Tabla de contenido

[Blockchain Programming: Guía e introducción para no morir en el intento 2](#_Toc117633572)

[Introducción, ¿Qué es la Blockchain? 2](#_Toc117633573)

[¿Como podemos desarrollar una Blockchain en Python? 4](#_Toc117633574)

[Capítulo 1. Bloques 5](#_Toc117633575)

[1.1 ¿Qué es un bloque? 5](#_Toc117633576)

[1.1 Concepto Hash 5](#_Toc117633577)

[1.2 ¿Qué información posee un bloque? 6](#_Toc117633578)

[Capítulo 2. Transacciones 7](#_Toc117633579)

[2.1 ¿Qué es una transacción? 7](#_Toc117633580)

[2.2 ¿Qué contiene una transacción? 7](#_Toc117633581)

[2.3 Proceso de Transacciones en una cadena de Bloques 8](#_Toc117633582)

[Capítulo 3. Usuarios. 9](#_Toc117633583)

[3.1 ¿Qué son los usuarios? 9](#_Toc117633584)

[3.2 Llaves públicas y privadas: Firmas digitales 9](#_Toc117633585)

[3.2.1 ¿Qué son? 9](#_Toc117633586)

[3.2.2 ¿Cuál es el fin de tener dos llaves? 10](#_Toc117633587)

[3.2.3 ¿Cómo se ven las llaves? ¿Cómo son? 10](#_Toc117633588)

[3.2.4 ¿Qué se puede hacer con cada llave? 10](#_Toc117633589)

[3.3 Firma digital 12](#_Toc117633590)

[3.4 ¿Cómo se relacionan con la Blockchain? 15](#_Toc117633591)

[Capítulo 4. Consensos 16](#_Toc117633592)

[4.1 Proof of Work 17](#_Toc117633593)

[4.1.1 ¿Qué significa Proof of Work? 17](#_Toc117633594)

[4.1.2 ¿Por qué existen los mineros? 18](#_Toc117633595)

[4.1.3 Muchos mineros, en muchas partes, y al mismo tiempo 18](#_Toc117633596)

[4.1.4 ¿Qué acertijos resuelven los mineros? 19](#_Toc117633597)

[4.2 Proof of Stake 22](#_Toc117633598)

[4.2.1 ¿Qué significa Proof of Stake? 22](#_Toc117633599)

[4.2.2 ¿Cómo funciona Proof of Stake? 22](#_Toc117633600)

[4.2.3 ¿Cuál es el trabajo de un forjador y del testigo? 23](#_Toc117633601)

[4.3 ¿Qué sigue? 24](#_Toc117633602)

[Capítulo 5. Implementación en Python usando Jupyter. 25](#_Toc117633603)

[5.0 Requerimientos 25](#_Toc117633604)

[5.1 Bloques 25](#_Toc117633605)

[5.2 Transacciones 25](#_Toc117633606)

[5.3 Proof of Work 25](#_Toc117633607)

[5.4 Proof of Stake 25](#_Toc117633608)

[5.5 Blockchain 25](#_Toc117633609)

[5.6 Ejemplo práctico 25](#_Toc117633610)

[Bibliografía 26](#_Toc117633611)

# Blockchain Programming: Guía e introducción para no morir en el intento

## Introducción, ¿Qué es la Blockchain?

Si la necesidad era suplir a los bancos, Blockchain[[1]](#footnote-1)[[2]](#footnote-2) entra como primer lugar.

La tecnología Blockchain no es más que una base de datos, distribuida a través de nodos en una red, que almacena cualquier tipo de información (transacciones, dinero, datos, etc.), y la guarda en **bloques**. Cada bloque contiene determinado número de transacciones e información, y al llegar al tamaño máximo del bloque (llenarse), se cierra y se vincula con el bloque anterior, creando uno nuevo. Por eso se llama “cadena de bloques”, debido a esa vinculación.

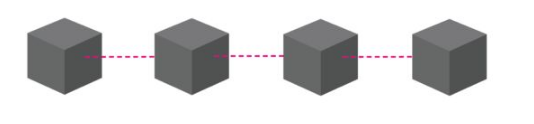
Toda la información nueva se compila en nuevos bloques, que luego se agregan a la cadena, y el proceso se repite con cada bloque.

Figura 0. 1. Ejemplo visual de una cadena de bloques. (Blockchain)

Blockchain tiene 4 características básicas:

* Descentralización
* Transparencia
* Inmutabilidad
* Escalabilidad

**Descentralización**

Blockchain no pertenece a ninguna persona, y cualquiera puede acceder a él. Una empresa esta centralizada cuando tiene su sede principal en un solo lugar. Digamos que una empresa telefónica tiene muchas sucursales alrededor de un país, pero cada una de ellas no puede subsistir si la sede central tiene una falla. Un ejemplo visual sería el siguiente:

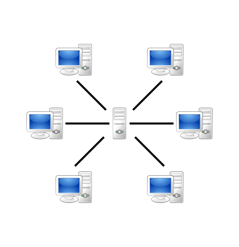
**

Figura 0. 2. Ejemplo de una red centralizada.

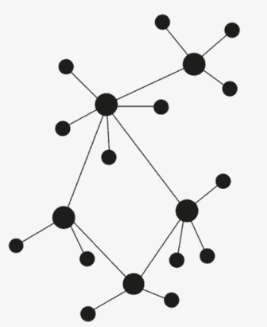
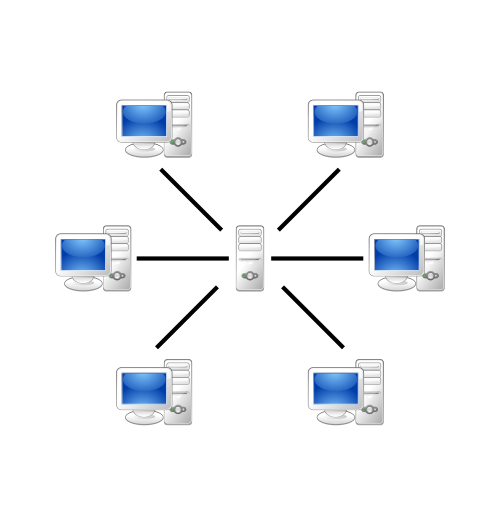
**La ventaja de una red **descentralizada** es que **no existe una sede central**. Todos los datos se distribuyen entre varios nodos de la red, donde ninguna entidad es una autoritaria.

Figura 0. 3. Ejemplo de una red descentralizada.

**Transparencia**

Las redes Blockchain son redes descentralizadas en donde su información está distribuida entre varios nodos de la red. Por lo tanto, cualquier nodo dentro de la red puede observar las transacciones que están sucediendo en la red. Cualquier transacción se puede rastrear, pero nunca manipular ni cambiar.

**Inmutabilidad**

Toda transacción o evento que se guarde en la Blockchain, se cifra y se encripta para darle seguridad en la red. Por lo tanto, nadie, ni el diseñador de la red misma, podría modificar el contenido de un bloque.

**Escalabilidad**

La escalabilidad se refiere a la capacidad de una cadena de bloques para manejar un mayor uso. Entre más popular se hace la cadena de bloques, más usuarios tendría que manejar en menos tiempo.

## ¿Como podemos desarrollar una Blockchain en Python?

Una Blockchain, viéndolo como si fuera un proyecto, es una plataforma en la que usuarios puede intercambiar información. La Blockchain es la que les ofrece un lugar para hacerlo de manera segura.

Para poder desarrollar una Blockchain necesitamos conocer el “que es”, “como funciona”, y “para que sirve” de cada uno de los siguientes conceptos en el contexto Blockchain:

* Bloque.
* Transacción.
* Usuarios.
* Consenso.

Vamos a explicar en este documento cada concepto, y en Jupyter[[3]](#footnote-3) veremos su aplicación en Python.

## Capítulo 1. Bloques

### 1.1 ¿Qué es un bloque?

La cadena de bloques es una cadena de bloques porque tiene… bloques. En esencia son lo más importante de la Blockchain, ya que, conceptualmente y filosóficamente, ellos son la Blockchain en sí.

Un **bloque[[4]](#footnote-4)** es un objeto en el cual se guarda información y se cifra, para después ser anexado a la cadena de bloques.

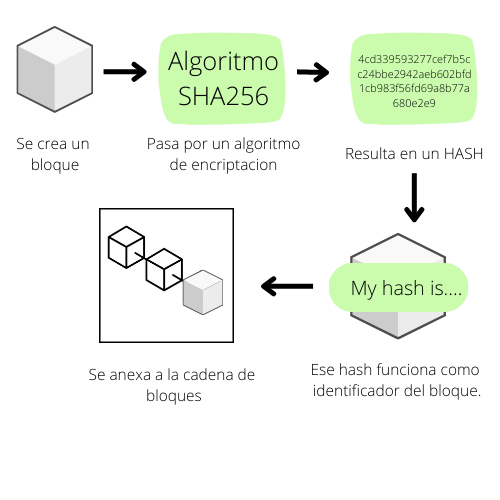
Un ejemplo visual sería el siguiente:

Figura 1. 1. 1. Ejemplo visual de cómo se trata un bloque en una cadena de bloques.

Para que el bloque pueda entrar a la cadena de bloques, si o si necesita pasar por un algoritmo de encriptación, ya que esto es lo que otorga seguridad e inmutabilidad.

### 1.1 Concepto Hash

Los códigos hash[[5]](#footnote-5) son creados por una función matemática que convierte la información digital en una cadena de números y letras. Los códigos son únicos. Por ejemplo:

|  |  |
| --- | --- |
| Mensaje | Mensaje encriptados con SHA256 |
| ‘Hola’ | e633f4fc79badea1dc5db970cf397c8248bac47cc3acf9915ba60b5d76b0e88f |
| ‘hola’ | b221d9dbb083a7f33428d7c2a3c3198ae925614d70210e28716ccaa7cd4ddb79 |
| ‘ola’ | 55a9f4f8994b1bbf2058ea38c8efb6c459000814d5f39c087002571639e6230e |

Tabla 2. 1. Ejemplo del uso del algoritmo SHA256.

Un ligero cambio en el contenido del mensaje, y el resultado será distinto.

### 1.2 ¿Qué información posee un bloque?

Los bloques generalmente incluyen estos elementos, pero pueden variar entre el objetivo del proyecto, por ejemplo:

* **Transacciones**: una lista de todas las transacciones almacenadas dentro de un bloque.
* **Altura del bloque:** Numero del bloque en toda la cadena de bloques actual.
* **Blocksize**: Establece el límite de tamaño en el bloque para que solo se pueda escribir una cantidad específica de información en él.
* **Encabezado del bloque**: Es una variable que contiene información que identifica a este bloque como único:
  + **Hash del bloque anterior**: Hash del bloque anterior en la cadena de bloques.
  + **Hash del bloque actual**: Hash del bloque actual en la cadena de bloques.
  + **Timestamp**: ‘Marca de tiempo’, viene a ser la hora en la que el bloque forma parte de la Blockchain.
  + **Difficulty:** Dificultad para resolver el problema en el consenso ‘Proof of Work’.
  + **Nonce:** Número encriptado que debe resolver un minero para resolver el problema en el consenso ‘Proof of Work’.

Los bloques contienen principalmente **transacciones**.

## Capítulo 2. Transacciones

### 2.1 ¿Qué es una transacción?

Si buscamos la definición, encontraríamos algo como lo siguiente:

“Trato o convenio por el cual dos partes llegan a un acuerdo comercial, generalmente de compraventa.”

De aquí, podemos obtener 3 elementos principales de la definición de arriba:

1. Dos personas/entidades queriendo intercambiar algo.
2. Ese algo.
3. El acuerdo.

Blockchain guarda la información como si la escribiera en piedra, una vez que se escribe nueva información, no se puede cambiar. Dentro de Blockchain, una transacción cambia un poco el concepto, pero mantiene el sentido:

**Transacciones**: “Operaciones que se realizan para agregar información a una Blockchain.”

Cada transacción contiene los mismos elementos que obtuvimos con anterioridad (Dos usuarios, “Algo” que intercambian y un acuerdo, que, en este caso, es mediado por la Blockchain). Por ello, hay que recordar que:

**Blockchain guarda información en bloques. Las transacciones añaden información a la Blockchain.[[6]](#footnote-6)**

Entonces, cada que se realiza una transacción, y se cumplan los 3 elementos principales, se anexara a un bloque para ser parte de la cadena de bloques.

Siendo más específicos…

### 2.2 ¿Qué contiene una transacción?

Contiene elementos como:

* Firma de la transacción: **Hash** que sirve como ID de la transacción.
* Bloque al que pertenece en la Blockchain.
* Timestamp: Hora en la que la transacción paso a ser parte de la Blockchain.
* Sender(from): Quien hace la transacción
* Receiver(to): Quien recibe el contenido de la transacción.
* Estatus: Estado en el que se encuentra la transacción. (Aceptada, en espera y rechazada.)

### 2.3 Proceso de Transacciones en una cadena de Bloques

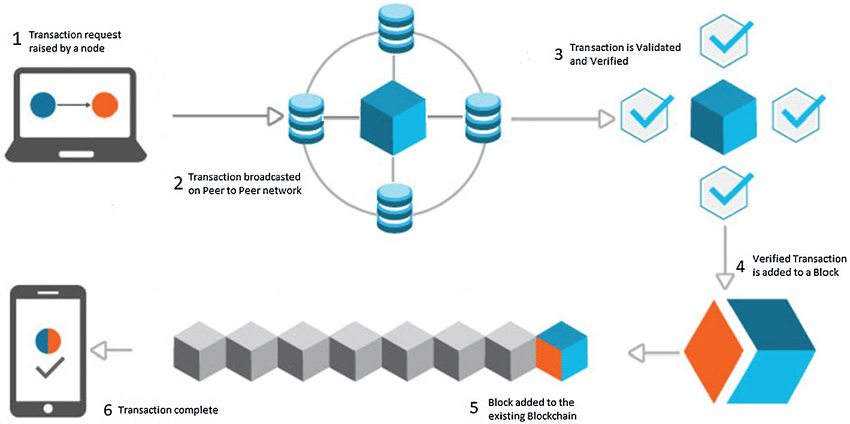


Figura 2. 3. 1. Ejemplo visual del proceso de una transacción en una cadena de bloques.

En la figura 2.3.1 podemos observar un ejemplo muy sencillo del proceso que conlleva una transacción.

1. Se crea una transacción en local.
2. La transacción llega a la red Blockchain con la que estemos trabajando.
3. La transacción se valida/verifica por los nodos de la red.
4. La transacción se añade al bloque más reciente (que aun tenga espacio para esta transacción)
5. El bloque entero se añade a la cadena de bloques.
6. La transacción se dice que esta verificada cuando existe dentro de la cadena de bloques.

Los que hacen transacciones dentro de la red son las cuentas, manejadas por usuarios.

## Capítulo 3. Usuarios.

### 3.1 ¿Qué son los usuarios?

Su definición es la siguiente:

“Persona que usa habitualmente un servicio.” En este caso, sería una persona que usa el servicio de la tecnología Blockchain, con algún propósito determinado (Intercambiar documentos, información o dinero).

La tecnología Blockchain termina siendo el **mediador** para que un usuario, pueda intercambiar con otros usuarios, elementos, regulado con **un acuerdo**, en el que todos los usuarios estén a favor.

Un usuario cuenta 2 características principales:

* Puede tener distintos activos y/o elementos. (Documentos, dinero o bienes)
* Puede recibir transacciones, y realizarlas con cualquier usuario dentro de la red.

Mas allá de eso, pueden interactuar con la red, participar en dinámicas, tener nicknames, etc.… En este trabajo vamos a enfocarnos solo en lo importante, las características mencionadas arriba y **su seguridad.**

Cada usuario va a contar con un sistema de seguridad para que su cuenta sea única, que solo él pueda acceder a ella, y solo él pueda realizar sus transacciones.

El tema que se presentara a continuación es de suma importancia para entender y comprender la relación que existe entre:

* Seguridad de la cuenta
* Seguridad de la transacción
* Validación y verificación de transacciones y bloques

### 3.2 Llaves públicas y privadas: Firmas digitales

#### 3.2.1 ¿Qué son?

Cada cuenta en la red va a tener una llave pública y una llave privada. Al sistema de llaves se le conoce como “Cifrado asimétrico[[7]](#footnote-7) de un solo camino.”

La criptografíaasimétrica es el método criptográfico, que asegura que un mensaje enviado no pueda ser leído por ninguna otra persona que la persona destinataria del mensaje.

En primer lugar, se crea un **par de llaves** (mejor conocido como *keypair*) mediante un proceso algorítmico se vinculan matemáticamente una con la otra. Gracias a su seguridad, costaría billones de billones de años solo hackear un mensaje.

#### 3.2.2 ¿Cuál es el fin de tener dos llaves?

Una llave pública es visible para cualquier persona y no pasa nada, la llave privada es de uso personal y **nadie debe de saberla más que el propietario**, es como una contraseña.

#### 3.2.3 ¿Cómo se ven las llaves? ¿Cómo son?

Hay varios algoritmos que se utiliza para el cifrado asimétrico, el más utilizado y el que usaremos en nuestro trabajo es el algoritmo RSA[[8]](#footnote-8).

El algoritmo crea dos llaves, puede variar en tamaño, pero en general una llave de 1024 bites es más que suficiente. Ejemplo con una llave de 512 (Para que se visualice en la pantalla):

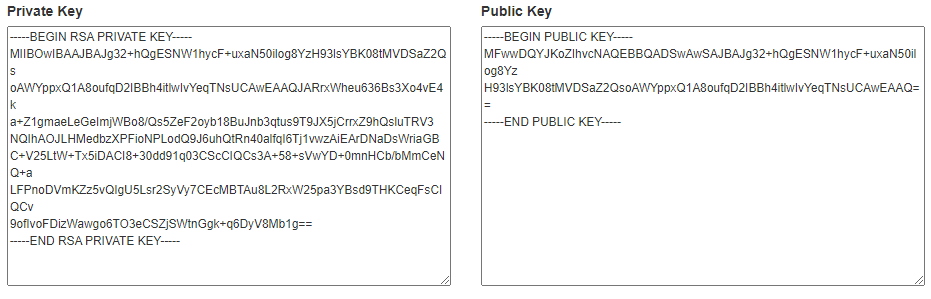


Figura 3.2.3.1 Ejemplo visual de dos llaves con el algoritmo RSA. [[9]](#footnote-9)

#### 3.2.4 ¿Qué se puede hacer con cada llave?

La **Llave pública** en la Blockchain es el ID de una cuenta. Cuando un usuario quiere transferirle activos a otro usuario en la red, va a necesitar la llave pública del destinatario. También sirve para verificar que un usuario ha hecho una transacción. (Las verificaciones las veremos más adelante)

Por otro lado, con la **llave privada** es como tener permisos de administrador de una cuenta. Al final, la llave privada es nuestro identificador, funge como una **contraseña** que **da acceso a la** cuenta y una **pluma** que nos **permite firmar transacciones**.

Figura 3.2.4.1. Infografía sobre el cifrado asimétrico.

Si pusiéramos el ejemplo de arriba en una Blockchain en la que se maneja dinero, de la figura 3.2.4.1 podemos obtener una estructura similar, veámoslo:

1. Pedro quiere realizar una transacción de 100MXN a María.
2. Pedro necesita la dirección (**Llave Pública**) de María para enviarle el dinero.
3. Para poder enviar la transacción a la red, Pedro **cifra el mensaje** con la llave pública de María y firma la transacción con su **Llave privada,** para después enviarla.
4. Al momento de firmar la transacción, se crea una **firma digital (HASH)** que sirve como identificador de la transacción.
5. En la red, un nodo en la Blockchainvalida si la **firma digital** concuerda con el remitente y el contenido del mensaje.
6. Una vez revisada la firma, se añade a un bloque. Hasta que la transacción forme parte de la cadena de Bloques, la transacción estará confirmada.
7. María, con su **llave privada**, y en su cuenta, **podrá ver el contenido del mensaje** y quien le envió la transacción.

Con nuestra llave privada podemos “*firmar*” transacciones, pero, ¿qué significa?, ¿o para que sirve?

### 3.3 Firma digital

Parte de lo que buscamos con utilizar un sistema de llaves es confidencialidad y **autenticidad**. Solo con nuestra llave privada podemos **autenticar** que somos nosotros los que queremos realizar la transacción, es la única forma en la que la computadora puede reconocernos como dueños de la cuenta.[[10]](#footnote-10)

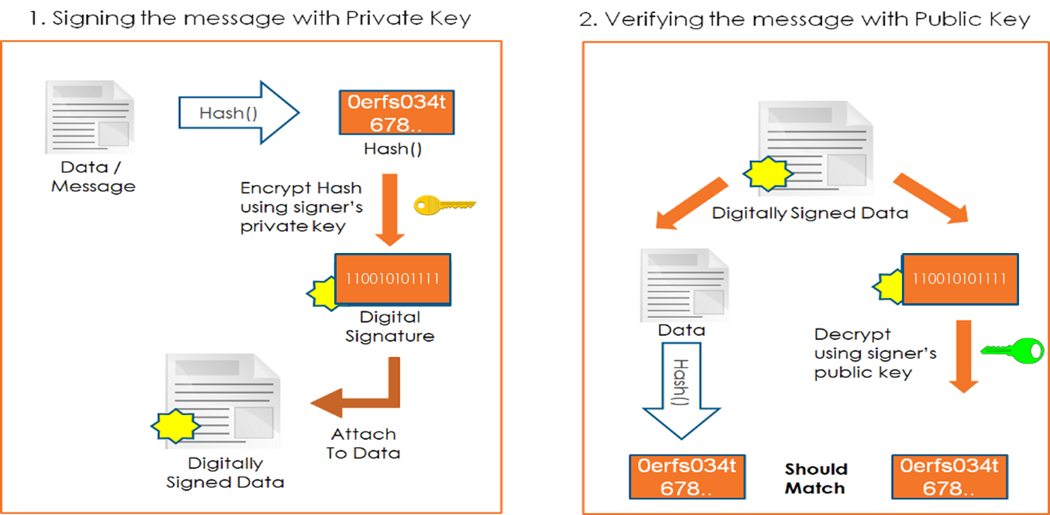
Veamos un ejemplo de cómo sería el proceso de firmado y verificación en una transacción.

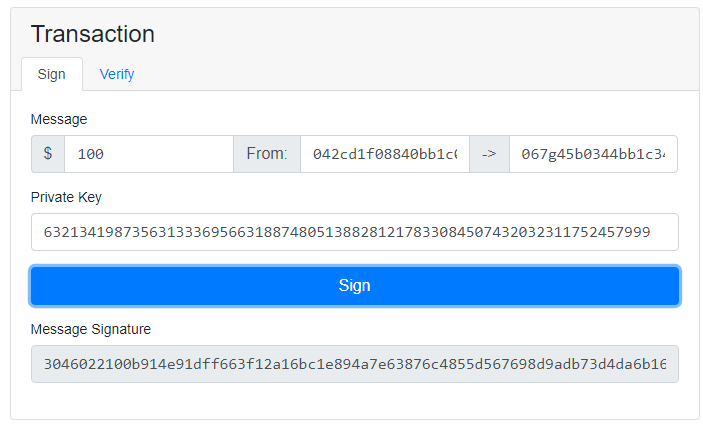
Figura 3.3.1. Ejemplo de una transacción firmada y verificada con el keypair.

1. El remitente crea una transacción (arriba seria Data).
2. La transacción pasa por un algoritmo HASH y resulta en un identificador único de la transacción.
3. Ese identificador se firma con la **llave privada**, lo que da como resultado una **firma digital**.
4. A la red se envía el contenido de la transacción y la firma resultante.
5. Después se procede a verificar que la firma de la transacción corresponda con el contenido de la misma. Esto se hace pasando comparando dos HASH.
   * El primer hash se obtiene pasando por el contenido de la transacción a un algoritmo HASH.
   * Y el segundo, **descifrando la firma digital** con la llave pública del remitente.
6. Si el resultado de los dos HASH es el mismo, la transacción esta verificada.

También podemos explorar este ejemplo para visualizarlo mejor:

Este ejemplo[[11]](#footnote-11) es una transacción entre dos llaves públicas; al darle click en **Sign** se crea una **firma digital** de la transacción **(**Message Signature). Hay que recordar que a la red se envía el contenido (que es toda la información de la transacción) y la firma digital.

Figura 3.3.2. Ejemplo de una firma con llave privada.



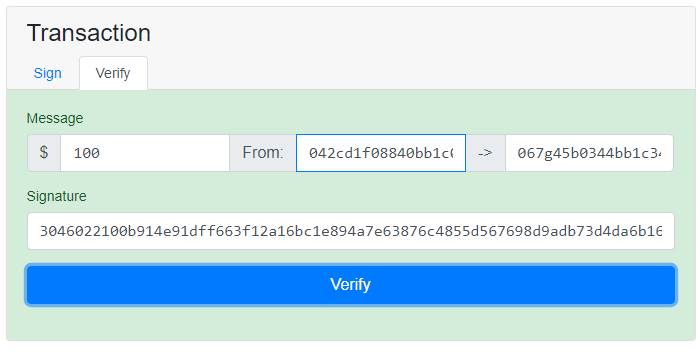
En la figura 3.3.3. Verificamos esa firma. Si el contenido de la transacción no está alterado, y la firma es la correcta. En la comparación dará como correcto.

Figura 3.3.3. Ejemplo de una verificación correcta de una firma digital.

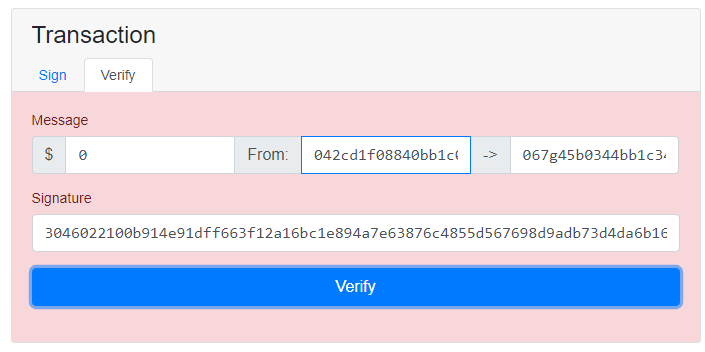
En los siguientes casos se presentan errores:

Figura 3.2.4. Error: El dinero que se envía no es el correcto.

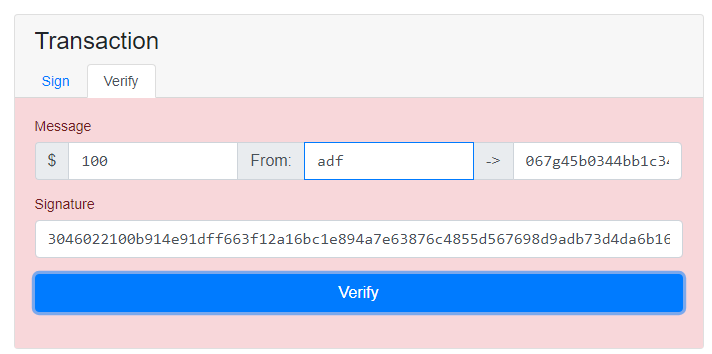
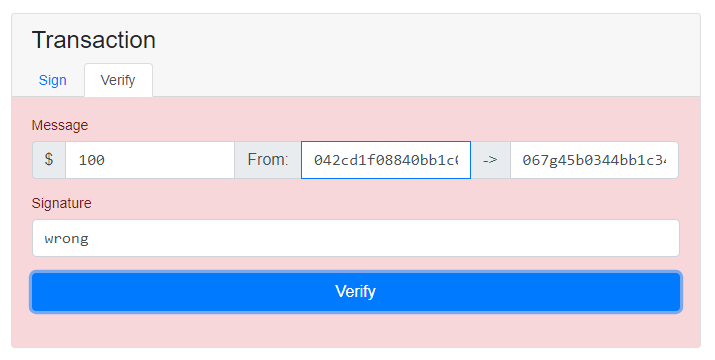


Figura 3.2.6. Error: La firma no concuerda con la transacción.

Figura 3.2.5. Error: El remitente no es quien firmo la transacción.

### 3.4 ¿Cómo se relacionan con la Blockchain?

Todo lo que podría suceder en una Blockchain son:

“Llaves públicas enviándose bienes a través de transacciones. Las transacciones se firman y se verifican. Las transacciones cambian la red. Mientras la transacción sea parte de la cadena de bloques, esta verificada. Todos contentos.”

Gracias a este mecanismo de seguridad, la Blockchain provee seguridad. Ahora, vamos con el siguiente tema, Consensos.

## Capítulo 4. Consensos

Ya vimos la relación que hay entre:

Un usuario hace una transacción -> la transacción se verifica -> se anexa a un bloque -> el bloque se anexa a la Blockchain -> Se confirma la transacción -> Todos felices

En este tema le vamos a dar más importancia a “el bloque se anexa a la Blockchain”, porque, ¿cómo lo hace? ¿Así nomás?

Las Blockchain necesitan nodos para existir, gracias a ellos, la información se distribuye y aumenta la seguridad. Los nodos son las encargadas también de anexar los bloques a la cadena original, textualmente, llegan a un consenso, y ese bloque pasa a ser el bloque más alto de la cadena:

La definición de consenso[[12]](#footnote-12), sería algo así:

**Por la RAE:**

“Acuerdo producido por consentimiento entre todos los miembros de un grupo, o entre varios grupos.” [[13]](#footnote-13)

**Por Ethereum**

“*Consensus mechanisms (also known as consensus protocols or consensus algorithms) allow distributed systems (networks of computers) to work together and stay secure.*

*Consensus protocols and consensus algorithms are often used interchangeably. However, protocols and algorithms are different. A protocol is a set of rules defined in a standard that governs how a system and its many functioning parts operate and interact. Algorithms are like exact recipes on how to solve a problem or calculate a result.”[[14]](#footnote-14)*

Aquí lo interesante, es que dependiendo del consenso que tengamos en nuestra Blockchain, va a ser la manera el que los bloques se anexen a la red. Para este tema, es importante dirigir la atención únicamente al proceso en sí.

Hablaremos sobre 2 consensos:

Proof of Work y Proof of Stake.[[15]](#footnote-15)

### 4.1 Proof of Work

#### 4.1.1 ¿Qué significa Proof of Work?

"Proof of Work”, traducido a **Prueba de Trabajo,** es el mecanismo que utiliza Bitcoin para anexar los bloques a su red. Es el mecanismo de consenso criptográfico original, así que vamos a explicarlo en el contexto de Bitcoin. [[16]](#footnote-16)

Recordemos que Bitcoin es una cadena de bloques en la que los usuarios poseen **dinero digital** en forma de **monedas digitales** llamadas **criptomonedas[[17]](#footnote-17)**.

En redes Blockchain hay varios tipos de nodos[[18]](#footnote-18). Hay nodos que almacenan toda la Blockchain y trabajan en hacerla más segura, y hay otros más pequeños encargados de otras tareas.

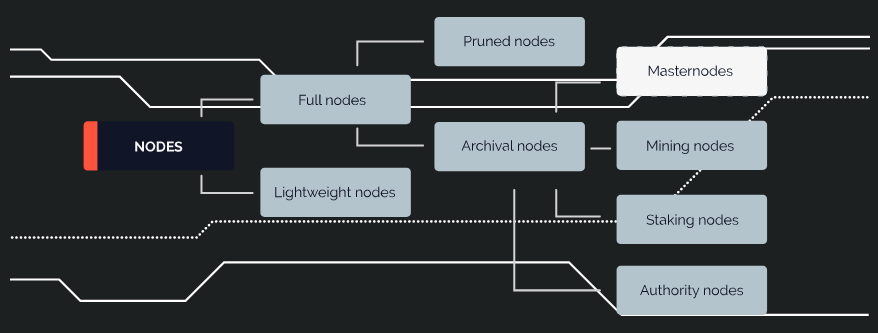
En este trabajo vamos a prestarle más atención a un tipo de nodo llamado **minero.**

Figura 4.1. 1. Imagen Ilustrativa de los nodos utilizados en una red Blockchain.

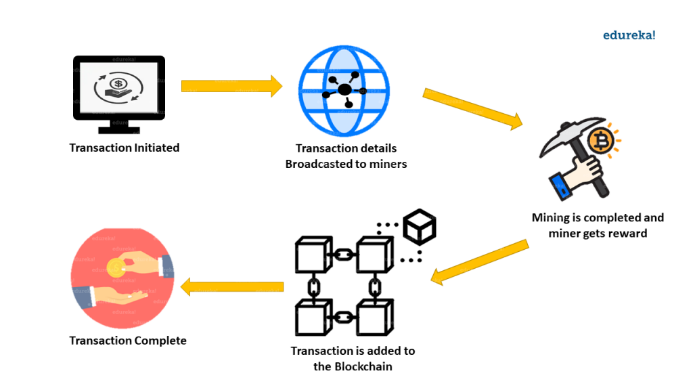
La minería es el proceso de crear un bloque, que contiene transacciones, para agregarlo a la cadena de bloques.

Figura 4.1. 2. Ejemplo visual de la minería.

La palabra minería se utiliza como una analogía hacia el oro. El oro es escaso, al igual que las monedas digitales, y la única forma de aumentar el volumen total es a través de la minería. A medida que el volumen de monedas va creciendo, la minería también es la forma de proteger la red al crear, verificar, publicar y propagar bloques en la cadena de bloques.

#### 4.1.2 ¿Por qué existen los mineros?

La minería es el elemento vital de la prueba de trabajo. Los mineros (computadoras que ejecutan software) usan su tiempo y poder de cómputo para procesar transacciones y producir bloques.

En sistemas descentralizados, debemos asegurarnos de que todos estén de acuerdo con las transacciones que suceden en la red. Los mineros ayudan a que esto suceda resolviendo acertijos computacionalmente difíciles para producir nuevos bloques.

Pero, si hay muchos mineros, ¿no sería una problemática? ¿Como se aborda ese problema?

También, hay mineros que trabajan resolviendo problemas para la red, pero ¿cuáles son los problemas que hay que resolver?

#### 4.1.3 Muchos mineros, en muchas partes, y al mismo tiempo

La razón por la que se llama "prueba de trabajo" es porque la red requiere una gran cantidad de potencia de procesamiento. Las cadenas de bloques de prueba de trabajo están aseguradas y verificadas por mineros virtuales de todo el mundo que compiten para ser los primeros en resolver un acertijo matemático.

Ethereum (otra Blockchain que en su momento utilizo Proof of Work) llego a tener más de 1 millón de transacciones por día. Imagínense el poder computacional que necesita la red para resolver 1 millón de acertijos matemáticos en la red al día.

Los mineros no sobran. Gracias a ellos, hay más nodos asegurando la red. El asunto aquí es que no trabajan en equipo, compiten entre ellos. Por lo tanto, el gasto energético en mucho mayor.

El ganador (quien resuelve el acertijo primero) puede actualizar la cadena de bloques con las últimas transacciones verificadas y la red lo recompensa con una cantidad predeterminada de dinero virtual.

#### 4.1.4 ¿Qué acertijos resuelven los mineros?

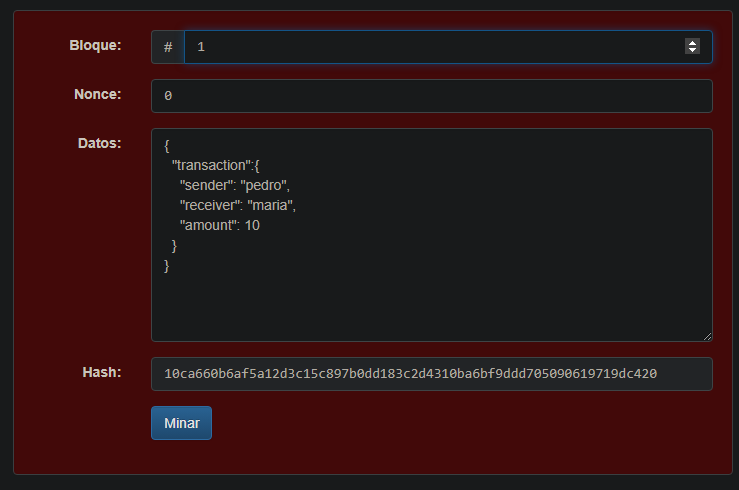
Los mineros buscan encontrar un “número mágico” llamado **Nonce**. El nonce significa “número que solo se usa una vez.”

Figura 4.1.4.1. Hash de una transacción.

En la figura 4.1.4.1 tenemos una imagen visual de lo que seria el resultado de pasar un bloque, con una transaccion, por un algoritmo Hash. El resultado es una cadena de longitud fija hexadecimal. Todo lo que es editable en la caja, pasa por el algoritmo hash (El bloque, el nonce y los datos).

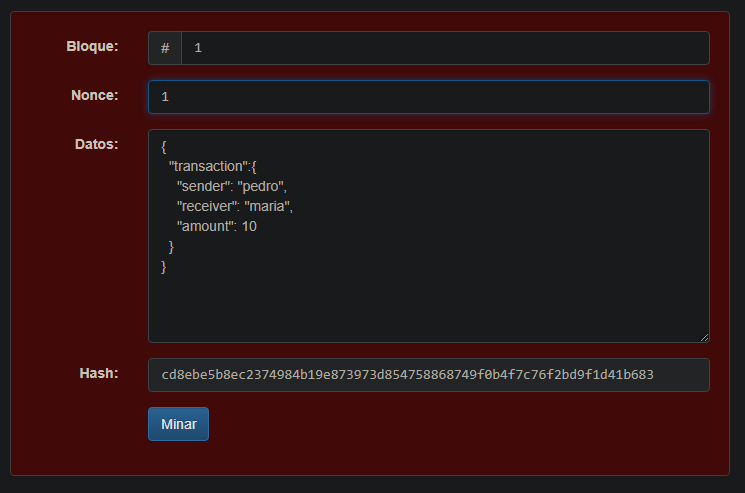
Si cualquiera de los datos cambia, el hash sera totalmente distinto.

Figura 4.1.4.2. Hash de una transacción.

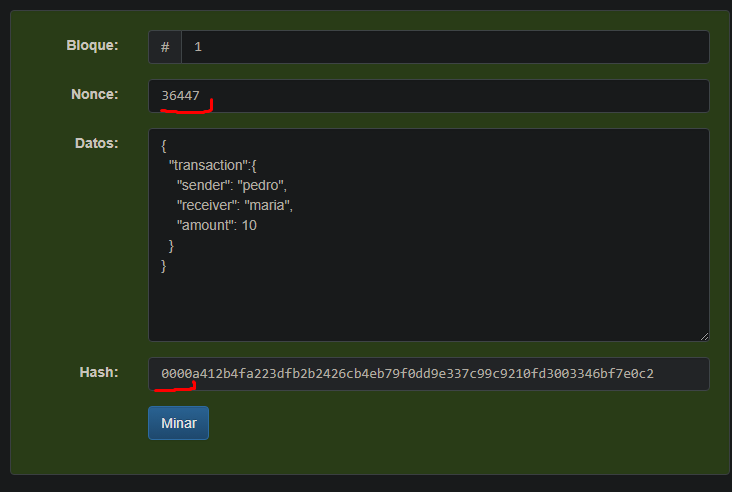
El trabajo de los mineros va a ser ir cambiando el “nonce” para encontrar con un número que resulte en un hash que venza una “dificultad” o un “target”.

Figura 4.1.4.3. Nonce encontrado. Target alcanzado.

En la figura 4.1.4.3 se encontró que el nonce 36447, dio como resultado un hash con sus primeros 4 dígitos en 0.

La **dificultad** es un valor que el hash tiene que igual o superar para considerarse valido. Recordemos que un hash tiene un valor hexadecimal. Entre más bajo sea un valor, más difícil va a ser encontrado. [[19]](#footnote-19)

No es lo mismo encontrar esto:

B5372154B53BF87D8E57B2329587D703981ED587AF0B5EB8418665FC7E2D4E76

A esto:

00000054B53BF87D8E57B2329587D703981ED587AF0B5EB8418665FC7E2D4E76

La dificultad indica que tan bajo tiene que resultar el hash para que sea considerado valido.

* Si quisiéramos un hash con dos ceros, el valor decimal tendría que ser menor a:
  + 452312848583266388373324160190187140051835877600158453279131187530910662655
* Si quisiéramos un hash con 16 ceros, el valor decimal tendría que ser menor a:
  + 6277101735386680763835789423207666416102355444464034512895

Se nota que hay una diferencia numérica evidente.

En redes como Bitcoin, la dificultad ajusta tomando en cuenta el número de mineros y su poder computacional en la red, para que siempre dura 10 minutos minar un bloque.

Dificultad es (Complejidad de la red) **una medida relativa sobre lo difícil que es minar un bloque nuevo en la cadena.[[20]](#footnote-20)**

### 4.2 Proof of Stake

#### 4.2.1 ¿Qué significa Proof of Stake?

Textualmente, “Proof of Stake”[[21]](#footnote-21) significa Prueba de Participación.[[22]](#footnote-22)

En Proof of Work, los mineros prueban que tienen capital en riesgo gastando energía resolviendo acertijos complejos. En Proof of Stake, los validadores apuestan explícitamente su capital (Stake) en forma de dinero virtual en la red. Este dinero apostado actúa como garantía que puede destruirse si el validador se comporta de manera deshonesta o perezosa.

El Staking es el acto de almacenar su dinero virtual en la red para ganar derechos de voto o recibir recompensas por parte de la Blockchain.

El validador es responsable de verificar que los nuevos bloques propagados a través de la red sean válidos y, ocasionalmente, de crear y propagar nuevos bloques por sí mismo.

Proof of Stake es un popular mecanismo de consenso alternativo a Proof of Work. En lugar de necesitar potencia informática para validar transacciones, los validadores deben hacer Staking de monedas. Este hecho reduce drásticamente el consumo de energía necesario.

#### **4.2.2 ¿Cómo funciona Proof of Stake?**

El algoritmo Proof of Stake cuenta con validadores en vez de mineros. Su responsabilidad es validar las transacciones de la red y anexarlas en bloques.

Para ser un validador, un nodo necesitar tener una determinada cantidad de dinero virtual almacenado.

Aquí, los bloques se "forjan" en lugar de ser minados. Los responsables de “forjar” (anexar) el bloque a la red son los Forjadores, y aquellos que atestiguan el trabajo del forjador, son llamados Testigos. Se forman grupos con un forjador y varios testigos, juntos, son los encargados de anexar bloque con transacciones verificadas en la red.

Los validadores participan en el proceso de elección en un algoritmo esperando a ser seleccionados como Forjadores o Testigos. Para seleccionar a cada uno de ellos, el sistema utiliza una combinación de factores como la antigüedad del Staking, un elemento de aleatoriedad y la riqueza del nodo.

El tamaño del Stake determina la probabilidad de que un nodo sea seleccionado como siguiente validador: cuanto mayor sea el Stake, mayor la probabilidad. Al proceso de selección se agregan métodos únicos para evitar favorecer únicamente a los nodos más ricos de la red. Los dos métodos más utilizados habitualmente son Randomized Block Selection (selección aleatoria de bloques) y Coin Age Selection (selección de monedas por antigüedad), aunque no los vamos a tocar en este tema[[23]](#footnote-23).

#### 4.2.3 ¿Cuál es el trabajo de un forjador y del testigo?

El forjador es el encargado de recibir las transacciones en espera, verificar cada una de ellas, y después agruparlas en un bloque. Cuando el bloque está listo, lo firma con su llave privada (Como si fuera una transacción) y lo envía al grupo de los testigos.

Los testigos están encargados de revisar que el bloque que ha creado el forjador este correcto y no sea un fraudulento. Si la mayoría está de acuerdo, el forjador puede anexarlo a la cadena de bloques.

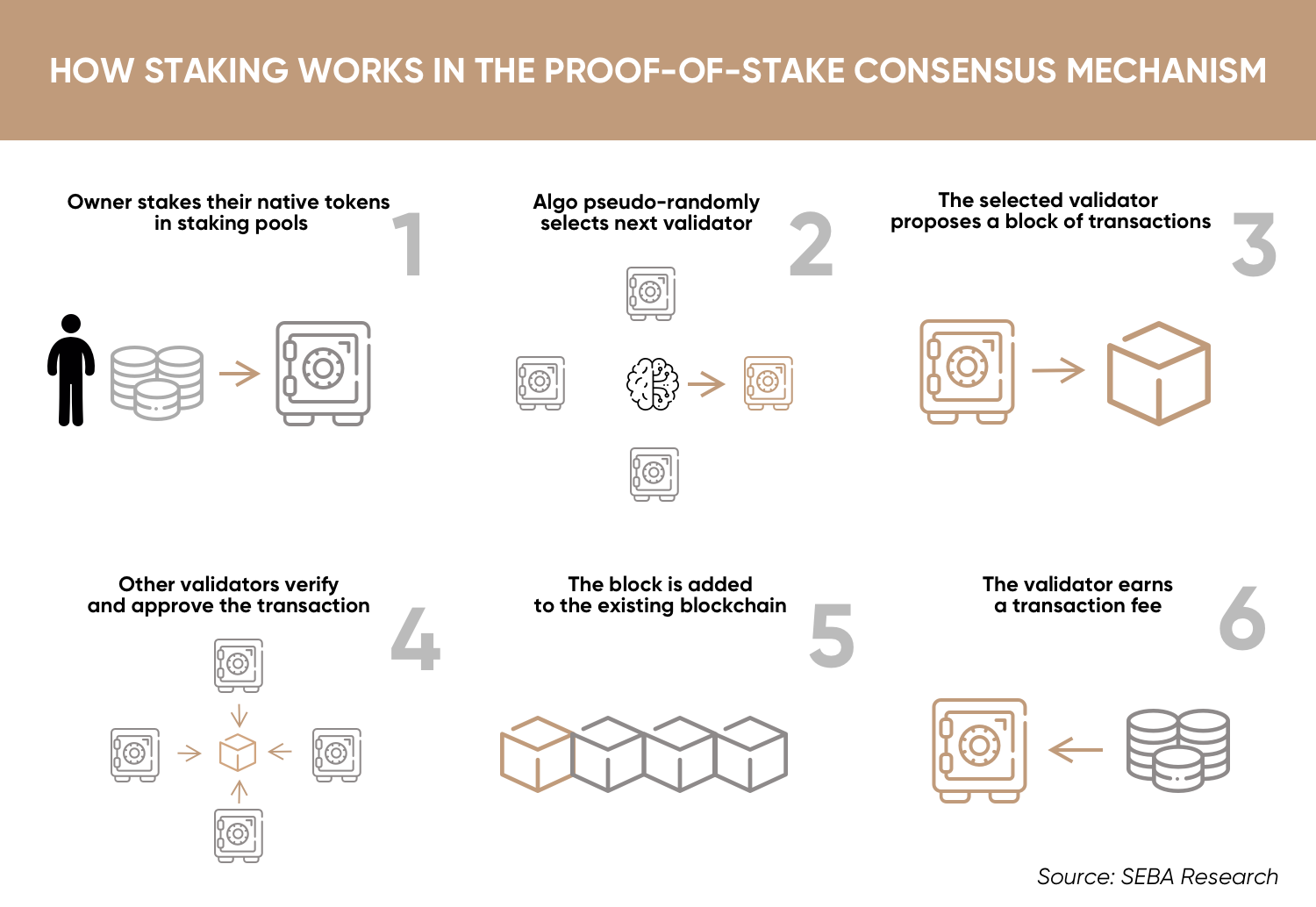


Figura 4.2.3.1. Cómo funciona el mecanismo de stacking en Proof of Stake.

### 4.3 ¿Qué sigue?

Cada mecanismo de consenso tiene sus ventajas y desventajas. Aprenderemos paso por paso a llevar a la implementación de cada una de ellas junto con los temas anteriores. Para ello, exploraremos los siguientes capítulos en Jupyter para tener una idea visual más clara y ver cómo se implementa.

Los capítulos anteriores nos ayudan a entender los conceptos importantes antes de ponerlos en práctica.

En este capítulo vamos a mostrar una implementación completa de una Blockchain, tomando en cuenta cuentas, transacciones, consensos y todo lo visto con anterioridad.

## Capítulo 5. Implementación en Python usando Jupyter.

### 5.0 Requerimientos

### 5.1 Bloques

### 5.2 Transacciones

### 5.3 Proof of Work

### 5.4 Proof of Stake

### 5.5 Blockchain

### 5.6 Ejemplo práctico

# Bibliografía

admin. (1 de Octubre de 2018). *Hash y sus ejemplos.* Obtenido de Altaruru: https://www.altaruru.com/hash-y-ejemplos-de-uso/

Binance. (20 de Octubre de 2022). *¿Qué es Proof of Stake (PoS)?* Obtenido de Binance: https://academy.binance.com/es/articles/proof-of-stake-explained

bit2me Academy. (s.f.). *¿Qué es bloque dentro de la blockchain?* Obtenido de Academy bit2me: https://academy.bit2me.com/que-es-un-bloque-dentro-de-la-blockchain/

bit2me Academy. (s.f.). *¿Qué es Prueba de participación / Proof of Stake (PoS)?* Obtenido de Academy bit2me: https://academy.bit2me.com/que-es-proof-of-stake-pos/

bitpanda. (s.f.). *¿Qué significa la dificultad de minado?* Obtenido de bitpanda: https://www.bitpanda.com/academy/es/lecciones/que-significa-la-dificultad-de-minado/

BLOCKCHAIN FEDERAL ARGENTINA. (s.f.). *Protocolos de consenso*. Obtenido de BFA: https://bfa.ar/blockchain/protocolos-de-consenso#:~:text=El%20Protocolo%20de%20Consenso%20es,ese%20bloque%20a%20la%20cadena

blockchain.com. (20 de 10 de 2022). *Network Difficulty*. Obtenido de https://www.blockchain.com/: https://www.blockchain.com/explorer/charts/difficulty

Coinbase. (s.f.). *What is "proof of work" or "proof of stake"?* Obtenido de coinbase: https://www.coinbase.com/learn/crypto-basics/what-is-proof-of-work-or-proof-of-stake

*CONSENSUS MECHANISMS*. (s.f.). Recuperado el 19 de 10 de 2022, de ethreum: https://ethereum.org/en/developers/docs/consensus-mechanisms/

Encryption Consulting. (s.f.). *What is RSA? How does an RSA work?* Obtenido de Encryptionconsulting: https://www.encryptionconsulting.com/education-center/what-is-rsa/

Evans, J. (s.f.). *Blockchain Nodes: An In-Depth Guide*. Obtenido de nodes: https://nodes.com/

Frankenfield, J. (02 de 05 de 2022). *What Is Proof of Work (PoW) in Blockchain?* Obtenido de https://www.investopedia.com/: https://www.investopedia.com/terms/p/proof-work.asp

*Guía para saber qué son las criptomonedas*. (19 de 10 de 2022). Obtenido de Santander: https://www.santander.com/es/stories/guia-para-saber-que-son-las-criptomonedas

Hayes, A. (27 de Septiembre de 2022). *Blockchain Facts: What It Is, How It Works, and How Can Be Used*. Obtenido de Investopedia: https://www.investopedia.com/terms/b/blockchain.asp

https://jupyter.org/. (s.f.).

IBM. (s.f.). *What is blockchain technology?* Obtenido de IBM: https://www.ibm.com/topics/what-is-blockchain

Porter, B. (20 de Abril de 2020). *What is Asymmetric Encryption?* Obtenido de Freedomben medium: https://freedomben.medium.com/what-is-asymmetric-encryption-64c74b2a0a82

*Proof-of-Work*. (s.f.). Recuperado el 19 de 10 de 2022, de ethereum: https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/

Ray, S. (6 de Octubre de 2017). *What is proof of stake?* Obtenido de hacekrmoon: https://hackernoon.com/what-is-proof-of-stake-8e0433018256

Real Academia Española. (s.f.). *Consenso*. Obtenido de Real Academia Española: https://www.rae.es/dpd/consenso

Soto, L. (18 de Agosto de 2021). *¿Qué es una firma digital?* Obtenido de Signaturit Blog: https://blog.signaturit.com/es/que-es-una-firma-digital

*Transactions*. (s.f.). Recuperado el 19 de 10 de 2022, de ethereum: https://ethereum.org/en/developers/docs/transactions/#:~:text=An%20Ethereum%20transaction%20refers%20to,takes%20place%20within%20a%20transaction.

1. Concepto de Blockchain por IBM: (IBM, n.d.) [↑](#footnote-ref-1)
2. Concepto de Blockchain por Investopedia. (Hayes, 2022) [↑](#footnote-ref-2)
3. Jupyter es una notebook virtual que nos permite crear un entorno de desarrollo en la red o local. Es muy útil para hacer pruebas sobre la marcha. Para más información, pueden visitar su sitio web. (<https://jupyter.org/>) [↑](#footnote-ref-3)
4. Definición más extensa del concepto de Bloque por bit2academy. (bit2me Academy, n.d.) [↑](#footnote-ref-4)
5. Definición y ejemplos de un algoritmo HASH. (admin, 2018) [↑](#footnote-ref-5)
6. Concepto de Transacción explicado por Ethereum. (Transactions, n.d.) [↑](#footnote-ref-6)
7. Articulo más extenso sobre la explicación del cifrado asimétrico. (Porter, 2020) [↑](#footnote-ref-7)
8. Artículo que explica más a fondo el algoritmo RSA. (Encryption Consulting, s.f.) [↑](#footnote-ref-8)
9. Página web que permite hacer el ejemplo visual del algoritmo SHA256. ([https://travistidwell.com/jsencrypt/demo](https://travistidwell.com/jsencrypt/demo/)) [↑](#footnote-ref-9)
10. Artículo que explica las firmas digitales. (Soto, 2021) [↑](#footnote-ref-10)
11. Ejemplo realizado en el siguiente sitio web:

    (<https://andersbrownworth.com/blockchain/public-private-keys/signatures>) [↑](#footnote-ref-11)
12. (BLOCKCHAIN FEDERAL ARGENTINA, s.f.) [↑](#footnote-ref-12)
13. (Real Academia Española, s.f.) [↑](#footnote-ref-13)
14. (CONSENSUS MECHANISMS, s.f.) [↑](#footnote-ref-14)
15. (Coinbase, s.f.) [↑](#footnote-ref-15)
16. (Proof-of-Work, s.f.) [↑](#footnote-ref-16)
17. (Guía para saber qué son las criptomonedas, 2022) [↑](#footnote-ref-17)
18. Página web dedicada a explicar qué son los nodos. (Evans, s.f.) [↑](#footnote-ref-18)
19. Página web que permite hacer el ejemplo visual.

    (<https://andersbrownworth.com/blockchain/block>) [↑](#footnote-ref-19)
20. Dificultad explicada a más detalle. (bitpanda, n.d.) [↑](#footnote-ref-20)
21. Concepto de PoS por Binance. (Binance, 2022) [↑](#footnote-ref-21)
22. Concepto de PoS por bit2academy. (bit2me Academy, n.d.) [↑](#footnote-ref-22)
23. Articulo que menciona los algoritmos Coin Age y Randomized Block Selection (Ray, 2017) [↑](#footnote-ref-23)