

TALLER BLOCKCHAIN

TECNOLOGÍA HYPERLEDGER

Dr. Iván S. Razo-Zapata

Dr. Alberto F. Martínez

Día 1, 24 de Febrero, 2021

AGENDA

- Presentación
- Conceptos básicos (técnicos) de blockchain
- Descripción del ecosistema de desarrollo
- Requerimientos básicos

AGENDA

- Presentación

AGENDA. PRESENTACIÓN.

El taller se divide en 6 sesiones teórico-prácticas, donde los asistentes al mismo:

- Estudiarán a grandes rasgos conceptos de criptografía y su relación con Blockchain
- Revisarán las bases técnicas de Blockchain
- Comprenderán la estructura de Blockchains "populares"
- Verán escenarios de aplicación de Blockchain orientados al sector energético
- Diseñarán e implementarán una Blockchain ... enfocada al sector energético

AGENDA. PRESENTACIÓN.

Instructores y sesiones:

- Dr. Alberto F. Martínez. Estará liderando las sesiones 1*, 3, 5.
- Dr. Iván S. Razo Zapata. Estará liderando las sesiones 2, 4, 6.

AGENDA. PRESENTACIÓN.

Objetivos generales del taller:

- Impulsar la innovación en el sector energético en Querétaro
- Promover el uso de Nuevas Tecnologías Digitales para catalizar dicha innovación
- Exponer algunas de las tendencias a nivel global en cuanto a la digitalización y el uso de blockchain en el sector energético
- Contribuir a la formación de capacidades locales

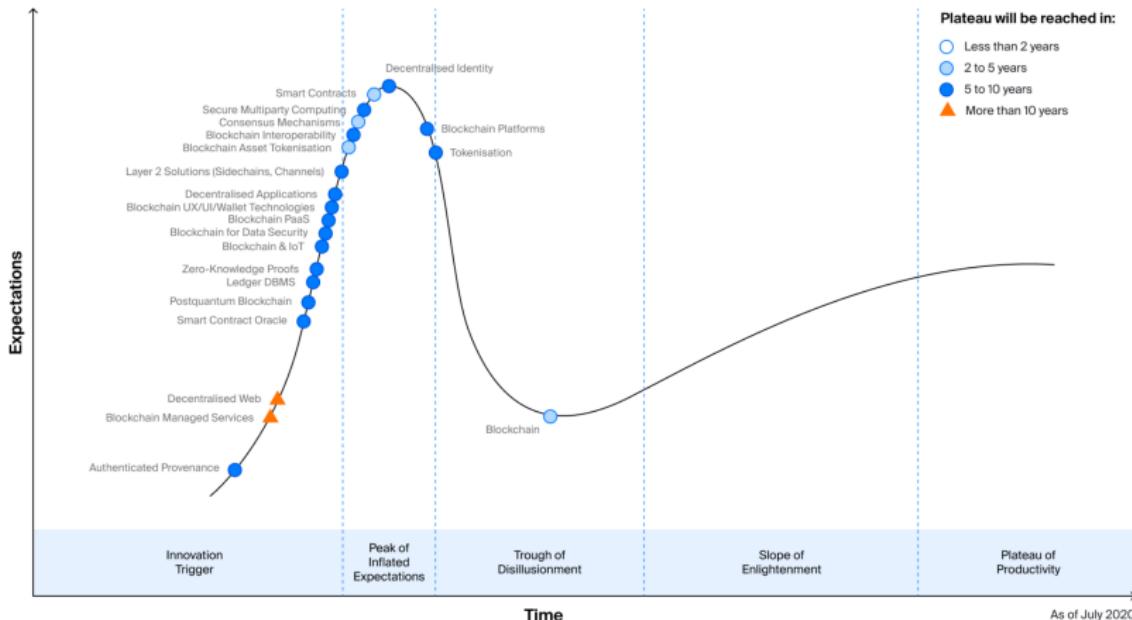
AGENDA. PRESENTACIÓN.

Sobre los asistentes:

- Afiliación
- Intereses en el sector energético
- Expectativas del taller

BIG PICTURE

Hype Cycle for Blockchain Technologies, 2020



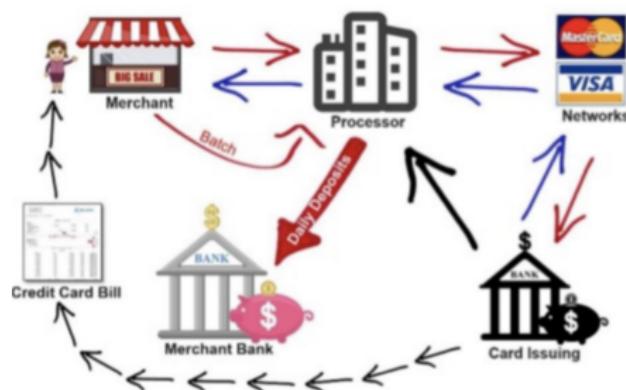
¿QUÉ ES UNA BLOCKCHAIN / CADENA DE BLOQUES?

- Un registro distribuido que solo permite agregar información
- No es una tecnología nueva ... más bien es una combinación ingeniosa de otras tecnologías
 - Sistemas distribuidos
 - Redes entre pares (P2P)
 - Hashing
 - Criptografía

¿CUÁL ES LA IDEOLOGÍA DETRAS DE LA TECNOLOGÍA?

Los intermediarios tiene demasiado poder

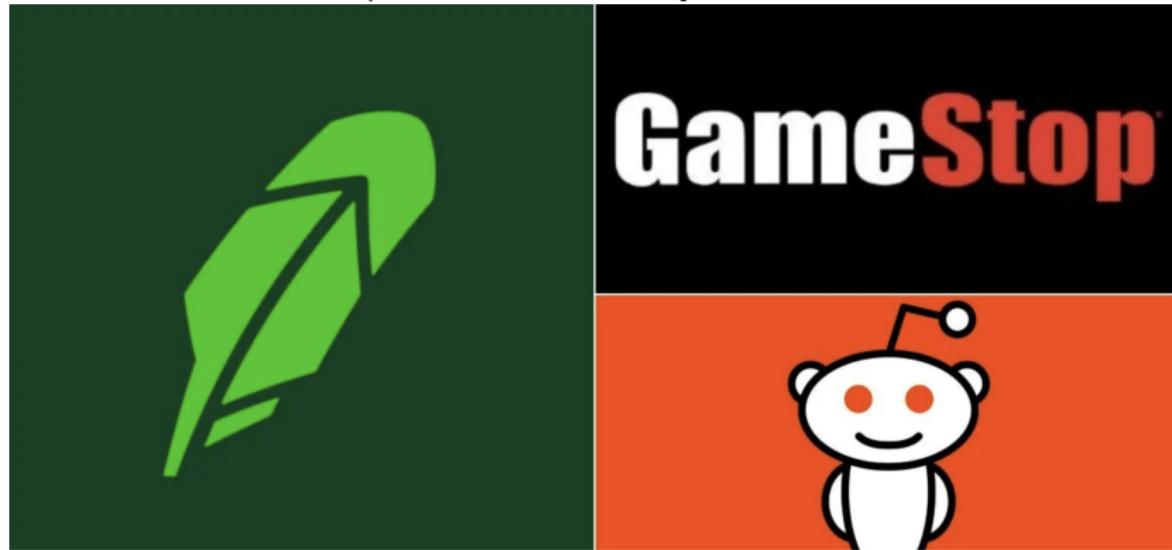
Imponen comisiones o censuran



¿CUÁL ES LA IDEOLOGÍA DETRAS DE LA TECNOLOGÍA?

Los intermediarios tiene demasiado poder

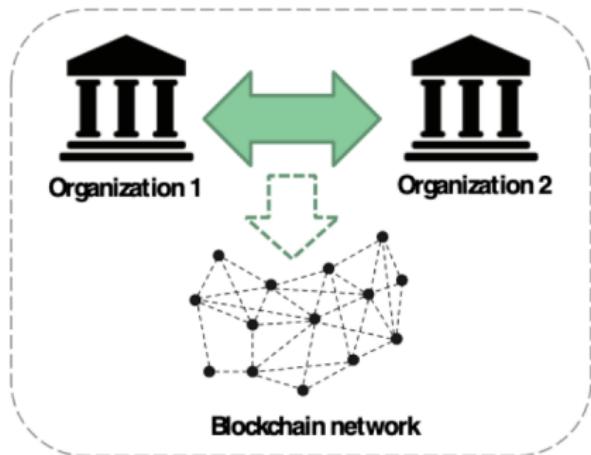
Imponen comisiones y censuran



INNOVACIÓN INSTITUCIONAL



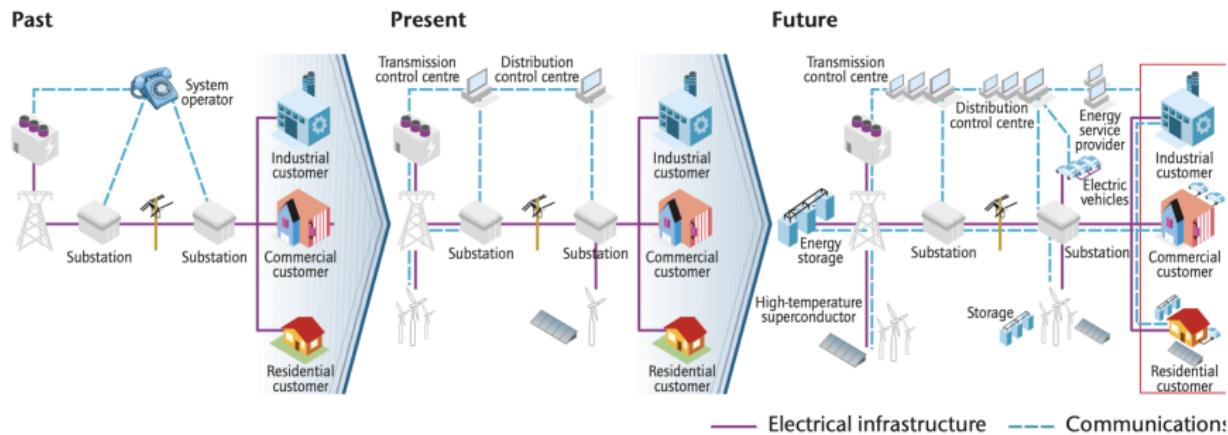
Superuser



Crowdsourced

INNOVACIÓN INSTITUCIONAL

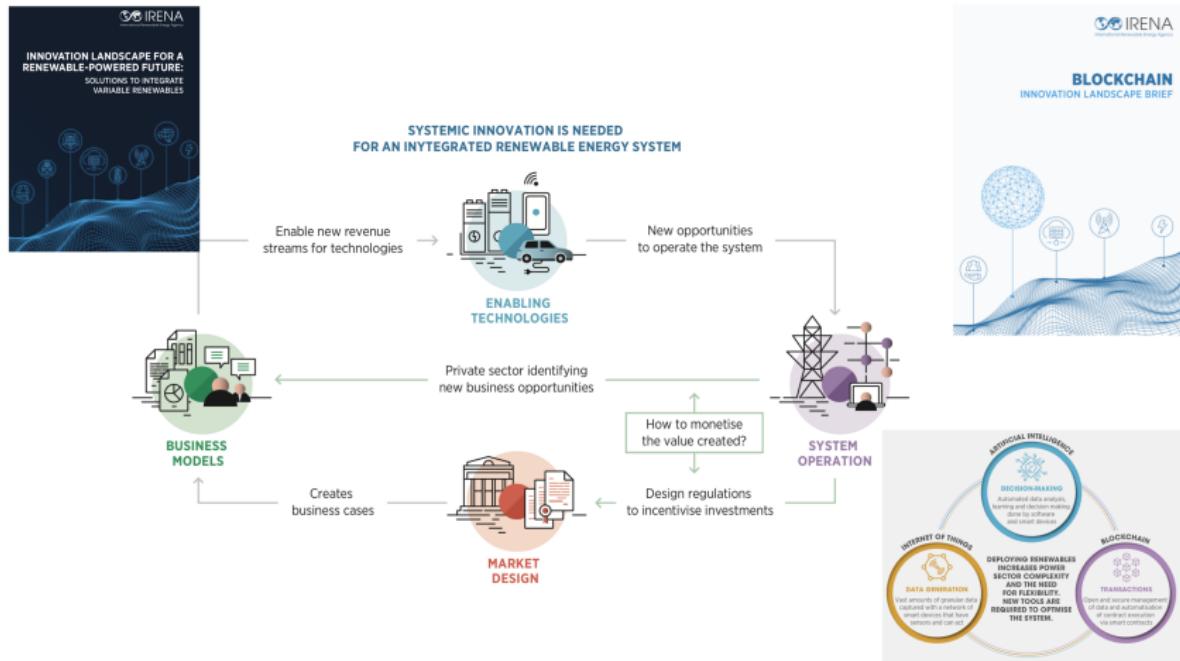
Impulsar la innovación en el sector energético



Source: IEA (2011), *Technology Roadmap: Smart Grids*, OECD/IEA, Paris.

INNOVACIÓN INSTITUCIONAL

Impulsar la innovación en el sector energético



AGENDA

- Conceptos básicos (técnicos) de blockchain



CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Distributed Systems. Un sistema distribuido es aquel sistema cuyos componentes están repartido entre varios entes (servidores), los cuales se comunican entre sí.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

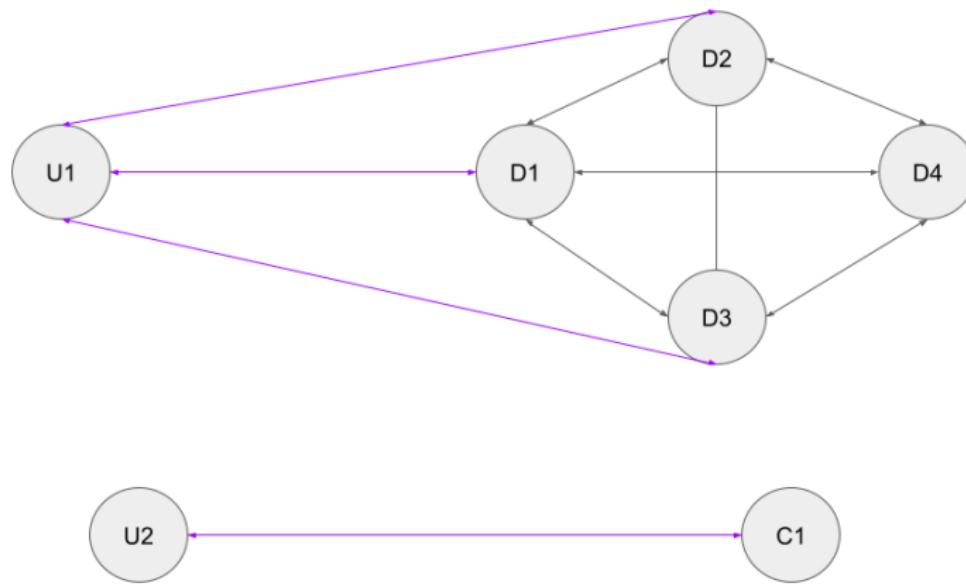


FIGURE: Un sistema descentralizado vs un sistema centralizado, desde el punto de vista de un usuario.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

La ventaja de un **sistema distribuido** vs un sistema centralizado es que hay **mayor redundancia** en caso de que exista un **fallo** o haya un ataque informático. Supóngase que únicamente se puede navegar en un sitio a través de un solo servidor y que haya un ataque de DDoS...

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Peer-to-peer networks. Una red peer-to-peer (P2P, entre pares) se caracteriza porque todos los componentes de la misma tienen la misma precedencia, es decir, todos tienen la misma jerarquía. Idealmente todos los participantes deberían ser capaces de comunicarse entre sí directamente, pero en la práctica eso no es así.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

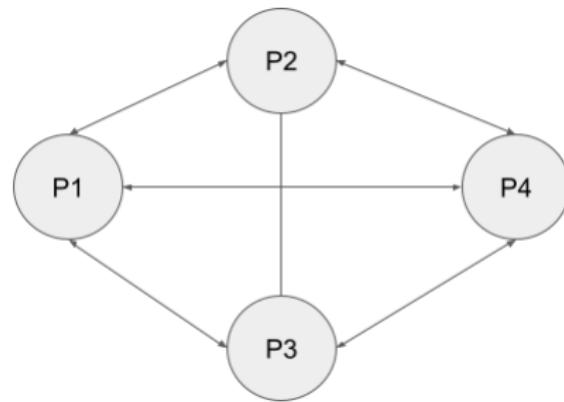


FIGURE: Un sistema P2P. Todos los nodos se ven entre sí.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Hashing algorithms. Una función resumen es aquella que convierte una cadena arbitraria de caracteres Y con tamaño m a una cadena X de tamaño fijo n , es decir:

$$X_m = \text{hash}(Y_n) \quad (1)$$

Visto como caja negra a como se muestra en la figura, entra la información en la función hash y se obtiene un resumen a la salida.



FIGURE: Función hash.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Una función hash debe cumplir con 3 requisitos para serlo

- Resistencia a primer pre-Imagen.
- Resistencia a segunda pre-Imagen.
- Resistencia a colisión.

Por lo tanto, su principal función es proveer la característica de integridad. O en términos prácticos, si un solo bit cambia en Y_n , la salida X_m será completamente diferente.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Resistencia a primer pre-Imagen

DEFINITION

Es computacionalmente intratable encontrar alguna entrada tal que

$$y = h(x') \tag{2}$$

Es decir, no se puede calcular x' a partir de y usando la inversa de $h(x')$.
Es decir, $h(x')$ es una función *one – way* (de una sola vía).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Resistencia a segunda pre-imagen

DEFINITION

Es computacionalmente intratable encontrar una segunda entrada tal que

$$h(x) = h(x') \quad (3)$$

donde $x \neq x'$

Es decir, no se pueden calcular ni x ni x' a partir de y usando la inversa de $h(x')$ o $h(x)$.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Resistencia a colisión

DEFINITION

Es computacionalmente intratable que a partir de dos entradas distintas x , x' encontrar:

$$h(x) = h(x') \quad (4)$$

Es decir, no es posible obtener un mismo hash de 2 entradas diferentes. Acá se debe tener cuidado porque pareciera que las propiedades 2 y 3 son iguales, cuando en realidad no lo son. La propiedad 2 está enfocada a las entradas x , x' y la propiedad 3 está enfocada a la salida.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Un ejemplo típico de verificación de integridad es cuando alguien baja una imagen de Linux desde el sitio del distribuidor.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

The screenshot shows a web browser window displaying a Canonical Ubuntu tutorial titled "How to verify your Ubuntu download". The page lists steps for verification, including "Retrieve the correct signature key". A terminal window at the bottom shows the command "gpg --keyid-format long --verify SHA256SUMS.gpg SHA256SUMS" being run, with its output indicating a successful verification.

How to verify your Ubuntu download

Getting Started

CANONICAL

ubuntu® Enterprise Developer Community Download

We are hiring Products

Search

1 Overview

2 Necessary software

3 Download checksums and signatures

4 Retrieve the correct signature key

5 Verify the SHA256 checksum

6 Check the ISO

7 What's next?

4. Retrieve the correct signature key

Depending on your platform, you may or may not need to download the public key used to authenticate the checksum file (Ubuntu and most variants come with the relevant keys pre-installed). The easiest way to find out if you need the key is to run the authentication command:

```
gpg --keyid-format long --verify SHA256SUMS.gpg SHA256SUMS
```

We use GnuPG's "long" (64-bit) key IDs throughout this tutorial, since "short" (32-bit) key IDs are insecure.

If you don't have the keys...

If there is no public key for Ubuntu already present, you will get an error message similar to the following:

```
gpg: Signature made Thu Apr  5 22:19:36 2018 EDT
      using DSA key ID 46181433FBB75451
gpg: Can't check signature: No public key
gpg: Signature made Thu Apr  5 22:19:36 2018 EDT
```

Type here to search

FIGURE: Uso de SHA-256 para verificar la integridad de una ISO de Ubuntu.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Existen diferentes **funciones hash** que han sido/son usadas en la vida cotidiana, sobretodo para firma digital (se explicará esto en esta sesión)

- RACE Integrity Primitives Evaluation Message Digest (RIPEMD-160, ya en desuso).
- Message Digest 5 (MD5, de 128 bits, ya en desuso).
- Familia SHA (SHA-0 -no puesto en uso por fallo en diseño-, SHA-1 -ya en desuso-, SHA-256, SHA-384, SHA-512).
- Keccak (estándar de reciente creación, también conocido como SHA-3).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Private key cryptography. La criptografía de clave privada involucra el uso de una clave secreta, que es usada para encriptar información almacenada o intercambiar información encriptada en un canal inseguro.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Private key cryptography. El uso de este esquema permite que exista la característica de **confidencialidad** (solo el emisor y el receptor saben el contenido del mensaje) y **privacidad** (solo personas autorizadas acceden a la información proporcionada).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

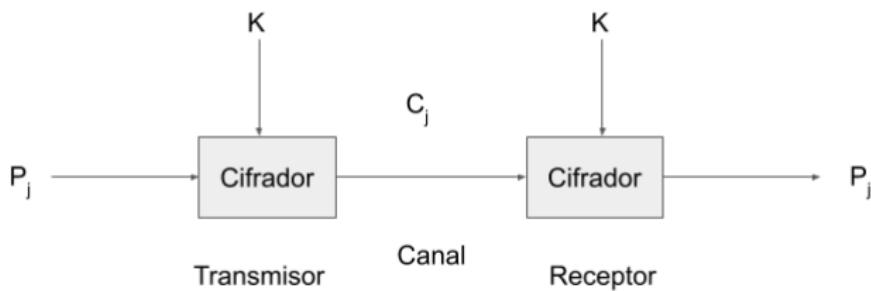


FIGURE: Uso de clave privada para el intercambio de información.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Los algoritmos de cifrado mas comunes encontrados en la literatura y usados en la industria son:

- DES (ex-estándar estadunidense). Clave efectiva de 56 bits. Existe su sucesor 3DES, con clave efectiva de 112 bits.
- AES (actual estándar estadunidense, el mas usado y estudiado alrededor del mundo). Con claves de 128, 192, y 256 bits.
- ARIA (estándar coreano). Con claves de 128, 192 y 256.
- SM4 (antes SMS4, estándar chino). Con clave de 128 bits.
- GOST (estándar ruso). Con claves de 256 bits.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Private key cryptography. Sin embargo, para usar este esquema, se requiere que la clave se haya acordado de antemano de manera segura, lo cual en la práctica no es viable. Además de que no garantiza características de **autenticidad** y **no repudio** de manera directa.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Public Key cryptography. La criptografía de clave pública involucra el uso de 2 claves, una que se hace **pública** (de allí su nombre) y otra que únicamente conoce el usuario. Con el uso de este esquema, es posible tener

- **Confidencialidad/Privacidad.**
- **Autenticidad.** La información no puede ser falsificada y el autor no puede ser suplantado.
- **No repudio.** El autor de la información no puede negar que lo hizo.

Y además, en lugar de tener un repositorio de claves privadas cual directorio telefónico, quien quiera comunicarse con alguien únicamente deberá 'bajar' la clave pública de esa persona (o entidad) cada vez que lo necesite, sin necesidad de almacenar dicha clave (ya que pueden incluso cambiarse).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Public Key cryptography. Si bien existen mas algoritmos de criptografía de clave pública, los 3 mas utilizados en la industria son

- **Exponenciación modular.** Se basa en la exponenciación de un número entero grande con otro número entero grande, y cuyo residuo (o módulo) es obtenido respecto a un número primo p . El tamaño de clave es de 1024 bits como mínimo.
- **Algoritmo RSA.** Parecido al anterior, pero ahora el módulo es obtenido respecto a un producto de los números primos grandes p , q . El tamaño de clave es de 1024 bits como mínimo.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Public Key cryptography.

- Criptografía de Curvas Elípticas. Se basa en la generalización (redefinición) de operaciones matemáticas tales como la suma y la multiplicación en unos entes matemáticos llamados Curvas Elípticas, donde la 'multiplicación' es la base fundamental de este algoritmo. Con uso de claves de al menos 256 bits. Además del tamaño de clave, su ventaja es la mayor versatilidad para ser implementado en hardware de manera relativamente directa.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

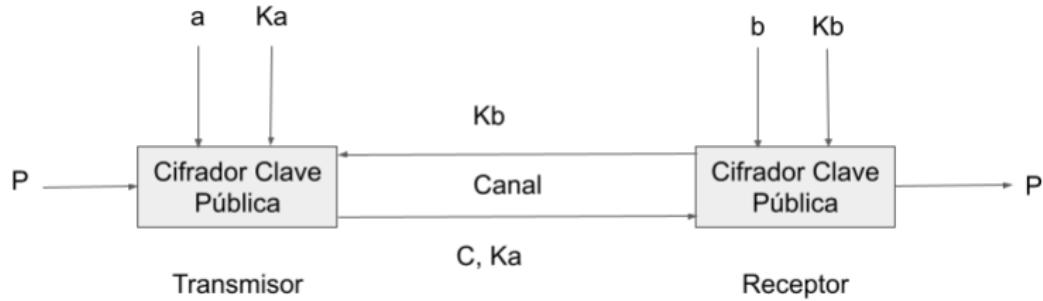


FIGURE: Uso de clave pública para el intercambio de información.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Public-Private key cryptography. Por supuesto, existen ventajas y desventajas entre ambos esquemas de clave pública y privada. Por ejemplo:

- Un esquema de clave privada es mas rápido que un esquema de clave pública en tiempos de ejecución.
- Un esquema de clave privada usa claves mas cortas.
- Pero como ya se ha mostrado, la clave privada tiene el problema de **reparto de claves** y que únicamente provee la característica de **confidencialidad** de manera directa.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Public-Private key cryptography. Por lo tanto, muchos protocolos de comunicaciones mezclan el uso de ambos esquemas para tener lo mejor de ambos mundos. El esquema de clave pública se usa para el intercambio de una clave secreta, y dicha clave secreta cifra la información intercambiada. Eso ocurre con el HTTPS al usar el protocolo SSL (Secure-Socket-Exchange, versión actual 3.0) o TLS (Transfer Layer Secure, versión actual 1.3).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

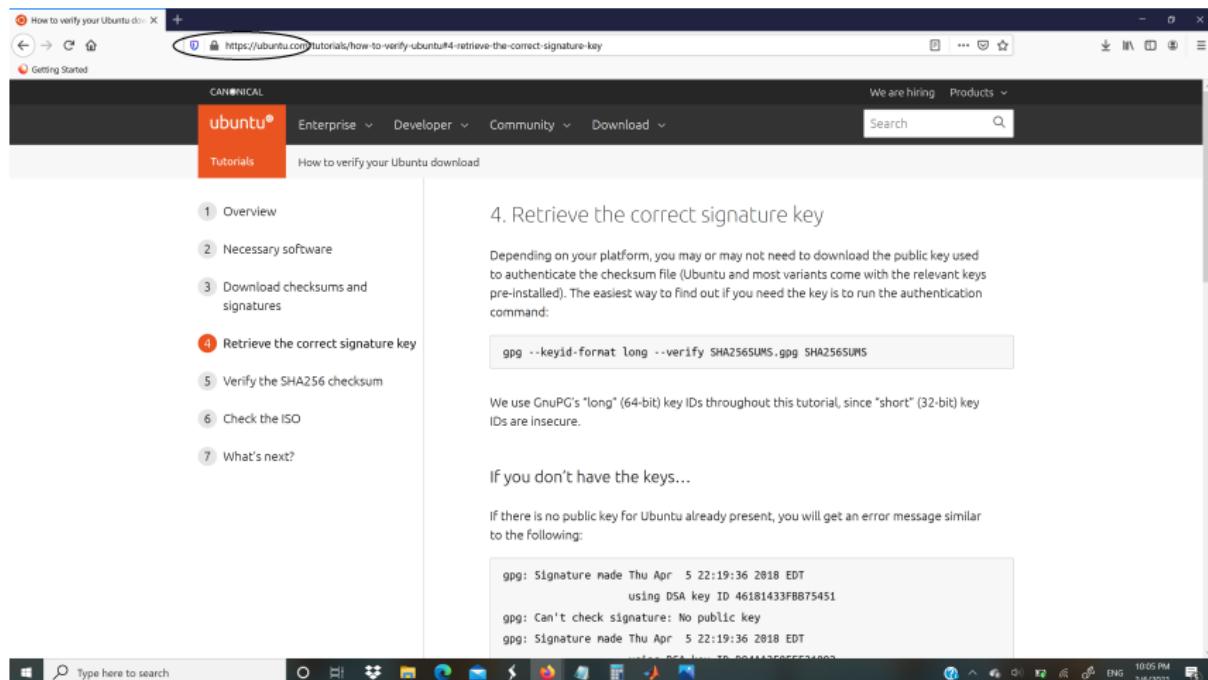


FIGURE: HTTPS presente en un navegador.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Cryptographic signatures. La firma digital es una operación criptográfica donde se busca llevar al mundo digital las propiedades de una firma autógrafo. Es decir:

- Que no pueda ser falsificable y que el autor no pueda ser suplantado (**autenticidad**).
- Que el autor no pueda negar su autoría (**no repudio**).
- Y en términos generales, incorpora la característica de **integridad** (que el mensaje o archivo no hayan sido alterados).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Cryptographic signatures. La firma digital posee las siguientes características:

- Debido a la naturaleza de los algoritmos criptográficos en cuestión de procesamiento, la firma digital varía de acuerdo a la información ingresada.
- Debido a la naturaleza de los algoritmos criptográficos en cuestión de resistencia a potenciales ataques (criptoanálisis), la firma digital tiene caducidad.

Un ejemplo de firma digital lo podemos encontrar en la FIEL (Firma Electrónica Avanzada, hoy e.firma) a la hora de realizar trámites ante el Sistema de Administración Tributaria (SAT).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Cryptographic signatures. Si bien depende del esquema a utilizar, la firma digital se compone de

- ① Un algoritmo de clave pública para garantizar la **autenticidad** y el **no repudio**.
- ② Una función hash que permita calcular el resumen de la información a ser firmada, proporcionando **integridad**.
- ③ Y naturalmente, la información a firmar digitalmente y que será procesada por la función hash.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

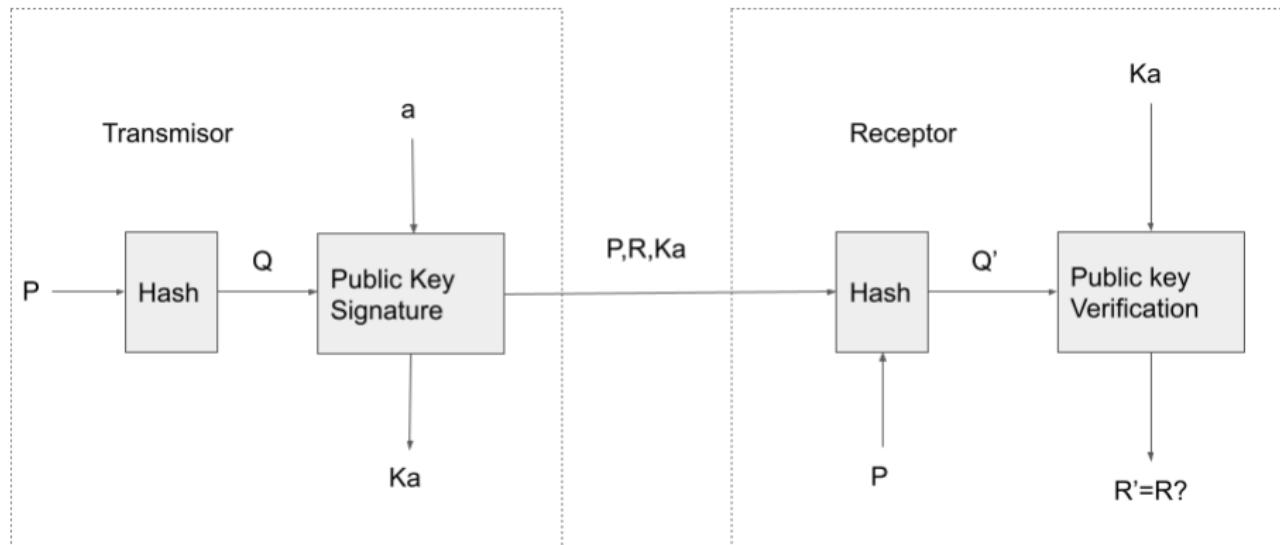


FIGURE: Esquema básico de firma digital.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Cryptographic signatures. Los esquemas de firma digital mas conocidos son

- ① **Digital Signature Algorithm (DSA).** Se basa en la operación matemática llamada **modulación exponencial** y operaciones adicionales con números modulo p , donde p es un primo entero muy grande (de 1024 bits en adelante).
- ② **Firma RSA.** Similar a DSA, pero acá se ocupa el **producto de 2 números primos p y q** como base del algoritmo (de 1024 bits en adelante).
- ③ **Elliptic Curve Cryptography Digital Signature Algorithm (ECDSA).** Es análogo a DSA, pero con el uso de **Criptografía de Curvas Elípticas**, con la ventaja de que pueden usarse directamente representaciones en bits y que las claves son mas cortas (de 256 bits en adelante).

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Merkle trees. Los árboles de Merkle son una derivación de los típicos árboles binarios comúnmente estudiados en Ciencias de la Computación. El árbol se construye **de abajo hacia arriba**, es decir, desde las ramas hasta la raíz.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Merkle trees. Dicha construcción se basa en la concatenación de los hashes obtenidos en cada rama del árbol, desde las ramas hasta la raíz, y donde en cada nodo se calcula el hash de los **nodos hijos**.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

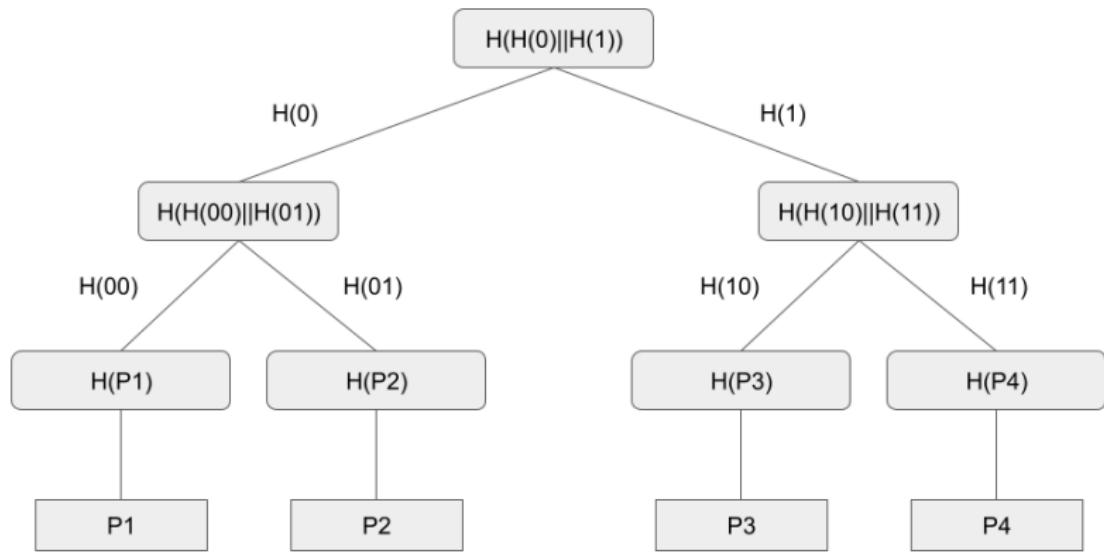


FIGURE: Árbol de Merkle. Basado en: https://es.wikipedia.org/wiki/%C3%81rbol_de_Merkle#/media/Archivo:Hash_Tree.svg

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Chain of blocks and consensus protocols.

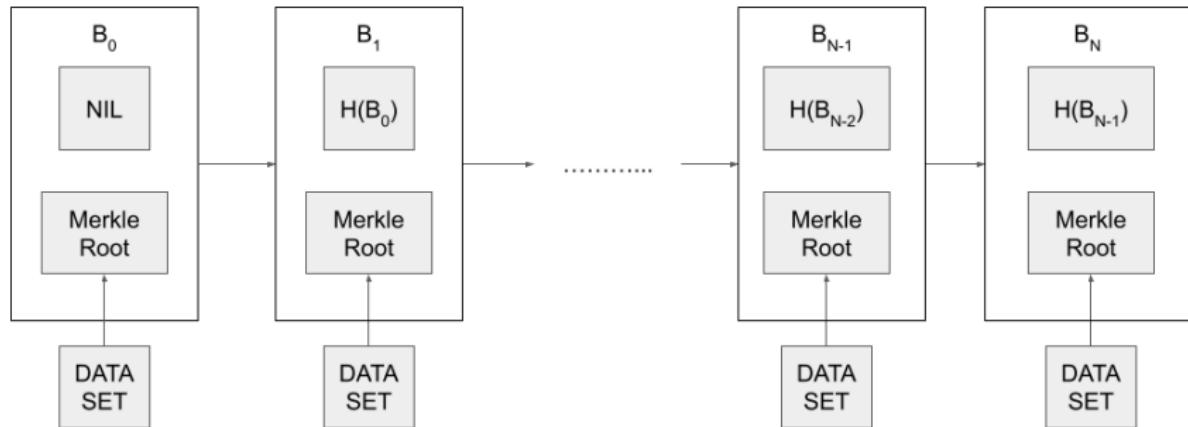


FIGURE: Cadena de bloques. Basada en: Uwe Roth, Blockchain for Dummies, everithing you never wanted to know about blockchain but you have been forced to find out, 2017, Luxembourg Institute of Science and Technology.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Chain of blocks and consensus protocols. Lo que podemos ver en la filmación anterior es

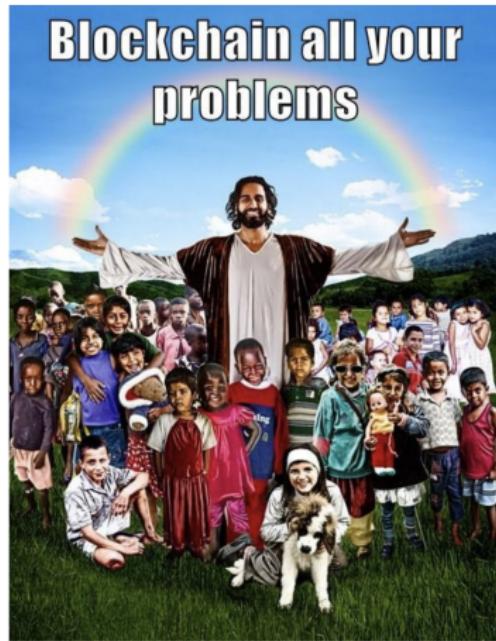
- El bloque B_0 es conocido como el bloque **Genesis**.
- Los **DATA SET** son arreglados de tal forma que puedan formar un **árbol de Merkle**, con el fin de hacer mas eficiente la operación de hashing para todos ellos. Hace referencia a las transacciones almacenadas en cada bloque.
- El hash de la raíz del árbol de Merkle (y elementos adicionales) forman el primer bloque (B_0).
- El bloque siguiente (B_1) se forma con otro árbol de Merkle para los datos referenciados en ese bloque junto al hash de B_0 y datos adicionales.
- Por lo general cada bloque tiene un nonce y un timestamp.

CONCEPTOS BÁSICOS (TÉCNICOS) DE BLOCKCHAIN

Chain of blocks and consensus protocols. Lo que podemos ver en la filmación anterior es

- El bloque B_i se conforma de otro árbol de Merkle para los datos referenciados en ese bloque junto al hash de B_{i-1} .
- El bloque B_N se conforma de otro árbol de Merkle para los datos referenciados en ese bloque junto al hash de B_{N-1} . Y se termina la cadena de bloques.

!!! BLOCKCHAIN !!!



**Blockchain all your
problems**



**YOU GET A BLOCKCHAIN!
AND YOU GET A BLOCKCHAIN!**

**EVERYBODY GETS A
BLOCKCHAIN!!!**

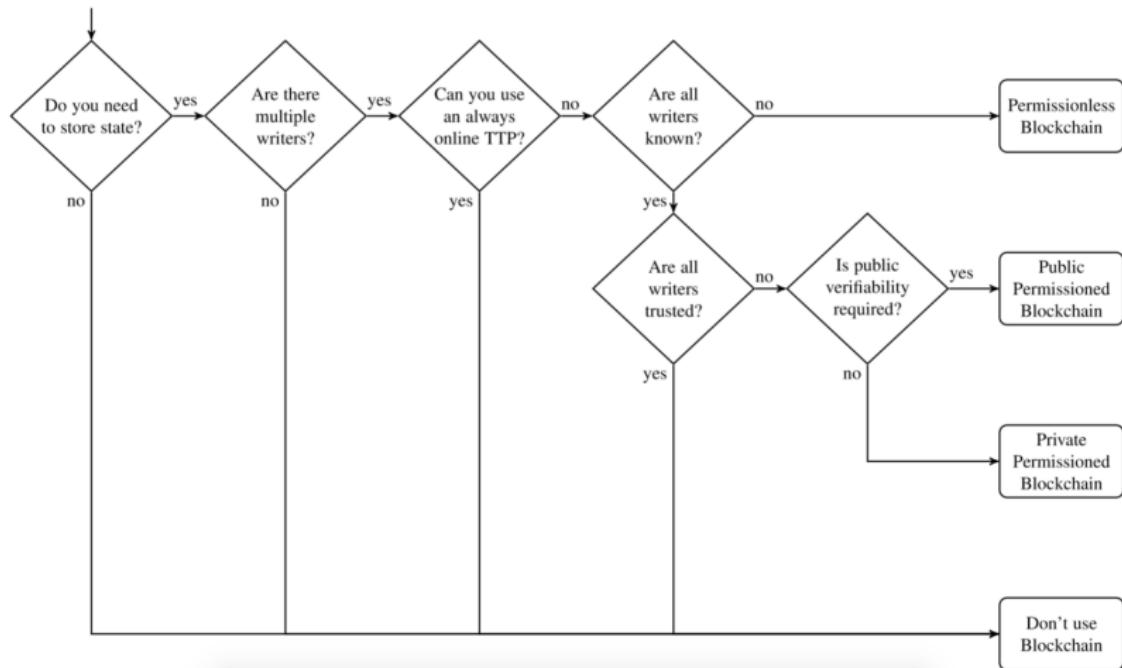
CONSIDERACIONES SOBRE EL USO DE BLOCKCHAINS

Sobre el problema y el uso de blockchain¹

- ¿Qué problema estás resolviendo?
- ¿Porqué es necesario resolverlo?
- ¿Cómo lo estas resolviendo?
- ¿Porqué necesitas una blockchain para resolverlo?

¹ Mihaylov, M., Razo-Zapata, I., and Nowe, A. (2018). NRGcoin—A blockchain-based reward mechanism for both production and consumption of renewable energy. In *Transforming Climate Finance and Green Investment with Blockchains* (pp. 111-131). Academic Press

CONSIDERACIONES SOBRE EL USO DE BLOCKCHAINS



CONSIDERACIONES SOBRE EL USO DE BLOCKCHAINS

Cambios / Transformaciones a considerar



Strategy

Disintermediation

Blockchains might remove companies from their strategic position within a business network, e.g. banks being replaced by Blockchain-based payments



Governance

From top-down to bottom-up business processes based on smart contracts



IT Systems

Control on systems supporting business processes

Access: Who can "see" things?

Control: Who can "change" (add) things?



People

New skills to be developed

E.g. partner and contract management, software engineering, and cryptography



Culture

From hierarchical (bureaucratic) structures to flexible & open structures

Adhocracy

Mendling, Jan, et al. "Blockchains for business process management-challenges and opportunities." *ACM Transactions on Management Information Systems (TMIS)* 9.1 (2018)

FUENTES Y MATERIAL EXTRA

Criptografía (funciones hash, clave pública y privada, firmas digitales, MD5, RIPEMD, DES).

- MENEZES, Alfred J.; VAN OORSCHOT, Paul C.; VANSTONE, Scott A. *Handbook of applied cryptography*. CRC press, 2018.
- RODRÍGUEZ-HENRÍQUEZ, Francisco, et al. *Cryptographic algorithms on reconfigurable hardware*. Springer Science & Business Media, 2007.
- STALLINGS, William; TAHILIANI, Mohit P. *Cryptography and network security: principles and practice*, vol. 6. 2014.
- HOLDEN, Joshua. *The mathematics of secrets: cryptography from caesar ciphers to digital encryption*. Princeton University Press, 2018.

Peer to Peer

- STEINMETZ, Ralf; WEHRLE, Klaus (ed.). *Peer-to-peer systems and applications*. Springer, 2005.

Sistemas distribuidos

- TANENBAUM, Andrew S.; VAN STEEN, Maarten. *Distributed systems: principles and paradigms*. Prentice-hall, 2007.

FUENTES Y MATERIAL EXTRA

Curvas Elípticas

- HANKERSON, Darrel; MENEZES, Alfred J.; VANSTONE, Scott. Guide to elliptic curve cryptography. Springer Science & Business Media, 2006.

AES

- PUB, N. F. Specification for the Advanced Encryption Standard (AES). Tech. Rep. FIPS PUB 197, Federal Information Processing Standards, 2001.

ARIA

- KWON, Daesung, et al. A Description of the ARIA Encryption Algorithm. RFC 5794, March 2010. <https://tools.ietf.org/html/rfc5794>, 2010.

SM4

- ZHANG, Jingbin; MA, Meng; WANG, Ping. Fast implementation for SM4 cipher algorithm based on bit-slice technology. En International Conference on Smart Computing and Communication. Springer, Cham, 2018. p. 104-113.
- GB/T 32907-2016 (el estándar está en idioma chino).

FUENTES Y MATERIAL EXTRA

GOST

- DOLMATOV, Vasily. GOST 28147-89: Encryption, decryption, and message authentication code (MAC) algorithms. Internet engineering task force RFC, 2010, vol. 5830.

SHA-1

- WANG, Xiaoyun; YIN, Yiqun Lisa; YU, Hongbo. Finding collisions in the full SHA-1. En Annual international cryptology conference. Springer, Berlin, Heidelberg, 2005. p. 17-36.

SHA-256, SHA-384, SHA-512

- GILBERT, Henri; HANDSCHUH, Helena. Security analysis of SHA-256 and sisters. En International workshop on selected areas in cryptography. Springer, Berlin, Heidelberg, 2003. p. 175-193.

Keccak

- BERTONI, Guido, et al. Keccak. En Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 2013. p. 313-314.
- DWORKIN, Morris J. SHA-3 standard: Permutation-based hash and extendable-output functions. 2015.

FUENTES Y MATERIAL EXTRA

Merkle Trees

- GEORG, B. Merkle Signature Schemes, Merkle Trees and Their Cryptanalysis. Ruhr-Universitat Bochum. Retrieved, 2013, vol. 20.

Chain of Blocks

- MONRAT, Ahmed Afif; SCHELÉN, Olov; ANDERSSON, Karl. A survey of blockchain from the perspectives of applications, challenges, and opportunities. IEEE Access, 2019, vol. 7, p. 117134-117151.

Consensus Protocols

- ANDONI, Merlin, et al. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. Renewable and Sustainable Energy Reviews, 2019, vol. 100, p. 143-174.

Distributed Ledger Technology

- ANDONI, Merlin, et al. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. Renewable and Sustainable Energy Reviews, 2019, vol. 100, p. 143-174.
- Natarajan, Harish; Krause, Solvej Karla; Gradstein, Helen Luskin. 2017. Distributed Ledger Technology (DLT) and blockchain (English). FinTech note; no. 1. Washington, D.C. : World Bank Group.
- Dave Birch (Consult Hyperion). UK Government of Science Report "Distributed Ledger Technology: beyond blockchain".

FUENTES Y MATERIAL EXTRA

Blockchain

- Blockchain Technology and Designing with Blockchain. Qinghua Lu and Len Bass. Slides at ICSA workshop. April 2018.
- Hasse, F., et al. "Blockchain—an opportunity for energy producers and consumers." PwC Global Power and Utilities (2016): 1-45.
- Blockchain for dummies: Everything You Never Wanted To Know About Blockchain But Have Been Forced To Find Out. LIST material by Dr. Uwe Roth
- Bitcoin and the Blockchain Technology. VUB material by Dr. Mihail Mihaylov
- <https://www.weforum.org/reports/the-future-of-financial-infrastructure-an-ambitious-look-at-how-blockchain-can-reshape-financial-services>
- <https://bitsonblocks.net/2016/02/29/a-gentle-introduction-to-immutability-of-blockchains/>

AGENDA

- Descripción del ecosistema de desarrollo

DESCRIPCIÓN DEL ECOSISTEMA



HYPERLEDGER



**HYPERLEDGER
FABRIC**

ubuntu The Ubuntu logo, featuring a red circle with a white dash and a green circle with a white dash.

docker

The Docker logo, featuring a blue whale carrying a stack of shipping containers.

Java

The Java logo, featuring a stylized orange flame above a blue coffee cup.

=GO

The Go programming language logo, featuring the word "GO" in a blue sans-serif font with a horizontal equals sign preceding it.

DESCRIPCIÓN DEL ECOSISTEMA

LF Energy:

part of The Linux Foundation ecosystem of sustainable open source



For the last 16 years, The Linux Foundation has provided unparalleled support for open source communities through financial and intellectual resources, governance structure, IT infrastructure, services, events, and training.

Dedicated to building sustainable ecosystems around open source projects, The Linux Foundation is working with the global technology community to solve the world's hardest problems through open source and **creating the largest shared technology investment in history**.

The Linux Foundation is the umbrella organization for **more than 350 open source projects** accelerating open technology development and commercial adoption. Some of the game-changing initiatives hosted by The Linux Foundation include:



The Power of Together

DESCRIPCIÓN DEL ECOSISTEMA



[Blockchain](#) [Products](#) ▾ [Solutions](#) ▾ [Services](#) [Learn](#) ▾ [Explore More](#) ▾

What is Hyperledger Fabric?

The flexible blockchain framework behind the IBM Blockchain Platform is helping innovators ignite a global business transformation

DESCRIPCIÓN DEL ECOSISTEMA

Premier Members



General Members



<https://www.hyperledger.org/about/members>

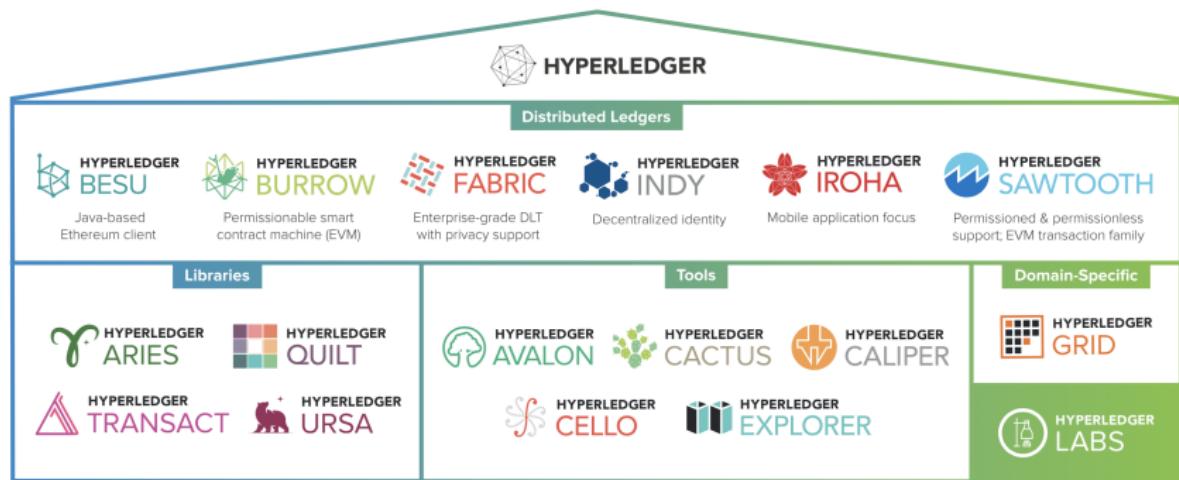
DESCRIPCIÓN DEL ECOSISTEMA



HYPERLEDGER

- Hyperledger es un proyecto de código abierto comunitario
- Enfocado en proporcionar soluciones **empresariales** de blockchain
- Brinda soporte para el desarrollo de una suite estable de:
 - Frameworks (Fabric, Sawtooth, Indy),
 - Herramientas (Caliper), y
 - Librerías (Ursa)

DESCRIPCIÓN DEL ECOSISTEMA



DESCRIPCIÓN DEL ECOSISTEMA



- Fabric es un framework que permite el diseño y desarrollo de blockchains permisionadas
- Proporciona modularidad y versatilidad para un rango amplio de casos industriales
- Emplea componentes plug & play
 - Protocolos de consenso
 - Privacidad, y
 - Membership services

DESCRIPCIÓN DEL ECOSISTEMA



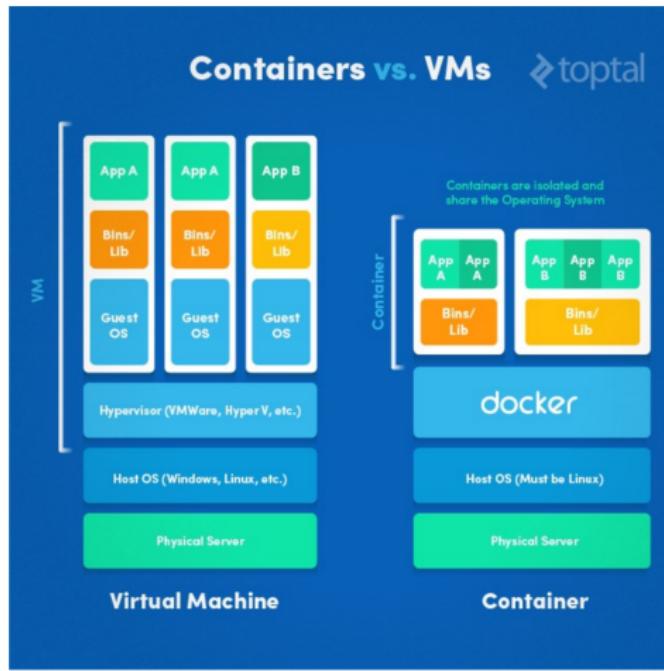
DOCKER

Es un proyecto de código abierto:

- Automatiza el despliegue de aplicaciones dentro de **contenedores** de software ... i.e. software que permite administrar contenedores
- Facilita la virtualización de dichas aplicaciones en múltiples sistemas operativos
- Permite que "contenedores" independientes se ejecuten dentro de una sola instancia del sistema operativo

DOCKER

Máquinas virtuales VS Contenedores:



DESCRIPCIÓN DEL ECOSISTEMA



AGENDA

- Requerimientos básicos

REQUERIMIENTOS BÁSICOS

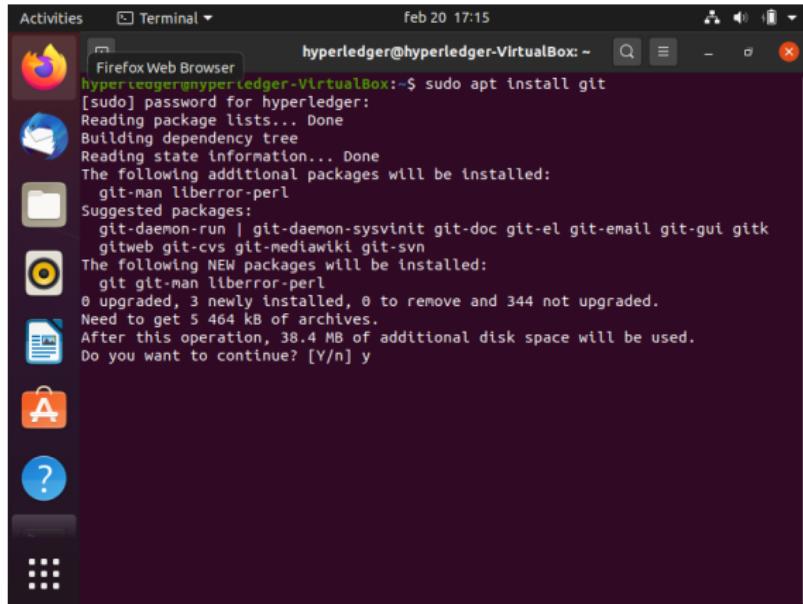
Paquete	Versión	Compatibilidad
Virtual Box	6.0 en adelante	6.0 en adelante
Ubuntu Linux	18 en adelante	18 en adelante
go	1.13.x	1.13.x
docker	Mayor a 1.14.0	1.14.10
Hyperledger	2.0	2.0

REQUERIMIENTOS BÁSICOS

Instalación de los componentes a utilizar:

https://github.com/Cocoa-ci/taller_blockchain

REQUERIMENTOS BÁSICOS

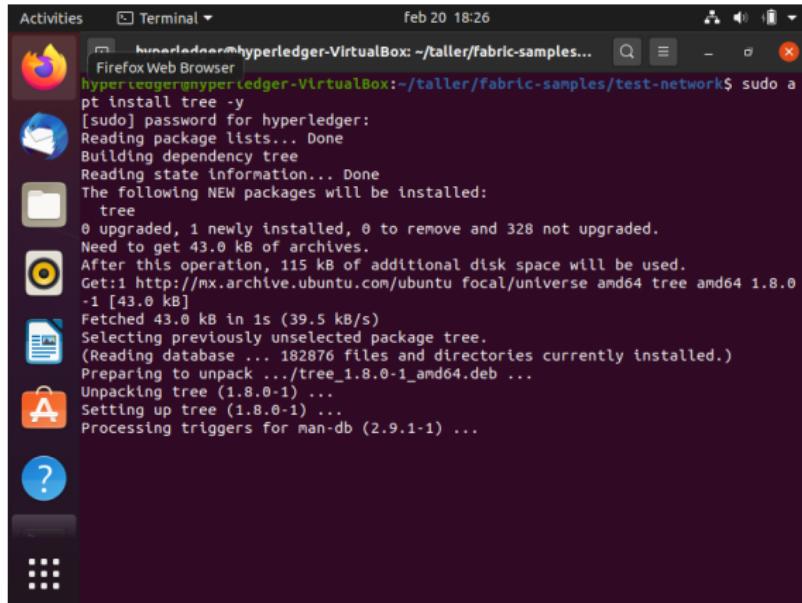


A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the date and time are "Feb 20 17:15". The user is running the command "sudo apt install git". The terminal output shows the package manager reading lists, building a dependency tree, and determining packages to install. It lists suggested packages like git-daemon-run and git-email, and new packages like git-man. It shows 0 upgraded, 3 newly installed, and 344 not upgraded. The total size of the download is 5 464 KB. The user is prompted with "Do you want to continue? [Y/n] y".

```
hyperledger@hyperledger-VirtualBox:~$ sudo apt install git
[sudo] password for hyperledger:
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 344 not upgraded.
Need to get 5 464 kB of archives.
After this operation, 38.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

FIGURE: Instalar GitHub.

REQUERIMENTOS BÁSICOS

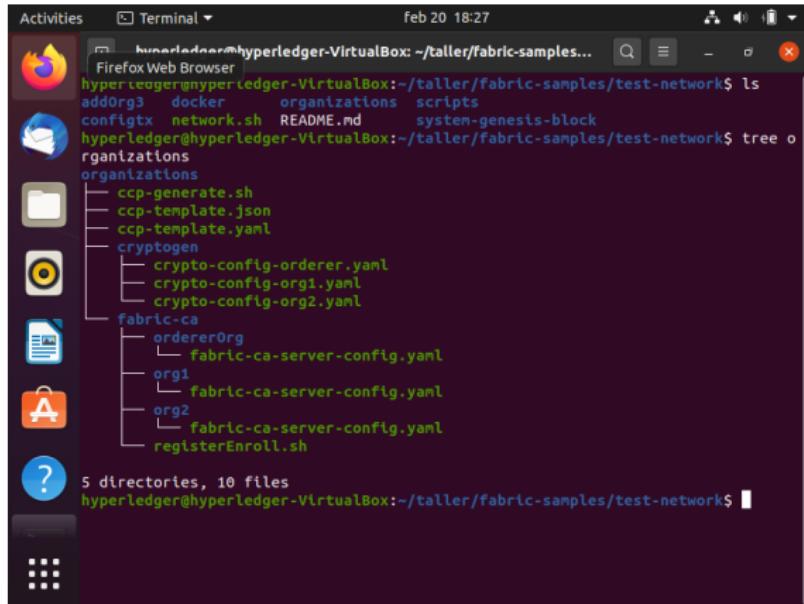


A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "hyperledger@hyperledger-VirtualBox: ~/taller/fabric-samples...". The date and time at the top right are "Feb 20 18:26". The terminal content shows the command "sudo apt install tree -y" being run, followed by the output of the package manager. The output includes package lists, dependency trees, state information, and the final result of 0 upgraded, 1 newly installed, and 328 not upgraded packages. It also shows the download of the tree package from the archive. The terminal window has a dark background and light-colored text.

```
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$ sudo apt install tree -y
[sudo] password for hyperledger:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 328 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://mx.archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 1s (39.5 kB/s)
Selecting previously unselected package tree.
(Reading database ... 182876 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
```

FIGURE: Instalar Tree.

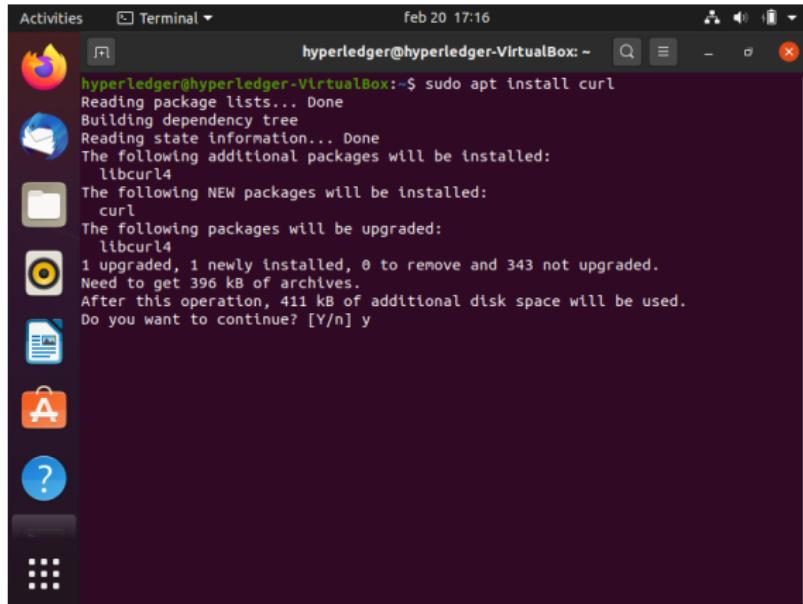
REQUERIMIENTOS BÁSICOS



```
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$ ls
addOrg3    docker    organizations scripts
configtx   network.sh README.md system-genesis-block
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$ tree organizations
organizations
├── ccp-generate.sh
├── ccp-template.json
└── ccp-template.yaml
organizations
├── cryptogen
│   ├── crypto-config-orderer.yaml
│   ├── crypto-config-org1.yaml
│   └── crypto-config-org2.yaml
└── fabric-ca
    ├── ordererOrg
    │   └── fabric-ca-server-config.yaml
    ├── org1
    │   └── fabric-ca-server-config.yaml
    └── org2
        └── fabric-ca-server-config.yaml
registerEnroll.sh
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$ tree
5 directories, 10 files
```

FIGURE: Prueba de Tree.

REQUERIMENTOS BÁSICOS



A screenshot of a Linux desktop environment showing a terminal window. The terminal title is "hyperledger@hyperledger-VirtualBox:~". The date and time at the top right are "Feb 20 17:16". The terminal content shows the command "sudo apt install curl" being run, along with the output of the package manager. It lists packages to be installed (libcurl4), new packages to be installed (curl), and packages to be upgraded (libcurl4). It also shows the total size of the download (396 KB) and the additional disk space required (411 kB). A question is asked at the end: "Do you want to continue? [Y/n]".

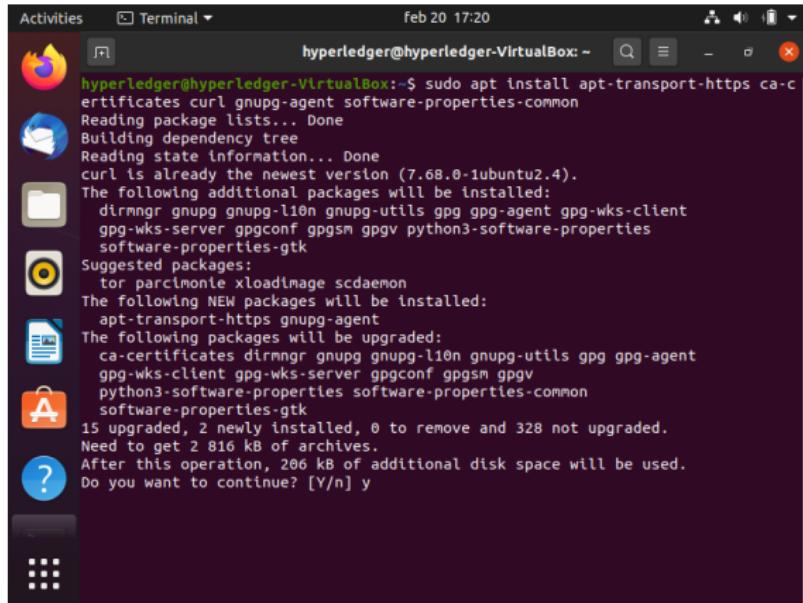
```
hyperledger@hyperledger-VirtualBox:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 upgraded, 1 newly installed, 0 to remove and 343 not upgraded.
Need to get 396 kB of archives.
After this operation, 411 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

FIGURE: Instalar Curl.

REQUERIMIENTOS BÁSICOS

Docker

REQUERIMENTOS BÁSICOS

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal window has a dark background and displays the following command and its output:

```
hyperledger@hyperledger-VirtualBox:~$ sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.4).
The following additional packages will be installed:
  dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client
  gpg-wks-server gpgconf gpgsm gpgv python3-software-properties
  software-properties-gtk
Suggested packages:
  tor parcmonte xloadimage scdaemon
The following NEW packages will be installed:
  apt-transport-https gnupg-agent
The following packages will be upgraded:
  ca-certificates dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent
  gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv
  python3-software-properties software-properties-common
  software-properties-gtk
15 upgraded, 2 newly installed, 0 to remove and 328 not upgraded.
Need to get 2 816 kB of additional disk space.
After this operation, 206 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

The desktop interface includes a dock with various icons for applications like a browser, file manager, terminal, and system settings.

FIGURE: Instalar Docker. Parte 1.

REQUERIMENTOS BÁSICOS

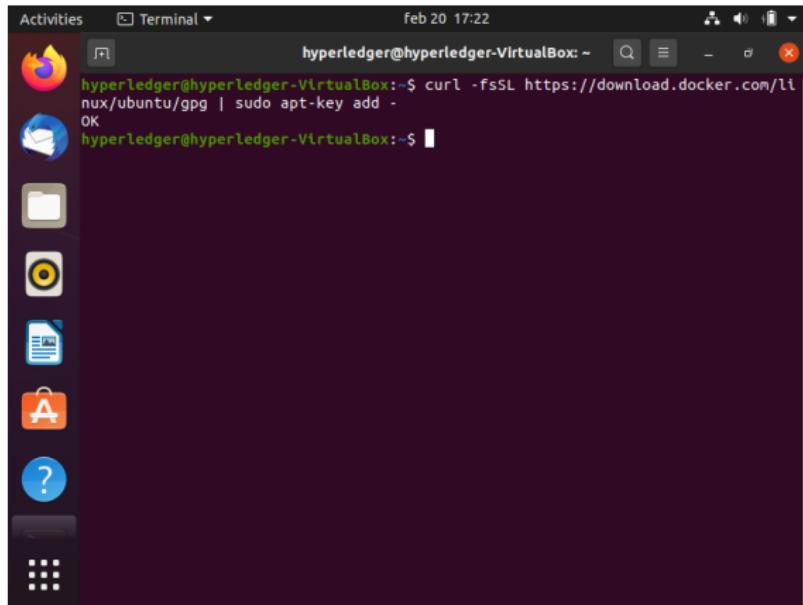
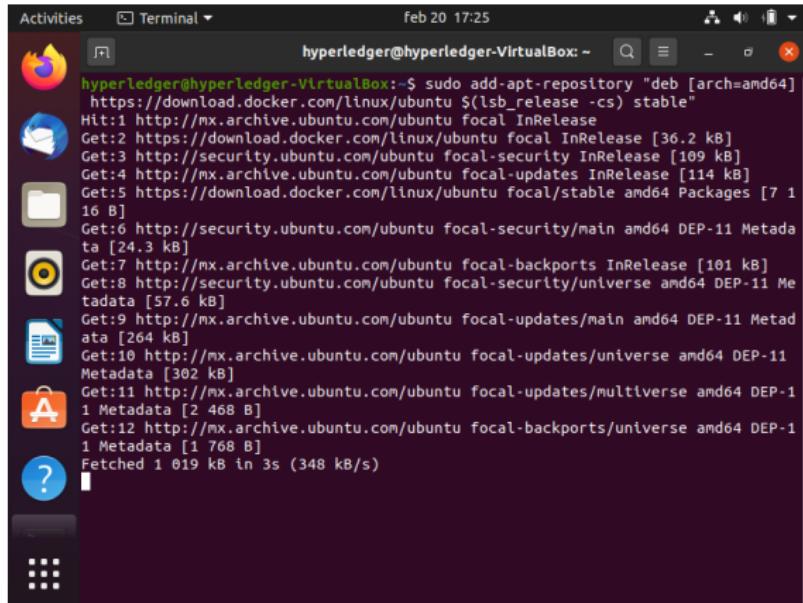


FIGURE: Instalar Docker. Parte 2.

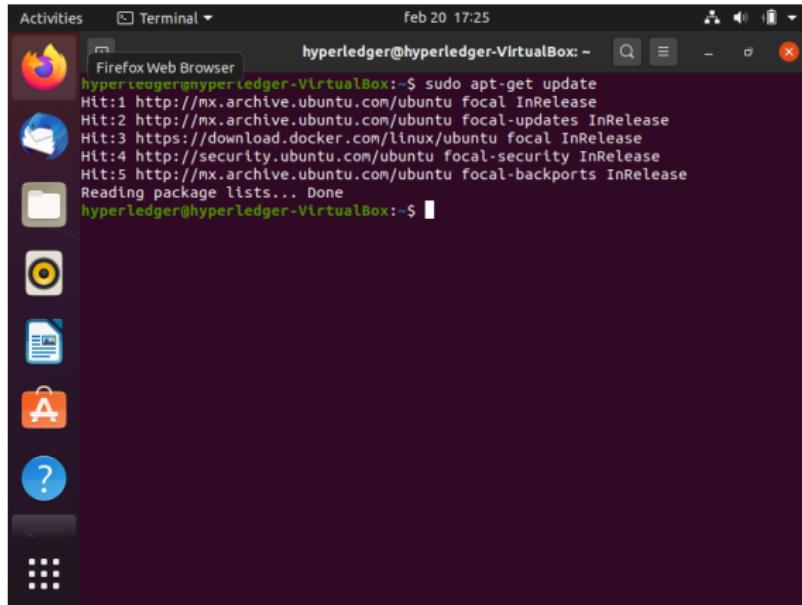
REQUERIMENTOS BÁSICOS

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal window has a dark background and displays a command-line session. The session starts with the user "hyperledger" logging in via SSH at "hyperledger@hyperledger-VirtualBox". The user runs the command "sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable"". The terminal then shows a series of "Get:" commands from the Ubuntu repositories, listing various packages and their metadata being downloaded. The log ends with "Fetched 1 019 kB in 3s (348 kB/s)".

```
hyperledger@hyperledger-VirtualBox:~$ sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Hit:1 http://mx.archive.ubuntu.com/ubuntu focal InRelease
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [36.2 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:4 http://mx.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [7 1
16 B]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metada
ta [24.3 kB]
Get:7 http://mx.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Me
tadata [57.6 kB]
Get:9 http://mx.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metad
ata [264 kB]
Get:10 http://mx.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11
Metadata [302 kB]
Get:11 http://mx.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-1
1 Metadata [2 468 B]
Get:12 http://mx.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-1
1 Metadata [1 768 B]
Fetched 1 019 kB in 3s (348 kB/s)
```

FIGURE: Instalar Docker. Parte 3.

REQUERIMENTOS BÁSICOS

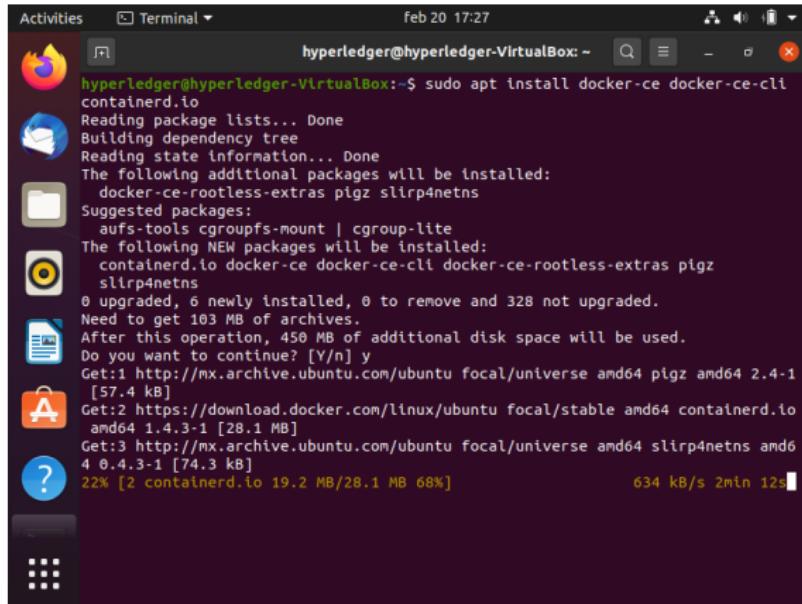


A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the date and time are "Feb 20 17:25". The user is "hyperledger@hyperledger-VirtualBox: ~". The terminal shows the command "sudo apt-get update" being run, followed by its output:

```
hyperledger@hyperledger-VirtualBox:~$ sudo apt-get update
Hit:1 http://mx.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://mx.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 https://download.docker.com/linux/ubuntu focal InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://mx.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
hyperledger@hyperledger-VirtualBox:~$
```

FIGURE: Instalar Docker. Parte 4 - Actualizar Ubuntu.

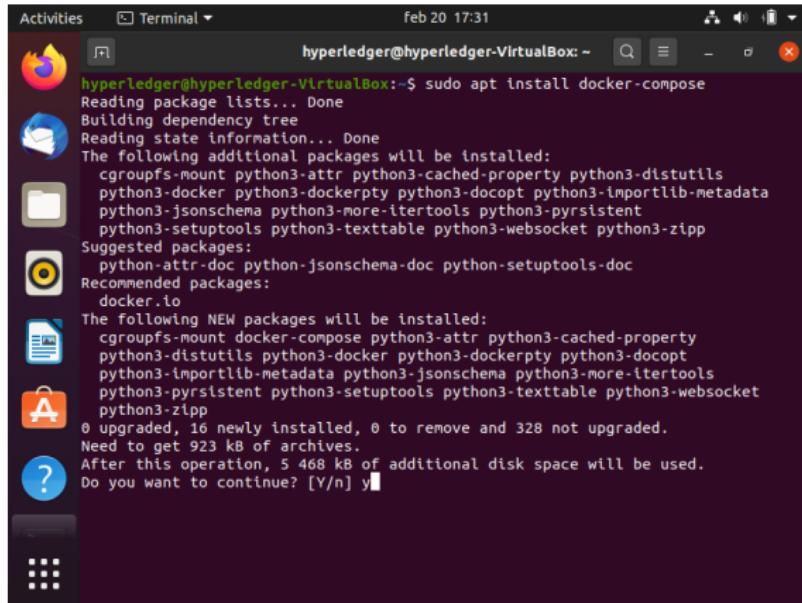
REQUERIMENTOS BÁSICOS

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal shows the command "sudo apt install docker-ce docker-ce-cli containerd.io" being run, followed by the output of the package manager. The output details the installation of Docker CE, including dependencies like slirp4netns and suggested packages like aufs-tools. It also lists NEW packages to be installed and provides download links from mx.archive.ubuntu.com and download.docker.com. The progress bar at the bottom indicates the download is at 22% completion.

```
hyperledger@hyperledger-VirtualBox:~$ sudo apt install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras pigz
  slirp4netns
0 upgraded, 6 newly installed, 0 to remove and 328 not upgraded.
Need to get 103 MB of archives.
After this operation, 450 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mx.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1
[57.4 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io
and64 1.4.3-1 [28.1 MB]
Get:3 http://mx.archive.ubuntu.com/ubuntu focal/universe amd64 slirp4netns amd64
4.0.4.3-1 [74.3 kB]
22% [2 containerd.io 19.2 MB/28.1 MB 68%] 634 kB/s 2min 12s
```

FIGURE: Instalar Docker. Parte 5.

REQUERIMENTOS BÁSICOS

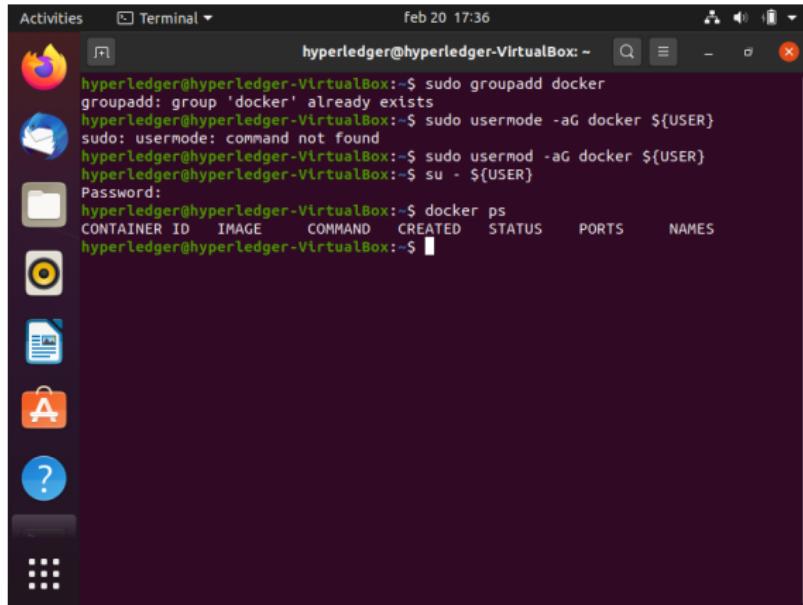


A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal window has a dark background and displays the following command and its output:

```
hyperledger@hyperledger-VirtualBox:~$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cgroupfs-mount python3-attr python3-cached-property python3-distutils
  python3-docker python3-dockerpty python3-docopt python3-importlib-metadata
  python3-jsonschema python3-more-itertools python3-persistent
  python3-setuptools python3-texttable python3-websocket python3-zipp
Suggested packages:
  python-attr-doc python-jsonschema-doc python-setuptools-doc
Recommended packages:
  docker.io
The following NEW packages will be installed:
  cgroupfs-mount docker-compose python3-attr python3-cached-property
  python3-distutils python3-docker python3-dockerpty python3-docopt
  python3-importlib-metadata python3-jsonschema python3-more-itertools
  python3-persistent python3-setuptools python3-texttable python3-websocket
  python3-zipp
0 upgraded, 16 newly installed, 0 to remove and 328 not upgraded.
Need to get 923 kB of additional disk space.
After this operation, 5 468 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

FIGURE: Instalar Docker. Parte 6.

REQUERIMENTOS BÁSICOS



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the date and time are "Feb 20 17:36". The user is "hyperledger@hyperledger-VirtualBox". The terminal output shows the following commands being run:

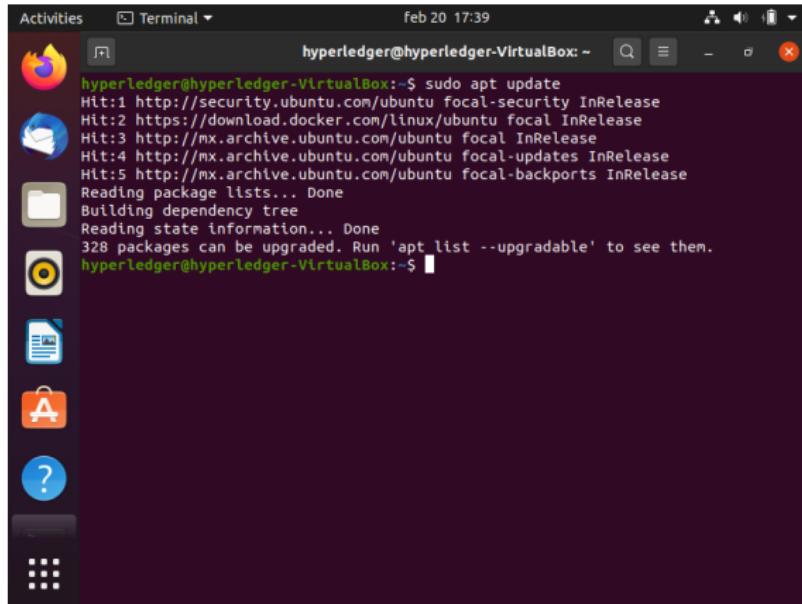
```
hyperledger@hyperledger-VirtualBox:~$ sudo groupadd docker
groupadd: group 'docker' already exists
hyperledger@hyperledger-VirtualBox:~$ sudo usermod -aG docker ${USER}
sudo: usermod: command not found
hyperledger@hyperledger-VirtualBox:~$ sudo usermod -aG docker ${USER}
hyperledger@hyperledger-VirtualBox:~$ su - ${USER}
Password:
hyperledger@hyperledger-VirtualBox:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS      NAMES
hyperledger@hyperledger-VirtualBox:~$
```

FIGURE: Alistando Docker.

REQUERIMIENTOS BÁSICOS

Instalar Golang

REQUERIMENTOS BÁSICOS



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the date and time are "Feb 20 17:39". The terminal content shows the output of the command "sudo apt update":

```
hyperledger@hyperledger-VirtualBox:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 https://download.docker.com/linux/ubuntu focal InRelease
Hit:3 http://mx.archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://mx.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://mx.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
328 packages can be upgraded. Run 'apt list --upgradable' to see them.
hyperledger@hyperledger-VirtualBox:~$
```

FIGURE: Actualizar Ubuntu.

REQUERIMENTOS BÁSICOS

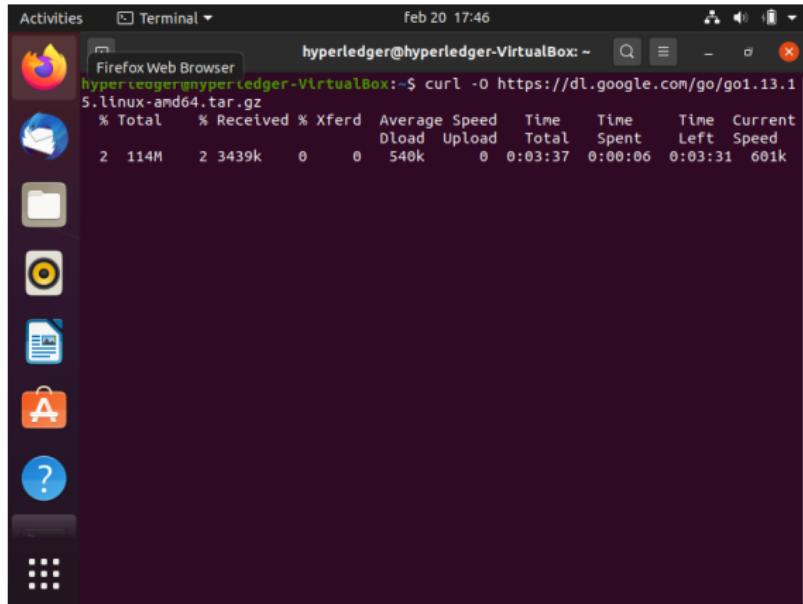


FIGURE: Bajar Go.

REQUERIMENTOS BÁSICOS

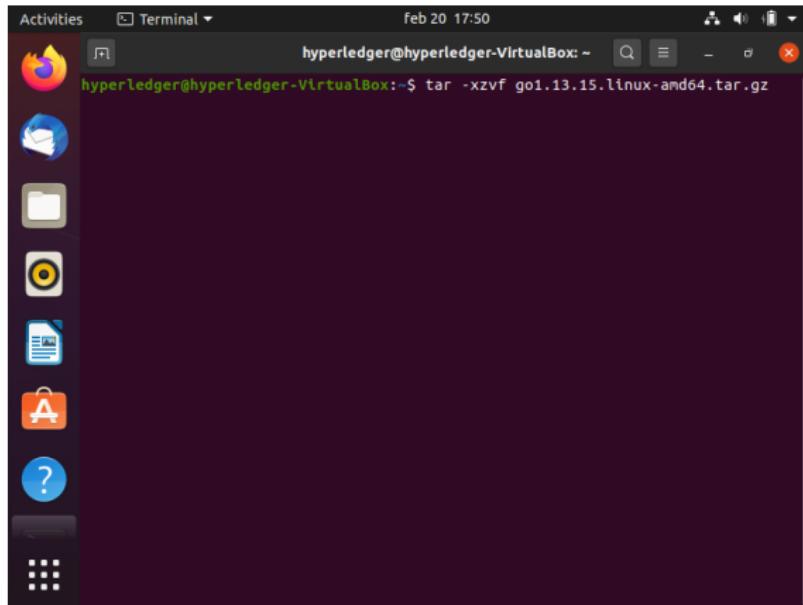


FIGURE: Descomprimir Go.

REQUERIMENTOS BÁSICOS

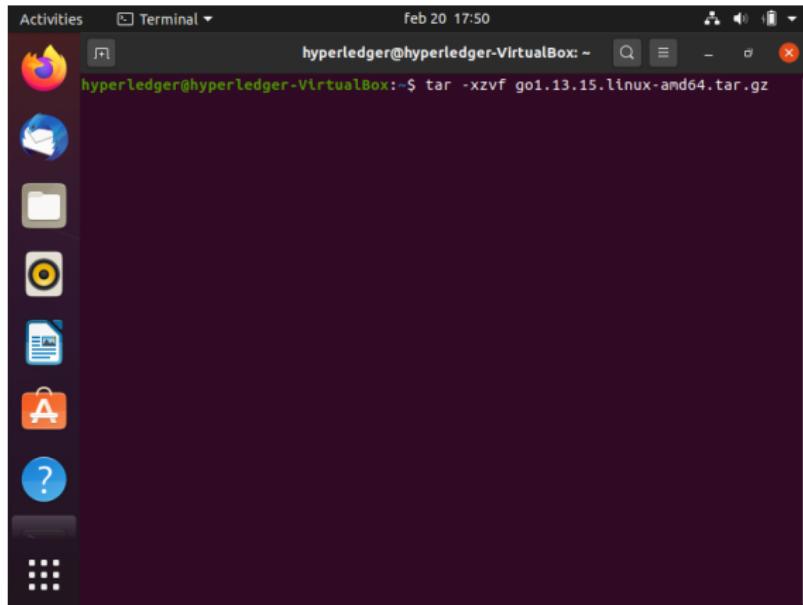


FIGURE: Descomprimir Go.

REQUERIMENTOS BÁSICOS

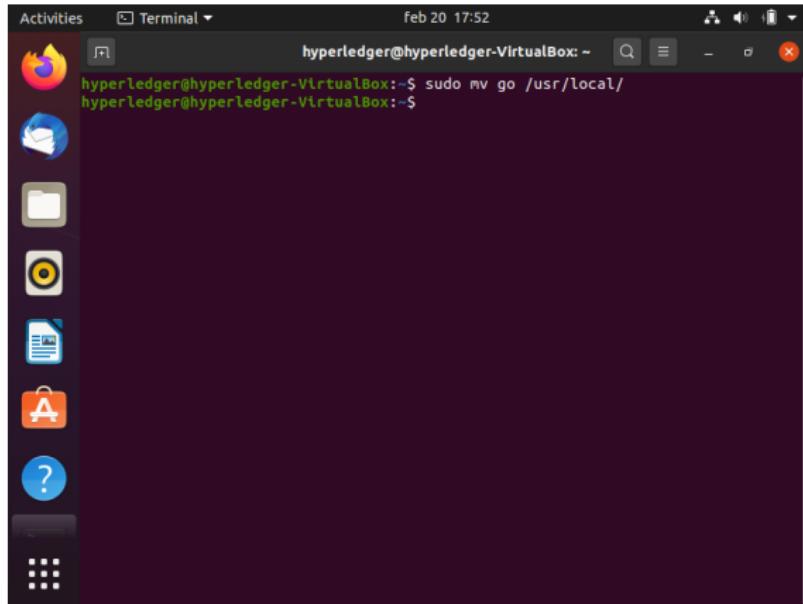
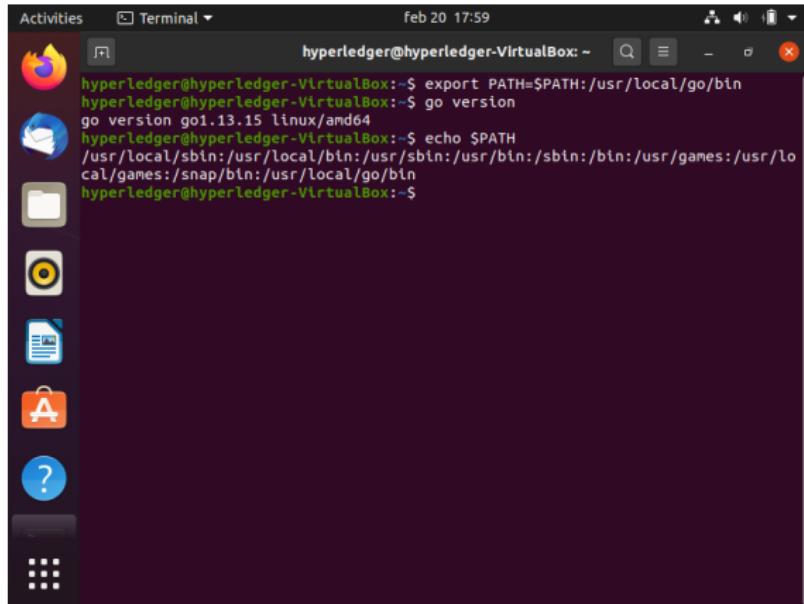


FIGURE: Mover a /usr/local.

REQUERIMENTOS BÁSICOS



The screenshot shows a terminal window titled "Terminal" with the command line "hyperledger@hyperledger-VirtualBox:~". The terminal displays the following commands and output:

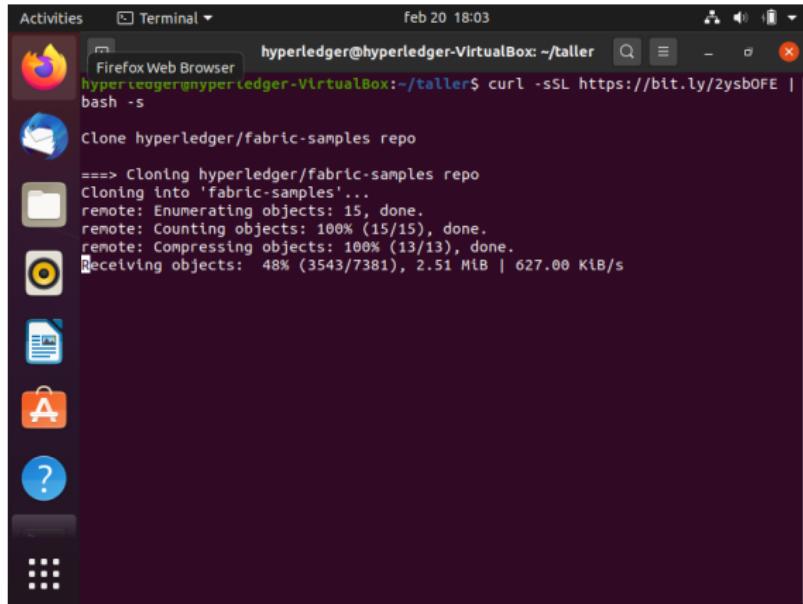
```
hyperledger@hyperledger-VirtualBox:~$ export PATH=$PATH:/usr/local/go/bin
hyperledger@hyperledger-VirtualBox:~$ go version
go version go1.13.15 linux/amd64
hyperledger@hyperledger-VirtualBox:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/local/go/bin
hyperledger@hyperledger-VirtualBox:~$
```

FIGURE: Anexar al path.

REQUERIMIENTOS BÁSICOS

Instalar Hyperledger Fabric

REQUERIMENTOS BÁSICOS

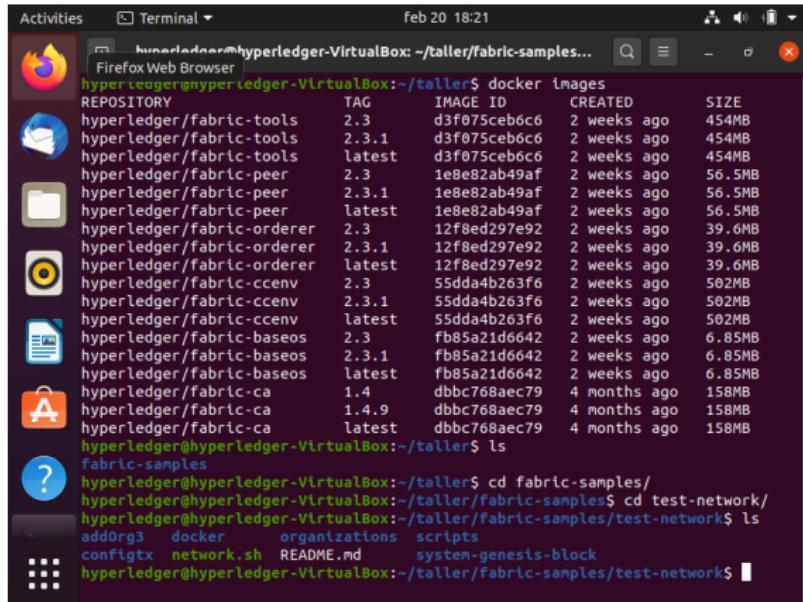


The screenshot shows a terminal window titled "hyperledger@hyperledger-VirtualBox: ~/taller". The command entered is "curl -sSL https://bit.ly/2ysb0FE | bash -s". The output of the command is displayed, showing the cloning of the "hyperledger/fabric-samples" repository. The process includes cloning into a directory named "fabric-samples", enumerating objects, counting objects at 100% (15/15), compressing objects at 100% (13/13), and receiving objects at 48% (3543/7381) with a speed of 2.51 MiB/s.

```
hyperledger@hyperledger-VirtualBox:~/taller$ curl -sSL https://bit.ly/2ysb0FE | bash -s
Clone hyperledger/fabric-samples repo
====> Cloning hyperledger/fabric-samples repo
Cloning into 'fabric-samples'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (13/13), done.
Receiving objects: 48% (3543/7381), 2.51 MiB | 627.00 KiB/s
```

FIGURE: Instalar Hyperledger Fabric.

REQUERIMENTOS BÁSICOS



Activities Terminal Feb 20 18:21

```
hyperledger@hyperledger-VirtualBox:~/taller$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
hyperledger/fabric-tools   2.3      d3f075ceb6c6  2 weeks ago   454MB
hyperledger/fabric-tools   2.3.1    d3f075ceb6c6  2 weeks ago   454MB
hyperledger/fabric-tools   latest   d3f075ceb6c6  2 weeks ago   454MB
hyperledger/fabric-peer    2.3      1e8e82ab49af  2 weeks ago   56.5MB
hyperledger/fabric-peer    2.3.1    1e8e82ab49af  2 weeks ago   56.5MB
hyperledger/fabric-peer    latest   1e8e82ab49af  2 weeks ago   56.5MB
hyperledger/fabric-orderer 2.3      12f8ed297e92  2 weeks ago   39.6MB
hyperledger/fabric-orderer 2.3.1    12f8ed297e92  2 weeks ago   39.6MB
hyperledger/fabric-orderer latest   12f8ed297e92  2 weeks ago   39.6MB
hyperledger/fabric-ccenv    2.3      55dda4b263f6  2 weeks ago   502MB
hyperledger/fabric-ccenv    2.3.1    55dda4b263f6  2 weeks ago   502MB
hyperledger/fabric-ccenv    latest   55dda4b263f6  2 weeks ago   502MB
hyperledger/fabric-baseos   2.3      fb85a21d6642  2 weeks ago   6.85MB
hyperledger/fabric-baseos   2.3.1    fb85a21d6642  2 weeks ago   6.85MB
hyperledger/fabric-baseos   latest   fb85a21d6642  2 weeks ago   6.85MB
hyperledger/fabric-ca       1.4      dbbc768aec79  4 months ago  158MB
hyperledger/fabric-ca       1.4.9    dbbc768aec79  4 months ago  158MB
hyperledger/fabric-ca       latest   dbbc768aec79  4 months ago  158MB
hyperledger@hyperledger-VirtualBox:~/taller$ ls
fabric-samples
hyperledger@hyperledger-VirtualBox:~/taller$ cd fabric-samples/
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples$ cd test-network/
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$ ls
addOrg3  docker  organizations  scripts
configtx  network.sh  README.md  system-genesis-block
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$
```

FIGURE: Ver imágenes de Docker.