



# Introducción a Hyperledger Fabric

Fundamentos de blockchains

René Dávila - Jorge Solano

# Índice

## 1 Introducción

## 2 Hyperledger Fabric v2.0

Componentes y características

## 3 Instalación de Hyperledger Fabric v2.0

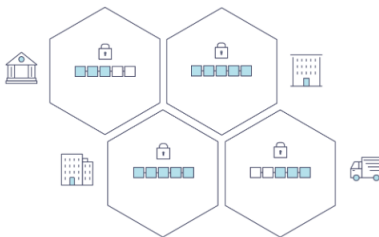
Prerrequisitos de instalación

Instalación

# Blockchain

En términos generales, una **blockchain** se podría considerar como un **libro mayor** en el área contable (**ledger**). Es un lugar donde se llevan a cabo transacciones inmutables dentro de una red de nodos distribuida.

Cada uno de estos **nodos** mantiene una **copia del libro mayor** aplicando transacciones que han sido validadas mediante el protocolo de consenso, agrupadas en bloques que incluyen un hash que permite enlazar cada bloque con su bloque anterior.



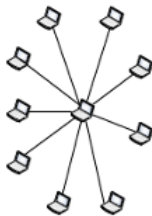
# Criptomonedas

La aplicación más reconocida de blockchain es la criptomoneda **Bitcoin**.

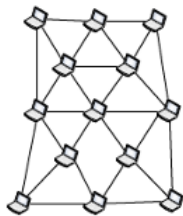
**Ethereum** es una criptomoneda alternativa, la cual integra muchas características de Bitcoin, pero agrega contratos inteligentes (smart contracts) con el fin de crear una plataforma para aplicaciones distribuidas.



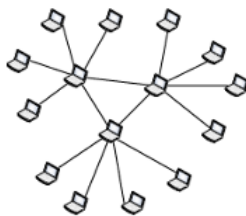
Tanto **Bitcoin** como **Ethereum** caen en la clasificación de blockchain, las cuales clasificaremos como tecnologías de **blockchain públicas y sin permiso**. Básicamente, son redes públicas, abiertas a cualquiera, donde los participantes interactúan de manera anónima.



Public  
Blockchain



Consortium  
Blockchain



Private  
Blockchain

En los casos de **uso empresarial** la **identidad** de los participantes es un **requisito indispensable**, como por ejemplo en una transacción financiera, donde las regulaciones el Know Your Customer (KYC) y el Anti-Money Laundering (AML) deben seguirse de manera puntual.



Para el uso empresarial de blockchain se deben considerar los siguientes requerimientos:

- Cada participante debe identificarse y ser identificable.
- Las redes deben estar autorizadas.
- Alto rendimiento durante las transacciones.
- Baja latencia en la confirmación de transacción.
- Privacidad y confidencialidad de las transacciones y de la información de las transacciones.

# Smart Contracts

Una red blockchain utiliza **Smart Contracts** para proporcionar acceso controlado al libro mayor.

No solo son un mecanismo clave para encapsular la información y mantenerla simple en toda la red, sino que también se pueden escribir para permitir a los participantes ejecutar ciertos aspectos de las transacciones automáticamente.



# Smart Contracts



## Hyperledger project community

La Linux Foundation se caracteriza por nutrir proyectos **open source** de **open governance** que genera comunidades sostenibles fuertes y ecosistemas prósperos.

**Linux Foundation** creó el proyecto **Hyperledger** en 2015 para avanzar en la industria de la tecnología de blockchain.



## Hyperledger project community

La comunidad ha desarrollado los siguientes **Distributed Ledgers**:

- **Hyperledger Besu.**
- **Hyperledger Burrow.**
- **Hyperledger Indy.**
- **Hyperledger Iroha.**
- **Hyperledger Sawtooth.**
- **Hyperledger Fabric.**

## Hyperledger project community

**Hyperledger Besu**, cliente *Ethereum* basado en Java, alternativa para crear casos de uso de redes públicas y autorizadas.

- Usa tecnología Apache.
- Usa redes de prueba Rinkeby, Ropsten y Görli.
- Usa algoritmos de consenso PoW, PoA, IBFT.



## Hyperledger project community

**Hyperledger Burrow**, distribución completa de blockchain binario único centrado en la simplicidad, la velocidad y la ergonomía del desarrollador.

- Admite Smart Contracts basados en EVM y WASM.
- Usa consenso BFT mediante el algoritmo Tendermint.



## Hyperledger project community

**Hyperledger Indy**, proporciona herramientas, bibliotecas y componentes reutilizables para proporcionar identidades digitales arraigadas en blockchains u otros ledgers distribuidos.

- Tiene como objetivo interoperabilidad entre dominios administrativos o entre aplicaciones.
- Interoperable con otras blockchain o de uso independiente para descentralización.



## Hyperledger project community

**Hyperledger Iroha**, está diseñado para ser simple y fácil de incorporar en proyectos de infraestructura o IoT que requieren tecnología de contabilidad distribuida.

- Construcción y diseño en C++ modular.
- Usa un algoritmo tolerante a fallas nuevo llamado YAC.



## Hyperledger project community

**Hyperledger Sawtooth**, ofrece una arquitectura flexible y modular que separa el sistema central del dominio de la aplicación.

- Los contratos inteligentes pueden especificar las reglas de negocio para las aplicaciones.
- Admite diversos algoritmos de consenso como PBFT o PoET.





## Hyperledger project community

**Hyperledger Fabric** es uno de los **proyectos de blockchain en Hyperledger**, el cual es un ledger que utiliza smart contracts y donde los participantes manejan las transacciones.



# Índice

## 1 Introducción

## 2 Hyperledger Fabric v2.0

Componentes y características

## 3 Instalación de Hyperledger Fabric v2.0

Prerrequisitos de instalación

Instalación

## Hyperledger Fabric v2.0

Hyperledger Fabric es una plataforma **open source** de grado **empresarial**, que permite manejar una tecnología distribuida de ledger (una blockchain).

Tiene algunas **capacidades claves** diferentes con respecto a otros ledger distribuidos o a otras plataformas de blockchain.

# Índice

## 1 Introducción

## 2 Hyperledger Fabric v2.0

Componentes y características

## 3 Instalación de Hyperledger Fabric v2.0

Prerrequisitos de instalación

Instalación

# Private and permissioned

**Hyperledger Fabric** destaca por sobre otros sistemas de blockchain porque **es privada y autorizada**, esto es, los miembros de la red se registran a través de un Membership Service Provider (MSP) de confianza.

Entonces, por ejemplo, si los participantes no pueden confiar completamente uno en el otro (por ejemplo, si son competidores en la misma industria), la red puede operar bajo el modelo de **gobierno basado en la confianza** que existe entre los participantes, como lo es un acuerdo legal o un marco de referencia para manejar disputas.

# Pluggable

**Hyperledger Fabric** ofrece varias **opciones enchufables**. La información del ledger puede guardarse en múltiples formatos, los mecanismos de consenso se pueden intercambiar y se pueden tener diferentes Proveedores de Servicio de Membresía (MSP).

Esto permite que la plataforma sea **personalizada** y se ajuste a casos de uso y modelos confiables particulares.

Por ejemplo, dentro de una empresa o en una empresa operada por una autoridad de confianza, el protocolo de consenso tolerante a fallas bizantinas podría considerarse innecesario y hasta excesivo, en su lugar se podría pensar que el **protocolo de consenso tolerante a fallas** sería más apropiado.



# Channels

**Hyperledger Fabric** ofrece la posibilidad de crear **canales**, lo que permite que un grupo de participantes tenga su propio ledger de transacciones.

Esta capacidad es importante en redes donde algunos participantes pueden ser competidores y no quieren que cada transacción (pe, una oferta especial en un producto) sea conocida por cada participante.

# Shared Ledger

El ledger de **Hyperledger Fabric** comprende dos componentes: el **world state** y la **transaction log**. Cada participante tiene una copia del ledger de cada red a la que pertenece.

El world state representa la base de datos del ledger. La transaction log guarda la historia actualizada del world state. El ledger es, entonces, la combinación de la **BD** del world state y la historia de la **transaction log**.

# Smart contracts

Los **smart contracts** de **Hyperledger Fabric** están escritos en chaincode y son invocados por una aplicación externa a la blockchain. En la mayoría de los casos, el chaincode interactúa directamente con la base de datos del ledger (world state) y no con la log transaction.

El **chaincode** puede ser implementado **en lenguajes de programación de uso general** (como Java, Go y Node.js), por lo que no se requiere aprender un lenguaje de dominio específico.

# Privacy

**Hyperledger Fabric** permite crear tanto redes donde la **privacidad** (utilizando canales) es un requerimiento operacional clave, así como redes abiertas.

# Consensus

**Hyperledger Fabric** está diseñado para permitir que las redes elijan el **mecanismo de consenso** que mejor se acomode a la relación que existe entre los participantes.

**Fabric** puede aprovechar los protocolos de consenso que **no requieren criptomonedas** nativas para incentivar la minería o la ejecución de contratos inteligentes. Evitar las criptomonedas **reduce** de manera significativa **el riesgo o los vectores de ataque**.

Además, la ausencia de las operaciones de minería criptográfica permite que la plataforma se pueda desplegar con prácticamente el mismo costo operacional que cualquier otro sistema distribuido.

# Modelo de Hyperledger Fabric

- **Assets.**- las definiciones de activos permiten el intercambio de casi cualquier cosa con valor monetario a través de la red, desde alimentos enteros hasta autos antiguos y futuros de divisas.
- **Chaincode.**- la ejecución de Chaincode se divide del orden de las transacciones, lo que limita los niveles requeridos de confianza y verificación en todos los tipos de nodos y optimiza la escalabilidad y el rendimiento de la red.
- **Ledger Features.**- el ledger inmutable y compartido codifica todo el historial de transacciones para cada canal e incluye la capacidad de consulta similar a SQL para una auditoría eficiente y resolución de disputas.

# Modelo de Hyperledger Fabric

- **Privacy.**- los canales y la recopilación de datos privados permiten transacciones multilaterales privadas y confidenciales que generalmente requieren las empresas competidoras y las industrias reguladas que intercambian activos en una red común.
- **Security & Membership Services.**- la membresía autorizada proporciona una red blockchain confiable, donde los participantes saben que todas las transacciones pueden ser detectadas y rastreadas por reguladores y auditores autorizados.
- **Consensus.**- un enfoque único para el consenso permite la flexibilidad y escalabilidad necesarias para la empresa.



# Índice

## 1 Introducción

## 2 Hyperledger Fabric v2.0

Componentes y características

## 3 Instalación de Hyperledger Fabric v2.0

Prerrequisitos de instalación

Instalación

Para poder ejecutar los proyectos de **Hyperledger Fabric** es necesario tener ciertas herramientas ya instaladas. Entonces, primero hay que verificar o instalar los **prerrequisitos de instalación**.

# Índice

## 1 Introducción

## 2 Hyperledger Fabric v2.0

Componentes y características

## 3 Instalación de Hyperledger Fabric v2.0

Prerrequisitos de instalación

Instalación

# Prerrequisitos de instalación

Para poder desarrollar aplicaciones con Hyperledger Fabric se deben tener instaladas las siguientes herramientas:

- Git
- curl
- Docker y Docker Compose (versión 17.06.2-ce o superior)
- Go
- Node.js y NPM
- Python

<https://hyperledger-fabric.readthedocs.io/en/release-2.0/prereqs.html>

# Git

**Git** es un **controlador de versiones distribuido**, gratuito y de código abierto diseñado para manejar proyectos de diferente tamaño con eficiencia.



<https://git-scm.com/downloads>

# curl

**curl** (Client URL) es una biblioteca y una herramienta de línea de comandos que permite **transferir datos** a través de diferentes protocolos.



`https://curl.haxx.se/download.html`

# Docker

**Docker** es un **contenedor de aplicaciones**. Un contenedor es una unidad estándar de software que empaqueta código y todas sus dependencias para que la aplicación se ejecute de manera confiable y rápida en cualquier entorno de computadora.



<https://www.docker.com/get-docker>

# Docker Compose

**Compose** es una herramienta para definir y **ejecutar múltiples contenedores de aplicaciones** Docker. Docker-compose ya está incluida en Docker Desktop.



<https://docs.docker.com/compose/install/>



# Go

**Go** es un **lenguaje de programación** de código abierto que permite crear software simple, confiable y eficiente.



<https://golang.org/dl/>

# Node.js

**Node.js** es un **entorno de ejecución de JavaScript**, de código abierto y multiplataforma, que permite ejecutar código JavaScript fuera de un navegador web.



<https://nodejs.org/en/download/>

# NPM

**NPM** es un **administrador de paquetes** para el lenguaje de programación JavaScript. Es el administrador de paquetes por defecto de Node.js.



`https://www.npmjs.com/get-npm`

# Python

**Python lenguaje de programación interpretado**, de alto nivel y de propósito general.



<https://www.python.org/downloads/>

# Índice

## 1 Introducción

## 2 Hyperledger Fabric v2.0

Componentes y características

## 3 Instalación de Hyperledger Fabric v2.0

Prerrequisitos de instalación

Instalación

# Instalación

Una vez **instalados los prerequisites**, se tiene el ambiente necesario para **descargar e instalar Hyperledger Fabric**. Actualmente el proyecto no cuenta con un instalador de los binarios de Fabric, pero los ejemplos que se descargan poseen scripts de ejecución.

<https://hyperledger-fabric.readthedocs.io/en/release-2.0/install.html>

# Instalación

Para la instalación simplemente hay que descargar los binarios y las imágenes de Fabric

```
curl -sSL https://bit.ly/2ysb0FE | bash -s
```

```
https://raw.githubusercontent.com/hyperledger/fabric/master/scripts/bootstrap.sh
```

# Instalación

La instrucción anterior descarga y descomprime los binarios requeridos y específicos para la plataforma y los guarda en el repositorio (**fabric-samples**) dentro de la **carpeta bin**.

```
bin
├── configtxgen
├── configtxlator
├── cryptogen
├── discover
├── fabric-ca-client
├── fabric-ca-server
├── idemixgen
├── orderer
└── peer

0 directories, 9 files
```



# Instalación

Por último, hay que mapear la carpeta **bin** al **PATH** del sistema. Aquí es importante comentar que también `go` y `go/bin` deben estar en el `PATH`.

```
export GOPATH=/usr/local/go  
export PATH=$PATH:$GOPATH/bin:/Users/jrg_sln/Fabric/fabric-samples/bin
```