

TALLER BLOCKCHAIN

DISEÑO Y DESPLIEGUE DE UNA RED BLOCKCHAIN (1ERA PARTE)

Dr. Iván S. Razo-Zapata
Dr. Alberto F. Martínez

Día 3, Miércoles 3 de Marzo, 2021

AGENDA

- Introducción técnica a Hyperledger Fabric
- Elementos clave de Hyperledger Fabric
- Diseño de red blockchain para tres organizaciones, dos ledgers (canales)

AGENDA

- Introducción técnica a Hyperledger Fabric y hands on

HYPERLEDGER FABRIC



- Fabric is a framework that allows building permissioned blockchain networks
- Offers modularity and versatility for a broad set of industry use cases
- Plug & play components
 - consensus ... i.e. not a hard-coded protocol
 - privacy and
 - membership services

HYPERLEDGER FABRIC



- Modular and configurable
- Fabric is the first distributed ledger platform to support smart contracts ("chaincode") authored in general-purpose programming languages

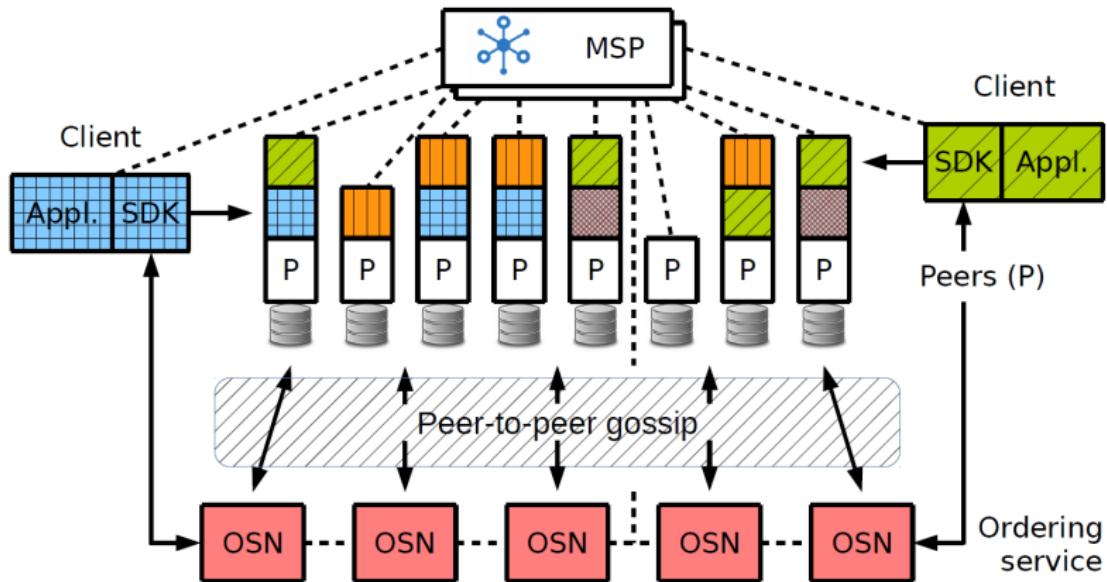


HYPERLEDGER FABRIC

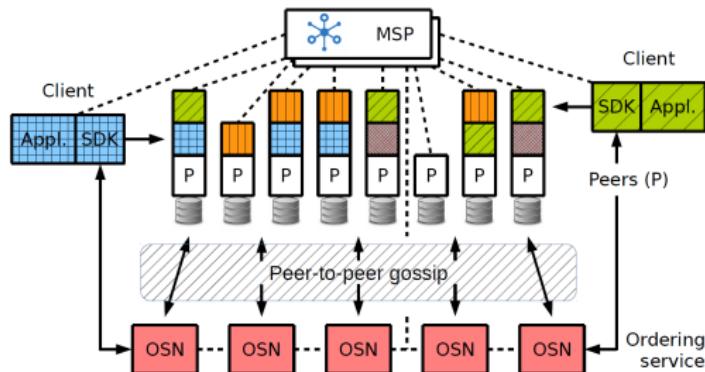


- For permissioned blockchains
- Support for pluggable consensus protocols
- Does not require a native cryptocurrency
 - No costly mining or fuel for smart contract execution
 - Solutions can be deployed with roughly the same operational cost as any other distributed system

HYPERLEDGER FABRIC - MODULARITY

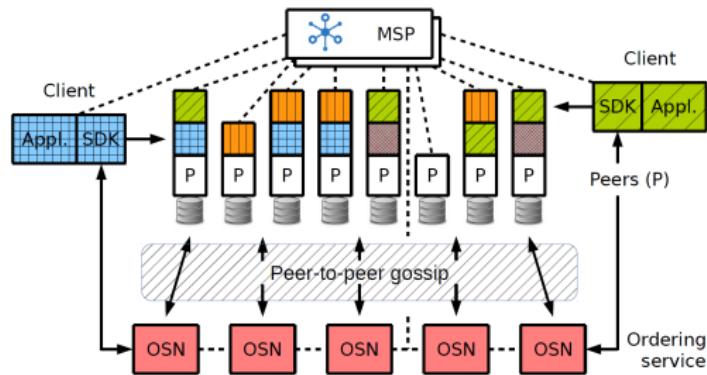


HYPERLEDGER FABRIC - MSP



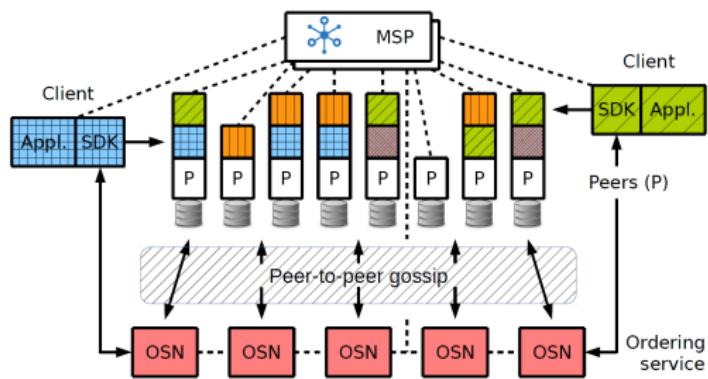
- ◊ Membership Service Provider (MSP)
- ◊ Responsable de asociar nodos con sus identidades criptograficas
- ◊ Mantiene la naturaleza permissionada de Fabric

HYPERLEDGER FABRIC - CLIENT



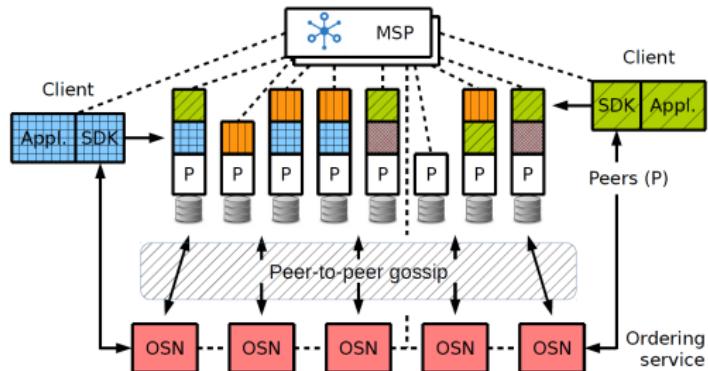
- ◊ "Usuario final"
- ◊ Aplicación del lado del cliente
- ◊ Java SDK (Python, NodeJS)

HYPERLEDGER FABRIC - CONTRATO INTELIGENTE



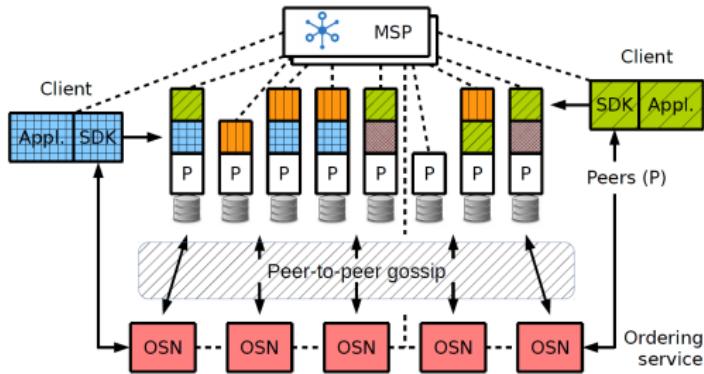
- ◊ Smart contract = Chain-code
- ◊ Aplicación distribuida
- ◊ Requiere ser instalado en los peers
- ◊ Tiene un determinado ciclo de vida
- ◊ Golang, Java, NodeJS

HYPERLEDGER FABRIC - PEERS



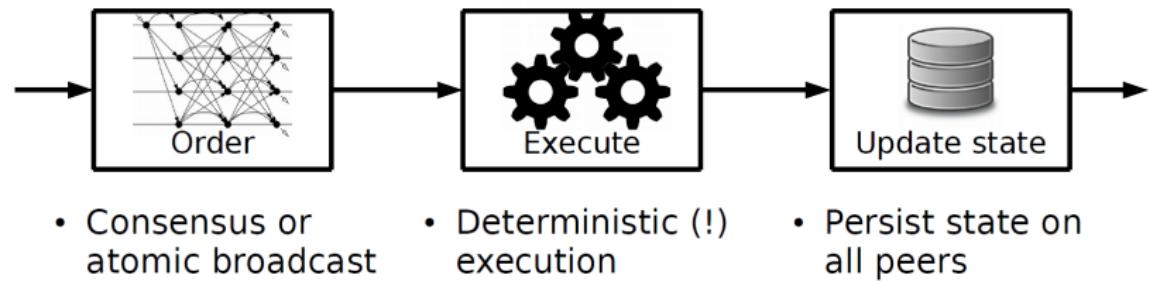
- ◊ Peers están agrupados en organizaciones
- ◊ Ejecutan* y validan propuestas de transacciones
- ◊ Mantienen copias de la blockchain / ledger

HYPERLEDGER FABRIC - ORDERING SERVICE

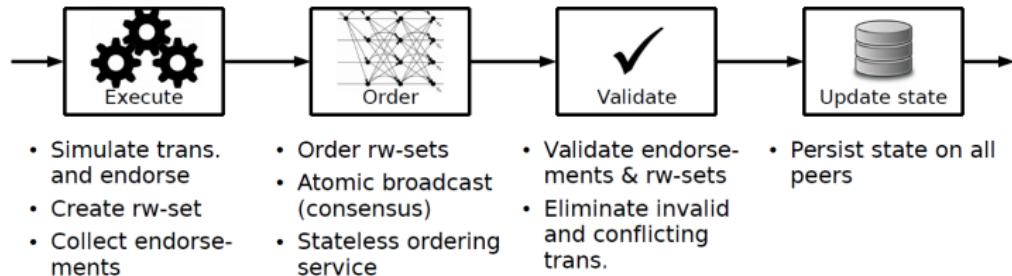


- ◊ Organizaciones (pueden ser las mismas de algunos peers) encargadas de llegar a consenso en el orden de las transacciones
- ◊ Diseminan transacciones apoyadas por nodos específicos
- ◊ Solo

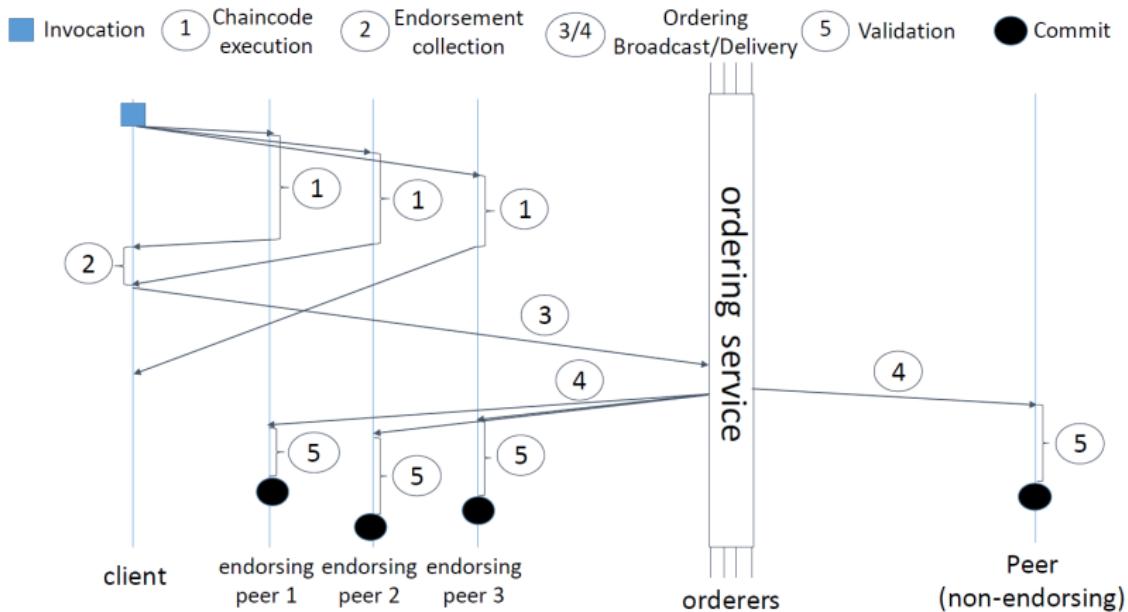
HYPERLEDGER FABRIC: ARQUITECTURA ORDER - EXECUTE



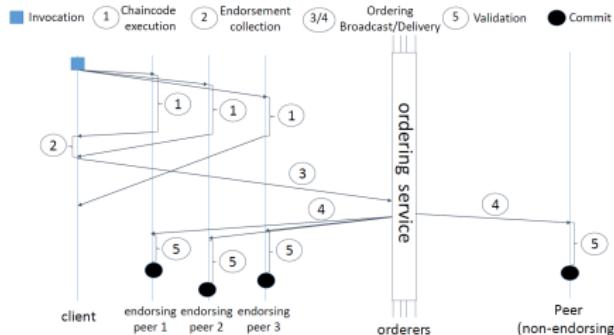
HYPERLEDGER FABRIC: ARQUITECTURA EXECUTE - ORDER - VALIDATE



HYPERLEDGER FABRIC



HYPERLEDGER FABRIC



- ◊ Flexibilidad en cuanto al número de "endorsing peers"
- ◊ Plug & Play ordering service ... adaptable a distintos casos

HYPERLEDGER FABRIC

Ejemplo básico

```
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$ ./network.sh up
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypto from 'cryptogen'
LOCAL_VERSION=2.3.1
DOCKER_IMAGE_VERSION=2.3.1
/home/hyperledger/taller/fabric-samples/test-network/../bin/cryptogen
Generating certificates using cryptogen tool
Creating Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
org1.example.com
+ res=0
Creating Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.example.com
+ res=0
Creating Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generating CCP files for Org1 and Org2
Creating network "fabric_test" with the default driver
Creating volume "docker_orderer.example.com" with default driver
Creating volume "docker_peer0.org1.example.com" with default driver
Creating volume "docker_peer0.org2.example.com" with default driver
Creating orderer.example.com ... done
Creating peer0.org1.example.com ... done
Creating peer0.org2.example.com ... done
Creating cli ... done
CONTAINER ID IMAGE NAMES COMMAND CREATED STATUS PORTS
1973ae0ba4e8 hyperledger/fabric-tools:latest "bin/bash" 1 second ago Up Less than a second
cli
3bdxfc004b29a hyperledger/fabric-peer:latest "peer node start" 2 seconds ago Up Less than a second 7051/tcp, 0.0.0.0:9051->7051/tcp
1->0$1/tcp peer0.org2.example.com
6dd1441d9360 hyperledger/fabric-peer:latest "peer node start" 2 seconds ago Up Less than a second 0.0.0.0:7051->7051/tcp
p peer0.org1.example.com
517ae0823a26 hyperledger/fabric-orderer:latest "orderer" 2 seconds ago Up Less than a second 0.0.0.0:7050->7050/tcp
p, 0.0.0.0:7053->7053/tcp orderer.example.com
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$ docker network ls
NETWORK ID NAME DRIVER SCOPE
4502f913824b bridge bridge local
29dd63eb4c92 fabric_test bridge local
a13d20be2a56 host host local
8c1b57a5d7ab none null local
hyperledger@hyperledger-VirtualBox:~/taller/fabric-samples/test-network$
```

HYPERLEDGER FABRIC - A DETALLE

- Elementos clave de Hyperledger Fabric

HYPERLEDGER FABRIC

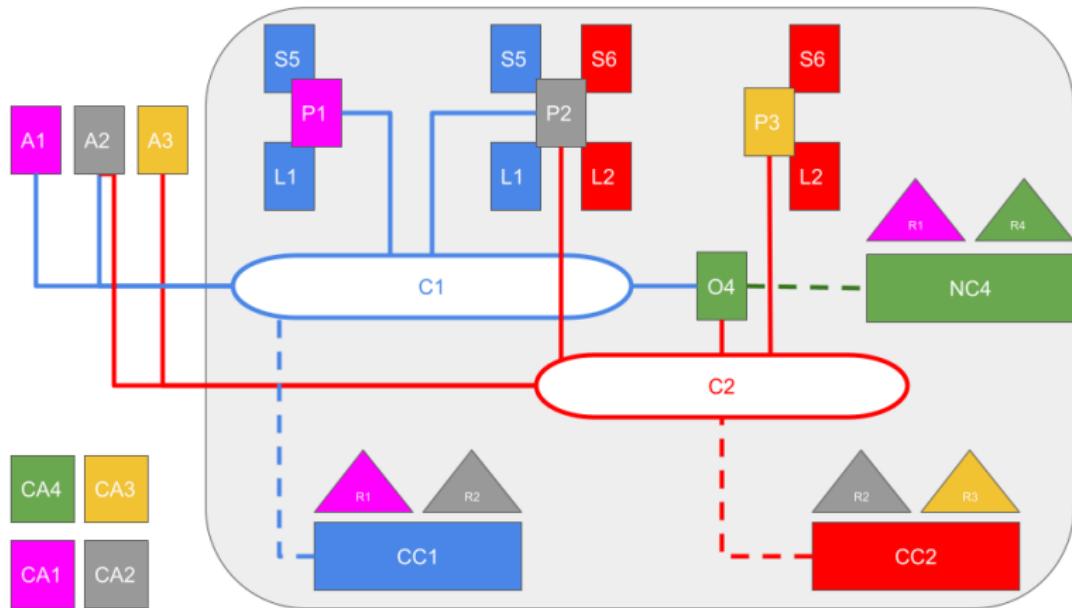


FIGURE: Ejemplo de red Blockchain y sus participantes. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Organizations.

Una organización es un ente que decide participar en la creación de la Blockchain mediante el uso de uno o varios peers, y que además procura su uso.



FIGURE: Como identificar a una Organización en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Applications. Una aplicación es aquella que nos permite interactuar con los registros que llevan el control de las transacciones hechas en la Blockchain, o agregar una nueva transacción. Es decir, permiten su consulta y actualización, pero de tal forma que la actualización pueda ser validada por toda la red Blockchain.



FIGURE: Como identificar a una Aplicación en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Channels

Un canal es el medio en donde se hacen los respectivos intercambios en una red

En Fabric un canal es sinónimo de una Blockchain, Ledger o registro distribuido

Como es de esperarse una red de nodos puede mantener más de un canal, i.e. más de una blockchain

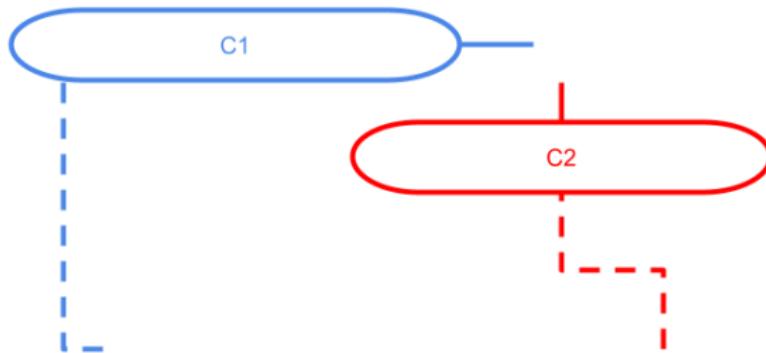


FIGURE: Como identificar a un Canal en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Channels

Dependiendo de la naturaleza con la que se busquen comunicar dichos elementos, los canales pueden ser de libre acceso (los usan todos los elementos de la red Blockchain) o de acceso privado (solo participan elementos autorizados en la red Blockchain).

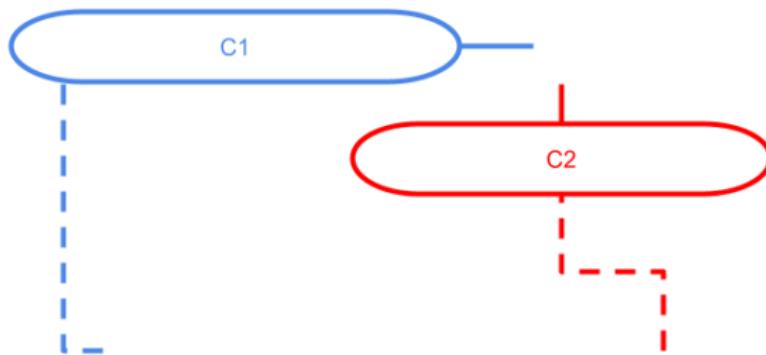


FIGURE: Como identificar a un Canal en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Private data.

Es evidente que algunas organizaciones desean que ciertos datos permanezcan **confidenciales**, por lo que requieren que algunos canales puedan proveer de características que ayuden a que dichos datos permanezcan de esa forma. En la práctica, conviene crear otro canal que contenga reglas mas estrictas de acceso.

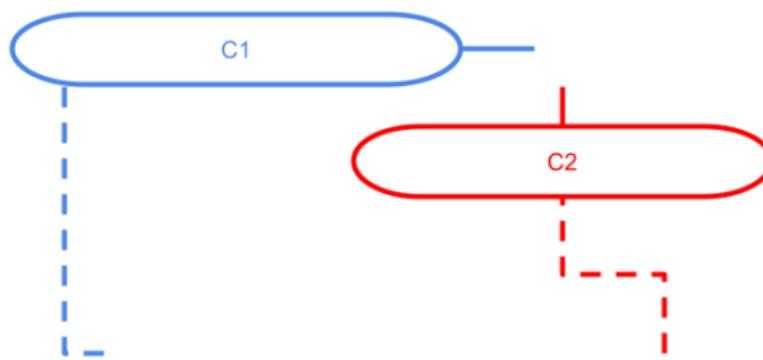


FIGURE: Como identificar a un Canal en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Peer nodes

Un peer node es donde se almacenan los registros de las transacciones que han existido en la Blockchain y a su vez son los que autorizan la actualización de las mismas mediante la anexión de nuevos bloques con el uso de protocolos de consenso.



FIGURE: Como identificar a un Peer en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Peer nodes

En el caso de Blockchain, el protocolo de consenso es Practical Byzantine Fault Tolerance (PBFT).



FIGURE: Como identificar a un Peer en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Ledgers

El ledger (L) es el registro donde se lleva el control de todas las transacciones confirmadas en la Blockchain. Todos los nodos tienen una copia de ese registro, el cual debe tener como propiedades el ser inmune a ser falsificado o alterado para que cada peer de la red Blockchain pueda validarla.



FIGURE: Como identificar a un Ledger en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Ledgers

Es sobre dicho registro donde se van a anexar los bloques nuevos que la Blockchain vaya generando conforme nuevas transacciones se vayan efectuando y validando, y el cual no puede cambiar al momento de validarse.



FIGURE: Como identificar a un Ledger en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Network configurations

La configuración de una **red** está relacionada con las organizaciones que participan y las reglas a seguir para participar en dicha **red**, entre ellas la admisión.



FIGURE: Como identificar a un Network Configuration en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Channel configuration.

La configuración de un **canal** está relacionada con las organizaciones que participan y las reglas a seguir para participar en dicho **canal**, entre ellas la admisión. Posee las siguientes propiedades:

- Versioned. Referente a la versión de la configuración.
- Permissioned. Referente a las políticas de configuración (permisos de modificación)
- Hierarchical. Las políticas pueden ser jerárquicas en base a los niveles existentes.



FIGURE: Como identificar a un Channel Configuration en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Ordering services.

Se encarga de la transmisión de las actualizaciones de los estados en una red Blockchain entre cada uno de sus nodos respectivos (peers), de establecer el consenso entre el orden de las transacciones hechas allí mismo, incluida su autenticidad. Y también se encarga de la reconfiguración del canal usado y del acceso al mismo. Todas estas acciones están registradas de tal manera que puedan reconstruir el orden de las mismas en la forma que se llevaron a cabo en la red, dado que los peers no están sincronizados.

O4

FIGURE: Como identificar a un Ordering Service en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

Identity.

En el mundo real, todos tenemos una identidad que nos hace visibles respecto al resto. Dicha identidad está respaldada por un documento oficial (pasaporte) emitido por un gobierno, la cual nos permite autenticarnos ante quien lo requiera.

Identity. En el mundo de Blockchain, dicha identidad es un certificado de autoridad que respalda dicha identidad. Y que es emitida por una autoridad de certificación reconocida (idealmente). Y que además ayuda a verificar a qué se tiene derecho en la red (los permisos).

Certificates of authority. Un certificado de autoridad (*CA*) es un documento electrónico emitido por una llamada **Autoridad de Certificación** en formato X.509, que avala que una organización realmente es y existe.

Certificates of authority. Un análogo en el mundo real es el pasaporte. Los datos que vienen allí pueden ser vistos por quien los quiere consultar, y hay una autoridad que ha certificado que esos datos son correctos en base a lo que la persona les ha entregado -con un comprobante de domicilio- y/o lo que la propia autoridad tiene registrada (en México vía el RENAPO).

Certificates of authority. Las partes que componen un certificado de autoridad son las siguientes

- Datos de la organización.
- Datos de la autoridad de certificación (CA).
- Número de serie del certificado, versión del estándar X.509 y algoritmo de firma.
- Inicio y expiración del certificado.
- Clave pública, firma e información asociada.
- Hash del certificado.

Mas detalles técnicos los podemos encontrar en el RFC 5280.

HYPERLEDGER FABRIC

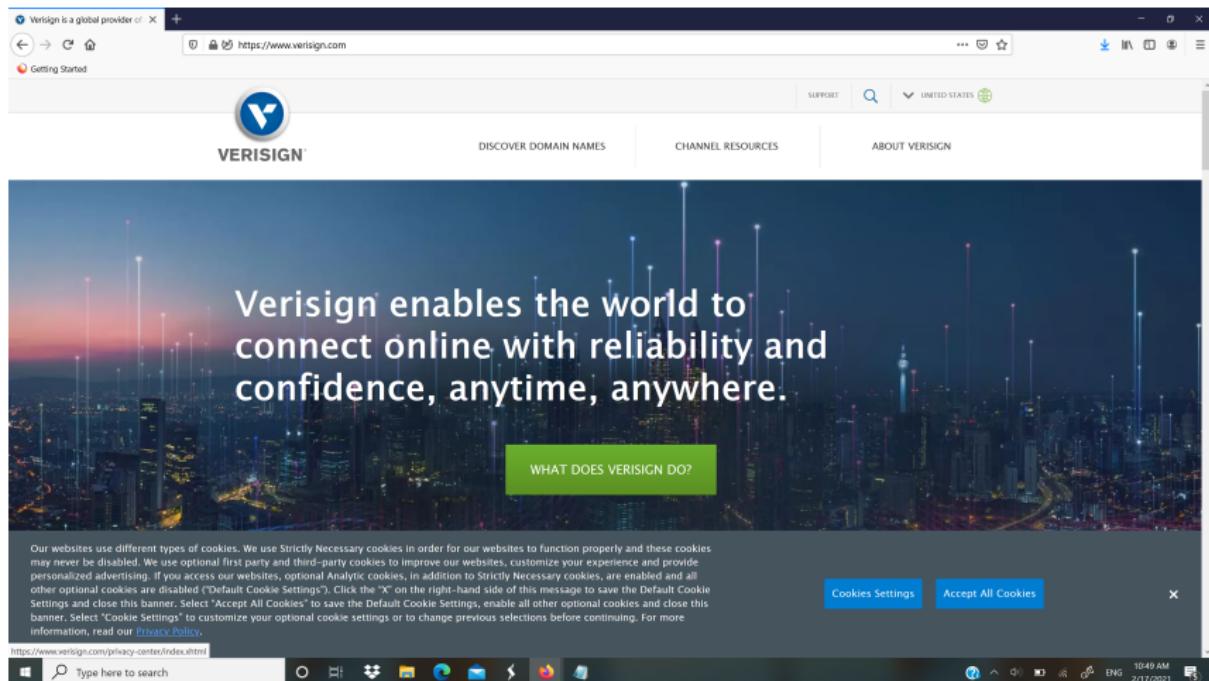


FIGURE: Página de Verisign.

HYPERLEDGER FABRIC

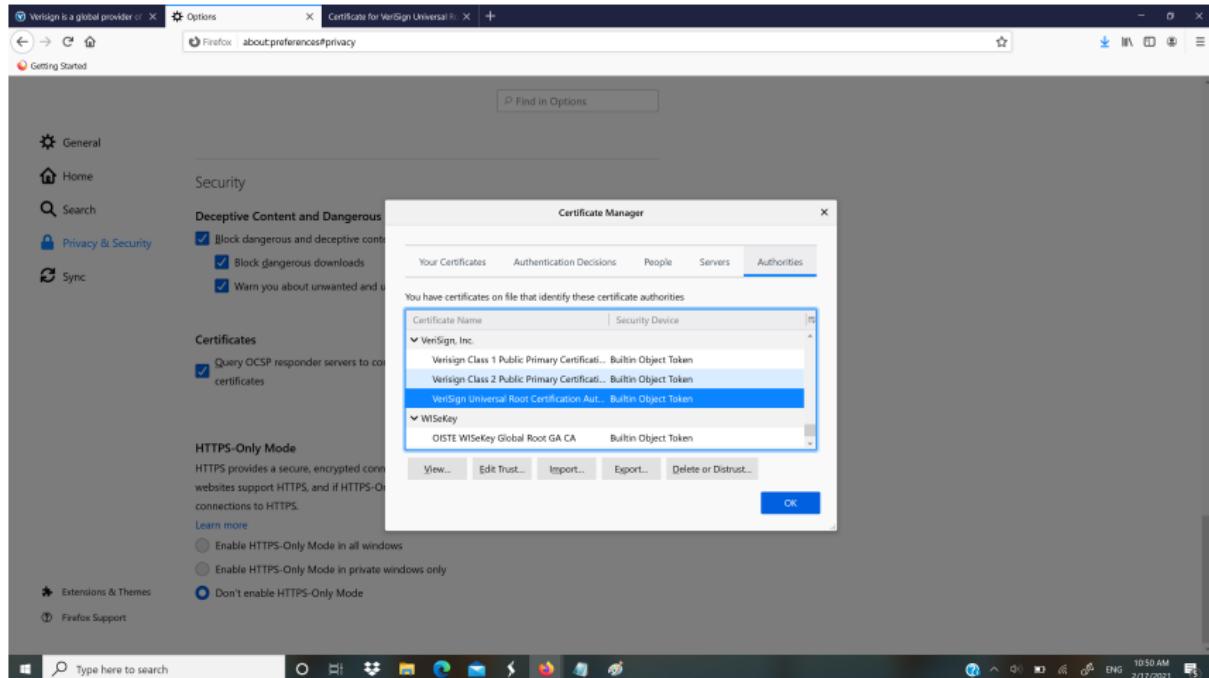


FIGURE: Cómo almacena Mozilla un certificado.

HYPERLEDGER FABRIC

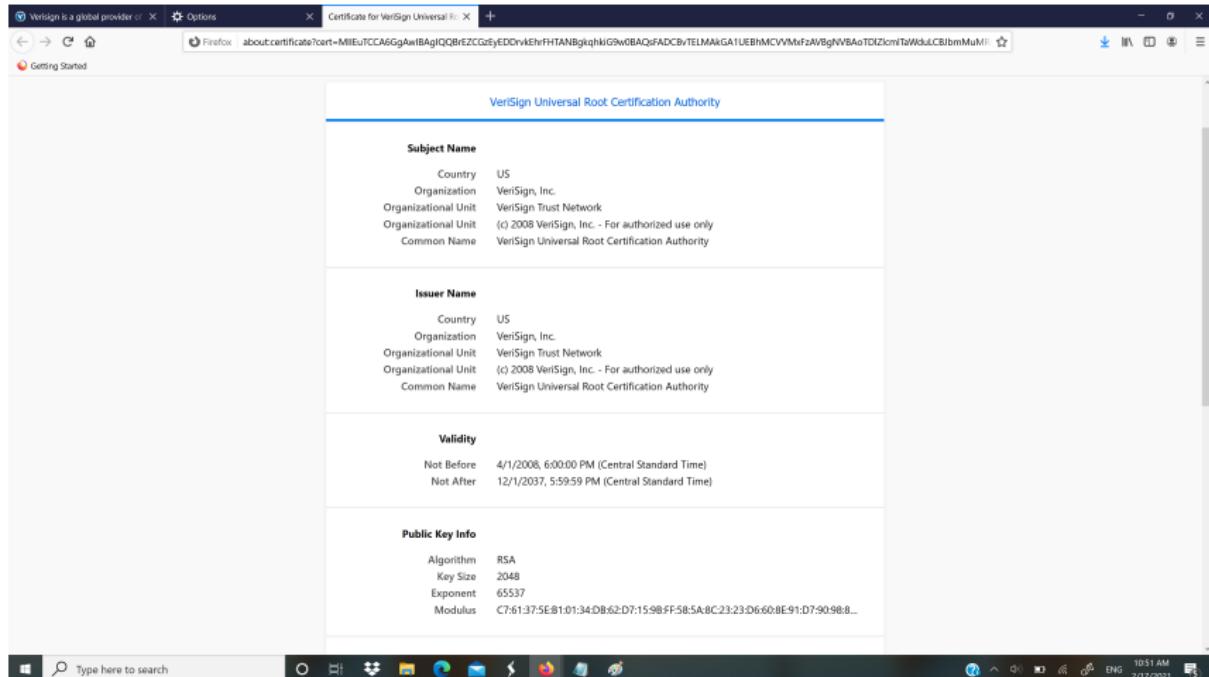


FIGURE: Partes de un certificado. Parte 1.

HYPERLEDGER FABRIC

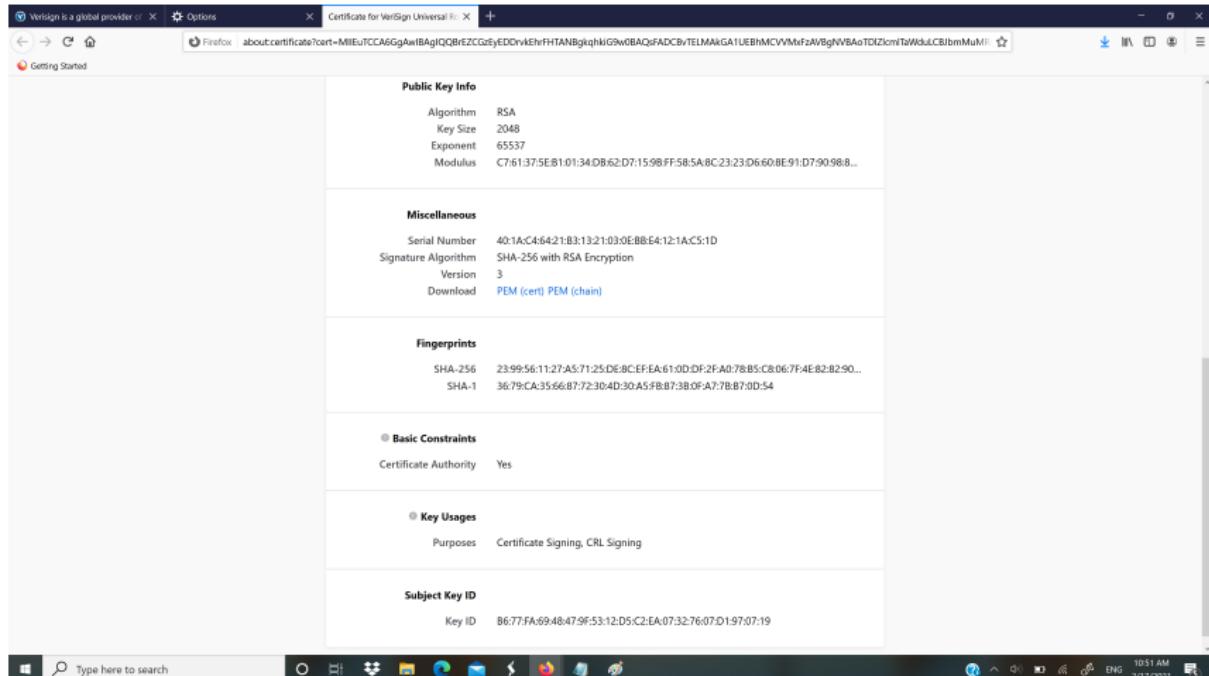


FIGURE: Partes de un certificado. Parte 2.

HYPERLEDGER FABRIC

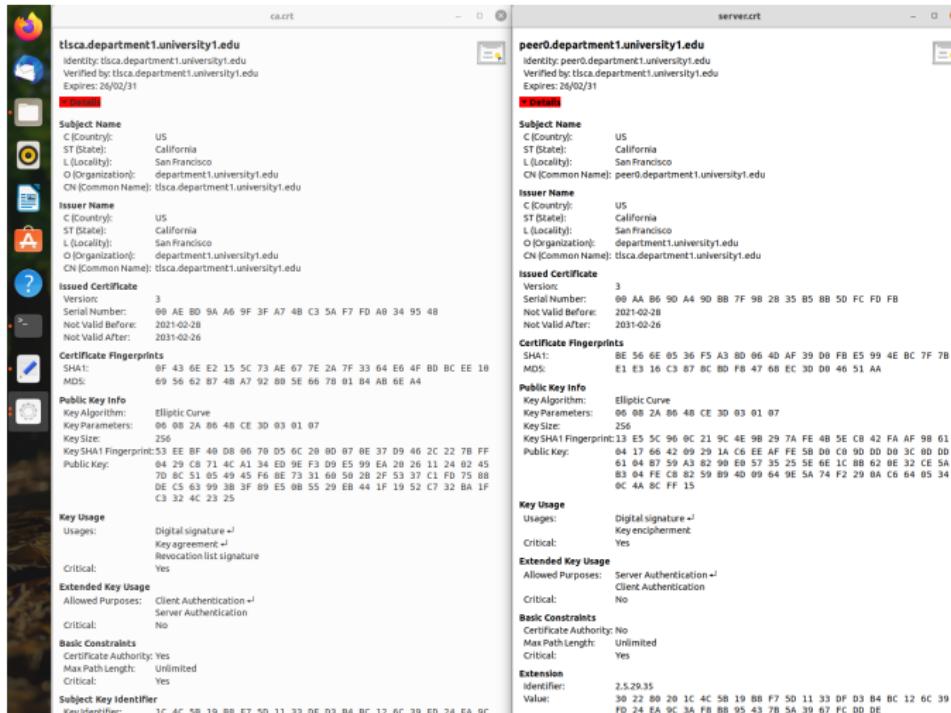


FIGURE: Partes de un certificado Blockchain. Parte 1.

HYPERLEDGER FABRIC

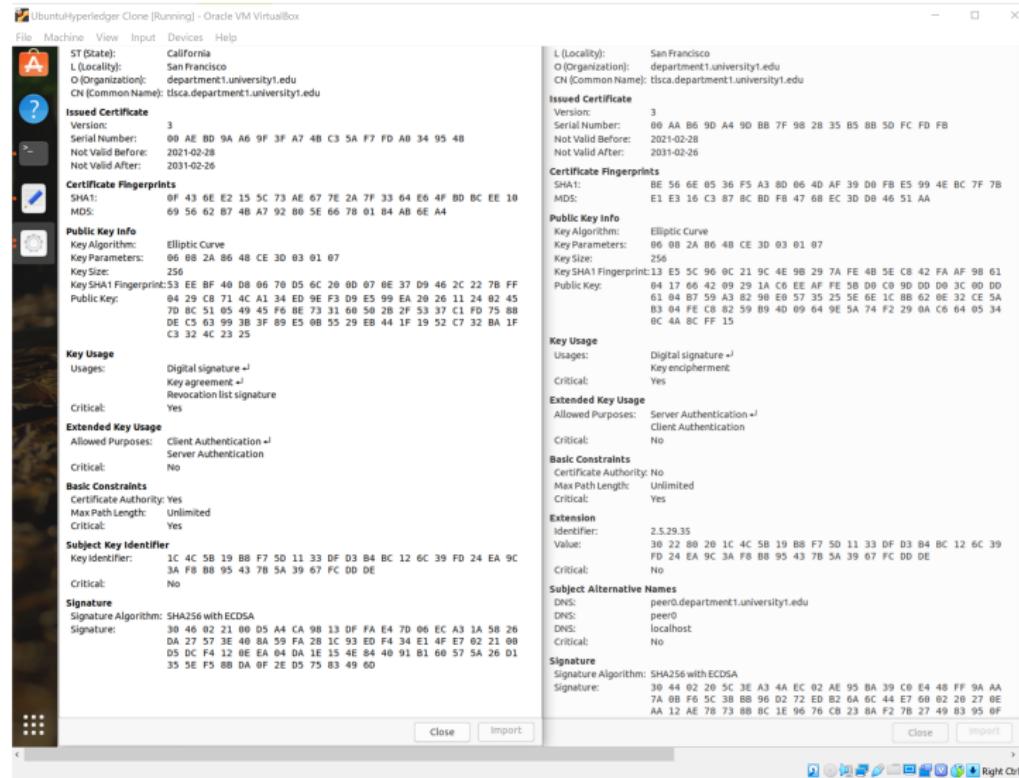


FIGURE: Partes de un certificado Blockchain. Parte 2.

HYPERLEDGER FABRIC

Se puede formar parte de algo conocido como **Chain of trust**. Un ejemplo de una **Chain of trust** puede ser

- La organización X se firma a sí misma y se constituye en una CA.
- La organización X firma a una organización Y. Por lo tanto a Y lo certifica X.
- La organización Y firma a una organización Z. Por lo tanto a Z la certifica Y.
- La organización Z firma a una organización U. Por lo tanto a U la certifica Z.

HYPERLEDGER FABRIC

También considerar que

- Un certificado pierde su vigencia.
- Un certificado puede ser revocado.
- Por lo general, cuando se emite un certificado, se verifica quien lo está pidiendo -de hecho, tarda un tiempo en emitirse-, y por lo general se paga dependiendo del tipo de certificado pedido.

HYPERLEDGER FABRIC

También considerar que

- Un certificado puede ser revocado a petición de la organización que lo posea.
- Un certificado puede ser revocado si se ha infringido una política.

HYPERLEDGER FABRIC

Es decir, no sólo se validan las firmas digitales hechas por la organización, sino que a su vez dichas firmas están respaldadas por un ente de confianza (idealmente) al validar la clave pública de la organización mediante el certificado respectivo.



FIGURE: Como identificar a una CA en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Certificates of authority. En una red Blockchain, cada miembro de la misma (orderers, peers, aplicaciones, etc.) tiene una identidad digital, avalada por una autoridad de certificación, lo que permite respaldar quien dice ser.



FIGURE: Como identificar a una CA en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Smart contracts. En el mundo real, un contrato es aquel que celebra un acuerdo de intercambio de productos y/o servicios entre 2 o mas entidades.



FIGURE: Como identificar a un Smart Contract en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Smart contracts. En Blockchain, es una transacción hecha en la red, la cual queda registrada dentro de ella como un bloque, el cual debe ser validado por todos los participantes de la misma antes de ser anexado al ledger.



FIGURE: Como identificar a un Smart Contract en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Smart contracts. Naturalmente, ese bloque debe quedar distribuido en todos los nodos de la red.



FIGURE: Como identificar a un Smart Contract en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Smart contracts. Se requiere el uso de lo que se conoce como **chaincode**, por lo que ambos términos son considerados sinónimos en la práctica. Implementa la lógica de la aplicación y también corre en fase de ejecución. Un chaincode es invocado por una aplicación que desee hacer una consulta o una actualización sobre el ledger sin tener acceso al al estado del mismo. Dicho chaincode es escrito en el lenguaje Golang, Java o NodeJS.



FIGURE: Como identificar a un Smart Contract en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

HYPERLEDGER FABRIC

Smart contracts. En términos técnicos, el contrato inteligente define la lógica de una transacción, y donde posteriormente es empacada en un chaincode para ser distribuida en la red Blockchain. Es decir, el smart contract contiene todos los pasos que crearon dicha transacción, y la chaincode permite codificar esa transacción de tal forma que pueda ser entendida por los peers de la red al momento de distribuirse en ella y ser registrada en el ledger.



FIGURE: Como identificar a un Smart Contract en una Blockchain. Figura basada en [hyperledger-fabricdocs Documentation](#).

Assets. Este término proviene del mundo financiero y hace referencia a todo aquello que tenga un valor monetario, independientemente si sea o no tangible (puede ser desde un commodity -un energético como el petróleo o el gas-, divisas, alimentos, o acciones -ya sea físicas o virtuales- que coticen en el sector financiero, por ejemplo en el Dow Jones o en el NASDAQ -la bolsa de valores de Nueva York-). Por lo tanto, el ledger (el registro de todas las transacciones validadas) se convierte en un **activo**.

HANDS ON

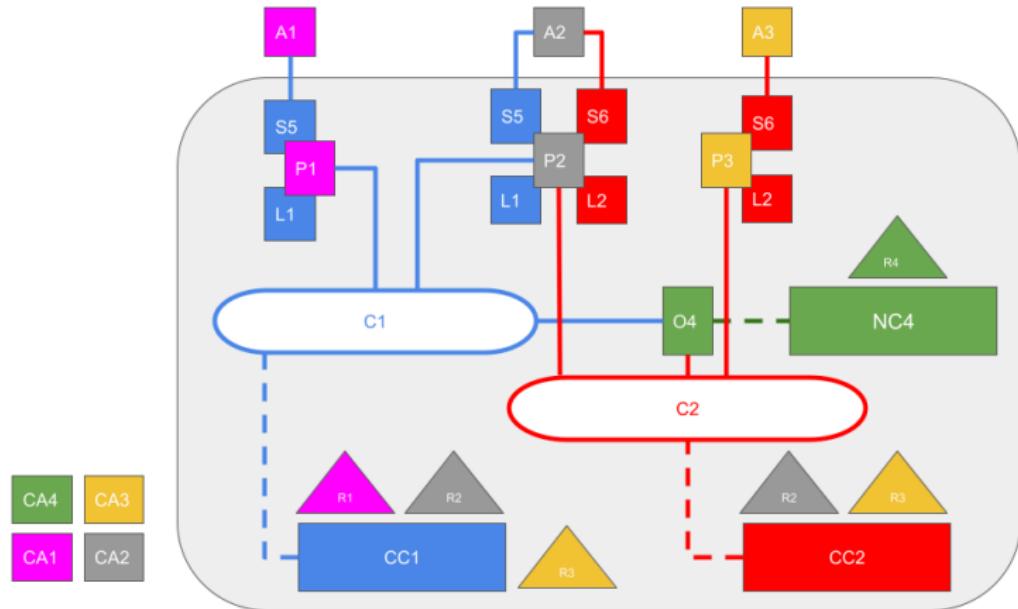


FIGURE: Se repite el ejemplo para referencia pero con puntuaciones. Figura basada en [hyperledger-fabricdocs Documentation](#).

HANDS ON

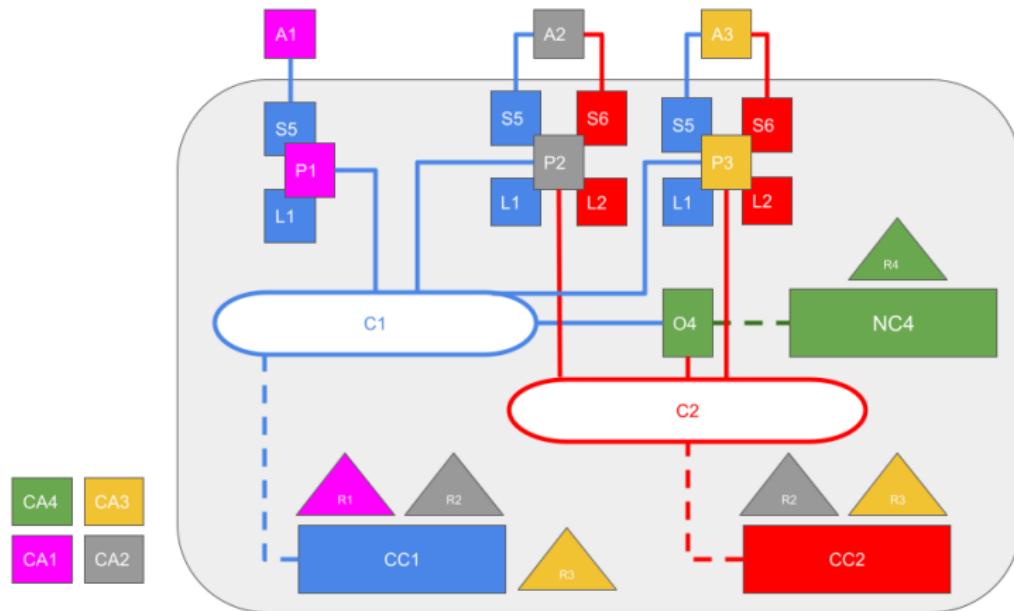


FIGURE: Nuestra red, hacer 2 papers con 3 autores (íconos basados en el manual de Hyperledger Fabric, [hyperledger-fabricdocs Documentation](#).)

FUENTES Y MATERIAL EXTRA

Criptografía (Clave pública y privada, firmas digitales, certificados).

- MENEZES, Alfred J.; VAN OORSCHOT, Paul C.; VANSTONE, Scott A. *Handbook of applied cryptography*. CRC press, 2018.
- Adams, C.; Lloyd, S. (2003). *Understanding PKI: concepts, standards, and deployment considerations*. Addison-Wesley Professional.
- Gutmann, P. (2002). PKI: it's not dead, just resting. *Computer*, 35(8), 41-49.
- 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.
- HANKERSON, Darrel; MENEZES, Alfred J.; VANSTONE, Scott. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

FUENTES Y MATERIAL EXTRA

Blockchain

- ANDONI, Merlinda, 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.
- DOCUMENTATION, Hyperledger-Fabricdocs. Release master. 2020.
- 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/>

- Red blockchain para un escenario simple
- Tres organizaciones
- Dos canales
- https://github.com/Cocoa-ci/taller_blockchain/tree/main/redes/redprueba
- Cambiar permisos a scripts sh

HANDS ON

- Algunos de los archivos a ocupar.

Archivo crypto-config.yaml.

- Una vista de alto nivel de la red
- Define el material criptográfico que debe generarse
- Se utiliza como entrada para el programa cryptogen

HANDS ON

Archivo crypto-config.yaml.

```
! crypto-config.yaml
9  # Modification done by: AFMH
10 #
11 #
12 #
13 # "OrdererOrgs" - Definition of organizations managing orderer nodes
14 #
15 OrdererOrgs:
16 #
17 # Orderer
18 #
19 - Name: Orderer
20 Domain: licore.com
21 EnableNodeOUs: true
22 #
23 # "Specs"
24 #
25 Specs:
26 | - Hostname: orderer
27 #
28 # "PeerOrgs" - Definition of organizations managing peer nodes
29 #
30 PeerOrgs:
31 #
32 # Org1 - Gobierno - government
33 #
34 - Name: Org1
35 Domain: department1.universityl.edu
36 EnableNodeOUs: true
37 #
38 # "Specs"
39 Template:
40 | Count: 1 # Numero de peers
41 #
42 # "Users"
```

HANDS ON

Estructura de directorios.

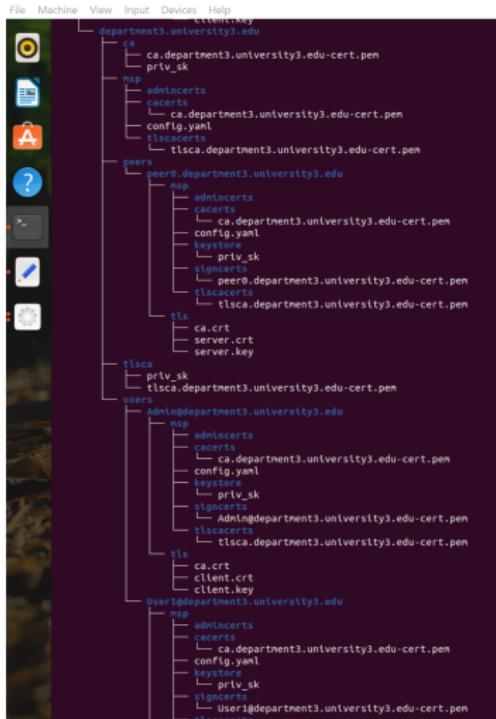


FIGURE: tree crypto-config

Archivo docker-compose-ca.yaml.

- Define la funcionalidad que debe proporcionar la autoridad certificadora (CA - Certification Authority)
- Tiene las direcciones de las carpetas donde se almacenan los certificados para cada organización (por ahora, todo está en un solo equipo).
- Cada certificado está en formato X.509 (Formato .pem, visto en las filminas anteriores).
- Usa el protocolo TLS.

HANDS ON

docker-compose-ca.yaml

```
Open | New | Help | Docker Compose
```

docker-compose-ca.yaml docker-compose-ca.yaml docker-compose-ca.yaml docker-compose-ca.yaml

```
Save | Exit
```

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#
# Project title: Red Frodo
# Project URL: https://github.com/redfrodo/ca-saml
# File creation date: February 27, 2021
# File last update date: February 27, 2021
# Modifications last update: None
#
# RFR
#
# version: "2"
#
# networks:
#   mynet:
#
# services:
#   ca:
#     image: hyperledger/fabric-ca:1.4.0
#     environment:
#       - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
#       - FABRIC_CA_SERVER_TLS_CERTIFICATE=/etc/hyperledger/fabric-ca-server-config/ca.department1.university1.edu-cert.pem
#       - FABRIC_CA_SERVER_TLS_PRIVATE_KEY=/etc/hyperledger/fabric-ca-server-config/ca.department1.university1.edu-cert.key
#       - FABRIC_CA_SERVER_PORT=7054
#     ports:
#       - 7054:7054
#     command: /bin/sh -c "fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-config/ca.department1.university1.edu-cert.pem --ca.keyfile /etc/hyperledger/fabric-ca-server-config/ca.department1.university1.edu-cert.key" & sleep infinity"
#     volumes:
#       - ./ca-server-config:/etc/hyperledger/fabric-ca-server-config
#     container_name: ca_peer0Org1
#     networks:
#       - mynet
#
# cat:
#   image: hyperledger/fabric-ca:1.4.0
#   environment:
#     - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
#     - FABRIC_CA_SERVER_TLS_CERTIFICATE=/etc/hyperledger/fabric-ca-server-config/ca.department2.university2.edu-cert.pem
#     - FABRIC_CA_SERVER_TLS_PRIVATE_KEY=/etc/hyperledger/fabric-ca-server-config/ca.department2.university2.edu-cert.key
#     - FABRIC_CA_SERVER_PORT=7054
#   ports:
#     - 7054:7054
#   command: /bin/sh -c "fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-config/ca.department2.university2.edu-cert.pem --ca.keyfile /etc/hyperledger/fabric-ca-server-config/ca.department2.university2.edu-cert.key" & sleep infinity"
#   volumes:
#     - ./ca-server-config:/etc/hyperledger/fabric-ca-server-config
#     - ./ca-peer0Org1:/etc/hyperledger/fabric-ca-server-config
#     container_name: ca_peer0Org2
#     networks:
#       - mynet
#
# ca0:
#   image: hyperledger/fabric-ca:1.4.0
```

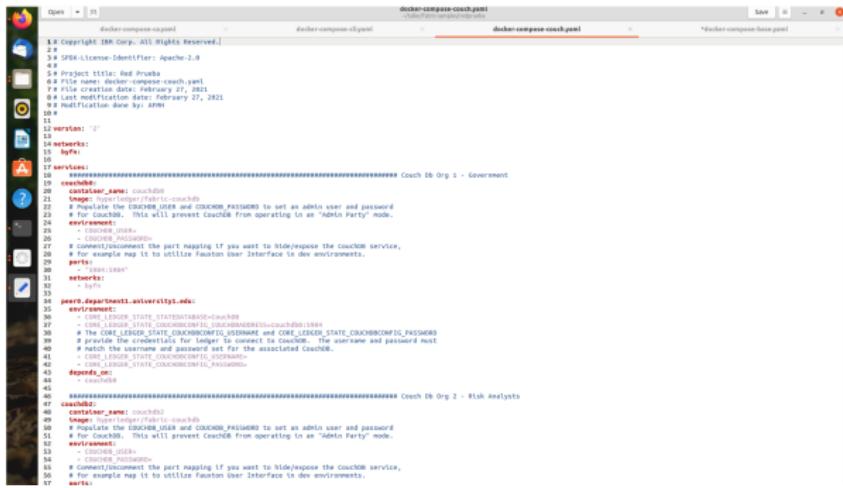
FIGURE: Ejemplo docker-compose-ca.yaml

Archivo docker-compose-couch.yaml.

- Define la funcionalidad para el almacenamiento del ledger en cada peer
- Permite mantener el ledger. Por ejemplo, su actualización.

HANDS ON

docker-compose-couch.yaml.



```
Open ⌂ ⌂ ⌂
docker-compose-couch.yaml docker-compose-couch.yaml docker-compose-couch.yaml *docker-compose-couch.yaml
1 # Copyright IBM Corp. All Rights Reserved.
2 #
3 # SPDX-License-Identifier: Apache-2.0
4 #
5 # Project title: Red Prueba
6 # File name: docker-compose-couch.yaml
7 # File creation date: February 17, 2021
8 # Last modification date: February 27, 2021
9 # Last modification done by: aemn
10 #
11
12 version: '3'
13
14 networks:
15   myNet:
16
17 services:
18
19   couchdb:
20     container_name: couchdb
21     image: hyperledger/couchdb:latest
22     environment:
23       - COUCHDB_USER=couchdb
24       - COUCHDB_PASSWORD=couchdb
25     ports:
26       - "5984:5984"
27     networks:
28       - myNet
29
30   peer1.department11.universityt.edu:
31     environment:
32       - CORE_LEDGER_STATE_STATEDATABASE=couchdb
33       - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=core_dbuser
34       - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=core_dbuser
35     depends_on:
36       - couchdb
37
38   peer2.department12.universityt.edu:
39     environment:
40       - CORE_LEDGER_STATE_STATEDATABASE=couchdb
41       - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=core_dbuser
42       - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=core_dbuser
43     depends_on:
44       - couchdb
45
46
47   database:
48     container_name: database
49     image: hyperledger/fabric-couchdb
50     environment:
51       - COUCHDB_USER=couchdb
52       - COUCHDB_PASSWORD=couchdb
53     ports:
54       - "5984:5984"
55     networks:
56       - myNet
57     depends_on:
58       - couchdb
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
138
139
139
140
141
142
143
144
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
217
218
219
219
220
221
222
223
224
225
226
227
227
228
229
229
230
231
232
233
234
235
236
237
237
238
239
239
240
241
242
243
244
245
246
247
247
248
249
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
267
268
269
269
270
271
272
273
274
275
276
277
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
317
318
319
319
320
321
322
323
324
325
326
327
327
328
329
329
330
331
332
333
334
335
336
337
337
338
339
339
340
341
342
343
344
345
346
347
347
348
349
349
350
351
352
353
354
355
356
357
357
358
359
359
360
361
362
363
364
365
366
366
367
368
368
369
369
370
371
372
373
374
375
376
377
377
378
379
379
380
381
382
383
384
385
386
387
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
407
408
409
409
410
411
412
413
414
415
415
416
417
417
418
419
419
420
421
422
423
424
425
426
426
427
428
428
429
429
430
431
432
433
434
435
436
436
437
438
438
439
439
440
441
442
443
444
445
446
446
447
448
448
449
449
450
451
452
453
454
455
456
456
457
458
458
459
459
460
461
462
463
464
465
466
466
467
468
468
469
469
470
471
472
473
474
475
476
476
477
478
478
479
479
480
481
482
483
484
485
486
486
487
488
488
489
489
490
491
492
493
494
495
496
496
497
498
498
499
499
500
501
502
503
504
505
505
506
507
507
508
508
509
509
510
511
512
513
514
515
515
516
517
517
518
518
519
519
520
521
522
523
524
525
525
526
527
527
528
528
529
529
530
531
532
533
534
535
535
536
537
537
538
538
539
539
540
541
542
543
544
545
545
546
547
547
548
548
549
549
550
551
552
553
554
555
555
556
557
557
558
558
559
559
560
561
562
563
564
565
565
566
567
567
568
568
569
569
570
571
572
573
574
575
575
576
577
577
578
578
579
579
580
581
582
583
584
585
585
586
587
587
588
588
589
589
590
591
592
593
594
595
595
596
597
597
598
598
599
599
600
601
602
603
604
604
605
606
606
607
607
608
608
609
609
610
611
612
613
614
614
615
616
616
617
617
618
618
619
619
620
621
622
623
624
624
625
626
626
627
627
628
628
629
629
630
631
632
633
634
635
635
636
637
637
638
638
639
639
640
641
642
643
644
644
645
646
646
647
647
648
648
649
649
650
651
652
653
654
654
655
656
656
657
657
658
658
659
659
660
661
662
663
664
664
665
666
666
667
667
668
668
669
669
670
671
672
673
674
674
675
676
676
677
677
678
678
679
679
680
681
682
683
684
684
685
686
686
687
687
688
688
689
689
690
691
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
701
702
703
703
704
705
705
706
706
707
707
708
708
709
709
710
711
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
721
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
731
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
741
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
751
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
761
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
771
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
781
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
791
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
801
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
811
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
821
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
831
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
841
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
851
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
861
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
871
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
881
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
891
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
911
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
921
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
931
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
941
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
951
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
961
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
971
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
981
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
991
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1011
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1021
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1031
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1041
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1051
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1061
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1071
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1141
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1161
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1381
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1411
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1421
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1431
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1441
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1451
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1461
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1471
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1481
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1511
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1521
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1531
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1541
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1551
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1561
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1571
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1611
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1621
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1631
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1641
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1651
1652
1653
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1661
1662
1663
1663
1664
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1671
1672
1673
1673
1674
1674
1675
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1683
1684
1684
1685
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1693
1694
1694
1695
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1703
1704
1704
1705
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1711
1712
1713
1713
1714
1714
1715
1715
1716
1716
1717
1717
1718
1718
1719
1719
1720
1721
1722
1723
1723
1724
1724
1725
1725
1726
1726
1727
1727
1728
1728
1729
1729
1730
1731
1732
1733
1733
1734
1734
1735
1735
1736
1736
1737
1737
1738
1738
1739
1739
1740
1741
1742
1743
1743
1744
1744
1745
1745
1746
1746
1747
1747
1748
1748
1749
1749
1750
1751
1752
1753
1753
1754
1754
1755
1755
1756
1756
```

HANDS ON

Archivo docker-compose-base.yaml.

- Se configuran los puertos de cada peer y del orderer (Por ahora será así ya que se montará de manera local. Es posible configurar una IP para cada elemento).
- Se configuran las carpetas correspondientes al Membership Service Provider (MSP).
- El certificado que permite verificar su identidad a cada elemento de la red y el certificado de la CA que soporta la validez de la identidad a cada uno.

HANDS ON

docker-compose-base.yaml

```
Open docker-compose-raymond docker-compose-client docker-compose-coach-pool *docker-compose-base-pool Save docker-compose-raymond
1 # Copyright IBM Corp. All Rights Reserved.
2 #
3 # SPDX-License-Identifier: Apache-2.0
4 #
5 # Project title: Red Phoenix
6 #
7 # File name: docker-compose-coach.yaml
8 # Last modified date: January 27, 2021
9 # Last modification done by: AFHM
10 # Modification done by: AFHM
11
12 version: "2"
13
14 services:
15
16   orderer.cleantca.edu:
17     container_name: orderer.cleantca.edu
18     environment:
19       FTLB: peer-base.json
20     services:
21       orderer-base
22     volumes:
23       - ./orderer-artifacts/genesis.block:/var/hyperledger/orderer/orderer-genesis.block
24       - ./orderer-artifacts/peer-organization/cleantca.edu/orderer/orderer_cleantca.edu:/var/hyperledger/orderer/msp
25       - ./crypto-config/ordererOrganization/cleantca.edu/orderers/orderer_cleantca.edu/tls:/var/hyperledger/orderer/tls
26       - ./crypto-config/ordererOrganization/cleantca.edu/orderers/orderer_cleantca.edu/msp:/var/hyperledger/orderer/msp
27     ports:
28       - 7050:7050
29
30 #####################################################
31 peers.department1.university.edu:
32   container_name: peer.department1.university.edu
33   environment:
34     FTLB: peer-base.json
35   services:
36     peer-base
37   volumes:
38     - CORE_PEER_ID=department1.university.edu
39     - CORE_PEER_ADDRESS=department1.university.edu:9801
40     - CORE_PEER_LISTENADDRESS=department1.university.edu:9801
41     - CORE_PEER_DISCOVERYADDRESS=department1.university.edu:9802
42     - CORE_PEER_GOSSIP_EXTERNALENDPOINT=department1.university.edu:9803
43     - CORE_PEER_GOSSIP_BOOTSTRAP=peer.department1.university.edu:9803
44     - CORE_PEER_GOSSIP_EXTERNALENDPOINT=department1.university.edu:9804
45     - CORE_PEER_MSCONFIG=group
46   version: '2'
47   ports:
48     - 7051:7051
49
50 #####################################################
51 peers.department2.university.edu:
52   container_name: peer.department2.university.edu
53   environment:
54     FTLB: peer-base.json
55   services:
56     peer-base
57   volumes:
58     - CORE_PEER_ID=department2.university.edu
59     - CORE_PEER_ADDRESS=department2.university.edu:9801
60     - CORE_PEER_LISTENADDRESS=department2.university.edu:9801
61     - CORE_PEER_DISCOVERYADDRESS=department2.university.edu:9802
62     - CORE_PEER_GOSSIP_EXTERNALENDPOINT=department2.university.edu:9803
63     - CORE_PEER_GOSSIP_BOOTSTRAP=peer.department1.university.edu:9803
64     - CORE_PEER_GOSSIP_EXTERNALENDPOINT=department2.university.edu:9804
65     - CORE_PEER_MSCONFIG=group
66   version: '2'
67   ports:
68     - 7052:7052
```

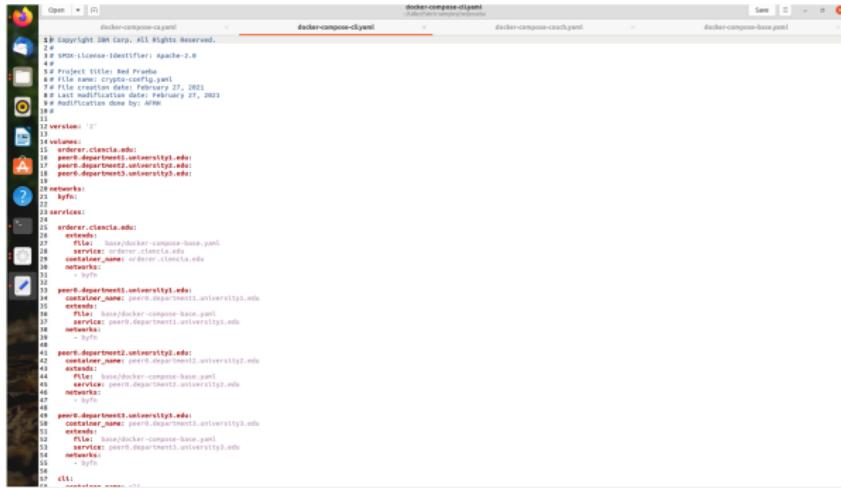
FIGURE: Ejemplo docker-compose-base.yaml

Archivo docker-compose-cli.yaml.

- Pone como punto de entrada a la blockchain a un nodo en específico. (Ejemplo: peer0.department1.university1.edu).
- Ayuda a configurar el levantamiento de cada peer de la red.

HANDS ON

docker-compose-cli.yaml.



```
# Copyright IBM Corp. All Rights Reserved.
# See License-Identifier: Apache-2.0
# Project title: Red Prairie
# File creation date: February 27, 2023
# Last modification date: February 27, 2023
# Modification done by: Arun
# id
# version: "2"
# volumes:
#   orderer.citela.edu
#     orderer.department1.university1.edu
#     peer0.department1.university1.edu
#     peer1.department1.university1.edu
#     peer2.department1.university1.edu
#   networks:
#     mynet
# services:
#   orderer.citela.edu:
#     extends:
#       file: base/docker-compose-base.yaml
#     service: orderer.citela.edu
#     container_name: orderer.citela.edu
#     networks:
#       - mynet
#     peer0.department1.university1.edu:
#     extends:
#       file: base/docker-compose-base.yaml
#       service: peer0.department1.university1.edu
#     networks:
#       - mynet
#     peer1.department2.university1.edu:
#     extends:
#       file: base/docker-compose-base.yaml
#       service: peer1.department2.university1.edu
#     networks:
#       - mynet
#     peer2.department2.university1.edu:
#     extends:
#       file: base/docker-compose-base.yaml
#       service: peer2.department2.university1.edu
#     networks:
#       - mynet
#     peer0.department3.university3.edu:
#     extends:
#       file: base/docker-compose-base.yaml
#       service: peer0.department3.university3.edu
#     networks:
#       - mynet
# cli:
```

FIGURE: Ejemplo docker-compose-cli.yaml