



# Commercial paper

## Fundamentos de blockchains

René Dávila - Jorge Solano

# Índice

## 1 Commercial paper

- PaperNet

- Crear smart contract propio (MagnetoCorp)

- Aprobar smart contract ajeno (DigiBank)

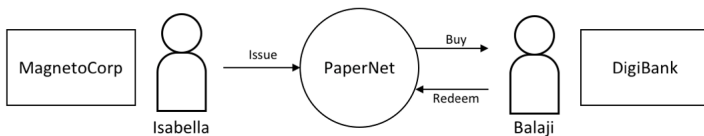
- Publicar smart contract (DigiBank)

- Emitir un papel comercial (MagnetoCorp)

- Comprar un papel comercial (DigiBank)

## 2 Referencias

Diagrama de las organizaciones:



Las organizaciones MagnetoCorp y DigiBank intercambian papeles comerciales en la red PaperNet.

# Índice

## 1 Commercial paper

### PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias

# Peers y canal de comunicación

En cd fabric-samples/test-network:

---

```
(PaperNet admin)$ ./network.sh up
```

---

```
(PaperNet admin)$ ./network.sh createChannel
```

---

# Red docker net\_test

En cd fabric-samples/commercial-paper

---

```
(PaperNet admin)$ ./network-starter.sh
```

---

```
(PaperNet admin)$ docker ps
```

---

```
(PaperNet admin)$ docker network inspect net_test
```

---

peer0.org1.example.com → DigiBank

peer0.org2.example.com → MagnetoCorp

# Monitor de la red

En cd commercial-paper/organization/magnetocorp

---

(MagnetoCorp admin)\$ ./monitordocker.sh net\_test

---

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias



# Código del smart contract

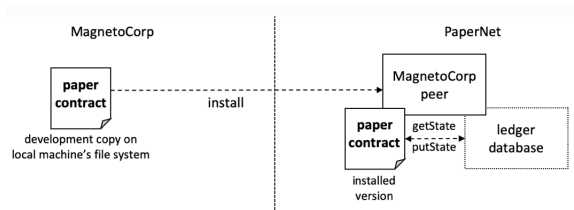
En cd commercial-paper/organization/magnetocorp

---

(MagnetoCorp dev)\$ code contract

---

# Ciclo de vida en Fabric



- El desarrollador empaqueta el chaincode ([1...n] smart contract)
- El administrador instalar el chaincode en cada organización.
- El administrador de cada organización debe aprobar el chaincode.
- El administrador publica en el chaincode en el ledger del canal asociado.

[https://hyperledger-fabric.readthedocs.io/en/release-2.0/chaincode\\_lifecycle.html#chaincode-lifecycle](https://hyperledger-fabric.readthedocs.io/en/release-2.0/chaincode_lifecycle.html#chaincode-lifecycle)

# Instalación y aprobación

**Variables de entorno:** En `cd commercial-paper/organization/magnetocorp` hay que establecer las variables de entorno para que el administrador pueda instalar y aprobar el chaincode en la organización Magnetocorp.

---

```
(Magnetocorp admin)$ source magnetocorp.sh
```

---

# Instalación y aprobación

**Empaquetar chaincode:** El smart contract se puede empaquetar en un chaincode utilizando el comando `peer lifecycle chaincode package`.

---

```
(Magnetocorp admin)$ peer lifecycle chaincode package cp.tar.gz -lang  
node -path ./contract -label cp_0
```

---

# Instalación y aprobación

**Instalar chaincode:** El smart contract se puede instalar utilizando el comando `peer lifecycle chaincode install`.

---

```
(MagnetoCorp admin)$ peer lifecycle chaincode install cp.tar.gz
```

---

# Instalación y aprobación

**Aprobar chaincode:** MagnetoCorp debe aprobar el papercontract que se acaba de instalar con el package ID del contrato.

---

```
(MagnetoCorp admin)$ peer lifecycle chaincode queryinstalled
```

---

```
(MagnetoCorp admin)$ export PACKAGE_ID=cp_0:ffda9...
```

---

```
(MagnetoCorp admin)$ peer lifecycle chaincode approveformyorg  
-orderer localhost:7050 -ordererTLSHostnameOverride  
orderer.example.com -channelID mychannel -name papercontract -v 0  
-package-id $PACKAGE_ID -sequence 1 -tls -cafile $ORDERER_CA
```

---

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

**Aprobar smart contract ajeno (DigiBank)**

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias

# Instalación y aprobación

Por defecto, el ciclo de vida del chaincode en Fabric requiere que la mayoría de las organizaciones apruebe la definición de un chaincode en el canal. En este caso, se tienen 2 organizaciones, por lo tanto, se requiere que las 2 organizaciones apruebe el chaincode.



# Instalación y aprobación

**Variables de entorno:** En `cd commercial-paper/organization/digibank/` hay que establecer las variables de entorno para que el administrador de DigiBank pueda instalar y aprobar el papernet chaincode.

---

```
(DigiBank admin)$ source digibank.sh
```

---

# Instalación y aprobación

**Empaquetar chaincode:** El smart contract se puede empaquetar en un chaincode utilizando el comando `peer lifecycle chaincode package`.

---

```
(DigiBank admin)$ peer lifecycle chaincode package cp.tar.gz -lang  
node -path ./contract -label cp_0
```

---

# Instalación y aprobación

**Instalar chaincode:** El smart contract se puede instalar utilizando el comando `peer lifecycle chaincode install`.

---

```
(DigiBank admin)$ peer lifecycle chaincode install cp.tar.gz
```

---

# Instalación y aprobación

**Aprobar chaincode:** DigiBank debe aprobar el papercontract que se acaba de instalar con el package ID del contrato.

---

```
(DigiBank admin)$ peer lifecycle chaincode queryinstalled
```

---

```
(DigiBank admin)$ export PACKAGE_ID=cp_0:ffda9...
```

---

```
(DigiBank admin)$ peer lifecycle chaincode approveformyorg -orderer  
localhost:7050 -ordererTLSHostnameOverride orderer.example.com  
-channelID mychannel -name papercontract -v 0 -package-id  
$PACKAGE_ID -sequence 1 -tls -cafile $ORDERER_CA
```

---

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

**Publicar smart contract (DigiBank)**

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias

# Publicación

Una vez que el contrato ha sido aprobado por la mayoría del consorcio (2/2) en el canal, se crea un nuevo contenedor docker para ejecutar el papercontract en los nodos de la red y así, éste pueda ser invocado por las aplicaciones cliente en el canal.

# Instalación y aprobación

**Publicar papercontract:** Ya aprobado, el administrador de cualquier organización puede publicar el smart contract utilizando el comando `peer lifecycle chaincode commit`.

---

```
(DigiBank admin)$ peer lifecycle chaincode commit -o localhost:7050
-ordererTLSHostnameOverride orderer.example.com -peerAddresses
localhost:7051 -tlsRootCertFiles $PEER0_ORG1_CA -peerAddresses
localhost:9051 -tlsRootCertFiles $PEER0_ORG2_CA -channelID
mychannel -name papercontract -v 0 -sequence 1 -tls -cafile
$ORDERER_CA -waitForEvent
```

---

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

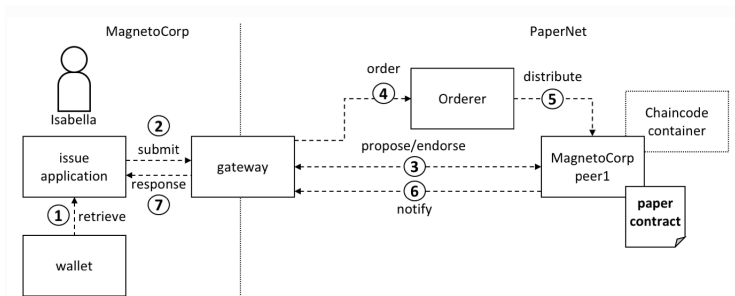
Comprar un papel comercial (DigiBank)

## 2 Referencias



# Diagrama para la emisión

Para emitir el papel comercial MagnetoCorp utiliza su aplicación cliente llamada issue.js.



# Emisión

En `cd commercial-paper/organization/magnetocorp/application/` se pueden ver los archivos **`addToWallet.js`**, **`issue.js`** y **`package.json`**.

Isabella va a utilizar `addToWallet.js` para agregar su identidad al wallet (Certificado X.509). `issue.js` utilizará esa identidad para crear el papel comercial a nombre de Magnetocorp invocando al `papercontract`.

# Emisión

Como se puede observar al inicio del archivo **issue.js** se requieren ciertas dependencias las cuales están declaradas en el archivo **package.json** y se pueden descargar con npm.

---

```
(Isabella)$ npm install
```

---

# Emisión

Ahora, Isabella debe agregar su credenciales X.509 a su wallet.

---

```
(Isabella)$ node addToWallet.js
```

---

```
(Isabella)$ ls ../identity/user/isabella/wallet/
```

---

# Emisión

Ahora sí, Isabella puede utilizar **issue.js** para enviar la transacción que emitirá el papel comercial 00001 de parte de MagnetoCorp.

---

(Isabella)\$ node issue.js

---

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias

# Compra

Para comprar el papel comercial que emitió MagnetoCorp, el usuario de DigiBank (Balaji) debe realizar una transacción de compra desde la aplicación de DigiBank.

# Compra

En `cd commercial-paper/organization/digibank/application/` se pueden ver los archivos **`addToWallet.js`**, **`buy.js`**, **`package.json`** y **`redeem.js`**

Balaji va a utilizar `buy.js` y `redeem.js` para comprar y, posteriormente, canjear el papel comercial que emitió MagnetoCorp.



# Compra

Al igual que en MagnetoCorp, DigiBank debe instalar los paquetes declarados en **package.json** a través de npm.

```
(Balaji)$ npm install
```

# Compra

Ahora, Balaji debe agregar sus identidad a su wallet ejecutando el programa **addToWallet.js**.

---

```
(Balaji)$ node addToWallet.js
```

---

# Compra

Finalmente, Balaji puede usar **buy.js** para enviar la transacción para transferir el papel comercial de MagnetoCorp a DigiBank.

```
(Balaji)$ node buy.js
```

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias

# Compra

La transacción final para el papel comercial será que DigiBank lo intercambie con MagnetoCorp. Para ello Balaji utilizará el **redeem.js**.

---

(Balaji)\$ node redeem.js

---

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias

# Compra

En `cd fabric-samples/commercial-paper.`

---

(Balaji)\$ `./network-clean.sh`

---

Este script dará de baja los peers, los contenedores, el nodo ordering service y el logspout. Además, eliminará las identidades de Isabella y Balaji.

# Índice

## 1 Commercial paper

PaperNet

Crear smart contract propio (MagnetoCorp)

Aprobar smart contract ajeno (DigiBank)

Publicar smart contract (DigiBank)

Emitir un papel comercial (MagnetoCorp)

Comprar un papel comercial (DigiBank)

## 2 Referencias



# Referencias I

- [https://hyperledger-fabric.readthedocs.io/en/release-2.0/tutorial/commercial\\_paper.html#prerequisites](https://hyperledger-fabric.readthedocs.io/en/release-2.0/tutorial/commercial_paper.html#prerequisites)