



Facultad de Estudios Estadísticos  
Universidad Complutense de Madrid



# Blockchain & BigData Canino

ANEXOII: Arquitectura Blockchain



## **Autores:**

Cristina Rodríguez Chamorro  
Daniel Lanzas Pellico  
Helena García Fernández  
José Bennani Pareja  
Juan José Lucas de la Fuente  
Unai Ares Icaran

## **Tutor:**

Sergio Torres Palomino

Documentación del TFM

*Máster Blockchain y Big Data. Curso 2019-2020*

# ÍNDICE

Tecnología a utilizar .....	2
Arquitectura del proyecto .....	4
INSTALACIÓN DE HYPERLEDGER FABRIC .....	6
ORGANIZACIÓN DE LA RED HYPERLEDGER FABRIC.....	8
ESTRUCTURA DEL PROYECTO .....	9
SCRIPTS PARA LEVANTAR EL PROYECTO .....	12
CONFIGURACIÓN DE LA RED .....	21
Puesta en marcha del proyecto.....	34
RED INICIAL SIN TLS.....	34
INSTALACIÓN DE HYPERLEDGER EXPLORER.....	43
RED CON TLS .....	50
RED CREADA ENTRE DOS SERVIDORES .....	54
• Creación de la red Docker Swarm.....	54
• Configuración de la red .....	56

## Tecnología a utilizar

Una vez definido el caso de uso, consistente en una red entre las organizaciones Federaciones Caninas y Colegios Veterinarios para facilitar la realización de las distintas acciones que los propietarios de perros tienen que llevar a cabo en cada una de estas organizaciones y evitar los posibles casos de fraude que puedan intentar cometerse, se procede a estudiar cuál será la mejor tecnología y arquitectura para generar la red entre ambas.

En primer lugar se decide el uso de la tecnología Blockchain para regir las relaciones entre ambas organizaciones, lo que redundará en que la actividad sea transparente para ambas y en que la información sea inmutable y no pueda ser modificada para realizar ningún tipo de fraude.

Por otra parte, dado que se trata de dos organizaciones privadas que deben interactuar sin llegar a tener confianza una en la otra y que los datos tienen una componente privada (siendo algunos de ellos sensibles y protegidos por el RGPD), se propone la utilización de una red Blockchain privada con la tecnología Hyperledger Fabric como base.

Los activos con los que contará la red son los perros, que tendrán un propietario que podrá cambiar, se les podrá poner el microchip y vacunas, participarán en concursos,...

Los participantes serán las Federaciones Caninas y los Colegios Veterinarios, que son los que ejecutarán las acciones sobre los perros.

Por otra parte que los usuarios serán las propias Federaciones, los Colegios Veterinarios, los propietarios, los veterinarios y las autoridades competentes (agentes de la autoridad, juzgados,...) a los que se permita el uso de la red.

En el siguiente gráfico se describen las relaciones entre los participantes en la red, que se gestionan a través de Hyperledger Fabric:

- Criadores: Solicitarán el afijo a la Federación correspondiente y notificarán la camada a la Federación, recibiendo la inscripción de la camada
- Veterinarios: Colocarán e inscribirán el microchip y las vacunas en la Federación. Tendrán relación con los propietarios, que solicitarán estas acciones.
- Como propuesta a futuro los veterinarios tomarán una muestra de ADN que quedará registrada en la Blockchain y además podría utilizarse para realizar un algoritmo que permita determinar los mejores cruces entre los perros para obtener los mejores ejemplares
- Federaciones: Inscribirán los afijos, las camadas, los microchips y vacunas, la propiedad de los perros, los resultados de los concursos y facilitarán el pedigrí a los propietarios que lo soliciten. Además podrá haber otros clubes y asociaciones que realicen concursos, pertenezcan o no a la Federación.

- Propietarios: Solicitarán la inscripción de la propiedad del perro a la Federación, el microchip y vacunas a los veterinarios, presentarán perros a concursos, solicitarán pedigries,...

Descripción:

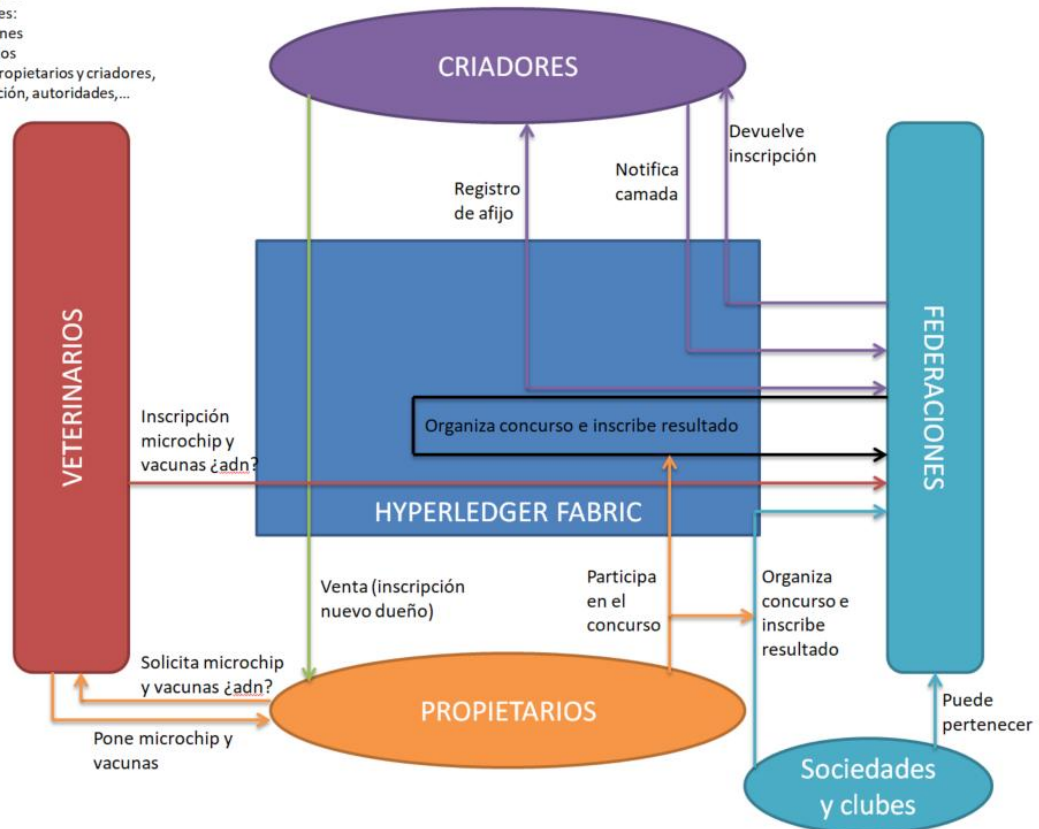
Activos: Perros

Participantes:

- Federaciones

- Veterinarios

Usuarios: Propietarios y criadores,  
administración, autoridades,...



## Arquitectura del proyecto

Dentro de Hyperledger Fabric se definen dos organizaciones con las siguientes características:

- **Federaciones Caninas**

Se trata de las organizaciones que llevan a cabo el control de los afijos autorizados para la cría de perros, control de las características de las razas, pedigrís,...

En este caso se tendrán en cuenta las cuatro mayores Federaciones existentes:

- FCI: Fédération Cynologique internationale
- RSCE: Real Sociedad Canina Española
- TKC: The Kennel Club
- ACW: Alianz Canine Worldwide

- **Colegios Veterinarios**

Se trata de las organizaciones que llevan el control de todo lo relativo a la sanidad de los perros: colocación de microchip, vacunas, pasaporte sanitario,...

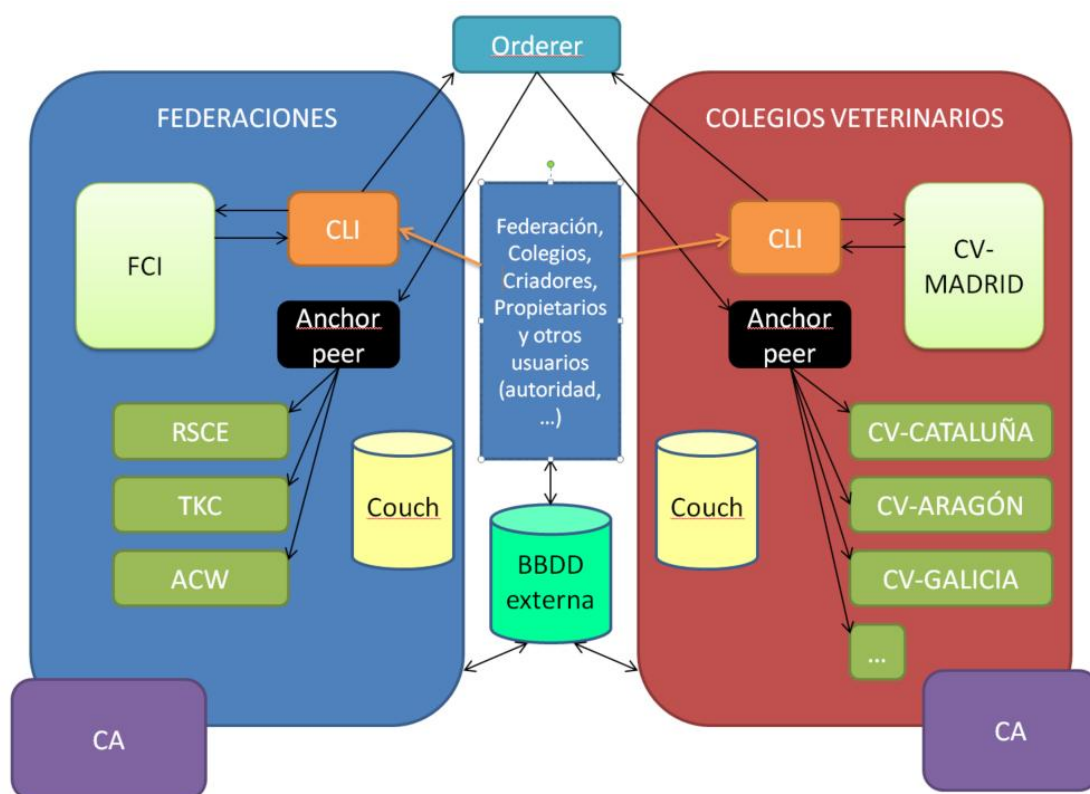
Se tomarán los Colegios Veterinarios a nivel de Comunidad Autónoma, por lo que tendremos diecisiete Colegios Veterinarios

Se utilizará CouchDB en vez de levelDB para aumentar las posibilidades de consulta a la red y cada organización tendrá una autoridad de certificación (CA).

Además se implementará una base de datos externa para guardar los datos sensibles y los destinados al uso por la aplicación de Big Data.

El tipo de consenso será “solo” para el desarrollo, aunque posteriormente debería implementarse “Kafka” en producción.

Una vez definidas las organizaciones la arquitectura será la siguiente:



Cada organización dispondrá de un nodo para cada uno de sus componentes, existiendo además un CLI al que atacarán los usuarios desde la API, un Orderer que distribuirá las transacciones por los nodos de la red, un Anchor Peer al que el Orderer enviará la información, una CouchDB en cada peer, una CA (autoridad de certificación) para cada organización y una base de datos externa en la que se guardarán los datos sensibles y los destinados al uso por la aplicación de Big Data.

## INSTALACIÓN DE HYPERLEDGER FABRIC

Para la instalación se utilizará un usuario al que daremos de alta con el nombre “hyperledger”.

Una vez instalados los prerequisites de hyperledger para Ubuntu que se encuentran en:

<https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh>

Y que incluyen docker y node.js, tras instalarlos se procede a instalar docker-compose desde la dirección:

[https://github.com/docker/compose/releases/download/1.21.2/docker-compose-\\$\(uname -s\)-\\$\(uname -m\)](https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m))

Y por último go:

<https://dl.google.com/go/go1.13.3.linux-amd64.tar.gz>

Las instrucciones concretas para realizarlo han sido:

```
# Añadir usuario hyperledger  
adduser hyperledger
```

```
# Dar permisos root al usuario  
sudo usermod -a -G sudo hyperledger  
sudo vi /etc/sudoers
```

```
# Añadir bajo el apartado User privilege specification: hyperledger  ALL=(ALL:ALL) ALL
```

```
logout
```

```
# Entrar al servidor con el usuario hyperledger
```

```
# Descargar los prerequisites  
curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh
```

```
# Dar permisos de ejecución  
chmod u+x prereqs-ubuntu.sh
```

```
# Ejecutar los prerequisites  
./prereqs-ubuntu.sh
```

```
# Actualizar docker-compose y dar permisos de ejecución  
sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose
```

```
# Actualizar
sudo apt-get update
sudo apt-get -y upgrade

/# Instalar GO
sudo curl -O https://dl.google.com/go/go1.13.3.linux-amd64.tar.gz
sudo tar -xvf go1.13.3.linux-amd64.tar.gz

/# Editar el archivo bashrc y añadir al final del fichero las siguientes variables de estado
sudo vi ~/.bashrc

export GOROOT=$HOME/go
export GOPATH=$HOME/work
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin

source ~/.bashrc

logout

/# Comprobar que se ha instalado GO
go version
```

Una vez preparado el servidor se procede a descargar e instalar Hyperledger Fabric.

Instalaremos Fabric bajo el home del usuario hyperledger con la siguiente ruta:

```
/home/hyperledger/work/src
```

Las instrucciones serán:

```
# Creamos la ruta a utilizar
mkdir work && cd work
mkdir src && cd src

# Descargamos el repositorio y los binarios de Fabric utilizando las versiones correctas
git clone -b master https://github.com/hyperledger/fabric-samples.git

cd ~/work/src/fabric-samples
git checkout v1.4.0
curl -sSL http://bit.ly/2ysbOFE | bash -s -- 1.4.0 1.4.0

# Comprobamos la ruta para exportar la variable de estado PATH
pwd

export PATH=/home/hyperledger/work/src/fabric-samples/bin:$PATH
```

En este punto el servidor ya está preparado para utilizar Hyperledger Fabric.



## ORGANIZACIÓN DE LA RED HYPERLEDGER FABRIC

El proyecto lo situaremos bajo la carpeta genérica que ha creado hyperledger fabric durante la instalación en:

```
/home/hyperledger/work/src/fabric-samples
```

Donde crearemos una carpeta llamada TFM, siendo la ruta del proyecto:

```
/home/hyperledger/work/src/fabric-samples/TFM
```

En primer lugar crearemos una red con la arquitectura descrita anteriormente:

- Dos organizaciones: Federaciones y ColegiosVeterinarios
- Cuatro peers para la primera organización y diecisiete para la segunda
- Orderer
- CLI
- CA's
- CouchDB

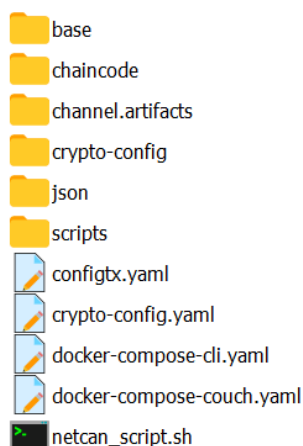
En la que instalaremos Hyperledger Explorer para poder visualizar la información de la red en el navegador, accediendo también a la CouchDB desde el navegador.

Posteriormente añadiremos TLS a la red, generando una red más segura en la que será necesario proveer los certificados correspondientes al efectuar cualquier acción.

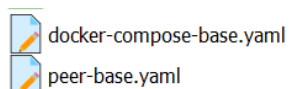
Por último desplegaremos la red entre dos servidores simulando la existencia real de dos las dos organizaciones.

## ESTRUCTURA DEL PROYECTO

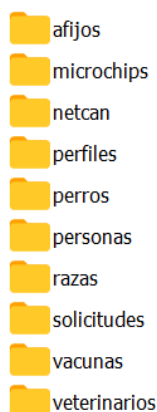
Para la realización del proyecto, bajo la carpeta del proyecto /TFM, se generarán:



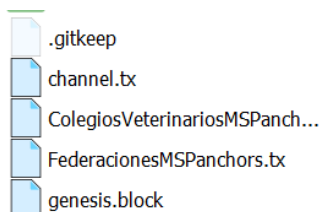
- Los archivos de configuración del proyecto de fabric: configtx.yaml, crypto-config.yaml, docker-compose-cli.yaml y docker-compose-couch.yaml



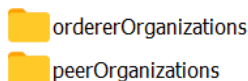
- La carpeta base en donde se situarán dos archivos de configuración de los que extienden los dos archivos docker-compose anteriores: docker-compose-base.yaml y peer-base.yaml



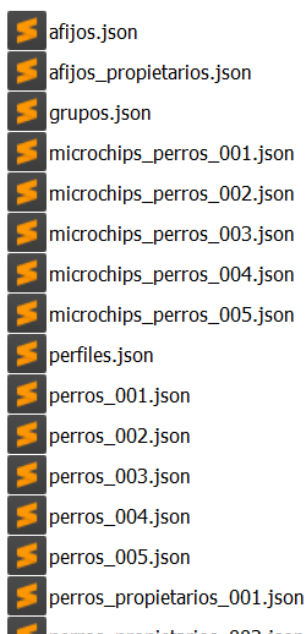
- La carpeta chaincode, donde se situarán los chaincodes desarrollados para la red: afijos, microchips, perfiles, perros, personas, razas, solicitudes, vacunas y veterinarios, y una carpeta netcan con datos comunes a todos ellos



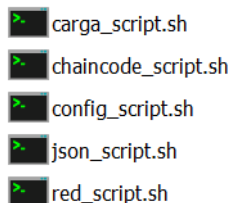
- La carpeta channel-artifacts, donde se generarán los artefactos del proyecto



- La carpeta crypto-config, donde se generará el material criptográfico



- La carpeta json, donde colocaremos los archivos json generados con los datos iniciales para cargar a la Blockchain: los afijos existentes, perros con sus dependencias (padres, hijos,...), propietarios, veterinarios, microchips, vacunas,...



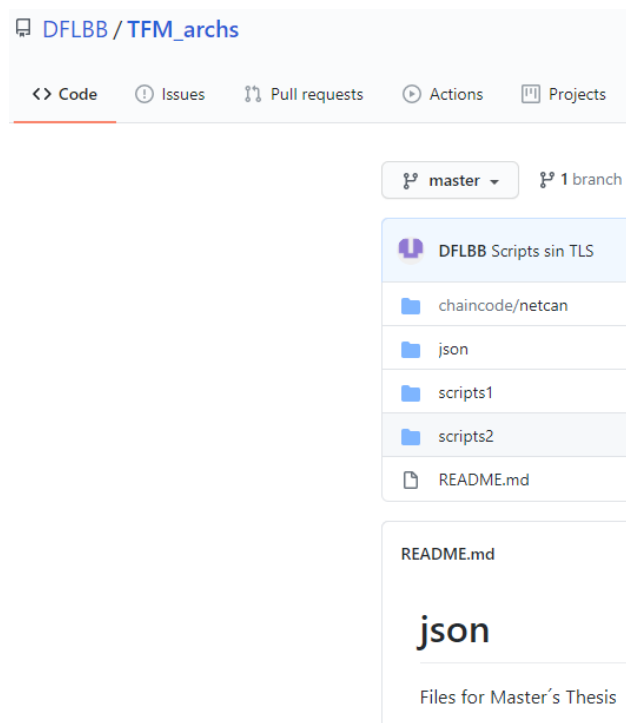
- La carpeta scripts, donde colocaremos una serie de scripts para levantar el proyecto de manera automática, de manera que se levante la red, se configure, se instalen e instancien los chaincodes desarrollados y se carguen los datos iniciales

```
netcan_script.sh
stop_netcan_script.sh
```

- Por último dos scripts para lanzar el proyecto y pararlo y limpiar del servidor los dockers y chaincodes levantados
- Al ejecutar el script netcan\_script.sh lo primero que hace es acceder a un repositorio en github creado para el proyecto donde se encuentran actualizados el resto de scripts, los archivos json de carga inicial actualizados y los chaincodes actualizados, descargando dichas carpetas al servidor y sustituyendo las existentes (si las hay), para que el proyecto esté siempre actualizado.

Dicho repositorio puede encontrarse en:

[https://github.com/DFLBB/TFM\\_archs](https://github.com/DFLBB/TFM_archs)



## SCRIPTS PARA LEVANTAR EL PROYECTO

- netcan\_script.sh

Es el único script que hay que lanzar para inicializar el proyecto.

```
#!/bin/bash

clear
echo "*****"
echo "                               Bienvenido a"
echo ""
echo "      _/      _/  _/ _/ _/      _/ _/      _/ _/      _/"
echo "      _/_/      _/  _/ _/ _/      _/ _/      _/ _/      _/"
echo "      _/_/      _/  _/ _/ _/      _/ _/      _/ _/      _/"
echo "      _/_/      _/  _/ _/ _/      _/ _/      _/ _/      _/"
echo ""
echo "*****"

echo ""
echo "*****"
echo "Limpiando la instalación"
echo "*****"

docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
docker volume prune --force
docker system prune --force
docker network prune --force

rm -rf ./json
rm -rf ./chaincode
rm -rf ./scripts
git clone https://github.com/DFLB/TFM_archs /home/hyperledger/work/src/fabric-samples/TFM/TFM_archs
mv ./TFM_archs/chaincode ./chaincode
mv ./TFM_archs/json ./json
mv ./TFM_archs/scripts1 ./scripts
cd scripts/ && chmod +x *.sh && cd ..
rm -rf ./TFM_archs

echo ""
echo "*****"
echo "Limpiando chaincodes"
echo "*****"

docker rmi $(docker images dev-fci.federaciones.netcan.com-* -q)

echo ""
echo "*****"
echo "Copiando los scripts de arranque al servidor"
echo "*****"

clear

./scripts/red_script.sh
#./scripts/Serv2_script.sh
docker exec cli /netcan_scripts/config_script.sh
docker exec cli /netcan_scripts/chaincode_script.sh
./scripts/json_script.sh
docker exec cli /netcan_scripts/carga_script.sh
```

En primer lugar limpia la instalación, a continuación accede al repositorio de Github, descarga las carpetas json, chaincode y scripts y sustituye las existentes por estas y posteriormente va lanzando el resto de scripts de configuración, ya sea para uno o dos servidores

Este script levanta la red hyperledger Fabric y copia los scripts necesarios al docker del CLI, donde luego habrá que realizar la configuración del proyecto

```
#!/bin/bash

echo "*****"
echo ""
echo "      _/      _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/"
echo "    _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/"
echo "   _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/"
echo "  _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/"
echo " _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/"
echo ""
echo "*****"
echo ""
echo "*****"
echo "Levantando la red"
echo "*****"

docker-compose -f docker-compose-cli.yaml -f docker-compose-couch.yaml up -d

echo "*****"
echo "Copiando scripts al CLI"
echo "*****"

docker cp /home/hyperledger/work/src/fabric-samples/TFM2/scripts cli:/netcan_scripts
export TERM=xterm
clear
```

Este script realiza la configuración en el servidor 2 en el caso de que se levante la red entre dos servidores

```
#!/bin/bash

echo "*****"
echo ""
echo ""
echo ""
echo ""
echo ""
echo ""
echo ""
echo "*****"
echo ""
echo ""

echo "*****"
echo "Configurando el servidor ColegiosVeterinarios"
echo "*****"

ssh hyperledger@15.188.185.49 bash -c ""

clear

echo "*****"
echo ""
echo ""
echo ""
echo ""
echo ""
echo ""
echo ""
echo "*****"
echo ""
echo ""
```

```
echo "*****"
echo "Limpiando la instalación"
echo "*****"

docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
docker volume prune --force
docker system prune --force
docker network prune --force

echo "*****"
echo "Levantando la red en el servidor ColegiosVeterinarios"
echo "*****"
docker-compose -f docker-compose-cli.yaml -f docker-compose-couch.yaml up -d

"
```

- config\_script.sh

Este script se ejecuta dentro del docker del CLI y realiza la configuración del canal, los peers y los pares de anclaje, ya sea con TLS o sin TLS

```
#!/bin/bash
clear

echo "*****"
echo ""
echo "      _/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/_/  _/"
echo ""
echo "*****"

echo ""

echo "*****"
echo "Estableciendo las variables de entorno del canal"
echo "*****"

export CHANNEL_NAME=netcanchannel

echo ""
echo "*****"
echo "Creando el canal"
echo "*****"

peer channel create -o orderer.netcan.com:7050 -t 15s -c $CHANNEL_NAME -f ./channel-artifacts/channel.tx --tls --
cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/netcan.com/orderers/orderer.netc
an.com/msp/tlsacerts/tlsca.netcan.com-cert.pem

echo ""
echo "*****"
echo "Adhiriendo el peer FCI de Federaciones al canal"
echo "Puede tardar un poco..."
echo "*****"
sleep 60

peer channel join -b netcanchannel.block

while [ $? -eq 1 ];
do
    echo ""
    echo "*****"
    echo "Reintentando en 5 segundos"
    echo "*****"
    sleep 5
    peer channel join -b netcanchannel.block;
done

echo ""
echo "*****"
```





```
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/colegiosveterinari
os.netcan.com/users/Admin@colegiosveterinarios.netcan.com/msp/ CORE_PEER_ADDRESS=cvgalicia.colegiosveterinarios.netcan
.com:7051 CORE_PEER_LOCALMSPID="ColegiosVeterinariosMSP" CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperl
edger/fabric/peer/crypto/peerOrganizations/colegiosveterinarios.netcan.com/peers/cvgalicia.colegiosveterinarios.netcan
.com/tls/ca.crt peer channel join -b netcanchannel.block
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/colegiosveterinari
os.netcan.com/users/Admin@colegiosveterinarios.netcan.com/msp/ CORE_PEER_ADDRESS=cvmurcia.colegiosveterinarios.netcan
.com:7051 CORE_PEER_LOCALMSPID="ColegiosVeterinariosMSP" CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperle
dger/fabric/peer/crypto/peerOrganizations/colegiosveterinarios.netcan.com/peers/cvmurcia.colegiosveterinarios.netcan.c
om/tls/ca.crt peer channel join -b netcanchannel.block
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/colegiosveterinari
os.netcan.com/users/Admin@colegiosveterinarios.netcan.com/msp/ CORE_PEER_ADDRESS=cvnavarra.colegiosveterinarios.netcan
.com:7051 CORE_PEER_LOCALMSPID="ColegiosVeterinariosMSP" CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperl
edger/fabric/peer/crypto/peerOrganizations/colegiosveterinarios.netcan.com/peers/cvnavarra.colegiosveterinarios.netcan
.com/tls/ca.crt peer channel join -b netcanchannel.block
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/colegiosveterinari
os.netcan.com/users/Admin@colegiosveterinarios.netcan.com/msp/ CORE_PEER_ADDRESS=cvpaisvasco.colegiosveterinarios.netc
an.com:7051 CORE_PEER_LOCALMSPID="ColegiosVeterinariosMSP" CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hype
rledger/fabric/peer/crypto/peerOrganizations/colegiosveterinarios.netcan.com/peers/cvpaisvasco.colegiosveterinarios.ne
tcan.com/tls/ca.crt peer channel join -b netcanchannel.block
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/colegiosveterinari
os.netcan.com/users/Admin@colegiosveterinarios.netcan.com/msp/ CORE_PEER_ADDRESS=cvlarioja.colegiosveterinarios.netcan
.com:7051 CORE_PEER_LOCALMSPID="ColegiosVeterinariosMSP" CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperl
edger/fabric/peer/crypto/peerOrganizations/colegiosveterinarios.netcan.com/peers/cvlarioja.colegiosveterinarios.netcan
.com/tls/ca.crt peer channel join -b netcanchannel.block

echo ""
echo "*****"
echo "Declarando los pares de anclaje"
echo "*****"

peer channel update -o orderer.netcan.com:7050 -c $CHANNEL_NAME -f ./channel-artifacts/FederacionesMSPanchors.tx --
tls --
cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/netcan.com/orderers/orderer.netc
an.com/msp/tlscacerts/tlsca.netcan.com-cert.pem
CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/colegiosveterinari
os.netcan.com/users/Admin@colegiosveterinarios.netcan.com/msp/ CORE_PEER_ADDRESS=cvmadrid.colegiosveterinarios.netcan.c
om:7051 CORE_PEER_LOCALMSPID="ColegiosVeterinariosMSP" CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperled
ger/fabric/peer/crypto/peerOrganizations/colegiosveterinarios.netcan.com/peers/cvmadrid.colegiosveterinarios.netcan.co
m/tls/ca.crt peer channel update -o orderer.netcan.com:7050 -c $CHANNEL_NAME -f ./channel-
artifacts/ColegiosVeterinariosMSPanchors.tx --tls --
cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/netcan.com/orderers/orderer.netc
an.com/msp/tlscacerts/tlsca.netcan.com-cert.pem
```

- chaincode\_script.sh

Este script también se ejecuta dentro del docker del CLI e instala e instancia los chaincodes desarrollados

```
#!/bin/bash

clear

echo "*****"
echo ""
echo "      _/      _/      _/      _/      _/      _/      _/      _/      _/"
echo "     _/_      _/_      _/_      _/_      _/_      _/_      _/_      _/"
echo "    _/_      _/_      _/_      _/_      _/_      _/_      _/_      _/"
echo "   _/_      _/_      _/_      _/_      _/_      _/_      _/_      _/"
echo "  _/_      _/_      _/_      _/_      _/_      _/_      _/_      _/"
echo ""
echo "*****"

echo ""
echo "*****"
echo "Copiando los chaincode al servidor"
echo "*****"

echo ""
echo "*****"
echo "Estableciendo las variables de entorno de chaincodes"
echo "*****"

export CHANNEL_NAME=netcanchannel
```

```
export CA_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/netcan.com/orderers/orde
rer.netcan.com/msp/tlscacerts/tlsca.netcan.com-cert.pem
export ORDERER_URL=orderer.netcan.com:7050
export CC_VERSION=1.0.0

LISTCHAINCODES=`ls /opt/gopath/src/github.com/chaincode/netcan`
for CHAINCODE in $LISTCHAINCODES; do
    if [ "$CHAINCODE" != "netcan" ]; then

        echo ""
        echo "*****"
        echo "Instalando e instanciando el chaincode $CHAINCODE"
        echo "*****"

        export CC_NOMBRE=$CHAINCODE
        export CC_FILE=github.com/chaincode/netcan/$CHAINCODE/cc

        peer chaincode install -n $CC_NOMBRE -v $CC_VERSION -p $CC_FILE
        peer chaincode instantiate -n $CC_NOMBRE -v $CC_VERSION -c '{"Args":["init"]}' -o $ORDERER_URL --tls --
cafile $CA_FILE -C $CHANNEL_NAME -P "OR ('FederacionesMSP.peer','ColegiosVeterinariosMSP.peer')"
    fi
done
```

- json\_script

Este script copia los archivos de carga iniciales a los dockers de los chaincodes para realizar las cargas iniciales de datos a la blockchain

```
#!/bin/bash

clear

echo "*****"
echo ""
echo "      _/      _/      _/      _/      _/      _/      _/      _/      _/"
echo "     _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/"
echo "    _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/"
echo "   _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/"
echo "  _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/_/     _/"
echo ""
echo "*****"

echo ""

echo "*****"
echo "Copiando los archivos json de carga inicial de datos"
echo "*****"

for DOCKER in $(docker ps --filter "since=cli" -q); do

    echo ""
    echo "*****"
    echo "Copiando los archivos json de carga inicial de datos al docker $DOCKER"
    echo "*****"

    docker cp /home/hyperledger/work/src/fabric-samples/TFM/json $DOCKER:./json
done
```

- carga\_script.sh

Este script se ejecuta dentro del docker del CLI y realiza las cargas iniciales de datos a la blockchain

```
#!/bin/bash

export CHANNEL_NAME=netcanchannel
export CA_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/netcan.com/orderers/orderer.netcan.com/msp/tlscacerts/tlsca.netcan.com-cert.pem
export ORDERER_URL=orderer.netcan.com:7050

echo "*****"
echo "Carga inicial de datos"
echo "*****"

echo "*****"
echo "Cargando datos de PERFILES DE PERSONAS"
echo "*****"

peer chaincode invoke -n perfiles -c '{"function": "cargarDatosIniciales", "Args": [".json/perfiles.json", "{\"IDPersona\":0}"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n perfiles -c '{"function": "asignarEstado_OnlyAdmin", "Args": ["PERFILES_PERSONAS", "{\"docType\": \"CONTADOR\", \"IDMaximo\":48}", {"IDPersona\":0}"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n perfiles -c '{"function": "cancelarPerfilPersona", "Args": [{"IDPersona\":0, \"CODPerfil\": \"ADMINISTRADOR\"}, {"IDPersona\":0}"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de GRUPOS"
echo "*****"

peer chaincode invoke -n razas -c '{"function": "cargarDatosIniciales_Grupos", "Args": [".json/grupos.json"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n razas -c '{"function": "asignarEstado", "Args": ["GRUPOS", "{\"docType\": \"CONTADOR\", \"IDMaximo\":12}"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de RAZAS"
echo "*****"

peer chaincode invoke -n razas -c '{"function": "cargarDatosIniciales", "Args": [".json/razas.json"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n razas -c '{"function": "asignarEstado", "Args": ["RAZAS", "{\"docType\": \"CONTADOR\", \"IDMaximo\":346}"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de PERSONAS"
echo "*****"

peer chaincode invoke -n personas -c '{"function": "cargarDatosIniciales", "Args": [".json/personas_001.json"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n personas -c '{"function": "asignarEstado", "Args": ["PERSONAS", "{\"docType\": \"CONTADOR\", \"IDMaximo\":1000}"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de AFIJOS"
echo "*****"

peer chaincode invoke -n afijos -c '{"function": "cargarDatosIniciales", "Args": [".json/afijos.json"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n afijos -c '{"function": "asignarEstado", "Args": ["AFIJOS", "{\"docType\": \"CONTADOR\", \"IDMaximo\":50}"]}' -o $ORDERER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de PROPIETARIOS DE AFIJOS"
echo "*****"
```

```
peer chaincode invoke -n afijos -
c '{"function":"cargarDatosIniciales_Propietarios","Args":["./json/afijos_propietarios.json"]}' -o $ORDENER_URL --
tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n afijos -
c '{"function":"asignarEstado","Args":["AFIJOS_PROPIETARIOS", '{"docType":"'CONTADOR','IDMaximo':50}"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de PERROS"
echo "*****"

peer chaincode invoke -n perros -c '{"function":"cargarDatosIniciales","Args":["./json/perros_001.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -c '{"function":"cargarDatosIniciales","Args":["./json/perros_002.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -c '{"function":"cargarDatosIniciales","Args":["./json/perros_003.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -c '{"function":"cargarDatosIniciales","Args":["./json/perros_004.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -c '{"function":"cargarDatosIniciales","Args":["./json/perros_005.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -
c '{"function":"asignarEstado","Args":["PERROS", '{"docType":"'CONTADOR','IDMaximo':5759}"]}' -o $ORDENER_URL --
tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de PROPIETARIOS DE PERROS"
echo "*****"

peer chaincode invoke -n perros -
c '{"function":"cargarDatosIniciales_Propietarios","Args":["./json/perros_propietarios_001.json"]}' -o $ORDENER_URL --
tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -
c '{"function":"cargarDatosIniciales_Propietarios","Args":["./json/perros_propietarios_002.json"]}' -o $ORDENER_URL --
tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -
c '{"function":"cargarDatosIniciales_Propietarios","Args":["./json/perros_propietarios_003.json"]}' -o $ORDENER_URL --
tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -
c '{"function":"cargarDatosIniciales_Propietarios","Args":["./json/perros_propietarios_004.json"]}' -o $ORDENER_URL --
tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -
c '{"function":"cargarDatosIniciales_Propietarios","Args":["./json/perros_propietarios_005.json"]}' -o $ORDENER_URL --
tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n perros -
c '{"function":"asignarEstado","Args":["PERROS_PROPIETARIOS", '{"docType":"'CONTADOR','IDMaximo':5759}"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de VETERINARIOS"
echo "*****"

peer chaincode invoke -n veterinarios -c '{"function":"cargarDatosIniciales","Args":["./json/veterinarios.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n veterinarios -
c '{"function":"asignarEstado","Args":["VETERINARIOS_PERSONAS", '{"docType":"'CONTADOR','IDMaximo':21}"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de MICROCHIPS"
echo "*****"

peer chaincode invoke -n microchips -
c '{"function":"cargarDatosIniciales","Args":["./json/microchips_perros_001.json"]}' -o $ORDENER_URL --tls --
cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
```

```
peer chaincode invoke -n microchips -
c '{"function":"cargarDatosIniciales","Args":["./json/microchips_perros_002.json"]}' -o $ORDENER_URL --tls --
cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n microchips -
c '{"function":"cargarDatosIniciales","Args":["./json/microchips_perros_003.json"]}' -o $ORDENER_URL --tls --
cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n microchips -
c '{"function":"cargarDatosIniciales","Args":["./json/microchips_perros_004.json"]}' -o $ORDENER_URL --tls --
cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n microchips -
c '{"function":"cargarDatosIniciales","Args":["./json/microchips_perros_005.json"]}' -o $ORDENER_URL --tls --
cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n microchips -
c '{"function":"asignarEstado","Args":["MICROCHIPS_PERROS", "{\"docType\":\"CONTADOR\",\"IDMaximo\":5759}"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "Cargando datos de VACUNAS"
echo "*****"

peer chaincode invoke -n vacunas -
c '{"function":"cargarDatosIniciales_VacunasProteccion","Args":["./json/vacunas_proteccion_001.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s

echo "*****"
echo "Cargando datos de VACUNACIONES"
echo "*****"

peer chaincode invoke -n vacunas -
c '{"function":"asignarEstado","Args":["VACUNAS_PROTECCION", "{\"docType\":\"CONTADOR\",\"IDMaximo\":11}"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n vacunas -c '{"function":"cargarDatosIniciales","Args":["./json/vacunas_perros_001.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n vacunas -
c '{"function":"asignarEstado","Args":["VACUNAS_PERROS", "{\"docType\":\"CONTADOR\",\"IDMaximo\":89}"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
peer chaincode invoke -n vacunas -
c '{"function":"cargarDatosIniciales_VacunasProteccion","Args":["./json/vacunas_perros_proteccion_001.json"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME
sleep 7s
peer chaincode invoke -n vacunas -
c '{"function":"asignarEstado","Args":["VACUNAS_PERROS_PROTECCION", "{\"docType\":\"CONTADOR\",\"IDMaximo\":512}"]}' -
o $ORDENER_URL --tls --cafile $CA_FILE -C $CHANNEL_NAME

echo "*****"
echo "DATOS INICIALES CARGADOS"
echo "*****"
```

- stop\_netcan\_script

Por último este script para la red, borra los dockers levantados y borra los chaincodes instanciados

```
#!/bin/bash

clear

echo "*****"
echo "Limpiando la instalación"
echo "*****"

docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
docker volume prune --force
docker system prune --force
```

```
docker network prune --force

echo "*****"
echo "Limpiando chaincodes"
echo "*****"

docker rmi $(docker images dev-fci.federaciones.netcan.com-* -q)

clear

echo "*****"
echo ""
echo ""
echo "      _/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/"
echo "      _/_/      _/  _/_/_/_/  _/_/_/_/  _/_/_/  _/_/  _/"
echo ""
echo "*****"
echo ""
echo "                Apagada y servidor limpio"
echo ""
```

## CONFIGURACIÓN DE LA RED

Para realizar la configuración de la red es necesario definir los archivos mencionados en la estructura del proyecto situados bajo la carpeta TFM (configtx.yaml, crypto-config.yaml, docker-compose-cli.yaml y docker-compose-couch.yaml) así como los situados bajo la carpeta base (docker-compose-base.yaml y peer-base.yaml).

- configtx.yaml

En este archivo se realiza la configuración de las organizaciones y del canal y es el que servirá para la generación de los artefactos del proyecto.

Las secciones más importantes son:

- Organizations

Donde se definen las organizaciones con sus anchor peers, los directorios de configuración del MSP (Membership Service Provider) y las políticas de acceso para lectura, escritura y administración

```
Organizations:

# SampleOrg defines an MSP using the sampleconfig. It should never be used
# in production but may be used as a template for other definitions
- &OrdererOrg
  # DefaultOrg defines the organization which is used in the sampleconfig
  # of the fabric.git development environment
  Name: OrdererOrg

  # ID to load the MSP definition as
  ID: OrdererMSP

  # MSPDir is the filesystem path which contains the MSP configuration
  MSPDir: crypto-config/ordererOrganizations/netcan.com/msp

  # Policies defines the set of policies at this level of the config tree
  # For organization policies, their canonical path is usually
```

```
# /Channel/<Application|Orderer>/<OrgName>/<PolicyName>
Policies:
  Readers:
    Type: Signature
    Rule: "OR('OrdererMSP.member')"
  Writers:
    Type: Signature
    Rule: "OR('OrdererMSP.member')"
  Admins:
    Type: Signature
    Rule: "OR('OrdererMSP.admin')"

- &Federaciones
  # DefaultOrg defines the organization which is used in the sampleconfig
  # of the fabric.git development environment
  Name: FederacionesMSP

  # ID to load the MSP definition as
  ID: FederacionesMSP

  MSPDir: crypto-config/peerOrganizations/federaciones.netcan.com/msp

  # Policies defines the set of policies at this level of the config tree
  # For organization policies, their canonical path is usually
  # /Channel/<Application|Orderer>/<OrgName>/<PolicyName>
  Policies:
    Readers:
      Type: Signature
      Rule: "OR('FederacionesMSP.admin', 'FederacionesMSP.peer', 'FederacionesMSP.client')"
    Writers:
      Type: Signature
      Rule: "OR('FederacionesMSP.admin', 'FederacionesMSP.client')"
    Admins:
      Type: Signature
      Rule: "OR('FederacionesMSP.admin')"

  # leave this flag set to true.
  AnchorPeers:
    # AnchorPeers defines the location of peers which can be used
    # for cross org gossip communication. Note, this value is only
    # encoded in the genesis block in the Application section context
    - Host: fci.federaciones.netcan.com
      Port: 7051

- &ColegiosVeterinarios
  # DefaultOrg defines the organization which is used in the sampleconfig
  # of the fabric.git development environment
  Name: ColegiosVeterinariosMSP

  # ID to load the MSP definition as
  ID: ColegiosVeterinariosMSP

  MSPDir: crypto-config/peerOrganizations/colegiosveterinarios.netcan.com/msp

  # Policies defines the set of policies at this level of the config tree
  # For organization policies, their canonical path is usually
  # /Channel/<Application|Orderer>/<OrgName>/<PolicyName>
  Policies:
    Readers:
      Type: Signature
      Rule: "OR('ColegiosVeterinariosMSP.admin', 'ColegiosVeterinariosMSP.peer', 'ColegiosVeterinariosMSP.client')"
    Writers:
      Type: Signature
      Rule: "OR('ColegiosVeterinariosMSP.admin', 'ColegiosVeterinariosMSP.client')"
    Admins:
      Type: Signature
      Rule: "OR('ColegiosVeterinariosMSP.admin')"

  AnchorPeers:
    # AnchorPeers defines the location of peers which can be used
    # for cross org gossip communication. Note, this value is only
    # encoded in the genesis block in the Application section context
    - Host: cvmadrid.colegiosveterinarios.netcan.com
      Port: 7051
```

## - Orderer

Define el orderer y las políticas de acceso para lectura, escritura y administración

```
Orderer: &OrdererDefaults

# Orderer Type: The orderer implementation to start
# Available types are "solo" and "kafka"
OrdererType: solo

Addresses:
- orderer.netcan.com:7050

# Batch Timeout: The amount of time to wait before creating a batch
BatchTimeout: 2s

# Batch Size: Controls the number of messages batched into a block
BatchSize:

# Max Message Count: The maximum number of messages to permit in a batch
MaxMessageCount: 100

# Absolute Max Bytes: The absolute maximum number of bytes allowed for
# the serialized messages in a batch.
AbsoluteMaxBytes: 99 MB

# Preferred Max Bytes: The preferred maximum number of bytes allowed for
# the serialized messages in a batch. A message larger than the preferred
# max bytes will result in a batch larger than preferred max bytes.
PreferredMaxBytes: 512 KB

Kafka:
# Brokers: A list of Kafka brokers to which the orderer connects
# NOTE: Use IP:port notation
Brokers:
- 127.0.0.1:9092

# Organizations is the list of orgs which are defined as participants on
# the orderer side of the network
Organizations:

# Policies defines the set of policies at this level of the config tree
# For Orderer policies, their canonical path is
#   /Channel/Orderer/<PolicyName>
Policies:
  Readers:
    Type: ImplicitMeta
    Rule: "ANY Readers"
  Writers:
    Type: ImplicitMeta
    Rule: "ANY Writers"
  Admins:
    Type: ImplicitMeta
    Rule: "MAJORITY Admins"
# BlockValidation specifies what signatures must be included in the block
# from the orderer for the peer to validate it.
BlockValidation:
  Type: ImplicitMeta
  Rule: "ANY Writers"
```

## - Channel & profiles

Define el canal, las organizaciones, el consorcio,...

```
Channel: &ChannelDefaults

# Policies defines the set of policies at this level of the config tree
# For Channel policies, their canonical path is
#   /Channel/<PolicyName>
Policies:
# Who may invoke the 'Deliver' API
Readers:
  Type: ImplicitMeta
```



```
Rule: "ANY Readers"
# Who may invoke the 'Broadcast' API
Writers:
  Type: ImplicitMeta
  Rule: "ANY Writers"
# By default, who may modify elements at this config level
Admins:
  Type: ImplicitMeta
  Rule: "MAJORITY Admins"

# Capabilities describes the channel level capabilities, see the
# dedicated Capabilities section elsewhere in this file for a full
# description
Capabilities:
  <<: *ChannelCapabilities

#####
#
# Profile
#
# - Different configuration profiles may be encoded here to be specified
#   as parameters to the configtxgen tool
#
#####
Profiles:

  NetCanOrdererGenesis:
    <<: *ChannelDefaults
    Orderer:
      <<: *OrdererDefaults
      Organizations:
        - *OrdererOrg
      Capabilities:
        <<: *OrdererCapabilities
    Consortiums:
      NetCanConsortium :
        Organizations:
          - *Federaciones
          - *ColegiosVeterinarios

  NetCanChannel:
    Consortium: NetCanConsortium
    Application:
      <<: *ApplicationDefaults
      Organizations:
        - *Federaciones
        - *ColegiosVeterinarios
      Capabilities:
        <<: *ApplicationCapabilities

  NetCanDevModeKafka:
    <<: *ChannelDefaults
    Capabilities:
      <<: *ChannelCapabilities
    Orderer:
      <<: *OrdererDefaults
      OrdererType: kafka
      Kafka:
        Brokers:
          - kafka.netcan.com:9092

      Organizations:
        - *OrdererOrg
      Capabilities:
        <<: *OrdererCapabilities
    Application:
      <<: *ApplicationDefaults
      Organizations:
        - <<: *OrdererOrg
    Consortiums:
      NetCanConsortium:
        Organizations:
          - *Federaciones
          - *ColegiosVeterinarios
```

- crypto-config.yaml

Define la arquitectura de la red con el orderer, las organizaciones y los peers para la generación del material criptográfico por medio de la herramienta de hyperledger fabric cryptogen

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#
# -----
# "OrdererOrgs" - Definition of organizations managing orderer nodes
# -----
OrdererOrgs:
  - Name: Orderer
    Domain: netcan.com

    Specs:
      - Hostname: orderer
# -----
# "PeerOrgs" - Definition of organizations managing peer nodes
# -----
PeerOrgs:
  - Name: Federaciones
    Domain: federaciones.netcan.com
    EnableNodeOUs: true

    Specs:
      - Hostname: fci
      - Hostname: rsce
      - Hostname: tkc
      - Hostname: acw

  Users:
    Count: 1

  - Name: ColegiosVeterinarios
    Domain: colegiosveterinarios.netcan.com
    EnableNodeOUs: true
    Specs:
      - Hostname: cvmadrid
      - Hostname: cvandalucia
      - Hostname: cvaragon
      - Hostname: cvasturias
      - Hostname: cvillesbalears
      - Hostname: cvcanarias
      - Hostname: cvcantabria
      - Hostname: cvcastillayleon
      - Hostname: cvcastillalamancha
      - Hostname: cvcataluna
      - Hostname: cvcomunitatvalenciana
      - Hostname: cvextremadura
      - Hostname: cvgalicia
      - Hostname: cvmurcia
      - Hostname: cvnavarra
      - Hostname: cvpaisvasco
      - Hostname: cvlarioja
      - Hostname: cvceuta
      - Hostname: cvmelilla

  Users:
    Count: 1
```

- docker-compose-cli.yaml

Es el archivo de configuración principal de Hyperledger Fabric. En él se define la estructura básica del proyecto.

En primer lugar se definen las CA's (certificate authorities). En este caso irán sin TLS (será lo único sin TLS del proyecto securizado ya que en conversaciones con el tutor nos ha recomendado no securizarlas ya que no es necesario) y se define también el puerto de las mismas y el comando para inicializarlas en el que hay que introducir la ca.certfile y la ca.keyfile correspondientes que se encuentran en la carpeta de la organización en peerOrganizations en la carpeta crypto-config.

A continuación se definirán el orderer y los peers, que extenderán de los archivos situados en la carpeta base.

Por último se define el CLI, que irá securizado con TLS por lo que se le pasarán los certificados correspondientes en las variables CORE\_PEER\_TLS\_CERT\_FILE, CORE\_PEER\_TLS\_KEY\_FILE y CORE\_PEER\_TLS\_ROOTCERT\_FILE

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2'

volumes:
  orderer.netcan.com:
  fci.federaciones.netcan.com:
  rsce.federaciones.netcan.com:
  tkc.federaciones.netcan.com:
  acw.federaciones.netcan.com:
  cvandalucia.colegiosveterinarios.netcan.com:
  cvaragon.colegiosveterinarios.netcan.com:
  cvasturias.colegiosveterinarios.netcan.com:
  cvillesbalears.colegiosveterinarios.netcan.com:
  cvcanarias.colegiosveterinarios.netcan.com:
  cvcantabria.colegiosveterinarios.netcan.com:
  cvcastillayleon.colegiosveterinarios.netcan.com:
  cvcastillalamancha.colegiosveterinarios.netcan.com:
  cvcataluna.colegiosveterinarios.netcan.com:
  cvcomunitatvalenciana.colegiosveterinarios.netcan.com:
  cvextremadura.colegiosveterinarios.netcan.com:
  cvgalicia.colegiosveterinarios.netcan.com:
  cvmadrid.colegiosveterinarios.netcan.com:
  cvmurcia.colegiosveterinarios.netcan.com:
  cvnavarra.colegiosveterinarios.netcan.com:
  cvpaisvasco.colegiosveterinarios.netcan.com:
  cvlarioja.colegiosveterinarios.netcan.com:

networks:
  netcan:

services:
  ca.federaciones:
    image: hyperledger/fabric-ca:1.4.0
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-federaciones
      - FABRIC_CA_SERVER_TLS_ENABLED=false
      - FABRIC_CA_SERVER_PORT=7054
    ports:
```

```
- "7054:7054"
  command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-
config/ca.federaciones.netcan.com-cert.pem --ca.keyfile /etc/hyperledger/fabric-ca-server-
config/ee7e9f81bf8ddfd11193119ffb4eb756a47d0aa326b921ff020179c30408826d_sk -b admin:adminpw -d --
cfg.identities.allowremove'
  volumes:
    - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
    - /home/hyperledger/work/src/fabric-samples/TFM/crypto-
config/peerOrganizations/federaciones.netcan.com/ca:/etc/hyperledger/fabric-ca-server-config
  container_name: ca_federaciones
  networks:
    - netcan

ca.colegiosveterinarios:
  image: hyperledger/fabric-ca:1.4.0
  environment:
    - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
    - FABRIC_CA_SERVER_CA_NAME=ca-colegiosveterinarios
    - FABRIC_CA_SERVER_TLS_ENABLED=false
    - FABRIC_CA_SERVER_PORT=7054
  ports:
    - "8054:7054"
  command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-
config/ca.colegiosveterinarios.netcan.com-cert.pem --ca.keyfile /etc/hyperledger/fabric-ca-server-
config/c605e45a562d0c01566a2e3bc452e653d7b92af1c04fdb0a79263b8df4dc0a7a_sk -b admin-cv:admin-cvpw -d --
cfg.identities.allowremove'
  volumes:
    - ./crypto-config/peerOrganizations/colegiosveterinarios.netcan.com/ca:/etc/hyperledger/fabric-ca-server-config
    - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
  container_name: ca_colegiosveterinarios
  networks:
    - netcan

orderer.netcan.com:
  extends:
    file: base/docker-compose-base.yaml
    service: orderer.netcan.com
  container_name: orderer.netcan.com
  networks:
    - netcan

fci.federaciones.netcan.com:
  container_name: fci.federaciones.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: fci.federaciones.netcan.com
  networks:
    - netcan

rsce.federaciones.netcan.com:
  container_name: rsce.federaciones.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: rsce.federaciones.netcan.com
  networks:
    - netcan

tkc.federaciones.netcan.com:
  container_name: tkc.federaciones.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: tkc.federaciones.netcan.com
  networks:
    - netcan

acw.federaciones.netcan.com:
  container_name: acw.federaciones.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: acw.federaciones.netcan.com
  networks:
    - netcan

cvandalucia.colegiosveterinarios.netcan.com:
  container_name: cvandalucia.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvandalucia.colegiosveterinarios.netcan.com
```

```
networks:
  - netcan

cvaragon.colegiosveterinarios.netcan.com:
  container_name: cvaragon.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvaragon.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvasturias.colegiosveterinarios.netcan.com:
  container_name: cvasturias.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvasturias.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvillesbalears.colegiosveterinarios.netcan.com:
  container_name: cvillesbalears.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvillesbalears.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvcanarias.colegiosveterinarios.netcan.com:
  container_name: cvcanarias.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvcanarias.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvcantabria.colegiosveterinarios.netcan.com:
  container_name: cvcantabria.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvcantabria.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvcastillayleon.colegiosveterinarios.netcan.com:
  container_name: cvcastillayleon.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvcastillayleon.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvcastillalamancha.colegiosveterinarios.netcan.com:
  container_name: cvcastillalamancha.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvcastillalamancha.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvcataluna.colegiosveterinarios.netcan.com:
  container_name: cvcataluna.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvcataluna.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvcomunitatvalenciana.colegiosveterinarios.netcan.com:
  container_name: cvcomunitatvalenciana.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvcomunitatvalenciana.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvextremadura.colegiosveterinarios.netcan.com:
  container_name: cvextremadura.colegiosveterinarios.netcan.com
  extends:
```

```
file: base/docker-compose-base.yaml
service: cvextremadura.colegiosveterinarios.netcan.com
networks:
  - netcan

cvgalicia.colegiosveterinarios.netcan.com:
  container_name: cvgalicia.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvgalicia.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvmadrid.colegiosveterinarios.netcan.com:
  container_name: cvmadrid.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvmadrid.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvmurcia.colegiosveterinarios.netcan.com:
  container_name: cvmurcia.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvmurcia.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvnavarra.colegiosveterinarios.netcan.com:
  container_name: cvnavarra.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvnavarra.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvpaisvasco.colegiosveterinarios.netcan.com:
  container_name: cvpaisvasco.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvpaisvasco.colegiosveterinarios.netcan.com
  networks:
    - netcan

cvlarioja.colegiosveterinarios.netcan.com:
  container_name: cvlarioja.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: cvlarioja.colegiosveterinarios.netcan.com
  networks:
    - netcan

cli:
  container_name: cli
  image: hyperledger/fabric-tools:1.4.0
  tty: true
  stdin_open: true
  environment:
    - GOPATH=/opt/gopath
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - FABRIC_LOGGING_SPEC=INFO
    - CORE_PEER_ID=cli
    - CORE_PEER_ADDRESS=fci.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
    - CORE_PEER_TLS_ENABLED=true
    - CORE_PEER_TLS_CERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/peers/fci.federaciones.netcan.com/tls/server.crt
    - CORE_PEER_TLS_KEY_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/peers/fci.federaciones.netcan.com/tls/server.key
    - CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/peers/fci.federaciones.netcan.com/tls/ca.crt
    - CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/users/Admin@Federaciones.netcan.com/msp
```

```
working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
command: /bin/bash
volumes:
  - /var/run:/host/var/run/
  - /home/hyperledger/work/src/fabric-samples/TFM/chaincode:/opt/gopath/src/github.com/chaincode/
  - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
  - ./scripts:/opt/gopath/src/github.com/hyperledger/fabric/peer/scripts/
  - ./channel-artifacts:/opt/gopath/src/github.com/hyperledger/fabric/peer/channel-artifacts
depends_on:
  - orderer.netcan.com
  - fci.federaciones.netcan.com
  - rsce.federaciones.netcan.com
  - tkc.federaciones.netcan.com
  - acw.federaciones.netcan.com
  - cvandalucia.colegiosveterinarios.netcan.com
  - cvaragon.colegiosveterinarios.netcan.com
  - cvasturias.colegiosveterinarios.netcan.com
  - cvillesbalears.colegiosveterinarios.netcan.com
  - cvcanarias.colegiosveterinarios.netcan.com
  - cvcantabria.colegiosveterinarios.netcan.com
  - cvcastillayleon.colegiosveterinarios.netcan.com
  - cvcastillalamancha.colegiosveterinarios.netcan.com
  - cvcataluna.colegiosveterinarios.netcan.com
  - cvcomunitatvalenciana.colegiosveterinarios.netcan.com
  - cvextremadura.colegiosveterinarios.netcan.com
  - cvgalicia.colegiosveterinarios.netcan.com
  - cvmadrid.colegiosveterinarios.netcan.com
  - cvmurcia.colegiosveterinarios.netcan.com
  - cvnavarra.colegiosveterinarios.netcan.com
  - cvpaisvasco.colegiosveterinarios.netcan.com
  - cvlarioja.colegiosveterinarios.netcan.com
networks:
  - netcan
```

- docker-compose-couch.yaml

Este archivo configura la red para el uso de CouchDB en vez de levelDB, lo que permite realizar índices y redunda en una mayor facilidad de uso y la posibilidad de realización de consultas más complejas.

En este caso no se define usuario y contraseña por facilidad de uso al ser seis componentes en el grupo de desarrollo, aunque en producción deberá definirse un usuario y contraseña para la couchDB.

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#
version: '2'

networks:
  netcan:

services:
  couchdb0:
    container_name: couchdb0
    image: hyperledger/fabric-couchdb:0.4.15
    # Populate the COUCHDB_USER and COUCHDB_PASSWORD to set an admin user and password
    # for CouchDB. This will prevent CouchDB from operating in an "Admin Party" mode.
    environment:
      # - COUCHDB_USER=netcan
      # - COUCHDB_PASSWORD=cannet
      - COUCHDB_USER=
      - COUCHDB_PASSWORD=
    # Comment/Uncomment the port mapping if you want to hide/expose the CouchDB service,
    # for netcan map it to utilize Fauxton User Interface in dev environments.
    ports:
      - "5984:5984"
    networks:
```

```
- netcan

fci.federaciones.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb0:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb0

couchdb1:
  container_name: couchdb1
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "6984:5984"
  networks:
    - netcan

rsce.federaciones.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb1:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb1

couchdb2:
  .
  .
  .
```

- docker-compose-base.yaml

En este archivo se definen el orderer y los peers. En particular se establece que los peers extienden a su vez del archivo peer-base.yaml y el uso de TLS en el orderer pasando los certificados correspondientes en las variables ORDERER\_GENERAL\_TLS\_PRIVATEKEY, ORDERER\_GENERAL\_TLS\_CERTIFICATE y ORDERER\_GENERAL\_TLS\_ROOTCAS

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2'

services:

  orderer.netcan.com:
    container_name: orderer.netcan.com
    image: hyperledger/fabric-orderer:1.4.0
    environment:
      - ORDERER_GENERAL_LOGLEVEL=INFO
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
      - ORDERER_GENERAL_GENESISMETHOD=file
      - ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block
      - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
      # enabled TLS
      - ORDERER_GENERAL_TLS_ENABLED=true
      - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=tfm_netcan
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric
    command: orderer
    volumes:
      - ../channel-artifacts/genesis.block:/var/hyperledger/orderer/orderer.genesis.block
```



```
- ../crypto-config/ordererOrganizations/netcan.com/orderers/orderer.netcan.com/msp:/var/hyperledger/orderer/msp
- ../crypto-config/ordererOrganizations/netcan.com/orderers/orderer.netcan.com/tls:/var/hyperledger/orderer/tls
- orderer.netcan.com:/var/hyperledger/production/orderer
ports:
  - 7050:7050

fci.federaciones.netcan.com:
  container_name: fci.federaciones.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=fci.federaciones.netcan.com
    - CORE_PEER_ADDRESS=fci.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=rsce.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=fci.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/fci.federaciones.netcan.com/msp:/etc/hyperledger/fabric/msp
- ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/fci.federaciones.netcan.com/tls:/etc/hyperledger/fabric/tls
- fci.federaciones.netcan.com:/var/hyperledger/production
ports:
  - 7051:7051
  - 7053:7053

rsce.federaciones.netcan.com:
  container_name: rsce.federaciones.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=rsce.federaciones.netcan.com
    - CORE_PEER_ADDRESS=rsce.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=rsce.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=fci.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/rsce.federaciones.netcan.com/msp:/etc/hyperledger/fabric/msp
- ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/rsce.federaciones.netcan.com/tls:/etc/hyperledger/fabric/tls
- rsce.federaciones.netcan.com:/var/hyperledger/production
ports:
  - 8051:7051
  - 8053:7053
.
```

- peer-base.yaml

Establece la configuración común a todos los peers, incluido el uso de TLS y sus correspondientes certificados

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2'

services:
  peer-base:
    image: hyperledger/fabric-peer:1.4.0
    environment:
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=tfm_netcan
      - FABRIC_LOGGING_SPEC=INFO
      - CORE_PEER_TLS_ENABLED=true
      - CORE_PEER_GOSSIP_USELEADERELECTION=true
```

```
- CORE_PEER_GOSSIP_ORGLEADER=false  
- CORE_PEER_PROFILE_ENABLED=true  
- CORE_PEER_ENDORSER_ENABLED=true  
- CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt  
- CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key  
- CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt  
working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer  
command: peer node start
```

## Puesta en marcha del proyecto

### RED INICIAL SIN TLS

Los archivos de configuración pueden encontrarse en el repositorio Github mencionado anteriormente:

[https://github.com/DFLBB/TFM\\_archs](https://github.com/DFLBB/TFM_archs)

En la carpeta Arquitectura bajo el nombre TFM\_sin\_TLS.tar

Se trata de una copia completa del proyecto.

Los scripts utilizados son los que se encuentran en la carpeta scripts2 del repositorio github.

Lo más relevante de esta versión es que no utiliza TLS, tal y como puede verse en los archivos de configuración docker-compose-cli.yaml, docker-compose-base.yaml y peer-base.yaml:

```
cli:
  container_name: cli
  image: hyperledger/fabric-tools:1.4.0
  tty: true
  stdin_open: true
  environment:
    - GOPATH=/opt/gopath
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - FABRIC_LOGGING_SPEC=INFO
    - CORE_PEER_ID=cli
    - CORE_PEER_ADDRESS=fci.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
    - CORE_PEER_TLS_ENABLED=false
    - CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/users/Admin@federaciones.netcan.com/msp
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
```

Una vez instalados los prerequisites y preparado el servidor para el uso de Hyperledger Fabric y una vez preparada la carpeta del proyecto en la que tendremos los archivos de configuración, la carpeta base con sus archivos de configuración, la carpeta canal-artifacts vacía (únicamente con un archivo .gitkeep vacío) y los scripts de inicio y parada del proyecto comenzamos con la instalación de la red.

Desde la carpeta del proyecto situada en /home/hyperledger/work/src/fabric-samples/TFM lanzaremos los siguientes comandos:

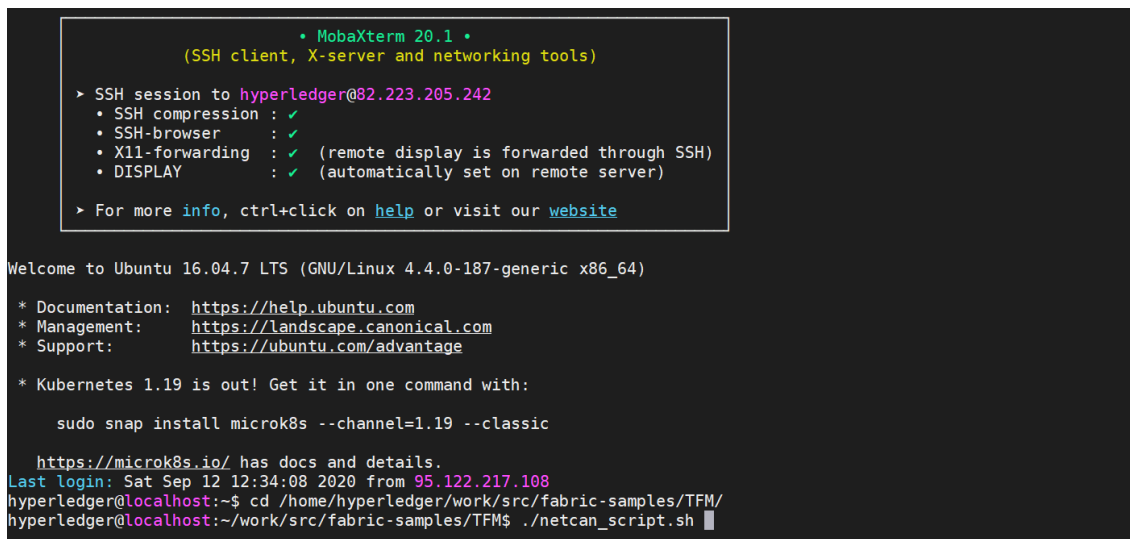
```
export PATH=/home/hyperledger/work/src/fabric-samples/bin:$PATH
# Generar material criptográfico /// NO HACER SI SE REINICIA Y NO SE QUIERE BORRAR
../bin/cryptogen generate --config=./crypto-config.yaml
export FABRIC_CFG_PATH=$PWD
# Generar EL bloque génesis
../bin/configtxgen -profile NetCanOrdererGenesis -outputBlock ./channel-artifacts/genesis.block

# Generar la configuración del canal
export CHANNEL_NAME=netcanchannel &&../bin/configtxgen -profile NetCanChannel -
outputCreateChannelTx ./channel-artifacts/channel.tx -channelID $CHANNEL_NAME

# Definir los pares de anclaje
../bin/configtxgen -profile NetCanChannel -outputAnchorPeersUpdate ./channel-
artifacts/FederacionesMSPanchors.tx -channelID $CHANNEL_NAME -asOrg
FederacionesMSP

../bin/configtxgen -profile NetCanChannel -outputAnchorPeersUpdate ./channel-
artifacts/ColegiosVeterinariosMSPanchors.tx -channelID $CHANNEL_NAME -asOrg
ColegiosVeterinariosMSP
```

En este punto se ha generado el material criptográfico en una carpeta llamada crypto-config y se han generado los artefactos en la carpeta channel-artifacts y podemos llamar al script netcan\_script.sh que levantará y configurará la red y los chaincodes desarrollados utilizando el resto de scripts explicados anteriormente.



```
• MobaXterm 20.1 •
(SSH client, X-server and networking tools)

> SSH session to hyperledger@82.223.205.242
• SSH compression : ✓
• SSH-browser : ✓
• X11-forwarding : ✓ (remote display is forwarded through SSH)
• DISPLAY : ✓ (automatically set on remote server)

> For more info, ctrl+click on help or visit our website

Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-187-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

* Kubernetes 1.19 is out! Get it in one command with:

  sudo snap install microk8s --channel=1.19 --classic

  https://microk8s.io/ has docs and details.
Last login: Sat Sep 12 12:34:08 2020 from 95.122.217.108
hyperledger@localhost:~$ cd /home/hyperledger/work/src/fabric-samples/TFM/
hyperledger@localhost:~/work/src/fabric-samples/TFM$ ./netcan_script.sh
```



ntic  
master  
revolucionamos la comunicación

[illegible]

## Se instalan e instancian los chaincodes

```
*****
Carga inicial de datos
*****
Cargando datos de PERFILES DE PERSONAS
*****
2020-09-12 12:43:28.443 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"48"
2020-09-12 12:43:28.622 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"{\docType\": \"CONTADOR\", \"IDMaximo\": 48}"
2020-09-12 12:43:28.732 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"{\docType\": \"PERFILES_PERSONAS\", \"IDPerfilPersona\": 0, \"IDPersona\": 0, \"CODPerfil\": \"ADMINISTRA
DOR\", \"FechaAlta\": \"2020-09-12 12:40:11.817017251 +0000 UTC m=+0.113422036\", \"FechaBaja\": \"2020-09-12 12:43:28.71
0732619 +0000 UTC m=+197.007137410\"}"
*****
Cargando datos de GRUPOS
*****
2020-09-12 12:43:28.950 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"12"
2020-09-12 12:43:29.154 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"{\docType\": \"CONTADOR\", \"IDMaximo\": 12}"
*****
Cargando datos de RAZAS
*****
2020-09-12 12:43:29.706 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"346"
2020-09-12 12:43:29.816 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"{\docType\": \"CONTADOR\", \"IDMaximo\": 346}"
*****
Cargando datos de PERSONAS
*****
```

Y se realiza la carga inicial de datos a la blockchain

Al estar implementada la CouchDB en el puerto 5984 podemos conectarnos para ver si todas las cargas han ido correctamente y están los datos (a pesar de que al realizar las cargas la CouchDB responde con un “OK”), comprobando que efectivamente todos los datos se han cargado a la CouchDB.

Name	Size	# of Docs	Actions
_replicator	2.3 KB	1	
_users	2.3 KB	1	
netcanchannel_	18.4 KB	2	
netcanchannel_atijos	31.4 KB	106	
netcanchannel_isc	6.0 KB	9	
netcanchannel_microchips	1.5 MB	4762	
netcanchannel_perfiles	16.3 KB	51	
netcanchannel_perros	3.1 MB	9524	
netcanchannel_personas	385.8 KB	1003	
netcanchannel_razas	103.9 KB	352	
netcanchannel_solitudes	0.5 KB	1	
netcanchannel_vacunas	37.9 KB	104	
netcanchannel_veterinarios	8.5 KB	24	

Pudiendo ver los datos cargados en cada tabla:

_id	_rev	IDMaximo	docType	~version	FechaAlta	FechaBaja	FechaDefund	FechaNacimiento	IDBajo	IDPerro	IDPerroMadre	IDPerroPadre	IDRaza	IDSexo	Nombre
PERROS1-12657...	5759	CONTA...	DCgMB...												
PERROS1-1-bae62...		PERROS	DCgMB...		2015-10...			2015-10...	1	1	797	800	126	0	BRISA
PERRO...-1-8163a...		PERROS	DCgMB...		2004-05...	2013-09...	2013-09...	2004-04...	1	10	0	0	126	0	CORAZ...
PERRO...-1-642a6...		PERROS	DCgMB...		2008-04...			2008-04...	2	100	49	50	126	0	VILMA
PERRO...-1-71465...		PERROS	DCgMB...		2014-04...			2014-03...	2	1000	402	413	126	1	ZAIMON
PERRO...-1-29134...		PERROS	DCgMB...		2014-04...			2014-03...	2	1001	414	53	126	1	STORM
PERRO...-1-bae80...		PERROS	DCgMB...		2014-04...	2015-07...	2015-06...	2014-03...	2	1002	414	53	126	1	PABLO
PERRO...-1-5c006...		PERROS	DCgMB...		2014-04...			2014-03...	2	1003	414	53	126	0	VICTORIA
PERRO...-1-99152...		PERROS	DCgMB...		2014-04...	2014-07...	2014-07...	2014-03...	2	1004	414	53	126	1	SATANAS
PERRO...-1-24026...		PERROS	DCgMB...		2013-12...	2020-08...	2020-07...	2013-11...	2	1005	417	405	126	1	BLACKY
PERRO...-1-17695...		PERROS	DCgMB...		2013-12...			2013-11...	2	1006	417	405	126	0	ROMELIA
PERRO...-1-c578...		PERROS	DCgMB...		2013-12...			2013-11...	2	1007	417	405	126	1	TINO
PERRO...-1-ad42...		PERROS	DCgMB...		2013-12...	2018-08...	2018-08...	2013-11...	2	1008	417	405	126	0	BRIZIA
PERRO...-1-8024...		PERROS	DCgMB...		2013-12...			2013-12...	3	1009	20	523	126	1	JUNIOR
PERRO...-1-12ee0...		PERROS	DCgMB...		2008-04...	2008-09...	2008-08...	2008-04...	2	101	49	50	126	1	ZIAN
PERRO...-1-c9c71...		PERROS	DCgMB...		2014-01...			2013-12...	3	1010	20	523	126	0	CALCETA
PERRO...-1-6f99...		PERROS	DCgMB...		2014-05...			2014-05...	3	1011	56	522	126	1	LAYO
PERRO...-1-ca51f...		PERROS	DCgMB...		2014-06...			2014-05...	3	1012	56	522	126	0	YGRITTE

Podemos comprobar que todos los dockers están corriendo correctamente mediante el comando:

`docker ps -a`

Obteniendo la siguiente salida en la que puede comprobarse que la red está corriendo correctamente:

```
hyperledger@localhost:~/work/src/fabric-samples/TFM$ docker ps -a
CONTAINER ID        IMAGE                                     COMMAND                  CREATED
STATUS            PORTS                               NAMES
8477766e969dc      dev-fci.federaciones.netcan.com-veterinarios-1.0.0-130c44c62ff0205de3b79eea6e6cfdb54e461931866f81fb4bf3ad7300101fd "chaincode -peer.add..." 13 minutes ago      Up 13 minutes
dev-fci.federaciones.netcan.com-veterinarios-1.0.0
```



8d4de91db75c	dev-fci.federaciones.netcan.com-vacunas-1.0.0-				
c0712aaa2c66f0eb4d3e68b9dca83fb740be117b8e8c9e54b50a94da72f3f366	"chaincode -peer.add..."	14 minutes ago	Up 14		
minutes	dev-fci.federaciones.netcan.com-vacunas-1.0.0				
969ccea9c3a4	dev-fci.federaciones.netcan.com-solicitudes-1.0.0-				
de8c71563a9bf41548efc3efd696676c22d66525e1f312b37574d60c1b17aab0	"chaincode -peer.add..."	14 minutes ago	Up 14		
minutes	dev-fci.federaciones.netcan.com-solicitudes-1.0.0				
ab54d7839973	dev-fci.federaciones.netcan.com-razas-1.0.0-				
b5e08411bd92c32ee9f31c54d803da0da9dbc07c35b86bdd9dbc66c7d9379d6c	"chaincode -peer.add..."	15 minutes ago	Up		
15 minutes	dev-fci.federaciones.netcan.com-razas-1.0.0				
91ade708082e	dev-fci.federaciones.netcan.com-personas-1.0.0-				
ec9fd3c3016785411df3b0cfb73fb0c4a2ff8f249ada3ae8604d52579cf2ca92	"chaincode -peer.add..."	15 minutes ago	Up 15		
minutes	dev-fci.federaciones.netcan.com-personas-1.0.0				
8e8a58923aee	dev-fci.federaciones.netcan.com-perros-1.0.0-				
46b41e94180f7b7b9d3a25abd665310a9fe68ed5639e9ab474ddf30c60d278f3	"chaincode -peer.add..."	16 minutes ago	Up 16		
minutes	dev-fci.federaciones.netcan.com-perros-1.0.0				
3a2a8692d58f	dev-fci.federaciones.netcan.com-perfiles-1.0.0-				
ee62113a5af70130e6a0df3310ec7555afcaee7f3659eb7a4191efd56bfa5d8	"chaincode -peer.add..."	17 minutes ago	Up 17		
minutes	dev-fci.federaciones.netcan.com-perfiles-1.0.0				
ffb55ddfcc92	dev-fci.federaciones.netcan.com-microchips-1.0.0-				
fb61a6a31883fa2a1b3280a342942d5462730c2b4264e4e39e8f7cec9785092c	"chaincode -peer.add..."	17 minutes ago	Up 17		
minutes	dev-fci.federaciones.netcan.com-microchips-1.0.0				
e4be7a159d3e	dev-fci.federaciones.netcan.com-afijos-1.0.0-				
33d55e901eadaa70caea32134c737a7fc4cae1ca66c9f8a937444c5383c8dbe	"chaincode -peer.add..."	18 minutes ago	Up 18		
minutes	dev-fci.federaciones.netcan.com-afijos-1.0.0				
49f928e0ac99	hyperledger/fabric-tools:1.4.0			"/bin/bash"	19 minutes
ago	Up 19 minutes				
eef9afbcdc31	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes	Up 19 minutes				
cvcastillalamancha.colegiosveterinarios.netcan.com	0.0.0.0:19051->7051/tcp,			0.0.0.0:19053->7053/tcp	
695f8a37dd86	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
6c8455958aaf	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
aa271f0e3eaa	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
cvcomunitatvalenciana.colegiosveterinarios.netcan.com	0.0.0.0:21051->7051/tcp,			0.0.0.0:21053->7053/tcp	
91468eeb882c	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
7c0675f67108	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
6254cb2d73bd	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
f9aeb8c5ea47	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
2a5892757a6f	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
ca243207d8f3	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
4a028d76740d	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
3a24bd3fcd56	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
2979b1ac8015	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
45b835a6641a	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 20 minutes				
7de244f6f55f	hyperledger/fabric-peer:1.4.0			"peer node start"	20 minutes
ago	Up 19 minutes				
5315084811fa	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
286bb539e7e8	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				
0aca3174afbe	hyperledger/fabric-peer:1.4.0			"peer node start"	20
minutes ago	Up 19 minutes				



21697bb1fac3	hyperledger/fabric-peer:1.4.0		"peer node start"	20
minutes ago	Up 20 minutes	0.0.0.0:10051->7051/tcp, 0.0.0.0:10053->7053/tcp	acw.federaciones.netcan.com	
5efbfe4d1272	hyperledger/fabric-peer:1.4.0		"peer node start"	20
minutes ago	Up 20 minutes	0.0.0.0:23051->7051/tcp, 0.0.0.0:23053->7053/tcp	cvgalicia.colegiosveterinarios.netcan.com	
84de82313932	hyperledger/fabric-peer:1.4.0		"peer node start"	20
minutes ago	Up 20 minutes	0.0.0.0:12051->7051/tcp, 0.0.0.0:12053->7053/tcp	cvandalucia.colegiosveterinarios.netcan.com	
b65b2075c00a	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:15984->5984/tcp	couchdb10	
16b300ca4219	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:17984->5984/tcp	couchdb12	
62cb45063c0d	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:18984->5984/tcp	couchdb13	
09283c58a6b9	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:14984->5984/tcp	couchdb9	
5133bdb07abf	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:16984->5984/tcp	couchdb11	
40eb43a49b91	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:12984->5984/tcp	couchdb7	
a46e2961fe0b	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:19984->5984/tcp	couchdb14	
a3785bbac749	hyperledger/fabric-ca:1.4.0		"sh -c 'fabric-ca-se..."	20
minutes ago	Up 20 minutes	0.0.0.0:7054->7054/tcp	ca_federaciones	
8607f1b75cbb	hyperledger/fabric-ca:1.4.0		"sh -c 'fabric-ca-se..."	20
minutes ago	Up 20 minutes	0.0.0.0:8054->7054/tcp	ca_colegiosveterinarios	
279fd7a1ce36	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:9984->5984/tcp	couchdb4	
acb5fe15bd11	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp	couchdb2	
88405564967c	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:13984->5984/tcp	couchdb8	
36b97918a0ba	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:24984->5984/tcp	couchdb19	
692c3207ca92	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:23984->5984/tcp	couchdb18	
fc096cd340a9	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:22984->5984/tcp	couchdb17	
6dc3acfc6566	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:25984->5984/tcp	couchdb20	
f675298d9f1d	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp	couchdb0	
4ba6d4aae8b9	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp	couchdb1	
e4789a01413e	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:20984->5984/tcp	couchdb15	
4e6b0db891c8	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:21984->5984/tcp	couchdb16	
e38a6faa457d	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:11984->5984/tcp	couchdb6	
1e4e76f0147f	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:10984->5984/tcp	couchdb5	
90c247abc1f7	hyperledger/fabric-couchdb:0.4.15		"tini -- /docker-ent..."	20
minutes ago	Up 20 minutes	4369/tcp, 9100/tcp, 0.0.0.0:8984->5984/tcp	couchdb3	
349d2601803d	hyperledger/fabric-orderer:1.4.0		"orderer"	20 minutes
ago	Up 20 minutes	0.0.0.0:7050->7050/tcp	orderer.netcan.com	

También podemos comprobar que se ha creado la red `tfm_netcan` que se definía en los archivos `docker-compose-base.yaml` y `peer-base.yaml`, en las variables `CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE`, utilizando el comando:

Docker network ls

```
hyperledger@localhost:~/work/src/fabric-samples/TFM$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
53a038bc3615        bridge              bridge              local
ea3f5cc505fa        host                host                local
619f53f028c1        none                null                local
19b1f0c96b46        tfm_netcan          bridge              local
```

Una vez levantada la red y comprobado que está funcionando correctamente podemos entrar a la CA para comprobar si está funcionando correctamente.

Para ello primero entramos a los logs de la CA para ver si está corriendo correctamente:

`docker logs -f ca_federaciones`

```
2020/09/12 12:37:06 [DEBUG] CA initialization successful
2020/09/12 12:37:06 [DEBUG] Initializing Idemix issuer...
2020/09/12 12:37:07 [INFO] The issuer key was successfully stored. The public key is at: /etc/hyperledger/fabric-ca-server/IssuerPublicKey, secret key is at: /etc/hyperledger/fabric-ca-server/msp/keystore/IssuerSecretKey
2020/09/12 12:37:07 [DEBUG] Initializing revocation authority for issuer 'ca-federaciones'
2020/09/12 12:37:07 [DEBUG] Initialize Idemix issuer revocation key material
2020/09/12 12:37:07 [INFO] Idemix issuer revocation public and secret keys were generated for CA 'ca-federaciones'
2020/09/12 12:37:07 [INFO] The revocation key was successfully stored. The public key is at: /etc/hyperledger/fabric-ca-server/IssuerRevocationPublicKey, private key is at: /etc/hyperledger/fabric-ca-server/msp/keystore/IssuerRevocationPrivateKey
2020/09/12 12:37:07 [DEBUG] Initializing nonce manager for issuer 'ca-federaciones'
2020/09/12 12:37:07 [INFO] Home directory for default CA: /etc/hyperledger/fabric-ca-server
2020/09/12 12:37:07 [DEBUG] 1 CA instance(s) running on server
2020/09/12 12:37:07 [INFO] Listening on http://0.0.0.0:7054
```

Abrimos una nueva consola, entramos al docker de la CA y nos logamos con el usuario `admin` de la CA mediante los comandos:

```
docker exec -it ca_federaciones bash
cd etc/hyperledger/fabric-ca-server
fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
```

```
hyperledger@localhost:~/work/src/fabric-samples/TFM$ docker exec -it ca_federaciones bash
root@a3785bbac749:/# cd etc/hyperledger/fabric-ca-server
root@a3785bbac749:/etc/hyperledger/fabric-ca-server# fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
2020/09/12 13:10:02 [INFO] Created a default configuration file at /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yaml
2020/09/12 13:10:02 [INFO] generating key: &{A:ecdsa S:256}
2020/09/12 13:10:02 [INFO] encoded CSR
2020/09/12 13:10:02 [INFO] Stored client certificate at /etc/hyperledger/fabric-ca-server/msp/signcerts/cert.pem
2020/09/12 13:10:02 [INFO] Stored root CA certificate at /etc/hyperledger/fabric-ca-server/msp/cacerts/localhost-7054.pem
2020/09/12 13:10:02 [INFO] Stored Issuer public key at /etc/hyperledger/fabric-ca-server/msp/IssuerPublicKey
2020/09/12 13:10:02 [INFO] Stored Issuer revocation public key at /etc/hyperledger/fabric-ca-server/msp/IssuerRevocationPublicKey
root@a3785bbac749:/etc/hyperledger/fabric-ca-server#
```

Una vez logados creamos un nuevo usuario, pero antes hay que crear la afiliación si no existe

```
fabric-ca-client affiliation add federaciones
```

```
fabric-ca-client affiliation add federaciones.RSCE
```

```
fabric-ca-client register --id.name Daniel --id.secret 22349 --id.affiliation federaciones.RSCE --id.attrs 'hf.Revoker=true,admin=true:ecert'
```

```
root@a3785bbac749:/etc/hyperledger/fabric-ca-server# fabric-ca-client affiliation add federaciones
2020/09/12 13:11:10 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yaml
Successfully added affiliation: federaciones
root@a3785bbac749:/etc/hyperledger/fabric-ca-server# fabric-ca-client affiliation add federaciones.RSCE
2020/09/12 13:11:17 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yaml
Successfully added affiliation: federaciones.RSCE
root@a3785bbac749:/etc/hyperledger/fabric-ca-server# fabric-ca-client register --id.name Daniel --id.secret 22349 --id.affiliation federaciones.RSCE --id.attrs 'hf.Revoker=true,admin=true:ecert'
2020/09/12 13:11:23 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yaml
Password: 22349
```

Comprobamos que esté registrado

```
fabric-ca-client enroll -u http://Daniel:22349@localhost:7054 -M msp
```

Y volvemos al usuario admin para revocar la identidad creada:

```
fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
```

```
fabric-ca-client identity remove Daniel
```

```
root@a3785bbac749:/etc/hyperledger/fabric-ca-server# fabric-ca-client enroll -u http://Daniel:22349@localhost:7054 -M msp
2020/09/12 13:13:43 [INFO] generating key: &{A:ecdsa S:256}
2020/09/12 13:13:43 [INFO] encoded CSR
2020/09/12 13:13:43 [INFO] Stored client certificate at /etc/hyperledger/fabric-ca-server/msp/signcerts/cert.pem
2020/09/12 13:13:43 [INFO] Stored root CA certificate at /etc/hyperledger/fabric-ca-server/msp/cacerts/localhost-7054.pem
2020/09/12 13:13:43 [INFO] Stored Issuer public key at /etc/hyperledger/fabric-ca-server/msp/IssuerPublicKey
2020/09/12 13:13:43 [INFO] Stored Issuer revocation public key at /etc/hyperledger/fabric-ca-server/msp/IssuerRevocationPublicKey
root@a3785bbac749:/etc/hyperledger/fabric-ca-server# fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
2020/09/12 13:13:52 [INFO] generating key: &{A:ecdsa S:256}
2020/09/12 13:13:52 [INFO] encoded CSR
2020/09/12 13:13:52 [INFO] Stored client certificate at /etc/hyperledger/fabric-ca-server/msp/signcerts/cert.pem
2020/09/12 13:13:52 [INFO] Stored root CA certificate at /etc/hyperledger/fabric-ca-server/msp/cacerts/localhost-7054.pem
2020/09/12 13:13:52 [INFO] Stored Issuer public key at /etc/hyperledger/fabric-ca-server/msp/IssuerPublicKey
2020/09/12 13:13:52 [INFO] Stored Issuer revocation public key at /etc/hyperledger/fabric-ca-server/msp/IssuerRevocationPublicKey
root@a3785bbac749:/etc/hyperledger/fabric-ca-server# fabric-ca-client identity remove Daniel
2020/09/12 13:13:59 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yaml
Successfully removed identity - Name: Daniel, Type: client, Affiliation: federaciones.RSCE, Max Enrollments: -1, Attributes: [{Name:hf.Revoker Value:true ECert:false} {Name:admin Value:true ECert:true} {Name:hf.EnrollmentID Value:Daniel ECert:true} {Name:hf.Type Value:client ECert:true} {Name:hf.Affiliation Value:federaciones.RSCE ECert:true}]
```

## INSTALACIÓN DE HYPERLEDGER EXPLORER

Para poder visualizar la información de la red en el navegador instalaremos Hyperledger Explorer, que nos permitirá ver los datos de la red, los bloques creados, las transacciones y los chaincodes.

Desde la carpeta `/home/hyperledger/work/src` instalaremos los prerequisites de Hyperledger Fabric. En particular estaremos atentos a las versiones ya que hasta el momento hemos utilizado Fabric 1.4.0, comprobando en las Release Notes de Hyperledger Fabric Explorer que deberemos utilizar la versión 8.11.1 de NodeJS y la versión 0.3.9.3 de Hyperledger Fabric Explorer.

### Release Notes

Hyperledger Explorer Version	Fabric Version Supported	NodeJS Version Supported
<a href="#">v1.1.2</a> (Aug 12, 2020)	<a href="#">v1.4.0 to v2.2.0</a>	<a href="#">12.16.x</a>
<a href="#">v1.1.1</a> (Jul 17, 2020)	<a href="#">v1.4.0 to v2.1.1</a>	<a href="#">12.16.x</a>
<a href="#">v1.1.0</a> (Jul 01, 2020)	<a href="#">v1.4.0 to v2.1.1</a>	<a href="#">12.16.x</a>
<a href="#">v1.0.0</a> (Apr 09, 2020)	<a href="#">v1.4.0 to v1.4.8</a>	<a href="#">10.19.x</a>
<a href="#">v1.0.0-rc3</a> (Apr 01, 2020)	<a href="#">v1.4.0 to v1.4.6</a>	<a href="#">10.19.x</a>
<a href="#">v1.0.0-rc2</a> (Dec 10, 2019)	<a href="#">v1.4.0 to v1.4.4</a>	<a href="#">8.11.x</a>
<a href="#">v1.0.0-rc1</a> (Nov 18, 2019)	<a href="#">v1.4.2</a>	<a href="#">8.11.x</a>
<a href="#">v0.3.9.5</a> (Sep 8, 2019)	<a href="#">v1.4.2</a>	<a href="#">8.11.x</a>
<a href="#">v0.3.9.4</a> (June 18, 2019)	<a href="#">v1.4.1</a>	<a href="#">8.11.x</a>
<a href="#">v0.3.9.3</a> (May 24, 2019)	<a href="#">v1.4</a>	<a href="#">8.11.x</a>
<a href="#">v0.3.9.2</a> (May 16, 2019)	<a href="#">v1.4</a>	<a href="#">8.11.x</a>
<a href="#">v0.3.9.1</a> (Feb 28, 2019)	<a href="#">v1.4</a>	<a href="#">8.11.x</a>
<a href="#">v0.3.9</a> (Feb 7, 2019)	<a href="#">v1.4</a>	<a href="#">8.11.x</a>
<a href="#">v0.3.8</a> (Dec 13, 2018)	<a href="#">v1.3</a>	<a href="#">8.x.x</a>

Partimos de la carpeta /home/hyperledger/work/src

```
cd ~/work/src
```

E instalamos los prerequisites (jq y NodeJS), pasando a la versión 8.11.1 de NodeJS.

```
sudo apt-get update
sudo apt-get install jq
nvm install 8.11.1
nvm use 8.11.1
```

A continuación es necesario instalar Postgresql ya que es la base de datos que utiliza Hyperledger Explorer, creando el usuario hyperledger

```
sudo apt-get install postgresql postgresql-contrib
```

Para conectarse a la consola utilizaremos:

```
sudo -u postgres psql // Para conectarse. También psql -h localhost -U hyperledger -d fabricexplorer
```

Y una vez dentro crearemos el usuario hyperledger y saldremos de la consola:

```
CREATE USER hyperledger PASSWORD 'hyperledger';
\q
```

Una vez configurado postgresql descargaremos el repositorio de Hyperledger Explorer y utilizaremos la versión 0.3.9.3:

```
git clone https://github.com/hyperledger/blockchain-explorer.git
cd blockchain-explorer
git checkout v0.3.9.3
```

Con Hyperledger Explorer ya instalado podemos pasar a configurarlo. Para ello hay que modificar los siguientes ficheros, que pueden encontrarse en el archivo Configuración\_explorer.tar de la carpeta Arquitectura del repositorio Github [https://github.com/DFLBB/TFM\\_archs](https://github.com/DFLBB/TFM_archs):

- ./app/explorerconfig.json, en el que tendremos que indicar el usuario y la contraseña establecidos en la base de datos postgresql

```
{
  "persistence": "postgresql",
  "platforms": ["fabric"],
  "postgresql": {
    "host": "127.0.0.1",
    "port": "5432",
    "database": "fabricexplorer",
    "username": "hyperledger",
    "passwd": "hyperledger"
  },
  "sync": {
    "type": "local",
    "platform": "fabric",
    "blocksSyncTime": "1"
  },
  "jwt": {
    "secret": "a secret phrase!!",
    "expiresIn": "2h"
  }
}
```

- ./app/platform/fabric/config.json, en donde hay que indicar el nombre de la red y el archivo de configuración

```
{
  "network-configs": {
    "netcan": {
      "name": "netcan",
      "profile": "./connection-profile/netcan_15.json"
    }
  },
  "license": "Apache-2.0"
}
```

- ./app/platform/fabric/connection-profile/netcan.json, donde definiremos la red, la organización y el canal al que se conectará el Hyperledger Explorer

```
{
  "name": "netcan",
  "version": "1.0.0",
  "license": "Apache-2.0",
  "client": {
    "tlsEnable": false,
    "adminUser": "admin",
    "adminPassword": "adminpw",
    "enableAuthentication": false,
    "organization": "Federaciones",
    "connection": {
      "timeout": {
        "peer": {
          "endorser": "300"
        },
        "orderer": "300"
      }
    }
  }
}
```

```
- "channels": {  
-   "netcanchannel": {  
-     "peers": {  
-       "fci.federaciones.netcan.com": {}  
-     },  
-     "connection": {  
-       "timeout": {  
-         "peer": {  
-           "endorser": "6000",  
-           "eventHub": "6000",  
-           "eventReg": "6000"  
-         }  
-       }  
-     }  
-   },  
-   "organizations": {  
-     "Federaciones": {  
-       "mspid": "FederacionesMSP",  
-       "fullpath": true  
-     }  
-   },  
-   "peers": {  
-     "fci.federaciones.netcan.com": {  
-       "url": "grpc://localhost:7051",  
-       "eventUrl": "grpc://localhost:7053",  
-       "grpcOptions": {  
-         "ssl-target-name-override": "fci.federaciones.netcan.com"  
-       }  
-     }  
-   }  
- }
```

Además estableceremos las variables de entorno ya que a veces dan problemas:

```
export DATABASE_HOST=127.0.0.1  
export DATABASE_PORT=5432  
export DATABASE_DATABASE=fabricexplorer  
export DATABASE_USERNAME=hyperledger  
export DATABASE_PASSWD=hyperledger
```

A continuación crearemos la base de datos de Hyperledger Explorer en postgresql:

```
cd /home/hyperledger/work/src/blockchain-explorer/app/persistence/fabric/postgreSQL/  
chmod -R 775 db  
cd db/  
./createdb.sh
```

Y por último ejecutaremos el script que instala Hyperledger Explorer:

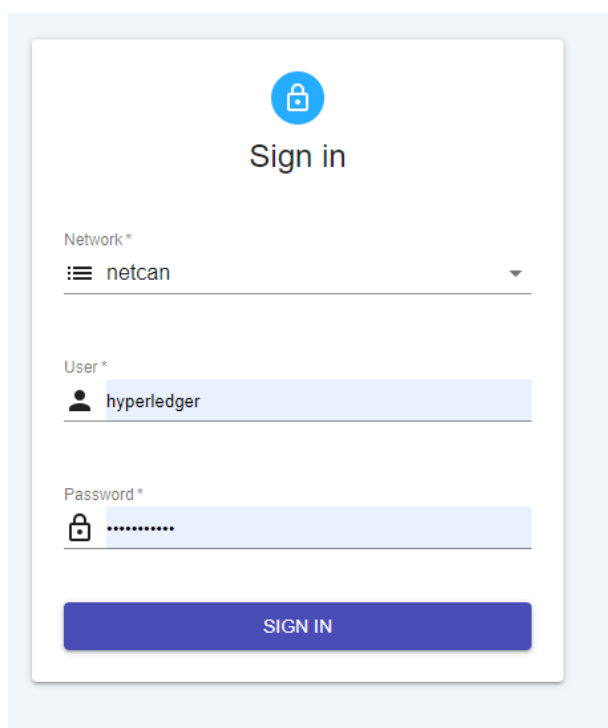
```
cd /home/hyperledger/work/src/blockchain-explorer/  
./main.sh install
```

Con la red levantada se utiliza el script start para inicial el servicio

```
./start.sh
```

Y podremos ver los logs en las carpetas app, console y db situadas en /home/hyperledger/work/src/blockchain-explorer/logs/

Por último se puede acceder en el navegador en el puerto :8080, en donde podemos observar el resumen de la actividad:



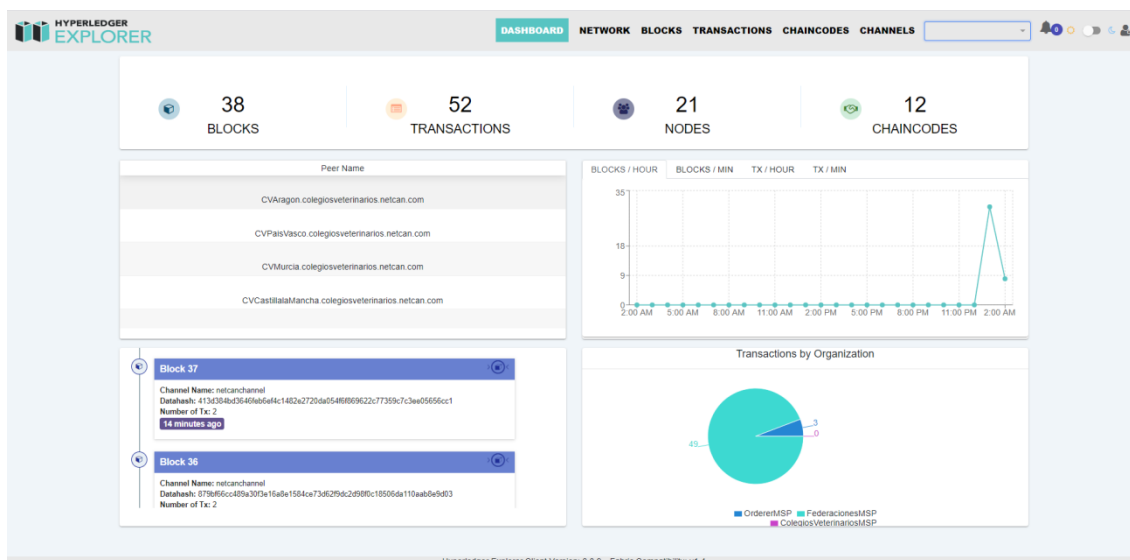
Sign in

Network \*  
netcan

User \*  
hyperledger

Password \*  
.....

SIGN IN





### La actividad en la red:

**HYPERLEDGER EXPLORER** DASHBOARD **NETWORK** BLOCKS TRANSACTIONS CHAINCODES CHANNELS

Peer Name	Request Url	Peer Type	MSPID	Ledger Height		
				High	Low	Unsigned
CVAragon.colegiosveterinario...	CVAragon.colegiosveterinario...	PEER	ColegiosVeterinariosMSP	0	38	true
CVPaísVasco.colegiosveterin...	CVPaísVasco.colegiosveterin...	PEER	ColegiosVeterinariosMSP	0	38	true
CVMurcia.colegiosveterinarios...	CVMurcia.colegiosveterinarios...	PEER	ColegiosVeterinariosMSP	0	38	true
CVCastillaLaMancha.colegiosv...	CVCastillaLaMancha.colegiosv...	PEER	ColegiosVeterinariosMSP	0	38	true
CVNavarra.colegiosveterinari...	CVNavarra.colegiosveterinari...	PEER	ColegiosVeterinariosMSP	0	38	true

Previous Page 1 of 5 5 rows Next

Hyperledger Explorer Client Version: 0.3.9 Fabric Compatibility: v1.4

### Los bloques creados:

**HYPERLEDGER EXPLORER** DASHBOARD NETWORK **BLOCKS** TRANSACTIONS CHAINCODES CHANNELS

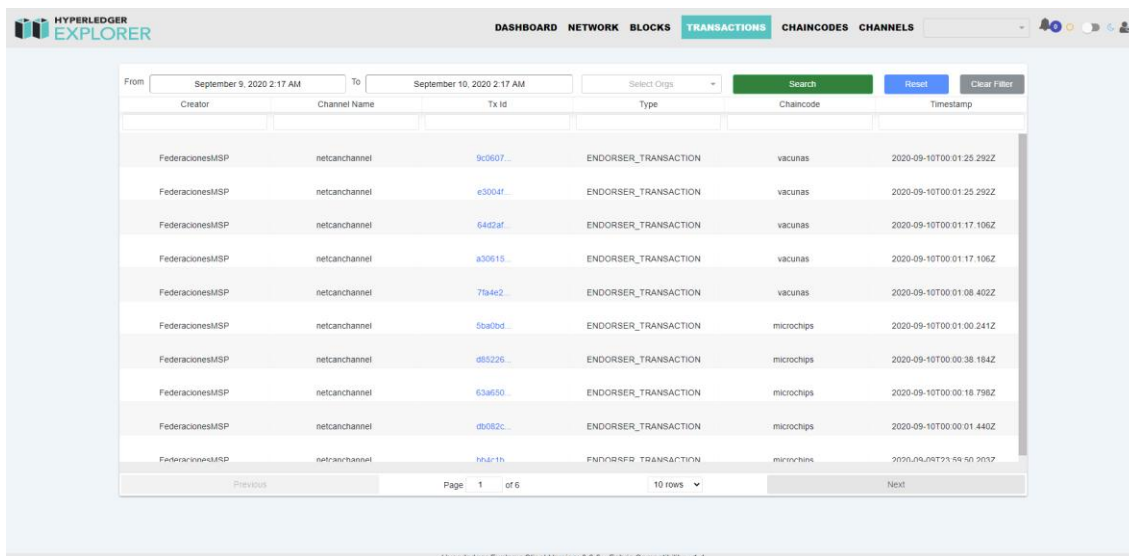
From September 9, 2020 2:17 AM To September 10, 2020 2:17 AM Select Orgs Search Reset Clear Filter

Block Number	Channel Name	Number of Tx	Data Hash	Block Hash	Previous Hash	Transactions
37	netcanchannel	2	413d38 ...	48e3be ...	7a70a5 ...	1972bb ... d72c7f ...
36	netcanchannel	2	879d6 ...	7a70a5 ...	971e35 ...	9c0607 ... e3004f ...
35	netcanchannel	2	b8b57e ...	971e35 ...	c28d5a ...	64d2af ... a30615 ...
34	netcanchannel	1	0a608 ...	c28d5a ...	d9eb44 ...	79a4e2 ...
33	netcanchannel	1	d71e18 ...	d9eb44 ...	863a80 ...	5ba0bd ...
32	netcanchannel	1	87367b ...	863a80 ...	55c662 ...	d85226 ...
31	netcanchannel	1	e0633c ...	55c662 ...	e7126a ...	63a650 ...

Previous Page 1 of 4 10 rows Next

Hyperledger Explorer Client Version: 0.3.9 Fabric Compatibility: v1.4

Las transacciones:

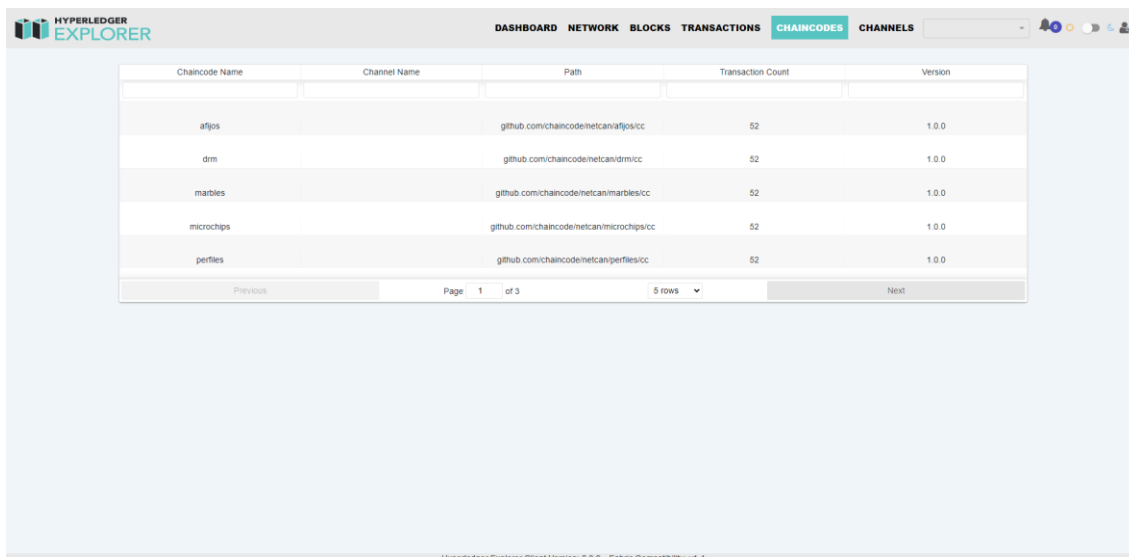


Hyperledger Explorer Client Version: 0.3.9 Fabric Compatibility: v1.4

From	To	Select Orgs	Search	Reset	Clear Filter
Creator	Channel Name	Tx Id	Type	Chaincode	Timestamp
FederacionesMSP	netcanchannel	9c3607...	ENDORSE_TRANSACTION	vacunas	2020-09-10T00:01:25.292Z
FederacionesMSP	netcanchannel	e3004f...	ENDORSE_TRANSACTION	vacunas	2020-09-10T00:01:25.292Z
FederacionesMSP	netcanchannel	6462af...	ENDORSE_TRANSACTION	vacunas	2020-09-10T00:01:17.106Z
FederacionesMSP	netcanchannel	a30615...	ENDORSE_TRANSACTION	vacunas	2020-09-10T00:01:17.106Z
FederacionesMSP	netcanchannel	75a462...	ENDORSE_TRANSACTION	vacunas	2020-09-10T00:01:08.402Z
FederacionesMSP	netcanchannel	5ba0bd...	ENDORSE_TRANSACTION	microchips	2020-09-10T00:01:00.241Z
FederacionesMSP	netcanchannel	d85226...	ENDORSE_TRANSACTION	microchips	2020-09-10T00:00:38.184Z
FederacionesMSP	netcanchannel	63a650...	ENDORSE_TRANSACTION	microchips	2020-09-10T00:00:18.798Z
FederacionesMSP	netcanchannel	d0882c...	ENDORSE_TRANSACTION	microchips	2020-09-10T00:00:01.440Z
FederacionesMSP	netcanchannel	hba11h...	ENDORSE_TRANSACTION	microchips	2020-09-10T00:00:00.203Z

Previous Page 1 of 6 10 rows Next

Y la actividad de los chaincode:



Hyperledger Explorer Client Version: 0.3.9 Fabric Compatibility: v1.4

Chaincode Name	Channel Name	Path	Transaction Count	Version
afips		github.com/chaincode/netcan/afips/cc	52	1.0.0
drm		github.com/chaincode/netcan/drm/cc	52	1.0.0
marbles		github.com/chaincode/netcan/marbles/cc	52	1.0.0
microchips		github.com/chaincode/netcan/microchips/cc	52	1.0.0
perfiles		github.com/chaincode/netcan/perfiles/cc	52	1.0.0

Previous Page 1 of 3 5 rows Next

## RED CON TLS

A continuación se modifica la red para implementar el protocolo TLS en el orderer, CLI y peers.

Los archivos de configuración pueden encontrarse en el repositorio Github mencionado anteriormente:

[https://github.com/DFLBB/TFM\\_archs](https://github.com/DFLBB/TFM_archs)

En la carpeta Arquitectura bajo el nombre TFM\_con\_TLS.tar. Los scrips se encuentran en la carpeta scripts1 del repositorio.

Se trata de una copia completa del proyecto.

El procedimiento realizado es el mismo y los archivos a definir también, tratándose realmente de la red definida en el apartado 2.5 de este documento, por lo que no repetiremos los pasos a realizar.

Únicamente varían con respecto a la red anterior los archivos docker-compose-cli.yaml, docker-compose-base.yaml y peer-base.yaml en los que debe introducirse todo lo referente al TLS tal y como se explicó en el apartado 2.5, es decir, definir correctamente las siguientes variables:

- CORE\_PEER\_TLS\_ENABLED=true
- CORE\_PEER\_TLS\_CERT\_FILE, CORE\_PEER\_TLS\_KEY\_FILE y CORE\_PEER\_TLS\_ROOTCERT\_FILE
- ORDERER\_GENERAL\_TLS\_PRIVATEKEY, ORDERER\_GENERAL\_TLS\_CERTIFICATE y ORDERER\_GENERAL\_TLS\_ROOTCAS

El resultado final tras lanzar el script de inicio es el mismo:

```
*****
Cargando datos de VACUNACIONES
*****
2020-09-12 16:24:00.933 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:{"docType":"CONTADOR","IDMaximo":11}
2020-09-12 16:24:01.184 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"89"
2020-09-12 16:24:08.342 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:{"docType":"CONTADOR","IDMaximo":89}
2020-09-12 16:24:09.001 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:"514"
2020-09-12 16:24:16.162 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: st
atus:200 payload:{"docType":"CONTADOR","IDMaximo":512}
*****
DATOS INICIALES CARGADOS
*****
```

Podemos comprobar también que los dockers están funcionando correctamente:

```
hyperledger@localhost:~/work/src/fabric-samples/TFM2$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND
CREATED	STATUS	PORTS
NAMES		
a586c944e147	dev-fci.federaciones.netcan.com-veterinarios-1.0.0-	
130c44c62ff0205de3b79eea6e6c6fdb54e461931866f81fb4bf3ad7300101fd	"chaincode -peer.add..."	7 minutes ago Up 7 minutes
f3195d77e341	dev-fci.federaciones.netcan.com-veterinarios-1.0.0	
c0712aaa2c66f0eb4d3e68b9dca83fb740be117b8e8c9e54b50a94da72f3f366	"chaincode -peer.add..."	7 minutes ago Up 7 minutes
6faa4b9c88b3	dev-fci.federaciones.netcan.com-vacunas-1.0.0-	
de8c71563a9bf41548efc3efd696676c22d66525e1f312b37574d60c1b17aab0	"chaincode -peer.add..."	8 minutes ago Up 8 minutes
c25afd673a69	dev-fci.federaciones.netcan.com-solicitudes-1.0.0-	
b5e08411bd92c32ee9f31c54d803da0da9dbc07c35b86bdd9dbc66c7d9379d6c	"chaincode -peer.add..."	9 minutes ago Up 9 minutes
d028e389859a	dev-fci.federaciones.netcan.com-solicitudes-1.0.0	
ec9fd3c3016785411df3b0cfb73fb0c4a2ff8f249ada3ae8604d52579cf2ca92	"chaincode -peer.add..."	9 minutes ago Up 9 minutes
bf5699f9e24d	dev-fci.federaciones.netcan.com-personas-1.0.0-	
46b41e94180f7b7b9d3a25abd665310a9fe68ed5639e9ab474ddf30c60d278f3	"chaincode -peer.add..."	10 minutes ago Up 10 minutes
4590c88cacba	dev-fci.federaciones.netcan.com-perros-1.0.0-	
ee62113a5af70130e6a0df3310ec7555afcaee7f3659eb7a4191efd56bfa5d8	"chaincode -peer.add..."	10 minutes ago Up 10 minutes
55601d19cebb	dev-fci.federaciones.netcan.com-perfiles-1.0.0-	
fb61a6a31883fa2a1b3280a342942d5462730c2b4264e4e39e8f7cec9785092c	"chaincode -peer.add..."	11 minutes ago Up 11 minutes
988d7da3eb9b	dev-fci.federaciones.netcan.com-microchips-1.0.0	
33d55e901eadaa70caea32134c737a7fc4cae1ca66c9f8a937444c5383c8dbe	"chaincode -peer.add..."	11 minutes ago Up 11 minutes
f7d350837a53	hyperledger/fabric-tools:1.4.0	"/bin/bash" 13 minutes ago Up 13 minutes
98607d432073	hyperledger/fabric-peer:1.4.0	cli "peer node start" 13 minutes ago Up 13 minutes
cvextremadura.colegiosveterinarios.netcan.com	0.0.0.0:22051->7051/tcp, 0.0.0.0:22053->7053/tcp	
9002d51c0353	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
b3c24b94564f	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
874859966c08	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
9bd477f12e2e	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
1be18d94cee9	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
1c342f976a4a	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
de6e006ec6af	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
cvcastillayleon.colegiosveterinarios.netcan.com	0.0.0.0:18051->7051/tcp, 0.0.0.0:18053->7053/tcp	
0d33b90f342c	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
44f70682c94b	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
aef5b718652f	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
7f4edd208e32	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
32456cd98b79	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
99ca8c595e72	hyperledger/fabric-peer:1.4.0	"peer node start" 13 minutes ago Up 13 minutes
fci.federaciones.netcan.com	0.0.0.0:7051->7051/tcp, 0.0.0.0:7053->7053/tcp	

e345ef4c99e1	hyperledger/fabric-peer:1.4.0			"peer node start"	13
minutes ago	Up 13 minutes	0.0.0.0:19051->7051/tcp,	0.0.0.0:19053->7053/tcp		
cvcastillalamancha.colegiosveterinarios.netcan.com					
bf74300b19e0	hyperledger/fabric-peer:1.4.0			"peer node start"	13
minutes ago	Up 13 minutes	0.0.0.0:21051->7051/tcp,	0.0.0.0:21053->7053/tcp		
cvcomunitatvalenciana.colegiosveterinarios.netcan.com					
372e2dd9a544	hyperledger/fabric-peer:1.4.0			"peer node start"	13
minutes ago	Up 13 minutes	0.0.0.0:9051->7051/tcp, 0.0.0.0:9053->7053/tcp	tkc.federaciones.netcan.com		
8976910066ce	hyperledger/fabric-peer:1.4.0			"peer node start"	13
minutes ago	Up 13 minutes	0.0.0.0:11051->7051/tcp, 0.0.0.0:11053->7053/tcp	cvmadrid.colegiosveterinarios.netcan.com		
782af996da0b	hyperledger/fabric-peer:1.4.0			"peer node start"	13
minutes ago	Up 13 minutes	0.0.0.0:8051->7051/tcp, 0.0.0.0:8053->7053/tcp	rsce.federaciones.netcan.com		
962cefc6f9d3	hyperledger/fabric-peer:1.4.0			"peer node start"	13
minutes ago	Up 13 minutes	0.0.0.0:12051->7051/tcp, 0.0.0.0:12053->7053/tcp	cvandalucia.colegiosveterinarios.netcan.com		
88d2c08cbfdd	hyperledger/fabric-peer:1.4.0			"peer node start"	13
minutes ago	Up 13 minutes	0.0.0.0:13051->7051/tcp, 0.0.0.0:13053->7053/tcp	cvaragon.colegiosveterinarios.netcan.com		
ea6d64f28ccc	hyperledger/fabric-ca:1.4.0			"sh -c 'fabric-ca-se..."	13
minutes ago	Up 13 minutes	0.0.0.0:8054->7054/tcp	ca_colegiosveterinarios		
aad365324580	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:14984->5984/tcp	couchdb9		
a71ce0ce9753	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp	couchdb2		
7463ccf0857a	hyperledger/fabric-orderer:1.4.0			"orderer"	13
minutes ago	Up 13 minutes	0.0.0.0:7050->7050/tcp	orderer.netcan.com		
19ec5a46ae2b	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:22984->5984/tcp	couchdb17		
53c7c6a4cc9e	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:8984->5984/tcp	couchdb3		
8fbe12607240	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:25984->5984/tcp	couchdb20		
a7b3ef2ba8cc	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:23984->5984/tcp	couchdb18		
00c7f01324bc	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:20984->5984/tcp	couchdb15		
458978aac3ec	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:18984->5984/tcp	couchdb13		
77768de52068	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:12984->5984/tcp	couchdb7		
3b13146d8511	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:21984->5984/tcp	couchdb16		
9c203052f7be	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:17984->5984/tcp	couchdb12		
73324fd3a54e	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:16984->5984/tcp	couchdb11		
295b8c6d4dd3	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp	couchdb0		
c2731bd80cea	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:9984->5984/tcp	couchdb4		
3fb914af4804	hyperledger/fabric-ca:1.4.0			"sh -c 'fabric-ca-se..."	13
minutes ago	Up 13 minutes	0.0.0.0:7054->7054/tcp	ca_federaciones		
499058f050c7	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:11984->5984/tcp	couchdb6		
1e979f2fd299	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:19984->5984/tcp	couchdb14		
7d6f86c8b9cb	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:24984->5984/tcp	couchdb19		
7caf66b62cba	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:15984->5984/tcp	couchdb10		
1fca5ebfddc	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp	couchdb1		
764fbc1e1eb7	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:13984->5984/tcp	couchdb8		
cfaf325c3109	hyperledger/fabric-couchdb:0.4.15			"tini -- /docker-ent..."	13
minutes ago	Up 13 minutes	4369/tcp, 9100/tcp, 0.0.0.0:10984->5984/tcp	couchdb5		

Y podemos igualmente logarnos a la CA y crear nuevas afiliaciones:

```
docker exec -it ca_federaciones bash
cd etc/hyperledger/fabric-ca-server
fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
fabric-ca-client affiliation add federaciones
fabric-ca-client affiliation add federaciones.RSCE
```

```
hyperledger@localhost:~/work/src/fabric-samples/TFM2$ docker exec -it ca_federaciones bash
root@3fb914af4804:/# cd etc/hyperledger/fabric-ca-server
root@3fb914af4804:/etc/hyperledger/fabric-ca-server# fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
2020/09/12 16:33:43 [INFO] Created a default configuration file at /etc/hyperledger/fabric-ca-server/fabric-ca-client
-config.yaml
2020/09/12 16:33:43 [INFO] generating key: &{A:ecdsa S:256}
2020/09/12 16:33:43 [INFO] encoded CSR
2020/09/12 16:33:43 [INFO] Stored client certificate at /etc/hyperledger/fabric-ca-server/msp/signcerts/cert.pem
2020/09/12 16:33:43 [INFO] Stored root CA certificate at /etc/hyperledger/fabric-ca-server/msp/cacerts/localhost-7054
.pem
2020/09/12 16:33:43 [INFO] Stored Issuer public key at /etc/hyperledger/fabric-ca-server/msp/IssuerPublicKey
2020/09/12 16:33:43 [INFO] Stored Issuer revocation public key at /etc/hyperledger/fabric-ca-server/msp/IssuerRevocat
ionPublicKey
root@3fb914af4804:/etc/hyperledger/fabric-ca-server# fabric-ca-client affiliation add federaciones
2020/09/12 16:33:43 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yam
l
Successfully added affiliation: federaciones
root@3fb914af4804:/etc/hyperledger/fabric-ca-server# fabric-ca-client affiliation add federaciones.RSCE
2020/09/12 16:33:43 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yam
l
Successfully added affiliation: federaciones.RSCE
```

Crear un nuevo usuario:

```
fabric-ca-client register --id.name Unai --id.secret 99876 --id.affiliation federaciones.RSCE --
id.attrs 'hf.Revoker=true,admin=true:ecert'
fabric-ca-client enroll -u http://Unai:99876@localhost:7054 -M msp
fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
```

```
root@3fb914af4804:/etc/hyperledger/fabric-ca-server# fabric-ca-client register --id.name Unai --id.secret 99876 --id.
affiliation federaciones.RSCE --id.attrs 'hf.Revoker=true,admin=true:ecert'
2020/09/12 16:34:25 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yam
l
Password: 99876
root@3fb914af4804:/etc/hyperledger/fabric-ca-server# fabric-ca-client enroll -u http://Unai:99876@localhost:7054 -M m
sp
2020/09/12 16:34:37 [INFO] generating key: &{A:ecdsa S:256}
2020/09/12 16:34:37 [INFO] encoded CSR
2020/09/12 16:34:37 [INFO] Stored client certificate at /etc/hyperledger/fabric-ca-server/msp/signcerts/cert.pem
2020/09/12 16:34:37 [INFO] Stored root CA certificate at /etc/hyperledger/fabric-ca-server/msp/cacerts/localhost-7054
.pem
2020/09/12 16:34:37 [INFO] Stored Issuer public key at /etc/hyperledger/fabric-ca-server/msp/IssuerPublicKey
2020/09/12 16:34:37 [INFO] Stored Issuer revocation public key at /etc/hyperledger/fabric-ca-server/msp/IssuerRevocat
ionPublicKey
root@3fb914af4804:/etc/hyperledger/fabric-ca-server# fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
2020/09/12 16:34:43 [INFO] generating key: &{A:ecdsa S:256}
2020/09/12 16:34:43 [INFO] encoded CSR
2020/09/12 16:34:43 [INFO] Stored client certificate at /etc/hyperledger/fabric-ca-server/msp/signcerts/cert.pem
2020/09/12 16:34:43 [INFO] Stored root CA certificate at /etc/hyperledger/fabric-ca-server/msp/cacerts/localhost-7054
.pem
2020/09/12 16:34:43 [INFO] Stored Issuer public key at /etc/hyperledger/fabric-ca-server/msp/IssuerPublicKey
2020/09/12 16:34:43 [INFO] Stored Issuer revocation public key at /etc/hyperledger/fabric-ca-server/msp/IssuerRevocat
ionPublicKey
```

Y borrarlo:

```
fabric-ca-client identity remove Unai
```

```
root@3fb914af4804:/etc/hyperledger/fabric-ca-server# fabric-ca-client identity remove Unai
2020/09/12 16:35:42 [INFO] Configuration file location: /etc/hyperledger/fabric-ca-server/fabric-ca-client-config.yam
l
Successfully removed identity - Name: Unai, Type: client, Affiliation: federaciones.RSCE, Max Enrollments: -1, Attrib
utes: [{Name:hf.Type Value:client ECert:true} {Name:hf.Affiliation Value:federaciones.RSCE ECert:true} {Name:hf.Revok
er Value:true ECert:false} {Name:admin Value:true ECert:true} {Name:hf.EnrollmentID Value:Unai ECert:true}]
root@3fb914af4804:/etc/hyperledger/fabric-ca-server#
```



## RED CREADA ENTRE DOS SERVIDORES

Por último se implementa una red más parecida a la realidad en la que cada organización se despliega en un servidor distinto, utilizándose una red Docker Swarm para comunicar ambos servidores.

Los archivos pueden encontrarse en el repositorio Github en la carpeta Arquitectura bajo el nombre TFM\_dos\_Servidores.rar y los scripts en scripts3.

- Creación de la red Docker Swarm

En el primer servidor, que en nuestro caso será el servidor de Federaciones, en la carpeta del proyecto (que en nuestro caso es TFM), buscamos en primer lugar la interfaz de red de la IP pública del servidor:

Ifconfig

```
root@localhost:~# ifconfig
docker0  Link encap:Ethernet  HWaddr 02:42:3c:91:bb:c2
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:3c:ff:fe91:bbc2/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0  errors:0 dropped:0 overruns:0 frame:0
          TX packets:69  errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:6198 (6.1 KB)

docker_gwbridge Link encap:Ethernet  HWaddr 02:42:47:7a:68:06
            inet addr:192.168.160.1  Bcast:192.168.175.255  Mask:255.255.240.0
            inet6 addr: fe80::42:47:ff:fe7a:6806/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:16914  errors:0 dropped:0 overruns:0 frame:0
            TX packets:17903  errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:2089040 (2.0 MB)  TX bytes:1867934 (1.8 MB)

ens192    Link encap:Ethernet  HWaddr 00:50:56:3b:30:09
          inet addr:82.223.122.137  Bcast:82.223.122.137  Mask:255.255.255.255
          inet6 addr: fe80::250:56:ff:fe3b:3009/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:45509944  errors:0 dropped:0 overruns:0 frame:0
          TX packets:44016475  errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11207888057 (11.2 GB)  TX bytes:8919839812 (8.9 GB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
```

A continuación creamos una red Docker Swarm y le indicamos la interfaz de red de la IP pública

```
docker swarm init --advertise-addr ens192
```

```
root@localhost:/home/hyperledger/work/src/fabric-samples# docker swarm init --advertise-addr ens192
Swarm initialized: current node (3n2b1hs0uv9nosptto2b80qtm) is now a manager.
```

To add a worker to this swarm, run the following command:

```
docker swarm join --token SWMTKN-1-62wv8gfk5g030i9rgv2ytnqpg4mazxyvm3tyzu51q616jppsto-746cpa6lvaapaf9foj3rhj64f
82.223.122.137:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

Como en el segundo servidor queremos tener también un nodo manager utilizamos la siguiente instrucción:

```
docker swarm join-token manager
```

Que genera un token que pegaremos en la consola del segundo servidor, en el que estaremos en la carpeta del proyecto:

```
root@localhost:/home/hyperledger/work/src/fabric-samples# docker swarm join-token manager
To add a manager to this swarm, run the following command:
```

```
docker swarm join --token SWMTKN-1-62wv8gfk5g030i9rgv2ytnqpg4mazxyvm3tyzu51q616jppsto-418znypmntsmx9az6ivzjco04
82.223.122.137:2377
```

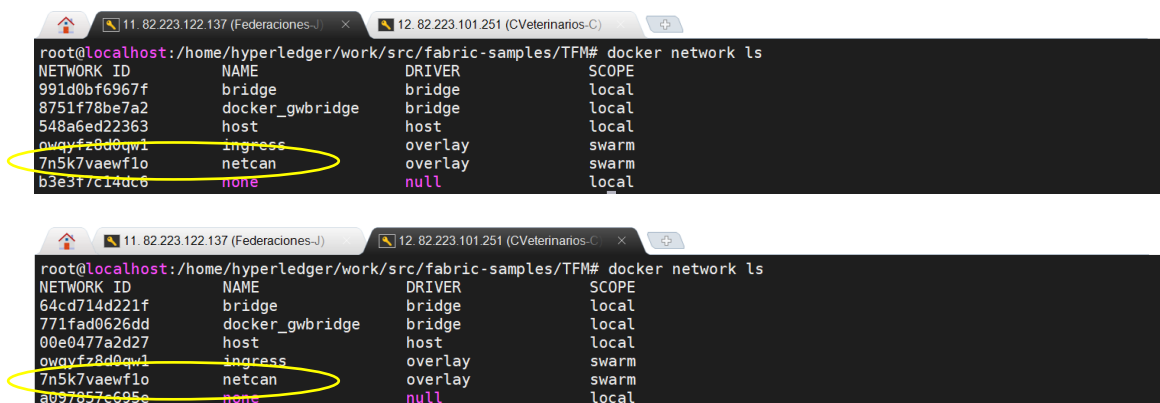
Una vez realizado esto regresamos al primer servidor en el que arrancaremos la red con el siguiente comando:

```
docker network create --attachable --driver overlay netcan
```

Pudiendo comprobar con

```
docker network ls
```

Que en ambos servidores aparece la misma red.



```
root@localhost:/home/hyperledger/work/src/fabric-samples/TFM# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
991d0bf6967f        bridge             bridge              local
8751f78be7a2        docker_gwbridge    bridge              local
548a6ed22363        host              host                local
0wqyfa2d0qw1        ingress            overlay             swarm
7n5k7vaewf1o        netcan             overlay             swarm
b3e3f7c14dc6        none              null                local

root@localhost:/home/hyperledger/work/src/fabric-samples/TFM# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
64cd714d221f        bridge             bridge              local
771fad0626dd        docker_gwbridge    bridge              local
00e0477a2d27        host              host                local
0wqyfa2d0qw1        ingress            overlay             swarm
7n5k7vaewf1o        netcan             overlay             swarm
a097857c695e        none              null                local
```



- Configuración de la red

En primer lugar, dado que queremos seguir manteniendo los scripts que levantan la red, necesitamos un método para poder conectar por medio de un script desde el primer servidor con la consola del segundo servidor y lanzar allí los comandos correspondientes, lo que realizamos configurando una clave público/privada con ssh-keygen:

Generamos el par de claves mediante el comando:

```
ssh-keygen -b 4096 -t rsa
```

```
root@localhost:/home/hyperledger/work/src/fabric-samples/TFM# ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:9zajio17DIKrCs2xsmOgNllgq9BS8xFcX8i4zA8WkEk root@localhost
The key's randomart image is:
+---[RSA 4096]-----+
| oE=.o .. |
| = o.o. |
| + o. |
|..o . * |
|+o+o o.oS . |
|=B.o.. ... |
|O.= . o = |
|+= o=. o o |
|o.... +++. |
+----[SHA256]-----+
```

Quedando la clave guardada en la carpeta .ssh del usuario en /home/hyperledger/.ssh/id\_rsa

A continuación es necesario pasar la clave pública al segundo servidor mediante el comando:

```
ssh-copy-id root@82.223.101.251
```

De esta manera cuando el primer servidor tenga que ejecutar cualquier comando en el segundo servidor podrá conectarse enviando la clave y no será necesario introducir contraseñas de manera manual.

Una vez configurada la conexión entre ambos servidores procedemos a modificar los archivos de configuración de la red `docker-compose-cli.yaml`, `docker-compose-couch.yaml` y `docker-compose-base.yaml`, partiendo de la red generada con TLS.

En el archivo `docker-compose-cli.yaml` es necesario quitar para el primer servidor todo lo relativo a los peers de la organización `ColegiosVeterinarios`, quedando la CA de federaciones, el orderer, el cli y los peers de Federaciones.

Es importante también cambiar la definición de la red que deberá definirse como externa y con el nombre de la red Swarm, aunque en la definición de cada servicio se utilizará el nombre de la red definida como externa y no el de la red Swarm:

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2'

volumes:
  orderer.netcan.com:
  FCI.federaciones.netcan.com:
  RSCE.federaciones.netcan.com:
  TKC.federaciones.netcan.com:
  ACW.federaciones.netcan.com:

networks:
  default:
    external:
      name: netcan

services:
  ca.federaciones:
    image: hyperledger/fabric-ca:1.4.0
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-federaciones
      - FABRIC_CA_SERVER_TLS_ENABLED=false
      - FABRIC_CA_SERVER_PORT=7054
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-
config/ca.federaciones.netcan.com-cert.pem --ca.keyfile /etc/hyperledger/fabric-ca-server-
config/876ab47bd6ac9132e57aa5e92828193fc70dc51522c932666e5e757a31b65de0_sk -b admin:adminpw -d --
cfg.identities.allowremove'
    volumes:
      - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
      - /home/hyperledger/work/src/fabric-samples/TFM/crypto-
config/peerOrganizations/federaciones.netcan.com/ca/:/etc/hyperledger/fabric-ca-server-config
    container_name: ca_federaciones
    networks:
      - default

  orderer.netcan.com:
    extends:
      file: base/docker-compose-base.yaml
      service: orderer.netcan.com
    container_name: orderer.netcan.com
    networks:
      - default

  FCI.federaciones.netcan.com:
    container_name: FCI.federaciones.netcan.com
    extends:
```

```
file: base/docker-compose-base.yaml
service: FCI.federaciones.netcan.com
networks:
  - default

RSCE.federaciones.netcan.com:
  container_name: RSCE.federaciones.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: RSCE.federaciones.netcan.com
  networks:
    - default

TKC.federaciones.netcan.com:
  container_name: TKC.federaciones.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: TKC.federaciones.netcan.com
  networks:
    - default

ACW.federaciones.netcan.com:
  container_name: ACW.federaciones.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: ACW.federaciones.netcan.com
  networks:
    - default

cli:
  container_name: cli
  image: hyperledger/fabric-tools:1.4.0
  tty: true
  stdin_open: true
  environment:
    - GOPATH=/opt/gopath
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - FABRIC_LOGGING_SPEC=INFO
    - CORE_PEER_ID=cli
    - CORE_PEER_ADDRESS=FCI.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
    - CORE_PEER_TLS_ENABLED=true
    - CORE_PEER_TLS_CERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/peers/FCI.federaciones.netcan.com/tls/server.crt
    - CORE_PEER_TLS_KEY_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/peers/FCI.federaciones.netcan.com/tls/server.key
    - CORE_PEER_TLS_ROOTCERT_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/peers/FCI.federaciones.netcan.com/tls/ca.crt
    - CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/federaciones.netcan.com/users/Admin@federaciones.netcan.com/msp
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
  command: /bin/bash
  volumes:
    - /var/run:/host/var/run/
    - /home/hyperledger/work/src/fabric-samples/TFM/chaincode:/opt/gopath/src/github.com/chaincode
    - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
    - ./scripts:/opt/gopath/src/github.com/hyperledger/fabric/peer/scripts/
    - ./channel-artifacts:/opt/gopath/src/github.com/hyperledger/fabric/peer/channel-artifacts
  depends_on:
    - orderer.netcan.com
    - FCI.federaciones.netcan.com
    - RSCE.federaciones.netcan.com
    - TKC.federaciones.netcan.com
    - ACW.federaciones.netcan.com
  networks:
    - default
```

En el segundo servidor únicamente quedará lo relativo a los peers y la CA de la organización ColegiosVeterinarios:

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2'

volumes:
  CVAndalucia.colegiosveterinarios.netcan.com:
  CVAragon.colegiosveterinarios.netcan.com:
  CVAsturias.colegiosveterinarios.netcan.com:
  CVIllesBalears.colegiosveterinarios.netcan.com:
  CVCanarias.colegiosveterinarios.netcan.com:
  CVCantabria.colegiosveterinarios.netcan.com:
  CVCastillayLeon.colegiosveterinarios.netcan.com:
  CVCastillalaMancha.colegiosveterinarios.netcan.com:
  CVCataluna.colegiosveterinarios.netcan.com:
  CVComunitatValenciana.colegiosveterinarios.netcan.com:
  CVExtremadura.colegiosveterinarios.netcan.com:
  CVGalicia.colegiosveterinarios.netcan.com:
  CVMadrid.colegiosveterinarios.netcan.com:
  CVMurcia.colegiosveterinarios.netcan.com:
  CVNavarra.colegiosveterinarios.netcan.com:
  CVPaisVasco.colegiosveterinarios.netcan.com:
  CVLaRioja.colegiosveterinarios.netcan.com:

networks:
  default:
    external:
      name: netcan

services:
  ca.colegiosveterinarios:
    image: hyperledger/fabric-ca:1.4.0
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-colegiosveterinarios
      - FABRIC_CA_SERVER_TLS_ENABLED=false
      - FABRIC_CA_SERVER_PORT=7054
    ports:
      - "8054:7054"
    command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-
config/ca.colegiosveterinarios.netcan.com-cert.pem --ca.keyfile /etc/hyperledger/fabric-ca-server-
config/1f0010447361e790c31d0e0c0fe5b6238804142c0ec886bc2e1c8cd5a18c3336_sk -b admin-cv:admin-cvpw -d --
cfg.identities.allowremove'
    volumes:
      - ./crypto-config/peerOrganizations/colegiosveterinarios.netcan.com/ca:/etc/hyperledger/fabric-ca-server-config
      - ./crypto-config/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
    container_name: ca_colegiosveterinarios
    networks:
      - default

  CVAndalucia.colegiosveterinarios.netcan.com:
    container_name: CVAndalucia.colegiosveterinarios.netcan.com
    extends:
      file: base/docker-compose-base.yaml
      service: CVAndalucia.colegiosveterinarios.netcan.com
    networks:
      - default

  CVAragon.colegiosveterinarios.netcan.com:
    container_name: CVAragon.colegiosveterinarios.netcan.com
    extends:
      file: base/docker-compose-base.yaml
      service: CVAragon.colegiosveterinarios.netcan.com
    networks:
      - default

  CVAsturias.colegiosveterinarios.netcan.com:
    container_name: CVAsturias.colegiosveterinarios.netcan.com
    extends:
      file: base/docker-compose-base.yaml
      service: CVAsturias.colegiosveterinarios.netcan.com
```

```
networks:
  - default

CVIllesBalears.colegiosveterinarios.netcan.com:
  container_name: CVIllesBalears.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVIllesBalears.colegiosveterinarios.netcan.com
  networks:
    - default

CVCanarias.colegiosveterinarios.netcan.com:
  container_name: CVCanarias.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVCanarias.colegiosveterinarios.netcan.com
  networks:
    - default

CVCantabria.colegiosveterinarios.netcan.com:
  container_name: CVCantabria.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVCantabria.colegiosveterinarios.netcan.com
  networks:
    - default

CVCastillayLeon.colegiosveterinarios.netcan.com:
  container_name: CVCastillayLeon.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVCastillayLeon.colegiosveterinarios.netcan.com
  networks:
    - default

CVCastillalaMancha.colegiosveterinarios.netcan.com:
  container_name: CVCastillalaMancha.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVCastillalaMancha.colegiosveterinarios.netcan.com
  networks:
    - default

CVCataluna.colegiosveterinarios.netcan.com:
  container_name: CVCataluna.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVCataluna.colegiosveterinarios.netcan.com
  networks:
    - default

CVComunitatValenciana.colegiosveterinarios.netcan.com:
  container_name: CVComunitatValenciana.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVComunitatValenciana.colegiosveterinarios.netcan.com
  networks:
    - default

CVExtremadura.colegiosveterinarios.netcan.com:
  container_name: CVExtremadura.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVExtremadura.colegiosveterinarios.netcan.com
  networks:
    - default

CVGalicia.colegiosveterinarios.netcan.com:
  container_name: CVGalicia.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVGalicia.colegiosveterinarios.netcan.com
  networks:
    - default

CVMadrid.colegiosveterinarios.netcan.com:
  container_name: CVMadrid.colegiosveterinarios.netcan.com
  extends:
```

```
file: base/docker-compose-base.yaml
service: CVMadrid.colegiosveterinarios.netcan.com
networks:
  - default

CVMurcia.colegiosveterinarios.netcan.com:
  container_name: CVMurcia.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVMurcia.colegiosveterinarios.netcan.com
  networks:
    - default

CVNavarra.colegiosveterinarios.netcan.com:
  container_name: CVNavarra.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVNavarra.colegiosveterinarios.netcan.com
  networks:
    - default

CVPaisVasco.colegiosveterinarios.netcan.com:
  container_name: CVPaisVasco.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVPaisVasco.colegiosveterinarios.netcan.com
  networks:
    - default

CVLaRioja.colegiosveterinarios.netcan.com:
  container_name: CVLaRioja.colegiosveterinarios.netcan.com
  extends:
    file: base/docker-compose-base.yaml
    service: CVLaRioja.colegiosveterinarios.netcan.com
  networks:
    - default
```

En el archivo docker-compose-couch.yaml en el primer servidor se dejará únicamente lo relativo a los peers de la organización Federaciones y se deberá tener cuidado en definir la red igual que en el anterior archivo:

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#
version: '2'
networks:
  default:
    external:
      name: netcan
services:
  couchdb0:
    container_name: couchdb0
    image: hyperledger/fabric-couchdb:0.4.15
    environment:
      - COUCHDB_USER=
      - COUCHDB_PASSWORD=
    ports:
      - "5984:5984"
    networks:
      - default
  FCI.federaciones.netcan.com:
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb0:5984
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb0
```

```
couchdb1:
  container_name: couchdb1
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "6984:5984"
  networks:
    - default

RSCE.federaciones.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb1:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb1

couchdb2:
  container_name: couchdb2
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "7984:5984"
  networks:
    - default

TKC.federaciones.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb2:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb2

couchdb3:
  container_name: couchdb3
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "8984:5984"
  networks:
    - default

ACW.federaciones.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb3:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb3
```

Y en el segundo servidor únicamente se dejará lo relativo a los peers de la organización ColegiosVeterinarios:

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2'

networks:
  default:
    external:
      name: netcan
```

```
services:
  couchdb4:
    container_name: couchdb4
    image: hyperledger/fabric-couchdb:0.4.15
    environment:
      - COUCHDB_USER=
      - COUCHDB_PASSWORD=
    ports:
      - "9984:5984"
    networks:
      - default

  CVMadrid.colegiosveterinarios.netcan.com:
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb4:5984
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb4

  couchdb5:
    container_name: couchdb5
    image: hyperledger/fabric-couchdb:0.4.15
    environment:
      - COUCHDB_USER=
      - COUCHDB_PASSWORD=
    ports:
      - "10984:5984"
    networks:
      - default

  CVAndalucia.colegiosveterinarios.netcan.com:
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb5:5984
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb5

  couchdb6:
    container_name: couchdb6
    image: hyperledger/fabric-couchdb:0.4.15
    environment:
      - COUCHDB_USER=
      - COUCHDB_PASSWORD=
    ports:
      - "11984:5984"
    networks:
      - default

  CVAragon.colegiosveterinarios.netcan.com:
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb6:5984
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
    depends_on:
      - couchdb6

  couchdb7:
    container_name: couchdb7
    image: hyperledger/fabric-couchdb:0.4.15
    environment:
      - COUCHDB_USER=
      - COUCHDB_PASSWORD=
    ports:
      - "12984:5984"
    networks:
      - default

  CVAsturias.colegiosveterinarios.netcan.com:
    environment:
      - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
      - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb7:5984
      - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
      - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
```



```
depends_on:
  - couchdb7

couchdb8:
  container_name: couchdb8
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "13984:5984"
  networks:
    - default

CVIllaBalears.colegiosveterinarios.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb8:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb8

couchdb9:
  container_name: couchdb9
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "14984:5984"
  networks:
    - default

CVCanarias.colegiosveterinarios.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb9:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb9

couchdb10:
  container_name: couchdb10
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "15984:5984"
  networks:
    - default

CVCantabria.colegiosveterinarios.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb10:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb10

couchdb11:
  container_name: couchdb11
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "16984:5984"
  networks:
    - default

CVCastillaLeon.colegiosveterinarios.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb11:5984
```

```
- CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=  
- CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=  
depends_on:  
- couchdb11  
  
couchdb12:  
  container_name: couchdb12  
  image: hyperledger/fabric-couchdb:0.4.15  
  environment:  
    - COUCHDB_USER=  
    - COUCHDB_PASSWORD=  
  ports:  
    - "17984:5984"  
  networks:  
    - default  
  
CVCastillalaMancha.colegiosveterinarios.netcan.com:  
  environment:  
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb12:5984  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=  
  depends_on:  
    - couchdb12  
  
couchdb13:  
  container_name: couchdb13  
  image: hyperledger/fabric-couchdb:0.4.15  
  environment:  
    - COUCHDB_USER=  
    - COUCHDB_PASSWORD=  
  ports:  
    - "18984:5984"  
  networks:  
    - default  
  
CVCataluna.colegiosveterinarios.netcan.com:  
  environment:  
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb13:5984  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=  
  depends_on:  
    - couchdb13  
  
couchdb14:  
  container_name: couchdb14  
  image: hyperledger/fabric-couchdb:0.4.15  
  environment:  
    - COUCHDB_USER=  
    - COUCHDB_PASSWORD=  
  ports:  
    - "19984:5984"  
  networks:  
    - default  
  
CVComunitatValenciana.colegiosveterinarios.netcan.com:  
  environment:  
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb14:5984  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=  
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=  
  depends_on:  
    - couchdb14  
  
couchdb15:  
  container_name: couchdb15  
  image: hyperledger/fabric-couchdb:0.4.15  
  environment:  
    - COUCHDB_USER=  
    - COUCHDB_PASSWORD=  
  ports:  
    - "20984:5984"  
  networks:  
    - default  
  
CVExtremadura.colegiosveterinarios.netcan.com:  
  environment:
```

```
- CORE_LEDGER_STATE_STATEDATABASE=CouchDB
- CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb15:5984
- CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
- CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
depends_on:
- couchdb15

couchdb16:
  container_name: couchdb16
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "21984:5984"
  networks:
    - default

CVGalicia.colegiosveterinarios.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb16:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb16

couchdb17:
  container_name: couchdb17
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "22984:5984"
  networks:
    - default

CVMurcia.colegiosveterinarios.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb17:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb17

couchdb18:
  container_name: couchdb18
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "23984:5984"
  networks:
    - default

CVNavarra.colegiosveterinarios.netcan.com:
  environment:
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb18:5984
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  depends_on:
    - couchdb18

couchdb19:
  container_name: couchdb19
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "24984:5984"
  networks:
    - default
```

```
CVPaisVasco.colegiosveterinarios.netcan.com:
environment:
  - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
  - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb19:5984
  - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
  - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
depends_on:
  - couchdb19

couchdb20:
  container_name: couchdb20
  image: hyperledger/fabric-couchdb:0.4.15
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - "25984:5984"
  networks:
    - default

CVLaRioja.colegiosveterinarios.netcan.com:
environment:
  - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
  - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb20:5984
  - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
  - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
depends_on:
  - couchdb20
```

En el archivo docker-compose-base.yaml, en el primer servidor se deja lo relativo al orderer y los peers de la organización Federaciones:

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#
version: '2'

services:

  orderer.netcan.com:
    container_name: orderer.netcan.com
    image: hyperledger/fabric-orderer:1.4.0
    environment:
      - ORDERER_GENERAL_LOGLEVEL=INFO
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
      - ORDERER_GENERAL_GENESISMETHOD=file
      - ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block
      - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
      # enabled TLS
      - ORDERER_GENERAL_TLS_ENABLED=true
      - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
      - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
      - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=tfm_netcan
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric
    command: orderer
    volumes:
      - ../channel-artifacts/genesis.block:/var/hyperledger/orderer/orderer.genesis.block
      - ../crypto-config/ordererOrganizations/netcan.com/orderers/orderer.netcan.com/msp:/var/hyperledger/orderer/msp
      - ../crypto-config/ordererOrganizations/netcan.com/orderers/orderer.netcan.com/tls:/var/hyperledger/orderer/tls
      - orderer.netcan.com:/var/hyperledger/production/orderer
    ports:
      - 7050:7050

  FCI.federaciones.netcan.com:
    container_name: FCI.federaciones.netcan.com
    extends:
      file: peer-base.yaml
      service: peer-base
    environment:
      - CORE_PEER_ID=FCI.federaciones.netcan.com
      - CORE_PEER_ADDRESS=FCI.federaciones.netcan.com:7051
      - CORE_PEER_GOSSIP_BOOTSTRAP=RSCE.federaciones.netcan.com:7051
```

```
- CORE_PEER_GOSSIP_EXTERNALENDPOINT=FCI.federaciones.netcan.com:7051
- CORE_PEER_LOCALMSPID=FederacionesMSP
volumes:
  - /var/run:/host/var/run/
  - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/FCI.federaciones.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/FCI.federaciones.netcan.com/tls:/etc/hyperledger/fabric/tls
  - FCI.federaciones.netcan.com:/var/hyperledger/production
ports:
  - 7051:7051
  - 7053:7053

RSCE.federaciones.netcan.com:
  container_name: RSCE.federaciones.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=RSCE.federaciones.netcan.com
    - CORE_PEER_ADDRESS=RSCE.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=RSCE.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=FCI.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/RSCE.federaciones.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/RSCE.federaciones.netcan.com/tls:/etc/hyperledger/fabric/tls
  - RSCE.federaciones.netcan.com:/var/hyperledger/production
ports:
  - 8051:7051
  - 8053:7053

TKC.federaciones.netcan.com:
  container_name: TKC.federaciones.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=TKC.federaciones.netcan.com
    - CORE_PEER_ADDRESS=TKC.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=TKC.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=FCI.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/TKC.federaciones.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/TKC.federaciones.netcan.com/tls:/etc/hyperledger/fabric/tls
  - TKC.federaciones.netcan.com:/var/hyperledger/production
ports:
  - 9051:7051
  - 9053:7053

ACW.federaciones.netcan.com:
  container_name: ACW.federaciones.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=ACW.federaciones.netcan.com
    - CORE_PEER_ADDRESS=ACW.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=ACW.federaciones.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=FCI.federaciones.netcan.com:7051
    - CORE_PEER_LOCALMSPID=FederacionesMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/ACW.federaciones.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/federaciones.netcan.com/peers/ACW.federaciones.netcan.com/tls:/etc/hyperledger/fabric/tls
  - ACW.federaciones.netcan.com:/var/hyperledger/production
ports:
  - 10051:7051
  - 10053:7053
```

Y en el segundo servidor se deja lo relativo a los peers de la organización ColegiosVeterinarios:

```
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

version: '2'

services:

  CVMadrid.colegiosveterinarios.netcan.com:
    container_name: CVMadrid.colegiosveterinario.netcan.com
    extends:
      file: peer-base.yaml
      service: peer-base
    environment:
      - CORE_PEER_ID=CVMadrid.colegiosveterinarios.netcan.com
      - CORE_PEER_ADDRESS=CVMadrid.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVMadrid.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_GOSSIP_BOOTSTRAP=CVAndalucia.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
    volumes:
      - /var/run:/host/var/run/
      - ../crypto-
    config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVMadrid.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
      - ../crypto-
    config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVMadrid.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
      - CVMadrid.colegiosveterinarios.netcan.com:/var/hyperledger/production
    ports:
      - 11051:7051
      - 11053:7053

  CVAndalucia.colegiosveterinarios.netcan.com:
    container_name: CVAndalucia.colegiosveterinarios.netcan.com
    extends:
      file: peer-base.yaml
      service: peer-base
    environment:
      - CORE_PEER_ID=CVAndalucia.colegiosveterinarios.netcan.com
      - CORE_PEER_ADDRESS=CVAndalucia.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVAndalucia.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
    volumes:
      - /var/run:/host/var/run/
      - ../crypto-
    config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVAndalucia.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
      - ../crypto-
    config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVAndalucia.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
      - CVAndalucia.colegiosveterinarios.netcan.com:/var/hyperledger/production
    ports:
      - 12051:7051
      - 12053:7053

  CVAragon.colegiosveterinarios.netcan.com:
    container_name: CVAragon.colegiosveterinarios.netcan.com
    extends:
      file: peer-base.yaml
      service: peer-base
    environment:
      - CORE_PEER_ID=CVAragon.colegiosveterinarios.netcan.com
      - CORE_PEER_ADDRESS=CVAragon.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVAragon.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
      - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
    volumes:
      - /var/run:/host/var/run/
```

```
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVAragon.colegiosveterinarios.netcan.com/msp:/etc/hyper
ledger/fabric/msp
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVAragon.colegiosveterinarios.netcan.com/tls:/etc/hyper
ledger/fabric/tls
- CVAragon.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
- 13051:7051
- 13053:7053

CVAsturias.colegiosveterinarios.netcan.com:
container_name: CVAsturias.colegiosveterinarios.netcan.com
extends:
file: peer-base.yaml
service: peer-base
environment:
- CORE_PEER_ID=CVAsturias.colegiosveterinarios.netcan.com
- CORE_PEER_ADDRESS=CVAsturias.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVAsturias.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
- CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
- /var/run:/host/var/run/
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVAsturias.colegiosveterinarios.netcan.com/msp:/etc/hyp
erledger/fabric/msp
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVAsturias.colegiosveterinarios.netcan.com/tls:/etc/hyp
erledger/fabric/tls
- CVAsturias.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
- 14051:7051
- 14053:7053

CVIllesBalears.colegiosveterinarios.netcan.com:
container_name: CVIllesBalears.colegiosveterinarios.netcan.com
extends:
file: peer-base.yaml
service: peer-base
environment:
- CORE_PEER_ID=CVIllesBalears.colegiosveterinarios.netcan.com
- CORE_PEER_ADDRESS=CVIllesBalears.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVIllesBalears.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
- CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
- /var/run:/host/var/run/
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVIllesBalears.colegiosveterinarios.netcan.com/msp:/etc
/hyperledger/fabric/msp
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVIllesBalears.colegiosveterinarios.netcan.com/tls:/etc
/hyperledger/fabric/tls
- CVIllesBalears.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
- 15051:7051
- 15053:7053

CVCanarias.colegiosveterinarios.netcan.com:
container_name: CVCanarias.colegiosveterinarios.netcan.com
extends:
file: peer-base.yaml
service: peer-base
environment:
- CORE_PEER_ID=CVCanarias.colegiosveterinarios.netcan.com
- CORE_PEER_ADDRESS=CVCanarias.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVCanarias.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
- CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
- /var/run:/host/var/run/
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCanarias.colegiosveterinarios.netcan.com/msp:/etc/hyp
erledger/fabric/msp
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCanarias.colegiosveterinarios.netcan.com/tls:/etc/hyp
erledger/fabric/tls
```

```
- CVCanarias.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 16051:7051
  - 16053:7053

CVCantabria.colegiosveterinarios.netcan.com:
  container_name: CVCantabria.colegiosveterinarios.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=CVCantabria.colegiosveterinarios.netcan.com
    - CORE_PEER_ADDRESS=CVCantabria.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVCantabria.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCantabria.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCantabria.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVCantabria.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 17051:7051
  - 17053:7053

CVCastillayLeon.colegiosveterinarios.netcan.com:
  container_name: CVCastillayLeon.colegiosveterinarios.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=CVCastillayLeon.colegiosveterinarios.netcan.com
    - CORE_PEER_ADDRESS=CVCastillayLeon.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVCastillayLeon.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCastillayLeon.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCastillayLeon.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVCastillayLeon.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 18051:7051
  - 18053:7053

CVCastillalaMancha.colegiosveterinarios.netcan.com:
  container_name: CVCastillalaMancha.colegiosveterinarios.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=CVCastillalaMancha.colegiosveterinarios.netcan.com
    - CORE_PEER_ADDRESS=CVCastillalaMancha.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVCastillalaMancha.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCastillalaMancha.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCastillalaMancha.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVCastillalaMancha.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 19051:7051
  - 19053:7053

CVCataluna.colegiosveterinarios.netcan.com:
```



```
container_name: CVCataluna.colegiosveterinarios.netcan.com
extends:
  file: peer-base.yaml
  service: peer-base
environment:
  - CORE_PEER_ID=CVCataluna.colegiosveterinarios.netcan.com
  - CORE_PEER_ADDRESS=CVCataluna.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVCataluna.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
  - /var/run:/host/var/run/
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCataluna.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVCataluna.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVCataluna.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 20051:7051
  - 20053:7053

CVComunitatValenciana.colegiosveterinarios.netcan.com:
container_name: CVComunitatValenciana.colegiosveterinarios.netcan.com
extends:
  file: peer-base.yaml
  service: peer-base
environment:
  - CORE_PEER_ID=CVComunitatValenciana.colegiosveterinarios.netcan.com
  - CORE_PEER_ADDRESS=CVComunitatValenciana.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVComunitatValenciana.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
  - /var/run:/host/var/run/
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVComunitatValenciana.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVComunitatValenciana.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVComunitatValenciana.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 21051:7051
  - 21053:7053

CVExtremadura.colegiosveterinarios.netcan.com:
container_name: CVExtremadura.colegiosveterinarios.netcan.com
extends:
  file: peer-base.yaml
  service: peer-base
environment:
  - CORE_PEER_ID=CVExtremadura.colegiosveterinarios.netcan.com
  - CORE_PEER_ADDRESS=CVExtremadura.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVExtremadura.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
  - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
  - /var/run:/host/var/run/
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVExtremadura.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVExtremadura.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVExtremadura.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 22051:7051
  - 22053:7053

CVGalicia.colegiosveterinarios.netcan.com:
container_name: CVGalicia.colegiosveterinarios.netcan.com
extends:
  file: peer-base.yaml
  service: peer-base
environment:
  - CORE_PEER_ID=CVGalicia.colegiosveterinarios.netcan.com
```

```
- CORE_PEER_ADDRESS=CVGalicia.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVGalicia.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
- CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
  - /var/run:/host/var/run/
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVGalicia.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVGalicia.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVGalicia.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 23051:7051
  - 23053:7053

CVMurcia.colegiosveterinarios.netcan.com:
  container_name: CVMurcia.colegiosveterinarios.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=CVMurcia.colegiosveterinarios.netcan.com
    - CORE_PEER_ADDRESS=CVMurcia.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVMurcia.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVMurcia.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVMurcia.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVMurcia.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 24051:7051
  - 24053:7053

CVNavarra.colegiosveterinarios.netcan.com:
  container_name: CVNavarra.colegiosveterinarios.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=CVNavarra.colegiosveterinarios.netcan.com
    - CORE_PEER_ADDRESS=CVNavarra.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVNavarra.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
  volumes:
    - /var/run:/host/var/run/
    - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVNavarra.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
  - ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVNavarra.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
  - CVNavarra.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
  - 25051:7051
  - 25053:7053

CVPaisVasco.colegiosveterinarios.netcan.com:
  container_name: CVPaisVasco.colegiosveterinarios.netcan.com
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=CVPaisVasco.colegiosveterinarios.netcan.com
    - CORE_PEER_ADDRESS=CVPaisVasco.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVPaisVasco.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
    - CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
  volumes:
    - /var/run:/host/var/run/
```

```
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVPaisVasco.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVPaisVasco.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
- CVPaisVasco.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
- 26051:7051
- 26053:7053

CVLaRioja.colegiosveterinarios.netcan.com:
container_name: CVLaRioja.colegiosveterinarios.netcan.com
extends:
file: peer-base.yaml
service: peer-base
environment:
- CORE_PEER_ID=CVLaRioja.colegiosveterinarios.netcan.com
- CORE_PEER_ADDRESS=CVLaRioja.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_EXTERNALENDPOINT=CVLaRioja.colegiosveterinarios.netcan.com:7051
- CORE_PEER_GOSSIP_BOOTSTRAP=CVMadrid.colegiosveterinarios.netcan.com:7051
- CORE_PEER_LOCALMSPID=ColegiosVeterinariosMSP
volumes:
- /var/run:/host/var/run/
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVLaRioja.colegiosveterinarios.netcan.com/msp:/etc/hyperledger/fabric/msp
- ../crypto-
config/peerOrganizations/colegiosveterinarios.netcan.com/peers/CVLaRioja.colegiosveterinarios.netcan.com/tls:/etc/hyperledger/fabric/tls
- CVLaRioja.colegiosveterinarios.netcan.com:/var/hyperledger/production
ports:
- 27051:7051
- 27053:7053
```

El archivo peer-base.yaml será idéntico para los dos servidores y no varía con respecto a la red desplegada en un servidor, al igual que el resto de archivos que hay que copiar al segundo servidor: material criptográfico y artefactos creados.

En cuanto a los scripts la única diferencia con la red anterior es que se copiarán desde la carpeta scripts3 del Github y que se utilizará el script Serv2\_script.sh que en las redes anteriores no se había utilizado.

Al lanzar el script de inicio (netcan\_script.sh) la red se levantará de la misma manera aunque primero se levantará la red Docker Swarm, luego lo harán los peers de la organización Federaciones:



Para apagar la red utilizaremos el script `stop_netcan_script.sh` que limpia los dockers y los chaincodes desplegados.

