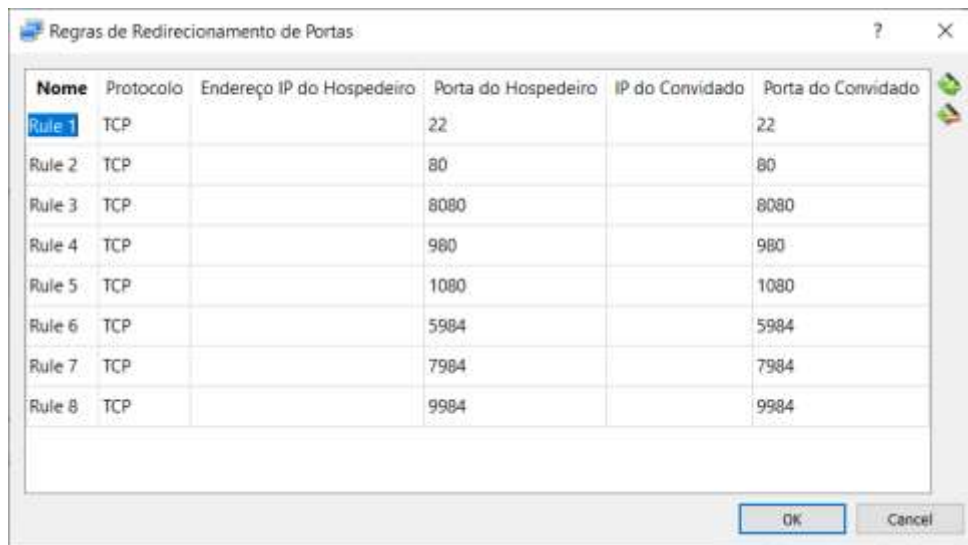


## Módulo 4 – Trabalhando o chaincode FabCar e primeiros passos com a biblioteca CC-Tools

### Configuração da VM

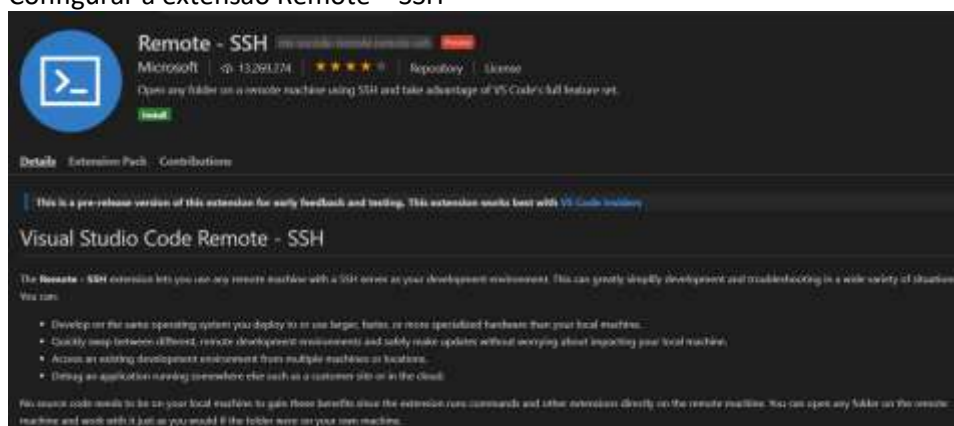
Configurar o redirecionamento de portas do Virtual Box.



Baixar o VSCode.

<https://code.visualstudio.com/>

Configurar a extensão Remote – SSH



Configurar um novo host no VSCode. Apertar CTRL+SHIFT+P e digitar *Remote SSH: Add New SSH Host*. Incluir o host digitando (Virtual Box deve estar ligado)

`ssh student@localhost`

Nessa aula vamos manter o ambiente da tarefa da aula 4 com as imagens Hyperledger Fabric 2.2.

Limpar os containers antigos.

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

# TRILHA BLOCKCHAIN HYPERLEDGER FABRIC

Dentro do diretório ***fabric-samples/test-network*** Subir a rede com o chaincode ***fabcar***

```
./network.sh up -ca
./network.sh createChannel
./network.sh deployCC -ccn fabcar -ccp ../chaincode/fabcar/go -ccl go
```

## Tarefa 1

Ir para o diretório ***/home/student/fabric-samples/test-network***

Configurar as variáveis de ambiente do peer

```
export FABRIC_CFG_PATH=$PWD/../config/
export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org1MSP"
export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
export CORE_PEER_ADDRESS=localhost:7051
```

Chamar a função para criar o carro

```
peer chaincode invoke \
-o localhost:7050 \
--ordererTLSHostnameOverride orderer.example.com \
--tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" \
-C mychannel -n fabcar \
--peerAddresses localhost:7051 \
--tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" \
--peerAddresses localhost:9051 --tlsRootCertFiles
"${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" \
-c '{"function": "CreateCar", "Args": ["CAR11", "Volks", "Passat", "preto", "Carlos"]}'
```

Criar os carros:

Ford Belina azul marinho, proprietário Luiz

Ford Variant cinza, proprietária Maria

Gravar os resultados das operações

## Tarefa 2

Ler a informação de 2 carros quaisquer.

```
peer chaincode query -C mychannel -n fabcar -c '{"Args": ["QueryCar", "CAR2"]}'
```

Gravar a operação.

## Tarefa 3

# TRILHA BLOCKCHAIN HYPERLEDGER FABRIC

Mudar o proprietário do carro CAR1 para “Marcelo”

Chamar a operação de QueryCar para mostrar o novo ativo.

## Tarefa 4

Utilizar o VSCode para atualizar o chaincode. Digitar CTRL+SHIFT+P e digitar *Remote-SSH:Connect to Host* e conectar ao *localhost*.

Digitar a senha 123456, na barra da esquerda, escolher Abrir o Folder e digitar */home/student/fabric-samples/chaincode/fabcar/go*

Criar um novo metodo para mudar a cor do carro.

Atualizar o chaincode usando o comando.

```
./network.sh deployCC -ccn fabcar -ccp ../chaincode/fabcar/go/ -ccl go -ccv 2.0 -ccs 2
```

Gravar o resultado da atualização do chaincode.

Atualizar a cor de um carro.

Gravar a operação de query.

## Tarefa 5

Apagar os containers antigos.

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

Subir uma rede usando o repositório **cc-tools-demo**

```
git clone https://github.com/goledgerdev/cc-tools-demo.git
```

Vendorar o chaincode

```
cd cc-tools-demo/chaincode  
go mod vendor  
cd ..
```

Vendorar o web service

```
cd rest-server  
docker network create cc-tools-demo-net  
./npmInstall.sh  
cd ..
```

Criar uma rede Hyperledger Fabric 2

```
./startDev2.sh
```

Gravar o resultado da criação da rede.

## Tarefa 6

Abrir o swagger acessando

<http://localhost/api-docs>

Chamar os endpoints *getTx* e *getSchema* (Get)

Gravar o resultado das 2 operações.

## Tarefa 7

Abrir o swagger acessando no browser.

<http://localhost/api-docs>

Chamar os endpoints *getTx* e *getSchema* (Get)

Gravar o resultado das 2 operações.

## Tarefa 8

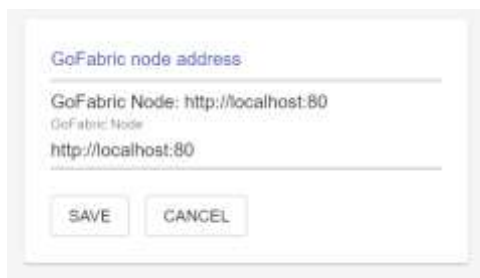
Abrir a aplicação Web.

`./run-cc-web.sh`

Abrir a aplicação acessando no browser.

<http://localhost:8080>

Configurar a aplicação e configurar para usar a org1.



GoFabric node address

GoFabric Node:

GoFabric Node:

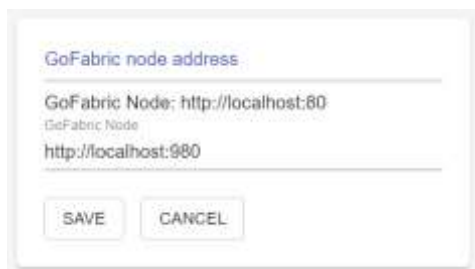
Utilizando o web app:

Criar uma pessoa.

Atualizar os dados de uma pessoa.

Mudar a aplicação para acessar a org2.

## TRILHA BLOCKCHAIN HYPERLEDGER FABRIC



GoFabric node address

GoFabric Node: `http://localhost:80`

GoFabric Node

`http://localhost:980`

Atualizar novamente os dados de uma pessoa.

Listar o histórico (ledger imutável) na aplicação web e gravar o resultado.