

# PROYECTO CESTA INSTALL

---

INSTALACIÓN CON DOCKER DE MYSQL Y ETHEREUM

```
docker run --name curso-mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:8.0.28
```

```
docker run --name curso-pg-14 -e POSTGRES_PASSWORD=my-secret-pw -p 5437:5432 -d postgres:13
```

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=my-secret-pw" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2019-CU15-ubuntu-20.04
```

```
docker run -d -p 1521:1521 --name curso-oracle-14 store/oracle/database-enterprise:12.2.0.1-Slim  
USUARIO:SYS AS SYSDBA password: Oradoc_db1
```

EN ESTE DIRECTORIO EN GITHUB TENEMOS LOS SCRIPT PARA CARGAR LA BASE DE DATOS NORTHWIND  
<https://github.com/Jviejo/curso-dbs-14/tree/master/northwind>

## ORACLE CONEXIÓN

General Oracle properties Driver properties SSH Proxy

Connection Type:

Basic TNS Custom

Host: localhost Port: 1521

Database: ORCLCDB.localdomain Service Name

Authentication

Authentication: Oracle Database Native

Nombre de usuario: sys Role: SYSDBA

Contraseña:  Save password local

Client:

*i* You can use variables in connection parameters.

Driver name: Oracle Edit Driver Settings

General Driver properties SSH Proxy SSL

Host: localhost Port: 1433

Database/Schema: northwind

Authentication: SQL Server Authentication

User name: sa

Password: ●●●●●●●●

Settings

Save password locally

Show All Schemas

[i](#) You can use variables in connection parameters.

Driver name: MS SQL Server / SQL Server [Edit Driver Settings](#)

## SQL SERVER CONEXIÓN

## MYSQL CONEXIÓN

General Driver properties SSH Proxy SSL

Server

Server Host: localhost Port: 3306

Database:

Authentication (Database Native)

Nombre de usuario: root

Contraseña:  Save password local

Advanced

Server Time Zone: Auto-detect

Local Client: MySQL Binaries

*i* You can use variables in connection parameters.

Driver name: MySQL [Edit Driver Settings](#)

General PostgreSQL Driver properties SSH Proxy SSL

Server

Host: localhost Port: 5439

Database: postgres

Authentication

Authentication: Database Native

Nombre de usuario: postgres

Contraseña:  Save password local

Advanced

Session role: Local Client: PostgreSQL Binario

📘 You can use variables in connection parameters.

Driver name: PostgreSQL Edit Driver Settings

## POSTGRES CONEXIÓN

```
docker run -d -v ${PWD}/data:/data -v ${PWD}/genesis.json:/genesis.json --
name eth-node ethereum/client-go init --datadir data /genesis.json
```

```
{
  "config": {
    "chainId": 8888,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "ethash": {}
  },
  "difficulty": "1",
  "gasLimit": "12000000",
  "alloc": {
    "0xff21E724B7D483fc93708855AbE6ee4f1eD97BF3": {
      "balance": "10000000000000000000000000000000"
    }
  }
}
```

Se ejecuta la imagen para inicializar el directorio con la base de datos de Ethereum

Se mapea `${PWD}/data` al directorio `data` del container  
Se mapea el fichero `genesis.json` al `/genesis.json`

Visitar el directorio `${PWD}/data` para ver el contenido. Ahí está la base de datos `leveldb` que soporta el blockchain

## LANZAMIENTO DEL NODO

```
docker run -d -p 8545:8545 -p:30303:30303 -v ${PWD}/data:/data --name eth-node-01
ethereum/client-go --datadir /data --http.api personal,eth,net,web3 --http --http.addr 0.0.0.0 --
http.port 8545 --http.corsdomain '*' --mine --miner.etherbase
0xff21E724B7D483fc93708855AbE6ee4f1eD97BF3 --miner.threads=1
```

Publica los puertos 8545 y 30303

Mapea el directorio local `${PWD}/data` al directorio `data` del container

El nombre del contenedor es `eth-node-01`

La imagen es `Ethereum/client-go`

El api esta activa en cualquier ip (0.0.0.0) en el 8545

Cualquier dominio puede acceder `http.corsdomain '*'`

La dirección de minería es `0xff...` y se dedica un thread