Práctica Lección 13. ***Blockchain.*** *Introducción a Ethereum*

Ejecutar un nodo de Ethereum

Configuración de un nodo de una red privada Ethereum.

Primero se crea el bloque génesis se define mediante el fichero "*genesis.json*" y contiene el identificador y parámetros de consenso. Luego se lanza el nodo especificando los siguientes parámetros:

- Identificador de red: 12342
- Directorio de datos: ./chaindata
- Bloque génesis: ./genesis.json

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.535]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd C:\Program Files\Geth

C:\Program Files\Geth>geth --networkid 12342 --datadir ./chaindata init ./genesis.json
INFO [01-02|01:04:57.980] Maximum peer count                       ETH=50 LES=0 total=50
INFO [01-02|01:04:58.314] Allocated cache and file handles         database="C:\\Program Files\\Geth\\chaindata\\geth\\chaindata" cache=16.00MiB handles=16
INFO [01-02|01:04:58.342] Writing custom genesis block
INFO [01-02|01:04:58.345] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-02|01:04:58.360] Successfully wrote genesis state         database=chaindata hash=6dd601…9339c2
INFO [01-02|01:04:58.369] Allocated cache and file handles         database="C:\\Program Files\\Geth\\chaindata\\geth\\lightchaindata" cache=16.00MiB handles=16
INFO [01-02|01:04:58.397] Writing custom genesis block
INFO [01-02|01:04:58.400] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-02|01:04:58.412] Successfully wrote genesis state         database=lightchaindata hash=6dd601…9339c2

C:\Program Files\Geth>
```

Al realizar esto se crean dos bases de datos, una donde se encuentra toda la información de los bloques encadenados y otra más ligera donde se almacena la configuración.

Asignación de Ether en el bloque génesis

Creación de una cuenta de Ethereum para poder asignarle los fondos.

```
C:\Program Files\Geth>geth account new --datadir ./chaindata
INFO [01-02|01:18:11.165] Maximum peer count                       ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0x64EaA97909b3Eef3A35545c47e2144E0e1A4b4F7
Path of the secret key file: chaindata\keystore\UTC--2020-01-02T00-18-37.565351000Z--64eaa97909b3eef3a35545c47e2144e0e1a4b4f7

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!


C:\Program Files\Geth>
```

Se modifica el bloque genesis para incluir fondos en la dirección Etherium creada:

```
{
"config": {
"chainId": 12342,
"homesteadBlock": 0,
"eip150Block": 0,
"eip155Block": 0,
"eip158Block": 0
},
"difficulty": "0x400",
"gasLimit": "2100000",
"alloc": {
"64EaA97909b3Eef3A35545c47e2144E0e1A4b4F7": {
"balance": "1000000000000000000000"
}
}
}
```

Como se creo la base de datos con el bloque genesis anterior, hay que destruir la base de datos y levantarla con el nuevo bloque; destrucción de base de datos:

```
C:\Program Files\Geth>geth removedb --datadir ./chaindata
INFO [01-02|01:25:48.518] Maximum peer count                       ETH=50 LES=0 total=50
Remove full node state database (C:\Program Files\Geth\chaindata\geth\chaindata)? [y/n] y
INFO [01-02|01:26:00.459] Database successfully deleted            path="C:\\Program Files\\Geth\\chaindata\\geth\\chaindata" elapsed=4.989ms
INFO [01-02|01:26:00.472] Full node ancient database missing       path="C:\\Program Files\\Geth\\chaindata\\geth\\chaindata\\ancient"
Remove light node database (C:\Program Files\Geth\chaindata\geth\lightchaindata)? [y/n] y
INFO [01-02|01:26:03.817] Database successfully deleted            path="C:\\Program Files\\Geth\\chaindata\\geth\\lightchaindata" elapsed=5.002ms
```

Creación de la nuevas bases de datos con el bloque genesis con saldo:

```
C:\Program Files\Geth>geth --networkid 12342 --datadir ./chaindata init ./genesis.json
INFO [01-02|01:26:41.509] Maximum peer count                       ETH=50 LES=0 total=50
INFO [01-02|01:26:41.874] Allocated cache and file handles         database="C:\\Program Files\\Geth\\chaindata\\geth\\chaindata" cache=16.00MiB handles=16
INFO [01-02|01:26:41.903] Writing custom genesis block
INFO [01-02|01:26:41.908] Persisted trie from memory database      nodes=1 size=149.00B time=997.6µs gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-02|01:26:41.928] Successfully wrote genesis state         database=chaindata hash=447df9…0d16fd
INFO [01-02|01:26:41.938] Allocated cache and file handles         database="C:\\Program Files\\Geth\\chaindata\\geth\\lightchaindata" cache=16.00MiB handles=16
INFO [01-02|01:26:41.974] Writing custom genesis block
INFO [01-02|01:26:41.978] Persisted trie from memory database      nodes=1 size=149.00B time=0s      gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-02|01:26:41.996] Successfully wrote genesis state         database=lightchaindata hash=447df9…0d16fd
```

Ejecución del nodo de Ethereum

En una terminal se ejecuta el nodo de Ethereum que hemos configurado anteriormente:

```
C:\Program Files\Geth>geth --identity "Practica13" --networkid 12342 --datadir ./chaindata --nodiscover -rpc --allow-insecure-unlock
INFO [01-02|03:11:32.550] Maximum peer count                       ETH=50 LES=0 total=50
INFO [01-02|03:11:32.892] Starting peer-to-peer node               instance=Geth/Practica13/v1.9.9-stable-01744997/windows-amd64/go1.13.4
INFO [01-02|03:11:32.900] Allocated trie memory caches             clean=256.00MiB dirty=256.00MiB
INFO [01-02|03:11:32.906] Allocated cache and file handles         database="C:\\Program Files\\Geth\\chaindata\\geth\\chaindata" cache=512.00MiB handles=8192
INFO [01-02|03:11:33.204] Opened ancient database                  database="C:\\Program Files\\Geth\\chaindata\\geth\\chaindata\\ancient"
INFO [01-02|03:11:33.215] Initialised chain configuration          config="{ChainID: 12342 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: 0 EIP155: 0 EIP158: 0 Byzan
tium: <nil> Constantinople: <nil> Petersburg: <nil> Istanbul: <nil>, Muir Glacier: <nil>, Engine: unknown}"
INFO [01-02|03:11:33.232] Disk storage enabled for ethash caches   dir="C:\\Program Files\\Geth\\chaindata\\geth\\ethash" count=3
INFO [01-02|03:11:33.240] Disk storage enabled for ethash DAGs     dir=C:\\Users\\34722\\AppData\\Local\\Ethash count=2
INFO [01-02|03:11:33.249] Initialising Ethereum protocol           versions="[64 63]" network=12342 dbversion=7
INFO [01-02|03:11:33.257] Loaded most recent local header          number=0 hash=447df9…0d16fd td=1024 age=50y8mo3w
INFO [01-02|03:11:33.265] Loaded most recent local full block      number=0 hash=447df9…0d16fd td=1024 age=50y8mo3w
INFO [01-02|03:11:33.274] Loaded most recent local fast block      number=0 hash=447df9…0d16fd td=1024 age=50y8mo3w
INFO [01-02|03:11:33.283] Loaded local transaction journal         transactions=0 dropped=0
INFO [01-02|03:11:33.294] Regenerated local transaction journal    transactions=0 accounts=0
INFO [01-02|03:11:33.877] Allocated fast sync bloom                size=512.00MiB
INFO [01-02|03:11:33.884] Initialized fast sync bloom              items=1 errorrate=0.000 elapsed=0s
INFO [01-02|03:11:33.930] New local node record                    seq=3 id=8befa5d1e17aa00a ip=127.0.0.1 udp=0 tcp=30303
INFO [01-02|03:11:33.959] Started P2P networking                   self="enode://7ce2af339cb67eef7d75d3299828e20d3527aaa8e8a7dfcf889a4fa810d5789fa5ac0b1ddf8af74e4a0791a
732880fb5626fccb3de988ccc4dbc4e505b2c2cd7@127.0.0.1:30303?discport=0"
INFO [01-02|03:11:33.930] IPC endpoint opened                      url=\\\\.\\pipe\\geth.ipc
INFO [01-02|03:11:34.005] HTTP endpoint opened                     url=http://127.0.0.1:8545 cors= vhosts=localhost
INFO [01-02|03:11:36.610] Mapped network port                      proto=tcp extport=30303 intport=30303 interface="UPNP IGDv2-IP2"
```

Ahora para poder interactuar con el nodo, tenemos que lanzar otra instancia de la aplicación especificándole dónde tiene que conectarse en una ventana diferente. Para ello se utiliza IPC (Inter-Process Communicactions) y se especifica la ruta donde está el fichero “geth.ipc”:

```
C:\Program Files\Geth>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/Practica13/v1.9.9-stable-01744997/windows-amd64/go1.13.4
coinbase: 0x64eaa97909b3eef3a35545c47e2144e0e1a4b4f7
at block: 0 (Thu, 01 Jan 1970 01:00:00 CET)
 datadir: C:\Program Files\Geth\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0
```

Se verifica la cuenta que nos creamos antes y se comprueba que tiene los fondos que estaban en el bloque génesis:

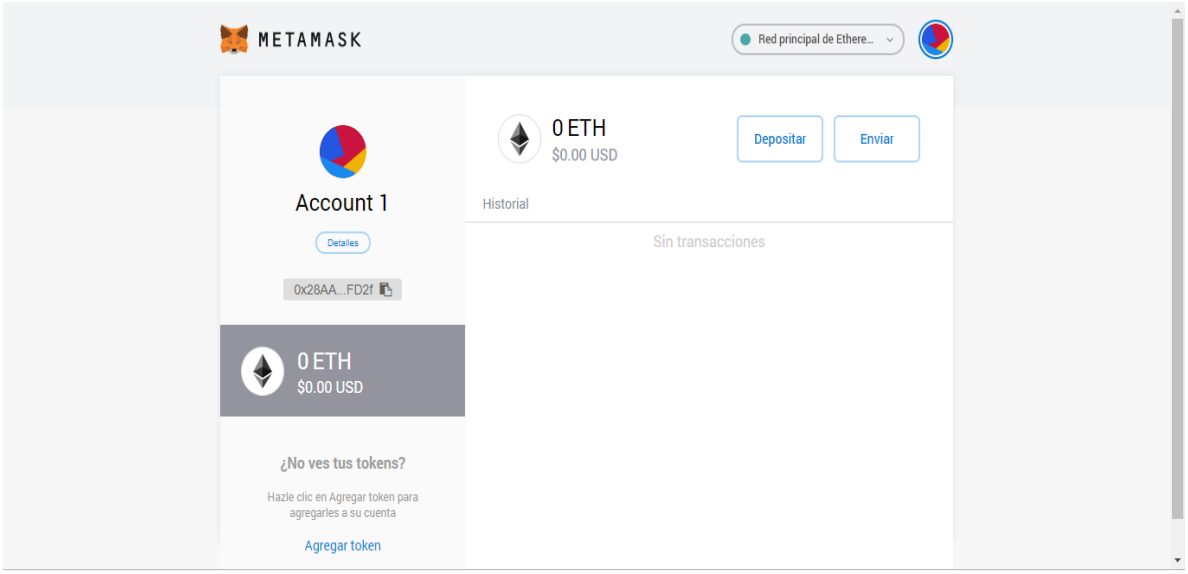
```
Administrador: Símbolo del sistema - geth  attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/Practica13/v1.9.9-stable-01744997/windows-amd64/go1.13.4
coinbase: 0x64eaa97909b3eef3a35545c47e2144e0e1a4b4f7
at block: 0 (Thu, 01 Jan 1970 01:00:00 CET)
 datadir: C:\Program Files\Geth\chaindata
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> eth.accounts
["0x64eaa97909b3eef3a35545c47e2144e0e1a4b4f7"]
> web3.fromWei(eth.getBalance(eth.accounts[0]), "ether")
100
>
```

Eth accounts hace una consulta de las cuentas creadas en el chaindata, informando que disponemos de 100 Eth que se corresponde a lo que hemos agregado en el genesis.json

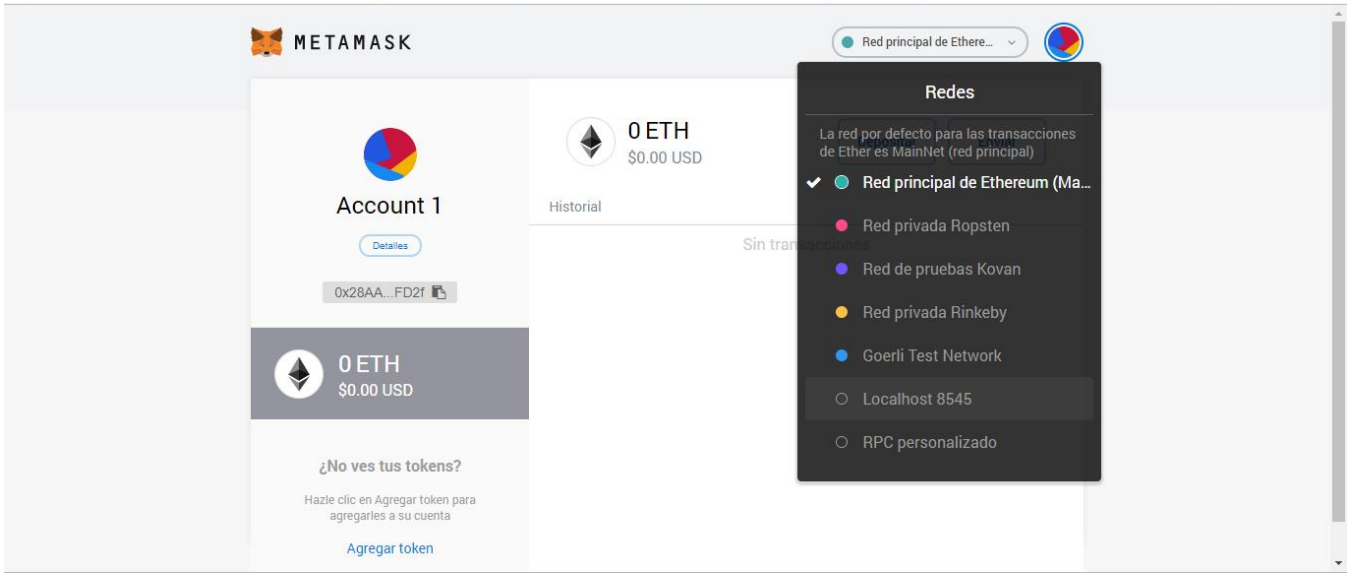
Interactuar con el nodo mediante una Wallet de Ethereum

Los “seed words”, generan todas las cuentas de Ethereum de la wallet de forma determinista, por lo que si se pierden o son robadas, los fondos se habrán perdido. Una wallet determinista se basa en derivar claves desde un punto de partida conocido como semilla maestra; La semilla permite al usuario realizar fácilmente una copia de seguridad y restaurar una billetera sin necesitar ninguna otra información.

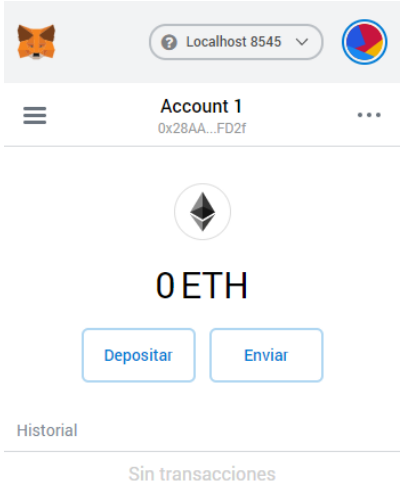
Se sigue el procedimiento para crear una wallets de Ethereum más utilizadas, Metamask:



Configurar Metamask con nuestro nodo



Metamask conectado al nodo local (localhost 8545):



Enviar fondos desde el nodo a Metamask

A continuación se va a mandar ether desde el nodo (que tiene 100 ETH desde el génesis block) a la wallet de Metamask.

```
> sender=eth.accounts[0]
"0x64eaa97909b3eef3a35545c47e2144e0e1a4b4f7"
> receiver="0x28AA527fDFB13df7EDe3E54108A1a8d9079AFD2f"
"0x28AA527fDFB13df7EDe3E54108A1a8d9079AFD2f"
> amount = web3.toWei(1,"ether")
"1000000000000000000"
> eth.sendTransaction({from:sender, to:receiver, value: amount})
Error: authentication needed: password or unlock
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at <anonymous>:1:1
```

El envío de la transacción falla debido a que la cuenta (que se creo al principio de la práctica) estaba protegida por una contraseña que no ha sido preguntada en ningún momento, luego de desbloquearla, se obtiene el resultado del envío es la identificación de la transacción que se corresponde con el valor de su hash.

```
> personal.unlockAccount(sender)
Unlock account 0x64eaa97909b3eef3a35545c47e2144e0e1a4b4f7
Password:
true
> eth.sendTransaction({from:sender, to:receiver, value: amount})
"0x0bd73d80444a89f9298edd2578ebce6e3003c5c83067b943a6aeffbaa0e72afa"
>
```

La transacción se encuentra en el “memory pool” del nodo, que es dónde se guardan todas aquellas transacciones recibidas que deberían ser incluidas en un bloque.


```
> txpool.status
{
  pending: 1,
  queued: 0
}
> txpool.content
{
  pending: {
    0x64EaA97909b3Eef3A35545c47e2144E0e1A4b4F7: {
      0: {
        blockHash: null,
        blockNumber: null,
        from: "0x64eaa97909b3eef3a35545c47e2144e0e1a4b4f7",
        gas: "0x5208",
        gasPrice: "0x3b9aca00",
        hash: "0x0bd73d80444a89f9298edd2578ebce6e3003c5c83067b943a6aeffbaa0e72afa",
        input: "0x",
        nonce: "0x0",
        r: "0x3e4a4403321ebdb8308eb42e2dc7f03f66e17286c419af83dfc4b32ff603f09e",
        s: "0x6a732f772f0dfb5cea579dc6f9b2efd66c8d869d64324a580731899cf3926c6b",
        to: "0x28aa527fdfb13df7ede3e54108a1a8d9079afd2f",
        transactionIndex: null,
        v: "0x608f",
        value: "0xde0b6b3a7640000"
      }
    }
  },
  queued: {}
}
```


Habilitar el minado en nodo

Por último se va a habilitar el minado en el nodo de forma que la transacción creada antes pueda ser incluida en un nodo y que los fondos lleguen a Metamask, A continuación en la ventana donde se está ejecutando el proceso del nodo:

```
INFO [01-02|03:36:51.986] Updated mining threads                   threads=4
INFO [01-02|03:36:51.990] Transaction pool price threshold updated price=1000000000
INFO [01-02|03:36:51.995] Commit new mining work                   number=1 sealhash=183917…4e213a uncles=0 txs=0 gas=0 fees=0 elapsed=0s
INFO [01-02|03:36:52.018] Commit new mining work                   number=1 sealhash=352edc…6e1106 uncles=0 txs=1 gas=21000 fees=2.1e-05 elapsed=23.013ms
INFO [01-02|03:36:54.767] Generating DAG in progress               epoch=0 percentage=0 elapsed=1.995s
INFO [01-02|03:36:56.746] Generating DAG in progress               epoch=0 percentage=1 elapsed=3.974s
INFO [01-02|03:36:58.649] Generating DAG in progress               epoch=0 percentage=2 elapsed=5.876s
INFO [01-02|03:37:00.683] Generating DAG in progress               epoch=0 percentage=3 elapsed=7.910s
INFO [01-02|03:37:02.661] Generating DAG in progress               epoch=0 percentage=4 elapsed=9.888s
INFO [01-02|03:37:05.073] Generating DAG in progress               epoch=0 percentage=5 elapsed=12.301s
INFO [01-02|03:37:07.967] Generating DAG in progress               epoch=0 percentage=6 elapsed=15.194s
INFO [01-02|03:37:10.778] Generating DAG in progress               epoch=0 percentage=7 elapsed=18.005s
INFO [01-02|03:37:13.238] Generating DAG in progress               epoch=0 percentage=8 elapsed=20.466s
INFO [01-02|03:37:15.586] Generating DAG in progress               epoch=0 percentage=9 elapsed=22.814s
INFO [01-02|03:37:17.752] Generating DAG in progress               epoch=0 percentage=10 elapsed=24.980s
INFO [01-02|03:37:19.761] Generating DAG in progress               epoch=0 percentage=11 elapsed=26.988s
INFO [01-02|03:37:21.868] Generating DAG in progress               epoch=0 percentage=12 elapsed=29.095s
INFO [01-02|03:37:23.958] Generating DAG in progress               epoch=0 percentage=13 elapsed=31.185s
INFO [01-02|03:37:26.064] Generating DAG in progress               epoch=0 percentage=14 elapsed=33.291s
INFO [01-02|03:37:28.158] Generating DAG in progress               epoch=0 percentage=15 elapsed=35.385s
INFO [01-02|03:37:31.372] Generating DAG in progress               epoch=0 percentage=16 elapsed=38.599s
INFO [01-02|03:37:34.036] Generating DAG in progress               epoch=0 percentage=17 elapsed=41.263s
INFO [01-02|03:37:36.424] Generating DAG in progress               epoch=0 percentage=18 elapsed=43.651s
INFO [01-02|03:37:38.935] Generating DAG in progress               epoch=0 percentage=19 elapsed=46.162s
INFO [01-02|03:37:41.335] Generating DAG in progress               epoch=0 percentage=20 elapsed=48.562s
INFO [01-02|03:37:43.850] Generating DAG in progress               epoch=0 percentage=21 elapsed=51.077s
INFO [01-02|03:37:46.210] Generating DAG in progress               epoch=0 percentage=22 elapsed=53.438s
INFO [01-02|03:37:48.462] Generating DAG in progress               epoch=0 percentage=23 elapsed=55.690s
INFO [01-02|03:37:51.625] Generating DAG in progress               epoch=0 percentage=24 elapsed=58.852s
INFO [01-02|03:37:55.597] Generating DAG in progress               epoch=0 percentage=25 elapsed=1m2.824s
INFO [01-02|03:37:59.127] Generating DAG in progress               epoch=0 percentage=26 elapsed=1m6.354s
INFO [01-02|03:38:02.167] Generating DAG in progress               epoch=0 percentage=27 elapsed=1m9.394s
```

Después de que la transacción haya sido minada podemos comprobar que el pool de transacciones está vacío y que la cuenta de Metamask contará con fondos recibidos del nodo.

```
> txpool.status
{
  pending: 0,
  queued: 0
}
> txpool.content
{
  pending: {},
  queued: {}
}
>
```