



---

# OPEN SOURCE BLOCKCHAIN

---

Desarrollo de Tecnologías Emergentes – TG3



7 DE MAYO DE 2018  
GRUPO T1

## Contenido

1. Autores del trabajo, planificación y entrega.....	3
1.1 Autores.....	3
1.2 Planificación .....	3
1.3 Entrega .....	3
2. Requisitos del prototipo a implementar .....	4
2.1 Requisitos funcionales.....	4
2.2 Requisitos NO funcionales .....	5
3. Criterios de comparación en la implementación.....	6
3.1 Criterio 1: Documentación .....	6
3.2 Criterio 2: Horas empleadas .....	6
3.3 Criterio 3: Líneas de código .....	6
3.4 Criterio 4: Facilidad de implementación .....	6
3.5 Criterio 5: Velocidad de arranque .....	6
3.6 Criterio 6: Velocidad de funcionamiento .....	7
3.7 Criterio 7: Cantidad de nodos .....	7
3.8 Criterio 8: Caída de los nodos .....	7
3.9 Criterio 9: Soporte .....	7
4. Proyecto de implementación de un prototipo del sistema utilizando MultiChain .....	8
4.1 Documentación de diseño .....	8
4.2 Documentación de construcción.....	8
4.3 Documentación de pruebas.....	14
4.3.1 Comprobaciones .....	14
4.3.2 Errores .....	15
4.4 Documentación de instalación.....	17
4.4.1 Proceso de instalación de las máquinas virtuales .....	17
4.4.2 Instalación.....	19
4.5 Manual de usuario.....	23
5. Proyecto de implementación de un prototipo del sistema utilizando la Monax .....	24
5.1 Documentación de diseño .....	24
5.2 Documentación de construcción.....	24
5.3 Documentación de pruebas.....	26
5.4 Documentación de instalación.....	28
5.4.1 Proceso de instalación de las máquinas virtuales .....	28
5.4.2 Instalación de Docker y Monax .....	30
5.5 Manual de usuario.....	32
6. Comparación de las dos implementaciones .....	33

6.1 Evaluación de los criterios en la implementación usando MultiChain .....	33
6.2 Evaluación de los criterios en la implementación usando Monax .....	33
7. Comparación de la implementación de las tecnologías.....	34
8. Conclusiones .....	36
ANEXO 1: Otros Requisitos .....	36
A.1. MultiChain.....	36
A.1.1. Requisitos Funcionales.....	36
A.1.2. Requisitos NO Funcionales .....	36
A.2 Monax.....	36
A.2.1. Requisitos Funcionales.....	36
A.2.2. Requisitos NO Funcionales .....	36

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

Este grupo es el T1 y está formado por:

- Alejandro Díaz Moreno (Coordinador del grupo T1)
- Daniel Ortega Expósito
- Daniela Guzmán Pisfil
- María Castro Vaquerizo
- Paula Hernández Jordá

## 1.2 Planificación

Como la herramienta GanttPro ya no permite compartir mediante URL la planificación hemos tenido que añadir al profesor de la asignatura a la misma.

URL →<https://app.ganttpro.com/#!/app/home>

## 1.3 Entrega

Indicamos el enlace (URL) a un repositorio en GitHub:

[https://github.com/alex7dm/DTE\\_T1\\_OpenSourceBlockchain](https://github.com/alex7dm/DTE_T1_OpenSourceBlockchain)

En dicho repositorio encontraremos:

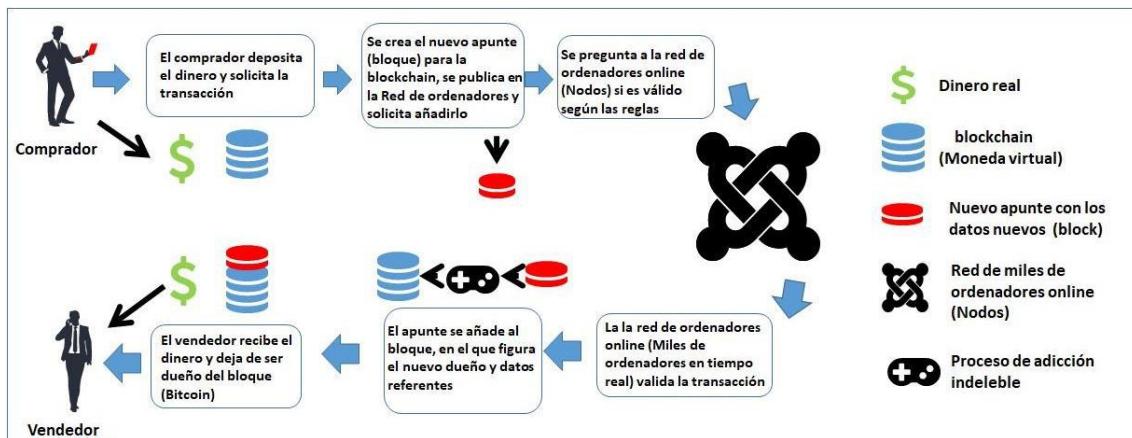
- El trabajo terminado con el nombre **TG3\_final.pdf**
- La presentación del trabajo con el nombre **TG3\_final.pptx**
- Prototipos obtenidos implementando con las tecnologías escogidas:
  - o **MultiChain\_final.zip**
  - o **Monax\_final.zip**

## 2. Requisitos del prototipo a implementar

Vamos a utilizar BlockChain para realizar una transacción que permita realizar un intercambio de activos entre dos o más partes.

Por ejemplo, el envío de un activo (dólares) de Marta a Jorge, al mismo tiempo que envía un activo en euros de Bob a Alicia. Como el intercambio se realiza con una sola transacción significa que, o todas las transacciones de activos tienen lugar simultáneamente o ninguna tiene lugar.

En el mundo de las finanzas este tipo de transacciones se denomina entrega contra pago (DvP)



A continuación, se detallan todos los requisitos para implementar:

### 2.1 Requisitos funcionales

Los requisitos funcionales deben ser los mismos para las dos implementaciones.

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema.

REQ.	DESCRIPCIÓN
RF01	Realizar la transacción
RF02	Instalar la BlockChain
RF03	Conectarse a la BlockChain creada
RF04	Instalación de los entornos
RF05	Cada transacción tendrá un numero identificativo diferente

## 2.2 Requisitos NO funcionales

En la siguiente tabla se indicará el catálogo de requisitos no funcionales del sistema.

REQ.	DESCRIPCIÓN
R01	Documentación
R02	El retardo de la transacción no debe ser excesivo
R03	Seguridad
R04	Implementación en Linux
R05	La implementación se debe poder llevar a cabo en una máquina virtual sencilla
R06	La implementación se debe poder llevar a cabo en una máquina con 2 GB de RAM
R07	La implementación se debe poder llevar a cabo en una CPU con 100% de límite de ejecución
R08	Se utilizará la versión 5.2.4 de Virtual Box
R09	Después de la implementación de la BlockChain, el número medio de errores cometidos por desarrollador no excederá de dos por día.

### **3. Criterios de comparación en la implementación**

En el trabajo TG2 se definieron criterios de comparación de las dos tecnologías a nivel teórico.

En este trabajo hay que definir criterios para la comparación de la implementación de las tecnologías en la construcción del prototipo de sistema de ejemplo, cuyos requisitos son los establecidos en el apartado 2.

#### **3.1 Criterio 1: Documentación**

**Nombre del criterio:** Documentación.

**Descripción:** Documentación encontrada para el desarrollo de la actividad.

**Tipo de valor:** Numérico

#### **3.2 Criterio 2: Horas empleadas**

**Nombre del criterio:** Horas empleadas en el desarrollo del sistema.

**Descripción:** Horas invertidas en el desarrollo de la tecnología

**Tipo de valor:** Numérico (horas).

#### **3.3 Criterio 3: Líneas de código**

**Nombre del criterio:** Líneas de código.

**Descripción:** Número de líneas programadas.

**Tipo de valor:** Numérico (líneas).

#### **3.4 Criterio 4: Facilidad de implementación**

**Nombre del criterio:** Facilidad de implementación

**Descripción:** Facilidad de reacción frente a la implementación o los fallos producidos.

**Tipo de valor:** Booleano (Si/No)

#### **3.5 Criterio 5: Velocidad de arranque**

**Nombre del criterio:** Velocidad de funcionamiento

**Descripción:** Tiempo que tarda en iniciarse el sistema y conectarse los nodos.

**Tipo de valor:** Texto libre

### **3.6 Criterio 6: Velocidad de funcionamiento**

**Nombre del criterio:** Velocidad de funcionamiento

**Descripción:** Tiempo que tarda en mostrar respuestas por pantalla.

**Tipo de valor:** Texto libre

### **3.7 Criterio 7: Cantidad de nodos**

**Nombre del criterio:** Cantidad de nodos

**Descripción:** Número de nodos necesarios para el funcionamiento de la BlockChain.

**Tipo de valor:** Numérico.

### **3.8 Criterio 8: Caída de los nodos**

**Nombre del criterio:** Caída de los nodos.

**Descripción:** Número de veces que se han caído los nodos desde su creación.

**Tipo de valor:** Numérico.

### **3.9 Criterio 9: Soporte**

**Nombre del criterio:** Soporte.

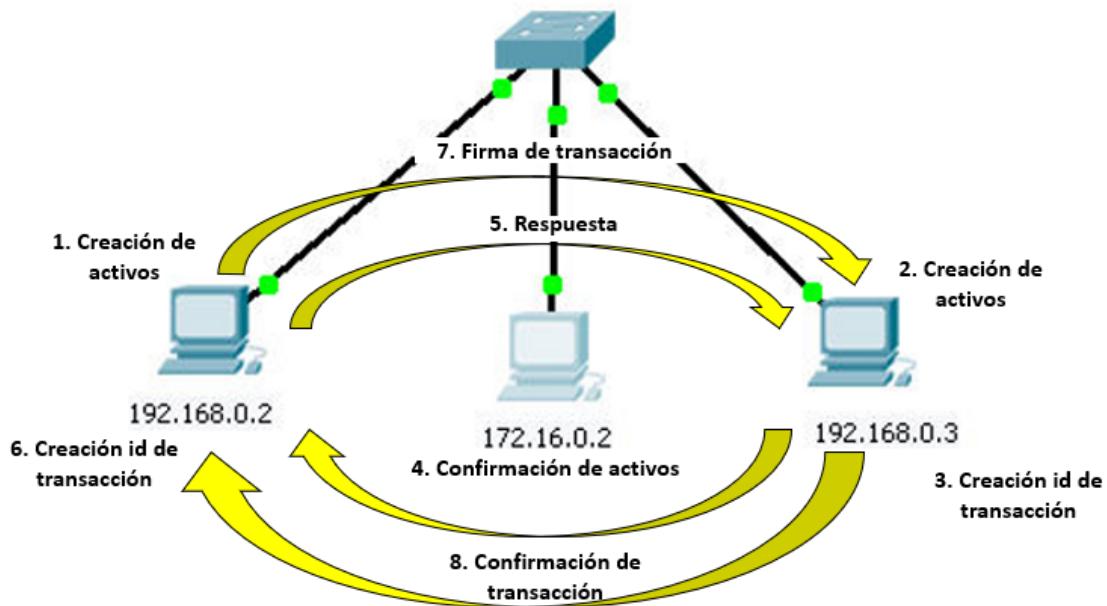
**Descripción:** Nivel de calidad de respuesta antes las dudas planteadas en su página oficial

**Tipo de valor:** Escala (1-5)

## 4. Proyecto de implementación de un prototipo del sistema utilizando MultiChain

### 4.1 Documentación de diseño

Diagrama de funcionamiento de MultiChain



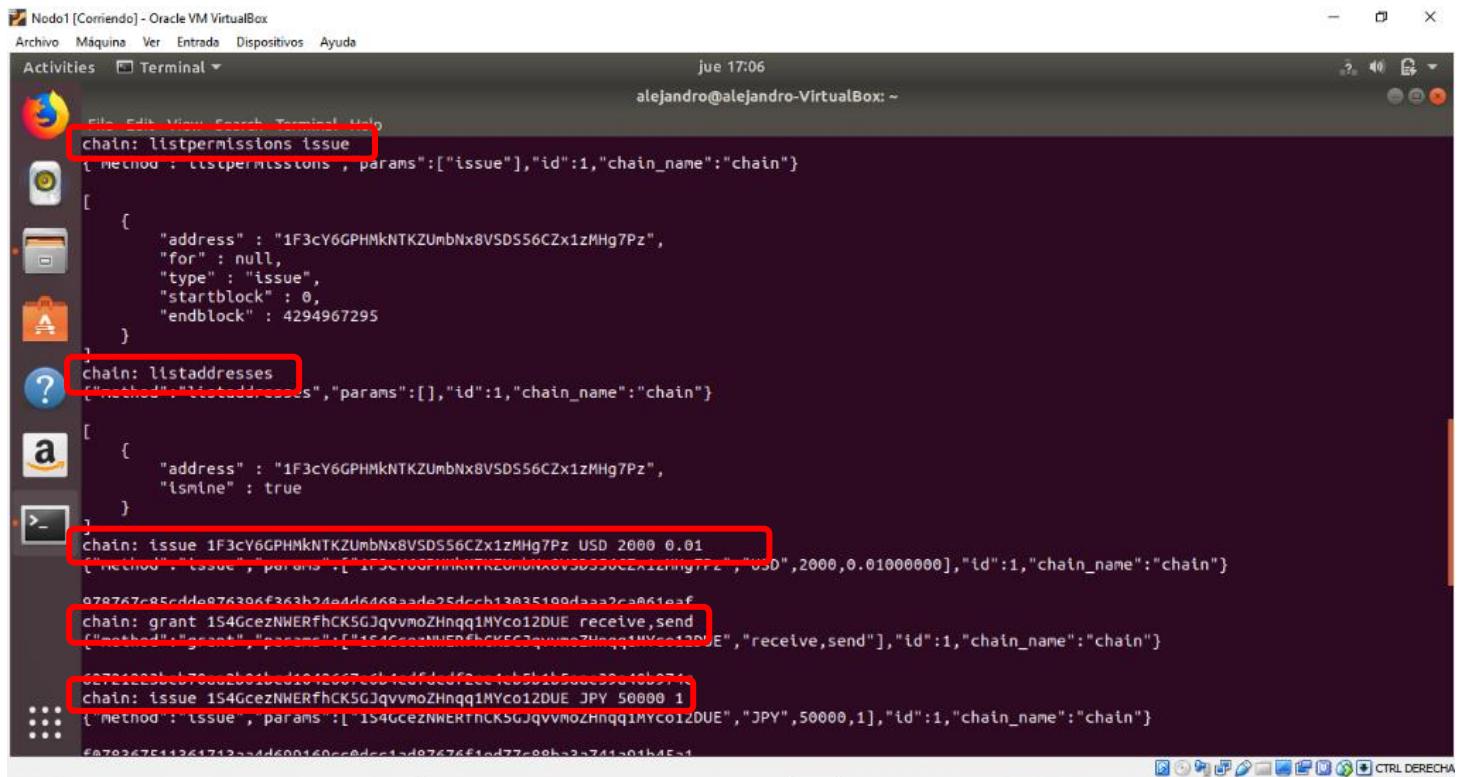
### 4.2 Documentación de construcción

A continuación, explicamos paso a paso como construimos y realizamos la transacción.

Primero hay que entrar en la chain en modo interactivo en ambos nodos usando el comando `multichain-cli {nombre de la chain}`

Listar la dirección que tiene permiso para crear activos. Se crea el activo en el nodo 1.

Se ve la dirección en el nodo 2 a la que hay que conceder permisos desde el nodo 1 para poder crear activos. Después de conceder los permisos se crean los activos.



The screenshot shows a terminal window titled "Nodo1 [Corriendo] - Oracle VM VirtualBox". The terminal displays a series of JSON objects representing blockchain operations:

- The first object is a response to the command "chain: listpermissions issue":

```
chain: listpermissions issue
{
  "method": "listpermissions",
  "params": ["issue"], "id": 1, "chain_name": "chain"
}
```
- The second object is a response to "chain: listaddresses":

```
chain: listaddresses
{
  "method": "listaddresses",
  "params": [], "id": 1, "chain_name": "chain"
}
```
- The third object is a response to "chain: issue 1F3cY6GPHMkNTKZUmbNx8VSDS56Cz1zMHg7Pz USD 2000 0.01":

```
chain: issue 1F3cY6GPHMkNTKZUmbNx8VSDS56Cz1zMHg7Pz USD 2000 0.01
{
  "method": "issue",
  "params": ["1F3cY6GPHMkNTKZUmbNx8VSDS56Cz1zMHg7Pz", "USD", 2000, 0.01], "id": 1, "chain_name": "chain"
}
```
- The fourth object is a response to "chain: grant 154GcezNWERfhCKSGJqvvm0ZHnqq1MYco12DUE receive,send":

```
chain: grant 154GcezNWERfhCKSGJqvvm0ZHnqq1MYco12DUE receive,send
{
  "method": "grant",
  "params": ["154GcezNWERfhCKSGJqvvm0ZHnqq1MYco12DUE", "receive,send"], "id": 1, "chain_name": "chain"
}
```
- The fifth object is a response to "chain: issue 154GcezNWERfhCKSGJqvvm0ZHnqq1MYco12DUE JPY 50000 1":

```
chain: issue 154GcezNWERfhCKSGJqvvm0ZHnqq1MYco12DUE JPY 50000 1
{
  "method": "issue",
  "params": ["154GcezNWERfhCKSGJqvvm0ZHnqq1MYco12DUE", "JPY", 50000, 1], "id": 1, "chain_name": "chain"
}
```

```

Nodo2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal lun 18:05
alejandro@alejandro-VirtualBox: ~

Interactive mode

chain: listpermissions issue
{"method": "listpermissions", "params": [{"issue": true}, {"id": 1, "chain_name": "chain"}]}
[{"for": null, "type": "Issue", "startblock": 0, "endblock": 4294967295}]

chain: listaddresses
{"method": "listaddresses", "params": [{"id": 1, "chain_name": "chain"}]}
[{"address": "1F3cY6GPHMkNTKZUmNx8VSDS56Cz1zMh7Pz", "lsmine": true}]

chain: grant 154GcezNWERfhCK5GJqvvmozHnqq1MYco12DUE
{"method": "grant", "params": [{"154GcezNWERfhCK5GJqvvmozHnqq1MYco12DUE": true}, {"id": 1, "chain_name": "chain"}]}

error code: -1
error message:
grant "address(es)" "permission(s)" ( native-amount startblock endblock "comment" "comment-to" )

Grant permission(s) to a given address.

Arguments:
1. "address(es)" (string, required) The multichain addresses to send to (comma delimited).
2. "permission(s)" (string, required) Permission strings, comma delimited.
  Global: connect,send,receive,issue,mine,admin,activate,create
  or per-asset: asset-identifier.issue,admin
  or per-stream: stream-identifier.write,activate,admin

3. native-amount (numeric, optional) Native currency amount to send. eg 0.1. Default - 0.0
4. startblock (numeric, optional) Block to apply permissions from (inclusive). Default - 0
5. endblock (numeric, optional) Block to apply permissions to (exclusive). Default - 4294967295
If -1 is specified default value is used.

6. "comment" (string, optional) A comment used to store what the transaction is for.
This is not part of the transaction, just kept in your wallet.
7. "comment-to" (string, optional) A comment to store the name of the person or organization
to which you're sending the transaction. This is not part of the
transaction, just kept in your wallet.

```

Listamos los activos que hemos creado:

```

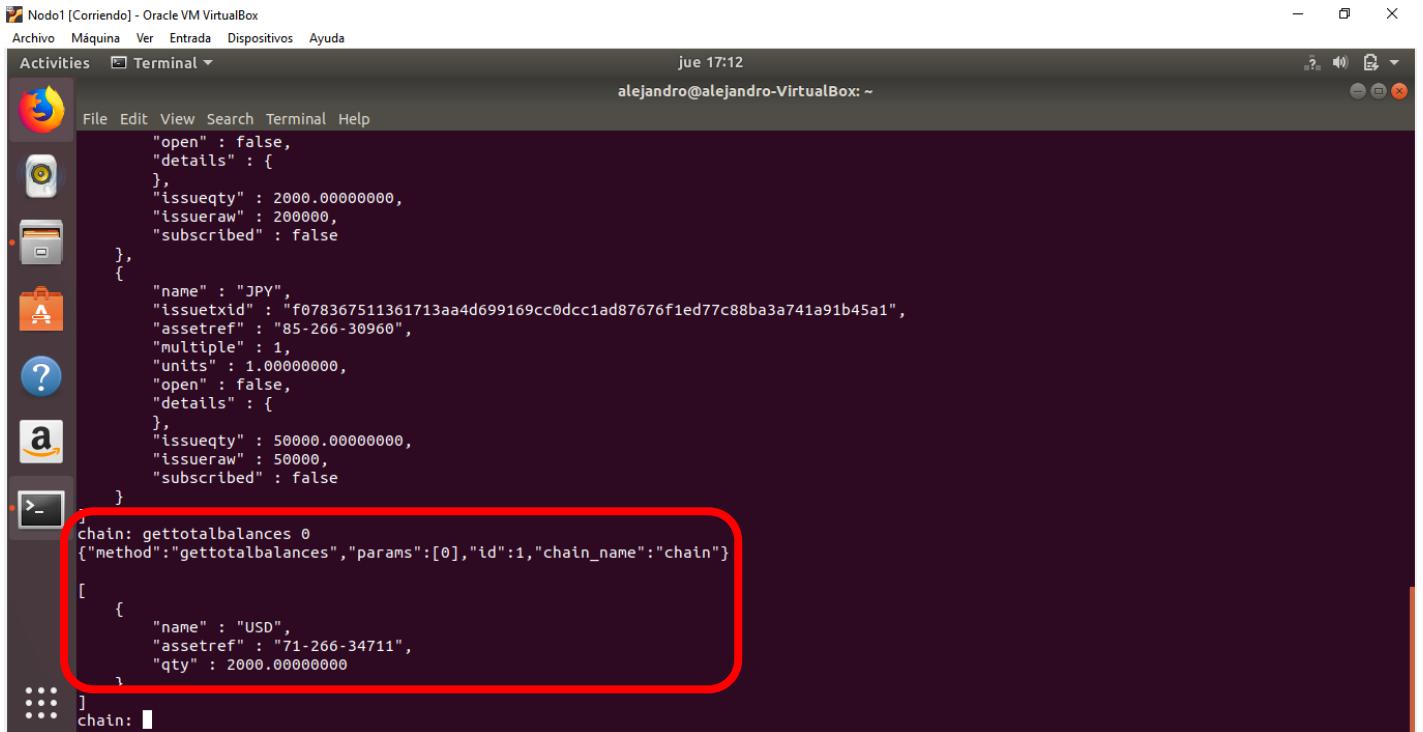
Activities Terminal jue 17:02
alejandro@alejandro-VirtualBox: ~

File Edit View Search Terminal Help
chain: listassets
{"method": "listassets", "params": [{"id": 1, "chain_name": "chain"}]}
[{"name": "USD", "issuetxid": "978767c85cdde876396f363b24e4d6468aade25dccb13035199daaa2ca061eaf", "assetref": "71-266-34711", "multiple": 100, "units": 0.01000000, "open": false, "details": {}, "issueqty": 2000.00000000, "issueraw": 200000, "subscribed": false}, {"name": "JPY", "issuetxid": "f078367511361713aa4d699169cc0dcc1ad87676f1ed77c88ba3a741a91b45a1", "assetref": "85-266-30960", "multiple": 1, "units": 1.00000000, "open": false, "details": {}, "issueqty": 50000.00000000, "issueraw": 50000, "subscribed": false}]

chain:

```

Se realiza get total balance para saber el saldo de ese nodo:



```
jue 17:12
alejandro@alejandro-VirtualBox: ~

File Edit View Search Terminal Help
"open" : false,
"details" : {
},
"issueqty" : 2000.00000000,
"issueraw" : 200000,
"subscribed" : false
},
{
  "name" : "JPY",
"issuetxid" : "f078367511361713aa4d699169cc0dcc1ad87676f1ed77c88ba3a741a91b45a1",
"assetref" : "85-266-30960",
"multiple" : 1,
"units" : 1.00000000,
"open" : false,
"details" : {
},
"issueqty" : 50000.00000000,
"issueraw" : 50000,
"subscribed" : false
}
]
chain: gettotalbalances 0
{"method":"gettotalbalances","params":[0],"id":1,"chain_name":"chain"}]

[
  {
    "name" : "USD",
    "assetref" : "71-266-34711",
    "qty" : 2000.00000000
  }
]
chain: 
```



```
lun 18:11
alejandro@alejandro-VirtualBox: ~

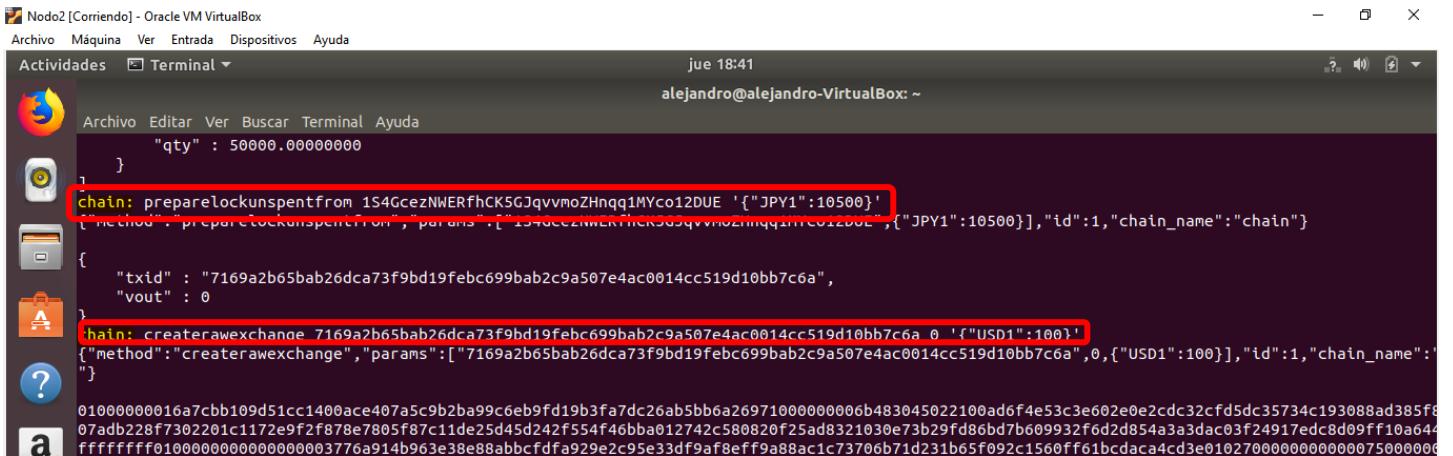
Archivo Editar Ver Buscar Terminal Ayuda
",

{
  "name" : "JPY",
"issuetxid" : "f078367511361713aa4d699169cc0dcc1ad87676f1ed77c88ba3a741a91b45a1",
"assetref" : null,
"multiple" : 1,
"units" : 1.00000000,
"open" : false,
"details" : {
},
"issueqty" : 50000.00000000,
"issueraw" : 50000,
"subscribed" : false
}
]
chain: gettotalbalance 0
{"method":"gettotalbalance","params":["0"],"id":1,"chain_name":"chain"}]

error code: -32601
error message:
method not found
chain: gettotalbalances 0
{"method":"gettotalbalances","params":[0],"id":1,"chain_name":"chain"}]

[
  {
    "name" : "USD",
    "assetref" : null,
    "qty" : 50000.00000000
  }
]
chain: 
```

En el nodo 2 creamos una transaccion de 10.500 yenes, pero a cambio pedimos 100 \$.

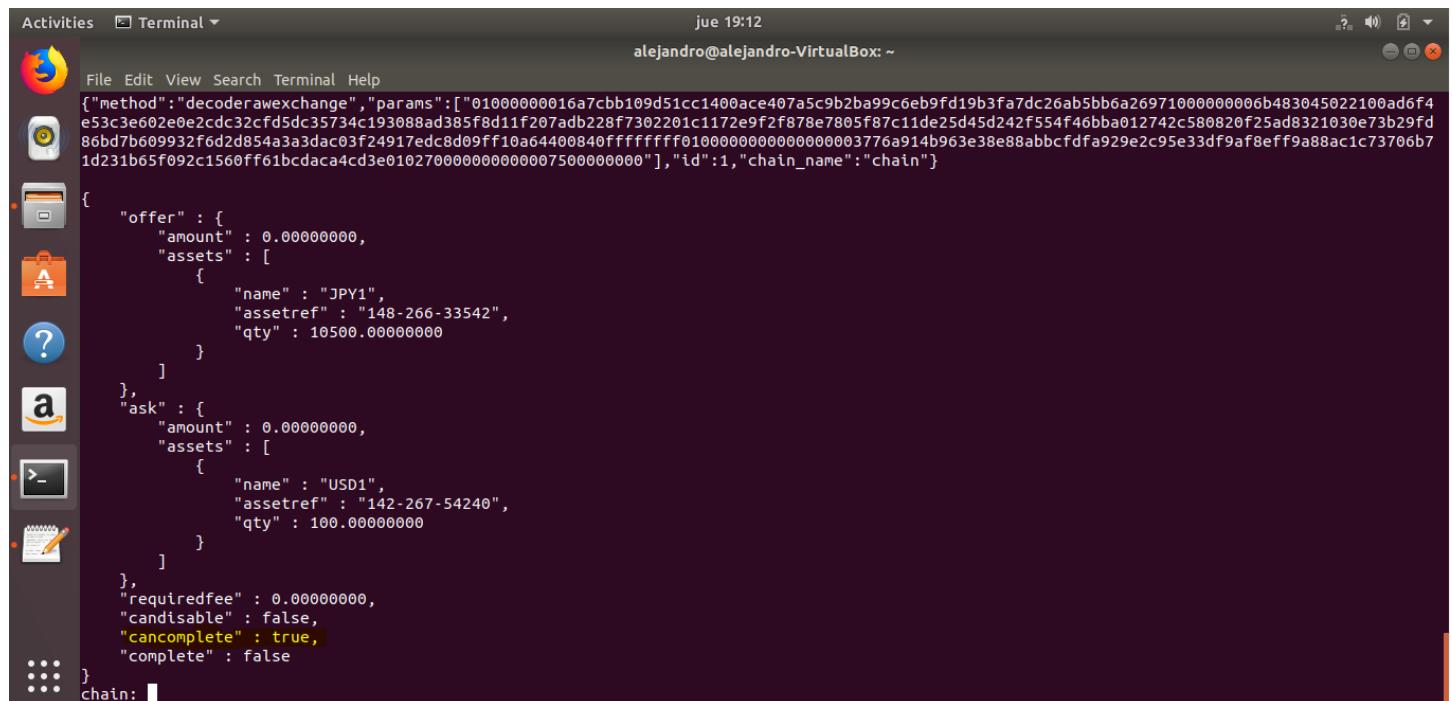


```
jue 18:41
alejandro@alejandro-VirtualBox: ~
{
  "method": "createRawExchange",
  "params": [
    {
      "txid": "7169a2b65bab26dca73f9bd19febcb699bab2c9a507e4ac0014cc519d10bb7c6a",
      "vout": 0
    }
  ],
  "chain": {
    "method": "prepareLockUnspentFrom",
    "params": [
      "1540ce2nWERThCK5GJqvvm0zHnqq1MYco12DUE",
      {"JPY1": 10500}
    ],
    "id": 1,
    "chain_name": "chain"
  }
}

{
  "offer": {
    "amount": 0.0000000,
    "assets": [
      {
        "name": "JPY1",
        "assetref": "148-266-33542",
        "qty": 10500.0000000
      }
    ],
    "ask": {
      "amount": 0.0000000,
      "assets": [
        {
          "name": "USD1",
          "assetref": "142-267-54240",
          "qty": 100.0000000
        }
      ],
      "requiredfee": 0.0000000,
      "candisable": false,
      "cancomplete": true,
      "complete": false
    }
  },
  "chain": []
}
```

El texto generado en hexadecimal es un id de transacción que contiene los datos de la misma también, y también muestra la información de la transacción

Hay que ejecutar el comando **decoderawexchange** y ese texto generado para confirmar si el nodo 1 tiene activos suficientes para poder llevarse a cabo la transacción.



```
jue 19:12
alejandro@alejandro-VirtualBox: ~
{
  "offer": {
    "amount": 0.0000000,
    "assets": [
      {
        "name": "JPY1",
        "assetref": "148-266-33542",
        "qty": 10500.0000000
      }
    ]
  },
  "ask": {
    "amount": 0.0000000,
    "assets": [
      {
        "name": "USD1",
        "assetref": "142-267-54240",
        "qty": 100.0000000
      }
    ],
    "requiredfee": 0.0000000,
    "candisable": false,
    "cancomplete": true,
    "complete": false
  }
}
```

Como cancomplete es true significa que tiene suficientes activos.



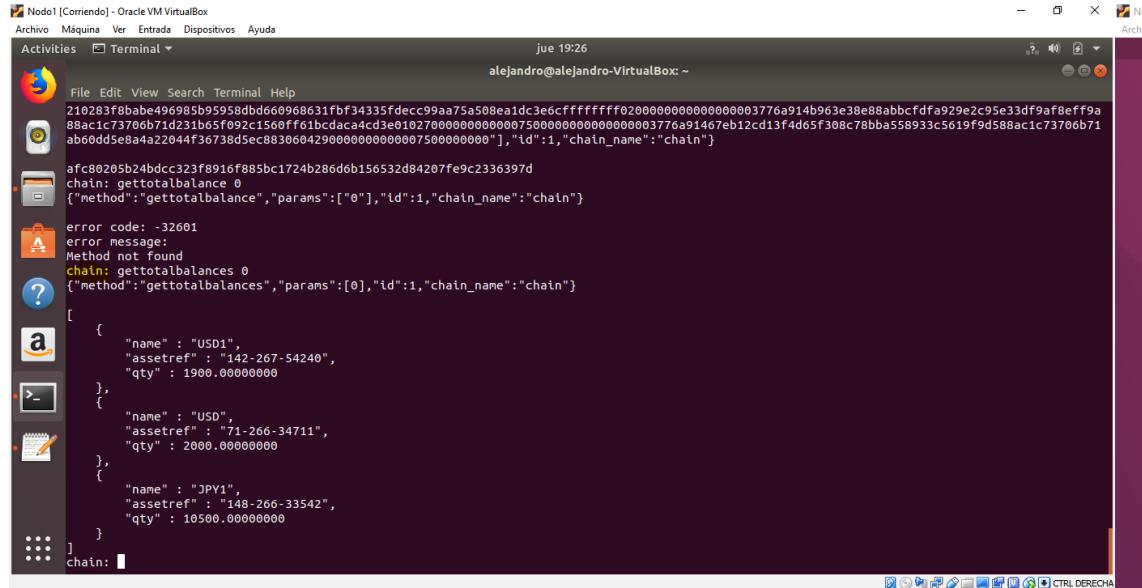
Ya solo queda hacer la comprobación de la transacción que se explicará en otro apartado.

## 4.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los casos de prueba.

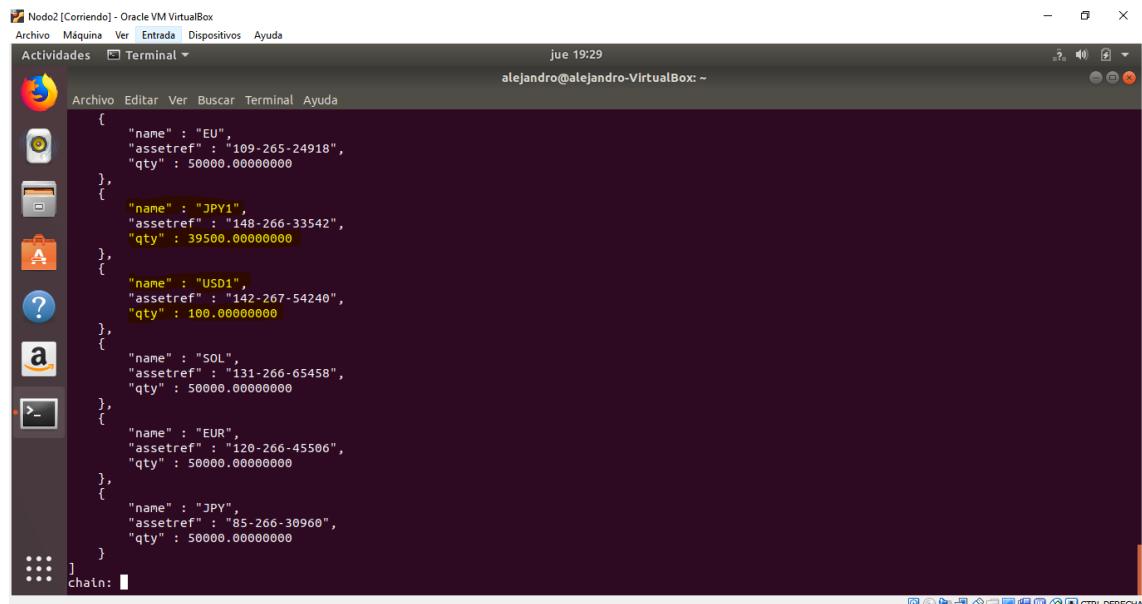
### 4.3.1 Comprobaciones

**Comprobación 1:** Utilizando “gettotalbalances 0” podemos comprobar en el nodo1 todas las monedas creadas para ambos nodos y la cantidad de cada una de ellas en qty:



```
Nodo1 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal jue 19:26
alejandro@alejandro-VirtualBox: ~
File Edit View Search Terminal Help
210283f6bab4496985b95958dbd660968631fb34335fdecc99aa75a508ea1dc3e6cfffffffff02000000000000000003776a914b963e38e88abcfdfa929e2c95e33df9af8eff9a
88ac1c73706b71d231b65f092c1560ff61bcdaca4cd3e01027000000000000750000000000000003776a91467eb12cd13f4d65f308c78bb558933c5619f9d588ac1c73706b7e
ab60dd55e84a2a20944f36738d5ec88306042900000000000007500000000
chain: gettotalbalance 0
{"method": "gettotalbalance", "params": ["0"], "id": 1, "chain_name": "chain"}
error code: -32601
error message:
Method not found
chain: gettotalbalances 0
{"method": "gettotalbalances", "params": [{}], "id": 1, "chain_name": "chain"}
[
  {
    "name": "USD1",
    "assetref": "142-267-54240",
    "qty": 1900.0000000
  },
  {
    "name": "USD",
    "assetref": "71-266-34711",
    "qty": 2000.0000000
  },
  {
    "name": "JPY1",
    "assetref": "148-266-33542",
    "qty": 10500.0000000
  }
]
chain: 
```

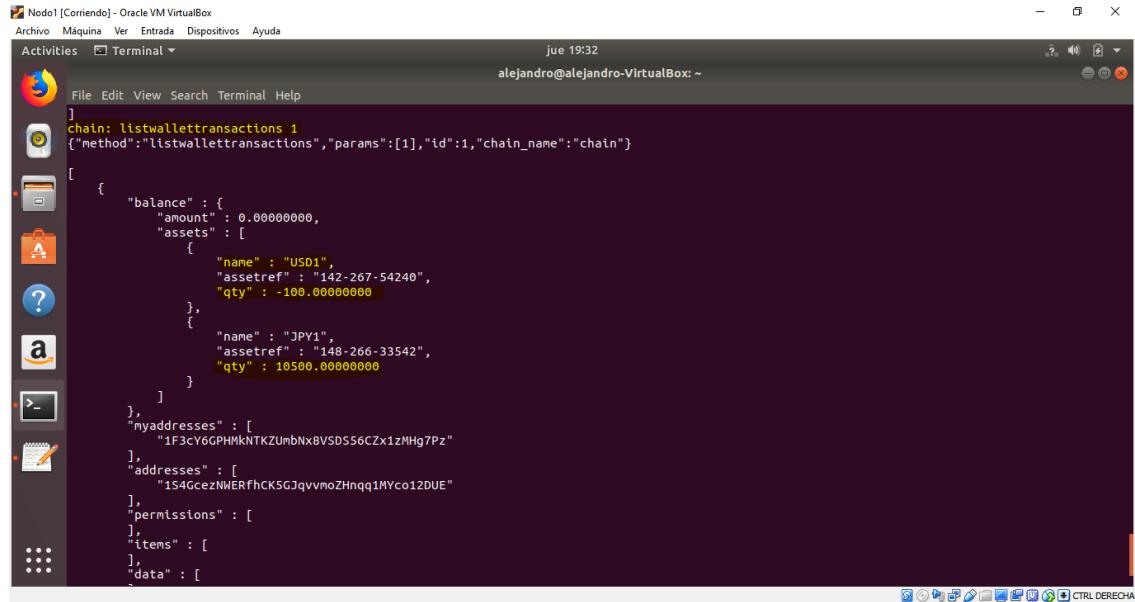
Se puede comprobar también en el nodo2 de la misma manera. Las dos monedas utilizadas en el ejemplo anterior son JPY1 y USD1 marcadas en amarillo:



```
Nodo2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal jue 19:29
alejandro@alejandro-VirtualBox: ~
File Edit Ver Buscar Terminal Ayuda
{
  "name": "EU",
  "assetref": "109-265-24918",
  "qty": 50000.0000000
},
{
  "name": "JPY1",
  "assetref": "148-266-33542",
  "qty": 39500.0000000
},
{
  "name": "USD1",
  "assetref": "142-267-54240",
  "qty": 100.0000000
},
{
  "name": "SOL",
  "assetref": "131-266-65458",
  "qty": 50000.0000000
},
{
  "name": "EUR",
  "assetref": "120-266-45506",
  "qty": 50000.0000000
},
{
  "name": "JPY",
  "assetref": "85-266-30960",
  "qty": 50000.0000000
}
]
chain: 
```

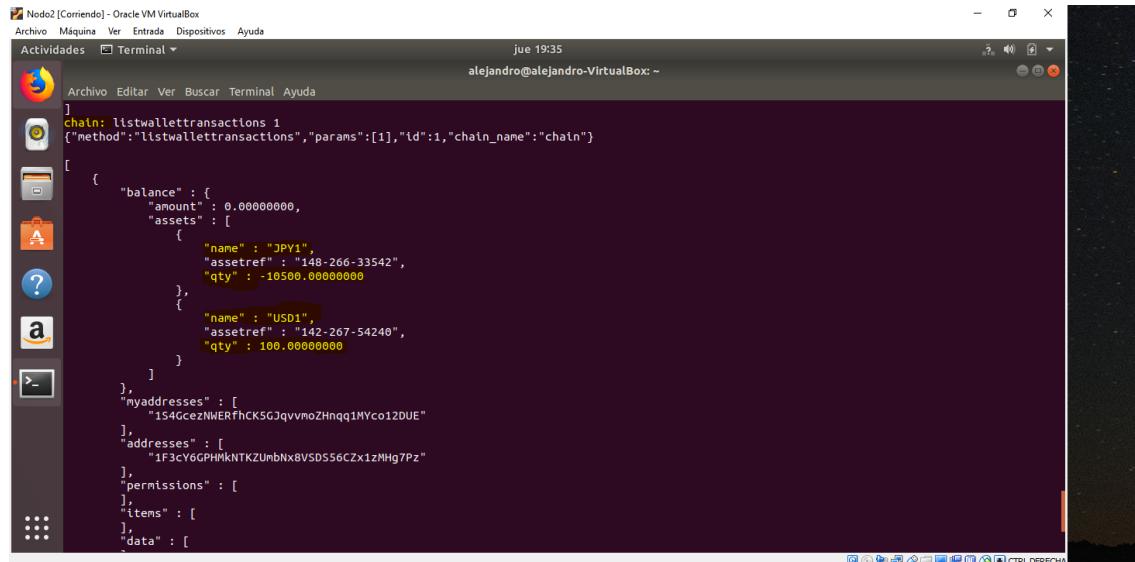
**Comprobación 2:** Utilizando “listwallettransactions 1” podemos ver las monedas con su cantidad pertenecientes explosivamente del nodo donde ejecutamos dicho comando.

Para nodo1:



```
chain: listwallettransactions 1
{"method": "listwallettransactions", "params": [1], "id": 1, "chain_name": "chain"}
[
  {
    "balance": {
      "amount": 0.00000000,
      "assets": [
        {
          "name": "USD1",
          "assetref": "142-267-54240",
          "qty": -100.00000000
        },
        {
          "name": "JPY1",
          "assetref": "148-266-33542",
          "qty": 10500.00000000
        }
      ]
    },
    "myaddresses": [
      "1F3cY6cPHMkNTKZUmbNx8VSDSS6CZx1zMHg7Pz"
    ],
    "addresses": [
      "1S4GcezNWERfhCK5GJqvvm0ZHnqq1MYco12DUE"
    ],
    "permissions": [
    ],
    "items": [
    ],
    "data": [
    ]
  }
]
```

Para nodo2:



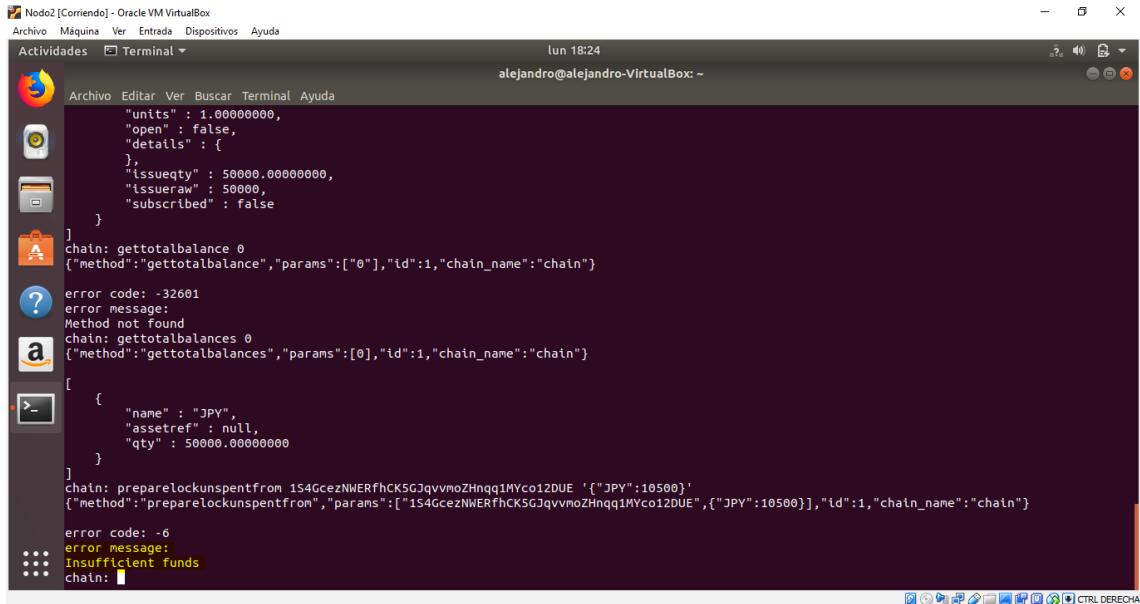
```
chain: listwallettransactions 1
{"method": "listwallettransactions", "params": [1], "id": 1, "chain_name": "chain"}
[
  {
    "balance": {
      "amount": 0.00000000,
      "assets": [
        {
          "name": "JPY1",
          "assetref": "148-266-33542",
          "qty": -10500.00000000
        },
        {
          "name": "USD1",
          "assetref": "142-267-54240",
          "qty": 100.00000000
        }
      ]
    },
    "myaddresses": [
      "1S4GcezNWERfhCK5GJqvvm0ZHnqq1MYco12DUE"
    ],
    "addresses": [
      "1F3cY6cPHMkNTKZUmbNx8VSDSS6CZx1zMHg7Pz"
    ],
    "permissions": [
    ],
    "items": [
    ],
    "data": [
    ]
  }
]
```

#### 4.3.2 Errores

**Error 1:** Al crear la salida de la transacción que contenga 10500 yenes aparece un mensaje de error “insufficient funds” o fondos insuficientes. Esto sucedió a que las JPY creadas no se habían creado bien inicialmente y por tanto no había monedas con las que realizar la transacción.

En caso de introducir una cantidad superior a la cantidad existente el error es el mismo. Se soluciona realizando la transacción con una cantidad inferior (dentro de las

existentes) o añadiendo la cantidad suficiente para poder realizar la transacción correctamente.



```
units : 1.00000000,
"open" : false,
"details" : {
},
"issueqty" : 50000.00000000,
"issueraw" : 50000,
"subscribed" : false
}
chain: gettotalbalance 0
{"method":"gettotalbalance","params":["0"],"id":1,"chain_name":"chain"}

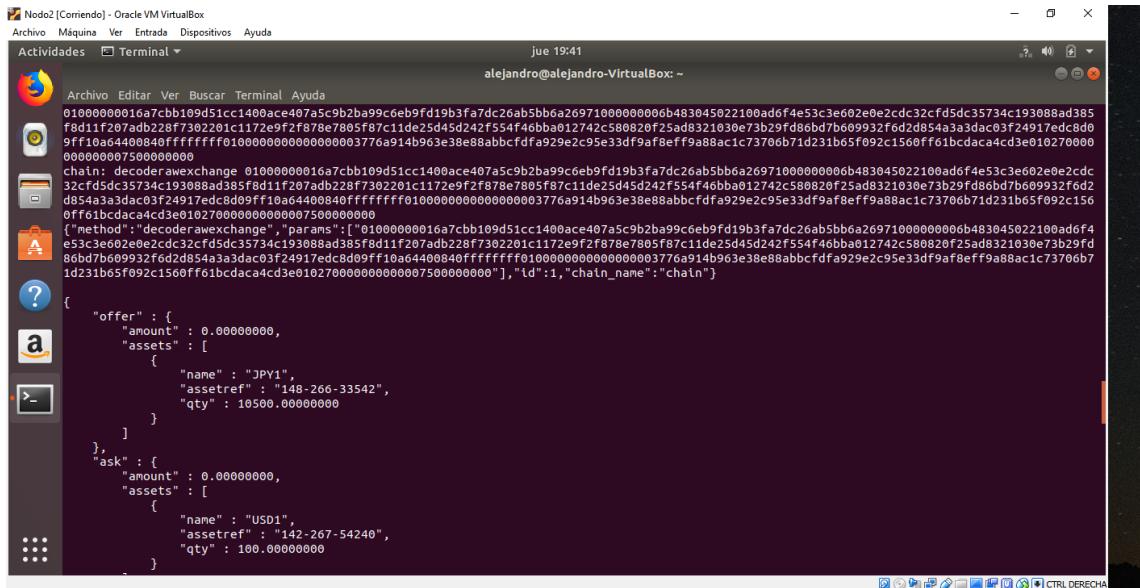
error code: -32601
error message:
Method not found
chain: gettotalbalances 0
{"method":"gettotalbalances","params":[0],"id":1,"chain_name":"chain"}

[
{
  "name" : "JPY",
  "assetref" : null,
  "qty" : 50000.00000000
}
]
chain: preparelockunspentfrom 154GcezNWERfhCK5GJqvvmozHnqq1MYco12DUE {"JPY":10500}
{"method":"preparelockunspentfrom","params":[{"154GcezNWERfhCK5GJqvvmozHnqq1MYco12DUE":{"JPY":10500}],"id":1,"chain_name":"chain"}

error code: -6
error message:
Insufficient funds
chain:
```

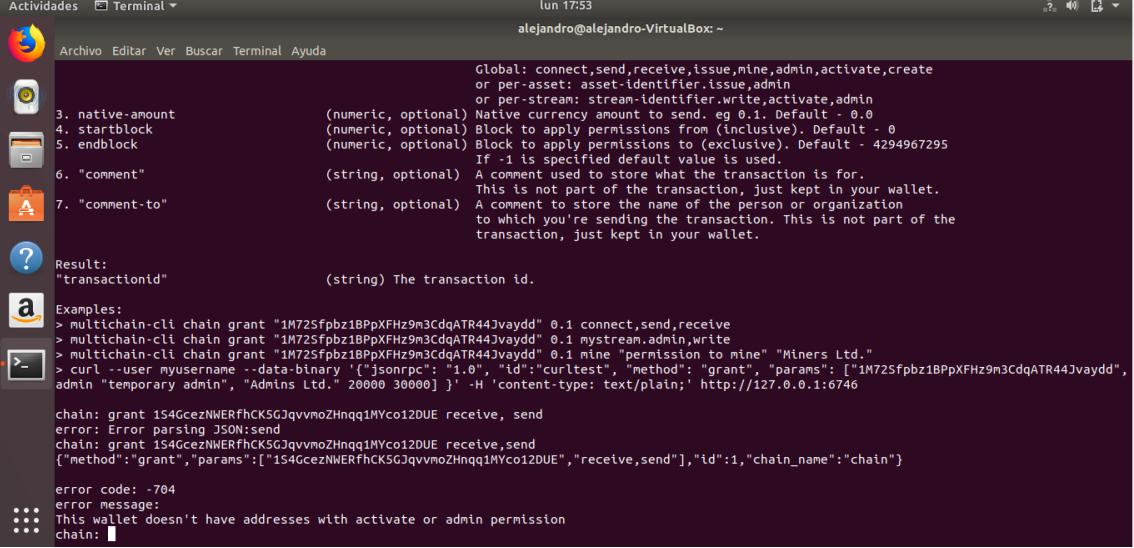
**Error 2:** Al ejecutar “createrwexchange [txid correspondiente] # ‘{“USD”:100}” en el nodo2 se genera un número hexadecimal que contiene los datos de transacción sin procesar que representa la oferta de intercambio. A continuación, hay que ejecutar “decoderawexchange [número hexadecimal]” en el nodo1, pero lo ejecutamos en el nodo2 y se produjo en el nodo1.

Por tanto, se produjo un error que solucionamos copiando el número hexadecimal en un blog de notas que tuvimos que pasar de un terminal a otro y ejecutando el comando decoderaweraxchange en el nodo1 como deberíamos haber hecho inicialmente.



```
offer" : {
  "amount" : 0.00000000,
  "assets" : [
    {
      "name" : "JPY1",
      "assetref" : "148-266-33542",
      "qty" : 10500.00000000
    }
  ],
  "ask" : {
    "amount" : 0.00000000,
    "assets" : [
      {
        "name" : "USD1",
        "assetref" : "142-267-54240",
        "qty" : 100.00000000
      }
    ]
  }
}, "id":1,"chain_name":"chain"]
```

**Error 3:** Realizamos desde el nodo1 la emisión de activos a la dirección del nodo2 de recibir y enviar sin otorgar los permisos previos al nodo1. Se produjo el error 704 y lo solucionamos otorgando los permisos necesarios al nodo1.



```

Actividades Terminal lun 17:53 alejandro@alejandro-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
Global: connect,send,receive,issue,mine,admin,activate,create
or per-asset: asset-identifier.issue,admin
or per-stream: stream-identifier.write,activate,admin
3. native-amount (numeric, optional) Native currency amount to send. eg 0.1. Default 0.0
4. startblock (numeric, optional) Block to apply permissions from (inclusive). Default - 0
5. endblock (numeric, optional) Block to apply permissions to (exclusive). Default - 4294967295
If -1 is specified default value is used.
6. "comment" (string, optional) A comment used to store what the transaction is for.
This is not part of the transaction, just kept in your wallet.
7. "comment-to" (string, optional) A comment to store the name of the person or organization
to which you're sending the transaction. This is not part of the
transaction, just kept in your wallet.

Result:
"transactionid" (string) The transaction id.

Examples:
> multichain-cli chain grant "1M72Sfpbz1BPPXFHz9m3CdqATR44Jvaydd" 0.1 connect,send,receive
> multichain-cli chain grant "1M72Sfpbz1BPPXFHz9m3CdqATR44Jvaydd" 0.1 mystream.admin,write
> multichain-cli chain grant "1M72Sfpbz1BPPXFHz9m3CdqATR44Jvaydd" 0.1 mine "permission to mine" "Miners Ltd."
> curl --user myusername --data-binary '[{"jsonrpc": "2.0", "id": "curltest", "method": "grant", "params": ["1M72Sfpbz1BPPXFHz9m3CdqATR44Jvaydd", "admin --temporary admin", "Admins Ltd.", "20000 30000] }]' -H 'content-type: text/plain;' http://127.0.0.1:6746

chain: grant 154GcezNWERfhCK5GjqvmoZHnqq1MYco12DUE receive, send
error: Error parsing JSON:send
chain: grant 154GcezNWERfhCK5GjqvmoZHnqq1MYco12DUE receive,send
{"method": "grant", "params": ["154GcezNWERfhCK5GjqvmoZHnqq1MYco12DUE", "receive,send"], "id": 1, "chain_name": "chain"}

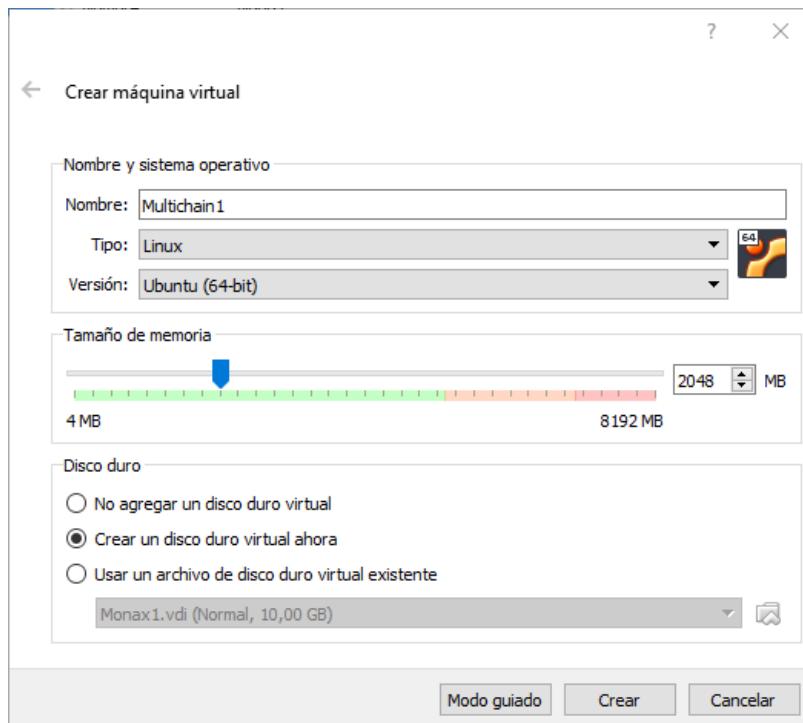
error code: -704
error message:
This wallet doesn't have addresses with activate or admin permission
chain: 
```

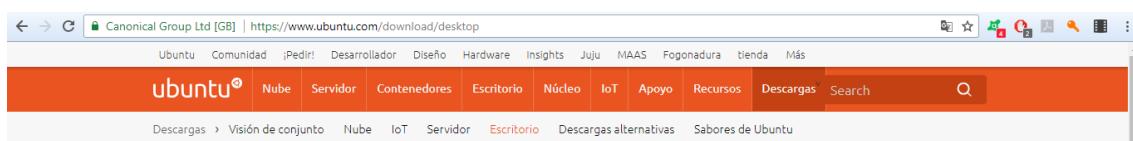
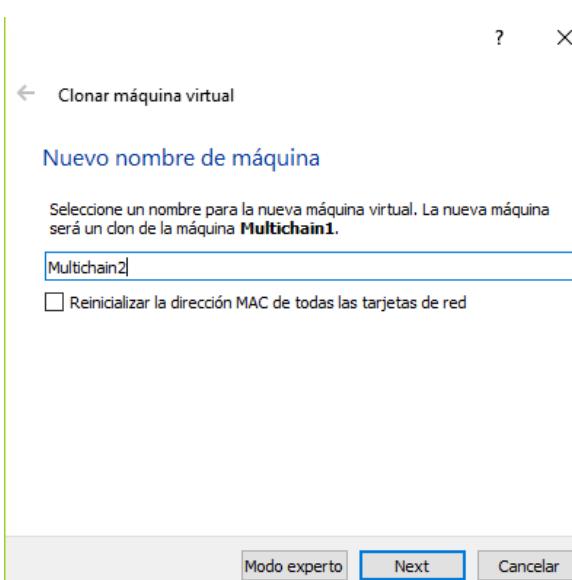
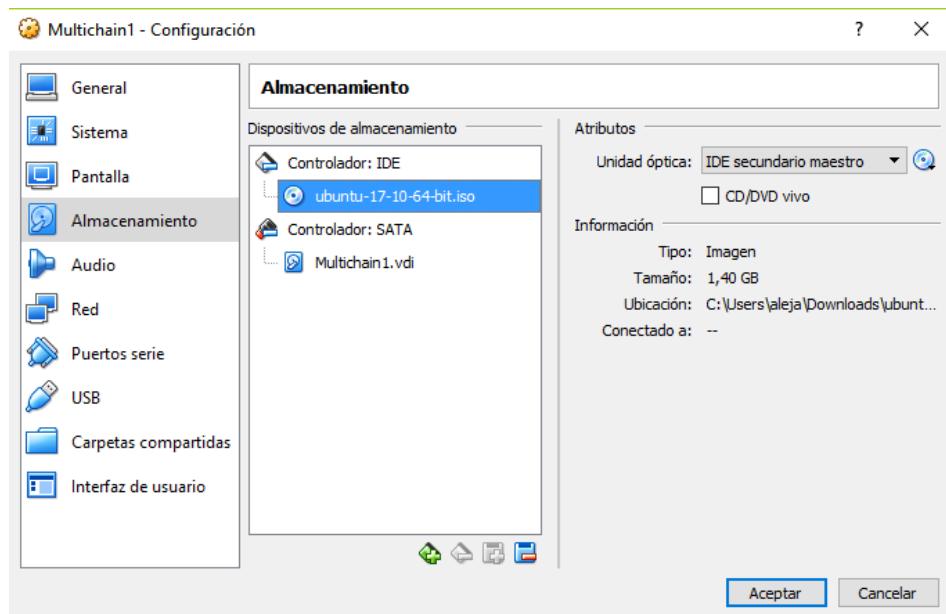
## 4.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

### 4.4.1 Proceso de instalación de las máquinas virtuales

Se le concede memoria a la máquina virtual, nombre, nombre, tipo de sistema operativo etc.





## Descargar Ubuntu Desktop

Ubuntu 18.04 LTS

Descargue la última versión LTS de Ubuntu, para PC de escritorio y portátiles. LTS significa soporte a largo plazo, lo que significa cinco años, hasta abril de 2023, de actualizaciones gratuitas de seguridad y mantenimiento, garantizadas.

[Notas de la versión de Ubuntu 18.04 LTS](#)

Requisitos del sistema recomendados:

- Procesador dual core de 2 GHz o mejor
- ...

Para otras versiones de Ubuntu Desktop, incluidos los torrents, el instalador de red, una lista de réplicas locales y versiones anteriores, vea nuestras [descargas alternativas](#).

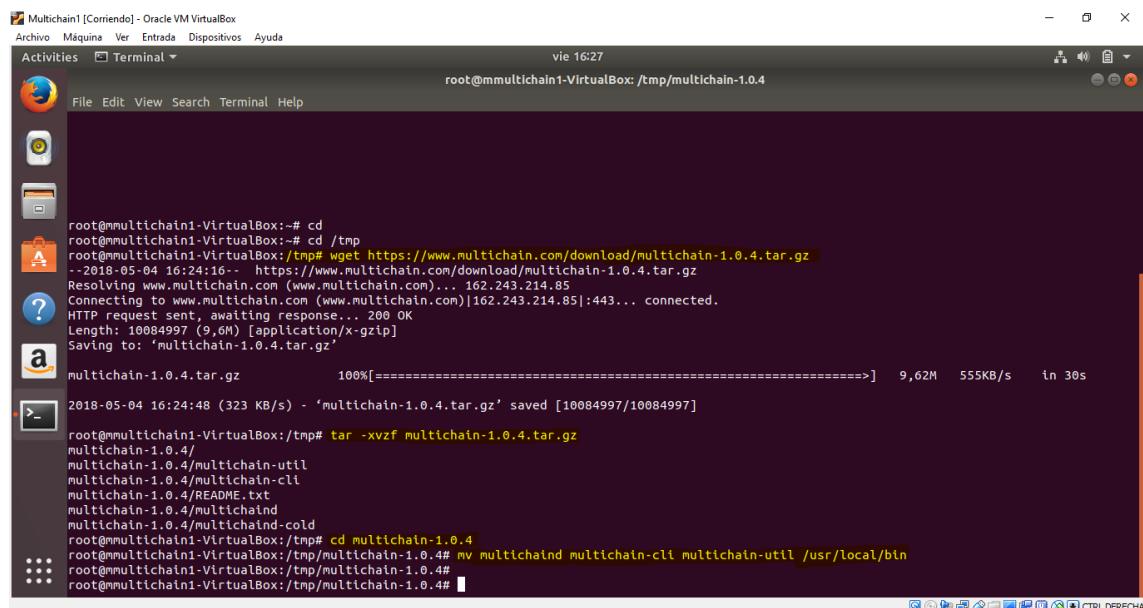
[Descargar](#)

#### 4.4.2 Instalación

En primer lugar, se descarga el programa en la propia página de MultiChain, dicho programa se descarga mediante comandos posteriormente escritos en el terminal de Linux.

Primero nos descargamos MultiChain con el primer comando subrayado en la siguiente imagen.

El segundo comando seleccionado sirve para descomprimir el archivo descargado, posteriormente nos movemos mediante cd a la carpeta descomprimida y, por último, movemos los archivos multichaind, multichain-cli y multichain-util a usr/local/bin para poder utilizarlos como comandos dentro del terminal Linux, cada uno tendrá una función determinada que será especificada posteriormente.



```
Multichain1 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal vie 16:27
root@mmultichain1-VirtualBox: /tmp/multichain-1.0.4

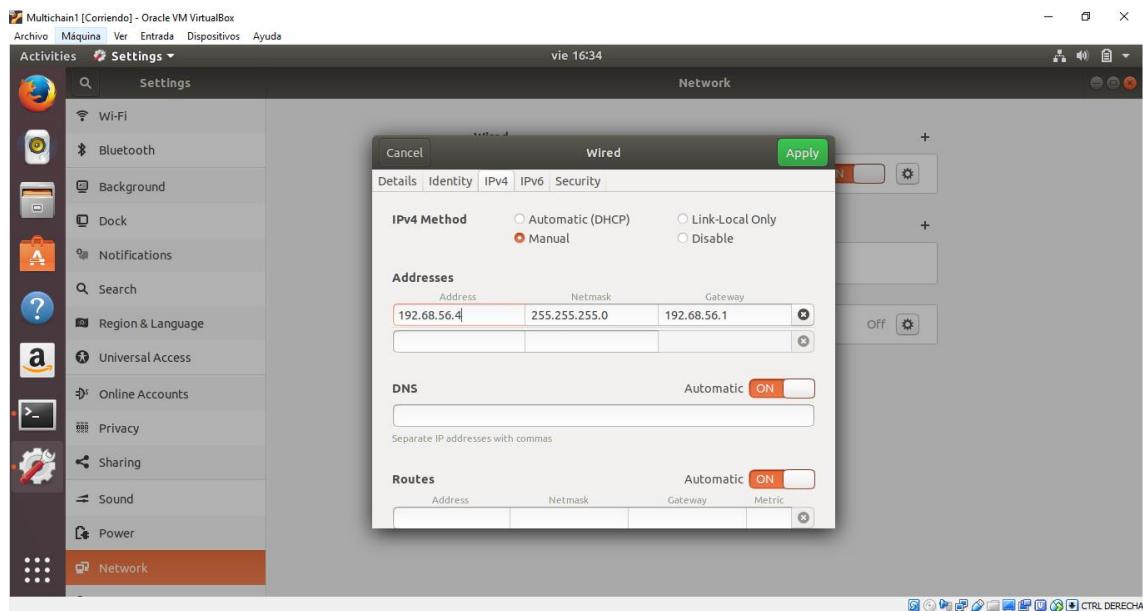
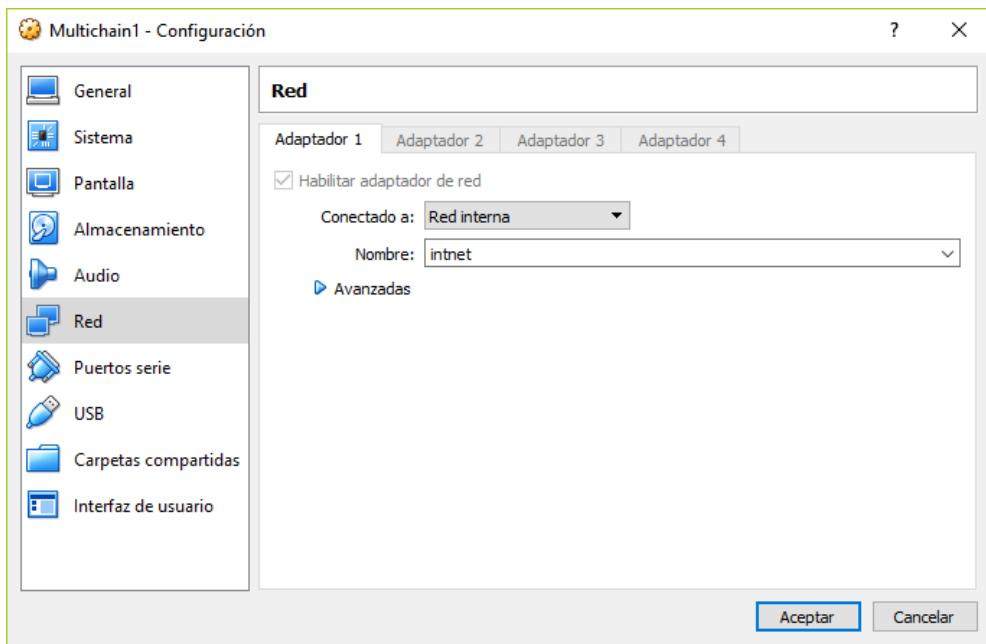
File Edit View Search Terminal Help

root@mmultichain1-VirtualBox:~# cd
root@mmultichain1-VirtualBox:~# cd /tmp
root@mmultichain1-VirtualBox:/tmp# wget https://www.multichain.com/download/multichain-1.0.4.tar.gz
--2018-05-04 16:24:16- https://www.multichain.com/download/multichain-1.0.4.tar.gz
Resolving www.multichain.com (www.multichain.com)... 162.243.214.85
Connecting to www.multichain.com (www.multichain.com)|162.243.214.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10084997 (9.6M) [application/x-gzip]
Saving to: 'multichain-1.0.4.tar.gz'

100%[=====] 9,62M 555KB/s in 30s
2018-05-04 16:24:48 (323 KB/s) - 'multichain-1.0.4.tar.gz' saved [10084997/10084997]

root@mmultichain1-VirtualBox:/tmp# tar -xvf multichain-1.0.4.tar.gz
multichain-1.0.4/
multichain-1.0.4/multichain-util
multichain-1.0.4/multichain-cli
multichain-1.0.4/README.txt
multichain-1.0.4/multichaind
multichain-1.0.4/multichaind-cold
root@mmultichain1-VirtualBox:/tmp# cd multichain-1.0.4
root@mmultichain1-VirtualBox:/tmp/multichain-1.0.4# mv multichaind multichain-cli multichain-util /usr/local/bin
root@mmultichain1-VirtualBox:/tmp/multichain-1.0.4#
```

Tras realizar esto, procedemos a conectar el nodo a la red interna para que se pueda ver con el nodo 2, para ello se dota al nodo de una ip como se muestra en la imagen:



Y comprobamos que la ip se ha cambiado correctamente:

```

mmultichain1@mmultichain1-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.68.56.4 netmask 255.255.255.0 broadcast 192.68.56.255
        inet6 fe80::cf9f:1066:4913:db3 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:d4:99:61 txqueuelen 1000 (Ethernet)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 111 bytes 11151 (11.1 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 694 bytes 43110 (43.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 694 bytes 43110 (43.1 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mmultichain1@mmultichain1-VirtualBox:~$

```

Utilizamos el comando multichain-util create chain\_test para crear la red blockchain:

```

Multichain1 [Corriendo] - Oracle VM VirtualBox
Archivo Maquina Ver Entrada Dispositivos Ayuda
Activities Terminal vie 16:43
mmultichain1@mmultichain1-VirtualBox: ~
File Edit View Search Terminal Help
mmultichain1@mmultichain1-VirtualBox:~$ multichain-util create chain_test
Multichain 1.0.4 Utilities (latest protocol 10010)

Blockchain parameter set was successfully generated.
You can edit it in /home/mmultichain1/.multichain/chain_test/params.dat before running multichaind for the first time.

To generate blockchain please run "multichaind chain_test -daemon".
mmultichain1@mmultichain1-VirtualBox:~$ 

```

A continuación, con el comando multichaind chain\_test -daemon iniciamos la BlockChain, nos proporciona un método para que otro nodo se pueda conectar a dicha BlockChain, el método se compone del comando multichaind la ip del nodo 1 y el puerto en el que hay que solicitar la conexión:

```
multichain1@multichain1-VirtualBox:~$ multichaind chain_test -daemon
MultiChain 1.0.4 Daemon (latest protocol 10010)
Starting up node...
Looking for genesis block...
Genesis block found
Other nodes can connect to this node using:
multichaind chain_test@192.68.56.4:7351
Listening for API requests on port 7350 (local only - see rpcallowip setting)
Node ready.
```

Se repiten todos los pasos anteriores en el nodo 2 a excepción de la creación y conexión de la red BlockChain.

En el nodo 2 se pone siguiente comando para conectar dicho nodo a la red creada anteriormente:

```
multichain2@multichain2-VirtualBox:~$ multichaind chain_test@192.68.56.4:7351
MultiChain 1.0.4 Daemon (latest protocol 10010)
Retrieving blockchain parameters from the seed node 192.68.56.4:7351 ...
Blockchain successfully initialized.
Please ask blockchain admin or user having activate permission to let you connect and/or transact:
multichain-cli chain_test grant 1CBEIAqe1zzjVRTimqEsDhElPww8KpWvueBf5 connect
multichain-ctl chain_test grant 1CBEIAqe1zzjVRTimqEsDhElPww8KpWvueBf5 connect,send,receive
multichain2@multichain2-VirtualBox:~$
```

En el nodo 1 se conceden permisos para poder conectar mediante el comando de la siguiente imagen:

```

Multichain1 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal vie 17:21
multichain1@multichain1-VirtualBox:~$ multichain-cli chain_test grant 1CBE1Aqe1ZzjVRT1mqEsDhElPww8KpWvueBf5 connect,send,receive
{"method": "grant", "params": ["1CBE1Aqe1ZzjVRT1mqEsDhElPww8KpWvueBf5", "connect,send,receive"], "id": 1, "chain_name": "chain_test"}
2bcdcec62c30e7fc3c14b92e5e5b3c62bb25008498feaeedba197097e12cf28
multichain1@multichain1-VirtualBox:~$ 

```

Y a continuación, con la información proporcionada por el comando anterior, en el nodo 2 se procede a copiarla para obtener los permisos e iniciar dicho nodo:

```

Multichain2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal vie 17:21
multichain2@multichain2-VirtualBox:~$ multichaind chain_test@192.68.56.4:7351
MultiChain 1.0.4 Daemon (latest protocol 10010)
Retrieving blockchain parameters from the seed node 192.68.56.4:7351 ...
Blockchain successfully initialized.
Please ask blockchain admin or user having activate permission to let you connect and/or transact:
multichain-cli chain_test grant 1CBE1Aqe1ZzjVRT1mqEsDhElPww8KpWvueBf5 connect
multichain-cli chain_test grant 1CBE1Aqe1ZzjVRT1mqEsDhElPww8KpWvueBf5 connect,send,receive
multichain2@multichain2-VirtualBox:~$ multichaind chain_test@192.68.56.4:7351 -daemon
MultiChain 1.0.4 Daemon (latest protocol 10010)
Starting up node...
Retrieving blockchain parameters from the seed node 192.68.56.4:7351 ...
Other nodes can connect to this node using:
multichain chain_test@192.68.56.5:7351
Listening for API requests on port 7350 (local only - see rpcallowip setting)
Node ready.
multichain2@multichain2-VirtualBox:~$ 

```

## 4.5 Manual de usuario

Esta tecnología no está preparada para ser empleada por un usuario que no tenga conocimientos técnicos.

## 5. Proyecto de implementación de un prototipo del sistema utilizando la Monax

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

### 5.1 Documentación de diseño

Diagrama de funcionamiento de Monax



### 5.2 Documentación de construcción

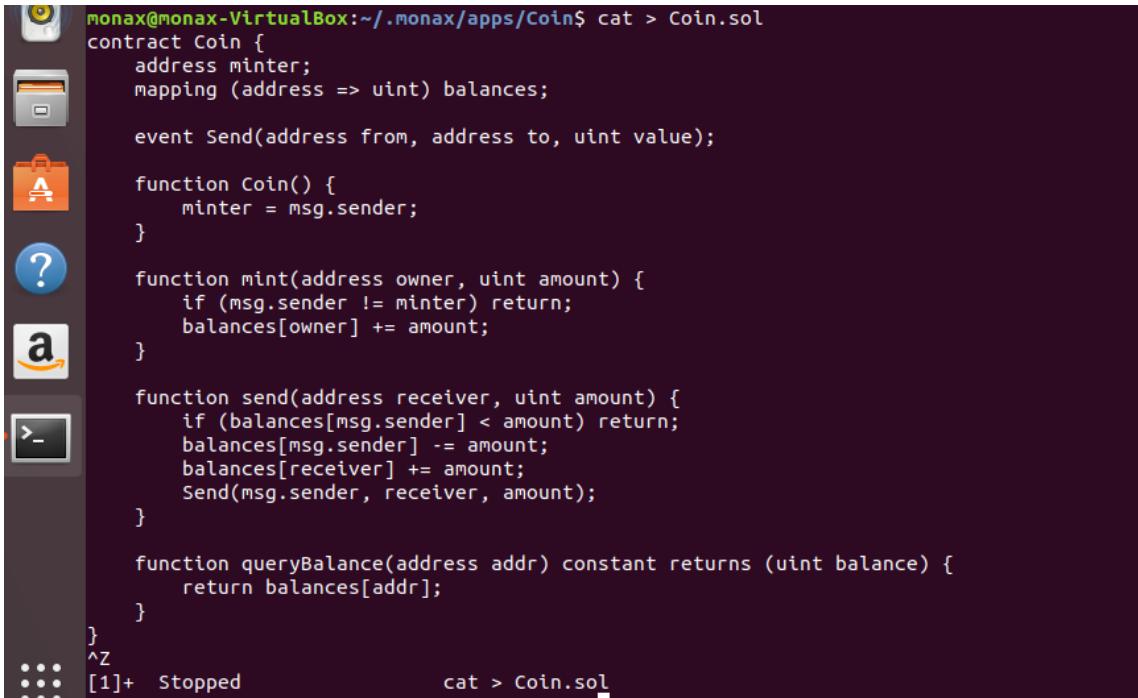
A continuación, explicamos paso a paso la creación de la transacción

A screenshot of a terminal window from a Linux-like environment. The window title is 'Terminal'. The command history shows the following steps:

```
monax@monax-VirtualBox:~/monax$ cd ~/monax/apps
monax@monax-VirtualBox:~/monax/apps$ mkdir Coin
monax@monax-VirtualBox:~/monax/apps$ cd Coin
monax@monax-VirtualBox:~/monax/apps/Coin$
```

The terminal window has a dark background with light-colored text. A small orange icon with a white letter 'A' is visible on the left side of the window.

Se crea el archivo correspondiente:



```

monax@monax-VirtualBox:~/monax/apps/Coin$ cat > Coin.sol
contract Coin {
    address minter;
    mapping (address => uint) balances;

    event Send(address from, address to, uint value);

    function Coin() {
        minter = msg.sender;
    }

    function mint(address owner, uint amount) {
        if (msg.sender != minter) return;
        balances[owner] += amount;
    }

    function send(address receiver, uint amount) {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        Send(msg.sender, receiver, amount);
    }

    function queryBalance(address addr) constant returns (uint balance) {
        return balances[addr];
    }
}
^Z
[1]+  Stopped                  cat > Coin.sol

```



```

monax@monax-VirtualBox:~/monax/apps/Coin$ cat > emp.yaml
jobs:
- name: sendTx
  job:
    send:
      destination: 58FD1799AA32DED3F6EAC096A1DC77834A446B9C
      amount: 1234
^Z
[2]+  Stopped                  cat > emp.yaml

```

Para poder seleccionar la adress que utilizará la transacción:



```

Monax [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal sáb 11:46
monax@monax-VirtualBox:~/monax/apps/Coin
monax@monax-VirtualBox:~/monax/apps/Coin$ chain_dir=$HOME/.monax/chains/test_chain
monax@monax-VirtualBox:~/monax/apps/Coin$ addr=$(cat $chain_dir/addresses.csv | grep test_chain_full_000 | cut -d ',' -f 1)

```

Se crea otra red BlockChain llamada receiver\_chain para que interactue con la inicial, siendo creada como la inicial.

Se ejecuta el contrato entre las dos chains:



```
Monax [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Terminal Help
monax@monax-VirtualBox:~/monax/apps/Coin$ /home/monax/monax pkgs do -c test_chain -a $addr
*****Executing Job*****
Job Name defaultAddr
*****Executing Job*****
Job Name sendCoin
Transaction Hash 6C0A8117135684A14507E53C84FBA927EF983CD4
*****Executing Job*****
Job Name deployCoin
Saving ABI abi/Coin
Deploying Contract name=Coin
addr=0C34B1EB647EA9E5971FF915E43B08B3FAAA4183
```

## 5.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los casos de prueba.

### 5.3.1 Comprobaciones

**Comprobación 1:** Comprobamos que el nodo se ha conectado bien al Docker

```
monax@monax-VirtualBox:~$ docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

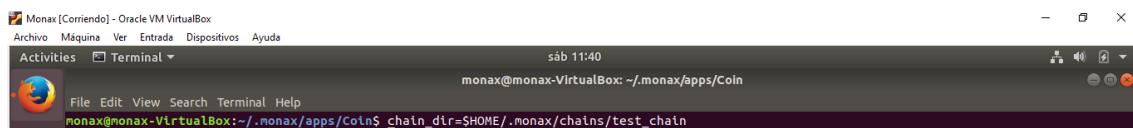
For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
```

**Comprobación 2:** Comprobamos que el Docker funciona correctamente y muestra el cliente y el servidor

```
monax@monax-VirtualBox:~$ docker version
Client:
  Version:      18.03.1-ce
  API version:  1.37
  Go version:   go1.9.5
  Git commit:   9ee9f40
  Built:        Thu Apr 26 07:17:38 2018
  OS/Arch:      linux/amd64
  Experimental: false
  Orchestrator: swarm

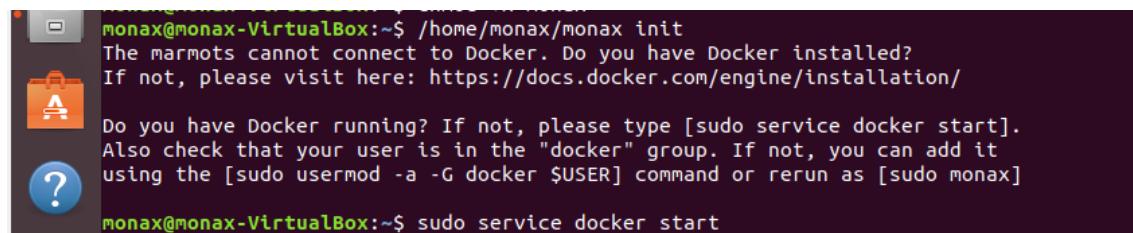
Server:
  Engine:
    Version:      18.03.1-ce
    API version:  1.37 (minimum version 1.12)
    Go version:   go1.9.5
    Git commit:   9ee9f40
    Built:        Thu Apr 26 07:15:45 2018
    OS/Arch:      linux/amd64
    Experimental: false
```

**Comprobación 3:** Comprobamos que encuentra el archivo o directorio y que hemos concedido los permisos correctamente.

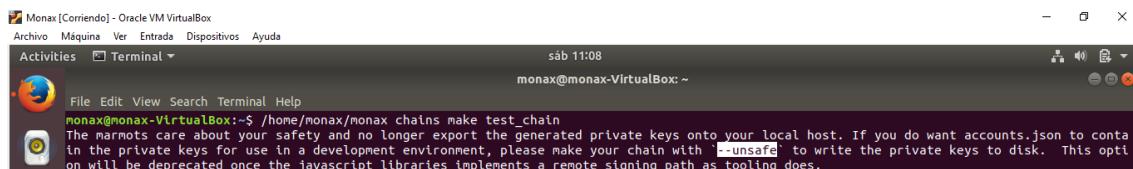


### 5.3.2 Errores

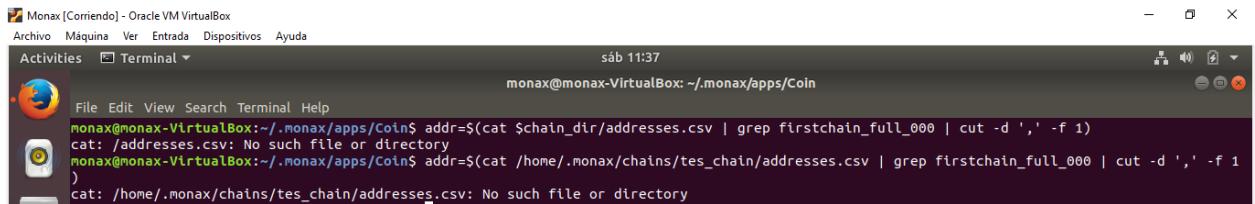
**Error 1:** Al intentar conectarse al Docker se produce un error ya que nos descargamos una versión de Docker no válida.



**Error 2:** Se produce un error a la hora de crear la cadena y encontramos otro comando que nos solucionaba este error, que se producía por la que no existía la carpeta en la que se producía la blockchain.

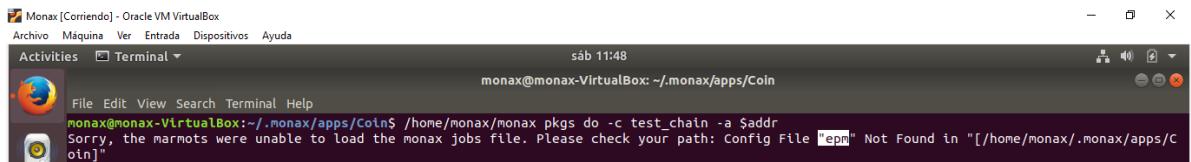


**Error 3:** Se produce un error al no encontrar el archivo directorio y no haber concedido los permisos correspondientes



```
Monax [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal sáb 11:37
monax@monax-VirtualBox: ~/monax/apps/Coin
File Edit View Terminal Help
monax@monax-VirtualBox:~/monax/apps/Coin$ addr=$(cat $chain_dir/addresses.csv | grep firstchain_full_000 | cut -d ',' -f 1)
cat: /addresses.csv: No such file or directory
monax@monax-VirtualBox:~/monax/apps/Coin$ addr=$(cat /home/.monax/chains/tes_chain/addresses.csv | grep firstchain_full_000 | cut -d ',' -f 1)
cat: /home/.monax/chains/tes_chain/addresses.csv: No such file or directory
```

**Error 4:** Se produce un error al escribir el nombre del archivo y para resolverlo comprobamos el nombre y resolvemos el error.



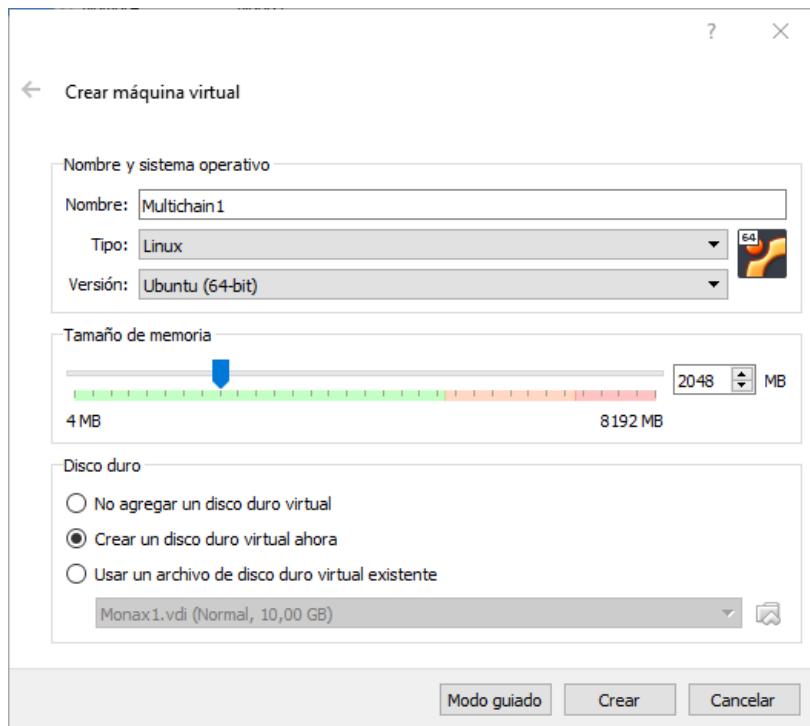
```
Monax [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal sáb 11:48
monax@monax-VirtualBox: ~/monax/apps/Coin
File Edit View Terminal Help
monax@monax-VirtualBox:~/monax/apps/Coin$ ./home/monax/monax pkgs do -c test_chain -a $addr
Sorry, the marmots were unable to load the monax jobs file. Please check your path: Config File "epm" Not Found in "[/home/monax/.monax/apps/Coin]"
```

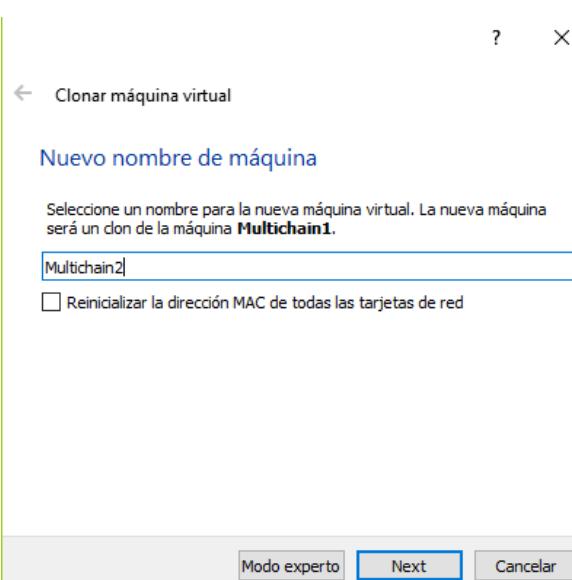
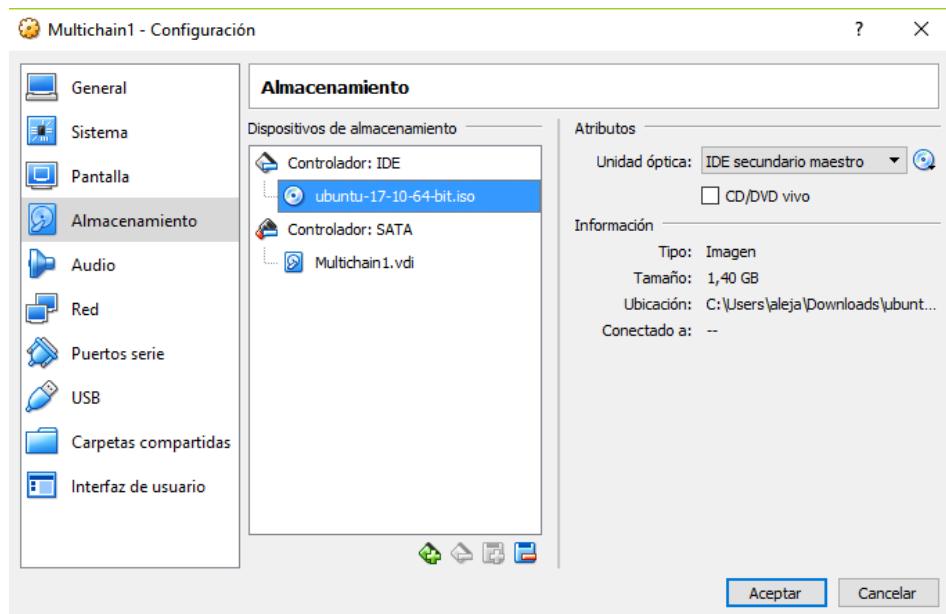
## 5.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

### 5.4.1 Proceso de instalación de las máquinas virtuales

Se le concede memoria a la máquina virtual, nombre, tipo de sistema operativo etc.





## Descargar Ubuntu Desktop

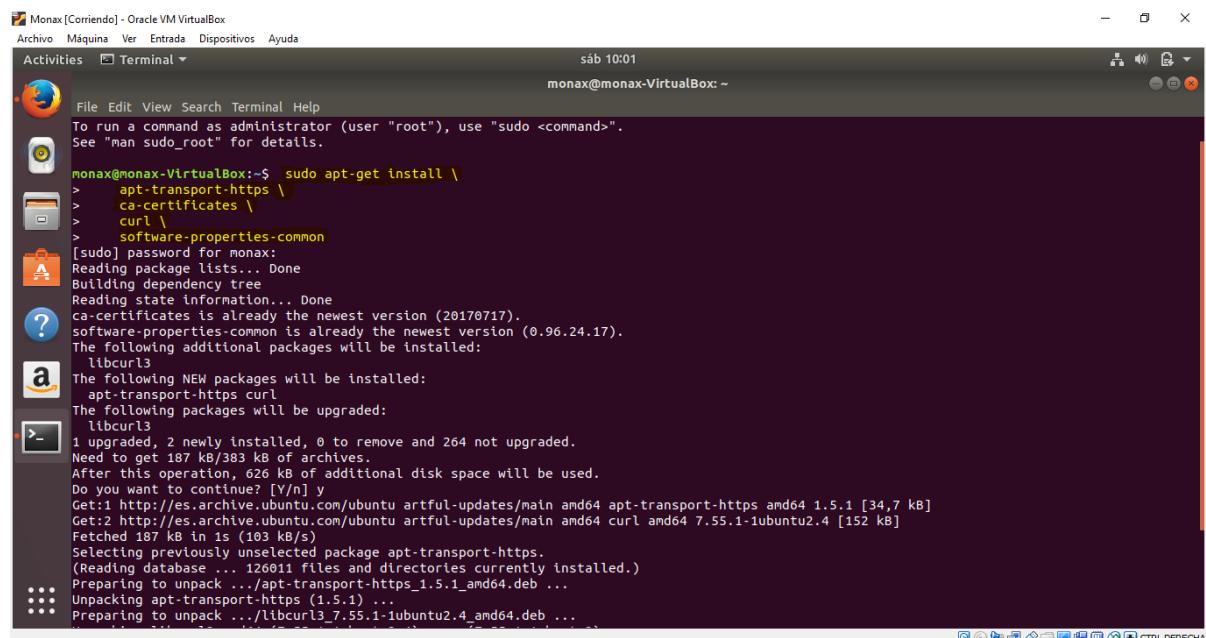
The screenshot shows the Ubuntu 18.04 LTS download page. The title is 'Ubuntu 18.04 LTS'. A paragraph describes it as the long-term support version for desktop and laptop PCs, with five years of security and maintenance updates. A green 'Descargar' (Download) button is prominent. Below it, a note says: 'Para otras versiones de Ubuntu Desktop, incluidos los torrents, el instalador de red, una lista de réplicas locales y versiones anteriores, vea nuestras descargas alternativas.' A link 'Notas de la versión de Ubuntu 18.04 LTS' is also present. A section titled 'Requisitos del sistema recomendados:' lists system requirements, with the first one checked: 'Procesador dual core de 2 GHz o mejor' (Dual-core processor at 2 GHz or better).

## 5.4.2 Instalación de Docker y Monax

En primer lugar, es necesario instalar el Docker:

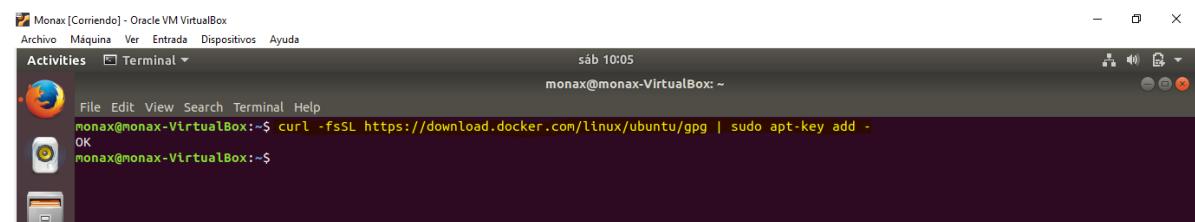
Para descargar Docker entramos en la página: <https://www.docker.com> y descargamos el archivo correspondiente.

Mediante los siguientes comandos en la consola Linux se procede a la preparación del equipo para instalar el Docker:

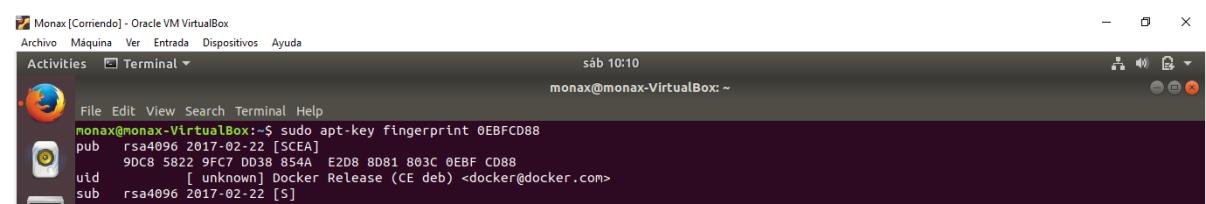


```
monax@monax-VirtualBox:~$ sudo apt-get install \apt-transport-https \ca-certificates \curl \software-properties-common
[sudo] password for monax:
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20170717).
software-properties-common is already the newest version (0.96.24.17).
The following additional packages will be installed:
libcurl3
The following NEW packages will be installed:
  apt-transport-https curl
The following packages will be upgraded:
  libcurl3
1 upgraded, 2 newly installed, 0 to remove and 264 not upgraded.
Need to get 187 kB/383 kB of archives.
After this operation, 626 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://es.archive.ubuntu.com/ubuntu artful-updates/main amd64 apt-transport-https amd64 1.5.1 [34,7 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu artful-updates/main amd64 curl amd64 7.55.1-1ubuntu2.4 [152 kB]
Fetched 187 kB in 1s (103 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 126011 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.5.1_amd64.deb ...
Unpacking apt-transport-https (1.5.1) ...
Preparing to unpack .../libcurl3_7.55.1-1ubuntu2.4_amd64.deb ...

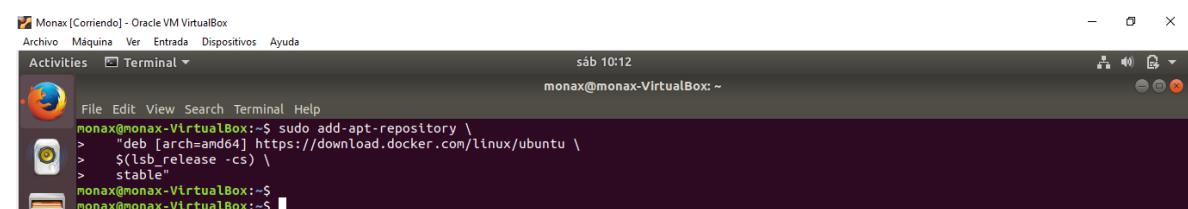
```



```
monax@monax-VirtualBox:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
monax@monax-VirtualBox:~$
```

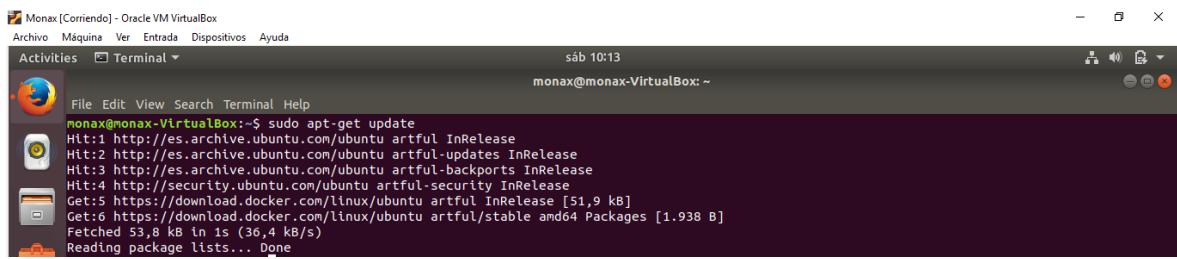


```
monax@monax-VirtualBox:~$ sudo apt-key fingerprint 0EBFCD88
pub   rsa496 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD3B 854A  E2D8 BD81 803C 0EBF CD88
uid            [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```



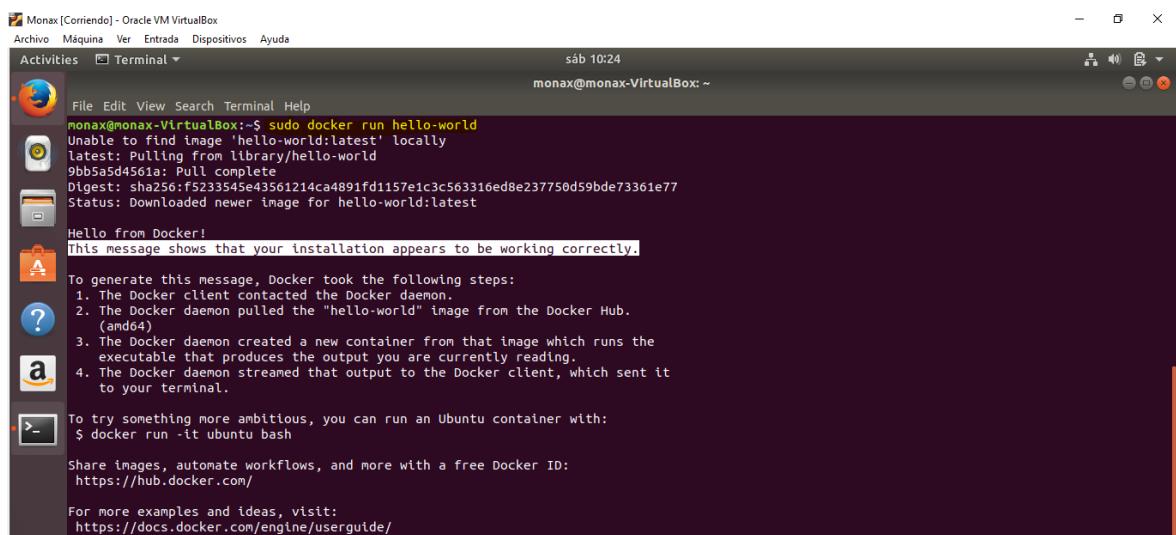
```
monax@monax-VirtualBox:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
monax@monax-VirtualBox:~$
```

A continuación, las siguientes imágenes muestran la instalación:



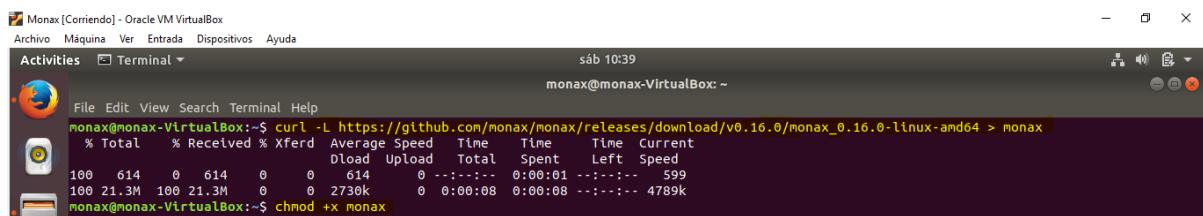
```
sáb 10:13  
monax@monax-VirtualBox:~  
monax@monax-VirtualBox:~$ sudo apt-get update  
Hit:1 http://es.archive.ubuntu.com/ubuntu artful InRelease  
Hit:2 http://es.archive.ubuntu.com/ubuntu artful-updates InRelease  
Hit:3 http://es.archive.ubuntu.com/ubuntu artful-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu artful-security InRelease  
Get:5 https://download.docker.com/linux/ubuntu artful/stable amd64 Packages [1.938 B]  
Fetched 53,8 kB in 1s (36,4 kB/s)  
Reading package lists... Done
```

Con el siguiente comando se especifica la versión a descargar:



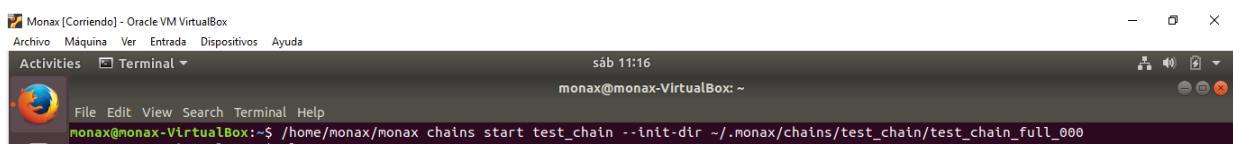
```
sáb 10:24  
monax@monax-VirtualBox:~  
monax@monax-VirtualBox:~$ sudo docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
9bb5a5d4561a: Pull complete  
Digest: sha256:f5233545e43561214ca4891fd1157e1c3c563316ed8e237750d59bde73361e77  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)  
3. The Docker daemon created a new container from that image which runs the  
executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/engine/userguide/
```

Tras instalar el Docker, se continua con la instalación de Monax:



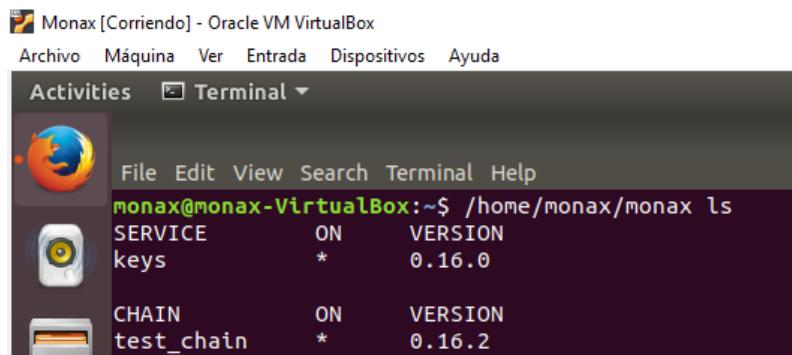
```
sáb 10:39  
monax@monax-VirtualBox:~  
monax@monax-VirtualBox:~$ curl -L https://github.com/monax/monax/releases/download/v0.16.0/monax_0.16.0-linux-amd64 > monax  
% Total % Received % Xferd Average Speed Time Time Current  
          Dload Upload Total Spent Left Speed  
100 614    0 614      0      0 0:00:01 --:--:-- 599  
100 21.3M 100 21.3M 0      0 2730k 0 0:00:08 0:00:08 --:--:-- 4789k  
monax@monax-VirtualBox:~$ chmod +x monax
```

Se crea la cadena de prueba:



```
sáb 11:16  
monax@monax-VirtualBox:~  
monax@monax-VirtualBox:~$ /home/monax/monax chains start test_chain --init-dir ~/.monax/chains/test_chain/test_chain_full_000
```

A continuación, se lista con el comando monax ls para comprobar que está activa:



A screenshot of a Linux desktop environment, specifically Monax running in Oracle VM VirtualBox. The desktop has a dark theme with icons for Firefox, a speaker, and a terminal. A terminal window is open, showing the command:

```
monax@monax-VirtualBox:~$ /home/monax/monax ls
SERVICE      ON      VERSION
keys          *       0.16.0
CHAIN         ON      VERSION
test_chain    *       0.16.2
```

## 5.5 Manual de usuario

Esta tecnología no está preparada para ser empleada por un usuario que no tenga conocimientos técnicos.

## 6. Comparación de las dos implementaciones

Se trata de dar valores a los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos.

### 6.1 Evaluación de los criterios en la implementación usando MultiChain

Se incluye una tabla donde se puede observar los criterios de evaluación que se han utilizado, su evaluación y comentarios para su mejor entendimiento.

CRITERIO	EVALUACIÓN
<b>Criterio 1:</b> Documentación	7 documentos
<b>Criterio 2:</b> Horas empleadas	80h
<b>Criterio 3:</b> Líneas de código	0 líneas → No hay que programar
<b>Criterio 4:</b> Facilidad de implementación	Si
<b>Criterio 5:</b> Velocidad de arranque	Tarda en arrancar.
<b>Criterio 6:</b> Velocidad de funcionamiento	A diferencia de la velocidad de arranque, la velocidad de respuestas por pantalla es mucho mayor.
<b>Criterio 7:</b> Cantidad de nodos	2 nodos
<b>Criterio 8:</b> Caída de los nodos	3 veces
<b>Criterio 9:</b> Soporte	4

### 6.2 Evaluación de los criterios en la implementación usando Monax

Se incluye una tabla donde se puede observar los criterios de evaluación que se han utilizado, su evaluación y comentarios para su mejor entendimiento.

CRITERIO	EVALUACIÓN
<b>Criterio 1:</b> Documentación	2 documentos
<b>Criterio 2:</b> Horas empleadas	120h
<b>Criterio 3:</b> Líneas de código	40 líneas
<b>Criterio 4:</b> Facilidad de implementación	No
<b>Criterio 5:</b> Velocidad de arranque	Tarda en iniciar el sistema y conectarse a los nodos.
<b>Criterio 6:</b> Velocidad de funcionamiento	Velocidad de respuestas por pantalla mayor a la de arranque.
<b>Criterio 7:</b> Cantidad de nodos	1 nodo
<b>Criterio 8:</b> Caída de los nodos	4 veces
<b>Criterio 9:</b> Soporte	1

## **7. Comparación de la implementación de las tecnologías**

Se incluye una tabla resumen donde cruzamos los criterios y los valores de cada tecnología estudiada. También se añade a dicha tabla una columna de comentarios sobre la comparación.

CRITERIOS	MULTICHAIN	MONAX	COMENTARIOS
<b>Criterio 1:</b> Documentación	7 documentos	3 documentos	Hay más documentación de MultiChain que de Monax, por lo que es más fácil poder solucionar más dudas y fallos que surjan en MultiChain.
<b>Criterio 2:</b> Horas empleadas	80h	120h	Hay una diferencia de 40 horas entre ambas, esto se debe a que para Monax hay una escasa documentación que dificulta poder continuar si tenemos un fallo.
<b>Criterio 3:</b> Líneas de código	0 líneas → No hay que programar	40 líneas	MultiChain nos da la facilidad de no programar el Smart Contract a diferencia de Monax.
<b>Criterio 4:</b> Facilidad de implementación	Sí	No	MultiChain tiene facilidad de implementación ya que si surge un error es más fácil saber cuál es el error y poder solucionarlo, en cambio Monax es muy poco intuitivo.
<b>Criterio 5:</b> Velocidad de arranque	Tarda en arrancar.	Tarda en iniciar el sistema y conectarse a los nodos.	Ambas tardan en iniciar el sistema y conectarse a los nodos.
<b>Criterio 6:</b> Velocidad de funcionamiento	A diferencia de la velocidad de arranque, la velocidad de respuestas por pantalla mayor a la de arranque.	Velocidad de respuestas por pantalla mayor a la de arranque.	Ambas tienen una velocidad de respuesta muy parecida.
<b>Criterio 7:</b> Cantidad de nodos	2 nodos	1 nodo	Monax al usar solo 1 nodo tiene un menor tiempo de implementación, en cambio en MultiChain se tienen que implementar 2 nodos.
<b>Criterio 8:</b> Caída de los nodos	3 veces	4 veces	Los nodos de MultiChain se han caído 3 veces a diferencia de Monax que se han caído 4 veces.
<b>Criterio 9:</b> Soporte	4	1	MultiChain cuenta con un mejor soporte que Monax. MultiChain tiene un foro de preguntas y en el caso de Monax tienen un chat de preguntas, MultiChain tiene un tiempo de respuesta máximo de 1 día y te intentan solventar fallos y dudas, en cambio Monax, además de tener que hacer las preguntas en inglés, tardan en responder o no responden directamente.

## 8. Conclusiones

A partir de lo expuesto en apartados anteriores y tras la realización de las aplicaciones con ambas tecnologías concluimos que esta tecnología es difícil de implementar si no eres experto.

Tras utilizar las dos plataformas y realizar la comparación, concluimos con que la plataforma MultiChain es más intuitiva, además de estar mejor documentada que Monax. También proporciona mayores facilidades en cuanto a instalación e implementación. Como ventaja destacable de Monax es que no se necesita nada más que un nodo para la realización de la red blockchain.

Cabe destacar que Multichain tiene integrados los Smart Contracts en sus comandos por lo que para realizar una transacción sólo hay que poner un comando y el Smart Contract que tiene asociado ese comando, en una capa más interna, se ejecutará y se producirá la transacción.

Por último, cabe destacar que MultiChain ofrece un soporte mayor al que ofrece Monax.

## ANEXO 1: Otros Requisitos

### A.1. MultiChain

#### A.1.1. Requisitos Funcionales

REQ.	DESCRIPCIÓN
RF01	Conectado a: Red interna
RF02	Se debe llevar a cabo una transacción entre dos nodos o más

#### A.1.2. Requisitos NO Funcionales

REQ.	DESCRIPCIÓN
R01	Se utilizará la versión 1.0.4 de MultiChain
R02	La implementación de la BlockChain y el desarrollo de la transacción han de llevarse a cabo en un tiempo inferior a 80 horas.

### A.2 Monax

#### A.2.1. Requisitos Funcionales

REQ.	DESCRIPCIÓN
RF01	Instalación del Docker
RF02	Conectado a: NAT
RF03	Se debe poder llevar a cabo una transacción en un solo nodo

#### A.2.2. Requisitos NO Funcionales

REQ.	DESCRIPCIÓN
R01	Se utilizará la versión 0.16.0 de Monax
R02	La implementación de la BlockChain y el desarrollo de la transacción han de llevarse a cabo en un tiempo inferior a 120 horas.