



# REDES P2P CON BLOCKCHAIN

ANDRÉS DAVID LUNA PEÑA

CHRISTIAN TENORIO

JUAN AGREDO

JOSE VELASQUES



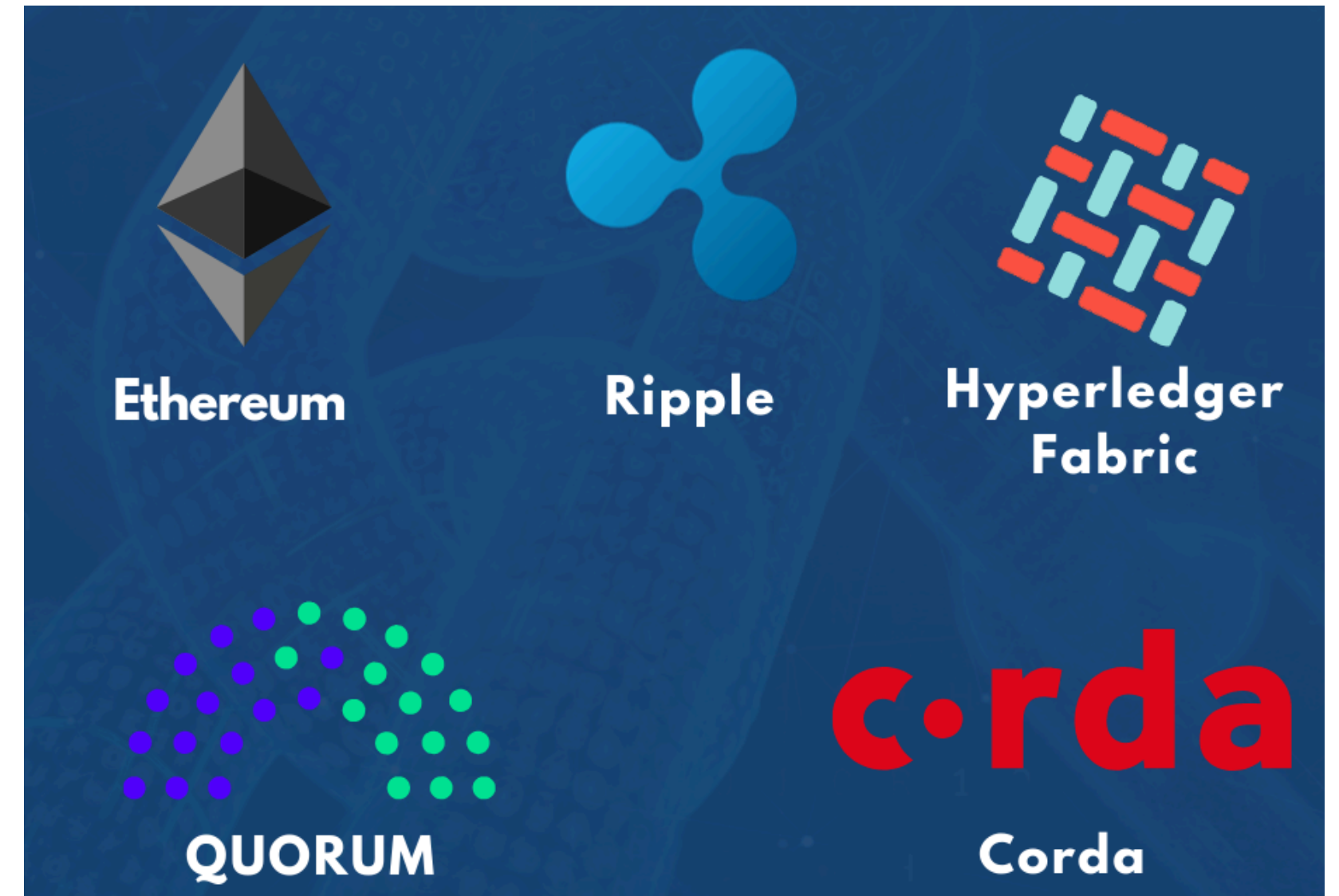
# ¿QUÉ ES PEER TO PEER?

El Peer-to-Peer (P2P) es una red descentralizada en la que dispositivos, llamados nodos o pares, se conectan directamente entre sí para compartir recursos como archivos, datos o servicios sin depender de un servidor central. Esto se aplica en compartir archivos, transacciones financieras en tecnologías blockchain y servicios como Uber o Airbnb. La legalidad varía según el contenido compartido, y existen tres tipos principales de redes P2P: híbridas, estructuradas y no estructuradas, cada una con sus propias características y usos específicos.



# P2P EN BLOCKCHAIN

En blockchain, el P2P permite que la información se comparta directamente entre usuarios sin intermediarios. La blockchain es un registro compartido y verificado por todos los participantes, lo que elimina la necesidad de servidores centrales. Cualquier usuario puede ayudar a validar transacciones convirtiendo su dispositivo en un nodo de la red, lo que refuerza la seguridad al detectar y rechazar actividades fraudulentas.



# DESASTRES NATURALES Y BLOCKCHAIN

- OPORTUNIDAD EN ALERTAMIENTOS
- VERACIDAD DE LA INFORMACIÓN
- COORDINACIÓN DE ESFUERZOS
- TRANSPARENCIA EN APORTACIONES

## ANTES: MITIGACIÓN DEL RIESGO

AUTORIDAD: **PRONÓSTICOS** CON "BIG DATA"  
PERSONA: **IDENTIDAD DIGITAL**  
COMUNIDAD: **"CROWDSOURCING"** DE RIESGOS

## DURANTE: ALERTA TEMPRANA

AUTORIDAD: **ALERTAS ABIERTAS**  
PERSONA: **SEÑAL DE EMERGENCIA**  
COMUNIDAD: **"CROWDSOURCING"**  
DE INCIDENTES

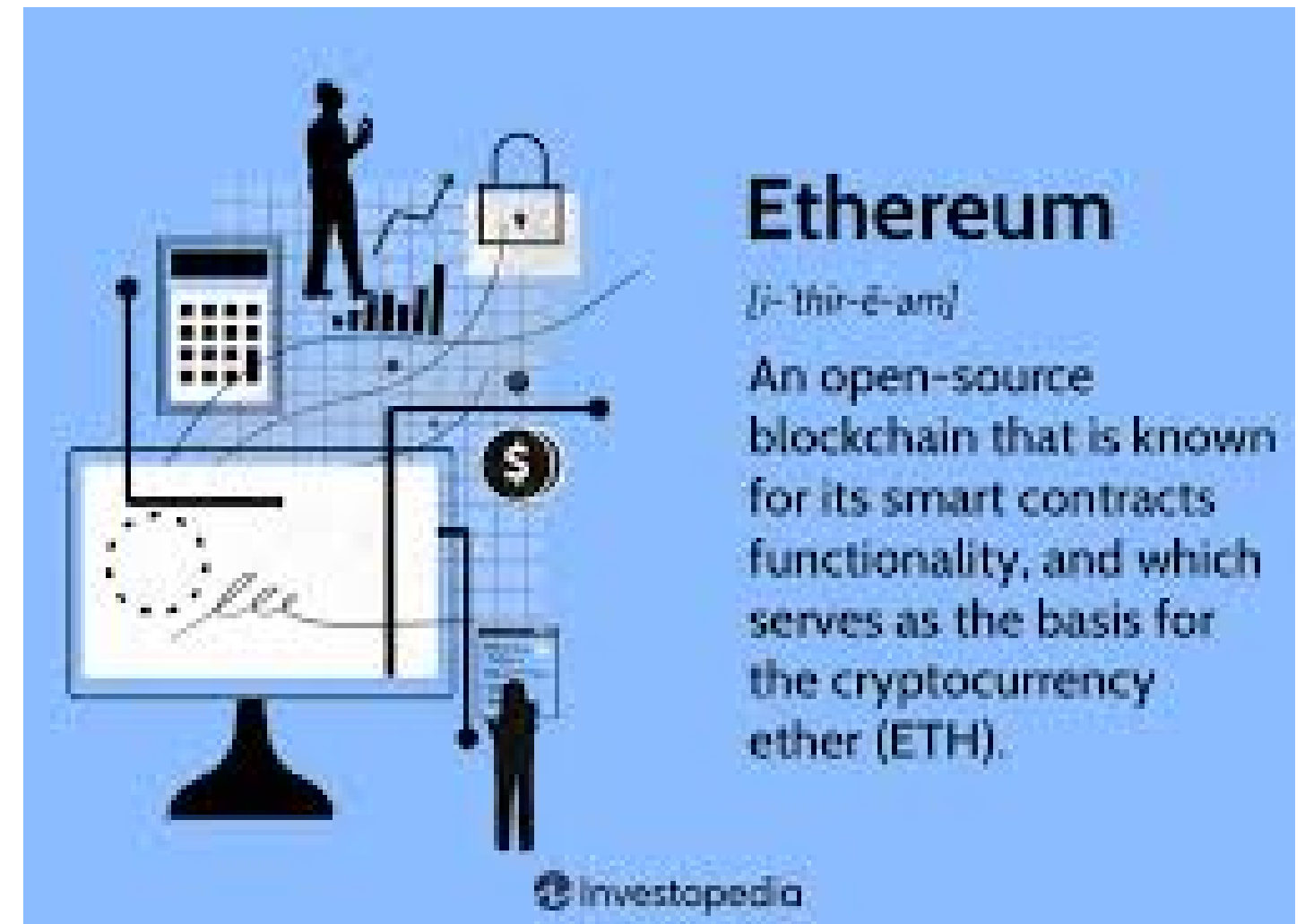
# DESASTRES NATURALES Y BLOCKCHAIN

**DESPUÉS: AYUDA HUMANITARIA**  
AUTORIDAD: **COMUNICADOS OFICIALES**  
PERSONA: SOLICITUDES **GENUINAS**  
COMUNIDAD: APORTACIONES  
**TRANSPARENTES**

Blockchain permite construir una fuente común de información confiable para coordinar una respuesta efectiva ante un desastre natural.

# PROPUESTA DE IMPLEMENTACIÓN

Instalar 3 nodos del cliente Geth de Ethereum para una red peer-to-peer usando el algoritmo de Ethereum. A lo largo del proceso, utilizaremos comandos básicos para configurar e interactuar con la red, permitiendo una integración de la propuesta que se buscará implementar.



# PROCESOS

1

Descargar el aplicativo de ethereum para Ubuntu a 64 bits.

**sudo apt-get update**

**sudo apt-get install software-properties-common**

**sudo apt-get install -y ethereum**

**sudo add-apt-repository -y ppa:ethereum/ethereum**

# PROCESOS

2

Revisamos la correcta instalación

**geth version**

3

Creamos el directorio, donde será nuestro espacio de trabajo y el directorio donde se almacenará los datos del nodo.

**mkdir ethereum-local.net**

**cd ethereum-local.net**



# PROCESOS

4

Creamos el archivo .json

```
{
  "config": {
    "chainId": 4777,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0
  },
  "difficulty" : "2",
  "gasLimit" : "2100000",
  "alloc" :{
    "0xf189ec0a4A111Df655cA4c11e5D340f820BD07F5": {
      "balance": "10000000"}
  }
}
```

5

Creamos la carpeta node-1 y si mismo el node-1

**mkdir node-1**

**geth -datadir node-1 account new**

# PROCESOS

6

Editamos el archivo genesis.json cambiando la llave publica del nodo 1

```
{
  "config": {
    "chainId": 4777,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0
  },
  "difficulty" : "2",
  "gasLimit" : "2100000",
  "alloc" :{
    "0xf189ec0a4A111Df655cA4c11e5D340f820BD07F5": {
      "balance": "10000000"
    }
  }
}
```

5

Actulizamos el nodo 1 con el archivo genesis.json

**geth --identity node-1 init genesis.json --datadir node-1**

# PROCESOS

6

Iniciamos el nodo 1

**geth --datadir node-1 --maxpeers 2 --networkid 13 --http -- http.addr  
"0.0.0.0" --http-api 8545 -- port 30301 --allow-insecure-unclock**

7

Hacemos el cambio de maquina para ingresar el usuario 1 en el nodo 1

De esta misma forma creamos los diferentes nodos.

**geth attach /home/usuario/desktop/ethereum-local.net/node-1/geth.ipc**

# RED P2P

```
vagrant@socorro: ~/blockcha  X + v
INFO [05-23|23:35:34.487] Looking for peers
INFO [05-23|23:35:44.591] Looking for peers
INFO [05-23|23:35:54.697] Looking for peers
INFO [05-23|23:36:05.741] Looking for peers
INFO [05-23|23:36:16.297] Looking for peers
INFO [05-23|23:36:26.542] Looking for peers
INFO [05-23|23:36:36.893] Looking for peers
INFO [05-23|23:36:47.672] Looking for peers
INFO [05-23|23:36:58.555] Looking for peers
INFO [05-23|23:37:08.944] Looking for peers
INFO [05-23|23:37:21.800] Looking for peers
INFO [05-23|23:37:31.975] Looking for peers
INFO [05-23|23:37:42.082] Looking for peers
INFO [05-23|23:37:52.219] Looking for peers
WARN [05-23|23:37:58.013] Post-merge network, but no beacon client seen. Please launch one to follow the chain!
INFO [05-23|23:38:02.325] Looking for peers
INFO [05-23|23:38:12.497] Looking for peers
INFO [05-23|23:38:22.987] Looking for peers
INFO [05-23|23:38:33.013] Looking for peers
INFO [05-23|23:38:47.080] Looking for peers
INFO [05-23|23:38:57.290] Looking for peers
INFO [05-23|23:39:07.315] Looking for peers
INFO [05-23|23:39:17.367] Looking for peers
INFO [05-23|23:39:27.505] Looking for peers
INFO [05-23|23:39:37.960] Looking for peers
INFO [05-23|23:39:47.977] Looking for peers
INFO [05-23|23:39:58.086] Looking for peers
INFO [05-23|23:40:08.191] Looking for peers
INFO [05-23|23:40:19.307] Looking for peers
INFO [05-23|23:40:29.367] Looking for peers
INFO [05-23|23:40:39.741] Looking for peers
INFO [05-23|23:40:50.159] Looking for peers
INFO [05-23|23:41:00.261] Looking for peers
INFO [05-23|23:41:10.307] Looking for peers

peercount=0 tried=24 static=0
peercount=0 tried=14 static=0
peercount=0 tried=6 static=0
peercount=0 tried=9 static=0
peercount=0 tried=9 static=0
peercount=0 tried=5 static=0
peercount=0 tried=4 static=0
peercount=0 tried=5 static=0
peercount=0 tried=6 static=0
peercount=0 tried=8 static=0
peercount=0 tried=13 static=0
peercount=0 tried=5 static=0
peercount=0 tried=42 static=0
peercount=0 tried=48 static=0
peercount=0 tried=40 static=0
peercount=0 tried=0 static=0
peercount=0 tried=5 static=0
peercount=0 tried=2 static=0
peercount=0 tried=4 static=0
peercount=0 tried=17 static=0
peercount=0 tried=40 static=0
peercount=0 tried=58 static=0
peercount=0 tried=8 static=0
peercount=0 tried=15 static=0
peercount=0 tried=20 static=0
peercount=0 tried=7 static=0
peercount=0 tried=8 static=0
peercount=0 tried=5 static=0
peercount=0 tried=35 static=0
peercount=0 tried=44 static=0
peercount=0 tried=8 static=0
peercount=0 tried=29 static=0
peercount=0 tried=6 static=0

Activar Windows
Ve a Configuración para activar Windows.
```

# GENESIS.JSON

```
{
  "config": {
    "chainId": 1337,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0
  },
  "gasLimit": "0x1c9c380",
  "difficulty": "0x020000",
  "alloc": {
    "0x3ac35Fa9477E2fA87d8f1D240DBaE514F5a30f3B": {
      "balance": "10000000"
    }
  }
}
```



# LANZAMIENTO GETH

```
vagrant@servidor:~/ethereum-nocal.net$ geth attach /home/vagrant/ethereum-nocal.net/node1/geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.13.15-stable-c5ba367e/linux-amd64/go1.21.6
at block: 0 (Thu Jan 01 1970 00:00:00 GMT+0000 (UTC))
  datadir: /home/vagrant/ethereum-nocal.net/node1
 modules: admin:1.0 debug:1.0 engine:1.0 eth:1.0 miner:1.0 net:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> eth.getBalance(eth.accounts[0])
0
> Connection to 127.0.0.1 closed by remote host.
```

# LANZAMIENTO GETH

fatal: invalid command line: too many arguments

```
vagrant@socorro:~$ geth attach /home/vagrant/blockchain_nodes/node1/geth.ipc
```

Welcome to the Geth JavaScript console!

instance: Geth/v1.14.3-stable-ab48ba42/linux-amd64/go1.22.3

at block: 0 (Thu Jan 01 1970 00:00:00 GMT+0000 (UTC))

datadir: /home/vagrant/blockchain\_nodes/node1

modules: admin:1.0 debug:1.0 engine:1.0 eth:1.0 miner:1.0 net:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit

```
> eth.getBalance(eth.accounts[0])
```

Error: invalid address

at inputAddressFormatter (web3.js:3955:11(39))

at web3.js:5060:37(8)

at map (native)

at web3.js:5059:35(12)

at web3.js:5085:34(15)

at send (web3.js:5110:39(11))

at <eval>:1:15(6)

```
> |
```

# GETH VERSION

```
INFO [05-23|23:26:06.629] Successfully wrote genesis state      database=lightchaindata hash=29495a
..5afb92
vagrant@servidor:~/ethereum-nocal.net$ geth --datadir node1 --maxpeers 2 --networkid 1337 --http --htt
p.addr "0.0.0.0" --port 30301 --allow-insecure-unlock
INFO [05-23|23:26:09.765] Maximum peer count                     ETH=2 total=2
INFO [05-23|23:26:09.768] Smartcard socket not found, disabling  err="stat /run/pcscd/pcscd.comm: no
such file or directory"
WARN [05-23|23:26:09.771] Sanitizing cache to Go's GC limits     provided=1024 updated=654
INFO [05-23|23:26:09.772] Set global gas cap                    cap=50,000,000
INFO [05-23|23:26:09.772] Initializing the KZG library           backend=gokzg
INFO [05-23|23:26:09.921] Allocated trie memory caches          clean=98.00MiB dirty=163.00MiB
INFO [05-23|23:26:09.921] Using pebble as the backing database
INFO [05-23|23:26:09.921] Allocated cache and file handles      database=/home/vagrant/ethereum-noc
al.net/node1/geth/chaindata cache=327.00MiB handles=524,288
INFO [05-23|23:26:09.933] Opened ancient database               database=/home/vagrant/ethereum-noc
al.net/node1/geth/chaindata/ancient/chain readonly=false
INFO [05-23|23:26:09.933] State scheme set to already existing  scheme=hash
Fatal: Failed to register the Ethereum service: ethash is only supported as a historical component of
already merged networks
vagrant@servidor:~/ethereum-nocal.net$ sudo vim genesis.json
vagrant@servidor:~/ethereum-nocal.net$ ^C
vagrant@servidor:~/ethereum-nocal.net$ geth version
Geth
Version: 1.13.15-stable
Git Commit: c5ba367eb6232e3eddd7d6226bfd374449c63164
```

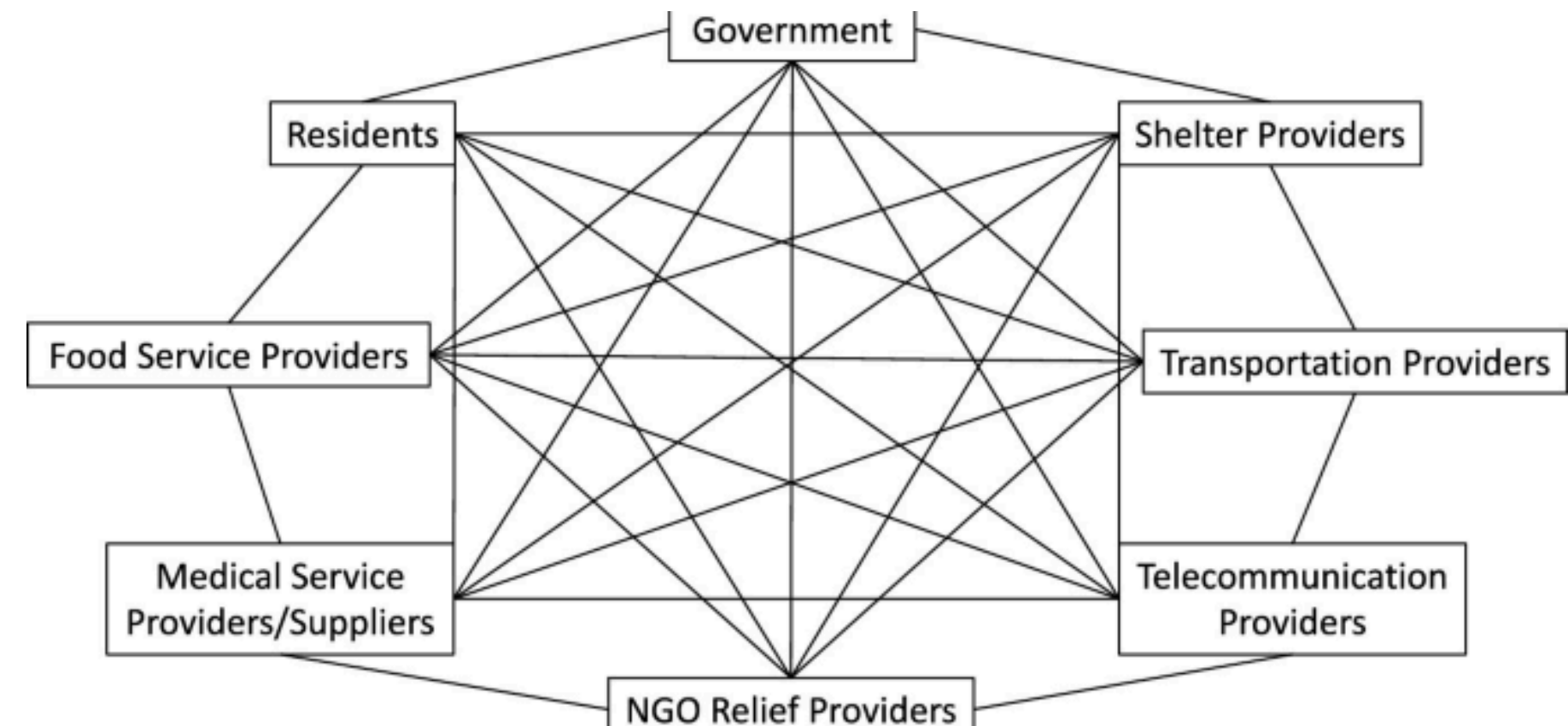


# CONCLUSIONES



- Las redes peer-to-peer (P2P) en combinación con la tecnología blockchain pueden ser extremadamente útiles para la gestión de procesos durante desastres naturales. Su arquitectura descentralizada garantiza la disponibilidad y la integridad de los datos, permitiendo una rápida recuperación y continuidad de las operaciones críticas sin depender de un servidor central, lo que es esencial en situaciones de emergencia.

- La implementación de la red blockchain no se pudo completar satisfactoriamente debido a problemas en la transferencia de datos entre nodos. La plataforma no funcionó correctamente con nuestra base de datos existente, lo que impidió realizar transacciones de manera efectiva y aseguró la integridad de la red.





¡MUCHAS  
GRACIAS!





# BIBLIOGRAFÍA



[HTTPS://WWW.CRYPTOPOLITAN.COM/ES/PEER-TO-PEER-EN-BLOCKCHAIN-COMO-FUNCIONA/](https://www.cryptopolitan.com/es/peer-to-peer-en-blockchain-como-funciona/)

[HTTPS://EDSP.PROXYUAO.ELOGIM.COM/EDS/PDFVIEWER/PDFVIEWER?  
VID=10&SID=DB36B8C7-3273-42CC-8077-83E951889344%40REDIS](https://edsp.proxyuao.elogim.com/eds/pdfviewer/pdfviewer?vid=10&sid=DB36B8C7-3273-42CC-8077-83E951889344%40REDIS)

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=K5LLT7PBOYK](https://www.youtube.com/watch?v=K5LLT7PBOYK)

[HTTPS://MEDIUM.COM/@EDUMAR111/GETH-ETHEREUM-PRIMEROS-PASOS-  
D71186CF9A70](https://medium.com/@edumar111/geth-ethereum-primeros-pasos-d71186cf9a70)

[HTTPS://YOUTU.BE/KUDLT2SF6WY?SI=46OAJLJGNT-WE6-S](https://youtu.be/KUDLT2SF6WY?si=46OAJLJGNT-WE6-S)

[HTTPS://GETH.ETHEREUM.ORG/DOCS/GETTING-STARTED/INSTALLING-GETH](https://geth.ethereum.org/docs/getting-started/installing-geth)

