UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Informe PTI Lab 2: Blockchain

Arnau Garcia Gonzalez

PTI Quadrimestre tardor Curs 2025-26

Índex

1. Introducció a la Blockchain

La tecnologia blockchain ha estat un avenç amb molt impacte en la societat, a causa de diferents sectors que s'han vist afectats. Tot i que cal destacar que recentment sembla que perd una mica d'importància, especialment pel fet que la IA està assolint un paper protagonista en el que fa a noves tecnologies. El blockchain es una tecnologia originalment dissenyada com a base tecnològica de Bitcoin, avui dia el seu abast transcendeix el sector financer i s'estén a múltiples àrees com la gestió d'identitats, les cadenes de subministrament, els contractes intel·ligents i l'administració pública.

Una blockchain es pot entendre com un llibre de registre distribuït i descentralitzat en què la informació s'emmagatzema en blocs enllaçats criptogràficament. Cada bloc conté un conjunt de transaccions que, un cop validades i segellades, s'incorporen de manera immutable a la cadena. Gràcies a aquesta estructura, es garanteixen la transparència, la resistència a la manipulació de dades i la confiança entre participants que no necessiten una autoritat central.

Els mecanismes de consens, com la Prova de Treball (Proof of Work, PoW), asseguren que tots els nodes de la xarxa comparteixin una única versió vàlida de la cadena, resolent així el problema de la doble despesa i garantint la coherència del sistema. Tanmateix, també han sorgit alternatives més eficients com Proof of Stake (PoS) o Proof of Authority (PoA), que busquen reduir el cost computacional i energètic elevat associat a PoW.

L'estudi de blockchain és especialment rellevant perquè combina aspectes de criptografia, sistemes distribuïts i economia digital. En aquest treball, ens centrarem a explorar els fonaments mitjançant una implementació simplificada en Python, amb l'objectiu de comprendre en la pràctica els conceptes clau com ara transaccions, mineria, consens i validació de cadenes.

2. Realització de la pràctica

2.1 Descripció de la pràctica i preparatius

En aquest treball, ens centrarem a explorar els fonaments de la tecnologia blockchain mitjançant una implementació simplificada en Python, amb l'objectiu de comprendre en la pràctica els conceptes clau com ara transaccions, mineria, consens i validació de cadenes.

Pel seguiment de la pràctica se seguirà la guia proporcionada pel professor i s'utilitzaran els arxius d'aquest mateix lloc: https://repo.fib.upc.es/felix.freitag/pti/-/tree/master/blockchain.

2.2 Instal·lació de l'entorn

Primer que tot s'ha comprovat que python3 estava instal·lat i també s'han instal·lat els paquets necessaris per l'execució del programa python (flask, requests i Werkzeug). Finalment s'ha executat el programa proporcionat (blockchain.py). Per fer tot això s'han executat les següents comandes:

```Shell
python3 --version   #Per comprobar que python3 està instalat

#Instala els paquets necesaris
python3 -m pip install Flask requests
python3 -m pip install Werkzeug

#Executa el programa - Comprobem que executa correctament
python3 blockchain.py -p 5000
```

Aquest procés mencionat no és el que s'ha seguit, ja que s'ha optat per crear un entorn virtual, i així evitar problemes futurs amb paquets. Per crear l'entorn virtual i fer el mateix que el mencionat just a sobre s'ha executat el següent:

```Shell
python3 -m venv prbc              #Crea l'entorn virtual
source prbc/bin/activate   #Activa l'entorn
pip3 install flask requests werkzeug    #Instal·la les dependències
python3 blockchain.py -p 5000    #Executa el programa
```

2.3 Proves de la blockchain

A continuació s'han dut a terme algunes proves, descrites al lloc ja mencionat, per familiaritzar-se amb el funcionament del programa i de la blockchain.

En aquest experiment s'han utilitzat dos nodes (el node 1 al port 5000 i el node 2 al port 5001) que participaran en la blockchain. Mitjançant un client HTTP s'han fet operacions sobre aquests nodes: s'han escrit transaccions, s'han minat blocs, aplicant la PoW (Proof of Work), i, finalment, s'ha aplicat el mecanisme de consens perquè tots dos nodes comparteixin i acceptin una mateixa cadena vàlida.

A continuació es descriuran els diferents passos de l'experiment. S'ha fet primer instrucció per instrucció i després s'ha elaborat un script per poder dur a terme tot l'experiment de manera ràpida (script.sh).

Primer que tot s'han executat els dos nodes:

```
Shell
#Run the two nodes
python3 blockchain.py -p 5000 &  #Executa el primer node al port 5000
python3 blockchain.py -p 5001 &  #Executa el primer node al port 5001
```

Una vegada executant-se escrivim les transaccions al node 1. Després s'han minat 3 blocs a aquest node, comprovant cada vegada l'estat de la cadena. A sobre de cada comanda s'especifica la funcionalitat d'aquesta.

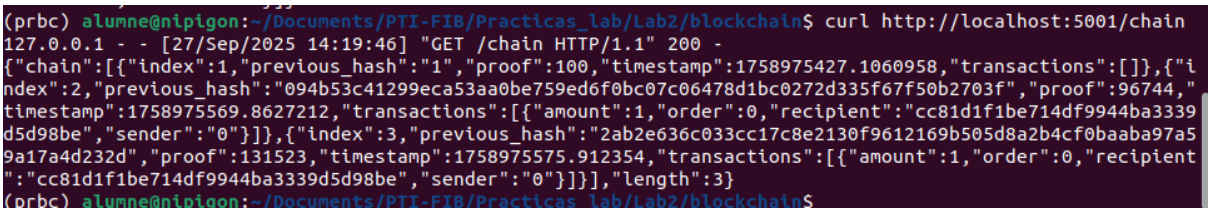
```
Shell
#Write two transactions into node 1
curl -X POST -H "Content-Type: application/json" -d '{"sender":
"A","recipient": "B", "amount": 8, "order": 1 }'
http://localhost:5000/transactions/new
curl -X POST -H "Content-Type: application/json" -d '{"sender":
"B","recipient": "C", "amount": 5, "order": 2 }'
http://localhost:5000/transactions/new

#Mine block at node 1
curl http://localhost:5000/mine
#See chain at node 1
curl http://localhost:5000/chain

#Mine another block at node 1
curl http://localhost:5000/mine
#See chain at node 1
curl http://localhost:5000/chain

#Mine another block at node 1
curl http://localhost:5000/mine
#See chain at node 1
curl http://localhost:5000/chain

#En aquest punt el node 1 té 3 blocs
```

Aquest és l'aspecte que té la chain del node 1 en aquest punt:

Després de les operacions al node 1, fem operacions al node 2. En concret minarem dos blocs.

```shell
#Mine block at node 2
curl http://localhost:5001/mine
#Mine another block at node 2
curl http://localhost:5001/mine
#See chain at node 2
curl http://localhost:5001/chain

#En aquest punt el node 2 té dos blocs
```

Aquest és l'aspecte que té la chain del node 2 en aquest punt:



Finalment, volem sincronitzar les dues cadenes, per això primer cal registrar-les el criteri de sincronització és que la cadena comuna acceptada per tots dos nodes sigui la que tingui més blocs i que sigui vàlida (és a dir, la cadena més llarga amb els hashes correctes). Per fer la sincronització s'ha executat el següent:

```shell
# We will now register the nodes to each other: Node 1 will know node 2 and
node 2 will know node 1.
#Register nodes in node 1
curl -X POST -H "Content-Type: application/json" -d
'{"nodes":"http://localhost:5001"}' http://localhost:5000/nodes/register

#Register nodes in node 2
```

```
curl -X POST -H "Content-Type: application/json" -d '{"nodes":"http://localhost:5000"}' http://localhost:5001/nodes/register


#node 1 will synchronize its chain
curl http://localhost:5000/nodes/resolve

#node 2 will synchronize its chain
curl http://localhost:5001/nodes/resolve
```

Aquest ha estat el resultat de les dues sincronitzacions:

```
(prbc) alumne@nipigon:~/Documents/PTI-FIB/Practicas_lab/Lab2/blockchain$ curl -X POST -H "Content-Type: application/json" -d '{"nodes":"http://localhost:5001"}' http://localhost:5000/nodes/register
127.0.0.1 - - [27/Sep/2025 14:38:45] "POST /nodes/register HTTP/1.1" 201 -
{"message":"New nodes have been added","total_nodes":["localhost:5001"]}
(prbc) alumne@nipigon:~/Documents/PTI-FIB/Practicas_lab/Lab2/blockchain$ curl -X POST -H "Content-Type: application/json" -d '{"nodes":"http://localhost:5000"}' http://localhost:5001/nodes/register
127.0.0.1 - - [27/Sep/2025 14:38:53] "POST /nodes/register HTTP/1.1" 201 -
{"message":"New nodes have been added","total_nodes":["localhost:5000"]}
(prbc) alumne@nipigon:~/Documents/PTI-FIB/Practicas_lab/Lab2/blockchain$ curl http://localhost:5000/nodes/resolve
127.0.0.1 - - [27/Sep/2025 14:39:00] "GET /chain HTTP/1.1" 200 -
127.0.0.1 - - [27/Sep/2025 14:39:00] "GET /nodes/resolve HTTP/1.1" 200 -
{"chain":[{"index":1,"previous_hash":"1","proof":100,"timestamp":1758976687.851917,"transactions":[]},{"index":2,"previous_hash":"546c66d17d0b485cff4b72f8c870c0ce91183c4fbe7cffd0bc97ebecf6f83c51","proof":75993,"timestamp":1758976690.7988307,"transactions":[{"amount":8,"order":1,"recipient":"B","sender":"A"},{"amount":5,"order":2,"recipient":"C","sender":"B"},{"amount":1,"order":0,"recipient":"df757e502c0f459aa409f937ca32833b","sender":"0"}]},{"index":3,"previous_hash":"74d4fb8756bb8d735bf08e03cb9f83a8d95334743a76c8e064f20f8f44d65f78","proof":58596,"timestamp":1758976690.8721387,"transactions":[{"amount":1,"order":0,"recipient":"df757e502c0f459aa409f937ca32833b","sender":"0"}]},{"index":4,"previous_hash":"3ec0f4b3431626a7662e52f82b4c462de0d19f8ba7d11d4ed476df3243ce77e3","proof":23591,"timestamp":1758976690.910783,"transactions":[{"amount":1,"order":0,"recipient":"df757e502c0f459aa409f937ca32833b","sender":"0"}]}],"message":"Our chain is authoritative"}
```

```
(prbc) alumne@nipigon:~/Documents/PTI-FIB/Practicas_lab/Lab2/blockchain$ curl http://localhost:5001/nodes/resolve
127.0.0.1 - - [27/Sep/2025 14:39:08] "GET /chain HTTP/1.1" 200 -
{'index': 1, 'previous_hash': '1', 'proof': 100, 'timestamp': 1758976687.851917, 'transactions': []}
{'index': 2, 'previous_hash': '546c66d17d0b485cff4b72f8c870c0ce91183c4fbe7cffd0bc97ebecf6f83c51', 'proof': 75993, 'timestamp': 1758976690.7988307, 'transactions': [{'amount': 8, 'order': 1, 'recipient': 'B', 'sender': 'A'}, {'amount': 5, 'order': 2, 'recipient': 'C', 'sender': 'B'}, {'amount': 1, 'order': 0, 'recipient': 'df757e502c0f459aa409f937ca32833b', 'sender': '0'}]}

----------

{'index': 2, 'previous_hash': '546c66d17d0b485cff4b72f8c870c0ce91183c4fbe7cffd0bc97ebecf6f83c51', 'proof': 75993, 'timestamp': 1758976690.7988307, 'transactions': [{'amount': 8, 'order': 1, 'recipient': 'B', 'sender': 'A'}, {'amount': 5, 'order': 2, 'recipient': 'C', 'sender': 'B'}, {'amount': 1, 'order': 0, 'recipient': 'df757e502c0f459aa409f937ca32833b', 'sender': '0'}]}
{'index': 3, 'previous_hash': '74d4fb8756bb8d735bf08e03cb9f83a8d95334743a76c8e064f20f8f44d65f78', 'proof': 58596, 'timestamp': 1758976690.8721387, 'transactions': [{'amount': 1, 'order': 0, 'recipient': 'df757e502c0f459aa409f937ca32833b', 'sender': '0'}]}

----------

{'index': 3, 'previous_hash': '74d4fb8756bb8d735bf08e03cb9f83a8d95334743a76c8e064f20f8f44d65f78', 'proof': 58596, 'timestamp': 1758976690.8721387, 'transactions': [{'amount': 1, 'order': 0, 'recipient': 'df757e502c0f459aa409f937ca32833b', 'sender': '0'}]}
{'index': 4, 'previous_hash': '3ec0f4b3431626a7662e52f82b4c462de0d19f8ba7d11d4ed476df3243ce77e3', 'proof': 23591, 'timestamp': 1758976690.910783, 'transactions': [{'amount': 1, 'order': 0, 'recipient': 'df757e502c0f459aa409f937ca32833b', 'sender': '0'}]}

----------

127.0.0.1 - - [27/Sep/2025 14:39:08] "GET /nodes/resolve HTTP/1.1" 200 -
{"message":"Our chain was replaced","new_chain":[{"index":1,"previous_hash":"1","proof":100,"timestamp":1758976687.851917,"transactions":[]},{"index":2,"previous_hash":"546c66d17d0b485cff4b72f8c870c0ce91183c4fbe7cffd0bc97ebecf6f83c51","proof":75993,"timestamp":1758976690.7988307,"transactions":[{"amount":8,"order":1,"recipient":"B","sender":"A"},{"amount":5,"order":2,"recipient":"C","sender":"B"},{"amount":1,"order":0,"recipient":"df757e502c0f459aa409f937ca32833b","sender":"0"}]},{"index":3,"previous_hash":"74d4fb8756bb8d735bf08e03cb9f83a8d95334743a76c8e064f20f8f44d65f78","proof":58596,"timestamp":1758976690.8721387,"transactions":[{"amount":1,"order":0,"recipient":"df757e502c0f459aa409f937ca32833b","sender":"0"}]},{"index":4,"previous_hash":"3ec0f4b3431626a7662e52f82b4c462de0d19f8ba7d11d4ed476df3243ce77e3","proof":23591,"timestamp":1758976690.910783,"transactions":[{"amount":1,"order":0,"recipient":"df757e502c0f459aa409f937ca32833b","sender":"0"}]}]}
```

2.4 Afegir funcions a la blockchain

En aquest apartat s'han afegit funcionalitats a la API Web. En concret s'han afegit funcions per: Llistar nodes registrats, validar la cadena de blocs i manipular la cadena d'un node. Per fer aquestes funcions s'ha utilitzat ChatGPT com a eina auxiliar.

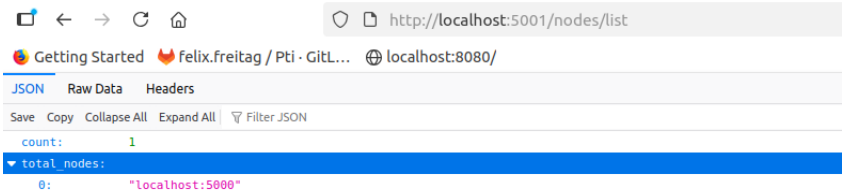
Per provar totes aquestes funcions s'ha utilitzat el script que executa totes les instruccions explicades a l'apartat 2.3 Probes de la blockchain. Les eines de les llibreries instal·lades a l'inici faciliten molt el codi d'aquestes funcions.

2.4.1 Llistar nodes registrats

Per fer aquesta primera funció de llistar els nodes registrats. Per aquesta funció s'ha utilitzat list (blockchain.nodes), que ja fa la majoria de la feina. Aquestes el codi de la implementació.

```python
@app.route('/nodes/list', methods=['GET'])
def list_nodes():
    response = {
        'total_nodes': list(blockchain.nodes),
        'count': len(blockchain.nodes)
    }
    return jsonify(response), 200
```

En accedir a `http://localhost:5001/nodes/list` o en executar la comanda `"curl http://localhost:5001/nodes/list"` aconseguim el següent ouput, la llista dels nodes registrats.



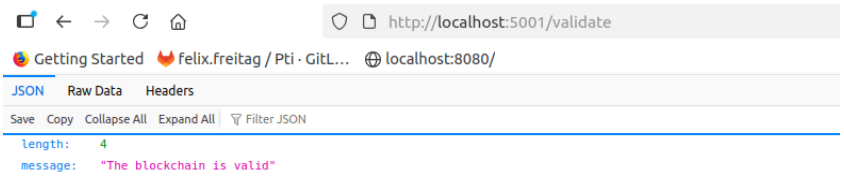
2.4.2 Validar la cadena de blocs

La segona funció que s'ha fet s'encarrega de validar la cadena de blocs d'un node. És a dir, valida els hashes del node. En aquest cas altra vegada tenim una funció a les llibreries que fa la validació, en aquest cas la funció implementada és la següent:

```python
app.route('/validate', methods=['GET'])
def validate_chain():
    is_valid = blockchain.valid_chain(blockchain.chain)
```

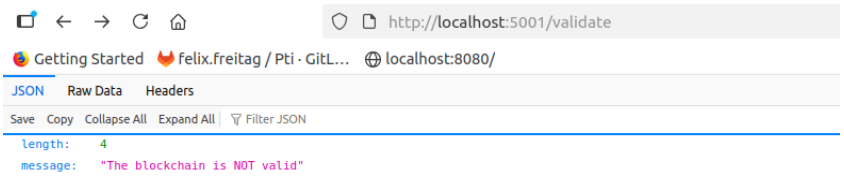
```
    if is_valid:
        response = {
            'message': 'The blockchain is valid',
            'length': len(blockchain.chain)
        }
    else:
        response = {
            'message': 'The blockchain is NOT valid',
            'length': len(blockchain.chain)
        }

    return jsonify(response), 200
```

En accedir a `http://localhost:5001/validate` o, cosa equivalent, en executar `"curl http://localhost:5001/validate"` aconseguim el següent ouput indicant si la cadena és vàlida o no:



Per aquest test també s'ha provat d'invalidar una cadena i veure si l'output és l'esperat. Aquest procediment d'invalidació s'explica a l'apartat 2.4.3 Manipular la cadena d'un node.

Una vegada invalidada i després d'executar `"curl http://localhost:5001/validate"` s'ha obtingut el següent resultat



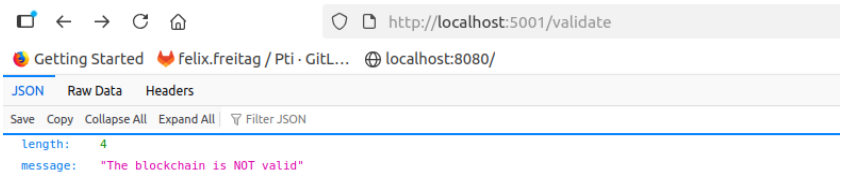
2.4.3 Manipular la cadena d'un node

La tercera funció implementada s'encarrega de manipular la cadena d'un node, per fer-la incorrecta. El codi de la implementació és el següent:

```
Python
@app.route('/nodes/manipulate', methods=['POST'])
def manipulate():
```

```
response = {}
# Replacing the proof of the second block by a random value,
# should invalidate the blockchain.
if len(blockchain.chain) < 2:
    response = {
    "message":"Not enough blocks stored in the blockchain."
}
else:
    blockchain.chain[1]['proof'] = 100
    response = {
        "message":"Blockchain manipulated successfully."
    }
return jsonify(response), 200
```

Una vegada executat amb `"curl -X POST http://localhost:5001/nodes/manipulate"` s'ha comprovat que efectivament la cadena s'ha tornat invàlida, accedint, d'igual forma que a l'apartat anterior a `http://localhost:5001/validate`. Aquest és el resultat:



2.5 Tasques addicionals

En aquest apartat es comenten les preguntes realitzades a l'enunciat.

2.5.1 Alternatives al mecanisme de Proof of Work

El mecanisme de Proof of Work (PoW) és efectiu per garantir el consens, però té un cost computacional molt elevat, pel que existeixen algunes alternatives per reduir la càrrega. Algunes d'aquestes són:

- Proof of Stake (PoS): el dret a validar blocs depèn de la quantitat de moneda que un node té bloquejada (stake). Redueix molt el consum energètic.
- Proof of Authority (PoA): la validació es delega a un conjunt limitat de nodes de confiança prèviament autoritzats. És més eficient però menys descentralitzat.
- Proof of Elapsed Time (PoET): assigna el dret de minar blocs de forma aleatòria, però controlada per hardware de confiança (p. ex. Intel SGX).

En el nostre codi, es podria integrar una versió senzilla de Proof of Stake, en què el "pes" del node es defineixi manualment o de forma aleatòria. La modificació principal seria substituir la funció de validació de la prova (valid_proof) per una lògica basada en el stake en lloc del càlcul de hash intensiu.

2.5.2 Mecanismes de protecció en la creació de transaccions

Actualment, qualsevol usuari pot escriure una transacció. Per protegir aquesta operació, es podrien implementar mecanismes com:

- Signatura digital: cada usuari tindria una clau privada per signar les seves transaccions i una clau pública que els altres nodes podrien verificar.
- Autenticació prèvia: afegir un sistema senzill d'API keys o tokens que validi l'accés al mètode /transactions/new.
- Límits i permisos: establir rols o límits de transaccions per usuari.

La signatura digital és l'opció més coherent amb l'esperit de la blockchain, ja que permet verificar l'autenticitat sense necessitat d'una autoritat central, és a dir protecció però sense perdre un dels factors més importants de la blockchain.

2.5.3 Emmagatzematge de l'estat en la blockchain

En l'exercici actual, la blockchain no guarda l'estat dels comptes (per exemple, els saldos de A o B després de cada transacció). Només s'emmagatzemen les transaccions com a registres independents.

Per afegir l'estat, es podria:

- Incorporar un diccionari global de comptes que es vagi actualitzant en cada transacció.
- Afegir aquesta informació en cada bloc (per exemple, un snapshot de l'estat després d'aplicar totes les transaccions del bloc).

D'aquesta manera, es podria consultar fàcilment el saldo actual de cada usuari sense haver de recórrer tota la cadena de blocs i aplicar totes les transaccions.

Pel nostre codi afegir la informació al bloc seria l'opció més senzilla. Afegir un camp nou al bloc, per exemple `state`, que contingui un diccionari `{adreça: saldo}` resultant després d'aplicar totes les transaccions del bloc. Tot i que aquesta opció té l'inconvenient de què pot ocupar més espai, en tenir un camp addicional.

2.5.4 Privacitat de les dades de les transaccions i possibles solucions

En l'exercici, les dades de les transaccions són públiques i visibles per tots els nodes. Aquesta obertura és necessària perquè els nodes puguin validar les transaccions i els blocs, però compromet la privacitat dels participants.

Algunes tècniques que podrien millorar la privacitat són:

- Transaccions xifrades amb sistemes que permetin validar-les sense revelar tota la informació.
- Zero-Knowledge Proofs (ZKP): permeten demostrar que una transacció és vàlida sense revelar-ne els detalls (per exemple, zk-SNARKs, usats a Zcash).
- Mixing/Tumbling services: barregen múltiples transaccions per dificultar el rastreig.

En el nostre cas, es podria explorar l'ús de signatures de coneixement zero senzilles per validar que un usuari té saldo suficient sense mostrar la quantitat exacta.