

Blockchain

Projecte Tecnologies d'Informació

Pau Coma Ninot

Àlex Aguilera Martínez

Grup 20

1. Objectius pràctica

L'objectiu d'aquesta pràctica és la creació d'un Web API per explorar alguns conceptes de blockchain fent ús del llenguatge de programació Python i d'un client HTTP per realitzar les operacions, Curl (ja utilitzat en pràctiques anteriors).

Aquesta pràctica, consistirà en afegir tres funcions noves al codi proporcionat per el professorat. Aquestes són:

- **/nodes/list** → Permet veure els altres nodes que es registre en un node
- **/validate** → Permet validar la cadena de hashes de la blockchain emmagatzemada en un node
- **/nodes/manipulate** → Manipular la cadena d'un node, de manera que quan es validi, esdevindrà incorrecte aquesta.

Per anar testejant el que fem, serà necessari també crear un script que crea els nodes, per facilitar-nos la feina.

1. Implementació

1- Instal·lació i configuració

Per aquesta pràctica, la instal·lació és ràpida. Hem d'instal·lar python i els paquets necessaris per la pràctica de la següent manera:

```
$sudo apt-get update (actualitzar, abans de res)
```

```
$sudo apt-get install python3-pip (instal·lació de python)
```

```
$python3.6 -m pip install Flask==0.12.2 requests==2.18.4 (instal·lació dels paquets)
```

Finalment, descarreguem el codi i descomprimim el tar.gz per poder començar a afegir les noves funcions a blockchain.py:

```
$git clone https://gitlab.fib.upc.edu/pti/pti.git
```

```
$tar -xzf blockchain.tar.gz
```

Una vegada fet això, per poder aprendre alguns conceptes de blockchain, creem dos nodes un amb el port 5000 i l'altre amb el port 5001 en mode d'experiment per fer ús del codi proporcionat. Seguidament, fem una serie de operacions amb els dos nodes (totes indicades per els professors) per veure que fa cadascuna de les funcions ja creades i poder afegir les nostres noves.

Com aquestes operacions s'han de fer cada vegada que volem testejar que funciona el nostre codi, creem un script.sh (es pot veure al rar proporcionat) per executar-lo cada vegada i d'aquesta manera facilitar-nos la feina.

2- List

En aquest cas, hem de llistar per consola els nodes registrats en un node determinat. Per fer-ho, afegim la següent funció llistar() al codi blockchain.py:

```
@app.route('/nodes/list', methods=['GET'])
def llistar():

    response = {
        'total_nodes': list(blockchain.nodes)
    }

    return jsonify(response), 200
```

Per fer-ho, hem entès la funció register_nodes() que una vegada registra un nou node, llista el seu contingut.

Per comprovar el correcte funcionament de la funció, hem de fer ús de curl amb un get, per obtenir els nodes i llistar-los. A continuació, hem introduït un exemple amb el node 5001 per veure el resultat:

```
alume@pcrecanvib5:~/pti/blockchain$ curl -X GET -H "Content-Type: application/json" http://localhost:5001/nodes/list
127.0.0.1 - - [05/Oct/2019 14:25:56] "GET /nodes/list HTTP/1.1" 200 -
{
  "total_nodes": [
    "localhost:5000"
  ]
}
```

3- Validate

Per aquesta funció, hem de validar la cadena de hashes de la blockchain emmagatzemada en un node. Per fer-ho, afegim una nova funció validar() al

```
@app.route('/validate', methods=['GET'])
def validar():

    response = {
        'validat': blockchain.valid_chain(blockchain.chain)
    }

    return jsonify(response), 200
```

codi:

Per aquesta funció, utilitzem una altre funció ja creada la qual valida la chain i retorna un boolea. Per tant, només hem cridar-la i escriure per consola el seu resultat.

Per comprovar el correcte funcionament, farem ús de curl una altre vegada amb el port 5000. Com podem veure al següent screenshot, ens retorna

```
alumni@pcrecanvib5:~/pti/blockchain$ curl http://localhost:5000/validate
127.0.0.1 - - [05/Oct/2019 14:36:42] "GET /validate HTTP/1.1" 200 -
{
  "validat": true
}
```

true, per tant es correcte:

4- Manipulate

I per últim, aquest funció ha de manipular el chain d'un node i quan aquest es validi amb la funció validar(), doni incorrecte aquesta vegada. Per fer-ho, afegim al codi que tenim, una nova funció manipular():

```
@app.route('/modify', methods=['GET'])
def modify():
    blockchain.modificar ()
    response = {
        'modificat': "blockchain has been modified",
    }
    return jsonify(response), 200
```

Per comprovar que la funció manipula correctament el chain del node, haurem d'utilitzar el curl amb la funció validar() i veure si ens retorna el resultat correcte. En aquest cas, ens retorna false, per tant sabem que ho hem fet bé.