INFO288 - Prueba 2
PARTE PRÁCTICA (6 Ptos)

Trabajo individual-grupal, máximo 2 personas.

Deben solucionar el problema a continuación y dejarlo en una carpeta El nombre del entregable será:

NombreApellidoIntegrante.zip o rar.
o
NombreApellidoIntegrante1_NombreApellidoIntegrante2.zip o rar.

Fecha límite de entrega: 30/05/2022 23:59.

Dudas del trabajo se contestan por discord, correo o en consultas en la oficina. Buena suerte.

CREANDO UNA CRIPTOMONEDA (6 Ptos)

Un grupo de emprendedores para nada fraudulentos quiere comenzar un negocio para nada fraudulento con una criptomoneda y les gusta la idea de blockchain pero nunca han intentado trabajar con una, por lo cual, este grupo le va a encargar a ud/uds que propongan una estructura para esta criptomoneda que se llamará la Scamcoin. Después de varias reuniones se llegó a un acuerdo sobre qué reglas debe seguir la Scamcoin. Para la Scamcoin se define lo siguiente como mínimo:

1. El bloque debe contener los siguientes componentes:
    a. Índice del bloque.
    b. Transacciones asociadas al bloque.
    c. Fecha cuando se minó.
    d. Proof of work.
    e. Dirección del minador que obtuvo el proof of work.
    f. Hash del bloque previo (formato hexadecimal).
    g. Hash del bloque (formato hexadecimal, obtener con lo anterior mencionado).
2. Para el Proof of Work se define que el problema para competir debe ser bicondicional y la probabilidad conjunta debe ser menor a (1x10^-9|1e-9), además, se debe trabajar con un algoritmo distinto al SHA256 usado en clases (investigar hashlib de python), por último, para armar el hash del proof se debe considerar una operación no simétrica y la fecha del momento.
3. Se debe considerar en el diseño que la Criptomoneda se trabaja en una red descentralizada, osea, que en cualquier momento un nodo se puede asociar a otros nodos que están minando la Scamcoin.
4. Se debe establecer un método de consenso que debe considerar lo siguiente:
    a. En caso de que compitan dos cadenas, debe prevalecer la más larga.
    b. En caso de que hayan 2 o más cadenas del mismo largo, se le da prioridad a la que obtuvo su último bloque antes que el resto.
    c. Lógicamente si una cadena en competencia no es válida, esta se descarta.

Una vez que se tenga la blockchain para la Scamcoin se debe levantar una aplicación en Flask que considere los siguientes métodos:

1. GetChain(): Devuelve la cadena entera mostrando a su vez el contenido de los bloques.
2. MineBlock(): Mina un bloque y devuelve los datos del bloque minado.
3. ValidateChain(): Analiza si la cadena es válida.
4. CorruptChain(): Modifica un bloque de la blockchain a propósito (elección o aleatorio), se utiliza para probar el funcionamiento de ValidateChain().
5. AddTransaction(): Agrega transacciones para un bloque que se va a minar y se devuelve un mensaje indicando que se agregaron.
6. ConnectNode(): Conecta un nodo a la red descentralizada.
7. DisconnectNode(): Desconecta al nodo de la red descentralizada.
8. ReplaceChain(): Aplica mecanismo de consenso sobre la cadena del nodo, la respuesta depende de 4 casos:
    a. Si la cadena ya era la más apropiada.
    b. Si la cadena no era la más larga.
    c. Si la cadena era igual de larga que otra pero perdió por tiempo de minado.
    d. Si la cadena no era válida.

Debe programar todo en un archivo llamado Scamcoin.py, y debe además probar su red descentralizada con un mínimo de 6 bloques distintos.

En el entregable final debe considerar lo siguiente:

1. Scamcoin.py (criptomoneda base).
2. nodes.json (archivo con los nodos).
3. Node#Number.py (Nodos minadores, puede también optimizar y usar un solo archivo el cual al momento de iniciarse se le pasa su ruta de conexión).
4. Pequeño reporte con fotos (no del hombre araña) de los siguientes casos:
    a. Nodo ejecutándose.
    b. Nodo agregando transacciones.
    c. Nodo minando bloque.
    d. Nodo obteniendo la cadena..
    e. Nodo conectado a una red.
    f. Nodo desconectandose de una red.
    g. Nodo validando su cadena (una para cadena válida y otra para cadena invalida).
    h. Nodo aplicando consenso (una por cada escenario posible del consenso).

    El formato del reporte es simplemente imagen + explicación de lo que sucede, nada más, fuente y tipo de letra a elección, cantidad de hojas a elección. El archivo idealmente que sea un pdf.

Para probar sus métodos puede usar Postman, la consola del navegador o cualquier herramienta que considere necesaria. También para propósitos de probar su criptomoneda pueden aumentar la probabilidad del problema asociado al proof of work, sin embargo, deben entregar el problema con lo que se pide en el archivo final.