Sistema de Transações com Consenso de Validação

Valor  
O trabalho tem peso de 3,00 (três pontos) na nota semestral A2.  
Quantidade de alunos  
O trabalho deve pode ser desenvolvido individualmente (1 aluno), duplas (2 alunos) ou trios (3 alunos).  
Data de entrega, defesa e pontuação  
A defesa é obrigatória e acontece no dia da entrega do trabalho;  
A entrega deve acontecer até o dia marcado no portal universitário;  
A entrega é composta por um link para o repositório do projeto, que deve estar disponível no GITHub;  
É obrigatório que todos os participantes façam commits relevantes no projeto, sendo que a faltas destes  
acarretará em zeramento da nota para aquele que não cumprir está diretiva;  
A nota será composta da seguinte forma: (Requisitos Cumpridos - Regras Não Cumpridas) \* %Defesa  
Objetivo  
Desenvolver um Sistema Distribuído capaz de simular o processo de validação por consenso de uma  
moeda digital, aqui denominada NoNameCoin;  
Contextualização  
Um sistema de transações por modelo de consenso é uma forma de processamento de transações em  
um ambiente distribuído, onde múltiplos participantes, também conhecidos como nós, concordam em  
validar e confirmar a legitimidade de transações sem a necessidade de uma autoridade central. O  
consenso é alcançado entre os nós para garantir que todas as transações sejam consideradas válidas e  
registradas de forma consistente em todos os nós da rede.  
Existem diferentes tipos de modelos de consenso utilizados em sistemas de transações, incluindo o Proof  
of Work (Prova de Trabalho), o Proof of Stake (Prova de Participação), o Delegated Proof of Stake (Prova  
de Participação Delegada), o Practical Byzantine Fault Tolerance (Tolerância a Falhas Bizantinas Práticas),  
entre outros.  
O sistema será modelado considerando o modelo de consenso Proof of Stake. Nesse caso, os nós da  
rede são selecionados para validar transações com base na quantidade de moeda que eles possuem e  
"apostam" como garantia. Quanto maior a quantidade de moeda que um nó possui, maior é a  
probabilidade de ser selecionado para validar transações e criar novos blocos. Esse modelo é  
considerado mais eficiente em termos de consumo de recursos computacionais e é usado em  
criptomoedas como o Ethereum.  
Modelagem  
O sistema será composto por três camadas: Banco, Seletor e Validador:  
• Banco: Essa camada tem por função distribuir as transações e informações das contas atreladas  
para a camada “Seletor”;  
• Seletor: Essa camada tem por função cadastrar e verificar a viabilidades de elementos  
computacionais serem validadores, assim com selecionar qual ou quais validadores receberam a  
transação ativa, além de gerenciar o consenso destes validados;  
• Validador: Essa camada tem por função validar, seguindo as regras do ecossistema, as transações  
que receber do seletor ao qual está cadastrada;  
Definições de Rotas

Programação Distribuída 2  
• As chamadas da API do banco estão explicitas na rota “/” do projeto;  
• Todas as rotas de “/trans” estão relacionadas as transações de ativos entre os clientes do sistema;  
• Todas as rotas de “/seletor” estão relacionadas aos sistemas de seleção;  
• Todas as rotas de “/validador” estão relacionadas aos sistemas de validação;  
• A rota “/hora” está relacionada ao horário atual do sistema para a sincronização;  
Regras de validação  
• O remetente deve ter um valor em saldo igual ou maior que o valor da transação, acrescido das  
taxas para a mesma ser válida;  
• O horário da transação deve ser menor ou igual ao horário atual do sistema e deve ser maior que  
o horário da última transação para ser válida;  
• Caso o remetente tenha feito mais que 100 transações no último minuto, as transações no  
próximo minuto devem ser invalidas;  
o Opcional: Aumentar o tempo de recusa, caso o problema persista;  
• Na hora do cadastro o validador recebe uma chave única do seletor. Em toda transação, o  
validador deve retornar a chave única que recebeu no cadastro. Caso as chaves sejam iguais, a  
transação é concluída, caso contrário, a transação não é concluída;  
• Status da Transação (Servem da camada “Validador” para a camada “Seletor”, assim como da  
camada “Seletor” para a camada “Banco”):  
o 1 = Concluída com Sucesso  
o 2 = Não aprovada (erro)  
o 0 = Não executada  
▪ Códigos opcionais podem existir, mas devem ser descritos para implementação  
na camada “Banco”;  
Regras de Seleção  
• O seletor deve escolher pelo menos três Validados para a transação ser concluída. Caso a  
quantidade mínima não esteja disponível, colocar a transação em espera, por no máximo, um  
minuto;  
• O consenso é gerado com mais de 50% de um status (Aprovada ou Não Aprovada);  
• O percentual de chance de escolha do validador é dinâmico, a depender da quantidade de  
moedas que o mesmo irá disponibilizar, tendo 50% de redução para Flag=1 e 75% de redução  
para Flag=2;  
• A percentual mínimo de escolha de um validador não é limitado e o percentual máximo de  
escolha deve ser de 20%;  
• Validadores inconsistentes ou mal intencionados serão identificados por uma FLAG de alerta.  
Caso o Validador receba mais que duas FLAGs, o mesmo deve ser eliminado da rede, perdendo  
seu saldo. Uma FLAG é retirada a cada 10.000 transações coerentes;  
• Caso o mesmo validador seja escolhido por cinco vezes seguidas, o mesmo deve ficar em HOLD  
pelas próximas 5 trasações;  
• Após a expulsão, um validador pode retorna a rede até duas vezes, sem obrigatório o dobro do  
saldo travado anteriormente para o retorno;  
• É obrigatório que o Validador se cadastre no Seletor, travando um saldo mínimo de 50  
NoNameCoins;  
• Para cada validação bem sucedida o seletor recebera 1,5% da quantidade de NoNameCoins  
transacionadas, ficando 0,5% travado para o validador e o restante distribuído igualitariamente  
entre os validadores;

Programação Distribuída 3  
Requisitos  
• O sistema deve trabalhar no formatado sistema Web Service (10%);  
• O sistema deve ter um algoritmo pertinente para a eleição dos validadores, podendo ser  
acionado a qualquer momento (10%);  
• A funcionalidade de inserção e remoção dos validadores deve estar disponível (10%);  
• O servidor deve armazenar todos os passos da eleição (10%);  
• O sistema deve ser capaz de diferenciar uma validação bem sucedida de uma avaliação mal  
sucedida (10%);  
• Sincronização do tempo entre o sistema seletor e o sistema validador, a partir do horário  
disponibilizado pelo banco (10%);  
• Desenvolvimento do conjunto regras para a validação da transação, no validador (10%);  
• Capacidade do sistema seletor identificar o validador falho, a partir de uma estrutura de  
“votação” interna (10%);  
• Interligação entre o sistema validador, o sistema seletor e o sistema de gerenciamento (10%);  
• Tolerância a falha no seletor e seletor (10%)