



Disciplina: Sistemas Distribuídos  
Professora: Ana Cristina Barreiras Kochem Vendramin

**Avaliação (valor 2,0)**  
**Arquitetura de Processos Pares e Segurança.**

Siga as instruções abaixo para desenvolver e testar um sistema de compra e venda de moedas virtuais.

1. Considere um conjunto mínimo inicial de quatro processos que desejam comercializar moedas virtuais em uma arquitetura *peer to peer*. Isto é, diferente de um sistema de câmbio de moedas desenvolvido no modelo tradicional cliente-servidor, na arquitetura P2P ocorrerá uma comunicação direta entre os pares, sem depender de uma instituição financeira central;
2. Cada processo mantém uma carteira para armazenar suas credenciais digitais (chaves pública e privada). Essas credenciais são essenciais para efetuar transações com moedas;
3. Todas as transações com moedas são públicas e o sistema provê um banco de dados distribuído por todos os processos da rede P2P para registrar todas as transações (ex.: "processo A enviou X moedas para processo B") e a quantia de moedas virtuais que cada processo possui (ex.: A tem -X moedas e B tem +X moedas). O sistema mantém réplicas idênticas dos dados;
4. Para prover funções básicas de segurança, como validar a autoria de uma transação (isto é, certificar que as moedas só podem ser vendidas pelo seu dono, será utilizada a criptografia de chaves assimétricas (chave pública e privada);
5. Qualquer processo, chamado de minerador, pode verificar se uma transação é ou não válida. Os mineradores tem a capacidade de escolher quais transações eles irão processar. Uma transação é válida se a quantia de moedas está sendo vendida pelo seu próprio dono e se este dono possui realmente essa quantia de moedas para vender;
6. Toda transação tem uma taxa que será atribuída como recompensa ao primeiro minerador que validá-la;

7. (valor 0,6) Um processo utilizará a comunicação em grupo (*multicast*) para os seguintes fins:
  - anunciar sua disponibilidade na rede (valor 0,15);
  - divulgar sua chave pública (valor 0,15);
  - anunciar o preço de venda de suas moedas (valor 0,15);
  - retornar o resultado de uma mineração para que o banco de dados distribuídos possa ser atualizado (valor 0,15);
8. (valor 0,4) O sistema deve permitir a entrada e saída de pares da rede durante o funcionamento da aplicação. Quando um novo par entrar na rede ele deverá se anunciar e divulgar sua chave pública aos demais utilizando a comunicação *multicast*. Os demais processos também devem se anunciar ao novo processo através de uma comunicação *unicast* e enviar o banco de dados;
9. (valor 0,4) A mensagem de compra de moedas deve ser enviada por *unicast* do comprador ao vendedor. A transferência de moedas se dá através de uma transação entre o remetente e o destinatário. Uma taxa de transação será paga pelo vendedor da moeda. Para criar uma transação, o processo remetente precisa apenas informar a quantia de moedas que está vendendo e qual o endereço do destinatário. O remetente autoriza a transferência assinando a transação com a sua chave privada-secreta (isto é, a mensagem da transação é criptografada com a chave privada do remetente). O remetente comunica essa transação de modo *multicast* aos outros processos na rede P2P. Os dois processos envolvidos na comunicação já poderão ver a transação em suas carteiras como “não confirmadas” (ou seja, esperando por um minerador para verificar e incluir a transação no BD distribuído);
10. (valor 0,4) Os processos da rede que desejam atuar como mineradores validam a assinatura criptográfica e as quantias envolvidas (processo chamado de mineração) para evitar falsificação e vendas duplas. O processo que validar primeiramente a transação receberá uma recompensa, isto é, receberá uma taxa da transação (por exemplo, assuma uma taxa única para toda transação). Para poder resgatar sua recompensa, uma transação especial chamada de recompensa é incluída pelo minerador junto com os pagamentos que ele processou;
11. (valor 0,2) Uma vez validada, a transação de venda, juntamente com a transação de recompensa, devem ser incluídas no BD distribuído.

Obs.: Utilizar sockets. É obrigatório documentar todo o código e a equipe é de dois programadores.