



Disciplina: Sistemas Distribuídos
Professora: Ana Cristina Barreiras Kochem Vendramin

Avaliação (valor 2,0)
Controle de Concorrência e Transações

Desenvolver uma aplicação que gerencie transações de consulta e troca de cartões postais entre um conjunto de pelo menos três colecionadores e um coordenador.

Requisitos da aplicação:

- Utilizar uma *middleware* (Java RMI, Web Services ou RPC/XML) para prover a comunicação.

COLEZIONADOR (valor 1,0)

Cada colecionador pode atuar como participante de uma ou mais transações.

Pode solicitar as seguintes operações ao coordenador:

- Consultar coleção de cartões postais de todos os colecionadores ativos;
- Trocar cartões com outros colecionadores;
- Obter informações sobre o estado das transações das quais é um participante.
- **(valor 0,4)** O colecionador deve implementar, ao menos, as seguintes operações: efetivar uma transação, abortar uma transação, obter decisão se pode ou não efetivar uma transação, consultar coleção e trocar cartão (remover ou inserir).
- **(valor 0,3)** É possível que algum objeto possa ser acessado por duas ou mais transações ao mesmo tempo. Por esse motivo, se faz necessária a implementação de um mecanismo que controle o acesso concorrente aos objetos (coleções) compartilhados.
 - Operações que acessam o mesmo objeto devem ser ordenadas.
 - Lembre-se que a sincronização pode afetar o desempenho da aplicação. Então, como regra deve-se manter bloqueios pelo menor tempo possível e apenas onde for necessário.
- **(valor 0,3)** O colecionador deve registrar logs das transações do qual é participante. Em caso de falha em um colecionador, este deve ser capaz de recuperar, através de *logs*, todos os dados de forma correta após ser restabelecido: (a) estado das transações; (b) dados intermediários de uma transação incompleta; (c) dados finais de transações efetivadas.

COORDENADOR DA TRANSAÇÃO (valor 1,0)

Responsável por gerenciar as requisições de cada colecionador

- **(valor 0,5)** Para realizar operações remotas entre dois ou mais colecionadores, o coordenador implementará o protocolo *two-phase commit* de uma transação;
- **(valor 0,5)** Dependendo da operação (consulta ou troca) solicitada pelo colecionador, o coordenador determinará se criará uma transação plana ou aninhada.
 - Operações não conflitantes devem ser aninhadas. Por exemplo, uma transação “Consultar os cartões postais dentro de um grupo de n colecionadores amigos pode ser composta por n operações (obterColeção_1, obterColeção_2, obterColeção_n) que serão executadas em paralelo.
 - Operações conflitantes devem ser tratadas como transações planas. Por exemplo, a transferência de cartões entre dois colecionadores é considerada como sendo um par de operações conflitantes (remove_Cartão_Coleção_1 e insere_Cartão_Coleção_2).

Requisitos de cada transação:

- Deve-se garantir que as quatro propriedades ACID sejam atendidas: **Atomicidade, Consistência, Isolamento e Durabilidade.**
- Deve-se evitar: escritas prematuras, leituras sujas, cancelamentos em cascata e impasses. Nenhum efeito de transações incompletas deve ser visível por outras transações.

Coleção

- Cada coleção é representada por um objeto com várias variáveis (exemplos: descr_coleção, nome_colecionador, quantidade, preço, trava, etc.);

Observações:

- Desenvolva uma interface com recursos de interação apropriados;
- Gerar a documentação completa de todas as classes e métodos de sua aplicação (obs.: será descontado 0,5 ponto na ausência da documentação completa);
- Equipe: dois programadores.