



Disciplina: Sistemas Distribuídos

Professora: Ana Cristina Barreiras Kochem Vendramin

Avaliação (valor 2,5)
**PyRO. Comunicação P2P. Detecção de falhas. Eleição.
Compartilhamento de Arquivos.**

Desenvolva uma aplicação distribuída que permita o **compartilhamento de arquivos** entre 5 **processos pares** (peer-to-peer - P2P). O sistema deve contar com um nó líder (**tracker**) responsável por manter um **índice centralizado** de todos os arquivos compartilhados. A comunicação entre os nós deve ser realizada utilizando o **middleware PyRO** (Python Remote Objects), baseado no padrão de **mensagens requisição-resposta**.

Serão executados **6 processos**:

- Um **serviço de nomes** (*binder*) do PyRO5: usado para **descoberta e registro das URIs dos nós**;
- Cinco **processos pares** (*peers*) que compartilham arquivos e participam de eleições. Um desses processos será eleito como *tracker* (indexador) e será responsável por manter um índice centralizado de todos os arquivos disponíveis na rede. O *tracker* também participa do compartilhamento de arquivos.

(0,4) Inicialização, Cadastro e Consulta no Serviço de Nomes e Cadastro no Tracker

- (0,2) Ao iniciar, os nós devem interagir com o serviço de nomes para: (i) informar sua referência (URI) e (ii) buscar a referência (URI) do *tracker* atual;
- (0,2) Com a referência do *tracker*, os nós devem cadastrar no *tracker* o nome dos arquivos que possuem.

(0,8) Eleição do Tracker

- (0,2) Quando um nó detecta que um *tracker* não existe, ele se declara candidato e inicia uma **eleição**, enviando mensagem aos demais nós. O candidato precisa do **voto da maioria dos nós (quórum)**, incluindo o **seu próprio voto**, para se tornar o novo *tracker*;
- (0,2) Depois que ele recebe o voto da maioria dos pares, o candidato é eleito *tracker* e se cadastra no serviço de nomes com o nome "*Tracker_Epoca_x*". Cada registro de *tracker* adicionado ao serviço de nomes está associado a uma **Época**. Uma **época** refere-se a um número sequencial (x) que é associado a um *tracker*. Esse número é utilizado para

identificar e diferenciar as várias instâncias de *trackers* ao longo do tempo. Quando um novo *tracker* é eleito, sua época é incrementada;

- (0,2) Um nó só pode votar em um único candidato por época;
- (0,2) Cada nó reinicia seu temporizador de *tracker* aleatório sempre que um novo *tracker* é eleito ou quando o nó recebe um *heartbeat* válido do *tracker* atual.

(0,4) Envio de *Heartbeat* e Detecção de Falha do *Tracker*

- (0,2) Periodicamente, o *tracker* deve enviar mensagens de *heartbeat* (ex.: a cada 100 ms) indicando que está ativo. Isso permite que os pares monitorem a sua disponibilidade;
- (0,2) Os nós escutam o *heartbeat* enviado pelo *tracker*. Cada nó possui um temporizador de *tracker* **aleatório** (ex: 150–300 ms). Assim, cada nó detectará a falha do *tracker* em tempos diferentes. Uma falha no *tracker* é detectada por um nó ao ocorrer um *timeout*.

(0,3) Atualização do *Tracker*

- (0,15) Sempre que um novo *tracker* for eleito, os nós devem informá-lo os nomes dos arquivos que possui.
- (0,15) Toda alteração local na lista de arquivos de um nó deve ser avisada ao *tracker*.

(0,6) Consulta ao *Tracker* e Transferência de Arquivos

- (0,2) Quando um nó deseja um arquivo, ele consultará o *tracker* para saber quem possui o arquivo desejado. O *tracker* retorna os nós que possuem o arquivo;
- (0,4) Após consultar o *tracker*, o nó escolhe de onde deseja baixar o arquivo e, então, a transferência é feita diretamente entre os pares (P2P);

Observações:

- Desenvolva uma interface com recursos de interação apropriados.
- É obrigatória a defesa da aplicação para obter a nota.
- O desenvolvimento pode ser individual ou em dupla.