

Trabalho Prático 2

A sua tarefa consiste na implementação de um sistema P2P básico, que deve ser organizado como uma arquitetura centralizada, onde o controle de toda a aplicação (lógica e estado) é concentrado em um *web service*. As seguintes funcionalidades devem ser implementadas:

- Os peers devem se registrar no servidor para poderem realizar a troca de arquivos entre peers. Isso deve ser feito por meio de uma REST.
- Durante o registro, cada peer informa seus recursos disponíveis (utilize um diretório com alguns arquivos, e calcule a *hash* de cada um). Para cada arquivo, o peer fornece ao servidor uma string ou o nome do arquivo e sua *hash* (utilize funções como MD5 ou SHA256, calculada sobre o conteúdo de cada arquivo).
- O servidor associa cada recurso em uma estrutura de dados. Cada recurso possui associado o IP do peer onde está o recurso e sua *hash*.
- Os peers podem solicitar uma lista de recursos (nomes dos arquivos / strings de identificação, IPs dos peers que contém os recursos e *hashes*) ao servidor ou um recurso específico. Isso deve ser feito por meio de uma REST.
- Ao solicitar um recurso ao servidor, o peer recebe a informação sobre sua localização (outro peer) e deve então realizar essa comunicação diretamente com o mesmo. A comunicação entre peers deve ser realizada com *sockets*.
- O servidor é responsável por manter a estrutura da rede de *overlay*. Para isso os peers devem consumir uma REST do servidor a cada 5 segundos. Caso um peer não fizer duas solicitações seguidas a um servidor, o mesmo é removido.

O trabalho deve ser realizado em grupos de 2 ou 3 integrantes. Qualquer linguagem de programação pode ser utilizada (preferencialmente Java).

Para o desenvolvimento, é sugerido que o grupo utilize uma rede com topologia definida, e que sejam realizados testes com um número suficiente de máquinas (pelo menos 3 VM/máquinas e pelo menos 5 terminais).

Juntamente com sua implementação, deve ser entregue um relatório descrevendo a mesma, incluindo os seguintes aspectos: 1) organização do código (por exemplo, descrição das funcionalidades dos módulos ou classes, definição das RESTs e estruturas de dados); 2) utilização dos programas (servidor e peers); 3) demonstração da implementação, apresentando casos de uso.