

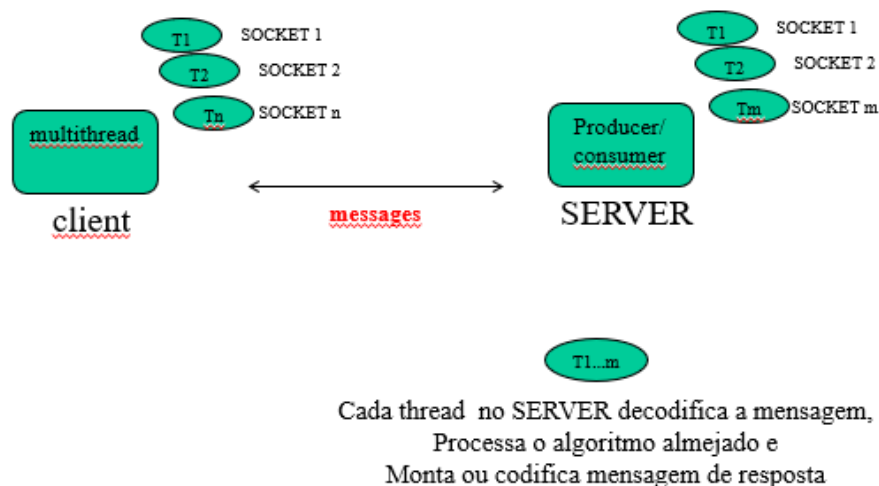
## Trabalho Prático 2 (TP 02) – Disciplina de Sistemas Distribuídos

Neste trabalho o aluno utilizará o conceito de SOCKETS, DATAGRAMS e THREADS para construir mais uma versão de uma solução distribuída. Não há utilização de middlewares neste TP.

Qualquer linguagem será aceita. Não serão aceitas soluções apenas multicore.

TEMA do TP 2.1: Implementar um servidor TCP ou UDP usando o a ideia de produtores e consumidores, onde elegantemente se efetua sincronizações tanto em sistemas paralelos quanto em sistemas distribuídos. O servidor ao receber mensagens se comporta como um produtor, portanto cabe a ele apenas inserir o Socket recebido, por exemplo, no recurso de sockets, indicando ao receber uma mensagem que os consumidores devem ser notificados para começarem a consumir as mensagens e as processar da forma devida. Há necessidade de criação de um cliente TCP ou UDP que estressa o servidor com múltiplas mensagens submetidas e respostas recebidas de tal servidor. Como são múltiplas e concorrentes, o cliente precisa ser multithread e pode ser necessário instalar múltiplos clientes em clusters. Ao final deste trabalho, precisamos ter: i) um servidor usando produtor/consumidor; ii) um servidor SEM produtor/consumidor; iii) um cliente multithread; iv) uma apresentação na forma de seminário, similar ao TP1. Na apresentação, o time de alunos apresentará experimentos com ambos os servidores e como foi desenhada a implementação da solução.

### Sockets



TEMA DO TP 2.2: Implementar um dos seguintes algoritmos de sincronização do livro "Distributed Systems: An Algorithmic Approach", capítulo 7: i) 7.2.1 Lamport's Solution; ii) 7.2.2 Ricart-Agrawala's Solution; iii) 7.2.3 Maekawa's Solution (valerá mais pontos devido à complexidade); iv) 7.3.1 Suzuki-Kasami Algorithm e v) 7.3.2 Raymond's Algorithm (valerá mais pontos uma vez que há necessidade de rede com topologia em árvore). O time de

alunos também poderá implementar um algoritmo que resolve o princípio da causalidade com ordenamento parcial de eventos, ou seja, a implementação de relógios vetoriais, inicialmente apresentado por Fidge em 1988 - <https://zoo.cs.yale.edu/classes/cs426/2012/lab/bib/fidge88timestamps.pdf>. O TP2.2 deverá ser apresentado na forma de seminário, similar ao TP1.

**USAR NUVEM do TP1 PARA IMPLEMENTAR SOLUÇÕES DO TP2. USAR LAB 30 ou 22 com devida consulta ao professor da disciplina.**