



## Sistemas Distribuídos

### Programando um CRUD com sockets

O objetivo deste trabalho é a programação de um sistema do tipo CRUD (*Create, Read, Update, Delete*) para recursos à sua escolha. Pode ser feito na forma de uma agenda (conjunto de pessoas), estoque de algum tipo de empresa, cadastros em geral. Você deverá decidir qual recurso será manipulado pelo programa, e comunicar ao professor para acertar os detalhes.

Os objetivos de aprendizagem são o entendimento do processo de criação de uma estrutura de comunicação (formato de pacote), processo de codificação de informação na troca de mensagens, processo de comunicação entre um cliente e um servidor e o processo de comunicação em rede.

Como descrito, seu sistema terá funções de inserção, leitura (a partir de um parâmetro, que será a chave daquele item), atualização (usando uma chave e passando os demais parâmetros) e remoção (também através de uma chave). Você deverá montar um formato de pacote adequado para a comunicação dos parâmetros, levando em conta os aspectos mencionados em aula (se algum dos campos tem tamanho variável, tipos de dados, etc).

#### Restrições do projeto:

- Os recursos manipulados devem ter pelo menos 5 atributos, sendo que devem possuir tipos variados (texto e número) e um deles deve ser uma chave primária. Caso se utilize um campo de id numérico como chave primária, deve haver ao menos um segundo campo numérico;
  - O id numérico não conta como atributo, de modo que, caso utilizado, o sistema deve manipular 6 atributos no total (os 5 pedidos mais o id);
- A descrição dos atributos deve fazer sentido com o recurso que se deseja manipular;

- Todos os procedimentos devem ser possibilitados através da mesma conexão de socket (como o exemplo de aula), até que o cliente deseje encerrar a conexão;
  - Ou seja, não feche o socket até que o cliente encerre o uso;
  - Não é necessário atender mais de um cliente por vez;
- Com o pacote enviado pelo cliente, seu programa deve ser capaz de identificar o método desejado, ler os parâmetros adequadamente e fazer o processamento pedido;
- Deve ser utilizada uma estrutura de banco de dados para armazenamento das informações. Devem ser armazenados os campos em separado, e não como uma única mensagem;
- Vocês podem usar os exemplos de aula, feitos em Python, ou seguir alguma outra linguagem que desejem;
- Os objetos **não podem ser serializados e enviados**, pois um dos critérios de avaliação é o processo de codificação, e não podem ser utilizadas bibliotecas para conversão dos dados em arquivos como JSON, XML e outros similares. Do mesmo modo, **não utilize bibliotecas que façam a pré-formatação dos dados** (para o caso de ponto flutuante, a função de conversão não conta como pré-formatação). Definam o formato e utilizem funções para realizar a codificação;
  - Para conversão de números de ponto flutuante, utilize o exemplo deixado no arquivo de demonstração;
- Caso tenham dúvida sobre alguma restrição, perguntuem antes de entregar. Após o prazo final, se alguma restrição for violada, descontos serão aplicados.

Sobre a entrega:

- Pode ser feito individualmente ou em dupla;
- Não esqueçam de enviar a descrição do sistema;
- Entreguem junto com o código a estrutura dos pacotes para os métodos pedidos (tanto de requisição quanto de resposta) e qual estrutura de banco de dados foi utilizada (não precisa enviar o banco, mas deixe uma função de criação das tabelas necessárias). Por estrutura do pacote, entenda o formato adotado para cada função, quantidade de bytes e o fluxo de pedido e resposta. Pode ser feito em formato de diagrama ou descrição textual, em um arquivo PDF.