



Roteiro de Aula Prática – Programação com Sockets

DISCIPLINA: DCA3605 – Redes de Computadores
PROFESSOR: Carlos Manuel Dias Viegas

Esta prática consiste em uma introdução ao desenvolvimento de aplicações de rede por meio de programação com Sockets.

- Os requisitos para a realização desta prática são a instalação do Python (versão 3);
- Esta prática consiste em realizar as tarefas descritas abaixo e desenvolver o respectivo código fonte.
- Os códigos fonte desenvolvidos deverão ser enviados em uma tarefa específica no SIGAA durante a aula em que está sendo realizada a atividade prática;
- Esta prática pode ser realizada em duplas.

Nome do discente (1): Ernane Ferreira Rocha Junior

Nome do discente (2): Quélita Miriam Nunes Ferraz

Os códigos fonte (em Python) necessários para iniciar esta prática estão disponíveis na seguinte página:

<https://www.dca.ufrn.br/~viegas/disciplinas/DCA0130/files/Sockets/>

Lembrando que, obrigatoriamente, deverão ser utilizados como base os códigos fonte disponibilizados acima!

Tarefa A: Desenvolver um sistema de transferência de arquivos

Crie uma aplicação utilizando sockets TCP na qual o cliente pode receber e enviar arquivos de/para o servidor.

- a. O cliente deve enviar os comandos:

`upload nome_do_arquivo` ou `download nome_do_arquivo`

- No caso de upload, o arquivo deve ser enviado ao servidor e armazenado localmente.
 - No caso de download, o servidor deve enviar o arquivo solicitado para o cliente.
- b. O servidor deve verificar se o arquivo existe antes de enviá-lo ao cliente.
- c. Trate erros, como tentativa de download de arquivos inexistentes ou comandos inválidos.

SUGESTÃO: Crie dois arquivos de texto simples (por exemplo: `arquivo1.txt` e `arquivo2.txt`) e escreva alguma informação em 1 linha nesses arquivos para que sejam os arquivos que serão transferidos entre cliente e servidor.

DICA: Use o método `open(arquivo.txt)` para abrir o arquivo solicitado e o método `.read()` para ler o seu conteúdo.

Tarefa B: Desenvolver um sistema de mensagens (bate-papo)

Crie uma aplicação utilizando socket UDP na qual o cliente envia mensagens para o servidor e este também possa responder para o cliente.

- O cliente deve digitar um texto e enviar para o servidor.
- O servidor deve receber e também enviar um texto digitado para o cliente.

c. Experimente também utilizar múltiplos clientes “conectados” ao servidor ao mesmo tempo.

DICA: Colocar o cliente em loop para que o socket não seja encerrado após o envio das mensagens.