



UNIVERSIDADE FEDERAL DE ITAJUBÁ – UNIFEI – CAMPUS ITABIRA
CURSO DE ENGENHARIA DA COMPUTAÇÃO - ECO
SISTEMAS PARALELOS E DISTRIBUÍDOS - ECO036
PROF. JULIANO MONTE-MOR

PROJETO PRÁTICO Nº 03

YAN ZINGRA PEREIRA - 31455
DOUGLAS SANTOS DE OLIVEIRA VENÂNCIO - 28614

Projeto 3 - Sockets

Itabira

Julho de 2018

Introdução

O socket é um mecanismo utilizado para a comunicação entre dois programas que são executados numa rede esta comunicação se dá em 2 sentidos tanto server cliente como cliente server. O pacote java.net disponibiliza duas classes para utilizar sockets elas são Socket e ServerSocket, a classe Socket implementa a ligação da comunicação do lado do cliente enquanto a ServerSocket do lado do servidor. A ligação se dá da seguinte forma o servidor com a aplicação Server Socket é executada em conjunto do servidor que liga o servidor a um socket ligado a uma porta específica do lado do cliente aplicação Socket é executada em conjunto com o cliente que solicita ao servidor ao se conectar com ele pela porta do servidor, a ServerSocket então designa uma porta diferente ao cliente e assim servidor e máquina se conectam.

Servidor de arquivos

Para criação do servidor implementado um servidor que tem a função de enviar e receber arquivos requisitados pelo cliente, listar os arquivos que possui, listar os clientes conectados, enviar mensagens em broadcast entre clientes quando requisitado por um cliente. Para comunicação e execução da mesma foi utilizado as classes `ServerSocket` que possui funções para criação da conexão cliente para cada cliente, para cada socket conectado foi criado uma thread para lidar com as conexão com o determinado cliente utilizando a classe `Socket`, o objeto utilizado para comunicação foi uma classe mensagem que possui em sua implementação objetos para enviar e receber mensagens como para o envio de arquivos, utilizando um vetor de bytes para armazenar os arquivos em seu objeto e enviando os entre os sockets ou seja de servidor para cliente ou cliente para o servidor com o fluxo de dados criado por `ObjectStreamInput()` para entrada e `ObjectStreamOutput()` para saída.

A classe mensagem utiliza a interface `Serializable` que transforma os dados de um objeto em um arquivo binário para enviá los entre o servidor e o cliente onde ele é remontado e extraído o objeto no seu destinatário.

Cliente

O cliente foi implementado com as seguintes funções requisitar ao servidor listar dos clientes conectados, listar arquivos no servidor, listar arquivos no cliente, enviar arquivos para o servidor, baixar arquivos do servidor, enviar mensagens em broadcast para todos os clientes conectados e desconectar. Assim como as threads cliente no servidor a programa do cliente foi criado utilizando a classe socket para requisitar e se ligar com servidor, seu fluxo de dados entre o cliente e servidor foi implementado da mesma forma como no servidor com a exceção que o cliente pode apenas requisitar a conexão com o servidor o contrário não pode ser feito.