



UFC – UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE SOBRAL
CURSO DE ENGENHARIA COMPUTAÇÃO
REDES DE COMPUTADORES
PROFESSOR: WENDLEY S. SILVA

SISTEMA DE CHAT EM PYTHON

Ionara Brandão Sant’anna 389107
Francisco Gilson Pereira Almeida Filho 401066

Link para o vídeo: https://www.youtube.com/watch?v=dBgEm_kn5k0

Sobral – CE
2019

SUMÁRIO

1. INTRODUÇÃO.....	3
2. COMO FUNCIONA.....	4
3. COMO UTILIZAR	5
4. REFERÊNCIAS.....	6

1. Introdução

O socket tem a finalidade de formar um elo bidirecional de comunicação entre dois programas, e é desenvolvido especificamente para aplicações Cliente/Servidor, permitindo as aplicações estarem em máquinas diferentes, onde a comunicação é realizada através de protocolos de transportes.

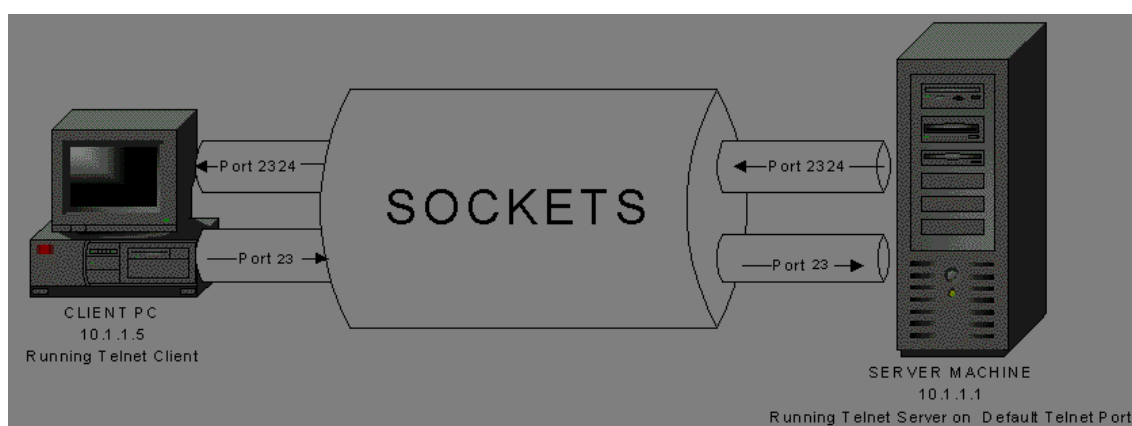
Dessa forma, quando o aplicativo interagir com o software de protocolo, ele deve informar se é um cliente ou um servidor.

Em outras palavras, o socket é uma abstração para endereços de comunicação, que mapeia diretamente a uma porta de transporte, utilizando o TCP ou UDP, mais um endereço de rede, tornando possível identificar unicamente um servidor ou um aplicativo na rede de comunicação IP.

Como já mencionado, o Socket utiliza os protocolos TCP ou o UDP, tendo uma grande diferença entre eles. No Socket UDP, o protocolo UDP não estabelece a conexão, dessa forma, os dados podem ser recebidos fora de ordem ou até mesmo se perderem.

Já O Socket TCP cria uma ligação stream bi-direcional entre os endereços 'hostC:portC' e 'hostS:portS', ou seja, entre uma aplicação cliente em execução na máquina 'hostC' consegue controlar também a porta 'portC' e outra aplicação servidora em execução na máquina 'hostS' conseguindo a visualização da porta 'portS' de 'hostS'. Diferente do UDP, o TCP é orientado a conexão, garantindo que os dados encaminhados pelo cliente para um servidor cheguem ao destino, sem erros.

Imagem 1: Comunicação TCP

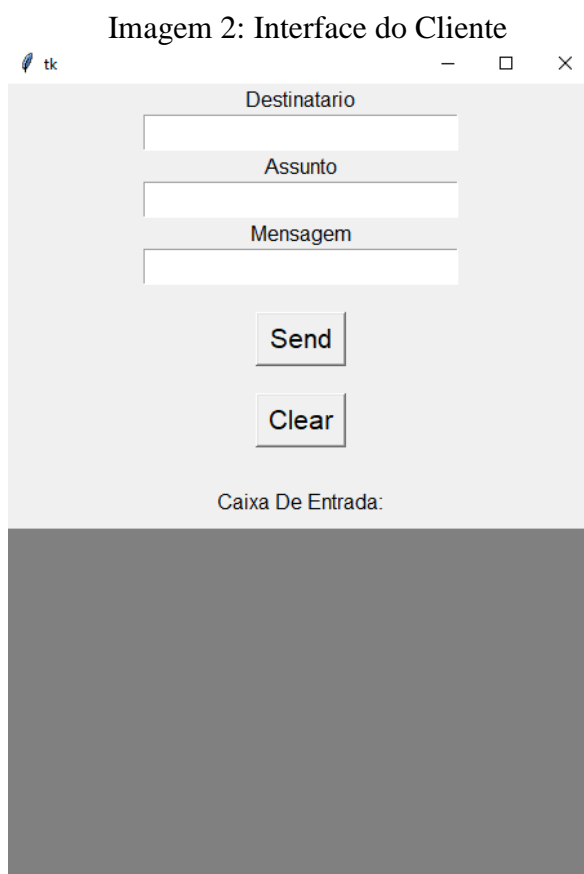


Fonte: <http://www.ortizonline.com/images/sockets.gif>

2. Como Funciona

Para implementação utilizamos a linguagem Python na versão 3. O endereço usado foi “127.0.0.1” – “localhost”, a porta utilizada foi : “44765”. O programa contém 4 arquivos, o servidor e os três clientes. No servidor temos as configurações de socket, onde também foi indicado o servidor e porta. Também, uma função `lidaCiente()`, que faz o tratamento de quantos clientes vão estar se comunicando, e referencia cada cliente com o padrão (Usuário(n)). Além disso, ele faz o tratamento para saber qual usuário vai receber determinada mensagem.

Os códigos dos clientes são idênticos, onde também vão ter as configurações de socket, servidor e porta. Foi usado a biblioteca ‘Tkinter’ para fazer a interface gráfica. No cliente, são indicados 3 campos, Destinatário, Assunto e Mensagem, levando em conta que o assunto pode ser vazio, o que resultara no padrão “Sem Assunto”, no recebimento da mensagem. Além disso, o cliente contém dois botões, um para enviar a mensagem e outro para limpar a caixa de entrada. As mensagens são recebidas em um Label que foi nomeada como ‘Caixa de Entrada’.



Fonte: Autores

3. Como utilizar

O arquivo zip enviado possui esse PDF e 4 arquivos .py. Com uma ide do python ou o próprio interpretador, inicie o servidor e logo após abra os clientes, que será iniciada as interfaces referentes. Logo após, você poderá fazer as comunicações, considerando que o primeiro usuário a entrar será o 'Usuario0', o segundo será o 'Usuario1' e assim por diante.

4. Referências

Sockets - Conceitos e definições básicas. Disponível em:

<<https://tencnolife.blogspot.com/2012/05/sockets-conceitos-e-definicoes-basicas.html>>.

Acesso em: 18 jun. 2019.