RELATÓRIO PARA PROJETO FINAL DE CONECTIVIDADE

Grupo: 9

Integrantes:

Pedro H. K. Vieira

Ana Julia Mazon

Bernardo Carlotto

1.0-Objetivo:

O objetivo desse projeto é usar socket’s para desenvolver uma API que possa promover a comunicação em tempo real entre três ou mais computadores

1.1-Requisitos:

R1 – A API deve poder conectar pelo menos três computadores simultaneamente e promover a comunicação entre eles

R2 – A API deve possibilitar a criação de um CHAT entre todos os computadores conectados ao servidor

R3- Deverá ser possível enviar mensagens broadcast e unicast para os computadores

R4- as mensagens enviadas deverão ser exibidas na tela dos dispositivos em menos de 1 segundo após o envio

R5- Deve haver um comando na API para fazer um computador desconectar-se do servidor

R6- O processo de troca de mensagens deverá ser feito via protocolo TCP

R7- Os dispositivos conectados deverão ser informados toda vez que um novo dispositivo for conectado

R8-Os dispositivos deverão ser identificados por um identificador único , que devera ser solicitado pelo sistema ao usuário antes que ele entre no chat

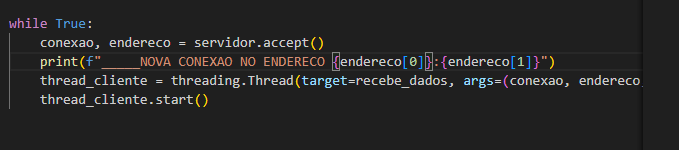
1.2-Implementação:

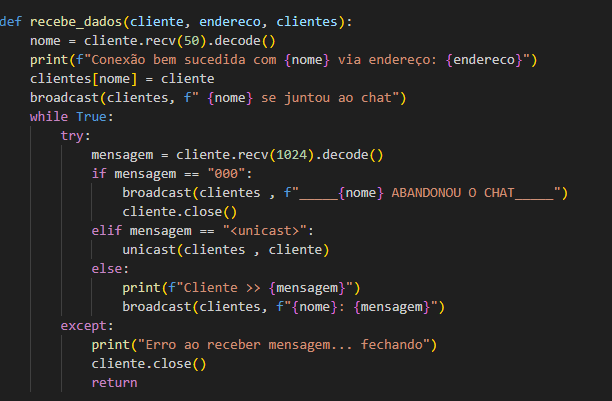
A implementação do projeto foi feita em Python e conta com dois tipos de programas , o código do servidor e do cliente , para a comunicação em tempo real foi usada a biblioteca Socket e para promover um chat mais fluido foi utilizada a biblioteca threading

1.2.1-Funcionamento do servidor:

O servidor foi programado e dividido varias threads , a thread principal é usada para criar novas conexões com o servidor e outra para gerir a comunicação entre os dispositivos.

Quando a thread principal faz a primeira conexão ela entra em um loop que gera uma nova thread para cada novo dispositivo que se conecta ao servidor

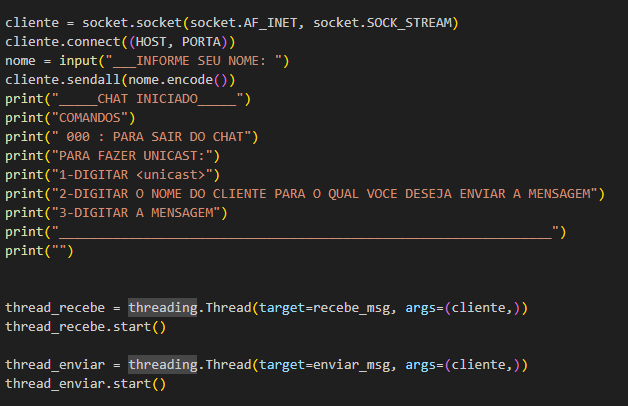


As threads criadas pela thread principal executam a a seguinte função: 

Que aramzena o nome fornecido pelo novo dispositivo no dicionario de clientes , que relaciona o nome do cliente ao Objeto que o representa dentro do servidor , após isso a thread entra num loop de processamento de mensagens do cliente. Se o cliente envia a mensagm 000 , ele está optando por desconectar do servidor e todos os outros dispositivos serão informados disso, se ele envia a mensagm <unicast> , ele está optando por mandar a mensagem apenas para um dos outros dispositivos conectados, se ele enviar ao servidor qualquer outra mensagem , o servidor irá envia-la para todos os outros dispositivos conectados atravez de uma função que simula o broadcast , selecionando todos os objetos de socket no dicionario de clientes , e enviando a mensagem para cada um deles.

1.2.2-Funcionamento dos clientes:

Os clientes estão divididos em três trheads , a thread principal que conecta o cliente ao servidor e outras duas , uma para gerenciar o envio de mensagens ao servidor e outra para receber e processar as mensagens do servidor:



2.0-Link para o projeto:

- O projeto se encontra no seguinte link :

<https://github.com/anajmaz/conectividade>

conta com três arquivos:

servidor.py: executa o servidor

cliente.py: executa o código do cliente

janela.py: executa os dois códigos juntos , para o caso de o computador que executará o executor

2.1-Apresentação:

-A apresentação será feita em sala de aula com um roteador trazido pela equipe