



# CES-35

## Lab 3

### Construindo uma aplicação em Rede

**Professores:** Cesar Augusto Cavalheiro Marcondes  
Cecília de Azevedo Castro César

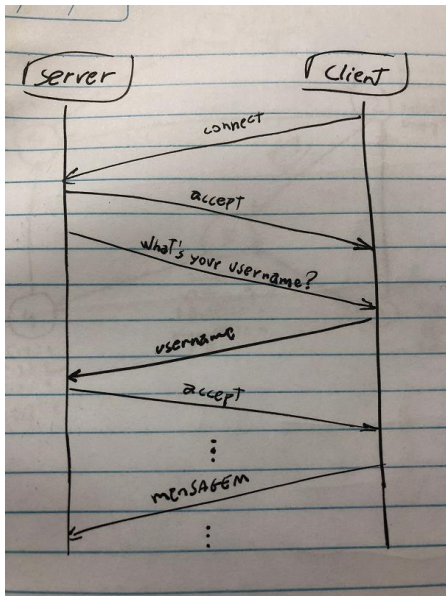
**Alunos:** Dennys Leandro Agostini Rocha  
Felipe Augusto Xavier Uchida

#### **Objetivo**

O objetivo desta experiência é o aprendizado dos princípios de programação em rede, bem como a discussão de um protocolo de comunicação.

## i. Introdução com a solução:

Protocolo:



O programa gira em torno de um jogo de duas pessoas mas que pode ter vários players online. Em azul está a parte do cliente, feito de um lado, enquanto que em vermelho está a parte do servidor, feita do outro lado.

Os clientes pedem conexão ao servidor, que responde. O servidor então pergunta qual nome aquele cliente gostaria de ter cadastrado; o cliente responde e o servidor diz se aceita.

No ambiente, um jogador (cliente) pede para jogar com outro jogador que está na rede, por meio do servidor, e se este aceitar, o jogo se inicia.

Os jogadores enviam suas mensagens para o servidor, já que cada cliente só poderá saber o que está escrevendo (nem do servidor ele pode receber alguma informação que não sua vitória ou derrota). O servidor trata essas mensagens de acordo com a formulação do jogo:

- Para cada jogador, o servidor gera, a cada rodada, um número aleatório de 1 a 19
- Os jogadores enviam um número de 1 a 19 para o servidor
- Para cada cliente, o servidor responde que a rodada foi ganha se:
  - A distância do número escolhido até o número aleatório gerado para o cliente for menor que a distância do número escolhido pelo outro cliente até o mesmo número aleatório
- Pode ocorrer dos dois jogadores ganharem a rodada, uma vez que os casos são analisados separadamente

Caso o cliente ganhe a rodada, este incrementará um contador que se inicia em zero.

No momento em que o contador atinge um limite (definido como igual a 3 no código fonte), os clientes enviam uma mensagem "ACABEI" ao servidor; caso o contador não alcance seu limite, a mensagem enviada é "NAOACABEI".

Se um dos clientes acabar, entao ambos param a partida.

Uma notificacao é transmitida pelo servidor informando aos clientes de sua derrota ou vitória.

## ii. Testes realizados com seus resultados:

Um teste foi feito usando uma mesma máquina com duas máquinas virtuais (clientes) operando com bridge e o host (servidor).

O ip da máquina hospedeira é impresso na tela assim que o servidor é aberto, portanto os clientes devem ajustar o ip host de seus códigos sempre que necessário (caso de desconexão do host com mudança de ip, por exemplo) .

Ao iniciar o servidor, tudo ocorreu perfeitamente. O servidor fica aguardando receber mensagens requisitando conexão dos clientes a todo momento, e quando as recebe, especializa o tratamento para cada cliente.

Quando um cliente começa sua tarefa, é necessário informar o nome de usuário (atentando-se ao fato de não repetir nomes). Em seguida, tem-se a opção de perguntar ao servidor a lista dos usuários conectados (comando “players”), verificar se existe algum desafio para si (comando “status”) ou então desafiar alguém (comando “jogar”).

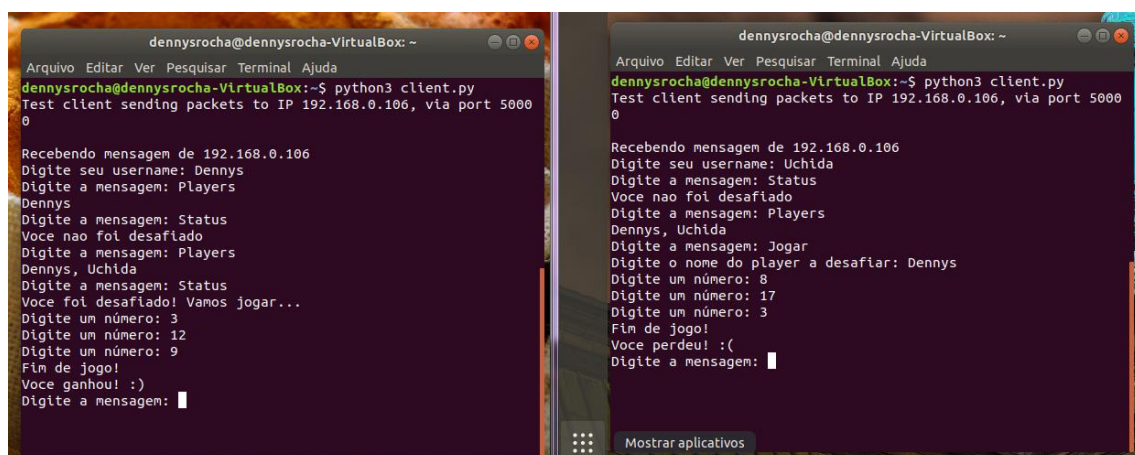
A partir do momento que o jogador desafia alguém, ele fica ocioso aguardando o desafiado aceitar. Quando aceita, inicia-se o jogo.

Da mesma forma, quando um cliente vai verificar seu status e existe algum player que o desafiou, este aceita o desafio de imediato. Então, inicia-se o jogo.

Durante o jogo, os players vão digitando números dentro dos conformes do jogo até que este se acabe.

As Figuras abaixo ilustram o cenário pós-jogo de dois clientes. Um vídeo mostrando a execução dos comandos das Figuras abaixo pode ser acessado no link:

<https://drive.google.com/open?id=1TCWC2Z3j0D3N3iXDYXVaKA8bDapZfLkz>



```
dennysrocha@dennysrocha-VirtualBox: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
dennysrocha@dennysrocha-VirtualBox:~$ python3 client.py  
Test client sending packets to IP 192.168.0.106, via port 5000  
0  
Recebendo mensagem de 192.168.0.106  
Digite seu username: Dennys  
Digite a mensagem: Players  
Dennys  
Digite a mensagem: Status  
Voce nao foi desafiado  
Digite a mensagem: Players  
Dennys, Uchida  
Digite a mensagem: Status  
Voce foi desafiado! Vamos jogar...  
Digite um número: 3  
Digite um número: 12  
Digite um número: 9  
Fim de jogo!  
Voce ganhou! :)  
Digite a mensagem: █  
  
dennysrocha@dennysrocha-VirtualBox: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
dennysrocha@dennysrocha-VirtualBox:~$ python3 client.py  
Test client sending packets to IP 192.168.0.106, via port 5000  
0  
Recebendo mensagem de 192.168.0.106  
Digite seu username: Uchida  
Digite a mensagem: Status  
Voce nao foi desafiado  
Digite a mensagem: Players  
Dennys, Uchida  
Digite a mensagem: Jogar  
Digite o nome do player a desafiar: Dennys  
Digite um número: 8  
Digite um número: 17  
Digite um número: 3  
Fim de jogo!  
Voce perdeu! :(  
Digite a mensagem: █
```

```
ca: Prompt de Comando - py server.py
Recebido: Status
do Dennys
Recebido: Uchida
do ('192.168.0.100', 47398)
Usuário adicionado!
Nova lista de usuários: ['Dennys', 'Uchida']
Recebido: Status
do Uchida
Recebido: Players
do Uchida
Recebido: Players
do Dennys
Recebido: Jogar
do Uchida
Recebido: Dennys
do Uchida
Recebido: Status
do Dennys
O valor randomico gerado para o Dennys eh 2
O valor randomico gerado para o Uchida eh 4
Recebido: 3
do Dennys
Recebido: 8
do Uchida
-----
O usuario Dennys ganhou a rodada
-----
Recebido: NAOACABEI
do Uchida
Recebido: NAOACABEI
do Dennys
O valor randomico gerado para o Uchida eh 5
O valor randomico gerado para o Dennys eh 2
Recebido: 12
do Dennys
Recebido: 17
do Uchida
-----
O usuario Dennys ganhou a rodada
-----
Recebido: NAOACABEI
do Uchida
Recebido: NAOACABEI
do Dennys
O valor randomico gerado para o Uchida eh 1
O valor randomico gerado para o Dennys eh 14
Recebido: 9
do Dennys
Recebido: 3
do Uchida
-----
O usuario Uchida ganhou a rodada
O usuario Dennys ganhou a rodada
-----
Recebido: NAOACABEI
do Uchida
Recebido: ACABEI
do Dennys
```

### **iii. Seção de possíveis melhorias e comentários:**

O código fica um tanto quanto frágil na troca de mensagens entre cliente e servidor se mudar o modo de entrega e recebimento das mensagens. Se não houver garantia de entrega de mensagens, ocorre deadlock em todas as transmissões de dados, o que não ocorre devido ao tipo de conexão feita no *connect()*.

O jogo pode ser melhorado de diversas formas, mas não vem ao caso modificá-lo.

## iv. Código fonte:

### Código server.py (primeiro lado da comunicacao)

```
#!/usr/bin/env python3

import socket
import sys
import numpy as np
import threading
import os, re
from random import randint
import time

port = 50000
backlog = 5
size = 1024
threads = []
usernames = list()
desafiados = [("example1", "example2")]
acabou = False
ganhador = []

# pegando meu NIC address
addresses = os.popen('IPCONFIG | FINDSTR /R "Ethernet adapter Local Area Connection .* Address.*[0-9][0-9]*\.[0-9][0-9]*\.[0-9][0-9]*\.[0-9][0-9]*')
nic = re.search(r'\b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b', addresses.read()).group()

host = nic
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((host, port))
s.listen(backlog)

print ("Servidor aberto em", nic, "\n")

def manager(client, address):
    """thread manager function"""

    # jogo: a cada rodada eh jogada uma moeda que resulta em um valor de 1 a 19; o player ganha se a distancia dele ate o valor gerado for menor ou igual a distancia
    # do outro ate o mesmo valor; é improvavel mas pode acontecer dos dois players ganharem a partida (falta ajustar esse caso)
    def rochida(desafiado):
        global acabou
        global usernames
        global ganhador
        meuNome = "".join([item[0] for item in usernames if item[2] == address])
        while not acabou:
            if acabou:
                break;
            randomico = randint(1,19) # convencao: 0 para "MENOR" e 1 para "MAIOR"
            time.sleep(0.5)
            print("O valor randomico gerado para o", meuNome, "eh", randomico)
            meuValor = client.recv(size).decode("utf-8") # valor numero digitado pelo usuario

            print("Recebido:", meuValor, "\ndo", meuNome)
            usernames = [item for item in usernames if item[0] != meuNome]
            usernames.append((meuNome, meuValor, address))

        while True:
            if ([item for item in usernames if (item[0] == desafiado and item[1] == -1)] == []): # verificar se a tupla ainda nao foi atualizada com o valor do desafiado
                break;
            valor = "".join([str(item[1]) for item in usernames if item[0] == desafiado]) # pega o valor inserido pelo desafiado

            # PROCESSO DE AVALIACAO PARA VER QUEM GANHOU A RODADA
            if abs(int(meuValor)-randomico) <= abs(int(valor)-randomico):
                print("-----")
                print("O usuario", meuNome, "ganhou a rodada")
                print("-----")
                myMessage = "GANHOURODADA"
                time.sleep(1)
            else:
                myMessage = "PERDEURODADA"
            client.send(myMessage.encode("utf-8")) # diz se perdeu ou ganhou a rodada

            usernames = [item for item in usernames if item[0] != meuNome]
            usernames.append((meuNome, -1, address))

    # PROCESSO DE AVALIACAO PARA VER SE O PLAYER ACABOU
    data = client.recv(size).decode("utf-8")
```

```

        time.sleep(1)
        print("Recebido:", data, "\ndo", meuNome)
        if data=="ACABEI":
            acabou = True
            ganhador = meuNome
        else:
            myMessage = "CONTINUAR"
            time.sleep(1)
            if not acabou:
                client.send(myMessage.encode("utf-8"))

    myMessage = "ACABOU"
    client.send(myMessage.encode("utf-8"))
    if ganhador == meuNome:
        client.send(meuNome.encode("utf-8"))
    else:
        client.send(desafiado.encode("utf-8"))

global desafiados
print("Conexao com", address, "estabelecida.")

text = "Digite seu username: "
client.send(text.encode("utf-8"))
data = client.recv(size).decode("utf-8")
print("Recebido:", data, "\ndo", address)
accepted = "Recebido!"
client.send(accepted.encode("utf-8"))
usernames.append((data,-1,address)) # (username, valorDigitado)

print("Usuário adicionado! \nNova lista de usuários:", [item[0] for item in usernames])
meuNome = "".join([item[0] for item in usernames if item[2] == address])

while True:
    data = client.recv(size).decode("utf-8")
    print("Recebido:", data, "\ndo", meuNome)
    if data:

        # O CLIENTE QUER SABER QUANTOS PLAYERS EXISTEM
        if data.upper()=="PLAYERS":
            text = ", ".join([x[0] for x in usernames])
            client.send(text.encode("utf-8"))

        # O CLIENTE QUER JOGAR COM ALGUEM
        elif data.upper()=="JOGAR":
            text = "DESAFIOU"
            client.send(text.encode("utf-8")) # envia o texto para entrar na
            # secao de desafio no cliente
            data = client.recv(size).decode("utf-8") # recebendo o nome do
            # desafiado
            print("Recebido:", data, "\ndo", meuNome)
            desafiados.append((data, meuNome)) # coloca o nome do desafiado na
            # lista "desafiados"
            while(any(data in item[0] for item in desafiados)):
                pass
                # aguarda até o desafiado verificar que foi desafiado e
            # aceitar o desafio
            text = "ACEITO"
            client.send(text.encode("utf-8"))
            rochida(data)

        # O CLIENTE QUER SABER SE TEM ALGUEM QUERENDO JOGAR COM ELE
        elif data.upper()=="STATUS":
            if([item[0] for item in desafiados if item[0] == meuNome] != []):
                text = "DESAFIADO" # voce esta na lista de pessoas
                # desafiadas: foi DESAFIADO!
                # remova seu nome dos desafiados!
                desafiador = "".join([item[1] for item in desafiados if
                item[0] == meuNome]) # encontra o nome do desafiador
                desafiados = [item for item in desafiados if item[0] !=
                meuNome]
                text = "DESAFIADO"
                client.send(text.encode("utf-8"))
                rochida(desafiador)
            else:
                text = "Voce nao foi desafiado"
                client.send(text.encode("utf-8"))

        # ALGUMA MENSAGEM QUE NAO ATENDA AOS CRITERIOS ANTERIORES
        else:
            text = "Servidor recebeu a mensagem."
            client.send(text.encode("utf-8"))

```



```

while True:
    client, address = s.accept()
    t = threading.Thread(target=manager, args=(client, address))
    threads.append(t)
    t.start()
sys.exit()

```

## Código client.py (segundo lado da comunicacao)

```

#!/usr/bin/env python3

import sys
import socket

host = '192.168.0.106'
port = 50000
size = 1024
pontos = 0

print ("Test client sending packets to IP {0}, via port {1}\n".format(host, port))

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((host,port))
data = s.recv(size).decode("utf-8")
print("Recebendo mensagem de", host)
print(data, end='')
myUsername = input()
s.send(myUsername.encode("utf-8"))
accepted = s.recv(size).decode("utf-8")

def jogar():
    global pontos
    global myUsername
    while pontos<3:
        myMessage = input("Digite um número: ")
        s.send(myMessage.encode("utf-8"))
        data = s.recv(size).decode("utf-8") # recebo a mensagem se ganhei ou perdi a rodada
        if data=="GANHOURODADA": # ganhou a rodada
            pontos=pontos+1

        # MANDO UMA MENSAGEM DIZENDO SE ALCANCEI 19 PTS
        if pontos>=3:
            myMessage = "ACABEI"
        else:
            myMessage = "NAOACABEI"
            s.send(myMessage.encode("utf-8"))

        data = s.recv(size).decode("utf-8") # recebo a mensagem se devo continuar ou se
acabou a partida
        if data=="ACABOU":
            break;
        data = s.recv(size).decode("utf-8") # quem ganhou
        if data==myUsername:
            print("Fim de jogo!\nVoce ganhou! :)")
        else:
            print("Fim de jogo!\nVoce perdeu! :)")
        pontos=0

    while True:

        # ESCOLHA QUAL O TIPO DE MENSAGEM QUER ENVIAR AO SERVIDOR
        myMessage = input("Digite a mensagem: ")
        s.send(myMessage.encode("utf-8"))

        # O SERVIDOR RESPONDEU À SUA MENSAGEM
        data = s.recv(size).decode("utf-8")

        # sua mensagem foi "JOGAR" e o servidor retornou "DESAFIOU"
        if data=="DESAFIOU":
            myMessage = input("Digite o nome do player a desafiar: ")
            s.send(myMessage.encode("utf-8"))
            data = s.recv(size).decode("utf-8")
            if data=="ACEITO": # o desafio foi aceito pelo adversario
                jogar()

```

```
# sua mensagem foi "STATUS" e o servidor disse que voce foi "DESAFIADO"
elif data=="DESAFIADO":
    print("Voce foi desafiado! Vamos jogar...")
    jogar()
else:
    print(data)
sys.exit()
```