

# Relatório da disciplina Laboratório de Redes

Universidade Federal Rural do Rio de Janeiro

Semestre:2019-1

Aluno: Fabio Luiz Silva Nogueira

Matricula: 201278012-1

# Sumário

Descrição do Jogo .....	3
Manual de Uso.....	3
Diagrama de Sequência .....	4
Código Fonte.....	5
Funcionamento.....	8
Referências .....	10

# Descrição do Jogo

O jogo trata-se de uma disputa de adivinhação entre dois jogadores. A adivinhação trata-se de um valor numérico aleatório entre 1 e 10. Para os jogadores disputarem no sistema, eles acessam a um servidor no qual fica armazenado os dados das partidas. Após a jogada, o servidor informa se o jogador acertou, caso não acerte, o próximo jogador realiza o seu palpite e assim sucessivamente, até alguém acertar. Além dos dois jogadores, o sistema contém um usuário Espectador, no qual pode acompanhar a partida sem interferir na mesma.

Após algum dos jogadores ganharem a partida, o servidor informa ao ganhador que ele ganhou, informa ao outro jogador que ele perdeu e informa para o espectador qual dos dois jogadores ganhou. Logo após o resultado, a conexão entre esses sistemas termina.

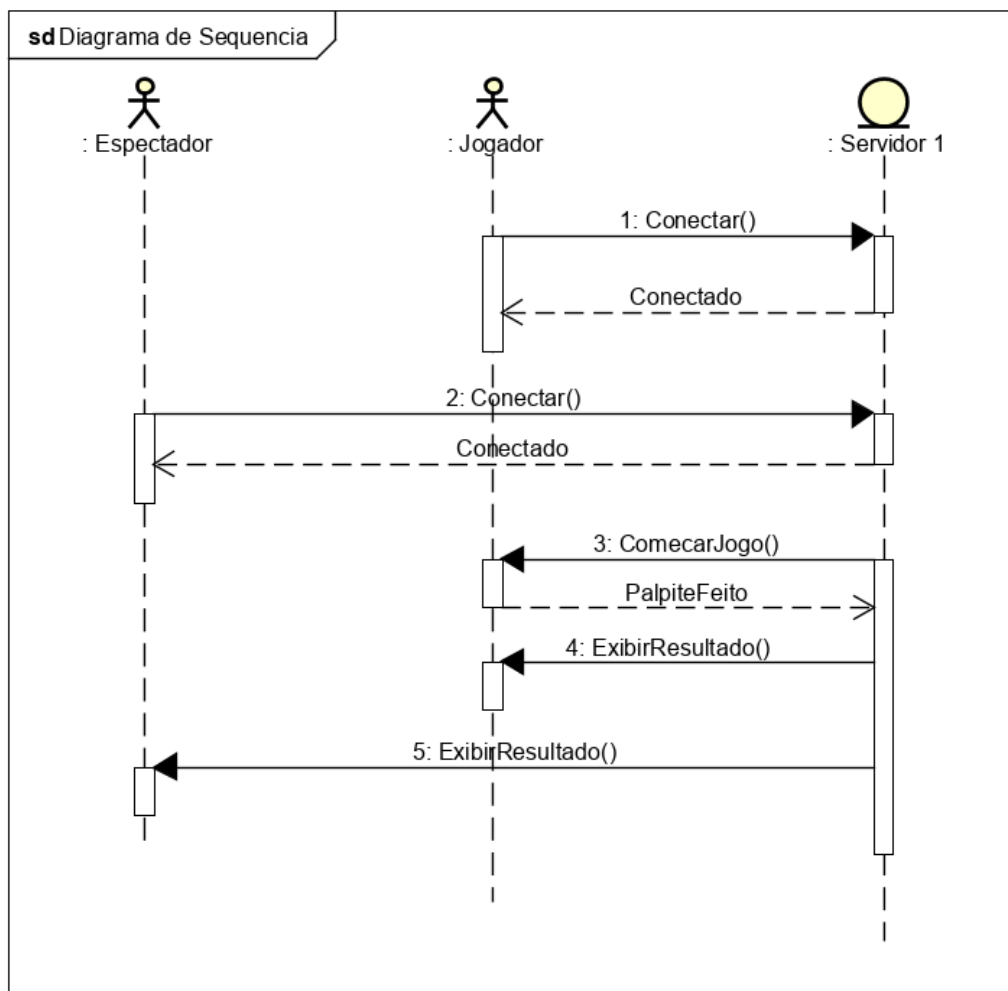
## Manual de Uso

Para poder utilizar o jogo, é preciso seguir algumas etapas à risca para que não ocorram erros. Seguem as etapas para a utilização de forma correta:

1. Servidor: Abrir em um terminal o arquivo "Servidor.py"
2. Jogador 1: Abrir em um terminal o arquivo "Cliente.py"
3. Jogador 2: Abrir em um terminal o arquivo "Cliente.py"
4. Espectador: Abrir em um terminal o arquivo "Espectador.py"

Logo após todos os usuários estiverem conectados, o servidor informará a todos que o jogo começou. O jogador só deverá realizar o palpite quando o servidor avisar que está na sua vez de jogar. No término do jogo, caso se deseje jogar novamente, deverá ser fechado todos os terminais e recomeçar o processo novamente.

## Diagrama de Sequência



# Código Fonte

O jogo está dividido em três arquivos, Seguem eles:

## Servidor:

```
servidor.py
1 import socket
2 import random
3 import sys
4
5 host = 'localhost'
6 port = 8880
7
8 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9 print("Socket criado")
10
11 try:
12     s.bind((host,port))
13 except socket.error:
14     print ("Binding falhou")
15     sys.exit
16
17 print("Socket foi designado")
18
19 s.listen(3)
20
21 print("Socket esta pronto")
22
23
24 conn1, addr1 = s.accept()
25 print("Conectado com " + addr1[0] + ":" + str(addr1[1]))
26 print("Esperando o jogador 2")
27 conn2, addr2 = s.accept()
28 print("Conectado com " + addr2[0] + ":" + str(addr2[1]))
29 print("Aguardando Espectador")
30 conn3, addr3 = s.accept()
31 print("Conectado com " + addr3[0] + ":" + str(addr3[1]))
32 numero = random.randint(1,10)
33 print("Numero sorteado: ", numero)
34 terminou=0
35
36 msgmEspect=("Todos os jogadores ja estao conectados ao servidor, o jogo comecou!").encode()
37 conn3.send(msgmEspect)
38
39
40 while True:
41     print("----Rodada do Jogador 1----")
42     jogadaUm = '1'
43     conn1.send(jogadaUm.encode())
44     palpite = int(conn1.recv(1024))
45     print("Palpite do Jogador 1: ", palpite)
46
47     msgmEspect = ("Palpite do Jogador 1: " + str(palpite)).encode()
48     conn3.send(msgmEspect)
```

```

49
50
51
52 ▼ if(palpite==numero):
53     msgGanhou='1'
54     conn1.send(msgGanhou.encode())
55     print("Jogador 1 GANHOU!!!!")
56     msgmEspect = ("Jogador 1 GANHOU!!!!")
57     conn3.send(msgmEspect.encode())
58
59
60     break
61
62     msgGanhou='0'
63     conn1.send(msgGanhou.encode())
64     jogadaUm = '0'
65     conn1.send(jogadaUm.encode())
66
67
68     print("----Rodada do Jogador 2----")
69     jogadaDois = '1'
70     conn2.send(jogadaDois.encode())
71     palpite = int(conn2.recv(1024))
72
73
74
75     print("Palpite do Jogador 2: ", palpite)
76
77
78     msgmEspect = ("Palpite do Jogador 2: " + str(palpite)).encode()
79     conn3.send(msgmEspect)
80
81 ▼ if(palpite==numero):
82     msgGanhou='1'
83     conn2.send(msgGanhou.encode())
84     print("Jogador 2 GANHOU!!!!")
85
86     msgmEspect = ("Jogador 2 GANHOU!!!!")
87     conn3.send(msgmEspect.encode())
88
89     break
90     conn2.send(msgGanhou.encode())
91     jogadaDois='0'
92     conn2.send(jogadaDois.encode())
93
94
95 s.close()
96

```

## Cliente:

```
cliente.py x
1 import socket
2 host = 'localhost'      # Endereco IP do Servidor
3 port = 8880             # Porta que o Servidor esta
4 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 dest = (host, port)
6
7 s.connect(dest)
8 print("Cliente.py: Conectado ao servidor")
9
10 while True:
11     try:
12         jogadaDisponivel = int(s.recv(1024))
13     except:
14         print("Voce perdeu :(")
15         break
16
17     if(jogadaDisponivel == 1):
18         print("Esta na sua rodada, Digite um palpite")
19         msg= input()
20         s.send(msg.encode())
21         jogadaDisponivel = 0
22         msgGanhou = int(s.recv(1024))
23         if (msgGanhou == 1):
24             print("Parabens! Voce Ganhou!!")
25             break
26 s.close()
27
```

## Espectador:

```
espectador.py x
1 import socket
2 import sys
3 host = 'localhost'      # Endereco IP do Servidor
4 port = 8880             # Porta que o Servidor esta
5 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6 dest = (host, port)
7
8 s.connect(dest)
9 print("Espectador: Conectado ao servidor")
10
11 ganhouUm= "Jogador 1 GANHOU!!!"
12 ganhouDois= "Jogador 2 GANHOU!!!"
13
14 while True:
15     msgm = s.recv(1024).decode()
16
17     if ((msgm==ganhouUm) or (msgm==ganhouDois)):
18         print(msgm)
19         break
20
21     print(msgm)
22 s.close
23
```

# Funcionamento

O jogo foi implementado usando a linguagem Python 3 com o API de Sockets e protocolos TCP. Para seu funcionamento correto, o jogo foi separado em 3 arquivos diferentes.

## Servidor

É o primeiro programa a ser executado. Ele funciona da seguinte forma:

- Importa as bibliotecas necessárias para sua execução (socket, random e sys). Escolhe um host e uma porta, no caso, localhost:8888.



- Cria um socket, e executa as funções sockets Bind e Listen.
- Armazena o Jogador 1 em conn1, Jogador 2 em conn2 e Espectador em conn3. Logo após informa o endereço e porta dos mesmos e que o jogo está pronto para começar
- Enquanto verdade, executa o código; Informa ao jogador 1 que está na vez dele através do envio do string '1' e recebe de volta essa string e converte para inteiro. Se o palpite do jogador 1 estiver correto: Envia ao Jogador 1 que ele ganhou, que o 2 perdeu e ao espectador que o 1 ganhou.
- Caso o Jogador 1 não acerte o primeiro palpite, o algoritmo segue o mesmo raciocínio do passo anterior com o Jogador 2.
- O algoritmo só vai parar de executar quando algum dos dois jogadores ganharem e no caso, entrar em um dos "ifs" e parar com o comando break.

### Cliente

Logo após o servidor está sendo executado, dois Jogadores devem entrar no jogo, pelo processo cliente.py, que funciona da seguinte forma:

- Importa a biblioteca socket e informa o Host/Porta no qual o cliente deve se conectar, no caso de conectar-se com sucesso, é informado na tela
- Enquanto verdade: Tenta: se está ou não em sua rodada, prossegue com o código, caso dê erro (except), significa que o outro jogador ganhou e o atual perdeu.
- Se estiver na partida do mesmo: Informa que está na vez e pede um palpite, captura o palpite no teclado e transforma numa string a ser enviada ao servidor. Se o servidor retornar um int 1, significa que o mesmo acertou o palpite

e o cliente notifica que ele ganhou e termina o código através do break e s.close().

### Espectador

Logo após os clientes estarem conectados, o espectador se conecta ao servidor para assistir a partida, que funciona da seguinte forma:

- Importa a biblioteca socket e informa o Host/Porta no qual o espectador deve se conectar, no caso de conectar-se com sucesso, é informado na tela
- O servidor envia mensagens do que está ocorrendo para o espectador, que recebe através da variável msgm e o mesmo imprime essas mensagens em tela. Caso uma dessas mensagens seja de que algum dos jogadores ganhou, o programa imprime quem ganhou e termina sua execução através do break e s.close().

## Referências

Documentação de Sockets Python:

<https://docs.python.org/3/library/socket.html>

Guia Socket Básico do site Python Brasil:

<https://wiki.python.org.br/SocketBasico>

Video - How To Create a Multithreaded Server and Allow Many Clients - Python Networking Using TCP:

<https://www.youtube.com/watch?v=ZwxTGGEx-1w>