

# Relatório 3º Trabalho de implementação

## Redes 1

**Professor: Luiz Carlos Pessoa Albini**

Alunos: Erick Eckermann Cardoso    GRR20186075

Dante Eleutério dos Santos    GRR20206686

### 1. Descrição Geral

Relatório técnico a respeito da implementação do 3º Trabalho da disciplina. Consiste num jogo de dados criado em sala, implementado sobre uma rede local em anel com 4 máquinas e passagem de bastão entre elas. Quem possui o bastão define uma combinação a apostar e um valor inicial, e as mensagens rodam pela rede em anel. As regras das jogadas foram criadas em sala e são de conhecimento geral, portanto dispensei descrição aqui.

#### 1.1 Linguagem e bibliotecas utilizadas

O trabalho foi implementado em Python 3. A comunicação entre as máquinas é feita utilizando o **import socket** com os parâmetros do SOCK\_DGRAM(Socket Data-Grama). As quatro máquinas utilizam o mesmo ip de rede local: "127.0.0.1". Cada máquina possui uma porta de comunicação para recebimento de mensagens e outra para saída. As portas escolhidas foram as seguintes:

- **Máquina A:**  
Porta receive = 6454  
Porta sent = 6451
- **Máquina B:**  
Porta receive = 6451  
Porta sent = 6452
- **Máquina C:**  
Porta receive = 6452  
Porta sent = 6453
- **Máquina D:**  
Porta receive = 6453  
Porta sent = 6454

Os imports de bibliotecas utilizados foram:

1. **import re**
2. **import socket**
3. **import numpy**
4. **import struct**
5. **import comp\_functions**
6. **import random**

## 1.2 Mensagem e formato da mensagem

A comunicação na rede em anel é feita por dois tipos de mensagem: mensagem padrão e a mensagem do bastão. Quem controla as jogadas e é designado como origem é quem possui o bastão. Todas as outras mensagens, além do bastão, são enviadas no mesmo formato.

### 1.2.1 Formato da mensagem padrão

1	2	3	4	5	6
Marc. Início	Char Origem	Char Holder	Inteiro	Inteiro	Paridade Vertical
8 bits	8 bits	8 bits	32 bits	32 bits	32 bits

O enquadramento da mensagem foi separado da seguinte maneira: Campos 1 e 6 correspondem ao marcador de início e detecção de erros, respectivamente, e campos 2, 3, 4, 5 correspondem ao campo de dados, guardando as informações necessárias para realizar as partidas do jogo. A mensagem possui esse formato fixo. O programa recebe as mensagens de 100 e 100 bytes.

Foi utilizado o método `struct.pack` para binarizar a mensagem e enviar. Esse método empacota vários elementos numa sequência de bits indicando a ordem dos elementos empacotados. Para receber, foi utilizado o método `struct.unpack`, que recebe a sequência de bits e interpreta eles de acordo com a ordem pré fixada, retornando uma tupla com os elementos. Após receber a mensagem, é feito o teste para ver se o marcador de início corresponde com a sequência desejada e confere se não houve erro na mensagem.

Segue uma explicação mais detalhada de cada campo:

- 1 - Campo do marcador de início da mensagem. A sequência escolhida de bits foi 01001011, correspondendo ao char "K".
- 2 - Guarda um char que indica sempre quem é a origem no momento, ou seja, quem possui o bastão.
- 3 - Guarda um char que indica o holder, ou seja, quem tomou a aposta e irá realizar as jogadas de dados.
- 4- Inteiro que pode guardar uma indicação da aposta ou o resultado das jogadas, em determinada parte.
- 5 - Inteiro que pode guardar o valor da aposta ou o saldo do holder após a jogada de dados.
- 6 - Paridade vertical com os dois inteiros que são enviados no campo 4 e 5. Realiza uma operação XOR com eles e guarda o resultado neste campo.

### 1.2.2 Formato da mensagem e lógica do bastão

O bastão é simbolizado por uma variável global de cada jogador. Ao se iniciar o jogo, o player A seta a variável, denominada "bastao", como True e os demais como False. Isso definirá se ele irá atuar como origem ou não na rodada. No final de cada rodada, a origem manda uma mensagem pela porta de saída com um único elemento sendo a string "bastao"

e seta seu bastão como False. Os demais conferem se a origem é quem envia mensagens para ele, e se for, fica aguardando o bastão, se não simplesmente reinicia a jogada. Ao receber a mensagem, quem estava aguardando confere se o elemento é a string “bastao” e se for, muda sua variável como True e inicia uma nova jogada, agora como origem.

## 2. Tomada de decisões do jogo

Cada jogador inicia com 5 fichas, e o jogo começa. Quem define a aposta é a origem, definindo qual a combinação e o valor inicial da aposta, e os outros podem decidir se tomam a aposta para eles ou não, aumentando o valor da aposta em 1 caso decidam tomar.

O último que decidiu tomar a aposta se torna o Holder e realiza as jogadas de dados. As jogadas foram implementadas como específicas em sala. Caso o holder tenha vencido, o valor da aposta é somado ao seu saldo. Caso contrário o valor da aposta é descontado de seu saldo.

Caso seu saldo se torne menor ou igual a zero, o jogo é encerrado, por meio de uma mensagem que circula a rede e avisa que o saldo do holder se tornou nulo ou negativo para cada jogador, encerrando o programa. Caso contrário, uma nova rodada é iniciada e o bastão é passado.

### 2.1 Tabela de pontuação

Combinação	Fichas
Par	1
Trio	2
Dois Pares	3
Full House	4
Sequência Baixa	5
Sequência Alta	6
Quadra	8
Quinteto	10