



Faculdades de Campinas  
Curso de Engenharia de Computação

SISTEMAS DISTRIBUÍDOS

# RELATÓRIO DO TRABALHO

## SNAKE GAME

**Aluno:**

LÍVIA SPILLER

RUBIA HELENA ARCHANJO

**RA:**

201910647

201910663

Campinas  
Junho de 2022

## SUMÁRIO

1. DESCRIÇÃO DO PROBLEMA/OBJETIVOS ESPECÍFICOS .....	3
2. MODELO DA SOLUÇÃO .....	4
3. LAYOUT DE TELAS .....	5
4. FORMATO DAS MENSAGENS TROCADAS.....	8
5. REQUISITOS E PROCEDIMENTOS PARA INSTALAÇÃO E TESTE .....	9

## 1. DESCRIÇÃO DO PROBLEMA/OBJETIVOS ESPECÍFICOS

O objetivo principal deste trabalho é desenvolver uma aplicação que envolva, pelo menos, a comunicação de dois clientes através de um servidor. O servidor deve agir como entidade mediadora e repositório de dados comuns aos clientes, de acordo com o problema resolvido. Devem ser utilizados delimitadores de campos nas mensagens trocadas entre clientes e servidor e, conseqüentemente, implementado o byte stuffing. Além de serem utilizados e adaptados os códigos exemplos fornecidos durante as aulas da disciplina, os clientes e os servidores devem ser multi-threaded e podem ser desenvolvidos em C ou em Java.

Em razão disso, foi desenvolvido um jogo popularmente conhecido como o da “cobrinha” ou “*snake game*”, na qual há a interação dessas partes envolvidas: dois clientes, se comunicando através de um servidor. A finalidade desse jogo é fazer a “cobrinha” comer a maior quantidade possível de maçãs, sem colidir com a tela ou com o próprio corpo. No final, vence o jogador que atingir o maior *score* durante a partida. Inicialmente, as duas cobrinhas não poderiam se colidir entre elas, porém no decorrer do desenvolvimento, achou-se mais interessante ter o objetivo de contabilizar somente a pontuação final de cada uma, baseada na quantidade de maçãs comidas pelas cobrinhas, a fim de tornar o jogo mais dinâmico e contínuo.

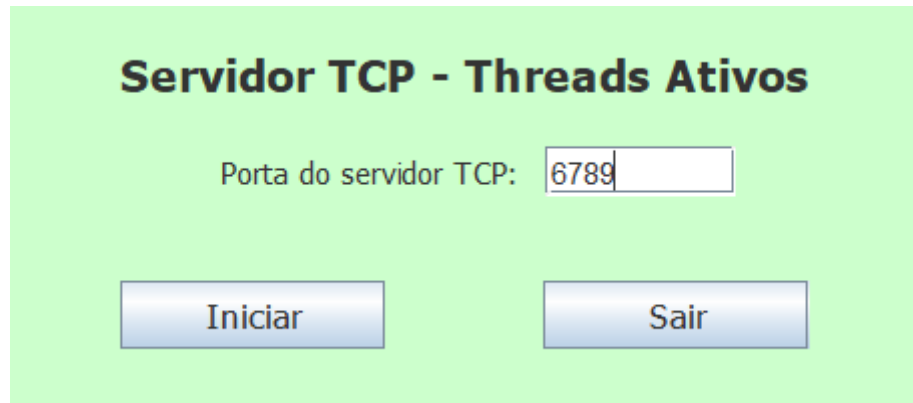
## 2. MODELO DA SOLUÇÃO

Neste modelo, o servidor é responsável por realizar os comandos e mandar essas respostas para os clientes. Os clientes, por sua vez, são responsáveis por receber essas respostas e enviar mensagens para o servidor que servirá como um intermédio entre a comunicação desses dois clientes. As classes presentes na aplicação são:

- **SnakeGame**: classe principal, responsável por chamar a classe do servidor;
- **Servidor**: responsável por inicializar a conexão com o servidor TCP;
- **Jogar**: responsável por iniciar e desconectar a conexão dos clientes, setar a cor de fundo do painel, desenhar na tela as cobrinhas, as maçãs, o game over e o *score*;
- **TCPClientHandler**: responsável por receber as informações do servidor para o cliente;
- **TCPClientMain**: responsável por conectar e enviar as mensagens do cliente ao servidor;
- **TCPServerAtivosHandler**: responsável por enviar as mensagens do servidor ao cliente e receber as mensagens dos clientes para o servidor;
- **TCPServerAtivosMain**: responsável por adicionar e remover um cliente, definir as variáveis e os calcular os comandos do jogo;
- **TCPServerConnection**: responsável por gerenciar as conexões com servidor.

### 3. LAYOUT DE TELAS

Figura 1 - Conexão com o servidor.



Fonte: autoria própria.

Figura 2 - Tela inicial e de conexão de um dos clientes.



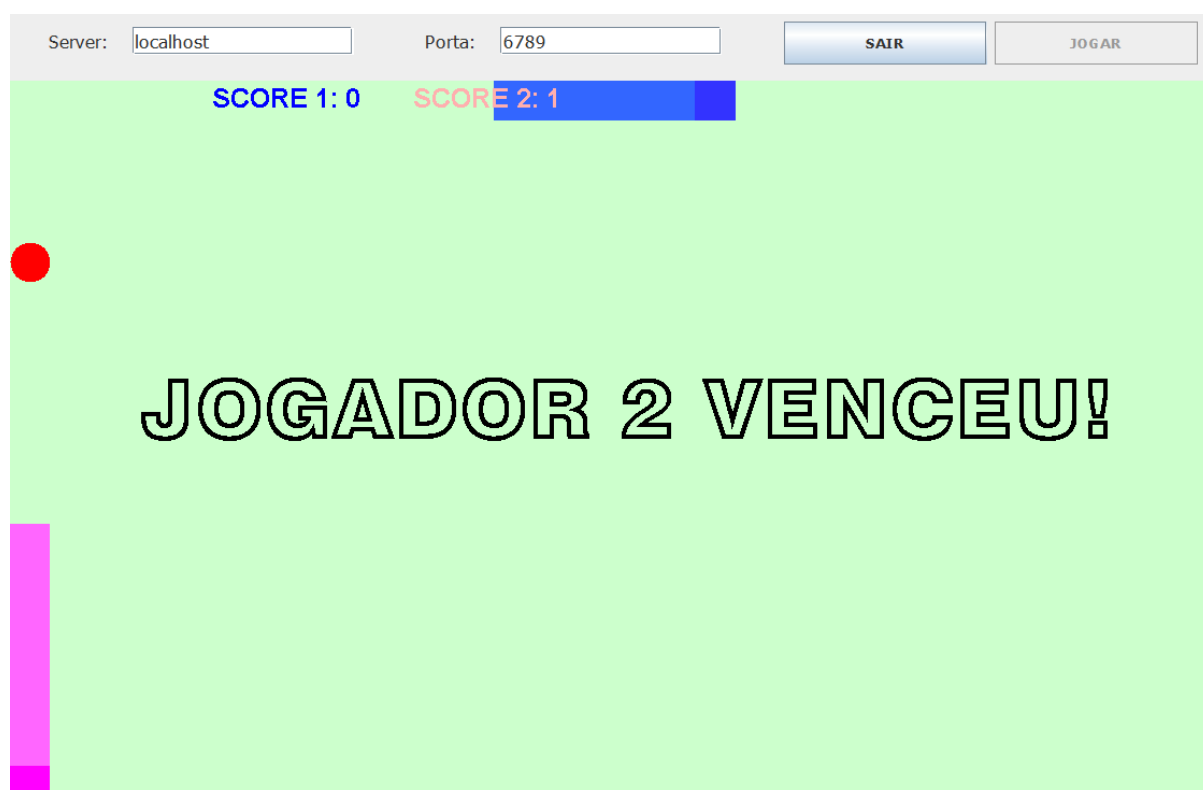
Fonte: autoria própria.

Figura 3 – Demonstração de um empate (score 1 = score 2)



Fonte: autoria própria.

Figura 4 – Demonstração de uma vitória (score 2 > score 1)



Fonte: autoria própria.

#### **4. FORMATO DAS MENSAGENS TROCADAS**

Foi criado um método `enviarMensagem()` na classe `TCPServerAtivosHandler` para enviar as mensagens do servidor aos clientes. E um método `lerMensagem()` na classe `TCPClientHandler` para ler essas mensagens que foram enviadas. As mensagens enviadas estão convertidas em um formato `String`, enquanto as mensagens que são lidas pelos clientes são convertidas em valores inteiros (`int` e `int[][]`) ou caracteres (`char`), para o caso das direções das cobrinhas.



## 5. REQUISITOS E PROCEDIMENTOS PARA INSTALAÇÃO E TESTE

Para jogar, é necessário conter apenas a instalação de uma plataforma que rode a linguagem Java para a inicialização do jogo. Diante disso, o *software* Netbeans IDE 8.2 foi utilizado como esse ambiente de desenvolvimento.

Para iniciar o jogo é necessário executar primeiramente a classe do servidor (Servidor.java), utilizando a porta do servidor TCP especificado. Logo após, é necessário conectar o primeiro cliente (Jogo.java) utilizando o servidor *localhost*, e o segundo cliente utilizando o endereço IP do primeiro. Após ambos apertarem o botão jogar, o jogo se inicia. O primeiro cliente é o primeiro jogador (cobrinha azul) e o segundo cliente é o segundo jogador (cobrinha rosa). Essa ordem é baseada em quem iniciou primeiro o jogo, apertando o botão jogar.