



**Universidade Federal de Alagoas**  
**Instituto de Computação - IC**  
**ENGENHARIA DE COMPUTAÇÃO**



**REDES DE COMPUTADORES - ECOM029T 2021.1**

**Prof. Leandro Melo de Sales**

**Karla Sophia Santana da Cruz - 19111113**

**Lilian Giselly Pereira Santos - 19111115**

## **Relatório de Atividades Desenvolvidas**

---

### **Descrição**

Projeto realizado para aquisição de nota da disciplina de Redes de Computadores, ministrada pelo prof. Leandro Sales no Instituto de Computação na Universidade Federal de Alagoas no semestre de 2021.1. Utilizando os conceitos de Socket, comunicação Cliente e Servidor e Thread aprendidos durante essa matéria, foi escolhido como aplicação ao projeto o famoso Jogo da Velha.

O tipo de socket utilizado no projeto foi o SOCK\_STREAM, que implementa o protocolo TCP de comunicação. No algoritmo do servidor, quando há uma conexão do cliente, após ele escolher o tipo de jogo que irá jogar, é iniciada uma thread para o tipo de jogo. Já no algoritmo do cliente, quando há uma conexão com o servidor, ele também inicia uma thread para executar o jogo da velha.

Na thread do servidor, caso o *gamemode* seja Player vs Player, sua função será de interligar os dois clientes, enviar as jogadas entre eles e avisar caso um jogador seja desconectado. Já para o caso da IA, o funcionamento de enviar as jogadas será similar, porém, o algoritmo da IA também rodará nessa thread. Já na thread no cliente, a função independe do modo de jogo, pois a execução do jogo da velha não muda, funcionando com as mesmas regras, como ganhar, perder e empate.

### **Principais Funcionalidades**

Alguma das principais funcionalidades presentes no jogo são:

- **Definição do nome do jogador:** o cliente pede o nome ao usuário para enviar ao servidor assim que realizar uma conexão com ele. Dessa forma, tanto o servidor quanto o possível oponente desse jogador (ler tópico “Modo Player vs Player”) saberão seu nome;
- **Escolha do modo de jogo:** após o servidor receber o nome do cliente, ele envia uma mensagem para que o cliente trate a questão da escolha do *gamemode*, no qual

o jogador pode escolher entre jogar contra outro jogador, tendo que esperar a conexão de outro cliente, ou jogar contra uma IA, que usa o algoritmo Minimax, executada diretamente no servidor. Vale salientar que as funções que rodam na thread variam de acordo com o *gamemode*;

- **Modo *Player vs Player*:** modo habilitado quando um cliente, ao se conectar ao servidor, escolhe jogar contra outro cliente. Assim, esse cliente fica em “modo de espera” até que outro cliente se conecte e escolha o mesmo modo de jogo. Quando os dois se conectarem, o servidor envia uma mensagem para ambos definindo quem inicia o jogo (sempre o jogador que se conectou primeiro) e o nome do seu oponente. Após isso, o servidor envia uma mensagem para informar ao cliente que o jogo começou e funciona recebendo a jogada de um jogador e enviando para o outro oponente, e vice-versa, até que haja um ganhador ou desista (desconecte-se);

- **Modo *IA*:** ao escolher esse modo, o funcionamento ocorre similarmente ao anterior. A diferença é que, ao cliente realizar sua jogada e enviar ao servidor, este passa ao algoritmo de IA que executa o Minimax e envia sua melhor jogada, ocorrendo ciclicamente até que o jogo finalize ou o cliente se desconecte;

## Dificuldades encontradas

Uma dificuldade encontrada durante a execução do projeto e que não foi resolvida é o fato de que, durante a conexão do primeiro cliente, o servidor envia uma mensagem para checar o *gamemode* que será jogado, e, conseqüentemente, espera uma resposta do cliente, o que acaba por “travar” o servidor nesse `.recv()`. Dessa forma, caso outro cliente venha a se conectar, ele ficará à espera da primeira mensagem do servidor, que não será enviada devido ao primeiro cliente, ocasionando numa espécie de “fila de espera” de conexões de clientes.

## Funcionalidades Futuras

- **Conceito de “Salas”,** onde será removida a limitação de apenas dois jogadores conectados caso um jogo “Player vs Player” esteja acontecendo. Com essa funcionalidade, múltiplas partidas PvP podem rodar simultaneamente.

- **Criação do cliente observador ou telespectador,** o qual será possível implementar com a funcionalidade acima criada, em que um cliente pode ingressar em uma sala para assistir.

- **Múltiplos jogos** em uma mesma sala, caso haja mais de 2 jogadores, numa espécie de campeonato.

- Desenvolvimento de uma **interface gráfica**.

- **Gravação do histórico de jogos** entre dois jogadores e a permissão para que eles joguem múltiplas rodadas juntos, bem como a exibição do placar.

## Modo de Execução

As instruções para execução do projeto se encontram no README.md do repositório do projeto, encontrado no link: [PROJETO DE REDES - GITHUB](#).