

# TicTacToeMP

## Trabalho de Redes de Computadores

Gilberto Antunes<sup>1</sup>, Henrique Tomé<sup>1</sup>

<sup>1</sup>Ciência da Computação – Universidade Estadual do Oeste do Paraná – (Unioeste)  
Caixa Postal 711 – 85.814-110 – Cascavel – PR – Brazil

{gilberto.junior2,henrique.gomes}@unioeste.br

### 1. O problema

O objetivo do trabalho é implementar um protocolo para o Jogo da Velha em rede utilizando uma arquitetura híbrida cliente/servidor e *Peer-to-Peer*. Inicialmente os jogadores devem se registrar em um servidor informando o seu nome e o número da porta em que estarão aguardando conexões. O endereço IP e porta do servidor são conhecidos e informados manualmente pelo jogador no início da execução do programa. O servidor deve armazenar e disponibilizar uma lista de jogadores conectados contendo o nome, o endereço IP (o mesmo utilizado no momento da conexão com o servidor) e o número da porta informados pelo jogador. Um jogador deve contatar o seu oponente para iniciar o jogo utilizando as informações do servidor.

### 2. Solução utilizada

A solução foi desenvolvida utilizando nodeJS para construção do *backend* (servidor UDP) de forma que este receba as inscrições dos jogadores e disponibilize as informações necessárias para outros jogadores. Tais trocas de informações é feita utilizando o protocolo definido na Seção 3. A interface foi construída utilizando VueJS e o *framework* Quasar e este consome as informações fornecidas por outros jogadores e pelo próprio servidor UDP por meio de um “*client side server*” também desenvolvido em nodeJS.

### 3. Protocolo Utilizado

“USER nome porta” essa deve ser a mensagem UDP enviada periodicamente (a cada 10 (dez) segundos) pelo jogador para o servidor indicando a sua presença no sistema. O nome é composto por uma única cadeia de caracteres e pode conter letras, números e  \_  (*underline*). O parâmetro porta indica em qual porta jogador aceitará conexões TCP de outros jogadores. O sucesso do registro é confirmado pela mensagem “USER OK”. Tentativas de registros de jogadores com parâmetros inválidos são recusadas com a mensagem “USER NOK”. Cada *host* (endereço IP) pode registrar um único jogador. O registro de um novo jogador do mesmo *host* implica na substituição automática do anterior. Se o servidor não receber a mensagem de anúncio após 1 (um) minuto, o jogador é removido pelo servidor da lista de jogadores ativos.

“LIST” é a mensagem UDP utilizada pelo jogador para obter a lista atualizada de jogadores registrados no servidor. O retorno é “LIST N jogador\_1 jogador\_2 .. jogador\_n”. O parâmetro N informa o número de jogadores da lista. Cada jogador é identificado por “nome:endereçoIP:porta”.

“EXIT” é a mensagem UDP enviada do jogador para o servidor indicando que o jogador está deixando o sistema. A ausência de mensagens “USER” após 1 (um) minuto tem o mesmo efeito.

Essas listadas acima são as funções utilizadas no serviço UDP, agora veremos as funções do serviço TCP que será utilizado para iniciar e durante o jogo.

“START nome” é a mensagem TCP enviada por um jogador A para o jogador B quando A deseja se comunicar com B. O parâmetro nome refere-se ao jogador A. A resposta deve ser “START OK” para confirmar o início do jogo ou “BYE” para indicar a recusa.

“BYE” é a mensagem TCP enviada por um jogador A para o jogador B indicando que ele recusando o jogo (após uma mensagem “START”) ou deixando o jogo em andamento (a qualquer momento).

“PLAY linha coluna” é a mensagem TCP enviada pelo jogador para indicar a sua jogada. linha e coluna referem-se à posição no tabuleiro que o jogador deseja fazer a marcação, sendo 0 menor ou igual linha, coluna menor que 3. Uma jogada válida é confirmada com “PLAY OK”. Uma indicação de jogada incorreta (fora dos limites do tabuleiro ou em uma posição já marcada) deve ser recusada com a mensagem “PLAY NOK”, indicando que o jogador deve refazer a jogada. Três jogadas incorretas consecutivas implicam na anulação da partida com o envio de “BYE” pelo jogador que está aguardando a jogada.

Após cada jogada válida (com confirmação “PLAY OK” do oponente) o sistema verifica se há ganhador ou empate e informa ao jogador local.

#### **4. Detalhes de implementação**

O servidor UDP é responsável pelas inscrições e disponibilização das informações, ele recebe as mensagens por meio de um *socket* UDP, esse da biblioteca *dgram*, e a cada 60000 mili segundos executa o serviço de verificação de usuários inativos, de forma que aqueles que não mandaram a mensagem de anuncio no último minuto são removidos da lista de jogadores, essa que têm como função guardar as informações de cada jogador que fez o registro no jogo, essa lista é aquela que é retornada quando solicitado pela mensagem “LIST”. As informações de registro de jogo são validadas de acordo com o protocolo (ver Seção 3) por meio de *regex* específicas para cada campo requerido. O servidor fica “ouvindo” em seu endereço IP (da máquina hospedeira) e uma porta específica (8000), essas informações que serão usadas para que o cliente anunciar seu registro no jogo.

O cliente é o responsável por se cadastrar no jogo (servidor), por meio de um *socket* UDP também da biblioteca *dgram*, fornecendo as informações requeridas e de acordo com o protocolo (ver Seção 3), esse processo é repetido automaticamente a cada 10 segundos, de acordo com o protocolo, dessa forma anunciando sua presença. A função de listagem é executada de forma automatizada para que o jogador tenha sempre uma lista atualizada de jogadores disponíveis. No momento do cadastro (assim que ação é confirmada pelo jogador, ao apertar o botão da interface “cadastrar”), um servidor TCP é aberto utilizando as informações fornecidas pelo usuário e de acordo com o protocolo. Após devidamente registrado o jogador é redirecionado para o *lobby*, onde a lista dos jogadores disponíveis é mostrada e atualizada a cada 5000 mili segundos. A partir deste momento o jogador pode selecionar um oponente da lista de jogadores disponíveis, fazendo isso

é criado um cliente TPC, que envia ao servidor TCP do oponente a mensagem de início de jogo “START” seguindo o protocolo. Aquele que inicia o jogo recebe a vantagem da primeira jogada, a partir daí o jogo é realizado por meio de troca de mensagens entre os dois cliente/servidor TCP de cada jogador seguindo o protocolo definido. A interface desenvolvida em Vue auxilia na jogabilidade.

## **5. Resultados**

Ao final obteve-se um jogo TicTacToe multiplayer utilizando os conceitos de conexões TCP e UDP, e seguindo um protocolo bem definido de forma a simular as especificações de uma RFC. O jogo funciona perfeitamente entre computadores, utilizando o navegador como base para rodar a aplicação, tendo em vista a linguagem utilizada. A interface apesar de simples é funcional e intuitiva e supre bem o seu papel e facilita o jogo.