



UNIVERSIDADE
ESTADUAL DE LONDRINA

VITOR HENRIQUE ROSA BATISTA DE OLIVEIRA
VITOR PALMA ADERALDO

JOGO EM REDE UTILIZANDO SOCKET: JOGO DA VELHA

LONDRINA - PR
2016

Sobre o jogo:

Dizem que o jogo da velha se popularizou no século 19, quando as mulheres inglesas costumavam se reunir às tardes para bordar e conversar, porém, ao passar da idade, a visão não permitia mais fazer boas costuras e bordados. Então, restava às velhas senhoras seguir conversando e se distrair jogando o jogo da velha.

O tabuleiro, na versão mais tradicional possui 3x3 casas, ou seja, nove espaços, e consiste em dois jogadores, um sendo X e outro O. Ganha quem formar primeiro uma linha horizontal, vertical ou diagonal com sua jogada (X ou O).

Funcionamento:

O jogo da velha nada mais é do que uma Matriz, no código ela foi vista como um vetor de 9 posições, porém para fazer um jogo da velha em rede foi necessário criar mais ferramentas, como um servidor e cliente.

Classes utilizadas: Matriz, Servidor, Cliente.

Matriz: Cria um vetor de 9 posições que é utilizado como um “tabuleiro”, todas as regras do jogo estão contidas nessa classe, essa classe retorna a string “333333333” inicialmente, indicando que todas as posições não foram marcadas.

Servidor: Aguarda clientes se conectarem, e determina a vez de cada cliente jogar. É o fluxo de comunicação entre dois Clientes.

Cliente: Realiza o movimento utilizando as regras da classe Matriz.

Para iniciar uma partida é necessário executar o Servidor, o Servidor inicia-se aguardando uma conexão que será a conexão do Cliente 0, após isso ele continua escutando mais uma conexão, a do Cliente 1. Após os dois clientes estarem conectados no Servidor, é gerado a Matriz, com todos os valores iniciados com 3, esse número indica que nenhuma posição foi preenchida. Após isso o Servidor envia uma mensagem para o Cliente 0, indicando que é a vez dele jogar, o cliente 0 então preenche a Matriz e gera uma String que possui informações a respeito da jogada, se ele preencheu a posição 2 da Matriz, será gerado uma String “330333333”, indicando que a posição dois da Matriz foi marcada pelo Jogador 0 (zero). Ao ser enviada essa string para o servidor, o servidor atualiza sua Matriz, verifica se houve ganhadores e então repassa essa String para o player 2, o qual atualiza sua Matriz em cima dessa String. Após atualizar, o player 2 pode realizar um movimento, ele então preenche a posição 1, será gerado uma string a respeito do movimento e então essa String é enviada para o

servidor, que atualiza sua matriz e envia a mesma string para o player 1.

Socket:

Socket é uma API que permite um fluxo de comunicação entre aplicações através de uma rede de computadores. É utilizado para criar aplicações que necessitam se comunicar pela rede através de protocolos.

A API Socket suporta os dois protocolos de transporte TCP e UDP todos utilizando o protocolo de rede IP.

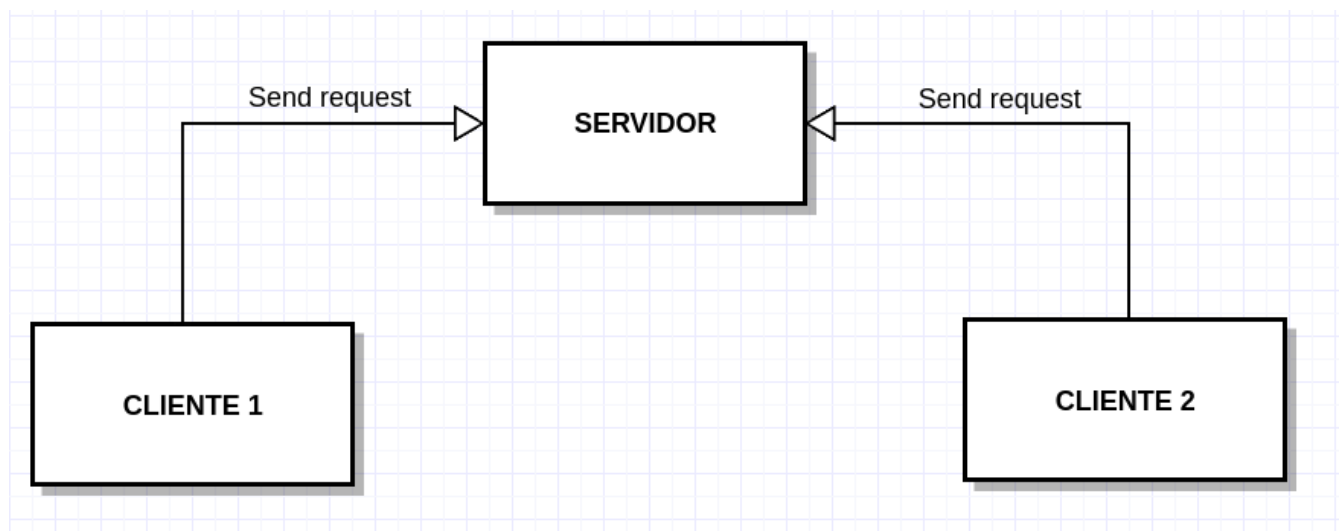
TCP: O qual é confiável por garantir que os dados cheguem.

UDP: Não confiável, a aplicação deve garantir que os dados cheguem.

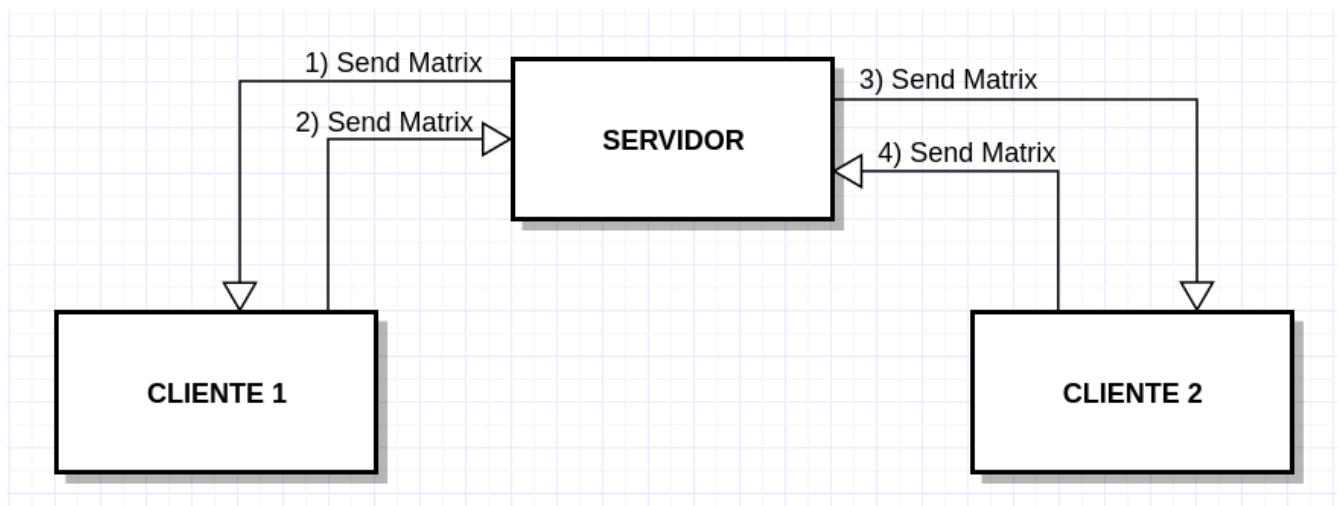
O protocolo utilizado neste jogo foi UDP, por permitir uma comunicação mais rápida, porém não foi realizado o tratamento para garantia do envio e recebimento dos pacotes, mas por enviar basicamente strings em uma rede local, o jogo não perde pacotes.

Fluxo:

1) Clientes se conectarem ao Servidor.



2) Após a conexão de ambos clientes, fluxo dos pacotes que contém a Matriz.



O fluxo sempre será assim, sempre após o envio de um pacote, é verificado se o houve vitória, derrota ou empate, assim não é necessario que essa informação seja transmitida. Ao receber o pacote essa verificação também é feita, antes de um player realizar o movimento.