

Trabalho Prático - Redes

Guilherme Resende, Gustavo Germano, Ramon Gonze

1º de dezembro de 2019

1 Introdução

Um dos alicerces sobre o qual se estabelece a estrutura de comunicação dos sistemas modernos em rede são os Sockets. Esses recursos são interfaces de comunicação responsáveis tanto pelo envio, quanto pelo recebimento de informações entre diferentes sistemas/dispositivos.

O modo trivial de operação dos Sockets é baseado no estabelecimento de uma rota de envio (IP público), e uma porta de comunicação no sistema destino. Tal interface torna possível a troca de mensagem entre processos de sistemas integrados à uma rede de comunicação.

Assim como os sockets, outro recurso vital ao funcionamento dos sistemas de internet são as diferentes linhas de processamento paralelo (Threads). As Threads trazem um componente crucial para os sistemas modernos, o paralelismo. Mais threads, principalmente integradas aos sockets de comunicação, agregam maior fluidez à experiência de usuário e maior capacidade de gerenciamento de mensagens simultâneas.

Este trabalho prático tem por foco o desenvolvimento de um sistema de comunicação em rede tal que implemente os conceitos de Socket e Thread descritos acima. De tema livre, o trabalho tem como única restrição o uso obrigatório das seguintes funções primitivas: socket, read, recv, write.

Tendo em vista a liberdade de escolha, optou-se pelo desenvolvimento do tradicional Jogo da Velha (ou Tik Tak Toe), onde se é possível ter a perspectiva da comunicação entre dois jogadores com o intermédio de um servidor no qual o jogo é hospedado.

2 Aplicação

O jogo desenvolvido se dá em um tabuleiro de dimensão 3x3 com dois oponentes. Cada jogador assume um tipo de marcação (X ou O), e a vitória consiste na formação de três marcações consecutivas - de um mesmo jogador, com alternância de turnos - tanto no sentido horizontal, como no vertical ou no diagonal.

Duas interfaces de uso foram criadas, uma gráfica, para os jogadores, e outra administrativa, para o servidor. A comunicação se baseia na transmissão de ações e coordenadas entre os atores a partir de gatilhos de eventos e utilizando a implementação de sockets referenciada na documentação oficial da linguagem Python ¹.

Ao fim de cada partida, os jogadores têm autonomia de prosseguir para outra partida, ou encerrar a execução. Entretanto, vale ressaltar que, caso pelo menos um dos jogadores opte pelo fim da partida, o sistema é interrompido. Uma vez expressado o desejo de permanência por ambas as partes, as configurações são reiniciadas.

Com base na descrição do jogo, pode-se definir 9 tipos de comunicações suportadas pela aplicação, onde duas mensagens partem dos jogadores com destino ao servidor, e todas as demais partem do servidor para os jogadores (Ver Tabela 2).

2.1 Servidor

Para o correto estabelecimento da interface de servidor, são necessários um endereço de IP e uma porta de comunicação, os quais serão utilizados pelos jogadores para o envio de suas respectivas mensagens. Uma vez em funcionamento, o servidor estabelece threads de execução exclusivas para o monitoramento da partida e das ações de cada usuário. As linhas de processamento se dividem em

- Gerenciamento de Usuário: Esta thread monitora eventos relacionados aos usuários por meio da função **manageClient**, invocada após um clique válido no tabuleiro de jogo. Tal função é responsável por consolidar uma jogada de usuário, entretanto, realiza também checagens no que diz

¹<https://docs.python.org/3/library/socket.html>

Mensagem	Valor	Ator
Indicação de Turno	0	Servidor
Turno do 1 Jogador	1	
Turno do 2 Jogador	2	
Atribuição como 1 Jogador	3	
Atribuição como 2 Jogador	4	
Vitória do 1 Jogador	5	
Vitória do 2 Jogador	6	
Jogada	7	Jogadores
Jogar Novamente	8	
Finalizar o Jogo	9	

Table 1: Índice de Possível Valores com seus Respectivos Atores.

respeito a uma possível vitória, ao desejo de fechar a aplicação após o termino de uma partida, ou mesmo a possibilidade de uma revanche;

- Gerenciamento de Partida: Esta thread monitora, por meio da função **manageMatch** as ações de usuário, realiza a alternância de turnos, e também atualiza a visualização do tabuleiro de jogo.

O jogo recebe o aval de início quando dois jogadores concordam em disputar uma partida.

2.2 Jogadores

Uma vez estabelecidas as configurações do servidor, os jogadores são capazes de criar instâncias de partidas. Para o início desse procedimento, é necessário que os usuários informem o endereço de IP do servidor responsável. Contudo, também é necessário saber a porta para comunicação do processo destino.

Cada jogador executa localmente uma interface do sistema que, por sua vez, possui uma thread exclusiva para o monitoramento das atualizações de jogo feitas por seu oponente e enviadas por intermédio do servidor.

De forma a reduzir o número de transações de mensagens entre os atores do sistema, cada jogador recebe uma cópia do estado atual do tabuleiro no início de seu turno, tornando possível a validação de jogadas do lado do cliente.

2.3 Protocolo de Aplicação

Assim como citado em 2, este trabalho considera a biblioteca de sockets fornecida na implementação padrão da linguagem Python. Para o escopo da aplicação aqui descrita foram utilizados sockets TCP para a comunicação inter-processos.

A aplicação possui dois tipos de mensagem que variam de acordo com os atores descritos na tabela 2, a dizer,

- Mensagem de Cliente: Composta por dois bytes correspondentes às coordenadas X e Y do tabuleiro. Este tipo de mensagem é utilizado pelas ações Jogada, Jogar Novamente e Finalizar o Jogo;
- Mensagem de Servidor: Composta por três bytes onde o primeiro indica a ação a ser tomada, e os dois restantes podem ser coordenadas X e Y, ou valores insignificantes (estritamente dependente da ação em questão). Este tipo de mensagem é utilizado para indicação de turno, atribuições ou vitórias.

3 Possíveis Melhoramentos

Um ponto de melhoramento é a interface gráfica. Nenhum dos integrantes do grupo possui extensa experiência em construção de interfaces ou de *User Experience*, além de pouca experiência na biblioteca *Tkinter*, utilizada na construção da interface do trabalho. Outra possível melhoria seria possibilitar ao usuário com qual símbolo ele quer jogar, X ou O.

4 Principais Dificuldades

As principais dificuldades foram lidar com a lentidão causada algumas vezes pela pacote *socket* do Python e gerenciar corretamente as *threads* e suas respectivas conexões. Ao fim de um jogo, ambos os usuários são perguntados se querem e continuar a jogar e, se ambos concordarem, uma novo jogo é iniciado. Manter e gerenciar as conexões corretamente após o início de um novo jogo foi um dos principais pontos de dificuldade do trabalho.

5 Demonstração em vídeo

Foi feita a demonstração da execução do jogo em um vídeo, que está disponível no YouTube através do seguinte link: <https://youtu.be/bjSy4wVoLBI>.

6 Código-Fonte

O código fonte está disponível no GitHub, através do seguinte link: <https://github.com/ramongonze/tic-tac-toe-online>.