

RELATÓRIO PROJETO REDES DE COMPUTADORES – 2021.1

Equipe: Lucas Alves de Barros e Jonathan Yuri da Silva Santos

O projeto foi desenvolvido utilizando a IDE “C++ Builder Community Edition” e é composto de duas aplicações: "cliente" e "servidor". Embora essa IDE seja baseada em Object Pascal, todo o código foi feito em C++ (com algumas peculiaridades da própria IDE), com exceção do código base do FORM e dos componentes utilizados (nativamente escritos em Pascal). A ideia inicial era desenvolver uma aplicação que tivesse apenas a função de trocar mensagens, porém, após iniciar a implementação, decidimos criar um desafio maior e implementar um jogo simples.

A aplicação tem as funcionalidades de convidar pessoas para jogar (desde que a outra pessoa não esteja jogando), trocar mensagens (somente durante a partida), ativar/desativar (servidor), conectar/desconectar (cliente). O servidor associa duas pessoas que estejam conectadas (na mesma partida) e redireciona as mensagens e dados de jogada recebidos para o respectivo oponente. A threads nativa do componente "TIdTCPServer" possibilita que mais de uma pessoa se conecte ao server. Foram criados arquivos externos com novas threads para possibilitar a leitura de dados recebidos enquanto alguma tarefa estivesse sendo executado ou até mesmo para executar tarefas paralelas na aplicação cliente. Os jogadores são automaticamente desconectados após uma partida ser finalizada ou caso algum dos dois fique offline.

Algumas coisas que poderiam ter sido implementadas:

- Enviar mensagens sem necessariamente estar jogando com a outra pessoa;
- Ver quais pessoas estão em jogo;
- Desistir da partida;
- Escolher manualmente entre jogar novamente com a mesma pessoa ou encerrar a conexão com a mesma;
- Receber e enviar mensagens para mais de uma pessoa.

Algumas dessas funcionalidades talvez fossem realmente úteis, mas como na maioria, até então, não foi visto necessidade, foi decidido finalizar o projeto, pois o objetivo principal foi alcançado.

Quanto às dificuldades, de forma resumida pode-se dizer que foram os conflitos no envio/recebimento dos dados, e nesse caso o dado era enviado pelo remetente, chegava até o servidor, mas não era redirecionado ao destinatário, pois por algum motivo o servidor perdia a referência do destino e ficava com esse dado retido. Esse problema causava o travamento do remetente e do servidor. O problema foi resolvido com a utilização de uma classe "map" para agilizar a busca do destinatário associado ao remetente e evitar que a referência da origem sobrescrevesse o destino.