

# iSocket

Julio Cesar F. Lima<sup>1</sup>

<sup>1</sup>Universidade Federal do Ceará (UFC)  
Sobral – CE – Brazil

**Abstract.** *This paper describes communication using sockets. Socket is a communication model commonly used to deploy server/client applications. Allowing the exchange of messages between the process.*

**Resumo.** *Esse trabalho descreve a comunicação utilizando sockets. Socket é um modelo de comunicação comumente utilizado para implementar servidor/cliente. Permitindo a troca de mensagens entre o processo.*

## 1. Introdução

A aplicação possui simples execução necessitando apenas da inicialização do arquivo executável. A aplicação conta com uma interface usuário simples que possibilita a troca de mensagens entre clientes e o servidor de maneira intuitiva.

A implementação do software foi facilitada devido as bibliotecas e larga implementação de terceiros, assim os erros foram facilmente resolvidos conforme algumas poucas buscas. As portas utilizadas para a troca de mensagens, foi a de número "8888" e para simulação de clientes diferentes, foi gerado um número aleatório entre 1 e 255 para gerar a terminação do IP do cliente "127.0.0.X". Assim com o mesmo programa pode se conectar ao servidor com uma probabilidade alta de não dar defeito, e caso haja será lançada uma exceção, não prejudicando o fluxo de código, dessa maneira deve ser apenas novamente tentar se conectar ao servidor em caso de falha.

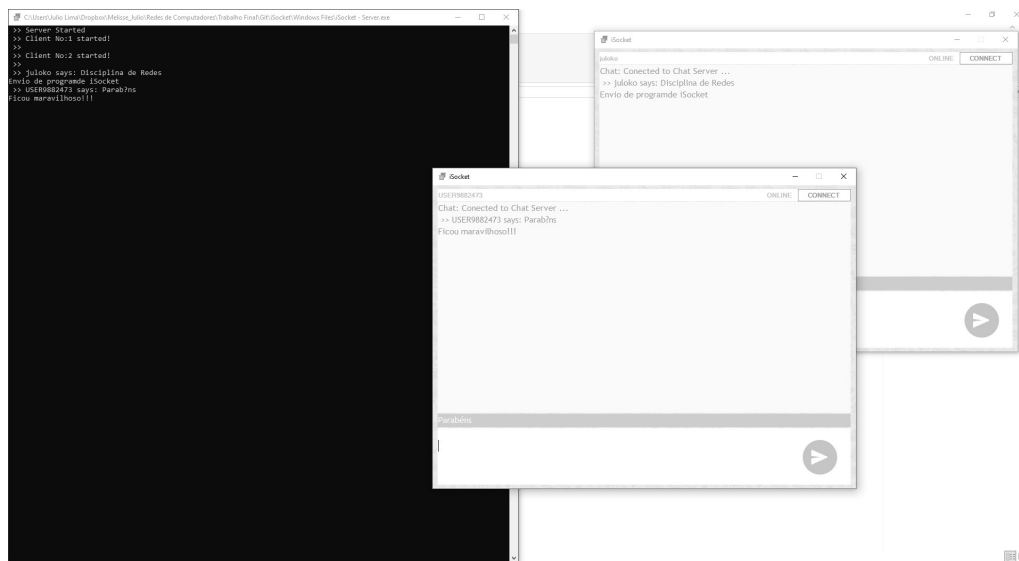
Os arquivos para teste estão na pasta "Windows Files".

O usuário para deve clicar no botão "CONNECT" de maneira a se conectar ao servidor. Após a abertura do programa do lado do cliente o software gera um número de usuário aleatório, no entanto o usuário poderá muda-lo ao seu gosto. Para o envio de mensagens o usuário terá a sua disposição um campo para o preenchimento do assunto da mensagem enviada e um campo para a mensagem. Há um painel onde são mostradas as mensagens enviadas. Junto com o envio do assunto e mensagem é enviado ao servidor o nome do usuário.

No lado do servidor é mostrado sempre que um cliente novo se conecta, assim como todas as mensagens enviadas pelos clientes. Nesse ponto o software está a um passo de funcionar como uma sala de chat, bastando após a recepção por meio do servidor das mensagens o envio em broadcast para cada um dos clientes. Caso houvesse a necessidade de salas, poderia facilmente separar os usuários por meio das portas.

## 2. Referências

A documentação para reprodução do software tanto do lado cliente quanto do lado do servidor está disponível nas referências.[NET 2010a], and [NET 2010b]. Assim como foi criado um repositório dos arquivos aqui produzidos [Lima 2018].



**Figure 1. Software desenvolvido.**

## References

- Lima, J. (2018). isocket. Disponível em: <https://github.com/juloko/iSocket>.
- NET, I. (2010a). C# multi threaded server socket programming. Disponível em: <http://csharp.net-informations.com/communications/csharp-multi-threaded-server-socket.htm>.
- NET, I. (2010b). How to c# chat client. Disponível em: <http://csharp.net-informations.com/communications/csharp-chat-client.htm>.