	<p style="text-align: center;"> <Atividade 2 Chatbox> por <Bruno Rodrigues, Philip Akpanyi > - 2020.1 Boa vista, 12/10/2020 </p>
--	---

Relatório Atividade 2

Bruno Rodrigues Caputo (brunorcx@hotmail.com)

Philip Mahama Akpanyi (pakpanyi@gmail.com)

DCC602-Sistemas Distribuídos 2020.1-Turma 01

Universidade Federal de Roraima

DCC -Departamento de Ciência da Computação - Bloco V

Campus Universitário do Paricarana - Aeroporto

69310-000 Boa vista, RR

Introdução

Este trabalho tem o objetivo de desenvolver uma aplicação cliente servidor para mostrar a comunicação usando sockets. O projeto foi desenvolvido utilizando a linguagem JavaScript. Para receber os dados do cliente, uma página básica foi criada para receber mensagens e ao enviar, o servidor retornar a mesma mensagem na forma revertida.

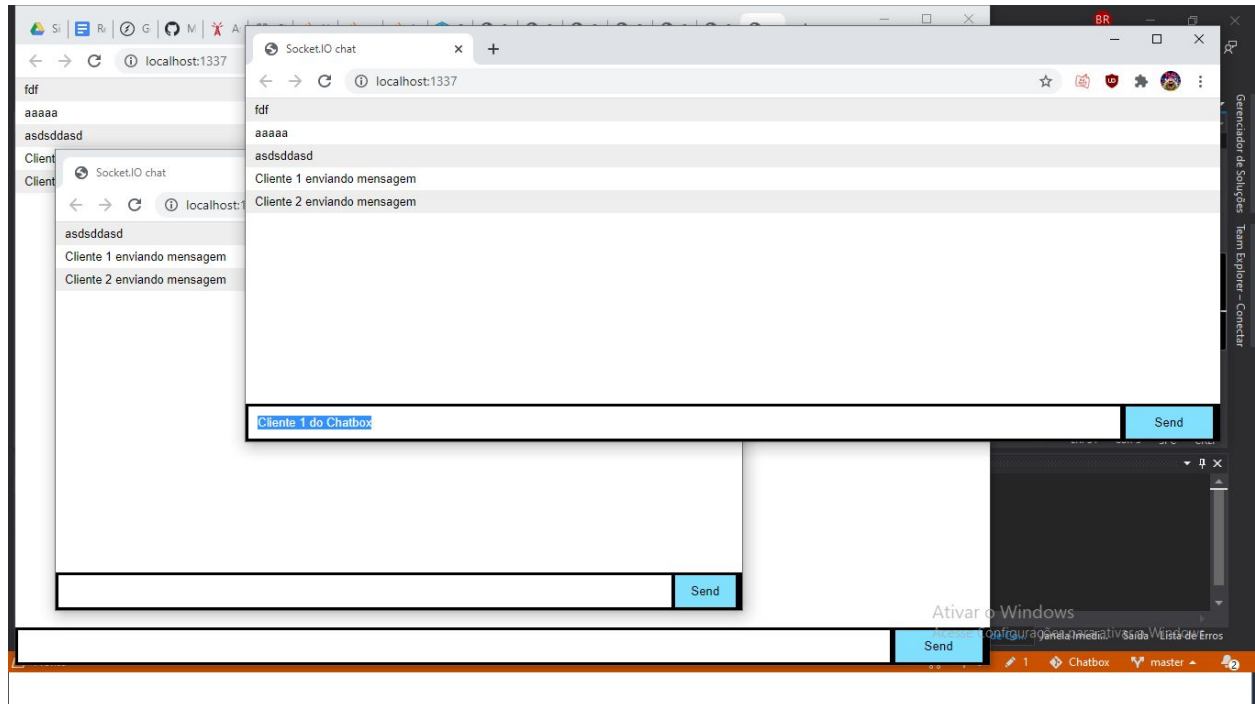
Execução

Para que o código execute, é preciso instalar as dependências. Para isso, deve navegar até o diretório principal usando o terminal. Do diretório principal, execute *npm install* e espere por alguns segundos. É preciso instalar dois pacotes (express e socket.io) como requisito para o funcionamento do código. Para isso, basta executar *npm i express* e *npm i socket.io*.



<Atividade 2 Chatbox>

por <Bruno Rodrigues, Philip Akpanyi > - 2020.1
Boa vista, 12/10/2020



Dois clientes se conectam ao servidor e conseguem conversar

Desempenho

Para medir o desempenho da aplicação, mensagens de 1byte, 512bytes, 1kb, etc foram enviadas e o desempenho avaliado pelo gerenciador de tarefas do computador. Estas mensagens serão executadas de acordo com a função:

```
function tamanhoMensagens (numByte)
```

Um *broadcast* será enviado para os clientes contendo o caracter “a” com o tamanho especificado pela variável *numByte*. Os testes para sua implementação se dá assim que o servidor detectar uma mensagem com número.



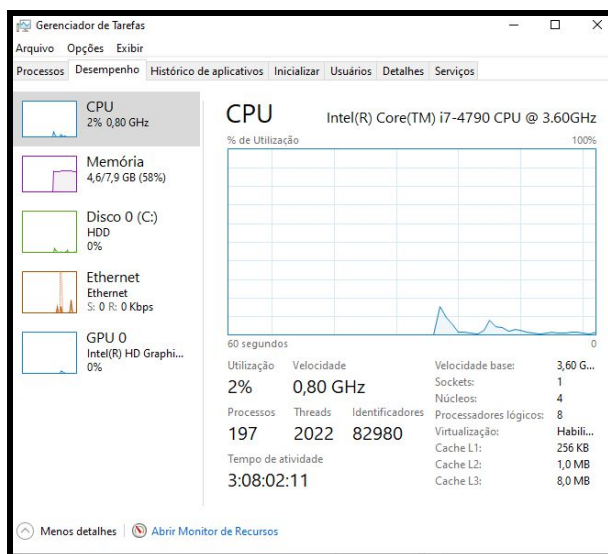
<Atividade 2 Chatbox>

por <Bruno Rodrigues, Philip Akpanyi > - 2020.1
Boa vista, 12/10/2020

```
C:\Darkb\Aplicativos\nodejs\node-v12.18.4-win-x64\node.exe
Debugger listening on ws://127.0.0.1:5858/14b35a9e-896a-4cdc-86a9-34f8f1308373
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
listening on *:1337
Um cliente se conectou
Um cliente se conectou
```

Servidor rodando através de *nodejs*

Como servidor está sendo usado em um computador local, portanto, alguns testes podem variar devido a outras aplicações fora o ChatBox. As fotos a seguir foram retiradas de acordo com os seguintes testes realizados em sequências:

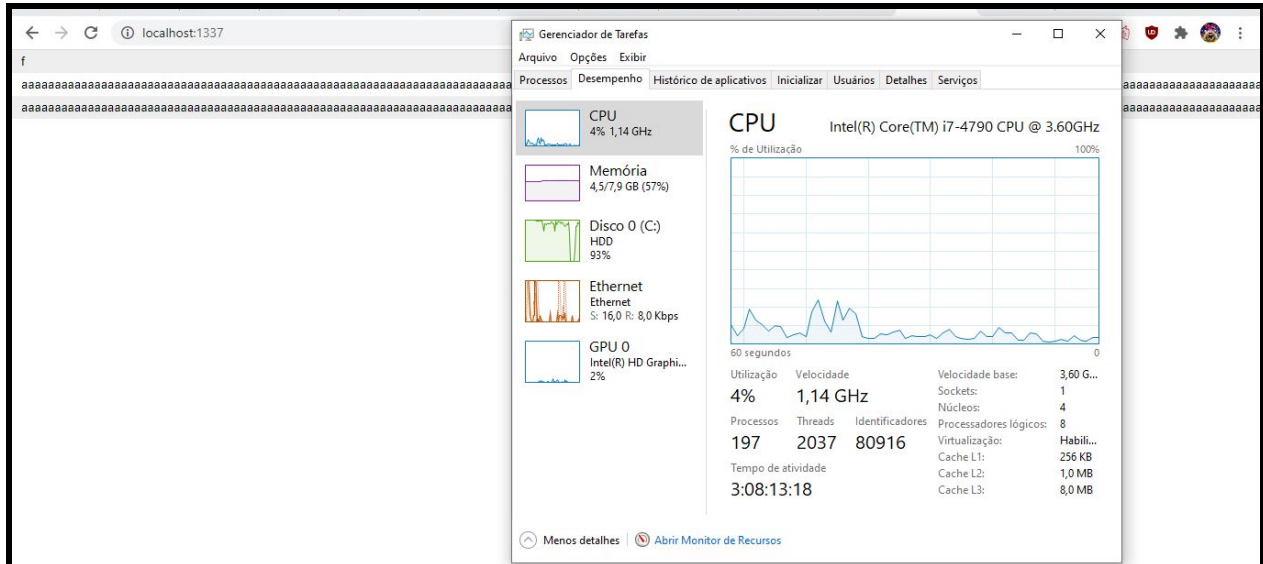


8 bytes

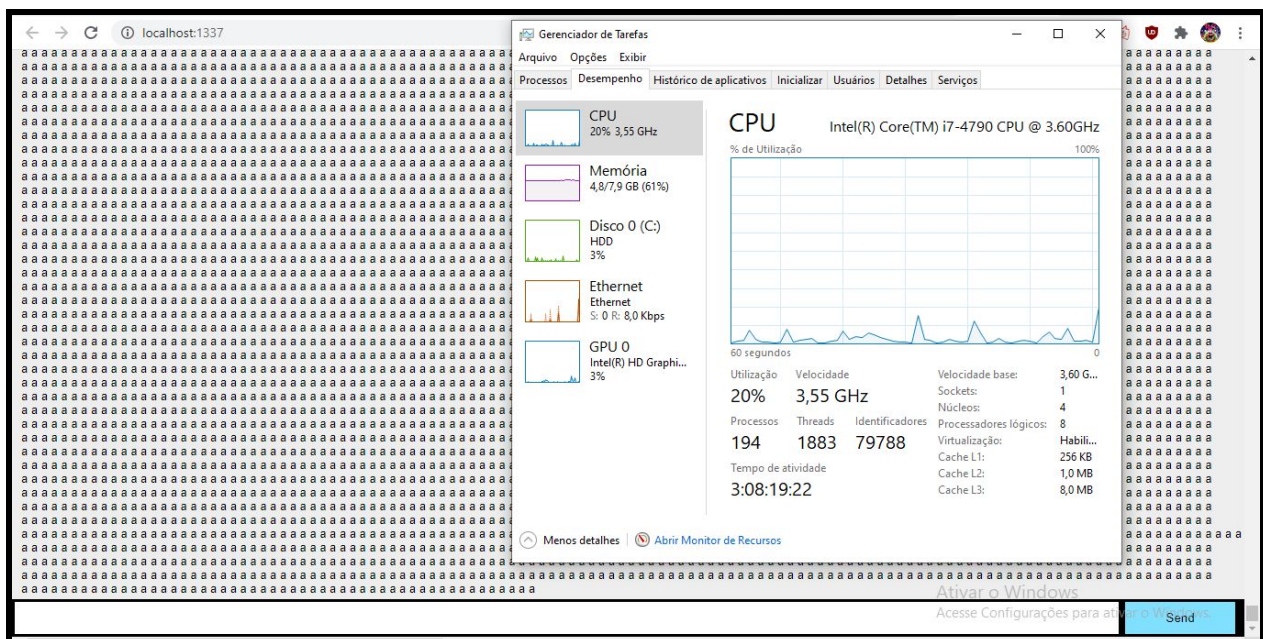


<Atividade 2 Chatbox>

por <Bruno Rodrigues, Philip Akpanyi > - 2020.1
Boa vista, 12/10/2020




256 bytes



6.000.000 bytes

Como esperado com o aumento no tamanho das mensagens enviadas o processamento para que o servidor seja capaz de enviar estas mensagens é bem mais alto, no último caso chegamos a 20% de uso do CPU. Um detalhe interessante sobre esse teste se diz

	<p><Atividade 2 Chatbox> por <Bruno Rodrigues, Philip Akpanyi > - 2020.1 Boa vista, 12/10/2020</p>
--	--

respeito a visibilidade sobre alguns dos caracteres enviados no ChatBox estarem atrás do Gerenciador de Tarefas. Vale ressaltar a proposta utilizada para ataques DoS, foi utilizado um *handler* para detectar entradas com valor estipuladas em com *requests* com até 5 ms. Diante disso se desconecta com o *socket* utilizado.

Bibliografia

<https://stackoverflow.com/questions/20435553/how-to-protect-against-distributed-denial-of-service-attacks-in-node-js-with-socket>

<https://socket.io/get-started/chat/#Getting-this-example>