



**FEPEG**

FÓRUM DE ENSINO,  
PESQUISA, EXTENSÃO  
E GESTÃO

TRABALHOS CIENTÍFICOS APRESENTAÇÕES ARTÍSTICAS E CULTURAIS DEBATES MINICURSOS E PALESTRAS

23 A 26 SETEMBRO DE 2015  
Campus Universitário Professor Darcy Ribeiro

ISSN 1806-549X

A HUMANIZAÇÃO NA CIÊNCIA, TECNOLOGIA E INOVAÇÃO



## **BUBBLE CHAT: UM SISTEMA DISTRIBUÍDO DE COMUNICAÇÃO USANDO SOCKETS EM UMA ARQUITETURA CLIENTE-SERVIDOR**

*Amanda Gabriely Santos Alves , Luanna Ferreira Neves , Joana Gabriela Ribeiro de Souza , João Paulo Pereira Nery , Lucas Araújo Borges , Rodrigo Mota Lacerda*

### **Introdução**

O presente trabalho descreve o desenvolvimento de um sistema de comunicação distribuído – *Bubble Chat* – contando também com breves conceitos acerca das tecnologias utilizados para o desenvolvimento deste software. O protótipo tem como objetivo demonstrar o desenvolvimento de uma aplicação distribuída com base na arquitetura Cliente/Servidor, visto que a utilização de aplicações distribuídas vem crescendo com o tempo e a necessidade de processamento mais rápido e eficiente, contribuiu para o crescimento dos sistemas distribuídos como uma ferramenta muito importante nos processos de comunicação, sincronização de dados e troca de recursos, fazendo com que o processamento possa ser exercido por vários agentes e removendo a distância do centro dos fatores que atrapalhavam a comunicação.

A motivação parte da crescente e constante necessidade das organizações e indivíduos de se comunicarem, prezando pela segurança e sigilo, o que fomenta a necessidade de ferramentas efetivas nestes aspectos. Portanto, este artigo visa descrever o desenvolvimento de um *chat* que foi formulado com o intuito de poder ser utilizado para fins básicos e diversos, desde universitários, amigos e familiares, para conversas informais, até organizações e equipes de trabalho, para efetivar a constante comunicação que é necessária na realização e cumprimento de projetos.

### **Material e Métodos**

#### *A. Sistemas Distribuídos*

Segundo Tanenbaum (2007), um sistema distribuído é um conjunto de computadores independentes entre si que se apresenta a seus usuários como um sistema único. No desenvolvimento do *Bubble Chat*, foi utilizada a arquitetura cliente-servidor, que consiste na distribuição de processamento entre fornecedores de serviço e requerentes destes, onde os fornecedores são denominados servidores, enquanto os requerentes são chamados de clientes.

A arquitetura desse trabalho possibilita a comunicação entre máquinas através de um elemento centra – servidor – que estabelece conexão com todas as outras máquinas que fazem requisições – clientes – permitindo a interação entre estes.

#### *B. Java e Java Virtual Machine – JVM.*

Por esta aplicação se apresentar aos usuários como uma única ferramenta integrada, apesar de possuir componentes distintos, é considerada um sistema distribuído. Para o desenvolvimento do sistema distribuído em questão foi utilizada a linguagem de programação Java, por ser altamente capacitada a atender o desenvolvimento de softwares completos, sendo hoje executada em mais de 850 milhões de computadores pessoais e também em bilhões de dispositivos móveis. Por ser uma linguagem interpretada, cada dispositivo que necessite executar usas linhas de código, deve possuir um interpretador chamado *Java Virtual Machine* – JVM. Esta que lê, no momento de execução, as linhas de código, permitindo uma grande portabilidade.

#### *C. Thread*

Nas aplicações distribuídas há uma recorrente necessidade de executar atividades diferentes ao mesmo tempo, ou seja, concorrentes, para isso faz-se o uso de *threads*, que possuem algumas vantagens em relação aos processos, por exemplo, a troca de mensagens entre elas é mais eficiente que a troca de mensagens entre processos. E por a linguagem Java dar suporte a criação e manipulação de *thread*, além da utilização de sockets, que consiste em um canal de comunicação entre duas máquinas através da abertura de portas, ela favorece o desenvolvimento de sistemas, abstraindo para o programador a utilização destas e tornando este desenvolvimento mais simples.

### **Desenvolvimento**

O sistema desenvolvido consiste em uma aplicação de comunicação em rede local (LAN), denominado *Bubble Chat*, que permite que pessoas em uma mesma rede troquem mensagens, onde todos os clientes que tiverem a aplicação em execução receberão as mensagens enviadas. Portanto, enquanto um cliente só se comunica com um servidor, o servidor



se comunica com todos aos clientes ativos, fazendo do servidor o elemento central que recebe todas as mensagens enviadas pelos clientes e redistribui para todos os clientes restantes. Foram utilizados *socket* e *threads* para a base do desenvolvimento, e seus processos ocorrem conforme mostra a tabela 1.

A interface da aplicação é intuitiva e bastante simples, contanto com apenas duas telas, a de login e a tela de conversação. Na tela de *login* o usuário insere o nome (*nickname*) que será usado no chat. Já na tela de conversação, o usuário conta com uma caixa para visualização das mensagens recebidas e enviadas, bem como uma caixa de inserção de texto e um botão para que este seja enviado. As telas podem ser observadas na Figura 1.

## Conclusões

Com o desenvolvimento do *Bubble Chat* foi possível reafirmar a complexidade do desenvolvimento de uma aplicação distribuída e todas as suas peculiaridades. Entretanto, através da efetivação de comunicação entre máquinas distintas, compreendemos como deve ser desenvolvido e como deve operar um sistema de comunicação e como as técnicas dos sistemas distribuídos podem ser utilizadas para alcançar este objetivo. Por fim, com o cumprimento do objetivo principal deste, o desenvolvimento do *Bubble Chat* com sua perfeita operação, foi possível visualizar como é possível desenvolver aplicações distribuídas e suas vantagens sobre as não-distribuídas, ressaltando a importância das técnicas de sistemas distribuídos e como estas contribuem para diversas ferramentas.

## Referências

- [1] CESTA, André Augusto. JAVATUTORIAL - A Linguagem de Programação Java e Orientação a Objetos/ Unicamp. - 1996. - Atualizado em 2009. - 01 de Junho de 2014. <<http://www.ic.unicamp.br/~cmrubira/JAVATUT14PDF.pdf>>. Acesso em 13 de maio de 2015.
  - [2] ORACLE. O que é a tecnologia Java e por que é necessária? Java. - 2013. - 01 de Junho de 2014. - <[http://www.java.com/pt\\_BR/download/faq/whatis\\_java.xml](http://www.java.com/pt_BR/download/faq/whatis_java.xml)>. Acesso em 13 de maio de 2015.
- SOMMERVILLE, Ian. Engenharia de Software. - São Paulo : Pearson, Addison Wesley, 2007.

**Tabela 1.** Processo de Comunicação.

SERVIDOR	CLIENTE
1. Abre a porta através de um <i>Socket</i> , especificando a porta - utilizamos 61000. 2.1 Aceita solicitação do Cliente e abre o canal de comunicação para envio de mensagens.	2. Cria um <i>Socket</i> , passando assim o IP do Servidor e especificando a porta.
3.1 Cria <i>Thread</i> responsável por escutar mensagens enviadas pelos Clientes.	3.2 Criar uma <i>Thread</i> responsável por escutar mensagens enviadas pelo Servidor.
	4. Envia mensagem para o Servidor.
5. Se houver mensagem em arquivo, carrega as mesmas para todos os Clientes.	
6. Armazena em arquivo as mensagens enviadas pelos Clientes	
7. Encaminha para todos os Clientes as mensagens enviadas.	

Fonte: Desenvolvida pelo autor, 2015.



**Figura 1.** Telas da aplicação.  
Fonte: Figura construída pelo autor, 2015.