**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC**

**CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ – CEAVI**

**JULIO CESAR BUENO DE OLIVEIRA JUNIOR & GABRIEL ANTÔNIO BERTOLDI PERINI**

**SISTEMA DISTRIBUÍDO PARA COMUNICAÇÃO EM CHAT EM SALAS DE GRUPO**

**IBIRAMA**

**2021**

**SUMÁRIO**

[1. RESUMO 3](#_Toc112864483)

[2. DESCRIÇÃO GERAL DO PROJETO 4](#_Toc112864484)

[3. REQUISITOS 5](#_Toc112864485)

[3.1. REQUISITOS NÃO FUNCIONAIS 5](#_Toc112864486)

[3.2. REQUISITOS FUNCIONAIS 5](#_Toc112864487)

[4. MENSAGENS PRINCIPAIS 6](#_Toc112864488)

[5. REFERÊNCIAS 7](#_Toc112864489)

# RESUMO

Este projeto se relaciona com o aprendizado da utilização de estruturas conhecidas como *Sockets,* muito famosas por representarem a parte fundamental do processo de utilização na programação distribuída e paralela. Por conta da popularização da Internet como um meio de conexão sem fio entre os dispositivos eletrônicos, esse tipo de estrutura se mostrou fundamental para difundir e popularizar aplicações que dependem de uma rede externa para funcionarem, como chats ou mesmo redes sociais. Como base de estudo, este projeto implementa uma estrutura de chat, utilizando a linguagem Java como ferramenta, da qual busca criar um sistema de mensagens de baixa latência, também conhecido como mensagens instantâneas, além de permitir múltiplas conexões em um único servidor que administra todo esse fluxo de troca de mensagens.

**Palavras-chave:** Socket; Mensagens instantâneas; Internet;

# DESCRIÇÃO GERAL DO PROJETO

A tecnologia cresce em um ritmo muito acelerado. Junto com esse crescimento, surgiu a necessidade de conexão entre sistemas, de uma maneira homogênea, e principalmente rápida o suficiente para garantir que os sistemas trabalhem uniformemente, a ponto de o usuário final não conseguir distinguir quando está utilizando um tipo de sistema ou outro. Como uma solução para esse dilema, foi desenvolvido o conceito de Socket. Segundo Claro e Sobral (2008, p.77) "Um socket é uma conexão de dados transparente entre dois computadores em uma rede." A conexão entre os dispositivos se deve a 2 estruturas popularmente conhecidas como "Cliente" e "Servidor". Servidor, para Claro e Sobral (2008, p.77) "Um servidor é um programa que fica executando em uma máquina, possui um nome (host) e este fica aguardando conexões de clientes em uma porta que é identificada por um número". Por sua vez, um cliente se baseia em uma aplicação que se conecta a um servidor. Com os conceitos iniciais elucidados, é realizado a proposta de sistema.

O sistema desenvolvido tem como propósito ser um serviço que será integrado ao projeto da disciplina de Desenvolvimento para Dispositivos Móveis (65DDM). Este sistema, por sua vez, busca replicar certas funcionalidades de redes sociais atuais em uma aplicação, de modo que o usuário possa interagir tanto por meio de postagens, quanto em salas de bate-papo, com trocas e mensagens e/ou chamadas de voz. Por conta do escopo de uma aplicação completa seria algo muito complexo e utilizaria estruturas e serviços que inviabilizaria a integração com Sockets, neste projeto, foi necessário isolar a funcionalidade de chat, com a finalidade de otimizar o aprendizado com a matéria. Estas últimas funções dependem de um serviço em tempo real que possa reconhecer a conexão de usuários, assim como providenciar um meio para envio e recebimento de texto. Portanto, a fim de elencar os assuntos apresentados na disciplina ao projeto citado anteriormente este sistema será desenvolvido com o propósito de conceder a infraestrutura necessária para suprir os requisitos funcionais elencados no desenvolvimento da rede social.

O documento conta com os requisitos funcionais e não funcionais assim como as especificações preliminares das mensagens propostas até o momento, que são exclusivas dos serviços disponibilizados pelo servidor.

# REQUISITOS

## REQUISITOS NÃO FUNCIONAIS

**RNF001 –** O sistema deve ser apresentar alguma forma de integração com a aplicação final, de forma a prestar os serviços requeridos para que esta possa atender seus respectivos requisitos funcionais;

## REQUISITOS FUNCIONAIS

**RF001 –** Verificar usuários online;

RN001: O sistema deve verificar quando algum usuário estiver conectado à aplicação;

**RF002 –** Permitir envio e recebimento de mensagens;

RN002: O sistema deve permitir que, uma vez que um usuário esteja conectado ao serviço, o mesmo possa se comunicar através de mensagens de texto em tempo real.

RN003: As mensagens enviadas por um usuário devem alcançar somente os demais usuários de uma mesma sala de bate-papo, estando estes conectados ao mesmo tempo.

# MENSAGENS PRINCIPAIS

|  |  |
| --- | --- |
| **Mensagem** | “NOME\_USUARIO;MENSAGEM;CONTEUDO\_DA\_MENSAGEM” |
| **Conteúdo** | nome: Nome do usuário que está enviando  operação: "mensagem"  mensagem: A mensagem que será enviada |
| **Descrição** | Envia uma mensagem para todos os usuários |
| **Retorno** | O servidor retorna a mensagem: “Mensagem enviada com sucesso” |

|  |  |
| --- | --- |
| **Mensagem** | “NOME;CONTEÚDO\_NOME” |
| **Conteúdo** | operação: “nome”  mensagem: nome do usuário |
| **Descrição** | Envia o nome do usuário para executar o login dentro do Server Socket |
| **Retorno** | O retorno é baseado em 3 informações que são coletadas em tempo de execução:  “sucesso”: que representa que o login ocorreu de maneira efetiva.  “IP”: que representa o IP do usuário em questão, utilizado para a conexão com o listener de mensagens.  “porta”: que representa uma porta única, utilizada no listener para receber as mensagens.  “nome”: o nome do usuário enviado ao sistema, e retornado para salvar. |

|  |  |
| --- | --- |
| **Mensagem** | “NOME;LISTAR” |
| **Conteúdo** | nome: Nome do usuário  operação: “LISTAR” |
| **Descrição** | Lista todos os usuários online no momento |
| **Retorno** | Retorna uma String com os nomes de todos os usuários separados por “//” |

|  |  |
| --- | --- |
| **Mensagem** | “NOME;SAIR” |
| **Conteúdo** | nome: Nome do usuário  operação: “SAIR” |
| **Descrição** | Desloga o usuário do sistema, removendo listeners e seus respectivos dados. |
| **Retorno** | O servidor retorna uma String com o conteúdo “saindo...” |

# REFERÊNCIAS

Claro D. B. e Sobral J. B. M. PROGRAMAÇÃO EM JAVA. Copyleft Pearson Education. Florianópolis,SC. Disponível em: [**http://www.lasid.ufba.br/teste/pessoal/danielaclaro/download/Programando%20em%20Java.pdf**](http://www.lasid.ufba.br/teste/pessoal/danielaclaro/download/Programando%20em%20Java.pdf) **Acesso em: 28 de Agosto de 2022.**