



UNIVERSIDADE FEDERAL DA FRONTEIRA SUL CAMPUS CHAPECÓ

Ciência da computação

Alunos:

Fernando Schreiner Magnabosco e Pedro Manfio Lill

Aplicação de Bate-Papo com Protocolo MQTT

GEX635 - TÓPICOS ESPECIAIS EM COMPUTAÇÃO XIII (60h)

Chapecó - SC
2023

1. Introdução

Neste trabalho desenvolvemos uma aplicação de chat baseada no protocolo MQTT(Message Queuing Telemetry Transport), optamos por desenvolvê-la usando o JavaScript. O MQTT é um protocolo leve, confiável e amplamente aproveitado em cenários que exigem entrega rápida de mensagens em tempo real. No decorrer do projeto, demonstraremos como essa tecnologia pode ser utilizada para criar uma aplicação de chat que permite comunicação um-a-um e em grupo, fornecendo uma visão prática de seu potencial de implementação em aplicações do mundo real.

2. Objetivo

O objetivo principal do projeto é desenvolver uma aplicação de bate-papo que utilize o protocolo MQTT para permitir comunicação eficiente entre usuários individuais e em grupos. O projeto visa atender aos seguintes objetivos específicos:

- Implementar um sistema de bate-papo que permita comunicação um-a-um e em grupo.

- Utilizar o protocolo MQTT para garantir a eficiência na comunicação.

- Criar uma interface de usuário amigável para interação com a aplicação.

- Gerenciar a persistência de mensagens para usuários offline.

- Manter um controle de status online/offline de usuários.

- Permitir a criação e gerenciamento de grupos de bate-papo.

3. Desenvolvimento

Sistema operacional: Linux.

Biblioteca: Paho para implementação do protocolo MQTT.

Linguagem de Programação: JavaScript

Usamos o comando `python3 -m http.server 5003` para iniciar um servidor web http simples usando o python3. Com esse comando o python inicia um servidor web na porta 5003 (número é optativo) e disponibiliza os arquivos no diretório atual para acesso via HTTP, depois acessamos o endereço `http://localhost:5003/` no navegador para acessar a aplicação.

Ou seja, para utilizar a aplicação, basta clonar o repositório em sua máquina, Iniciar o servidor MQTT com o comando `"python3 -m http.server 5003"`,

iniciar o aplicativo web para inicializar o cliente MQTT e acessar o endereço `http://localhost:5003/` no navegador para acessar a aplicação.

4. Arquitetura do Sistema

O sistema de chat com MQTT é construído com base na arquitetura cliente-servidor, onde os clientes se comunicam por meio de um servidor MQTT central. A arquitetura do sistema é a seguinte:

Clientes MQTT: Cada usuário é representado como um cliente MQTT que se conecta ao broker. Esses clientes podem ser aplicativos ou dispositivos capazes de publicar e assinar tópicos para trocar mensagens em tempo real.

Servidor MQTT: O broker é o componente central do sistema. Ele gerencia a troca de mensagens entre os clientes, mantém os tópicos de controle e garante a entrega confiável das mensagens. O servidor MQTT é responsável por rotear as mensagens entre os clientes e manter a integridade do sistema.

Cada cliente possui um tópico de controle exclusivo identificado pelo seu ID de usuário. Por exemplo, se o ID de um cliente for "yztfds", o tópico de controle será "yztfds_Control."

Quando um cliente deseja iniciar uma conversa com outro usuário, ele publica uma mensagem no tópico de controle deste usuário. Por exemplo, se yztfds deseja falar com cdasds, yztfds publica no tópico "cdasds_Control" para iniciar a solicitação de conversa.

Os principais tópicos são o ID_control: que é um tópico individual para cada cliente. Utilizado para solicitar/negociar novas sessões e comunicação de controle. E o tópico users: um tópico de controle geral, onde os usuários publicam seu status (online/offline).

Os tópicos de chat, tanto para conversas individuais como em grupo, são dinamicamente criados através de identificadores únicos (UUIDs). Esses identificadores são comunicados via publicação nos respectivos tópicos de controle.

Essa arquitetura proporciona uma comunicação eficiente entre os usuários, permitindo a criação dinâmica de tópicos de chat exclusivos para cada interação. O uso de tópicos específicos facilita o controle e a organização das mensagens, garantindo uma experiência de bate-papo robusta e escalável.

4.1 Cliente

O componente cliente MQTT é uma classe derivada da biblioteca Paho MQTT, responsável por gerenciar a comunicação entre os usuários.

Cada cliente possui um ID único, um status que indica se está online, offline ou em estado desconhecido para outros usuários. Além disso, mantém informações sobre chats individuais, grupos e uma lista de solicitações pendentes.

4.2 Chats Individuais e Grupos

A classe Client mantém um dicionário denominado "chatUsers", onde as chaves representam os nomes dos chats (usuários individuais ou nomes de grupos) e os valores são instâncias das classes Chat ou Group.

Chats individuais são representados por instâncias da classe Chat, enquanto grupos são instâncias da classe Group.

A classe Chat contém informações sobre o tópico e mensagens trocadas. Mensagens são armazenadas em dicionários dentro das classes Chat e Group, sendo cada mensagem identificada por um ID único e o Timestamp + Remetente.

Grupos mantêm informações como tópico, data de criação, membros e mensagens. Membros são representados por objetos da classe Member, contendo informações sobre o ID do usuário e permissões no grupo

4.3 Solicitações Pendentes:

As solicitações pendentes, como pedidos de chat ou convites para grupos, são armazenadas na lista "requests" dentro da classe Client.

Cada solicitação é representada por um dicionário contendo informações como remetente, conteúdo da mensagem, e timestamp com data e hora

5. Descrição das funcionalidades

Inicialização de Usuário:

A função User inicializa um usuário com suas propriedades e se conecta a um broker MQTT.

Inicialização da Interface do Usuário (UI):

O método initUI inicializa os elementos da interface do usuário e configura manipuladores de eventos para interações do usuário.

Interações do Usuário:

Os usuários podem iniciar chats, aceitar solicitações de chat, enviar mensagens e visualizar o histórico de chat.

Os usuários podem criar grupos, aceitar ou rejeitar pedidos de associação a grupos e visualizar os membros do grupo.

Comunicação MQTT:

Os métodos connect e disconnect lidam com a conexão e desconexão MQTT. O método onConnect se inscreve em tópicos MQTT relevantes e publica o status online do usuário.

Funcionalidade de Chat:

Os usuários podem iniciar um chat, aceitar solicitações de chat e enviar mensagens. Existe um modal para exibir e enviar mensagens de chat.

Funcionalidade de Grupo:

Os usuários podem criar grupos, visualizar informações de grupo e conversar dentro dos grupos. Pedidos de associação a grupos podem ser aceitos ou rejeitados.

Manipulação de Eventos:

Vários métodos lidam com eventos, como recebimento de mensagens, atualizações de status do usuário e mensagens de grupo.

Histórico e Registro:

O sistema mantém o histórico relacionado a usuários e grupos, incluindo chat e pedidos de associação.