ESCOLA POLITÉCNICA
SISTEMAS DE INFORMAÇÃO

BRUNO PENTEADO CARRARA
FILIPE DIAS CARVALHO DE AZEVEDO
GABRIEL HENRIQUE DA SILVA
RAPHAELA DE SOUZA RIBEIRO

RELATÓRIO DO PROJETO:

Trabalho Prático 01 - Desenvolvimento de Aplicação Utilizando Protocolos de Mensageria Mosquitto (MQTT)

Link para repositório: REPOSITÓRIO

Orientador: Prof Douglas Henrique Siqueira Abreu.

CAMPINAS
2024

1. DESCRIÇÃO DA APLICAÇÃO DESENVOLVIDA

A aplicação desenvolvida é um sistema de chat em tempo real que permite a comunicação entre múltiplos usuários através de uma rede. Utilizando o protocolo MQTT (Message Queuing Telemetry Transport) em conjunto com o broker Mosquitto, a aplicação permite que clientes enviem e recebam mensagens instantaneamente, promovendo uma interação dinâmica.

O sistema possui várias características principais que o tornam eficiente e funcional. Primeiramente, oferece comunicação em tempo real, garantindo que as mensagens sejam entregues instantaneamente a todos os usuários conectados. Além disso, suporta a conexão simultânea de múltiplos clientes, permitindo que diversos usuários interajam ao mesmo tempo. A interface é simples e operada via linha de comando, facilitando tanto o uso quanto a compreensão do seu funcionamento. Por fim, a arquitetura do sistema é escalável, permitindo expansão para suportar um número maior de usuários e funcionalidades adicionais.

2. JUSTIFICATIVA DA ESCOLHA DO PROTOCOLO (MQTT COM MOSQUITTO)

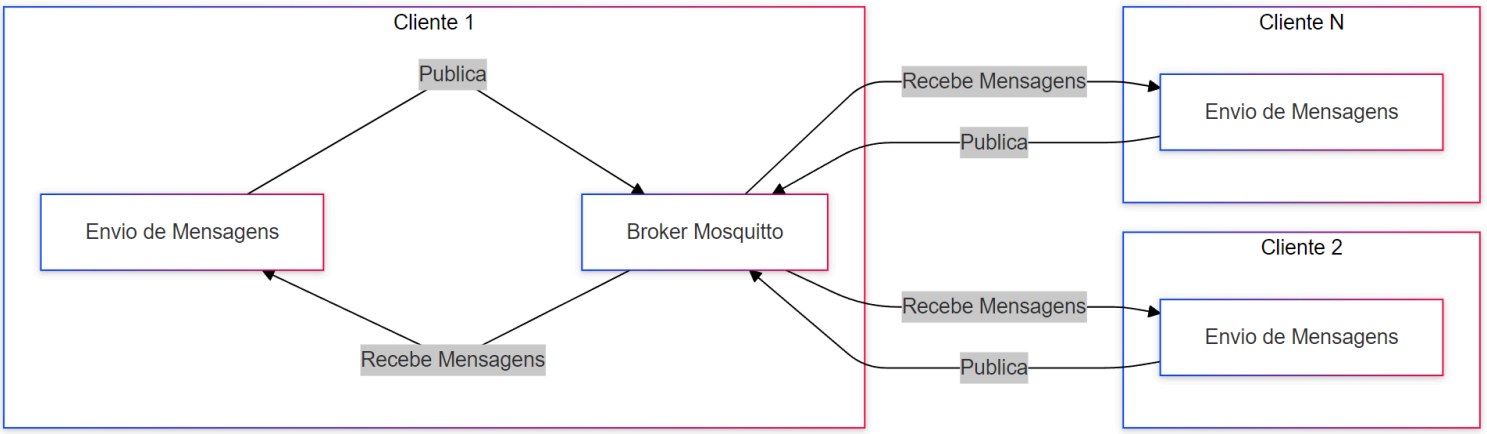
A escolha do protocolo MQTT com o broker Mosquitto foi fundamentada em várias razões. Primeiramente, a leveza e eficiência do MQTT são notáveis; sendo projetado como um protocolo leve com cabeçalhos mínimos, ele reduz o uso de banda e recursos computacionais, proporcionando alto desempenho. Isso o torna ideal para aplicações que requerem comunicação rápida e eficiente, como sistemas de chat em tempo real.

Além disso, o modelo publish/subscribe do MQTT oferece um desacoplamento entre os clientes. Esse modelo permite que os clientes publiquem mensagens em tópicos específicos sem a necessidade de conhecer outros clientes, facilitando a escalabilidade e a manutenção do sistema. A flexibilidade na comunicação é outro benefício, pois os usuários podem se inscrever em diferentes tópicos, permitindo a criação de múltiplas salas de chat ou canais temáticos.

A simplicidade de implementação é uma vantagem significativa. Com a biblioteca Paho-MQTT para Python, a implementação de clientes MQTT torna-se direta e acessível. O Mosquitto, sendo um broker open-source amplamente utilizado, oferece vasta documentação e suporte da comunidade, o que facilita ainda mais o desenvolvimento.

Por fim, em comparação com o AMQP e o RabbitMQ, o MQTT apresenta benefícios claros para este contexto. O AMQP com RabbitMQ é mais complexo e robusto, adequado para aplicações que exigem transações confiáveis e recursos avançados, o que pode ser excessivo para um simples sistema de chat. Além disso, o MQTT apresenta menor overhead em comparação com o AMQP, tornando-o mais apropriado para aplicações que priorizam desempenho e simplicidade.

3. DIAGRAMA DA ARQUITETURA DA APLICAÇÃO



O diagrama descreve um sistema de comunicação baseado no protocolo MQTT, envolvendo clientes, um broker Mosquitto e o fluxo de mensagens entre eles.

Os Clientes(Cliente 1, Cliente 2, ..., Cliente N) são instâncias de um script Python em execução em diferentes terminais. Cada cliente representa um usuário distinto, participando do sistema de comunicação de forma independente. Esses clientes se conectam ao broker para enviar e receber mensagens, interagindo através de tópicos específicos.

O Broker Mosquitto funciona como o servidor central do sistema. Ele é responsável por gerenciar o recebimento e a distribuição das mensagens que os clientes enviam. Quando um cliente publica uma mensagem em um tópico, o broker garante que ela seja encaminhada corretamente para todos os clientes inscritos nesse mesmo tópico.

O Fluxo de Comunicação no sistema se dá da seguinte maneira:

- Publicação: Os clientes enviam mensagens ao broker utilizando um tópico de chat específico. Cada mensagem publicada é direcionada ao broker que centraliza essas informações.
- Recepção: Após receber as mensagens, o broker as distribui para todos os clientes que estão inscritos no tópico, garantindo que todos os participantes recebam as informações enviadas em tempo real.

Dessa forma, o sistema permite que os usuários se comuniquem de maneira eficaz, com o broker Mosquitto atuando como intermediário no processo de troca de mensagens.

4. APLICAÇÃO DE CONCEITOS DE REDES DE COMPUTADORES NO DESENVOLVIMENTO

O desenvolvimento desta aplicação incorporou diversos conceitos fundamentais de redes de computadores.

Protocolo de Comunicação (MQTT): O MQTT opera na camada de aplicação do modelo OSI, possibilitando a troca de mensagens entre terminais conectados em rede. Para garantir a entrega confiável das mensagens, utiliza os protocolos subjacentes TCP/IP.

Modelo Cliente-Servidor: O broker Mosquitto atua como servidor central, responsável por receber e distribuir mensagens. Os clientes, por sua vez, desempenham o papel de produtores ao enviar mensagens e de consumidores ao receber mensagens distribuídas pelo broker.

Publicação/Assinatura (Publish/Subscribe): Esse modelo permite o desacoplamento temporal e espacial entre os clientes, que não precisam estar cientes da existência uns dos outros nem conectados simultaneamente para trocar mensagens. Isso proporciona escalabilidade e flexibilidade, facilitando a adição de novos clientes e tópicos sem necessidade de grandes alterações na infraestrutura.

Sockets e Conexões de Rede: A comunicação é estabelecida por meio de conexões TCP, que garantem canais de comunicação confiáveis entre clientes e broker. A porta de rede padrão utilizada para comunicação MQTT é a 1883, essencial para o roteamento correto das mensagens.

Programação Concorrente: No lado do cliente, a aplicação implementa threads para possibilitar o envio e a recepção simultânea de mensagens, o que melhora a responsividade e eficiência do sistema.

5. INSTRUÇÕES PARA A EXECUÇÃO E TESTES DA APLICAÇÃO

O código fonte, assim como um vídeo prático da execução podem ser encontrados no repositório -> Repositório

Pré-requisitos:

Para o funcionamento adequado da aplicação, é necessário garantir que alguns componentes estejam instalados e configurados. Primeiro, certifique-se de que o Python 3.x está instalado no sistema. Em seguida, instale a biblioteca Paho-MQTT executando o comando `pip install paho-mqtt`. Além disso, é fundamental que o broker Mosquitto esteja instalado e corretamente configurado.

Passos para Execução:

1. Instalar o Broker Mosquitto:

Baixe o instalador disponível em Mosquitto Download e siga as instruções fornecidas pelo instalador.

2. Iniciar o Broker Mosquitto:

Abra a pasta fonte onde o mosquito foi instalado e rode o comando `mosquitto`
mosquitto

3. Configurar o Client:

Baixe o arquivo client.py e se necessário troque o valor de broker_address(porta). A padrão é a 1883.

4. Executar múltiplas instâncias para teste:

   Abra um terminal onde estiver o código client.py, de um split no terminal quantas vezes quiser e por fim rode o programa através de `python client.py` .

   Após isso, insira um nome para cada terminal como solicitado. E pronto, ele indicará que conseguiu conectar e você está livre para enviar mensagens.