

a) Descrição do projeto

O projeto final da disciplina de Redes de Computadores será a implementação de um sistema de monitoramento e controle da iluminação de ambientes. Será utilizado o protocolo MQTT para a comunicação entre o sensor, servidor e cliente.

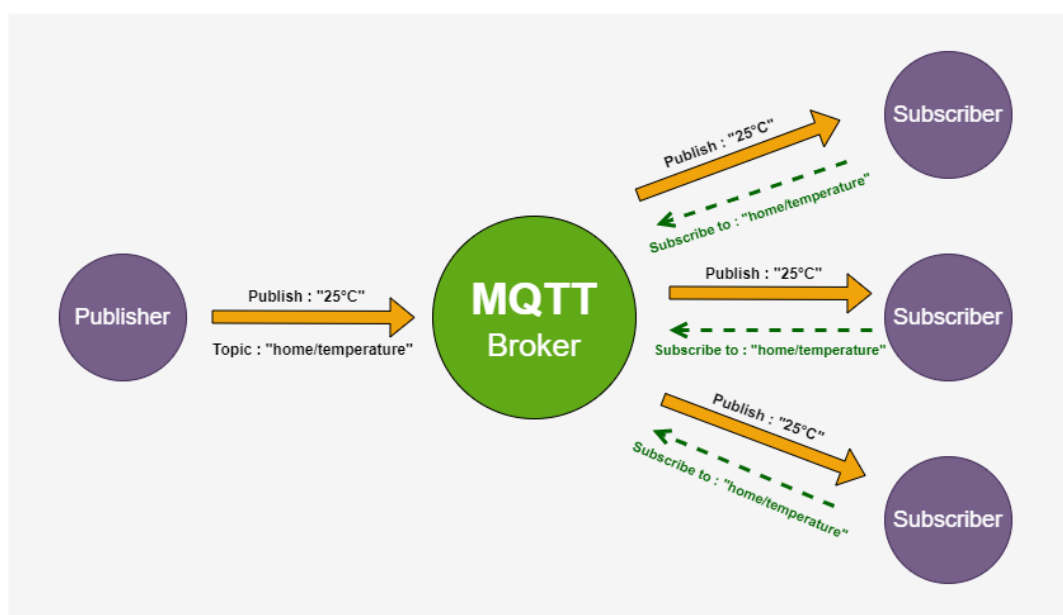


Figura 1 - Comunicação MQTT

(Fonte: <https://dzone.com/articles/eclipse-mosquitto-connector-for-jolie-language>)

O MQTT funciona com o paradigma de conexão publisher/subscriber. Os dados são associados a tópicos, como "home/temperature", ilustrado na figura 1. O pacote MQTT que contém o dado de temperatura só será recebido e lido caso algum dispositivo tenha enviado ao servidor (Broker) a requisição de inscrição (Subscriber) naquele tópico específico. Todo redirecionamento e comunicação é feito pelo broker MQTT.



### a.1) Cenário de aplicação

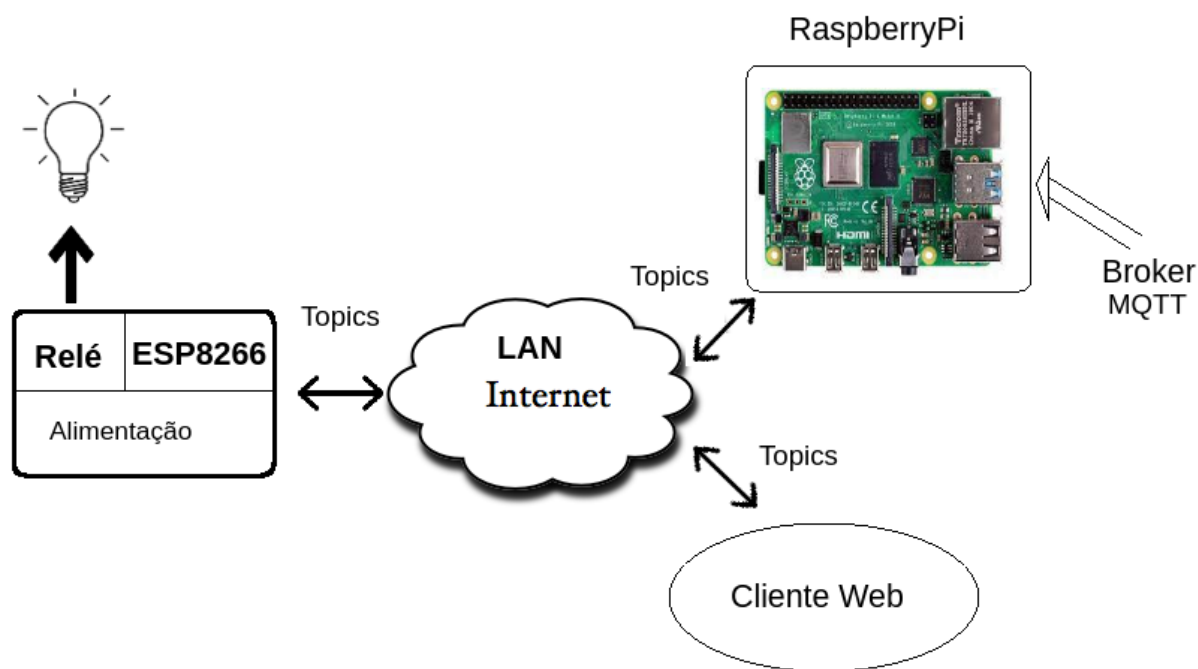


Figura 2 - Sistema de monitoração usando MQTT

Conforme a figura 2, o módulo ESP8266 será utilizado para fazer a conexão com a internet e enviar/receber os dados MQTT para/de um servidor (broker). Este também controlará o relé que ficará conectado no circuito que energiza as lâmpadas.

O broker MQTT será instalado em uma placa Raspberry Pi (RPI), que deverá ser responsável pela conexão com o hardware de controle e monitoração.

Uma interface WEB deverá ser desenvolvida para a monitoração e acionamento do relé. O serviço web pode ser executado diretamente na RPI ou em alguma plataforma de serviço (PaaS).

### b) Etapas

#### b.1) Implementar o acionamento do relé

Codificar em linguagem C a parte de acionamento do relé que conecta a alimentação das lâmpadas. Lembrando que a escrita e leitura de uma porta de I/O é o suficiente para esta tarefa.

#### b.2) Implementação MQTT no ESP8266: publicações e inscrições nos tópicos de monitoramento e acionamento.



Codificar em linguagem C a comunicação MQTT com o broker através de um link **criptografado**. Tópicos com o status do relé e comando de acionamento deverão ser implementados.

b.3) Instalação e operação do broker MQTT

Instalar algum broker *open source* (Eclipse Mosquitto é um excelente broker) na Raspberry Pi. Para verificar o funcionamento, pode-se testar a comunicação implementada na etapa b.2.

b.4) Visualização e controle do status dos relés a partir do acesso de um cliente remoto via HTTP.

Uma vez implementada a etapa b.3, deve-se visualizar o status de iluminação do ambiente. Nesta etapa é necessário codificar ou utilizar uma página Web para visualização das informações sobre a luminosidade. O servidor web pode ser desenvolvido e executado em alguma plataforma em nuvem, pode-se usar o próprio Web Client de validação do Broker ou criar hospedar e criar a visualização web na própria placa Raspberry Pi. A ideia é que se tenha uma interface amigável para monitorar o status das lâmpadas/luminárias do ambiente.

c) Cronograma de entrega

**Todas** as etapas (b.1, b.2, b.3 e b.4) deverão ser implementadas e integradas até dia 16 de Dezembro de 2022.

d) Modificações

d.1) O ESP8266 (ESP01) utilizado no hardware deve rodar o RTOS desenvolvido pelo fabricante Espressif. Os guidelines para configuração do SDK está em:

<https://docs.espressif.com/projects/esp8266-rtos-sdk/en/latest/get-started/index.html>

d.2) A interface WEB de visualização dos dados deve possuir uma tela inicial com campos para login e senha, validados através de um banco de dados.

d.3) Um histórico do status de iluminação deve ser gerado, apresentando data, hora e status do hardware (iluminação On/Off).

d.4) Um web server deve ser criado e configurado para armazenar os objetos web que compõem a interface de acesso aos hardwares que controlam a iluminação.