Universidade Federal do Ceará – UFC
Centro de Ciências – CC
Mestrado e Doutorado em Ciências da Computação - MDCC
Sistemas Distribuídos

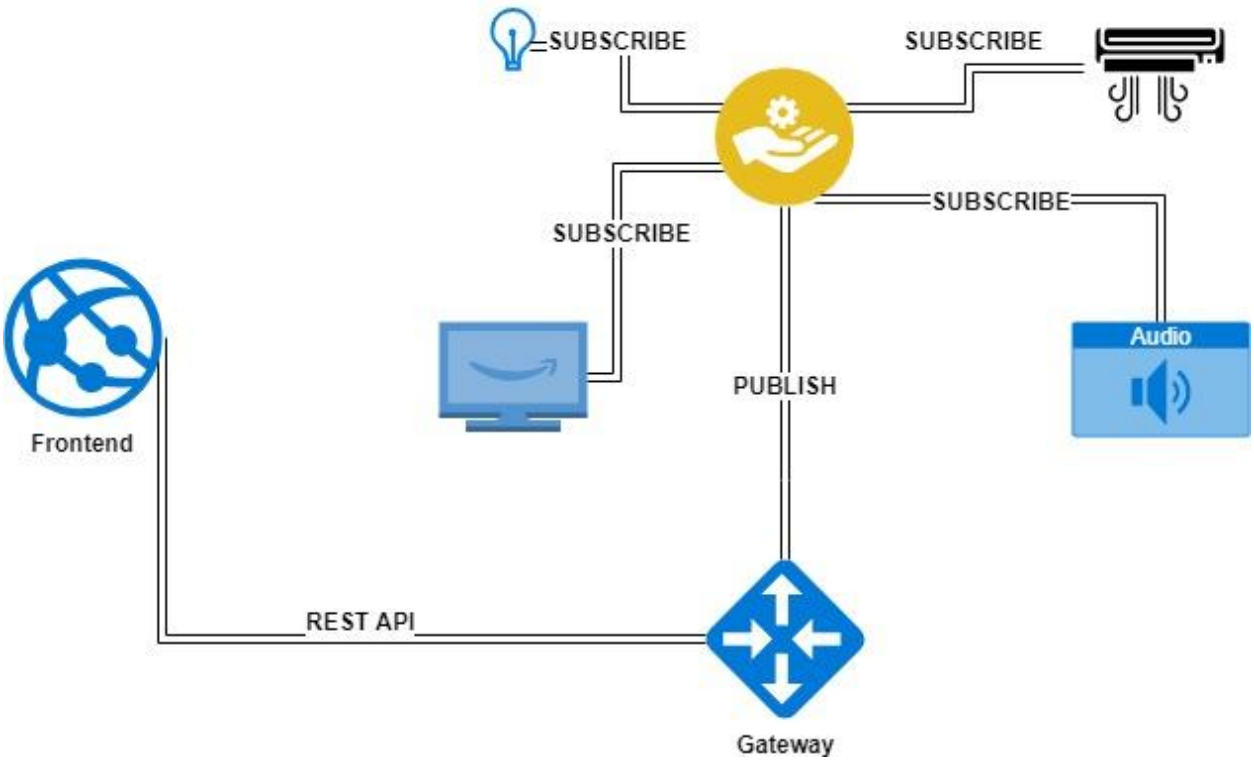
Trabalho: MQTT X Socket
Objetivos: Exercitar os conceitos de Complexidade de Algoritmos.

NOME: Antonio Alex Monteiro de Sousa **MATRÍCULA:** 409542
Repositório git: https://github.com/aalexmonteiro/domotic

MQTT X SOCKET

Esse relatório tem o objetivo de descrever como foi realizada a implementação do trabalho de MQTT vs. Socket, mostrando a arquitetura que a solução foi implementada, as tecnologias utilizadas e por fim as diferenças entre os modelos de implementação.

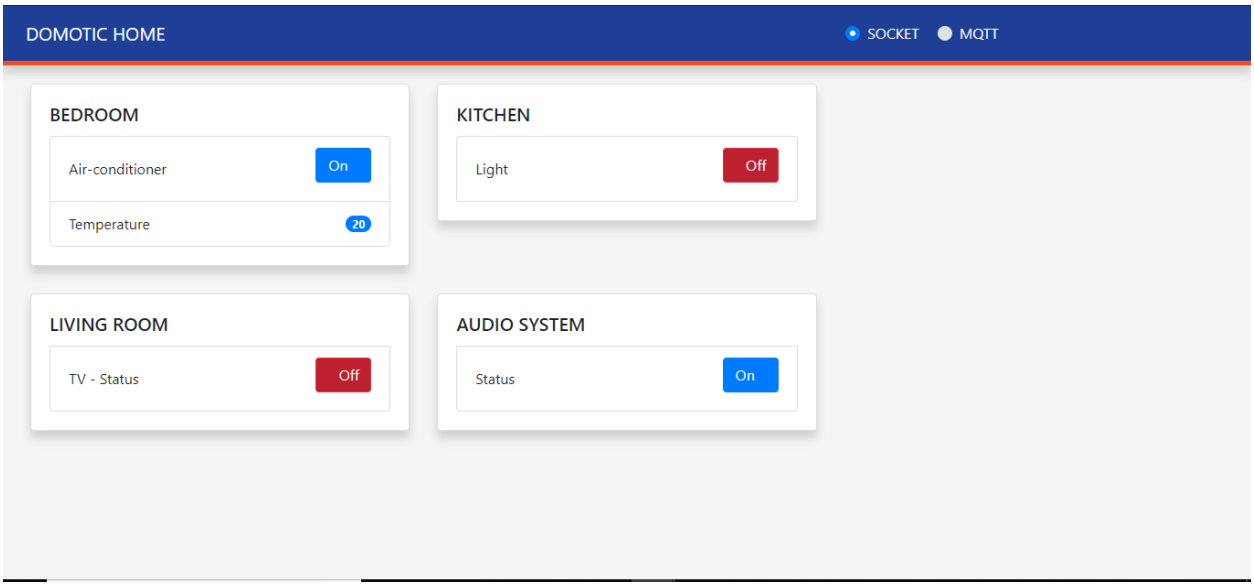
Desenho da Arquitetura



Na figura acima podemos visualizar a arquitetura criada. A solução escolhida foi uma aplicação web no qual o frontend se comunica com o gateway através de uma REST API. No gateway foram desenvolvidos 4 objetos simulados (ar-condicionado, lâmpada, TV e sistema de áudio). O

objeto real não foi implementado nessa solução. A ação desenvolvida nos objetos simulados foi o ligamento e desligamento desses objetos. Para ajudar na simulação, foi utilizado o Redis, um banco NO-SQL que pode armazenar somente em memória. O motivo do uso do Redis, foi para guardar o estado (ON/OFF) dos objetos. Foram feitos dois tipos de implementação, uma em socket utilizando a biblioteca nativa do Python que foi a linguagem utilizada na solução e outra implementação em MQTT utilizando o Paho MQTT. Na implementação com MQTT, foi utilizado um broker instalado e configurado na máquina local.

Frontend da Aplicação



Tecnologias Utilizadas

Na solução implementada foram utilizadas as seguintes tecnologias:

- Frontend:
  - HTML5
  - JavaScript (JQuery)
  - Bootstrap
- Gateway:
  - Python 3.5
  - Flask (Rest API)
  - Paho MQTT (Cliente MQTT em Python)
  - Mosquito MQTT Broker (instalado na máquina local)
  - Biblioteca nativa socket do Python
  - Redis (Armazenamento em memória para guardar o estado dos objetos IoT simulados)

Diferenças entre os modelos de programação

Os sockets são mais complexos de implementar e gerenciar do que o MQTT que é bem mais simples.

No socket é aberto uma comunicação cliente para se comunicar com um servidor que está em um endereço IP e porta específicos. Para um socket enviar uma mensagem para o servidor, é necessário ter a informação de qual endereço IP e porta o servidor está sendo executado. Na implementação, cada objeto simulado tinha um IP e porta específico, no qual quando era necessário executar alguma ação, o socket devia ser aberto para se comunicar com o IP e porta do objeto que se desejava comunicar. Desenvolver em socket é mais complexo do que o MQTT, pois se codifica mais e o seu gerenciamento é mais complicado.
No MQTT, a implementação foi bem mais simples, pois a configuração dos subscribers e publishers é bem simples. Em poucas linhas é possível desenvolver essa solução. Para isso, basta associar os subscribers e publishers a um broker que está sendo executado em um endereço IP e porta específica. No subscriber, é criado um tópico no qual as mensagens publicadas pelos publishers serão recebidas. No publisher são publicadas as mensagens para um tópico específico que foi criado em um subscriber.
Ao fim da implementação foi possível verificar que MQTT é um modelo melhor do que o modelo usando sockets, além de ser eficiente e leve.