Trabalho Prático 1: Comunicação *Publish*/*Subscribe*

Augusto Sergio dos Santos
Bruno Goulart Meneghin
Lucas Batista Magalhães Carvalhais

1 - Introdução

A Internet das Coisas visa conectar dispositivos eletrônicos comuns utilizados no dia a dia através da internet. Essa comunicação é feita através de trocas de mensagens através de rede, como, por exemplo, um sensor pode detectar um movimento no ambiente e enviar mensagem para acender uma lâmpada. Outro uso é um servidor de mensagens, em que um usuário envia uma mensagem de um dispositivo e outro usuário a recebe em seu próprio dispositivo.

Nesse último exemplo pode-se pensar que nem sempre um usuário vai estar conectado a rede, e se alguém lhe envia alguma mensagem ela deve ser guardada até que ele possa recebe-la. Um dos protocolos usados para a esse tipo de situação é o MQTT (*Message Queuing Telemetry Transport*). Baseado no modelo Publish/Subscribe o MQTT recebe publicações de clientes para que outros clientes possam acessar em tempo real. O uso deste protocolo será apresentado ao longo deste trabalho.

2. Projeto do Sistema

O projeto do sistema se divide em três partes:

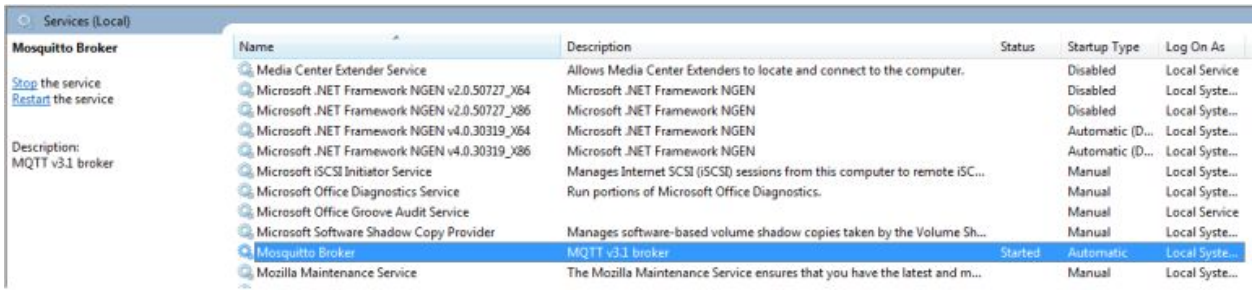
1.Broker MQTT

No protocolo MQTT Publish/Subscribe as mensagens são armazenadas num servidor que gerencia os eventos de publicação e transmissão de mensagens, e esse servidor é o Broker.

O Broker funciona como um nó intermediário na comunicação entre dois dispositivos, e o uso desse tipo de serviço gera desacoplamento desacoplamento do espaço (não há necessidade dos clientes conhecerem um do outro), desacoplamento do tempo (clientes não precisam estar *online* para enviar e receber, podem realizar a qualquer momento) e é assíncrono (não há tempo global).

No caso desse projeto foi utilizado o broker Mosquitto, que é um Broker open source que implementa o protocolo MQTT.

O broker funciona como um serviço rodando em background na máquina, que fica escutando na porta 1883.



Dessa forma qualquer cliente (nesse ponto clientes na mesma rede) podem publicar mensagens ou dar subscribe em algum tópico de interesse.

2.Android MQTT

Para efetuar a comunicação a partir de um dispositivo android foi utilizado o *Eclipse Paho*, que é um projeto Open Source que disponibiliza a implementação do protocolo MQTT para clientes voltado a Internet das Coisas.

Para funcionar no Android foi necessário importar o repositório Maven "https://repo.eclipse.org/content/repositories/paho-snapshots/" e os repositórios "org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.0.2"

e

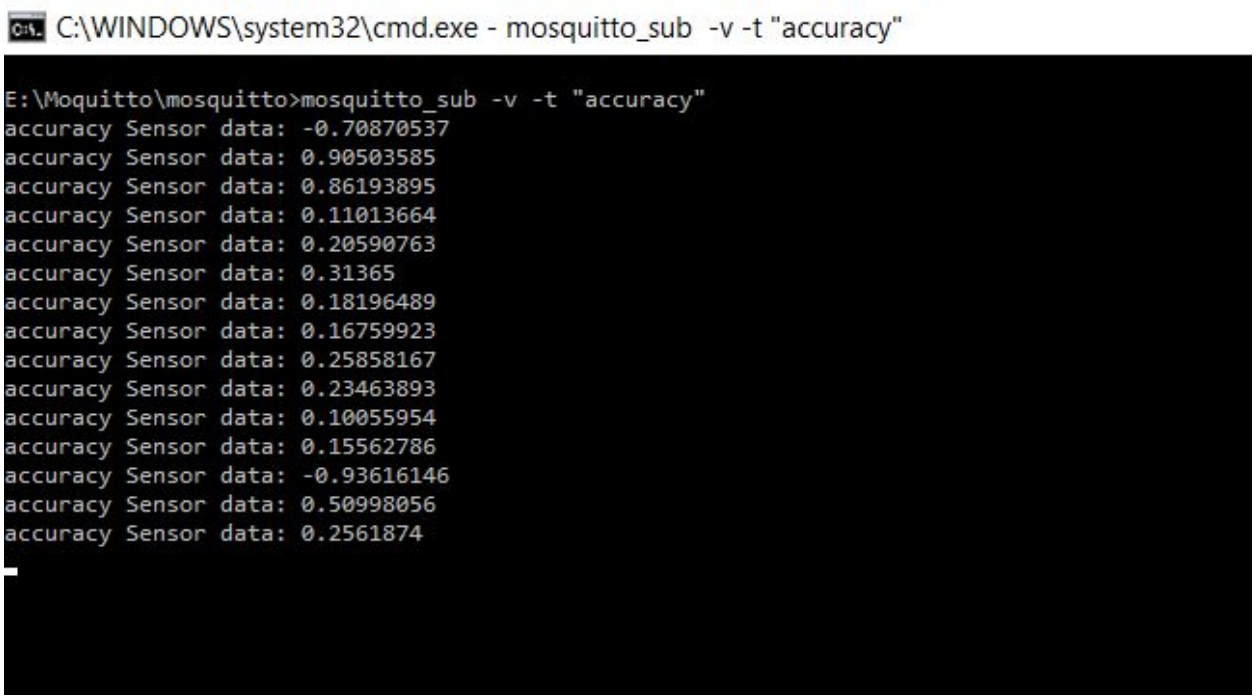
"Org.eclipse.paho:org.eclipse.paho.android.service:1.0.2".

Além disso o Android precisa saber que tem que rodar o serviço Mqtt quando a aplicação estiver rodando, e por isso foi necessário adicionar ao Manifest do aplicação o serviço MQTT para que fosse iniciado ao abrir.

Já na atividade principal da aplicação é feito o seguinte:

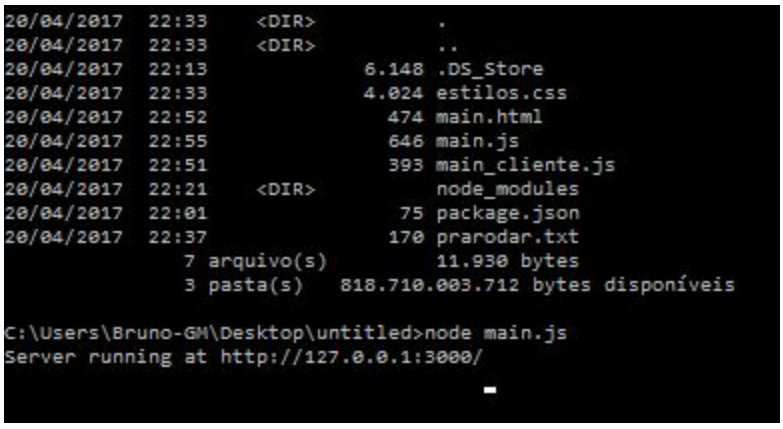
1. Instaciamos um objeto cliente que leva como parâmetro o IP do broker com sua respectiva porta, o ID do cliente e seu modelo de persistência.
2. Instaciamos um objeto de configuração de opções de conexão entre o cliente e o Broker.
3. Colocamos a flag "CleanSession" com True, pois se a conexão for perdida por algum motivo os dados de conexão são limpos do lado do cliente.
4. Utilizando o objeto de configuração criado no tópico 2 é feita a conexão com o broker.
5. Após a conexão criamos a mensagem Mqtt que será enviada no formato correto.
6. O cliente faz o publish no broker no tópico específico.
7. A conexão é desfeita.

Utilizamos o *sensor.TYPE_ALL do Android para publicar a acurácia dos sensores no broker em um tópico, e em um terminal que foi subscribed nesse tópico chegamos ao seguinte resultado:*



O terminal cliente recebeu o dado do tópico (accuracy) ao que estava inscrito, mostrando assim a comunicação realizada entre um Publisher (Smartphone), o Broker (Mosquitto) e um Subscriber (Terminal rodando “Mosquitto_sub” e inscrito no tópico).

3.Servidor Web MQTT



Para a implementação do Servidor Web foi utilizado node para o Back-end e HTML, CSS, javascript para o Front-end. O Back-end ficou responsável pelo gerenciamento de mensagens MQTT vindas do Broker e também pela integração entre o browser e o

Broker. O Front-end ficou responsável por atribuir estilo com CSS e exibir as mensagens dos tópicos subcribed no browser.

Foram necessárias as instalações do MQTT e npm, para assim executar as funcionalidades necessárias no Servidor web.

3. Dificuldades

Ao longo do desenvolvimento encontramos problema na comunicação entre o broker e clientes, pois inicialmente a conexão entre os dois não acontecia. Porém detectamos que era uma configuração de rede que estava impedindo tal conexão, e depois de configurado tudo ocorreu normalmente.

Outra dificuldade foi configurar as dependências e repositórios do Android Studio no primeiro momento, pois certas bibliotecas estavam causando conflito e gerando a quebra do aplicativo em tempo de execução, e foi necessário gastar um tempo pesquisando a solução desse problema, porém quando as referências certas foram encontradas tudo funcionou corretamente.

4.Conclusão

Nessa etapa do projeto foi possível absorver de forma satisfatória como o protocolo funciona e testar, pela primeira vez, a troca de mensagem entre sistemas diferentes. Apesar das dificuldades encontradas que geraram o atraso na entrega foi bom para entender como as entidades se comunicam e o que é necessário para tal utilizando MQTT Publish/Subscribe.