

Projeto de Graduação



06 de Dezembro de 2021

## **Atuação remota de objetos inteligentes e a latência do meio para estudos de eficiência**

Felipe Nogueira de Souza

## **Atuação remota de objetos inteligentes e a latência do meio para estudos de eficiência**

**Aluno(s): Felipe Nogueira de Souza**

**Orientador(es): Markus Endler**

Trabalho apresentado com requisito parcial à conclusão do curso de Engenharia de Controle e Automação na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

## **Agradecimentos:**

Gostaria de fazer um breve agradecimentos a todos que participaram, ou ainda participam, do LAC - Laboratory of Advanced Collaboration - e que me motivaram sem saber a continuar em minha jornada acadêmica até o fim da mesma. Espero que pessoas interessantes continuem a frequentar o laboratório e tenham uma experiência tão boa e produtiva quanto eu e outros alunos que passaram por lá .

## Resumo

No Laboratory of Advanced Collaboration (LAC), da PUC - Rio, foi publicado um artigo sobre controle de objetos inteligentes através de Smart Phones, Middleware Support for Generic and Flexible Actuation in the Internet of Mobile Things. Através desse estudo e pesquisa foi criada uma nova versão do Mobile Hub, software de coleta, processamento de dados e atuação remota na internet das coisas, e muitas ideias para testes e pesquisa surgiram.

A necessidade de um programa com serviço de controle integrado ao gerenciador de sensores e atuadores dos objetos inteligentes surgiu, e para testar a eficiência do middleware ContextNet, em questão de latência, escolhi usar o protocolo MQTT para fins de comparação com a utilização do middleware do LAC, assim testando especificamente a aptidão do projeto para atuação remota.

**Palavras-chave:** Internet das Coisas, Internet das Coisas Móveis, Atuação Remota

## Sumário

- 1. Introdução - página 1**
- 2. ContextNet e o Mobile Hub - página 2**
- 3. Mact, SOM e o Mact User - página 4**
- 4. Novo Mobile Hub e a implementação do teste - página 9**
- 5. Conclusão - página 12**
- 6. Bibliografia - página 13**

## Lista de Figuras:

- 1. Arquitetura Contextnet [5] - Página 1**
- 2. Esquema Mobile Hub [5] - Página 3**
- 3. Mact User Imagem 1 - Página 4**
- 4. Mact User Imagem 2 - Página 5**
- 5. Mact User Imagem 3 - Página Página 5**
- 6. Mact User Imagem 4 - Página 6**
- 7. Mact User Imagem 5 - Página 6**
- 8. Mact User Imagem 6- Página 7**
- 9. Arquitetura Mact User - Página 8**
- 10. Arquitetura Mact User com MQTT - Página 8**
- 11. Resultado Teste 1 - Página 10**
- 12. Resultado Teste 2 - Página 10**
- 13. Resultado Teste 3 - Página 11**

## 1 Introdução

### 1.1 Motivação:

Esse trabalho se insere no contexto de Internet das Coisas (ou Internet of Things - IoT) e tem como objetivo criar e avaliar um serviço para atuação genérica em dispositivos IoT que atendem uma demanda específica do Laboratory for Advanced Collaboration (LAC) e seu middleware ContextNet [1]

A Internet das coisas se refere a uma evolução tecnológica dos sistemas em rede e de Computação Pervasiva que tem como objetivo conectar uma grande parcela de aparelhos e dispositivos usados no nosso dia a dia à rede Internet. Cada vez mais surgem eletrodomésticos, veículos, robôs, drones, e dispositivos vestíveis (wearables), roupas e maçanetas conectadas à Internet, além de outros dispositivos, como computadores convencionais e smartphones.

O LAC vem trabalhando em específico com IoMT - *Internet of Mobile Things* - que se baseia no contexto de dispositivos inteligentes móveis, e como devemos interagir com esses dispositivos para conseguirmos extrair dados relevantes ou até atuar sobre dispositivos inteligentes de maneira segura. Os trabalhos do laboratório se pautam sobre dois pilares: o *Mobile Hub* [2], e o *ContextNet*, o pilar principal onde se implementa a comunicação de nós, sejam nós móveis (*smartphones*, por exemplo) ou estáticos, via o SDDL - *Scalable Data Distribution Layer* - core, logo, toda a comunicação com a internet se dá via ContextNet.



Arquitetura Contextnet [5]

O Mobile Hub atual é capaz de se comunicar com dispositivos inteligentes e atuar sobre eles utilizando tecnologias de rádio de curto alcance. Para isso uma diversidade de serviços e programas-cliente devem existir, sendo um destes programas-cliente o foco deste trabalho. A necessidade de um programa-cliente para enviar mensagens de atuação - sejam automáticas, ou enviadas manualmente - surgiu e em conjunto a necessidade de criar um ambiente de testes para medir o quão eficientemente a comunicação entre os programas funciona.

### 1.1 Objetivo:

O mesmo cliente que obtém os atuadores móveis disponíveis também deve enviar as mensagens de atuação. O primeiro experimento do *Mobile Hub* voltado para atuação remota utilizou o cliente que desenvolvi, que na época foi chamado de *Mact User - Mobile Actuation Service User - um programa desenvolvido em Java, explicarei mais adiante sobre o Mact Service e o Mact User*. Um teste de latência, que consistiu em medir *em quanto tempo um driver para atuação chega no Mobile Hub*

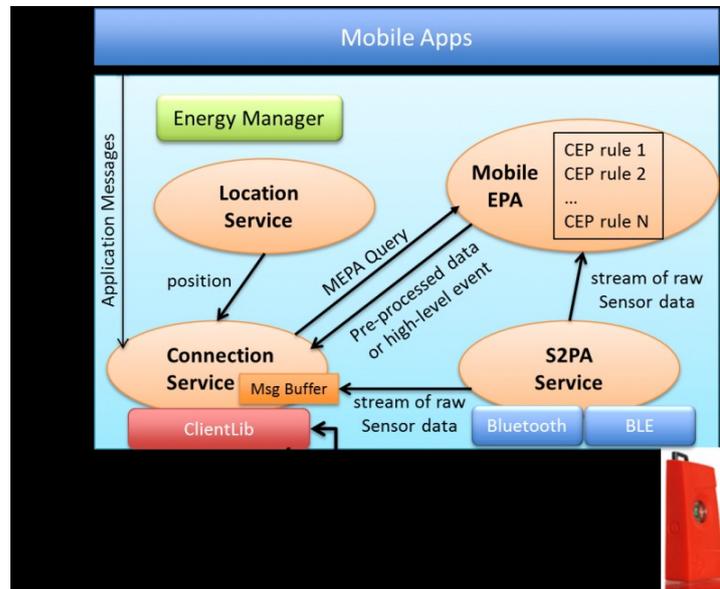
*após ser requisitado pelo mesmo*, este teste foi feito para testar se este projeto de atuação remota era de fato viável.

O problema deste teste de latência é que ele seria muito mais rico para a continuação do desenvolvimento, do *ContextNet*, *Mobile Hub* e do programa cliente caso houvesse uma perspectiva diferente para comparação de desempenho; sendo muito mais rico para publicação de futuros artigos.

Aproveitando um novo desenvolvimento do *Mobile Hub*, *Mobile Hub 2.0*, implementei no *Mact User* uma comunicação MQTT que se dá fora do contexto do *ContextNet*. Assim propus e desenvolvi uma comunicação MQTT entre o cliente emissor de mensagens de atuação e o *Mobile Hub*, com o objetivo de medir a latência de uma mensagem de atuação enviada do cliente ao *Mobile Hub*, utilizando especificamente a biblioteca Eclipse Paho, implementação do protocolo MQTT desenvolvido pela Eclipse Foundation.

## 2 ContextNet e o Mobile Hub

O *Mobile Hub* é um projeto que se iniciou com intuito de tornar o smartphone pessoal em um nó móvel na ponta de uma rede de dispositivos inteligentes, coletando, e processando via CEP - Complex Event Processing -, dados de sensores de dispositivos inteligentes; a comunicação destes *smartphones* com os dispositivos inteligente se dá via rádio de curto alcance (BLE, Zigbee e etc), através de um serviço interno do *Mobile Hub* chamado S2PA, que lida tanto com o descobrimento tanto quanto com a conexão e a comunicação com os dispositivos inteligentes. Essa arquitetura de serviços abrange toda estrutura do *Mobile Hub*, podendo-se escalar o projeto de maneira a conter mais serviços caso necessário. Os principais serviços do *Mobile Hub*, sendo também os cruciais para seu funcionamento, são o próprio S2PA, explicado anteriormente, o *Location Service* que cuida da obtenção e da localização do dispositivo móvel, o *MEPA Service* - que cuida da criação das regras CEP e do funcionamento das mesmas para o processamento de dados, e o *Connection Service* que utiliza o *ContextNet* como ferramenta para comunicação com a internet, logo com outros nós.



Esquema Mobile Hub [5]

A comunicação se dá da seguinte maneira: cada dispositivo móvel é um nó que se comunica com a rede através da camada SDDL do ContextNet. Um gateway para a camada SDDL é criado na etapa inicial, a mensagem é formatada para o padrão JSON e encapsulada como uma *Application Message*. A *Application Message* então é enviada para o SDDL Core que cuida dos tratamentos necessários da mensagem e a repassa para aplicações-cliente que utilizarão desta mensagem para armazenar ou fazer maior tratamento dos dados conforme a necessidade do usuário.

A criação do conceito do *Mobile Hub* cita possível uso para atuação, mas a implementação inicial do mesmo não se considerou a atuação, somente a coleta de dados de sensoriamento de dispositivos inteligentes e seu processamento através de *Complex Event Processing*. Mais tarde com a criação do serviço de atuação dentro do Mobile Hub, *Mact* - Mobile Actuation Service - [3], surgiu a necessidade da criação de clientes e serviços externos ao Mobile Hub. Um serviço essencial tanto para atuadores quanto para os sensores é o SOM [3] - *Smart Object Manager* -, o Mobile Hub precisa de drivers para comunicação correta com os dispositivos inteligentes, sendo que cada sensor/atuador necessita de um driver específico para se extrair as informações corretamente, ou interpretar a mensagem de atuação corretamente. Inicialmente o conceito do SOM foi criado somente para atuação, porém expandiu-se para incluir Sensores recentemente.

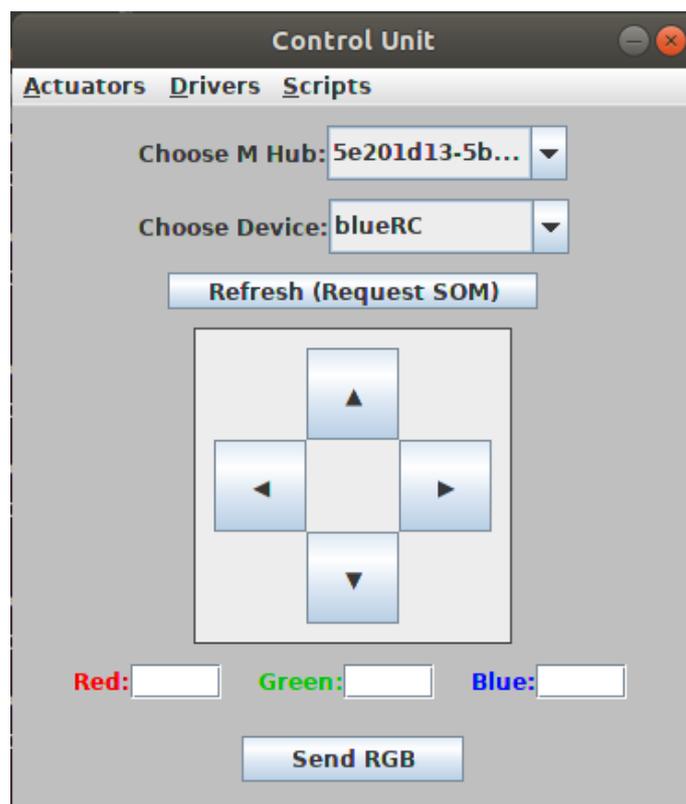
A necessidade de um cliente que consiga obter a lista de atuadores disponíveis - por *Mobile Hub* conectado à rede - e envie mensagens de atuação para cada dispositivo individualmente, ou para um grupo de dispositivos, também surgiu. Este cliente também é responsável por mandar as mensagens de atuação para os atuadores comandados pelos *Mobile Hubs* mais próximos e é neste programa que reside o foco deste trabalho.

### 3 Mact, SOM e o Mact User

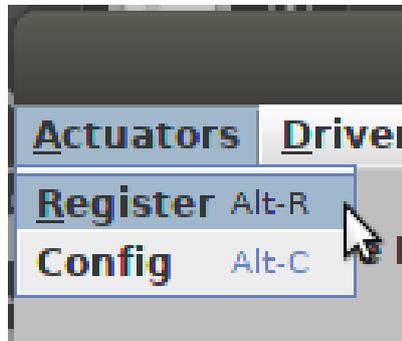
#### 3.1 Primeira Implementação

O *Mact* como já dito anteriormente é um serviço que foi criado dentro do Mobile Hub para que o dispositivo móvel consiga atuar sobre dispositivos inteligentes conectados via uma tecnologia WPAN. O processo de atuação ocorre da seguinte maneira: 1. o S2PA detecta o dispositivo inteligente dentro do alcance do Bluetooth (por exemplo), 2. a conexão é estabelecida através do próprio S2PA, 3. caso o dispositivo seja um atuador um evento é acionado e o objeto do atuador é enviado para o *Mact*, 4. o *Mact* faz um "request" através do Connection Service para o SOM, que retorna como resposta um driver 5. nesta etapa o Mobile Hub está apto a receber mensagens de atuação direcionadas para este dispositivo inteligente específico, podendo ele receber uma ordem de atuação por mensagem ou uma lista de ordens. Aproveitando a criação do *Mact User* e a necessidade de se registrar previamente os drivers no SOM, para serem requisitados posteriormente pelo Mobile Hub, o *Mact User* também recebeu a função de enviar drivers para o SOM, onde lá são registrados em um banco de dados.

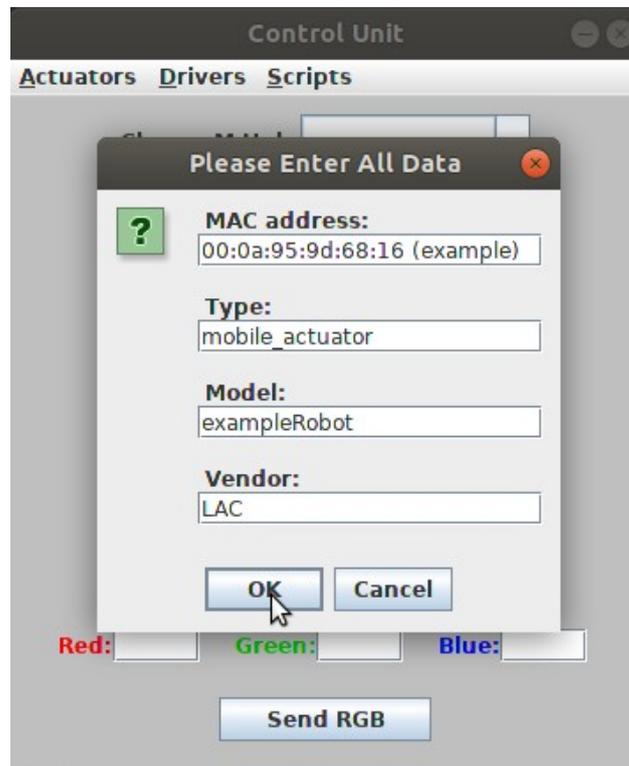
Com o objetivo inicial para o *Mact User* de facilitar que um usuário mais leigo possa enviar as mensagens de atuação e registrar os drivers no SOM, com uma interface de usuário simples de entender, projetei a interface descrita pelas imagens seguintes:



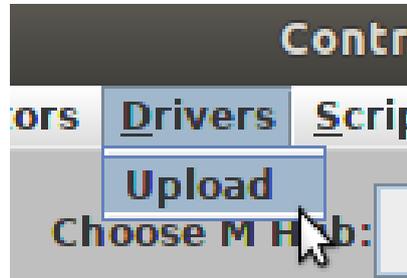
Mact User Imagem 1



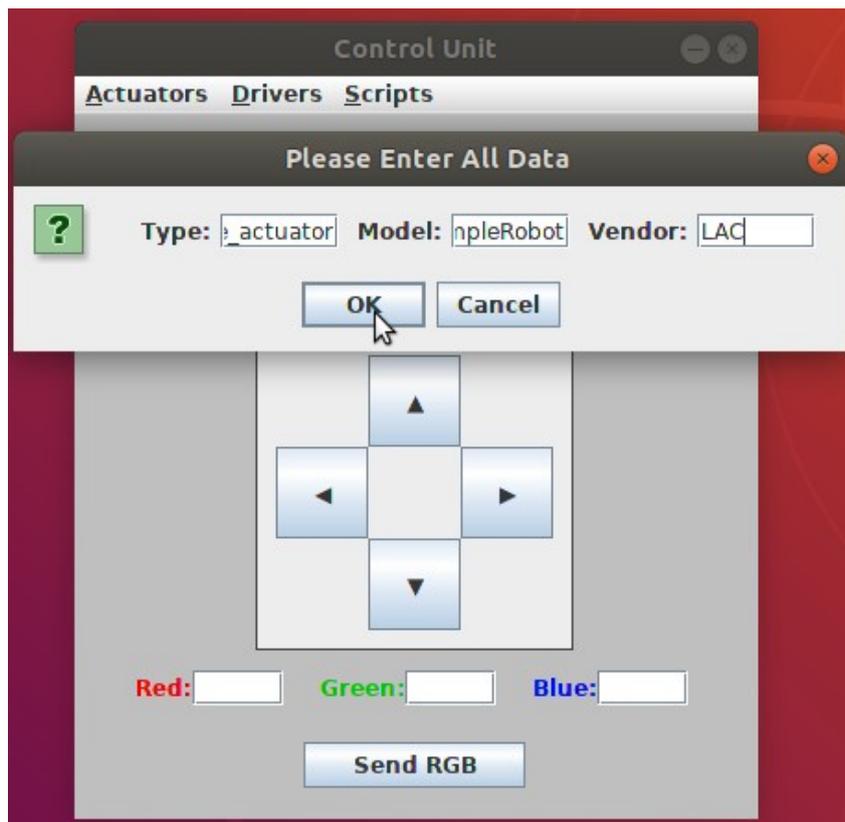
Mact User Imagem 2



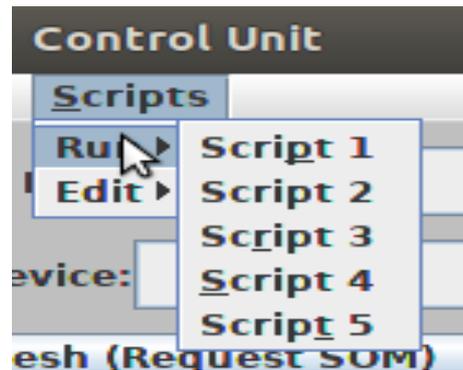
Mact User Imagem 3



Mact User Imagem 4



Mact User Imagem 5



Mact User imagem 6

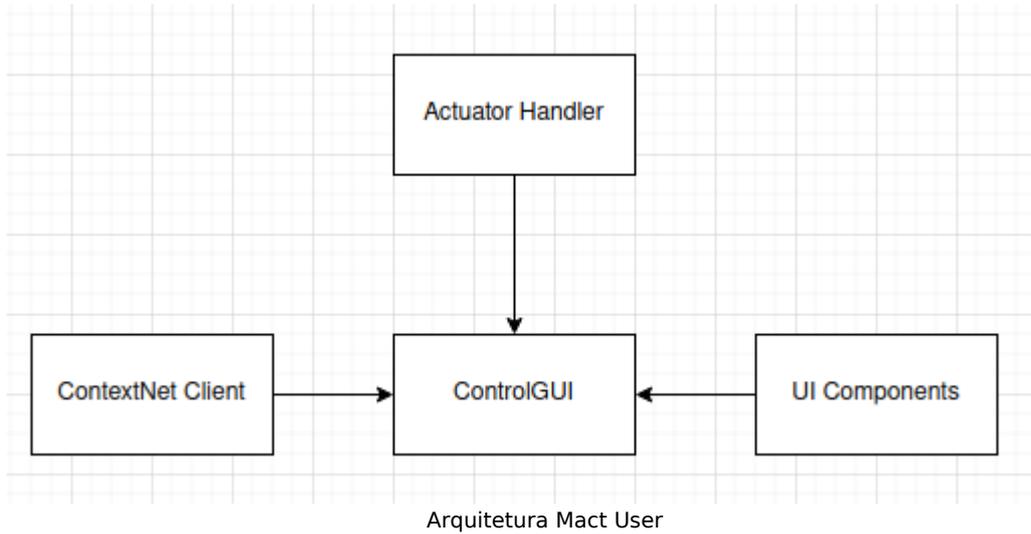
Esta versão do *Mact User* foi testada e utilizada em testes de controle de ar condicionado (projeto BlueRC) e no controle do Robot Car, um carrinho de controle remoto montado a partir de um Arduino entre outros. A arquitetura do código se baseou em utilizar o padrão Facade para separar completamente a interface da comunicação com a internet e a lógica. A classe *ControlPanel*, sendo essa Facade mencionada, detém tanto uma referência ao método de comunicação (Contextnet ou MQTT Paho) quanto uma referência à interface do usuário.

Vale ressaltar também que para futuros testes de atuação e automação incluí no programa uma linguagem única de scripts feita para facilitar o envio de mensagens sequenciais. Uma linguagem simples que é constituída de um comando com a mesma nomenclatura que é incluída no respectivo driver, e seus argumentos em sucessão. Assim, caso seja necessário enviar uma sequência de comandos, ou caso o atuador exija mais de um comando para atuação - caso que já aconteceu com o robô BB8, um brinquedo robô utilizado pelo laboratório para testes que exige um comando de permissão antes do comando de atuação propriamente dito. A seguir um exemplo do script de envio de múltiplos comandos através de uma única mensagem:

```
cmd toggleLED args R 100 G 200 B 255;  
cmd move args heading 100 speed 200;!
```

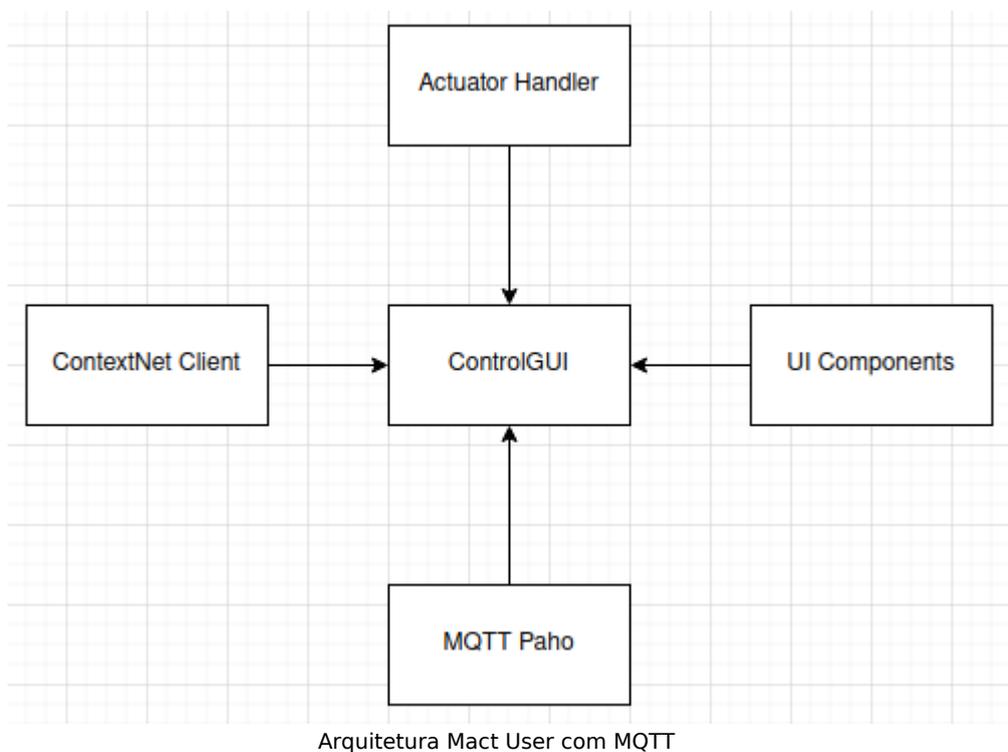
O script funciona da seguinte maneira, após "cmd" deve-se passar o nome do comando, e após args deve-se passar o nome do argumento seguido de seu valor. Todos estes dados constam no driver do dispositivo, dentro do Mobile Hub, e devem ser disponibilizados pelo criador do driver. O ponto e vírgula sinaliza o término de um comando, e a exclamação o término da sequência de comandos, podendo-se deixar comentários sem problemas após a exclamação.

Retornando a arquitetura do programa, segue o diagrama de sua arquitetura:



### 3.2 Inclusão do protocolo MQTT

Como dito anteriormente: a classe ControlPanel, dentro do pacote ControlGUI, se tornou a facade do projeto, dando independência aos demais elementos. Assim, após a implementação da arquitetura MQTT, a adição ficou da seguinte maneira:



Desta maneira é possível escolher utilizar a implementação MQTT ou utilizar o cliente do middleware do ContextNet caso necessário.

O protocolo MQTT é um protocolo de comunicação entre máquinas famoso por ser utilizado em IoT, e ele se baseia em um formato de publisher x subscriber, onde o subscriber basicamente se inscreve a um tópico, e caso alguém (publisher) publique neste tópico, todos inscritos este tópico receberão esta informação.

Para este protocolo funcionar corretamente é necessário um broker. Um broker é um servidor que gere as informações para os inscritos, servindo como um intermediário entre subscriber e publisher, sendo este broker organizado em tópicos [7]. Para este trabalho utilizei o Mosquitto [6], um broker popularmente utilizado em testes de pequena escala.

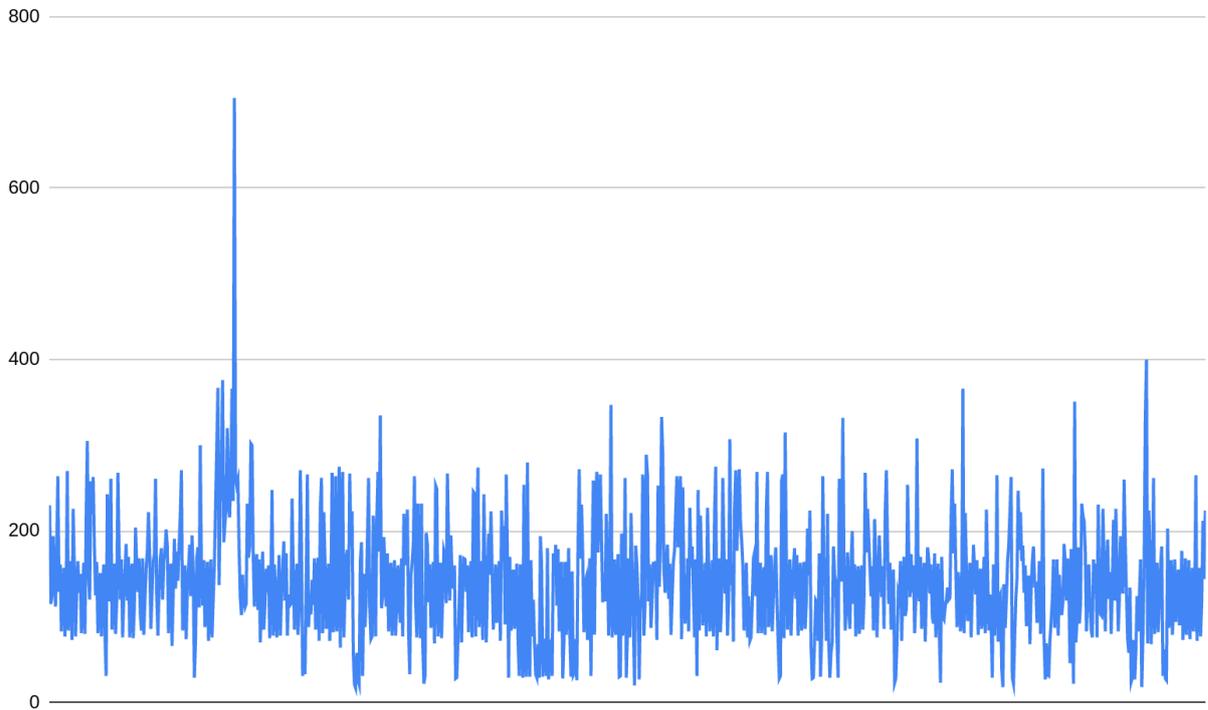
#### **4 Novo Mobile Hub e a implementação do teste**

O LAC recentemente passou a utilizar uma nova versão do Mobile Hub, ainda em fase de testes; ainda está para ser atrelado a ele, também, uma versão nova do ContextNet e por isso este teste carrega grande importância para comparações com futuros testes.

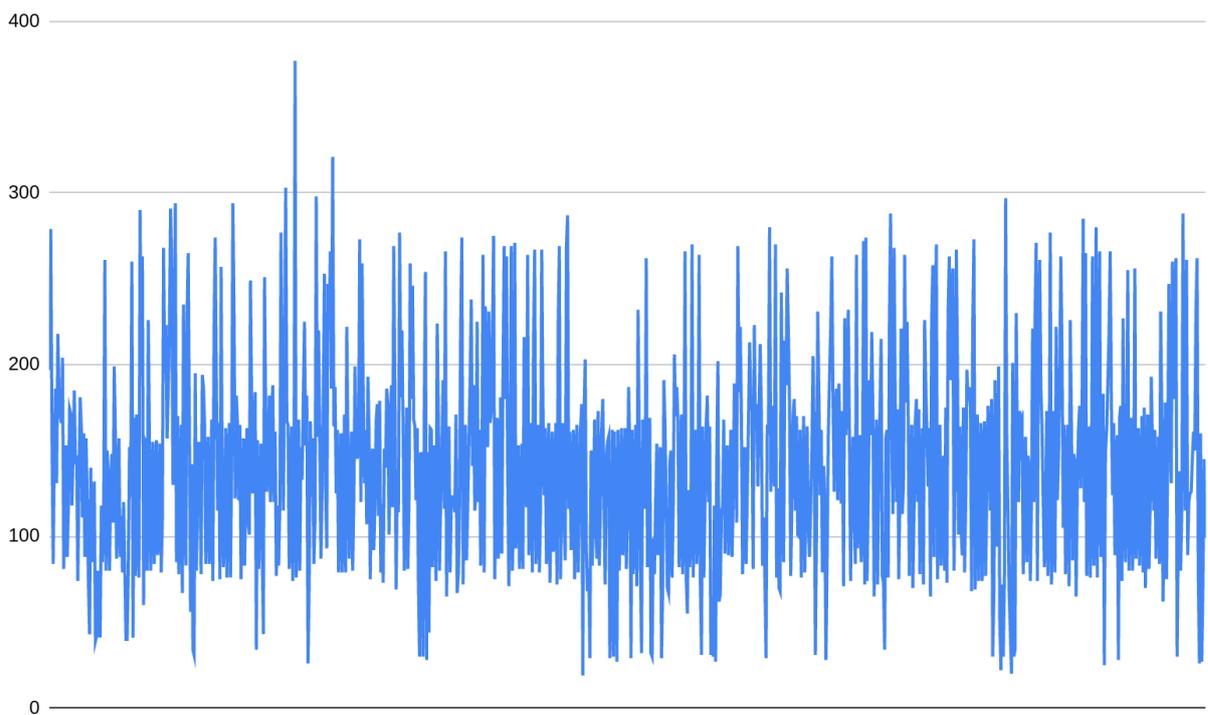
Na atuação o tempo de resposta a um comando é crucial, e para saber as dimensões deste tempo de resposta é necessário averiguar se o sistema, o ambiente, pelo qual esta mensagem de comando está sendo passada não está atrasando o resultado, assim descobrimos se a proposta do projeto de atuação neste ambiente é aceitável.

O novo mobile hub mantém o mesmo conceito do mobile hub anterior, porém agora utiliza tecnologias e arquitetura mais recentes. Utilizando a "clean architecture" o mobile hub possui um core - que possui as regras de comportamento do código - e em camadas mais externas possui interfaces para servirem de base para implementação das tecnologias (WPAN, WLAN, CEP etc) que vão ser adicionadas.

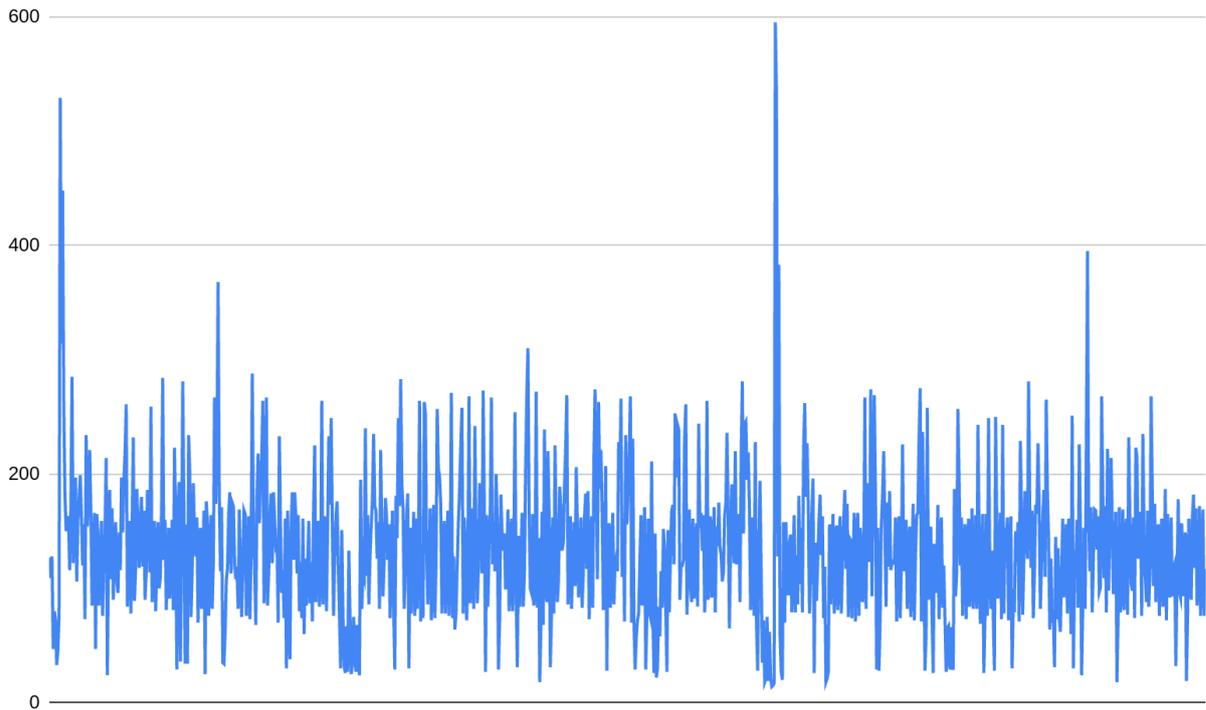
Colocando em pauta agora o teste propriamente dito, o mesmo consistiu de 1000 mensagens de teste de atuação enviadas do Mact User, de um notebook Linux Ubuntu 18.04 com processador I7-9750H e foram recebidas por um celular Moto g(6) com Android versão 9 utilizando Wi-Fi 5G. A latência, como já dito anteriormente, foi medida do momento do envio da mensagem de atuação pelo Mact User até o momento em que antecede o envio da mensagem de atuação do Mobile Hub para um dispositivo. Uma mensagem de resposta MQTT é enviada em retorno para o Mact User e o relógio então é parado, não se considerou o tempo de retorno da mensagem para parar o relógio. Seguem os resultados de três baterias de teste com suas médias e contagens considerando um corte de valores muito acima da curva, por serem prováveis perdas de pacote do meio (WiFi), considerei valores maiores que 750 ms como alguma falha externa ao experimento.



Resultado Teste 1



Resultado Teste 2



Resultado Teste 3

- Resultado Teste 1:

Contagem pós-corte: 983

Latência média: 141,9084435 ms

- Resultado Teste 2:

Contagem pós-corte: 985

Latência média: 142,1411168 ms

- Resultado Teste 3:

Contagem pós-corte: 982

Latência média: 135,7057026 ms

## 5 Conclusão

O teste foi bem sucedido e fornece uma base para testes futuros. Não encontrei, porém, testes similares sobre este tipo de atuação, logo não fui capaz de concluir sobre a natureza dos valores, e mais testes futuramente talvez elucidem-nos sobre a qualidade dos valores encontrados, sendo que quanto menor a latência melhor será o desempenho da atuação remota - diminuindo o tempo de resposta [4].

## Bibliografia

- [1] M. Endler; F. Silva e Silva, Past, Present and Future of the ContextNet IoMT Middleware,  
Open Journal of Internet of Things (OJIOT), vol.4, nr. 1, pages 7-23, ISSN = 23647108,  
July 2018.  
Presented at the Intl. Workshop on Very Large Internet of Things (VLIoT 2018)
- [2] L. E. Talavera; M. Endler; I. Vasconcelos; R. Vasconcelos; M. Cunha; Francisco J. da Silva e Silva, The Mobile Hub concept: Enabling applications for the Internet of Mobile Things,  
Publicado em: 23-27 March 2015, 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)
- [3] Sheriton R. Valim; Felipe Nogueira; Flávia Pisani; Markus Endler; Middleware Support for Generic and Flexible Actuation in the Internet of Mobile Things  
Publicado em: 2-16 June 2020 , 2020 IEEE 6th World Forum on Internet of Things (WF-IoT)
- [4] André Barros de Sales; MEDIDAS DE LATÊNCIA EM AMBIENTES DE PROCESSAMENTO DE ALTO DESEMPENHO; Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.  
Orientador: Prof. Dr. Carlos Becker Westphall
- [5] <http://www.lac.inf.puc-rio.br/index.php/software-contextnet/>
- [6] <https://mosquitto.org/>
- [7] <https://www.hitecnologia.com.br/blog/o-que-e-protocolo-mqtt/>
- [8] <https://www.eclipse.org/paho/clients/java/>