

# MAC0352-EP1

Luca Assumpção Dillenburg  
NUSP: 11796580

# Descrição da Apresentação em Alto Nível

Podemos entender o broker como um servidor que aceita todas as conexões, criando um novo processo filho para lidar com cada uma delas.

Esses processos se comunicam através de FIFOs: arquivos especiais utilizados para o envio de mensagens entre processos. Os processos que lidam com clientes publishers enviam as mensagens para os arquivos FIFOs de um certo tópico e os processos que lidam com clientes subscribers leem esses arquivos e depois enviam as mensagens aos clientes.

# Decisões Relevantes de Projeto: Sistema de Arquivos

Uma decisão importante na organização do sistema foi onde seriam armazenados os arquivos FIFO de cada tópico. Ao meu ver, haveriam duas formas de fazer: (1.) armazená-las num hashtable que parearia os tópicos (keys) e os arquivos FIFO inscritos em cada tópico ou (2.) criar uma estrutura de arquivos que, por si só, informasse os arquivos FIFO inscritos em cada tópico.

Optei pela segunda opção por remover a necessidade de lidar com o compartilhamento de memória entre diferentes processos. Por isso, tornou-se necessário criar cada arquivo FIFO dentro de uma pasta dedicada ao tópico em que o mesmo está inscrito, possibilitando o envio de mensagens pelos processos do broker responsáveis por encaminhar as mensagens dos publishers.

# Decisões Relevantes de Projeto: Sistema de Arquivos

Por conta dessa organização de pastas, tornou-se necessário fazer um encoding dos nomes dos tópicos para garantir que é um nome válido para uma pasta no Linux.

Segue um exemplo (ao lado) da estrutura de arquivos quando há com alguns tópicos e subscribers.

```
[/tmp/MAC0352_mqtt_broker]$ tree
```

```
.
├── topico_1
│   ├── M18h8sh
│   ├── dw8rsd
│   └── ks7h2
├── topico_2
│   └── jrhs7k
└── topico_3
    ├── bvy13wu
    ├── faoa234
    ├── hui74
    └── pi31fr
```

```
3 directories, 8 files
```

# Descrições dos Comandos

- CONNECT: é necessário enviar um packet de resposta CONNACK para iniciar o recebimento de outras mensagens de controle
- SUBSCRIBE: é necessário enviar um packet de resposta SUBACK confirmando a inscrição no tópico, e, depois localmente, criar um FIFO do tópico escolhido para ler as mensagens enviadas e repassá-las para o cliente.  
ps: Por conta das chamadas de controle que poderão ocorrer enquanto o processo está bloqueado esperando a leitura do FIFO, foi necessário criar um processo filho para ler a FIFO, enquanto o processo pai continua a tratar as chamadas de controle.
- PINGREQ: é necessário enviar um packet de resposta PINGRES
- PUBLISH: não é necessário enviar um packet de resposta, porém é necessário enviar a mensagem para os arquivos FIFOs referentes ao tópico do packet
- DISCONNECT: não é necessário fazer nenhuma alteração, pois após o disconnect, o cliente irá encerrar a comunicação por socket, saindo do while que espera por mensagens de controle e, assim, finalizando o fluxo normalmente.

# Análise de Desempenho

Vimos que o MQTT é muito utilizado em IoT, onde os dispositivos que estão enviando e recebendo mensagens muitas vezes tem uma baixa capacidade de processamento. Por essa razão, procurei utilizar uma máquina antiga para rodar os clientes. O servidor, em contrapartida, geralmente é mais robusta, então utilizei uma máquina mais nova para rodar o servidor.

A máquina que está rodando o broker/servidor é o Macbook Air M1 com 16gb de RAM, enquanto os clientes estão rodando em um laptop da com um Core 2 Duo e 4gb de RAM.

# Análise de Desempenho

Os dados a serem disponibilizados a seguir são da máquina que rodava o broker (Macbook Air), os dados foram capturados pelo Wireshark (com o filtro da porta e MQTT) e pelo aplicativo Activity Monitor da Apple.

Foram feitas diversas medições com diferentes mensagens e tópicos para ter validade estatística.

Informações de Rede: rede cabeada no PC e rede Wi-Fi no Macbook, rede com 10mb de upload, 20mb de download e 127ms de latência.

# Análise de Desempenho: apenas broker

## Bytes Transferidos:

0 bytes

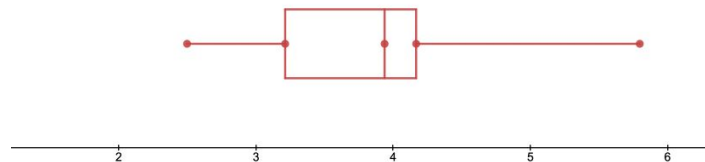
*Nenhuma informação foi transferida.*

## Uso de CPU:

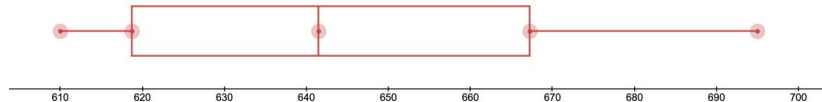
Média: 3.9% com 645.83 processos

Desvio Padrão: 1.138% com 33.58 processos

## Gráfico % de CPU:



## Gráfico Número de Processos:



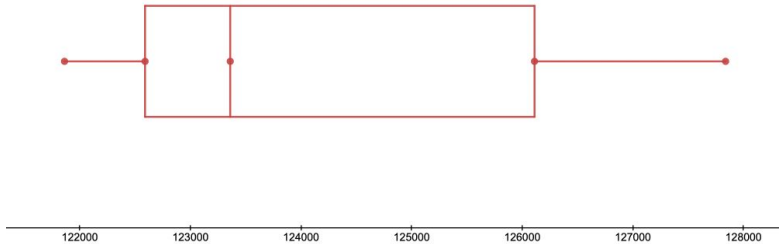


# Análise de Desempenho: broker + 100 clientes (5 tópicos)

## Bytes Transferidos:

Média: 124299 bytes

Desvio Padrão: 2471.2 bytes

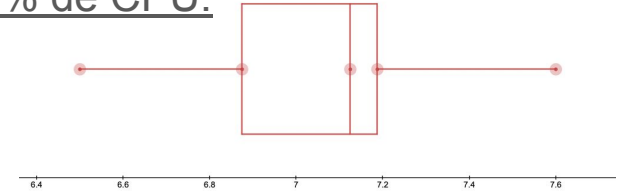


## Uso de CPU:

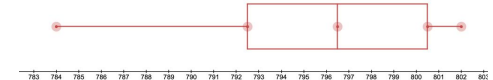
Média: 7.058% com 795.3 processos

Desvio Padrão: 0.37% com 6.80 processos

## Gráfico % de CPU:



## Gráfico Número de Processos:

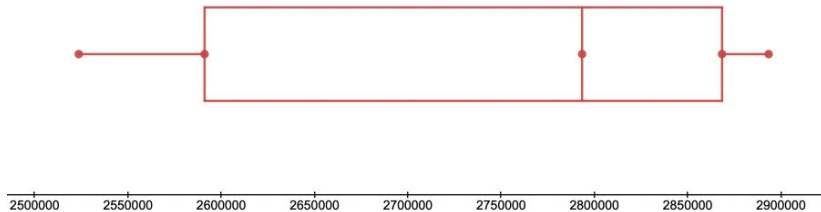


# Análise de Desempenho: broker + 1000 clientes (20 tópicos)

**Uso de Rede** (bytes transferidos):

Média: 2737019 bytes

Desvio Padrão: 171084 bytes



**Uso de CPU:**

Média: 12.1% com 1668.8 processos

Desvio Padrão: 1.89% com 10.34 processos

Gráfico % de CPU:

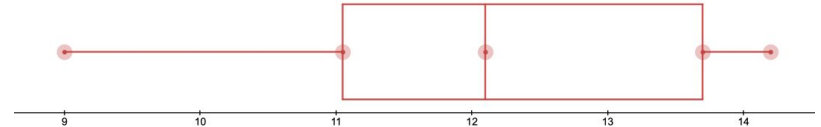
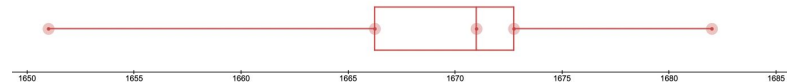


Gráfico Número de Processos:



# **Análise de Desempenho: conclusão**

Apesar de ser possível observar um aumento significativa no uso da CPU, o uso não cresceu de forma linear e continuou ocupando um porcentagem pequena do processador.

O uso da rede por outro lado, aumentou de forma linear, porém ela não me pareceu sobrecarregada, nem impactou significativamente o uso de outros dispositivos conectados a ela.