

# Desempenho e Escalabilidade de Plataformas Livres de IoT

Dener Ottolini, Ivan D. Zyrianoff, Alexandre Heideker, João H. Kleinschmidt,  
Carlos A. Kamienski

<sup>1</sup>Universidade Federal do ABC

{dener.silva, ivan.dimitry, alexandre.heideker, joao.kleinschmidt, cak}  
@ufabc.edu.br

**Abstract.** *IoT platforms manage distributed sensors, actuators and data, connecting heterogeneous devices with applications that add value to the collected data. Although there are several IoT platforms, many are complex, unversatile and expensive, still lacking information about their performance and scalability, impairing a direct comparison between them. This article evaluates and compares three open source IoT platforms - FIWARE, Konker and ThingsBoard - through quantitative metrics to understand their operation, scalability, reliability and hardware resources needed for its operation in city and health scenarios. The results show that greater computational power directly affects the scalability of each platform, but does not directly interfere with the processing time of each message. The results also show that, although the platforms evaluated have similar structures and proposals, their performances are very different, showing that each platform is suitable for a different solution.*

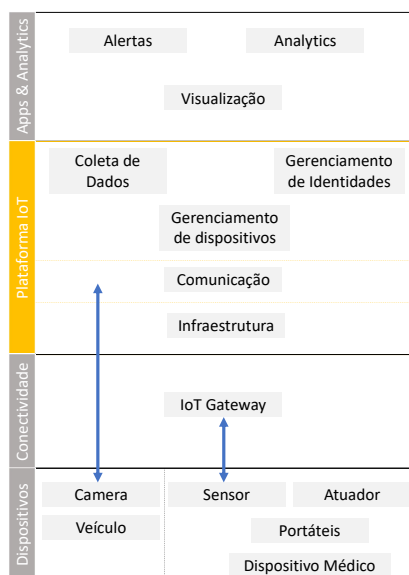
**Resumo.** *Plataformas de IoT gerenciam sensores, atuadores e dados distribuídos, conectando dispositivos heterogêneos com aplicações que agregam valor aos dados coletados. Embora existam diversas plataformas de IoT, muitas são complexas, pouco versáteis e caras, carecendo ainda de informações sobre o seu desempenho e escalabilidade, prejudicando uma comparação direta entre elas. Este artigo avalia e compara três plataformas de IoT de código aberto - FIWARE, Konker e ThingsBoard - através de métricas quantitativas com o objetivo de compreender seu funcionamento, escalabilidade, confiabilidade e recursos de hardware necessários para sua operação em cenários de cidades e saúde. Os resultados mostram que maior poder computacional afeta diretamente a escalabilidade de cada plataforma, mas não interfere diretamente no tempo de processamento de cada mensagem. Os resultados mostram ainda que, embora as plataformas avaliadas tenham estruturas e propostas semelhantes, seus desempenhos são muito distintos, mostrando que cada plataforma é adequada para uma solução diferente.*

## 1. Introdução

Novas tecnologias baseadas no conceito de Internet das Coisas (IoT) surgem a cada dia, seja em número de dispositivos conectados, soluções de software ou aplicações que se beneficiam com as informações coletadas por esses dispositivos. À medida que a IoT se dissemina, os dispositivos também evoluem, tornando-se cada vez mais comum a criação de dispositivos conectados à Internet por padrão, permitindo uma comunicação e integração mais complexas entre dispositivos, usuários e aplicações. Em 2008, o número de dispositivos conectados à Internet superou o número de pessoas na Terra [Liu 2019]. Estima-se

que na década de 2020 o mundo terá dezenas de bilhões de dispositivos conectados. Os números variam de 19 bilhões de dispositivos relacionados à soluções de IoT em 2022, a 42 bilhões em 2022 e 79 bilhões em 2025 [Statista 2019]. Entretanto, uma das desvantagens desta rápida evolução é a proliferação de redes de sensores heterogêneos que carecem de interoperabilidade [Luca Calderoni 2019].

Para acompanhar este desenvolvimento, as plataformas de IoT surgem como soluções multicamadas que possibilitam funções de provisionamento, gerenciamento e automação de dispositivos, tendo como característica sua adaptabilidade a diferentes cenários e tecnologias. As plataformas de IoT agem como *middleware*, cuja função é mediar a transmissão de dados entre as camadas de aplicação e hardware. A Figura 1 ilustra a estrutura multicamadas comum à grande parte das Plataformas de IoT atuando como *middleware*. As plataformas devem ser capazes de operar com diferentes protocolos de transmissão para coleta de dados dos sensores, como ilustrado na Figura 1. A plataforma pode receber dados em um protocolo que poderá tratar diretamente, como os objetos deste estudo, FIWARE, Thingsboard e Konker que são capazes de receber os dados através de uma API Rest. Um gateway IoT pode ser necessário para converter os formatos de dados de diferentes protocolos para os formatos que podem ser tratados pela plataforma, como acontece na plataforma FIWARE que dispõe de um *General Enabler* - microserviço fornecido pela solução FIWARE - denominado IoT Agent que recebe uma mensagem em diferentes protocolos e converte para o protocolo NGSI utilizado pelo FIWARE. O uso de *gateways* permitem às plataformas a integração com diferentes tipos de dispositivos, possibilitando ainda o fornecimento destes dados a aplicações que poderão consumi-los para execução de tarefas de análise, Big Data ou apenas visualização [Technologies 2019].



**Figura 1. Plataformas de IoT multi-camadas**

De acordo com um estudo elaborado por [Liu 2017], em 2017 o mercado dispunha de cerca de 450 plataformas de IoT - este número tende a ser muito maior nos dias atuais - sendo os líderes destes mercados as grandes empresas de TI como Microsoft, Google

e Amazon. Entretanto, mesmo com grandes empresas de TI envolvidas no desenvolvimento de novas tecnologias e plataformas de IoT, ainda não há uma solução que domine o mercado. Embora as Plataformas de IoT exerçam papel crucial no desenvolvimento de novas tecnologias e aplicações inteligentes, as atenções e esforços são normalmente direcionados à compatibilidade ou conectividade destas plataformas com novos dispositivos e tecnologias de transmissão. A necessidade de adaptação a múltiplos protocolos e padrões gera ainda problemas de interoperabilidade entre as plataformas, uma vez que cada plataforma pode operar com seus próprios protocolos e padrões, exigindo adaptação por parte dos desenvolvedores [Broring 2017].

Este artigo avalia de maneira quantitativa três das mais populares plataformas de IoT de código livre, FIWARE<sup>1</sup>, ThingsBoards<sup>2</sup> e Konker<sup>3</sup>. As plataformas foram selecionadas com base na experiência prévia dos autores e pela semelhança em suas estruturas, protocolos em comum com os quais operam e capacidade de operar através de contêineres Docker, o que permite uma comparação justa. O objetivo é analisar, identificar e medir os componentes de cada plataforma e seu comportamento em diferentes cenários que são comuns a aplicações de IoT, como o envio de dados de sensores para processamento e armazenamento. Utilizando tráfego sintético, as plataformas foram submetidas a cargas de trabalho leves, moderadas e altas, sendo observado a todo momento o consumo de recursos computacionais como CPU e memória RAM. Foi realizada a avaliação do tempo de processamento de cada mensagem para permitir a análise do desempenho de cada solução proposta. Os resultados mostraram que embora as plataformas sejam similares em suas propostas e arquiteturas, seu desempenho é diferente diante dos diferentes cenários avaliados. Desta forma, foi possível avaliar que as plataformas de código aberto estudadas podem ser utilizadas em diferentes soluções de IoT que exigem diferentes tipos de comportamento, seja a baixa necessidade de recursos apresentada pela Thingsboard, a facilidade de implementação do FIWARE ou ainda a versatilidade da Konker.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 apresenta as plataformas analisadas, a metodologia, ferramentas e métricas que foram utilizadas para a elaboração dos experimentos; a Seção 4 apresenta os resultados que são discutidos na Seção 5; por fim são apresentadas as conclusões do trabalho.

## 2. Trabalhos Relacionados

Devido ao grande número de plataformas de IoT disponíveis no mercado, que ocasiona um alto grau de heterogeneidade - seja por diferença de protocolos, estruturas necessárias ou tecnologias necessárias para implementação - estudos que analisam as características destas plataformas são geralmente qualitativos como em [Hamdan Hejazi 2018], [Daniel Happ 2018] e [Singh 2017], devido ao alto número de opções e grau de heterogeneidade, sendo inviável uma comparação direta entre um grande número de aplicações. No caso de plataformas comerciais como AWS [Ltda. a], Google [Ltda. 2019a], IBM [Ltda. 2019b] e Microsoft Azure [Ltda. b], se tornam inviáveis uma vez que não se tem acesso às tecnologias, plataformas e hardware, tornando uma possível avaliação injusta, sendo que não é possível garantir paridade entre as diferentes infraestruturas de cada fornecedor.

---

<sup>1</sup><https://www.fiware.org/>

<sup>2</sup><https://thingsboard.io/>

<sup>3</sup><http://www.konkerlabs.com/>

Estudos experimentais são apresentados em [Zyrianoff 2018] e [Ramón Martínez and Iborra 2016] que avaliam a plataforma FIWARE voltada para cenários de *Smart Agriculture* sem a variação de infraestrutura. Os resultados de [Zyrianoff 2018] mostraram o funcionamento da plataforma FIWARE em operação em diferentes cenários com névoa e nuvem computacional, além de avaliar o impacto causado pelo uso da tecnologia LoRaWAN em soluções de IoT. A plataforma ThingsBoard é avaliada em [Ahmed A. Ismail 2018] para um cenário de *Smart City* com um baixo número de sensores, sendo o primeiro cenário com apenas 10 sensores, e novamente, sem variação de infraestrutura. Em [Mauro A. A. 2018] é reforçada a falta de análises quantitativas na literatura, sendo que análises qualitativas não são o suficiente para auxiliar na escolha de uma solução em IoT. Ainda em [Mauro A. A. 2018] é apresentada uma abordagem através do método de decisão multicritério PROMETHEE que analisa os critérios qualitativos encontrados na literatura com critérios quantitativos e atribui pontuações para cada característica de cada plataforma, sendo o FIWARE uma das plataformas analisadas.

O diferencial da proposta deste trabalho é submeter plataformas de IoT a diferentes cargas de trabalho e *payloads*, possibilitando a compreensão do seu comportamento para diferentes soluções de IoT, além da variação nas infraestruturas que permitiram a avaliação dos requisitos de hardware. As avaliações utilizaram tráfego real e foram analisadas de maneira quantitativa, diferente de muitas avaliações encontradas na bibliografia que se baseiam nas experiências do usuário.

### **3. Metodologia**

#### **3.1. Plataformas de IoT**

##### **3.1.1. Plataforma FIWARE**

Promovida e financiada pela Comissão Europeia, a Plataforma FIWARE consiste em diferentes *Generic Enablers* (GE) que atuam como microsserviços que podem ser então combinados para a criação de uma aplicação. Através de seu catálogo, a comunidade FIWARE fornece e compartilha diferentes GEs e aplicações que possibilitam a criação das mais diversas soluções. A comunicação entre os GEs é realizada através de uma API REST com o padrão NGSI (*OMA NGSI standard* [Mobile Alliance 2019]). Entre os diversos componentes disponíveis, foi avaliado neste trabalho o GE mais popular do FIWARE, o Orion Context Broker. Operando através de *publish/subscribe*, o Orion é responsável pela criação, registros, atualizações e subscrições na plataforma FIWARE, sendo ainda possível a criação de notificações ao se atender uma determinada condição, como a alteração de algum valor de atributo. Entretanto, o GE Orion só é capaz de armazenar o último estado de cada atributo. A fundação FIWARE foi criada em 2016 pela Atos, Engigeering, Orange e Telefonica, se tornando uma organização sem fins lucrativos que encoraja a adoção de um *framework* para compartilhamentos de dados comum, colaborativa e interoperável baseada em padrões de tecnologias de código livre. A comunidade FIWARE ainda provê cursos, seminários e compartilhamento de dados obtidos com o uso da plataforma para proposta de melhoria contínua das tecnologias de IoT, que podem ser ainda implementadas de maneira gratuita por uma nuvem computacional de testes disponibilizada por parceiros da fundação. Entre as soluções que implementam a plataforma FIWARE podemos citar o projeto SWAMP [SWAMP 2017] que visa utilizar a solução FIWARE para coleta de dados para uso em *Smart Agriculture* e o projeto Bettair [Cities

S.L. 2018] que mapeia a qualidade do ar em alta resolução com sensores de gases de alta precisão.

### 3.1.2. Plataforma ThingsBoard

ThingsBoard é uma Plataforma IoT de código livre que permite a coleta, processamento, visualização de dados e gerenciamento de dispositivos. Compatível com a maioria dos protocolos de transmissão do mercado (MQTT, CoAP) a plataforma ThingsBoard se popularizou devido à sua flexibilidade e facilidade de uso, como o fornecimento de *dashboards* para visualizações de dados. A plataforma ThingsBoard permite diferentes formas de configuração, possibilitando a utilização de diferentes bancos de dados para metadados e para dados dos dispositivos conectados através de uma solução híbrida que conecta um banco de dados relacional Postgres com um banco de dados NoSQL (Cassandra), sendo o Postgres responsável por armazenar dados históricos enquanto o NoSQL armazena dados de séries temporais [Inc. 2019b]. Para esta avaliação foi utilizada uma versão da plataforma que opera com um banco de dados relacional Postgres tanto para armazenamento de metadados de cada sensor - id, nome, descrição - quanto para o armazenamento de valores. Esta escolha foi necessária pois a configuração mínima proposta pela página do desenvolvedor que utiliza outros bancos de dados não relacionais é superior à capacidade computacional dos cenários propostos neste artigo. Dentre os diversos usos da plataforma, os desenvolvedores oferecem soluções prontas de controle inteligente de insumos de fazendas, rastreamento de frotas, e gerenciamento de sensores de *Smart Cities*.

### 3.1.3. Plataforma Konker

A plataforma Konker é desenvolvida e mantida pela Konker Labs, empresa brasileira sediada em São Paulo. A plataforma oferece uma versão gratuita capaz de operar com protocolos HTTP e MQTT. A principal característica que a difere das demais é a função nativa de segurança através da criação de senhas para cada dispositivo, diferente da solução comum na qual a segurança é feita através da autenticação pela API utilizando senhas. A Konker opera de maneira centralizada, com todas as funções operando em apenas uma instância, onde fornece ferramentas de visualização e encaminhamento de dados entre sensores através da criação de rotas, facilitando a implementação da comunicação M2M (Machine-to-Machine), permitindo a comunicação entre dois dispositivos sem a necessidade de interação humana. Dentre as soluções que adotam a plataforma Konker, estão a solução de rastreamento de frota da Toyota IoT Tracking e soluções de gerenciamento inteligente de ar condicionado que reduziram o consumo de energia em 15% para a Inmetrics e solução de organização e disponibilidades de salas para a TOTVS [kon 2020].

## 3.2. Ferramentas Computacionais

Para a construção dos cenários e coleta de métricas, foram utilizadas as seguintes ferramentas: SenSE, IMAIoT e Docker.

O SenSE - *Sensor Simulation Environment* [Zyrianoff 2017] é um gerador de tráfego sintético de sensores em larga escala de código aberto, com o objetivo de simular o tráfego de dados em ambientes IoT em cenários complexos. Desenvolvido em linguagem

Java, a ferramenta geradora de tráfego SenSE é adequada para comprovar a eficiência e escalabilidade de sistemas *middleware* em tempo real, por possuir a capacidade de gerar cargas de trabalho de milhares de sensores propagando dados simultaneamente. A utilização do SenSE permite a abstração dos sensores sem prejuízo à hipótese proposta neste trabalho de avaliar o comportamento de cada plataforma em diferentes cenários e com diferentes cargas de trabalho uma vez que, embora os sensores sejam virtuais, o tráfego gerado é real. A utilização de sensores sintéticos não afetará o estudo proposto, sendo este focado na análise do comportamento das plataformas IoT, seus componentes e consumo de recursos computacionais frente a diferentes cargas de trabalho.

Para coletar as métricas de sistema, utilizou-se o IoTMA [Heideker A. 2019], uma ferramenta de monitoramento que utiliza protocolos de IoT para publicar suas métricas. A versatilidade da ferramenta permite o monitoramento de uso de CPU e memória do sistema assim como de contêineres Docker onde as plataformas serão executadas. A ferramenta foi configurada para armazenar os dados em formato JSON com intervalo de 5 segundos entre as amostras.

Finalmente, todas as plataformas foram executadas em contêineres Docker. O Docker é um software que fornece uma camada de abstração e automação para virtualização de sistemas operacionais - também conhecida como *lightweight virtualization*. Com o Docker, desenvolvedores conseguem empacotar soluções de software de maneira padronizada, permitindo a criação de ambientes isolados que já contam com as bibliotecas e links necessários, simplificando a implantação destas soluções [Inc. 2019a].

### 3.3. Métricas

Para análise das plataformas foram consideradas duas métricas de desempenho: o tempo de processamento e o uso de recursos de sistema.

A métrica de tempo de processamento refere-se ao tempo decorrido entre a obtenção do dado pelo sensor até o mesmo atingir o componente de armazenamento para que o dado esteja disponível para uma aplicação consumidora. Desta forma, pretende-se avaliar o tempo entre a criação do dado até o momento em que este esteja disponível para consumo de uma aplicação externa, como por exemplo uma *dashboard* que exhibe os dados ou um módulo de análise dos dados.

Para medir o uso de recursos de sistema, foram coletadas métricas de sistema operacional do uso de CPU e memória RAM, considerando o consumo total de todos os seus respectivos módulos de software. Em cenários como a plataforma FIWARE, o banco de dados e o Context Broker Orion são executados em contêineres Docker distintos. Para análise comparativa, o consumo de cada componente foi somado para que se tenha um valor de consumo total para a plataforma. Cada amostra foi coletada através do IMAIoT em um intervalo de 5 segundos, capturando dados de uso de CPU e memória de cada contêiner.

### 3.4. Cenários

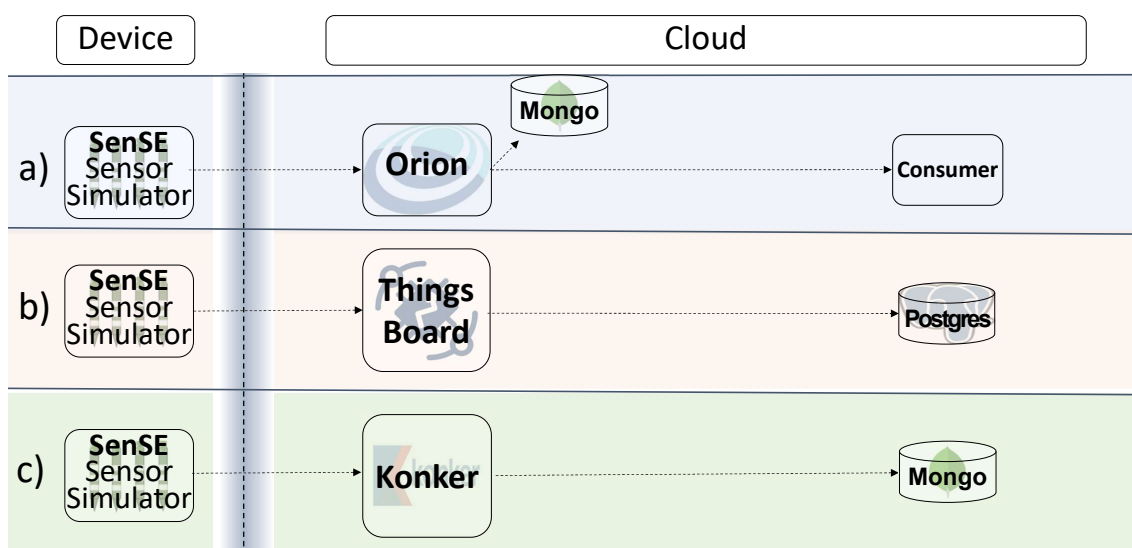


Figura 2. Cenários e Ambiente Experimental

A Figura 2 apresenta os cenários utilizados para este trabalho. O simulador de sensores SenSE foi configurado para enviar o *timestamp* do momento da criação do dado para cada plataforma através do protocolo HTTP, simulando um sensor que envia apenas um dado simples com um único campo. No cenário a, a Plataforma FIWARE recebe os dados através do padrão NGSI. Todos os equipamentos foram sincronizados com o protocolo NTP (*Network Time Protocol*) para evitar problemas de diferenças nos *timestamps*.

No cenário a, o Context Broker Orion receberá as informações enviadas pelo SenSE e encaminhará a informação através de uma notificação para uma aplicação consumidora, desenvolvida em NodeJS especificamente para este experimento, que será responsável por armazenar o *timestamp* do momento em que o dado foi recebido e poderá ser consumido.

No cenário b está ilustrada a configuração da Plataforma ThingsBoard. A aplicação armazena no banco de dados o *timestamp* do momento em que o dado foi gravado e está disponível para consumo.

No cenário c está ilustrada a configuração da Plataforma Konker, que opera com um banco de dados não relacional MongoDB tanto para armazenamento de metadados quanto para o armazenamento de dados históricos. Assim como na plataforma ThingsBoard, o *timestamp* do momento do armazenamento também é adicionado ao registro do dado.

Portanto, para cada cenário, o tempo de processamento será calculado subtraindo-se o valor do *timestamp* do momento da criação da mensagem por parte do SenSE do *timestamp* do momento em que o dado foi armazenado no banco de dados de série histórica.

Para a execução dos experimentos não foram utilizados emuladores de rede, uma vez que o objetivo deste trabalho é identificar o comportamento das plataformas de IoT experimentalmente.

### 3.5. Fatores e Níveis

Para avaliar o desempenho e escalabilidade de cada plataforma, os experimentos foram realizados através da adaptação de cenários reais. No primeiro cenário avaliado o simulador SenSE foi configurado para enviar um *payload* de 25 bytes, informando apenas o *timestamp* do momento da criação do dado. Desta forma o cenário simula uma solução genérica de IoT voltada para *Smart City*, com um sensor transmitindo um único dado, como por exemplo uma medida de temperatura ou umidade.

No segundo cenário o SenSE foi configurado para enviar dados comuns a aplicações de *Smart Health* que podem ser encontrados na literatura [Galli 2019] [K. Natarajan 2016] [Pascoli and Iannaccone 2016] [Zhe Yang 2016]. O *payload* passa a ser estruturado, fornecendo os seguintes dados: identificação do paciente, temperatura, nível de glicose, batimento cardíaco, passos, geolocalização, medicamentos, eletrocardiograma, pressão sanguínea, umidade, altura e peso. Com o novo formato, o *payload* deste cenário totaliza 3.685 bytes, que permitirão avaliar o desempenho de cada plataforma com cenários que necessitam de maiores detalhes de informações ao invés de informações pontuais.

Os experimentos realizados visam ainda analisar como cada plataforma gerencia seus recursos computacionais e, para tal, cada cenário foi executado em diferentes infraestruturas com variação na carga de trabalho. Em um primeiro cenário, cada plataforma foi executada em um contêiner Docker em uma máquina virtual localizada no serviço Amazon AWS do tipo *t2.Medium*, contando com 2 vCPUS, 4Gb de memória RAM e 20Gb de disco rígido. Um segundo cenário foi introduzido replicando as mesmas plataformas em uma instância virtual Amazon AWS com o tipo *c5d.xlarge*, que conta com o dobro da capacidade, ou seja, 4 vCPUS, 8Gb de memória RAM e 40Gb de disco rígido. Esta variação, nos permitirá verificar como cada plataforma gerencia o consumo dos recursos computacionais em situações de escassez ou sobra de recursos.

Para análise da escalabilidade, cada cenário foi avaliado com diferentes cargas de trabalho de acordo com a infraestrutura no qual a plataforma foi executada. Em instâncias AWS.t2.Medium foram utilizadas as cargas de trabalho de 8, 16 e 24 mensagens por segundo. Já nas instâncias AWS.c5d.Large foram utilizadas as cargas de trabalho de 24, 32 e 40 mensagens por segundo. Desta forma, será possível a comparação direta do uso de recursos computacionais em diferentes condições. Todos os 36 cenários propostos são listados na Tabela 1.

Mensagens por segundo	Recursos de Hardware	Payload
8, 16 e 24	Medium	Smart City
24, 32 e 40	Large	Smart City
8, 16 e 24	Medium	Smart Health
24, 32 e 40	Large	Smart Health

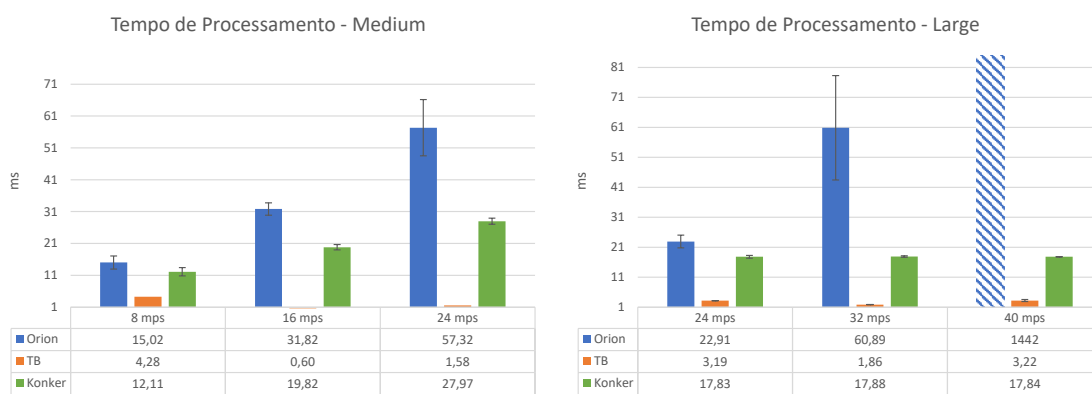
**Tabela 1. Fatores e Níveis**

Cada experimento teve a duração de 2 minutos com 30 replicações, totalizando 36 horas de execução.

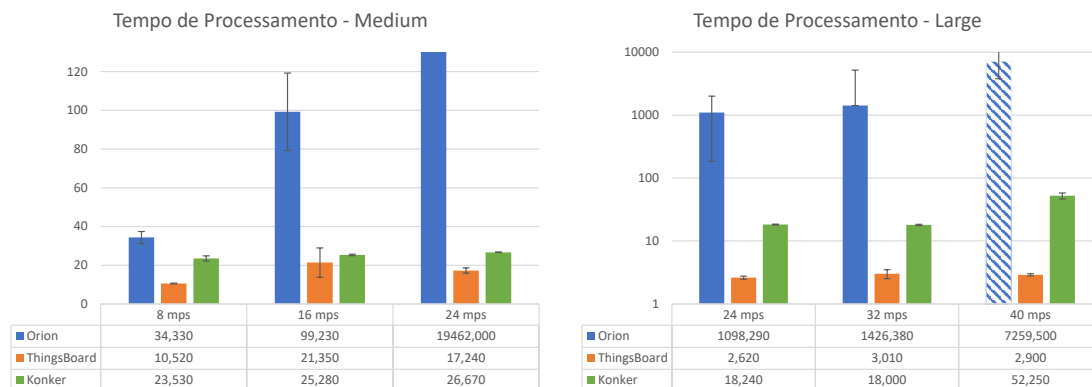


## 4. Resultados

A Figura 3 apresenta o tempo de processamento em diferentes infraestruturas para o cenário de *Smart City*, enquanto a Figura 4 apresenta o mesmo dado em relação ao cenário de *Smart Health*. Para todas as avaliações foi utilizado o intervalo de confiança de 95%. A primeira observação que é destacada na Figura 3 é o desempenho significativamente melhor da plataforma ThingsBoard em ambos os cenários, mantendo um tempo de processamento similar independente da quantidade de mensagens em um cenário de *Smart City*. A aplicação mostrou-se ainda estável na relação de consumo de memória RAM e CPU para o cenário de *Smart City*, como observado nas Figuras 5 e 7, onde a plataforma mostra um crescimento regular proporcional ao aumento da carga, apresentando um pico apenas em uso de memória e CPU em um cenário com 40 mensagens por segundo.



**Figura 3. Tempo de Processamento - Smart City**



**Figura 4. Tempo de Processamento - Smart Health**

A plataforma FIWARE mostrou-se muito sensível em relação à quantidade de mensagens por segundo, sendo incapaz de executar até o fim o experimento de 40 mensagens por segundo em um cenário de *Smart City*, ilustrado na Figura 3 com a barra hachurada. Além do alto tempo de processamento, a aplicação apresentou ainda intervalos de confiança muito elevados em relação às demais soluções analisadas, caracterizando um cenário pouco estável. Comparando o cenário de 24 mensagens por segundos em ambas as infraestruturas, ilustrado na Figura 3, podemos observar ainda que a plataforma se beneficia muito de mais recursos computacionais, diminuindo de um tempo de processamento

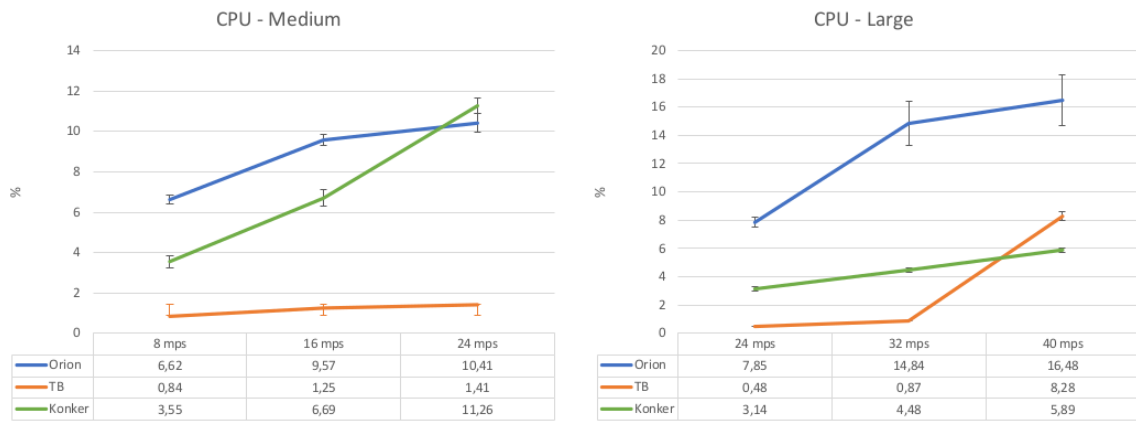
de 20 segundos em uma infraestrutura *Medium* para 1 segundo em uma infraestrutura *Large*. Este comportamento é corroborado ao observarmos o uso de CPU representado na Figura 5, mostrando que a solução FIWARE manteve um consumo percentual de CPU similar em ambas infraestruturas mas, obviamente, com o dobro de processamento efetivo na infraestrutura *Large*. A disponibilidade de maior recurso de processamento causou uma menor necessidade de uso de memória, caindo de 3359 Mb em uma infraestrutura *Medium* para 2411 Mb em uma infraestrutura *Large*.

A plataforma Konker sofreu um impacto menor da quantidade de mensagens por segundo em relação à plataforma FIWARE. Como mostrado na Figura 5, sua execução em uma infraestrutura *Medium* foi proporcional à quantidade de mensagens enquanto seu desempenho na infraestrutura *Large* mostrou um comportamento estável em relação ao tempo de processamento. A estabilidade está ligada diretamente a quantidade de recursos de processamento disponíveis, como observado na Figura 5 em ambos os cenários. A Figura 7 mostra ainda que a estabilidade da plataforma depende diretamente da quantidade de memória disponível, com valores muito próximos ao limite da infraestrutura, tornando a quantidade de memória um fator de impacto maior em relação à capacidade de CPU.

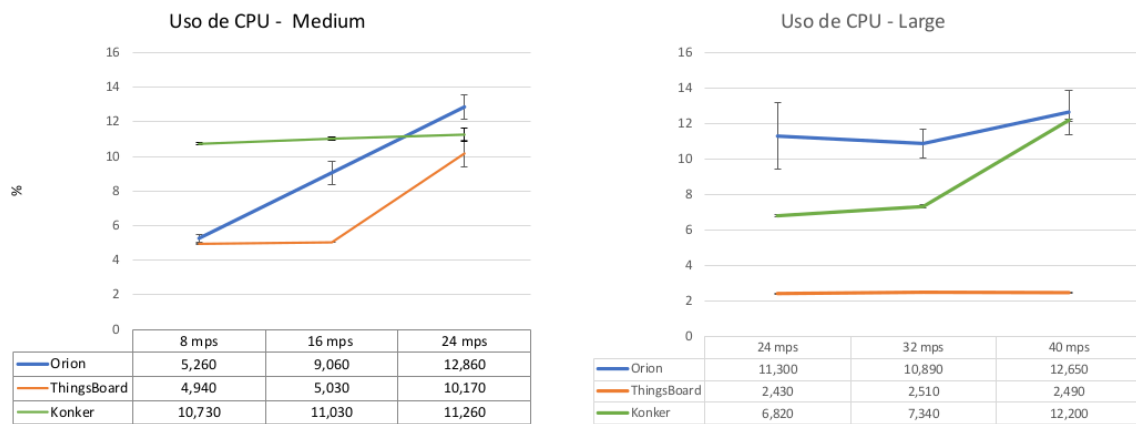
Analisando os cenários de Smart Health podemos observar na Figura 4 que novamente a plataforma ThingsBoard teve o melhor desempenho mas com menor significância que a plataforma Konker em relação ao cenário de Smart City. A plataforma teve o maior crescimento em relação ao uso de CPU, aumentando o consumo de processamento em dez vezes em uma infraestrutura *Medium* com 24 mensagens por segundo. O mesmo pode ser observado em relação ao uso de memória RAM em ambas as infraestruturas. Na comparação entre as infraestruturas, a plataforma ThingsBoard mostrou um aumento significativo no uso de memória, porém estável em relação ao uso de CPU e em tempo de processamento.

A plataforma FIWARE mostrou-se novamente a mais impactada pela quantidade de mensagens por segundo, assim como a com menor estabilidade dados os grandes intervalos de confiança observáveis na Figura 4. A escalabilidade também se torna um problema, não sendo capaz de executar até o fim o experimento de 40 mensagens por segundo em uma infraestrutura *Large*. Assim como nos cenários de *Smart City*, a plataforma FIWARE tirou grandes benefícios de maior capacidade de processamento, reduzindo significativamente seu tempo de processamento.

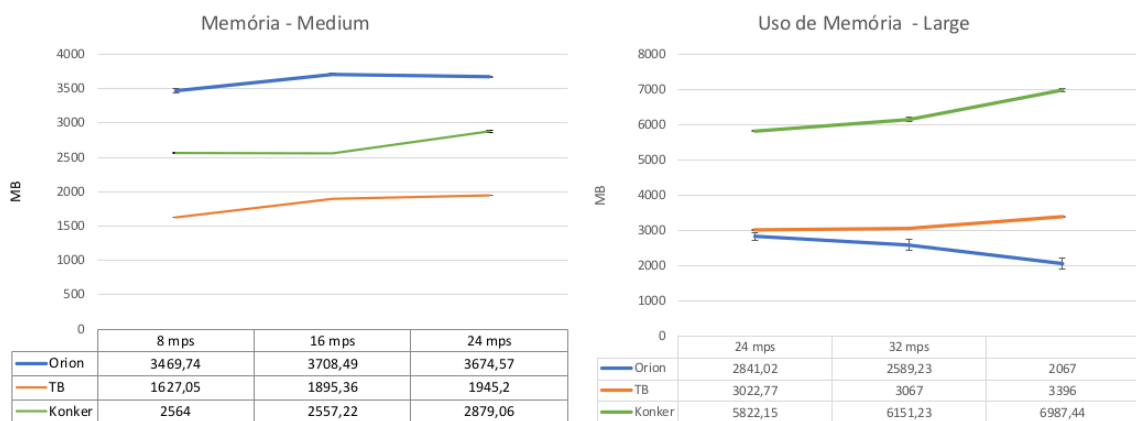
A plataforma Konker mostrou novamente um comportamento estável em um cenário de *Smart Health* mesmo com um tempo de processamento significativamente maior em uma infraestrutura *Large* quando comparada à estrutura *Medium* com o mesmo número de mensagens, o que pode ser visualizado na Figura 4. Entretanto, podemos observar na Figura 8 que o consumo de memória da plataforma Konker foi muito superior às demais plataformas, chegando novamente próximo ao limite da infraestrutura.



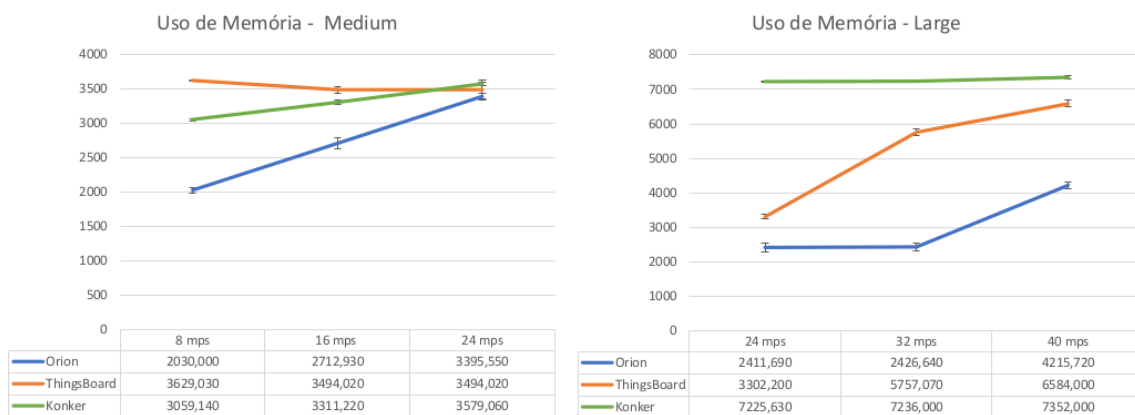
**Figura 5. Uso de CPU - Smart City**



**Figura 6. Uso de CPU - Smart Health**



**Figura 7. Uso de Memória - Smart City**



**Figura 8. Uso de Memória - Smart Health**

## 5. Discussão

A análise nos mostra que sem detrimento de escalabilidade, a plataforma ThingsBoard foi capaz de prover um baixo tempo de processamento, provando-se adequada para aplicações que exigem baixo tempo de processamento, como veículos autônomos ou aplicações de segurança. Em todos os cenários, a ThingsBoard teve uma média de tempo de processamento muito abaixo das demais aplicações. A mesma mostrou-se ainda leve o suficiente para ser capaz de operar em infraestruturas que dispõem de baixo poder computacional, como Raspberry Pi, tornando-se a plataforma avaliada mais estável e versátil.

A plataforma FIWARE se mostrou versátil e capaz de operar com diferentes tecnologias, tornando-se a solução avaliada mais abrangente e com mais recursos dado o grande número de GE's desenvolvidos e mantidos pela comunidade. Entretanto, sua operação demanda uma infraestrutura computacional mínima para tornar-se viável, e ainda assim, apenas para soluções que não demandam de tempo de processamento mínimo, sendo viável e prática para serviços como *Smart Agriculture* ou apenas recebimento e armazenamento de dados para utilização em aplicações de *Big Data* e *Analytics*. A plataforma FIWARE mostrou-se pouco escalável e mesmo com um consumo de recursos de hardware regular, não há o *tradeoff* de tempo de processamento por este consumo. A aplicação mostrou-se pouco estável em relação às demais, tendo um intervalo de confiança significativamente maior mesmo em cenários mais simples com apenas 8 mensagens por segundo.

Embora a plataforma Konker tenha se provado estável e escalável, a mesma mostrou-se altamente dependente de recursos computacionais, chegando próximo aos recursos totais da infraestrutura em muitos cenários. Mesmo com o alto consumo de recursos de hardware, a aplicação foi capaz de executar todos os experimentos até o fim. Portanto, a plataforma mostra-se viável porém não tão versátil quanto a plataforma ThingsBoard.

## 6. Conclusão

Este artigo explorou de maneira quantitativa o desempenho e escalabilidade de plataformas IoT em cenários reais. Os resultados mostram que plataformas de IoT de código livre são viáveis, estáveis e possuem desempenho adequado.

Embora as três plataformas tenham tido desempenhos aceitáveis, ficou claro que cada uma é viável para soluções distintas. A plataforma Thingsboard é adequada para aplicações que demandam baixo tempo de processamento ou ainda apresentam escassez de recursos computacionais. A plataforma FIWARE mostrou-se versátil por trabalhar com diferentes tecnologias, mas apresenta um maior custo computacional. Portanto, é recomendada para soluções de coleta e organização de dados, ou quando se dispõe de ambientes computacionais mais robustos. A plataforma Konker tem uma exigência maior de recursos computacionais quando comparada à plataforma Thingsboard, sendo recomendada para ambientes de nuvem.

Embora haja um grande número de plataformas no mercado, livres ou comerciais, no processo de escolha é importante que o desenvolvedor considere o tipo de recurso necessário. Ou seja, a decisão não deve ser apenas na facilidade de implementação ou nos protocolos disponíveis. Trabalhos futuros incluem aumentar a quantidade de plataformas avaliadas, explorando também diferentes protocolos, como MQTT, assim como explorar diferentes meios de transmissão (Wi-Fi e LoraWAN).

## Referências

- (2020). Konker Labs. [konkerlabs.com](http://konkerlabs.com) . Acessado em Março de 2020.
- Ahmed A.Ismail, Haitham S.Hamza, A. M. (2018). Performance evaluation of open source IoT platforms. 2018 IEEE Global Conference on Internet of Things (GCIoT).
- Broring, A., e. a. (2017). Enabling IoT ecosystems through platform interoperability. *IEEE Softw.* 2017, 34, 54–61.
- Cities S.L., B. (2018). Bettair - the fiware-powered solution on a mission to map urban air quality in high resolution.
- Daniel Happ, Niels Karowski, e. a. (2018). Meeting IoT platform requirements with open pub/sub solutions, 72: 41. *Annals of Telecommunications*.
- Galli, A., e. a. (2019). Multi-user ECG monitoring system based on IEEE standard 802.15.6. 2019 IEEE International Symposium on Measurements Networking (MN).
- Hamdan Hejazi, e. a. (2018). Survey of platforms for massive IoT. 2018 IEEE International Conference on Future IoT Technologies (Future IoT).
- Heideker A., e. a. (2019). IMAIoT Infrastructure Monitoring Agent for IoT: Um Agente Monitor de Infraestruturas para Ambientes de IoT. *Anais Estendidos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.
- Inc., D. (2019a). [docker.com/why-docker](https://docker.com/why-docker) ; Acessado em Dezembro de 2019.
- Inc., T. (2019b). Thingsboard documentation. [thingsboard.io/docs/reference/](https://thingsboard.io/docs/reference/) ; Acessado em Dezembro de 2019].
- K. Natarajan, B. Prasath, P. K. (2016). Smart health care system using internet of things. *Journal of Network Communications and Emerging Technologies (JNCET)*.
- Liu, S. (2017). Platforms used to run connected devices in software projects -2017. IoT - number of connected devices worldwide 2015-2025. [statista.com/statistics/869359/worldwide-technologies-used-for-connected-devices-in-software-projects](https://statista.com/statistics/869359/worldwide-technologies-used-for-connected-devices-in-software-projects); Acessado em Novembro de 2019.

- Liu, S. (2019). Internet of Things in the U.S. - Statistics Facts. Internet of Things - number of connected devices worldwide 2015-2025.
- Ltda., A. Amazon AWS IoT.
- Ltda., G. (2019a). Google Cloud IoT. cloud.google.com ; Acessado em Dezembro de 2019.
- Ltda., I. (2019b). IBM. ibm.com/br-pt/cloud ; Acessado em Dezembro de 2019.
- Ltda., M.
- Luca Calderoni, Antonio Magnani, D. M. (2019). IoT Manager: a case study of the design and implementation of an Open Source IoT Platform. IEEE 5th World Forum on Internet of Things (WF-IoT).
- Mauro A. A., e. a. (2018). Performance evaluation of IoT middleware through multicriteria decision-making. 2018 IEEE Global Communications Conference - GLOBECOM.
- Mobile Alliance, O. (2019). Open Mobile Alliance. NGSi requirements, OMA-RD-NGSI-V1.0. openmobilealliance.org ; Acessado em Dezembro de 2019.
- Pascoli, E. S. S. D. and Iannaccone, G. (2016). Low-power wearable ECG monitoring system for multiple-patient remote monitoring. IEEE SENSORS JOURNAL, V.
- Ramón Martínez, Juan Ángel Pastor, B. and Iborra, A. (2016). A testbed to evaluate the FIWARE-Based IoT platform in the domain of precision agriculture. Sensors 2016 Nov; 16(11): 1979.
- Singh, K.; Kapoor, D. (2017). A survey of IoT platforms: Create your own internet of things. IEEE Consumer Electronics Magazine.
- Statista, R. D. (2019). Internet of things - number of connected devices worldwide 2015-2025. Statista Research Department. statista.com/statistics/471264/iot-number-of-connected-devices-worldwide ; Acessado em Novembro de 2019.
- SWAMP (2017). Smart Water Management Platform. swamp-project.org ; Acessado em Dezembro de 2019.
- Technologies, K. (2019). Kaa Webinar - introducing new kaa 1.1. kaaproject.org/what-is-iot-platform ; Acessado em Novembro de 2019.
- Zhe Yang, Qihao Zhou, L. L. K. Z. W. X. (2016). An IoT-cloud based wearable ECG monitoring system for smart healthcare. Journal of Medical Systems volume 40, Article number: 286 (2016).
- Zyrianoff, Ivan; Heideker, A. S. D. K. C. (2018). Scalability of an Internet of Things Platform for Smart Water Management for Agriculture. 2018 23rd Conference of Open Innovations Association (FRUCT). pp. 432-439.
- Zyrianoff, I.; Borelli, F. K. C. (2017). SenSE – Sensor Simulation Environment: Uma ferramenta para geração de tráfego IoT em larga escala. SBRC 2017. Salão de Ferramentas.