

# Utilizando uma plataforma de IoT open source: a Plataforma Konker

Luis Fernando Gomez Gonzalez<sup>1</sup>, Douglas Oliveira Polinário<sup>1</sup>, Alexandre Junqueira<sup>1</sup>

<sup>1</sup>Konker

Av. Brg. Faria Lima, 1982 – 10º Andar – 01451-000 – São Paulo – SP – Brazil

{luis, douglas, alexjunq}@konkerlabs.com

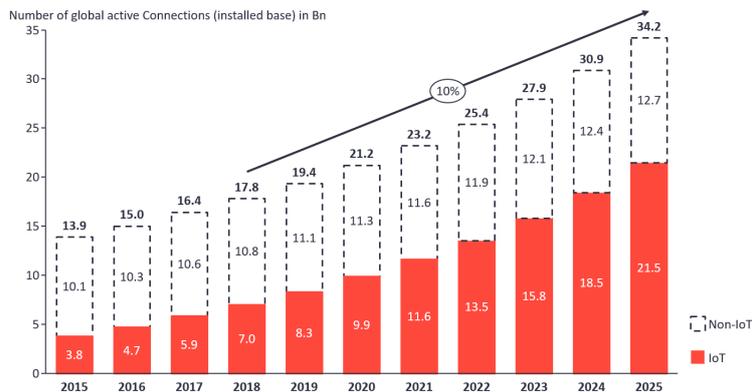
**Abstract.** *In this paper we will discuss the soaring need for a Internet of Things (IoT) platform to operate on increasingly complex real-world problems. In this scenario, the architecture and deployment used on the open source Konker Platform will be shown and it's use for a simple device based on the ESP8266 microcontroller will be demonstrated using the platform operating on the cloud.*

**Resumo.** *Neste trabalho será discutida a necessidade crescente do uso de uma plataforma para dispositivos de Internet das Coisas (IoT, do inglês Internet of Things) para atuação em problemas reais cada vez mais complexos. Nesse cenário, serão exibidas a arquitetura e a implementação usadas na plataforma open source de IoT Konker assim como será demonstrado o uso de um dispositivo simples, baseado no microcontrolador ESP8266, acessando a plataforma baseada em nuvem.*

## 1. Introdução

Dados atuais estimam que hoje existem mais de 17,8 milhões de dispositivos conectados na internet, dos quais 10,8 milhões são dispositivos de comunicação pessoal, como smartphones e computadores [Iot-Analytics 2018]. Dessa forma, estima-se que existam hoje mais de 7 bilhões de dispositivos enviando dados para a internet que não são relativos a comunicação pessoal, ou seja, que coletam informações de forma autônoma ou pseudo-autônoma, caracterizando-se como dispositivos de Internet das Coisas ou IoT, do inglês *Internet of Things*. Essa classe de dispositivos possui um crescimento mais rápido que os dispositivos de comunicação pessoal e devem se tornar a maioria dos dispositivos na internet em 2021, como ilustra a Figura 1.

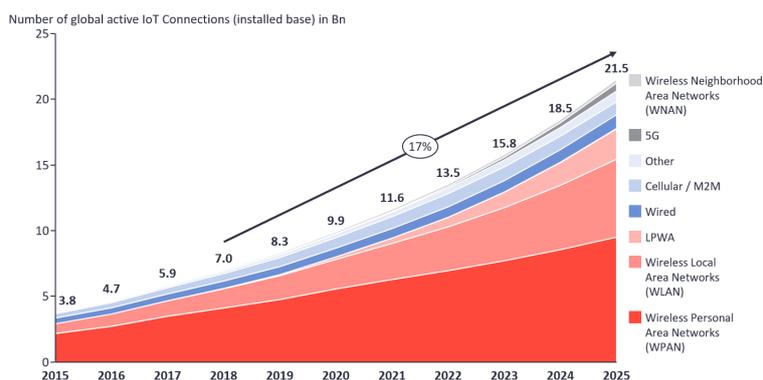
### Total number of active device connections worldwide



**Figura 1. Número projetado de dispositivos de comunicação pessoal (non-IoT) e de IoT até o ano de 2025 (gráfico retirado de [IoT-Analytics 2018]).**

Dentre os dispositivos conectados, existe uma miríade de modelos diferentes com sensores e atuadores próprios para as tarefas para as quais foram designados. Dessa forma, além de medirem grandezas físicas distintas e atuarem das mais variadas formas, até mesmo a configuração dos dispositivos possui particularidades quanto a taxa de transferência de dados, periodicidade e robustez além dos mais variados modelos de conectividade para suporte das características desejadas para o dispositivo. Hoje o modelo mais comum é a conectividade por *Wireless Personal Area Networks* (WPAN), que compreende redes de pequeno alcance como Zigbee, Z-Wave e Bluetooth, e conectividade usando *Wireless Local Area Networks* (WLAN) que compreende conexões WiFi. Conexões de longo alcance e baixo consumo, conhecidas como LPWAN, estão começando a se tornar mais comum, tendo um crescimento expressivo esperado para os próximos anos, como ilustrado na Figura 2.

### Global Number of Connected IoT Devices



**Figura 2. Número projetado de conexões a internet por dispositivos de IoT, divididos por tipo de conexão (gráfico retirado de [IoT-Analytics 2018]).**

Nesse contexto, é necessário uma forma comum e transparente de administrar e se comunicar, independente do dispositivo ou de como ele está conectado. Além disso, ter os dados consolidados em um só lugar permite explorar de forma facilitada o potencial de IoT em sua plenitude, permite tomar decisões baseadas em dados em tempo

real, ou próximo a tempo real, usando ferramentas capazes de extrair valor de dados com alta dimensionalidade, como métodos de *Machine Learning*. Uma solução para essa complexidade é a utilização de uma Plataforma de IoT que englobe essas funções e permita uma análise de dados simples, utilizando APIs de acesso bem documentadas.

## 2. Plataforma de IoT Konker

Construir soluções de IoT que resolvam problemas do mundo real não é uma tarefa trivial [Konker 2018]. Em geral, é necessária uma combinação de habilidades:

**Operational Technology:** Trata-se de entender o problema específico que se está tentando resolver e como resolvê-lo no mundo real. Se a solução é uma solução de irrigação na agricultura, é necessário conhecer quais os problemas associados, quais os equipamentos são usados para irrigar, de que forma trabalham, onde estão suas limitações, e assim por diante.

**Hardware:** Trata-se de construir o equipamento que trabalha como ponte entre OT e a Internet. Esse equipamento, dependendo das condições de trabalho, talvez tenha que ser resistente à umidade, sujeira, vibração, temperatura, conectividade. entre outros aspectos importantes para uma instalação em campo.

**Software:** Trata-se do I (Internet) em "Internet das Coisas". Construir uma solução de software na Internet capaz de receber, armazenar, processar e analisar grandes volumes de dados, com alta disponibilidade, segurança e desempenho, além de ser capaz de auxiliar no processo de gestão da rede de dispositivos utilizados para alimentar e operacionalizar o fluxo de dados da solução.

A Plataforma Konker é uma ferramenta que endereça o último ponto desta lista: o Software na Internet. Trata-se de uma Plataforma na nuvem que opera num modelo *PaaS* (Platform as a Service) e que realiza o trabalho de receber, armazenar, processar, enriquecer, transformar os dados coletados por qualquer dispositivo, assim como simplifica o trabalho de integração e gestão deles. O objetivo é liberar tempo e esforço para que o criador de uma solução baseada em IoT possa concentrar seus esforços na solução em si, no hardware, no problema que deve ser resolvido e em seu modelo de negócio e mesmo assim tenha uma solução final altamente robusta e escalável. A prototipação de soluções é especialmente simples: em alguns minutos é possível começar a enviar e visualizar dados. A principal contribuição para o estado da arte com relação a outras plataformas de IoT se encontra no desenvolvimento pensado no paradigma de comunicação brasileiro, onde a maior parte das comunicações são bidirecionais e pouco resilientes, além de permitir o uso de dispositivos muito simples, com baixo poder computacional. A arquitetura da plataforma Konker trata as conexões downstream e upstream da mesma forma idêntica, permitindo as mesmas operações para sensores e atuadores, incluindo dispositivos que operam como ambos. Com relação a resiliência, é previsto nativamente o uso de *Gateways* que permitem mapear de forma transparente dispositivos que podem ter conectividade restrita ou mesmo não-IP, permitindo o uso de dispositivos com hardware bastante simples.

### 2.1. Arquitetura da Plataforma

A Plataforma Konker foi desenvolvida usando componentes de código aberto e está disponível no GitHub [Konker 2019b] sob licença *Apache-2.0*. O uso de uma licença bastante permissiva tem como premissa a ideia de que soluções baseadas em Internet das

Coisas são muito amplas para ser construídas por uma única pessoa ou empresa: elas serão construídas por muitos, e que para atingir todo seu potencial de uso ela deve ser amigável com a criação e aceleração de novos negócios. Ela também está disponível no formato *Docker* [Konker 2019a], possibilitando a instalação em ambientes locais sem preocupações com dependências e inconsistências de bibliotecas e softwares. Para instalação usando o *Docker* é necessário: um Sistema com o Docker instalado (Linux, OSX, Windows 10), 1 GB de memória RAM e 1 GB de espaço em disco disponível.

Em sistemas Linux ou OSX, basta executar o comando abaixo para baixar e instalar o docker contendo a plataforma:

```
sudo docker run -p 80:80 -p 1883:1883 -v /yourpersistentfolder:/data/db -itd konkerlabs/konker-platform
```

Após a instalação, a interface da plataforma se encontrará em <http://localhost>, com as credenciais padrões **usuário: admin@localhost** e **senha: changeme**, e pode ser aberta no navegador de preferência. Maiores detalhes da instalação podem ser encontrados na referência [Konker 2019a]. Embora seja possível instalar localmente, a plataforma foi projetada para rodar em nuvem e pode ser acessada em <https://demo.konkerlabs.net>. Nessa página é possível criar um usuário com capacidade de utilizar até 5 dispositivos gratuitamente. É importante observar que rodando na nuvem é possível atingir mais facilmente o objetivo principal da plataforma de conectar dispositivos localizados em qualquer parte do planeta de forma transparente, robusta e consistente.

Como discutido no início da seção, a arquitetura da Plataforma Konker na nuvem é composta por módulos distribuídos por um cluster de máquinas em diferentes Data Centers, orquestrada usando o *Marathon/Mesos* [Marathon 2019] para garantir alta disponibilidade.

Um macro desenho de todo o espaço da solução da plataforma Konker pode ser visto na Figura 3. Nela é observada a solução desde o lado do dispositivo e Gateway até a aplicação por parte do usuário. Embora seja importante observar o todo, o escopo desse trabalho se concentra em três principais componentes na plataforma: *Data*, *Core Platform* e *API*. Esses componentes podem ser vistos em detalhes na Figura 4.

**Data:** Componente que recebe e envia as mensagens via *MQTT* ou *REST* [Salman and Jain 2019] para os dispositivos e Gateways, além de enfileirar as requisições usando o RabbitMQ [RabbitMQ 2019]. O *broker* utilizado para comunicação *MQTT* é o *Mosquitto* [Mosquitto 2019], com planejamento para utilização futura do *VerneMQ* [VerneMQ 2019]. A ingestão de dados via *HTTP* é feita utilizando como *backend* o *NGINX* [NGINX 2019], juntamente com código em Java para migração dos dados. Nessa camada também ocorrem os processamentos de rotas e regras simples na ingestão dos dados bem como *Bridges* de transformação de formato dos dados, para ingestão de dispositivos com pacotes de dados diferenciados, como dispositivos de *tracking* (GPS) comerciais.

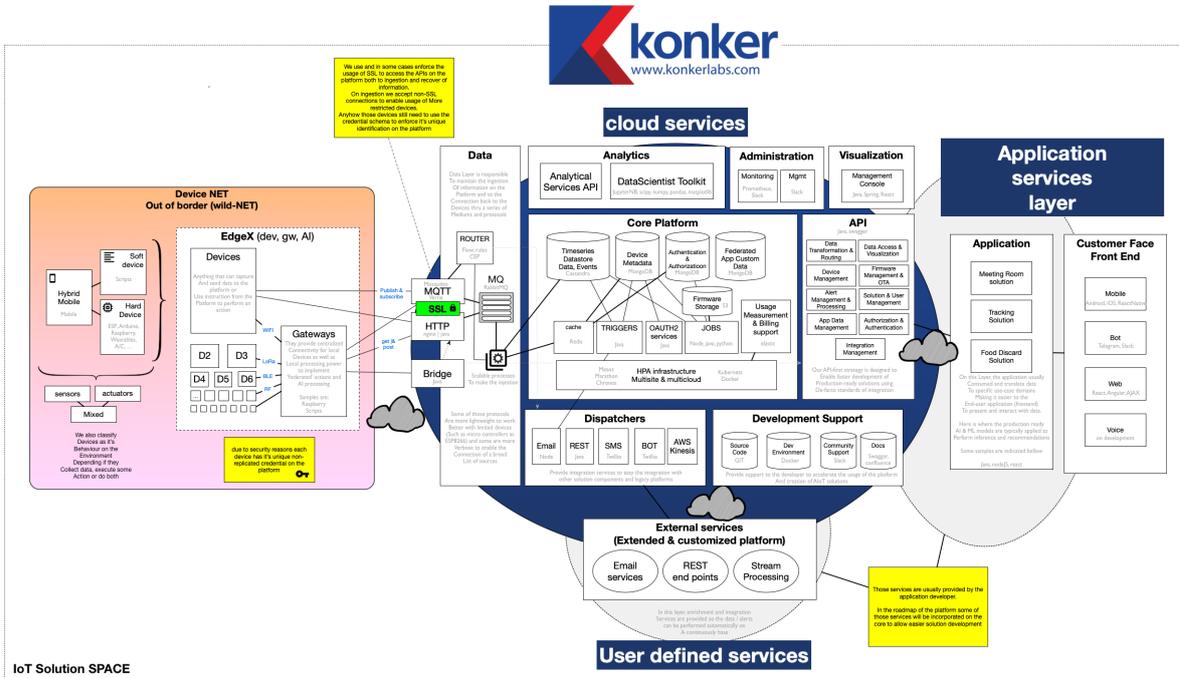


Figura 3. Espaço de solução da Plataforma Konker.

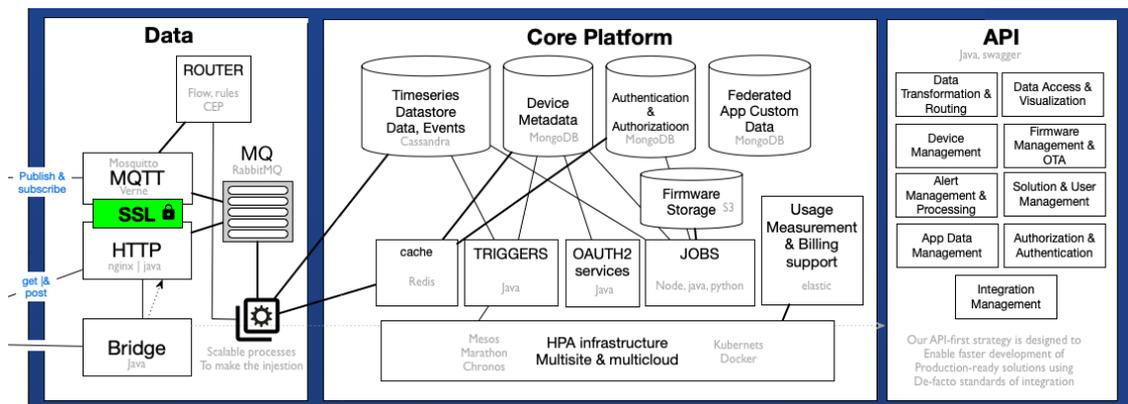


Figura 4. Detalhe dos 3 componentes da plataforma Konker abordados nesse trabalho.

**Core Platform:** Componente central da plataforma onde ocorre leitura e gravação dos dados nos diferentes bancos de dados utilizados além da autenticação dos dispositivos. Para os dados provenientes de dispositivos (*payload* das mensagens) é utilizado um banco de dados distribuído *Cassandra* [Cassandra 2019] enquanto para os metadados desses dispositivos se encontra em um *Database Mongo DB* [MongoDB 2019]. Outro banco de dados usado nessa camada é o *Redis* [Redis 2019] como cache para processos como Alertas gerados na plataformas por dispositivos. Existe também um *storage* para arquivos, como *firmwares*, que se utiliza do serviço S3 da Amazon [Amazon 2019]. O software desenhado para integração de todos os componentes dessa camada se utilizam principalmente das linguagens: *Java*, *Node.JS* e *Python*.

**API:** Componente projetado para controle de todos os itens configuráveis pelo usuário, ou seja, essa camada é responsável pela criação, modificação e exclusão dos metadados de dispositivos, usuários, gateways, alertas e rotas. O principal componente dessa camada é o *Swagger* [Swagger 2019] e ela possui toda a comunicação com o usuário usando *REST*. Também faz parte dessa camada a própria documentação da API também baseada em ferramentas do *Swagger*. A maior parte do código dessa camada, assim como das demais, é escrito em *Java*.

### 3. Usando a Plataforma Konker

O modo mais simples de testar a plataforma é utiliza-la na nuvem, gratuita para até 5 dispositivos, disponibilizada no site: <https://demo.konkerlabs.net>. Caso haja a necessidade de usar mais dispositivos, existe a possibilidade de contratação de mais dispositivos na plataforma na nuvem na modalidade *PaaS* ou usando uma instalação particular como descrito em detalhes na referência [Konker 2019a]. Quatro conceitos importantes antes de iniciar a criação de dispositivos e envio de dados são [Konker 2018]:

**Dispositivo:** Um dispositivo na Plataforma Konker é uma representação de um dispositivo físico (ou virtual) que envia ou recebe dados. É o equivalente a "Things" quando se fala em "Internet of Things". Apenas dispositivos cadastrados na plataforma podem enviar e receber dados.

**Canal:** O conceito de canal está associado ao dispositivo e é uma maneira de agrupar mensagens com conteúdo semelhante para que possam ser tratadas em conjunto. Um exemplo é o caso em que um dispositivo tem 2 sensores: um sensor de umidade e um sensor de temperatura. Nesse caso, é aconselhável que os dados enviados pelo dispositivo sejam enviados para um canal "temperatura" e os dados de umidade para um canal de "umidade". Dessa forma, pode-se aplicar filtros ou visualizações apenas para um tipo de informação ou de outra independentemente. Canais não precisam ser cadastrados: basta informar seu nome durante o envio da informação que ele será automaticamente criado.

Os canais também são relevantes para o dispositivo que recebe os dados. Suponha que um dispositivo receba da plataforma 2 tipos de informação, como por exemplo comandos e configurações. Nesse caso, o dispositivo pode fazer uma

assinatura para receber apenas os dados de comandos ou apenas os dados de configuração.

**Rota:** Uma rota é um caminho que liga um dispositivo de entrada a um dispositivo (ou webservice) de saída. Dessa forma, se há um dispositivo que contém um botão chamado "Botao 01" e um dispositivo que contém um LED chamado "LED 01", pode-se criar uma rota que liga o dispositivo "Botao 01" ao dispositivo "LED 01", de forma que as mensagens enviadas ao pressionar o botão sejam encaminhadas ao dispositivo que acende ou apaga o LED. Nesse cenário, também seria necessário especificar, na rota, qual o canal em que os dispositivos se comunicam.

**Transformação:** Um ponto de transformação permite que uma mensagem seja arbitrariamente manipulada durante o processamento da rota para que ela esteja no formato que o dispositivo de saída possa entender. Isso é realizado através de chamadas a serviços externos à plataforma que devem receber as mensagens, processá-las e devolvê-las à plataforma para continuar o processamento.

Um tutorial com todos os passos necessários para conectar um dispositivo na plataforma pode ser encontrado no vídeo: <https://www.youtube.com/watch?v=1L6Fsi1kDSw>. Nesse exemplo é conectado à plataforma um medidor da tensão de alimentação de um *NodeMCU*, que é uma placa de ensino baseada no microcontrolador ESP8266 [Espressif 2019]. Caso uma utilização mais avançada ou automatizada seja necessária, também está disponível em vídeo o tutorial de utilização da API no link: <https://www.youtube.com/watch?v=HWRTcOwh978>.

A documentação completa, em texto, descrevendo em detalhes o uso da plataforma e da API, pode ser encontrada em: <https://www.konkerlabs.com/developers/developers-pt.html>

#### 4. Descrição da Demonstração planejada para o SBRC

A demonstração planejada para o O Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos consiste em reproduzir os passos necessários para conectar um dispositivo na Plataforma Konker (PaaS), criar uma rota entre dois dispositivos, visualizar os dados de forma rápida na própria plataforma e rodar uma análise em *Python* nos dados obtidos em tempo real. Também serão exibidos dados obtidos em tempo real dentro do Campus na Universidade Estadual de Campinas (dentro do programa SmartCampus) e ao menos uma aplicação consumindo dados da plataforma. Como dispositivos serão usados microcontroladores *ESP8266* e *ESP32* [Espressif 2019] além de sistemas mais complexos rodando Linux, como o *Raspberry Pi* [Raspberrypi 2019]. Todos os sensores, atuadores e computadores serão fornecidos pela Konker, sendo necessário no local apenas uma conexão com a Internet (WiFi para os dispositivos), uma mesa ou bancada para interação das pessoas com os dispositivos, acesso a rede elétrica e dois monitores ou televisores para exibição dos dados, da plataforma e aplicações.

## Referências

- Amazon (2019). Amazon Simple Storage Service (S3). <https://aws.amazon.com/s3/> - Acessado em 23/03/2019.
- Cassandra, A. (2019). Apache Cassandra: Manage massive amounts of data, fast, without losing sleep. <http://cassandra.apache.org/> - Acessado em 23/03/2019.
- Espressif (2019). MCUs Datasheet. <https://www.espressif.com/sites/default/files/documentation/> - Acessado em 25/03/2019.
- Iot-Analytics (2018). State of the IoT 2018: Number of IoT devices now at 7B - Market accelerating. <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b> - Acessado em 16/03/2019.
- Konker (2018). Guia de Uso da Plataforma Konker. <https://konker.atlassian.net/wiki/spaces/DEV/pages/28180518/Guia+de+Uso+da+Plataforma+Konker> - Acessado em 16/03/2019.
- Konker (2019a). Docker Hub da Konker - Plataforma. <https://hub.docker.com/r/konkerlabs/konker-platform> - Acessado em 16/03/2019.
- Konker (2019b). Github da Konker - Plataforma. <https://github.com/KonkerLabs/konker-platform> - Acessado em 16/03/2019.
- Marathon (2019). Marathon: A container orchestration platform for Mesos and DC/OS. <https://mesosphere.github.io/marathon/> - Acessado em 16/03/2019.
- MongoDB (2019). MongoDB: Open Source Document Database. <https://www.mongodb.com/> - Acessado em 23/03/2019.
- Mosquitto (2019). Eclipse Mosquitto: An open source MQTT broker. <https://mosquitto.org/> - Acessado em 23/03/2019.
- NGINX (2019). NGINX — High Performance Load Balancer, Web Server. <https://www.nginx.com/> - Acessado em 23/03/2019.
- RabbitMQ (2019). RabbitMQ: Messaging that just works. <https://www.rabbitmq.com/> - Acessado em 23/03/2019.
- Raspberrypi, F. (2019). Raspberry Pi: A small and affordable computer that you can use to learn programming. <https://www.raspberrypi.org/> - Acessado em 25/03/2019.
- Redis (2019). Redis - An open source, in-memory data structure store, used as a database, cache and message broker. <https://redis.io/> - Acessado em 23/03/2019.
- Salman, T. and Jain, R. (2019). A Survey of Protocols and Standards for Internet of Things. *arXiv e-prints*, page arXiv:1903.11549.
- Swagger (2019). Swagger - The Best APIs are Built with Swagger Tools. <https://swagger.io/> - Acessado em 25/03/2019.
- VerneMQ (2019). VerneMQ - A MQTT broker that is scalable, enterprise ready, and open. <https://vernemq.com/> - Acessado em 23/03/2019.