

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/354603541>

Plataforma FIWARE para monitorização de poluição

Conference Paper · September 2021

CITATIONS

0

3 authors, including:



Nuno Datia

Instituto Politécnico de Lisboa

26 PUBLICATIONS 44 CITATIONS

SEE PROFILE



Nuno Cruz

Instituto Politécnico de Lisboa

16 PUBLICATIONS 27 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



5G-MOBIX [View project](#)

Plataforma FIWARE para monitorização de poluição *

Ivo Pedroso¹, Nuno Datia^{1,2}, and Nuno Cruz^{1,3}

¹ FIT - Future Internet Technologies, ISEL - Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa

A25450@alunos.isel.pt, {nuno.datia, nuno.cruz}@isel.pt

² NOVALINCS, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

³ LASIGE, Faculdade de Ciências, Universidade de Lisboa

Resumo A poluição atmosférica afeta a saúde da população e é um factor de aumento da taxa de mortalidade. Entidades responsáveis pela gestão de grandes cidades procuram soluções que permitam monitorizar em tempo real os níveis de poluição em grandes áreas urbanas. No entanto, as soluções existentes para a monitorização de poluição são proprietárias ou não apresentam um grau de maturidade de integração, que facilite a expansão de uma rede de sensores com custos baixos de manutenção. Neste trabalho apresenta-se uma arquitectura da solução e uma implementação de referência, para uma plataforma distribuída de código aberto, que permita gerir uma rede de sensores. Esta plataforma permite acoplar e desacoplar, em tempo real, os sensores de poluição atmosférica, armazenar o histórico de medições registadas, agregar dados e disponibilizar um *dashboard* que permita posterior análise da informação armazenada. A arquitetura de solução é disponibilizada em contentores de execução virtual, recorrendo a diversos componentes do quadro de referência FIWARE e a componentes desenvolvidos à medida. O meio de comunicação usado na transmissão de dados entre os sensores e a plataforma, é uma rede baseada em LoRaWAN, a qual permite comunicações sem fios de longo alcance e de baixa potência. Foi assim selecionada a *The Things Network* (TTN), uma rede pública que implementa o protocolo LoRaWAN, que permite a comunicação entre sensores e a plataforma proposta. Foi selecionado o protocolo MQTT como forma de comunicação entre a plataforma proposta e a TTN. A plataforma proposta disponibiliza uma interface com o utilizador, implementada como um *dashboard* acessível a partir de um navegador, onde é possível observar a localização dos sensores registados sobre um mapa, a informação sobre seu estado operacional, bem como a informação sobre as medições observadas. Todos os componentes da plataforma são parametrizados através de um único ficheiro de configuração, que, através de um processo de automação, possibilita a instalação e execução da plataforma através de um único comando.

* Gostaríamos de agradecer à equipa do CGIUL o apoio prestado ao abrigo do Protocolo “Dados ao Serviço de Lisboa”. Este trabalho é apoiado pelo LASIGE (UIDB / 00408/2020) e pela NOVA LINCS (UIDB / 04516/2020) com o apoio financeiro da FCT- Fundação para a Ciência e a Tecnologia, através de fundos nacionais.

Keywords: Cidades Inteligentes · FIWARE · IoT · LoRaWAN.

1 Introdução

A poluição atmosférica tem um grande impacto na saúde da população [10]. A OMS estima que a poluição atmosférica seja responsável por 4.2 milhões de mortes anuais devido a acidente vascular cerebral, doença cardíaca, cancro do pulmão, doenças respiratórias crónicas, e que 91% da população mundial viva em locais onde o nível da qualidade do ar excede os limites definidos pela OMS [10]. A exposição a partículas finas, de diâmetro inferior a $2.5 \mu\text{m}$ (pm2.5) e inferior a $10 \mu\text{m}$ (pm10), são consideradas a sexta maior causa de morte prematura no Sudeste Asiático [1]. Estes poluentes penetram facilmente nos pulmões, provocando dificuldades respiratórias e várias doenças pulmonares [15]. Estima-se que sejam causadores de 43% da doença pulmonar obstrutiva crónica (DPOC) e 29% das causas associadas ao cancro do pulmão.

Entidades com responsabilidade na gestão dos grandes centros urbanos, têm vindo a aumentar o investimento em sensores de monitorização, como é o caso da Câmara Municipal de Lisboa [2], com o objetivo de monitorizar a qualidade do ar em tempo real. O uso de estações de monitorização especializadas, apesar de terem uma melhor precisão nas medições, trazem algumas desvantagens, nomeadamente o elevado custo financeiro na aquisição e manutenção, a pouca ou nenhuma mobilidade, levando a uma reduzida área de cobertura dos centros urbanos. Estas limitações fomentam as entidades a procurarem novas soluções que permitam poupança na aquisição e gestão dos sensores, com maior mobilidade e com melhor capacidade de escala. Uma solução possível é o uso de pequenos sensores de monitorização da qualidade do ar, móveis, auto-alimentados, com capacidade de comunicação de longo alcance e de baixo consumo [14]. Para suportar essa rede de dispositivos disseminados pelos centros urbanos, com potencial gerador de grande volume de dados, é necessária uma plataforma que recolha os dados registados, os armazene e disponibilize uma forma de obter e/ou visualizar, que permita posterior análise dos dados registados. Esta plataforma deverá simplificar o processo de integração de novos sensores, permitir observar o estado da rede de sensores, bem como apresentar um resumo dos dados recolhidos num *dashboard*. Os dispositivos IoT ao serem integrados na plataforma, além de transmitir as métricas recolhidas pelos seus sensores, deverão também transmitir a suas coordenadas geográficas, para que seja possível à plataforma mostrar aos utilizadores a localização dos dispositivos registados, num mapa geográfico na interface de utilizador da plataforma.

Neste artigo propõe-se uma arquitectura de solução, e uma implementação de um protótipo, para uma plataforma de código aberto, descentralizada, baseada em contentores, com *deploy* através da execução de um simples comando. Esta arquitetura faz uso da *framework* FIWARE [3], assumindo que os sensores comunicam através de LoRaWAN [13].

A FIWARE é uma *framework* de código aberto cujo intuito é a disponibilização de componentes para desenvolvimento de soluções aplicadas a diferentes

domínios de aplicação, incluindo as *Cidades Inteligentes*. Todos componentes FIWARE disponibilizam uma interface comum, NGSIv2 [7], que é uma interface API RESTful [12] orientada para a transmissão e interrogação de informação, permitindo a interação entre componentes FIWARE e componentes terceiros, minimizando o esforço de integração e troca de componentes, se necessário.

Para realizar a troca de dados entre os dispositivos *Internet of Things* (sensores) e a plataforma proposta, dadas as restrições já indicadas dos dispositivos ao nível da mobilidade e consumo energético, é utilizado o protocolo de comunicação LoRaWAN. Para tal, é necessário que exista uma entidade que possibilite a troca de mensagens entre os dispositivos numa rede LoRaWAN e a plataforma proposta. Foi escolhida plataforma TTN ⁴ (*The Things Network*) para fazer essa interação, pois disponibiliza um conjunto de servidor de rede e servidor aplicacional, com capacidade de comunicação e registos de dispositivos numa rede LoRaWAN, e disponibiliza uma fila MQTT [6] que permite a aplicações interessadas, subscreverem a notificações de envio de mensagens por parte dos dispositivos IoT. Um dos fatores para a escolha da TTN foi também de disponibilizar um acesso comunitário, sem custos financeiros associados.

O contributos deste artigo são: (i) uma proposta de arquitectura de uma plataforma para gestão de sensores; (ii) um protótipo demonstrador da solução.

O resto do artigo encontra-se organizada da seguinte forma: na secção 2 descrevem-se alguns trabalhos com objectivos semelhantes; de seguida, na secção 3 é apresentada a arquitectura de solução, sendo discutidos em detalhe os componentes implementados; na secção 4 são discutidas as vantagens da plataforma proposta, terminando o artigo com a secção 5 que formula as conclusões do trabalho.

2 Trabalho relacionado

Dada a crescente preocupação na monitorização da poluição, encontramos diversas plataformas baseada na tecnologia FIWARE, que propõem soluções de recolha, transformação e disponibilização de dados de dispositivos *Internet of Things*(IoT).

A plataforma proposta por [5] tem como objetivo recolher medições da poluição do ar, usando dispositivos Arduino ⁵, munidos com sensores de medição de diferentes poluentes. Esta plataforma foi usada para monitorizar a qualidade do ar na Cidade do México. A plataforma utiliza componentes da *framework* FIWARE, nomeadamente o *Orion Context Broker* [4] para conter os dados de contexto da plataforma e permitir notificar outras aplicações interessadas na evolução das leituras. Também usa o componente FIWARE QuantumLeap [11], responsável por receber os dados transmitidos pelos dispositivos, converte-los para o formato NGSIv2 [7], de modo a registar esses dados no *Orion Context Broker*. Por sua vez é notificado pelo *Orion Context Broker* de novos dados e

⁴ Ver www.thethingsnetwork.org

⁵ Ver www.arduino.cc

armazena os mesmo com base numa série temporal no sistema de gestão de bases de dados CrateDB ⁶. Na comunicação entre os dispositivos e plataforma, os autores usam o microprocessador Cloudino ⁷ em paralelo com o Arduino, o qual permite a comunicação através de uma rede WiFi, se disponível, ou da rede GSM em alternativa. A visualização e análise de dados é possível através da aplicação Grafana ⁸, que, após ligada à base de dados, permite construir diferentes *dashboards* para análise dos dados. Também é possível ligar a aplicação Green Route ⁹ ao *Orion Context Broker*, uma aplicação baseada no suporte à navegação por GPS com informação da qualidade do ar, a qual também faz parte do portfólio FIWARE.

Kaivonen and NGai [8] propõem um sistema de monitorização da poluição do ar através de dispositivos IoT instalados em autocarros da cidade Sueca de Uppsala, complementados com sensores estacionários, instalados em localizações não cobertas pelo percurso dos autocarros utilizados. Os dados dos sensores nos autocarros são transmitidos via 4G. Por sua vez os sensores estacionários transmitem os dados através de um protocolo de comunicação de baixa potência *6LoWPAN* para *gateway relays*. No entanto, estes usam WiFi para comunicar com o servidor da aplicacional existente na nuvem. Os dados observados após transmitidos para o servidor, são armazenados numa base de dados NoSQL e publicados num *Context Broker* MQTT, disponível para outras aplicações. Os sensores usados nos dispositivos instalados em autocarros são Waspote ¹⁰ de baixa potência do fabricante Libelium. A informação gerada pode ser observada num mapa, na forma de heatmap sobre o conjunto das métricas observadas, ou com marcadores das últimas 10 medições observadas relativas a um autocarro concreto.

Ambas as propostas estão dependentes de dispositivos IoT específicos, Arduino e Cloudino em [5] e Libelium WaspMote em [8]. Apesar de [5] usar componentes da *framework* FIWARE com modelo de dados NGSIv2, o qual promove a interoperabilidade entre aplicações, as comunicações são suportadas por WiFi ou GSM, meios que não suportam dispositivos de baixa potência, ou que requerem custos de operação. Por sua vez, [8] não disponibiliza um modelo de dados standard e interoperável, as medições observadas são dependentes das rotas dos autocarros usados e usam como meio de comunicação o 4G, com os custos de operação e energia associados.

Na proposta de solução apresentada neste artigo estas limitações foram tidas em conta, garantido uma solução final compatível com o FIWARE NGSIv2 e sem custos fixos na comunicação.

⁶ Ver crate.io/products/cratedb

⁷ Ver www.cloudino.io

⁸ Ver grafana.com

⁹ Ver green-route-manual

¹⁰ Ver www.libelium.com/iot-products/waspote

3 Implementação

Para a implementação da plataforma proposta, foi necessário definir a arquitetura da mesma, identificando quais os componentes necessários e como estão ligados entre si. Como um dos requisitos propostos é a implementação de uma plataforma de código aberto com base na *framework* FIWARE, foi avaliado o portfólio de componentes desta *framework*, identificando quais os componentes úteis na definição da arquitetura da plataforma e também identificando quais os componentes, que dada a sua especificidade, são necessários desenvolver.

3.1 Arquitetura

A arquitetura proposta é composta por múltiplos componentes, a maioria dos quais são componentes da *framework* FIWARE, tirando partido do conjunto rico e diversificado de componentes genéricos oferecido pela *framework*, que procuram solucionar diversos problemas relacionados com o desenvolvimento de soluções Smart. Os componentes implementados foram criados por necessidades particulares na arquitetura da plataforma, para os quais não foram identificados componentes desenvolvidos por entidades terceiras, que resolvessem tais necessidades. Cada componente usado na plataforma, é disponibilizado através de imagem de contentor de execução virtual, permitindo a execução de todos os componentes, tanto *on-premises*, como na nuvem. Com diversos componentes, cada um com a sua própria funcionalidade e contentorizado, é possível implementar uma solução descentralizada, que em caso de falha de um componente, não deixe de funcionar na sua totalidade, mas apenas parcialmente, mantendo uma disponibilidade parcial através dos componentes não afetados pela falha. Esta arquitetura com componentes individuais contentorizados permite também a substituição individual, sem um elevado esforço associado, da maioria dos componentes da plataforma, por outro componente de funcionalidade similar. A Figura 1 ilustra a arquitectura proposta para a solução, identificando a azul os componentes desenvolvidos por entidades terceiras e a cinza os componentes desenvolvidos especificamente para esta solução. Note-se que, na totalidade, a implementação desta arquitectura configura um *Generic Enabler FIWARE*. Os componentes da arquitectura são:

- **Orion (*Context Broker*)**¹¹: Componente responsável pela gestão dos dados de contexto actuais.
- **IoT Agent LoRaWAN**¹¹: Componente responsável por obter dados oriundos de dispositivos IoT numa rede LoRaWAN [9] e registar esses dados no *Orion Context Broker*.
- **MongoDB**¹²: SGBD NoSQL, permite armazenar dados de contexto, criando um registo histórico dos dados e armazenar dados de estado interno de outros componentes.

¹¹ Ver www.fiware.org

¹² Ver www.mongodb.com

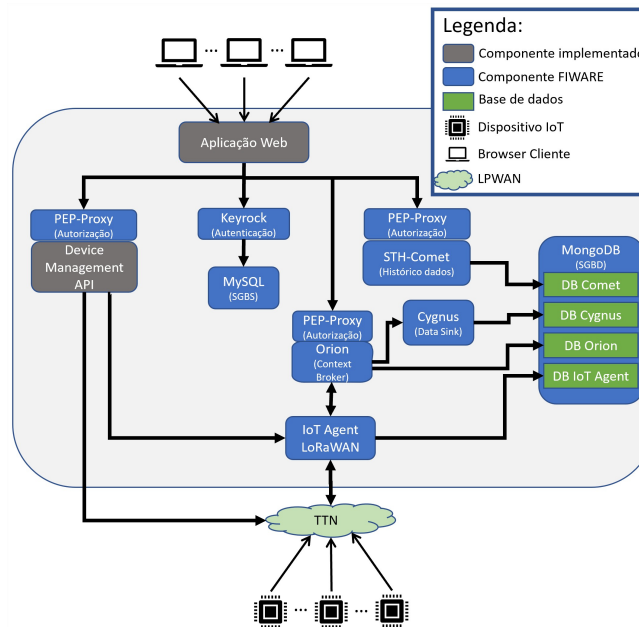


Figura 1. Arquitetura proposta para a plataforma.

- **Cygnus**¹¹: Conector responsável pela persistência do histórico dos dados de contexto do *Orion Context Broker*, em base de dados MongoDB.
- **STH-Comet**¹¹: Permite interrogar o histórico de contexto armazenado em base de dados. Disponibiliza algumas funções de agregação sobre os dados.
- **KeyRock**¹¹: Gestor de autenticação e autorização de acesso à plataforma.
- **MySQL**¹³: SGBD para base de dados do Keyrock.
- **Device Management API**: Permite o aprovisionamento de dispositivos IoT na plataforma.
- **Aplicação Web** Disponibiliza um *dashboard* com informação sobre os dispositivos e dados gerados por estes.
- **PEP-Proxy (Wilma)**¹¹: Assegura o control de acessos a outros componentes.

3.2 Componentes desenvolvidos

Device Management API¹⁴ Os dispositivos IoT (sensores), para serem aco-
plados à plataforma, têm como requisitos mínimos o envio das valores medidos
através de LoRaWAN. A comunicação é suportada pela TTN, sendo necessária
a existência de *gateways* LoRaWAN registados (na TTN) no raio de alcance dos
dispositivos. Para os gerir, foi implementado o componente Device Management

¹³ Ver www.mysql.com

¹⁴ Ver github.com/IvoPedroso/Management-API

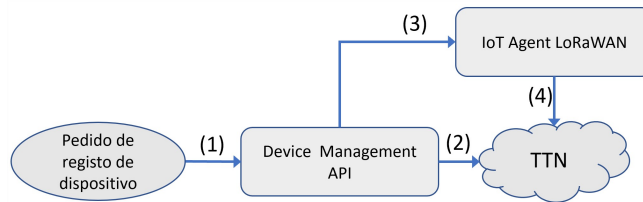


Figura 2. Processo de registo de um dispositivo na plataforma.

API. Este aceita pedidos de registo, remoção e listagem de dispositivos IoT na plataforma. Estes pedidos têm como origem a aplicação Web da plataforma. A implementação deste componente é feita na linguagem Javascript e corre sobre um serviço Node JS ¹⁵, acessível via HTTP. O componente, ao receber um pedido de registo para um dispositivo, efectua duas tarefas: (i) regista o dispositivo com os dados recebidos, na plataforma TTN, a qual permite o dispositivo IoT usar esta plataforma na transmissão das suas mensagens através do Protocolo LoRaWAN; (ii) através do IoT Agent LoRaWAN, regista-se automaticamente na fila MQTT da plataforma TTN, como subscritor das mensagens enviadas pelo dispositivo identificado no registo. A Figura 2 ilustra o processo de aprovisionamento de um dispositivo na plataforma, iniciado por um pedido de registo de um dispositivo:

- (1) O pedido de registo de dispositivo chega ao Device Management API, via HTTP, contendo os dados do dispositivo IoT a registar.
- (2) É feito um pedido à plataforma TTN, via API REST, registando o dispositivo na mesma. Este pedido contém os dados do dispositivo e as chaves que permitem a TTN autenticar o dispositivo na rede LoRaWAN. Estas chaves são chaves de negociação, que permitem definir chaves únicas de comunicação nas próximas mensagens entre o dispositivo e a TTN.
- (3) É feito um pedido HTTP ao IoT Agent LoRaWAN com os dados do dispositivo, ficando o dispositivo registado neste agente.
- (4) Ao receber um registo de um dispositivo, o IoT Agent LoRaWAN regista-se na fila MQTT da TTN, como subscritor de todas as mensagens provenientes do dispositivo. A TTN, ao receber uma mensagem do dispositivo, irá notificar os subscritores registados, na fila de mensagem associada ao dispositivo.

A remoção de dispositivo usa um processo similar ao processo de registo, mas realiza a dissociação do dispositivo tanto na TTN como no IoTAgent.

Dinâmica das mensagens transmitidas pelos dispositivos. Depois de estar aprovisionado, quando um dispositivo transmite uma mensagem, é necessário uma série de interações, nomeadamente entre a TTN e a plataforma, e também entre alguns componentes da própria plataforma, para que a informação transmitida pelo dispositivo fique corretamente armazenada na plataforma. A figura

¹⁵ Ver nodejs.org

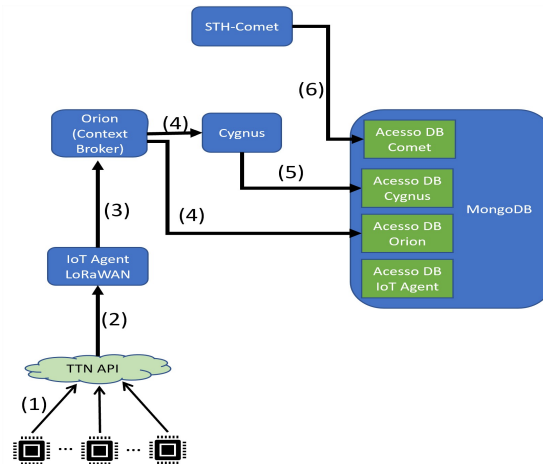


Figura 3. Dinâmica das mensagens transmitidas pela plataforma.

3 ilustra o percurso de uma mensagem enviada por um dispositivo, ao longo da TTN e da plataforma implementada. Os passos necessários para a transmissão das mensagens são:

- (1) O dispositivo envia um mensagem via LoRaWAN, que é recebida na plataforma TTN;
- (2) O IoT Agent LoRaWAN é notificado e recebe a nova mensagem, pois está registado com subscritor na fila MQTT (na TTN), associada ao dispositivo que efetuou a transmissão da mensagem;
- (3) O *Orion Context Broker* recebe a mensagem via IoT Agent LoRaWAN, pois todas as mensagens de todos os dispositivos são-lhe transmitidas por este, já no formato NGSIv2;
- (4) O *Orion Context Broker* regista os dados em base de dados e notifica o Cygnus com a nova mensagem. O Cygnus é subscritor de alterações de contexto no Orion, desde o momento da instanciação da plataforma;
- (5) O Cygnus regista a mensagem recebida em base de dados, pois tem como função registar a evolução dos dados do Orion;
- (6) O STH-Comet observa a evolução dos dados registados pelo Cygnus e gera agregações desses dados e armazena-as também em base de dados.

Após o registo na plataforma dos dados transmitidos pelos dispositivos, estes dados ficam disponíveis para consulta, nomeadamente os dados atuais de contexto, através do *Orion Context Broker*, e os dados históricos com base numa série temporal, através do STH-Comet.

Aplicação Web Foi implementada uma aplicação Web ¹⁶, para servir de interface com os utilizadores. Esta permite visualizar num mapa geográfico, a loca-

¹⁶ O código está disponível em github.com/IvoPedroso/DashboardWeb

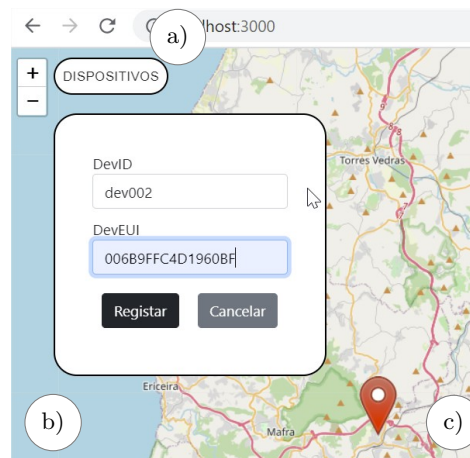


Figura 4. Interface de utilizador do protótipo implementado.

lização dos dispositivos registados. As coordenadas dos dispositivos são extraídas da informação registada na plataforma, com origem em mensagens enviadas pelos dispositivos. A Figura 4 ilustra a interface de utilizador da aplicação Web, a qual tem um mapa geográfico onde o utilizador pode movimentar a vista sobre o mapa (arrastando o mapa com o ponteiro do rato). O utilizador pode aceder à opções de adicionar e remover dispositivos na plataforma através do botão identificado na Figura 4.a). Por exemplo na Figura 4.b), é mostrado o formulário que permite registar um dispositivo na plataforma. Todos os dispositivos registados e que já tenham transmitido mensagens com informação de localização, são referenciados no mapa nas coordenadas transmitidas, com um marcador, como demonstrado em Figura 4.c). O utilizador, ao interagir com um marcador no mapa, espoleta uma janela *pop-up* com informação relativa aos dados enviados pelo dispositivo em questão, como indicado na Figura 5. Estes dados incluem a data da última comunicação e o último valor transmitido para cada métrica enviada, sendo esta informação obtida do componente *Orion Context Broker* (Figura 5.a)). Também é mostrada informação histórica, num calendário, relativo aos últimos 30 dias. Este calendário faz uso de uma escala de cores, para representar gamas de valores. Cada dia no calendário tem uma cor que representa o valor máximo obtido no dia para a métrica representada, sendo a gama de valores dependente do que está a ser medido pelo sensor (Figura 5.b)). Acima do calendário é possível alternar a métrica representada, através de um tabulador (Figura 5.c)). A informação de histórico é obtida do componente STH-Comet.

3.3 Instanciação da plataforma

Um dos requisitos definidos para a plataforma proposta neste artigo, é a existência de um processo de instanciação simples e com pouco esforço associado. Todos os

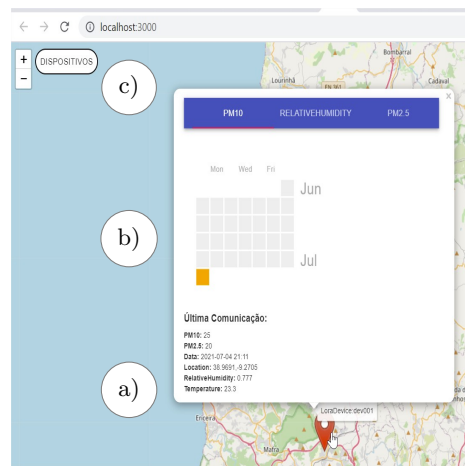


Figura 5. Informação histórica de cada dispositivo.

componentes usados são contentorizados usando Docker¹⁷: quer os componentes da *framework* FIWARE, quer os componentes desenvolvidos em específico para a plataforma. Assim, todas as imagens de contentores Docker usadas na plataforma estão disponíveis para livre utilização no portal DockerHub¹⁸. A disponibilização dos componentes através de imagens de contentores permite o *download* das imagens dos componentes no momento da instanciação dos mesmos, possibilitando também a execução de forma individual de cada componente, aumentando a flexibilidade da plataforma, nomeadamente a substituição de componentes, mesmo com a plataforma em execução. A parametrização dos diversos componentes, bem como a interligação entre estes, é realizada através da definição de variáveis de ambiente obtidas de um único ficheiro que define o valor das variáveis usadas, as quais são importadas para os contentores durante a instanciação dos mesmos.

Para mitigar a complexidade associada à instanciação dos diversos contentores e dos inúmeros parâmetros necessários, foram definidos dois *scripts* DockerCompose¹⁹. Não foi tudo definido num único *script*, porque durante a instanciação de alguns componentes da plataforma, nomeadamente na instanciação dos componentes PEP-Proxy, é necessária informação de autenticação, a qual é gerada pelo componente KeyRock, aquando do seu primeiro arranque. Tal obriga a instanciação da plataforma em dois passos distintos. Para execução destes dois *scripts*, bem como realizar operações intermédias e operações pós instanciação, nomeadamente o registo do Componente Cygnus como subscritor no componente *Orion Context Broker* (para obtenção de informação de contexto), foi definido um *Shell Script*, que realiza todos os passos necessário para instanciação de toda

¹⁷ Ver www.docker.com

¹⁸ Ver hub.docker.com

¹⁹ Ver docs.docker.com/compose

a plataforma. Assim, para instanciar toda a plataforma, basta a execução de um único comando ²⁰.

4 Discussão

A plataforma tem como características: (i) a facilidade de instanciação, pois através da execução de um único comando, os componentes que a compõe são descarregados da internet e a plataforma é instanciada como um todo; (ii) uso de representação normalizada da informação de contexto (NGSIv2); (iii) a flexibilidade na sua constituição, pois sendo uma plataforma descentralizada, composta por elementos em execução em contentores, é possível realizar a troca de componentes com o resto da plataforma em execução; e, (iv) a sua capacidade de evolução, com componentes adicionais, ou mesmo outros substitutos, alterando apenas os *scripts* de instanciação. O registo de dispositivos IoT na plataforma também é realizado sem necessidade de conhecer em detalhe a TTN e o LoRaWAN. Como ilustrado na Figura 4, apenas é necessário inserir a identificação do dispositivo. Os resultados das observações transmitidas pelos dispositivos podem ser analisadas através da interface do utilizador como ilustrado na Figura 5. Os últimos valores transmitidos são mostrados, bem como os valores máximos por dia das métricas transmitidas são mostrados num calendário com os últimos 30 dias, representado numa escala de cores. A cor associada para classificar os valores máximos de cada métrica, está definido na plataforma, mas está previsto a criação de um formulário na interface de utilizador, que permita a definição ou correção da escala apresentada para cada métrica.

5 Conclusão

A plataforma proposta, é uma plataforma descentralizada baseada na *framework* FIWARE. Toda a plataforma é instanciável através da execução de um único comando e os seus componentes executados em contentores. Faz uso da plataforma TTN como meio de comunicação com dispositivos IoT sobre uma rede LoRaWAN. Disponibiliza uma interface Web com os utilizadores, permitindo o registo de dispositivos IoT na plataforma, a visualização da localização de dispositivos registados num mapa, visualizar o valor das últimas métricas enviadas pelos dispositivos e visualizar os valores máximos de cada métrica, numa escala de cores, relativos aos últimos 30 dias. A implementação desta plataforma procurou disponibilizar à comunidade uma solução de código aberto que permita integrar diversos dispositivos IoT, que operam sobre uma rede móvel de baixa potência, cuja observações seja armazenadas e agregadas. Motivada pelo intuito de auxiliar a população e agentes com responsabilidade na gestão de centros urbanos, a analisar e avaliar os níveis de poluição existentes nos centros urbanos, parece-nos que esta arquitectura pode ser usada noutros domínios em cidades inteligentes e IoT.

²⁰ Consultar github.com/IvoPedroso/PlataformaFIWAREInstancia para mais detalhes sobre o aprovisionamento da solução.

Referências

1. Chowdhury, S., Dey, S.: Cause-specific premature death from ambient pm2.5 exposure in India: Estimate adjusted for baseline mortality. *Environment International* **91**, 283–290 (2016)
2. CM Lisboa: Rede de sensores monitoriza a qualidade do ar, níveis de ruído e trânsito. online: <https://www.lisboa.pt/atualidade/noticias/detalhe/rede-de-sensores-monitoriza-a-qualidade-do-ar-niveis-de-ruído-e-transito> (2021)
3. Fernández Maceira, A., et al.: Fiware-based application for control of smart cities (2020)
4. FIWARE Foundation: A c++ implementation of the ngsiv2 rest api. online: <https://fiware-orion.readthedocs.io/en/master/> (2021)
5. Fuertes, W., Carrera, D., Villacís, C., Toulkeridis, T., Galárraga, F., Torres, E., Aules, H.: Distributed system as internet of things for a new low-cost, air pollution wireless monitoring on real time. In: 2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT). pp. 58–67. IEEE (2015)
6. Hillar, G.C.: MQTT Essentials-A lightweight IoT protocol. Packt Publishing Ltd (2017)
7. Italia, T., I+D, T., NEC: Ngsiv2 api specification (v2.0). online: <https://fiware.github.io/specifications/ngsiv2/stable/> (2018)
8. Kaivonen, S., Ngai, E.C.H.: Real-time air pollution monitoring with sensors on city bus. *Digital Communications and Networks* **6**(1), 23–30 (2020). <https://doi.org/https://doi.org/10.1016/j.dcan.2019.03.003>
9. LoRa Alliance: Lorawan. online: <https://lora-alliance.org/about-lorawan/> (2021)
10. OMS: Ambient air pollution - a major threat to health and climate. online: <https://www.who.int/airpollution/ambient/en/> (2021)
11. OrchestraCities: Quantumleap: a service for storing, querying and retrieving spatial-temporal data. online: <https://quantumleap.readthedocs.io/en/latest/> (2021)
12. Richardson, L., Ruby, S.: RESTful web services. "O'Reilly Media, Inc." (2008)
13. Sornin, N., Luis, M., Eirich, T., Kramp, T., Hersent, O.: Lorawan specification. LoRa alliance (2015)
14. Taborda, R., Datia, N., Pato, M., Pires, J.M.: Exploring air quality using a multiple spatial resolution dashboard — a case study in lisbon. In: 2020 24th International Conference Information Visualisation (IV). pp. 140–145 (2020). <https://doi.org/10.1109/IV51561.2020.00032>
15. Xing, Y., Xu, Y., Shi, M., Lian, Y.: The impact of pm2.5 on the human respiratory system. *Journal of Thoracic Disease* **8**(1), E69–74 (2016)