INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS – *CAMPUS* INHUMAS
DEPARTAMENTO DE ÁREAS ACADÊMICAS
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RÚBEN FRANÇA XAVIER

DESENVOLVIMENTO DE UMA APLICAÇÃO DE MONITORAMENTO UTILIZANDO A PLATAFORMA IoT DOJOT

INHUMAS
2020

RÚBEN FRANÇA XAVIER

DESENVOLVIMENTO DE UMA APLICAÇÃO DE MONITORAMENTO UTILIZANDO A PLATAFORMA IoT DOJOT

Trabalho de Conclusão de Curso, apresentado como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação ao Departamento de Áreas Acadêmicas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás – *Campus* Inhumas.

Orientador: Rogério Sousa e Silva

INHUMAS

2020

Dados Internacionais de Catalogação na Publicação (CIP)

Xavier, Rúben França

X3 Desenvolvimento de uma aplicação de monitoramento utilizando a plataforma IoT Dojot [Manuscrito]. / Samuel Neves de Melo Silva. – – Inhumas: IFG, 2020.

45 f. : figs.

Bibliografia.

Orientador: Prof. Me. Rogério Sousa e Silva

Trabalho de conclusão de curso de graduação em Bacharelado em Sistemas de Informação – IFG/Câmpus Inhumas, 2019.

1. Plataforma de Middleware. 2. Internet das coisas. 3. Dojot. 4. Silva, Rogério Sousa e (orientador). I. Título.

CDD 006.6

Ficha catalográfica elaborada pela bibliotecária Maria Aparecida Rodrigues de Souza CRB/1-1497

Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Câmpus Inhumas – Biblioteca Atena

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO

NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

[ ] Tese	[ ] Artigo Científico
[ ] Dissertação	[ ] Capítulo de Livro
[ ] Monografia – Especialização	[ ] Livro
[X] TCC - Graduação	[ ] Trabalho Apresentado em Evento
[ ] Produto Técnico e Educacional - Tipo: _____	

Nome Completo do Autor: RÚBEN FRANÇA XAVIER
Matrícula: 20161030090080
Título do Trabalho: DESENVOLVIMENTO DE UMA APLICAÇÃO DE MONITORAMENTO UTILIZANDO A PLATAFORMA IoT DOJOT

Autorização

1. (X) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ___/___/_____ (Embargo);
3. ( ) Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

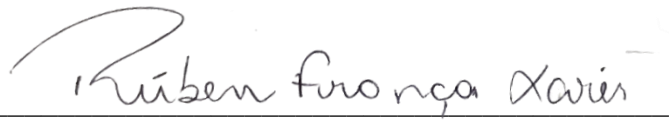
( ) O documento está sujeito a registro de patente.
( ) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
( ) Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

INHUMAS , 19/02/2020.



Assinatura do Autor e/ou Detentor dos Direitos Autorais

RÚBEN FRANÇA XAVIER

DESENVOLVIMENTO DE UMA APLICAÇÃO DE MONITORAMENTE UTILIZANDO A PLATAFORMA IoT DOJOT

Trabalho de Conclusão de Curso, apresentado como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação ao Departamento de Áreas Acadêmicas do Instituto Federal de Educação, Ciência e Tecnologia de Goiás – *Campus* Inhumas.

Aprovado(a) em 17 de fevereiro de 2020.

COMISSÃO EXAMINADORA

_____

Prof. Me. Rogério Sousa e Silva

Instituto Federal de Goiás – Campus Inhumas

Orientador

_____

Prof. Dr. Leandro Alexandre Freitas

Instituto Federal de Goiás – Campus Inhumas

_____

Prof. Me. Cleiton José da Silva

Instituto Federal de Goiás

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade de poder desfrutar de vida e saúde para estudar e batalhar por mais conhecimento. Agradeço aos meus pais, Valmir e Romilda, e também ao meu irmão, Ítalo, pelo suporte durante todo o meu período de estudo, por nunca terem me deixado desistir. Agradeço aos meus professores Leandro e Rogério por todo o apoio e incentivo na busca do conhecimento e por terem me orientado no desenvolvimento deste trabalho. Agradeço a professora Doralicy que também auxiliou no desenvolvimento textual deste trabalho. Agradeço a cada um dos meus amigos que me apoiaram nos momentos difíceis, sempre acreditando em mim e me dando forças para prosseguir. Por fim, porém não menos importante, quero agradecer a toda equipe do Núcleo de Estudos em Redes de computadores e Sistemas distribuídos (NumbERS) que me deram suporte no desenvolvimento do meu trabalho e também no meu desenvolvimento pessoal.

"*O insucesso é apenas uma oportunidade para recomeçar com mais inteligência.*"

Henry Ford

RESUMO

A Internet das Coisas (IoT) é um paradigma no qual objetos inteligentes participam ativamente no ambiente, colaborando com outros dispositivos físicos e virtuais. O alto grau de heterogeneidade de dispositivos, protocolos de comunicação e dados, provocaram a necessidade do desenvolvimento de plataformas de *middleware*. Estas plataformas surgem como uma solução para os problemas apresentados anteriormente, e também passam a agregar serviços disponíveis ao desenvolvedor e/ou usuário. Dessa forma, diversos grupos de pesquisa no contexto mundial projetaram e desenvolveram plataformas de *middleware* que atendam aos principais requisitos para cidades inteligentes. Com a redução da complexidade de desenvolvimento e a disponibilização de serviços, as plataformas passaram a servir não só para cidades inteligentes, mas também a aplicações em geral. Este trabalho tem por objetivo desenvolver uma aplicação de monitoramento de distância para avaliar as funcionalidades da plataforma IoT Dojot.

Palavras-chave: Plataforma de *Middleware*. Internet das Coisas. Dojot.

ABSTRACT

The Internet of Things (IoT) is a paradigm in which intelligent objects participate actively in the environment, collaborating with other physical and virtual devices. The high degree of heterogeneity of devices, communication protocols and data, caused the need to develop middleware platforms. These platforms appear as a solution to the problems previously presented, and also start to add services available to the developer and / or user. In this way, several research groups in the world context designed and developed middleware platforms that meet the main requirements for smart cities. With the reduction of development complexity and the availability of services, the platforms started to serve not only for smart cities, but also for applications in general. This work aims to develop a distance monitoring application to assess the functionality of the IoT Dojot platform.

Keywords: Middleware Platform. Internet of things. Dojot.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A computação ubíqua é um paradigma de interação entre o usuário e o computador que integra de forma invisível ambientes físicos com objetivo de ajudar pessoas na realização de tarefas cotidianas (WEISER, 1991 / WEISER, 1993). Um tipo característico da computação ubíqua são os espaços inteligentes (*smart spaces*). Esses são áreas físicas delimitadas, tais como, salas, blocos, casas, rodeados por serviços computacionais que controlam uma infraestrutura que possui um ou mais aspectos da computação ubíqua, como mobilidade e pervasividade (ROMÁN, et. al., 2002 / RORIZ, et. al. 2013).

A forma como é vista a informática e os sistemas de informação evoluiu com o tempo. Para tanto é possível imaginar e ver a computação saindo das estações de trabalho e atuando de forma pervasiva em nosso dia a dia. Dispositivos móveis e estacionários administram entre si formas de entregar ao usuário acesso a serviços com objetivo de ampliar as capacidades humanas. O conceito de ubiquidade está cada vez mais presente no nosso dia a dia graças, entre outros, a convergência, disseminação e popularização de tecnologias de rádio, associados a paradigmas da Internet das Coisas (Internet of Things - IoT) (ATZORI, 2010).

A Internet das Coisas faz parte de um conjunto de fatores determinantes para alavancar os espaços inteligentes (ABUARQOUB, 2017). Ela visa tornar dispositivos de uso diário autônomos, flexíveis e acessíveis a qualquer momento, de qualquer lugar, para qualquer usuário e em todo o mundo. Na IoT, os objetos inteligentes como salas de aulas, saberão o contexto para monitoramento do ambiente e sua interação. De forma simplificada, o conceito de IoT traz os objetos ou coisas inseridas, de forma difusa, ao nosso cotidiano cooperando entre si para atingir objetivos comuns (CAVALCANTE, 2017).

A computação ubíqua e a Internet das Coisas permitem expandir o conceito de espaços inteligentes. As cidades inteligentes são áreas geográficas bem definidas, em que Tecnologias da Informação e Comunicação (TIC), logística, produção de energia, entre outros, se juntem com a finalidade de oferecer ao cidadão bem-estar, inclusão, participação, desenvolvimento sustentável e inteligente (DAMERI, 2013). Além desses conceitos, é possível mensurar o quão inteligente uma cidade é a partir das dimensões descritas por Giffinger *et al*. (2007) que são: *Smart Economy*, *Smart People*, *Smart Governance*, *Smart Mobility*, *Smart Environment* e *Smart Living*. Nas cidades inteligentes, as principais tecnologias para elaboração da infraestrutura são: (1) Internet das Coisas para fornecimento de conexão entre dispositivos; (2) Big Data propiciando o armazenamento e processamento do grande volume de dados; e (3)

Computação em Nuvem fornecendo ambientes escaláveis e elásticos (KON, 2016). Em uma cidade inteligente é preciso considerar que dispositivos dos mais variados tipos, com tecnologias heterogêneas coexistem dentro do contexto. Esses dispositivos interagem entre si para executar uma infinidade de serviços. Logo, o produto poderá apresentar problemas como: disponibilidade, dependência de localização, segurança, mobilidade, escalabilidade e a tolerância ao erro (PIRO, 2014). A Figura 1 ilustra uma cidade inteligente com aplicações nos vários domínios existentes em uma cidade inteligente.

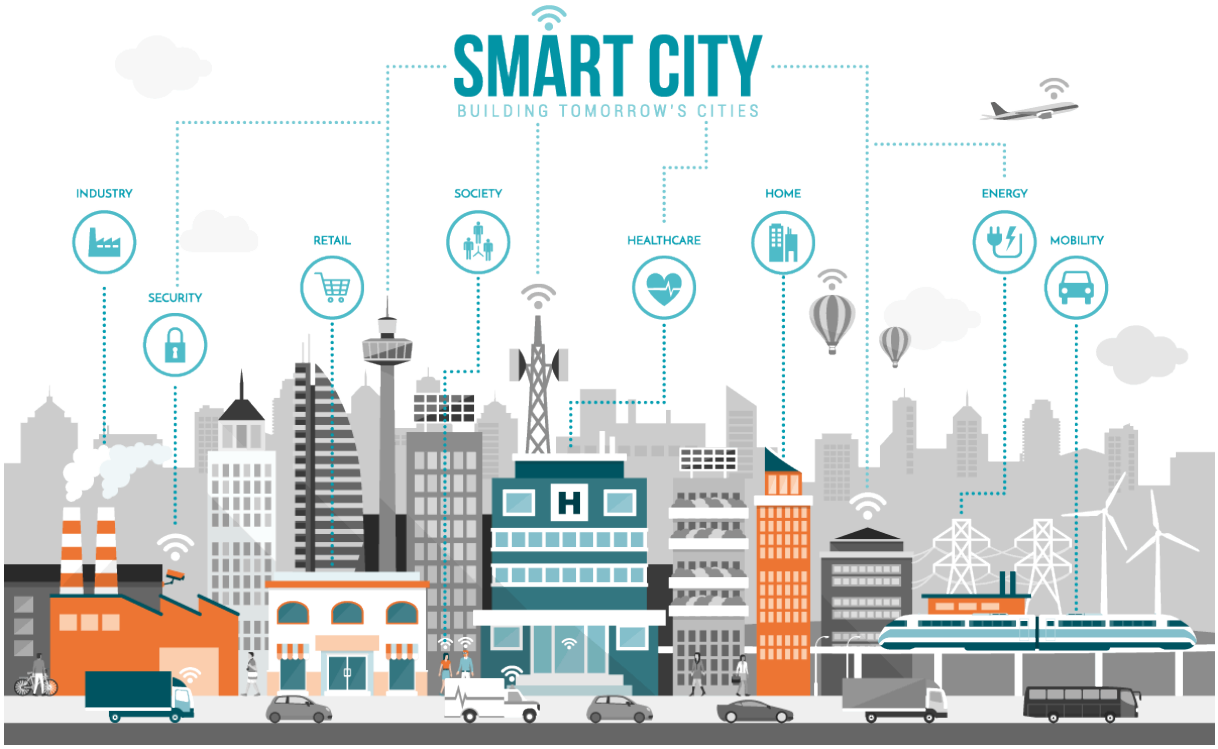


Figura 1 - Ilustração de uma cidade inteligente. Fonte: LABORATORY OF VISUAL COMMUNICATIONS, 2020

Os maiores desafios na construção de sistemas IoT está no alto grau de heterogeneidade de software e hardware nos ambientes (PIRES, 2015). Para isso, surge como solução em diversas pesquisas acadêmicas e também em ambientes reais, um conjunto de plataformas de middleware, tais como CiDAP, Sentilo, OpenIoT e SmartSantander (KON, 2016) (TEIXEIRA et al., 2011). Essas plataformas se apresentam com o objetivo de prover abstrações para as aplicações e também disponibilizar diversos serviços, criando camadas que funcionam em modo plug-n-play (BANDYOPADHYAY, 2011).

A plataforma de middleware Dojot, foi lançada pelo Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD). Originalmente a Dojot foi projetada com o intuito de facilitar o desenvolvimento de aplicações para cidades inteligentes (DOJOT, 2017), embora possa ser usada para fins diversos, como discutiremos no decorrer deste trabalho.

1.1 OBJETIVOS

O objetivo geral deste trabalho é desenvolver uma aplicação de monitoramento de distância para avaliar as funcionalidades da plataforma IoT Dojot.

Os objetivos específicos abaixo são desdobramento do objetivo geral:

- Realizar um levantamento bibliográfico das principais plataformas de middleware IoT;
- Estudar sobre os componentes da plataforma Dojot na implementados na forma de microsserviços;
- Projetar e implementar uma aplicação de monitoramento de distância utilizando a Dojot;
- Comparar as principais funcionalidades da Dojot com a plataforma IoT FIWARE.

1.2 METODOLOGIA

Inicialmente, realizamos um levantamento bibliográfico sobre um conjunto plataformas de *middleware* IoT apresentados no Capítulo 3. Utilizamos como ferramenta de busca o Google Scholar e, algumas das palavras chaves foram: plataforma IoT, plataforma de middleware, middleware IoT, e utilizamos algumas associações com os termos Smart City ou Smart Cities.

Logo após, estudamos os componentes da plataforma Dojot desenvolvidos na forma de microsserviços. Utilizamos como base a documentação disponível no site da plataforma e também realizamos alguns testes como a criação de *templates* e *devices*, como demonstraremos no Capítulo 2.

Com o conhecimento sobre as funcionalidades disponíveis na plataforma, projetamos e desenvolvemos uma aplicação com o objetivo de avaliar suas principais características. Após isso, discutimos e escolhemos a plataforma Arduino, como base para nosso hardware.

Em seguida, desenvolvemos a aplicação Arduino e construímos a estrutura para tomada de decisão utilizando a ferramenta *flow* da Dojot. Por fim, realizamos uma avaliação das plataformas de *middleware* Dojot e FIWARE com o intuito de detectar os principais pontos fortes e fracos de cada plataforma.

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado da seguinte forma: Capítulo 2 apresenta os trabalhos relacionados. Capítulo 3 apresenta uma visão geral da plataforma de *middleware* Dojot. Capítulo 4 demonstra a implementação de uma aplicação IoT. Capítulo 5 compara a plataforma Dojot à FIWARE. o Capítulo 6 traz a conclusão do trabalho. Capítulo 7 o referencial teórico.

2 TRABALHOS RELACIONADOS

As cidades inteligentes têm evoluído de forma constante e gerado cada vez mais dados de tipos variados. Os sistemas e dispositivos que integram arquiteturas de uma cidade inteligente podem atuar como produtores de dados e consumidores de dados. Um produtor de dados é qualquer tipo de recurso físico e/ou virtual que forneça dados para os consumidores de dados (OLIVEIRA, 2015). Como forma de integrar produtores e consumidores, algumas soluções estão centradas no desenvolvimento de plataformas de middleware para coleta, gerenciamento de dados e comunicação de dispositivos.

Neste capítulo, apresentaremos algumas propostas de plataformas de middleware que estão presentes na literatura. Algumas dessas estruturas já são utilizadas oficialmente em produção como a Sentilo em Barcelona, outras ainda estão em desenvolvimento como a FIWARE.

2.1 SENTILO

Sentilo é uma plataforma de *middleware* que teve seu desenvolvimento financiado pela Prefeitura Municipal de Barcelona. A plataforma que já está em uso, é parte do projeto da Prefeitura Municipal via Instituto Municipal de Informática (IMI), que iniciou-se em 2012, com o objetivo de posicionar globalmente a cidade de Barcelona como referência dentro das Cidades Inteligentes (SENTILO, 2013).

A plataforma é um dos componentes da arquitetura da cidade que tem como função isolar aplicações que explorem dados “gerados pela cidade” e os sensores. Sentilo foi projetada para atender “*qualquer cidade que procura abertura e fácil interoperabilidade*”. Com seu código aberto, traz o conceito de plataforma cruzada para facilitar o compartilhamento de informações entre sistemas variados e também integração de aplicativos legados (SENTILO, 2013). A Figura 2 ilustra a arquitetura da Sentilo.

A plataforma de middleware Sentilo traz um conjunto inicial de agentes, porém, a arquitetura de componentes é extensível tornando possível a ampliação das funcionalidades sem alteração no sistema principal. Além disso, um *front-end* para processamento de mensagens, console de administração, banco de dados de memória e banco de dados no-SQL são alguns dos componentes que são fornecidos na plataforma.

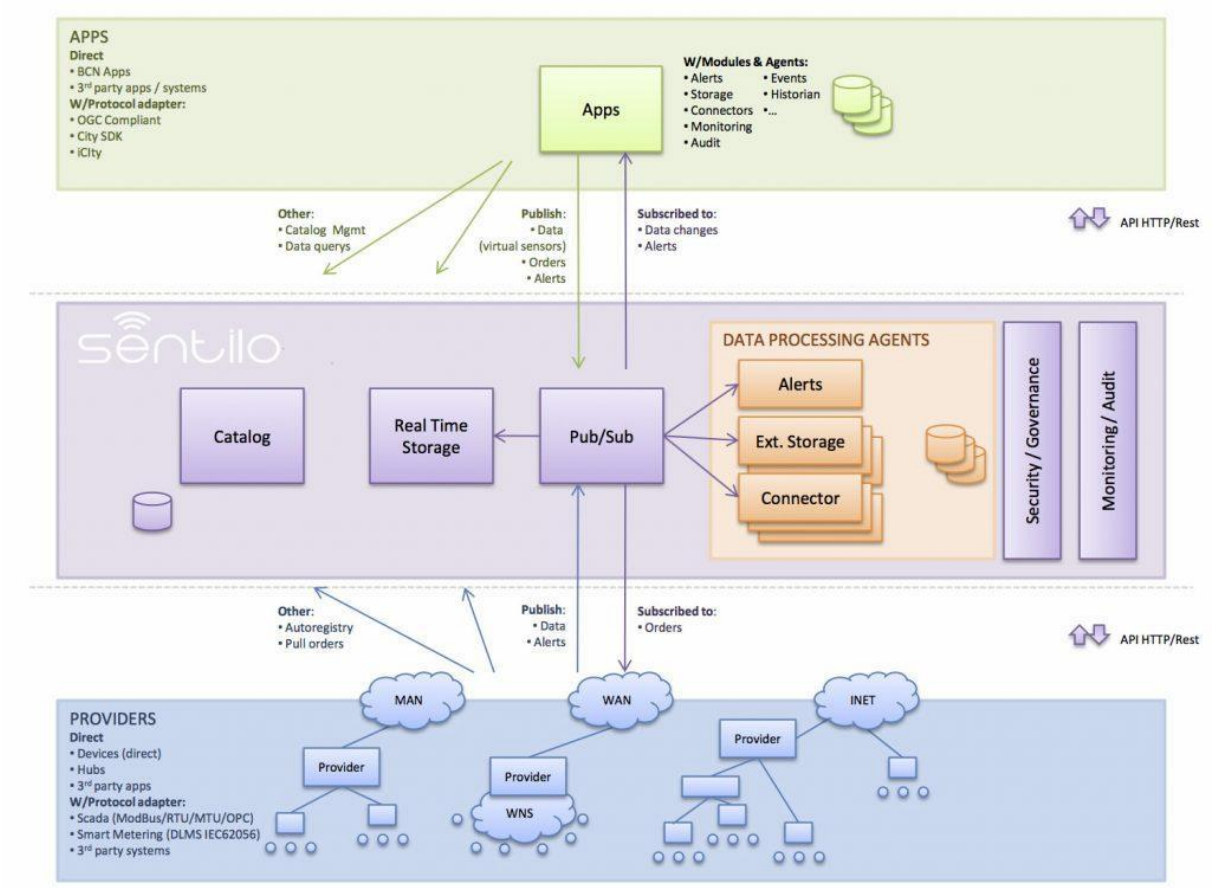


Figura 2 - Arquitetura funcionamento Sentilo. Fonte: sentilo.io/wordpress/.

2.2 FIWARE

A **FIWARE** é uma plataforma de middleware apresentada pela Comissão Europeia, fruto do projeto *Future Internet Public Private Partnership* (FI-PPP) lançado em 2011. A plataforma tem como missão "*construir um ecossistema aberto e sustentável em torno de padrões de plataforma de software públicos, isentos de royalties e orientados à implementação que facilitarão o desenvolvimento de novos aplicativos inteligentes em múltiplos setores*" e, trabalha com seus colaboradores para que busquem a materialização desta missão (FIWARE FOUNDATION, 2017)(FIWARE FOUNDATION, 2020a).

O objetivo da plataforma é oferecer um ambiente em que haja suporte para o desenvolvimento de aplicações de diversas áreas. Além de Cidades inteligentes, é possível também desenvolver pensando em logística, energias renováveis ou até onde sua imaginação e capacidade o levarem (TRINDADE et. al.). Os componentes da **FIWARE** são chamados de habilitadores genéricos (do inglês *Generic Enablers* - GE). Os GEs possuem APIs bem definidas que disponibilizam os serviços ao usuário. Entre os mais conhecidos e que não estão

em desenvolvimento podemos citar por exemplo o Orion Context Broker que é utilizado para gerenciar informações de contexto (TRINDADE *et al*. / SILVA, 2016 / SILVA, 2019).

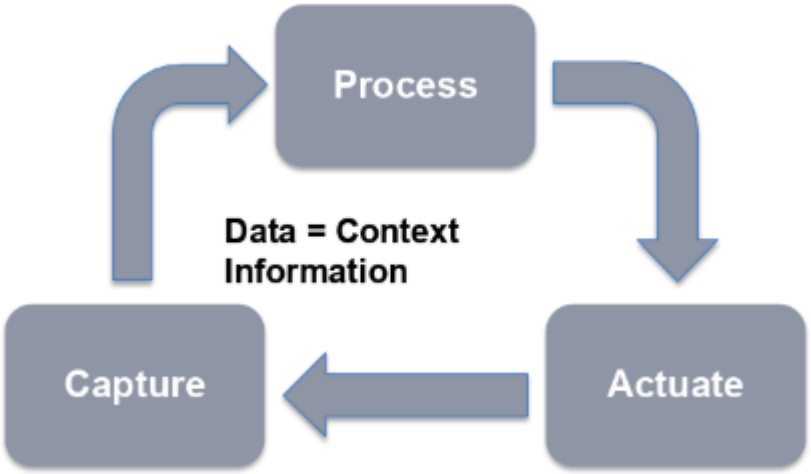


Figura 3 - FIWARE como plataforma para informação de contexto. Fonte: fiware.org

Na **FIWARE**, os dados são chamados de informação de contexto (do inglês *context information*). O componente principal da plataforma, o *context broker*, permite atualizações e acesso ao estado contexto naquele instante. Na Figura 3 observa-se o **FIWARE** como plataforma para informação de contexto. Em torno do FIWARE *context broker*, foram construídos outros componentes que complementam e lidam com interfaces IoT, gerenciamento de dados, processamento, análise e visualização de informações do contexto. Essa estrutura pode ser observada na Figura 4 (FIWARE FOUNDATION, 2020b).

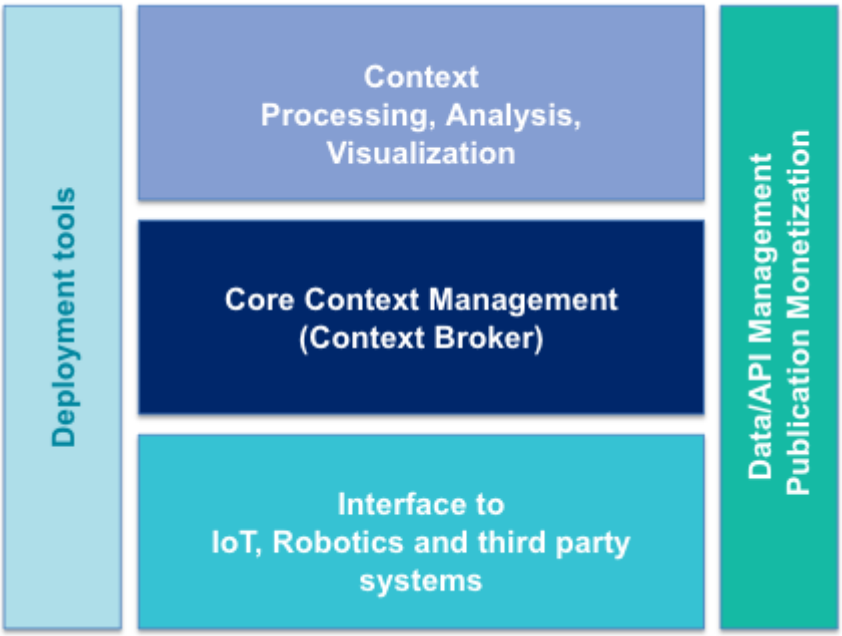


Figura 4 - Estrutura Fiware. Fonte: fiware.org

2.3 INTERSCITY

A InterSCity é um projeto que busca desenvolver pesquisas científicas e tecnológicas de forma multidisciplinar com o objetivo de enfrentar os desafios relacionados à infraestrutura das cidades inteligentes em relação a software. Um dos focos deste projeto é o desenvolvimento de tecnologias e métodos que possam ser reutilizáveis e tenham código aberto para apoiar futuras cidades inteligentes (DEL ESPOSTE, 2017).

O desenvolvimento da plataforma se deu de forma incremental, baseando-se em microsserviços de código aberto para permitir pesquisa, desenvolvimento e experimentos em cidades inteligentes. Seu objetivo é oferecer uma infraestrutura de *middleware* que seja modular, de alta qualidade, escalável e que consiga suportar as soluções de cidades inteligentes (DEL ESPOSTE, 2017).

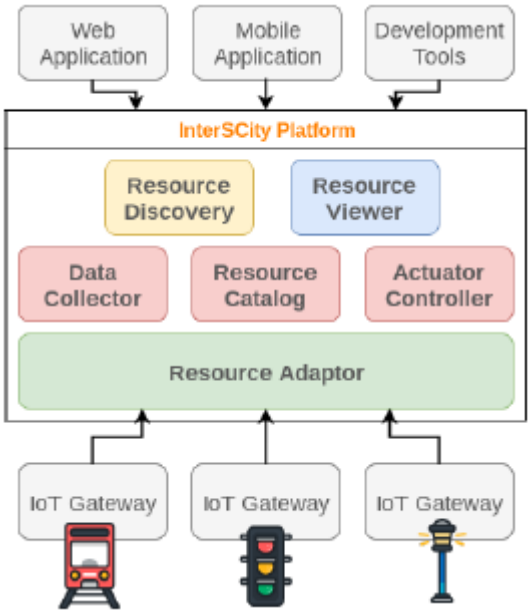


Figura 5 - Arquitetura InterSCity. Fonte: DEL ESPOSTE, 2017

A Figura 5 apresenta a arquitetura da plataforma InterSCity. Com 6 (seis) microsserviços, os limites de comunicação com dispositivos e aplicativos são bem definidos e oferecem recursos para a integração de dispositivos IoT a partir do *Resource Adaptor*, gerenciamento de dados e recursos a partir do *Resource Catalog*, *Data Collector* e *Actuator Controller*, descoberta de recursos a partir de dados de contexto no *Resource Discovery*, e visualização com o *Resource Viewer*. Além disso, a arquitetura desenvolvida permite que os recursos sejam gerenciados de forma descentralizada, ou seja, dividindo as funções entre os microsserviços (DEL ESPOSTE, 2017).

2.4 OPENIOT

O OpenIoT é uma plataforma desenvolvida com o objetivo de permitir o desenvolvimento de aplicações que se baseiam nos conceitos de Internet das Coisas (KON, 2016). No Projeto Vital (Pertrolo et al., 2014) a plataforma foi utilizada para implantar ambiente de Cidade Inteligente em cidades da Europa. A plataforma possui três planos: Físico, Virtualizado e de Utilidade e Aplicações (KON, 2016). A Figura 6 apresenta a estrutura da plataforma OpenIoT.

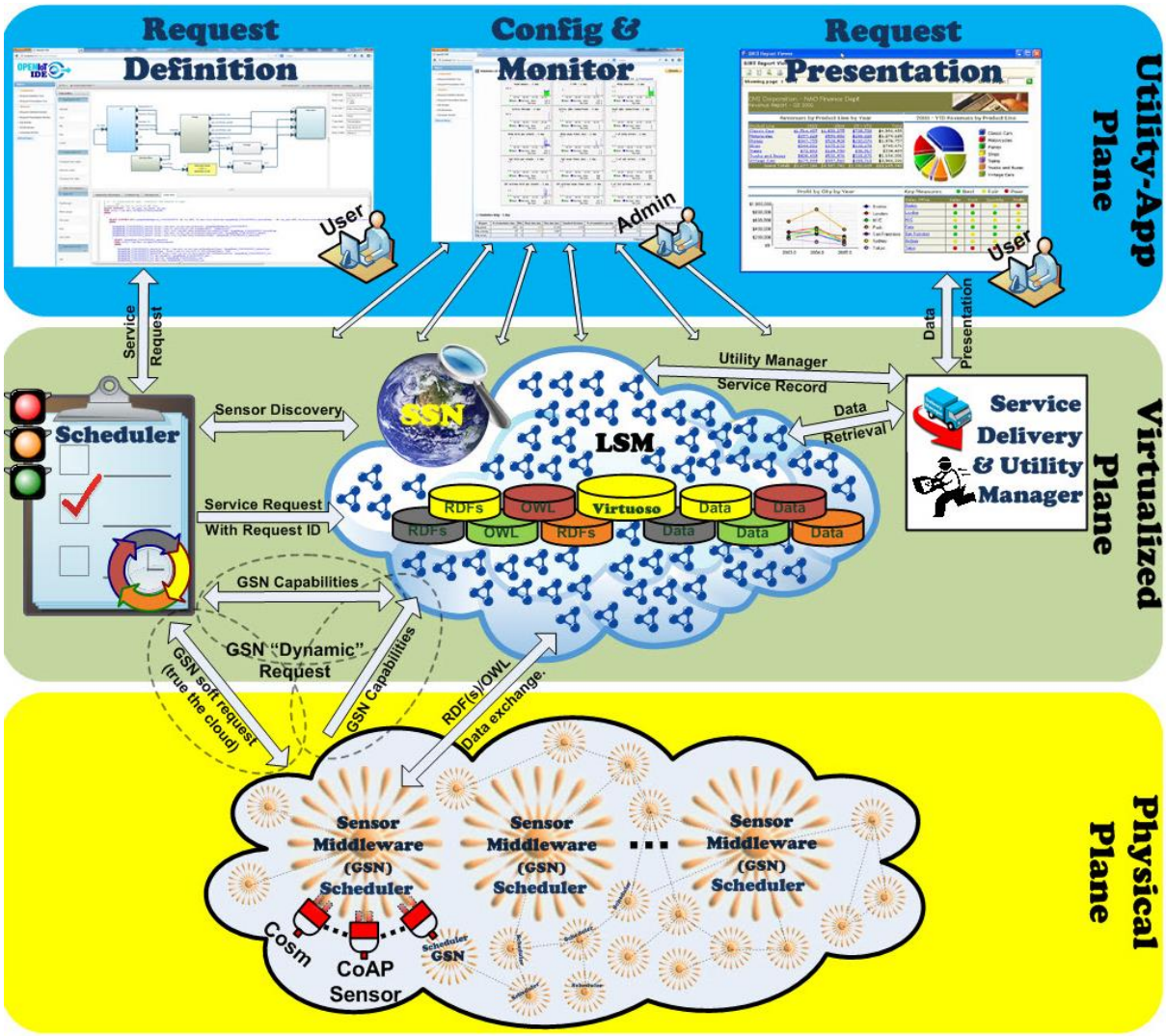


Figura 6 - Estrutura OpenIoT. Fonte: Pertrolo et al., 2014.

Cada plano possui sua função específica. O Plano Físico (do inglês *Physical Plane*) faz a interface entre o mundo físico e a plataforma, ou seja, gerencia, controla e monitora os dispositivos IoT e os dados gerados por eles. Com 3 componentes principais (*Scheduler*, *Cloud Data Storage*, *Service Delivery and Utility Manager*), a função do Plano Virtualizado (do inglês *Virtualized Plane*) é armazenar os dados, agendar e executar serviços. Por fim, o Plano de Utilidades e Aplicações (do inglês *Utility-App Plane*) faz a interface da plataforma com o usuário através de 3 componentes (KON, 2016):

1. **Request Definition**: permite a definição de novas aplicações utilizando serviços e dados da plataforma (KON, 2016);

2. **Request Presentation**: executa a aplicação feita no componente Request Definition e fazendo a comunicação com componentes do plano virtualizado para recuperação de dados (KON, 2016);

3. **Configuration and Monitoring**: permite configurar parâmetros como, por exemplo, intervalos de leitura feitos por sensores (KON, 2016).

2.5 SMARTSANTANDER

SmartSantander é uma plataforma que foi desenvolvida em um projeto experimental financiado pela Comissão Europeia (KON, 2016). A construção da plataforma seguiu uma arquitetura com 3 camadas: *IoT device* ou *IoT Node*, *IoT Gateway* e *Server* (SANCHEZ, 2014). Toda a arquitetura da plataforma pode ser observada na Figura 7.

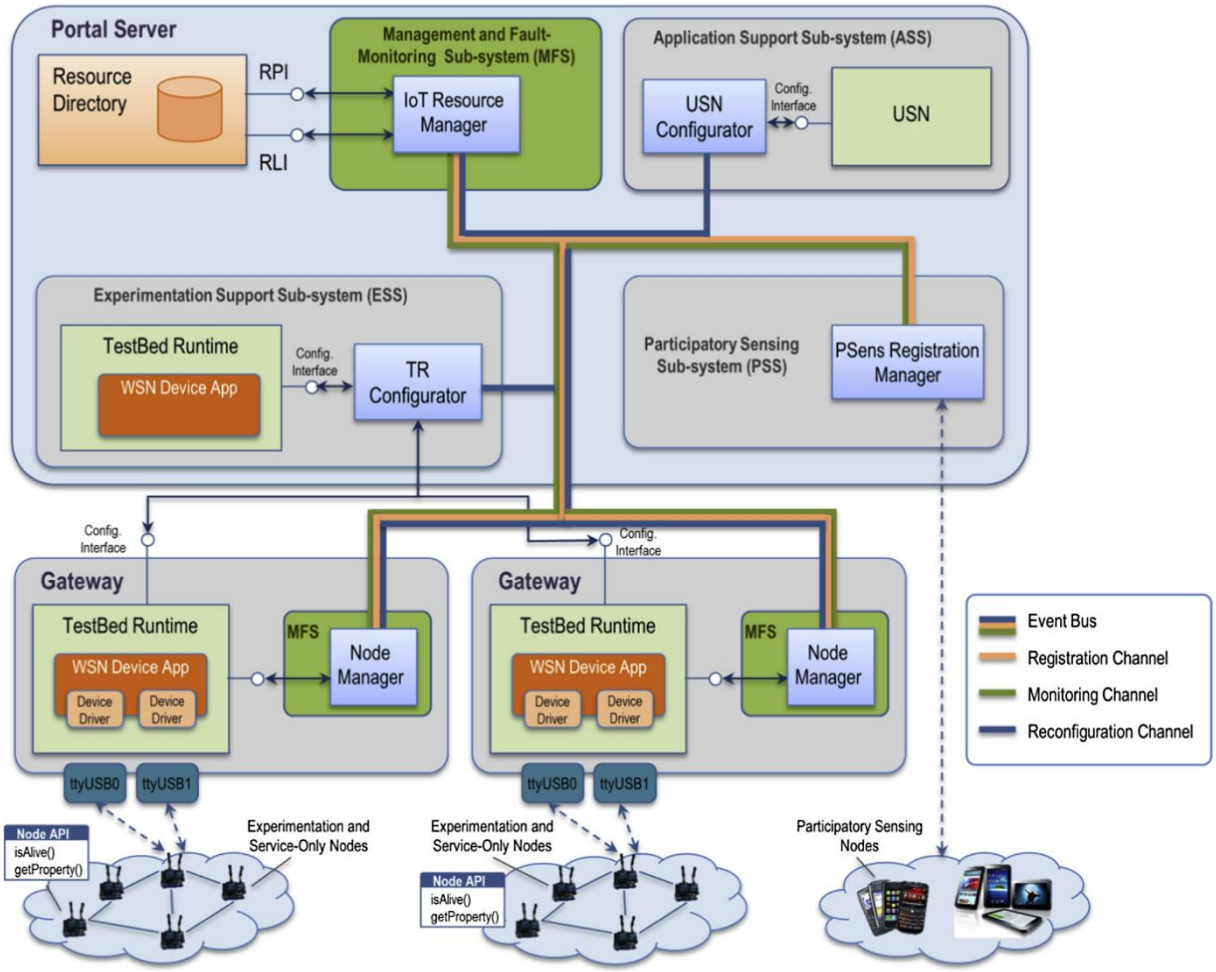


Figura 7 - Arquitetura da Plataforma SmartSantander. Fonte: Sanchez, 2014

O *IoT Node* fornece o que é necessário para os experimentos: dispositivos. Estes dispositivos geralmente são limitados em relação a recursos como energia, memória e processamento, ou seja, passam a operar apenas como sensores enviando esses dados ou possibilitando a atuação. Esses dispositivos também possuem meios para que qualquer tipo de

mal funcionamento seja identificado de forma rápida e eficaz. Este monitoramento é necessário por estar em área externa e sujeito ao clima e também a vandalismo (SANCHEZ, 2014).

A camada *IoT Gateway* é responsável pela comunicação dos dispositivos com os servidores. Esta comunicação leva os dados que serão processados e armazenados até os servidores. Ela também monitora os dispositivos conectados a cada gateway que faz a gestão e também podem reconfigurar esses dispositivos em tempo de execução (KON, 2016).

Por fim, a camada *Server* é onde os dados são processados e configurados. Hardware de alto poder de processamento são utilizados a fim de garantir escalabilidade e elasticidade da plataforma. Além do processamento de dados e armazenamento, também podem se tornar servidores de aplicações e serviços, entre outros (KON, 2016).

2.6 CIDAP

A plataforma CiDAP (*City Data and Analytics Platform*) utiliza ferramentas de *Big Data* para processar um grande volume de dados para adicionar inteligência e contexto nas aplicações e serviços que são desenvolvidos para a cidade. Um *middleware* independente faz a coleta dos dados pós processamento (KON, 2016 / CHENG, 2015). A Figura 8 apresenta os 5 principais componentes da plataforma.

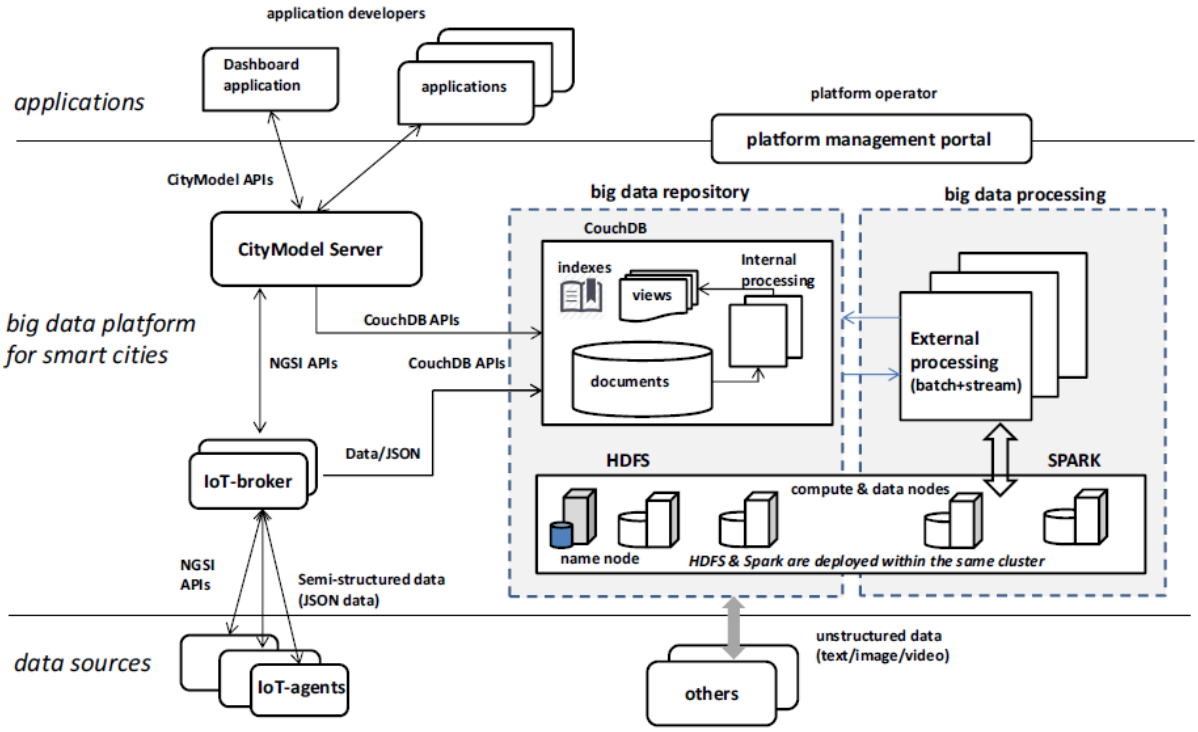


Figura 8 - Plataforma CiDAP. Fonte: CHENG, 2015

Os *IoT-Agents* fazem a conexão do middleware IoT como um gateway para poder buscar os dados dos dispositivos para armazenamento. As fontes de dados do middleware são mapeadas, cada uma, em um *IoT-Agent*. O *IoT-Broker* trabalha criando uma interface para facilitar o acesso dos dados nos *IoT-Agents* pelo *middleware*. O armazenamento de dados coletados e processados são armazenados pelo *Big Data Repository* utilizando componentes de Big Data. O *Big Data Processing* é o componente responsável por processamentos complexos e/ou demorados, processamento de dados históricos a partir de processamento em lote e também processamento de dados em tempo através de fluxos de dados. O *City Model Server* é implementado como uma API pois é esse componente o responsável por fazer a interface da plataforma com aplicações externas (KON, 2016 / CHENG, 2015).

2.7 CONCINNITY

A Plataforma Concinnity foi construída com o objetivo de facilitar o desenvolvimento de aplicações que utilizem dados providos a partir de sensoriamento, contribuição coletiva e a disponibilização de dados e serviços (KON, 2016). A plataforma implementa interfaces de acesso a dados em um ambiente de computação em nuvem além de uma ferramenta que possibilita a criação de *workFlows* (KON, 2016 / Wu et al., 2014). A Figura 9 apresenta as camadas e os componentes que fazem parte da arquitetura da aplicação.

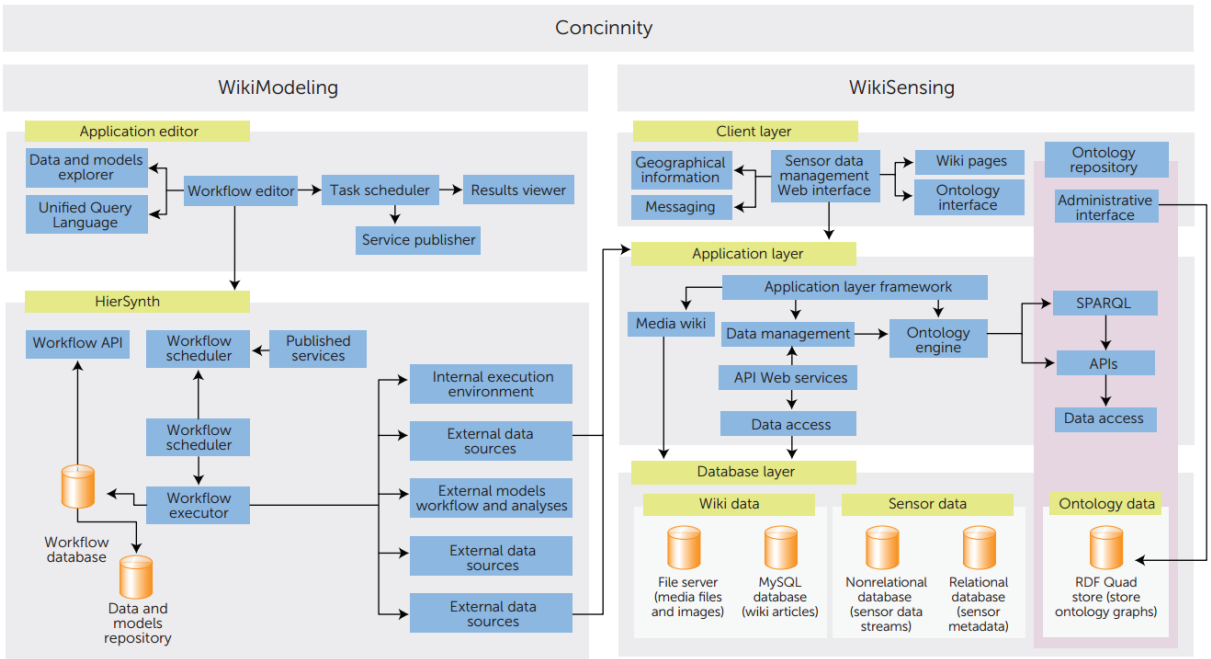


Figura 9 - Arquitetura da plataforma Concinnity. Fonte: Wu et. al., 2014

A arquitetura da plataforma é separada por camadas, cada uma exercendo uma função específica no conjunto. No total são 5 camadas:

- **Application Editor** é uma ferramenta online para construção de aplicações utilizando dados de sensores. Ela possui linguagem de consulta que permite recuperar e filtrar dados, além disso, a ferramenta também permite a criação de *workFlows*, agendar tarefas e um publicador de serviços (KON, 2016 / Wu et al., 2014).
- **HierSynth** é um servidor disponível em nuvem que recebe aquilo que foi desenvolvido no *Application Editor* e executa os serviços necessários para que a aplicação funcione corretamente (KON, 2016).
- **Data Base Layer** é o repositório de dados na plataforma onde são armazenados dados de sensoriamento, aplicações e também da própria plataforma. A plataforma utiliza banco de dados NoSQL MongoDB para facilitar a agregação dos dados, e também relacional MySQL para armazenar dados das aplicações e também as configurações da plataforma (KON, 2016).
- **Application Layer** é onde está inserida toda a lógica da aplicação que gerencia a plataforma. Ela é usada para controlar a execução dos serviços, gestão dos dados e para permitir a colaboração interna. Além disso, ela é responsável pela comunicação com os serviços que fazem a execução de *workflows* e também para acesso aos dados (KON, 2016).
- **Client Layer** implementa uma interface WEB que permite a visualização dos dados de sensores, catálogo de aplicações e serviços além de uma aplicação que permite a contribuição e compartilhamento de conhecimento dentro da plataforma baseada em Wiki (KON, 2016).

2.8 CLOUT

Com o objetivo de conscientizar seus cidadãos sobre os recursos da cidade e para ajudar a usar e cuidar desses recursos por meio de serviços de IoT, o projeto ClouT surge por meio do 7º *Framework programme* da Comissão Europeia e o *National Institute of Information and Communications Technology* do Japão (TEI, 2014). Com uma abordagem centrada no usuário, traz em seu conceito o aproveitamento da computação em nuvem como um facilitador de conexão entre a Internet das Coisas e a Internet das Pessoas por meio da Internet dos Serviços. Trabalha também para desenvolver uma plataforma de alta eficiência que utilize de

todas as fontes de informações possíveis para tornar cidades mais inteligentes e dar meios que enfrentam desafios emergentes (KON, 2016 / TEI, 2014). A Figura 10 apresenta a estrutura da plataforma.

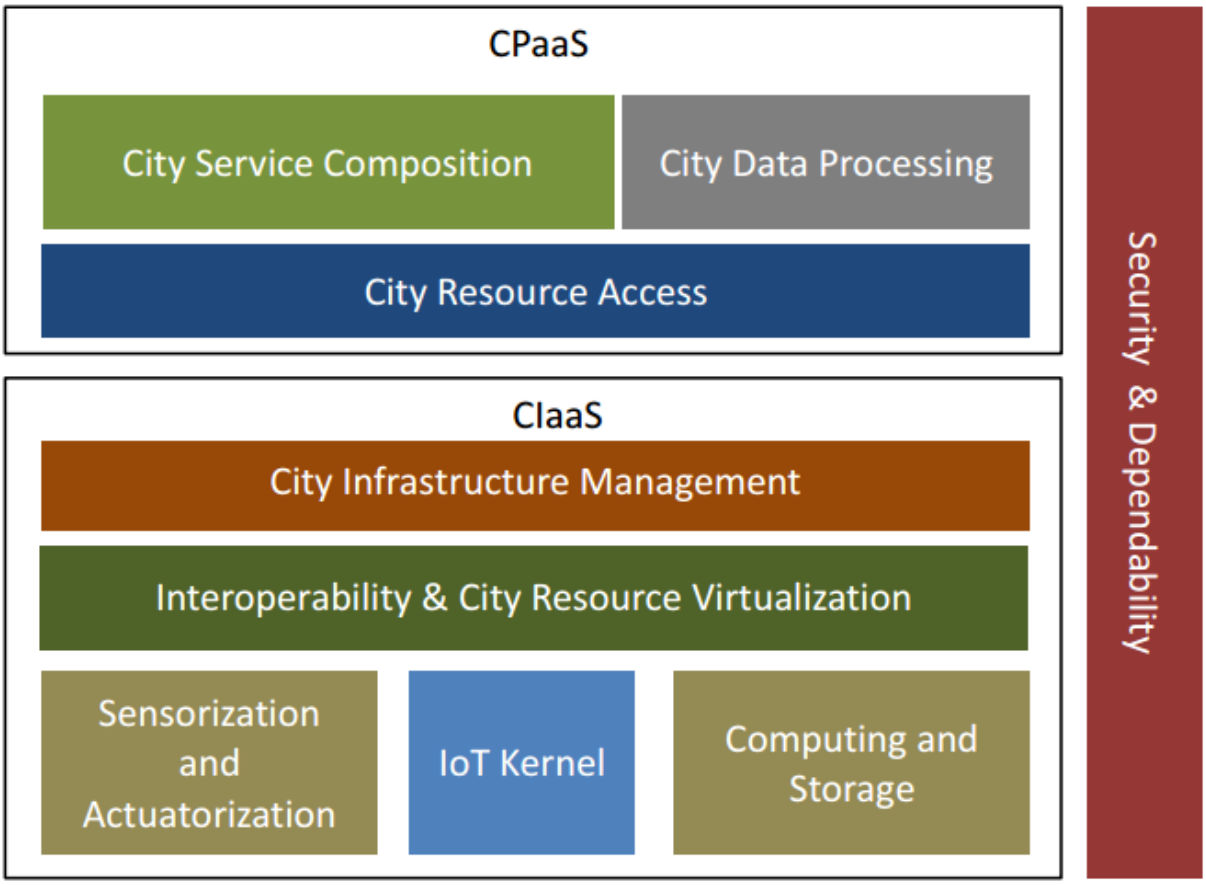


Figura 10 - Estrutura ClouT. Fonte: KON, 2016.

A plataforma possui duas camadas que disponibilizam seus serviços através de computação em nuvem: CpaaS (*City Platform as a Service*) e CIaaS (*City Infrastructures as a Service*). CpaaS oferece um conjunto de serviços que permitem o desenvolvimento de aplicações para cidades através da utilização do conjunto de componentes presentes na estrutura dessa camada. A ClaaS atua na gerência dos recursos físicos, garantindo a comunicação de diferentes tipos de sistemas/dispositivos. Os recursos dessa camada são acessados a partir de APIs, que permitem o acesso a recursos físicos disponíveis (KON, 2016).

3 DOJOT

A Dojot é uma plataforma de *middleware* aberta que foi lançada pelo Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD) buscando facilitar o desenvolvimento de aplicações para cidades inteligentes, com foco nos pilares de segurança pública, mobilidade urbana e saúde (DOJOT, 2017). A plataforma brasileira, tem em sua base a FIWARE, que foi desenvolvida por um consórcio *Fiware Foundation* no continente europeu e seus conceitos, inseridos no projeto brasileiro. A FIWARE é fruto do programa *Future Internet Public and Private Partnership* (FI-PPP) da Comissão Europeia (EC, de *European Commission*). Com a plataforma, um programa de aceleramento foi criado, e foram investidos 80 milhões de euros para promover sua adoção (FAZIO et. al., 2015).

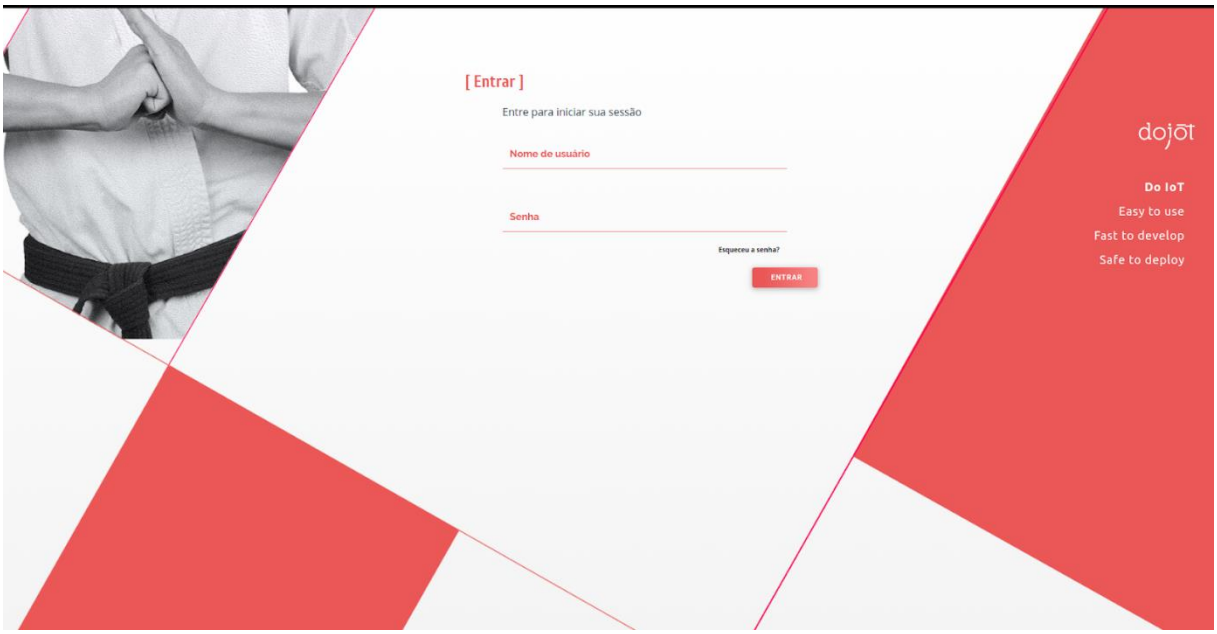


Figura 11 - Tela de Login Dojot.

A Dojot foi desenvolvida no projeto “Plataforma Aberta para IoT e suas Aplicações”, que contou com o apoio da FUNTTEL, MCTIC e Finep. O Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD) é o principal executor deste projeto e conta com outras instituições que apoiam e participam de seu progresso (DOJOT, 2017). Com uma proposta *open source*, a ideia é que seja facilitado o desenvolvimento de soluções e do ecossistema IoT, partindo das necessidades locais de cada projeto e, desta forma, contribuindo para o aperfeiçoamento e evolução constante da plataforma (PIMENTA, 2017).

Com uma estrutura de microsserviços, a Dojot oferta a possibilidade de criar diversas

aplicações sob demanda. Sua organização integra ainda outros recursos, como um painel de controle com interface gráfica, conhecida como *dashboard*, uma *Application Programing Interface* (API) aberta para acesso aos serviços e, ainda oferece proteção a seu usuário com segurança, bem como o gerenciamento de usuários utilizando tokens JWT, autenticação dos dispositivos (PIMENTA, 2017). O ambiente para cada usuário é separado, ou seja, um usuário está sempre dentro do seu próprio contexto sem compartilhamento de dados (DOJOT, 2019). A Figura 11 apresenta uma visão da tela de login.

Um dos diferenciais da Dojot é o recurso que possibilita a criação do Produto Mínimo Viável (do inglês *Minimum Viable Product* - MVP), que permite ao desenvolvedor checar e validar conceitos antes do início da implementação do produto final. Isso é possível graças a interface gráfica na qual, através do painel de controle, cria-se dispositivos físicos e virtuais gerenciam fluxos e aplicações (PIMENTA, 2017).

3.1 ARQUITETURA DA DOJOT

A Dojot possui uma arquitetura completa, com serviços variados que entregam a melhor experiência possível de desenvolvimento ao usuário. A ideia é que a plataforma seja de fácil utilização, escalável e robusta (DOJOT, 2019).

Para tanto a implementação de diversos Agentes IoT com diferentes protocolos de comunicação, fica a cargo do desenvolvedor adotar a tecnologia que melhor se adapta ao seu produto/conceito. A associação entre Apache Kafka e DataBroker os dados são isolados dentro do contexto. A utilização do Kong API Gateway centraliza o acesso aos dados e controla o tráfego. A Dojot permite que os dados recebidos por ela, sejam armazenados, para isso, o History torna-se um guia para dados e eventos que devem ser persistidos em um banco de dados. Tais informações podem ser armazenadas internamente em banco não-relacional MongoDB, e também externamente com a configuração do Logstash (DOJOT, 2019).

Além dos componentes citados acima, a Dojot possui diversos outros componentes que constroem a sua arquitetura. A subseção seguinte descreve cada um desses componentes.

3.1.1 COMPONENTES INTERNOS DA PLATAFORMA

A Dojot possui vários componentes e interfaces que formam sua arquitetura. Esses elementos são realizados a partir de microsserviços, o que contribui para sua execução distribuída, por meio de diferentes hosts ou máquinas virtuais. As subseções a seguir descrevem individualmente cada um dos componentes.

3.1.1.1 KAFKA

O Apache Kafka é uma plataforma de streaming distribuída. Portanto oferece recursos como a publicação e assinatura em fluxos de registros, armazenamento de fluxos de registros a longo prazo e com tolerância a falhas e também o processamento dos fluxos sob demanda (KAFKA, 2017). Na Dojot os fluxos são chamados de tópicos, e esta estrutura é utilizada para que os dados sejam isolados de usuários e aplicações de forma que garanta uma estrutura multi-tenant (DOJOT, 2019).

3.1.1.2 DATA BROKER

Desenvolvido pelo CPqD, o DataBroker exerce duas funções específicas na Dojot: 1) Gerenciar os tópicos do Kafka usados na distribuição de informações, controlando a criação dos tópicos em tempo de execução e restringindo os eventos aos quais um serviço deve ser exposto e é autorizado processar; 2) gerenciar os dados em tempo real, direcionando-os ao seu respectivo dispositivo para ser apresenta na GUI. (DATABROKER).

3.1.1.3 DEVICE MANAGER

O Device Manager é responsável por gerenciar dispositivos e *templates*. Além disso, também faz a publicação de atualizações para os componentes interessados utilizando o Kafka (DOJOT, 2019).

3.1.1.4 IoT AGENT

O IoT Agent é o elo entre os dispositivos físicos e componentes principais da Dojot, permitindo a comunicação através de protocolos implementados pelos agentes. Portanto garantir que a comunicação entre os dispositivos seja segura também é função deste componente (DOJOT, 2019).

3.1.1.5 SERVIÇO DE AUTORIZAÇÃO DE USUÁRIOS

Serviço que trabalha diretamente com controle de acesso e gerenciamento de perfis dos usuários. Todas as requisições de API feitas através do API Gateway são autenticadas por este serviço. O não uso de cache, não armazenamento de estados e a escalabilidade horizontal torna o serviço capaz de atender um grande volume de chamadas (DOJOT, 2019).

3.1.1.6 FLOWBROKER

O Flowbroker permite e provê meios para que o usuário contribua com fluxos de processamento de dados para execução das ações. Além dos blocos internos, é permitido a utilização de processamento externo utilizando APIs REST (DOJOT, 2019).

3.1.1.7 DATA MANAGER

Este serviço implementa configurações de gerência, importação e exportação de dados (DOJOT,2019). Os elementos importados e exportados para a plataforma trazem configurações de dispositivos, sejam sensores ou atuadores.

3.1.1.8 CRON

Apesar de ainda estar em fase experimental, o Cron possibilitará agendar eventos que serão lançados para outros microsserviços (DOJOT,2019). Neste sentido, a equipe do CPqD continua trabalhando no desenvolvimento deste componente.

3.1.1.9 HISTORY

O History tem por objetivo gerenciar dados e eventos que podem ser persistidos no banco de dados. Os dados que são armazenados, antes, são colocados em uma estrutura de específica para depois serem enviados ao banco. O armazenamento interno é feito em banco não-relacional, mas também pode ser usado armazenamento externo (DOJOT,2019).

3.1.1.10 SERVIÇO DE REGISTRO E AUTORIA

Assim como o Cron, este serviço ainda está em desenvolvimento, contudo, pretende gerar métricas de uso de recursos. Dessa forma, aplicações que possuem sistemas baseado no consumo de recursos e cotas podem ser controlados (DOJOT, 2019).

3.1.1.11 KONG API GATEWAY

O Kong API Gateway é o serviço que faz a conexão entre aplicações e serviços externos e os serviços internos da Dojot. Isso cria, por exemplo, um único ponto de acesso, o que facilita o uso da plataforma (DOJOT, 2019).

3.1.1.12 GUI

Um dos componentes mais importantes, a Interface Gráfica de Usuário (GUI) é provida através de uma aplicação WEB responsiva. Com ela é possível gerenciar perfis de usuários, e limitá-los a acessar APIs específicas. Também é possível administrar usuários, modelos de dispositivos, dispositivos, fluxos de processamento com ações de criação, visualização, edição e exclusão, além de visualizar as notificações do sistema em tempo real. Sendo assim, é possível ter controle de todas as funcionalidades que a Dojot disponibiliza via GUI (DOJOT, 2019). Uma parte da interface gráfica da Dojot pode ser observada na Figura 12, que mostra o dashboard com os dispositivos.

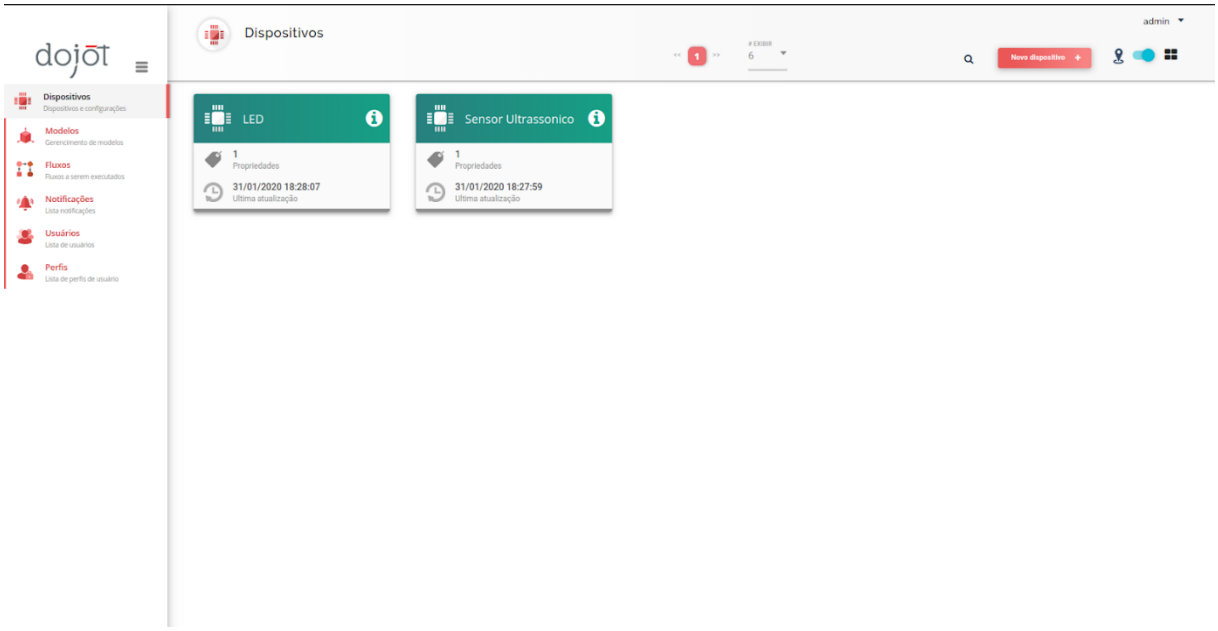


Figura 12 - Dashboard Dojot.

3.1.1.13 IMAGE MANAGER

Este componente está relacionado com todas as operações e imagens de firmware dos dispositivos (DOJOT, 2019).

4 AMBIENTE DE DESENVOLVIMENTO

Para avaliar a plataforma Dojot, projetamos uma aplicação que fez uso de suas principais funcionalidades. Para isso, desenvolvemos uma aplicação que identificasse uma situação de alerta baseada na distância entre objetos. Com o desenvolvimento desta aplicação, conseguimos fazer uso da plataforma para monitoramento de variáveis do ambiente e atuação.

Esse capítulo está organizado da seguinte forma: a Seção 4.1 apresenta todos os equipamentos utilizados. Na Seção 4.2, temos uma visão de como foi a implementação e, por fim, a Seção 4.3 discute os resultados.

4.1 EQUIPAMENTOS

Durante as pesquisas, optamos por escolher ferramentas que atendessem a necessidade da aplicação e com uma boa relação custo-benefício. Conseguimos chegar à plataforma Arduino que é amplamente utilizada em projetos do tipo *Maker*, que oferece todos os requisitos necessários para nossa aplicação desde microcontroladores, sensoriamento e atuação.

4.1.1 MICROCONTROLADORES

Microcontroladores são unidades de processamento que possuem uma unidade central de processamento, memória de dados, memória de programa, comunicação e conversores analógico-digitais entre outros (STEVAN,2015). Isso permite a interação com o ambiente, através de sensores e, a possibilidade de controle, por meio de atuadores.

Neste trabalho, utilizamos o microcontrolador NodeMCU que é um microcontrolador que combina o chip ESP8266, criado pela empresa chinesa *Espressif*, a interface usb-serial e um regulador de tensão 3,3V (LIMA, s. D.) e, em alguns casos, até 5V. Esse microcontrolador foi escolhido por oferecer uma boa relação custo-benefício, ou seja, o custo para adquirir é baixo e oferece ao desenvolvedor uma facilidade para o desenvolvimento e poder de processamento suficiente para as aplicações com baixa complexidade. A Figura 13 ilustra o NodeMCU.

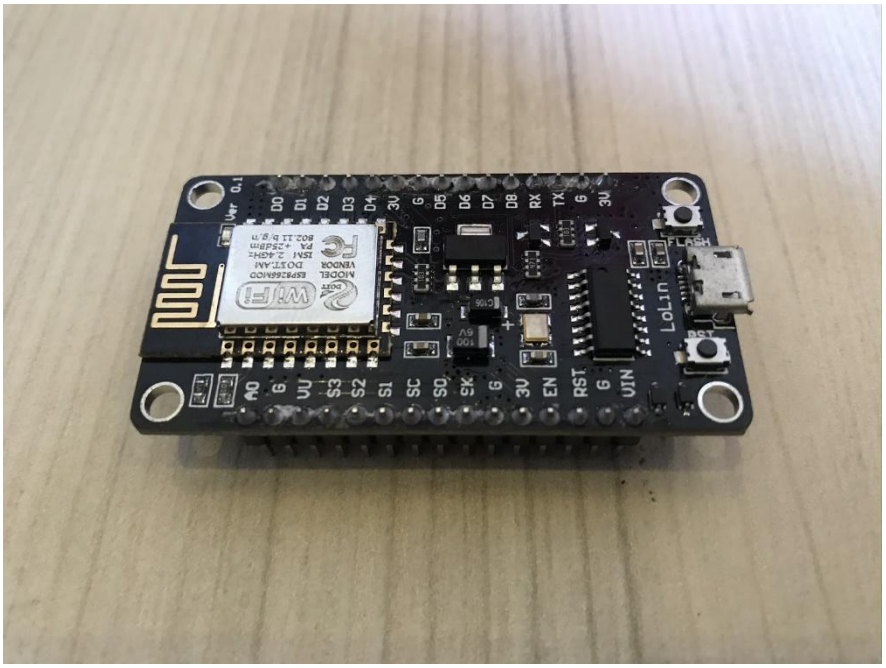


Figura 13 - Um dos NodeMCUs utilizados na aplicação.

4.1.2 SENSORES

Sensores são componentes eletrônicos que permitem a interação entre mundo real e equipamentos eletrônicos como geladeiras, forno microondas, ar-condicionado (BANZI, 2011). Sensores também são definidos por Marcelo Marchesan (2012, p. 19) como “dispositivos que mudam de estado conforme interação com ambiente”.

Microcontroladores são computadores simples, e processam apenas sinais elétricos. Para tornar possível a captação de qualquer variável de ambiente e/ou dados físicos, é necessário que haja algo para converter esses sinais em eletricidade e os sensores são exemplos disso (BANZI, 2011).

Na aplicação projetada e desenvolvida, escolhemos o sensor ultrassônico HC-SR04. Com ele, medimos a distância entre o sensor e demais objetos.

4.1.3 ATUADORES

Atuador é um dispositivo físico ou virtual responsável por realizar atividades ou tarefas computacionais. Na aplicação, sempre que um objeto se aproximar do sensor ultrassônico a uma distância previamente definida, um alerta será emitido. Para isso, escolhemos emitir um alerta visual, utilizando um LED simples, de cor vermelha.

Para controlar esse LED, um NodeMCU foi utilizado como atuador. Nessa situação, o *node* atua sobre o led, enviando sinal *on*/*off* baseado na informação que recebe, se enquadrando na condição de atuador.

4.1.4 COMPUTADORES

Utilizamos três computadores para realizarmos os testes que serão descritos na tabela 1.

Tabela 1 - Descrição dos computadores utilizados.

Origem	**Tipo**	**Sistema Operacional**	**Configuração**
Autor	Desktop	Ubuntu 18.04 LTS	Processador Intel i5 7400, 8GB, SSD 120 GB, HD 1Tb.
Autor	Notebook	Windows 10 Home Single Language	Processador Intel i5 3210M, 8GB, SSD 240Gb.
Núcleo NumbERS	Desktop	Ubuntu 18.04 LTS	Processador AMD A10 5800b, 8GB, SSD 240GB.

Os computadores com sistema operacional Ubuntu foram utilizados para hospedar a plataforma Dojot, ou seja, os testes foram replicados em máquinas diferentes de modo a garantir o seu funcionamento independente da arquitetura do processador. Utilizamos a máquina com Windows 10 para fazer o *deploy* dos algoritmos nos microcontroladores por termos enfrentado dificuldades ao realizar o *deploy* nas máquinas com Ubuntu.

4.1.5 SISTEMA OPERACIONAL

Utilizamos nos experimentos dois sistemas operacionais (SO): Ubuntu e Windows. O Linux Ubuntu foi utilizado nas duas máquinas desktop que hospedaram a Dojot. Esse SO foi escolhido por ser um software livre e por oferecer maior suporte às operações necessárias desde a instalação da plataforma até os testes.

O Windows não foi utilizado por escolha, mas por facilidade. Durante os testes com Ubuntu, encontramos dificuldades em fazer *deploy* dos algoritmos para os node que não encontramos no Windows. Com a instalação do *driver* do *node*, foi possível o *deploy* sem nenhuma dificuldade.

5 IMPLEMENTAÇÃO

Projetamos a aplicação sobre as ferramentas que a Dojot nos oferece. A aplicação visa o sensoriamento de distância entre objetos e o sensor para emitir alerta a partir de uma determinada distância. Trabalhamos com alertas visuais, contudo, é possível emitir qualquer tipo de alerta ou ação programável a partir da estrutura utilizada em nosso trabalho.

Inicialmente, definimos o protocolo de comunicação. Escolhemos o protocolo *Message Queue Telemetry Transport* (MQTT) por ser amplamente utilizado em aplicações IoT e também por ter sido projetado para ser leve e que trabalhe em condições que a oferta de banda e processamento é baixa (HWANG, 2016).

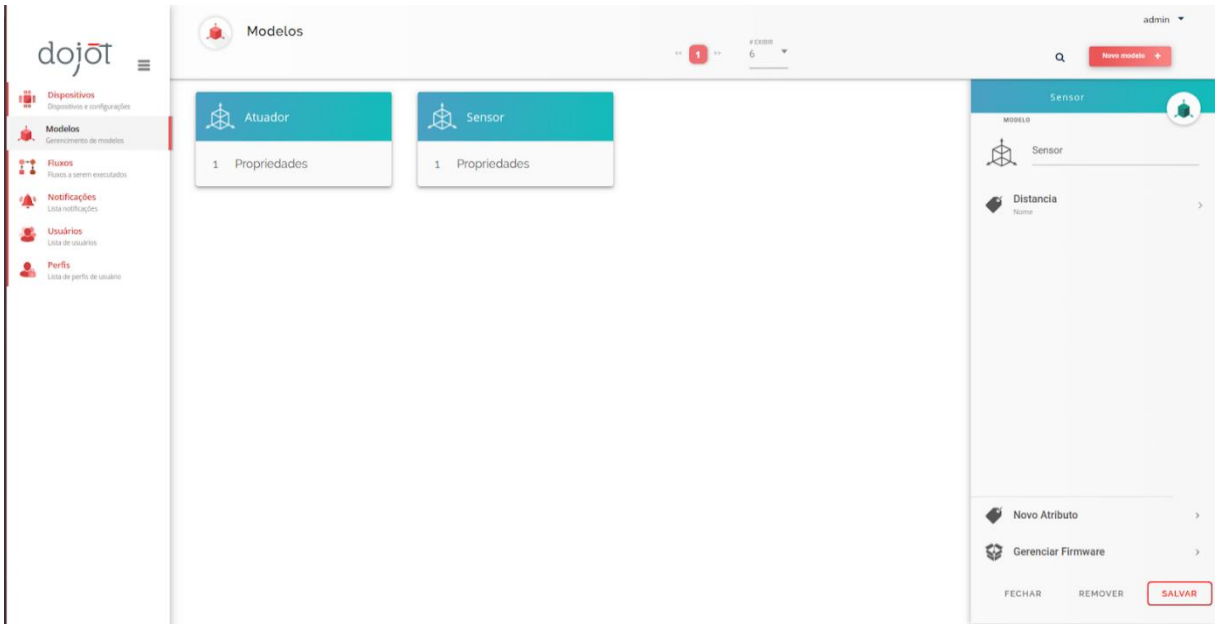


Figura 14 - Template sensor

Após isso, iniciamos a implementação do algoritmo para o primeiro NodeMCU, responsável por controlar o sensor ultrassônico e, através do protocolo MQTT, publicar os dados na plataforma. Devemos citar que para os dados serem publicados, há a necessidade da criação de um *template* definindo os atributos e seus tipos, como é possível observar na Figura 14, para, posteriormente, criar dispositivos que sejam identificáveis através de um ID único, este, fornecido pela plataforma. Além do *template* de *device* para o sensor, foi necessário também a criação de outro modelo para o atuador como é possível observar na Figura 15.

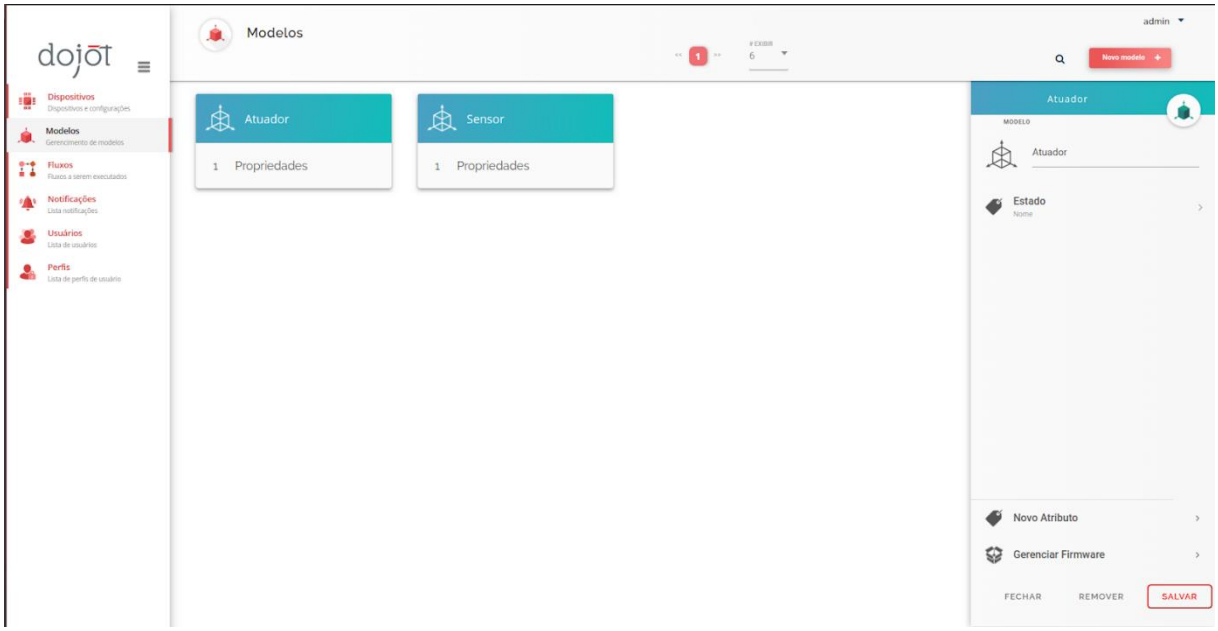


Figura 15 - Template Atuador

Com a plataforma recebendo dados, passamos para a criação do fluxo. O fluxo deve executar testes para cada dado recebido baseando-se nos parâmetros estabelecidos, e então decidir se a luz deve ou não ser ligada. Na construção do fluxo, utilizamos 1 componente de entrada (evento dispositivo), 2 de função (definir valor e interruptor), e 1 de saída (multi atuador). O fluxo construído é apresentado na Figura 16. É importante ressaltar que foi utilizado como padrão números inteiros para indicar o estado “1” para ligado e, “0” para desligado.

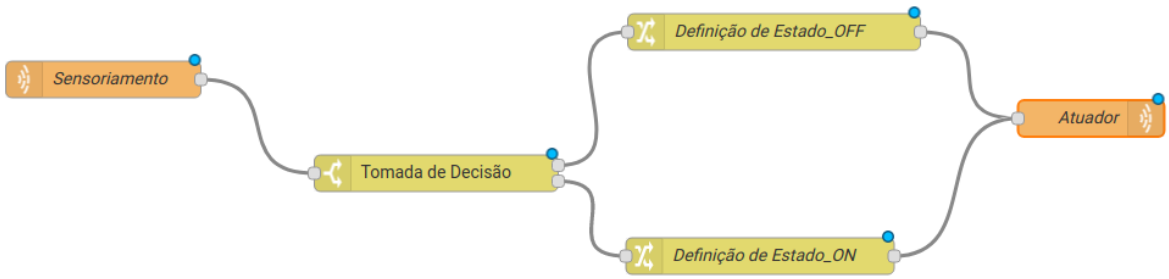


Figura 16 - Flow construído na plataforma Dojot.

Por fim, passamos a desenvolver para o atuador. A criação de um *template* com atributo do tipo atuador e a criação de um *device*. A criação de um *device* virtual estabelece o uso da fila do brokerMQTT. As informações do dispositivo ficam disponíveis na fila que pode

ser acessada por dispositivos externos. Com essas informações, configuramos o nodeMCU para buscá-las na plataforma o que ocorreu sem maiores problemas.

5.1 RESULTADOS

Com o ambiente configurado, testamos e visualizamos o funcionamento de toda a aplicação. Definimos como parâmetro a distância de 45 centímetros e dessa forma, espaço menor ou igual altera o estado da lâmpada para *on* e maior para *off*. A distância definida tem caráter experimental. A finalidade deste teste é de validar o funcionamento da aplicação, sendo assim, a distância definida como parâmetro se torna indiferente, podendo ser usados valores maiores ou menores, porém, deve ser respeitado os limites impostos pelo sensor utilizado que é de 4 metros de distância máxima e 2 centímetros para distância mínima. Os testes foram feitos com sucesso, alcançando totalmente o objetivo da aplicação projetada.

Durante a avaliação, notamos que ao realizar a ação de ligar ou de desligar o led há um *delay* de cerca de 2 a 3 segundos entre o envio da informação e a atuação. Como tentativa de redução desse tempo de resposta, reduzimos o tempo de envio dos dados do sensor ultrassônico à plataforma, porém, não obtivemos resultados consideráveis.

6 COMPARAÇÃO ENTRE DOJOT E FIWARE

As plataformas Dojot e FIWARE serão comparadas a partir de avaliações técnicas de utilização e documentação. A plataforma Dojot exige a criação de modelos (ou *templates* na tradução para a língua inglesa) para que se crie dispositivos. Nela, a relação é de 1:N, ou seja, é possível criar N dispositivos com as mesmas características utilizando apenas 1 *template*. Já na FIWARE, é necessário criar o que é chamamos de serviço, e que ainda não é a utilização de protocolos de comunicação, que gerará uma chave, e depois associar o serviço a um dispositivo. A criação de um dispositivo na FIWARE não está associada à criação de um serviço, porém, há a necessidade de associação de ambos via chave para que seja possível a utilização do dispositivo, como por exemplo, publicação de dados.

Por meio da interface que o FIWARE *Labs* implementa, é possível gerenciar a criação de serviços, instanciando-os sob demanda. Entretanto, diferentemente da Dojot, que suporta a utilização de *workflows*, a FIWARE não permite a criação de uma estrutura para tomada de decisão, sendo necessário o desenvolvimento de aplicações específicas. O *workflow* criado a partir da Dojot, permite a definição de regras para tratamento dos dados monitorados, bem como ações a partir de atuadores, seja em nível de software ou hardware.

Por fim, a FIWARE utiliza a instalação de componentes sob demanda, ou seja, não há a necessidade de instalação de todos os componentes disponíveis para utilização da plataforma. Além disso, cada componente FIWARE é instalado separadamente sem a necessidade de um orquestrador de containers. Do outro lado dessa mesma moeda, a Dojot requer a utilização de um orquestrador de container, o Docker-compose. A plataforma une todos os componentes em containers e os disponibiliza para uso. É possível escolher quais componentes serão conteinerizados, contudo, é necessário que o usuário altere o arquivo de configuração que é escrito em *yaml* antes do processo de conteinerização feito pelo Docker-compose.

7 CONCLUSÃO

O desenvolvimento e expansão da tecnologia no mundo fez com que diversos aparelhos eletrônicos como *smartphones*, televisores, computadores, entre outros, tivessem maior influência no nosso dia a dia. É fato que, o uso desses dispositivos, transformou a forma com que interagimos com outras pessoas, aumentando exponencialmente a quantidade de dados que trafegam na rede.

Os trabalhos de Mark Weiser (WEISER, 1991/WEISER, 1993) vislumbraram o que hoje chamamos de Internet das Coisas. Além dos vários dispositivos eletrônicos que fazem parte da nossa rotina, tornou-se possível a implementação de inteligência em dispositivos nunca antes imaginados, nas "coisas" e, também o desenvolvimento de aplicações com objetivo de melhorar o ambiente ao qual participamos. A utilização de sensores possibilitou que informações como temperatura, umidade, luminosidade, entre outros, fossem utilizadas para adaptar ambientes às suas necessidades de forma autônoma.

O crescimento da IoT se deu com tamanha intensidade e força que o conceito expandiu para as cidades inteligentes. O monitoramento de variáveis de ambiente como temperatura e umidade, condições estruturais como pontes e estradas, naturais como nível de rios e de chuvas, e diversas outras necessidades de uma cidade, chamou a atenção em níveis distintos da hierarquia política no mundo. A oportunidade de possuir todos esses dados para tomar decisões, associada aos problemas gerados pelo grande volume de dados produzidos em uma cidade, heterogeneidade de dispositivos, protocolos de comunicação, deu-se início a criação de plataformas de *middleware*.

A utilização de plataformas tem crescido cada vez mais por tornarem o desenvolvimento menos complexo. Com disponibilização de diversos protocolos de comunicação e também a concentração de dados, tudo isso em um único lugar, tornaram as plataformas de *middleware* a melhor opção para a construção de uma cidade inteligente. Com um conjunto de plataformas formado por SmartSantander, Sentilo, OpenIoT, entre outras, a Dojot surge como opção brasileira para integrar um sistema.

7.1 CONTRIBUIÇÕES

Neste trabalho, investigamos um conjunto de plataformas de *middleware* IoT para desenvolvimento de aplicações, e utilizamos a Dojot como ferramenta principal para essas implementações. Neste sentido, nossas principais contribuições foram:

- Visão geral de um conjunto de plataformas de *middleware* IoT existentes na literatura;
- Utilização de plataformas de *middleware* para desenvolvimento de aplicações IoT;
- Utilização de plataforma de *middleware* Dojot:
- Validação de componentes de comunicação (*IoT agent*);
- Validação de componentes de visualização de dados (*dashboard*);
- Validação de estrutura para tomada de decisão (*workflow*);
- Validação de conexão e uso com atuador (*IoT Agent* + *workflow*);

7.2 TRABALHOS FUTUROS

Como trabalhos futuros, pretendemos estabelecer testes com os outros *IoT agents* e protocolos disponíveis na plataforma Dojot. É também objetivo, explorar outros blocos disponíveis no *workflow* da Dojot e quais as possibilidades de uso.

Além disso, pretendemos investigar problemas relacionados à latência na comunicação entre sensor e atuador, apresentados na plataforma Dojot. Por fim, gostaríamos de apresentar a parceria realizada com o CPqD, no que tange a cooperação entre este centro de pesquisa e o Núcleo de Estudos aplicados a Redes de computadores e Sistemas distribuídos (NumbERS). Esta parceria tem por objetivo colaborar no desenvolvimento de microsserviços da plataforma ainda não finalizados.

8 REFERÊNCIAS

DOJOT. **Documentação do dojot**, c2019. Docs. Disponível em: < https://dojotdocs.readthedocs.io/pt_BR/stable/index.html >. Acesso em: 29 de janeiro de 2020.

DOJOT. **Dojot Soluções para IoT**. c2017. Disponível em: <http://www.dojot.com.br/sobre-a-dojot-iot/>. Acesso em: 29, jan. 2020.

PIMENTA, Regina. CPQD lança dojot, plataforma aberta para o desenvolvimento de aplicações de IoT. **CPQD**, 2017. Disponível em: <https://www.cpqd.com.br/releases/cpqd-lanca-dojot-plataforma-aberta-para-o-desenvolvimento-de-aplicacoes-de-iot/>. Acesso em: 29 de janeiro de 2020.

FAZIO, Maria et al. Exploiting the FIWARE cloud platform to develop a remote patient monitoring system. In: **2015 IEEE Symposium on Computers and Communication (ISCC)**. IEEE, 2015. p. 264-270.

KAFKA. **Documentação Kafka**, c2017. Documentation. Disponível em: <http://kafka.apache.org/documentation/#gettingStarted>. Acesso em: 29 de janeiro de 2020.

DATABROKER. **Repositório DataBroker**, s. D. README. Disponível em: < https://github.com/dojot/data-broker >. Acesso em: 29 de janeiro de 2020.

LIMA, Iago O.; MAIA, Marcio EF; REGO, Paulo AL. Sistema de identificação de dispositivos utilizando NodeMCU.

STEVAN, Sergio Luiz; SILVA, Rodrigo Adamshuk. **Automação e instrumentação industrial com Arduino: teoria e projetos**. Saraiva Educação SA, 2015.

BANZI, Massimo; SHILOH, Michael. Primeiros passos com o Arduino. **São Paulo: Novatec**, p. p1, 2011.

OLIVEIRA, Marcelo; GAMA, Kiev; LÓSCIO, Bernadette. Waldo: Servico para publicacao e descoberta de produtores de dados para Middleware de cidades inteligentes. In: **Anais do XI Simpósio Brasileiro de Sistemas de Informação**. SBC, 2015. p. 71-78.

SENTILO. **Sentilo**. c2013. Disponível em: < http://www.sentilo.io/wordpress/ >. Acesso em: 05 de fevereiro de 2020.

FIWARE FOUNDATION. **After the Open Day: From the FI-PPP to the FIWARE Foundation**. 2017. Disponível em: <https://www.fiware.org/2017/03/09/ after-the-open-day-from-the-fi-ppp-to-the-fiware-foundation/>. Acessado em: 05 de fevereiro de 2020.

FIWARE FOUNDATION. **About us**. c2020a. Disponível em: <https://www.fiware.org/about-us/>. Acesso em: 05 de fevereiro de 2020.

TRINDADE, Cláudio et al. Usando o FIWARE para Desenvolvimento de Aplicações de Cidades Inteligentes.

SILVA, Felipe Matheus Conceição da. Estudo de componentes de FIWARE para IoT e cidades inteligentes. 2019

SILVA, Pablo Henrique P.. **O que diabos é FIWARE?**, 2016. Disponível em: <https://medium.com/@pablohpsilva/o-que-diabos-%C3%A9-fiware-6b1cb80714c8>. Acesso em: 05 de fevereiro de 2020.

FIWARE FOUNDATION. **Developers**. c2020b. Disponível em: <https://www.fiware.org/developers/>. Acesso em: 05 de fevereiro de 2020.

KON, Fabio; SANTANA, Eduardo Felipe Zambom. Cidades Inteligentes: Conceitos, plataformas e desafios. In: **Congresso da Sociedade Brasileira de Computação**. 2016. p. 2-49.

PETROLO, Riccardo; LOSCRI, Valeria; MITTON, Nathalie. Towards a cloud of things smart city. **IEEE COMSOC MMTC E-Letter**, v. 9, n. 5, 2014.

SANCHEZ, Luis et al. SmartSantander: IoT experimentation over a smart city testbed. **Computer Networks**, v. 61, p. 217-238, 2014.

CHENG, Bin et al. Building a big data platform for smart cities: Experience and lessons from santander. In: **2015 IEEE International Congress on Big Data**. IEEE, 2015. p. 592-599.

DEL ESPOSTE, Arthur M. et al. InterSCity: A Scalable Microservice-based Open Source Platform for Smart Cities. In: **SMARTGREENS**. 2017. p. 35-46.

WU, Chao et al. Concinnity: A generic platform for big sensor data applications. **IEEE Cloud Computing**, v. 1, n. 2, p. 42-50, 2014.

TEI, Kenji; GÜRGEN, Levent. ClouT: Cloud of things for empowering the citizen clout in smart cities. In: **2014 IEEE World Forum on Internet of Things (WF-IoT)**. IEEE, 2014. p. 369-370.

HWANG, Hyun Cheon; PARK, JiSu; SHON, Jin Gon. Design and implementation of a reliable message transmission system based on MQTT protocol in IoT. **Wireless Personal Communications**, v. 91, n. 4, p. 1765-1777, 2016.

WEISER, Mark. Some computer science issues in ubiquitous computing. **Communications of the ACM**, v. 36, n. 7, p. 75-84, 1993.

WEISER, Mark. **The computing for the 21st century**. Scientific American, 265(3):66-75, Sept. 1991.

ROMÁN, Manuel et al. A middleware infrastructure for active spaces. **IEEE pervasive computing**, v. 1, n. 4, p. 74-83, 2002.

RORIZ, Marcos Paulino et al. C3S: A content sharing middleware for smart spaces. In: **2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)**. IEEE, 2013. p. 163-168.

ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The internet of things: A survey. **Computer networks**, v. 54, n. 15, p. 2787-2805, 2010.

ABUARQOUB, Abdelrahman et al. A survey on internet of things enabled smart campus applications. In: **Proceedings of the International Conference on Future Networks and Distributed Systems**. 2017. p. 1-7.

CAVALCANTE, Everton et al. Challenges to the development of smart city systems: A system-of-systems view. In: **Proceedings of the 31st Brazilian Symposium on Software Engineering**. 2017. p. 244-249.

PIRES, Paulo F. et al. Plataformas para a internet das coisas. **Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, p. 110-169, 2015.

BANDYOPADHYAY, Soma et al. Role of middleware for internet of things: A study. **International Journal of Computer Science and Engineering Survey**, v. 2, n. 3, p. 94-105, 2011.

LABORATORY OF VISUAL COMMUNICATIONS. **Smart Cities Course**. C2020. Disponível em: <http://lcv.fee.unicamp.br/index.php/courses/smart-cities-course>. Acesso em: 13 de fevereiro de 2020.