UMA PLATAFORMA PARA GERENCIAMENTO E APLICAÇÕES EM INTERNET DAS COISAS

JOSUÉ BATISTA ANTUNES

UMA PLATAFORMA PARA GERENCIAMENTO E APLICAÇÕES EM INTERNET DAS COISAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Daniel Fernandes Macedo
Coorientador: Prof. Dr. Aldri Luiz dos Santos

Belo Horizonte
Junho de 2016

Antunes, Josué Batista

A636p Uma Plataforma para Gerenciamento e Aplicações em Internet das Coisas / Josué Batista Antunes. — Belo Horizonte, 2016

xxiii, 87 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais

Orientador: Prof. Dr. Daniel Fernandes Macedo
Coorientador: Prof. Dr. Aldri Luiz dos Santos

1. Computação - Teses. 2. Internet das coisas. 3. Gerenciamento da internet das coisas. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*71(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Uma plataforma para gerenciamento e aplicações em internet das coisas

JOSUÉ BATISTA ANTUNES

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. DANIEL FERNANDES MACEDO - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ALDRI LUIZ DOS SANTOS - Coorientador
Departamento de Informática - UFPR

PROFA. FÁTIMA DE LIMA PROCÓPIO D. FIGUEIREDO
Departamento de Ciência da Computação - PUC-MG

PROF. JOSÉ MARCOS SILVA NOGUEIRA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 07 de junho de 2016.

Aos meus pais, Batista e Elza, por acreditarem na educação desde o princípio e minha esposa Gislane Cerqueira pelo apoio e confiança.

Agradecimentos

À minha família, que sempre dedicou todos os seus recursos para apoiar minhas escolhas acadêmicas e profissionais, em especial aos meus pais, esposa e irmãos.

Ao IFNMG campus Araçuaí, em especial aos colegas do Núcleo de Informática por atender o meu pedido e permitir o afastamento para capacitação.

Ao meu orientador, professor Daniel Macedo, pela enorme paciência com as minhas deficiências. Agradeço pelo apoio e pela dedicação, sempre indicando o melhor caminho para que este trabalho se concretizasse.

Ao meu coorientador, professor Aldri dos Santos, que através da sua extensa experiência, contribuiu para o desenvolvimento e a escrita deste trabalho.

Ao programa PBQS-IFNMG, pelo apoio financeiro concedido durante o período de capacitação, que foi de suma importância para alcançar as metas traçadas no seu devido tempo.

Aos colegas e professores do PPGCC-UFMG e do Laboratório Winet, em especial aos integrantes do projeto de pesquisa em IoT, Tiago, Marconi e István que tiveram uma contribuição fundamental no desenvolvimento deste trabalho.

A todos os amigos que fiz durante o tempo que fui Belo Horizontino, ao meu amigo e colega Carlos Anderson pelo apoio nas primeiras semanas, aos meus amigos Claudionor e Val pelos momentos de lazer. Enfim, a todos aqueles que não citei aqui, mas que diretamente ou indiretamente contribuíram de alguma forma para a execução deste trabalho.

A todos, os meus sinceros agradecimentos.

"A tarefa de viver é dura, mas fascinante."
(Ariano Suassuna)

Resumo

A Internet das Coisas (*Internet of Things* - IoT) é um paradigma que envolve uma variedade de dispositivos, denominados coisas, com capacidade de se conectarem à Internet e interagirem uns com os outros para atingir objetivos comuns. Tomando como base essa capacidade de comunicação, podemos criar soluções para integrar e gerenciar as funcionalidades individuais dos dispositivos e fornecer novos serviços, como utilizar os dados de um sensor para configurar a funcionalidade de um atuador. Um dos desafios da IoT consiste na heterogeneidade dos dispositivos que compõem a rede, onde cada objeto pode possuir diferentes capacidades de processamento ou padrão de comunicação diferente. Isso demanda um sistema de gerenciamento de configuração dinâmico e ciente de contexto. Contudo, os trabalhos atualmente encontrados na literatura não apresentam plataformas gerenciais que tratem de modo integrado questões essenciais à IoT, como gerenciamento multinível, por contexto e a escalabilidade. Logo, elas se limitam a solucionar problemas de um domínio específico, e portanto sendo insuficientes. Esta dissertação apresenta uma plataforma para o gerenciamento dos dispositivos e aplicações em IoT, chamada ManIoT (*Management for Internet of Things*), que, tendo como base os requisitos para a IoT, atua em âmbito local e global e considera a percepção do contexto. A divisão em dois níveis objetiva fornecer serviços dentro de um cenário (âmbito local) e para múltiplos cenários a partir de diretivas de alto nível (âmbito global). Essa dissertação aborda ainda o gerenciamento de múltiplas aplicações e múltiplos usuários, que prevê e trata problemas como conflitos e prioridade de execução. Um protótipo da plataforma para monitorar aplicações e dispositivos foi implementado. Os experimentos com cenários distintos mostraram que ManIoT fornece a estrutura necessária para a execução de múltiplas aplicações e serviços, tratando a integração entre dispositivos, aplicações e dispositivos heterogêneos. Os resultados mostraram ainda que ManIoT faz baixo uso de recursos computacionais, podendo, assim, ser executada em dispositivos residenciais com capacidade de processamento.

Palavras-chave: Internet das Coisas, Plataforma de Gerenciamento, ManIoT.

Abstract

The Internet of Things (IoT) paradigm encompasses a variety of devices, called things, with the ability to connect to the Internet and interact with each other to achieve common goals. Due to this ability to communicate, it is necessary to create solutions to integrate and manage the individual features of devices and provide new services, such as making use of data from a sensor to decide on an actuator. One of the challenges of IoT consists on the heterogeneity of the devices in the network, since each object may have different processing capabilities and/or communication standards. This requires a dynamic and context-aware configuration management system. However, the state of the art lacks management platforms that address the key issues of IoT in an integrated manner, such as multi-level management, context awareness and scalability. Furthermore, those works are restricted to a specific domain, and are therefore insufficient. This dissertation presents a platform that manages devices and applications on the Internet of Things, called ManIoT (Management for Internet of Things). It operates in local and global levels, and considers the perception of context. The division into two levels allows local control within a scenario (local level), while high-level policies are used to control multiple scenarios (global level). This dissertation also deals with the management of multiple applications and multiple users, treating problems such as conflicting commands and execution priority. A prototype of the platform was implemented, and experiments with different scenarios have shown that ManIoT supports applications and services that employ heterogeneous devices. The results indicate that ManIoT makes minimal use of computing resources and can thus be run in residential devices with modest processing capacity.

Keywords: Internet of Things, Management Platform, ManIoT.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

A Internet das Coisas (IoT - *Internet of Things*) é um paradigma onde uma variedade de dispositivos, como *Tags* e leitoras RFID (*Radio-Frequency IDentification*), nós sensores, telefones celulares, e objetos cotidianos (como lâmpadas, geladeiras e outros), são habilitados para interagir entre si e atingir objetivos comuns [Atzori et al., 2010]. A IoT traz, entre outros benefícios, a coleta de dados dos dispositivos e das suas condições operacionais em tempo real. Com esses dados, podemos automatizar tarefas domésticas, melhorar as tomadas de decisões nas empresas e entender os processos que ocorrem no ambiente onde as redes IoT estão inseridas.

De acordo com IBSG-Cisco [2011] (*Internet Business Solutions Group*), em 2010, mais de 12,5 bilhões de dispositivos estavam conectados à Internet. E a previsão é para cerca de 50 bilhões de dispositivos conectados até 2020. Diante do grande número de dispositivos conectados à Internet, surge a necessidade de criar mecanismos que proveem gerenciamento e controle desses dispositivos. Acreditamos que o gerenciamento eficiente dos dispositivos poderá garantir a continuidade da conexão, melhorar o uso dos recursos disponíveis e contribuir para alcançar uma rede com serviços de melhor qualidade.

Através do gerenciamento, podemos usar os recursos oferecidos pela IoT para fornecer serviços que vão atender diversas áreas. Entre as mais promissoras estão os serviços de apoio à saúde, infraestrutura e serviços do setor público. O monitoramento remoto em saúde, por exemplo, proporciona uma grande diferença na vida de pessoas com doenças crônicas e, simultaneamente, diminui os custos de cuidados com a saúde desses pacientes. A automação residencial usando IoT permite que dispositivos, tais como termostatos inteligentes, adequem a temperatura do ambiente e que lâmpadas sejam controladas à distância, em tarefas de segurança e economia de energia. No entanto, a IoT apresenta muitos desafios no seu gerenciamento até que se propicie

serviços inteligentes e integrados a qualquer momento e em qualquer lugar.

Podemos elencar alguns desafios ligados à IoT. O primeiro, e principal, está relacionado à heterogeneidade dos dispositivos que compõem a IoT. Segundo Atzori et al. [2010] a IoT utiliza diferentes formatos e tipos de dados para descrever o *status* das "coisas". Esses tipos de dados variam entre inteiro, caractere, e outros, incluindo dados semi-estruturados e não estruturados, tais como áudio e vídeo. Além disso, cada objeto tem uma capacidade de processamento diferente e possui uma forma de comunicação própria. Finalmente, fatores como a imprecisão dos dados produzidos (sistemas RFID podem gerar entre 60 e 70% de dados incorretos), o grande conjunto de dados produzidos em tempo real e a semântica implícita impõem desafios na configuração dos ambientes da IoT [Meng et al., 2013].

O gerenciamento de IoT é mais complexo que o gerenciamento de Redes de Sensores Sem Fio (RSSF) ou de redes IP. As redes IP, apesar de tratarem dispositivos com *hardware* e *software* heterogêneos, empregam mecanismos de comunicação homogêneos devido ao uso do protocolo IP. As RSSF devem gerenciar as falhas frequentes de comunicação e a baixa segurança dos enlaces sem fio (por exemplo a arquitetura MANNA [Ruiz et al., 2003]) e este gerenciamento deve ser ciente do contexto. Entretanto, os dispositivos de uma RSSF, em geral, tendem a ser mais homogêneos em configuração que na IoT. Nas redes IoT, além dos desafios acima, elas precisam suportar aplicações e serviços que envolvem: (*i*) o uso de dispositivos de características diferentes; (*ii*) a interação entre redes IoT, necessitando gerenciamento local (por exemplo o dono da casa) e global (a concessionária de energia, que procura reduzir a demanda em horas de pico), ambos cientes do contexto. As arquiteturas (ver Figura 2.2, [Bassi et al., 2013]) e plataformas de gerenciamento em IoT existentes, entretanto, atendem parcialmente a estes requisitos [Pires et al., 2015].

Neste trabalho desenvolvemos uma plataforma que atua em âmbito local (dentro de uma residência) e global e faz uso de dispositivos heterogêneos, Figura 1.1. No Capítulo 4 detalhamos todos os componentes da plataforma.

1.1 Objetivos

Neste trabalho, propomos um projeto para plataforma para gerenciamento de dispositivos em Internet das Coisas chamada ManIoT (*Management for Internet of Things*) e implementamos e avaliamos um protótipo. A plataforma integra e gerencia as funcionalidades individuais dos dispositivos em uma rede IoT e permite a criação de novos serviços cientes do contexto. O projeto ManIoT define uma estrutura de gerenciamento

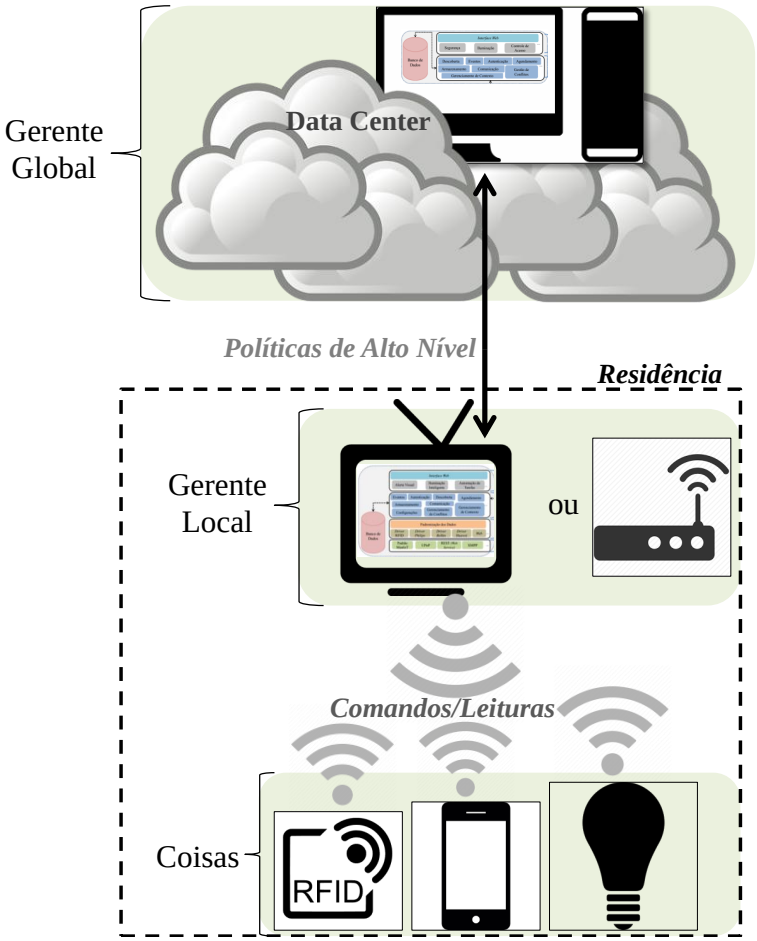

Figura 1.1. Topologia da Plataforma ManIoT

em dois escopos: gerenciamento local, onde a aplicação é executada no mesmo ambiente que os dispositivos, e gerenciamento global, onde as aplicações fornecem indicativos de operação para as aplicações locais através de diretivas de alto nível. Para executar o módulo local pretendemos utilizar equipamentos comuns em residências e que tenham capacidade modestas de processamento, como TV's e roteadores domésticos. O projeto da plataforma prevê ainda serviços genéricos, tais como descoberta de nós, armazenamento de dados e autenticação, que são blocos básicos para a construção de aplicações IoT. Além disso, a estrutura da plataforma é expansível, permitindo a adição de novos tipos de dispositivos.

Além da integração e controle das funções dos dispositivos, a plataforma proposta prevê o gerenciamento de conflitos, usuários e de aplicações. Entre os dispositivos que serão gerenciados pelo protótipo temos lâmpadas inteligentes, controladores/sensores de tomadas, recursos de um *smartphone* (localização e sensor de luminosidade), dispositivos de identificação (RFID) e nós de redes de sensores sem fio.

O trabalho foi desenvolvido em fases. Inicialmente, definimos os requisitos e especificamos as ações que devem ser tomadas no gerenciamento. Detalhamos o que iremos gerenciar nos dispositivos. Em seguida, criamos os casos de uso e definimos os detalhes do projeto, como o modelo de camadas da plataforma ManIoT, ferramentas e protocolos que devem ser utilizados, entre outros. Após a fase de especificação, implementamos o protótipo ManIoT e realizamos os testes. Através do protótipo, avaliamos parte das funcionalidades especificadas e do desempenho da plataforma proposta, como uso de rede, CPU e tempo de reação. Nas avaliações utilizamos dispositivos e cenários reais. Por fim, concluímos a pesquisa com a apresentação da dissertação.

Avaliamos a plataforma proposta através de um protótipo que integra dispositivos com características diferentes. Os resultados dos testes com cenários e dispositivos reais mostraram que a plataforma ManIoT faz uso mínimo de recursos computacionais e de rede, podendo assim ser executada em equipamentos de baixo poder computacional, como *TV* ou *Gateway*.

1.2 Contribuições

O desenvolvimento da proposta, do protótipo e os respectivos resultados obtidos nos testes da plataforma ManIoT trazem as seguintes contribuições:

- Especificação de uma plataforma para gerenciamento de dispositivos composta por: (*i*) um modelo de gerenciamento com dois escopos, (local e global) que possibilitam o gerenciamento em dois níveis, onde as decisões globais (executadas remotamente) podem sobrescrever as decisões locais; (*ii*) funções modeladas em camadas; (*iii*) mecanismos que permitem a expansão de serviços oferecidos por cada dispositivo individualmente.

- Modelagem de uma proposta envolvendo o uso de dados de contexto para dinamizar os serviços oferecidos.

- A implementação de um protótipo composto por dispositivos com diferentes características.

- Avaliação da plataforma proposta utilizando, para isso, cenários e dispositivos reais.

Em paralelo com a escrita desta dissertação elaboramos e publicamos um artigo chamado “ManIoT: Uma Plataforma para Gerenciamento de Dispositivos da Internet

das Coisas" no XXI Workshop de Gerência e Operação de Redes e Serviços (WGRS)[1]. Além do artigo, dois alunos de Iniciação Cientifica e um de Monografia Final, ambos do Departamento de Ciência da Computação da UFMG, desenvolveram seus trabalhos utilizando como base a proposta da plataforma ManIoT. E, por fim, temos um projeto de pesquisa aceito (Edital Nº01/2016 - IFNMG - *Campus* Araçuaí), com dois alunos de iniciação cientifica (Programa PIBIC-Jr do IFNMG), que também tem este trabalho como base.

1.3 Organização do Texto

A dissertação está organizada em sete capítulos. O capítulo 2 apresenta os conceitos fundamentais sobre internet das coisas e gerenciamento de redes. Identificamos os elementos básicos, desafios e as áreas de aplicação das redes IoT, em seguida, fizemos uma revisão do gerenciamento para redes tradicionais. Revisamos também o gerenciamento para redes IoT, apresentando as características e desafios específicos encontrados nessas redes. Concluímos o capítulo fazendo uma relação entre gerenciamento para redes de sensores sem fio e redes IoT.

No capítulo 3, apresentamos as arquiteturas e plataformas encontradas na literatura voltadas para gerenciamento de redes IoT. Relacionamos as arquiteturas que proveem os requisitos de ciência do contexto, segurança e extensibilidade. Diferenciamos, para contextualização deste trabalho, as arquiteturas das plataformas e apresentamos exemplos, destacando as principais características de cada solução. Em seguida, concluímos o capítulo fazendo a observação que as soluções analisadas não apresentam de forma satisfatória um modelo que possa atender todos os requisitos das redes IoT.

No capítulo 4, apresentamos a plataforma ManIoT. Destacamos inicialmente os requisitos e terminologias utilizadas neste trabalho, e logo em seguida, iniciamos a descrição da plataforma. Descrevemos o projeto do gerente local e gerente global que se apresenta como uma das principais contribuições deste trabalho. Destacamos e detalhamos os componentes de *software* que formam a plataforma e o relacionamento/comunicação entre eles. Criamos, ainda, um modelo de informação da plataforma que encapsula as operações sobre os elementos gerenciados. Em seguida, apresentamos o gerenciamento para múltiplas aplicações e múltiplos usuários da plataforma ManIoT. Destacamos a definição de um algoritmo para gerenciamento de conflitos que deve definir qual usuário ou aplicação tem permissão sobre um dispositivo de forma

[1]WGRS 2016 - http://www.sbrc2016.ufba.br/workshop/wgrs/

simples e prevendo o uso de dados do contexto. Por fim, sintetizamos e destacamos os pontos mais relevantes da plataforma desenvolvida.

O capítulo 5 descreve a implementação do protótipo da plataforma ManIoT. Relacionamos os equipamentos utilizados, modelo de dados definido para criação do banco de dados. Em seguida descrevemos a comunicação entre plataforma e os dispositivos, e caracterizamos os cenários utilizados para realização dos testes. Concluímos o capítulo enfatizando que as decisões sobre o uso de ferramentas, protocolos, entre outros, usados na implementação tiveram como embasamento teórico os trabalhos descritos no Capítulo 3, além disso, criamos novas abordagens, como três cenários para realização dos testes com dispositivos e ambientes reais.

No capítulo 6 apresentamos os resultados dos testes. O capítulo foi dividido em três partes, onde cada parte aborda um cenário. Em cada cenário destacamos o consumo de recursos, como memória, CPU e banda de rede, pelo protótipo da plataforma ManIoT. Apresentamos, ainda, outros parâmetros de avaliação de desempenho, como tempo de resposta. Por fim, concluímos o capítulo ressaltando que o protótipo ManIoT faz uso mínimo de recursos computacionais e poderia ser executado em dispositivos menos potentes, como televisores ou *gateways* IoT.

O capítulo 7 apresenta as conclusões obtidas com o trabalho. Destacamos as contribuições deste trabalho, como a definição de um plataforma genérica e escalável e que emprega uma estrutura em dois níveis de gerenciamento e promove a integração de vários dispositivos. Destacamos também que ManIoT faz uso dos dados de contexto e cria possibilidades para expansão dos serviços oferecidos pela rede. Na conclusão destacamos ainda o projeto para gerenciamento de múltiplas aplicações e múltiplos usuários. Em seguida, enfatizamos que um protótipo da plataforma foi implementado e experimentos foram realizados considerando um ambiente residencial composto por três cenários. Por fim, para trabalhos futuros propomos a definição de novos componentes da plataforma e de novos cenários.

Capítulo 2

Conceitos Fundamentais

Uma rede formada por coisas/dispositivos envolve diversos aspectos, a saber, *hardware*, *software* (protocolos), modelos, padrões, entre outros. Assim, é necessário definir alguns conceitos iniciais, elencar os desafios, e destacar a relação entre IoT e outras redes similares.

Neste capítulo iremos descrever os conceitos básicos sobre Internet das Coisas, gerenciamento de redes e gerenciamento para IoT. Na Seção 2.1 apresentamos os conceitos de Internet das Coisas e relacionamos as características dos dispositivos que compõem essas redes. Na Seção 2.2 descrevemos os conceitos ligados ao gerenciamento de redes, como formas de autenticação e de controle de acesso. E na Seção 2.3 tratamos o gerenciamento para Internet das Coisas, destacando os requisitos, as arquiteturas e plataformas de referência e os aspectos ligados a segurança.

2.1 Internet das Coisas

A Internet das Coisas (IoT - *Internet of Things*) é um novo paradigma que surgiu e está rapidamente ganhando terreno no cenário das telecomunicações. A ideia básica desse conceito é a possibilidade de interação, através de uma conexão com a Internet, entre diversos dispositivos chamados de Coisas - como Sensores, Celulares, Etiquetas, Lâmpadas, etc. - para alcançar objetivos comuns [Atzori et al., 2010].

A IoT é baseada no sucesso das redes móveis e da Internet. De acordo com a ITU [2005], as redes IoT do futuro serão capazes de detectar e monitorar em tempo real as mudanças no estado físico de dispositivos interligadas em rede. Segundo Miorandi et al. [2012] a IoT traz a mudança de uma Internet usada para interligar dispositivos de usuários finais para uma Internet usada para interligar objetos físicos que comunicam uns com os outros e/ou com seres humanos para oferecer um dado serviço.

A IoT trouxe o conceito de dispositivos inteligentes, e esses dispositivos podem interagir com os componentes de redes existentes, como terminais, roteadores, entre outros, [Marotta et al., 2013]. Esses dispositivos inteligentes devem possuir algumas características básicas como:

- Ter um formato físico, como tamanho, forma, etc.
- Ter um conjunto mínimo de funcionalidades de comunicação, como capacidade de ser descoberto, aceitar mensagens e responder a elas.
- Possuir um identificador único e responder através desse identificador.
- Possuir alguma capacidade de computação, que vai desde o envio de resposta para uma mensagem até a execução de operações complexas.
- Em alguns casos, meios para detectar fenômenos físicos, por exemplo, sensores de temperatura, de iluminação e umidade.

A IoT é um paradigma que une componentes físicos com soluções em *software*. A próxima seção descreve esses componentes e suas respectivas funções.

2.1.1 Elementos

A Internet das Coisas é formada por três componentes [Gubbi et al., 2013]: pelo *Hardware*, responsável pelo processamento e armazenamento das informações, através dos sensores e atuadores e *hardware* de comunicação embutidos; pelo *Middleware*, na demanda por armazenamento e ferramentas computacionais para análise de dados e; *Software* de Apresentação, ferramentas para entender, visualizar e interpretar os dados de diversas plataformas projetadas para diferentes aplicações. Existem algumas tecnologias facilitadoras que representam os três componentes acima referidos.

Como exemplo para componente de *hardware* da IoT temos a tecnologia RFID. Ela auxilia na identificação automática de qualquer coisa, agindo como um código de barras eletrônico. As aplicações que utilizam RFID podem ser encontradas no transporte de pessoas(substituindo os bilhetes e registos de automóveis) e aplicações de controle de acesso. Entre as várias aplicações do RFID, a principal está na identificação de *containers* portuários para monitoramento de carga [ITU, 2005].

Um dos resultados mais importantes do crescimento da IoT é o surgimento de uma quantidade dados sem precedentes. O armazenamento, a propriedade e a validade dos dados tornam-se questões críticas. Além disso, como o dinamismo da IoT novos algoritmos de fusão de dados precisam ser desenvolvidos para dar sentido aos dados

coletados. Métodos de aprendizagem de máquina temporais com base em algoritmos evolutivos, algoritmos genéticos, redes neurais e outras técnicas de inteligência artificial são necessárias para alcançar tomadas de decisões automatizadas [Gubbi et al., 2013].

A comunicação entre dispositivos da IoT é realizada através de diversos protocolos e essa conexão entre múltiplos dispositivos, segundo ITU [2005], é uma tarefa desafiadora. Os dispositivos podem ser ligados através de redes cabeadas (*wireline*) ou redes sem fio (*wireless*). Entretanto, a fixação e manutenção de cabos pode ter um custo alto além de demandar muito tempo. Assim, a IoT usa prioritariamente os protocolos de redes sem fio para ligar os dispositivos.

A IoT faz uso de protocolos de comunicação padronizados e protocolos específicos para dispositivos restritos. Os protocolos padronizados são aqueles já conhecidos e usados em outras redes, como as redes de computadores tradicionais, Ad Hoc e RSSF (redes constituídas por centenas ou milhares de nós sensores e que têm a capacidade de detecção, processamento e comunicação, usando para isso meios sem fios [Ruiz et al., 2003]). Já os protocolos específicos são adaptados para atender às limitações dos dispositivos conectados. Entre os procolos padrões temos o *Wi-Fi, Bluetooth*[1]*, Ethernet*, 3G, 4G-LTE e HTTP (*Hypertext Transfer Protocol*). Já no grupo dos protocolos específicos temos como uma das principais características a aptidão para funcionamento em dispositivos de baixa capacidade, temos como exemplos o ZigBee[2], Z-Wave[3], NFC[4] (*Near Field Communication*), 6LoWPAN (*IPv6 over Low power Wireless Personal Area Networks*) [Mulligan, 2007], RPL (*Routing Protocol for Low Power and Lossy Networks*) [Winter, 2012], COAP[5] (*The Constrained Application Protocol*), RFID, entre outros.

Com os recentes avanços em tecnologias, soluções como telas sensíveis ao toque, tornaram-se muito intuitivas. Para que um leigo possa se beneficiar plenamente da IoT, devem ser criadas formas atraentes de visualização e compreensão dos dados. À medida que avançamos de telas 2D para 3D, mais informações podem ser fornecidas de forma significativa para os consumidores. A extração de informações significativas a partir de dados brutos não é trivial. Isto engloba tanto a detecção e visualização de eventos quanto a identificação de dados relacionados [Gubbi et al., 2013].

[1]Bluetooth - https://www.bluetooth.com/
[2]ZigBee - http://www.zigbee.org/
[3]Z-Wave - http://www.z-wave.com/
[4]NFC - http://nfc-forum.org/
[5]COAP - http://coap.technology/

2.1.2 Desafios

A IoT possui alguns desafios que estão sendo tratados pela indústria e comunidade acadêmica em geral. Entre os desafios mais relevantes podemos destacar [Atzori et al., 2010]:

Definição de Padrões: existem diversas propostas de normalização, mas elas não estão integradas em um mesmo *framework*. Cabe destacar aqui duas formas de padronização de dados para prover comunicação entre os módulos de uma solução de gerenciamento chamados modelo de dados e modelo de informação.

O Modelo de Informação (IM - *Information Model*) é uma representação conceitual ou abstrata das operações possíveis nos elementos gerenciados, já o Modelo de Dados (DM - *Data Model*) especifica os dados com detalhes suficientes para armazenar e/ou transmitir as informações. IM e DM têm um papel de modelar e criar uma *interface* comum para permitir que o gerente da rede possa extrair suas informações e executar seu controle sobre os equipamentos gerenciados. Segundo Pras & Schoenwaelder [2003], o principal objetivo de um IM é modelar objetos gerenciados em um nível conceitual, independente da implementação ou protocolo específico usado para transportar os dados. Ainda segundo Pras & Schoenwaelder [2003], o grau de detalhe das abstrações definidas no IM depende das necessidades de modelagem de seus projetos. Assim, para deixar o projeto claro, um IM deve esconder todos os detalhes do protocolo e implementação. Outra característica importante de um IM é que ele define as relações entre objetos gerenciados. IM's podem ser definidos usando uma linguagem formal ou uma linguagem estruturada semi-formal. Uma das possibilidades para especificar formalmente um IM é utilizar diagramas de classe da UML (*Unified Modeling Language*) [Pras & Schoenwaelder, 2003]. Já os DM's são definidos em um nível mais baixo de abstração, se comparados com os IM's. O DM deve incluir muitos detalhes sobre o formato e codificação dos dados. A maioria dos modelos padronizados em IoT são DM's.

Suporte à Mobilidade: existem várias propostas para gerenciamento de dispositivos, mas nenhuma trata ou dá apoio à mobilidade no cenário da Internet das Coisas, onde a escalabilidade e capacidade de adaptação às tecnologias heterogêneas representam problemas cruciais.

Serviço de Nomes: uma rede IoT requer um servidor ONS - *Object Name Servers*(similar aos serviços de DNS - *Domain Name System*, para redes tradicionais). Esse servidor é necessário para mapear uma referência para descrição de um dispositivo específico e um identificador relacionado, e *vice-versa*.

Protocolo de Transporte: os protocolos de transporte existentes, segundo

Atzori et al. [2010], falham nos cenários da IoT pois seus mecanismos para estabelecimento de conexão e controle de congestionamento podem ser inúteis devido a falta de estabelecimento de conexão, comunicação fim-a-fim e *buffer* de memória na origem e destino.

Caracterização de Tráfego e Suporte a QoS - *Quality of Service*: IoT gera um tráfego de dados com padrões significativamente diferentes daqueles observados na Internet atual. Segundo Atzori et al. [2010] as características do tráfego em redes de sensores sem fio dependem fortemente do cenário de aplicação. Consequentemente, também será necessário definir novos requisitos de QoS.

Autenticação: a autenticação é difícil pois requer infraestrutura apropriada que não estarão disponíveis em cenários da IoT. Além disso, os dispositivos têm recursos escassos quando comparados com dispositivos de comunicação e computação atuais.

Privacidade: diversas de informações privadas sobre uma pessoa, como localização, podem ser coletadas pelas aplicações sem a consciência dessa pessoa. O controle sobre a coleta e difusão de todas as informações, segundo Atzori et al. [2010], é impossível com técnicas atuais.

Esquecimento Digital: as informações coletadas pela IoT sobre uma pessoa podem ser mantidas por longo período. Técnicas de mineração de dados também podem ser usadas para recuperar facilmente qualquer informação, mesmo depois de vários anos. É necessário então definir mecanismos para destruir ou proteger esses dados.

Assim como a quantidade de desafios, o número de áreas e aplicações que podem se beneficiar com as soluções da IoT também são grandes. Na próxima seção faremos um levantamento das principais aplicações para IoT.

2.1.3 Aplicações

Segundo Miorandi et al. [2012] existem diversas classes/categorias de aplicações e setores do mercado onde as soluções da IoT podem proporcionar vantagens competitivas sobre as soluções atuais. Essas aplicações abrangem diversos domínios: monitoramento ambiental, cidades inteligentes, *smart business*, casas inteligentes, saúde, segurança e vigilância, entre outros.

Os **edifícios ou casas inteligentes** podem ser equipados com tecnologias avançadas da IoT. A IoT pode ajudar tanto na redução do consumo de recursos associados aos edifícios (eletricidade, água), bem como na melhoria do nível de satisfação das pessoas. O impacto é tanto em termos econômicos (na redução de despesas operacionais), quanto em termos sociais (com redução da emissão de carbono). Nessas aplicações, um

papel chave é desempenhado pelos sensores, que são usados para ambos os monitores de consumos, bem como detectar proativamente as atuais necessidades dos usuários. Esse cenário pode integrar um certo número de subsistemas diferentes, e, portanto, exigir um elevado nível de padronização para assegurar interoperabilidade.

As **cidades inteligentes** surgem por meio da implantação de infraestrutura de comunicação e serviços sobre toda a cidade. As tecnologias da IoT podem, por exemplo, ser usadas para fornecer sistemas de controle de tráfego avançados. É possível monitorar o tráfego de automóveis nas grandes cidades ou estradas e implantar serviços que ofereçam sugestões de rotas de tráfego para evitar congestionamento. Os sistemas de estacionamento inteligente, baseados em tecnologias RFID e sensores, podem monitorar vagas livres e melhorar assim a mobilidade na área urbana. Além disso, os sensores podem monitorar o fluxo de tráfego de veículos em rodovias e recuperar informações agregadas, como velocidade média e número de carros. Por fim, os sensores podem ser utilizados em um ambiente forense, através da detecção de violações e transmitindo os dados relevantes para as agências de aplicação da lei, a fim de identificar o infrator.

As soluções da IoT podem ser aplicadas para **monitoramento ambiental**. Nessas soluções, um papel fundamental da IoT é desempenhado pela capacidade de detecção, de forma distribuída e auto-gestão de fenômenos e processos naturais (por exemplo, temperatura, vento, precipitação, níveis do rios), bem como a capacidade de integrar esses dados heterogêneos em aplicações remotas/globais. O processamento de informações em tempo real, juntamente com a capacidade de um grande número de dispositivos para comunicar entre si, fornecem uma plataforma sólida para detectar e monitorar anomalias que podem implicar riscos para a vida humana e animal. Os dispositivos da IoT podem atuar no acesso a áreas críticas, em que a presença de operadores humanos podem representar um risco (por exemplo, áreas vulcânicas, abismos oceânicos, áreas remotas). A segurança ambiental pode se beneficiar da capacidade de detecção dos dispositivos da IoT através da identificação de incêndios (por exemplo, através de sensores de temperatura).

A IoT pode fornecer uma série de aplicações para **cuidados com a saúde**. Elas podem ser utilizadas para auxiliar no tratamento de pessoas assistidas. Os pacientes podem transportar sensores médicos para monitorar parâmetros, tais como, temperatura corporal, pressão arterial e atividade de respiração. Outros sensores (por exemplo, acelerômetros, giroscópios e proximidade) podem ser usados para reunir dados e monitorar as atividades de pacientes em seus ambientes de vida. As informações podem ser agregadas localmente e transmitidas para centros médicos remotos, que podem realizar o monitoramento, com respostas rápidas quando necessárias. Outro setor de aplicação diz respeito a cuidados de saúde personalizados e soluções para bem-estar. O uso

de sensores como acessórios, em conjunto com aplicações adequadas permite que as pessoas rastreiem suas atividades diárias (quantidade de passos, calorias queimadas, exercícios realizados, etc.), fornecendo sugestões para melhorar o seu estilo de vida e prevenir o aparecimento de problemas de saúde.

Na **gestão e monitoramento de produtos** a tecnologia RFID predomina. As soluções baseiam-se na capacidade das tecnologias RFID em identificar e fornecer suporte para rastreamento de mercadorias. Normalmente, as etiquetas RFID são ligadas diretamente nos itens ou nos recipientes que os carregam, enquanto os leitores são colocados em locais estratégicos. Em aplicações para o varejo, as tecnologias da IoT podem ser usadas para monitorar a disponibilidade do produto em tempo real e manter o inventário de estoque correto. Eles também podem desempenhar um papel no apoio ao pós-venda, através do qual os usuários podem recuperar automaticamente todos os dados sobre os produtos comprados. Além disso, tecnologias de identificação podem ajudar a limitar roubos e auxiliar na luta contra a falsificação, fornecendo produtos com um identificador único, incluindo uma descrição completa e confiável do bem em si. Nos processos de produção, os sensores, em combinação com a tecnologia de RFID, podem permitir o controle de qualidade do produto final e evitar uma possível deterioração durante a vida de prateleira do produto.

Por fim, na área de **segurança e vigilância** os benefícios oferecidos pela IoT são diversos. A vigilância e segurança tornou-se uma necessidade para edifícios corporativos, *shoppings*, chão de fábrica, parques de estacionamento e muitos outros locais públicos. Tecnologias IoT podem ser usadas para aumentar consideravelmente o desempenho das soluções atuais, proporcionando alternativas mais baratas e menos invasivas para a implantação generalizada de câmeras enquanto, ao mesmo tempo, preserva a privacidade dos usuários. Sensores de ambientes podem ser utilizados para monitorar a presença de produtos químicos perigosos. Sensores que monitoram o comportamento das pessoas podem ser utilizados para avaliar a presença de suspeitos, gerando alertas eficientes e antecipando os riscos. A identificação pessoal por meios de RFID ou outras tecnologias semelhantes também é uma opção.

Portanto, o conjunto de aplicações para IoT é extremamente grande [Miorandi et al., 2012]. Mas os dispositivos, quando atuam de forma individual, fornecem capacidades restritas, surge então a necessidade de integração e gerenciamento desses dispositivos. As próximas seções detalham os aspectos relacionados ao gerenciamento nas redes tradicionais e, em seguida, nas redes envolvendo dispositivos da IoT.

2.2 Gerenciamento para Redes Tradicionais

As redes de computadores são compostas por uma grande variedade de dispositivos. Elas têm como objetivos fornecer comunicação e compartilhamento de recursos. Em grande parte, a eficiência dos serviços prestados está associada ao bom desempenho dos sistemas da rede. O gerenciamento das redes de computadores surgiu após a rápida evolução das tecnologias de redes, aliada a essa evolução houve também grande redução de custos dos recursos computacionais. Os serviços oferecidos passaram de simples compartilhamento de recursos para aplicações bem mais complexas como correio eletrônico, transferência de arquivos, aplicações multimídia, até chegarmos hoje na Internet das Coisas.

O objetivo da gerência de redes é monitorar e controlar os elementos da rede, físicos ou lógicos, assegurando certo nível de qualidade de serviço. Para realizar esta tarefa, os gerentes de redes são auxiliados por um sistema de gerência de redes. Um sistema de gerência de rede pode ser definido como uma coleção de ferramentas integradas atendendo algumas funcionalidades com o objetivo de monitoração e controle da rede. Este sistema oferece uma interface única, com informações sobre a rede e pode oferecer também um conjunto poderoso e amigável de comandos que são usados para executar quase todas as tarefas da gerência da rede, [Stallings, 2005].

O gerenciamento em redes de computadores envolve as atividades de administração, manutenção e uso eficiente dos equipamentos e sistemas em rede. O modelo clássico de gerenciamento de redes pode ser sumarizado em três etapas, [Kurose & Ross, 2010]:

- Coleta de dados: consiste na monitoração dos recursos gerenciados, normalmente, de forma automática.
- Diagnóstico: consiste no tratamento e análise dos dados monitorados para identificar o problema e propor uma solução para o mesmo.
- Ação ou controle: uma vez diagnosticado o problema, pode ser realizada uma ação ou controle sobre o recurso para alterar seu estado ou resolver o problema.

Podemos observar a importância do gerenciamento de redes quando deparamos com problemas, em alguns casos críticos, que podem levar à indisponibilidade dos serviços oferecidos. Segundo Kurose & Ross [2010], uma rede sem mecanismos de gerência pode apresentar problemas como interferência no tráfego de dados, falta de integridade dos dados, altas taxas de congestionamento, recursos podem ser mal utilizados ou sobrecarregados, além de diversos problemas de segurança.

De acordo com Comer [2009], o gerenciamento de redes pode ser difícil por três motivos: (*i*) na maioria das organizações, a rede gerenciada é heterogênea - contém componentes de *hardware* e *software* fabricados por várias empresas. (*ii*) a tecnologia continua a mudar, o que significa que novos serviços aparecem constantemente. (*iii*) a maioria das redes gerenciadas são grandes, o que significa que algumas partes podem ser distantes das outras. Diante desses problemas, é necessário definir as áreas onde o gerenciamento deve atuar.

Segundo Kurose & Ross [2010], a ISO (*International Organization for Standardization*) criou um modelo de gerenciamento de redes onde são definidas cinco áreas funcionais, são elas gerenciamento desempenho, falhas, configuração, contabilização e segurança:

- Desempenho. Tem como objetivo quantificar, medir, informar, analisar e controlar o desempenho de diferentes componentes da rede. Entre esses componentes estão dispositivos individuais - por exemplo, roteadores e hospedeiros - bem como abstrações fim a fim, como uma rota da rede.

- Falhas. O objetivo é registrar, detectar e reagir às condições de falha da rede. A divisão entre o gerenciamento de falha e o gerenciamento de desempenho é indefinida segundo segundo Kurose & Ross [2010]. O gerenciamento de falha pode ocorrer através do tratamento imediato de falhas transitórias da rede - como interrupção de serviço em um enlace ou *software* de roteadores. Já o gerenciamento de desempenho adota uma abordagem a longo prazo.

- Configuração. O gerenciamento de configuração permite que um administrador de rede saiba quais dispositivos fazem parte da rede administrada e quais são suas configurações de *hardware*. Sanchez et al. [2001] apresentam uma visão geral de gerenciamento e requisitos de configuração para redes IP.

- Contabilização. Tem como objetivo permitir que o administrador da rede especifique, registre e controle o acesso de usuários e dispositivos aos recursos da rede. As quotas e cobranças de utilização para acesso privilegiado aos recursos rede compõem o gerenciamento de contabilização.

- Segurança. A meta do gerenciamento de segurança é controlar o acesso aos recursos da rede de acordo com as políticas definidas na organização.

A arquitetura geral dos sistemas de gerência de redes, segundo Kurose & Ross [2010], apresenta quatro componentes básicos (Figura 2.1): elementos gerenciados, estações de gerência, protocolos de gerência e informações de gerência.

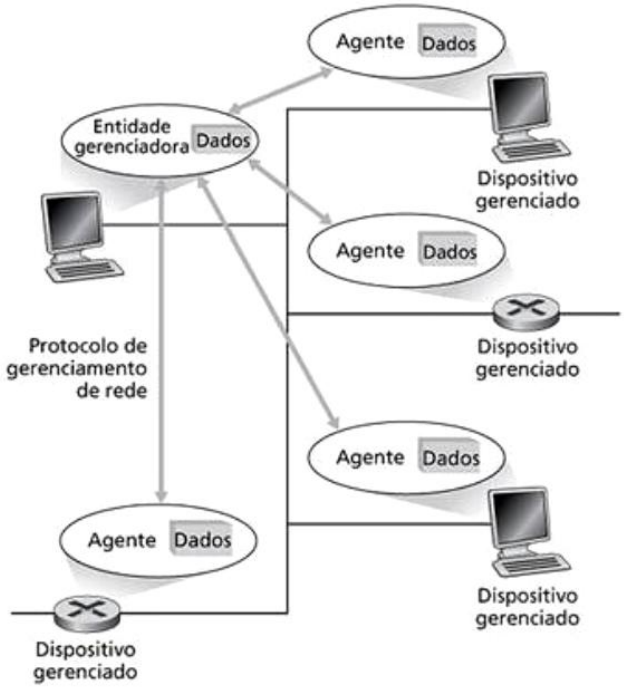

Figura 2.1. Componentes dos sistemas de gerência de redes, [Kurose & Ross, 2010].

- Agente: os elementos gerenciados possuem um *software* especial chamado agente. Este *software* permite que o equipamento seja monitorado e controlado através de uma ou mais estações de gerência;

- Gerente: chamamos de gerente o *software* da estação de gerência que comunica diretamente com os agentes nos elementos gerenciados, seja com o objetivo de monitorá-los, seja com o objetivo de controlá-los. A estação de gerência normalmente oferece ainda uma interface através da qual usuários autorizados podem gerenciar a rede;

- Protocolo: para que a troca de informações entre gerente e agentes seja possível é necessário que ambos utilizem o mesmo protocolo. Este protocolo permite operações de monitoramento (leitura) e controle (escrita);

- Informações de gerência: gerentes e agentes podem trocar informações. As informações de gerência definem os dados que podem ser referenciados nas operações do protocolo de gerência.

Os equipamentos de um rede (roteadores, comutadores, repetidores, impressoras, servidores e estações clientes) podem ter agentes instalados. A estação de gerência

deve obter informações de gerência destes agentes usando, por exemplo, o protocolo SNMP [Peterson & Davie, 2012].

Devido às características específicas da IoT, citadas na Seção 2.1, os dispositivos não podem ser gerenciados utilizando apenas as ferramentas de gerenciamento de tradicionais. A próxima seção descreve o modelo de gerenciamento para IoT.

2.3 Gerenciamento para IoT

A Cisco estima que o número de dispositivos que vão compor a IoT pode chegar a 50 bilhões em 2020, enquanto a IDC[6] extrapola esse valor com uma previsão de 212 bilhões de dispositivos. Nesse contexto, surge a necessidade de soluções para prover um gerenciamento eficiente desses dispositivos. Essas soluções devem fornecer interoperabilidade, segurança e suportar a crescente variedade de dispositivos associados às aplicações, bem como o consumo de dados por parte dos usuários finais.

O gerenciamento para dispositivos da IoT precisa adaptar a topologia dinâmica e, frequentemente, desconhecida dessas redes, fornecendo, por exemplo, informações de localização e estado dos dispositivos. Os serviços do gerenciamento devem desconectar dispositivos roubados, modificar configurações de segurança, localizar dispositivos perdidos, apagar dados sensíveis de dispositivos, entre outros. Segundo Delicato et al. [2013], o gerenciamento deve considerar ainda a possibilidade de dispositivos serem integrados ao ambiente e utilizados de maneira oportunista e não previamente planejada. Dessa forma, é importante que uma plataforma de gerenciamento possibilite a descoberta de dispositivos presentes no ambiente em questão, de forma dinâmica, a fim de atender os requisitos das aplicações.

O gerenciamento de IoT é mais complexo que o gerenciamento de redes de sensores sem fio (RSSF) ou de redes IP. As redes IP, apesar de tratarem dispositivos com *hardware* e *software* heterogêneos, empregam mecanismos de comunicação homogêneos devido ao uso do protocolo IP. As RSSF devem gerenciar as falhas frequentes de comunicação e a baixa segurança dos enlaces sem fio (por exemplo a arquitetura MANNA, [Ruiz et al., 2003]), e este gerenciamento deve ser ciente do contexto [Loureiro et al., 2003]. Entretanto, os dispositivos de uma RSSF em geral tendem a ser mais homogêneos em configuração que na IoT.

Já as redes MANETs (*Mobile Ad hoc Networks*), segundo Cordero et al. [2013] consistem em um conjunto de plataformas móveis, ou simplesmente "nós", que tem como característica a livre mobilidade. Ainda segundo o autor, as MANETs são siste-

[6]IDC Analyse the Future - www.idc.com/

mas móveis autônomos que podem operar isoladamente ou ter um *gateway* para fazer *interface* com uma rede fixa. Assim como as RSSF, as MANETs também possuem configurações homogêneas, o que viabiliza o uso de protocolos e soluções em escala global. Já na IoT, por exemplo, podemos ter como dispositivos de uma mesma rede um Sensor de Presença, um Servidor de Rede, uma Agenda *Web* e um *Smartphone*. Assim a principal diferença entre as RSSF, redes Ad Hoc (MANETs) e redes IoT é a característica dos dispositivos que formam essas redes.

Parte das ferramentas desenvolvidas para essas redes (RSSF e Ad Hoc) podem ser usadas nas redes IoT, mas devemos considerar também o fato que a todo momento surgem novos dispositivos aptos a comporem essas redes. Com isso as plataformas para gerenciamento em IoT requerem um módulo de *software* (*driver*) para traduzir as funcionalidades do dispositivo para a plataforma (por exemplo, a plataforma SmartThings faz uso desses *drivers*, [SmartThings, 2015]). Esse *driver* deve ser personalizado de acordo com o tipo de dispositivo (como aqueles citados no parágrafo anterior).

2.3.1 Requisitos

As soluções para gerenciamento em IoT devem tratar alguns requisitos [Pires et al., 2015], como:

- **Interoperabilidade** entre os diversos dispositivos e plataformas disponíveis no ambiente.
- **Descoberta e gerenciamento de dispositivos** presentes no ambiente em questão, realizada dinamicamente a fim de atender os requisitos das aplicações.
- **Ciência do contexto**, onde as informações, como localização e estado dos objetos da rede, são utilizadas para efetuar ações ou reagir a estímulos.
- **Escalabilidade** para aceitar a expansão e funcionar corretamente mesmo em situações de uso intenso.
- **Segurança e adaptação dinâmica** a fim de manter a integridade e privacidade dos dados disponibilizados e garantir a disponibilidade e qualidade das aplicações durante a sua execução.

Além dos requisitos citados, o gerenciamento para redes IoT deve suportar aplicações e serviços que envolvem: (*i*) o uso de dispositivos com características diferentes; (*ii*) a interação entre diferentes redes IoT, necessitando de gerenciamento local (por exemplo o dono da casa) e global (a concessionária de energia, que procura reduzir a

demanda em horas de pico), ambos cientes do contexto. As plataformas de gerenciamento em IoT existentes, entretanto, atendem parcialmente a estes requisitos [Pires et al., 2015].

Devido à falta de padronização, o gerenciamento dos dispositivos para IoT precisa definir uma forma de tratar essa lacuna. Para isso é necessário especificar modelos de dados e de informação, Seção 2.1.2. Esses modelos devem indicar a estrutura dos dados da rede. Com isso é possível definir um formato para armazenamento e implementação de outros serviços de gerenciamento.

Além da padronização, outros aspectos, como segurança, autenticação e autorização devem ser considerados. Diversos trabalhos buscam soluções para esses problemas. A próxima subseção destaca as características desejáveis nas plataformas para fornecer serviços de qualidade, considerando todas as peculiaridades da IoT.

2.3.2 Arquiteturas de Referência para IoT

As arquiteturas de referência podem ser definidas como um tipo de arquitetura abstrata que envolve conhecimento e experiências acerca de como projetar sistemas em um determinado domínio sendo, portanto, capaz de guiar o seu desenvolvimento e evolução. Além disso, as arquiteturas de referência podem ser utilizadas como um artefato de padronização para permitir interoperabilidade entre sistemas ou componentes de sistemas [Pires et al., 2015].

Os objetivos de uma arquitetura de referência são: facilitar o desenvolvimento de sistemas, promovendo redução de tempo e custo; padronizar arquiteturas de sistemas em um domínio; guiar a evolução de sistemas existentes. Segundo Pires et al. [2015], assim como em outros domínios, o estabelecimento de arquiteturas de referência é uma questão importante em IoT. Em primeiro lugar, os direcionamentos providos por uma arquitetura de referência são elementos essenciais para guiar e facilitar a construção de sistemas de IoT, considerando sua crescente escala e complexidade. Mais ainda, por proverem os blocos de construção fundamentais à construção das arquiteturas concretas de tais sistemas, arquiteturas de referência permitem construir sistemas capazes de atender aos requisitos existentes nesse domínio.

Como exemplo, o Modelo de Arquitetura de Referência (MAR) para IoT "*IoT Architectural Reference Model*" desenvolvido no contexto do projeto europeu *Internet of Things Architecture - IoT-A* é definida em um alto nível de abstração, fornecendo visões arquiteturais e perspectivas que são relevantes para a construção de várias arquiteturas para IoT [Bassi et al., 2013].

A visão funcional definida no MAR do IoT-A, Figura 2.2, possui nove grupos de

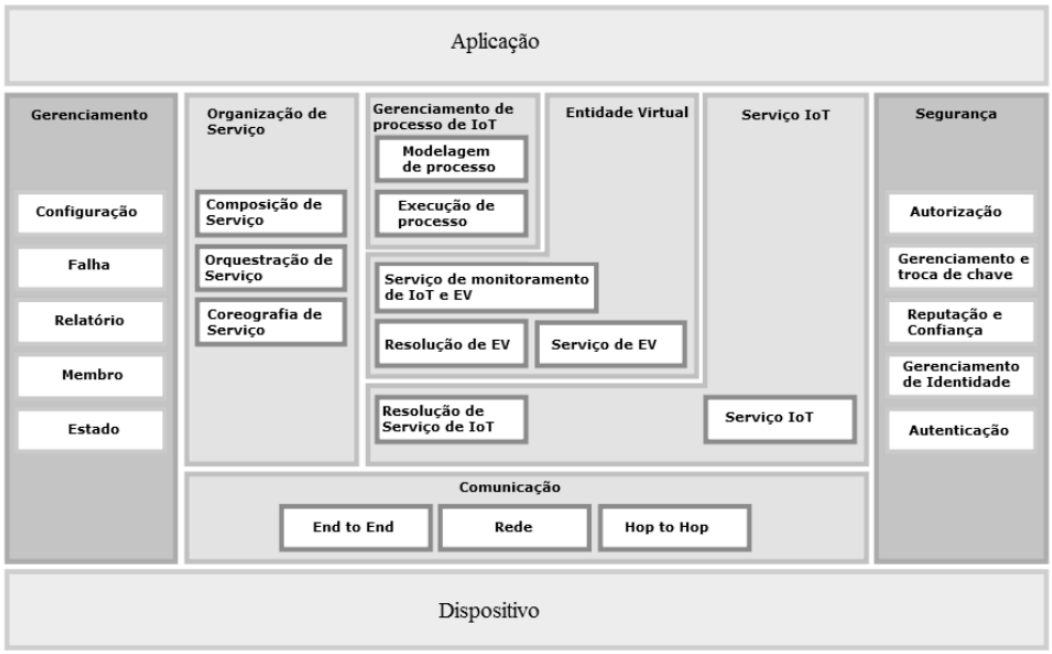

Figura 2.2. Arquitetura de Referência para IoT (IoT-A), [Bassi et al., 2013].

funcionalidades (GFs), a saber: (*i*) aplicação; (*ii*) gerenciamento; (*iii*) organização de serviço; (*iv*) gerenciamento de processo de IoT; (*v*) entidade virtual; (*vi*) serviço IoT; (*vii*) segurança; (*viii*) comunicação, e; (*ix*) dispositivo. Cada um desses grupos envolve um ou mais componentes funcionais. Entretanto, apesar da visão funcional descrever os componentes, ela não especifica as interações que ocorrem entre esses elementos pelo fato de tais interações serem tipicamente dependentes de escolhas de projetos, sendo portanto realizadas durante o desenvolvimento da arquitetura concreta [Pires et al., 2015].

Por envolver informações de usuários, alguns aspectos, como segurança e suas subáreas (controle de acesso, autenticação, etc) merecem destaque no gerenciamento para IoT. A próxima subseção trata desses aspectos.

2.3.3 Aspectos Ligados a Segurança

No contexto da Internet das Coisas, na maioria das situações, o papel dos dispositivos/Coisas é de coletar dados privados que podem inclusive serem transportados através de redes sem segurança adequada. Por isso, é importante que uma plataforma de IoT forneça estratégias de segurança, a fim de manter a integridade e privacidade dos dados disponibilizados, além de proteger tanto os dispositivos envolvidos quanto os recursos expostos à rede. Diversas técnicas, como prevenção à modificação maliciosa de dados ou ofuscação de código podem ser utilizadas para fornecer segurança aos dispositivos,

enquanto os recursos podem ser protegidos através de bloqueios de portas abertas e não usadas, uso de protocolos de segurança, uso de mecanismos para autorização e autenticação [Pires et al., 2015].

Segundo Jan et al. [2014], a maior dificuldade para prover segurança acontece devido ao fato de que cada dispositivo tem suas próprias características e requisitos. Jan et al. [2014] destacam ainda que a identidade de cada pessoa, objeto ou sistema ligado ao IoT deve ser estabelecida, pois na ausência dessa informação os intrusos podem ter acesso à rede e quebrar a sua segurança. As ameaças podem ser de natureza diversas, como a desativação de alarme, transmissão de informações falsas sobre estado de saúde ou falsos alarmes de incêndio.

Devido a vulnerabilidades inerentes à IoT, questões de segurança e privacidade devem ser consideradas no projeto de uma plataforma. De acordo com Liu et al. [2012] esses dois temas são os mais desafiadores para os projetistas. Antes do surgimento da IoT as violações dos sistemas computacionais não atingiam o mundo físico, com a IoT as falhas de segurança podem influenciar e comprometer o mundo físico.

Segundo Janak et al. [2012] a maioria das estruturas que fornecem controle de acesso são baseadas em funções: em um primeiro momento a identidade do usuário é estabelecida - autenticação, e logo depois seus privilégios de acesso são determinados a partir da função deste usuário dentro da organização. Esse modelo é utilizado na maioria dos sistemas existentes, como RADIUS, LDAP, IPSec, Kerberos e SSH. Ainda segundo Janak et al. [2012] os sistemas de controle de acesso baseados em funções não são adequados para dispositivos da IoT, pois a identidade do dispositivo pode não ser conhecida ou não importar. Então, segundo Janak et al. [2012], o controle de acesso deve ser baseado em outros critérios como localização, proximidade, esforço investido, entre outros. Os autores exemplificam que para gerenciar cenários relativamente simples, como "controlar a luz em um quarto somente se o usuário estiver localizado dentro deste mesmo quarto", é necessário um controle de acesso baseado em atributos genéricos.

Devido às peculiaridades da IoT, segundo Gusmeroli et al. [2013] o controle de acesso deve atender algumas características, são elas:

- Deve enfrentar o desafio da escalabilidade da IoT.

- Deve ser fácil de gerenciar. Devido à quantidade generalizada de dispositivos na IoT, os usuários estão mais envolvidos com atividades de segurança, como autorização, do que no passado.

- O sistema de controle de acesso deve apoiar as características avançadas, por exemplo, delegação de direitos, prover auditoria, entre outros.
- Por último deve ser flexível para se adaptar a diferentes contextos, adaptar às comunidades de usuários e suas necessidades.

2.4 Resumo

Este capítulo apresentou os conceitos fundamentais para formar o embasamento teórico sobre Internet das Coisas e Gerenciamento. Mostramos que gerenciamento em IoT é diferente do gerenciamento de redes tradicional pois as redes formadas por dispositivos da IoT precisam se adaptar a topologias dinâmicas e, frequentemente, desconhecidas. Além da possibilidade de dispositivos serem integrados ao ambiente e utilizados de maneira oportunista e não previamente planejada.

Discutimos e destacamos ainda neste capítulo que o gerenciamento para IoT é mais complexo que o gerenciamento para RSSF. As RSSF devem gerenciar as falhas frequentes de comunicação e a baixa segurança dos enlaces sem fio, e este gerenciamento deve ser ciente do contexto. Entretanto, os dispositivos de uma RSSF em geral tendem a ser mais homogêneos em configuração que na IoT. Devemos considerar ainda a falta de estrutura dos dados capturados pelos dispositivos da IoT.

Definimos também neste capítulo alguns elementos da IoT, relacionamos os protocolos de comunicação e apresentamos diversas áreas que podem se beneficiar com as aplicação da IoT. Por fim, tratamos do gerenciamento para os dispositivos da IoT e apresentamos o conceito e um exemplo de arquitetura de referência.

No próximo capítulo apresentaremos as arquiteturas e plataformas de gerenciamento para IoT dando enfoque especial para alguns requisitos, como ciência do contexto, escalabilidade, entre outros. Faremos ainda um relação entre as características dos trabalhos relacionados com aquelas que são importantes para este trabalho.

Capítulo 3

Arquiteturas e Plataformas de Gerenciamento para Internet das Coisas

Determinados trabalhos encontrados na literatura tratam arquiteturas e plataformas de gerenciamento como algo similar; neste trabalho iremos especificar uma plataforma de gerenciamento. As arquiteturas (ou projetos de arquiteturas) buscam atender todos os aspectos gerenciais e fornecer serviço completo. Stallings caracteriza arquitetura como "atributos de um sistema que são visíveis para o programador ou, em outras palavras, aos atributos que têm impacto direto sobre a execução lógica de um programa.", [Stallings et al., 2006]. O Já uma plataforma de gerenciamento fornece serviços pontuais e, em alguns casos, podem requerer um módulo complementar para tratar outros aspectos. Uma plataforma fornece suporte para outros serviços. No trabalho de Pires et al. [2015] os autores utilizam o termo "arquitetura" para descrever conhecimentos e experiências acerca de como projetar sistemas em um determinado domínio, sendo, portanto capaz de guiar o seu desenvolvimento e evolução. Neste mesmo trabalho o termo "plataforma" é usado para tratar todas as propostas implementadas.

A seção seguinte relaciona as plataformas de gerenciamento encontradas na literatura. Apesar do conceito "Internet das Coisas" ser relativamente novo, existem diversas soluções que buscam realizar esse gerenciamento de forma eficiente. Procuramos detalhar neste capítulo aquelas soluções que possuem características ou elementos que aproximam da plataforma ManIoT proposta neste trabalho. No restante deste capítulo, iremos apresentar as características das plataformas criadas para integrar e gerenciar os dispositivos que compõem a IoT. Por fim, na última seção, relacionamos as características comuns entre ManIoT e as arquiteturas e plataformas revisadas.

3.1 Arquiteturas para Gerenciamento dos Dispositivos e Dados da IoT

Os dispositivos da IoT geram grande volume de dados [Atzori et al., 2010]. Por isso, parte dos trabalhos relacionados estão voltados para construção de modelos para gerenciamento desses dados. O gerenciamento dos dispositivos, foco deste trabalho, é tratado por um número menor de autores. Nesta seção apresentaremos os trabalhos que buscam soluções para gerenciamento dos dispositivos e dos dados coletados por esses dispositivos, destacando aspectos importantes como ciência do contexto e segurança.

Os trabalhos de Guiping [2013], Bin et al. [2011], Ning et al. [2007] e Sehgal et al. [2012] propõem soluções para interação dos dispositivos de forma segura, sem falhas e com bom desempenho. Guiping [2013] propõe a criação de um protocolo para gerenciamento de dispositivos chamado TMP - *Things Management Protocol*. Segundo o autor, a motivação para criação do TMP veio da necessidade de gerenciar os dispositivos de forma independente. O TMP é um protocolo da camada de aplicação que deve fazer a ligação entre dispositivos, dispositivos-aplicação ou até mesmo entre aplicação-aplicação. O TMP inclui diversas operações - como *GetInformationObject* e *SetInformationObject*, PDU - *Protocol Data Unit* de operações, fluxo de processamento do protocolo emissor e receptor além da ligação do protocolo TMP com protocolos da camada de transporte.

Bin et al. [2011] propõem um sistema de gestão de equipamentos em edifícios chamado BEIOT - *Building Equipment Internet of Things*. Esse sistema, segundo os autores, deve permitir o acesso aos dados dos equipamentos de um edifício, como estado do ar condicionado e das lâmpadas dos quartos, e enviar comandos para alterar seus estados. Outras contribuições do BEIOT, segundo Bin et al. [2011], são a possibilidade de localização de pessoas através dos controladores de quarto e a otimização do consumo de energia que é feita tomando como base a quantidade de pessoas presentes em determinado ambiente.

Outro trabalho, desenvolvido por Ning et al. [2007], aborda o gerenciamento de sistemas RFID. Os autores utilizaram o protocolo SNMP (*Simple Network Management Protocol*) para gerenciar o dispositivo que recebe as informações das etiquetas e propõem a criação do protocolo RFID-MP (*Radio Frequency IDentification - Managing Protocol*), que controla o subsistema de terminal RFID.

Sehgal et al. [2012] trabalham com a gestão de dispositivos com recursos limitados. Os autores pesquisaram como o gerenciamento de rede baseado em IP (*Internet*

Protocol) pode ser implementado em dispositivos com recursos limitados e fizeram uma comparação entre o SNMP e o NETCONF (*Network Configuration Protocol*). Sehgal et al. [2012] enfatizam que a maioria dos dispositivos desenvolvidos para IoT são concebidos levando em conta as restrição recursos, onde o principal deles é o consumo de energia. Os autores identificaram os recursos mínimos exigidos pelos dispositivos que farão parte da IoT. Com os resultados obtidos, os autores observaram que SNMP faz uso mais eficiente dos recursos, chegando a responder um pedido de processamento até 10 vezes mais rápido que NETCONF. Outro fator observado por Sehgal et al. [2012] foi que as questões de segurança geram um custo alto no processamento e que esse custo poderia ser reduzido com a implementação do suporte a criptografia em *hardware*. Os autores chegaram à conclusão que os protocolos que utilizam pequenas mensagens da camada de aplicação e que cabem em um pacote IPv6, são muito mais adequados para dispositivos com recursos limitados.

A próxima subseção destaca os trabalhos que abordaram o uso de dados do contexto. A ciência do contexto é um dos principais requisitos para gerenciamento em IoT, segundo Atzori et al. [2010], ITU [2005] e Pires et al. [2015].

3.1.1 Ciência do Contexto

A ciência do contexto, em linhas gerais, é a coleta de entradas capazes de refletir as condições atuais dos usuários, do ambiente no qual o mesmo se encontra e do próprio dispositivo computacional utilizado [Loureiro et al., 2009]. Esse paradigma é abordado em algumas soluções de gerenciamento. Nessas soluções, as informações, como localização e horário de coleta, são utilizadas para caracterizar os dados levantados. Nos trabalhos chamados Ubiware [Nagy et al., 2009] e LinkSmart [Lang, 2014] os autores fizeram uso dos dados de contexto.

A plataforma Ubiware incorpora princípios de sistemas multi-agentes, entidades computacionais com comportamento autônomo que facilitam o desenvolvimento de sistemas complexos. Ubiware é estruturado no núcleo da plataforma chamado Ubicore. Esse componente provê a todos os dispositivos conectados a possibilidade de serem inteligentes ao conectá-los a um agente de *software*. Dessa forma, tais objetos ganham recursos de comunicação, autocontrole e auto-monitoramento através da utilização do conhecimento e das funcionalidades previamente adquiridos a partir de eventos próprios e externos. Sobre a relação com ciência de contexto, o fato de que cada objeto tem um agente vinculado a ele implica que tal agente tem pleno conhecimento acerca de seu estado. Dessa forma, tal conhecimento pode ser trocado com outros agentes e utilizado para melhorar a execução de outros objetos vinculados à plataforma [Nagy et al., 2009].

LinkSmart é uma plataforma baseada em Arquitetura Orientada a Serviços (*SOA – Service-Oriented Architecture*) para IoT que oferece suporte ao desenvolvimento de aplicações formadas por dispositivos físicos heterogêneos que operam com recursos limitados em termos de poder computacional, energia e memória [Pires et al., 2015]. Ela oferece interfaces de serviços web para controle de dispositivo físico e permite que desenvolvedores incorporem dispositivos físicos heterogêneos em suas aplicações. Sua arquitetura possui três camadas principais, sendo uma delas a camada semântica. Nessa camada é realizado o tratamento de informações de contexto dos dispositivos através do uso de ontologias de dispositivos. Responsável por representar todas as meta-informações sobre os dispositivos, a ontologia usada é baseada na ontologia de dispositivos FIPA (*Foundation for Intelligent Physical Agents*) e permite a parametrização semântica para incluir informações dos dispositivos, como seus recursos de segurança [Pires et al., 2015].

Em outros trabalhos, como Elkhodr et al. [2016] e Ribeiro & Metrôlho [2016], os autores projetaram soluções que também procuram fazer uso de dados de contexto. Em Elkhodr et al. [2016], os autores desenvolveram um projeto para gestão e preservação da privacidade e localização dos dispositivos da IoT. Eles desenvolveram um projeto de *middleware* que utiliza uma abordagem de contexto adaptativo, que permite aos usuários realizar o gerenciamento das informações de localização divulgadas pelo dispositivos com base nos dados de contexto e nas políticas de execução. Este mecanismo leva em conta tanto o consentimento informado pelo usuário quanto suas preferências. Já no trabalho de Ribeiro & Metrôlho [2016], os autores desenvolveram uma solução que objetiva integrar, gerenciar e utilizar informações sobre padrão e objetos de personalização dos usuários. Os objetos podem ser registrados na plataforma e enviar suas informações, como identificação do objeto, o tipo de informação que eles representam e também o nível de acessibilidade às suas informações. Entretanto nenhum desses trabalhos apresentaram testes com dispositivos e cenários reais.

Além da ciência do contexto, os requisitos segurança e extensibilidade devem ser tratados ou previstos nas soluções para IoT. A próxima subseção descreve trabalhos que tratam do requisito segurança.

3.1.2 Segurança

O requisito de segurança é subdivido em autenticação, controle de acesso, criptografia de dados, entre outros. Os trabalhos encontrados na literatura, de alguma forma, fornecem um nível mínimo de segurança. Relacionamos brevemente alguns trabalhos que relataram os problemas e soluções envolvendo as redes IoT.

Jan et al. [2014] propõem um esquema de autenticação que verifica a identidade dos participantes e servidores em um ambiente utilizando protocolo CoAP (*Constrained Application Protocol*). Neste modelo cada cliente mantém uma chave de sessão com um servidor que garante para ambas as partes a autenticação. O servidor, então, notifica os clientes quando as condições são satisfeitas. Com esse modelo, é possível reduzir o número de transmissões indesejáveis e, consequentemente, diminuir o congestionamento na rede. De acordo com Jan et al. [2014] essa solução não é eficiente contra ataques Sybil. No ataque Sybil, um único nó malicioso coloca múltiplas identidades para os dispositivos de comunicação em um dado tempo. Essas identidades são produzidas ou roubadas desativando os nós legítimos da rede. Assim, um dispositivo físico único pode prejudicar vários recursos da rede.

Já Flood & Schukat [2014] propõem um novo método para fornecer autenticação e criptografia em redes M2M (*Machine-to-Machine*). O método é baseado no protocolo GMW (*Goldreich-Micali-Wigderson*) e na troca de chaves Diffie-Hellman. A abordagem, de acordo com Flood & Schukat [2014] é estruturalmente semelhante a uma chave simétrica compartilhada.

Pereira et al. [2014] apresenta um *framework* baseado em CoAP e propõe um método especifico para autenticação. A autenticação funciona como um primeiro passo para prover o controle de acesso. Assim, o sistema deve reconhecer o usuário através de uma chave compartilhada ou outro validador e informar ao CoAP-NAS (*Network Access Server* CoAP - Figura 3.1) sobre o usuário, permissões, grupo e tempo total da permissão. A Figura 3.1 apresenta ainda a sequência temporal desse processo. Podemos perceber que, se a permissão expirar, o sistema requisitará uma nova autenticação e o processo se reiniciará.

Segundo Janak et al. [2012] o padrão XACML (*Extensible Access Control Mark-up Language*) e a ferramenta OAuth também podem serem utilizados para descrever regras de controle de atributo e controle de acesso para aplicativos da IoT, respectivamente. OAuth requer a identidade dos aplicativos através da apresentação de *tokens*.

O *framework* OAuth, na sua versão 2.0, definido na RFC 6749 [Hardt, 2012] é utilizado por diversas implementações, como a plataforma WSO2 [Cavalcante et al., 2015]. OAuth 2.0 define quatro funções no controle de acesso, são elas:

- Proprietário do recurso: uma entidade capaz de possibilitar o acesso a um recurso protegido.

- Servidor de recursos: dispositivo que hospeda os recursos protegidos. Ele deve ser capaz de aceitar e responder a solicitações utilizando permissões de acesso.

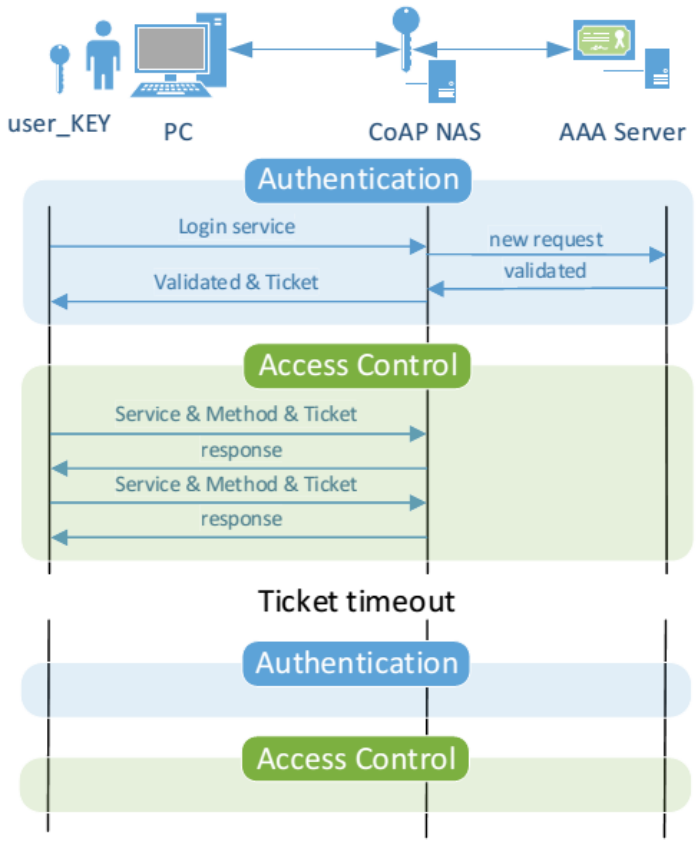

Figura 3.1. Processo de Autenticação definido por Pereira et al. [2014]

- Cliente: um aplicativo que faz solicitações de recursos protegidos em nome do proprietário do recurso e com sua autorização.

- Servidor de Autorização: dispositivo que emite permissões de acesso para o cliente após a autenticação e uma autorização para utilizar determinado recurso.

XACML, segundo Wu & Periorellis [2005], utiliza XML (*Extensible Markup Language*) para comunicar as políticas de controle de acesso entre os serviços. Ele fornece um esquema para expressar as políticas e regras baseadas nessas políticas. O trabalho de Cavalcante et al. [2015] também utiliza esse padrão. As atualizações e outros detalhes do padrão podem ser encontradas em OASIS[1].

[1]OASIS - *Advancing Open Standards for the Information Society*, disponível em http: www.oasis-open.org

3.1.3 Extensibilidade

A extensibilidade em redes é a capacidade de absorver e tratar uma porção crescente de trabalho ou de fornecer estrutura para receber novos componentes. Além da ciência do contexto e segurança, a extensibilidade é requisito tratado por alguns trabalhos existentes na literatura.

O requisito de extensibilidade foi tratado nos trabalhos de Fan & Chen [2010], Ding et al. [2011] e Jara et al. [2012]. Fan & Chen [2010] pesquisam o gerenciamento de dados gerado por diferentes dispositivos e utilizam uma abordagem SOA (*Service Oriented Architecture*). Essa abordagem é justificada pelas vantagens apresentadas pelo SOA, como redução do custo de desenvolvimento e o encapsulamento dos detalhes específicos dentro da lógica do serviço. Segundo Fan & Chen [2010], uma das dificuldades encontradas é a heterogeneidade dos dispositivos. Gerenciar a heterogeneidade é um problema comum em IoT, que deverá ser enfrentado também pelo nosso trabalho.

Ding et al. [2011] propõe uma base para gerenciamento de dados chamado "SeaCloudDM". Segundo os autores, SeaCloudDM reduz consideravelmente os dados que precisam ser gerenciados através de atribuições de compartilhamento entre a camada *sea-computing*, definida no trabalho, e a camada de gerenciamento de dados nas nuvens.

Jara et al. [2012] abordam o gerenciamento de dispositivos móveis voltados para a área de saúde pessoal e sua integração com dispositivos clínicos. A solução proposta, chamada YOAPY, utiliza RFID para identificação do paciente e 6LoWPAN - (*IPv6 over Low Power Wireless Personal Area Networks*) para transmissão dos dados. YOAPY visa otimizar o desempenho da comunicação para objetos de baixa capacidade, como largura de banda e energia. YOAPY analisa os sinais vitais contínuos do paciente para detectar anomalias e não sobrecarregar a comunicação.

A próxima seção destaca as plataformas desenvolvidas para gerenciar os dispositivos da IoT e a capacidade de integrar dispositivos diferentes (interoperabilidade), que é a principal característica dessas soluções.

3.2 Plataformas de Gerenciamento para IoT

Na literatura, há diversas plataformas de gerenciamento para IoT. Essas plataformas são comumente denominadas *middleware*, pois são camadas de *software* que ocultam dos desenvolvedores as complexidades e heterogeneidades referentes ao *hardware*, aos protocolos de rede, às plataformas e às dependências do sistema operacional [Razzaque et al., 2016]. Elas também facilitam o gerenciamento de recursos e aumentam a

previsibilidade da execução de aplicações. A interoperabilidade entre os dispositivos é o principal requisito para as plataformas de gerenciamento, [Pires et al., 2015].

A interoperabilidade é a capacidade de prover comunicação entre dois ou mais sistemas e/ou dispositivos. A grande maioria das plataformas para IoT fornecem, de alguma forma, essa interação. As plataformas RestThing [Qin et al., 2011], EcoDiF [Delicato et al., 2013] e SmartThings [SmartThings, 2015] abstraem a heterogeneidade dos usuários e aplicações. A plataforma RestThing visa permitir que desenvolvedores criem aplicações usando REST (*REpresentational State Transfer*), combinando recursos físicos e *Web*, de modo que dispositivos e informações *Web* sejam manipulados por uma *interface* REST. A plataforma EcoDiF (Ecossistema *Web* de Dispositivos Físicos) integra dispositivos físicos heterogêneos e os conecta à Internet, fornecendo funcionalidades de controle, visualização, processamento e armazenamento de dados em tempo real. A plataforma *open source* SmartThings permite que os usuários criem aplicativos e os conectem aos dispositivos, às ações e aos serviços oferecidos pela plataforma. SmartThings permite ainda a integração de novos dispositivos e fornece suporte para os aplicativos (*SmartApps*) na comunicação com serviços *Web* externos através do envio de notificações *Push*, SMS e da apresentação do seu terminal REST.

Outra plataforma chamada IFTTT (*If This, Then That*) [IFTTT, 2016], fornece interoperabilidade entre dispositivos físicos e serviços da rede. IFTTT é um serviço que trabalha com sistemas *Web*, *Android* e iOS, e permite criar conexões com uma declaração simples do tipo "se isso acontecer, então faça aquilo". Ele combina duas contas de diferentes sites ou redes sociais para interagir e funcionarem juntas. Os serviços são baseados em três elementos: o canal de gatilho, o canal de reação e a descrição da ação. Por exemplo, o gatilho pode ser o GPS do celular e o canal de reação o SMS. A descrição seria "enviar uma mensagem à minha mulher avisando que eu estou chegando em casa". Em outros exemplos, o IFTTT pode "*twittar* automaticamente o *status* postado no Facebook." Além disso, o usuário pode criar serviços personalizados, ligando dois ou mais eventos. Além do Twitter e do Facebook, o IFTTT tem a capacidade de atuar com os serviços web, como Instagram, Dropbox, Gmail, YouTube, Google Calendar, entre outros. O IFTTT pode atuar também com dispositivos físicos como Philips Hue e WeMo Belkin.

Na próxima seção apresentamos uma relação com as principais características dos trabalhos relacionados. Destacamos ainda, entre essas características, aquelas que são implementadas ou desejáveis para a plataforma ManIoT, proposta neste trabalho.

3.3 Principais Características das Arquiteturas e Plataformas para IoT

Nesta seção, relacionamos as características mais importantes dos trabalhos analisados e destacamos aquelas que foram implementadas (marcações ✓), aquelas que não foram implementadas (marcações −) e aquelas desejadas (marcações O). Essa classificação (Tabela 3.1) tem como objetivo mostrar de forma clara e resumida os requisitos, como gerenciamento de contexto e abordagem multi-cenários, que são tratados ou previstos pela arquitetura ManIoT e pelos principais trabalhos encontrados na literatura que fazem revisão sobre IoT, como [Razzaque et al., 2016], [Pires et al., 2015], [Atzori et al., 2010] e [ITU, 2005].

A heterogeneidade é um dos requisitos mais importantes para gerenciamento de dispositivos da IoT. Essa característica é tratada pela maioria das soluções relacionadas e também está prevista neste trabalho quando utilizamos diferentes dispositivos como lâmpadas inteligentes, sensores de luminosidade e leitores RFID.

Além da heterogeneidade, a segurança e a privacidade são requisitos tratados por grande parte dos trabalhos. No contexto de IoT, muitas vezes o papel dos dispositivos integrados é o de coletar dados privados que podem ser transportados através de redes sem segurança adequada [Pires et al., 2015]. Por essa razão, é importante que as soluções forneçam estratégias de segurança, a fim de manter a integridade e a privacidade dos dados disponibilizados, além de proteger tanto os dispositivos envolvidos quanto os recursos expostos à rede.

Diversas organizações, como Cisco [IBSG-Cisco, 2011], preveem que bilhões de dispositivos estarão aptos a serem utilizados por aplicações em curto prazo de tempo. Dessa forma, as soluções para IoT devem dar suporte à escalabilidade e à confiabilidade. As soluções devem prever um número crescente de dispositivos e requisições e funcionar corretamente, mesmo em situações de uso intenso. A plataforma ManIoT, ao criar uma camada que possibilita a inclusão de novos drivers para gerenciar novos dispositivos, provê escalabilidade.

A Tabela 3.1 relaciona também os procolos e estilos de projetos (RFID, SNMP, NETCONF, 6LoWPAN e SOA) citados como características da IoT pelos autores, [Atzori et al., 2010], [Cavalcante et al., 2015] e [Delicato et al., 2013]. Podemos observar que 6LoWPAN e NETCONF não são amplamente adotados pelas soluções existentes. Mas, devido as suas características, como aptidão para trabalhar com dispositivos de baixa capacidade, esses protocolos são desejáveis para a plataforma ManIoT.

A classificação das soluções entre aquelas que tratam do gerenciamento dos da-

Tabela 3.1. Características dos trabalhos relacionados e da solução proposta

	Fan & Chen [2010]	Ding et al. [2011]	Jara et al. [2012]	Guiping [2013]	Bin et al. [2011]	Ning et al. [2007]	Sehgal et al. [2012]	Delicato et al. [2013]	Qin et al. [2011]	SmartThings [2015]	IFTTT	ManIoT
Heterogeneidade	✓	–	–	✓	✓	–	✓	✓	✓	✓	✓	✓
Segurança e Privacidade	–	✓	–	–	✓	–	✓	✓	–	✓	✓	✓
Escalabilidade e Confiabilidade	–	✓	–	–	–	–	–	✓	✓	✓	✓	✓
Aborda RFID	✓	–	✓	–	–	✓	–	–	✓	–	–	✓
Utiliza SNMP	–	–	–	–	–	✓	✓	–	–	–	–	–
Utiliza NETCONF	–	–	–	–	–	–	✓	–	–	–	–	*O*
Utiliza 6LoWPAN	–	–	✓	–	–	–	–	–	–	–	–	*O*
Utiliza SOA	✓	–	–	–	–	–	–	✓	✓	✓	–	–
Gerenciamento dos Dados	✓	✓	✓	–	–	–	–	–	–	–	–	–
Gerenciamento dos Dispositivos	–	–	–	✓	✓	✓	✓	✓	✓	✓	✓	✓
Aborda outros Cenários	–	–	–	–	–	–	✓	✓	–	✓	✓	✓
Define um Modelo de Informação	–	–	–	–	–	–	–	–	–	✓	–	✓
Define um Modelo de Dados	–	–	–	✓	✓	✓	–	–	–	✓	–	✓
Gerenciamento Local	–	–	✓	–	–	✓	–	–	–	✓	–	✓
Gerenciamento Global	–	–	✓	–	–	✓	–	✓	✓	✓	✓	✓
Gerenciamento Remoto	✓	✓	✓	–	✓	✓	–	✓	✓	✓	✓	✓
Gerenciamento por Contexto	–	–	–	–	–	✓	–	–	–	–	✓	✓

dos e aquelas que tratam gerenciamento dos dispositivos da IoT dividem os trabalhos analisados. Três entre as onze soluções tratam especificamente do gerenciamento dos dados coletados pelos dispositivos, enquanto as outras nove soluções (incluindo este trabalho) tratam do gerenciamento dos dispositivos (como definição de parâmetros de funcionamento e configuração). Essa divisão entre os tipos de gerenciamento é impor-

tante para entender as prioridades adotadas pelas soluções diante dos outros requisitos. Por exemplo, no gerenciamento de dados os trabalhos tendem a fornecer soluções especificas de criptografia de dados, tipo de armazenamento (local ou nas nuvens), entre outros.

Os cenários são ambientes onde atuam os dispositivos gerenciados. Algumas soluções foram desenvolvidas para tratar apenas um cenário, como gerenciamento predial ou assistência a saúde. Ao tratar outros cenários, as soluções expandem as possibilidades de criação de novos serviços e uso de novos dispositivos.

Os modelos de dados e modelos de informação são utilizados por um pequeno número de soluções. A definição desses modelos fornecem a estrutura para entender como os dados são tratados e armazenados pela plataforma. A plataforma ManIoT cria um modelo de dados, descrito na Seção 5.2, e um modelo de informação, descrito na Subseção 4.2.4.

O gerenciamento local trata os serviços dentro de um cenário (uma rede local), gerenciando os dispositivos que compõem este cenário, normalmente, utilizando informações de contexto. Já o gerenciamento global procura uniformizar as ações realizadas em múltiplos cenários a partir de diretivas de alto nível. Por isso, essas duas características são importantes para as soluções de gerenciamento para IoT. A plataforma ManIoT tem o gerenciamento local e global dos dispositivos como uma das principais características. Esse gerenciamento é implementado através dos módulos gerente global e gerente local, descritos nas Subseções 4.2.1 e 4.2.2.

O gerenciamento remoto é outra característica das soluções analisadas. O gerenciamento global é realizado de forma remota, mas nem todos os gerenciamentos remotos são gerenciamentos globais. O gerenciamento global tem a capacidade de unir vários cenários e utilizar os dados de um para prover serviços para outros. Por implementar gerenciamento global, ManIoT também implementa gerenciamento remoto.

Por fim, temos o gerenciamento por contexto, provido por soluções que de alguma forma definem uma maneira de coletar entradas capazes de refletir as condições atuais do usuário, do ambiente no qual o mesmo se encontra e do próprio dispositivo computacional utilizado.

3.4 Resumo

Em geral, observamos que o estado da arte trata parcialmente de aspectos como adaptação dinâmica, ciência de contexto e a interação entre redes IoT [Pires et al., 2015]. As soluções analisadas (arquiteturas e plataformas) não apresentam, de forma satisfatória,

um modelo que possa ser expansível para possibilitar a inclusão de novos dispositivos ou novos ambientes aliado a um gerenciamento remoto/global.

As principais características da plataforma ManIoT, como suporte a diversos cenários e gerenciamento por contexto, foram destacadas na Tabela 3.1 da Seção 3.3. Podemos observar que apesar dos autores reconhecerem sua importância (Pires et al. [2015] e Atzori et al. [2010]), a maioria dos trabalhos relacionados não trata esses requisitos.

No próximo capítulo iremos descrever a plataforma de gerenciamento ManIoT, suas camadas e funções, o modelo de gerenciamento local (gerente local) e de gerenciamento global (gerente global). Apresentaremos ainda o modelo de informação que guiou a implementação do protótipo da plataforma.

Capítulo 4

Plataforma ManIoT - Management for the Internet of Things

ManIoT é uma plataforma para gerenciamento de dispositivos em Internet das Coisas. A plataforma ManIoT tem como objetivos integrar e gerenciar as funcionalidades individuais dos dispositivos em uma rede IoT para permitir a criação de novos serviços cientes do contexto, por exemplo, utilizar os dados coletados por sensor de luminosidade para controlar taxa de emissão de luz de uma lâmpada ou controlar uma lâmpada através da localização dos usuários de uma casa. A plataforma realiza gerenciamento local e remoto e é expansível, permitindo a adição de novos tipos de dispositivos. O projeto da plataforma prevê serviços genéricos, tais como descoberta de nós, armazenamento de dados e autenticação, que são blocos básicos para a construção de aplicações IoT.

Neste capítulo descreveremos a plataforma ManIoT. Na Seção 4.1 apresentamos os requisitos para plataforma ManIoT. Na Seção 4.2 definimos os escopos de gerenciamento local e global com suas camadas e componentes. Na Subseção 4.2.4 apresentamos o modelo informação. Por fim, na Seção 4.3 apresentamos uma proposta para gerenciamento multiusuário e multiaplicação e detalhamos o modelo para autorização e para gerenciamento de conflitos.

4.1 Requisitos e Terminologia

A plataforma ManIoT atende a alguns requisitos já conhecidos em Internet das Coisas, como a heterogeneidade, além de outros relacionados aos serviços como gerenciamento local e global. Assim, os principais requisitos para ManIoT são:

- Tratar a Heterogeneidade: ManIoT deve trabalhar com dispositivos de diferentes capacidades de comunicação e de processamento e diferentes protocolos, como sensores e *notebooks*.

- Prover Autenticação e Controle de Acesso: as informações dos dispositivos devem ser protegidas. Assim, a autenticação e controle de acesso das aplicações e usuários são dois dos principais pontos de discussão no âmbito de Internet das Coisas e a plataforma ManIoT deve prover esses serviços.

- Prover Extensibilidade: devido à infinidade de dispositivos existentes, um dos principais requisitos da plataforma ManIoT é fornecer suporte para adicionar novos dispositivos e criar novos cenários.

- Usar Protocolos Conhecidos: é desejável para plataforma ManIoT o uso de tecnologias, protocolos e plataformas existentes e reconhecidas na área de gerenciamento em redes de computadores.

- Realizar o Gerenciamento dos Dispositivos: o gerenciamento dos dados não é função da plataforma ManIoT. ManIoT deve gerenciar os dispositivos, manipulando o estado, as configurações e o modo de operação dos dispositivos que formam a rede IoT.

- Definir um Modelo de Dados: ManIoT deve fazer uso de um modelo de dados bem definido e que encapsule os dados coletados pelos dispositivos implementados.

- Definir um Modelo de Informação: além do modelo de dados, ManIoT deve prover um modelo de informação para os dispositivos suportados. Esse modelo facilitará a programação, pois utilizamos dispositivos de fabricantes diferentes. O modelo de informação é uma representação abstrata/conceitual, enquanto o modelo de dados especifica os detalhes da informação, como tipo de dados.

Para facilitar a compreensão do restante do capítulo, definimos abaixo os principais termos usados na construção do projeto da plataforma ManIoT.

Ambientes Inteligentes: são locais delimitados no espaço, por exemplo, uma casa ou escritório. Esses ambientes envolvem um conjunto de dispositivos da IoT.

Cenários: são domínios de aplicações controlados por um gerente local, onde um conjunto de dispositivos (reais ou virtuais) realizam tarefas para os usuários. Como exemplo de cenários da plataforma ManIoT temos, Iluminação Inteligente, Automação de Tarefas e Tecnologia Assistiva.

Tarefas: são atividades realizadas pelas aplicações. Cada aplicação realiza uma ou mais tarefas, como "verificar localização", "ler o estado dos dispositivos", "controlar luzes da sala".

Aplicações: fazem uma interface de comunicação entre a plataforma e os usuários. As aplicações implementam os cenários e fornecem um conjunto de tarefas para os usuários. Exemplos: aplicação para monitoramento de pessoas, aplicação para gerenciamento de iluminação, entre outras.

Dispositivos: são os nós da rede IoT. Formados por dispositivos físicos, como Lâmpadas, *Smartphones*, *Notebooks*, Sensores, entre outros, e dispositivos virtuais, como calendário *online*, e-mail, agenda *online*, entre outros.

Recursos: são os serviços oferecidos por cada dispositivo. Um dispositivo pode ter um ou mais recursos. Além disso, os recursos podem realizar a função de sensoriamento ou atuação. Por exemplo, o dispositivo *Smartphone* com Android pode ter diversos recursos de sensoriamento (Posição GPS, Sensor de Luminosidade, Sensor Acelerômetro etc.). Em outro exemplo, as lâmpadas inteligentes são atuadores devido à capacidade de alterar a cor e intensidade da iluminação de um ambiente.

Camadas: são subdivisões da plataforma onde definimos um agrupamento de funcionalidades semelhantes. Por exemplo, na camada Aplicação definimos os componentes de *software* que fazem a interface entre a plataforma e os usuários.

Serviços: são as atividades principais da plataforma. Realizam operações mais pontuais quando comparadas com as aplicações. Exemplos de serviços: autenticação, comunicação e gerenciamento de conflitos. Uma aplicação pode utilizar um ou vários serviços, assim como um serviço pode ser utilizado por uma ou mais aplicações.

Usuários: são pessoas que usam ou administram o sistema. Eles podem ser usuários comuns ou usuários administradores. Os usuários comuns são pessoas que utilizam os dispositivos de terceiros através da plataforma ManIoT. Já os administradores são usuários donos e responsáveis por um recurso ou dispositivo. Esse usuário tem controle total e irrestrito sobre seu recurso. Por exemplo, em uma aplicação que gerencia lâmpadas inteligentes em residências, os usuários podem ser todos os moradores e possíveis visitantes.

Ações: um certo recurso pode sofrer ações. Essas ações são baseadas nas restrições impostas. Exemplo ligar/desligar um *switch*. Alterar a cor de emissão de luz de uma lâmpada. Obter a temperatura atual capturada por um sensor, entre outros.

Permissões: são autorizações atribuídas a determinados usuários para executarem uma ou mais tarefas. Por exemplo, uma determinada aplicação do usuário só terá permissão para desligar uma lâmpada quando este usuário obtiver essa permissão.

Modo de Operação: define como os recursos serão gerenciados. Por exemplo, no modo IC - "*Informação de Contexto*" da ManIoT, as permissões serão dadas através das regras que utilizam informações de contexto. Já o modo ADM - "Administrador", as permissões serão delegadas diretamente pelo administrador do recurso.

Restrições: definem quais tipos de ações serão possíveis em um recurso. As restrições são divididas em três categorias. (1) R - "Leitura", ato de ler o estado atual de um determinado recuso. (2) W - "Escrita", ato de gravar uma informação ou alterar uma configuração/estado de um determinado recurso e (3) C - "Controle", ato que permite gerenciar um recurso, atribuindo permissão e alterando o modo de operação. Ex: Leitura - *ler posição do celular android*, Escrita - *gravar novo percentual de iluminação em 90%*, Escrita - *alterar estado da lâmpada para desligado*, Controle - *dar a permissão de escrita para usuário André sobre os recursos do dispositivo WeMo Insight Switch*, Controle - *alterar modo de operação da lâmpada philips hue para IC*.

Conflitos: são situações onde a plataforma ManIoT, diante de duas ou mais opções, precisa decidir sobre uma tarefa. Os conflitos aparecem, por exemplo, quando um usuário deseja apagar uma lâmpada do quarto e outro usuário deseja deixá-la acesa.

4.2 Descrição da Plataforma ManIoT

A plataforma ManIoT estabelece dois escopos de gerenciamento, local e global. O gerente local atua dentro de um cenário, gerenciando os dispositivos que compõem este cenário, geralmente, a partir de informações sobre o contexto. Desta forma, por exemplo, o gerente local pode controlar os eventos que uma aplicação ou usuário pode realizar, como ligar ou desligar uma lâmpada. Já o gerente global procura uniformizar as ações realizadas em diferentes cenários a partir de diretivas de alto nível. Assim,

uma concessionária de energia através do gerente global, por exemplo, poderia definir cotas máximas de consumo por área ou residência em períodos de potenciais *blackouts*.

A plataforma prevê vários gerentes locais em espaços físicos diferentes, mas todos devem se comunicar com apenas um gerente global, conforme apresentado na Figura 4.1. Acreditamos que a separação em dois níveis de gerenciamento traz os seguintes benefícios: (1) podemos implementar políticas de gerenciamento e tomar decisões locais. Cada gerente local trabalha com as características de um cenário especifico sem preocupar com detalhes de outros cenários; (2) o gerente global pode tomar as decisões mais importantes através de informações fornecidas por cada gerente local. O gerente global não deverá se preocupar com os detalhes específicos de cada cenário; (3) através dessa proposta podemos adicionar novos cenários com novos gerentes locais para gerenciar novos dispositivos sem alterar a estrutura e serviços existentes - isso irá prover extensibilidade para a plataforma ManIoT.

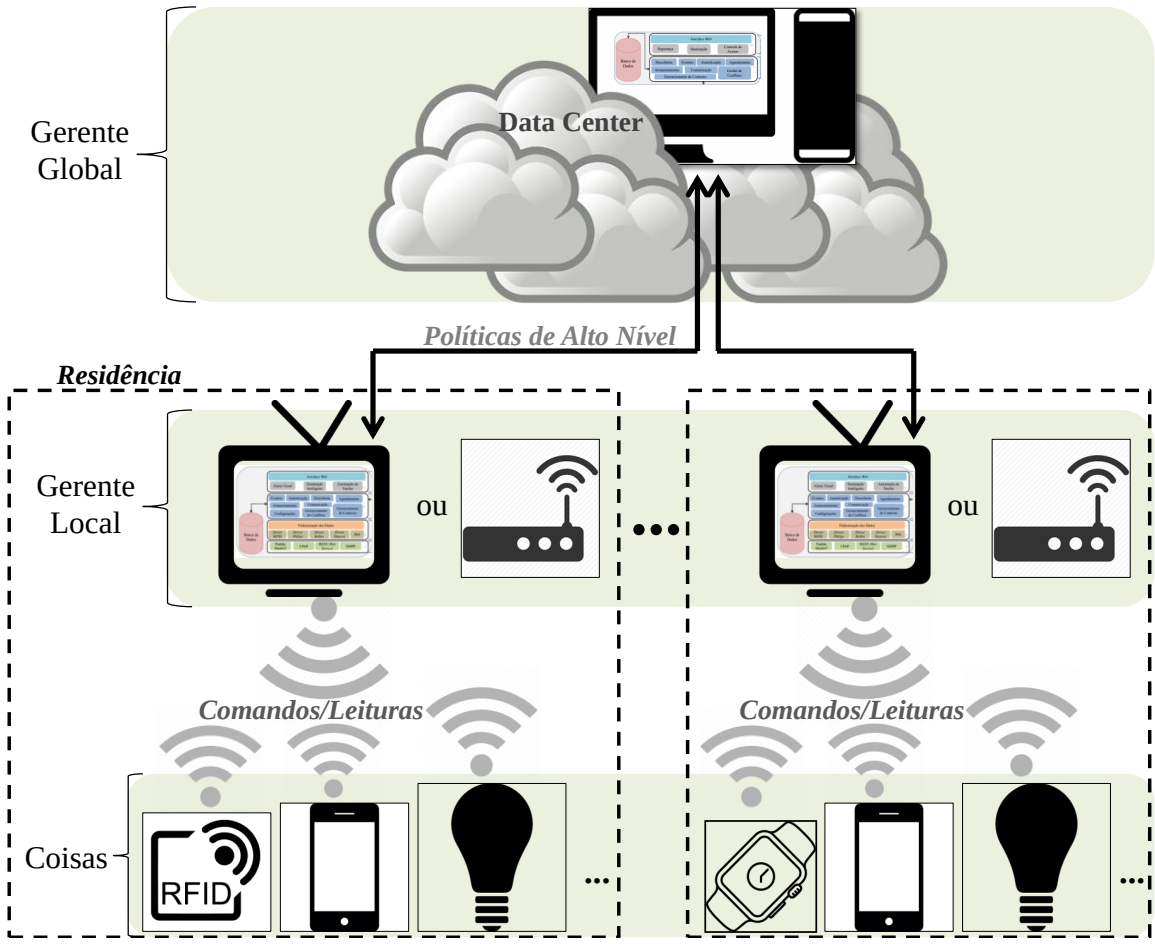

Figura 4.1. Topologia da Plataforma ManIoT, Destacando os Gerentes Global, Locais e os Dispositivos.

A plataforma ManIoT também leva em conta a heterogeneidade dos dispositivos ou coisas. Assim, ManIoT não requer modificações ou instalações de *softwares* adicionais nos dispositivos da rede ou nos dispositivos dos usuários, exceto no uso de *Smartphone*, Seção 5.3. No projeto da plataforma ManIoT o acesso às aplicações é realizado através de uma *interface Web*. O projeto prevê ainda que o acesso pelos usuários deve ser restringido por contas de usuários e por um administrador que define as aplicações e recursos dos dispositivos que esses usuários podem acessar, Seção 4.3.

A plataforma especifica um modelo de dados e um modelo de informação com o objetivo de padronizar o formato dos dados utilizados na comunicação entre aplicações, serviços e dispositivos. O estado dos dispositivos (ligado/desligado) e o id (identificação do dispositivo) são exemplos de características utilizadas no modelo de informação. Ainda, visando a extensibilidade e integração com outros sistemas, o projeto da plataforma prevê o uso de protocolos e padrões populares da indústria para modelos de dados, como o XML (*eXtensible Markup Language*) e o REST.

Cada gerente local roda em um servidor de rede ou dispositivo embarcado de maior poder computacional e sempre ligado à rede elétrica (por exemplo um roteador doméstico, uma televisão) dentro de um cenário e comunica com os dispositivos através da troca de comandos. A comunicação entre o gerente local e o gerente global se dá via conexão TCP/IP e uso de políticas de alto nível, Figura 4.1. O projeto do gerente local para a plataforma ManIoT é uma das principais contribuições deste trabalho. A próxima subseção detalha os componentes do gerente local.

4.2.1 Componentes de Software do Gerente Local

Os *softwares* que compõem a plataforma ManIoT no gerente local, Figura 4.2, possuem cinco camadas bem definidas e são detalhadas abaixo.

Camada de aplicação. A primeira camada é composta pelas aplicações. Cada aplicação usa os dados providos por um ou vários dispositivos, bem como os serviços da plataforma. Os usuários da rede devem acessar as aplicações através de uma *interface web* e essas aplicações, por sua vez, devem interagir com ManIoT utilizando chamadas de função. Cada aplicação solicita à plataforma a execução de ações sobre os dispositivos tendo em vista o cenário implementado. Por exemplo, uma aplicação de gerenciamento de energia solicita desligar ou ligar um aparelho de ar condicionado tendo em vista a redução do consumo.

Camada de serviços. A segunda camada é formada pelos serviços. Os serviços dão suporte às aplicações e utilizam as abstrações implementadas pelos *drivers* para realizar a comunicação com os dispositivos. Entre os itens dessa camada temos:

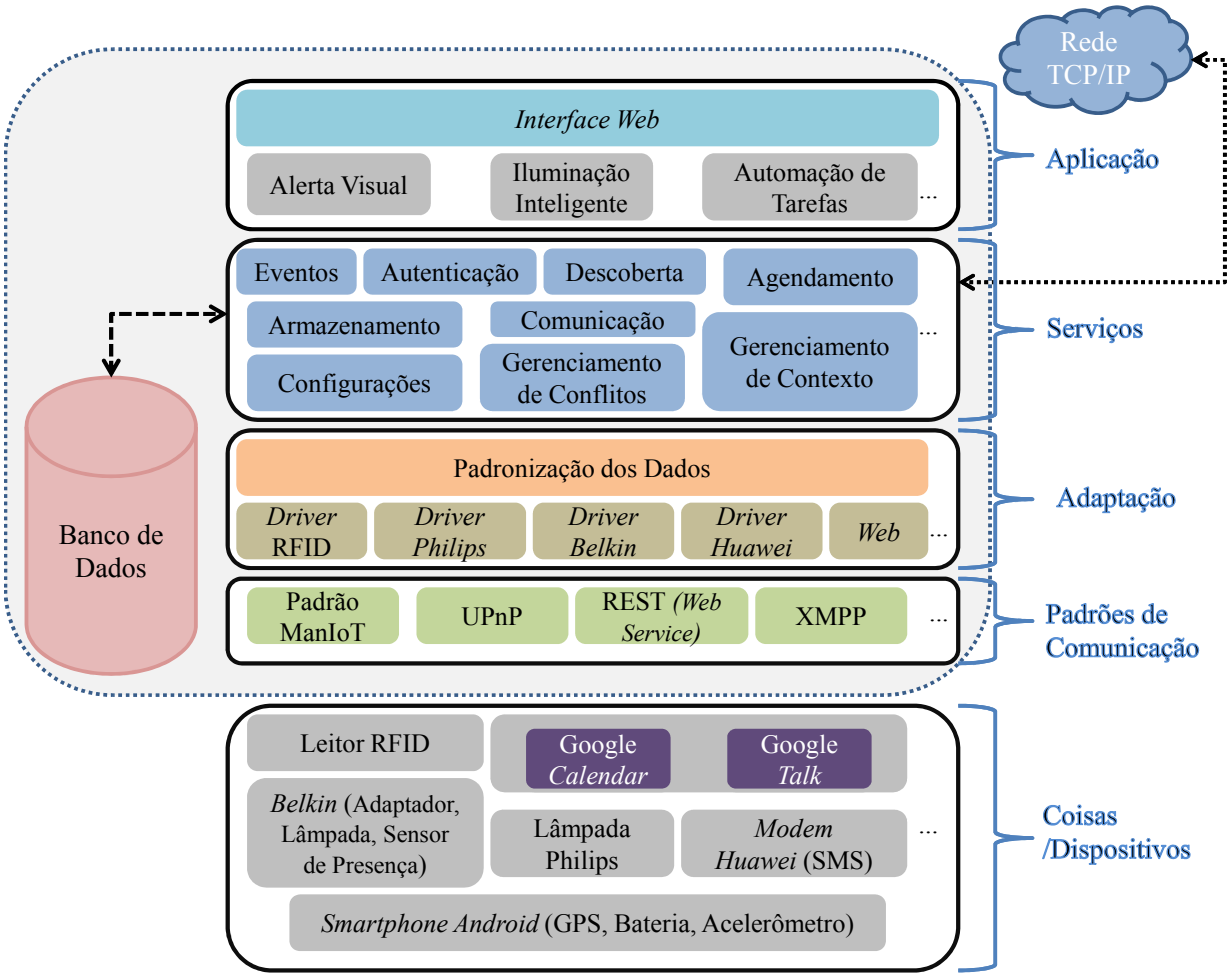

Figura 4.2. Plataforma ManIoT: gerente local.

- **Armazenamento.** Mantém um histórico dos dados coletados, eventos capturados pelos dispositivos e informações sobre as características e funcionalidades de cada dispositivo e de cada cenário;

- **Agendamento.** Programa ações futuras ou que se repetem periodicamente, tais como consultas a sensores ou alterações no estado de um dispositivo;

- **Autenticação.** Realiza a identificação dos usuários (utilizando *login* e senha) e dos dispositivos (utilizando um identificador único);

- **Descoberta.** Realiza a identificação de novos dispositivos (com *drivers* previamente cadastrados) que foram adicionados a rede IoT, identificando também os serviços oferecidos por esses dispositivos;

- **Configurações.** Gerencia as configurações básicas de cada dispositivo e fornece uma interface para as aplicações acessarem essas configurações;
- **Comunicação.** controla a comunicação entre os gerentes locais e o gerente global;
- **Eventos.** Permite que as aplicações sejam notificadas através de eventos disparados pelos dispositivos, como detecção de movimento pelo sensor de presença, condições da temperatura ambiente, localização de um dispositivo, etc;
- **Gerenciamento de conflitos.** Diante de operações conflitantes entre duas aplicações, como "desligar" e "ligar" uma lâmpada, determina qual usuário ou aplicação possui prioridade na execução;
- **Gerenciamento de contexto.** Realiza a percepção de informações contextuais, como localização e tempo (data e hora) para prover funções relevantes aos usuários e outros serviços, como agendamento e gestão de conflitos.

Camada de adaptação. A terceira camada é divida em duas partes, sendo a primeira responsável pela padronização dos dados e a segunda por tratar as especificidades de cada dispositivo - subcamada *driver*. Cada *driver* gerencia um ou uma família de dispositivos compatíveis e abstrai as especificidades do acesso aos seus sensores e atuadores, o que permite o gerenciamento por parte dos serviços de forma integrada. Esta camada e todas as camadas acima utilizam o modelo de informação definido na Seção 4.2.4 e as camadas abaixo utilizam representações próprias de cada protocolo ou dispositivo.

Camada de comunicação. A quarta camada é composta pelos diferentes protocolos de acesso aos dispositivos. Como mencionado anteriormente, a rede será composta por dispositivos que podem utilizar protocolos de aplicação (por exemplo, UPNP ou um protocolo proprietário) e de redes (ZigBee, WiFi) diferentes. Por exemplo, o padrão REST é utilizado pelo "dispositivo Virtual" Google *Calendar*, já dispositivos WeMo Insight Switch, fabricados pela Belkin, utilizam UPnP (*Universal Plug and Play*). Como a plataforma ManIoT faz uso de protocolos que os próprios dispositivos já usam, não são necessárias grandes mudanças na maioria dos dispositivos. Os dispositivos que precisam de mudança são, por exemplo, sensores da plataforma Iris, para definir o formato das leituras e *Smartphone* porque não existe uma interface padronizada para obter dados dos seus sensores.

Camada de coisas/dispositivos. A última camada é formada pelas Coisas. Existem dois tipos diferentes de dispositivos, a saber, os dispositivos reais e os disposi-

tivos virtuais. Os dispositivos reais são sensores e atuadores físicos, por exemplo, uma lâmpada inteligente (atuador), um sensor de pressão (sensor). Já os dispositivos virtuais capturam informações de um servidor conectado a uma rede TCP/IP. Por exemplo, um serviço de calendário ou de *e-mail*, ou um servidor de redes sociais.

4.2.2 Componentes de Software do Gerente Global

O Gerente Global possui duas camadas - Aplicação e Serviços, como mostra a Figura 4.3.

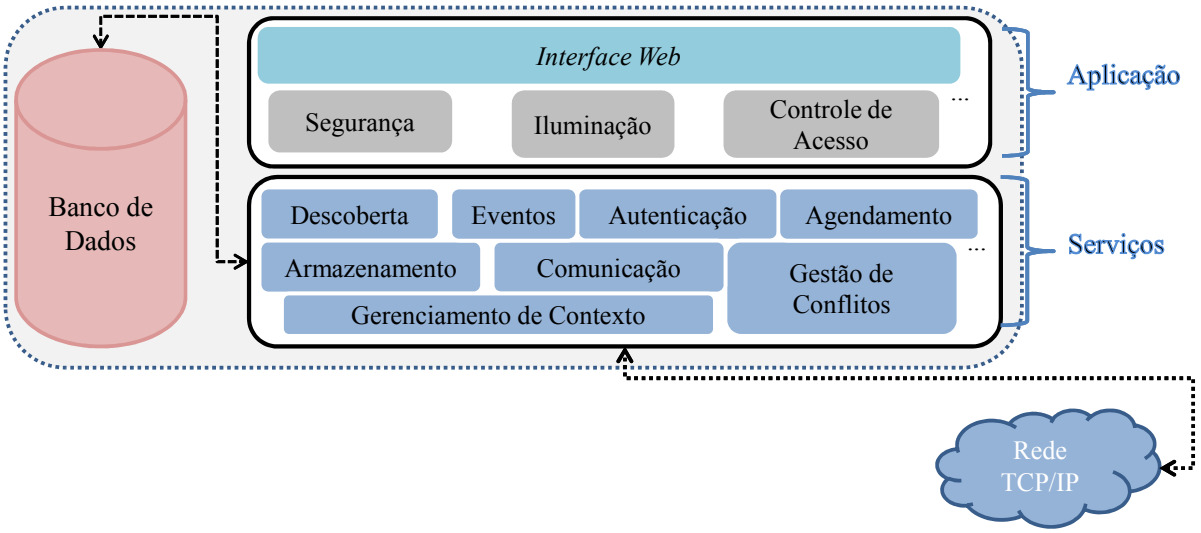

Figura 4.3. Plataforma ManIoT: gerente global.

Os serviços implementados no escopo global possuem as mesmas funções daqueles implementados no escopo local, segunda camada da Figura 4.2 e Figura 4.3 . Os serviços do escopo global tratam conjuntos de dados maiores e fornecem suporte para aplicações mais abrangentes. Por exemplo, no contexto do gerenciamento de energia elétrica o gerente global deve ter a capacidade de gerenciar possíveis faltas de energia em diversas residências de um bairro. As decisões definidas pelos serviços globais são enviadas para os serviços do gerente local que, por sua vez, procura a melhor forma de implementá-las, para tanto empregando uma conexão TCP/IP. Por exemplo, na iminência de um apagão, a companhia que fornece energia elétrica (através do gerente global) pode enviar uma notificação para o gerente local solicitando a redução instantânea do consumo e o gerente local deve, por exemplo, desligar temporariamente aqueles dispositivos que consomem mais energia.

4.2.3 Comunicação entre Gerente Local e Global

O gerente global da plataforma ManIoT tem como objetivo controlar dois ou mais gerentes locais. Para isso definimos nesta subseção a comunicação entre gerente global e gerente local.

De acordo com as Figuras 4.3 e 4.2, a comunicação utiliza o padrão TCP/IP para interligar os gerentes. Além disso, a plataforma define a comunicação em alto nível através de um conjunto de parâmetros de gerenciamento. Esses parâmetros são dependentes da aplicação e através deles os gerentes locais devem notificar o gerente global e este, caso seja necessário, enviar indicativos com novos valores de operação. Assim, cada gerente local deve definir e publicar os seus parâmetros gerenciáveis.

Na rotina de operação, a aplicação do gerente local escreve os valores dos seus parâmetros periodicamente ou quando achar relevante e o gerente global verifica esses valores quando achar necessário. Após a verificação, o gerente global pode enviar indicativos de gerenciamento.

Utilizamos o escopo [<PARÂMETRO>, <VALOR>], onde o parâmetro indica o que deve ser gerenciado, por exemplo "consumo_instantaneo_de_energia", e o valor indica o valor de operação.

Para exemplificar utilizaremos o gerenciamento de gastos com energia elétrica. O gerente local publica dois parâmetros gerenciáveis ["CONSUMO_INSTANT", <VALOR>] e ["CONSUMO_MENSAL_MEDIO", <VALOR>] e através desses parâmetros o gerente global pode reunir dados de vários cenários (fornecidos pelos respectivos gerentes locais) e indicar o consumo (instantâneo e previsão média mensal) desejado. Assim, o gerente local informa o gerente global sobre o consumo mensal médio em kWh ["CONSUMO_MENSAL_MEDIO", 400] e, caso necessite, o gerente global indica que gerente local deve reduzir o consumo em 50 kWh, enviando os dados ["CONSUMO_MENSAL_MEDIO", 350]. Portanto, o gerente local deverá definir formas para chegar a este valor. Em outro exemplo, os gerentes locais podem indicar os gastos instantâneos de cada cenário e, reunindo dados de vários cenários, na eminência de um *blackout*, o gerente global indica os valores desejados para serem implementados pelo gerentes locais.

O gerente global não interfere diretamente nas tarefas desempenhadas pelos gerentes locais, mas apenas fornece indicativos baseados nos parâmetros fornecidos. Assim a comunicação entre os gerentes locais e global utiliza um esquema genérico e extensível para outras aplicações e permite assim a escalabilidade.

Portanto, o ideal é que os parâmetros sejam algo que sumarize as ações do gerente local. Essa sumarização é importante pelo fato que o gerente local pode tratar de

muitos dados e que mudam frequentemente, e se o controle de todos esses dados forem repassados ao gerente global, não conseguimos prover a escalabilidade.

Para guiar a implementação do protótipo da plataforma ManIoT, descrita no Capítulo 5, definimos na próxima subseção o modelo de informação.

4.2.4 Modelo de Informação

Para implementação da plataforma ManIoT foram definidos o modelo de dados e o modelo de informação. O modelo de informação é uma representação conceitual ou abstrata das operações possíveis nos elementos gerenciados. O modelo de dados, definido na Seção 5.2, especifica os dados com detalhes para armazenar e/ou transmitir as informações [Pras & Schoenwaelder, 2003].

A Figura 4.4 apresenta as classes dos componentes e dos dispositivos utilizados no modelo de informação. Empregamos uma modelagem baseada em objetos, onde esses objetos possuem propriedades e métodos. As propriedades de cada objeto são manipuladas pela plataforma ManIoT através de leituras, no caso dos sensores, ou leituras e escritas, no caso de atuadores. Utilizamos ainda no modelo o conceito de herança, onde a classe de um objeto herda as propriedades e métodos de outra classe, que por sua vez generaliza diversas outras classes. Como exemplo de herança, na Figura 4.4 a classe do objeto Sensor de Luminosidade herda o método *Ligar()* da classe Dispositivo.

Na plataforma ManIoT todos os dispositivos possuem um nome, um tipo e propriedades específicas, como *Latitude* e *Longitude* do objeto Localização. Essas propriedades são manipuladas através de métodos que definem as principais operações sobre cada objeto. Através das operações cada aplicação pode definir e modelar seus eventos.

Os eventos são serviços oferecidos pela plataforma ManIoT para as aplicações. Os eventos podem ser entendidos como notificações geradas ou interpretadas pela plataforma. Como exemplo de um evento, sempre que um sensor de luminosidade fizer uma leitura com diferença de 50 *lúmens* para a última leitura, a plataforma deve gerar um evento notificando as lâmpadas para atualização da luminosidade emitida. Em outro exemplo, quando um usuário estiver próximo à sua residência (definido através da localização) a plataforma deve ser notificada (pelo *smartphone*) e gerar eventos como, ligar ar condicionado, ligar cafeteira, desligar sensor de presença, entre outros.

As classes Usuário e Aplicação encapsulam os usuários da rede IoT e das aplicações que utilizam os serviços oferecidos. A classe Usuário possui as propriedades *Nome, Login, Senha e Status* que indica a situação do usuário, ativo ou inativo e os métodos *Cadastrar(), Alterar(), Excluir()* e *AlterarStatus()*. A classe Aplicação possui como

propriedades o *Nome, Descricao* e *Status* que também tem função de indicar se uma aplicação está ou não ativa.

A classe Agendamento envolve os usuários, as aplicações e os dispositivos e agendam operações para serem executadas em data posterior. A classe possui uma propriedade chamada *Data_ e_ Hora*, que indica o tempo que a ação agendada deve ser executada e *Operacao* que indica qual tipo de ação deve ser executada. Os métodos são *Incluir(), Excluir(), Alterar()* e *Consultar()*.

A classe Histórico tem função de encapsular as atividades de leituras realizadas pelos dispositivos da rede. Assim, as propriedades envolvem *Data_ e_ Hora, Dispositivo* e *Operacao*. Para exemplificar uma *Operacao* temos, uma nova leitura realizada pelo sensor luminosidade, a detecção de nova etiqueta, etc. Os métodos são similares àqueles da classe Agendamento. Em outro exemplo, sempre que usuário solicitar os dados de *status* da classe Sensor de Tomada, essas informações devem ser armazenadas no Histórico.

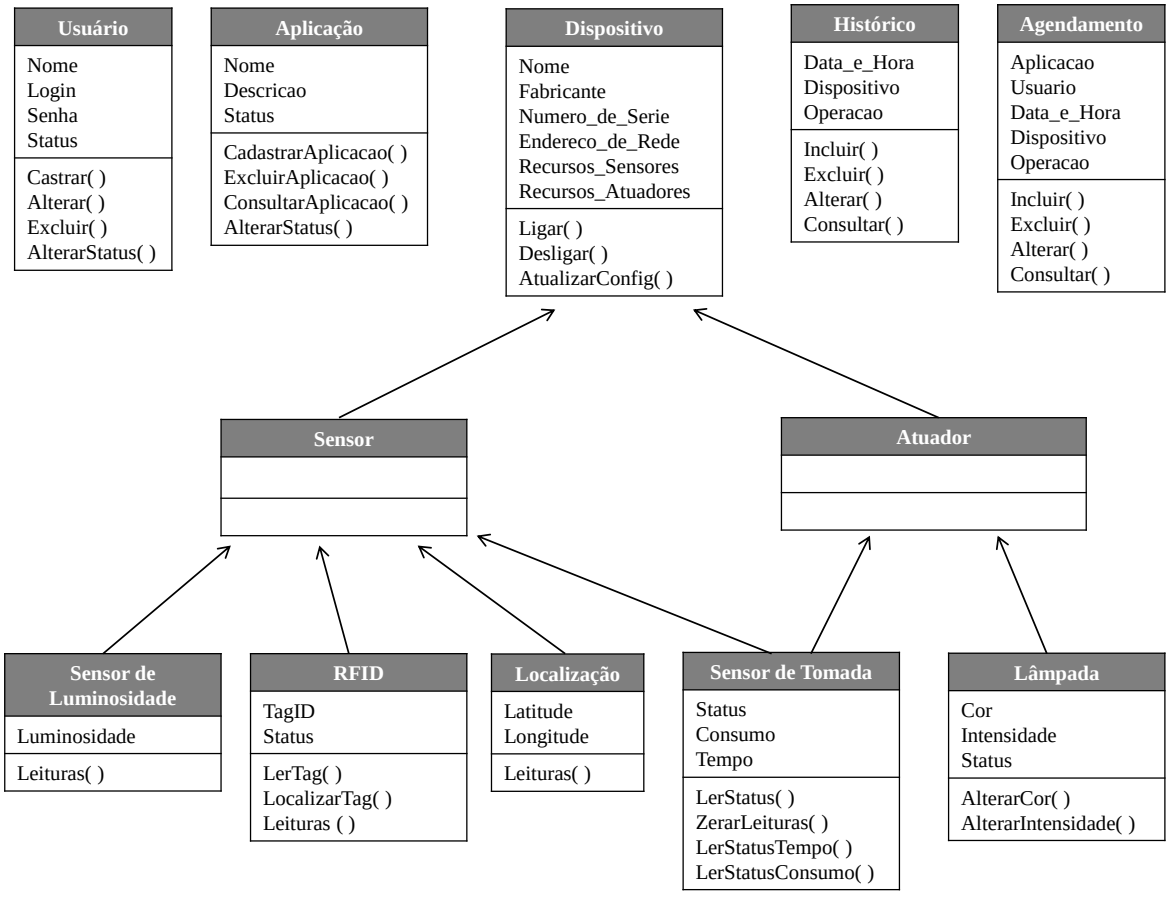

Figura 4.4. Exemplo do Modelo de Informação da Plataforma ManIoT

Os dispositivos são encapsulados pela classe genérica Dispositivo, pelas classes Sensor e Atuador e por suas respectivas classes. Os dispositivos possuem nome, fabricante, número de série, endereço de rede (IP ou outro), lista de recursos do tipo sensores e lista de recursos do tipo atuadores. A classe dos dispositivos possui também os métodos básicos para ligar, desligar e alterar as configurações de um dispositivo. As subclasses genéricas Sensor e Atuador não possuem propriedades ou métodos pois cada dispositivo tem seus próprios parâmetros gerenciáveis, além disso essas duas classes servem para definir o domínio de atuação de cada dispositivo. O modelo de informação é extensível, mas para efeito de implementação, o protótipo ManIoT prevê 5 dispositivos, encapsulados pelas classes:

- Sensor de Luminosidade, que encapsula os sensores de luminosidade, possui a propriedade *Luminosidade*, que armazena os *lúmens* lidos em um ambiente. Possui ainda o método *leitura()* que trata das operações de leitura sobre o dispositivo.

- RFID, com as propriedades *TagID* e *Status* e os métodos *LerTag()*, *LocalizarTag()*, *Leituras()* - que trata da leitura simples dos dados.

- Localização, que encapsula os dispositivos para localização *outdoor*, como *smartphones*. Possui as propriedades *Latitude* e *Longitude* e o método *Leitura()*.

- Sensor de Tomada, que encapsula dispositivos controladores de tomadas e funções de ler dados de consumo elétrico, ativar e desativar seu uso. As propriedades são *Status*, *Consumo* e *Tempo* e os métodos *LerStatus(), ZerarLeituras(), LerStatusTempo()* e *LerStatusConsumo()*.

- Lâmpada, que encapsula as lâmpadas inteligentes com capacidade de gerenciamento remoto. As propriedades são *Cor* (para alterar a cor de emissão de luz) *Intensidade* (de emissão de luz) e *Status*. Os métodos são *AlterarCor()* e *AlterarIntensidade()*.

De forma geral um dispositivo tem (ou pode ter) um conjunto de sensores e atuadores que são encapsulados pelas classes genéricas sensor e atuador, respectivamente. Dessa forma, no protótipo temos os sensores de presença, luminosidade, corrente elétrica, tensão e localização (*indoor* e *outdoor*). E os atuadores de iluminação, intensidade de luz e corrente elétrica.

Para exemplificar temos o dispositivo Controlador WeMo Insight Switch que faz uso da classe Sensor de Tomada e atua como sensor e atuador. Esse dispositivo tem capacidade de ler o consumo dos equipamentos conectados e apresentar informações, como tempo de uso e energia gasta. Agindo como atuador o controlador tem capacidade

de ligar e desligar a energia dos equipamentos conectados. Todas essas operações são encapsuladas pelos métodos *Ler()*, *LerStatusTempo()* e *LerStatusConsumo()* da classe especialista Sensor de Tomada.

Devido à grande quantidade de dispositivos que compõem a IoT, as plataformas de gerenciamento devem prever a inclusão de novos componentes [Pires et al., 2015]. Podemos observar que as características gerais, como *Nome* e *Tipo* e as operações *Ler()*, *Ligar()* e *Desligar()* dos dispositivos da IoT são previstas. Além das operações temos as classes de uso geral, como Aplicação e Agendamento que não restringem o modelo do dispositivo. Portanto o modelo de informação da plataforma ManIoT é extensível, pois aceita a inclusão de novos dispositivos com novas características e serve de base para o modelo do banco de dados.

Além do modelo de informação, uma plataforma para IoT é composta por aplicações, pelos dispositivos e pelos usuários. Devido à pluralidade desses componentes, podemos dizer que a essência do IoT é ser multi: multiaplicações, multi-hardware e multiusuários. Na próxima seção trataremos do projeto para gerenciamento multiusuário e multiaplicação.

4.3 Gerenciamento Multiusuário e Multiaplicação

De acordo com IBSG (*Internet Business Solutions Group*) [IBSG-Cisco, 2011], existe a previsão para cerca 50 bilhões de dispositivos compondo a IoT em 2020. Diante do grande número de dispositivos e respectivos usuários conectados à Internet, as plataformas para IoT precisam implementar serviços para gerenciar as necessidades desses usuários, principalmente aquelas que geram conflitos. Além disso, de acordo com Atzori et al. [2010] os componentes básicos de uma plataforma para IoT são as aplicações, os serviços e os objetos (dispositivos). Devido à pluralidade desses componentes, podemos dizer que a essência do IoT é ser multi: multi-hardware, multiusuários e multiaplicações. As aplicações utilizam os dispositivos através dos serviços disponíveis e essas aplicações são utilizadas por diversos usuários com necessidades diferentes. Nesta seção apresentamos uma proposta desenvolvida para gerenciamento de múltiplos usuários e múltiplas aplicações para plataforma ManIoT.

Os conflitos aparecem, por exemplo, quando um usuário deseja apagar uma lâmpada do quarto e outro usuário deseja deixá-la acesa. Utilizando como exemplo as lâmpadas inteligentes, onde podemos executar operações como ligar/desligar e mudar cor e intensidade remotamente, é necessário definir quem poderá executar essas operações conflitantes. Em outro exemplo, o conflito aparece quando uma aplicação

de economia de energia deseja desligar uma lâmpada e outra aplicação, de segurança, deseja deixá-la acesa. Nesses exemplos quando a plataforma executar a operação solicitada por uma entidade (usuário ou aplicação) ela irá deixar de atender outra, gerando portanto um conflito. Com isso existe a necessidade de definir quem terá prioridade, quais regras serão utilizadas para definir essas prioridades e quem será o intermediador desses conflitos.

As plataformas encontradas na literatura possuem diversos desafios ainda abertos e que requerem soluções em nível de *hardware* e de *software*. As plataformas existentes não proveem mecanismos que permitam o uso harmonioso dos dispositivos por múltiplos usuários e múltiplas aplicações. Certas plataformas, como EcoDiF [Delicato et al., 2013], implementam módulos que permitem aos usuários realizar o gerenciamento dos dispositivos conectados, mas restringe o acesso, liberando apenas aqueles dispositivos criados pelo próprio usuário. Outros modelos de plataformas permitem a realização de buscas por dados de dispositivos e aplicações através da interface *Web*, mas restringem as operações que alteram o modo de funcionamento dos dispositivos. Outras soluções, como Xively [LogMeIn, 2015] e Carriots [Carriots, 2015] utilizam serviços de nuvem para gerenciar os dados providos pelos dispositivos. Nesse tipo de plataforma os usuários podem visualizar os dados e seus históricos mas não podem realizar operações de controle mais avançadas, como por exemplo, alterar a taxa de leitura de um sensor.

Uma das políticas usadas para gerenciar ambientes com recursos compartilhados é criar uma ordem de prioridade para determinadas tarefas. A ordem de prioridade tem como objetivo eleger um usuário ou aplicação que deverá executar uma operação em um cenário onde existam conflitos de interesses, como desligar ou ligar um dispositivo. É importante definir então quem poderá executar determinada ação e qual o momento de início e a duração dessa ação. A falta de uma política de gerenciamento pode levar o sistema a situações de inconsistência, como desligar um sensor de presença para economizar energia em um momento de risco iminente.

Na próxima seção relacionamos os requisitos para o gerenciamento proposto neste capítulo. Apresentamos também os modelos de autorização e gerenciamento de conflitos criados.

4.3.1 Modelo de Autorização e Gerenciamento de Conflitos

A plataforma ManIoT fornece serviços para usuários através de aplicações que, por sua vez, utilizam os recursos dos dispositivos da rede. Esses recursos são compartilhados entre as aplicações e os respectivos usuários. Devido ao uso compartilhado existe a necessidade de gerenciar as permissões para executar ações, como ligar ou desligar um

serviço. Nesse sentido podemos identificar dois domínios que devem ser gerenciados: usuários e aplicações.

As aplicações habilitam um conjunto de tarefas para os usuário. Assim, todos os usuários da plataforma utilizam uma ou mais aplicações que por sua vez executam serviços e ações programadas, normalmente em monitoramento de ambientes, como sensores de presença, alarme, iluminação, câmeras para aplicações de segurança. Como exemplo, uma ação programada pode acender uma lâmpada quando o sensor de presença detectar um movimento.

O gerenciamento requer certos requisitos. O primeiro requisito para gerenciamento de usuários é a criação de um perfil, onde cada usuário deve ter um *login* e uma senha. Através desse perfil o usuário poderá acessar a plataforma de qualquer local, via interface *Web*, e executar as aplicações para as quais possuem permissão. O segundo requisito é que cada usuário deve ter mapeado em seu perfil o tipo de permissão para cada recurso. Por questões de segurança, o valor padrão deve ser "sem permissão". O terceiro requisito diz que, além da autenticação e controle de acesso, cada usuário terá uma prioridade na execução de ações sobre os recursos dos dispositivos. Essa prioridade deve escalonar a execução e definir a vigência de cada ação, mais detalhes na Subseção 4.3.1. Por fim, cada aplicação deve indicar quais ações e privilégios necessita para executar. Com isso, se determinado usuário não tiver todos os pré-requisitos, automaticamente ele não terá permissão sobre tal aplicação.

As permissões são autorizações atribuídas a determinados usuários para executarem uma ou mais tarefas. As Leituras são, a principio, operações permitidas a todos os usuários da plataforma, mas a revogação dessa permissão pode ser feita a qualquer momento pelo Administrador do Recurso. Portanto devemos gerenciar os usuários e suas aplicações para fazer o uso coordenado e eficiente dos recursos mantidos pela plataforma. A Figura 4.5 apresenta o modelo exemplo, onde um usuário tem permissão de administrador sobre os quatro recursos enquanto os outros usuários possuem permissões restritas. A marcação e as linhas vermelhas da figura representam as permissões de administrador do recurso enquanto as linhas verdes representam os usuários com permissões restritas, ou seja, aqueles que não tem controle sobre os recursos.

Para realizar o gerenciamento de conflitos, cada recurso de um dispositivo possui um usuário administrador. Os administradores possuem controle total e irrestrito sobre seu recurso, podendo delegar permissões a outros usuários de forma permanente, ou temporária, baseado no contexto (como localização do usuário). Pensando na implementação, a localização poderia ser fornecida por dispositivos como leitores RFID, GPS em *smartphones* ou sensores de presença. Tomemos como exemplo uma lâmpada inteligente na porta de uma casa: Alice, um usuário comum, não poderia desligar a

lâmpada do quarto de Maria que é outro usuário comum, se Maria estiver no seu quarto. Já Joana, a mãe e administradora da lâmpada, poderia desligar ou ligar a lâmpada a qualquer momento, sobrescrevendo as ações de Maria e Alice.

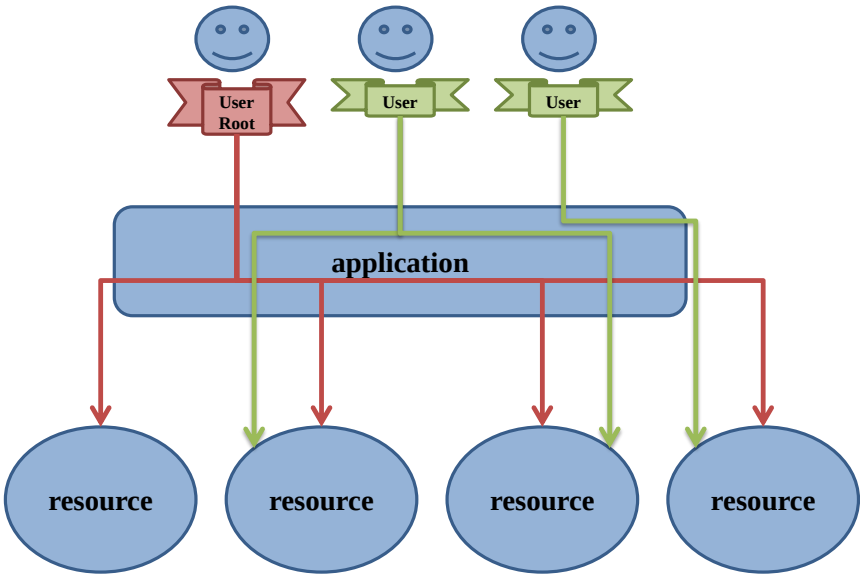

Figura 4.5. Relação entre usuários, aplicações e recursos da plataforma ManIoT.

Para tratar a restrição mostrada na Figura 4.5 precisamos implementar uma forma de delegar essas permissões a outros usuários. O administrador do recurso atribui a permissão de escrita ou leitura do seu recurso a outro usuário utilizando critérios como localização ou prioridade. Essa atribuição deve seguir um dos formatos:

TF - Tempo Fixo: (*Ex: TF 1, 10, 30 ou 60*) onde a permissão expira automaticamente após determinado tempo pré-fixado em 1, 10, 30 ou 60 minutos.

TE - Tempo Especificado: (*Ex: TE "Value"*) onde, a partir do momento da solicitação, o administrador do recurso especifica o tempo total da permissão em minutos.

TP - Tempo Programado: (*Ex: TP "Timeline"*) onde o administrador do recurso pode criar um cronograma de permissões para um usuário específico executar determinada ação sobre um recurso. Por exemplo, "*O usuário João poderá desligar o sensor de presença somente aos domingos no período da manhã*".

TI - Tempo Indeterminado: (TI) A Permissão não terá um limite e somente o administrador poderá revogá-la.

Alguns dispositivos, como lâmpadas, que atuam em pequenos ambientes, como uma sala ou quarto, poderão ser gerenciados através das informações de contexto. Essas informações são, a princípio, a Localização e/ou Data/Hora. O administrador do recurso deve permitir ou definir que seu recurso seja gerenciado através das informações de contexto - IC (*Informação de Contexto*). Portanto um recurso pode estar em modo IC ou modo ADM - Administrador. As regras para implementação do modo IC serão especificadas por e para cada aplicação. Por exemplo, "O usuário que tiver mais próximo da lâmpada poderá ligar/desligar ou alterar sua intensidade". Em outro exemplo, a regra de contexto pode dar permissão total sobre uma lâmpada para um usuário que utiliza determinada aplicação de segurança quando o horário estiver entre 00h:00min e 06h:00min.

4.3.1.1 Algoritmo para Gerenciamento de Conflitos

Um aspecto importante em uma rede com múltiplos serviços se refere ao gerenciamento de conflitos. Os conflitos podem ocorrer quando duas ou mais aplicações tentam alterar a mesma configuração de um recurso. Por exemplo, quando uma aplicação solicita o desligamento de uma lâmpada, e uma segunda aplicação solicita que esta fique ligada. Este gerenciamento é implementado pelos serviços gerenciamento de conflitos e gerenciamento de contexto.

Algorithm 1 Gerenciamento de operações conflitantes no ManIoT.

```
1:  procedure Conflito(Recurso, Operacao, idUsuarios[])
2:      idUsuario ← 0
3:      if Operacao = IC then idUsuario ← BuscarMaisProximo(idUsuarios)
4:          return idUsuário
5:      else
6:          for i ← 1, n do
7:              if idUsuarios[i].Tipo = Administrador then
8:                  idUsuario ← idUsuarios[i]
9:                  return idUsuario
10:     if idUsuario.Tipo ≠ Administrador then
11:         idUsuario ← BuscarPermissao(idUsuarios, Recurso)
12:         return idUsuario
```

O Algoritmo 1 descreve a operação do serviço de gerenciamento de conflitos. O serviço recebe a lista de usuários que enviaram comandos ao dispositivo, verifica se o dispositivo (que contém o recurso) está habilitado para ser gerenciado no modo IC

e retorna o usuário mais apto ($l.3$). Se o modo IC não estiver habilitado, a lista de usuários candidatos deverá ser verificada e escolhido aquele que possui permissão de administrador sobre o recurso ($l.7$). Por fim, se não existir um administrador entre os candidatos, cada usuário candidato poderá solicitar essa permissão diretamente ao administrador do recurso ($l.11$).

A ordem de prioridade sobre os recursos tem o objetivo de definir um único usuário para executar a operação de escrita. O administrador do recurso é o único que tem a permissão de controle, e essa permissão pode delegar e revogar as permissões de leitura e escrita. Consequentemente o administrador é aquele que tem maior prioridade. O usuário administrador pode alterar ainda seu recurso para o modo de operação IC e assim as regras preestabelecidas irão controlar as permissões de escrita, podendo delegar escrita para um usuário que atenda os critérios da aplicação.

As regras que utilizam IC terão, neste caso, o papel de administrador para delegar e revogar a permissão de escrita. Quando o usuário administrador habilita o modo IC para determinado recurso, as permissões delegadas serão substituídas pelas regras do IC, portanto permissões delegadas tem menor prioridade.

Como definição de projeto, a qualquer momento o usuário administrador pode alterar entre o modo de operação entre IC e ADM. Caso o administrador queira dar permissão a um determinado usuário e o seu recurso está no modo IC, este deve primeiro alterar o modo de operação para ADM e logo depois delegar escrita. Portanto, de forma geral a ordem de prioridade entre os usuário é a seguinte:

1. (Maior) - Usuário com Permissão de Controle - Administrador.
2. - Usuário com Permissão Delegada por IC.
3. (Menor) - Usuário com Permissão Delegada por ADM.

O contexto é utilizado para autorizar a execução de determinada aplicação. As informações de localização (*indoor* - ex: dentro de uma sala, quarto, escritório; *outdoor* - ex: no trabalho, em um bairro, próximo a residência) indicam se determinado usuário poderá executar uma ação, como desligar um sensor de presença ou ligar uma lâmpada.

O gerenciamento de contexto na plataforma ManIoT utiliza a localização, data e hora. Através de leitores RFID para localização *indoor* e coordenadas GPS para localização *outdoor* a plataforma ManIoT define a posição do usuário e pode autorizar ou não uma determinada ação. As informações de data e hora são utilizadas para definir ações agendadas ou ações com horário de execução pré-definidos.

O exemplo a seguir ilustra o uso de dados do contexto. Em uma residência moram João e Maria. Eles possuem em seus respectivos quartos lâmpadas inteligentes. Eles

são responsáveis pelo gerenciamento (como ligar e desligar). Quando Maria não está em casa João recebe a permissão para executar as ações através do contexto localização. Essas operações são possíveis porque os donos dos recursos (João e Maria) habilitaram o modo de gerenciamento IC.

Na próxima subseção detalhamos modelagem das permissões. Apresentamos ainda alguns exemplos utilizando usuários.

4.3.2 Modelagem das Permissões

Para chegarmos a um gerenciamento eficiente as regras devem ser claras e todos os usuários devem conhecê-las. Uma das tarefas da modelagem do problema é abstrair certas funcionalidades e apresentar o sistema de forma simples, permitindo visualizar o papel de cada usuário sem duplo entendimento. Como forma de visualizar as permissões de Leitura, Escrita e Controle, criamos uma matriz chamada Matriz de Permissões, onde as constantes NP, RD, WR e CR representam Nenhuma Permissão, Permissão de Leitura, Permissão de Escrita e Permissão de Controle, respectivamente. Assim através do exemplo mostrado na Tabela 4.1 podemos visualizar quem tem determinadas permissões e verificar inconsistências, como dois administradores para o mesmo recurso.

Tabela 4.1. Matriz de Permissões (NP - nenhuma permissão; RD - permissão de leitura; WR - permissão de escrita; CR - permissão de controle)

Recursos/ Usuários	Lâmpadas	Luminosidade	Proximidade	Sensor WeMo	RFID
João	RD	NP	RD	CR	WR
Ana	WR	CR	RD	RD	RD
Carlos	RD	RD	CR	NP	RD
Pedro	RD	NP	NP	WR	CR

4.4 Resumo

Neste capítulo descrevemos a plataforma ManIoT. Na Seção 4.1 definimos os requisitos para a plataforma ManIoT, como tratar a heterogeneidade, definir um modelo de dados e de informação e realizar o gerenciamento local e global. Em seguida definimos cada componente do gerente global e do gerente local. Nos componentes que formam o gerente local destacamos a camada de Serviços e a camada Adaptação que fornece abstração entre os dispositivos, serviços e aplicações da plataforma ManIoT.

Na Subseção, 4.2.4, definimos o modelo de informação. Criamos, através do modelo uma representação abstrata das operações que podem ser realizadas nos elementos gerenciados. Além das operações o modelo define quais características e quais dados dos dispositivos são tratadas pelas aplicações.

Apresentamos também uma proposta desenvolvida para gerenciamento de múltiplos usuários e múltiplas aplicações. Definimos as terminologias usadas, requisitos e modelos para autorização e gerenciamento de conflitos. Relacionamos as formas de delegar as permissões a outros usuários. E apresentamos um algoritmo para gerenciamento de conflitos que prevê o uso de dados de contexto.

Por questão de foco e para sintetizar o trabalho, não iremos implementar o gerenciamento multiusuário e multiaplicação no protótipo. Mas pretendemos realizar essa implementação em melhorias futuras da plataforma. Portanto, o conteúdo da Seção 4.3 ficará no nível de projeto da plataforma ManIoT.

No próximo capítulo iremos discutir a modelagem e implementação do protótipo da plataforma ManIoT. Descreveremos as ferramentas, plataformas e abordagens utilizadas no desenvolvimento do protótipo além do modelo de dados, dispositivos e cenários utilizados.

Capítulo 5

Implementação do Protótipo

O desenvolvimento de um protótipo teve como objetivo final avaliar determinadas funcionalidades da plataforma proposta. O protótipo da plataforma ManIoT permite a realização de testes de consumo de memória, CPU e banda da rede além da disponibilidade e confiabilidade.

Neste capítulo iremos descrever a implementação do protótipo da plataforma ManIoT. A Seção 5.1 descreve as ferramentas, plataformas e abordagens utilizadas no desenvolvimento do protótipo. Na Seção 5.2 definimos o modelo de dados. Na Seção 5.3 descrevemos os dispositivos utilizados na implementação dos cenários. Na Seção 5.4 apresentamos a comunicação entre os dispositivos que formam a plataforma ManIoT. Por fim, na Seção 5.5 detalhamos os cenários implementados e tratados pelo protótipo.

5.1 Descrição da Implementação

Para avaliar a aplicabilidade dos mecanismos de gerenciamento, implementamos um protótipo do gerente local da plataforma com alguns estudos de caso e uso de dispositivos reais. Devido à grande heterogeneidade dos dispositivos em IoT, apenas as funcionalidades necessárias para os estudos de caso foram implementadas, blocos destacados na Figura 5.1. Junto com o protótipo, implementamos, ainda, as aplicações para testar e validar a plataforma ManIoT. Essas aplicações fornecem serviços diretamente aos usuários. O gerente global deverá ser implementado em trabalhos futuros.

Procuramos utilizar ferramentas, linguagens e adotar padrões conhecidos na indústria e na academia. Utilizamos o sistema operacional CentOS[1], instalado no laboratório Winet[2] do Departamento de Ciência da Computação da Universidade Federal

[1]CentOS - Disponível em https://www.centos.org/

[2]Winet - Wireless Networking - UFMG - www.winet.dcc.ufmg.br/

de Minas Gerais. Os componentes para comunicação com os dispositivos foram desenvolvidos em linguagem Java, bem como as aplicações e módulos de padronização e armazenagem de dados.

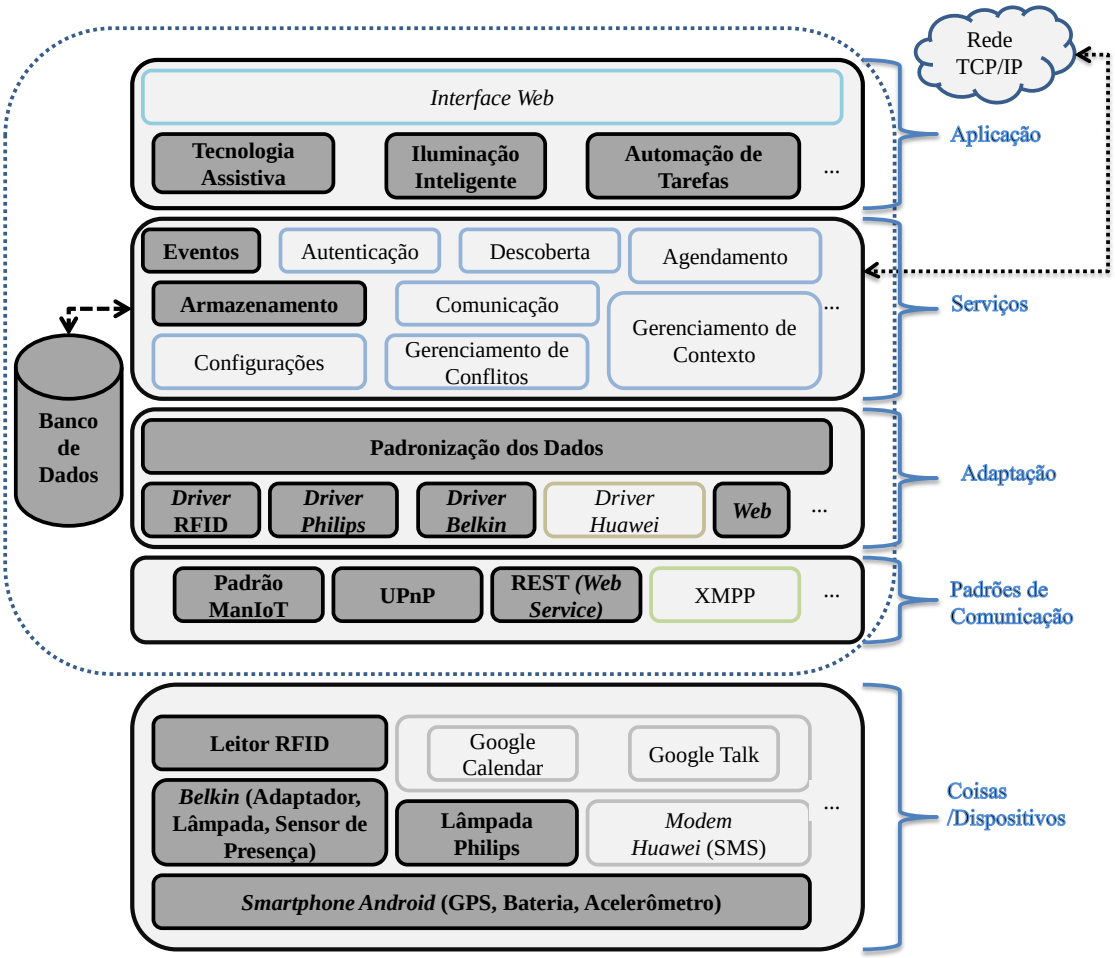

Figura 5.1. Gerente Local da Plataforma ManIoT com Destaque para os Componentes Implementados no Protótipo.

No gerente local armazenamos informações sobre o ambiente, sobre os dispositivos, além dos dados coletados pelos dispositivos. Utilizamos o banco de dados MySQL[3] por ser um dos sistemas de gerenciamento de banco de dados mais usados e pelas seguintes características: (1) alta compatibilidade com diversas linguagens, inclusive com o Java. (2) baixa exigência de processamento quando comparado com outros bancos como PostgreSQL[4]. (3) instruções em SQL (*Structured Query Language*). (4) e por fim, seu uso é gratuito.

[3]MySQL - Disponível em https://www.mysql.com/
[4]PostgreSQL - Disponível em www.postgresql.org/

Para o desenvolvimento do protótipo utilizamos a abordagem de programação baseada em eventos. Analisando as características dos dispositivos e considerando o fato que grande parte das ações realizadas pelas aplicações é motivada por eventos que acontecem em determinado dispositivo, esta abordagem é aquela que melhor se adapta.

A comunicação dos *drivers* com as aplicações foi realizada através do banco de dados ManIoT. Entre os componentes de um mesmo *driver* utilizamos chamada de métodos das classes Java.

5.2 Modelo de Dados Para Armazenamento e Comunicação

Além do modelo de informação, Subseção 4.2.4, definimos o modelo de dados para implementação da plataforma ManIoT. O modelo de dados especifica os dados com detalhes suficientes para armazenar e/ou transmitir as informações [Pras & Schoenwaelder, 2003]. Assim, o modelo de dados foi criado através de instâncias do modelo de informação, tendo como resultado as tabelas criadas no banco de dados MySQL.

A Figura 5.2 apresenta parte do modelo de dados para os dispositivos Lâmpadas Philips Hue, Sensor de Tomada WeMo Insight Switch e Sensor de Luminosidade na Plataforma Iris. Assim, é possível modelar e criar uma *interface* comum para permitir que as aplicações da plataforma possam coletar as informações fornecidas pelos dispositivos, armazenar essas informações nas bases de dados e executar o controle sobre os respectivos dispositivos gerenciados.

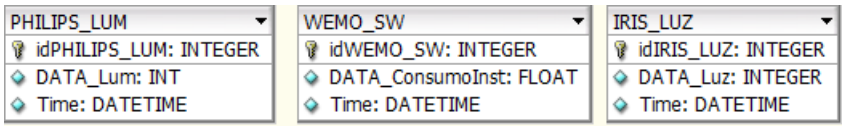

Figura 5.2. Exemplo do Modelo de Dados da Plataforma ManIoT

Na próxima seção descrevemos os dispositivos que foram utilizados nos testes do protótipo.

5.3 Dispositivos Utilizados

O protótipo ManIoT foi implementado em uma máquina desktop rodando um sistema operacional Linux CentOS 7, processador i3-4160 com 3.60GHz e 16GB de memória RAM. Foram utilizados ainda cinco dispositivos, são eles:

Lâmpadas Philips Hue: dispositivos que, através de acesso remoto, podem ligar/desligar, alterar a intensidade de emissão de luz (*lúmen*) e mudar a cor de emissão de luz seguindo o padrão RGB (*Red Green Blue*). A Figura 5.3 apresenta o *kit* utilizado na implementação. Esse *kit* é composto por três lâmpadas e um *gateway* Ethernet que comunica com as lâmpadas via ZigBee *Light Link*.

Figura 5.3. Kit - Lâmpadas Inteligentes Philips Hue

Controlador WeMo Insight Switch: dispositivo que, através de acesso remoto, tem a funcionalidade de ligar/desligar e medir o consumo de energia dos dispositivos conectados. A Figura 5.4 apresenta o dispositivo utilizado.

Figura 5.4. Dispositivo Controlador de Tomada WeMo Insight Switch

Sensor de Luminosidade na Plataforma Iris: também possui a funcionalidade de medir a iluminação disponível em um ambiente. A Figura 5.5 apresenta um nó da plataforma Iris Mote. Esse nó é genérico e possui um barramento onde é possível acoplar diversos sensores. Neste caso utilizamos a placa sensor de luminosidade modelo MDA 100.

Figura 5.5. Dispositivo Plataforma Iris Mote

Leitor RFID Alien 9900: dispositivo que identifica as Etiquetas RFID em um determinado cenário. A Figura 5.6 apresenta um *kit* com um Leitor, duas Antenas e algumas etiquetas.

Tablet para Localização GPS e Sensoriamento de Luminosidade: através do sistema operacional Android, provê funcionalidade de localizar um dispositivo. Já o Sensoriamento de Luminosidade tem a funcionalidade de medir a iluminação disponível em um ambiente. A Figura 5.7 apresenta o modelo utilizado, Samsung Galaxy Tab 2 7.0.

Os dispositivos escolhidos foram selecionados devido às seguintes características: popularidade nas implementações para IoT, como RFID; dispositivos que usam diferentes padrões de comunicação, como ZigBee e UPnP; e dispositivo com funções distintas, como sensoriamento, iluminação e identificação. Assim, acreditamos que esses cinco dispositivos representam uma ampla classe dos dispositivos que compõem a IoT. A comunicação entre eles e a plataforma ManIoT é descrita na próxima seção.

Figura 5.6. Kit - Dispositivo RFID Alien 9900

Figura 5.7. Dispositivo Tablet Samsung Galaxy Tab 2 7.0

5.4 Comunicação entre o Protótipo e os Dispositivos

Os componentes implementados no protótipo foram: os protocolos de comunicação (REST, UPnP, entre outros); os *drivers* da camada de adaptação, responsáveis pela

comunicação com os recursos dos dispositivos utilizados; o banco de dados local; e os serviços, como armazenamento e eventos.

A comunicação entre os dispositivos é realizada através de inserções e consultas periódicas aos dados no banco de dados ManIoT. Assim, quando o protótipo, através do dispositivo que faz sensoriamento da luminosidade, inserir um novo valor no banco de dados, outros dispositivos (por exemplo, lâmpadas) podem receber esse valor através da plataforma e usá-lo para definir seus parâmetros de operação.

Cada dispositivo possui um *driver* específico que trata a comunicação (envio e recebimento de dados e informações de controle). Os *drivers* dos dispositivos, desenvolvidos na linguagem Java, fornecem serviços para outras camadas da plataforma. Como exemplo, o *driver* do sensor de luminosidade recebe e envia para o serviço de armazenamento o valor lido em um determinado ambiente e o *driver* da lâmpada inteligente recebe esse valor e utiliza para gerenciar a intensidade de emissão de luz. Para cada dispositivo novo, com novas características, a plataforma requer o desenvolvimento de um *driver* específico. Entre os padrões suportados pelos *drivers* implementados estão UPnP e REST. Além disso, implementamos *drivers* para protocolos proprietários, como o protocolo de comunicação com o dispositivo RFID.

O banco de dados do protótipo ManIoT armazena os dados dos eventos dos dispositivos. Utilizamos a ferramenta DBDesign[5] 4 para modelagem. As tabelas projetadas armazenam informações sobre Usuários, Aplicações, Dispositivos, Recursos e Permissões. As tabelas 5.1 e 5.2 definem os campos para armazenar as informações dos dados coletados pelos recursos dos dispositivos. Na Tabela 5.1 o campo idRECURSO identifica cada recurso de forma única, os campos idCOISA e idADM identificam os dispositivos e o administrador do recurso, respectivamente. Temos ainda o campo "nome", que armazena o nome dos respectivos recursos. Os campos idCOISA e idADM são chaves primárias das suas respectivas tabelas.

RECURSO			
idRECURSO	idCOISA	nome	idADM
01	01	GPS	000
02	01	Acelerômetro	001
03	02	Tag1	000

Tabela 5.1. Tabela RECURSO - Banco de Dados ManIoT

Como pode ser observado na Tabela 5.2, além da identificação do recurso (chave estrangeira), o protótipo armazena os valores dos dados capturados pelos dispositivos

[5]DBDesigner - https://dbdesigner.net/

e o momento da coleta - *time* (Data e Hora). Com esse formato qualquer tipo de dado (texto, inteiro ou real) pode ser armazenado no banco. Nos exemplos apresentados pelas tabelas temos as informações sobre os recursos GPS, Acelerômetro e Etiqueta e os valores coletados pelos recursos (01 e 02) com data e hora de coleta.

DADOS		
idRECURSO	valor	time
01	"12.002343, 23.456837"	01/05/2015 22:45:32
02	"8882321123233"	10/03/2015 02:45:32

Tabela 5.2. Tabela DADOS - Banco de Dados ManIoT

5.5 Cenários Implementados

Para realizar os testes com os dispositivos descritos na Seção 5.3, e integrar suas funcionalidades, foram projetados três cenários, chamados **Tecnologia Assistiva**, **Iluminação Inteligente** e **Automação de Tarefas**. A Figura 5.8 apresenta os três cenários.

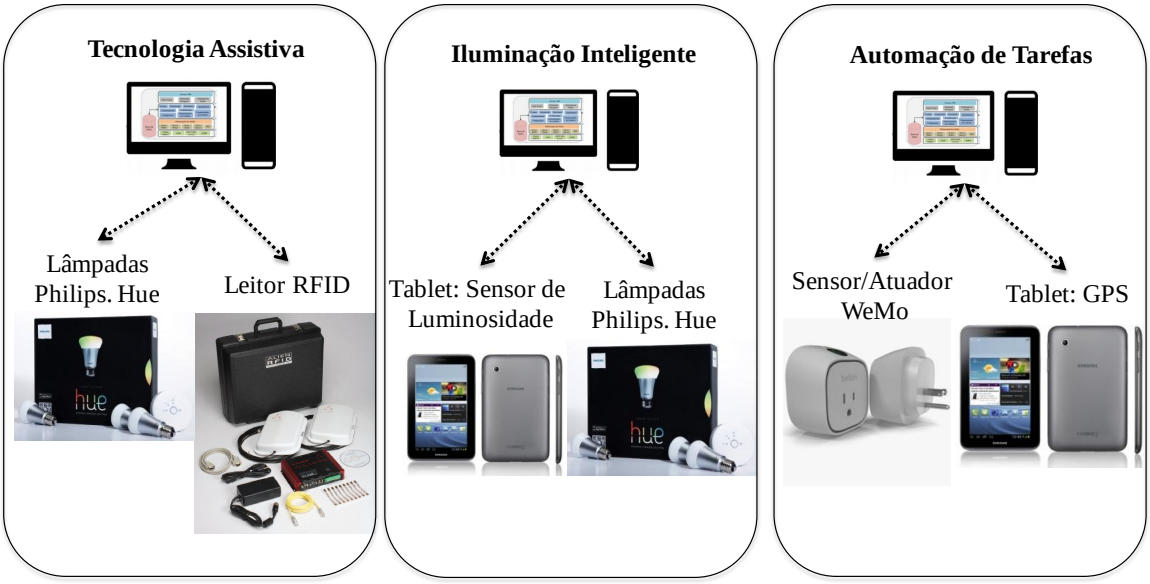

Figura 5.8. Cenários Implementados no Protótipo ManIoT

Cenário 1: Tecnologia Assistiva

A Internet das Coisas pode ajudar as pessoas com dificuldades de locomoção, audição, visão entre outras. O cenário **Tecnologia Assistiva** provê serviços para pessoas com deficiência auditiva e idosos com dificuldade de locomoção. Neste cenário, os usuários podem se beneficiar com os seguintes serviços: Serviço 1 - Alerta sobre presença de pessoas tocando a campainha de uma casa. Este serviço liga uma lâmpada para indicar a presença de uma pessoa. Serviço 2 - Monitoramento de pessoas dentro da casa. O monitoramento controla o deslocamento da pessoa dentro da residência que pode ser usado para detectar possíveis urgências, como tempo excessivo dentro do banheiro, e oferecer serviços automáticos, como ligar luzes.

A rotina do Serviço 1 consiste em utilizar o leitor RFID, modelo Alien 9900, para a detecção de presença (de pessoas previamente autorizadas) e as Lâmpadas para emitir o alerta luminoso. O Serviço 2 utiliza o leitor RFID com leituras recorrentes. Ao detectar uma etiqueta o leitor iria monitorar o tempo que essa etiqueta se manteve naquele ambiente (sala, quarto, banheiro) e fornecer os serviços, como ligar ou desligar uma lâmpada.

O Algoritmo 2 representa a sequência simplificada do protótipo para o cenário tecnologia assistiva. O execução inicia com o recebimento dos dados de uma etiqueta, linha $l.1$. Logo em seguida a plataforma armazena esse valor no banco de dados, linha $l.2$. Na linha $l.3$ o algoritmo verifica qual a localização do dispositivo que realizou leitura da etiqueta, sendo um leitor interno à residência, é verificado o quarto, linha $l.4$, em seguida, na linha $l.5$ é enviado o comando para ligar uma luz do quarto. Se a leitura da etiqueta foi realizada na área externa, linha $l.8$, o algoritmo requisita a ligação de uma lâmpada específica, que para um surdo, a luz de certa cor indica uma presença, como uma campainha para pessoas ouvintes, linha $l.9$.

Algorithm 2 Algoritmo do Cenário Tecnologia Assistiva

```
 1: procedure Assistiva(localizacaoTAG)
 2:     GravaBD(localizacaoTAG)
 3:     if localizacaoTAG.Leitor = Interno then
 4:         idQuarto ← BuscarQuarto(localizacaoTAG)
 5:         LigarLampada(idQuarto)
 6:         return 0
 7:     else
 8:         if LocalizacaoTAG.Leitor = Externo then
 9:             EmitirAltertaCampanhia()
10:             return 0
```

Outro serviço que pode ser oferecido em implementações futuras é a abertura de

portas automaticamente através da detecção de presença. Através de uma etiqueta RFID na cadeira de rodas, por exemplo, ao aproximar de uma porta, outro dispositivo (atuador) iria receber o comando e abrir automaticamente essa porta.

Cenário 2: Iluminação Inteligente

A economia de energia elétrica é uma necessidade nas indústrias, repartições públicas e residências. A IoT pode fornecer serviços para ajudar nessa economia. O cenário **Iluminação Inteligente** fornece meios para reduzir os gastos de energia elétrica com uso de lâmpadas inteligentes.

Este cenário tem o objetivo de controlar a iluminação ligando/desligando e definindo a quantidade de *lúmens* de acordo com a presença de pessoas no recinto e da iluminação natural disponível. Assim, o serviço oferecido por este cenário consiste em controlar as lâmpadas (modelo Philips Hue) com os dados coletados pelo sensor de luminosidade (de um dispositivo Android). O sensor de luminosidade, localizado em ponto estratégico do ambiente, indica a quantidade de luz natural. Portanto, o controle das lâmpadas de uma sala pode ser realizado de forma automática pela plataforma ManIoT.

O Algoritmo 3 representa a sequência simplificada do protótipo para o cenário iluminação inteligente. O algoritmo recebe como parâmetros os valores de luminosidade atual e anterior (penúltimo valor) capturados pelo sensor do *Tablet* e a identificação da lâmpada, linha $l.1$. Na linha $l.2$ o algoritmo verifica se houve uma variação maior que 50 *lúmens* na disponibilidade de luz ambiente capturada pelo sensor, e havendo essa variação o algoritmo: grava a luminosidade capturada no banco de dados, linha $l.3$; converte a luminosidade capturada pelo sensor em taxas utilizadas pelas lâmpadas, linha $l.4$; e altera a taxa de emissão de luz da lâmpada passando sua identificação e o valor da nova luminosidade, linha $l.5$. Se houver mudança na taxa de emissão de luz o algoritmo retorna 1, senão retorna 0.

Algorithm 3 Algoritmo do Cenário Iluminação Inteligente

```
1: procedure Iluminacao(luminosidadeAtual, lumininosidadeAnterior, idLampada)
2:     if (ABS(luminosidadeAtual − luminosidadeAnterior) >= 50) then
3:         GravaBD(luminosidadeAtual)
4:         luminosidadeLamp ← ConverteLuminosidade(luminosidadeAtual)
5:         AlteraTaxaEmissao(idLampada, luminosidadeLamp)
6:         return 1
7:     else
8:         return 0
```

Cenário 3: Automação de Tarefas

A IoT pode automatizar tarefas pré-definidas. Neste terceiro cenário, chamado **Automação de Tarefas**, foram projetados serviços que proveem algumas facilidades na realização de tarefas do usuário tendo como base sua localização.

A localização utilizando GPS, implementada no sistema operacional Android, fornece a posição do usuário. Classificamos as localizações pelos seguintes tipos: "Casa", "Trabalho" ou "Outro". Ao detectar a proximidade ou entrada em uma dessas localizações, a plataforma dispara determinada tarefa. O Controlador WeMo Insight Switch pode, por exemplo, ligar a energia para uma máquina de café na residência assim que o usuário deixar seu local de trabalho ou desligar o alarme de segurança quando este chegar na residência.

Utilizamos uma fronteira virtual determinada por um raio. Consideramos como ponto central a casa ou local de trabalho do usuário. Nos testes utilizamos as coordenadas do centro 8.3591729 e 14.1581481, e um raio de 50 metros. Com isso realizamos as ações no momento que o usuário cruza essa fronteira virtual (em ambos os sentidos).

O Algoritmo 4 representa a sequência simplificada do protótipo para o cenário automação de tarefas. O algoritmo recebe como parâmetros a localização capturada pelo dispositivo de GPS do usuário, o ponto de referência (que neste caso é a residência do usuário) e o raio, que determina uma fronteira para alteração (ou não) do sensor de tomada, linha $l.1$. O algoritmo calcula a distância entre a localização e o ponto de referência, linha $l.2$ e utiliza essa distância para verificar se o usuário está ou não dentro do raio determinado, linha $l.4$. Caso esteja dentro do raio, linha $l.5$, o sensor de tomada WeMo é ativado e caso contrário o sensor de tomada é desativado, linha $l.7$.

Algorithm 4 Algoritmo do Cenário Automação de Tarefas.

```
1: procedure Automacao(localizacaoGPS, pontoReferencia, raio)
2:     distancia = CalculaDistancia(localizacaoGPS, pontoReferencia)
3:     GravaBD(localizacaoGPS)
4:     if distancia <= raio then
5:         AlteraControladorWemo(Ligado)
6:     else
7:         AlteraControladorWemo(Desligado)
```

Considerando o fato que o dispositivo WeMo Insight Switch é um controlador de tomada, este cenário fornece possibilidades diversas. A princípio, qualquer dispositivo pode ser ligado ou desligado automaticamente pela plataforma ManIoT.

5.6 Resumo

Neste capítulo descrevemos os modelos utilizados na implementação do protótipo do gerente local ManIoT. Fizemos uma descrição geral, justificando as escolhas, como sistema de gerência de banco de dados. Mostramos parte do modelo de dados que serviu de base para a estrutura do banco de dados e descrevemos os dispositivos, a comunicação e os cenários desenvolvidos para os testes.

As decisões sobre os recursos usados nesse capítulo (ferramentas, protocolos, entre outros) tiveram como embasamento teórico os trabalhos descritos no Capítulo 3. Criamos ainda novas abordagens, como o modelo de comunicação utilizando banco de dados e o projeto e implementação dos três cenários para realização dos testes com dispositivos e ambientes reais. O gerente global ficou definido no escopo de projeto e será implementado em trabalhos futuros.

Tomando como base os cenários descritos na Seção 5.5, no próximo capítulo apresentaremos os resultados obtidos em cada teste e a análise dos mesmos.

Capítulo 6

Avaliação do Protótipo da Plataforma ManIoT

Após a implementação do protótipo realizamos os testes seguindo a descrição dos três cenários definidos na Seção 5.5. As aplicações e cenários implementados visam criar facilidades para a vida do usuário em um ambiente residencial. Assim, além da verificação funcional, a avaliação consistiu em analisar a capacidade do protótipo. Portanto avaliamos o desempenho do protótipo através dos parâmetros, consumo de memória, processamento e banda. A escolha dessas métricas ocorreram devido o objetivo futuro de migrar o protótipo ManIoT para equipamentos domésticos, como TV's e roteadores.

A próxima seção descreve os resultados obtidos no primeiro cenário (Tecnologia Assistiva), seguindo pela descrição do cenário 2 (Iluminação Inteligente) e por fim o cenário 3 (Automação de Tarefas) é descrito na última seção.

6.1 Cenário 1: Tecnologia Assistiva

Neste cenário utilizamos o dispositivo RFID para detectar a presença de pessoas e as lâmpadas inteligentes Philips Hue para notificar essa presença. A detecção de presença deve auxiliar os usuários com determinadas limitações através de três aplicações. A primeira avisa sobre a presença de uma pessoa autorizada (simulando uma campainha para pessoas ouvintes). A segunda aplicação liga ou desligar uma lâmpada automaticamente em ambientes de uma residência. A terceira e última aplicação monitora o tempo que o usuário permanece em cada ambiente da casa através dos dados gravados (local e momento da gravação). No protótipo testamos a primeira aplicação.

Realizamos os testes inserindo e removendo as etiquetas na área de cobertura do leitor e verificando a ativação da lâmpada. Os resultados apresentados nesta seção

descrevem o desempenho dos subsistemas que compõem este cenário.

O protótipo é composto por dois subsistemas ambos representados por processos. O primeiro processo, que chamaremos de P1, é responsável pela comunicação com o leitor RFID e pela gravação dos dados das etiquetas no banco e o segundo, que chamaremos de P2, é responsável por ler esses dados gravados no banco e, de acordo com os dados lidos, controlar as lâmpadas. Os processos P1 e P2 foram executados no *Desktop* e no *Tablet*, respectivamente. Ambos os equipamentos foram descritos na seção Dispositivos Utilizados (Seção 5.3).

O protótipo, processo P1, busca por etiquetas e grava os dados encontrados no banco a cada segundo. E o protótipo, processo P2, realiza as consultas no banco de dados e altera o estado da lâmpada em intervalos fixos de um segundo. Assim o tempo entre a detecção de uma etiqueta até a notificação pode variar no intervalo de 0 a 2 segundos. Esse valor define o tempo de reação deste cenário.

6.1.1 Consumo de Recursos: Memória e CPU

A Figura 6.1 apresenta uma amostra do uso da CPU do processo P1, responsável pela comunicação com o leitor RFID. Podemos observar que o valor máximo de uso ficou em aproximadamente 7%, com média de 3.6%. Vale ressaltar que esse teste foi realizado através do comando top (que exibe dados dos processos em execução nos sistemas Linux).

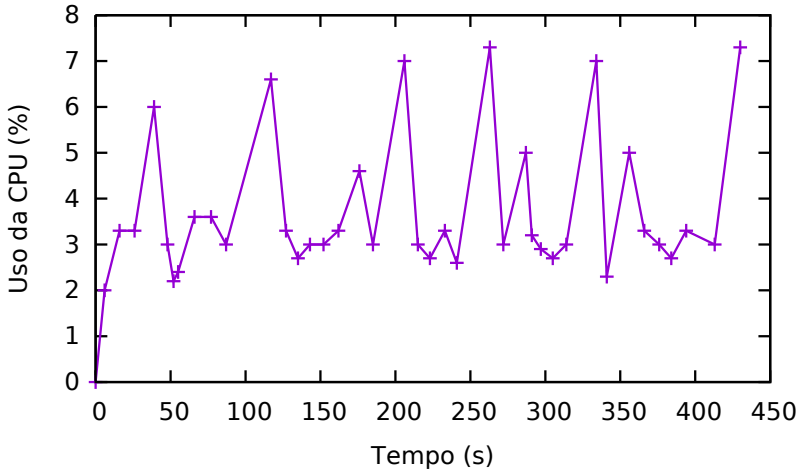

Figura 6.1. Percentual de consumo de CPU - Cenário Tecnologia Assistiva (Processo P1).

A Figura 6.2 apresenta uma amostra do uso da CPU do processo P2, responsável pela busca dos dados das etiquetas no banco e notificação das lâmpadas. Podemos

verificar que o consumo foi próximo do consumo verificado no processo P1, mas com valor de consumo médio menor (igual a 1,8%). Assim podemos concluir que os dois subsistemas não necessitam de alto poder de processamento.

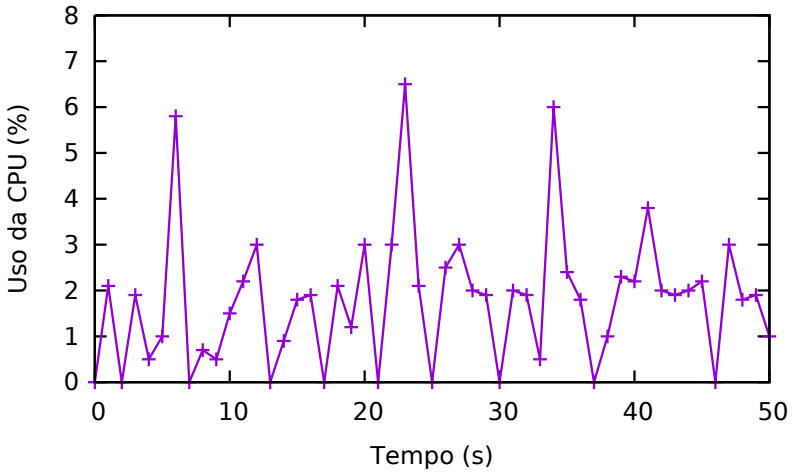

Figura 6.2. Percentual de consumo de CPU - Cenário Tecnologia Assistiva (Processo P2).

Constatamos que o consumo de memória RAM varia entre 3 e 3.5MB, mantendo um valor médio de 3.3MB nos subsistemas implementados. O protótipo opera em um ciclo, realizando buscas por etiquetas, gravando no banco e lendo esses dados do banco para configurar as lâmpadas. Assim, verificamos que o protótipo realiza o gerenciamento utilizando poucos dados em memória. Esse fato justifica o baixo uso desse recurso.

6.1.2 Consumo de Banda

A Figura 6.3 apresenta uma amostra das taxas de troca de dados entre o protótipo (processo P1) da plataforma ManIoT e o leitor RFID a cada segundo. Os valores de *Download* representam as taxas dos dados enviados dos dispositivos para o protótipo e o *Upload* são as taxas dos dados enviados da plataforma ManIoT para os dispositivos. Podemos observar que os dados enviados (*upload*) tiveram variação no intervalo de 0 a 2,08kbps e os dados recebidos (*download*) variaram entre 0 e 4,37kbps, ou seja, o uso de banda de rede neste cenário é mínimo.

Já a Figura 6.4 apresenta uma amostra das taxas de troca de dados entre o protótipo da plataforma ManIoT e as lâmpadas a cada segundo (processo P2). Os dados recebidos variam entre 5.38 e 5.52kbps, enquanto os dados enviados variam entre 3.88 e 3.87kbps. Podemos verificar que praticamente não houve grandes alterações no

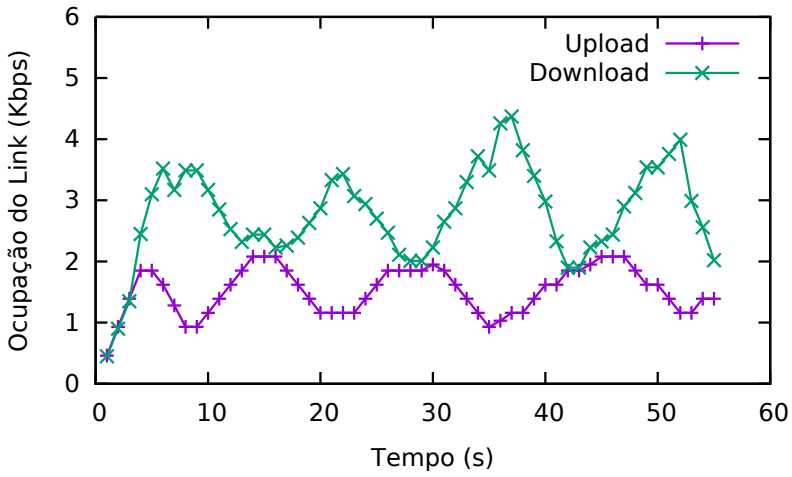

Figura 6.3. Troca de dados entre ManIoT e os dispositivos - Cenário Tecnologia Assistiva (Processo P1).

volume dos dados trocados. Observamos que o pequeno aumento nos dados recebidos (*download*) ocorre quando o protótipo faz leituras no banco de dados e, como resposta, traz informações também da etiqueta lida.

Neste cenário não é possível determinar o momento da entrada de uma nova etiqueta na área monitorada e isso requer uma comunicação permanente entre o protótipo da plataforma e dispositivos. Vale ressaltar ainda que mesmo mantendo uma comunicação constante, o volume de dados trocados é baixo e não compromete outros serviços da rede.

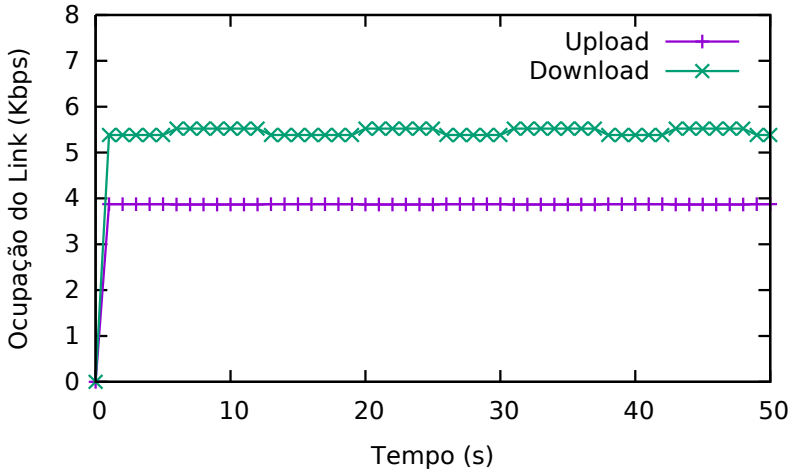

Figura 6.4. Troca de dados entre ManIoT e os dispositivos - Cenário Tecnologia Assistiva (Processo P2).

6.2 Cenário 2: Iluminação Inteligente

Neste cenário, a iluminação de um ambiente é ajustada de acordo com a presença de pessoas e com a existência de luz natural. As lâmpadas são ligadas somente quando há uma pessoa no ambiente. Já a intensidade da luz é inversamente proporcional à quantidade de luz natural. Empregamos as lâmpadas Philips Hue, bem como um *tablet*. O *tablet* fornece, via GPS, a localização da pessoa, e o seu sensor de luminosidade indica a quantidade de luz natural no ambiente.

Medimos o tempo de reação entre a detecção da variação de luminosidade e a atuação da plataforma sobre a intensidade de emissão de luz da lâmpada. Como não é possível inserir código nas lâmpadas inteligentes para notificar a modificação da sua luminosidade, este teste foi feito visualmente, onde tampamos o sensor de luminosidade do *tablet*, e verificamos que o ajuste da intensidade da lâmpada ocorreu após menos de um segundo, o que consideramos aceitável para a aplicação. Vale ressaltar que o sensor de luminosidade atualiza seus dados quando ocorre uma diferença de 50 *lúmens* para mais ou para menos, de acordo com o último valor lido.

A Tabela 6.1 apresenta 10 testes onde houve variação no tempo de resposta, mas a média ficou abaixo de 1 segundo e desvio padrão em 0.25. Assim, como a aplicação não exige execução em tempo real, para a mudança na intensidade de emissão de luz esse tempo médio reação igual a 0.79 é satisfatório.

Tabela 6.1. Tempo de Reação - Iluminação Inteligente

Número de Sequência	Tempo de Reação (Segundos)
1	0.69
2	1.23
3	0.77
4	0.85
5	0.66
6	1.25
7	0.98
8	0.49
9	0.80
10	0.71
Média	**0.79**

6.2.1 Consumo de Recursos: Memória e CPU

A Figura 6.5 apresenta uma amostra do uso da CPU no computador rodando o ManIoT. Esse percentual é variável, se mantendo entre 0 e 6%. Alguns picos de processamento

são observados. Eles são causados por solicitações da plataforma para que a lâmpada modifique a sua intensidade.

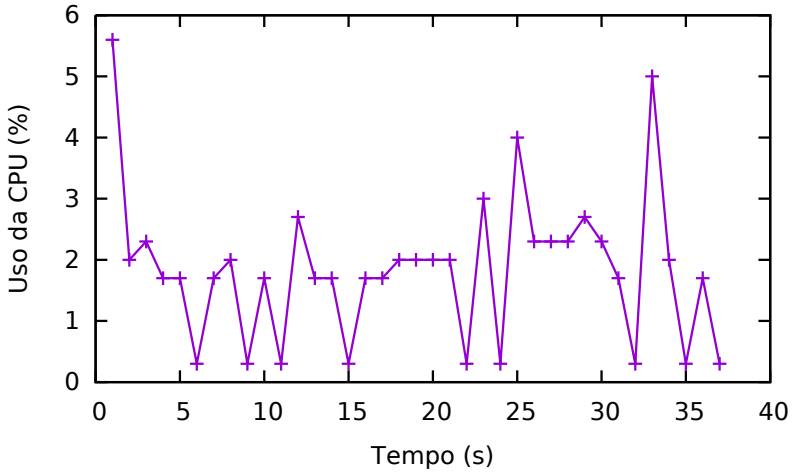

Figura 6.5. Percentual de consumo de CPU - Cenário Iluminação Inteligente.

Já o uso de memória RAM do gerente local variou entre 3 e 7MB, mantendo a média de 3.5MB. Esses valores são justificados devido à pequena quantidade de dados trocados entre o protótipo do gerente local e os dispositivos, reforçando assim o uso mínimo dos recursos de *hardware*.

6.2.2 Consumo de Banda

A Figura 6.6 apresenta uma amostra das taxas de troca de dados entre a plataforma ManIoT e os dispositivos a cada segundo. Os valores de *Download* são os dados enviados dos dispositivos para a plataforma e o *Upload* são dados enviados da plataforma ManIoT para os dispositivos. Os dados recebidos dos sensores variam entre 0 e 50kbps, enquanto os dados enviados variam entre 0 e 32kbps. As trocas de dados são mais intensas quando o gerente local notifica as lâmpadas para uma troca de intensidade luminosa. Mesmo considerando o uso do UPNP, que é extremamente verboso pois emprega XML-RPC, o volume de dados trocado é relativamente baixo. Podemos considerar ainda o fato que as redes Ethernet e WiFi aceitam 100 e 54mbps, respectivamente, o protótipo ManIoT consome aproximadamente 0.05% da banda dessas redes no pior caso.

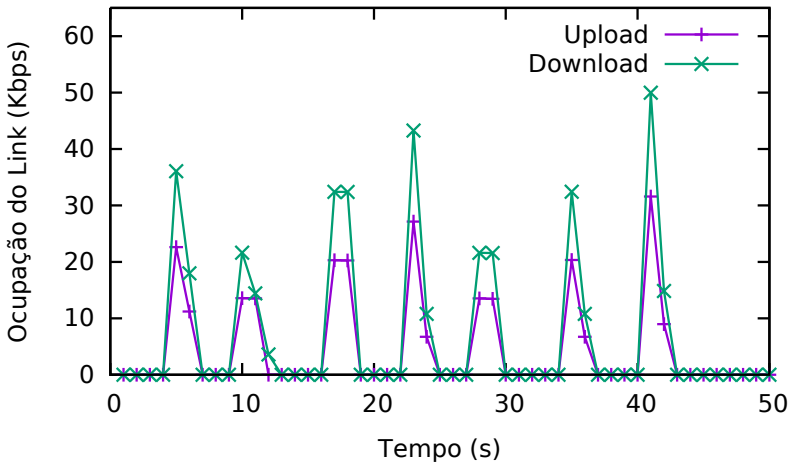

Figura 6.6. Troca de dados entre ManIoT e os dispositivos - Cenário Iluminação Inteligente.

6.3 Cenário 3: Automação de Tarefas

Neste cenário realizamos a ativação automática de dispositivos elétricos de acordo com a localização do usuário. Realizamos os testes utilizando a função GPS do *tablet*. Caso o usuário esteja em uma região geográfica bem definida, ativamos o sensor de tomada WeMo Insight Switch. Quando o usuário sai dessa região, a tomada é desligada. Este cenário pode, por exemplo, ligar uma cafeteira ou alarme de segurança, quando o usuário chega ou sai da sua casa, respectivamente.

No teste realizado, classificamos a localização do usuário em três zonas: "Casa", "Trabalho" ou "Outro". Ao detectar a proximidade ou entrada em uma dessas zonas, a plataforma dispara determinada tarefa. As zonas são definidas como um ponto central (a casa ou local de trabalho do usuário) e um raio. Com isso o protótipo dispara as ações no momento que o usuário cruza essa fronteira virtual. Simulamos a saída ou chegada de uma pessoa na sua residência. No momento de saída do usuário, o sensor de tomada WeMo Insight Switch era ativado (ligando o dispositivo conectado no mesmo), e na chegada, o sensor era desativado. Para tanto, utilizamos um *software* chamado *Fake GPS*[1] para definir a localização do usuário.

Medimos o tempo de reação, que é o tempo decorrido entre a ultrapassagem da fronteira virtual e o acionamento do dispositivo conectado ao sensor de tomada. A medição foi realizada através da diferença entre os tempos dos *logs* de envio e recebimento de uma notificação. A Tabela 6.2 apresenta 8 testes onde houve variação no tempo de resposta com média de 0.158 segundos e desvio padrão de 0.045. Este tempo de reação

[1]Fake GPS, disponível em https://play.google.com

considera o tempo entre o envio da notificação pelo *tablet* até a ativação da tomada WeMo.

Tabela 6.2. Reação da plataforma - Automação de Tarefas

Número de Sequência	Tempo de Reação (Segundos)
1	0.141
2	0.179
3	0.149
4	0.125
5	0.219
6	0.125
7	0.227
8	0.101
Média	**0.158**

Outros resultados obtidos nos testes serão descritos nas próximas subseções.

6.3.1 Consumo de Recursos: Memória e CPU

Já o consumo de memória RAM se manteve entre 3 e 3.75MB, com média de 3.4MB. Portanto, assim como no cenário iluminação inteligente, o protótipo faz uso moderado do recurso memória. Observamos ainda que os valores médios também foram próximos daqueles encontrados no primeiro cenário e na comparação com o cenário Iluminação Inteligente apresentou uso máximo menor. No uso de CPU, constata-se que o protótipo fez uso moderado. A Figura 6.7 apresenta uma amostra de execução onde o valor máximo aproximou-se de 6%, mas a média (considerando somente o efetivo processamento) ficou em 3%. Por fim, vale ressaltar que os valores de uso de CPU também foram próximos aos valores percebidos nos cenários Iluminação Inteligente e Tecnologia Assistiva.

6.3.2 Consumo de Banda

Mensuramos o consumo de banda para transmissão de dados entre o gerente local e os dispositivos. A Figura 6.8 apresenta uma amostra dos testes onde percebemos os valores de 4.5kbps para envio e 1.5 a 2kbps para recebimento de dados. Essa troca de dados ocorre nos momentos em que o *tablet* cruza a fronteira virtual. Neste cenário os valores apresentados foram ainda menores que no cenário anterior, chegando a taxas de 10% dos valores percebidos no cenário iluminação inteligente. Essa diminuição ocorre devido ao fato do protótipo, neste cenário, utilizar uma menor quantidade de eventos.

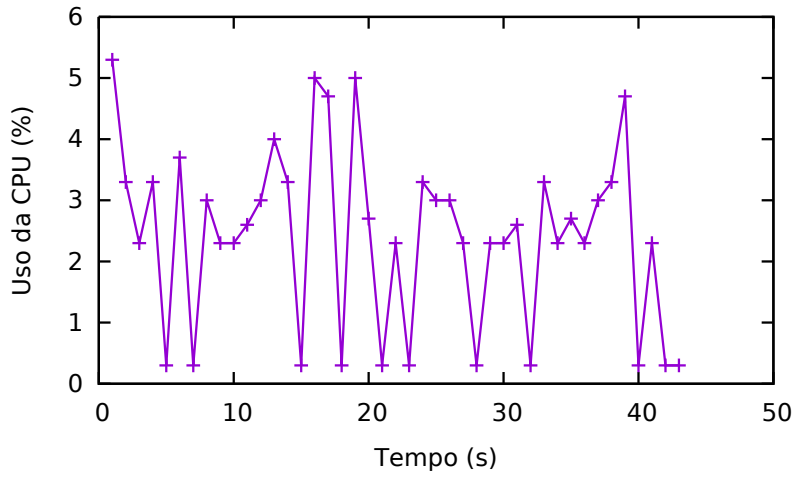

Figura 6.7. Percentual de consumo de CPU - Cenário Automação de Tarefas.

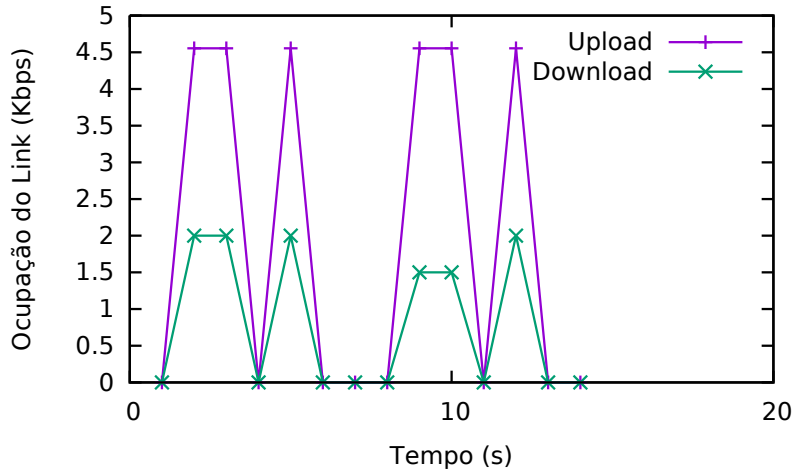

Figura 6.8. Troca de dados entre ManIoT e os dispositivos - Cenário Automação de Tarefas.

6.4 Discussão dos Resultados

Os subsistemas do protótipo ManIoT para o cenário Tecnologia Assistiva apresentaram desempenho satisfatórios. Verificamos que a proposta de notificação de presença através de alerta luminoso é funcional. Verificamos ainda que durante os testes com o cenário, o protótipo não apresentou erros que comprometessem a funcionalidade proposta. Nos resultados verificamos que os subsistemas fazem baixo uso de recursos de rede, CPU e memória. Portanto, o protótipo da plataforma ManIoT para fornecer auxílio a pessoas com dificuldades apresentou resultados satisfatórios e pode ser integrado com outros cenários.

O bom desempenho do protótipo que trata do cenário Iluminação Inteligente foi comprovada com a combinação dos tempos de resposta apresentados e dos recursos computacionais exigidos. Verificamos ainda, através de repetições sucessivas de testes, que os resultados seguiram o mesmo padrão de resposta. Vale ressaltar ainda que durante os testes com o cenário, o protótipo não apresentou erros que comprometessem a funcionalidade proposta pela aplicação. Portanto, assim como o protótipo para o primeiro cenário, este também apresentou resultados satisfatórios e pode ser integrado a outros cenários.

No terceiro e último protótipo, Automação de Tarefas, verificamos que os resultados apresentaram baixo tempo de resposta, e atende as aplicações propostas para esse cenário. Verificamos ainda que o uso de memória RAM ficou dentro do padrão já verificado no cenário 1 e 2. Além da memória RAM constatamos que neste cenário o protótipo fez menor uso de banda e uso aproximado de CPU quando comparamos com o cenário 2. Vale ressaltar também que durante os testes com o cenário, o protótipo não apresentou erros que comprometessem a funcionalidade proposta. Portanto, o protótipo da plataforma ManIoT para automatização de tarefas apresentou resultados satisfatórios e também pode ser integrado a outros cenários.

Então, partindo dos resultados apresentados pelos protótipos nos três cenários, podemos fazer duas constatações gerais: a primeira é que os testes apresentaram o mesmo padrão no consumo de recursos computacionais; e a segunda é que esse padrão de consumo é baixo, permitindo que os protótipos sejam executados em dispositivos residenciais com restrições de recursos (principalmente banda de rede e memória RAM).

6.5 Resumo

Neste capítulo analisamos os resultados obtidos pelos testes do protótipo em três cenários, Tecnologia Assistiva, Iluminação Inteligente e Automação de Tarefas. Verificamos parâmetros de operação, como Consumo de Memória, CPU, Banda de Rede e Tempo de Resposta. Além desses parâmetros analisamos também a confiabilidade, taxa de erros, entre outros.

Os resultados dos testes mostraram que o protótipo da plataforma ManIoT nos três cenários reage com bons tempos de resposta e requer baixo poder computacional. Assim, podemos concluir que o módulo gerente local da plataforma ManIoT pode ser implementado em diversos *hardware*, como roteadores, televisores e até dispositivos móveis. No próximo capítulo concluímos o trabalho e apresentamos as propostas de trabalhos futuros.

Capítulo 7

Conclusões

A Internet das Coisas envolve uma variedade de dispositivos, que possuem a capacidade de interagir uns com os outros e cooperar com seus vizinhos para alcançar objetivos comuns. O grande número de dispositivos decorrentes da IoT naturalmente demanda por soluções de gerenciamento e controle dos diversos serviços, e portanto o provimento de plataformas, que integram esses serviços. Através do gerenciamento podemos usar os recursos oferecidos pelos dispositivos da IoT para fornecer serviços que vão atender diversas áreas, como automação residencial. Entretanto, as plataformas de gerenciamento existentes em IoT atendem parcialmente os requisitos definidos na literatura.

Entre os requisitos não atendidos por uma mesma solução estão interoperabilidade, ciência do contexto, escalabilidade, gerenciamento em diferentes níveis. O requisito interoperabilidade permitiria a integração entre diferentes dispositivos e plataformas disponíveis. O requisito ciência do contexto possibilitaria o uso de informações, como localização e estado dos objetos da rede, para aperfeiçoar ou automatizar as ações e reações dos serviços da plataforma. O requisito escalabilidade proporcionaria a capacidade de expansão e de funcionar corretamente mesmo em situações de uso intenso, e o requisito gerenciamento em diferentes níveis forneceria escopos com variadas possibilidades de tomada de decisão.

Este trabalho apresentou o projeto de uma plataforma para o gerenciamento dos dispositivos em Internet das Coisas, chamada ManIoT, que prevê a escalabilidade e promove a integração de vários dispositivos. ManIoT é genérica e pode ser utilizada em vários cenários, visto que ela emprega uma estrutura em dois níveis de gerenciamento, que permite a inclusão de novos dispositivos e novos cenários gerenciáveis. Além disso, ManIoT faz uso dos dados de contexto e cria possibilidades para expansão dos serviços oferecidos pela rede e a para especificação de novos serviços.

Apresentamos ainda o projeto para gerenciamento de conflitos envolvendo múlti-

plas aplicações e múltiplos usuários. Definimos um algoritmo para tratar os conflitos, especificamos a autenticação e as formas de permissão de execução. E para ilustrar a proposta, criamos modelos que mostram a relação entre usuários, aplicações e recursos.

Um protótipo da plataforma foi implementado, e experimentos foram realizados considerando um ambiente residencial composto por três cenários (Tecnologia Assistiva, Iluminação Inteligente e Automação de Tarefas) e por cinco dispositivos (*tablet*, lâmpadas inteligentes, sensores de luminosidade, sensores/controladores de tomada, leitores RFID).

Os resultados apresentados pelos testes com o protótipo da plataforma nos três cenários mostraram que ManIoT mantém um padrão de consumo de recursos, reage com bons tempos de resposta e requer baixo poder computacional. Assim, podemos concluir que o módulo gerente local da plataforma ManIoT pode ser implementado em diversos *hardware*, como roteadores e até dispositivos móveis.

7.1 Trabalhos Futuros

Como trabalhos futuros e melhorias pretendemos:

- Implementar outras funcionalidades do nível de gerência global. Propomos uma plataforma completa, mas o protótipo prevê apenas o gerente local. Pretendemos implementar o gerente global e a comunicação entre este os gerentes locais, definido na Subseção 4.2.3, e expandir as possibilidades de gerenciamento.

- Propor e implementar outros cenários para plataforma ManIoT. A IoT é abrangente e existem diversos cenários que podem ser criados. Iremos definir novos cenários e através dele habilitar novos serviços.

- Definir novos componentes para a camada de Serviços, Subseção 4.2.1.

- Expandir o uso de dados de contexto para envolver dados remotos e virtuais. Existem outras fontes de dados, como "agenda online" e "previsão do tempo/trânsito", que são úteis para o gerenciamento e pretendemos inserir nas melhorias da plataforma ManIoT.

- Implementar o gerenciamento ManIoT com múltiplos usuários e múltiplas aplicações, modelado na Seção 4.3.

- Integrar cenários distintos. Como foi previsto na plataforma, para um gerenciamento completo devemos utilizar informações de um cenário para habilitar

serviços em outros cenários (integrados). Iremos criar os mecanismos para realizar essa integração.

- Implementar o mecanismo para a execução paralela de diversos cenários em um mesmo gerente local. Como o passo seguinte a integração iremos habilitar o gerente local e global para coordenar os serviços de cada cenário e fornecer um protótipo que ofereça a funcionalidade de gerenciar múltiplos cenários.

Referências Bibliográficas

Atzori, L.; Lera, A. & Morabito, G. (2010). The Internet of things: A survey. *Computer Network*, pp. 2787--2805.

Bassi, A.; Bauer, M.; Fiedler, M.; Kramp, T.; Kranenburg, R.; Lange, S. & Meissner, S. (2013). *Enabling things to talk: designing IoT solutions with the IoT architectural reference model*. Springer.

Bin, S.; Guiqing, Z.; Shaolin, W. & Dong, W. (2011). The development of management system for building equipment Internet of things. Em *3rd International Conference on Communication Software and Networks (ICCSN), 2011*, pp. 423–427.

Carriots, C. S. (2015). Carriots: Carrying the internet of things. `https://www.carriots.com/`. Acessado em: 20-10-2015.

Cavalcante, E.; Alves, M. P.; Batista, T.; Delicato, F. C. & Pires, P. F. (2015). An analysis of reference architectures for the internet of things. Em *Proceedings of the 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures*, pp. 13--16. ACM.

Comer, D. E. (2009). *Computer Network and Internets*. Pearson Prentice Hall, 5th edição.

Cordero, J.; Yi, J.; Clausen, T. & Baccelli, E. (2013). Enabling multihop communication in spontaneous wireless networks. *ACM SIGCOMM*, 1:413--457.

Delicato, F. C.; Pires, P. F. & Batista, T. (2013). *Middleware solutions for the Internet of Things*. Springer.

Ding, Z.; Yang, Q. & Wu, H. (2011). Massive heterogeneous sensor data management in the Internet of things. Em *IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing (CPSCom)*, pp. 100–108.

Elkhodr, M.; Shahrestani, S. & Cheung, H. (2016). A middleware for the internet of things. *International Journal of Computer Networks and Communications (IJCNC)*.

Fan, T. & Chen, Y. (2010). A scheme of data management in the Internet of things. Em *2nd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pp. 110–114.

Flood, P. & Schukat, M. (2014). Peer to peer authentication for small embedded systems: A zero-knowledge-based approach to security for the internet of things. Em *Digital Technologies (DT), 2014 10th International Conference on*, pp. 68–72.

Gubbi, J.; Buyya, R.; Marusic, S. & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645--1660.

Guiping, D. (2013). Design and implementation on a things management protocol for Internet of things. Em *Proceedings of the 32nd Chinese Control Conference (CCC)*, pp. 7361–7364.

Gusmeroli, S.; Piccione, S. & Rotondi, D. (2013). A capability-based security approach to manage access control in the internet of things. *Mathematical and Computer Modelling*, 58:1189 – 1205.

Hardt, D. (2012). *RFC 6749: The OAuth 2.0 Authorization Framework*. `http://tools.ietf.org/html/rfc6749`. Acesso em: 30 Jun 2015.

IBSG-Cisco (2011). The internet of things. `http://share.cisco.com/internet-of-things.html`. Acesso em: 10 Out 2014.

IFTTT (2016). Connect the apps you love. `https://ifttt.com/`. Acessado em: 03-01-2016.

ITU, I. T. U. (2005). The internet of things. Relatório técnico, ITU, `http://www.itu.int/internetofthings`. Acesso em: 20 Ago 2014.

Jan, M.; Nanda, P.; He, X.; Tan, Z. & Liu, R. P. (2014). A robust authentication scheme for observing resources in the internet of things environment. Em *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*, pp. 205–211.

Janak, J.; Nam, H. & Schulzrinne, H. (2012). On access control in the internet of things. Em *IAB Workshop on Smart Object Security, Paris, France*.

Jara, J. A.; Zamora, M. A. & Skarmeta, A. F. (2012). Knowledge acquisition and management architecture for mobile and personal health environments based on the Internet of things. Em *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1811–1818.

Kurose, J. F. & Ross, K. W. (2010). *Computer Networking: a Top-Down Approach Featuring the Internet*. Pearson, 5th edição.

Lang, J. P. (2014). Linksmart middleware platform portal. `https://linksmart.eu/redmine/`. Acessado em: 10-09-2015.

Liu, J.; Xiao, Y. & Chen, C. (2012). Authentication and access control in the internet of things. Em *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pp. 588–592.

LogMeIn (2015). Xively connected product management. `https://xively.com/`. Acessado em: 21-10-2015.

Loureiro, A. A.; Nogueira, J. M. S.; Ruiz, L. B.; Mini, R. A. d. F.; Nakamura, E. F. & Figueiredo, C. M. S. (2003). Redes de sensores sem fio. Em *Simpósio Brasileiro de Redes de Computadores (SBRC)*, pp. 179--226. sn.

Loureiro, A. A. F.; Oliveira, R. A. R.; Silva, T.; Júnior, W. R. P.; Oliveira, L. d.; Moreira, R.; Siqueira, R.; Rocha, B. & Ruiz, L. (2009). Computação ubíqua ciente de contexto: Desafios e tendências. *27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 99--149.

Marotta, M. A.; Carbone, F. J.; Cardoso de Santanna, J. J. & Rockenbach Tarouco, L. M. (2013). Through the internet of things–a management by delegation smart object aware system (mbdsas). Em *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual*, pp. 732--741. IEEE.

Meng, M.; Wang, P. & Chao-Hsien, C. (2013). Data management for internet of things: Challenges, approaches and opportunities. Em *IEEE International Conference on and IEEE Cyber, Physical and Social Computing (CPSCom)*, pp. 1144–1151.

Miorandi, D.; Sicari, S.; Pellegrini, F. D. & Chlamtac, L. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, pp. 1497–1516.

Mulligan, G. (2007). The 6lowpan architecture. Em *Proceedings of the 4th workshop on Embedded networked sensors*, pp. 78--82. ACM.

Nagy, M.; Katasonov, A.; Szydlowski, M.; Khriyenko, O.; Nikitin, S. & Terziyan, V. (2009). *Challenges of middleware for the internet of things*. INTECH Open Access Publisher.

Ning, H.; Ning, N.; Qu, S.; Zhang, Y. & Yang, H. (2007). Layered structure and management in Internet of things. Em *Future Generation Communication and Networking (FGCN 2007)*, volume 2, pp. 386–389.

Pereira, P.; Eliasson, J. & Delsing, J. (2014). An authentication and access control framework for coap-based internet of things. Em *Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE*, pp. 5293–5299.

Peterson, L. L. & Davie, B. S. (2012). *Computer networks : a systems approach*. The Morgan Kaufmann series in networking.

Pires, P.; Delicato, F.; Batista, T. V.; Avila, T.; Cavalcante, E. & Pitanga, M. (2015). *Capítulo 3: Plataformas para a Internet das Coisas*. SBRC 2015.

Pras, A. & Schoenwaelder, J. (2003). *RFC3444: On the Difference between Information Models and Data Models*. `http://tools.ietf.org/html/rfc3444`. Acesso em: 13 Set 2014.

Qin, W.; Li, Q.; Sun, L.; Zhu, H. & Liu, Y. (2011). Restthing: A restful web service infrastructure for mash-up physical and web resources. Em *Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on*, pp. 197–204.

Razzaque, M. A.; Milojevic-Jevric, M.; Palade, A. & Clarke, S. (2016). Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, 3(1):70–95.

Ribeiro, F. & Metrôlho, J. (2016). An internet of things platform to facilitate the development of context aware applications. overview, challenges and experiments.

Ruiz, L.; Nogueira, J. & Loureiro, A. (2003). Manna: a management architecture for wireless sensor networks. *Communications Magazine, IEEE*, 41(2):116–125.

Sanchez, L.; McCloghrie, K. & Saperia, J. (2001). *RFC3139: Requirements for Configuration Management of IP-based Networks*. `https://tools.ietf.org/html/rfc3139`. Acesso em: 13 Set 2014.

Sehgal, A.; Perelman, V.; Kuryla, S. & Schonwalder, J. (2012). Management of resource constrained devices in the Internet of things. *IEEE Communications Magazine*, 50(12):144–149.

SmartThings (2015). *Developer Documentation: Release 1.0*. `https://media.readthedocs.org/pdf/smartthings/latest/smartthings.pdf`. Acesso em: 15 Dez. 2015.

Stallings, W. (2005). Data and computer communications. *Prentice Hall*.

Stallings, W.; de Figueiredo, C. C. & de Figueiredo, L. C. (2006). *Arquitetura e organização de computadores: projeto para o desempenho*. Prentice-Hall.

Winter, T. (2012). Routing protocol for low-power and lossy networks. Relatório técnico, rfc 6550, 6551, 6552. IETF.

Wu, J. & Periorellis, P. (2005). *Authorization-Authentication Using XACML and SAML*. University of Newcastle upon Tyne, Computing Science.