



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
TECNÓLOGO EM REDES DE COMPUTADORES

ALISSON DE LIMA E SILVA

MONITORAMENTO NÃO INVASIVO DE COLMEIAS ATRAVÉS DA IOT

QUIXADÁ

2017

ALISSON DE LIMA E SILVA

MONITORAMENTO NÃO INVASIVO DE COLMEIAS ATRAVÉS DA IOT

Monografia apresentada ao curso de Redes de Computadores da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Redes de Computadores. Área de concentração: Computação.

Orientador: Prof. Me. Antonio Rafael Braga

QUIXADÁ

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S578m Silva, Alisson de Lima e.

Monitoramento não invasivo de colmeias através da IOT / Alisson de Lima e Silva. – 2017.
47 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,
Curso de Redes de Computadores, Quixadá, 2017.

Orientação: Prof. Me. Antonio Rafael Braga.

1. Apicultura. 2. Abelha africanizada. 3. Internet das coisas. 4. Sensoriamento remoto. 5. Arduíno. I.
Título.

CDD 004.6

ALISSON DE LIMA E SILVA

MONITORAMENTO NÃO INVASIVO DE COLMEIAS ATRAVÉS DA IOT

Monografia apresentada ao curso de Redes de Computadores da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Redes de Computadores. Área de concentração: Computação.

Aprovada em: ____/____/____

BANCA EXAMINADORA

Prof. Me. Antonio Rafael Braga (Orientador)
Universidade Federal do Ceará – UFC

Prof. Dr. Marcio Espíndola Freire Maia
Universidade Federal do Ceará - UFC

Profa. Dra. Atslands Rego da Rocha
Universidade Federal do Ceará - UFC

Aos meus pais, Margarida e Jocivaldo.
À minha namorada Cintya.

AGRADECIMENTOS

À UFC, por me proporcionar apoio financeira crucial para minha estadia e permanência no curso.

Ao professor Rafael Braga, pela excelente orientação, disponibilidade, paciência, por tudo que foi investido e por ter me incentivado do começo ao fim para a realização desse trabalho.

Ao Felipe Anderson, por te me auxiliado na implantação da proposta desse trabalho, pela disponibilidade e paciência.

Ao meu tio Francisco Lima pela ajuda inicial, seu apoio e incentivo.

Aos professores participantes da banca examinadora Atslands Rego da Rocha e Marcio Espíndola Freire Maia pela disponibilidade, pelo tempo, pelas valiosas colaborações e sugestões.

À minha namorada Cintya Maria e sua família por todo apoio e incentivo.

Aos meus amigos da turma de 2013.1 por todos os momentos vividos durante esses anos.

À meus amigos pela paciência em momentos de estresse e pelo apoio em momentos de dificuldades.

À todos os meus tios, tias, primos, primas pelo apoio e incentivo.

À meus pais pela ajuda financeira, pelo apoio, pelo incentivo, por tudo que fizeram por mim em toda minha vida.

“O mundo não é um mar de rosas. É um lugar sujo, um lugar cruel, que não quer saber o quanto você é forte. Vai colocar você de joelhos e você vai ficar de joelhos para sempre se você deixar. Você, eu, ninguém vai bater tão forte quanto a vida, mas não se trata de bater forte, se trata de quanto você aguenta apanhar e seguir em frente, o quanto você é capaz de aguentar e continuar tentando. É assim que se consegue vencer”

(Rocky Balboa)

RESUMO

A apicultura, definida como criação racional de abelhas, é uma atividade de fácil manutenção e de baixo custo inicial em relação a outras atividades agropecuárias. Tornou-se uma importante atividade econômica, movimentando um mercado milionário, responsável pela criação de milhares de empregos. Além disto, as abelhas são importantes agentes do processo de polinização, contribuindo para o mantimento do ecossistema. No entanto, diversos fatores como desmatamento, mudanças climáticas, entre outros, houve uma significativa queda no número de abelhas, comprometendo a polinização e ocasionando uma redução na fabricação nos produtos produzidos pelas abelhas. O monitoramento de colmeias se mostra com uma ótima solução para entender melhor o estado de funcionamento das colmeias e através das informações geradas, definir estratégias e técnicas eficientes que possam prevenir perdas na colônia e maximizar a produtividade das abelhas. Esse trabalho apresenta o Mociot, uma solução que através da utilização de sensores e uma rede sem fio, oferece um sistema de monitoramento em tempo real e de baixa intrusão. O sistema tem como intuito, reduzir a inspeção manual, atividade altamente intrusiva que desestabiliza a homeostase no interior das colmeias causando estresse nas abelhas, consequentemente reduzindo sua produtividade.

Palavras-chave: Apicultura. Abelha africanizada. Internet das coisas. Sensoriamento remoto. Arduino.

ABSTRACT

Beekeeping, defined as the rational creation of bees, is an activity that is easy to maintain and low initial cost in relation to other agricultural activities, has become an important economic activity, moving a millionaire market, responsible for the creation of thousands of jobs. In addition, bees are important agents of the pollination process, contributing to the maintenance of the ecosystem. However due to several factors such as deforestation, climatic changes, among others, there was a significant decrease in the number of bees, compromising the pollination and causing a reduction in the production of the products produced by the bees. The monitoring of beehives is shown with a great solution to better understand the state of operation of the hives and through the information generated, to define strategies and efficient techniques that can prevent losses in the colony and to maximize the productivity of the bees. This work presents Mociot, a solution that through the use of sensors and a wireless network, offers a real-time low intrusion monitoring system to reduce manual inspection, a highly intrusive activity that destabilizes homeostasis inside the Hives causing stress on bees, consequently reducing their productivity.

Keywords: Beekeeping. Africanized bee. Internet of things. Remote sensing. Arduino.

LISTA DE FIGURAS

Figura 1 – Morango não polinizado à esquerda e morango polinizado à direita	11
Figura 2 – Termorregulação para aquecimento e para resfriamento	17
Figura 3 – Arquitetura MOCIoT	23
Figura 4 – Trecho do script que envia dados ao Middleware	24
Figura 5 – Demonstração do padrão <i>Publish/Subscribe</i>	26
Figura 6 – Exemplo de criação de notificação	26
Figura 7 – Telas do aplicativo	28
Figura 8 – Nó sensor	30
Figura 9 – Gateway	31
Figura 10 – Estrutura de um documento Mongo	32
Figura 11 – Nó sensor implantado na colmeia	34
Figura 12 – Gateway implantado no ambiente	35
Figura 13 – Tensão da bateria da colmeia	36
Figura 14 – Tensão da bateria do gateway	36
Figura 15 – Histórico de coletas de temperatura	37
Figura 16 – Histórico de coletas de umidade	37

LISTA DE ABREVIATURAS E SIGLAS

ABEMEL	Associação Brasileira dos Exportadores de Mel
API	<i>Application Programming Interface</i>
BBB	<i>Beaglebone Black</i>
DZ	Departamento de Zootecnia
GCP	<i>Google Cloud Platform</i>
GE	<i>Generic Enabler</i>
I2ND	<i>Interface to Network and Devices</i>
IBGE	Instituto Brasileiro de Geografia e Estatísticas
IoT	<i>Internet of Things</i>
LTE	<i>Long Term Evolution</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
NGSI	<i>Next Generation Service Interface</i>
OMA	<i>Open Mobile Alliance</i>
REST	<i>Representational State Transfer</i>
SDN	<i>Software Defined Network</i>

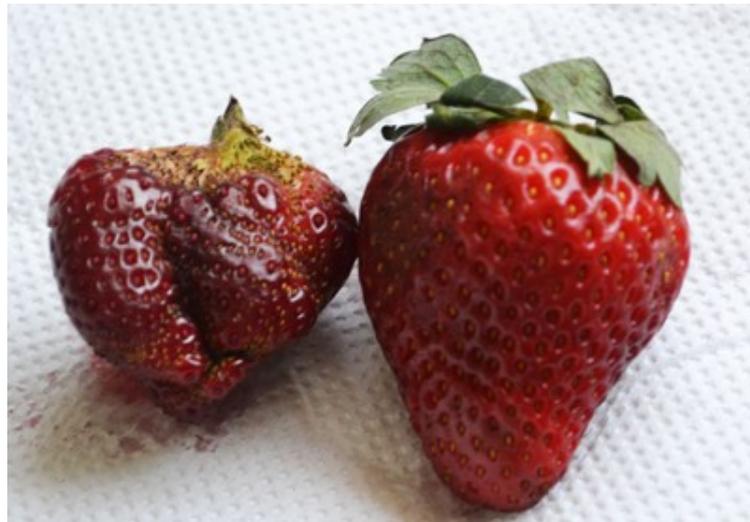
SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos	13
<i>1.1.1</i>	<i>Objetivo Geral</i>	<i>13</i>
<i>1.1.2</i>	<i>Objetivos Específicos</i>	<i>13</i>
2	TRABALHOS RELACIONADOS	14
3	FUNDAMENTAÇÃO TEÓRICA	16
3.1	Comportamento das abelhas	16
<i>3.1.1</i>	<i>Termorregulação</i>	<i>17</i>
<i>3.1.2</i>	<i>Apicultura de precisão e monitoramento de colmeias</i>	<i>18</i>
3.2	Internet das coisas	20
<i>3.2.1</i>	<i>Fiware</i>	<i>21</i>
4	MONITORAMENTO DE COLMEIAS COM IOT (MOCIoT)	23
4.1	Arquitetura do MOCIoT	23
<i>4.1.1</i>	<i>Sensores</i>	<i>24</i>
<i>4.1.2</i>	<i>Gateway</i>	<i>24</i>
<i>4.1.3</i>	<i>FIWARE</i>	<i>25</i>
<i>4.1.4</i>	<i>Webservice</i>	<i>27</i>
<i>4.1.5</i>	<i>Aplicação móvel</i>	<i>27</i>
5	VALIDAÇÃO DO SISTEMA	29
5.1	Planejamento	29
<i>5.1.1</i>	<i>Sensoriamento</i>	<i>29</i>
<i>5.1.2</i>	<i>Ambiente de processamento e armazenamento de dados</i>	<i>31</i>
<i>5.1.3</i>	<i>Política de sensoreamento e economia de energia</i>	<i>33</i>
5.2	Instanciação	34
5.3	Resultados	35
6	CONSIDERAÇÕES FINAIS	38
	REFERÊNCIAS	39
	APÊNDICE A – IMPLEMENTAÇÃO DE REPETIDOR	42
	APÊNDICE B – CRIAÇÃO DE ENTIDADE	44

1 INTRODUÇÃO

A apicultura, por ser uma atividade de fácil manutenção e de baixo custo inicial em relação a outras atividades agropecuárias, tornou-se uma opção de renda para o homem do campo (FREITAS; KHAN; SILVA, 2004). A apicultura também desperta interesse em diversos segmentos da sociedade pela sua capacidade de gerar empregos no campo, possibilidade de obtenção de bons lucros e não menos importante, o fato das abelhas atuarem como polinizadores naturais, contribuindo para o equilíbrio do ecossistema e manutenção da biodiversidade (BROWN; PAXTON, 2009). A Figura 1 apresenta a comparação entre um morango polinizado e não polinizado.

Figura 1 – Morango não polinizado à esquerda e morango polinizado à direita



Fonte: Salles (2014)

No entanto, fatores como destruição do habitat natural, aumento das atividades agrícolas, perda da biodiversidade de flora causadas pelo desmatamento, mudanças climáticas e utilização de pesticidas são os principais causadores da atual e preocupante queda no número de abelhas em todo mundo, conseqüentemente, comprometendo a polinização (POTTS et al., 2010). Fatores estes que colaboram para a redução na produção dos produtos fabricados pelas abelhas como cera, própolis, geleia real, apitoxina (veneno retirado das abelhas fêmeas, utilizado para tratamentos de saúde e estéticos) e principalmente o mel.

Como consequência dos fatores citados acima, muitos criadores estão tendo severas perdas nas suas produções em períodos com temperaturas mais elevadas devido a enxameação

por abandono (ALMEIDA, 2008). A enxameação por abandono ou migração é o processo em que todas as abelhas abandonam a colmeia a procura de um novo ambiente com melhores condições.

De acordo com o IBGE ¹ e a ABEMEL ², o Brasil foi do 4º para 11º lugar no ranking dos maiores exportadores de mel do mundo. Essa queda na produção foi causada por uma longa seca na região nordeste do Brasil, umas das principais produtoras de mel, em 2012. Devido às grandes temperaturas predominantes nessa região durante o ano, é comum acontecer um enorme número de fugas das colônias, causando a perda de muitas colmeias (KRIDI; CARVALHO; GOMES, 2014).

São vários os possíveis fatores que podem estar influenciando no mau funcionamento da colônia e provocando a enxameação, como: doenças, stress, pólen contaminado, temperatura, umidade relativa do ar, seca, chuva, falta de alimento, tamanho populacional, e dentre outros (MURPHY et al., 2015).

Nesse contexto, podemos destacar a importância de monitorar colônias de abelhas. Com o monitoramento é possível gerar valiosas informações sobre o comportamento e saúde da colônia e este conhecimento pode viabilizar estratégias com intuito de melhorar a produção e prevenir uma enxameação por abandono (KRIDI; CARVALHO; GOMES, 2014).

Sem o monitoramento por sensores, para conhecer a situação de uma colmeia era necessário uma verificação local, porém é um método muito invasivo, estressante para a colônia, além de causar a morte de grande quantidade de abelhas. Um monitoramento por sensores demonstra ser uma técnica eficiente por ter características de ser pouco invasivo, menos estressante para as abelhas, resultados rápidos, limpo e barato.

Diante dessa problemática, este trabalho tem como propósito elaborar uma solução capaz de realizar monitoramento de apiários, com foco no processo de termorregulação. O sistema identifica um possível abandono da colônia através de anomalias na temperatura, umidade relativa do ar dentro da colmeia e sons criados pela colônia.

A solução proposta neste trabalho é constituída por uma plataforma de internet das coisas (Internet of Things - IoT), uma rede de sensores e uma aplicação móvel. A plataforma de IoT utilizada será o Fiware, cuja função é o recebimento e armazenamento de dados coletados. A arquitetura da rede de sensores é formada por nós sensores e sorvedouro, os nós sensores localizados no apiário, responsáveis pela coleta de dados, o nó sorvedouro incumbido de receber

¹ Instituto Brasileiro de Geografia e Estatística *website*: <<http://www.ibge.gov.br>>.

² Associação Brasileira dos Exportadores de Mel *website*: <<http://www.abemel.com.br>>.

os dados coletados, processar e fazer o envio para a plataforma de IoT. A aplicação móvel é encarregada de apresentar os dados coletados de forma clara e simples para o usuário final.

O trabalho está organizado da seguinte maneira. O capítulo 2 descreve os trabalhos relacionados que serviram de base, nela são demonstrados as semelhanças e diferenças entre eles. No capítulo 3, são fundamentados os conceitos necessários para compreensão deste trabalho. No capítulo 4 é apresentado nossa solução para Monitoramento de Colmeias através da IoT, o MOCIoT. No capítulo 5 são detalhados todos os experimentos realizados e são analisados os resultados obtidos nos experimentos. Por fim, no capítulo 6 são apresentadas as conclusões obtidas.

1.1 Objetivos

A seguir, apresentamos os objetivos do nosso trabalho.

1.1.1 Objetivo Geral

Utilizar conceitos de Internet das coisas, Sensoriamento e Apicultura de precisão para criar um sistema de monitoramento em tempo real de colmeias minimamente intrusivo e de baixo custo.

1.1.2 Objetivos Específicos

1. Implementação de ambiente para gerenciamento de contexto, armazenamento e consulta de dados;
2. Criação de protótipo de sistema de sensoriamento;
3. Implantação do protótipo, coleta de dados e validação do sistema.

2 TRABALHOS RELACIONADOS

O sensoriamento IoT de colmeias e apiários é um assunto frequentemente abordado em trabalhos que visam o entendimento e descrição de padrões comportamentais em situações normais e de estresse das colônias. Nesse sentido, nessa seção, apresentaremos alguns trabalhos que utilizam de sensoriamento para estudar o comportamento das abelhas e um trabalho que utiliza sensoriamento para auxiliar na tomada de decisão sobre o objeto estudado.

Lewis (2014) apresentou uma solução que através de uma balança digital ligada por uma porta serial a um computador de baixo custo que lia o peso da colmeia e enviava as informações por e-mail para o apicultor. No nosso trabalho as leituras são feitas de forma periódica com intervalos de 60 segundos entre cada leitura e podem ser consultados através de uma aplicação para *smartphones*, enquanto o trabalho do Lewis (2014), os valores são solicitados pelo apicultor ao sistema e o mesmo retornava um e-mail contendo as informações sobre o peso da colmeia. Outra vantagem do nosso trabalho é sua capacidade de um armazenamento histórico dos valores coletados e isso é bastante útil por favorecer uma melhor tomada de decisão pelo apicultor em relação a colmeia. Além disso, nosso trabalho emprega um número maior de métricas monitoradas e isso é um grande diferencial pois pode prevenir diversos problemas com a colmeia e oferece ao apicultor um maior conhecimento sobre as possíveis causas de problemas com sua criação.

Kandepi (2015) tinha como propósito em seu trabalho construir um hardware que monitorasse a temperatura e o som de uma colmeia, as informações coletadas eram enviadas por uma rede de sensores sem fio a um display LCD onde eram exibidas em tempo real. Tanto nosso trabalho, quanto este trabalho relacionado empregam o uso de sensores para realizar coletas de temperatura e som de uma colmeia e utilizam de uma rede sem fio para o envio das informações. Em relação as diferenças entre este trabalho e o trabalho Kandepi (2015), destacamos a forma de exibição dos dados coletados, nosso sistema dispõe de uma aplicação móvel que apresenta em tempo real os valores coletados pelos sensores e também se difere pela sua capacidade de armazenar os dados coletados de forma histórica, no trabalho de Kandepi (2015) não foi idealizado outra forma de apresentar o dados coletados, assim como também não oferecia um armazenamento para esses dados. O trabalho de Kandepi (2015) utilizava-se de energia renovável, através do uso de placas solares, para alimentação dos sensores, enquanto este trabalho não conseguiu implementar ainda uma solução que oferecesse energia renovável para os sensores.

Murphy et al. (2015) apresentou um sistema alimentado por energia solar capaz de manter o sistema funcionando por vários dias sem sol, capturando 5 amostras de imagens (infravermelho e térmicas) por dia e armazenando em um cartão de memória. O sistema também capturava ruídos e movimentações bruscas na colmeia através de um sensor de ruído e um acelerômetro. Caso o sistema detectasse essas movimentações bruscas ou ruídos anormais que atingissem um valor predeterminado, seria gerado um alerta e enviado através da tecnologia GSM/GPRS por meio de um SMS. O diferencial do trabalho do Murphy et al. (2015) em relação ao nosso é sua capacidade de monitoramento através de imagens, além de sua capacidade de auto alimentação por energia solar. Porém nosso trabalho apresenta o monitoramento de um número maior de métricas e o desenvolvimento de uma aplicação móvel que permite a leitura dos valores em tempo real dos valores coletados pelos sensores.

Abreu e Perez (2017) propuseram um sistema capaz de monitorar diversas variáveis importantes para o funcionamento de ambientes protegidos e automatizados para cultivo, ou como conhecidos, abrigos de cultivo. O autor cita que mesmo esses ambientes sendo automatizados, o produtor tem que manter constante atenção e alerta ao ambiente pois o mesmo pode apresentar anomalias em seu sistema ou fatores externos como falta de energia. O objetivo principal do trabalho é, através de sensores, monitorar temperatura, umidade, luminosidade e tensão elétrica dos abrigos de cultivo, realizar processamento das informações coletadas e acionamento de equipamentos responsáveis pela alteração do ambiente caso necessário, além de definir um tempo entre cada leitura dos sensores que possa reduzir o consumo energético mas que não comprometa a operação de automação do abrigo. Apesar de serem trabalhos voltados para áreas de estudo diferentes, esse trabalho se assemelha com o nosso pelo uso de sensores e uma rede de sensores sem fio para propiciar o acompanhamento de algum objeto de estudo, além de oferecer uma aplicação móvel que permita um acompanhamento em tempo real das informações coletadas.

A tabela 1 apresenta um comparativo dos trabalhos citados acima.

Tabela 1 – Comparativo entre trabalhos

Trabalho	Grandezas monitoradas	Persistência dos dados	Apresentação dos dados
Lewis (2014)	Peso	Não	E-mail
Kandepi (2015)	Temperatura, som	Não	Display LCD
Murphy et al. (2015)	Imagem, som, movimentações bruscas	Não	Alerta SMS
Abreu e Perez (2017)	Temperatura, umidade, luminosidade, tensão elétrica	Sim	Aplicativo móvel

3 FUNDAMENTAÇÃO TEÓRICA

Nessa seção será apresentada uma visão geral sobre os conceitos abordados nesse trabalho. Na seção 3.1 é apresentado conceitos sobre o o comportamento das abelhas, desde seu comportamento, até informações sobre as grandezas estudadas e sua importância em uma colônia de abelhas . Na seção 3.2 é apresentada conceitos sobre a tecnologia e as ferramentas que auxiliarão na criação do ambiente e funcionalidades propostas pelo trabalho.

3.1 Comportamento das abelhas

A classe animal de insetos é a mais numerosa e mais difundida. Os insetos podem ser encontrados em quase todos os lugares do planeta. Constituindo um grupo de animais de grande importância ecológica como agentes de polinização, vetores de doenças, pragas na agricultura, parasitismo, produção de alimentos, entre outros.

O grupo de insetos que chamam mais atenção para estudos são os insetos sociais, devido principalmente ao seu papel ecológico e sua capacidade organização. No grupo dos insetos sociais se destacam as abelhas por possuir umas das mais complexas organizações sociais e tendo a espécie *Apis mellifera* como a mais estudada (KRONENBERG; HELLER, 1982). Esta espécie é de grande importância para a sociedade pela produção de mel e pela manutenção da vida vegetal feita por meio da atividade de polinização, contribuindo em um aumento notável da produtividade agrícola.

A espécie objeto de estudo deste trabalho é a *Apis mellifera*. Essa espécie é pertencente ao reino Animal, filo Arthropoda, classe Insecta, ordem Hymenoptera, subordem Apocrita e superfamília Apoidea. São insetos sociais e apresentam 3 castas de indivíduos: rainha, operarias e zangões. Todas essas castas passam pelas fases de ovo, larva, pupa até atingir a fase adulta e possuem tarefas definidas na colônia (RAMOS; CARVALHO, 2007).

As abelhas são famosas por manter cooperativamente a homeostase dentro da colmeia, regulando propriedades biofísicas tais como: temperatura, umidade e gases respiratórios. Ao fazerem isso, elas criam um ambiente adequado, regulando condições adversas (WILLIAMS et al., 2013). A atividade de controle de temperatura no ninho é conhecido como termorregulação, processo melhor explicado na seção 3.1.1.

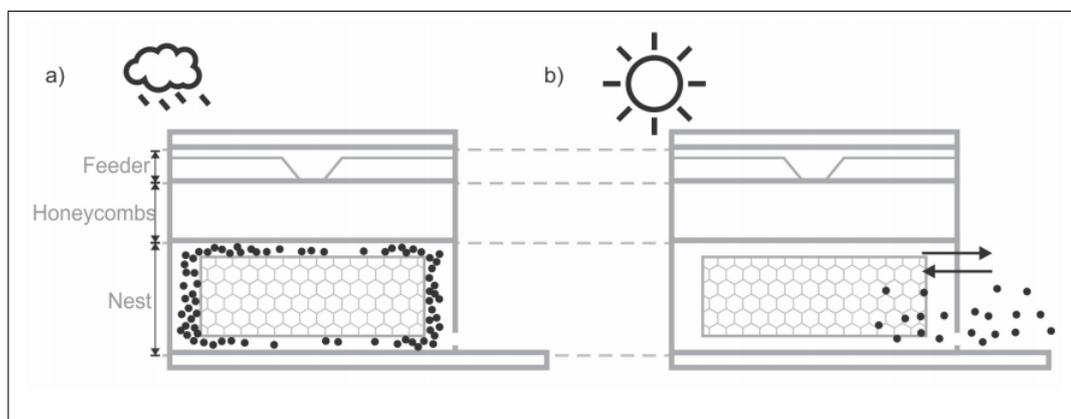
3.1.1 Termorregulação

Insetos são animais pecilotérmicos ou "animais de sangue frio", pois sua temperatura corpórea varia de acordo com a temperatura do meio em que vivem (RODRIGUES, 2004). As abelhas tem seu metabolismo e atividade influenciados pela temperatura corpórea, que por sua vez, está quase inteiramente na dependência da temperatura do ambiente. Temperaturas baixas normalmente dificulta a atividade, enquanto altas temperaturas estimulam o animal (ALMEIDA, 2008).

No caso dos insetos sociais, controlar variáveis de temperatura é fundamental para garantir uma gestação apropriada das crias e sobrevivência da colonia em diferentes temperaturas, como no inverno, por exemplo (LOLI, 2008). Mardan e Kevan (2002) afirmam que longas exposições a temperaturas elevadas ou muito baixas podem levar a mortes ou má formações das crias, assim como o colapso fisiológico de operarias ou de forrageiras em voo. ALMEIDA (2008) revelou em seus experimentos que atingindo uma temperatura interna do ninho em 41 °C, as abelhas abandonam a colmeia, porém as colonias só abandonaram o ninho após uma longa exposição à essas altas temperaturas. Isso só reforça que a termorregulação é umas das atividades mais importantes para sobrevivência e bom funcionamento da colmeia e o seu mau funcionamento pode gerar complicações a colonia.

As abelhas buscam controlar rigorosamente o microclima da sua colmeia, principalmente no ninho, mantendo a temperatura dentro de uma faixa térmica entre 33 °C e 36 °C (ALMEIDA, 2008). A Figura 2 apresenta algumas movimentações que as abelhas realizam para manter o controle da temperatura do ninho.

Figura 2 – Termorregulação para aquecimento e para resfriamento



Fonte – (KRIDI; CARVALHO; GOMES, 2014)

Abelhas do gênero *Apis* controlam a temperatura do ninho por meio de práticas como conservar calor através da concentração de indivíduos, produzir calor através de tremores musculares, fazer o controle da temperatura por ventilação e entre outros. Essas características que permitiram sua migração para regiões cuja temperatura varia regularmente ao longo do ano (CARVALHO, 2009). A Figura 2a mostra a termorregulação interna para aquecer a colmeia. As abelhas formam um aglomerado em torno do ninho e fazem contrações musculares para esquentar a colmeia, quanto mais baixo as temperaturas mais apertado é o agrupamento. Na Figura 2b, para promover o resfriamento em condições de altas temperaturas, as abelhas utilizam-se da ventilação promovida pelo batimento das asas, retirando o ar quente de dentro da colmeia e ajudando na circulação da corrente de ar (DOMINGOS; GONÇALVES, 2014). Outra forma de resfriamento observada no trabalho de ALMEIDA (2008) é a formação de barbas (“*cluster*”) fora do ninho, assim elas não esquentam o ninho com suas atividades metabólicas e permitem uma melhor ventilação dentro do ninho (ALMEIDA, 2008).

Problemas com termorregulação e outros inúmeros fatores de estresse generalizado como por exemplo, altas temperaturas externas, variações de umidade relativa, falta de água e alimentos, doenças e entre outros, podem fazer com que a colônia inicie o processo de enxameação parcial ou total, processo denominado de enxameação por abandono (ALMEIDA, 2008).

Com intuito de supervisionar a colônia e coletar informações sobre esses fatores tão importantes para a saúde da colmeia, existe atualmente vários estudos e soluções que buscam identificar problemas que podem vir a causar a enxameação de uma colônia. Esse assunto será tratado na seção 3.1.2.

3.1.2 Apicultura de precisão e monitoramento de colmeias

A apicultura de precisão é uma estratégia de gerenciamento de apiários baseado no monitoramento de colônias de abelhas individuais para minimizar o consumo de recursos e maximizar a produtividade das abelhas. A apicultura de precisão tem como principal requisito a possibilidade de coletar e analisar dados sobre a colmeia de forma contínua, em tempo real e de forma automatizada (ZACEPINS; STALIDZANS; MEITALOVIS, 2012).

Os sistemas de monitoramento modernos podem fornecer ao apicultor dados em tempo real e informações sobre parâmetros importantes de uma colônia e com base nessas informações o apicultor pode tomar conclusões e em caso de necessidade tomar medidas para

obter uma maximização da produção da colmeia. E as tecnologias de comunicação sem fio e sensores podem ser aplicadas na apicultura para medir parâmetros necessários de uma colônia (MEZQUIDA; MARTÍNEZ, 2009).

Baseado na leitura realizada para este trabalho, os parâmetros mais usados como medidas para monitoramento de colmeias são: temperatura, umidade relativa, peso da colmeia, ruído, luminosidade, dióxido de carbono e imagens. Como exemplo da utilização de algumas das métricas citadas acima, a partir de informações de temperatura combinados com umidade relativa podem ser detectados eventos relacionados a saúde da colmeia, como aumento no consumo de alimentos ou possível abandono da colmeia (ALMEIDA, 2008). O peso da colmeia reflete na saúde e produtividade da colônia (FITZGERALD et al., 2015). Monitoramento de áudio podem ser aplicados como estimativa no comportamento das abelhas e utilizados para identificar padrões de zumbido do enxame que indique um possível abandono em massa da colônia (BENCSEK et al., 2011) ou possíveis acidentes com a colmeia através da detecção de movimentações bruscas (MURPHY et al., 2015).

Já existe no mercado produtos e projetos que oferecem serviços de monitoramento de colmeias, como por exemplo o Arnia¹, o ITAPIC², BeeWise³ e entre outros. Arnia é um sistema de monitoramento remoto que fornece informações vitais da colmeia em tempo real para *Smartphones* através redes sem fio e utilização de sensores para coletar dados de algumas das métricas citadas acima. Oferecendo serviços voltados desde apicultores entusiastas, que praticam a apicultura como um *hobby*, apicultores comerciais e até mesmo cientistas de abelhas. ITAPIC tem como objetivo aplicar princípios agrícolas na apicultura usando recursos industriais existentes e tecnologias de informação e comunicação. O principal objetivo do projeto é identificar diferentes estados de colônia de abelhas e prevenir perdas de colônias. O sistema prevalente pode ajudar a determinar o estado da colônia alertando o apicultor para qualquer anomalia de estados normais (KVIESIS; ZACEPINS, 2015). BeeWise é um sistema que através de sensores de massa, permite que os apicultores obtenham atualizações sobre o peso de suas colmeias por SMS.

Como conclusão, a aplicação de tecnologias oferece novas possibilidades para o desenvolvimento de uma atividade de apicultura eficiente e sustentável. A utilização de técnicas eficientes podem mudar e melhorar a compreensão dos apicultores sobre o comportamento das

¹ Arnia. *website*: <<http://www.arnia.co.uk/>>

² ITAPIC, do inglês *Application of Information Technologies in Precision Apiculture* ou Aplicação de tecnologias da informação na apicultura de precisão. *website*:<<http://www.itapic.eu/>>

³ BeeWise. *website*:<<http://www.beewise.eu/>>

abelhas, maximizando suas produções e melhorando a cooperação entre a colmeia e o apicultor. (ZACEPINS; STALIDZANS; MEITALOV, 2012).

3.2 Internet das coisas

Do inglês *Internet of Things* (IoT) é um paradigma inovador que está rapidamente ganhando terreno no cenário das modernas telecomunicações sem fio. A ideia básica deste conceito é a presença generalizada de nós de uma variedade de coisas ou objetos como tags de identificação por radiofrequência (RFID - *Radio Frequency Identification*), sensores, atuadores, telefones celulares e etc., que através de esquemas de endereçamento únicos, são capazes de interagir uns com os outros e cooperar com seus vizinhos para alcançar objetivos comuns (PAN; PAUL; JAIN, 2011).

Internet das Coisas refere-se à integração de objetos físicos e virtuais em redes conectadas à Internet, permitindo que “coisas” colem, troquem e armazenem informações a respeito do meio que estão inseridas, isso implica em uma enorme quantidade de dados gerados, em que uma vez processados e analisados, geram informações e serviços em escala inimaginável. Apontada como uma revolução tecnológica iminente e com mercado mundial estimado em 1,7 trilhão de dólares em 2020 (SANTUCCI, 2010), a IoT gera impacto em todas as áreas, incluindo eletrônica de consumo, saúde, e de maneira transversal, na forma como a sociedade consome informação (ALMEIDA, 2012). Essas informações podem ser acessadas a qualquer momento através da Internet usando tecnologias de rede como: RFID, Wi-Fi, WAN, *Long Term Evolution* (LTE). No entanto, nesses objetos não estão incluídos apenas os dispositivos eletrônicos ou os produtos com maior desenvolvimento tecnológico, como veículos e equipamentos, mas também se incluem coisas como alimentos, roupas, materiais diversos, matérias primas e outras (RAO et al., 2012). Dessa forma, a IoT compõe-se de dispositivos heterogêneos conectados a rede.

Para simplificar o desenvolvimento do sistema proposto neste trabalho, utilizamos um *middleware* IoT. Para a escolha do nosso *middleware*, de acordo com as leituras realizadas, foram levados em consideração dois requisitos: interoperabilidade (capacidade de tratar uma grande quantidade de dispositivos com diversidade de hardwares, softwares e protocolos de forma transparente) e escalabilidade (capacidade de assimilar um número crescente de dispositivos e requisições e funcionar corretamente mesmo em situações de uso intenso). O *middleware* também deve conseguir tratar uma grande quantidade de dados com segurança e que ofereça uma variedade de ferramentas que possibilitem e facilitem o desenvolvimento de aplicações.

Levando em consideração os requisitos citados acima, para nosso trabalho o middleware escolhido foi o FIWARE, sendo melhor conceituado na seção 3.2.1.

3.2.1 *Fiware*

FIWARE é uma ferramenta para auxiliar a criação de *Smart Metr6poles*. É um projeto p6blico-privado na Uni6o Europeia, com objetivo de criar um ecossistema aberto com base em tecnologias da Internet do futuro para estimular a inova6o envolvendo pequenas e m6dias empresas. Este projeto resultou em uma s6rie de servi6os de modo a fomentar, e suportar, o ecossistema de inova6o. O objetivo é facilitar o custo e efic6cia da cria6o e da entrega de aplica6es e servi6os da Internet do futuro em diversas 6reas, incluindo cidades inteligentes, transportes sustent6veis, log6stica, energia renov6vel e sustentabilidade ambiental (RAMPARANY et al., 2014).

FIWARE *Platform* é uma plataforma de *middleware* que fornece um conjunto de APIs que facilitam o desenvolvimento de aplica6es para cidades inteligentes. A plataforma é de c6digo aberto e com base em ferramentas de programas de *state-of-the-art* e solu6es como OpenStack⁴ e Apache Hadoop⁵. Ela é composta de habilitadores gen6ricos (*Generic Enablers* - GE), tais GEs s6o os elementos principais que constituem o FIWARE e implementam as especifica6es abertas disponibilizadas pelas APIs da plataforma (FERN6NDEZ et al., 2016).

Os GEs s6o agrupados em cap6tulos t6cnicos de acordo com um conjunto de funcionalidades a qual est6o relacionadas, esses cap6tulos s6o *Cloud Hosting*, *Data/Context Management*, *Security*, *Internet of Things (IoT) Services Enablement*, *Applications*, *Services and Data Delivery*, *Interface to Network and Devices (I2ND) Architecture* e *Advanced Web-based User Interface*.

Cloud Hosting é composto por GEs comprometidos com a concep6o de uma moderna infraestrutura de hospedagem em computa6o em nuvem, isto é, uma nuvem FIWARE. O cap6tulo *Data/Context Management* visa proporcionar uma plataforma com alto desempenho na gest6o, processamento e explora6o de informa6es de contexto. O cap6tulo *Internet of Things (IoT) Services Enablement* é respons6vel por possibilitar que coisas (do ingl6s, *Things*) se tornem dispon6veis, pesquis6veis, acess6veis e utiliz6veis como recursos de um contexto para fomento 6 base de intera6o de aplicativos FIWARE com objetos reais. O cap6tulo *Applications, Services*

⁴ Openstack. website: <<https://www.openstack.org/>>

⁵ Apache Hadoop website: <<http://hadoop.apache.org/>>

and Data Delivery busca apoiar a criação de um ecossistema de aplicações, serviços e dados que seja sustentável e que promova a inovação. O capítulo *Security* busca demonstrar que os conceitos de segurança podem ser incorporados à realidade do design da Internet do futuro. O capítulo *Interface to Network and Devices Architecture* tem um foco mais abrangente, sendo dividido em três domínios: Redes definidas por Software (*Software Defined Network - SDN*), dispositivos robóticos e sua integração com outros GEs e um *middleware* de integração avançada voltado para a comunicação entre todos os GEs. O capítulo *Advanced Web-based User Interface* busca oferecer um conjunto abrangente de serviços e aplicações para implementar interfaces de usuários altamente interativas (NAMIOT; SNEPS-SNEPPE, 2014).

Serão utilizados dois GEs neste trabalho, o Cosmos GE e o *Context Broker* GE. O Cosmos, também chamado de *Big Data Analysis*, é um GE que tem como objetivo a implantação de meios que permitem a análise e processamento de dados, ele tem em seu ecossistema ferramentas que auxiliam o desenvolvedor a realizar análise de *Big Data*. Utilizamos também o Cygnus, ferramenta que faz integração com o *Context Broker* para prover armazenamento de contexto persistentes em base de dados externas (FIWARE-COSMOS, 2017).

O Orion *Context Broker* é a implementação do *Context Broker* GE que utiliza do padrão *Publish/Subscribe* e fornece as interfaces NGSI9 e NGSI10. Essas interfaces foram criadas para gerenciar a interação entre usuários e GEs de forma padronizada. Essas interfaces seguem a especificação NGSI1 (*Next Generation Service Interface*) definida pela OMA (*Open Mobile Alliance*). A interação e comunicação é feita através do protocolo HTTP com seus métodos (GET, POST, DELETE, PUT) por meio de uma API REST. O Context broker GE é baseado no funcionamento do protocolo para IoT MQTT (*Message Queuing Telemetry Transport*) e o seu padrão de comunicação *Publish/Subscribe* (FIWARE-ORION, 2017). O MQTT divide sua arquitetura em *broker*, provedores de contexto e consumidores de contexto. O *broker* funciona como intermediador entre os elementos da rede MQTT permitindo um desacoplamento entre eles e reduzindo a quantidade de tráfego (HUNKELER; TRUONG; STANFORD-CLARK, 2008).

Na seção 4.1.3 são melhor detalhados o funcionamento do padrão *Publish/Subscribe*, o funcionamento do Cygnus e Orion e sua implementação para este trabalho.

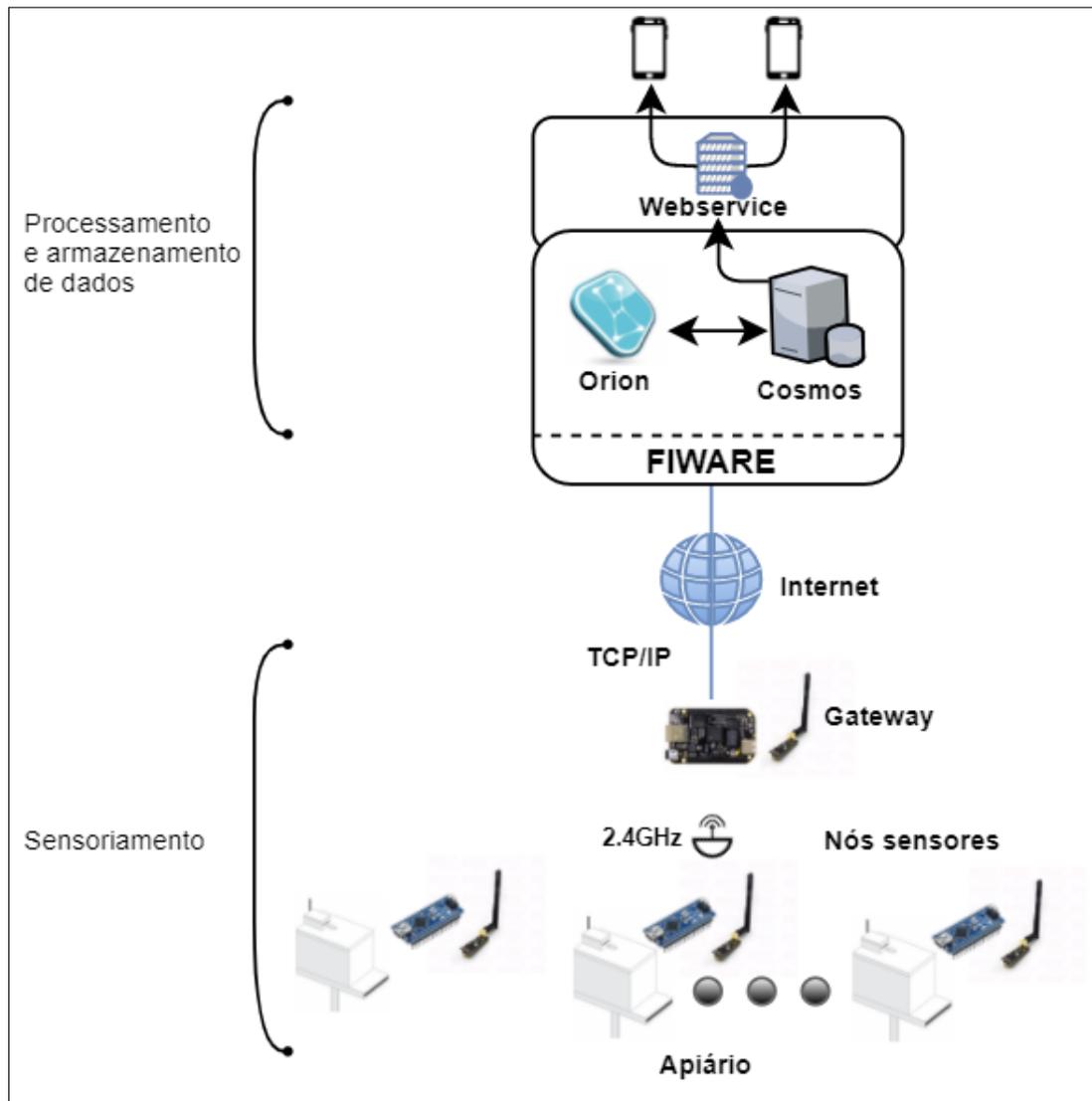
4 MONITORAMENTO DE COLMEIAS COM IOT (MOCIoT)

Esta seção apresenta detalhes e arquitetura do MOCIoT (Monitoramento de Colmeia com IoT), ela foi desenvolvida seguindo a premissa apresentada na introdução deste trabalho.

4.1 Arquitetura do MOCIoT

A arquitetura do MOCIoT, apresentada na Figura 3, é composta por 4 elementos principais: sensores, *gateway*, *middleware*, *webservice* e aplicações móveis. As subseções seguintes apresentam detalhadamente cada elemento da arquitetura.

Figura 3 – Arquitetura MOCIoT



Fonte – Elaborada pelo autor

4.1.1 Sensores

Os sensores são responsáveis pela coleta, processamento dos dados coletados e envio para o nó gateway através de ondas de rádio na frequência 2.4 GHz. Os módulos sensores conectados a placa Arduino coletam as grandezas de temperatura, umidade e ruído, o Arduino processa os dados lidos pelos sensores e utiliza um módulo transmissor à rádio para transmitir os dados coletados para o gateway através da frequência 2.4 GHz. Cada colmeia monitorada contém um nó sensor com seus sensores localizados dentro da colmeia, agrupados entre os quadros centrais do ninho. As leituras são feitas de forma periódica, a cada intervalo de tempo determinado para economia de bateria.

4.1.2 Gateway

A função do Gateway é receber os dados coletados pelos nós sensores e enviar diretamente ao Middleware. A recepção dos dados é feita da seguinte forma, o Arduino recebe os dados transmitidos pelos nós sensores através do módulo à radio, processa essas informações e escreve em uma porta serial no Beaglebone Black(BBB) através da porta USB, ao mesmo tempo, no BBB, um script escrito na linguagem Python, tem como tarefa ler os dados escritos na porta serial, processar e enviar ao Middleware. O envio é feito utilizando a biblioteca Python-requests que envia um JSON através de uma mensagem HTTP à porta 1026. A figura 4 apresenta trechos do script em que as informações são enviadas ao Middleware.

Figura 4 – Trecho do script que envia dados ao Middleware

```
urlOrion = "http://35.184.233.148:1026/v1/updateContext"

#Adiciona os cabecalhos a requisicao
headers = {
    'Content-Type': 'application/json',
    'Fiware-Service': 'tcc',
    'Fiware-ServicePath': '/',
}
#Preparacao do json
data = open('json.json', 'r')
data = data.read()
r = requests.post(urlOrion, headers=headers, data=data)
```

4.1.3 FIWARE

O ambiente de processamento e armazenamento de dados é constituído pelo *middleware* Fiware, que como foi explicado na seção 3.2.1, disponibiliza GEs com determinadas funções na plataforma. No caso da arquitetura proposta, utilizamos os GEs *Orion Context Broker* e o Cosmos GE.

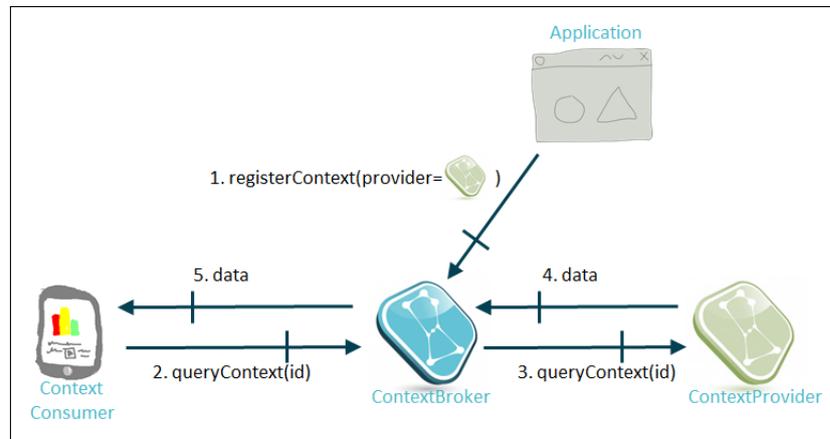
O *Orion Context Broker* é o GE da plataforma FIWARE que gerencia o Data/Context, e consiste em um broker para geração de contexto que utiliza o padrão *Publish/Subscribe* (EUGSTER et al., 2003). Esse broker é responsável por receber e gerenciar os dados de contexto das aplicações que utilizam o FIWARE. No caso deste trabalho os dados de contexto são os dados coletados pelo sensores e as aplicações que utilizam o Fiware são: os próprios sensores, os aplicativos e *Cosmos*.

O padrão *Publish/Subscribe* utilizado pelo Orion Context Broker funciona dividindo clientes entre provedores de contexto e consumidores de contexto. Os provedores de contexto não encaminham diretamente suas mensagens para um consumidor específico. Ao invés disso, a mensagem é encaminhada ao Orion. Da mesma forma, os consumidores expressam interesse em um ou mais contextos fazendo subscrições no Orion e somente recebem mensagens de interesse expressados nessas subscrições (MOLTCHANOV; ROCHA, 2013).

O Orion disponibiliza as interfaces NGSII9 e NGSII10 para que clientes possam fazer operações de *Publish/Subscribe* como registrar produtores de contexto (exemplo, adicionar um sensor de temperatura em uma sala), um provedor fazer atualizações de valores (exemplo, um sensor atualizar a temperatura da sala), uma aplicação fazer uma assinatura para receber notificações após mudanças em um contexto (exemplo, uma aplicação receber uma notificação de mudança de temperatura da sala monitorada por determinado sensor) e realizar operações de consulta (exemplo, consultar a temperatura a cada minuto) (FIWARE-ORION, 2017). A Figura 5 demonstra o funcionamento do padrão *Publish/Subscribe*.

O Cosmos contém um ecossistema de ferramentas voltadas para análise e armazenamento de dados. O Cygnus faz parte do GE Cosmos e foi projetado para prover uma forma de armazenamento persistente de dados de contexto para o *Orion Context Broker* em diferentes bancos de dados externos (MySQL, MongoDB, Postgres e entre outros). Em outras palavras, o Orion Context Broker armazena somente o último valor coletado referente a cada provedor de contexto. Nesse caso, se alguma aplicação desejar acessar dados históricos, é

Figura 5 – Demonstração do padrão *Publish/Subscribe*



Fonte – Fiware-Orion (2017)

necessário a persistência de dados em outra forma de armazenamento (FIWARE-COSMOS, 2017).

Neste trabalho, os provedores de contexto são os sensores localizados nas colmeias, eles terão a tarefa de gerar dados de contexto e enviar ao Orion Context Broker através do gateway, todos já citados acima. (MARTÍNEZ et al., 2016).

O funcionamento do Fiware na arquitetura proposta é da seguinte forma. Como consumidor, o Cygnus precisa fazer uma subscrição para receber os dados coletados dos sensores. Isto é feito adicionando uma notificação no Orion Context Broker. A figura 6 apresenta um exemplo de um JSON enviado ao Orion para a criação de uma notificação.

Figura 6 – Exemplo de criação de notificação

```

{
  "entities": [
    {
      "type": "Colmeia",
      "isPattern": "false",
      "id": "Colmeia1"
    }
  ],
  "attributes": [
    "temperatura",
    "umidade",
    "data",
    "hora",
    "som"
  ],
  "reference": "http://10.128.0.5:5050/notify",
  "duration": "P100Y",
  "notifyConditions": [
    {
      "type": "ONCHANGE",
      "condValues": []
    }
  ],
  "throttling": "PT3S"
}

```

Fonte – Elaborada pelo autor

O JSON tem em seu conteúdo dados relevantes para a criação da notificação. Na chave *entities*, um dos valores é o ID do provedor de contexto, no caso nossa colmeia. Na chave *attributes* é informados quais valores o Cygnus receberá caso esta notificação seja acionada. Na chave *reference* é informada a URL onde o Cygnus estará recebendo as informações oriundas do Orion. A chave *duration* representa a duração da notificação e por fim, a chave *notifyConditions* representa um conjunto de informações, dentre outras o tipo e as condições para acionar esta notificação. Ou seja, o JSON representado nesta figura cria uma notificação que aciona o Cygnus caso o provedor de contexto Colmeia1 envie dados ao Orion Context Broker e qualquer um dos valores monitorados tenham sofrido mudanças.

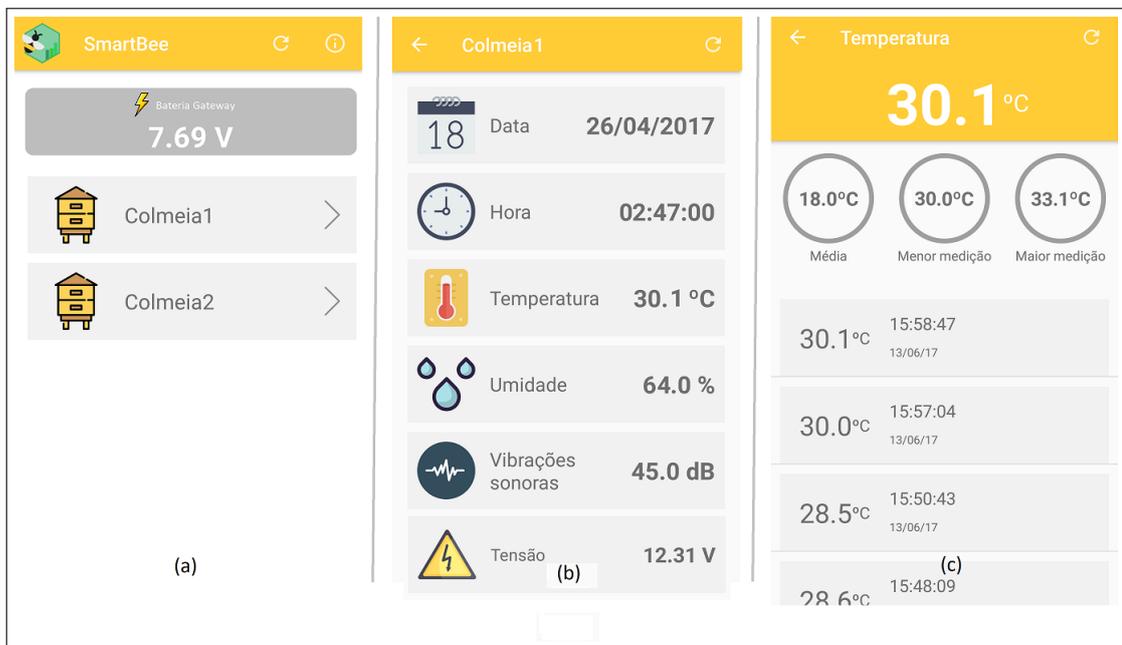
4.1.4 Webservice

O *Webservice* utiliza o banco de dados alimentado pelo Cygnus para realizar consultas SQL e disponibilizar através de uma API REST (*Representational State Transfer*, em português Transferência de Estado Representacional) essas informações para fácil acesso pelas aplicações móveis. A comunicação entre as aplicações *mobile* e o *Webservice* é feita trocando mensagens JSON através de protocolo HTTP.

4.1.5 Aplicação móvel

Para melhor apresentação dos dados coletados, foi desenvolvido também um aplicativo móvel. O aplicativo foi desenvolvido na plataforma Android, com versão mínima SDK 16 (Jelly Bean 4.1). O aplicativo monitora em tempo real os valores coletado pelos sensores. Ele obtém dados em formato JSON através da API Retrofit, fazendo requisições HTTP GET ao *web service*. A figura 7 apresenta as principais telas do aplicativo. Vemos na Figura 7a a listagem de todas as colmeias monitoradas. A figura 7b é apresentada após seleciona uma colmeia na primeira tela, nela são mostrados os valores monitorados (temperatura, umidade e vibrações sonoras), a hora e data em que esses valores foram coletados. Após selecionar uma das três métricas, é chamada a tela mostrada na Figura 7c, nela é apresentada detalhes sobre a métrica selecionada, podemos ver a temperatura atual no topo da tela, logo abaixo o menor valor coletado, a média e o maior valor, logo abaixo é mostrado as ultimas 6 mudanças nos valores coletados, acompanhado da sua data e hora da coleta.

Figura 7 – Telas do aplicativo



Fonte – Elaborada pelo autor

5 VALIDAÇÃO DO SISTEMA

A validação desta proposta foi realizada buscando-se verificar o funcionamento das funcionalidades básicas da aplicação MOCIoT, tais como consumo de energia e obtenção das grandezas monitoradas. Essa seção é dividida em planejamento, experimentos e resultados. A seção de planejamento detalha todo material utilizado e configurações realizadas para criação do sistema. A seção de experimento detalha todos os passos para a implementação do sistema no ambiente de estudo e por fim, na seção de resultados mostramos a análise dos dados coletados pelo sistema.

5.1 Planejamento

As subseções abaixo detalham todo o material e configurações utilizadas para criação do protótipo e implementação da arquitetura proposta na seção 4.1.

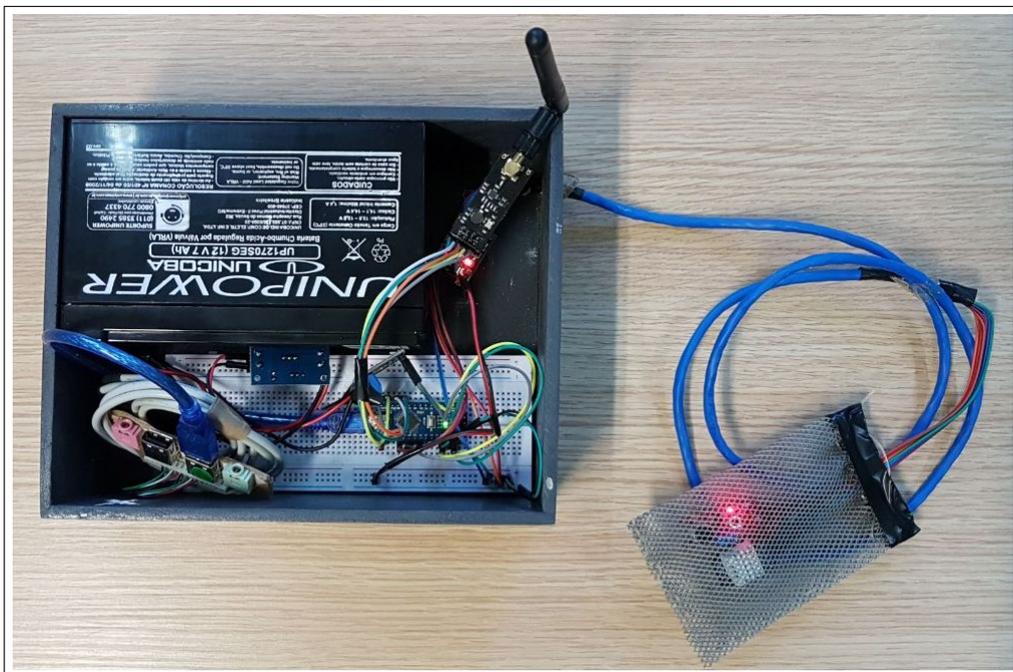
5.1.1 Sensoriamento

O ambiente de sensoriamento proposto na seção 4.1 é dividido entre nós sensores e nó gateway. A criação de nosso protótipo de sensoriamento foram feitas utilizando: duas placas Arduino, um Beaglebone Black, dois módulos de transmissão à radio, um módulo sensor de temperatura e umidade relativa, um módulo detector de ruído, duas baterias e um cabo RJ-45 de aproximadamente 40 metros.

Os módulos para sensoriamento utilizados foram o DHT22 e KY-038. O DHT22 é utilizado para coletar medições de temperatura e umidade, apresentando uma faixa de 0 a 100% com resolução de 0.1% para medições de umidade relativa do ar e uma faixa de -40°C a 80°C com resolução de 0.1°C para medições de temperatura. Sua escolha foi tomada devido capacidade de coletar duas grandezas com um único módulo, seu preço baixo, dimensões reduzidas e apresenta uma medição precisa se comparado a outros módulos estudados para este trabalho, como o DHT11 e o DS18B20. O módulo KY-038 foi o escolhido para realizar medições de ruído, ele funciona variando a tensão de saída em uma porta analógica de acordo com a intensidade sonora captada. O módulo acompanha um regulador de potência e requer uma calibragem para uma conversão fiel para decibéis. Todos os sensores utilizados neste projeto estão respeitando a proposta de serem minimamente invasivos para a colmeia, são de baixo custo e integração simples com a plataforma Arduino.

O nó sensor, mostrado na figura 8, é formado por uma placa Arduino, uma bateria para alimentação, um módulo para transmissão de dados e uma unidade dos módulos DHT22 e KY-038. O Arduino utilizado foi o Nano, que tem como especificações: Microcontrolador Atmel ATmega328, memória Flash com capacidade de 16 Kbytes, velocidade de clock de 16 MHz e dimensões 1,85cm x 4,318 cm. Para alimentação do nó, foi utilizado uma bateria selada com tensão de 12 volts e capacidade de 7 ampere, recarregável e com dimensões de 151 X 65 X 100mm.

Figura 8 – Nó sensor



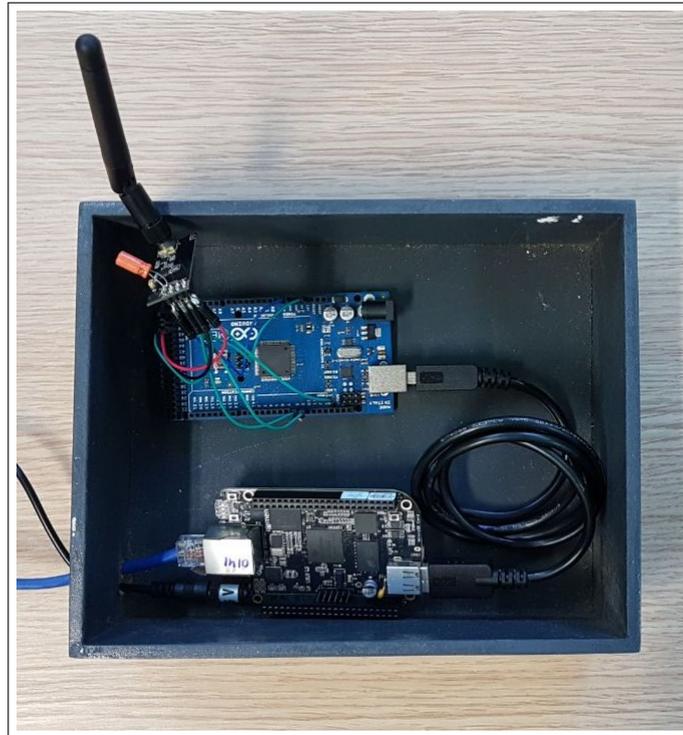
Fonte – Elaborada pelo autor

O módulo de transmissão utilizado nos nós sensores e gateway é o NRF24L01+ versão 5.1 com antena externa. O módulo opera à radio em uma banda de 2.4 GHz, tendo como potência máxima de 20 dBmW. Ele tem como especificações: antena externa de 2dBI, taxas de transmissão de 250Kbps a 2MB, o alcance de transmissão deste módulo varia entre 520 a 1000 metros, dependendo da quantidade de obstáculos e velocidade de transmissão. Suas dimensões são de 41mm x 15mm e comprimento da antena de 107mm.

O gateway, apresentado na figura 9, é formado por Beaglebone Black(BBB), uma placa Arduino e um módulo de transmissão/recepção de dados e uma bateria de mesmo modelo da utilizada no nó sensor. O BBB tem como especificações: Processador Cortex-A8, 1 GHz de clock, 512MB de memória RAM DDR3, armazenamento flash de 4GB, interface Ethernet

(conector RJ45), porta USB 2.0 e sistema operacional Debian 7 instalado. A placa Arduino utilizada foi o Arduino Uno com as seguintes especificações: Microcontrolador ATmega2560, memória flash de 256Kbytes, velocidade de clock de 16 MHz e dimensões de 101.52 x 53.3mm. O módulo de transmissão é o mesmo utilizado no nó sensor.

Figura 9 – Gateway



Fonte – Elaborada pelo autor

5.1.2 Ambiente de processamento e armazenamento de dados

O ambiente de processamento e armazenamento de dados é composto por uma plataforma computação em nuvem, um servidor web e pelos GEs do Fiware *Orion Context Broker* e *Cosmos*.

A solução de computação em nuvem escolhida para este trabalho foi a Google Cloud Platform (referida no restante deste documento como GCP), a decisão de optar por esta plataforma foi por oferecer todos os recursos necessários para o desenvolvimento do trabalho (IP público estático, alta disponibilidade, segurança e entre outros) e oferecer um período de uso gratuito suficiente para a execução do trabalho.

Para a configuração do ambiente foi necessário a utilização de duas máquinas virtuais de nomes *Orion* e *Cygnus*, estas mantidas na GCP. A primeira com as seguintes configurações: 1

CPU, 5.25 Gigabytes de memória RAM, 30 Gigabytes de disco e sistema operacional CentOS 6. A segunda com: 1 CPU, 1.27 Gigabytes de memória RAM, 10 Gigabytes de disco e sistema operacional CentOS 6. Cada máquina virtual é responsável por manter em execução serviços importantes para o funcionamento do sistema.

A máquina virtual Orion é responsável por executar os serviços do *Orion Context Broker*, um banco de dados que armazena os dados do Orion e o servidor Web. O banco de dados executando na instância Orion é o Mongo DB, a utilização deste banco foi devido ele ser um dos requisitos para a instalação do *Orion Context Broker*. Chodorow (2013) define o banco Mongo DB como um tipo relativamente novo de banco de dados que não utiliza conceitos de tabelas, esquemas, SQL (*Structured Query Language* - Linguagem de consulta estruturada) ou linhas. MongoDB é um banco de dados de código aberto que provê alta performance e alta escalabilidade, utilizando conceito de documentos, uma estrutura de dados, semelhante ao JSON (*JavaScript Object Notation*), composta por campos e chaves usados para identificar um registro (MONGODB, 2017). Na Figura 10 é apresentada a representação de um documento em um banco Mongo.

Figura 10 – Estrutura de um documento Mongo



Fonte – MongoDB (2017)

O Mongo DB mostrou ser uma solução eficiente para nosso trabalho por sua escalabilidade e sua característica de não ter tabelas com campos estáticos, isso facilita uma possível ampliação no número de sensores utilizados neste projeto. Como por exemplo, no momento estamos monitorando temperatura, umidade relativa do ar e áudio, mas posteriormente planejamos adicionar mais sensores, e. Em um banco convencional seria necessário fazer alterações na tabela através de SQLs, já no MongoDB é automaticamente adicionado um novo par chave/valor à estrutura do documento.

O (*web service*) também é mantido na máquina virtual Orion, ele foi implementando

na linguagem Java utilizando Jersey. *Jersey* é um *framework* REST, que provê um kit de características e utilidades que simplificam o desenvolvimento de um serviço RESTful. Foi escolhido para o desenvolvimento do *web service* devido a sua simplicidade e por ser uma ferramenta poderosa. Para disponibilizar o *web service* foi utilizado o servidor HTTP Apache Tomcat 7 ¹.

A máquina virtual Cygnus é responsável por manter o serviço do Cygnus e o banco de dados utilizado pelo Cygnus para armazenar o histórico dos dados. O banco de dados configurado foi o MySQL, vale ressaltar que para as configurações do banco de dados, o Cygnus disponibiliza duas opções de armazenamento, a primeira é a opção *row*, funciona criando apenas uma tabela e é criado uma linha para cada nova entrada de contexto no banco, sua vantagem é a criação automática das tabelas, sua desvantagem é a dificuldade em realizar consultas devido todos os dados serem armazenados em uma única tupla no banco, já a segunda opção é a *column*, na qual é utilizado uma tabela para cada nova entidade cadastrada, por exemplo, a cada colmeia adicionada ao sistema, uma nova tabela é adicionada ao banco de dados, sua vantagem é a separação dos valores por coluna, onde cada coluna representa um valor do contexto, a desvantagem é a necessidade de criar manualmente as tabelas.

5.1.3 *Política de sensoriamento e economia de energia*

A política de coleta de dados possui dois objetivos: reduzir a quantidade de informações enviadas por cada nó sensor a fim de diminuir o consumo energético e manter a confiabilidade das medições realizadas.

Para isso, foi definido para este trabalho o intervalo de 60 segundos entre leituras, inicialmente o intervalo seria de 10 segundos, mas por questões de economia de bateria, esse intervalo foi aumentado. Para essa decisão foi levado em consideração um intervalo em que fosse possível realizar uma economia de bateria, mas sem perda na veracidade dos valores coletados.

Para obter mais ganhos na economia de bateria, foi utilizado a biblioteca JeeLib ². Ela é capaz de colocar o Arduino em um estado de sono profundo, desligando todas as funções do Arduino, reduzindo a velocidade do clock e desativando as portas, conseqüentemente diminuindo o seu consumo de energia.

¹ Apache Tomcat *website*: <<https://tomcat.apache.org/>>.

² JeeLib *website*: <<http://jeelabs.org/pub/docs/jeeLib/>>

5.2 Instanciação

Os testes foram executados com uma colmeia da *Apis Mellifera* disponibilizada pelo Departamento de Zootecnia (DZ) da Universidade Federal do Ceará. Os experimentos foram realizados no apiário deste departamento (38°44'33.7"S 38°34'46.5"O) no dia 06/07/2017.

Após os códigos fonte serem finalizado e embarcados nos kits Arduino e realizado últimos baterias de testes de comunicação entre os nós, o nó sensor foi implantado na colmeia. A caixa que armazenava o Arduino e a bateria ficou abaixo da colmeia, apoiada por um suporte e os sensores foram posicionados entre os quadros centrais dentro da colmeia. A imagem 11 mostra o nó sensor implantado na colmeia.

Figura 11 – Nó sensor implantado na colmeia

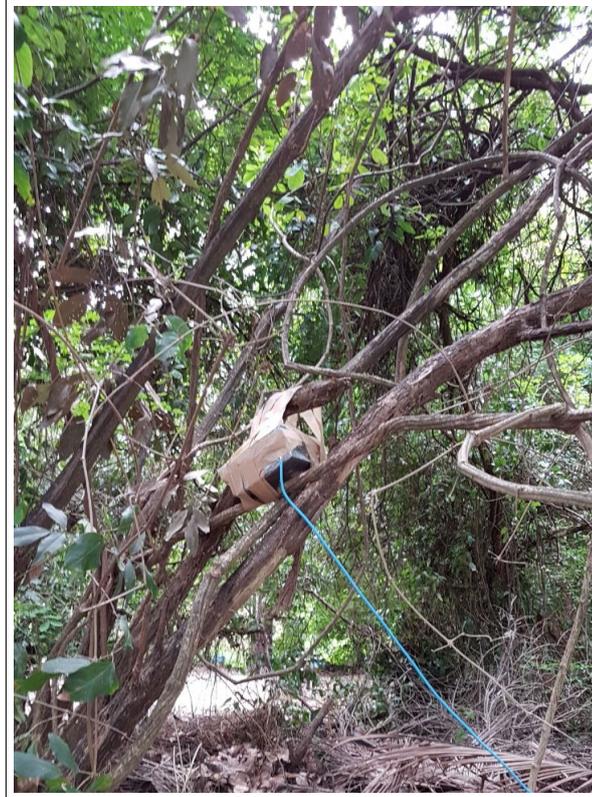


Fonte – Elaborada pelo autor

Após seguidas tentativas de implantação com o uso da comunicação sem fio via sinal de rádio frequência sem obter sucesso na comunicação, tentamos realizar a implantação de um nó que servisse como repetidor e ficasse localizado entre o nó sensor e o gateway, porém também não obtemos sucesso novamente. O repetidor é melhor abortado no A. Para que nós conseguissem trocar informações foi necessário pôr o gateway mais perto do nó sensor, então utilizamos um longo cabo de rede para fornecer conexão com a Internet. A figura 12 apresenta o

gateway implantado.

Figura 12 – Gateway implantado no ambiente



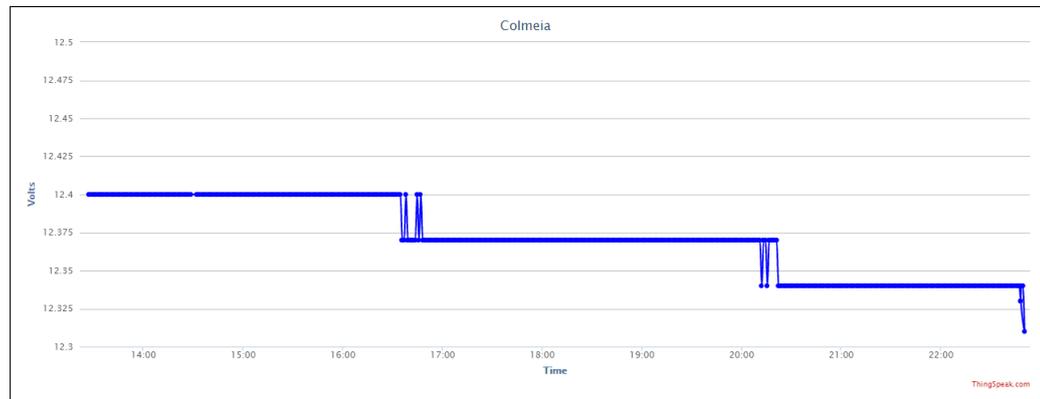
Fonte – Elaborada pelo autor

5.3 Resultados

A aplicação foi iniciada por volta das 13:21 horas e executou até 22:50 horas e foram coletadas 509 amostras. Em relação ao consumo de energia tivemos resultados distintos entre o nós sensor e gateway como é demonstrado nas figuras 13 e 14, respectivamente. Obtivemos sucesso em nosso objetivo de economizar energia no nó sensor, em um experimento que durou mais de 9 horas, a bateria manteve um valor quase constante de 12.34 volts, com pequenas variações, tendo no final do experimento uma queda de apenas 0.03 volts, indo de 12.34 volts iniciais à 12.31 volts ao final. No entanto, não tivemos o mesmo sucesso na autonomia de energia com o gateway, durante o experimento teve uma queda de 4.61 volts, variando sua tensão de 12.3 volts até 7.69 volts. Ao chegar aos 7.69 volts o nó gateway desligou, finalizando nosso experimento.

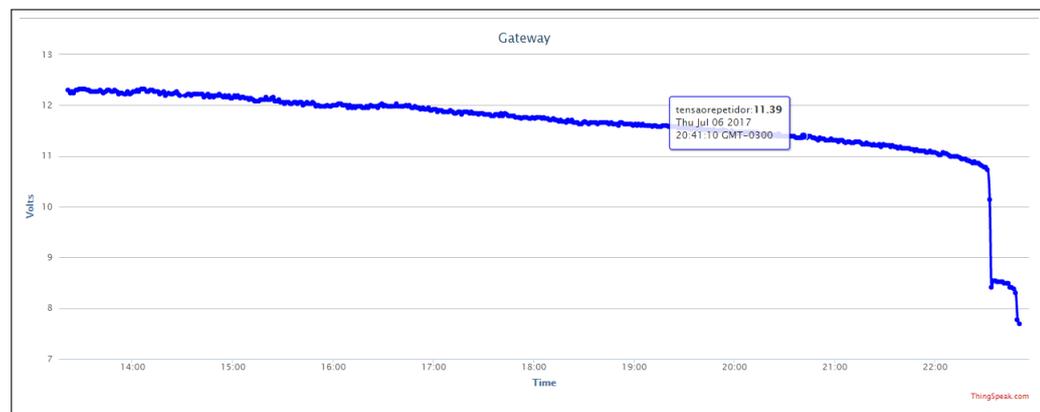
Em relação ao monitoramento das variáveis de temperatura e umidade, foram

Figura 13 – Tensão da bateria da colmeia



Fonte – Elaborada pelo autor

Figura 14 – Tensão da bateria do gateway



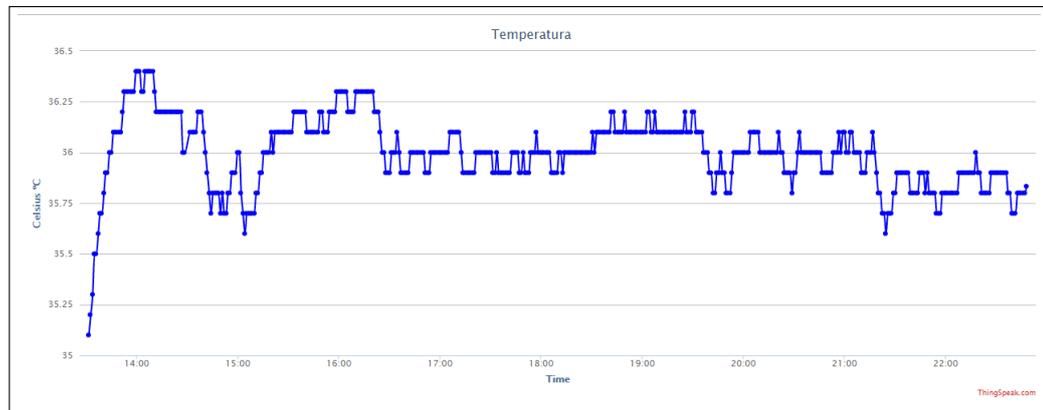
Fonte – Elaborada pelo autor

descartados os primeiros dez minutos de amostras, foram considerados *outliers* pois o microclima do ninho sofre uma variação de valores em resposta a abertura da colmeia, vale ressaltar que para a escolha dos outliers não foi utilizado nenhum método, somente foi retirado os primeiro 10 valores por estarem muito diferentes dos valores coletados..

Como mostrado na figura 15, os valores coletados ficaram em uma faixa de valores entre 35 a 36.5 graus Celsius. O aplicativo apresenta que o maior valor coletado foi de 36.4 graus Celsius entre 14 e 14:10hs. O menor valor foi de 35.4 graus Celsius, coletado por volta de 13:35 e a média ficou em 36 graus. Os dados de temperatura coletados corresponderam a faixa que ALMEIDA (2008) indica como ideal para o microclima de uma colmeia, em torno de 33 graus a 36 graus.

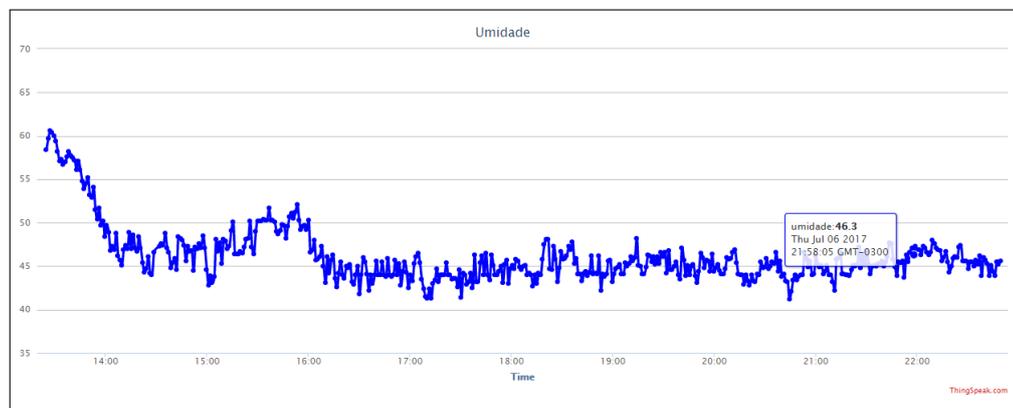
A figura 16 apresenta o histórico dos valores de umidade coletados. A umidade relativa ficou em torno de 42% a 59%, tendo como maior valor 58.2%, como menor valor 41.2% e apresentou uma média de 46%.

Figura 15 – Histórico de coletas de temperatura



Fonte – Elaborada pelo autor

Figura 16 – Histórico de coletas de umidade



Fonte – Elaborada pelo autor

O sensor de ruído foi implantado juntamente ao de temperatura e umidade mas como foi citado na seção 4.1.1, o sensor KY-038 requer uma calibragem para obter dados em decibéis, sua unidade de medida padrão. Entretanto, essa calibragem não foi concluída até a realização dos experimentos.

6 CONSIDERAÇÕES FINAIS

Esse trabalho apresentou o MOCIoT, um sistema minimamente invasivo e de baixo custo que utiliza de tecnologias como redes de sensores, Internet das Coisas e Computação em nuvem para prover uma ferramenta de monitoramento em tempo real de grandezas importantes de uma colmeia e que apresenta um aplicativo na plataforma Android que expõe os dados coletados de forma simples e organizada. Possibilitando ao apicultor uma opção que lhe permita um auxílio no cuidado do apiário e nas tomadas de decisões, evitando perdas nas colmeias e possibilitando um aumento na produtividade das abelhas.

Como trabalhos futuros, pretende-se realizar a calibragem do sensor de ruído e adição de um sensor de dióxido de carbono, e com isso, agregar mais valor ao sistema, sendo capaz de monitorar uma gama maior de variáveis de uma colmeia. Também pretendemos utilizar um painel solar para alimentação dos nós através de energia renovável com objetivo de aumentar a autonomia de energia do MOCIoT e utilizar internet móvel para o envio dos dados coletados, sem a necessidade de uma conexão cabeada. O principal ensinamento que ficou com o desenvolvimento desse trabalho foi a importância do monitoramento de colmeias e que essa atividade pode ser realizada utilizando de materiais de baixo custo, podendo evitar inúmeros problemas.

Todos os códigos fonte embarcados no nó sensor e gateway, o projeto do servidor web e aplicação móvel, *script* para criação de entidades no Fiware e arquivos de configuração do Cygnus estão disponíveis na GitHub <<https://github.com/alissonlimasilva/smartbee>>. Uma melhor visualização dos gráficos referentes ao nó sensor¹, gateway², temperatura³ e umidade⁴ estão disponíveis nos links do rodapé. O aplicativo está disponível na PlayStore através do endereço <https://play.google.com/store/apps/details?id=com.ufc.alima.mociot&hl=pt_BR>. Os arquivos de configuração do Cygnus estão disponíveis no GitHub <<https://github.com/alissonlimasilva/smartbee/tree/master/Fiware/Cygnus>>.

¹ Tensão nó sensor: <<http://bit.ly/2tWHQpn>>

² Tensão gateway: <<http://bit.ly/2szXV0T>>

³ Gráficos de temperatura: <<http://bit.ly/2tWneOn>>

⁴ Gráfico de umidade: <<http://bit.ly/2sWY2mv>>

REFERÊNCIAS

- ABREU, C. S. P. de; PEREZ, A. L. F. Redução do consumo energético de um sensor sem fio utilizado na automação de um abrigo de cultivo. **Anais do Computer on the Beach**, p. 220–229, 2017.
- ALMEIDA, G. F. D. **Fatores que interferem no comportamento enxameatório de abelhas africanizadas**. Tese (Doutorado) — Universidade de São Paulo, 2008.
- ALMEIDA, H. **Computação Brasil**. 2012.
<<http://sbc.org.br/publicacoes-2/298-computacao-brasil>>. Acessado em 17/06/2017.
- BENCSIK, M.; BENCSIK, J.; BAXTER, M.; LUCIAN, A.; ROMIEU, J.; MILLET, M. Identification of the honey bee swarming process by analysing the time course of hive vibrations. **Computers and electronics in agriculture**, Elsevier, v. 76, n. 1, p. 44–50, 2011.
- BROWN, M. J.; PAXTON, R. J. The conservation of bees: a global perspective. **Apidologie**, EDP Sciences, v. 40, n. 3, p. 410–416, 2009.
- CARVALHO, M. D. F. d. **Temperatura da superfície corpórea e perda de calor por convecção em abelhas (*Apis mellifera*) em uma região semi-árida**. Dissertação (Mestrado) — Universidade Federal Rural do Semi-Árido, 2009.
- CHODOROW, K. **MongoDB: the definitive guide**. [S.l.]: "O'Reilly Media, Inc.", 2013.
- DOMINGOS, H. G. T.; GONÇALVES, L. S. Termorregulação de abelhas com ênfase em *apis mellifera*. **Acta Veterinaria Brasilica**, v. 8, n. 3, p. 150–154, 2014.
- EUGSTER, P. T.; FELBER, P. A.; GUERRAOU, R.; KERMARREC, A.-M. The many faces of publish/subscribe. **ACM computing surveys (CSUR)**, ACM, v. 35, n. 2, p. 114–131, 2003.
- FERNÁNDEZ, P.; SANTANA, J. M.; ORTEGA, S.; TRUJILLO, A.; SUÁREZ, J. P.; DOMÍNGUEZ, C.; SANTANA, J.; SÁNCHEZ, A. Smartport: a platform for sensor data monitoring in a seaport based on fiware. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 16, n. 3, p. 417, 2016.
- FITZGERALD, D. W.; MURPHY, F. E.; WRIGHT, W. M.; WHELAN, P. M.; POPOVICI, E. M. Design and development of a smart weighing scale for beehive monitoring. In: IEEE. **Signals and Systems Conference (ISSC), 2015 26th Irish**. [S.l.], 2015. p. 1–6.
- FIWARE-COSMOS. **BigData Analysis - Cosmos**. 2017.
<<https://catalogue.fiware.org/enablers/bigdata-analysis-cosmos>>. Acessado em 23/06/2017.
- FIWARE-ORION. **Publish/Subscribe Context Broker - Orion Context Broker**. 2017.
<<https://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker>>. Acessado em 23/06/2017.
- FREITAS, D. G. F.; KHAN, A. S.; SILVA, L. M. R. Nível tecnológico e rentabilidade de produção de mel de abelha (*apis mellifera*) no ceará. **Revista de Economia e Sociologia Rural**, SciELO Brasil, v. 42, n. 1, p. 171–188, 2004.

HUNKELER, U.; TRUONG, H. L.; STANFORD-CLARK, A. Mqtt-s—a publish/subscribe protocol for wireless sensor networks. In: IEEE. **Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on.** [S.l.], 2008. p. 791–798.

KANDEPI, M. **Bee hive temperature and sound monitor.** Dissertação (Mestrado) — Graduate school of Cornell University, 2015.

KRIDI, D. S.; CARVALHO, C. G. N. d.; GOMES, D. G. A predictive algorithm for mitigate swarming bees through proactive monitoring via wireless sensor networks. In: ACM. **Proceedings of the 11th ACM symposium on Performance evaluation of wireless ad hoc, sensor, & ubiquitous networks.** [S.l.], 2014. p. 41–47.

KRONENBERG, F.; HELLER, H. C. Colonial thermoregulation in honey bees (*apis mellifera*). **Journal of Comparative Physiology B: Biochemical, Systemic, and Environmental Physiology**, Springer, v. 148, n. 1, p. 65–76, 1982.

KVIESIS, A.; ZACEPINS, A. System architectures for real-time bee colony temperature monitoring. **Procedia Computer Science**, Elsevier, v. 43, p. 86–94, 2015.

LEWIS, T. W. Beehive monitor. 2014.

LOLI, D. **Termorregulação colonial e energética individual em abelhas sem ferrão *Melipona quadrifasciata* Lepelletier (Hymenoptera, Apidae, Meliponini).** Tese (Doutorado) — Universidade de São Paulo, 2008.

MARDAN, M.; KEVAN, P. G. Critical temperatures for survival of brood and adult workers of the giant honeybee, *apis dorsata* (hymenoptera: Apidae). **Apidologie**, EDP SCIENCES, v. 33, n. 3, p. 295–302, 2002.

MARTÍNEZ, R.; PASTOR, J. Á.; ÁLVAREZ, B.; IBORRA, A. A testbed to evaluate the fiware-based iot platform in the domain of precision agriculture. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 16, n. 11, p. 1979, 2016.

MEZQUIDA, D. A.; MARTÍNEZ, J. L. Platform for bee-hives monitoring based on sound analysis. a perpetual warehouse for swarm apos; s daily activity. **Spanish Journal of Agricultural Research**, v. 7, n. 4, p. 824–828, 2009.

MOLTCHANOV, B.; ROCHA, O. R. Pub/sub and semantic annotation enablers for future internet in the ppp eu project test-bed. In: **AAAI 2013 Workshop on Context-Aware System Architectures.** [S.l.: s.n.], 2013. p. 48–52.

MONGODB. **Introduction to MongoDB.** 2017.

<<https://docs.mongodb.com/manual/introduction/>>. Acessado em 14/06/2017.

MURPHY, F. E.; MAGNO, M.; O’LEARY, L.; TROY, K.; WHELAN, P.; POPOVICI, E. M. Big brother for bees (3b)—energy neutral platform for remote monitoring of beehive imagery and sound. In: IEEE. **Advances in Sensors and Interfaces (IWASI), 2015 6th IEEE International Workshop on.** [S.l.], 2015. p. 106–111.

NAMIOT, D.; SNEPS-SNEPPE, M. On software standards for smart cities: Api or dpi. In: IEEE. **ITU Kaleidoscope Academic Conference: Living in a converged world-Impossible without standards?, Proceedings of the 2014.** [S.l.], 2014. p. 169–174.

- PAN, J.; PAUL, S.; JAIN, R. A survey of the research on future internet architectures. **IEEE Communications Magazine**, IEEE, v. 49, n. 7, 2011.
- POTTS, S. G.; BIESMEIJER, J. C.; KREMEN, C.; NEUMANN, P.; SCHWEIGER, O.; KUNIN, W. E. Global pollinator declines: trends, impacts and drivers. **Trends in ecology & evolution**, Elsevier, v. 25, n. 6, p. 345–353, 2010.
- RAMOS, J. M.; CARVALHO, N. d. Estudo morfológico e biológico das fases de desenvolvimento de *apis mellifera*. **Revista Científica Eletrônica de Engenharia Florestal**, v. 6, n. 10, p. 1–21, 2007.
- RAMPARANY, F.; MARQUEZ, F. G.; SORIANO, J.; ELSALEH, T. Handling smart environment devices, data and services at the semantic level with the fi-ware core platform. In: IEEE. **Big Data (Big Data), 2014 IEEE International Conference on**. [S.l.], 2014. p. 14–20.
- RAO, B. P.; SALUIA, P.; SHARMA, N.; MITTAL, A.; SHARMA, S. V. Cloud computing for internet of things & sensing based applications. In: IEEE. **Sensing Technology (ICST), 2012 Sixth International Conference on**. [S.l.], 2012. p. 374–380.
- RODRIGUES, W. C. Fatores que influenciam no desenvolvimento dos insetos. **Info Insetos**, v. 1, n. 4, p. 14–17, 2004.
- SALLES, M. **Produção de abelhas in vitro abre novo horizonte para polinização em lavouras**. 2014. <<http://revistagloborural.globo.com/Noticias/Agricultura/noticia/2014/09/producao-de-abelhas-vitro-abre-novo-horizonte-para-polinizacao-em-lavouras.html>>. Acessado em 14/06/2017.
- SANTUCCI, G. The internet of things: Between the revolution of the internet and the metamorphosis of objects. **Vision and Challenges for Realising the Internet of Things**, p. 11–24, 2010.
- WILLIAMS, G. R.; ALAUX, C.; COSTA, C.; CSAKI, T.; DOUBLET, V.; EISENHARDT, D.; FRIES, I.; KUHN, R.; MCMAHON, D. P.; MEDRZYCKI, P. et al. Standard methods for maintaining adult *apis mellifera* in cages under in vitro laboratory conditions. **Journal of Apicultural Research**, Taylor & Francis, v. 52, n. 1, p. 1–36, 2013.
- ZACEPINS, A.; STALIDZANS, E.; MEITALOVS, J. Application of information technologies in precision apiculture. In: **Proceedings of the 13th International Conference on Precision Agriculture (ICPA 2012)**. [S.l.: s.n.], 2012.

APÊNDICE A – IMPLEMENTAÇÃO DE REPETIDOR

Durante as tentativas de implantação, surgiu um problema de comunicação entre o nó sensor e gateway. Devido ao apiário está localizado em uma mata com muitas árvores e obstáculos, os dados coletados pelo nó sensor não estavam chegando ao Gateway. Então vimos a necessidade de utilizar um nó que intermediasse a comunicação entre o nó sensor e o gateway, um repetidor de sinal. Entretanto, mesmo com a adição desse nó na arquitetura, também não foi possível estabelecer a comunicação entre o nó sensor e o repetidor. O nó repetidor era constituído por um Arduino Uno, o módulo a radio NRF24L01+ e sua alimentação era feita por uma bateria de 12 volts e um regulador de tensão. Abaixo é apresentado o código embarcada no Arduino Uno para implementar o repetidor.

```
1 #include <RF24Network.h>
2 #include <RF24.h>
3 #include <SPI.h>
4
5 RF24 radio(8,9);
6 RF24Network network(radio);
7 const uint16_t this_node = 01;
8 const uint16_t other_node = 00;
9 int SENSORTENSAO = A0;
10 float tensao_lida = 0;
11 struct payload_t {
12     char id[30];
13     float temperatura;
14     float umidade;
15     float co2;
16     float som;
17     float tensaocolmeia;
18     float tensaorepetidor;
19 };
20 void lerTensao() {
```

```
21 float valor_lido_tensao = analogRead(SENSORTENSAO);
22 tensao_lida=((valor_lido_tensao*0.004887586)*4.62);
23 }
24 void setup(void)
25 {
26     radio.begin();
27     radio.setPALevel(RF24_PA_MAX);
28     radio.setDataRate(RF24_250KBPS);
29     network.begin(90, this_node);
30 }
31
32 void loop(void) {
33     network.update();
34     while ( network.available() ) {
35         RF24NetworkHeader header;
36         struct payload_t payload;
37         network.read(header, &payload, sizeof(payload));
38         RF24NetworkHeader header_recebido(other_node);
39         bool ok = network.write(header_recebido, &payload, sizeof
40             (payload));
```

APÊNDICE B – CRIAÇÃO DE ENTIDADE

Este script realiza a criação de novas entidades no Fiware Orion. Ao executar esse script, é enviada uma mensagem HTTP POST contendo um JSON que é recebida pela interface NGSI do Orion na porta 1026.

```
1 #!/bin/bash
2 #Script tem que receber 7 parametros, senao nao executara
3 if [ $# -lt 7 ]; then
4     echo "./criar_entidade [id] [tipo] [temperatura] [umidade]
5         [som] [co2] [tensaocolmeia] [tensaogateway]"
6     exit 1
7 fi
8 (curl 35.184.233.148:1026/v1/updateContext -s -S --header
9     Content-Type: application/json \
10    --header Accept: application/json --header Fiware-
11    Service: tcc --header Fiware-ServicePath: / -d @- |
12    python -mjson.tool > resposta | grep "code") <<EOF
13 {
14     "contextElements": [
15         {
16             "type": "$2",
17             "isPattern": "false",
18             "id": "$1",
19             "attributes": [
20                 {
21                     "name": "temperatura",
22                     "type": "float",
23                     "value": "$3"},
24                 {
25                     "name": "umidade",
26                     "type": "float",
27                     "value": "$4"},
28                 {
29                     "name": "data",
```

```
23         "type": "string",
24         "value": "26/04/2017"},
25     {
26         "name": "hora",
27         "type": "string",
28         "value": "11:42:51"},
29     {
30         "name": "som",
31         "type": "float",
32         "value": "$5"},
33     {
34         "name": "dioxido",
35         "type": "float",
36         "value": "$6"},
37     {
38         "name": "tensaocolmeia",
39         "type": "float",
40         "value": "$7"},
41     {
42         "name": "tensaogateway",
43         "type": "float",
44         "value": "$8"}
45     ]
46 }
47 ],
48 "updateAction": "APPEND"
49 }
50 EOF
```