



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM TECNOLOGIA EM REDES DE COMPUTADORES

JULIANA DE CASTRO RABELO

**NOTIFYME! UM SISTEMA DE NOTIFICAÇÕES DE GRANDEZAS FÍSICAS: UM
ESTUDO DE CASO COM O MONITORAMENTO DE COLMEIAS**

QUIXADÁ
2018

JULIANA DE CASTRO RABELO

NOTIFYME! UM SISTEMA DE NOTIFICAÇÕES DE GRANDEZAS FÍSICAS: UM
ESTUDO DE CASO COM O MONITORAMENTO DE COLMEIAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Tecnologia em Redes de Computadores do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de tecnologia em Tecnologia em Redes de Computadores.

Orientador: Prof. Me. Antonio Rafael Braga

QUIXADÁ

2018

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)

R114n Rabelo, Juliana de Castro.

Notifyme! Um sistema de notificações de grandezas físicas : um estudo de caso com o monitoramento de colmeias / Juliana de Castro Rabelo. – 2018.

71 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Redes de Computadores, Quixadá, 2018.

Orientação: Prof. Me. Antonio Rafael Braga.

1. Notificações. 2. Apicultura. 3. Monitoramento ambiental. I. Título.

CDD 004.6

JULIANA DE CASTRO RABELO

NOTIFYME! UM SISTEMA DE NOTIFICAÇÕES DE GRANDEZAS FÍSICAS: UM
ESTUDO DE CASO COM O MONITORAMENTO DE COLMEIAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Tecnologia em Redes de Computadores do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de tecnóloga em Tecnologia em Redes de Computadores.

Aprovada em: __/__/__

BANCA EXAMINADORA

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

Prof. Dr. Arthur de Castro Callado
Universidade Federal do Ceará (UFC)

Prof. Dra. Atslands Rego da Rocha
Universidade Federal do Ceará (UFC)

À todas as mulheres, pela coragem, lutas que enfrentaram e contribuições que deixaram.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela vida e paciência que destinou a mim.

Agradeço à minha mãe, Diomar, e ao meu pai, Paulo Nogueira, que, apesar de não entenderem muitas vezes que era necessário eu trilhar por um caminho pensado por mim, fizeram sacrifícios para que eu pudesse possuir uma boa educação e por me darem tempo para estudar.

Agradeço ao Prof. Me. Antonio Rafael Braga, por ter me aceitado como orientanda, pela paciência, e pela excelente orientação que permitiu a realização deste trabalho.

Agradeço ao meu irmão, Diego, que, apesar de me perturbar muito perguntando quando eu iria trabalhar, sempre esteve ao meu lado quando precisei.

Agradeço aos meus amigos, principalmente ao Fábio e Roseli, pela ajuda, paciência, e amizade durante o curso. E ao Júlio, Ícaro, Sting, Pedro, Luan, Alisson, Mateus, Mateus Sousa, Micaele, Tiago, Isac, Renan, Walafi, Ana Lúcia, João Batista, Caleb, Leilton, Samuel, Wilkinson, Cleilson e Sara, por me aguentarem todo esse tempo, escutando minhas reclamações e me ajudando quando necessário.

Agradeço aos professores Dr. Arthur Callado e Dra. Atslands Rocha, pela disponibilidade em participar da banca desse trabalho.

Agradeço aos professores Paulo Rego, Lívia Almada, Ticiania Linhares, João Marcelo, Marcos Dantas e Alisson Barbosa, pelos conselhos, ensinamentos e pelo trabalho que exercem com excelência no campus UFC-Quixadá.

Por fim, agradeço-me, pela pessoa que sou e pela perseverança que tenho.

“Mesmo que eu possua o dom de profecia e conheça todos os mistérios e toda a ciência, e ainda tenha uma fé capaz de mover montanhas, se não tiver amor, nada serei.”

(Coríntios 13:2)

RESUMO

A apicultura possui um grande impacto em todo o campo agrícola, pois as abelhas são os principais insetos polinizadores e desempenham um papel importante na produção de culturas inteiras e na sobrevivência de plantas selvagens. Além disso, a apicultura pode ser vista também como uma fonte secundária de renda, principalmente para as pequenas famílias rurais. Essas famílias são os responsáveis pela maior parte da produção nacional de mel. Dentro desse contexto, existe um grande interesse por parte de pesquisadores, agricultores, ecologistas e decisores políticos em relação à saúde e ao comportamento das abelhas. Diversos fatores podem influenciar na saúde e comportamento das mesmas, como estresses causados por patógenos, o manejo inadequado das colônias, uso de agrotóxicos, má nutrição e os fatores combinados. Uma das formas de reduzir o manejo inadequado das colmeias e monitorar a saúde das abelhas é o envio de notificações/alertas sobre dados coletados por meio de sensores. O presente estudo apresenta o *NotifyMe!*, uma solução de envio de notificações via Telegram, E-mail e SMS. As notificações alertam quanto ao nível de temperatura, umidade, som, dióxido de carbono, oxigênio, peso da colmeia e atraso na coleta. A partir desses dados, pesquisadores e apicultores podem ficar informados e realizar mudanças nos locais das colmeias, evitar catástrofes e possíveis doenças. Os resultados obtidos em relação ao tempo de processamento no envio de mensagens, mostrou que os envios realizados via SMS e Telegram possuem um menor tempo de processamento se comparado ao envio via E-mail. Em relação ao envio das notificações de acordo com as preferências do usuário, todas as notificações foram enviadas corretamente.

Palavras-chave: Notificações. Apicultura. Monitoramento ambiental.

ABSTRACT

Beekeeping has a major impact on the entire agricultural field since bees are the main pollinator insects and play an important role in the production of whole crops and the survival of wild plants. In addition, beekeeping can also be seen as a secondary source of income, especially for small rural families. These families are responsible for most of the national honey production. Within this context, there is a great interest on the part of researchers, farmers, ecologists and policy makers regarding bee health and behavior. Several factors can influence their health and behavior, such as stress caused by pathogens, inadequate management of colonies, use of pesticides, malnutrition and a combination of these factors. One of the ways to reduce inappropriate management of hives and monitor bee health is to send notifications/alerts on data collected through sensors. The present study presents NotifyMe!, a solution for sending notifications through Telegram, E-mail and SMS. The notifications warn about the level of temperature, humidity, sound, carbon dioxide, oxygen, hive weight and delay in collection. From this data, researchers and beekeepers can be informed and make changes in the locations of hives, avoid catastrophes and possible diseases. The results obtained in relation to the processing time in the sending of messages, showed that the messages sent via SMS and Telegram have a shorter processing time compared to the sending via E-mail. In regards to sending notifications according to user preferences, all notifications were sent correctly.

Keywords: Notifications. Beekeeping. Environmental monitoring.

LISTA DE FIGURAS

Figura 1 – Modelo Incremental	17
Figura 2 – Polinização de Abelhas	18
Figura 3 – Sistema Sm@rtbee	21
Figura 4 – Diagrama de Interação dos Módulos do <i>Beehiveior</i>	28
Figura 5 – Etapas funcionais do <i>software</i> para os índices de temperatura e umidade	29
Figura 6 – Módulo de Alertas	33
Figura 7 – Listagem das Configurações Cadastradas das Variáveis Coletadas	39
Figura 8 – Formulário de Edição das Configurações	39
Figura 9 – Formulário de Cadastro de Nova Notificação Via E-mail	40
Figura 10 – Formulário de Personalização das Grandezas	41
Figura 11 – Formulário de Cadastro de Nova Notificação Via Telegram	42
Figura 12 – Modal Explicativo Para a Criação de um <i>Username</i>	43
Figura 13 – Modal Explicativo Para o Envio de Mensagens Para o Bot	44
Figura 14 – Aviso de Sucesso no Cadastro de Username	44
Figura 15 – Listagem das Notificações Cadastradas Via E-mail	45
Figura 16 – Listagem das Notificações Enviadas Via E-mail	45
Figura 17 – Listagem das Notificações Enviadas Via SMS	45
Figura 18 – Listagem de Colmeias Disponíveis Para a Criação de uma Nova Notificação	46
Figura 19 – Listagem das Notificações Cadastradas	47
Figura 20 – Erro na Exclusão de uma Notificação	47
Figura 21 – Sucesso na Exclusão de uma Notificação	47
Figura 22 – Listagem das Notificações Enviadas	48
Figura 23 – Tempo de processamento A	54
Figura 24 – Tempo de processamento B	55
Figura 25 – Tempo de Processamento de Todos os Períodos	57
Figura 26 – Mensagens Recebidas Via E-mail	58
Figura 27 – Mensagens Recebidas Via SMS	59
Figura 28 – Mensagens Recebidas Via Telegram	59

LISTA DE TABELAS

Tabela 1 – Tecnologias de Envio de Notificações	23
Tabela 2 – Trabalhos Relacionados	34
Tabela 3 – Ferramentas e linguagens utilizadas	36
Tabela 4 – Distribuição dos Períodos Observados	52
Tabela 5 – Valores da Médias e Intervalos de Confiança da Média	56
Tabela 6 – Descrição das Legendas Utilizadas	66
Tabela 7 – Notificações do Capitão América	66
Tabela 8 – Notificações da Mulher Gato	67
Tabela 9 – Notificações do Batman	67
Tabela 10 – Configurações do Capitão América	67
Tabela 11 – Configurações da Mulher Gato	67
Tabela 12 – Configurações do Batman	67

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
<i>1.1.1</i>	<i>Objetivo Geral</i>	<i>14</i>
<i>1.1.2</i>	<i>Objetivos Específicos</i>	<i>14</i>
1.2	Organização	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Engenharia de Software	16
<i>2.1.1</i>	<i>Modelo Incremental</i>	<i>16</i>
2.2	Monitoramento Ambiental	17
<i>2.2.1</i>	<i>Monitoramento de Colmeias</i>	<i>18</i>
2.3	O Sistema Sm@rtbee	19
2.4	Tecnologias de Rede de Envio de Notificações	21
<i>2.4.1</i>	<i>Nagios</i>	<i>21</i>
<i>2.4.2</i>	<i>Zabbix</i>	<i>22</i>
2.5	Tecnologias de Rede Agregadas de Envio de Notificações	22
<i>2.5.1</i>	<i>E-mail</i>	<i>23</i>
<i>2.5.2</i>	<i>SMS</i>	<i>24</i>
<i>2.5.2.1</i>	<i>SMS Gateway</i>	<i>25</i>
<i>2.5.3</i>	<i>Mensagens Instantâneas</i>	<i>25</i>
<i>2.5.3.1</i>	<i>Telegram</i>	<i>25</i>
<i>2.5.3.1.1</i>	<i>Bots do Telegram</i>	<i>26</i>
3	TRABALHOS RELACIONADOS	27
3.1	<i>Beehiveior</i> - Sistema de monitoramento e controle de colmeias de produção apícola	27
3.2	Apropriação intelectual da tecnologia (<i>software e hardware</i>) do controle de qualidade da cadeia industrial de mel	28
3.3	<i>Developing an underwater monitoring and alert system for water-borne disease management</i>	29
3.4	Alerta poá - um sistema de comunicação apoiado em redes de sensores sem fio para monitoração do córrego itaim	30

3.5	<i>Interfacing atmospheric variables to web interface using arduino</i>	31
3.6	<i>Web based online bakery system with short messaging service and email notification</i>	32
3.7	Um modelo proativo de antecipação de ações de times de resposta rápida baseado em análise preditiva	33
4	METODOLOGIA	35
4.1	Análise dos Requisitos	35
4.2	Desenvolvimento do Sistema Notifyme	35
4.3	Testes e validação do sistema	36
5	SISTEMA DE NOTIFICAÇÕES NOTIFYME	37
5.1	Requisitos	37
5.1.1	<i>Apresentação do Sistema</i>	38
5.1.1.1	<i>Configuração de Grandezas</i>	38
5.1.1.2	<i>Cadastrros de Novas Notificações</i>	40
5.1.1.3	<i>Listagem, Edição e Exclusão das Notificações</i>	42
5.1.1.4	<i>Listagem das Notificações Enviadas</i>	43
5.1.1.5	<i>Cadastro, listagens, edições e exclusões do Administrador</i>	46
5.1.1.6	<i>Scripts de Envio das Notificações</i>	48
6	RESULTADOS E DISCUSSÃO	51
6.1	Planejamento e Execução	51
6.2	Resultados dos Experimentos	52
7	CONCLUSÕES E TRABALHOS FUTUROS	60
	REFERÊNCIAS	61
	APÊNDICES	66
	APÊNDICE A – Experimento A	66
	APÊNDICE B – Código-fonte utilizado para o envio das coletas	69

1 INTRODUÇÃO

A apicultura é um dos ramos tradicionais da agricultura, e possui um enorme impacto em todo o campo agrícola, pois as abelhas são os principais insetos polinizadores e desempenham um papel importante na produção de culturas inteiras e na sobrevivência de plantas selvagens (ZACEPINS *et al.*, 2016). De acordo com Kviešis e Zacepins (2015) as abelhas podem polinizar cerca de 85% das flores de maneira geral, o que as torna um dos insetos mais importantes para a agricultura.

A apicultura pode ser vista também como uma fonte secundária de renda, principalmente para as pequenas famílias rurais. Essas famílias são os responsáveis pela maior parte da produção nacional de mel e representam uma parcela significativa na geração de renda oriunda do extrativismo apícola (ALMEIDA *et al.*, 2013).

De acordo com ADECE (2018), a apicultura no estado do Ceará caracteriza-se, quase que exclusivamente, pela produção de mel de abelhas africanizadas (*Apis mellifera L.*). Já o perfil da maioria dos apicultores cearenses é de agricultor familiar, possuindo em média apenas 35 colmeias. A apicultura familiar está presente em quase todos os municípios do Estado, gerando renda e ocupação para muitas famílias de baixa renda.

No que diz respeito a produção de mel no Brasil, no ano de 2016 foi de 39,59 mil toneladas, com alta de 5,1% em relação ao ano de 2015. Já o valor da produção foi de R\$ 470,51 milhões, com alta de 31,5%. O principal responsável pela produção nacional foi o Rio Grande do Sul com 15,8%, apresentando uma alta de 26,6% em relação a 2015. Em segundo o Paraná, onde a produção caiu 4,7% devido ao excesso de chuvas e à contaminação por inseticidas (IBGE, 2017). Já o estado do Ceará se encontra na nona posição, representando aproximadamente 2,9% da produção (IBGE, 2016).

As colônias de abelhas podem sofrer problemas como o *Colony Collapse Disorder* CDD ou distúrbio do colapso das colônias que se caracteriza pela “rápida perda de abelhas operárias, ausência de crias e abelhas adultas mortas dentro ou fora da colmeia e invasão da colmeia por pragas” (PIRES *et al.*, 2016). Esse distúrbio tem como principais fatores estresses causados por patógenos, o manejo inadequado das colônias, uso de agrotóxicos, má nutrição e os fatores combinados (PIRES *et al.*, 2016). Para reduzir tais fatores e realizar a avaliação da saúde das colônias, os apicultores e cientistas começaram a monitorar colmeias (KVIŠIS; ZACEPINS, 2015).

Dentro desse cenário, há um crescente interesse nas vidas e no comportamento

das abelhas (KVIESIS; ZACEPINS, 2015), e a saúde das mesmas se constitui como uma crescente preocupação entre pesquisadores, ecologistas, agricultores e ainda decisores políticos (ZACEPINS *et al.*, 2016).

Graças ao rápido desenvolvimento de tecnologias que utilizam sensores, os estados de colônias de abelhas podem ser monitorados remotamente, reduzindo conseqüentemente as inspeções manuais. Sendo assim, existe um grande interesse no desenvolvimento de novos métodos não invasivos que possam contribuir para avaliar o estado da colônia (ZACEPINS *et al.*, 2016). Uma das formas de monitorar a saúde das abelhas pode ser o envio de notificações/alertas sobre dados coletados por meio de sensores. Dentro do contexto de monitoramento remoto e envio de notificações, o presente estudo buscou resolver o seguinte problema de pesquisa: como informar os apicultores, pesquisadores e gestores sobre a saúde das abelhas sem a necessidade de atenção constante a um sistema de monitoramento *web*?

Portanto, o presente estudo, criou um sistema de notificações denominado *Notifyme!* acoplado ao sistema *Sm@rtbee* em desenvolvimento para o monitoramento de colmeias. O mesmo realiza o envio de notificações pré-configuradas no sistema *Sm@rtbee* pelos usuários. As tecnologias de envio são: E-mail, Telegram e SMS (*Short Message Service*). As notificações alertam quanto ao nível de temperatura, umidade, som, dióxido de carbono, oxigênio, peso da colmeia e atraso na coleta. A partir desses dados, pesquisadores e apicultores podem ficar informados e realizar mudanças nos locais das colmeias, evitar catástrofes e possíveis doenças.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um sistema de notificações de grandezas para um Apiário.

1.1.2 Objetivos Específicos

- Integrar o sistema de notificações ao sistema de gerenciamento *Sm@rtbee*.
- Criar as configurações de grandezas de acordo com as preferências dos usuários.

1.2 Organização

A seguir são apresentados os conceitos da fundamentação teórica no Capítulo 2. O Capítulo 3 traz os trabalhos relacionados. Já o Capítulo 4 apresenta a metodologia do presente estudo. O Capítulo 5 apresenta o sistema proposto, o Capítulo 6 a avaliação do sistema e o Capítulo 7 as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo são discutidas as tecnologias utilizadas para a implementação do sistema.

2.1 Engenharia de *Software*

A engenharia de *software* engloba um conjunto de processos, métodos e técnicas de forma sistemática, disciplinada, organizada e qualificada para a construção de um produto final de qualidade (GOMES, 2014).

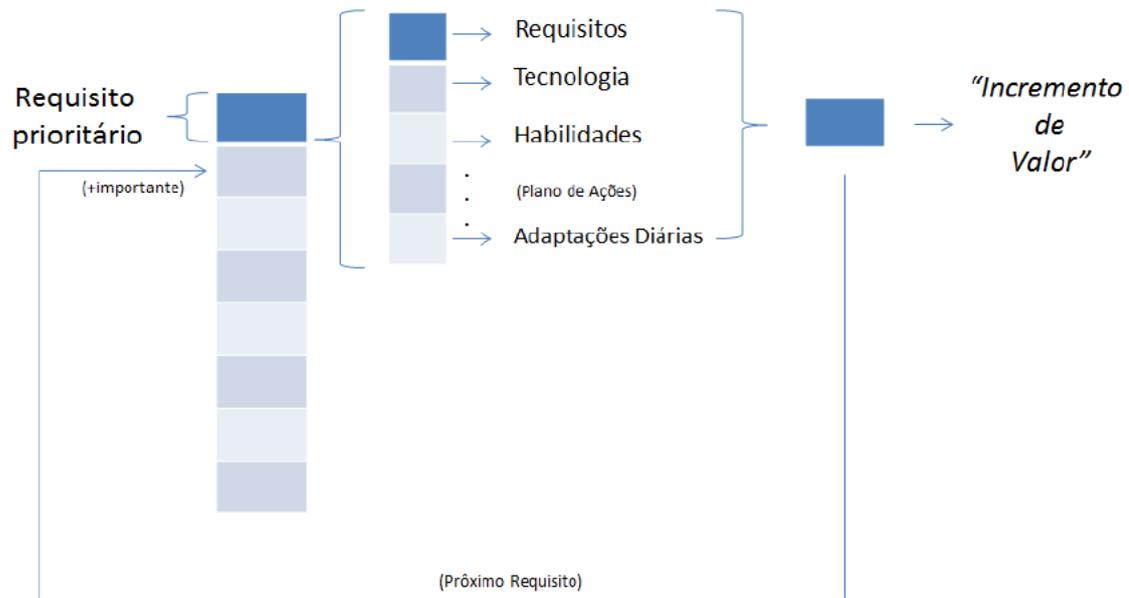
Utilizando a abordagem de engenharia de *software*, na fase inicial o problema a ser tratado deve ser analisado e decomposto em sub problemas. Após isso, uma solução deve ser elaborada para cada sub problema. Por fim, solucionados os sub problemas deve-se integrar as soluções. Para realizar essas ações, uma arquitetura deve ser estabelecida. E “para apoiar a resolução de problemas, procedimentos, métodos e técnicas devem ser utilizados, bem como ferramentas para parcialmente automatizar o trabalho” (GOMES, 2014). Sendo assim, o modelo de processo que se encaixa ao presente estudo é o modelo incremental.

2.1.1 Modelo Incremental

O modelo incremental é um modelo de processo que permite a liberação de uma serie de versões denominadas “incrementos” aos clientes. Inicialmente, os requisitos básicos são atendidos, porém os requisitos complementares ainda não são entregues. A primeira entrega é o produto mínimo viável. O mesmo pode ser utilizado ou pode passar por uma avaliação detalhada do cliente. Com o resultado de uso/avaliação são realizados planejamentos para o incremento seguinte. No planejamento, já são consideradas as mudanças no produto essencial para se adequar as necessidades do cliente e a entrega de recursos e funcionalidades adicionais. Esse processo é repetido após a entrega de cada incremento até a entrega do produto final (PRESSMAN; MAXIM, 2016).

Conforme Figura 1, “a cada iteração a equipe analisa os requisitos, a tecnologia e suas habilidades e então se dividem para construir e entregar o melhor produto possível adaptando-se diariamente conforme surjam as complexidades e dificuldades” (SILVA; MELO, 2016).

Figura 1 – Modelo Incremental



Fonte: Silva e Melo (2016).

2.2 Monitoramento Ambiental

O homem provocou e vem provocando diversas alterações no ambiente, transformando paisagens e, conseqüentemente, reduzindo a biodiversidade. Uma das conseqüências é a possibilidade de exclusão de espécies-chave dos ecossistemas, o que pode afetar na flora, na fauna, nas relações ecológicas entre os organismos e prejudicar cada vez mais a qualidade de vida no planeta (OLIVEIRA *et al.*, 2015).

Dentro desse contexto, o monitoramento ambiental se faz necessário, seja ele manual, ou seja, com o uso de instrumentos de medidas, e/ou automatizado por meio de equipamentos ou sensores de medidas que atuam sem intervenção constante. O monitoramento ambiental consiste em um conjunto de observações qualitativas e quantitativas, contínuas ou periódicas, de parâmetros ambientais utilizados para a realização da gestão ambiental (OLIVEIRA; PORTEZANI, 2015). Ele pode ser realizado em micro e/ou macro escalas. Em micro escala pode ser realizado em pequenas e limitadas áreas geográficas, como por exemplo o monitoramento de temperatura em um apiário (DUTRA, 2016) ou na análise dos poluentes emitidos por uma chaminé industrial. Já em macro escala é realizado em áreas geográficas mais amplas, como a análise da qualidade da água ao longo de todo o trajeto de um rio (OLIVEIRA; PORTEZANI, 2015) ou a verificação da qualidade da água, através do uso de sensores para a medição de índices como pH, temperatura, turbidez e condutividade elétrica como apresentada por Almeida *et al.* (2018).

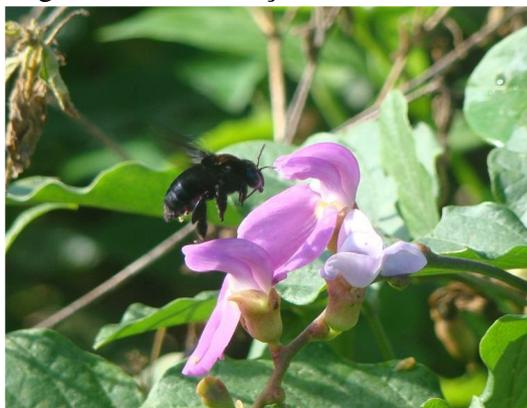
2.2.1 Monitoramento de Colmeias

No estudo literário realizado por Oliveira *et al.* (2015) sobre os bioindicadores ambientais, os pesquisadores mostraram como um dos interesses o grupo *Hymenoptera*, que são representados pelas abelhas, formigas e vespas. Esse grupo possui cerca de 115.000 espécies descritas, e várias delas com importância comprovada no ambiente. Dentre essas espécies, podem ser citadas as vespas e abelhas solitárias, que “são importantes pelas interações que mantêm com outros grupos de organismos (parasitismo, predação, polinização, dispersão) e também porque as variações espaciais e temporais de sua abundância e riqueza têm demonstrado correlações importantes com as alterações na estrutura do ambiente e com a diversidade de outros organismos”.

Já abelhas sem ferrão do gênero *Melipona*, especificamente na floresta tropical de Rondônia, são sensíveis ao desmatamento. De modo geral, os insetos deste grupo são sensíveis às mudanças ecológicas. Essa sensibilidade tem sido utilizada como ferramenta para avaliar a qualidade dos ecossistemas, em áreas de mineração, de agricultura, ou urbanas (OLIVEIRA *et al.*, 2015).

As abelhas, especificamente, são uma parte importante do nosso eco-sistema. Polinizadores, como elas, ajudam pelo menos 30% das lavouras do mundo e até 90% das plantas selvagens do mundo a prosperar através da polinização cruzada (LOFARO, 2017). A polinização realizada pelos insetos ocorre quando estes visitam flores diferentes (POLIZEL *et al.*, 2018), a Figura 2 mostra a visitação de uma abelha em uma flor.

Figura 2 – Polinização de Abelhas



Fonte: Brito *et al.* (2018).

De acordo com Lofaro (2017) estudos mostram que entre os anos de 2015 e 2016 durante o inverno os apicultores perderam 44% das colônias de abelhas, o que representa um

aumento de 10% em relação à década anterior. As causas dessa queda estão relacionadas à saúde geral da colmeia. Quanto mais saudável é a colmeia, maior a probabilidade dela sobreviver ao inverno.

A principal técnica de inspeção usada atualmente, ou seja, inspeções manuais, exigem que as pessoas verifiquem as colmeias quanto a presença de infestações de ácaros *Varroa*, baixos níveis de umidade entre outros sinais que revelam que uma colmeia não está saudável. Apesar dessas inspeções acontecerem uma ou duas vezes por ano, elas são invasivas para as colônias de abelhas (LOFARO, 2017). Para reduzir essas perdas, estudos como de Lofaro (2017) que propõem criar um sistema de monitoramento denominado *Smart Hive*, que traz os dados de sensores acoplados nas colmeias, tais dados são disponíveis na nuvem para auxiliar os apicultores a manterem suas colmeias saudáveis. Outro estudo que realizou o monitoramento de colmeias foi Dutra (2016), que criou um sistema denominado *Beehiveior*, com o objetivo de auxiliar pesquisas na área apícola.

2.3 O Sistema Sm@rtbee

No estudo de Silva (2017), foi criada uma arquitetura para realizar o monitoramento de colmeias denominado MOCIoT (Monitoramento de Colmeia com IoT). A mesma é composta por 4 elementos principais: sensores, *gateway*, *middleware*, *webservice* e aplicação móvel.

Na arquitetura MOCIoT a placa Arduino possui módulos sensores conectados a ela que atualmente coletam grandezas como: temperatura, umidade e ruído (SILVA, 2017). Porém, serão implementados posteriormente sensores para medição de dióxido de carbono, oxigênio e peso. O Arduino realiza o processamento dos dados lidos pelos sensores e utiliza um módulo transmissor à rádio para transmitir esses dados para o *gateway* através de ondas de rádio na frequência 2.4 GHz. Cada colmeia monitorada possui um nó sensor com seus sensores localizados dentro da colmeia. As leituras são realizadas periodicamente em um intervalo de tempo de 5 minutos para economia de bateria (SILVA, 2017).

O Arduino processa os dados recebidos e os escreve em uma porta serial no *Beaglebone Black* (BBB) por meio de uma porta USB. Ao mesmo tempo, no BBB, um *script* escrito na linguagem *Python* realiza a leitura e processamento dos dados escritos na porta serial, e envia-os para o *Middleware*. Esse envio é realizado utilizando a biblioteca *Python-requests* que envia um JSON (*JavaScript Object Notation*) através de uma mensagem HTTP (*Hypertext Transfer Protocol*) à porta 1026. De forma resumida, o nó *Gateway* recebe os dados coletados

pelos nós sensores e envia-os diretamente para o *Middleware* na nuvem. O *middleware Fiware* é o ambiente de processamento e armazenamento de dados, para um melhor entendimento do funcionamento do *middleware* é recomendável a leitura do estudo de Silva (2017).

De acordo com Silva (2017) o *webservice* utiliza um banco de dados para realizar consultas SQL (*Structured Query Language*) e disponibilizar através de uma API RESTb (*Representational State Transfer*) ou Transferência de Estado Representacional, essas informações para fácil acesso pelas aplicações móveis. A aplicação móvel realiza a comunicação com o *webservice* trocando mensagens JSON através de protocolo HTTP.

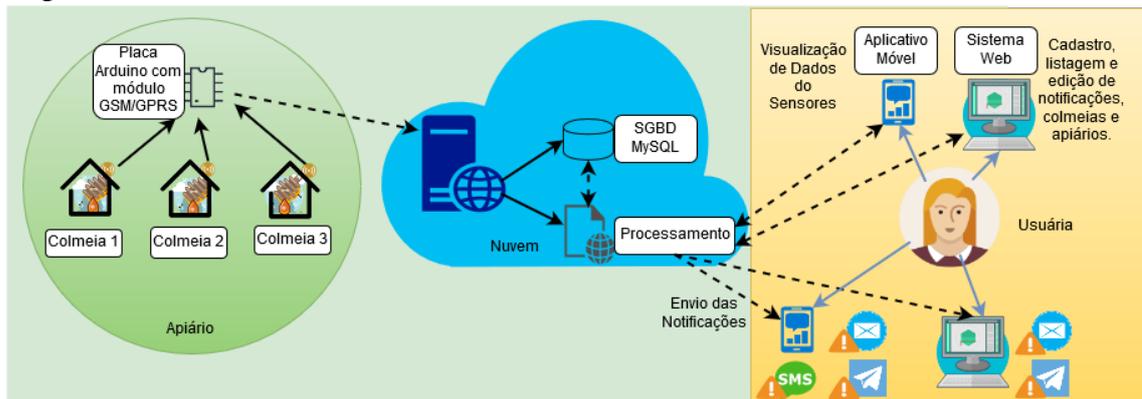
O aplicativo móvel foi desenvolvido na plataforma Android, com versão mínima SDK 16 (Jelly Bean 4.1). O mesmo é capaz de monitorar em tempo real os valores coletados pelos sensores. Ele obtém dados em formato JSON através da API Retrofit, fazendo requisições HTTP GET ao *webservice* (SILVA, 2017). Atualmente o aplicativo está na versão 5.0 e disponível na *Play Store*.

Atualmente a arquitetura que está sendo utilizada pelo sistema *web Sm@rtbee* possui os sensores, a placa Arduino e conectada a ela se encontra um módulo GSM (*Global Service for Mobile*)/GPRS (*General Packet Radio Service*). O módulo GSM/GPRS utiliza de um chip de operadora de telefonia móvel para comunicação, o que permite o acesso a redes de dados de comutação de pacotes, como internet através do serviço GPRS. Dessa forma, o módulo GSM/GPRS envia os dados para um agente IoT (*Internet of Things*) na nuvem que é responsável por enviar os dados para o *Middleware*, o mesmo realiza o processamento e armazenamento dos dados no banco de dados MySQL.

A Figura 3 apresenta um apiário com 3 colmeias sendo monitoradas. Cada uma possui seus próprios sensores, onde a Placa Arduino envia os dados para a nuvem através do módulo GSM/GPRS. Após o tratamento dos dados, o usuário poderá criar novas notificações via sistema *web*, receber notificações via E-mail, Telegram ou SMS, e visualizar os dados através do aplicativo móvel.

No presente estudo, os dados não foram enviados ou processados de acordo com a arquitetura citada, pois os sensores referentes ao dióxido de carbono, oxigênio e peso ainda não estão em funcionamento. Nesse caso, os mesmos foram salvos em um banco de dados MySQL através de um *script* que simula o envio das coletas e os testes foram realizados localmente.

Figura 3 – Sistema Sm@rtbee



Fonte: Autora.

2.4 Tecnologias de Rede de Envio de Notificações

Um sistema de notificação “notifica os usuários quando ocorrem eventos importantes ou quando ocorre um evento que requer atenção especial” (SAAD *et al.*, 2017). As tecnologias de rede que realizam o monitoramento e envio de notificações são diversas. Porém, podem ser citados o Nagios e o Zabbix.

2.4.1 Nagios

O Nagios é uma ferramenta *open source* distribuída através de uma licença GPL, criada nos anos 90 por Ethan Galstad. Em relação a suas funcionalidades, o Nagios não permite que o servidor execute tarefas de pedidos e verificações, ele delega essa função a componentes externos (*plugins*) disponíveis, que ficarão encarregados de realizar esses pedidos e verificações. Os *plugins* coletam a informação e repassam-na para o servidor, que então realiza a análise dessa informação e a converte para a interface *web* com os vários estados dos objetos (ALMEIDA, 2015).

Com o Nagios é possível monitorar serviços como correio eletrônico, páginas *web* e outros, e dispositivos físicos como servidores, impressoras e sensores. Ele realiza o envio de notificações quando é detectado um problema e quando o mesmo é resolvido (ALMEIDA, 2015). As notificações podem ser enviadas por e-mail e SMS (AMNUR *et al.*, 2014). O autor Pucci (2017) realizou a integração do Nagios e Telegram criando um módulo *open source* para o sistema operacional CentOS 5.11. Nos testes realizados pelo autor o Nagios foi capaz de realizar o envio de mensagens para o Telegram.

2.4.2 Zabbix

O Zabbix é um *software* de monitorização distribuído e *Open Source*. Foi desenvolvido por Alexei Vlashev em 1998. Em 2001 ocorreu o lançamento da primeira versão não estável sobre uma licença GPL. É estruturado por três componentes essenciais, o servidor Zabbix, a base de dados e a interface *web* (ALMEIDA, 2015).

O servidor zabbix é responsável pela coleta e armazenamento de todos os dados monitorados. Possui ainda a tarefa de criação de gráficos e mapas, bem como o cálculo e o acionamento de *triggers* (são gatilhos que são disparados, respondendo a eventos em particular) e execução de ações a eles inerentes, como por exemplo o envio de notificações e execuções de scripts. A base de dados é o local onde serão armazenadas todas as configurações do Zabbix assim como toda a informação coletada. Todos os dados coletados são centralizados em uma única interface *web*, que permite a consulta/análise dos dados, assim como a criação de todas as configurações desejadas. Este permite ainda a criação de perfis de utilizadores que podem ter regras quanto aos acessos e quais ações este podem executar (ALMEIDA, 2015).

Além de possuir um painel de alertas, o Zabbix pode realizar o envio de mensagens de e-mail e até mesmo de SMS (AGUIAR, 2017). Segundo Thamrin *et al.* (2017) alguns recursos, como a API do Telegram e a API do SMS, podem ser adicionados ao Zabbix. Ao fazer isso, o Zabbix pode enviar uma mensagem de notificação através do Telegram ou SMS para o administrador da rede caso ocorra algum problema na rede.

2.5 Tecnologias de Rede Agregadas de Envio de Notificações

Apesar de existirem tecnologias de rede para o envio de notificações como o Zabbix e o Nagios, o presente estudo integrou as funcionalidades de envio de notificações ao sistema Sm@rtbee que está em desenvolvimento. Essa escolha se deu pela facilidade no uso para customização do sistema *web* Sm@rtbee, tanto no que diz respeito ao desenvolvimento de funções no sistema quanto ao uso. Além disso, o mesmo está voltado exclusivamente para o monitoramento de colmeias.

As tecnologias de rede agregadas para o envio de notificações são: E-mail, Telegram e SMS (*Short Message Service*), em português, Serviço de Mensagens Curtas, conhecido também como torpedo. A tabela Tabela 1 traz uma breve comparação entre as tecnologias escolhidas para o envio das notificações.

Tabela 1 – Tecnologias de Envio de Notificações

Tecnologia	Valor do Serviço
E-mail	Gratuito
SMS	9 centavos por mensagem
Telegram	Gratuito

Fonte: Autora (2018).

Nas seções seguintes, são apresentadas tais tecnologias. Como apresentado na Tabela 1 no envio de SMS há custo de envio da mensagem. Já as notificações via E-mail e Telegram não possuem custo. Todas as tecnologias necessitam de acesso à Internet para o envio de notificações no presente sistema.

2.5.1 E-mail

O e-mail ou correio eletrônico é um método que permite criar, enviar e receber mensagens através de sistemas eletrônicos de comunicação (GOMES, 2014). O serviço de e-mail tem sua origem na ARPANET e sua primeira transmissão inter-institucional foi realizada no início da década de 70 (MARCHI *et al.*, 2014).

Segundo Gomes (2014) o termo e-mail pode ser aplicado tanto para os sistemas que utilizam a Internet e que são baseados nos protocolos POP3 (*Post Office Protocol*), IMAP (*Internet Message Access Protocol*) e SMTP (*Simple Mail Transfer Protocol*), como aos sistemas “conhecidos como *intranets*, que permitem a troca de mensagens dentro de uma empresa ou organização e que são, normalmente, baseados em protocolos proprietários”.

A popularidade do e-mail diminuiu com o surgimento de diversos sistemas de comunicação alternativos, porém atualmente o e-mail continua a ser um dos métodos de comunicação em muitas empresas (NDECKY, 2017). A tecnologia de e-mail pode ser acessada e usada em qualquer lugar através de telefones celulares e outros dispositivos de computador (HAMIDI *et al.*, 2018) que tenham acesso a internet, portanto pode ser usado para o envio de notificações para os usuários. No presente estudo a classe PHPMailer versão 5.2.4 foi utilizada para o envio de e-mails. PHPMailer é uma classe de criação e transferência de e-mails com recursos completos para PHP, e é distribuída sob a licença LGPL 2.1 (BOINTON,).

2.5.2 SMS

O SMS é um serviço utilizado para o envio de mensagens de texto com um número limitado de caracteres por mensagem, é rápido e eficiente, e funciona através de telefones móveis ou pela Internet (GOMES, 2014).

O conceito de *Short Message Service* (SMS) foi desenvolvido pelo engenheiro finlandês Matti Makkonen, em meados da década de 1980 (LANZA, 2016), porém o primeiro SMS foi enviado a partir de um telemóvel apenas em dezembro de 1992, pela operadora Vodafone, no Reino Unido, com o texto “Feliz Natal” (GOMES, 2014). O Sistema Global de Comunicações Móveis (GSM) padronizou o SMS em 1985, porém o mesmo agora é mantido pelo 3GPP como TS 23.040 (ISLAM *et al.*, 2017). Esta tecnologia está sendo explorada no Brasil e no mundo desde 2001 com êxito por empresas privadas de vários setores, como (emissoras de televisão, bancos, seguradoras de móveis e automóveis) (LANZA, 2016).

No Brasil, não é necessário que o usuário possua créditos em seu telefone celular para receber mensagens via SMS, ou seja, todos os telefones celulares móveis do país com um chip SIM estão habilitados para receber mensagens via SMS de qualquer serviço (LANZA, 2016).

Em relação ao total de caracteres, uma mensagem SMS convencional pode ter 160 caracteres (codificados pelo conjunto de caracteres de 7 bits), 140 caracteres (codificados pelo conjunto de caracteres de 8 bits) e 70 caracteres (codificados por 16 bits para o conjunto de caracteres *Unicode*). Assim sendo, na prática, o tamanho do SMS é de 140 bytes. As mensagens maiores que 140 bytes são agrupadas em uma mensagem de 134 bytes com 6 bytes de cabeçalho de dados do usuário (UDH) para cada mensagem, que é chamado SMS concatenado (ISLAM *et al.*, 2017). O armazenamento de dados SMS é realizado na memória do próprio dispositivo móvel.

O SMS é enviado do remetente, ou seja, do aparelho móvel, para um centro de serviço de mensagens curtas (SMSC) e, posteriormente, entregue ao aparelho celular do destinatário usando um mecanismo chamado “armazenar e encaminhar”. Se o destinatário não estiver acessível, a mensagem é armazenada e enfileirada no SMSC e depois existe uma nova tentativa de envio para o destinatário. O *Mobile Application Part* (MAP) do protocolo SS7 é usado para enviar e receber mensagens curtas entre o SMSC e o telefone do usuário. O SMS do celular é chamado SMS-MO e o SMS do SMSC para o celular é chamado SMS-MT (ISLAM *et al.*, 2017).

2.5.2.1 *SMS Gateway*

O envio de mensagens pode ser realizado por meio de um *gateway*. O *gateway* pode ser interpretado como uma ponte conectando um sistema com outro sistema diferente, para que possa ocorrer uma troca de dados entre os sistemas. Dessa forma, o *SMS Gateway* pode ser entendido como uma ligação para o tráfego de dados SMS, seja ele enviado ou recebido (SIREGAR; TAUFIK, 2017).

Os recursos normalmente desenvolvidos em aplicativos do *SMS Gateway* são: Resposta automática, mensagem de entrega / transmissão em massa e entrega agendada. Tais recursos podem ser modificados conforme a necessidade. As mensagens são enviadas para o celular do solicitante, o mesmo pode escolher qual dado deseja que seja enviado (SIREGAR; TAUFIK, 2017). De acordo com essa tecnologia, no presente estudo, para o envio de mensagens via SMS a API da TotalVoice foi integrada ao sistema. A API da TotalVoice é um serviço com um custo sobre cada mensagem enviada. A mesma foi escolhida pela facilidade de integração, pelo custo fixo no valor da mensagem, e por possuir um valor de pacote mínimo reduzido (apenas R\$ 20,00 reais) em relação aos demais serviços. (TOTALVOICE,).

2.5.3 *Mensagens Instantâneas*

Atualmente os aplicativos de mensagens instantâneas do inglês *Instant messaging* ou ainda IM foram desenvolvidas e usadas por quase todos os usuários de *smartphones* (CANDRA *et al.*, 2016). A IM é um tipo de mídia que possibilita a transmissão de texto e a comunicação online através de transmissão de sincronização de voz e vídeo entre pessoas ou grupos de pessoas em uma determinada plataforma (WU *et al.*, 2017).

Um dos sistemas operacionais móveis mais populares atualmente é o Android. De acordo com loja de aplicativos Android ou no Google Play, centenas de aplicativos de mensagens instantâneas estão disponíveis para download (CANDRA *et al.*, 2016). Dentre essas aplicações podem ser citadas o WhatsApp, Skype, Facebook Messenger e Telegram. O Telegram é um aplicativo considerado seguro do Android e usado por muitas pessoas (CANDRA *et al.*, 2016).

2.5.3.1 *Telegram*

O Telegram é um *software* popular entre os usuários de *smartphones*, possui uma API pública (MORAIS *et al.*, 2017), e tem como função básica o envio e recebimento de mensagens

instantâneas. De origem russa, o *software* foi criado por Pavel e Nikolai Durov, e atende a requisitos como: segurança, confiabilidade e velocidade em qualquer rede. A versão para iOS foi lançada em 14 de agosto de 2013, já a versão para Android foi lançada oficialmente em 20 de outubro de 2013 (ARAÚJO *et al.*, 2017).

Seu uso pode ser realizado através de várias plataformas como: *smartphone*, *tablet* e computador. O aplicativo possui capacidade de enviar e receber mensagens, fotos, vídeos, áudios e qualquer arquivo com até 1,5 GB (gigabyte) (ARAÚJO *et al.*, 2017). Além de poder enviar e receber mensagens individuais, segundo Araújo *et al.* (2017) o Telegram pode criar grupos ou canais com até 5000 mil pessoas. Busca de pessoas pelo nome de usuário, e ainda criação de bots (MORAIS *et al.*, 2017).

O Telegram realiza o armazenamento de dados na nuvem, e portanto, não necessita de realização de backups. Visto isso, o Telegram realiza a junção de SMS e e-mail ao mesmo tempo em um só aplicativo, e ainda possui como um de seus benefícios em relação ao SMS não cobrar pelo envio de mensagens (ARAÚJO *et al.*, 2017).

2.5.3.1.1 Bots do Telegram

A API pública do Telegram é apropriada para o desenvolvimento de bots. Os bots são aplicativos de terceiros que funcionam dentro do Telegram (MUFTINISA *et al.*, 2017) e usuários podem interagir com bots enviando e recebendo mensagens (MORAIS *et al.*, 2017). A Bot API do Telegram possui vários recursos que vão desde bots que interagem em grupos, com múltiplos usuários ao mesmo tempo, bots de pesquisa ou *inline bots*, que pesquisam mídias na internet, por exemplo a busca de imagens, vídeos ou gifs ao serem solicitados pelo usuário (MORAIS *et al.*, 2017) e sistemas que usam bots para o envio de alertas, como no presente estudo.

No estudo de Morais *et al.* (2017), o autor usou a Bot API do Telegram para a criação de um Quiz chamado Donuts, no modelo cliente-servidor, ou seja, o bot (servidor) aplica questões e o usuário (cliente) responde. As perguntas respondidas corretamente geram pontos e assim o usuário pode avançar de nível.

A criação do bot do presente estudo se deu utilizando o *BotFather*. Ao adicionar o *BotFather* e enviar o comando `/start` esse bot apresenta uma lista de opções, para criação e edição de bots. Ao criar um bot ele libera um *token* para o acesso do telegram via HTTP.

3 TRABALHOS RELACIONADOS

Atualmente os alertas/notificações são usados em diferentes áreas. As subseções seguintes trazem os trabalhos relacionados do presente estudo.

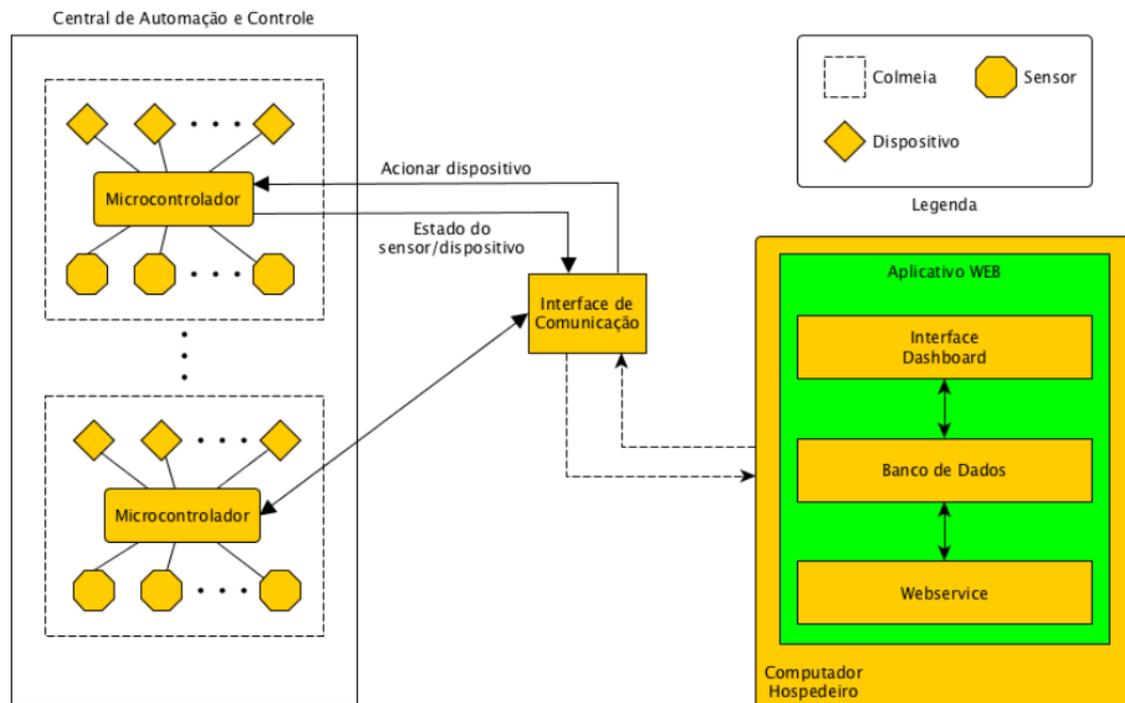
3.1 *Beehiveior* - Sistema de monitoramento e controle de colmeias de produção apícola

Com o objetivo de auxiliar as pesquisas da área apícola da UEPA-EAJ (Unidade de Extração de Produtos Apícolas, da Escola Agrícola de Jundiá) e facilitar as atividades no manejo das colmeias o trabalho de Dutra (2016) desenvolveu uma ferramenta de monitoramento e controle de colmeias de um apiário chamada *Beehiveior*. Esse sistema realiza coleta de variáveis das colmeias e exibe-os em um sistema *web*. O sistema foi construído utilizando a placa Arduino modelo UNO acoplada a um *shield* para o fornecimento de conexão *Ethernet* e armazenamento via cartão SD, a linguagem de programação Java e o SGBD PostgreSQL.

O *Beehiveior* é composto da Central de Automação e Controle e Aplicativo *web*. A primeira parte é a responsável pela coleta dos dados através dos sensores acoplados na colmeia e envio dos dados para a interface de comunicação. Além disso, é responsável por receber comandos da interface de comunicação e realizar ações para alterar o estado dos dispositivos acoplados. Já o aplicativo *web* é responsável por realizar o processamento, armazenamento e exibição dos dados enviados pela central de automação e controle. O mesmo ainda é responsável por executar ações programadas ou *online* sobre os dispositivos acoplados a colmeia, ele realiza também a verificação dos dados e geração dos alertas. No que se refere a emissão de alertas, apesar de ser um dos objetivos do estudo, o mesmo não concluiu o módulo de alertas (DUTRA, 2016). A Figura 4 mostra como é a interação entre esses módulos.

As variáveis coletadas foram: temperatura, umidade, ruído, sensor de presença e contagem de abelhas (infravermelho). Tais dados são mostrados através do aplicativo *web* por três tipos de visualização: por apiário e colmeia, em forma de tabela e em forma de gráfico. Foram realizados testes controlados em uma colmeia. Os dados foram coletados pelos sensores com um intervalo de 1 minuto. Foram comparadas as duas interfaces de comunicação *Ethernet* cabeada e cartão SD (DUTRA, 2016). O presente estudo não realizou a criação de uma central de automação e controle, pois a mesma já existe, o módulo de sensoriamento do sistema Sm@rtbee. Porém, foi realizada a integração de formulários de criação de notificações ao sistema Sm@rtbee. As variáveis observadas são: temperatura, umidade, som, dióxido de carbono, oxigênio e peso.

Figura 4 – Diagrama de Interação dos Módulos do *Beehiveior*



Fonte: Dutra (2016).

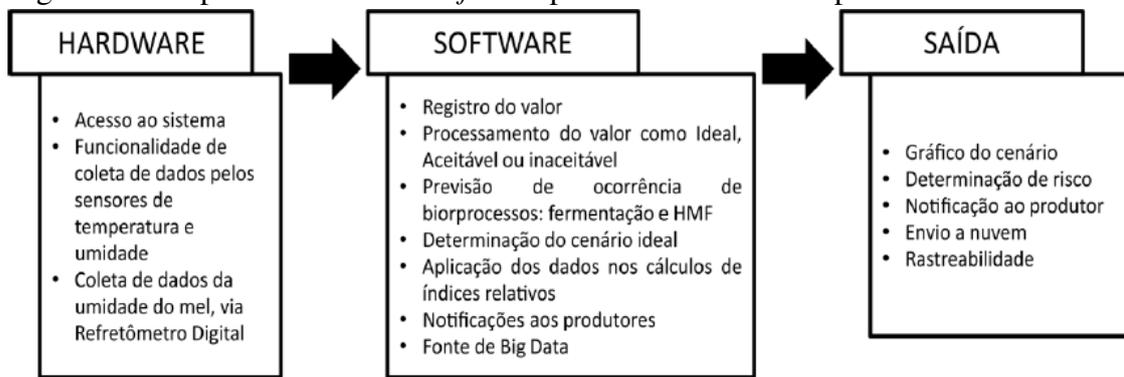
3.2 Apropriação intelectual da tecnologia (*software e hardware*) do controle de qualidade da cadeia industrial de mel

O estudo de Linhares e Quintella (2018) apresenta um conjunto de tecnologias denominado BIPP (Bloco Inteligente do Processo de Produção), o mesmo é formado por um *hardware* e quatro *softwares* embarcados. Ele funciona "relacionando os dados de análises de cor, umidade do mel, umidade relativa do ar, temperatura ambiente e de análises laboratoriais de componentes físico-químico-microbiológicos" para garantir a qualidade na produção do mel e monitorar os fatores de risco como fermentação e o aumento de HMF (Hidroximetilfurfural) (LINHARES; QUINTELLA, 2018).

Para a análise da cor do mel e o HMF, são utilizados *softwares* embarcados no *hardware*, que tem como funcionalidades: instrumentalizar, documentar e parametrizar os dados. No que se refere as análises físico-químico-microbiológicas, os autores desenvolveram um sistema, que após a coleta dos dados, realiza o tratamento estatístico buscando relações e interferências dos resultados das análises com as condições críticas do mel. Além disso, o *software* embarcado no *hardware*, tem como a principal funcionalidade de avaliação e monitoramento dos resultados colhidos nos laboratórios e qualificar o mel em função dos parâmetros exigidos (LINHARES; QUINTELLA, 2018).

Já para as medições de umidade e temperatura como mostrado na Figura 5, os dados são coletados e documentados automaticamente por meio sensores instalados no *hardware*. Esses sensores utilizam a tecnologia Arduino e processam em tempo real os dados relacionados a umidade relativa do ar e de temperatura, alimentando o sistema embarcado, responsável por todo o processamento. A coleta dos dados relativos a umidade do mel é realizada por meio de um refratômetro digital, que informa o dado que será inserido no *software*, com o objetivo de gerar padrões, previsões e aplicação dos cálculos de relação com a umidade relativa do ar e com a temperatura ambiente (LINHARES; QUINTELLA, 2018).

Figura 5 – Etapas funcionais do *software* para os índices de temperatura e umidade



Fonte: Linhares e Quintella (2018).

Apesar de mostrar as etapas funcionais do software Linhares e Quintella (2018) incluindo as notificações aos produtores como saída, o estudo não descreveu qual a tecnologia de envio da notificação.

A seguir serão apresentados trabalhos relacionados ao controle ambiental, vendas e saúde.

3.3 *Developing an underwater monitoring and alert system for water-borne disease management*

No artigo de Varma *et al.* (2017) são apresentadas propostas de melhorias ao sistema *Underwater acoustic sensor network* (UWASN) que é uma rede de sensores acústicos subaquáticos que pode coletar dados do oceano, como “temperatura, salinidade, pH, turbidez e oxigênio dissolvido”, entre outros, com a ajuda de nós sensores subaquáticos. Através do *gateway* de superfície subaquática, os nós de sensores subaquáticos encaminham os dados para análise na rede terrestre. Esse mecanismo pode ser usado para detectar quais áreas estão mais propensas a ter doenças que podem ser transmitidas pela água. A melhorias propostas estão relacionadas

ao pré-processamento de dados e adição da funcionalidade de envio de alertas para o centro de saúde local. Tais melhorias são baseadas em regras para o *gateway* de superfície subaquática.

O *gateway* de superfície subaquática possui os seguintes componentes: módulo de comunicação, microprocessador, camada de abstração de hardware e unidade de fonte de energia. O módulo de comunicação é formado pelo módulo RF e módulo acústico. O módulo de RF é usado para comunicação entre o centro de saúde local e o *gateway*. Já o módulo acústico é usado para comunicação entre os nós do sensor subaquático e o *gateway*. Os autores adicionaram o módulo de pré-processamento ao *gateway* de superfície subaquática para realizar o gerenciamento dos dados. O módulo de pré-processamento consiste em mecanismos baseados em regras e o sistema de alerta/mecanismo de notificação. O módulo de pré-processamento do *gateway* de superfície encontra os dados necessários e os compara com os valores mínimos e máximos padrões. Tais valores são baseados nas condições de crescimento favoráveis das doenças causadas pela água. Se o valor atual dos dados for superior ao valor máximo, o *gateway* de superfície subaquática envia mensagem de alerta para o centro de saúde local para tomar medidas cautelares para controlar as doenças transmitidas pela água. Se os valores de dados forem menores que os valores mínimos, os dados serão enviados ao servidor de banco de dados para a análise dos dados. O servidor de banco de dados pode processar esses dados e enviar a informação dos dados processados ao centro de saúde do governo (VARMA *et al.*, 2017).

Os autores Varma *et al.* (2017) realizaram emulações, e usaram uma função aleatória que gerava valores de salinidade, pH e temperatura. Foram considerados os valores aleatórios cinco vezes e esses foram considerados como cinco locais diferentes. No estudo de Varma *et al.* (2017) não houve descrição de qual tecnologia de envio de notificação foi escolhida. O sistema de notificações do presente estudo também compara os valores coletados das colmeias e os compara com os valores definidos como máximos e mínimos de cada usuário, caso o usuário opte pela comparação. Já para o envio de coletas, foi utilizado um *script* com valores pré-definidos das variáveis, e foi considerado um número de 8 colmeias para a realização das comparações.

3.4 Alerta poá - um sistema de comunicação apoiado em redes de sensores sem fio para monitoração do córrego itaim

A fim de monitorar o nível de água de córregos e alertar a população a cerca de possíveis enchentes. A dissertação de Cardoso *et al.* (2016) desenvolveu o projeto Alerta Poá, que se trata de um conjunto de *hardware* e *software* com o objetivo de realizar o monitoramento de

córregos e rios e a divulgar os dados coletados. O estudo criou um novo modelo de monitoramento utilizando a plataforma Rádium, que realiza a interface de um ou vários sensores com a Internet. O sistema foi desenvolvido e instalado na cidade de Poá de forma experimental, e permitiu testar a eficiência e a fidelidade dos dados coletados pelos sensores a um custo extremamente baixo se comparado a aplicações comerciais. Os dados coletados são transmitidos em tempo real para o Centro de Controle de Segurança da Cidade, o mesmo realiza o tratamento dos dados, convertendo os dados brutos colhidos pelos sensores em informações úteis e de fácil compreensão sobre o nível de água do córrego através de uma página *web* da Prefeitura.

No que diz respeito a alertas, quando é observado um sensor com situação de alerta são enviados e-mails para os agentes interessados nos dados. Além disso, são enviadas mensagens para os usuários que utilizam a aplicação de monitoramento móvel “Alerta Poá”. No aplicativo ainda é mostrado a visualização atual da porcentagem de água no córrego (CARDOSO *et al.*, 2016).

Para a coleta dos dados dos sensores foi utilizado um código em *Python*. Já o Sistema de Gerenciamento de Banco de Dados utilizado foi o MySQL. Para criação do Aplicativo Alerta Poá, foi utilizada a plataforma Xamarin, que é um software para desenvolvimento de aplicativos móveis para os sistemas operacionais IOS, Android e Windows. A página *web* foi construída através da linguagem PHP para consulta de dados no banco de dados, e a linguagem JavaScript para a ilustração desses através de gráficos interativos. Também foi utilizada a linguagem CSS para a formatação da página *web* (CARDOSO *et al.*, 2016).

3.5 *Interfacing atmospheric variables to web interface using arduino*

A fim de monitorar os gases dióxido de carbono (CO₂) e metano (CH₄) em áreas residenciais, o estudo de Mishra e Kaavya (2017) propôs um dispositivo denominado WSN (*Wireless Sensor Network*) que engloba o sensor de gás CO₂ MG-811, o sensor de gás CH₄ TGS 2600, o sensor de temperatura e umidade DTH-11, o *shield* que possui um chip SIM900A GSM e uma antena de banda dupla, todos conectados com um dispositivo Arduino UNO R3. Nos experimentos realizados o dispositivo foi capaz de medir a quantidade dos gases dióxido de carbono (CO₂) e metano (CH₄) continuamente e alimentar os dados para uma interface *web*. A interface da *web* usou o Serviço de Mapa da *web Bhuvan* do Centro Nacional de Sensores Remotos como Interface do Usuário. A interface *web* mostra as tendências de mudança na concentração dos gases.

Em relação ao envio dos dados, o dispositivo realiza o envio a cada hora para o servidor da *web* usando o HTTP (*Hyper Text Transfer Protocol*) através do Modem GPRS (*General Packet Radio Service*). O dispositivo é dividido em três partes lógicas: (i) parte de detecção - os sensores coletam os dados e envia-os para a parte de processamento conforme pedido, os sensores se comunicam através da interface serial; (ii) parte de processamento - todas as medidas da parte de detecção são acumuladas e filtradas, essas medidas são então empacotadas em um único arquivo JSON (*JavaScript Object Notation*) que é enviado para o módulo de transmissão; (iii) parte transmissora - consiste em todos os componentes que são utilizados para se comunicar com outros dispositivos, como um servidor *web*. Ele deve poder se comunicar com o nó de extremidade usando a tecnologia suportada por ele (MISHRA; KAAVYA, 2017). Apesar de citar que a interface *web* pode enviar alertas imediatos às autoridades para tomar as medidas necessárias para a saúde e o melhoramento das pessoas que vivem nessas áreas, o artigo não citou como isso foi configurado, ou seja, não descreveu o tipo de alerta e não o apresentou como um resultado.

3.6 *Web based online bakery system with short messaging service and email notification*

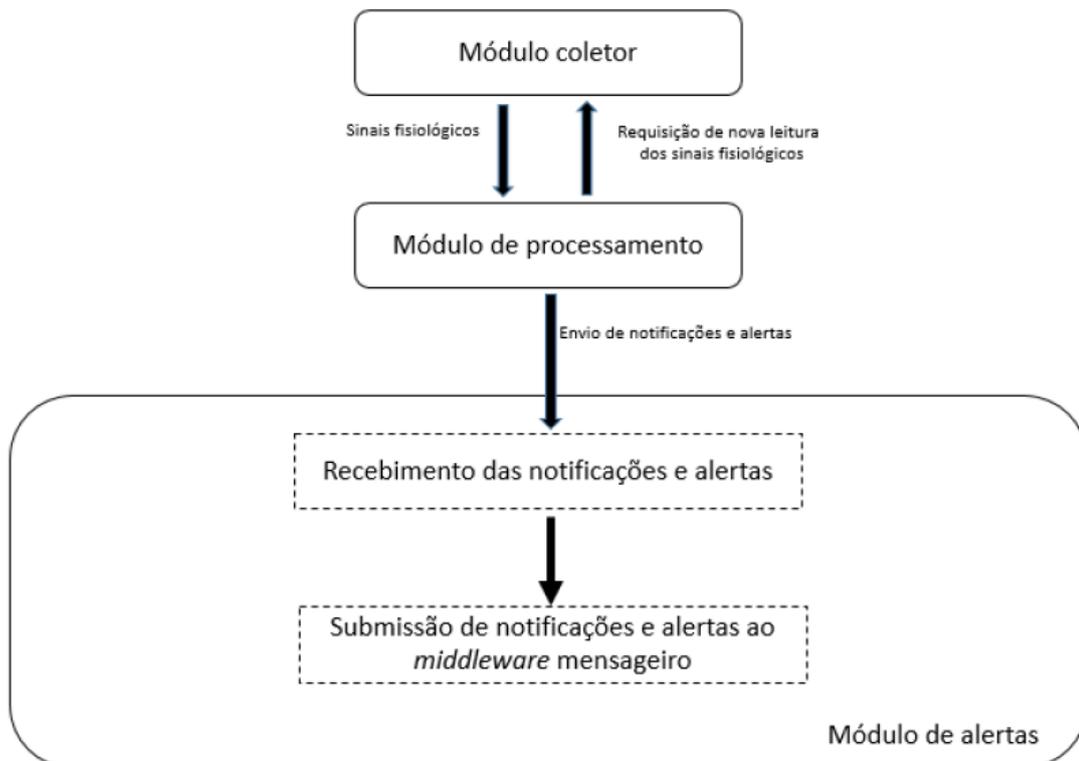
(RAZAK *et al.*, 2017) desenvolveram o *Cakes By Fara* (CBS) que é um sistema de pedidos *online* de padaria onde os principais produtos a serem vendidos são bolos de vários tipos. O sistema é baseado na *web* e faz notificações através do *Short Message Service* (SMS) e por e-mail. As notificações são enviadas automaticamente para o e-mail do cliente após o seu cadastro, compra de produtos e na alteração de status do pedido. Já o envio de SMS é realizado pelo administrador quando o pedido está pronto. O cliente receberá a notificação de SMS de entrega ou notificação de retirada com base na opção de entrega escolhida, ou seja, para notificar o cliente sobre o status da entrega. Em relação aos usuários do sistema existem três tipos, que são: clientes registrados, clientes não registrados e administrador. Os clientes registrados possuem mais privilégios no sistema em comparação com clientes não registrados, pois conseguem comprar os produtos online e receberem notificações por e-mail e SMS. A metodologia *Rational Unified Process* (RUP) foi usada para desenvolver o sistema. O sistema de (RAZAK *et al.*, 2017), bem como o presente sistema, foram construídos usando a linguagem de programação PHP, CSS e JavaScript. No presente estudo, as tecnologias utilizadas para o envio de notificações foram SMS, Telegram e e-mail, as mesmas são enviadas de forma automática de acordo com o cadastro de notificações realizado pelos usuários.

3.7 Um modelo proativo de antecipação de ações de times de resposta rápida baseado em análise preditiva

O estudo de Dias (2017) propôs um modelo chamado Predictvs, que consiste em um Time de Resposta Rápida (equipe de saúde) e um conjunto de *softwares* (sensores, sistema *web* e sistema de alerta) para a prevenção de mortes de pacientes que possam ter piora clínica fora de ambientes de Unidades de Tratamento Intensivo (UTIs) em hospitais. O mesmo realiza a análise de sinais vitais em tempo real dos clientes com o uso de escores de alerta precoce e regressão linear.

No que diz respeito ao módulo de alertas Figura 6, o *front-end* do mesmo foi desenvolvido em JavaScript, e recebe os alertas do módulo de processamento. Utilizando as bibliotecas *tg* e *PyTg* do Python foi realizada a integração do módulo de alertas com o aplicativo Telegram. Assim, os alertas são enviados diretamente para a conta do profissional de saúde envolvido de forma individual ou coletiva (grupos) (DIAS, 2017).

Figura 6 – Módulo de Alertas



Fonte: Dias (2017).

A Tabela 2 traz uma comparação entre o presente estudo com os trabalhos relacionados a apicultura, ao controle ambiental e os demais trabalhos.

Tabela 2 – Trabalhos Relacionados

Estudo	Tipo de Notificação	Fenômeno Observado
Dutra (2016)	N/C ¹	Monitoramento de Colmeia
Linhares e Quintella (2018)	N/D ²	Monitoramento da Qualidade do Mel
Varma <i>et al.</i> (2017)	N/D ²	Saúde da água
Cardoso <i>et al.</i> (2016)	E-mail e alertas pelo aplicativo	Monitoramento de córregos
Mishra e Kaavya (2017)	N/D ²	Monitoramento de Gases
Razak <i>et al.</i> (2017)	E-mail e SMS	Vendas de Bolos
Dias (2017)	Telegram	Sinais Vitais
Presente Estudo	E-mail, SMS e Telegram	Monitoramento de Colmeia

Fonte: Autora (2018).

¹ Não concluiu o módulo de envio de notificações/alertas.

² Não descreveu a tecnologia de envio de notificações/alertas.

4 METODOLOGIA

O desenvolvimento do sistema foi realizado seguindo-se seguintes etapas:

- a) Análise dos requisitos;
- b) Desenvolvimento do sistema Notifyme;
- c) Testes e validação do sistema.

Os detalhes sobre cada uma das etapas são descritos ao longo desta seção.

4.1 Análise dos Requisitos

Nessa fase foram levantados os requisitos do sistema. Os requisitos foram definidos a partir de conversas realizadas com um dos professores responsáveis pelo projeto *Sm@rtbee* e com os outros desenvolvedores do sistema. As conversas foram realizadas a partir da troca de e-mails, ligações realizadas por meio do *WhatsApp*, *Hangouts* e grupo de desenvolvimento no *Telegram*. Nessas conversas foram feitas perguntas sobre o funcionamento do sistema *web*, sobre o envio de notificações, quais tecnologias eram mais viáveis, além de avaliação de incrementos para acréscimos de requisitos. A partir disso foram levantados os requisitos que poderão ser vistos na seção de apresentação do sistema.

4.2 Desenvolvimento do Sistema Notifyme

Nesta fase, foi desenvolvido o protótipo do sistema de notificação para o envio de notificações/alertas para os usuários solicitantes. O sistema foi desenvolvido já integrado ao projeto do sistema *web Sm@rtbee*. O sistema *web Sm@rtbee* utiliza o *framework Laravel* versão 5.6.23. Sendo assim, para o desenvolvimento do *front-end* foi utilizado o *framework Laravel* na mesma versão. Esse *framework* foi criado para ajudar os desenvolvedores na criação de sistemas *web* com a linguagem PHP, e possui como características a simplicidade e flexibilidade do design. Esse *framework* aplica o método *Model-View-Controller* (MVC), que separa os dados (*Model*) da *View* e a lógica da aplicação (*Controller*) (ANIF *et al.*, 2017).

Para realizar *downloads* e atualizações de dependências do *Laravel*, foi utilizado o *Composer*. O *Composer* trata-se de um gerenciador de dependências em projetos, para programação em PHP, ou seja, permite gerenciar (declarar, baixar e manter atualizado) os pacotes de software nos quais o projeto PHP está baseado (ESPINOZA *et al.*, 2017).

Juntamente com o *Laravel*, para o sistema de notificações foi utilizada também a

linguagem JavaScript. JavaScript é muito utilizada em aplicações *web*. Os navegadores da *web* permitem que o códigos em JavaScript interajam com os usuários, reaja a suas ações (YOUNANG; LU, 2015) como a ocultando ou mostrando opções de um menu.

Já para o *back-end*, ou seja, para os *scripts* de conexão com o banco de dados para o envio de notificação e o envio de notificações em si, foi utilizado a linguagem PHP versão 7.0. A linguagem de *script* PHP (*Hypertext Preprocessor*) é *open source*, ou seja, de código aberto de uso geral, muito utilizada e adequada para o desenvolvimento *web* (PHP, 2017). A base de dados utilizada foi o MySQL. Segundo Hardono *et al.* (2017) o MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) do tipo relacional que permite aos usuários armazenar, pesquisar, classificar e recuperar dados de maneira eficiente. O MySQL é conhecido por ser muito rápido e robusto.

Para o gerenciamento de código fonte e controle de versão do sistema, foi utilizado o Git. Ele é um sistema de controle de versão de código, distribuído, gratuito e de código aberto (GIT,). Para o gerenciamento de repositório Git foi utilizado o GitLab, pois o projeto Sm@rtbee já estava sendo controlado através dele. De acordo com MORAES (2013) o GitLab é *open source* e suas principais funcionalidades são a criação de projetos, repositórios e gerenciamento de usuários, bem como suas permissões de acesso ao código fonte.

Para a execução do sistema, foi utilizado o Servidor Apache. Ele é um servidor *web* que responde a solicitações de navegadores da *web*. O Apache é um dos servidores *web* mais populares do mundo, possui alta qualidade e é de código aberto (ESPINOZA *et al.*, 2017).

A Tabela 3 traz um resumo de ferramentas e linguagens utilizadas para a criação do sistema.

Tabela 3 – Ferramentas e linguagens utilizadas

Sistema	Framework	Linguagens	SGBD	Controle de Versão	Servidor <i>web</i>
Notifyme	Laravel	PHP e JavaScript	MySQL	Git e GitLab	Apache

Fonte: Autora (2018).

4.3 Testes e validação do sistema

Ao longo do desenvolvimento, foram realizados testes unitários com o envio de notificações separadamente até não surgirem erros. Após isso, para a validação do sistema, foi criado um *script* que simula o envio de coletas (Apêndice B), descrito no Apêndice A.

5 SISTEMA DE NOTIFICAÇÕES NOTIFYME

O sistema de notificações é baseado nas funcionalidades de envios de notificações/alertas para usuários que possuam colmeias cadastradas e que necessitam de saber como está a saúde da colmeia, através do envio de notificações com o valor das variáveis coletadas. O sistema foi desenvolvido com o propósito de disponibilizar ao usuário formas de envios de notificações diferentes.

Este capítulo contém os requisitos do sistema, suas telas e funcionalidades.

5.1 Requisitos

Esta seção tem o objetivo de apresentar os requisitos levantados para o desenvolvimento do sistema. Foram definidos os seguintes requisitos para o sistema:

- Requisitos funcionais
 - Cadastro de Notificação: Somente o usuário logado poderá cadastrar novas notificações para o mesmo. Onde cada colmeia poderá ter apenas um usuário e as notificações cadastradas serão únicas, ou seja, uma colmeia poderá ter apenas uma configuração de envio para cada tecnologia de envio (exceto se o usuário for o administrador);
 - Escolha de Preferências de Envio: Ao realizar o cadastro de uma nova notificação, o usuário poderá escolher as condições de envio de notificações, caso ocorram mudanças nos valores de limiar da colmeia ou não. O mesmo também poderá escolher se deseja que o envio seja realizado de todas as variáveis ou das escolhidas pelo mesmo. Além disso, o usuário terá um intervalo de envio de notificação a ser escolhido, os mesmos são: 30 minutos, 1 hora, 2 horas, 4 horas, 8 horas, 16 horas e 24 horas. Nesse cadastro, o usuário poderá escolher se deseja enviar alertas caso ocorram atrasos nas coletas ou não, e o período de checagem poderá ser escolhido onde os valores serão os mesmos do intervalo das notificações, nesse caso, se houver atraso ele receberá uma notificação de acordo com o intervalo de atraso;
 - Listagem de Notificação: O usuário logado poderá visualizar suas notificações cadastradas;
 - Edição de Notificação: O usuário logado poderá editar as notificações cadastradas. Na edição, o mesmo poderá desativar a notificação;
 - Exclusão da Notificação: O usuário logado poderá apagar as notificações cadastradas;

- Configuração dos Valores de Limiares: O usuário ao cadastrar-se possuirá configurações padrões de grandezas. O mesmo poderá editá-las de acordo com suas preferências e limites de valores;
 - Paralisação no Envio de Notificações: Caso ocorram atrasos ou ausência nas coletas, só serão enviadas dez notificações consecutivas, ou seja, só serão enviadas notificações caso exista uma nova coleta;
 - Acesso Privilegiado ao Administrador: O administrador do sistema terá acesso a listagem das notificações cadastradas de todos os usuários e poderá realizar o cadastro de notificações de colmeias de outros usuários para ele mesmo. Ele não poderá editar ou apagar uma notificação que não pertence a ele.
- Requisitos não funcionais
 - O sistema requer um sistema operacional Ubuntu com versão igual ou superior a 16.04;
 - O SGBD utilizado deve ser o MySQL;
 - Para as atualizações do sistema, o servidor deverá ter o Git e *Composer* instalados;
 - Para o funcionamento, o servidor deverá ter o Apache, o PHP 7.0 e o cURL instalados.
 - Restrições
 - O sistema requer conexão à internet para ser acessado pelo usuário e para o envio das notificações, tanto de E-mail, quanto Telegram e SMS.

5.1.1 Apresentação do Sistema

Esta seção tem como objetivo apresentar as telas do sistema referentes às notificações e configurações e explicar o funcionamento de cada uma, bem como apresentar o formato dos *scripts* de envio das notificações.

5.1.1.1 Configuração de Grandezas

Por padrão, após o cadastro de um novo usuário no sistema uma *trigger* é acionada no banco de dados e é criado um novo registro de configurações com valores padrões do sistema relacionados ao usuário. A Figura 7 apresenta a listagem de configurações de uma usuária.

Na tela de listagem existe a opção de edição das configurações. No formulário de edição, como pode ser visto na Figura 8, as variáveis utilizadas são: temperatura, umidade, som, dióxido de carbono, oxigênio e peso. O usuário deve escolher os valores mínimos e máximos

Figura 7 – Listagem das Configurações Cadastradas das Variáveis Coletadas



GRANDEZAS	VALORES MÍNIMOS	VALORES MÁXIMOS
Temperatura	40 °C	50 °C
Umidade	30 %	35 %
Som	30 dB	60 dB
Oxigênio	36 %	50 %
Dióxido de Carbono	350 ppm	500 ppm
Peso	40 Kg	80 Kg

Fonte: Autora (2018).

para cada variável de acordo com suas preferências e abelhas estudadas.

Figura 8 – Formulário de Edição das Configurações



Passo 1: Configure a Faixa de Temperatura
 Temperatura Mínima em °C: 40 | Temperatura Máxima °C: 50

Passo 2: Configure a Faixa de Umidade
 Umidade Mínima em %: 30 | Umidade Máxima em %: 35

Passo 3: Configure a Faixa de Som
 Som Mínimo em dB: 30 | Som Máximo em dB: 60

Passo 4: Configure a Faixa de Oxigênio
 Oxigênio Mínimo em %: 36 | Oxigênio Máximo em %: 50

Passo 5: Configure a Faixa de Dióxido de Carbono
 Dióxido de Carbono Mínimo em ppm: 350 | Dióxido de Carbono Máximo em ppm: 500

Passo 6: Configure a Faixa de Peso
 Peso Mínimo em KG: 40 | Peso Máximo em KG: 80

Fonte: Autora (2018).

5.1.1.2 Cadastros de Novas Notificações

A Figura 9 apresenta a tela de cadastro de uma nova notificação via E-mail. Como pode ser visto, o formulário é bastante informativo. Inicialmente o usuário deve escolher de qual colmeia ele deseja que sejam enviados os dados. No passo 2, deve-se nomear a notificação. O passo 3, refere-se ao critério de verificação e possui duas opções, a primeira se refere ao envio das notificações sem comparações com as configurações cadastradas de valores mínimos e máximos das variáveis coletadas, já a segunda opção mostrada na Figura 11 realiza as comparações com as configurações cadastradas.

Figura 9 – Formulário de Cadastro de Nova Notificação Via E-mail

The screenshot shows a web interface for configuring email notifications. The title is 'NOVA NOTIFICAÇÃO DE E-MAIL DA COLMEIA'. The breadcrumb trail is 'DASHBOARD / NOVA NOTIFICAÇÃO DE E-MAIL DA COLMEIA / LISTA DE NOTIFICAÇÕES DE E-MAIL'. The main heading is 'Configurar Notificações de E-mail'. The steps are as follows:

- Passo 1: Escolha a Colmeia**: A table with columns 'ESCOLHA A COLMEIA' and 'NOME'. It lists 'Colmeia 2' and 'Colmeia 3'. A red dot is positioned under 'Colmeia 3'.
- Passo 2: Nome da Notificação**: A text input field.
- Passo 3: Escolha o Critério de Verificação**: A dropdown menu with the selected option 'Sem mudanças nos padrões (mínimo e máximo)'. A link 'Listar Configurações' is below it.
- Passo 4: Verificar Atraso na Coleta**: Radio buttons for 'Sim' (selected) and 'Não'. A dropdown menu for 'Escolha o intervalo de checagem' is set to '30 minutos'.
- Passo 5: Configure o Envio da Notificação**: Radio buttons for 'Ativar todas as notificações de grandezas' (selected) and 'Personalizar as notificações de grandezas'. A dropdown menu for 'Escolha o intervalo' is set to '30 minutos'.

At the bottom, there is a footer with the address: 'Endereço: Rua Campus do Pici - Campus do Pici - Bloco 942 - A Fortaleza | Ceará | Brasil CEP: 60440-554'.

Fonte: Autora (2018).

No passo 4 o usuário poderá escolher entre verificar se existe atraso nas coletas ou não. O intervalo de checagem poderá ser escolhido entre 30 minutos, 1, 2, 4, 8, 16 ou 24 horas. Caso o usuário opte por checar se existe atraso ele receberá uma notificação informando o atraso de acordo com o intervalo de checagem, as notificações de atraso possuem um limite de 10 mensagens consecutivas, atingindo as 10 mensagens o usuário só receberá novas notificações na existência de uma nova coleta.

Já no passo 5, o usuário poderá escolher entre ativar o envio das notificações referentes a todas as variáveis coletadas podendo escolher o intervalo de envio ou personalizar o envio, podendo escolher sobre quais variáveis serão enviadas as notificações e o intervalo de envio personalizado de cada variável. Caso o usuário escolha personalizar a Figura 10 mostra o menu com as variáveis e os intervalos. O intervalo de envio escolhido pode ser 30 minutos, 1, 2, 4, 8, 16 ou 24 horas, tanto para as notificações ativadas quanto para as notificações personalizadas.

Figura 10 – Formulário de Personalização das Grandezas

Personalizar as Notificações de Grandezas

Notificações de Temperatura

Sim Não

30 minut

30 minutos

1 hora

2 horas

4 horas

8 horas

16 horas

24 horas

Notificações de Umidade

Sim Não

Notificações de Som

Sim Não

Notificações de Oxigênio

Sim Não

Notificações de Dióxido de Carbono

Sim Não

Notificações de Peso

Sim Não

Fonte: Autora (2018).

Os formulários de cadastro de nova notificação via E-mail e SMS são semelhantes alterando-se apenas os títulos do formulário. Já o formulário de cadastro de nova notificação via Telegram possui algumas diferenças em relação aos demais. Como no cadastro de usuário do sistema Sm@rtbee não existia um campo para o Telegram, apenas para E-mail e Número de celular, o mesmo foi criado no presente formulário.

Assim como no cadastro de novas notificações via E-mail e SMS, no cadastro de uma nova notificação via Telegram o passo 1 diz respeito a escolha da colmeia, o passo 2 ao nome dado a notificação. A Figura 11 mostra o formulário de cadastro de uma nova notificação via Telegram.

Os passos 3, 4 e 5 são relacionados ao cadastro do *username* (nome de usuário do telegram). No passo 3, se o usuário já possuir um *username* ele deve coloca-lo, e se não possuir existe um modal conforme é mostrado na Figura 12 explicando como criar um *username*.

O passo 4 informa que o usuário deve enviar qualquer palavra para o bot denominado “bot_smartbee_server”, nesse mesmo passo existe um modal explicativo conforme a Figura 13. O

Figura 11 – Formulário de Cadastro de Nova Notificação Via Telegram

NOVA NOTIFICAÇÃO VIA TELEGRAM DA COLMEIA

Configurar Notificações de Telegram

Passo 1: Escolha a Colmeia

Colmeias Disponíveis (2)

ESCOLHA A COLMEIA	NOME
<input type="radio"/>	Colmeia 2
<input checked="" type="radio"/>	Colmeia 3

Passo 2: Nome da Notificação

Passo 3: Username do Telegram

Passo 4: Agora envie a palavra "cadastro" para o usuário "server_smartbee_ufc" a partir de seu telegram Link

Passo 5: Clique no botão "Confirmar Username" para certificar-se que seu usuário foi cadastrado, após isso prossiga com as configurações!

Passo 6: Escolha o Critério de Verificação

Passo 7: Verificar Atraso na Coleta

Passo 8: Configure o Envio das Notificações

Ativar todas as Notificações de Grandezas

Endereço: Rua Campus do Pici - Campus do Pici - Bloco 942-A Fortaleza | Ceará | Brasil CEP: 60440-554

Fonte: Autora (2018).

passo 5 informa apenas que o usuário deve clicar no botão confirmar para verificar se o cadastro do *username* obteve sucesso ou não. Essa checagem de *username* é realizada por meio de uma função no *Controller* que faz uma requisição a uma página que lista as atualizações no bot (como as mensagens recebidas pelo bot) por meio do método *getUpdates* do Telegram. O retorno dos dados é no formato JSON. A função realiza uma busca pelos dados JSON, comparando se existe alguma mensagem enviada do *username* digitado, se sim, ele retorna o valor do *username* e id do usuário.

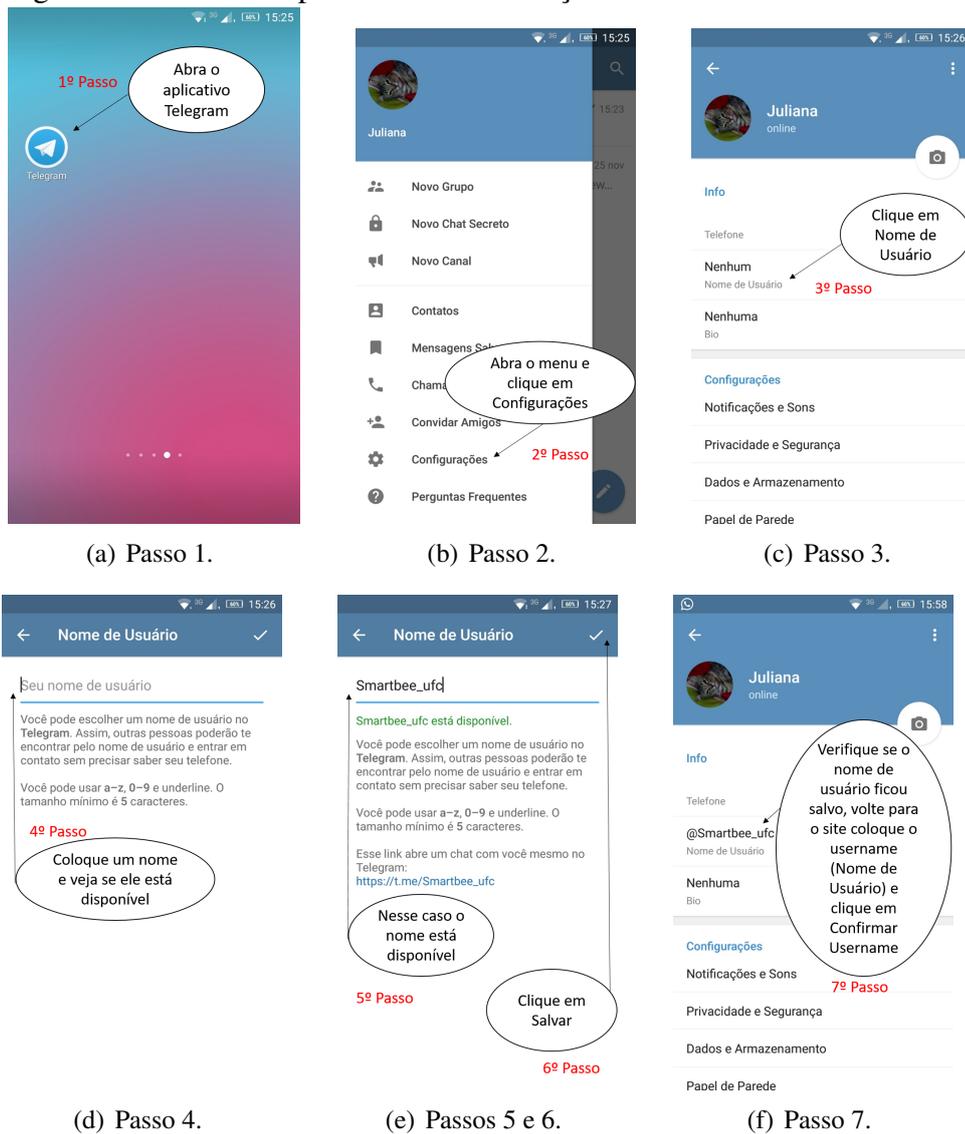
A Figura 14 mostra um caso em que o *username* foi cadastrado com sucesso. Os demais campos são semelhantes aos campos 3, 4 e 5 dos cadastros de Telegram e SMS referentes a checagem de configuração, verificação de atraso e escolha do estado da notificação.

5.1.1.3 Listagem, Edição e Exclusão das Notificações

Na listagem das notificações cadastradas, cada entrada possui o ID da notificação, nome da notificação, nome da colmeia que está sendo monitorada, o estado das notificações, um *link* para a edição da notificação observada e um botão caso o usuário deseje apagar a notificação. O formulário de edição é idêntico ao de cadastro de nova notificação, o que diferencia os dois é que o formulário de edição já vem preenchido com os dados cadastrados anteriormente.

O botão de apagar é simples e ao excluir uma notificação ele retorna sucesso ou erro na exclusão e redireciona para a tela atual de listagem. A listagem das notificações é semelhante para todas as tecnologias estudadas. A Figura 15 mostra a listagem de notificações cadastradas via E-mail.

Figura 12 – Modal Explicativo Para a Criação de um *Username*



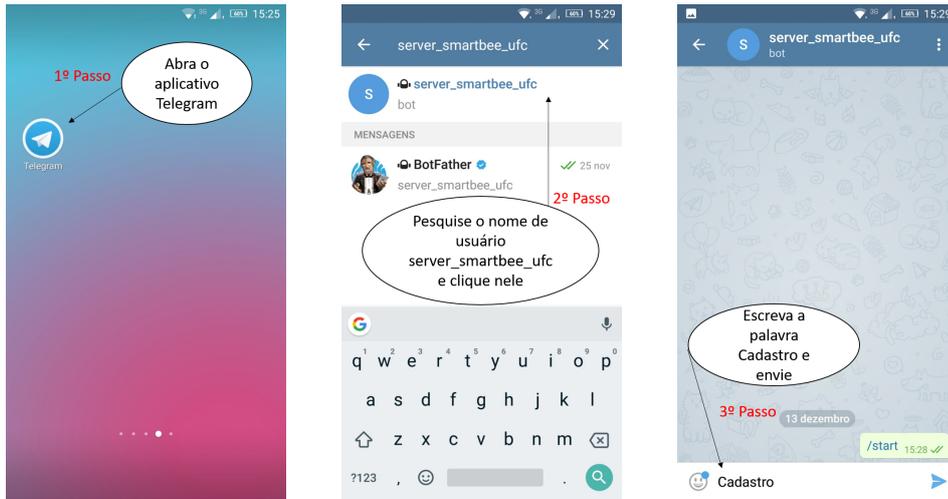
Fonte: Autora (2018).

5.1.1.4 Listagem das Notificações Enviadas

A listagem das notificações enviadas via E-mail e Telegram segue o padrão da Figura 16. Para cada mensagem enviada são mostrados o id da mensagem, nome da notificação, nome da colmeia e data e horário do envio.

Já a listagem das mensagens enviadas via SMS possuem o id, nome da notificação, a colmeia, o número de mensagens agregadas, o valor da notificação, a data e a hora do envio. Além de possuir o valor total gasto até o momento com o envio de notificações via SMS. A Figura 17 mostra a listagem de notificações enviadas para uma determinada usuária.

Figura 13 – Modal Explicativo Para o Envio de Mensagens Para o Bot



(a) Passo 1.

(b) Passo 2.

(c) Passo 3.

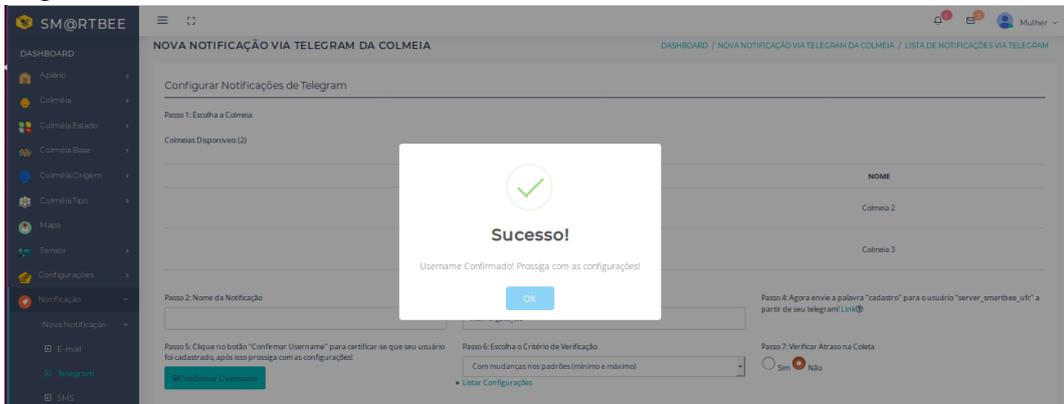


(d) Passo 4.

(e) Passo 5.

Fonte: Autora (2018).

Figura 14 – Aviso de Sucesso no Cadastro de Username



Fonte: Autora (2018).

Figura 15 – Listagem das Notificações Cadastradas Via E-mail

SM@RTBEE | Colméia Base | Colméia Origem | Colméia Tipo | Mapa | Sensor | Configurações | **Notificação** | Nova Notificação | Listar Notificações | E-mail | Telegram | SMS | Notificações Enviadas

LISTA DE NOTIFICAÇÕES DE E-MAIL DAS COLMEIAS

Estado da Notificação

LISTA DE ESTADOS DAS NOTIFICAÇÕES (3)

ID DA NOTIFICAÇÃO	NOME DA NOTIFICAÇÃO	NOME DA COLMEIA	ESTADO DA NOTIFICAÇÃO	AÇÕES
20	AP0C1-C4	Colmeia 4	ativadas	Apagar, Editar
21	PP0C0-C5	Colmeia 5	personalizadas	Apagar, Editar

Endereço: Rua Campus do Pici - Campus do Pici - Bloco 942-A Fortaleza | Ceará | Brasil CEP: 60440-554

Fonte: Autora (2018).

Figura 16 – Listagem das Notificações Enviadas Via E-mail

SM@RTBEE | Apêlrio | Colméia | Colméia Estado | Colméia Base | Colméia Origem | Colméia Tipo | Mapa | Sensor | Configurações | **Notificação** | Nova Notificação | Listar Notificações | Notificações Enviadas | E-mail | Telegram | SMS

LISTA DE NOTIFICAÇÕES ENVIADAS VIA E-MAIL

Nome da Colmeia

TOTAL DE NOTIFICAÇÕES ENVIADAS (28)

ID	NOME DA NOTIFICAÇÃO	NOME DA COLMEIA	HORÁRIO E DATA DO ENVIO
5	PP1C1-C6	Colmeia 6	18:00:22 09/05/2018
6	AP0C1-C4	Colmeia 4	18:00:24 09/05/2018
8	PP0C0-C5	Colmeia 5	18:00:30 09/05/2018
14	AP0C1-C4	Colmeia 4	18:30:13 09/05/2018
19	AP0C1-C4	Colmeia 4	19:00:13 09/05/2018

Endereço: Rua Campus do Pici - Campus do Pici - Bloco 942-A Fortaleza | Ceará | Brasil CEP: 60440-554

Fonte: Autora (2018).

Figura 17 – Listagem das Notificações Enviadas Via SMS

SM@RTBEE | Colméia Base | Colméia Origem | Colméia Tipo | Mapa | Sensor | Configurações | **Notificação** | Nova Notificação | Listar Notificações | Notificações Enviadas | E-mail | Telegram | **SMS**

LISTA DE NOTIFICAÇÕES ENVIADAS VIA SMS

Nome da Colmeia

TOTAL DE NOTIFICAÇÕES ENVIADAS (28)
 *TOTAL DE MENSAGENS (54)
 *VALOR TOTAL DAS MENSAGENS R\$ 4.86

ID	NOTIFICAÇÃO	COLMEIA	MENSAGENS AGREGADAS	VALOR	HORÁRIO E DATA DO ENVIO
1	PP1C1-C6	Colmeia 6	1	R\$ 0.09	18:00:10 09/05/2018
6	AP0C1-C4	Colmeia 4	1	R\$ 0.09	18:00:11 09/05/2018

Endereço: Rua Campus do Pici - Campus do Pici - Bloco 942-A Fortaleza | Ceará | Brasil CEP: 60440-554

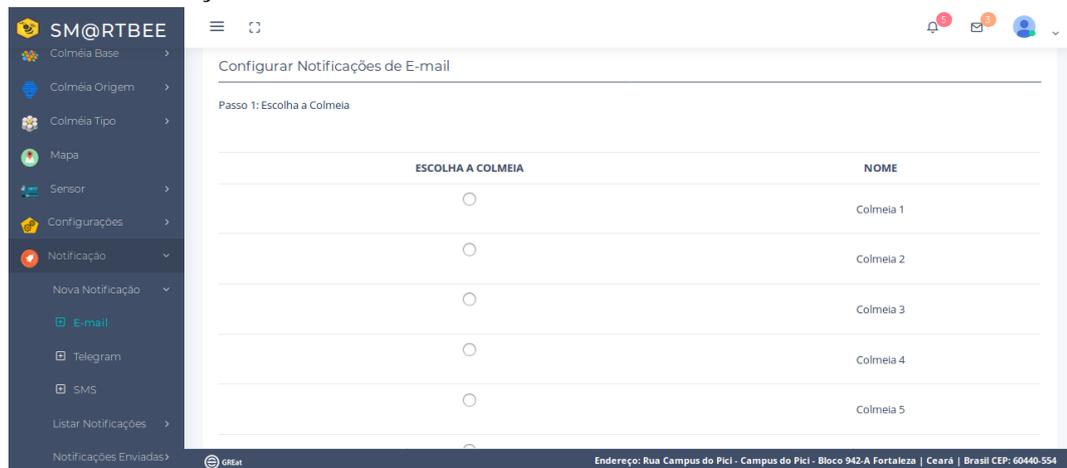
Fonte: Autora (2018).

5.1.1.5 Cadastro, listagens, edições e exclusões do Administrador

O usuário administrador pode visualizar e editar todas as notificações cadastradas, cadastrar novas notificações de colmeias para outros usuários, mas só pode excluir as notificações pertencentes a ele (administrador). Para realização dessas atividades, foram realizadas checagens individuais em cada tela. A primeira refere-se ao cadastro de novas notificações, a segunda a listagem das notificações de todos os usuários, a terceira é relacionada a exclusão de notificações apenas se o usuário logado for o dono da notificação, a quarta checagem diz respeito a listagem de todas as notificações enviadas.

O cadastro de uma nova notificação a ser realizada pelo administrador possui os mesmos campos das notificações já descritas nos tópicos acima, a única alteração é que todas as colmeias são listadas independentemente do dono da mesma. A listagem das colmeias que podem ser escolhidas pode ser vista na Figura 18.

Figura 18 – Listagem de Colmeias Disponíveis Para a Criação de uma Nova Notificação

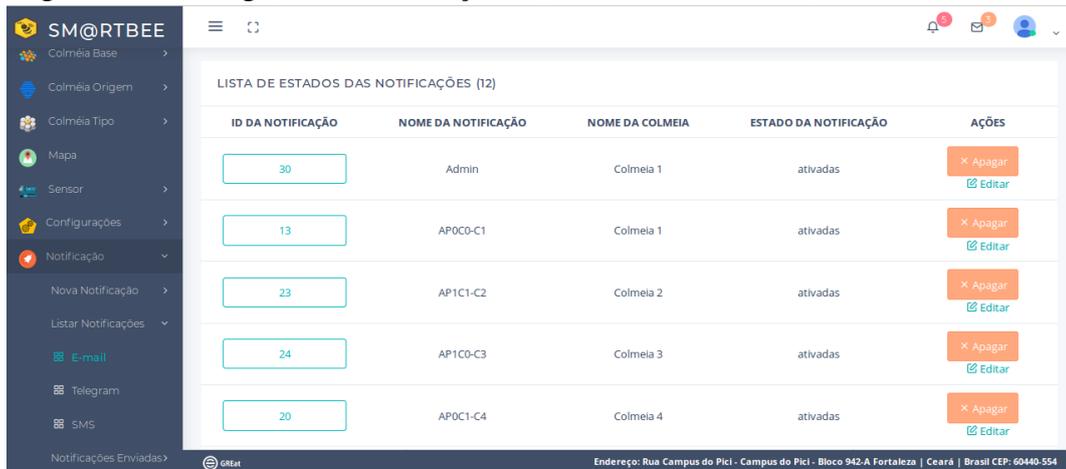


Fonte: Autora (2018).

A listagem das notificações cadastradas de todos os usuários pode ser vista na figura Figura 19.

Nesta mesma página o administrador pode optar por editar ou excluir alguma notificação. Se a notificação pertencer ao usuário logado ele poderá editá-la, se não, ele é redirecionado para a listagem de notificações. Se o usuário optar por excluir a notificação e a notificação não pertencer a ele uma mensagem é retornada avisando que a notificação não pertence a ele conforme a Figura 20. Caso a notificação pertencer a ele, exclui-se a notificação e é retornada uma mensagem de sucesso de acordo com a Figura 21.

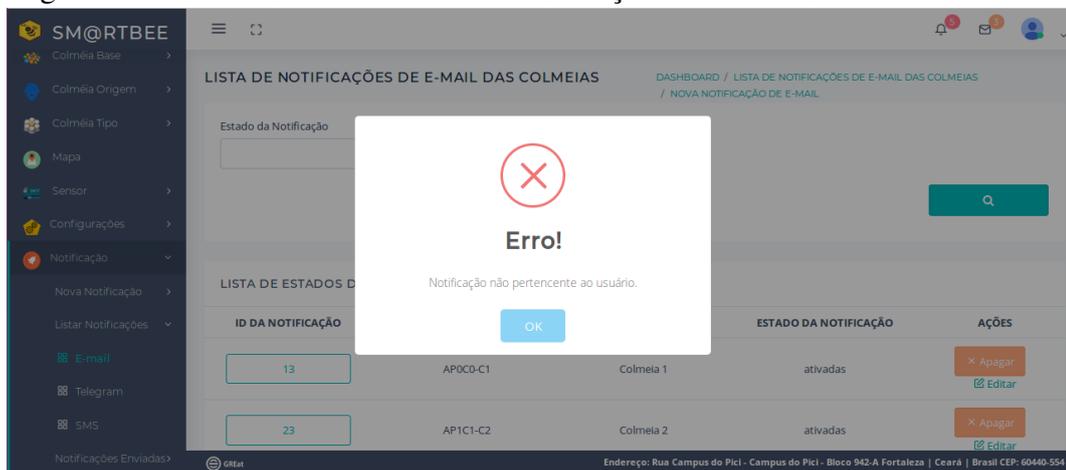
Figura 19 – Listagem das Notificações Cadastradas



ID DA NOTIFICAÇÃO	NOME DA NOTIFICAÇÃO	NOME DA COLMEIA	ESTADO DA NOTIFICAÇÃO	AÇÕES
30	Admin	Colmeia 1	ativadas	Apagar, Editar
13	APOC0-C1	Colmeia 1	ativadas	Apagar, Editar
23	AP1C1-C2	Colmeia 2	ativadas	Apagar, Editar
24	AP1C0-C3	Colmeia 3	ativadas	Apagar, Editar
20	APOC1-C4	Colmeia 4	ativadas	Apagar, Editar

Fonte: Autora (2018).

Figura 20 – Erro na Exclusão de uma Notificação



Dashboard / LISTA DE NOTIFICAÇÕES DE E-MAIL DAS COLMEIAS / NOVA NOTIFICAÇÃO DE E-MAIL

Estado da Notificação

Erro!

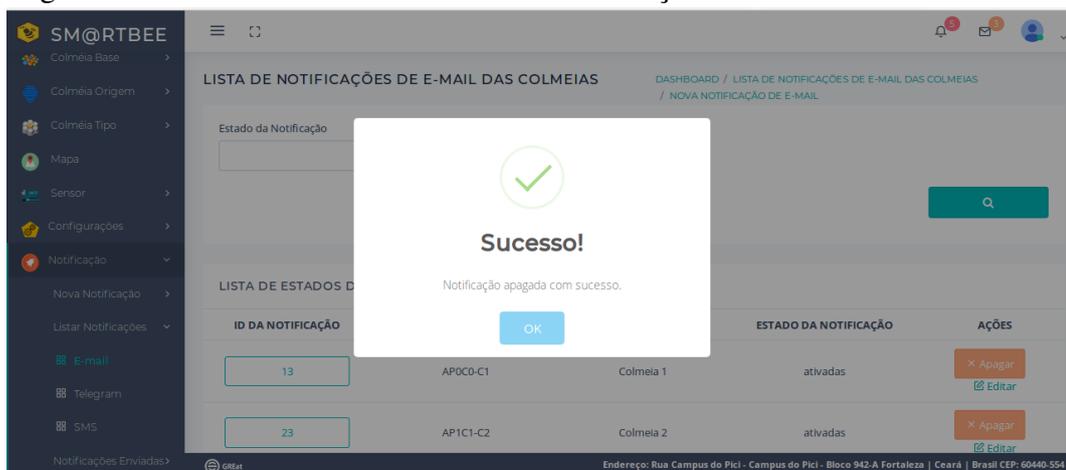
Notificação não pertencente ao usuário.

OK

ID DA NOTIFICAÇÃO	NOME DA NOTIFICAÇÃO	NOME DA COLMEIA	ESTADO DA NOTIFICAÇÃO	AÇÕES
13	APOC0-C1	Colmeia 1	ativadas	Apagar, Editar
23	AP1C1-C2	Colmeia 2	ativadas	Apagar, Editar

Fonte: Autora (2018).

Figura 21 – Sucesso na Exclusão de uma Notificação



Dashboard / LISTA DE NOTIFICAÇÕES DE E-MAIL DAS COLMEIAS / NOVA NOTIFICAÇÃO DE E-MAIL

Estado da Notificação

Sucesso!

Notificação apagada com sucesso.

OK

ID DA NOTIFICAÇÃO	NOME DA NOTIFICAÇÃO	NOME DA COLMEIA	ESTADO DA NOTIFICAÇÃO	AÇÕES
13	APOC0-C1	Colmeia 1	ativadas	Apagar, Editar
23	AP1C1-C2	Colmeia 2	ativadas	Apagar, Editar

Fonte: Autora (2018).

A listagem de todas as notificações enviadas pode ser vista na figura Figura 22. Vale ressaltar que o valor retornado no caso da listagem de notificações enviadas via SMS é apenas

das notificações do administrador.

Figura 22 – Listagem das Notificações Enviadas

ID	NOTIFICAÇÃO	COLMEIA	MENSAGENS AGREGADAS	VALOR	HORÁRIO E DATA DO ENVIO
1	PP1C1-C6	Colmeia 6	1	R\$ 0.09	18:00:10 09/05/2018
2	AP1C0-C9	Colmeia 9	2	R\$ 0.18	18:00:10 09/05/2018
3	AP1C1-C2	Colmeia 2	1	R\$ 0.09	18:00:10 09/05/2018
4	AP0C0-C1	Colmeia 1	2	R\$ 0.18	18:00:11 09/05/2018
5	PP1C0-C11	Colmeia 11	2	R\$ 0.18	18:00:11 09/05/2018
6	AP0C1-C4	Colmeia 4	1	R\$ 0.09	18:00:11 09/05/2018

Fonte: Autora (2018).

5.1.1.6 Scripts de Envio das Notificações

Foram criados três *scripts* para o processamento das notificações, um para cada tecnologia de envio de notificação. Um *script* com funções para E-mail e Telegram e um *script* com funções para as notificações via SMS. Houve essa separação devido a formatação das mensagens, pois o mensagens via SMS não aceitam caracteres especiais, e foi reduzido o tamanho das mensagens para uma maior economia. Todos os *scripts* foram feitos utilizando a linguagem PHP. O formato dos *scripts* criados para o processamento das notificações segue as etapas a seguir:

- a) Conexão com o banco de dados;
- b) Consulta ao banco de dados de todas as notificações cadastradas e coletas;
- c) As notificações entram em um laço, e para cada colmeia na notificação é realizada uma nova consulta no banco de dados para pegar as 10 últimas notificações enviadas;
- d) É chamada uma função que transforma a data e hora coletada em minutos, retorna esse tempo e a data e hora do sistema;
- e) Cada notificação é comparada no que diz respeito se são notificações ativadas (todas as grandezas) ou personalizadas (somente as grandezas que o usuário escolheu);
- f) Se forem ativadas:

1. A primeira condicional compara se não existe coleta para a colmeia, se a notificação necessita de checagem de atraso e se o tempo passado por parâmetro é igual ao do atraso. Após isso é verificado se a soma das ausências é menor que 10 ou vazia, se sim, é enviada a notificação informando a ausência da coleta;
 2. A segunda condicional compara se o período é igual a "0", ou seja, se o usuário optou por não comparar os valores coletados com as configurações dele.
 - Nesse caso, a primeira condicional compara se a notificação exige checagens de atraso, se o tempo de atraso da coleta é maior que o intervalo de atraso da notificação, se o intervalo de atraso é igual o tempo recebido por parâmetro e se existe coleta para a colmeia. Após isso é verificado se a soma dos atrasos é menor que 10 ou vazia, se sim, é enviada a notificação informando o atraso na coleta;
 - A segunda condicional compara se a notificação exige comparações com o atraso, se o tempo de atraso na coleta é menor ou igual ao intervalo do atraso, se o intervalo da notificação é igual ao tempo recebido por parâmetro e se a coleta existe. Ou ainda se a notificação não exige comparações com o atraso e se o intervalo de envio é igual ao tempo recebido por parâmetro. Nesse caso é chamada uma função que checa para cada grandeza se existe coleta para a mesma, e então é realizado o retorno de todas as mensagens concatenadas. É então chamada uma nova função para o envio da notificação com todos os valores coletados, se houver sucesso no envio é salvo no banco de dados os valores referentes ao nome da notificação, nome e id da colmeia, atraso e id do usuário.
 3. A Terceira condicional compara se o período é igual a "1", ou seja, se o usuário optou por comparar os valores coletados com as configurações dele. As condicionais são iguais ao período igual a "0", porém somente são enviados os valores que possuem alterações em comparação com as configurações do usuário.
- g) Se forem personalizadas:
- Seguem as mesmas condições das notificações ativadas, porém, o tempo

recebido como parâmetro é passado para uma função que realiza a checagem do intervalo de envio para cada grandeza que o usuário escolheu.

6 RESULTADOS E DISCUSSÃO

A validação do sistema proposto foi realizada buscando-se verificar o correto funcionamento das funcionalidades de envio de notificações via E-mail, Telegram e SMS no sistema *web Sm@rtbee*. Foram observados o tempo de processamento no envio das mensagens e o correto envio, ou seja, se a mensagem foi entregue de acordo com as configurações do usuário. O tempo considerado para a comparação no tempo de processamento foi a hora atual do sistema que é enviada juntamente com a notificação e não o horário que a notificação foi salva no banco, pois a notificação só é salva no banco de dados após o envio.

Essa seção é dividida em (i) planejamento e execução e (ii) resultados. A seção de planejamento e execução detalha os passos para realização dos teste e por fim, na seção de resultados mostramos o tempo de processamento no envio das notificações e algumas notificações recebidas.

6.1 Planejamento e Execução

O sistema *web Sm@rtbee* foi instalado e configurado em uma máquina virtual 64 bits rodando o sistema operacional Ubuntu 16.04 no VirtualBox versão 5.1.26. Após isso, foram adicionadas as funcionalidades de cadastro, listagem, edição e exclusão de notificações.

Para executar os scripts de envio, o cron usa o arquivo crontab (MALHEIROS, 2016). Para isso, utilizou-se o comando "crontab -e" e adicionou-se os scripts de envio para cada tecnologia, especificando-se o tempo de execução de acordo com os intervalos de envio das notificações. Para deixar o script pronto para qualquer intervalo, ao executar o script é passado como parâmetro o tempo de envio.

Para verificar o tempo de processamento e a verificação de envio das notificações foi criado um *script* na linguagem PHP disponível no Apêndice B simulando o envio de coletas de 8 colmeias. O *script* foi executado uma vez e iniciado as 18:02:06 no dia 09 de Maio de 2018 enviando dados alterados para cada colmeia. O intervalo entre as inserções foi de cinco minutos. As 19:57:17 iniciou-se uma pausa de duas horas para gerar atrasos nas coletas. As 21:57:17 foi reiniciada a inserção de coletas, sendo que a primeira coleta é alterada e as demais dentro dos padrões. O *script* encerrou-se às 22:57:19. A Tabela 4 mostra a distribuição dos períodos observados.

¹ Em algumas notificações estudadas a coleta não estaria em atraso.

² Apesar das variáveis estarem dentro dos padrões algumas notificações foram cadastradas para enviar os dados

Tabela 4 – Distribuição dos Períodos Observados

Período	Horário	Quantidade de Notificações Por Tecnologia	Estado das Variáveis Coletadas
1	18:00:00	8	Ausentes
2	18:30:00	2	Alteradas
3	19:00:00	7	Alteradas
4	19:30:00	2	Alteradas
5	20:00:00	9	Alteradas
6	20:30:00	4	Em atraso ¹
7	21:00:00	6	Em atraso ¹
8	21:30:00	2	Em atraso ¹
9	22:00:00	9	Alteradas
10	22:30:00	2	Dentro dos padrões ²
11	23:00:00	5	Dentro dos padrões ²
12	23:30:00	4	Em atraso ¹
13	00:00:00	6	Em atraso ¹

Fonte: Autora (2018).

Para verificar se as tecnologias estavam comparando a existência de atraso ou ausência de coletas, as notificações foram observadas a partir do primeiro período, pois não existiam coletas para nenhuma das colmeias devido o *script* ter sido iniciado às 18:02:07.

Foram criados 3 usuários e nomeados da seguinte forma: Capitão América (proprietário das colmeias 1, 2 e 3), Mulher Gato (proprietária das colmeias 4, 5 e 6) e Batman (Proprietário das colmeias 7 e 8). O Apêndice A descreve as notificações e configurações de cada usuário.

6.2 Resultados dos Experimentos

Todas as mensagens enviadas foram recebidas de acordo com as notificações cadastradas dos usuários. O tempo de processamento não ultrapassou a faixa de segundos. Como no tempo observado não existiriam 10 mensagens consecutivas de ausência ou atraso de coletas, foram criadas novas colmeias sem coletas e verificado se as mensagens não eram enviadas após 10 mensagens consecutivas de atraso ou ausência da coleta, nesse caso todas as tecnologias não enviaram após as 10 mensagens.

A Figura 23 mostra os gráficos dos intervalos de confiança da média e média do processamento do envio de notificações das tecnologias estudadas para os períodos 1, 2, 3, 4, 5, 6, 7 e 8.

A Figura 24 mostra os gráficos dos intervalos de confiança da média e média do processamento de envio das tecnologias estudadas para os períodos 9, 10, 11, 12 e 13. O intervalo independente das configurações.

de confiança usado no experimento foi de 95%.

Como pode ser observado na Figura 23 de acordo com os períodos 1, 2, 3, 4, 5, 6, 7 e 8 e Figura 24 períodos 9 e 10 o envio de notificações por E-mail é o que demanda um maior tempo de processamento. Já o tempo de processamento no envio de notificações via Telegram e SMS são relativamente menores.

O gráfico da Figura 24 para os períodos 11, 12 e 13, foram os únicos casos onde houveram diferenças em relação aos já citados. Como pode ser visto nos períodos 11 e 13 os limites inferiores dos intervalos de confiança do E-mail ficaram próximos dos limites superiores dos intervalos de confiança do Telegram. O único período onde houve proximidade no tempo de processamento entre as três tecnologias foi o período 12.

A Tabela 5 mostra os valores referentes ao tempo de processamento médio e intervalo de confiança da média em segundos das notificações enviadas por cada tecnologia.

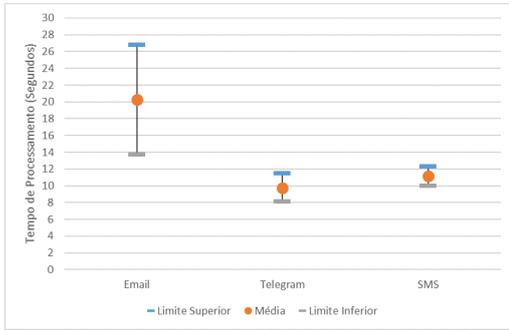
Somando-se o total de todas as notificações enviadas por cada tecnologia obtém-se um total de 66 notificações por tecnologia. O gráfico da Figura 25 mostra o intervalo de confiança da média do tempo de processamento encontrada em segundos.

A média do tempo processamento via E-mail foi de 15,24 segundos, limite superior de 17,43 segundos e limite inferior de 13,14 segundos. Para o envio via Telegram a média foi de 4,53 segundos, limites superior e inferior de 5,25 e 3,80 respectivamente. Para o envio via SMS a média, limite superior e limite inferior os valores foram: 4,19; 4,96 e 3,42 segundos respectivamente.

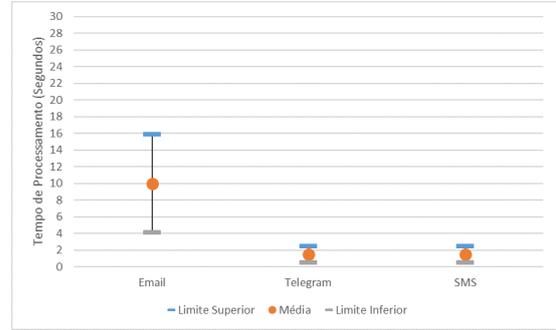
De acordo com os dados apresentados pode-se concluir que a tecnologia de envio de E-mail possui um maior tempo de processamento em relação as demais. Um fator que pode resultar nessa diferença é que no envio de notificações por E-mail foi utilizada a classe do PHPMailer onde é necessário iniciar uma conexão com o servidor de e-mail SMTP (*Simple Mail Transfer Protocol*) (no caso do próprio gmail), para autenticação e somente após isso é que a mensagem é enviada. Já para o envio via Telegram é necessário apenas o *token* do BOT do Telegram e a mensagem é passada via URL sem a necessidade de autenticação.

Para o envio via SMS, a mensagem é encaminhada para a API da TotalVoice juntamente com o *AccessToken* de acesso via URL. Apesar dos resultados encontrados em relação ao tempo de processamento serem reduzidos, existe um tempo adicional na entrega das mensagens, visto que a mensagem é enviada ao servidor da TotalVoice onde existe um atraso no envio do servidor para o cliente. O tempo de envio pode ser verificado acessando a conta criada para este

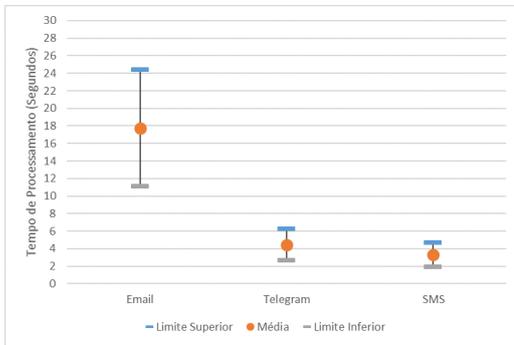
Figura 23 – Tempo de processamento A



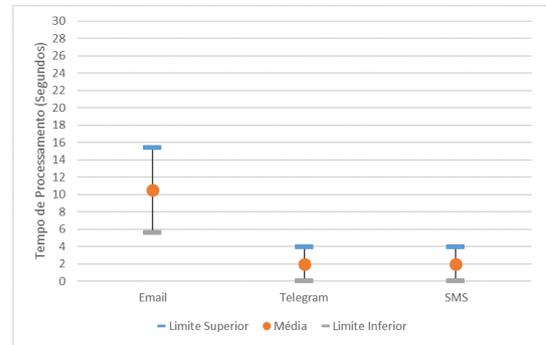
(a) Período 1.



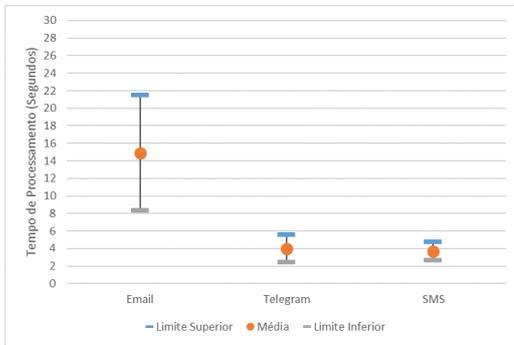
(b) Período 2.



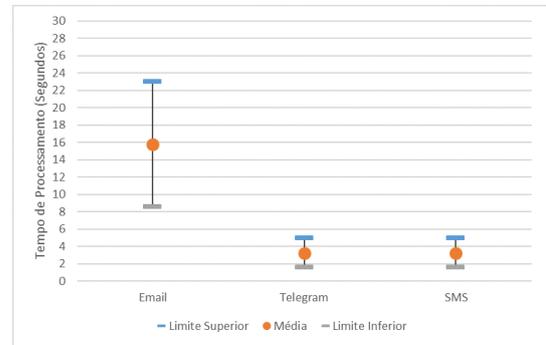
(c) Período 3.



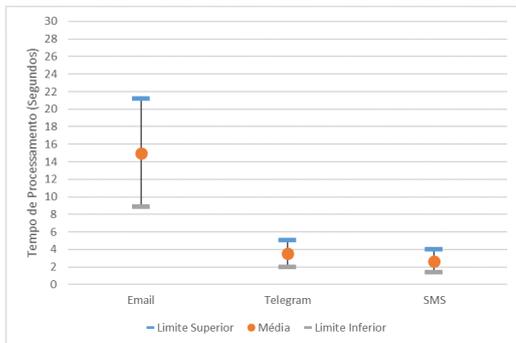
(d) Período 4.



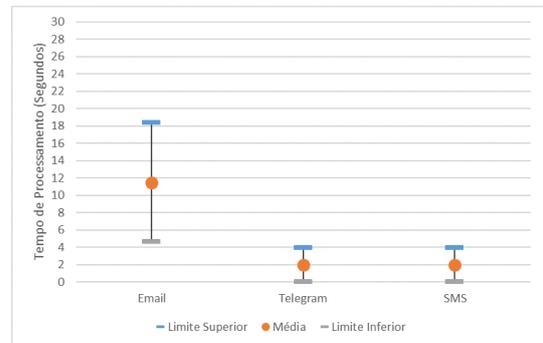
(e) Período 5.



(f) Período 6.



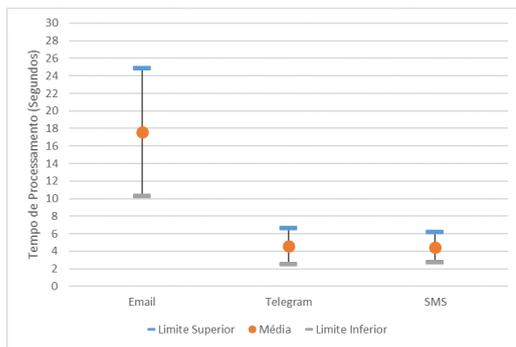
(g) Período 7.



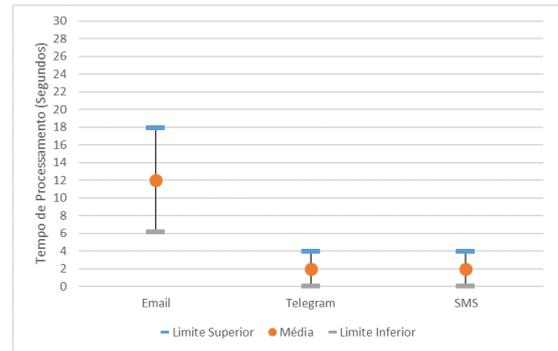
(h) Período 8.

Fonte: Autora (2018).

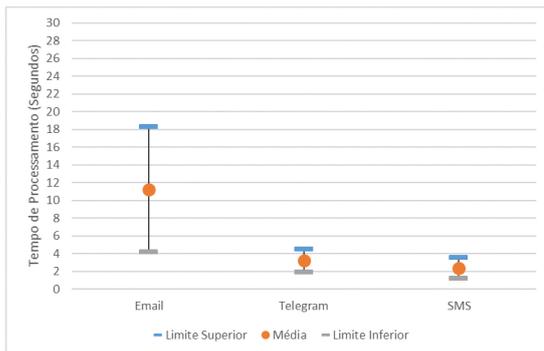
Figura 24 – Tempo de processamento B



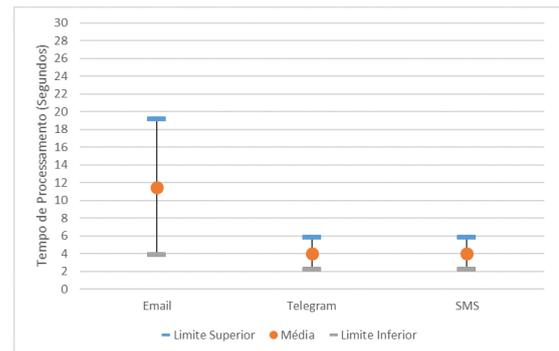
(a) Período 9.



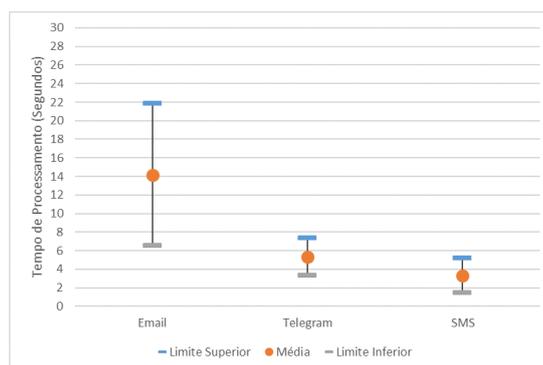
(b) Período 10.



(c) Período 11.



(d) Período 12.



(e) Período 13.

Fonte: Autora (2018).

Tabela 5 – Valores da Médias e Intervalos de Confiança da Média

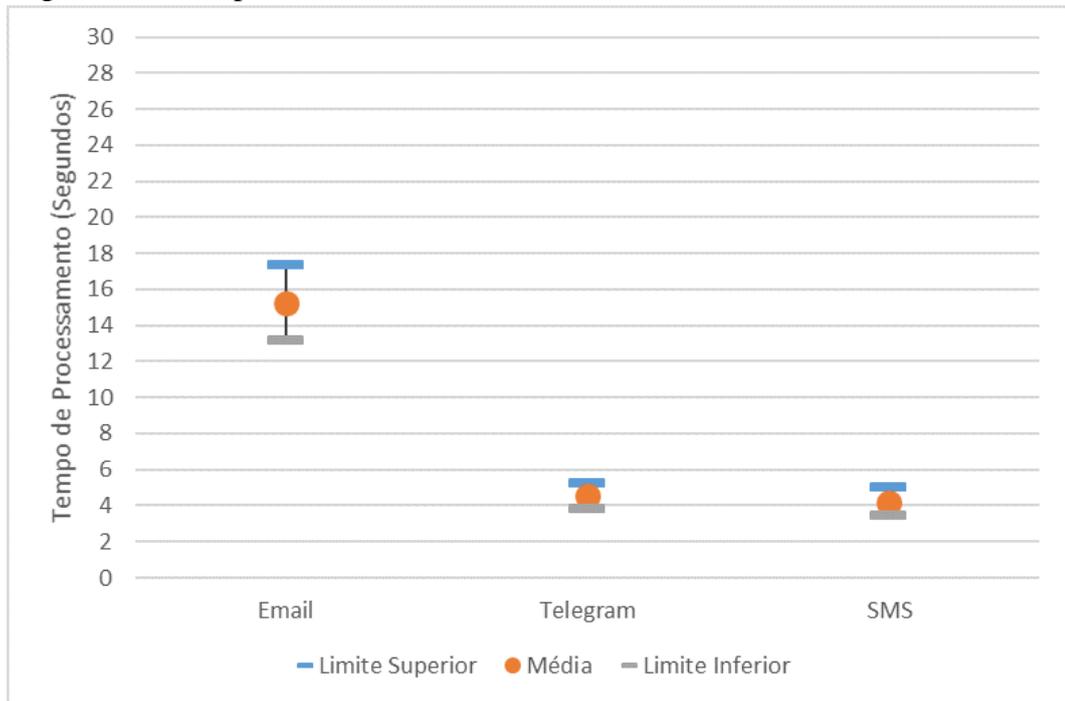
Período	Número de Amostras	Métrica (Segundos)	Email	Telegram	SMS
1	8	Limite Superior	26,78	11,43	12,26
1	8	Média	20,25	9,75	11,12
1	8	Limite Inferior	13,71	8,06	9,98
2	2	Limite Superior	15,87	2,47	2,47
2	2	Média	10	1,5	1,5
2	2	Limite Inferior	4,12	0,52	0,52
3	7	Limite Superior	24,36	6,23	4,68
3	7	Média	17,71	4,42	3,28
3	7	Limite Inferior	11,06	2,62	1,88
4	2	Limite Superior	15,39	3,95	3,95
4	2	Média	10,5	2	2
4	2	Limite Inferior	5,6	0,04	0,04
5	9	Limite Superior	21,48	5,56	4,69
5	9	Média	14,88	4	3,66
5	9	Limite Inferior	8,29	2,43	2,63
6	4	Limite Superior	22,96	4,92	4,92
6	4	Média	15,75	3,25	3,25
6	4	Limite Inferior	8,53	1,57	1,57
7	6	Limite Superior	21,17	4,99	3,97
7	6	Média	15	3,5	2,66
7	6	Limite Inferior	8,82	2	1,36
8	2	Limite Superior	18,35	3,95	3,95
8	2	Média	11,5	2	2
8	2	Limite Inferior	4,64	0,04	0,04
9	9	Limite Superior	24,81	6,62	6,14
9	9	Média	17,55	4,55	4,44
9	9	Limite Inferior	10,29	2,48	2,74
10	2	Limite Superior	17,87	3,95	3,95
10	2	Média	12	2	2
10	2	Limite Inferior	6,12	0,04	0,04
11	5	Limite Superior	18,25	4,5	3,57
11	5	Média	11,2	3,2	2,4
11	5	Limite Inferior	4,14	1,89	1,22
12	4	Limite Superior	19,11	5,78	5,78
12	4	Média	11,5	4	4
12	4	Limite Inferior	3,88	2,21	2,21
13	6	Limite Superior	21,83	7,33	5,2
13	6	Média	14,16	5,33	3,33
13	6	Limite Inferior	6,5	3,33	1,46

Fonte: Autora (2018).

fim no site da TotalVoice. Para identificar o atraso total no envio das notificações é necessário um estudo mais amplo, incluindo o tempo de processamento do servidor Sm@rtbee, o tempo de envio para o servidor que de fato irá enviar a mensagem e o tempo em que o cliente recebeu a mensagem. Esse estudo não faz parte do escopo desse trabalho.

No recebimento das mensagens via E-mail, Telegram e SMS o atraso foi inferior a 60 segundos. Isto foi verificado mensagem por mensagem onde são apresentados hora e minutos de recebimento.

Figura 25 – Tempo de Processamento de Todos os Períodos



Fonte: Autora (2018).

Em comparações com os trabalhos relacionados apenas três estudos realizaram e descreveram o envio de alertas. No trabalho de Dias (2017) não houve comparações em relação ao tempo de processamento no envio da notificação via Telegram. O estudo de Cardoso *et al.* (2016) que realizou o envio de alertas via E-mail e aplicação móvel sobre o monitoramento de córregos, no que tange ao envio de e-mails não foi realizado experimento em relação ao tempo de processamento no envio das mensagens. Já para os resultados em relação ao aplicativo móvel nenhuma consulta ao servidor demorou mais que 3 segundos para ser apresentada no aplicativo. O estudo de Razak *et al.* (2017) também não realizou experimentos acerca do tempo de processamento no envio das mensagens.

Em relação o formato das mensagens enviadas, para a tecnologia E-mail e Telegram é possível adicionar caracteres especiais. Já para envio via SMS não é possível e as palavras são enviadas sem acentuação. A Figura 26 mostra alguns tipos de mensagens recebidas pelo usuário Batman via E-mail.

A Figura 27 mostra 2 tipos de mensagens recebidas pelo usuário Capitão América via SMS. Já a Figura 28 mostra notificações recebidas pelo Capitão América via Telegram.

Figura 26 – Mensagens Recebidas Via E-mail

<p>Usuário: Batman Nome da Notificação: PP0C1-C8 Nome da Colmeia: Colmeia 8 Assunto: Alerta Smartbee</p> <p>Dados</p> <p>Coleta ausente!  Envio da Notificação: 09-05-2018 18:00:35</p>	<p>Usuário: Batman Nome da Notificação: PP1C0-C7 Nome da Colmeia: Colmeia 7 Assunto: Alerta Smartbee</p> <p>Dados</p> <p>Coleta de Temperatura Ausente Coleta de Umidade Ausente  Coleta Não Existe  Envio da Notificação: 09-05-2018 18:00:13</p>
(a) Coleta ausente.	(b) Coleta ausente.
<p>Usuário: Batman Nome da Notificação: PP0C1-C8 Nome da Colmeia: Colmeia 8 Assunto: Alerta Smartbee</p> <p>Dados</p> <p>Temperatura: 68 °C Umidade: 80 %  Última Coleta: 09-05-2018 18:27:12  Envio da Notificação: 09-05-2018 18:30:13</p>	<p>Usuário: Batman Nome da Notificação: PP0C1-C8 Nome da Colmeia: Colmeia 8 Assunto: Alerta Smartbee</p> <p>Dados</p> <p>Som: 68 dB Dióxido de Carbono: 800 ppm Oxigênio: 80 % Peso: 88 Kg  Última Coleta: 09-05-2018 18:57:15  Envio da Notificação: 09-05-2018 19:00:32</p>
(c) Coleta sem comparações com os padrões.	(d) Coleta sem comparações com os padrões.

Usuário: Batman
Nome da Notificação: PP1C0-C7
Nome da Colmeia: Colmeia 7
Assunto: Alerta Smartbee

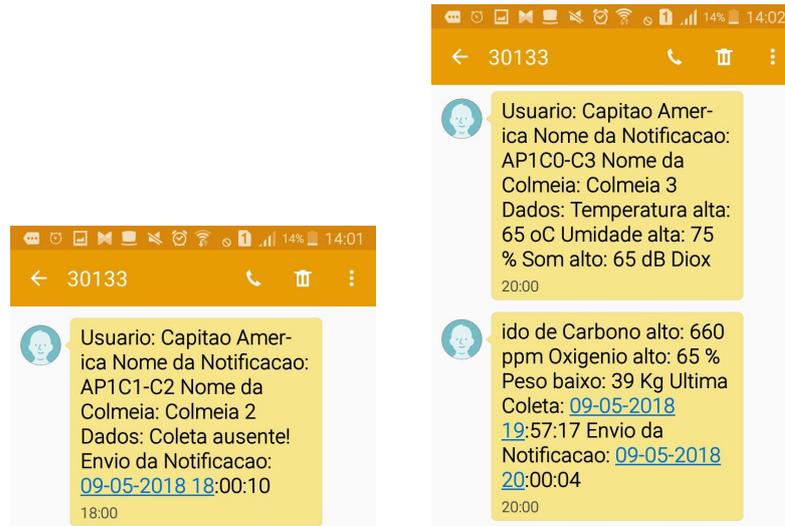
Dados

↑ Temperatura alta: 69°C
↑ Umidade alta: 70%
 Última Coleta: 09-05-2018 19:57:17
 Envio da Notificação: 09-05-2018 20:00:07

(e) Coleta com comparações com os padrões.

Fonte: Autora (2018).

Figura 27 – Mensagens Recebidas Via SMS

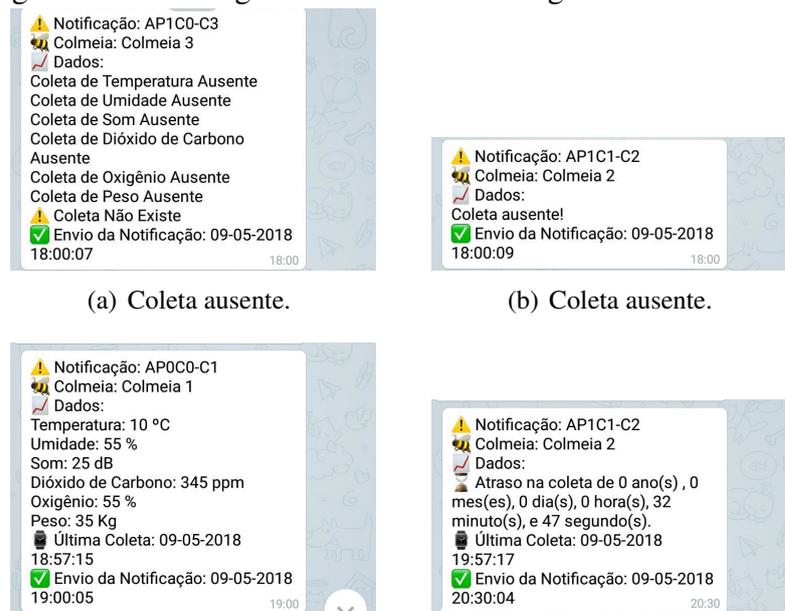


(a) Coleta ausente.

(b) Coleta com comparações com os padrões.

Fonte: Autora (2018).

Figura 28 – Mensagens Recebidas Via Telegram

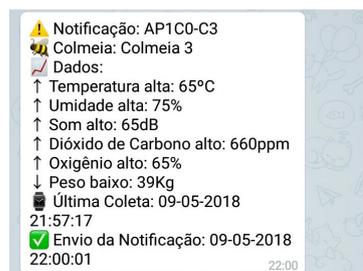


(a) Coleta ausente.

(b) Coleta ausente.

(c) Coleta sem comparações com os padrões.

(d) Coleta em atraso.



(e) Coleta com comparações com os padrões.

Fonte: Autora (2018).

7 CONCLUSÕES E TRABALHOS FUTUROS

Apesar do sistema de notificações não ter sido testado em um ambiente real com sensores enviando os dados coletados ao banco de dados, pode-se concluir que os objetivos foram devidamente alcançados.

De acordo com os resultados apresentados, o tempo de processamento no envio das mensagens foi considerado aceitável apesar do número de notificações enviadas no experimento ter sido reduzido. Todas as mensagens enviadas foram recebidas de acordo com as notificações cadastradas dos usuários.

As tecnologias de envio via E-mail e Telegram não implicam em custos, e é enviada uma única mensagem por notificação. Já via SMS, como existe um limite de 160 caracteres por mensagem, podem ocorrer envios de duas mensagens de uma única notificação, principalmente nas notificações personalizadas, acarretando assim em um custo que pode ser considerado pelo usuário como alto. Porém, o envio de mensagens via SMS apesar das demais tecnologias serem gratuitas, é essencial para os apicultores, pois caso os mesmos não possuam acesso a internet, poderão receber as notificações sem problemas caso possuam cobertura do sinal celular.

Para uma avaliação mais realística das tecnologias de envio, como um trabalho futuro a ser realizado, seria um estudo utilizando toda a arquitetura do sistema Sm@artbee, verificando não apenas o tempo de processamento no envio, mas também o atraso total no envio.

Os códigos-fonte se encontram disponíveis em <https://gitlab.com/julianacastro/smartbeev2/>.

REFERÊNCIAS

- ADECE. **Mel**. Fortaleza-CE, 2018. Disponível em: <<http://www.adece.ce.gov.br/index.php/mel>>. Acesso em: 01 jan. 2018.
- AGUIAR, Í. F. **Proposta de utilização da ferramenta Zabbix no gerenciamento de redes: um estudo de caso no ambiente da fab segundo boas práticas de governança de TI**. 2017. 59 f. Monografia (Trabalho de conclusão de especialização) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017. Disponível em: <<http://hdl.handle.net/11422/3300>>. Acesso em: 02 abr. 2018.
- ALMEIDA, A. R. *et al.* Projeto e desenvolvimento de laboratório móvel para monitoramento ambiental dos níveis de temperatura e humidade baseado em internet das coisas (iot). **CRICTE**, n. 28, 2018. Disponível em: <<https://publicacoeseventos.unijui.edu.br/index.php/cricte/article/view/8855>>. Acesso em: 04 abr. 2018.
- ALMEIDA, A. W. S. de; SILVA, T. F. dos S.; SILVA, T. M. **Proposta e modelagem de um sistema de informação para gerenciamento de apiário**. 2013. 64 f. Monografia (Bacharelado em Sistemas de Informação) - Instituto Federal de Minas Gerais, São João Evangelista - Minas Gerais, 2013. Disponível em: <http://www.sje.ifmg.edu.br/portal/images/artigos/biblioteca/TCCs/Sistemas_de_informacao/2013/ALBERT_WESLEY_SILVA_DE_ALMEIDA_THALLYS_FREED_DOS_SANTOS_SILVA_THIAGO_MOREIRA_SILVA.pdf>. Acesso em: 02 abr. 2018.
- ALMEIDA, C. M. P. d. **Sistema de monitorização e alarmística de ativos de rede**. 2015. 143 f. Dissertação (Mestrado em Engenharia Informática) — Instituto Superior de Engenharia do Porto, Porto, 2015. Disponível em: <http://recipp.ipp.pt/bitstream/10400.22/8034/1/DM_CarlosAlmeida_2015_MEI.pdf>. Acesso em: 02 abr. 2018.
- AMNUR, H. *et al.* Perancangan dan implementasi network monitoring sistem menggunakan nagios dengan email dan sms alert (design with implementation of network monitoring system using nagios by email and sms alert). **POLI REKAYASA**, Politeknik Negeri Padang, v. 10, n. 1, p. 42–50, 2014. Disponível em: <<http://repo.polinpdg.ac.id/65/1/854-819-1-PB.pdf>>. Acesso em: 02 abr. 2018.
- ANIF, M.; DENTHA, A.; SINDUNG, H. W. S. Designing internship monitoring system web based with laravel framework. In: **IEEE International Conference on Communication, Networks and Satellite (Comnetsat), 2017 IEEE International Conference on**. IEEE, 2017. p. 112–117. Disponível em: <<https://ieeexplore.ieee.org/document/8263583>>.
- ARAÚJO, A. *et al.* Redes sociais-telegram no ambiente escolar. **Realização e Organização**, p. 17, 2017. Disponível em: <http://mpet.ifam.edu.br/wp_seta/wp-content/uploads/2016/07/Anais_Seta-2017.pdf#page=64>. Acesso em: 01 mar. 2018.
- BOINTON, M. **PHPMailer**. Disponível em: <<https://github.com/Synchro/PHPMailer>>. Acesso em: 15 mai. 2018.
- BRITO, R. de A.; BRITO, L. A. de; MENDES, M. R. A. Flora melitófila próxima ao cultivo de aceroleiras no distrito de irrigação dos tabuleiros litorâneos (ditalpi), parnaíba-pi. **Heringeriana**, v. 11, n. 1, p. 28–38, 2018. Disponível em: <http://revistas.jardimbotanico.ibict.br/index.php/heringeriana/article/view/230/pdf_16>. Acesso em: 20 jan. 2018.

CANDRA, A.; KURNIAWAN, Y.; RHEE, K. H. Security analysis testing for secure instant messaging in android with study case: Telegram. In: **International Conference on System Engineering and Technology (ICSET)**. IEEE, 2016. p. 92–96. Disponível em: <<https://ieeexplore.ieee.org/document/7849630/>>. Acesso em: 25 fev. 2018.

CARDOSO, M. A. R. *et al.* **Alerta Poá-Um sistema de comunicação apoiado em redes de sensores sem fio para monitoração do córrego Itaim**. 2016. 127 f. Dissertação (Mestrado em Engenharia Elétrica) — Pontifícia Universidade Católica de Campinas, Campinas-SP, 2016. Disponível em: <<http://tede.bibliotecadigital.puc-campinas.edu.br:8080/jspui/handle/tede/894>>. Acesso em: 10 mar. 2018.

DIAS, F. d. O. **Um modelo proativo de antecipação de ações de times de resposta rápida baseado em análise preditiva**. 2017. 104 f. Dissertação (Mestrado Acadêmico em Computação Aplicada) — Universidade do Vale do Rio dos Sinos, São Leopoldo-RS, 2017. Disponível em: <http://www.repositorio.jesuita.org.br/bitstream/handle/UNISINOS/6228/F%c3%a1bio%20de%20Oliveira%20Dias_.pdf?sequence=1&isAllowed=y>. Acesso em: 20 fev. 2018.

DUTRA, T. F. S. **Beehiveior-Sistema de monitoramento e controle de colmeias de produção apícola**. 2016. 81 f. Dissertação (Mestrado em Engenharia de Software) — Universidade Federal do Rio Grande do Norte, Natal- RN, 2016. Disponível em: <https://repositorio.ufrn.br/jspui/bitstream/123456789/21775/1/ThiagoFernandesSilvaDutra_DISSERT.pdf>. Acesso em: 15 fev. 2018.

ESPINOZA, L. Y. A.; SALAZAR, V.; ROXANA, C. **Sistema Web de Optimizacion para la Adquisición de Equipos y Control de Servicios Técnicos de la empresa "NPC BALANZA" con Framework Laravel**. 2017. 129 f. Tese (Doutorado em Engenharia em Sistemas Computacionais) — Universidad de Guayaquil. Facultad de Ciencias Matematicas y Fisicas, 2017. Disponível em: <<http://repositorio.ug.edu.ec/bitstream/redug/19964/1/B-CISC-PTG.1284%20Ayala%20Espinoza%20Leslye%20Yanina.pdf>>. Acesso em: 20 fev. 2018.

GIT. **Git –local-branching-on-the-cheap**. Disponível em: <<https://git-scm.com/>>. Acesso em: 03 abr. 2018.

GOMES, I. **Sistema de alerta por sms e e-mail**. 2014. 107 f. Monografia (Licenciatura em Informática de Gestão) - Universidade do Mindelo, Mindelo-São Vicente, Cabo Verde, 2014. Disponível em: <<http://www.portaldoconhecimento.gov.cv/bitstream/10961/3644/1/Gomes%202014.%20Sistema%20de%20alerta%20por%20sms%20e%20e-mail.pdf>>. Acesso em: 10 abr. 2018.

HAMIDI, E. *et al.* The implementation of alert system for lan network monitoring using the dude based email. In: IOP PUBLISHING. **IOP Conference Series: Materials Science and Engineering**. 2018. v. 288, n. 1, p. 012054. Disponível em: <<http://iopscience.iop.org/article/10.1088/1757-899X/288/1/012054/pdf>>. Acesso em: 01 jan. 2018.

HARDONO *et al.* Development of theses categorization system search engine using php and mysql. In: **International Conference on Information Technology Systems and Innovation (ICITSI)**. IEEE, 2017. p. 194–199. Disponível em: <<https://ieeexplore.ieee.org/document/8267942/>>. Acesso em: 15 abr. 2018.

IBGE. **Pesquisa Pecuária Municipal**. [S.l.], 2016. Disponível em: <<https://sidra.ibge.gov.br/pesquisa/ppm/quadros>>. Acesso em: 01 jan. 2018.

- IBGE. **Pecuária Municipal 2016**:: Centro-oeste concentra 34,4% do rebanho bovino do país. [S.l.], 2017. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2013-agencia-de-noticias/releases/16992-pecuaria-municipal-2016-centro-oeste-concentra-34-4-do-rebanho-bovino-do-pais.html>>. Acesso em: 01 jan. 2018.
- ISLAM, R. U.; ANDERSSON, K.; HOSSAIN, M. S. Heterogeneous wireless sensor networks using coap and sms to predict natural disasters. In: **IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. IEEE, 2017. p. 30–35. Disponível em: <<https://ieeexplore.ieee.org/document/8116348/>>. Acesso em: 10 mar. 2018.
- KVIESIS, A.; ZACEPINS, A. System architectures for real-time bee colony temperature monitoring. **Procedia Computer Science**, v. 43, n. Supplement C, p. 86 – 94, 2015. ISSN 1877-0509. ICTE in Regional Development, December 2014, Valmiera, Latvia. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050914015804>>. Acesso em: 10 jan. 2018.
- LANZA, B. B. B. **Configurações e modelo de negócios em governo móvel**: o uso do short message service (sms). 2016. 251 f. Tese (Doutorado em Administração) — Universidade Federal do Paraná, Curitiba, 2016. Disponível em: <<https://acervodigital.ufpr.br/bitstream/handle/1884/43951/R\%20-\%20T\%20-\%20BEATRIZ\%20BARRETO\%20BRASILEIRO\%20LANZA.pdf?sequence=1&isAllowed=y>>. Acesso em: 10 mar. 2018.
- LINHARES, M. V. D.; QUINTELLA, C. M. Apropriação intelectual da tecnologia (software e hardware) do controle de qualidade da cadeia industrial de mel. **Revista GEINTEC-Gestão, Inovação e Tecnologias**, v. 8, n. 1, p. 4259–4270, 2018. Disponível em: <<http://www.revistageintec.net/index.php/revista/article/view/923/851>>. Acesso em: 10 jan. 2018.
- LOFARO, D. M. The honey bee initiative 2014; smart hive. In: **International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)**. IEEE, 2017. p. 446–447. Disponível em: <<https://ieeexplore.ieee.org/document/7992772/>>. Acesso em: 20 jan. 2018.
- MALHEIROS, L. **Sistema de detecção de quedas e de posicionamento corporal com monitoramento de batimentos cardíacos**. 2016. 82 f. Monografia (Engenharia de Redes de Comunicação) - Universidade de Brasília, Brasília. Disponível em: <http://bdm.unb.br/bitstream/10483/15260/1/2015_LarinniMalheiros.pdf>. Acesso em: 15 fev. 2018.
- MARCHI, A. A.; FOSCARINI, E. D.; RIBEIRO, R. d. Q. Chasquemail: O email da ufrgs. In: **WORKSHOP DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO DAS IFES (8.: 2014: BRASÍLIA)**. [Anais..]. 2014. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/105099/000943850.pdf;sequence=1>>. Acesso em: 15 fev. 2018.
- MISHRA, G.; KAAVYA, D. Interfacing atmospheric variables to web interface using arduino. In: **International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)**. [s.n.], 2017. p. 1–6. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/8186709/>>. Acesso em: 20 abr. 2018.
- MORAES, R. A. C. **CollabDev**:: Gerenciador de repositórios para ambientes colaborativos de desenvolvimento. 2013. 60 f. Monografia (Bacharelado em Sistemas de Informação) - Centro

Universitário Eurípides de Marília, Marília, 2013. Disponível em: <http://aberto.univem.edu.br/bitstream/handle/11077/991/rodolfo_adhenawer.pdf?sequence=1&isAllowed=y>. Acesso em: 20 fev. 2018.

MORAIS, C. *et al.* Donuts:: um bot como instrumento facilitador do processo de ensino-aprendizagem na disciplina “construção de algoritmos”. **Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação**, v. 1, n. 7, 2017. ISSN 2446-7634. Disponível em: <<https://revistas.setrem.com.br/index.php/reabtic/article/view/244>>. Acesso em: 01 mar. 2018.

MUFTINISA, A. *et al.* Development and implementation of fixed asset management system. In: **Second International Conference on Informatics and Computing (ICIC)**. [s.n.], 2017. p. 1–6. Disponível em: <<https://ieeexplore.ieee.org/document/8280591/>>. Acesso em: 10 jan. 2018.

NDECKY, G. P. **Visual smart mail**: uma aplicação para visualização exploratória de email. 2017. 102 f. Dissertação (Mestrado em Engenharia Informática) — Universidade de Évora, Évora, 2017. Disponível em: <<http://hdl.handle.net/10174/21328>>. Acesso em: 20 abr. 2018.

OLIVEIRA, A. C. de; PORTEZANI, C. H. Transmissão de parâmetros ambientais através da tecnologia sem fio zigbee. **ANAIS DO ENIC**, n. 6, 2015. Disponível em: <<https://anaisonline.uems.br/index.php/enic/article/view/2533>>. Acesso em: 04 abr. 2018.

OLIVEIRA, M. A. de *et al.* Bioindicadores ambientais: insetos como um instrumento desta avaliação. **Ceres**, v. 61, n. 7, 2015. Disponível em: <http://www.scielo.br/scielo.php?pid=S0034-737X2014000700005&script=sci_abstract&tlng=es>. Acesso em: 04 abr. 2018.

PHP. **O que é o PHP?** 2017. Disponível em: <https://secure.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 01 abr. 2018.

PIRES, C. S. S. *et al.* Enfraquecimento e perda de colônias de abelhas no brasil: há casos de ccd? **Pesquisa Agropecuária Brasileira**, v. 51, n. 5, p. 422–442, 2016. Disponível em: <<http://www.scielo.br/pdf/pab/v51n5/1678-3921-pab-51-05-00422.pdf>>. Acesso em: 01 jan. 2018.

POLIZEL, A. L.; NANYA, S.; CONTE, H. Insetos dipteras como polinizadores em orchidaceae. **REVISTA UNINGÁ**, v. 46, n. 1, 2018. ISSN 2318-0579. Disponível em: <<http://revista.uninga.br/index.php/uninga/article/view/1249>>. Acesso em: 01 fev. 2018.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software**: Uma abordagem profissional. 8. ed. São Paulo: AMGH Editora Ltda, 2016. Disponível em: <<https://books.google.com.br/books?id=wexzCwAAQBAJ&printsec=frontcover&dq=PRESSMAN+Engenharia+de+Software-8\%C2\%AA+Edi\%C3\%A7\%C3\%A3o+modelo+incremental&hl=pt-PT&sa=X&ved=0ahUKEwjLq7aHgPDbAhXFj5AKHQoLB3cQ6AEIKDAA#v=onepage&q=modelo\%20incremental&f=false>>. Acesso em: 10 fev. 2018. Cap. 4.

PUCCI, D. **Integration module between nagios and telegram-cli**. 2017. Disponível em: <<https://github.com/streambinder/nagios-telegram-integration>>. Acesso em: 29 mar. 2018.

RAZAK, N. F. A. *et al.* Web based online bakery system with short messaging service and email notification. In: **ICT International Student Project Conference (ICT-ISPC)**. [s.n.], 2017. p. 1–4. Disponível em: <<https://ieeexplore.ieee.org/document/8075355/>>. Acesso em: 05 jan. 2018.

SAAD, M. H. M. *et al.* Smart pump operation monitoring and notification (puma) via telegram social messaging application. **JOIV: International Journal on Informatics Visualization**, v. 1, n. 3, p. 57–60, 2017. Disponível em: <<http://joiv.org/index.php/joiv/article/view/26>>. Acesso em: 10 jan. 2018.

SILVA, A. de Lima e. **Monitoramento não invasivo de colmeias através da IOT**. 2017. 47 f. Monografia (Tecnologia em Redes de Computadores) - Universidade Federal do Ceará, Quixadá, 2017. Disponível em: <<http://www.repositoriobib.ufc.br/00003a/00003ac0.pdf>>. Acesso em: 10 fev. 2018.

SILVA, R. F. da; MELO, F. C. L. de. Modelos híbridos de gestão de projetos como estratégia na condução de soluções em cenários dinâmicos e competitivos. **Revista Brasileira de Gestão e Desenvolvimento Regional**, v. 12, n. 3, 2016. Disponível em: <<http://www.rbgdr.net/revista/index.php/rbgdr/article/view/2532>>. Acesso em: 10 mar. 2018.

SIREGAR, I. K.; TAUFIK, F. Perancangan aplikasi sms alert berbasis web. **JIMP-Jurnal Informatika Merdeka Pasuruan E-ISSN 2503-1945**, v. 2, n. 2, 2017. Disponível em: <<http://ejurnal.unmerpas.ac.id/index.php/informatika/article/view/68/40>>. Acesso em: 06 mar. 2018.

THAMRIN, I.; ANDJARWIRAWAN, J.; NOERTJAHYANA, A. Pembuatan alert sistem komputer kabel dan nirkabel di uk. petra. **Jurnal Infra**, v. 5, n. 1, p. 271–274, 2017. Disponível em: <<https://media.neliti.com/media/publications/112016-ID-pembuatan-alert-sistem-komputer-kabel-da.pdf>>. Acesso em: 02 abr. 2018.

TOTALVOICE. **API de Comunicação por Voz e Texto**. Disponível em: <<http://www.totalvoice.com.br/>>. Acesso em: 04 abr. 2018.

VARMA, N. M. K. *et al.* Developing an underwater monitoring and alert system for water-borne disease management. In: **OCEANS 2017 - Aberdeen**. [s.n.], 2017. p. 1–4. Disponível em: <<https://ieeexplore.ieee.org/document/8084747/>>. Acesso em: 10 jan. 2018.

WU, C. Y. *et al.* A study of impact of instant messaging on job performance through employee empowerment. In: **Portland International Conference on Management of Engineering and Technology (PICMET)**. [s.n.], 2017. p. 1–10. Disponível em: <<https://ieeexplore.ieee.org/document/8125343/>>. Acesso em: 10 mar. 2018.

YOUNANG, A.; LU, L. Improving precision of java script program analysis with an extended domain of intervals. In: **IEEE 39th Annual Computer Software and Applications Conference**. [s.n.], 2015. v. 3, p. 441–446. Disponível em: <<https://ieeexplore.ieee.org/document/7273400/>>. Acesso em: 10 fev. 2018.

ZACEPINS, A. *et al.* **Beekeeping in the future — Smart apiary management**. 2016. 808-812 p. Disponível em: <<https://ieeexplore.ieee.org/document/7501207/>>. Acesso em: 10 fev. 2018.

APÊNDICE A – EXPERIMENTO A

O experimento tratou-se de verificar o correto envio das notificações cadastradas de três usuários denominados Capitão América, Mulher Gato e Batman. Ao total foram criadas 8 colmeias distribuídas conforme as Tabelas 7, 8 e 9.

A verificação dos envios iniciou-se no dia 09 de Maio de 2018, iniciando-se às 18:00:00 horas e encerrando-se as 00:00:00 do dia 10 de Maio de 2018.

Para um melhor entendimento, a Tabela 6 traz uma lista de legendas e suas descrições utilizadas no experimento.

Tabela 6 – Descrição das Legendas Utilizadas

Legenda	Descrição
A	Ativadas (Todas as grandezas serão enviadas)
P	Personalizadas (Todas as grandezas selecionadas serão enviadas)
D	Desativadas (As notificações não serão enviadas)
P0	Sem comparações de mínimos e máximos
P1	Com comparações de mínimos e máximos
C0	Sem verificação de atraso na coleta
C1	Com verificação de atraso na coleta
T	Temperatura
U	Umidade
S	Som
D	Dióxido de Carbono
O	Oxigênio
P	Peso

Fonte: a autora.

As Tabelas 7, 8 e 9 mostram como foram cadastradas as notificações de acordo com o usuário. Já as Tabelas 10, 11 e 12 apresentam as configurações das grandezas utilizadas pelos usuários.

Tabela 7 – Notificações do Capitão América

Tecnologia	Colmeia	Notificações	Intervalo de Envio	Configurações	Atraso/Intervalo
E-mail, Telegram e SMS	1	A	1 hora	P0	C0
E-mail, Telegram e SMS	2	A	1 hora	P1	C1 (30 minutos)
E-mail, Telegram e SMS	3	A	2 horas	P1	C0

Fonte: a autora.

Para o envio das coletas foi criado um *script* de envio para o banco de dados, o mesmo iniciou-se as 18:02:07 e encerrou-se as 22:57:19. O funcionamento do *script* é exemplificado nos itens abaixo:

- a) Envio de coletas com intervalo entre inserções de 5 minutos;

Tabela 8 – Notificações da Mulher Gato

Tecnologia	Colmeia	Notificações	Intervalo de Envio	Configurações	Atraso/Intervalo
E-mail, Telegram e SMS	4	A	30 minutos	P0	C1 (1 hora)
E-mail, Telegram e SMS	5	P	1 hora (Todas)	P0	C0
E-mail, Telegram e SMS	6	P	1 hora (T e U)	P1	C1 (30 minutos)

Fonte: a autora.

Tabela 9 – Notificações do Batman

Tecnologia	Colmeia	Notificações	Intervalo de Envio	Configurações	Atraso/Intervalo
E-mail, Telegram e SMS	7	P	2 horas (T e U)	P1	C0
E-mail, Telegram e SMS	8	P	30 minutos (T e U) 1 hora (S, D, O e P)	P0	C1 (1 hora)

Fonte: a autora.

Tabela 10 – Configurações do Ca-pitão América

Grandeza	Mínimo	Máximo
Temperatura	20	50
Umidade	30	35
Som	30	60
Oxigênio	36	50
Dióxido	350	500
Peso	40	80

Fonte: a autora.

Tabela 11 – Configurações da Mu-her Gato

Grandeza	Mínimo	Máximo
Temperatura	40	50
Umidade	30	35
Som	30	60
Oxigênio	36	50
Dióxido	350	500
Peso	40	80

Fonte: a autora.

Tabela 12 – Configurações do Bat-man

Grandeza	Mínimo	Máximo
Temperatura	40	50
Umidade	30	35
Som	30	60
Oxigênio	36	50
Dióxido	350	500
Peso	40	80

Fonte: a autora.

b) O *script* executa por cerca de 2 horas inserindo coletas alteradas para as 8

colmeias;

- c) Pausa por 2 horas para gerar o atraso;
- d) Após o atraso, envia uma coleta com dados alterados de cada colmeia;
- e) Finaliza enviando coletas dentro dos padrões de configurações por 1 hora para testar se as notificações não são enviadas para as pessoas que estão comparando com suas configurações.

O *script* de envio das coletas é apresentado no apêndice B.

APÊNDICE B – CÓDIGO-FONTE UTILIZADO PARA O ENVIO DAS COLETAS

Código-fonte 1 – Script de Envio de Coletas

```
1 <?php
2
3 include "/var/www/html/smartbeev2/tcc/classes/funcoes/conectaBD.php
4     ";
5 $conexao2 = $link;
6
7 mysqli_select_db($conexao2, 'coletas');
8
9 $colmeias = array(1,2,3,4,5,6,7,8);
10 $colmeias2 = array(1,2,3,4,5,6,7,8);
11
12 //Valores Fora dos Limites
13 $temperatura = array(10,60,65,55,55,69,69,68);
14 $umidade = array(55,75,75,75,75,60,70,80);
15 $som = array(25,25,65,25,25,65,67,68);
16 $oxigenio = array(55,55,65,35.9,25,25,70,80);
17 $dioxido = array(345,345,660,660,300,650,700,800);
18 $peso = array(35,35,39,85,95,100,87,88);
19
20 //Valores Dentro dos Limites
21 $temperatura2 = array(25,25,25,45,46,46,47,48);
22 $umidade2 = array(33,33,33,35,35,33,33,33);
23 $som2 = array(35,35,35,40,36,60,37,38);
24 $oxigenio2 = array(40,40,40,38,38,36,37,38);
25 $dioxido2 = array(358,400,400,351,350,356,357,358);
26 $peso2 = array(45,45,45,46,48,46,47,48);
27
28 $cont = 0;
29 $cont2 = 0;
30 $cont_i = 0;
```

```
31 $cont3 = 0;
32
33 //24 coletas em 1 hora e 55 minutos 24x8 colmeias = 192 coletas
34 while($cont_i<192){
35     for ($j=0; $j < count($colmeias); $j++) {
36         $sql ="INSERT INTO coletas(amostra, idColmeia, datahora,
37             temperatura, umidade, som, oxigenio, dioxido, peso)
38             VALUES (default,$colmeias[$j],NOW(),$temperatura[$j]
39                 , $umidade[$j], $som[$j], $oxigenio[$j], $dioxido[
40                 $j], $peso[$j])";
41         $resultado = mysqli_query($conexao2, $sql);
42         echo "Teste 1: Coletas Alteradas ".$j."\n";
43         $cont_i++;
44         echo "Total de Coletas Inseridas: ".$cont_i."\n";
45     }
46
47     if ($cont_i == 192) {
48         //2 horas de atraso
49         sleep(7200);
50         //Insere dados alterados mais uma vez para cada colmeia
51         for ($k=0; $k < count($colmeias); $k++) {
52             $sql ="INSERT INTO coletas(amostra, idColmeia, datahora
53                 , temperatura, umidade, som, oxigenio, dioxido,
54                 peso)
55                 VALUES (default,$colmeias[$k],NOW(),$temperatura
56                 [$k], $umidade[$k], $som[$k], $oxigenio[$k],
57                 $dioxido[$k], $peso[$k])";
58             $resultado = mysqli_query($conexao2, $sql);
59             echo "Teste 2: Coleta Alterada Depois do Atraso: ".$k."
60                 \n";
61             $cont2++;
62             $cont_i++;
63             echo "Total de Coletas Inseridas: ".$cont_i."\n";
64         }
65     }
```

```
57 //cinco minutos (Intervalo)
58 sleep(300);
59
60 if ($cont2 == 8) {
61     while ($cont3 < 96) {
62         for ($i=0; $i < count($colmeias2); $i++) {
63
64             $sql ="INSERT INTO coletas(amostra, idColmeia,
65                 datahora, temperatura, umidade, som,
66                 oxigenio, dióxido, peso) VALUES (default,
67                 $colmeias2[$i],NOW(),$temperatura2[$i],
68                 $umidade2[$i],$som2[$i],$oxigenio2[$i],
69                 $dióxido2[$i],$peso2[$i])";
70
71             $resultado = mysqli_query($conexao2,$sql);
72             echo "Teste 3: Coletas Dentro dos Limites: ".$i
73                 ."\n";
74             $cont3++;
75             $cont_i++;
76             echo "Total de Coletas Inseridas: ".$cont_i."\n
77                 ";
78         }
79         sleep(300);
80     }
81 }else{
82     echo "Processando!". "\n";
83 }
84 }else{
85     echo "Processando!". "\n";
86 }
87 sleep(300);
88 }
```