

# BeeNotified! A Notification System of Physical Quantities for Beehives Remote Monitoring

BeeNotified! Um Sistema de Notificações de Grandezas Físicas para Monitoramento Remoto de Colmeias de Abelhas

Antonio Rafael Braga<sup>1\*</sup>, Juliana de Castro Rabelo<sup>1</sup>, Arthur de Castro Callado<sup>1</sup>, Atslands Rego da Rocha<sup>2</sup>, Breno M. Freitas<sup>3</sup>, Danielo G. Gomes<sup>2</sup>

**Resumo:** One of the ways to reduce inappropriate management of hives and monitor bee health is to send notifications/alerts about the data collected through sensors. This study presents BeeNotified!, 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 data gathering. From this data, researchers and beekeepers can be informed and make changes in the locations of the hives, avoiding catastrophes and possible diseases. The results obtained with 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.

**Resumo:** Uma das maneiras de reduzir o gerenciamento inadequado de colmeias e monitorar a saúde das abelhas é enviar notificações/alertas sobre os dados coletados por meio de sensores. O presente estudo apresenta o BeeNotified!, uma solução para envio de notificações através de Telegram, E-mail e SMS. As notificações alertam sobre o nível de temperatura, umidade, som, dióxido de carbono, oxigênio, peso da colmeia e atraso na coleta de dados. A partir desses dados, pesquisadores e apicultores podem ser informados e fazer mudanças nos locais das colmeias, evitando catástrofes e possíveis doenças. Os resultados obtidos com o tempo de processamento no envio de mensagens mostraram que as mensagens enviadas via SMS e Telegram têm um tempo de processamento menor em relação ao envio via e-mail. Em relação ao envio de notificações de acordo com as preferências do usuário, todas as notificações foram enviadas corretamente.

**Palavras-Chave:** Notificações — Apicultura — Monitoramento ambiental

<sup>1</sup> *Redes de Computadores, Campus Quixadá, Universidade Federal do Ceará, Quixadá, Ceará, Brasil.*

<sup>2</sup> *GREat, Departamento de Engenharia de Teleinformática, Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, Ceará, Brasil.*

<sup>3</sup> *Setor de Abelhas, Departamento de Zootecnia, Centro de Ciências Agrárias, Universidade Federal do Ceará, Fortaleza, Ceará, Brasil.*

\* **Corresponding author:** rafaelbraga@ufc.br

**DOI:** <https://doi.org/10.22456/2175-2745.90724> • **Received:** 06/03/2019 • **Accepted:** 17/05/2020

CC BY-NC-ND 4.0 - This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

## 1. Introdução

A apicultura é um dos ramos tradicionais da pecuária, 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 agrícolas e na sobrevivência de plantas silvestres [1]. As abelhas podem polinizar cerca de 85% das espécies vegetais de maneira geral, o que as torna um dos insetos mais importantes para a agricultura [2]. A apicultura pode ser vista também como uma fonte secundária de renda, principalmente para as pequenas famílias rurais, que são as responsáveis pela maior parte da produção nacional de mel [3].

As colônias de abelhas podem sofrer uma série de problemas que levam a (i) rápida perda de abelhas operárias, (ii) ausência de crias, (iii) morte de abelhas adultas e (iv) invasão da colmeia por pragas [4]. Esse distúrbio tem como principais fatores os estresses causados por patógenos, manejo inadequado das colônias, uso de agrotóxicos, má nutrição e os fatores combinados [4]. Dentro desse cenário, há um grande interesse nas vidas e no comportamento das abelhas, e a saúde das mesmas se constitui como uma crescente preocupação entre pesquisadores, ecologistas, agricultores e ainda tomadores de decisão [1].

Nesse sentido, tem-se observado um interesse cada vez maior entre os apicultores em usar sensores em colmeias, o

que pode ser explicado pelo progresso da Internet das Coisas, em inglês, *Internet of Things* (IoT). Seja na cidade ou no campo, é possível criar a chamada apicultura de precisão [5, 6] através da IoT e da Tecnologia da Informação como um todo [7]. Entre outras magnitudes sensoreáveis que podem ser monitoradas, se destacam: temperatura, umidade, concentração de dióxido de carbono ( $CO_2$ ) e oxigênio nas colmeias, massa da colmeia, padrões de imagem e emissão de intensidade sonora [8].

Uma vez coletadas, essas magnitudes devem ser analisadas e/ou informadas para o apicultor. Idealmente, uma análise prévia automática das grandezas deve ser feita para posterior envio de informações mais relevantes para o apicultor. Uma das maneiras de informar ao apicultor sobre as grandezas coletadas é por meio do envio de notificações/alertas [9, 10, 11]. Essas notificações podem ser extremamente úteis para indicar qual ou quais itens devem receber mais atenção pelo apicultor durante os manejos (ou inspeções) preventivos ou corretivos [12].

Dentro desse contexto, o presente estudo buscou resolver o seguinte problema de pesquisa: como informar aos apicultores, pesquisadores e gestores sobre a saúde de uma colônia de abelhas sem a necessidade de atenção constante a um sistema de monitoramento *web*? Portanto, o presente estudo, criou um sistema de notificações denominado *BeeNotified!* acoplado ao sistema *Sm@rtBee*<sup>1</sup>, que realiza o monitoramento de colmeias. O *BeeNotified!* faz o envio de notificações pré-configuradas pelos usuários. As tecnologias de envio são: e-mail, Telegram e SMS (*Short Message Service*).

## 2. Fundamentação Teórica

Nessa sessão são discutidas os conceitos, metodologias e as tecnologias utilizadas para a implementação do sistema *BeeNotified!*.

### 2.1 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 [13].

Dentro desse contexto, o monitoramento ambiental se faz necessário, seja ele manual, ou 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 [14]. 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 [15] 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 [14] ou a verificação da qualidade da água, através do uso de sensores para a medição de índices como pH, temperatura, turbidez e condutividade elétrica [16].

#### 2.1.1 Monitoramento de Colmeias

As abelhas, formigas e vespas, espécies da Ordem Hymenoptera, estão recendo atenção especial nos últimos anos, pois podem ser vistos como bioindicadores ambientais [13].

As vespas e abelhas solitárias, 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.

As abelhas sem ferrão do gênero *Melipona* 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 [13].

As abelhas melíferas (*Apis mellifera*) são uma parte importante do nosso ecossistema. Como agente polinizador, ajudam pelo menos 30% das lavouras do mundo e até 90% de todas as plantas selvagens a prosperar através da polinização cruzada [17]. A polinização realizada pelos insetos pode ocorrer quando estes visitam as flores de uma mesma espécie vegetal [18].

Estudos mostram que entre os anos de 2015 e 2016 durante o inverno os apicultores dos EUA perderam 44% das colônias de abelhas, o que representa um aumento de 10% em relação à década anterior [17]. As causas dessa queda estão relacionadas à saúde geral da colônia. Quanto mais saudável é a colônia, maior a probabilidade dela sobreviver ao inverno.

As inspeções manuais, a principal técnica de inspeção usada atualmente, 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 colônia não está saudável. Essas inspeções são invasivas para as colônias de abelhas [17]. Para reduzir essas perdas, estudos como de [17] propuseram 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 manter suas colmeias saudáveis. Outro estudo que realizou o monitoramento de colmeias foi Dutra [15], que criou um sistema denominado *Beehiveior*, com o objetivo de auxiliar pesquisas na área apícola. Os sistemas *Smart Hive* e *Beehiveior* serão melhor detalhados na Seção 3.

<sup>1</sup><http://www.smartbee.great.ufc.br/>.

## 2.2 Tecnologias de Rede para Enviar Notificações

Um sistema de notificação é responsável pelo envio de mensagens pré-configuradas aos usuários quando ocorrem eventos importantes ou quando ocorre um evento que requer atenção especial do usuário do sistema [19]. As tecnologias de rede que realizam o monitoramento e envio de notificações são diversas.

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 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	Pago
Telegram	Gratuito

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.

### 2.2.1 E-mail

O e-mail ou correio eletrônico é um método que permite criar, enviar e receber mensagens através de sistemas eletrônicos de comunicação [20]. O serviço de e-mail tem sua origem na ARPANET e sua primeira transmissão inter-institucional foi realizada no início da década de 70 [21].

Segundo Gomes [20] 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 para os sistemas conhecidos como *intranets*, que permitem a troca de mensagens dentro de uma organização.

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 [22]. A tecnologia de e-mail pode ser acessada e usada em qualquer lugar através de telefones celulares e outros dispositivos de computador [23] 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 [24].

### 2.2.2 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 [20].

O conceito de *Short Message Service* (SMS) foi desenvolvido pelo engenheiro finlandês Matti Makkonen, em meados da década de 1980 [25], 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” [20]. O Sistema Global de Comunicações Móveis (GSM) padronizou o SMS em 1985, porém o mesmo agora é mantido pelo 3GPP como TS 23.040 [26]. 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) [25].

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 [25].

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 [26]. 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 [26].

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 [27].

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 [27]. 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. [28].

### 2.2.3 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* [29]. 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 [30].

Um dos sistemas operacionais móveis mais populares atualmente é o Android. De acordo com loja de aplicativos Android ou Google Play, centenas de aplicativos de mensagens instantâneas estão disponíveis para download [29]. 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 [29].

O Telegram é um *software* popular entre os usuários de *smartphones*, possui uma API pública [31], 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 [32].

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) [32]. Além de poder enviar e receber mensagens individuais, segundo [32] o Telegram pode criar grupos ou canais com até 5000 mil pessoas.

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 [32]. Permite ainda a busca de pessoas pelo nome de usuário, e ainda a criação de bots [31].

Os bots são aplicativos de terceiros que funcionam dentro do Telegram [33] e usuários podem interagir com bots enviando e recebendo mensagens [31]. A API pública do Telegram é apropriada para o desenvolvimento de bots. 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 [31] e sistemas que usam bots para o envio de alertas, como no presente estudo.

No estudo de Moraes [31], 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

Em Kvisis e Zacepins (2015) [2], os autores descrevem abordagens para o monitoramento de temperatura no local e remoto, assim os dados de temperatura de colônias podem ser transferidos para o apicultor em tempo real. O objetivo do trabalho é distinguir 6 diferentes arquiteturas de sistema para monitoramento em tempo real da temperatura das colônias de abelhas. Em 5 das 6 arquiteturas apresentadas, os autores incluem o envio de notificações para o usuário final, exceto na arquitetura onde a temperatura é armazenada apenas localmente esse envio não é proposto.

As notificações são enviadas por mensagens de SMS ou e-mail e são geradas pelo aplicativo de notificação de erro embarcado em uma Raspberry Pi. Embora Kvisis e Zacepins (2015) tenham proposto 5 arquitetura diferentes que notificam o apicultor em caso de anormalidade das condições das colmeias (por exemplo: enxameamento ou morte de colônias), os autores indicam a necessidade de um sistema de apoio à decisão para gerar essas mensagens.

Em Balta et. al. (2015) [11] é apresentado um modelo de software para monitoramento remoto de colmeias. Para resolver a distância de comunicação entre colmeias e apicultores, a arquitetura do software proposta garante que o envio de dados pode ser feito para a Nuvem através de sinais de rádio frequência, fibra óptica ou até mesmo via sinais de satélite. A arquitetura proposta permite o monitoramento de temperatura, umidade e do número de abelhas entrando e saindo da colmeia.

As informações medidas são armazenadas em um servidor para posteriormente alertar os apicultores através de mensagens SMS sobre situações que poderiam ser perigosas para as colmeias. Os autores informam que o mecanismo de notificação pode ser fornecido através de mensagens MQTT para o aplicativo móvel ou por meio de APIs externas. Os autores planejam desenvolver alertas via Twitter, Whatsapp, E-mail e SMS.

No trabalho de Murphy et. al. (2015) [34], os autores apresentam o projeto de um sistema de monitoramento de

colônias totalmente autônomo, que realiza o monitoramento de imagem e som internos da colmeia, bem como um sistema de alerta para eventos de emergência, tais como aumento abrupto da atividade das abelhas ou danos físicos à colmeia. Na solução apresentada, quando os níveis de som dentro da colmeia aumentam substancialmente, o apicultor é alertado por SMS.

Uma série de testes foi realizada na função de alerta de texto para testar o tempo que o apicultor levou para receber um alerta de texto após um evento. O tempo médio entre o evento anômalo e a notificação no celular do apicultor foi de aproximadamente 7,56 segundos. Isso foi considerado apropriado para os autores, pois sem esses alertas os apicultores não seriam capazes de perceber os problemas dentro de uma colmeia até que visitasse ou inspecionasse a mesma, o que poderia levar vários dias.

Com o objetivo de auxiliar as pesquisas da área apícola da Unidade de Extração de Produtos Apícolas, da Escola Agrícola de Jundiá (UEPA-EAJ) e facilitar as atividades no manejo das colmeias, o trabalho de Dutra (2016) [15] desenvolveu uma ferramenta de monitoramento e controle de colmeias de um apiário chamada *Beehiveior*. Esse sistema realiza coleta das variáveis temperatura, umidade, ruído, sensor de presença e contagem de abelhas (infravermelho) das colmeias e exibe-os em um sistema *web*. O sistema de monitoramento foi construído utilizando a placa Arduino modelo UNO acoplada a uma *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 por uma Central de Automação e Controle e um Aplicativo *web*. A central 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. O aplicativo *web* é responsável por realizar o processamento, armazenamento e exibição dos dados enviados pela central de automação e controle. O aplicativo *web* 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 [15].

O estudo de Linhares e Quintela (2018) [35] apresenta um conjunto de tecnologias denominado *Bloco Inteligente do Processo de Produção* (BIPP), o mesmo é formado por um *hardware* e quatro *softwares* embarcados. Os dados de análises de cor, umidade do mel, umidade relativa do ar, temperatura ambiente e de análises laboratoriais de componentes físico-químico-microbiológicos são analisados para garantir a qualidade na produção do mel e monitorar os fatores de risco como fermentação e o aumento de HMF (Hidroximetilfurfural) [35]. 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.

Para as medições de umidade e temperatura, os dados são coletados e documentados automaticamente por meio sensores instalados. 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 [35]. Apesar de mostrar as etapas funcionais do *software*, incluindo as notificações aos produtores como saída, o estudo não descreveu a tecnologia de envio da notificação.

A Tabela 2 apresenta uma comparação entre o presente estudo com os trabalhos relacionados a apicultura que foram discutidos.

## 4. O Sistema BeeNotified!

O Sistema BeeNotified! foi desenvolvido como uma extensão do sistema web do Projeto Sm@rtBee. O Sm@rtBee é composto por 6 elementos principais, a saber: *sensores*, *gateway*, *middleware*, *webservice* e as aplicações *móvel e web*.

Na arquitetura do Sm@rtBee, em cada colmeia há uma placa Arduino Nano que possui módulos *sensores* conectados ao Arduino que coletam grandezas como: temperatura, umidade, peso e ruído [36]. Porém, serão implementados posteriormente sensores para medição de dióxido de carbono e oxigênio. 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 [36]. O Arduino realiza o processamento dos dados coletados 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.

A função do *gateway* é receber os dados coletados pelos nós sensores e enviar diretamente ao *middleware*. O gateway possui também uma Arduino onde uma antena de rádio frequência receptora fica instalada. Esse Arduino processa os dados recebidos dos sensores e os escreve na porta serial de um *Beaglebone Black* (BBB). 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. Para envio da mensagem HTTP, um *script* que executa do BBB configura uma conexão com a Internet através de 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

Tabela 2. Trabalhos Relacionados

Estudo	Tipo de Notificação	Grandeza Monitorada
Kviesis e Zacepins (2015) [2]	SMS ou e-mail	Temperatura
Balta et. al. (2015) [11]	Não implementada	Temperatura, umidade e número de abelhas
Murphy et. al. (2015) [34]	SMS	Ruído e imagem
Dutra (2016) [15]	Não implementada	Temperatura, umidade, ruído, presença e número de abelhas
Linhares e Quintela (2018) [35]	Não descrita	Temperatura e umidade
Presente Estudo	E-mail, SMS e Telegram	Temperatura, umidade, ruído, $O_2$ , $CO_2$ e peso

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.

O *middleware* Fiware<sup>2</sup> é o ambiente de processamento e armazenamento de dados, criado por [36], onde foi utilizado os GEs Orion Context Broker e o Cosmos GE. O *Orion Context Broker* gerencia o *Data/Context*, e consiste em um *broker* para geração de contexto que utiliza o padrão *Publish/Subscribe*. O *broker* é responsável por receber e gerenciar os dados de contexto das aplicações que utilizam o Fiware. No ambiente estudado, os dados de contexto são os dados coletados pelo sensores e as aplicações que utilizam o Fiware são os próprios sensores, os aplicativos e Cosmos. Este último contém um ecossistema de ferramentas voltadas para análise e armazenamento de dados. Para um melhor entendimento do funcionamento do *middleware* é recomendável a leitura do estudo de Silva [36].

De acordo com Silva [36] o *webservice* utiliza um banco de dados para realizar consultas SQL (*Structured Query Language*) e disponibilizar através de uma API REST (*Representational State Transfer*) ou Transferência de Estado Representacional, essas informações para fácil acesso pelas aplicações web e móvel. A aplicação móvel realiza a comunicação com o *webservice* trocando mensagens JSON através de protocolo HTTP.

A aplicação *móvel* foi desenvolvida 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* [36]. Atualmente o aplicativo está na versão 5.0 e disponível na *Play Store*<sup>3</sup>.

A aplicação *web* do Sm@rtbee foi desenvolvida utilizando o *framework* Laravel<sup>4</sup> versão 5.6.23. Sendo assim, para o desenvolvimento do *front-end* do BeeNotified! foi utilizado o *framework* Laravel na mesma versão, além da linguagem JavaScript<sup>5</sup>. Para o *back-end*, ou seja, para os *scripts* de conexão com o banco de dados e o envio de notificações em si, foi utilizada a linguagem PHP versão 7.0<sup>6</sup>. A base de

dados utilizada foi o MySQL<sup>7</sup>. Para a execução do sistema, foi utilizado o Servidor Apache<sup>8</sup>.

A Figura 1 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.

Na fase de desenvolvimento do sistema BeeNotified!, foi criado o protótipo do sistema de notificação para o envio de notificações/alertas aos usuários solicitantes. O desenvolvimento foi realizado seguindo-se as seguintes etapas: a) análise dos requisitos, b) desenvolvimento e c) testes e validação do sistema. Na fase de análise de requisitos foram levantados os requisitos do sistema. Os requisitos foram definidos a partir de conversas realizadas com um dos professores responsáveis pelo projeto 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 *Telegram*.

Na fase de testes e validação do sistema, 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 com o qual foram realizados experimentos para avaliar o tempo de processamento no envio das mensagens.

## 4.1 Funcionalidades

O sistema foi desenvolvido com o propósito de disponibilizar ao usuário formas de envios de notificações diferentes. Possui como funcionalidades a configuração de grandezas, o cadastro de notificações via E-mail, Telegram e SMS, listagem das notificações cadastradas e enviadas e opções de edição e exclusão das notificações cadastradas tanto para usuários comuns como para usuários do tipo administrador. A seguir são descritas as funcionalidades.

### 4.1.1 Configuração de Grandezas

Por padrão, após o cadastro de um novo usuário no sistema um *trigger* é acionado no banco de dados e é criado um novo registro de configurações com valores padrões do sistema relacionados ao usuário. Na tela de listagem existe a opção

<sup>2</sup><https://www.fiware.org/>

<sup>3</sup><https://play.google.com/store/apps/details?id=com.ufc.smartbee>

<sup>4</sup><https://laravel.com/>

<sup>5</sup><https://www.javascript.com/>

<sup>6</sup><http://php.net/>

<sup>7</sup><https://www.mysql.com/>

<sup>8</sup><https://www.apache.org/>

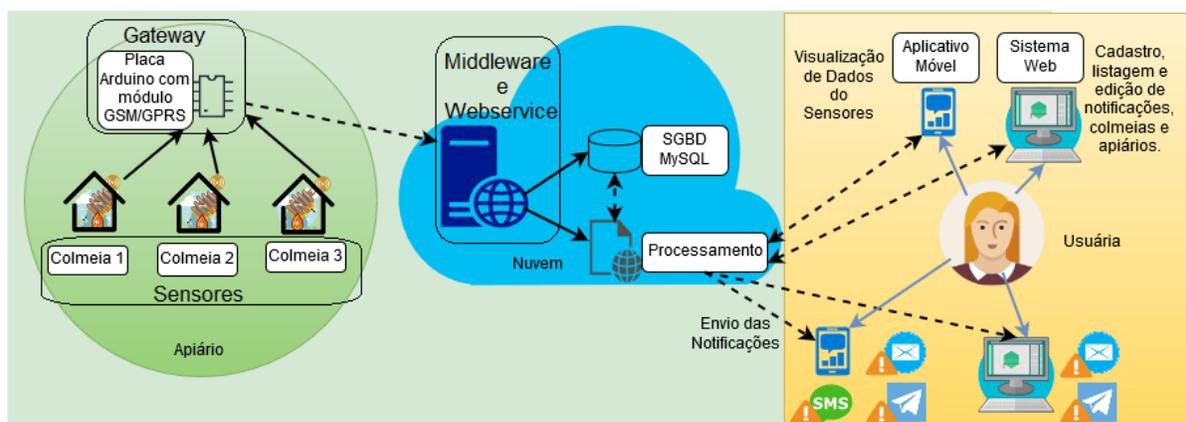


Figura 1. Arquitetura dos Sistemas Sm@rtBee e NotefyMe!

de edição das configurações. No formulário de edição, as variáveis utilizadas são: temperatura, umidade, som, dióxido de carbono, oxigênio e peso. O usuário deve escolher os valores mínimos e máximos para cada variável de acordo com suas preferências e abelhas estudadas. A Figura 2 apresenta o formulário de edição dos valores mínimos e máximos para cada variável.

Edição de configurações no sistema Sm@rtBee. O formulário mostra uma tabela de configurações de padrões e dois passos de configuração para a faixa de temperatura e umidade.

GRANDEZAS	VALORES MÍNIMOS	VALORES MÁXIMOS
Temperatura	33 °C	36 °C
Umidade	42 %	47.5 %
Som	50 dB	60 dB
Oxigênio	33 %	50 %
Dióxido de Carbono	350 ppm	500 ppm
Peso	29 Kg	39 Kg

Passo 1: Configure a Faixa de Temperatura  
 Temperatura Mínima em °C: 33  
 Temperatura Máxima em °C: 36

Passo 2: Configure a Faixa de Umidade  
 Umidade Mínima em %: 42  
 Umidade Máxima em %: 47,2

Figura 2. Edição dos limiares das grandezas

#### 4.1.2 Cadastros de Novas Notificações

O cadastro de uma nova notificação (Figura 3) via E-mail ou SMS possui um formulário onde o usuário, no passo 1, escolhe a colmeia para a qual ele deseja que sejam enviada uma notificação. 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 realiza as comparações com as configurações cadastradas.

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 para não sobrecarregar o usuário, assim, ao atingir 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, e assim escolher sobre quais variáveis serão enviadas as notificações e o intervalo de envio personalizado de cada variável. O intervalo de envio da notificação escolhido pode ser de 30 minutos, 1, 2, 4, 8, 16 ou 24 horas. A Figura 3 apresenta o formulário de cadastro de uma nova notificação via E-mail.

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 à escolha da colmeia, o passo 2 ao nome dado à notificação. 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 colocá-lo, se não possuir existe um modal explicando como criar um *username*. O passo 4 informa que o usuário deve enviar qualquer palavra para o bot denominado *bot\_smartbee\_server*. O passo 5 informa apenas que o usuário deve clicar no botão confirmar para verificar se o cadastro do *username* obteve sucesso ou não. Os demais campos são semelhantes aos campos 3, 4 e 5 dos cadastros de E-mail e SMS referentes à checagem de configuração, à verificação de atraso e à escolha do estado da notificação.

Figura 3. Cadastro de Nova Notificação Via E-mail

#### 4.1.3 Listagem, Edição e Exclusão das Notificações

Na listagem das notificações cadastradas (Figura 4), 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 de listagem. A listagem das notificações é semelhante para todas as tecnologias estudadas.

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

Figura 4. Listagem de Notificações Cadastradas

#### 4.1.4 Listagem das Notificações Enviadas

A listagem das notificações enviadas via E-mail e Telegram segue um mesmo padrão. Para cada mensagem enviada são mostrados o ID da mensagem, o nome da notificação, o nome da colmeia e a data e o 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. Ainda mostra o valor

total gasto até o momento com o envio de notificações via SMS (que é pago).

#### 4.1.5 Cadastro, listagens, edições e exclusões do Administrador

O usuário administrador pode visualizar todas as notificações e cadastrar novas notificações de colmeias de outros usuários, mas só pode editar e excluir as notificações pertencentes a ele (administrador). Para a realização dessas atividades, foram realizadas checagens individuais em cada tela. A primeira refere-se ao cadastro de novas notificações; a segunda, à 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 à 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. 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 ela não pertencer a ele, uma mensagem é retornada avisando que a notificação não pertence a ele.

## 5. Avaliação do sistema

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. O *script* de envio de coletas foi criado na linguagem PHP simulando o envio de coletas de 8 colmeias.

Apesar do sistema BeeNotified! estar totalmente integrado ao sistema *web* do sistema Sm@rtbee, os experimentos do estudo foram realizados de forma simulada para que os diversos eventos possíveis fossem facilmente experimentados, ou seja, os dados enviados para o banco de dados MySQL foram gerados através de um *script* de dados controlados a fim de verificar o correto funcionamento do sistema. 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*. 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.

O *script* foi executado uma vez e iniciado às 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. Às 19:57:17 iniciou-se uma pausa de duas horas para gerar atrasos nas coletas. Às 21:57:17 foi reiniciada a inserção de coletas, sendo que a primeira coleta é alterada e as demais foram dentro dos padrões. O *script* encerrou-se às 22:57:19. A Tabela 3 mostra a distribuição dos períodos observados. 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 ao fato do *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).

**Tabela 3.** Distribuição dos Períodos Observados

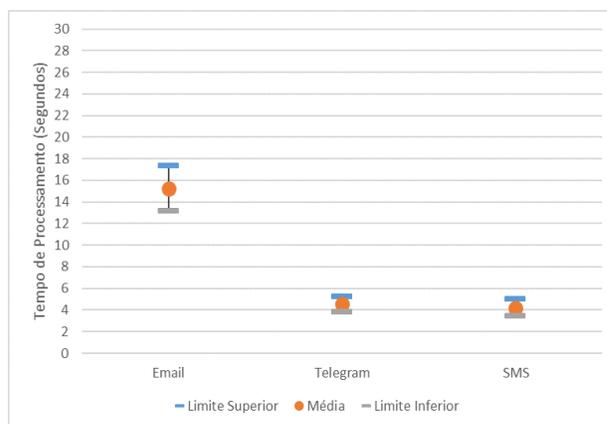
Período	Quantidade de Notificações	Variáveis
18:00:00	8	Ausentes
18:30:00	2	Alteradas
19:00:00	7	Alteradas
19:30:00	2	Alteradas
20:00:00	9	Alteradas
20:30:00	4	Em atraso <sup>9</sup>
21:00:00	6	Em atraso <sup>9</sup>
21:30:00	2	Em atraso <sup>9</sup>
22:00:00	9	Alteradas
22:30:00	2	Normais <sup>10</sup>
23:00:00	5	Normais <sup>10</sup>
23:30:00	4	Em atraso <sup>9</sup>
00:00:00	6	Em atraso <sup>9</sup>

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. 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 5 mostra na vertical 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

<sup>9</sup>Em algumas notificações estudadas a coleta não estaria em atraso devido ao intervalo de checagem de atraso de cada usuário ser diferente.

<sup>10</sup>Apesar das variáveis estarem dentro dos padrões dos usuários, algumas notificações foram cadastradas para enviar os dados independente das configurações.



**Figura 5.** Tempo de Processamento de Todos os Períodos

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.

Em relação o formato das mensagens enviadas, para as tecnologias 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 6 mostra um exemplo de notificação recebida pelo usuário Batman via E-mail. A Figura 7 mostra um tipo de mensagem recebida pelo Capitão América via SMS e a Figura 8 mostra uma notificação recebida pelo Capitão América via Telegram.

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

#### Dados

Coleta de Temperatura Ausente  
 Coleta de Umidade Ausente  
 ⚠ Coleta Não Existe  
 ✅ Envio da Notificação: 09-05-2018 18:00:13

**Figura 6.** Mensagem via E-mail do usuário Batman

## 6. Discussão

De acordo com os resultados obtidos pode-se concluir que o envio de mensagens por 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 (no caso, o 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



Figura 7. Mensagem via SMS do Capitão América

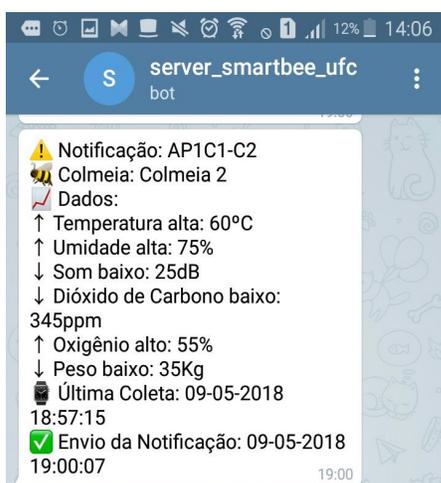


Figura 8. Mensagem via Telegram do Capitão América

mensagem é encaminhada para a API da TotalVoice<sup>11</sup> (API usada para o envio de SMS) juntamente com o *AccessToken* de acesso via URL.

Apesar dos tempos de processamento obtidos serem reduzidos, existe um tempo adicional na entrega das mensagens, visto que a mensagem é enviada aos servidores do Gmail, do Telegram ou da TotalVoice e, somente após isso, a mensagem é enviada para o cliente (apicultor). No caso das mensagens SMS, o tempo de envio pode ser verificado acessando a conta criada para este fim no site da TotalVoice.

O envio via SMS foi o que apresentou menor tempo médio de processamento, 4,19 segundos. Esse tempo foi menor do que o obtido em Murphy et. al. (2015) [34]. Vale destacar que essa diferença pode estar ligada a: (i) às implementações realizadas, às APIs de envio de SMS's ou à rede telefônica de entrega. Vale destacar também que Murphy et. al. (2015) [34] mediu o tempo desde a coleta de dados na colmeia até a chegada da mensagem no apicultor.

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. Contudo, foi possível verificar que, para todas as mensagens enviadas e para todas as tecnologias estudadas, o recebimento das mensagens o atraso foi inferior a 60 segundos.

Apesar do sistema de notificações não ter sido testado com sensores enviando os dados coletados ao banco de dados, pode-se concluir que os objetivos foram alcançados. Pois o sistema apresentou os comportamentos esperados para todos os casos de testes propostos (medições ausentes, medidas com alteração na faixa normal de valores, medidas em atraso e medidas na faixa normal de valores). Se comparado com os trabalhos relacionados, os casos de testes aqui propostos foram muito mais amplos. Por exemplo, em [34] os autores apenas avaliaram a chegada de notificações de medidas que se apresentavam fora da faixa normal de valores. Em Kviesis e Zacepins (2015) [2] os autores não fizeram avaliação do envio das notificações. Em relação a tecnologias de envio, este trabalho apresentou três tipos de tecnologias para o envio de notificações, e-mail, telegram e SMS. Ao passo que os trabalhos relacionados apresentados apenas em Kviesis e Zacepins (2015) [2] foi apresentado mais de uma opção de tecnologia, no caso SMS ou e-mail. Contudo, a opção de envio através de mensagem instantânea é de suma importância pra garantir mais facilidade de acesso e rápida notificação.

## 7. Conclusões

Este estudo apresentou um sistema para notificações de grandezas físicas monitoradas de colônias de abelhas. De acordo com os resultados apresentados, o tempo de processamento no envio das mensagens foi considerado aceitável. Todas as mensagens enviadas foram recebidas de acordo com as notificações cadastradas pelos usuários.

As tecnologias de envio via E-mail e Telegram não implicam em custos, e é enviada uma única mensagem por notificação. Via SMS, como existe um limite de 160 caracteres por mensagem, podem ocorrer envios de duas mensagens para uma única notificação, 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 à internet, poderão receber as notificações sem problemas caso possuam cobertura do sinal celular.

Em relação às perspectivas de trabalhos futuros, destaca-se a realização de novos experimentos utilizando toda a arquitetura do sistema Sm@rtbee, inclusive verificando-se o atraso total no envio. A implementação de outros meios de entrega de mensagens como o WhatsApp e Messenger<sup>12</sup>.

<sup>11</sup><http://www.totalvoice.com.br/>

<sup>12</sup><https://developers.facebook.com/docs/messenger-platform/reference/send-api/>

## Contribuições dos autores

Todos os autores (A.R.B., J.C.R., A.C.C., A.R.R., B.M.F e D.G.G.) conceberam e conceituaram o protocolo do estudo. A.C.C. e A.R.R. conceberam e desenvolveram a metodologia; A.R.B. e J.C.R. trabalharam no desenvolvimento do software; A.C.C. e A.R.R. fizeram a análise formal; A.R.B., J.C.R. e A.R.R. escreveram o primeiro rascunho do manuscrito; A.R.B., A.C.C., B.M.F e D.G.G. revisaram o manuscrito e produziram a versão final; B.M.F e D.G.G. trabalharam na supervisão deste artigo; Todos os autores concordaram com a versão publicada do manuscrito.

## Agradecimentos

Esse estudo foi financiado em parte pela CAPES - código de financiamento 001. Danielo G. Gomes e Breno M. Freitas agradecem o suporte financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) [processos #302934/2010-3, #432585/2016-8, #129426/2018-0 e #310317/2019-3].

## Referências

- [1] 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.
- [2] SYSTEM Architectures for Real-time Bee Colony Temperature Monitoring. *Procedia Computer Science*, v. 43, n. Supplement C, p. 86 – 94, 2015. ICTE in Regional Development, December 2014, Valmiera, Latvia.
- [3] 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. Disponível em: [http://www.sje.ifmg.edu.br/portal/imagens/artigos/biblioteca/TCCs/Sistemas\\_de\\_informacao/2013/ALBERT\\_WESLEY\\_SILVA\\_DE\\_ALMEIDA\\_THALLYS\\_FREED\\_DOS\\_SANTOS\\_SILVA\\_THIAGO\\_MOREIRA\\_SILVA.pdf](http://www.sje.ifmg.edu.br/portal/imagens/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.
- [4] 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.
- [5] ZACEPINS, A. et al. Challenges in the development of precision beekeeping. *Biosystems Engineering*, v. 130, p. 60 – 71, 2015.
- [6] ZACEPINS, A. et al. Development of internet of things concept for precision beekeeping. In: *2017 18th International Carpathian Control Conference (ICCC)*. [S.l.: s.n.], 2017. p. 23–27.
- [7] BRAGA, A. R. et al. Gerenciamento térmico e elétrico de um centro de dados utilizando sensoriamento iot. In: *Anais do IX Simposio Brasileiro de Computação Ubíqua e Pervasiva*. Porto Alegre, RS, Brasil: SBC, 2017. Disponível em: <https://portaldeconteudo.sbc.org.br/index.php/sbcup/article/view/3306>.
- [8] MEIKLE, W. G.; HOLST, N. Application of continuous monitoring of honeybee colonies. *Apidologie*, v. 46, n. 1, p. 10–22, Jan 2015.
- [9] KVIESIS, A.; ZACEPINS, A. System architectures for real-time bee colony temperature monitoring. *Procedia Computer Science*, Elsevier, v. 43, p. 86–94, 2015.
- [10] DINEVA, K.; ATANASOVA, T. *COMPUTER SYSTEMS USING INTERNET OF THINGS FOR MONITORING OF BEE HIVES*. 2017. 169-176 p. Disponível em: <https://search.proquest.com/docview/2185586788?accountid=26598>.
- [11] BALTA, A. et al. Software modeling of remote controlled beehive design. In: *International Conference on Advances and Innovations in Engineering ICAIE*. [S.l.: s.n.], 2017. p. 1133–1138.
- [12] A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Computers and Electronics in Agriculture*, v. 169, p. 105161, 2020.
- [13] OLIVEIRA, M. A. de et al. Bioindicadores ambientais: insetos como um instrumento desta avaliação. *Ceres*, v. 61, n. 7, 2015.
- [14] 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.
- [15] 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](https://repositorio.ufrn.br/jspui/bitstream/123456789/21775/1/ThiagoFernandesSilvaDutra_DISSERT.pdf). Acesso em: 15 fev. 2018.
- [16] ALMEIDA, A. R. et al. Projeto e desenvolvimento de laboratório móvel para monitoramento ambiental dos níveis de temperatura e umidade 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.
- [17] 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.

- [18] POLIZEL, A. L.; NANYA, S.; CONTE, H. INSETOS DIPTERAS COMO POLINIZADORES EM ORCHIDA-CEAE. *REVISTA UNINGÁ*, v. 46, n. 1, 2018. Disponível em: <http://revista.uninga.br/index.php/uninga/article/view/1249>. Acesso em: 01 fev. 2018.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] BOINTON, M. *PHPMailer*. Disponível em: <https://github.com/Synchro/PHPMailer>. Acesso em: 15 mai. 2018.
- [25] 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.
- [26] 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.
- [27] 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://ejournal.unmerpas.ac.id/index.php/informatika/article/view/68/40>. Acesso em: 06 mar. 2018.
- [28] TOTALVOICE. *API de Comunicação por Voz e Texto*. Disponível em: <http://www.totalvoice.com.br/>. Acesso em: 04 abr. 2018.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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](http://mpet.ifam.edu.br/wp-seta/wp-content/uploads/2016/07/Anais_Seta-2017.pdf#page=64). Acesso em: 01 mar. 2018.
- [33] 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.
- [34] MURPHY, F. et al. Big brother for bees (3b) - energy neutral platform for remote monitoring of beehive imagery and sound. *Proceedings - 2015 6th IEEE International Workshop on Advances in Sensors and Interfaces, IWASI 2015*, p. 106–111, 2015.
- [35] 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.
- [36] 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.