



## Modelagem de uma RSSF para manejo de pragas em pomares usando o simulador Eboracum

Felipe Frantz Zanini<sup>1</sup>, Weslen Schiavon de Souza<sup>1</sup>, Lisane Brisolara de Brisolara<sup>1</sup>, Paulo R. Ferreira Jr.<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Computação  
Cyber-Physical Systems Engineering Laboratory  
Centro de Desenvolvimento Tecnológico  
Universidade Federal de Pelotas  
Pelotas, Brasil

{felipe.zanini, wsdsouza, lisane, paulo.ferreira}@inf.ufpel.edu.br

**Abstract.** *Eboracum is an extensible modeling simulation framework for evaluating wireless sensor networks (WSN). This provides high-level primitives for the modeling of platform and application aspects of WSNs, and thus allowing to evaluate efficiency of application-specific WSN solutions. In this work, we employ Eboracum for modelling of solutions for integrated pest management based on smart traps. A small extension of simulator was proposed and implemented in order to model the smart trap as sensor nodes. Based on the proposed modeling and simulation preliminary results, the efficiency of the solution is discussed, and optimizations are planned.*

**Resumo.** *O Eboracum é um framework extensível para modelagem e simulação de redes de sensores sem fio (RSSF). Este fornece primitivas de alto nível de abstração para modelagem da plataforma de hardware bem como aspectos da aplicação, permitindo assim avaliar a eficiência de soluções de RSSF dedicadas a uma dada aplicação. Neste trabalho, empregamos o Eboracum para a modelagem de uma solução de RSSF para o manejo integrado de pragas baseado em armadilhas eletrônicas. Uma pequena extensão do simulador foi proposta para a modelagem das armadilhas como nodos sensores. Com base na modelagem proposta e em resultados preliminares de simulação, a eficiência desta solução é discutida e prospeeta-se otimizações.*

### 1. Introdução

As Redes de Sensores sem Fio (RSSF) têm sido empregadas no sensoriamento e monitoramento de ambientes para as mais diversas aplicações, dentre estas destaca-se a Agricultura

de Precisão (AP), onde redes de sensores tem sido amplamente empregadas para gerenciamento de irrigação, fertilização e controle de pragas [Manikandan e Jayapriya 2018]. Estas redes são normalmente auto-organizáveis e formadas por nós sensores, sendo estes geralmente conectadas entre si através de módulos de rádio frequência (RF), com baixo custo energético [Alsharif et al. 2019].

O aprimoramento da captura de dados e a evolução das tecnologias de informação e comunicação, alavancaram consideráveis melhorias na eficiência, efetividade e produtividade de vários setores agrícolas [Zhang et al. 2018]. A Agricultura de precisão é uma das áreas de aplicação de RSSF, onde sensores são geralmente empregados para monitorar a qualidade do solo e da plantação, coletando dados como umidade, pressão e temperatura. Ainda, RSSF têm sido amplamente empregadas nessa área para o gerenciamento de irrigação, fertilização e para o controle de pragas [Manikandan e Jayapriya 2018].

Com o intuito de identificar a proliferação de pragas em lavouras, armadilhas são comumente espalhadas pelo cultivo, de modo que um responsável fique encarregado de verificá-las periodicamente [Murtaza et al. 2019]. A coleta de dados das armadilhas é feita por humanos, o que não garante a precisão dos dados e a periodicidade requeridas, prejudicando o manejo. A frequência da coleta é muito importante para o controle de infestações em uma dada plantação e RSSF são empregadas para realizar este monitoramento [Shelton e Badenes-Pérez 2006]. Estas redes são usualmente compostas por nodos mantidos por bateria e, portanto, o consumo energético deve ser minimizado de forma a prolongar o tempo de vida da rede.

Para mensurar a eficiência de uma solução de RSSF dedicada para uma aplicação, experimentos precisam ser realizados que permitam avaliar o tempo de vida da rede, bem como a cobertura de sensoriamento e comunicação. Experimentos no campo apesar de oferecerem maior precisão nas análises demandam muitos esforços e custo de aquisição prévia de equipamentos. Por outro lado, simuladores permitem avaliar diferentes soluções com menor custo e esforço, postergando a aquisição de equipamentos e os testes diretamente em campo. No entanto, na avaliação através de simulação, é necessário emular os fenômenos do ambiente real, considerando sua distribuição espacial na área de interesse e a frequência de ocorrência destes eventos usando os modelos estocásticos apropriados para a aplicação alvo [Haghighi e Cliff 2013].

A ferramenta Eboracum [2019] permite a modelagem e avaliação de RSSF, fornecendo primitivas de alto nível para modelar tanto aspectos da camada de hardware quanto de aplicação. Em [Marques et al. 2016], a modelagem de alguns aspectos da aplicação de manejo de pragas com armadilhas eletrônicas foram discutidos. No entanto, neste estudo preliminar, extensões do simulador não foram de fato implementadas e modelos estocásticos para representação da infestação foram generalizados. Estes modelos precisam ser mais precisos uma vez que emulam a carga de trabalho a qual a rede será submetida no campo e afetam diretamente a avaliação da eficiência das soluções de rede.

Este artigo contribui com a modelagem e simulação de soluções de RSSF dedicadas a uma dada aplicação empregando os modelos de alto nível de abstração suportados pelo Eboracum. Por meio de um estudo de caso empregando a aplicação de manejo de pragas, será demonstrado como empregar esta infraestrutura para modelar e simular soluções de RSSF. Neste estudo de caso, uma solução dedicada de RSSF reativa será avaliada.

O artigo está organizado da seguinte forma: a Seção 3 revisa o framework Eboracum, enfatizando as primitivas empregadas ou estendidas para realização dos estudos de caso. Na Seção 4 é apresentado o modelo de armadilha inteligente que será considerado para modelagem dos estudos de caso. O modelo construído para o estudo de caso será detalhado na Seção 5 e os resultados preliminares são apresentados na Seção 6. Por fim, conclusão e trabalhos futuros serão discutidos na Seção 7.

## 2. Trabalhos Relacionados

O emprego de armadilhas eletrônicas para o manejo de pragas vem sendo estudado em vários grupos de pesquisa e por equipes de pesquisa e desenvolvimento de empresas [Parsons et al. 2019] [Ding e Taylor 2016] [Remboski et al. 2018] [Martins et al. 2019]. Em [Parsons et al. 2019] uma revisão é apresentada de várias soluções para controle de pragas em geral envolvendo armadilhas inteligentes. No trabalho de [Remboski et al. 2018] foi discutido o emprego de armadilhas eletrônicas inteligentes no manejo de pragas em pomares, empregando redes neurais para a identificação dos insetos. A identificação de moscas da fruta também foi objeto de estudo de [Martins et al. 2019], onde diferentes técnicas de aprendizado de máquina foram avaliadas para aplicação nas armadilhas eletrônicas. A técnica proposta por [Martins et al. 2019] está sendo prototipada em uma parceria com o Grupo de Pesquisa de Ciberfísicos da UFPel e a empresa Partmon. Este protótipo serviu de base para a caracterização apresentada neste trabalho.

As armadilhas inteligentes podem ser integradas à módulos comunicadores, como discutido em [Parsons et al. 2019], podendo assim ser conectadas de forma a compor uma rede de sensores sem fio. A ideia de modelar uma RSSF dedicada para manejo de pragas no simulador Eboracum já foi proposta em [Marques et al. 2016]. No entanto, o referido trabalho possui limitações pois vários pontos foram apenas prospectados porém sem de fato terem sido experimentados, além de várias caracterizações ainda não estarem definidas pela ausência de protótipos ou trabalhos que suportassem esta caracterização. O presente trabalho contribui não só com o modelo de simulação e as respectivas extensões do simulador mas também com caracterizações que propiciarão novos experimentos desta aplicação no simulador Eboracum, incluindo avaliações de otimizações.

## 3. Eboracum

Eboracum é um framework para projeto e avaliação de RSSFs, que visa modelar e simular RSSFs em alto nível de abstração. Esta infraestrutura de simulação permite avaliar o consumo energético dos nodos da rede ao executar uma aplicação, e consequentemente estimar o tempo de vida da rede.

O Eboracum dispõe de classes que permitem representar os componentes que compõem a rede. O nodo sensor da rede é representado por instâncias da Classe *SimpleWSNNode*, já a estação base ou Sink é representada pela classe *NetworkMainGateway*. Estes representam as principais primitivas para a representação dos componentes que compõem a plataforma de hardware. Ambos possuem capacidade de comunicação, e um parâmetro define o raio de alcance do módulo transmissor. Mas, o nodo sensor também tem capacidade de captura de fenômenos e esta é delimitada por um raio de sensoriamento. Além disso, o nodo sensor tem capacidade de processamento, assim o nodo processa os dados capturados, gerando a informação que será transmitida pela rede.

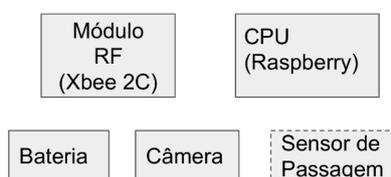
Este simulador permite dois modos de simulação uma sincronizada com relógio, e outra seguindo um modelo de simulação de eventos discretos. O simulador suporta a avaliação do consumo, permitindo analisar o tempo de vida da rede considerando uma dada carga de trabalho. A carga de trabalho de trabalho será modelada através de eventos gerados pelo simulador. A descarga da bateria deverá ser caracterizada no modelo de simulação considerando três modos de operação para os nodos sensores: idle, processamento e transmissão.

Nesta simulação baseada em eventos discretos, as atualizações só ocorrem quando algum evento acontece, o que permite acelerar a simulação. Desta forma a carga de trabalho da aplicação é representada por eventos que acionam os atores do modelo. O Eboracum oferece várias classes para representação de diferentes fenômenos da natureza. A classe *PeriodicEvent* representa o acionamento de nodos sensores para capturas periódicas, com períodos pré-definidos. Dentre as primitivas providas, destaca-se ainda classes que modelam eventos usando distribuições de probabilidade, tal como a classe *StochasticPeriodicJumperEvent*. Esta classe permite representar um conjunto grande de eventos não simultâneos que são distribuídos espacialmente e temporalmente seguindo modelos estocásticos. Muitos fenômenos da natureza, inclusive infestações de pragas, requerem modelos estocásticos para uma simulação mais precisa. Então em nosso estudo de caso, esta classe será empregada no modelo proposto a fim de emular a incidência de insetos à área de interesse que representa o cultivo.

#### 4. Armadilhas Inteligentes

Existe uma considerável variedade de insetos que afetam negativamente a produção de frutas [Sharma et al. 2017]. No sul do Brasil, pesquisadores apontaram a mosca-das-frutas *Ceratitis Capitata* como uma das pragas mais críticas presentes em árvores frutíferas [Mello e Caimi 2008]. Para mitigar as infestações por este inseto, os produtores usam pesticidas. O monitoramento da população de moscas-das-frutas e a tomada de decisões sobre sua mitigação baseiam-se em armadilhas de feromônios.

As armadilhas inteligentes consideradas neste trabalho são baseadas no trabalho de [Martins et al. 2019] e utilizam feromônios para atrair insetos, que ao entrarem nas armadilhas ficam presos em seu interior.



**Figura 1. Diagrama de blocos da armadilha inteligente**

A Figura 1 ilustra os principais módulos que são utilizados no protótipo da armadilha inteligente, que são: CPU, Módulo Raio-frequência (RF), Câmera e uma bateria para alimentar todo o sistema. A câmera captura a imagem do piso adesivo da armadilha. A abordagem proposta usa uma técnica de processamento digital de imagens para detectar regiões de interesse (ROI - *Region of Interest*) que provavelmente representem moscas e uma rede neural artificial para analisar e identificar a classe do inseto capturado. Essas

técnicas são executadas em uma Raspberry Pi III. O módulo de comunicação empregado neste trabalho é o XBee S2C [Digi International Inc. 2020]. Um sensor infra-vermelho de passagem é usado para ativar o sistema quando uma mosca entra na armadilha, fornecendo assim um comportamento reativo para nodos sensores. Estas informações sobre o hardware empregado nas armadilhas são importantes para a construção dos modelos, sobretudo para a caracterização do consumo dos nodos, que será discutida na Seção 5.

## 5. Modelagem Proposta

Neste trabalho, considera-se o emprego das armadilhas propostas em [Martins et al. 2019] para compor uma RSSF dedicada ao manejo de pragas. Estas armadilhas eletrônicas são baseadas em feromônio e possuem recursos para captura de imagens, processamento e comunicação, como detalhado na Seção 4. Cada armadilha é um nodo da rede, que deve processar a imagem capturada e comunicar o total de moscas da fruta detectadas fazendo chegar esta informação através de múltiplos saltos (*multihop*) até a estação base. Estas mensagens serão de 36 Bytes empregando o protocolo Zigbee [Zigbee Alliance], considerando 24 Bytes de cabeçalho e 12 Bytes de Payload.

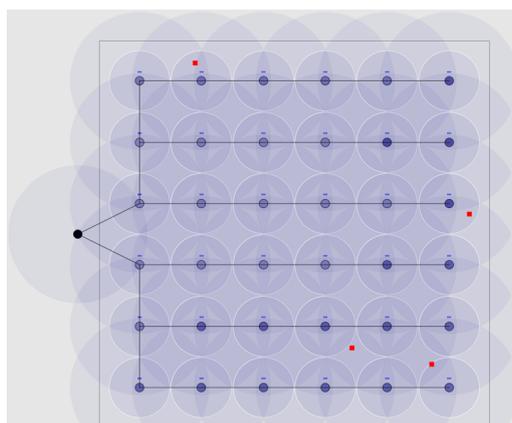


Figura 2. RSSF dedicada ao manejo de pragas

Para avaliar nosso modelo, foi construído um cenário com 36 nodos sensores implantados através de uma área quadrada de 810.000 m<sup>2</sup> (900 m x 900 m) e uma estação base localizada na lateral esquerda, conforme ilustrado na Figura 2. Todos os nodos possuem um módulo RF S2C. O alcance do radio foi considerado equivalente a 160 metros conforme experimentos em campo relatados em [Freitas et al. 2019].

A estação base é representada por uma instância da classe *NetworkMainGateway*. Este nodo só tem capacidade de comunicação. Já os nodos sensores (os quais representam as armadilhas), além da capacidade de comunicação, também são caracterizados por um raio de sensoriamento de 70 metros. Este raio foi definido considerando dados obtidos com especialistas da Partmon que sugerem uma armadilha a cada um hectare quando o feromônio Bio Trimedlure é utilizado.

O nodos sensores também foram caracterizados pela quanto ao custo energético referente ao processamento da imagem capturada pelas armadilhas e técnica de classificação. Este custo depende do número de insetos ou regiões de interesse (ROI) desta imagem. O número de regiões de interesse dependem do número de insetos capturados. Assim, a classe *SimpleWSNNodeWithEventCounter* foi criada no simulador de

forma a representar as armadilhas reativas e contabilizar o número de insetos capturados. Esta classe estende a classe *SimpleWSNNode*, permitindo que a descarga da bateria dos nodos possa considerar a contagem de eventos capturados para a estimativa do consumo energético referente ao processamento

Assim, quando o nodo captura um evento, este nodo deve incrementar seu número de regiões de interesse. Este valor será multiplicado pelo custo energético referente ao processamento de um único ROI para determinar a descarga da bateria do nodo quando este estiver processando. Assim como empregado no protótipo de armadilha desenvolvido, os nodos sensores da solução modelada são mantidos por uma bateria de lítio, a qual fornece uma carga de 25200000 mAs. Este valor portanto é usado como bateria inicial dos nodos sensores em nossos experimentos.

Nosso modelo considera as armadilhas reativas, assim os nodos reagem quando algum evento é detectado em seu raio de sensoriamento. O evento acionará a captura da imagem e o tratamento dos dados, e conseqüentemente o envio da nova informação para a estação base. Na modelagem proposta para a aplicação alvo, os eventos representam os insetos que entram nas armadilhas. Para que a armadilha seja acionada desta forma, um sensor infra-vermelho foi incluído no hardware do nodo sensor. Com a passagem do inseto, este sensor aciona a captura da imagem e na sequência aciona a Raspberry para o processamento da imagem capturada. Concluído o processamento, uma mensagem será enviada via rede até alcançar a estação base.

Os eventos aparecem no espaço seguindo uma distribuição normal invertida, visando emular as moscas chegando prioritariamente no pomar pelas das bordas da área de monitoramento. Para representar aspectos temporais da incidência de insetos na rede, o modelo aqui proposto considera que a maior probabilidade de chegada de insetos ocorre ao entardecer. A frequência dos eventos é dada por uma distribuição de Poisson com uma média de intervalo de 68399 segundos (ou 19 h), desvio padrão de 199 s (2 h) e intervalo de 0 a 86399 segundos (24h). Para representar estes eventos foi criada a classe *StochasticDailyJumperEvent*, que similar a *StochasticPeriodicJumperEvent*, permite criar vários eventos com esta característica, espaçados em um período de 24h. Como o pomar vai estar constantemente monitorado e pesticidas serão usados sempre que necessário, nunca o nível de insetos será muito alto. Assim, forma considerados quatro insetos por dia nesta primeira modelagem, representados por quatro instâncias de *StochasticDailyJumperEvent*.

**Tabela 1. Caracterização do modelo proposto - Custo energético**

Parâmetro	Custo (mAs)
CPUCost	720 x NROI
ComCost	0,038016
IdleCost	38

A Tabela 1 apresenta os custos energéticos dos três modos de operação dos nodos sensores correspondente ao modelo de solução proposto. O custo de processamento neste caso terá um custo dependente do número de ROIs (NROI) na imagem. Cada nodo terá este controle e na descarga da bateria este número será considerado. Assim, o custo *CPUCost* é 720 multiplicado pelo NROI. Para processar a imagem de 1 ROI, a Raspberry

trabalha a 100% de sua capacidade, consumindo 800mA por 0,9 segundos, o que equivale a uma descarga de 720 mAs. O custo de *idle* considera o consumo do módulo RF quando não está transmitindo, mas está com recepção habilitada (28 mA - *idle*) acrescido do consumo do sensor infra-vermelho que consome 10 mA. O custo de comunicação refere-se ao consumo do módulo RF quando transmitindo mensagem. Este custo é mensurado considerando o tamanho do pacote (36 Bytes), taxa de transmissão do módulo (250 Kbps) e a corrente elétrica de transmissão (33 mA), ambos informados no *datasheet* do equipamento.

## 6. Experimentos e Resultados Preliminares

O modelo proposto foi empregado em experimentos de simulação visando avaliar o tempo de vida da rede considerando a carga de trabalho representada neste modelo. Os experimentos foram repetidos trinta vezes e os resultados médios foram considerados nas análises. O teste estatístico *t-student* foi empregado para avaliar a significância dos resultados obtidos e as médias das trinta execuções foram comparadas.

O modelo proposto com sua caracterização de custos foi empregado em um primeiro conjunto de experimentos. Em média a solução permanece ativa por sete dias. Na carga modelada acontecem 4 eventos por dia, mas alguns destes eventos podem ocorrer fora da área de cobertura dos nodos e portanto não são detectados pela rede.

Embora dentre os custos considerados no modelo proposto, o maior deles seja associado ao processamento, como as armadilhas são reativas, estas só consomem este valor quando um evento acontece e aciona-se a captura e classificação da imagem. Desta forma, o custo que se torna mais impactante para a eficiência das soluções é o custo referente ao modo *idle*. Isso porque em um maior número de ciclos de simulação a rede opera neste modo. Qualquer otimização que seja aplicada para reduzir o consumo do processamento ou da transmissão terá pouco impacto, se o custo de *idle* permanecer nesta proporção.

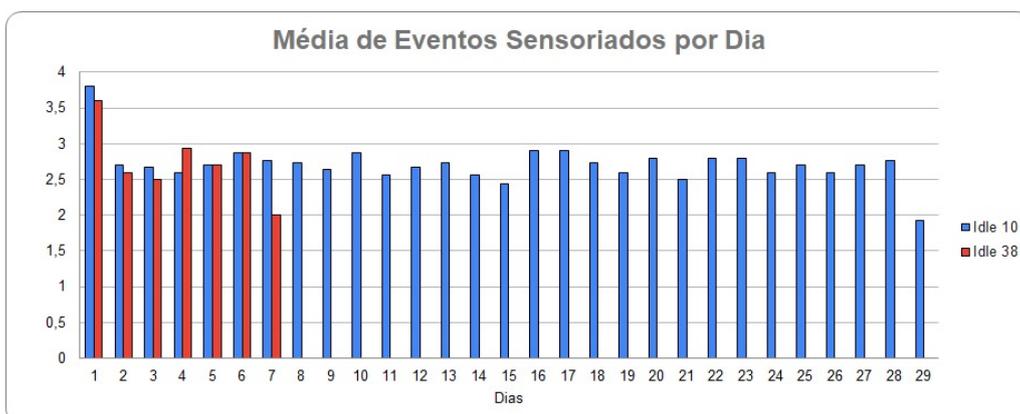


Figura 3. Número de eventos sensorizados por dia - comparativo das soluções

Uma maneira de reduzir este consumo seria colocar o módulo RF a operar em modo *sleep*, assim o nodo consumiria uma corrente de 10,001 mA durante os ciclos inativos. Porém, quando operando neste modo, o nodo não recebe mensagens, diferentemente do modo *idle* suportado pelo Eboracum atualmente. Então, para permitir a otimização de

consumo através deste quarto modo, o simulador precisa ser estendido de forma a permitir simular o chaveamento de nodos do modo *sleep* para modo *idle* quando mensagens tiverem de ser capturadas e retransmitidas.

Para demonstrar esta possível otimização e prospectar sobre o seu ganho, foi realizado um segundo conjunto de experimentos, considerando este valor reduzido de consumo no lugar do custo *idle* no modelo de simulação. Os resultados apresentados na Figura 3 representam a média de eventos sensoreados pela rede por dia nas duas soluções, *idle* de 38 e *idle* de 10, e permitem observar um aumento significativo no tempo de vida da rede com a redução do custo de *idle*. No entanto, é preciso destacar que neste segundo conjunto de experimentos foi considerada uma simplificação dos custos energéticos do módulo RF, a qual despreza o custo das recepções. Para avaliar com maior precisão o ganho de uma solução de RSSF reativa que mantém os nodos em modo *sleep* sempre que possível, migrando para *idle*, processamento ou comunicação somente quando necessário, o simulador precisará ser estendido e o modelo de simulação deverá prever quando o chaveamento de *sleep* para *idle* ocorrerá.

## 7. Conclusão e trabalhos futuros

O simulador Eboracum provê uma infra-estrutura extensível para a modelagem e simulação de RSSF. Neste trabalho é proposta a modelagem de RSSF dedicadas ao manejo de pragas no simulador Eboracum. Neste modelo os nodos são armadilhas eletrônicas que atraem e contabilizam moscas da fruta, auxiliando assim no controle de pragas em pomares.

O artigo contribui para a discussão da modelagem de aspectos da aplicação em um simulador de alto nível de abstração. Ainda, o presente trabalho contribui com extensões para o simulador Eboracum, além de caracterizações realizadas para consumo energético que serão de suma importância a trabalhos que visem a otimização soluções dedicadas para esta aplicação alvo para estender ao máximo as baterias dos nodos.

A distribuição de probabilidade atualmente empregada para determinar a ocorrência dos eventos no tempo, embora relacione a presença do inseto ao horário do dia pode ser aprimorada através de discussão com especialistas. Como trabalhos futuros, pretende-se discutir a definição de um modelo mais preciso.

Os transmissores disponíveis no mercado, inclusive o S2C, possuem um recurso de modo *sleep*, onde o consumo reduzido. Nossos experimentos demonstraram que o tempo de vida da rede pode ser aumentado significativamente ao manter os rádios neste modo. Porém, este modo não é suportado pelo simulador atualmente, que considera apenas o modo *idle* onde o radio está apto a receber mensagens e portanto tem um consumo mais alto. A inclusão do suporte ao modo *sleep* no Eboracum permitirá uma avaliação mais precisa de soluções otimizadas para esta aplicação alvo, bem como representará uma melhoria significativa no simulador como uma ferramenta para exploração do espaço de soluções dedicadas de RSSF.

## 8. Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## Referências

- (2019). Eboracum. Disponível em: <https://sourceforge.net/projects/eboracum/>. Acesso em: 25 de julho 2021.
- Alsharif, M. H., Kim, S., e Kuruoğlu, N. (2019). Energy harvesting techniques for wireless sensor networks/radio-frequency identification: a review. *Symmetry*, 11(7):865.
- Digi International Inc. (2020). Xbee/xbee-pro s2c zigbee: Rf module. Disponível em: <https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf>. Acesso em: 13 de julho 2021.
- Ding, W. e Taylor, G. (2016). Automatic moth detection from trap images for pest management. *Computers and Electronics in Agriculture*, 123:17–28.
- Freitas, L., Martins, V., Ferreira Jr, P., e Brisolara, L. (2019). Aplicação de redes de sensores sem fio para o manejo de pragas em pomares: Avaliação de módulos de comunicação. In *CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UFPEL*.
- Haghighi, M. e Cliff, D. (2013). Multi-agent support for multiple concurrent applications and dynamic data-gathering in wireless sensor networks. In *7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS*.
- Manikandan, S. e Jayapriya, P. (2018). Precision agriculture using wireless sensor network system: Opportunities and challenges. *International Journal Of Engineering And Computer Science*.
- Marques, M. R. S., Brisolara, L., Jr., P. R. F., e Indrusiak, L. S. (2016). Eboracum: An extensible framework for high-level modeling and evaluation of reactive and adaptable wsns. In *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8.
- Martins, V. A. M., Freitas, L. C., de Aguiar, M. S., de Brisolara, L. B., e Ferreira, P. R. (2019). Deep learning applied to the identification of fruit fly in intelligent traps. In *IX Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–8.
- Mello, B. A. d. e Caimi, L. L. (2008). Simulação na validação de sistemas computacionais para a agricultura de precisão. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 12:666 – 675.
- Murtaza, G., Ramzan, M., Ghani, M. U., Munawar, N., Majeed, M., Perveen, A., e Umar, K. (2019). Effectiveness of different traps for monitoring sucking and chewing insect pests of crops. *Egyptian Academic Journal of Biological Sciences. A, Entomology*, 12(6):15–21.
- Parsons, L., Ross, R., e Robert, K. (2019). A survey on wireless sensor network technologies in pest management applications. *SN Applied Sciences*, 2(1).
- Remboski, T., Souza, W., Aguiar, M., e Ferreira Jr, P. (2018). Identification of fruit fly in intelligent traps using techniques of digital image processing and machine learning. In *33rd Annual ACM Symposium on Applied Computing*, pages 260–267.
- Sharma, S., Kooner, R., e Arora, R. (2017). Insect pests and crop losses. In *Breeding insect resistant crops for sustainable agriculture*, pages 45–66. Springer.

- Shelton, A. e Badenes-Pérez, F. (2006). Concepts and applications of trap cropping in pest management. *Annual review of entomology*, 51:285–308.
- Zhang, Y., Chen, D., Wang, S., e Tian, L. (2018). A promising trend for field information collection: An air-ground multi-sensor monitoring system. *Information Processing in Agriculture*, 5.
- Zigbee Alliance. Zigbee specification. Disponível em: <http://www.zigbee.org/>. Acesso em: 25 de setembro 2021.