

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE  
TECNOLOGIAS**

**OLACIR CYPRIANO TOMASINI**

**AVALIAÇÃO DE ESTRATÉGIA DE *SLEEPMODE*  
PARA RSSF APLICADA À MONITORAÇÃO DE  
AMBIENTE**

**CAMPINAS  
2018**

**OLACIR CYPRIANO TOMASINI**

**AVALIAÇÃO DE ESTRATÉGIA DE SLEEPMODE PARA  
RSSF APLICADA À MONITORAÇÃO DE AMBIENTE**

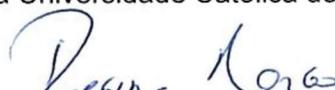
Dissertação apresentada ao Curso de Mestrado em Sistemas de Infraestrutura Urbana do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito parcial para obtenção do título de Mestre em Sistemas de Infraestrutura Urbana.  
Área de Concentração: Sistemas de Infraestrutura Urbana.  
Orientador (a): Prof. (a). Dr. (a). Omar Carvalho Branquinho.

Dissertação defendida e aprovada em 27 de fevereiro de 2018 pela Comissão Examinadora constituída dos seguintes professores:



---

Prof. Dr. Omar Carvalho Branquinho  
Orientador da Dissertação e Presidente da Comissão Examinadora  
Pontifícia Universidade Católica de Campinas



---

Profa. Dra. Regina Márcia Longo  
Pontifícia Universidade Católica de Campinas



---

Profa. Dra. Thais Queiroz Zorzeto Cesar  
Universidade Estadual de Campinas - UNICAMP

Ficha catalográfica elaborada por Marluce Barbosa – CRB 8/7313  
Sistema de Bibliotecas e Informação - SBI - PUC-Campinas

t621.3851 Tomasini, Olacir Cypriano.  
T655a Avaliação de estratégia de sleepmode para RSSF aplicada à monitoração de ambiente / Olacir Cypriano Tomasini. - Campinas: PUC-Campinas, 2018.  
87 f.

Orientador: Omar Carvalho Branquinho.  
Dissertação (mestrado) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Sistemas de Infraestrutura Urbana.  
Inclui anexo e bibliografia.

1. Redes de sensores sem fio. 2. Monitoramento ambiental. 3. Proteção ambiental. 4. Catástrofes naturais. 5. Energia - Fontes alternativas. I. Branquinho, Omar Carvalho. II. Pontifícia Universidade Católica de Campinas. Centro Ciências Exatas, Ambientais e de Tecnologias. Pós-Graduação em Sistemas de Infraestrutura Urbana. III. Título.

CDD – 22. ed. t621.3851

Dedico este trabalho primeiramente à essa energia universal que chamamos de Deus, por me proporcionar o dom da capacidade e a possibilidade de sonhar em todos os momentos dessa jornada vivência. Em especial, também dedico este trabalho à minha fé. Fé que me ofereceu apoio durante todos os meses, equilibrando minha sanidade.

## AGRADECIMENTOS

À professora Dr<sup>a</sup>. Marina Lavorato,  
Pela ajuda na estruturação do projeto e co-orientação.

Ao professor Dr. Alexandre de Assis Mota (*in memoriam*) e à professora Dr<sup>a</sup>. Lia Toledo M. Mota,  
Pela confiança, competência, apoio inicial para execução do projeto e por toda a ajuda no decorrer do mesmo.

À toda minha família,  
Especialmente minha mãe Valquiria e irmã Débora, Mylow e Meg, pela confiança que depositaram em mim em todas as fases da minha vida. Sempre com muita dedicação, carinho e amor.

À professora Dr<sup>a</sup>. Lia Lorena Pimentel e ao Projeto FAPESP 2014/20486-8,  
Pela ajuda na aquisição dos materiais imprescindíveis para o projeto.

À professora Dr<sup>a</sup>. Regina Márcia Longo,  
Pelo auxílio nas questões e análises envolvidas na área da engenharia ambiental.

Aos professores do programa de pós-graduação em Sistemas de Infraestrutura Urbana,  
Pelo apoio e auxílio em todos os momentos do mestrado.

Aos amigos Camila Madeiros, Roger Prior Gregio, Lilian Corpas, Giovanna Rizzo e em especial à Paula Minorin,  
Pelo apoio emocional, amizade e pela imensa contribuição na evolução do meu trabalho.

À todos os alunos da turma do Mestrado de Infraestrutura Urbana com início em 2016,  
Pela descontração e enriquecimento de conhecimento nas aulas.

Ao técnico Eduardo Veiga do laboratório de Meios de Transmissão e Redes, ao laboratório de Materiais de Construção Civil e Estruturas, ao Departamento de Manutenção, à diretoria do CCHSA e à todo o corpo de funcionários da PUC Campinas,  
Pela competência e presteza.

Ao professor Dr. Omar Carvalho Branquinho,  
Pelo acompanhamento deste trabalho.

“Sem música, a vida seria um erro.”  
Friedrich Nietzsche (1844-1900)

## RESUMO

TOMASINI, Olacir Cypriano. *Avaliação de estratégia de sleepmode para rede de sensores sem fio aplicada à monitoração de ambiente*. Dissertação (Mestrado em Sistemas de Infraestrutura Urbana) – Programa de Pós-graduação em Sistemas de Infraestrutura Urbana, Pontifícia Universidade Católica de Campinas, Campinas, 2018.

A preocupação com o meio ambiente intensificou-se nos últimos anos em decorrência das mudanças climáticas e dos desastres ambientais, exigindo uma maior conscientização ambiental, aliada à necessidade de se preservar o meio ambiente de forma inteligente e sustentável. Logo a monitoração ambiental aparece com a finalidade de avaliar e controlar possíveis impactos ambientais, utilizando por exemplo uma Rede de Sensores Sem Fio – RSSF. As RSSF são um meio eficiente de acompanhar locais remotos e de difícil acesso que necessitem de monitoração para prováveis condições emergenciais, como por exemplo deslizamentos. Para isso utilizaram-se também tecnologias que permitiram o monitoramento, o controle e a coleta de parâmetros ambientais. Instalou-se a RSSF em um local sem disponibilidade de energia cabeada, utilizando-se assim o nó sensor sem fio energizado através de uma bateria, carregada por um painel fotovoltaico. Além disso, empregaram-se estratégias de economia de energia (*SleepMode*) via *software* para aumentar o tempo de vida da bateria. Os resultados mostraram uma RSSF capaz de monitorar grandezas ambientais por longos períodos, uma vez que foram implementados algoritmos sofisticados visando a economia de energia, tratamento dos dados e utilização de um servidor de aplicação para visualização em tempo real dos dados em forma gráfica. Deste modo, a tecnologia se mostrou eficiente ao atender o proposto.

**Palavras-chave:** Condições Emergenciais, Monitoração Ambiental, *Radiuino*, Rede de Sensores Sem Fio, *SleepMode*.

# ABSTRACT

TOMASINI, Olacir Cypriano. *Evaluation of sleepmode strategy for wireless sensor network applied to environment monitoring*. 2018. Master's thesis, the Master's course in Urban Infrastructure Systems, Pontifical Catholic University of Campinas, Campinas, 2018.

Concern for the environment has intensified in recent years as a result of climate change and environmental disasters, requiring greater environmental awareness, together with the need to preserve the environment in an intelligent and sustainable way. Therefore the environmental monitoring appears with the purpose of evaluating and controlling possible environmental impacts, using for example a Network of Wireless Sensors - WSN. WSNs are an efficient means of tracking remote and difficult access sites that require monitoring for probable emergency conditions, such as landslides. In order to do so, we also used technologies that allowed the monitoring, control and collection of environmental parameters. The WSNF was installed in a location without the availability of wired power, using the wireless sensor node powered by a battery, charged by a photovoltaic panel. In addition, energy-saving strategies (SleepMode) were used via software to increase battery life. The results showed a WSNF capable of monitoring environmental quantities for long periods, since sophisticated algorithms were implemented aiming at energy saving, data processing and use of an application server for real time visualization of the data in graphic form. In this way, the technology proved to be efficient in meeting the proposal.

**Keywords:** *Emergency Conditions, Environmental Monitoring, Wireless Sensor Network, Rarduino, SleepMode.*

## LISTA DE ILUSTRAÇÕES

<b>Figura 1</b> – Representação do consumo dos componentes de um NSSF.....	22
<b>Figura 4</b> – Visão sistêmica da aplicação do sistema na Fase 1.....	27
<b>Figura 5</b> – Foto dos componentes utilizados na Fase 1 do projeto. ....	28
<b>Figura 6</b> – Identificação dos principais componentes da placa <i>DK106</i> .....	30
<b>Figura 7</b> – Módulo de radiofrequência <i>BE990</i> . ....	30
<b>Figura 8</b> – Bateria do tipo polímero de íon de lítio 3,7V – 1800 mAh. ....	34
<b>Figura 9</b> – Programador <i>UartsBee v4</i> . ....	34
<b>Figura 10</b> – Antena mono polo $\frac{1}{4} \lambda$ . ....	35
<b>Figura 11</b> – Preenchimento da estrutura em PVC com cimento e argamassa. ....	35
<b>Figura 12</b> - Caixa hermética aberta mostrando o circuito interno e, a direita, estrutura montada.....	36
<b>Figura 13</b> – Mapa do pacote de 52 <i>bytes</i> da plataforma <i>Radiuino</i> . ....	37
<b>Figura 14</b> – Representação dos estados de funcionamento do módulo RF. ....	38
<b>Figura 15</b> – Interface de trabalho do <i>software RadiuinoTest</i> . ....	39
<b>Figura 16</b> – Ilustração do cenário de testes. ....	40
<b>Figura 17</b> – <i>Interface</i> primária do <i>software</i> gerenciador codificado em <i>Python</i> . ....	41
<b>Figura 18</b> – <i>Interface Shell</i> de execução da aplicação em <i>Python</i> . ....	42
<b>Figura 19</b> – Ativando a função <i>SleepMode</i> através do <i>Python Shell</i> . ....	44
<b>Figura 20</b> – Visão sistêmica da aplicação na Fase 2. ....	46
<b>Figura 21</b> – <i>Kit IoT</i> . ....	47
<b>Figura 22</b> - <i>Raspberry Pi 3</i> . ....	48
<b>Figura 23</b> – Sensor de umidade desolo e nível de água <i>Vegetronix VH400</i> .....	49
<b>Figura 24</b> – Sensor de umidade e temperatura do ar <i>DHT22</i> . ....	50
<b>Figura 25</b> – Painel fotovoltaico 1 W. ....	51
<b>Figura 26</b> – Fluxograma do ciclo de funcionamento das condições emergenciais.....	54
<b>Figura 27</b> – Imagem via satélite das áreas de interesse. ....	56
<b>Figura 28</b> – Distância entre NSSF (Área 1) e estação base.....	56
<b>Figura 29</b> – Local de alocação interna da estação base, e à direita externa. ....	57
<b>Figura 30</b> – Área de interesse (Área 1). ....	58
<b>Figura 31</b> – Área de interesse com NSSF já fixado (Área 1). ....	58
<b>Figura 32</b> – Sensor de umidade do solo <i>VH400</i> à 18 cm da superfície (Área 1). ....	59
<b>Figura 33</b> – Sensor de umidade do solo <i>VH400</i> instalado (Área 1).....	59
<b>Figura 34</b> – Prato para proteção da radiação solar direta no sensor <i>DHT22</i> (Área 1).....	60
<b>Figura 35</b> – Suporte cedido e, à direita, tripé substituído e já fixado (Área 1).....	60
<b>Figura 36</b> – NSSF e seu aspecto final em campo (Área 1).....	61
<b>Figura 37</b> – Distância entre NSSF (Área 2) e estação base.....	62
<b>Figura 38</b> – Grade de metal circular e declínio do solo na Área 2. ....	62
<b>Figura 39</b> – NSSF fixado na Área 2. ....	63
<b>Figura 40</b> – <i>Interface</i> inicial do <i>Zabbix</i> . ....	63
<b>Figura 41</b> – Acesso à plataforma <i>Grafana</i> .....	64
<b>Figura 42</b> – Indicadores animados em tempo real de uma simulação teste. ....	65
<b>Figura 43</b> - Gráfico do processo de descarga da bateria do NSSF sem a função <i>SleepMode</i> . ....	66
<b>Figura 44</b> - Gráfico do processo de descarga da bateria do NSSF com a função <i>SleepMode</i> . ....	67
<b>Figura 45</b> - Tempos de vida do NSSF sem <i>SleepMode</i> e NSSF com função <i>SleepMode</i> .....	67
<b>Figura 46</b> – Gráfico demonstrativo dos resultados obtidos.....	68
<b>Figura 47</b> – Detalhamento da <i>Interface</i> dos dados.....	69
<b>Figura 48</b> – Detalhamento gráfico da variável bateria nos últimos 3 meses.....	69

<b>Figura 49</b> – <i>Zoom</i> de 28 dias da variável bateria. ....	70
<b>Figura 50</b> - Detalhamento gráfico da variável coeficiente volumétrico nos últimos 3 meses.....	70
<b>Figura 51</b> – Gráficos temporais sobrepostos das variáveis temperatura e umidade. ....	71
<b>Figura 52</b> – Indicador animado e gráfico temporal sobrepostos em tempo real. ....	71
<b>Figura 53</b> – Indicador animado e gráfico temporal sobrepostos em tempo real. ....	72

## SUMÁRIO

1.	INTRODUÇÃO .....	13
	1.1 Delineamento do problema .....	13
	1.2 Objetivo geral e objetivos específicos .....	15
	1.3 Justificativa .....	15
	1.4 Organização .....	15
2	REVISÃO DA LITERATURA.....	17
	2.1 Monitoramento ambiental .....	17
	2.2 Redes de Sensores Sem Fio – RSSF .....	19
	2.3 Energias renováveis .....	23
	2.3.1 Energia solar.....	24
3	MATERIAIS E MÉTODOS .....	26
	3.1 Fase 1 .....	26
	3.1.1 Nó sensor sem fio .....	27
	3.1.2 Placa de desenvolvimento DK106 .....	28
	3.1.3 Módulo de radiofrequência BE990.....	30
	3.1.4 Bateria íon de lítio 3,7V – 1800 mAh e 3600 mAh .....	33
	3.1.5 Programador UartsBee v4 .....	34
	3.1.6 Antena mono polo $\frac{1}{4} \lambda$ (lambda).....	35
	3.1.7 Estrutura de suporte do NSSF.....	35
	3.1.8 Plataforma Arduino e Firmware .....	36
	3.1.9 Plataforma Radiuino .....	36
	3.1.10 Função SleepMode.....	37
	3.1.11 Radiuino Test.....	38
	3.1.12 Python.....	39
	3.1.13 Testes em ambiente protegido.....	39
	3.2 Fase 2 .....	44
	3.2.1 Zabbix.....	47

	3.2.2	Raspberry Pi 3 .....	48
	3.2.3	Vegetronix VH400.....	49
	3.2.4	DHT22 .....	50
	3.2.5	Painel fotovoltaico 1W .....	51
	3.2.6	Função SleepMode 2.....	52
	3.2.7	Condições emergenciais.....	53
	3.2.8	Testes em ambiente externo .....	55
4		RESULTADOS .....	66
	4.1	Fase 1 .....	66
	4.2	Fase 2 .....	68
5		CONCLUSÕES .....	74
6		TRABALHOS FUTUROS.....	76
7		REFERÊNCIAS BIBLIOGRÁFICAS.....	77
8		APÊNDICES.....	82
	8.1	Código Python.....	82
9		ANEXOS .....	88
	9.1	Anexo A – Esquema elétrico da placa DK106.....	88

# 1. INTRODUÇÃO

## 1.1 *Delineamento do problema*

A preocupação com o meio ambiente intensificou-se nos últimos anos, em especial após uma série de alterações climáticas já conhecidas acontecerem por consequência do aumento da temperatura média da atmosfera da Terra e dos oceanos, conhecida como aquecimento global. Tais alterações acarretaram no derretimento de geleiras, na interrupção do ciclo de chuvas de certas regiões e interferiu diretamente, de forma nociva, no ritmo natural do meio ambiente.

Dentre os fatores-causa, muitas atividades humanas insustentáveis como a queima de combustíveis fósseis, o uso indevido da terra, o desmatamento e a utilização de energias poluidoras, conciliadas com a explosão populacional e o crescimento econômico, causando massivas emissões de gases poluentes que intensificam, em especial, o efeito estufa (MENDONÇA, 2003).

Tais fatores têm exigido das pessoas uma maior conscientização ambiental, aliada a necessidade de se preservar o meio ambiente de forma inteligente e sustentável, através da monitoração ambiental.

A monitoração ambiental permite adquirir informações e possibilitar a observação contínua e detalhada das variáveis ambientais, objetivando reconhecer e aquilatar as circunstâncias dos recursos naturais em um certo momento – de forma qualitativa ou quantitativa - bem como as tendências ao longo do tempo (CAROLINA, 2013).

O monitoramento ambiental fornece informações a respeito dos fatores que possuem uma influência direta na condição de conservação, preservação, deterioração e recuperação ambiental da região em estudo, auxiliando também na definição e melhoria de políticas ambientais. A monitoração e análise de variáveis ambientais oferece desde avaliações no impacto urbano até mesmo uma melhor compreensão da relação das ações do homem com o meio ambiente.

O monitoramento ambiental pode ser feito em escalas pequenas ou grandes, através da utilização de câmeras fotográficas de alta resolução embutidas em satélites, sondas, raios X e sensores remotos (constituindo uma Rede de Sensores Sem Fio - RSSF) para mensuração de grandezas ambientais. (PODELESKI, 2013).

Uma RSSF é uma rede de sensores cujo objetivo é monitorar algum fenômeno. A principal função de uma RSSF é coletar informações medindo alguma grandeza do meio físico, processar a informação e propagá-la através de um meio de transmissão não cabeado. Uma RSSF é composta por uma unidade de processamento conectada a uma base (estação base responsável pelo *link* direto com o Nó Sensor Sem Fio - NSSF), *software* para gerência e decodificação dos dados coletados, rádio para comunicação sem fio e NSSF. (ASSUMPÇÃO, 2011). O NSSF é um dispositivo eletrônico que abriga os sensores responsáveis pela leitura das grandezas a serem medidas. Tais sensores podem ser analógicos ou digitais.

A energia elétrica cabeada não existe ou é deficiente em grande parte dos locais que necessitam de monitoração, exigindo então uma outra fonte de energia para os dispositivos eletrônicos que realizarão as medidas das grandezas físicas. Para que nenhuma outra forma de energia poluidora seja utilizada, esse trabalho fará a utilização de energia renovável sustentável.

São consideradas fontes de energia renováveis todas aquelas em que seus recursos possuem a capacidade de regeneração, ou seja, inesgotáveis, diferente de algumas fontes não renováveis como o urânio, petróleo e carvão, que são recursos naturais, porém em quantidades limitadas.

Entre as fontes de energias renováveis, destacam-se a eólica (geração de energia elétrica utilizando a energia dos ventos), biomassa (realizando combustão de matéria orgânica para produção de energia) e na energia solar (oriunda do calor ou luz do Sol). A energia solar é utilizada por meio de diferentes tecnologias, como a heliotérmica, o aquecimento solar e a energia solar fotovoltaica (INTERNATIONAL ENERGY AGENCY, 2010).

A energia solar fotovoltaica é obtida utilizando sistemas fotovoltaicos chamados células fotovoltaicas, que transformam a radiação solar em energia elétrica através do efeito fotovoltaico, gerando uma fonte de energia que servirá como principal meio de alimentação do NSSF.

## 1.2 *Objetivo geral e objetivos específicos*

O objetivo principal desse trabalho é criar e avaliar uma solução para monitoramento dos ambientes urbanos, florestas ou recortes de mata, de forma automatizada e com qualidade, através do desenvolvimento de uma RSSF e um NSSF alimentados por energia solar fotovoltaica com o uso, foco e análise da tecnologia *Sleepmode* como atributo de melhoria em eficiência energética para que tal artifício auxilie o *time to live* do sistema de monitoração, minimizando os problemas de energia encontrados em RSSF, possibilitando que o dispositivo mantenha-se ativo em quaisquer condições de clima, mesmo que a fonte de energia solar esteja em baixo potencial por dias consecutivos.

## 1.3 *Justificativa*

Muitos dos locais que necessitam de monitoração são áreas perigosas e de difícil acesso, fazendo com que a utilização de tecnologias sem fio seja necessária.

Os sistemas de monitoramento atuais (como por exemplo as estações meteorológicas convencionais) não disponibilizam uma plataforma modular para que a troca de sensores seja possível, se adequando a cada tipo de monitoramento.

Em algumas épocas do Outono ou Inverno, a incidência de Sol é menor devido as características dessas estações do ano, bem como dias seguidos de chuva e céu nublado, dificultando ou impossibilitando a coleta de energia fotovoltaica, demandando a necessidade de melhorias em relação a eficiência energética.

Com base nos princípios apresentados, é evidente a importância em técnicas e aprimoramento de *software* e *hardware*, para otimizar o consumo de energia de dispositivos que usam energias renováveis como fonte principal de alimentação.

## 1.4 *Organização*

Esta dissertação está organizada, como se segue.

No capítulo 2, é apresentada a revisão de literatura dos temas abordados no projeto, bem como a descrição de trabalhos relacionados.

No capítulo 3, as duas fases de execução do projeto são expostas, os materiais utilizados são descritos e a metodologia aplicada nos testes em ambientes controlado e externo são detalhadas.

No capítulo 4, os resultados parciais coletados nos testes são apontados.

No capítulo 5, uma análise dos dados é feita, acrescida de conclusões finais sobre ambas as fases.

No capítulo 6, os trabalhos futuros referentes ao projeto são apresentados.

A dissertação é finalizada com a descrição dos trabalhos futuros, referências bibliográficas e a disponibilização dos apêndices e anexos.

## 2 REVISÃO DA LITERATURA

### 2.1 *Monitoramento ambiental*

Os grandes tratados internacionais sobre o clima evidenciaram a necessidade da monitoração ambiental em diversos locais do mundo, objetivando a coleta de resultados para um melhor entendimento dos impactos que as ações antrópicas causaram no decorrer das últimas décadas.

É de extrema importância o conhecimento dos padrões climáticos para todo sistema de controle e monitoração. Tais padrões asseguram valiosas informações a respeito das condições ambientais, possibilitando o diagnóstico e classificação dos impactos ambientais e posteriormente optar por melhores medidas de controle ambiental de forma mais confiável pois, ao compreender os impactos diagnosticados será possível elaborar de forma planejada as ações a serem executadas (CAROLINA, 2013).

O monitoramento de grandezas no meio ambiente tem crescido, em especial após o reconhecimento do efeito estufa, principal causador do aquecimento global. O Painel Intergovernamental de Mudanças do Clima (IPCC) foi o primeiro mecanismo de caráter científico, criado em 1990, no intuito de alertar o planeta sobre os problemas ocasionados pelo aquecimento global. (*INTERGOVERNMENTAL PANEL ON CLIMATE CHANGE*, 1988). O IPCC realizou projeções referentes aos impactos ocasionados pelas mudanças climática para diversos países. No que tange ao Brasil, projetou aumento de temperatura 2°C até 4°C em relação a temperatura média aferida entre 1961 e 1990, comparando as diferenças climáticas desses períodos (MARENGO, 2006).

Após grande pressão exercida sobre os governos, fruto de ONG's ambientais, aliados a opinião pública e meios de comunicação, foram desenvolvidos tratados internacionais sobre clima e meio ambiente, como o Protocolo de Quioto. O Protocolo de Quioto é um tratado complementar à Convenção-Quadro das Nações Unidas sobre Mudança do Clima, em que foram definidas metas de redução na emissão de poluentes, em especial relativos aos países desenvolvidos ou aos que estavam próximos de se tornarem desenvolvidos.

Ele foi criado em 1997, e entrou em vigor oficialmente em 2005 (*UNITED NATIONS – CLIMATE CHANGE*, 2017).

Muitas são as grandezas que podem ser acompanhadas, porém as mais frequentes são: umidade, temperatura, pressão, CO<sub>2</sub> (em gasodutos, indústrias e edificações) utilizando sensores no meio urbano e GEE (Gases de Efeito Estufa). (MELLO et al., 2007). Para Queiroz et al. (2008), acompanhar e monitorar as variáveis envolvidas nas diferentes cadeias ambientais é necessário para gerar as informações precisas nas tomadas de decisões, tanto em sua gestão quanto em sistemas de controle.

São diversas as formas de monitoramento ambiental encontradas na literatura. Segundo Molin (2004), os sistemas de monitoramento relacionados ao meio ambiental e que são comumente utilizados, em especial na Agricultura de Precisão (AP), são videografias, imagens de satélites, fotografias aéreas, observações; e possuem inúmeras limitações, representando uma avaliação tardia da área de interesse, uma vez que o interessado na monitoração só poderá aplicar correções posteriormente. Destaca-se, então, a necessidade de monitoração baseada em princípios tecnológicos, que ofereça informações em tempo real e *on line* (MOLIN, 2004).

Muitas vezes, áreas de interesse a serem monitoradas estão situadas em locais inóspitos, perigosos ou de difícil acesso. Além disso, inúmeros processos no ramo agrícola necessitam da mobilidade dos dispositivos que realizam as medidas devido às limitações quanto ao uso de cabos ou fios. Tais regiões também podem necessitar de vigilância constante devida à suscetibilidade de riscos ou situações tendenciosas cujo monitoramento possibilite o alerta sobre as condições emergenciais causadas por desastres naturais.

Os desastres naturais demandam atenção. Discussões e diversos debates são realizados com o intuito de explorar e aprimorar melhores soluções que diminuam os danos causados por tais fenômenos. (REIS et al., 2011). Nesse contexto, os sistemas de monitoramento têm ganhado notoriedade, devido à sua capacidade de prever, minimizar a deterioração e diminuir perdas de vidas e danos ao meio ambiente, dessa forma uma das melhores opções para monitoramento é a utilização de uma Rede de Sensores Sem Fio - RSSF (SANTOS et al., 2010).

## 2.2 Redes de Sensores Sem Fio – RSSF

As Redes de Sensores Sem Fio (RSSF) podem ser descritas sucintamente como redes constituídas por dispositivos de pequeno porte (Nó Sensor Sem Fio – NSSF), munidos de sensores, com a capacidade de transmissão de dados sem fio entre si para a disponibilização dos dados de sensoriamento onde eles são necessários (ASSUMPÇÃO, 2011).

O NSSF é um dispositivo eletrônico que abriga os sensores responsáveis pela leitura das grandezas a serem medidas. A configuração básica de *hardware* para o funcionamento de um NSSF constitui-se de unidade de processamento, módulo-rádio transceptor de dados através de radiofrequência (RF), sensores e alimentação energética. Tais sensores agregados podem ser confeccionados para que funcionem nas formas analógica ou digital.

Akyildiz et al. (2002) e Wang & Balasingham (2010) descrevem uma RSSF como uma rede caracterizada por microprocessadores aptos a mensurar, acompanhar e reger dispositivos e é constituída de um ou mais NSSF, comunicando-se entre si via radiofrequência. Loureiro et al. (2003) complementam, ainda, que a tecnologia de RSSF contempla diversas áreas do conhecimento, onde inúmeros tópicos de telecomunicações são agregados (antenas, rádios e etc), com a utilização de microcontroladores, sistema de controle digital e analógico, entre outros.

A área militar dos Estados Unidos criou, na década de cinquenta, uma rede com o intuito de detectar e rastrear submarinos soviéticos. Foram espalhados pelo Atlântico e Pacífico inúmeros nós sensores equipados com sensores acústicos à prova d'água e que se comunicavam entre si, dando origem às RSSF. (WINKLER et al., 2008). Posteriormente a utilização expandiu-se para diversas áreas além da militar e industrial, em particular após a popularização dos controladores e microcontroladores programáveis (ZHAO, 2011).

É possível encontrar a utilização de RSSF para monitoramento de variáveis ambientais em diversas pesquisas e aplicações. O trabalho de Barros et al. (2012) mostra a utilização de uma RSSF para o monitoramento de grandezas como umidade e temperatura em uma cultura experimental de milho (BARROS et al., 2012). Teruel et al. (2012) empregaram uma RSSF para avaliar o desempenho de redes de monitoramento ambiental indoor em casas de vegetação. (TERUEL et

al., 2012). Segundo Moraes et al. (2004) e Jackson et al. (2007), a utilização de uma RSSF para monitoramento de umidade de solo é viável. Além da umidade de solo, Podeleski (2013) comprovou também a eficácia da utilização de RSSF para medidas de temperatura do solo.

No ramo hospitalar, alguns autores fizeram uso dos recursos de RSSF para monitorar o nível de eficiência de um esterilizador de ozônio, mensurando concentração de gases, temperatura e umidade, solucionando assim a problemática da esterilização de materiais que não suportam quantidades altas de calor (LUQUETA et al., 2016). O trabalho de Lima (2016) usa uma RSSF em conjunto com um medidor de sinais vitais e um sistema supervisório, aumentando a eficiência e proporcionando facilidade na disponibilização dos dados vitais medidos dos pacientes hospitalares (LIMA et al., 2016).

As RSSF utilizadas para monitoramento estão presentes também em monitorações de condições específicas de clima, para determinar o potencial risco de deriva na pulverização de pesticidas agrícolas (TIEMI et al., 2017), bem como no sensoriamento climático em salas de servidores (*Data Centers*), auxiliando na melhor eficiência da refrigeração necessária para esses tipos de dependências (CAMARGO & MIERS, 2015).

Moncks et al. (2016) aplicaram uma RSSF para solucionar os eventuais problemas de detecção de incêndios, oriundos das formas convencionais de monitoramento para esse tipo de aplicação (geralmente realizada com câmeras espalhadas pelas florestas). Os autores também citaram falhas e problemas técnicos nos métodos de monitoramento de incêndios mais usados atualmente:

- Alto índice de alarmes falsos e desempenho reduzido a noite;
- Necessidade de grande uso de largura de banda na transmissão das imagens;
- Custo elevado e variação constante da luminosidade das cenas;
- Impossibilita a visualização de focos de incêndio por detrás de relevos acidentados ou árvores;
- Extremas e desfavoráveis condições climáticas restringem a eficiência.

É exequível a utilização de RSSF para monitoramento de outra RSSF. Garcia et al. (2013) propõem um sistema de monitoramento passivo para RSSF.

No monitoramento passivo, uma RSSF adicional depura e analisa o funcionamento de uma RSSF em operação (chamada de rede alvo), coleta pacotes enviados e recebidos da rede alvo e possibilita um acompanhamento completo da rede para que melhores condições energéticas possam ser também aplicadas (GARCIA et al., 2013).

A ascensão das RSSF está ligada e é impulsionada pela Internet das Coisas ou *Internet Of Things* (CLAUDIO et al., 2016), pois outorga ao mundo físico conectar-se ao digital, através da comunicação entre os dispositivos (e objetos) e o envio de seus dados a uma Nuvem ou *Data Center*.

Destacam-se as vantagens e as desvantagens de uma RSSF na Tabela 1:

**Tabela 1 – Vantagens e desvantagens de uma RSSF**

<b>Vantagens</b>	Cobertura abrangente da área à ser monitorada; Possibilidade de gerenciamento de energia; Maior discriminação dos dispositivos; Suporte à diversos tipos de sensores e transdutores; Diminuição de gastos com cabeamento; Mobilidade dos NSSF; Alimentação por energias renováveis; Envio dos dados a longa distância e em tempo real.
<b>Desvantagens</b>	Dificuldade técnica para o desenvolvimento e aplicação (dependendo da plataforma utilizada e do local monitorado); Necessidade da disposição de energia elétrica; Pouca capacidade de processamento devido ao tamanho reduzido dos dispositivos; Consumo de energia.

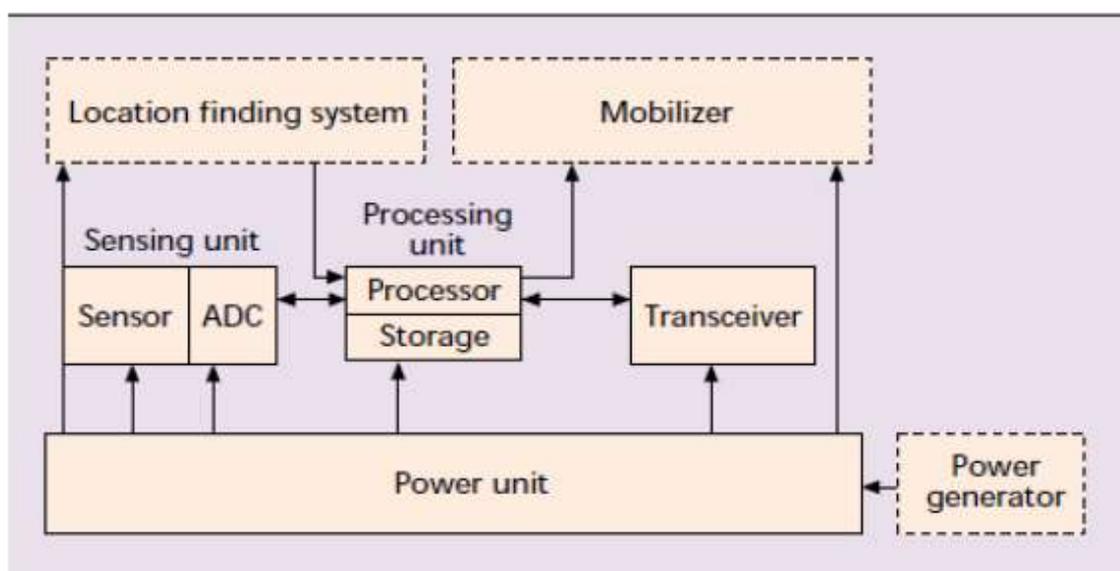
Roseline e Sumathi (2014) apontam que o maior desafio que as RSSF podem encontrar é a energia. O consumo de energia numa RSSF deve ser considerado com extrema prioridade, uma vez que a energia disponível nem sempre origina-se de uma fonte que em boa parte do tempo está acessível, como a energia elétrica cabeada convencional. Logo, entende-se que a eficiência energética é fator primordial para os futuros desenvolvimentos de tecnologias e processos em RSSF (ROSELINE & SUMATHI, 2014).

Muitas vezes, os territórios onde as RSSF estão situadas são regiões afastadas, carentes de infraestrutura básica e que não possuem disponibilidade de energia elétrica cabeada, tornam-se então tais fatores preocupantes, tanto devido à grande parte dessas áreas serem reservas ambientais ou protegidas, quanto à inacessibilidade do local, onde flui a necessidade de utilização de energias renováveis disponíveis na zona de interesse.

Para Akyildiz et al. (2002), pode-se classificar o consumo de uma RSSF em três pilares: sensoriamento, comunicação e processamento dos dados.

A Figura 1 disponibiliza uma visão sistêmica da composição dos três pilares de consumo citados.

**Figura 1** – Representação do consumo dos componentes de um NSSF.



**Fonte:** Akyildiz et al. (2002).

As RSSF comportam diversas formas de sensoriamento, podendo ser com baixa ou alta amostragem de dados. É possível programar o uso dos NSSF para realizar medidas contínuas ou somente para a percepção de algum fato, possibilitando também o acionamento de atuadores locais. A maneira de aplicação dos sensores e da forma de sensoriamento impacta diretamente no consumo de energia dos NSSF.

Quanto à comunicação exercida pelo módulo-rádio, responsável pela recepção e transmissão dos dados, deve-se considerar com cautela, pois os NSSF demandam grande parte de sua energia na recepção e na transmissão de dados. No processamento dos dados, levando em conta a contínua evolução dos

processadores no quesito eficiência energética, o consumo também torna-se fator crítico.

O consumo energético é diferente nos estados de funcionamento dos NSSF. Constata-se que a comunicação, ou o processo de transmissão dos dados, gasta mais energia que o sensoriamento e o próprio processamento dos dados sensorizados (ANASTASI et al., 2009). Outros aspectos como arquitetura de rede, dimensão da rede, tipo de população de nós sensores, a proporção do engendramento de informações de sensoriamento, a energia disponível na fase inicial de uso da bateria de cada sensor e os protocolos de comunicação, afetam diretamente a vida útil ou *time to live* de uma RSSF (FUNG & LIU, 2008).

### 2.3 *Energias renováveis*

O crescimento populacional desenfreado e não planejado, aliado ao uso e exploração dos recursos naturais, geraram dependência humana de energias fósseis, resultando em impactos ambientais de grande porte e muitas vezes irreversíveis no cotidiano atual. Cerca de 80% das necessidades energéticas do mundo são supridas com o uso de combustíveis fósseis – petróleo, gás natural, carvão. A utilização dos combustíveis fósseis incentivou economias industriais, tornando-se indispensável para diversas ações produtivas e da vida moderna atual. Tais indícios ocorrem (à) uma realidade ainda com grande participação dos combustíveis fósseis, pelo menos pelas próximas décadas (ARÇO et al., 2011).

As ações antrópicas ocorridas ao longo dos últimos anos resultaram no aumento de destaque da sustentabilidade energética, em particular a partir de 1980 graças ao Relatório de Brundtland, que propõe desenvolvimento sustentável como “[...] aquele que atende às necessidades do presente sem comprometer a possibilidade de as gerações futuras atenderem às suas necessidades [...]”. No contexto de sustentabilidade, enquadra-se o quesito de geração de energia limpa, renovável e sustentável. (BRUNDTLAND; ONU, 1987).

A Agência Internacional de Energia (IEA) descreve que as energias renováveis são todas aquelas provenientes de processos naturais que em geral são regularmente renovados, mesmo que alguns deles necessitem de um período de tempo maior para se regenerar, gerando assim controvérsias sobre o assunto.

Definem-se como “energias renováveis” os métodos de obtenção de energia direta ou indiretamente pelo sol ou quaisquer formas de extração de energia a partir do vento, biomassa, recursos hidrológicos e geotérmicos. (INTERNATIONAL ENERGY AGENCY, 2010).

### 2.3.1 Energia solar

Castro (2002) define que o Sol é a fonte energética mais farta da Terra. Além de ser cinco vezes maior que a soma de todas as outras fontes energéticas e irradiar dez mil vezes a energia consumida por toda a população no mesmo período de tempo, a sua temperatura de superfície (6000°C) o torna um reator capaz de transformar quatro milhões de toneladas por segundo de hidrogênio em hélio.

A radiação solar constitui-se numa fonte de energia abundante. Além de possuir grande capacidade de utilização através de outros sistemas de captação e conversão de energia, a radiação solar tem papel fundamental na responsabilidade pela manutenção da vida terrestre (LION F. & QUEIROZ, 2007).

O potencial solar do Brasil é significativo e promissor em quase todo o ano. O aproveitamento dessa energia, tanto como fonte de luz quanto de calor, enquadra-se hoje numa das opções energéticas mais promissoras, e ecologicamente corretas, para geração de energia elétrica e solução de problemas derivados da alta demanda de consumo (ALDABÓ, 2002).

Dentre as formas de geração de energia através de recursos solares, a utilização de painéis fotovoltaicos se enquadra nas necessidades de alimentação de NSSF em RSSF.

A tecnologia solar fotovoltaica utiliza células fotovoltaicas, constituídas por semicondutores, para realizarem a conversão direta dos fótons de luz em eletricidade, utilizando células solares. A energia luminosa emanada pelo Sol ou por qualquer outra fonte de luz é transformada em energia elétrica através dessas células (REN21, 2016).

A tecnologia de células fotovoltaicas modernas teve início em meados de 1954, após o *Bell Telephone Laboratories* adotar definitivamente o silício como mineral semicondutor e, através de estudos e experimentos, atingir uma eficiência de conversão da luminosidade solar de 4%. Em 1958, a tecnologia envolvendo

silício passa a ser considerada a mais adequada para aplicações espaciais. Os satélites *Vanguard* I e II, *Explorer* III e *Sputnik* III foram desenvolvidos e lançados utilizando painéis fotovoltaicos, com células solares de silício, em suas estruturas. A energia fotovoltaica é vantajosa, pois utiliza o recurso mais abundante na Terra, a radiação solar, fazendo com que essa tecnologia facilite o fornecimento de energia elétrica para áreas remotas (ARÇO et al., 2011).

O fenômeno de conversão fotovoltaica foi descrito em 1839 por Becquerel, que relatou a geração de uma tensão elétrica (*Volts*) entre as duas faces (positiva e negativa) de um material condutor após a absorção de luz no ambiente, dando origem à primeira tecnologia de célula fotovoltaica (unidade primordial na conversão de energia solar em elétrica). Para fabricação dos painéis fotovoltaicos, o material semiconductor mais utilizado é o silício, que pode ser monocristalino, policristalino ou de silício amorfo de junção dupla ou tripla (FUNDAÇÃO ECOLÓGICA NACIONAL, 2013). Cada material oferece uma porcentagem de conversão de energia (Tabela 3):

**Tabela 3** – Tipos de silício e fator de conversão.

<b>Tipo de Tecnologia de Silício Utilizada</b>	<b>Fator de Conversão</b>
Monocristalino	15% a 18%
Policristalino	12,5%
Silício Amorfo	5% a 7%
Amorfo Dupla/Tripla Junção	8% a 10%

**Fonte:** Adaptado de Fundação Ecológica Nacional (2013).

A manipulação e a utilização do silício nos painéis fotovoltaicos é definido por Amerlis (2001) *apud* SILVA (2014) como um processo de dopagem do silício em forma pura e é feito utilizando fósforo (para induzir uma carga negativa) e boro (para cargas positivas). Quando a luz solar é incidida sobre a superfície do painel, os fótons que o integram colidem-se com os elétrons da estrutura do silício, gerando energia através da diferença de potencial causada pelas polaridades da célula fotovoltaica.

### 3 MATERIAIS E MÉTODOS

O projeto do sistema de monitoramento ambiental proposto ocorreu em duas fases.

A Fase 1 corresponde à definição de plataforma de comunicação, *software*, *hardware* e confecção da estrutura do NSSF, onde foca-se somente em testes de funcionamento do NSSF e esgotamento de bateria.

A Fase 2 do projeto abrange *upgrade* de *software* e *hardware*, a utilização de sensores, testes com o sensor de umidade do solo, medidas de RSSI (*Received Signal Strength Indicator* – indicador da potência do sinal), definição e utilização de sistema supervisorio e testes de esgotamento de energia da bateria de longa duração em ambientes externos.

#### 3.1 Fase 1

O *setup* prático inicial de testes resume-se a uma rede formada por uma estação de trabalho (microcomputador), onde está conectado, via USB, o programador *UartSBee* e nele acoplado o módulo *BE990* com antena mono polo. Este *setup* será a estação base na RSSF, atuando sob controle do *software* gerenciador em *Python*, comunicando-se com um NSSF.

Na Fase 1 do projeto, após desenvolvimento do *software* gerente e *firmware*, definiram-se os endereços de NSSF e base, canal de comunicação e outras configurações padrões referentes à plataforma *Radiuino*. Foram então realizados testes de comunicação entre NSSF e base em distâncias aproximadas de 6 metros, tanto com o *Radiuino Test* quanto com o próprio *Python*.

Os testes executados na Fase 1 ocorreram em ambientes controlados, pois o intuito era apenas a verificação de funcionalidade do *SleepMode*.

Pode-se definir o ciclo de funcionamento do *setup* do sistema na Fase 1 baseado na Figura 4.

**Figura 4** – Visão sistêmica da aplicação do sistema na Fase 1.



A Figura 4 pode ser descrita da seguinte forma:

1. *Software* gerenciador em *Python* hospedado em uma estação de trabalho gera tarefas e as envia para a base em forma de solicitação;
2. O módulo rádio *BE990* acoplado à estação base converte as informações solicitadas e as transmite em um pacote via radiofrequência;
3. O NSSF recebe as instruções solicitadas pelo gerente, executa-as e após isso envia outro pacote como resposta à base;
4. A estação decodifica os dados do pacote recebido e os envia para o *software* gerenciador;
5. O *software* gerenciador interpreta os dados, disponibiliza uma visualização dos mesmos e armazena automaticamente as informações num *log* de texto em formato *.txt*.

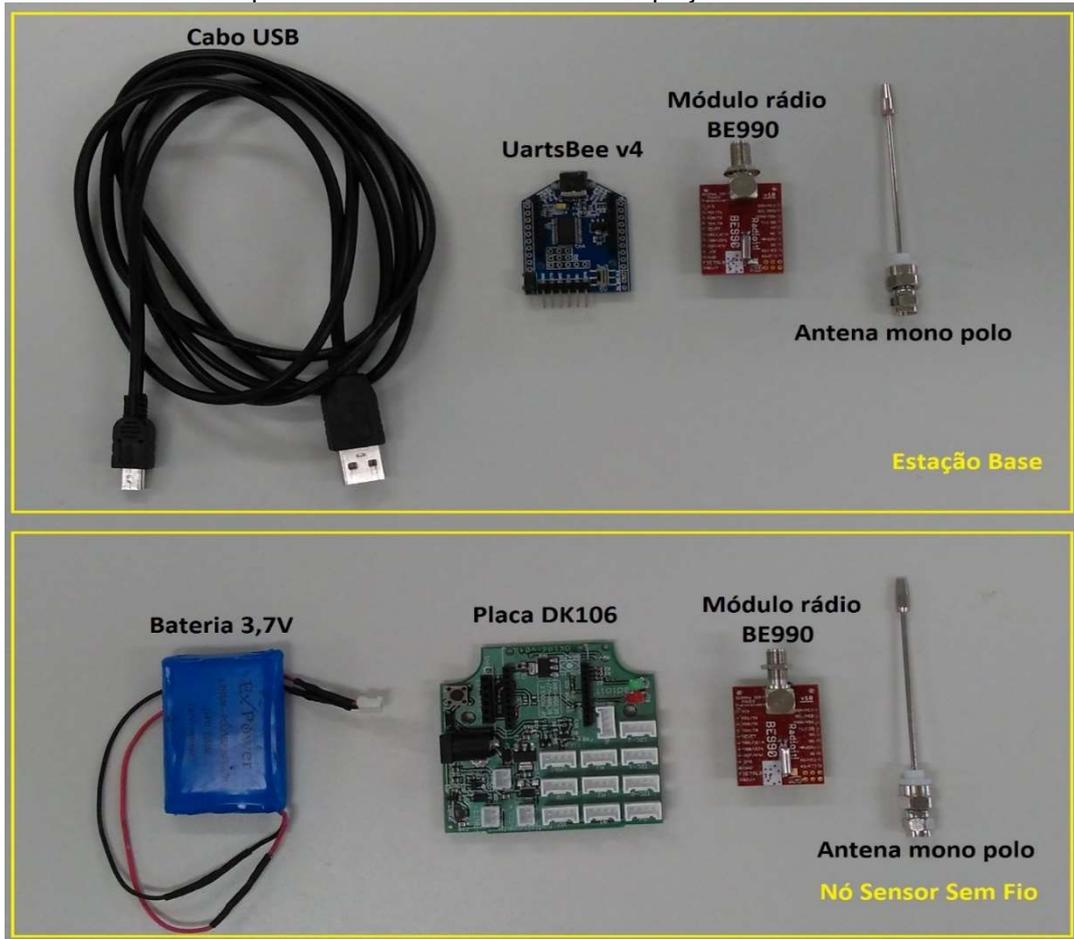
### 3.1.1 Nó sensor sem fio

O NSSF é constituído por cinco componentes além do protocolo *Radiuino*, escolhido como plataforma de comunicação. O que difere o *setup* das duas fases do projeto é o acréscimo dos sensores para mensuração das grandezas nos testes em ambiente externo. Segue caracterização e imagem (Figura 5) dos componentes:

- NSSF – Composto pela placa de desenvolvimento *DK106*, módulo rádio *BE990*, bateria do tipo polímero de íon de lítio de 3.7V e Antena Mono Polo  $\frac{1}{4}$  de  $\lambda$  (*lambda*).

- Base – Módulos gravador *Uartsbee* e rádio *BE990* para recepção e transmissão de dados gerados e informações advindas do sistema gerenciador. A base fica conectada a uma estação de trabalho operada por *Windows 7* de 64 *bits*, onde um *software* gerenciador atuará.

**Figura 5** – Foto dos componentes utilizados na Fase 1 do projeto.



Foi aplicada a função *SleepMode* 1 como ferramenta principal para melhoria da eficiência energética do NSSF visando o uso inteligente da energia disponível e consequentemente aumentando o *time to live*.

### 3.1.2 Placa de desenvolvimento DK106

A placa de desenvolvimento *DK 106* é produzida e comercializada pela empresa *Radioit*. Esta placa possui o acoplamento para rádio compatível com diversos módulos RF disponíveis no mercado, funcionando como uma extensão

para acesso aos pinos do rádio, possibilitando o aumento de capacidade do NSSF e agregando componentes essenciais para uso de energias renováveis devido sua entrada para painéis fotovoltaicos.

O dispositivo pode ser alimentado por fonte de alimentação direta, bateria ou painel solar fotovoltaico. O controle inteligente de carga fica sob responsabilidade do controlador de carga *MCP73831-2*, desenvolvido pela *Microchip Technology*, para aplicações que façam uso de baterias do tipo polímero de íon de lítio. O controlador suporta uma tensão máxima de entrada de até 15V por fonte externa, até 6V via painel fotovoltaico ou baterias de até 4,5V. Ele também possibilita um carregamento de carga de até 500 mAh e possui um sistema de proteção de descarga completa, já que as baterias de íons de lítio correm o risco de não funcionarem, caso ocorra uma descarga total.

O *DK106* possui conectores tipo *Grove* para que os sensores possam ser conectados à placa com facilidade e sem a necessidade de soldas. Os conectores tipo *Grove* possuem 4 pinos, em que 2 deles estão reservados para o sinal, 1 para a alimentação VCC (corrente contínua) e outro para GND (terra). A placa *DK106* possibilita a conexão de até 10 sensores diferentes, desde que os sensores trabalhem nas plataformas analógica, digital ou I2C. Um dos conectores, denominado UART, é reservado para programações avançadas no *bootloader* referentes ao hardware do *DK106*.

O diagrama elétrico da placa está disponível conforme Anexo A, e a organização dos pinos destinados as plataformas está alocada conforme a Tabela 4:

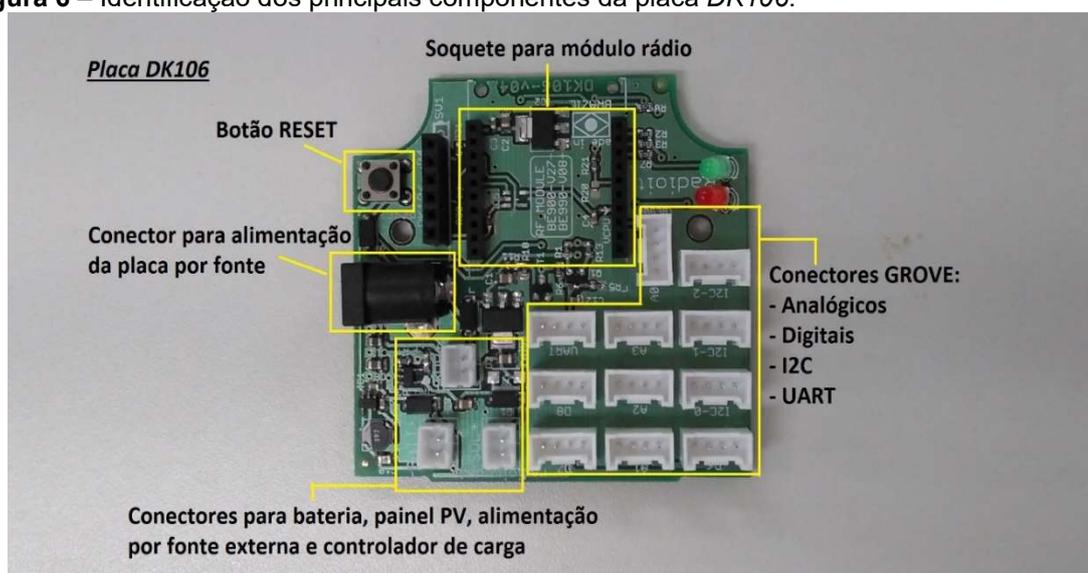
**Tabela 4** – Pinos disponíveis na placa *DK106*.

	Analógico	Digital	I2C	UART
PINOS	4	3	3	1

**Fonte:** Adaptado de *Radioit* (2017).

Os pinos dos conectores *Grove* correspondentes ao sinal ficam em contato direto com o módulo rádio *BE990* através de trilhas na placa *DK106*. Os principais componentes da placa *DK106* são identificados na Figura 6:

**Figura 6** – Identificação dos principais componentes da placa *DK106*.



### 3.1.3 Módulo de radiofrequência *BE990*

O *BE990* (Figura 7) foi desenvolvido com o transceptor de RF *CC1101*, ajustado para operar na banda de 915MHz (902-907,5MHz e 915-928MHz), que trabalha diretamente em conjunto com um amplificador de RF, o TI *CC1190*, possibilitando uma comunicação confiável em grandes distâncias (*indoor* até 1000m e *outdoor* até 8000m em linha de visada).

Optou-se pelo *BE990* pois a frequência de transmissão em 915MHz é mais adequada para ambientes com vegetação e sua potência de transmissão possibilita a conexão dos dispositivos em longas distâncias. A frequência de 2,4 GHz não é adequada devido à atenuação do sinal causada pelas plantas, por serem materiais ricos em água (TERUEL *et al.*, 2012).

**Figura 7** – Módulo de radiofrequência *BE990*.



As dimensões da placa e as especificações do transceptor podem ser conferidos conforme a Tabela 5:

**Tabela 5** - Especificações técnicas do transceptor de RF e dimensões do *BE990*.

<b>RF</b>	
<b>Frequência de Operação</b>	902-907,5 MHz e 915-928 MHz
<b>Modulação</b>	2FSK(Configurável)
<b>Tecnologia</b>	Modulação Digital
<b>Taxa de dados do RF</b>	Até 250Kbps
<b>Potência de TX</b>	Até +17dBm
<b>Sensibilidade de RX</b>	Até -112dBm (com ~1% de PER)
<b>Alcance Indoor</b>	Até 1000m
<b>Alcance Outdoor</b>	Até 8000m
<b>Regulamentação</b>	Atende à Anatel, FCC, Austrália

<b>Placa</b>	
<b>Dimensões</b>	24,4mm x 32mm x 10,5mm
<b>Número de Pinos</b>	20
<b>Espaçamento dos Pinos</b>	2mm
<b>Conector de RF (opcional)</b>	SMA Reto ou SMA 90°
<b>Pinos de E/S</b>	E/S digitais, UART, I2C, SPI, ADC, PWM

**Fonte:** *Radioit* (2017).

O *BE990* também é equipado com o microcontrolador *Atmega328*, de 8 *bits* com baixo consumo e alto desempenho; e atende à regulamentação de funcionamento para radiofrequência no Brasil, exigido pela ANATEL.

A potência máxima de operação é 17 dBm, taxa de RF é 4,8 kbps e o canal escolhido para operação é o 2. A Tabela 6 mostra as características do microcontrolador.

**Tabela 6** – Detalhes técnicos do microcontrolador *Atmega 328*.

<b>Microcontrolador</b>	
<b>Memória</b>	32KB Flash, 2KB RAM, 1KB EEPROM
<b>Clock da CPU</b>	8MHz
<b>RTC (Relógio de tempo real)</b>	32768KHz (+/- 10ppm)
<b>Conversões AD</b>	Até 6 canais de 10 bits
<b>Saídas PWM</b>	2
<b>Entradas/Saídas Digitais</b>	Até 4 entradas ou saídas

Fonte: *Radioit* (2017).

A Tabela 7 mostra as especificações elétricas do módulo RF *BE990* e a Tabela 8 disponibiliza a relação dos pinos do *BE990* com a plataforma *Arduino*:

**Tabela 7** – Especificações elétricas do módulo RF *BE990*.

<b>Especificações elétricas</b>	<b>Min</b>	<b>Tip.</b>	<b>Max</b>	<b>Un.</b>
<b>Tensão de Entrada (VCC)</b>	3,0	3,3	3,6	V <sub>DC</sub>
<b>Tensão de Entrada (VCPU)</b>	VCC	3,3	5,0	V <sub>DC</sub>
<b>Corrente de transmissão</b>		310,3		mA
<b>Corrente de Recepção</b>		27,2		mA
<b>Corrente de Inatividade</b>		5,2		mA
<b>Corrente de Dormência</b>		< 0,3		mA
<b>Temperatura de Operação</b>	-50		125	°C

Fonte: *Radioit* (2017).

**Tabela 8** – Relação dos pinos do *BE990* com a plataforma *Arduino*.

<b>Pino</b>	<b>Nome</b>	<b>Tipo</b>	<b>Função no ATmega328</b>	<b>Pino do Arduino</b>
1	3V3	Entrada alim.	VCC (3.3V)	-
2	PD1/TX	Entrada/Saída	TXD (Pino de saída da USART), PCINT17	1 (E/S digital)
3	PD0/RX	Entrada/Saída	RXD (Pino de entrada da USART), PCINT16	0 (E/S digital)
4	PD4/T0	Entrada/Saída	XCK (Clock externo da USART), T0 (entrada externa do Timer/Counter 0), PCINT20	4 (E/S digital)
5	/RESET	Entrada	Pino de reset	-
6	PB1/OC1A	Entrada/Saída	OC1A (Saída da Comparação A do Timer/Conter1), PCINT1	9 (E/S digital)
7	PB0/ICP1	Entrada/Saída	ICP1 (Entrada de captura do Timer/Counter1), CLK0 (Saída do clock do sistema), PCINT0	8 (E/S digital)
8	PD7/AIN1	Entrada/Saída	AIN1 (Entrada negativa do comparador analógico), PCINT23	7 (E/S digital)
9	DTR	Entrada	Para a programação do ATmega328	-
10	GND	Terra	Terra	-
11	A3/PC3	Entrada/Saída	ADC3, PCINT11	3 (Entrada Analógica/ 17 (E/S digital)
12	A2/PC2	Entrada/Saída	ADC2, PCINT10	2 (Entrada Analógica/ 16 (E/S digital)
13	NC	-	Não Conectada	-
14	VCPU	Entrada	Alimentação opcional para o microcontrolador (requer jumper SJ1 em aberto)	-
15	NC	-	Não Conectada	-
16	A7	Entrada	ADC7	7 (Entrada Analógica)
17	T1/PD5	Entrada/Saída	T1 (Entrada externa do Timer/Counter 1), OC0B (Saída da Comparação B do Timer/Counter0), PCINT21	5 (Entrada Analógica/
18	AIN0/PD6	Entrada/Saída	AIN0 (Entrada positiva do comparador analógico), OC0A (Saída da Comparação A do da Comparação A do Timer/Counter0), PCINT22	6 (E/S digital)
19	A5/PC5	Entrada/Saída	ADC5 (Entrada do ADC canal 5), SCL (linha de clock do barramento serial), PCINT13	5 (Entrada Analógica/ 19 (E/S digital)
20	A4/PC4	Entrada/Saída	ADC4 (Entrada do ADC canal 4),SDA (linha de dados do barramento serial), PCINT12	4 (Entrada Analógica/ 18 (E/S digital)

Fonte: *Radioit* (2017).

As configurações de características do módulo rádio são determinadas no desenvolvimento do *firmware*. Um importante ponto a ser ajustado é o *offset* dos módulos. O *offset* é a diferença de frequência entre um valor obtido e o valor da frequência de referência. O valor de ajuste do *offset* é geralmente fornecido pelo fabricante, porém é possível encontrar o valor de *offset* utilizando um analisador de espectro para radiofrequência.

#### 3.1.4 Bateria íon de lítio 3,7V – 1800 mAh e 3600 mAh

Para a execução do trabalho, optou-se primeiramente por utilizar uma bateria recarregável do tipo polímero de íon de lítio, devido à compatibilidade da placa *DK106* com esse tipo de bateria. Além disso, as baterias de íon de lítio apresentam certas vantagens entre as outras baterias disponíveis no mercado, como as confeccionadas com hidreto metálico de níquel ou níquel cádmio. Dentre as vantagens podem ser citadas:

- Ausência do efeito memória, ou seja, não há a necessidade de carregar a bateria até o total de sua capacidade e descarregar até o mínimo possível;
- Suportam centenas de ciclos de carga/descarga sem perder quantidades significativas de rendimento;
- Tamanhos reduzidos com rendimentos elevados, devido à alta reatividade do lítio e suas ligações atômicas.

Algumas desvantagens das baterias de lítio englobam medidas a serem tomadas na Fase 2, onde os testes serão realizados *outdoor* e com presença frequente do calor solar. Tais desvantagens são:

- Extrema sensibilidade a temperaturas elevadas;
- Grande possibilidade de não funcionamento caso a carga seja descarregada por completo (vertente onde o controlador de carga da placa *DK106* atua como dispositivo de proteção para esse tipo de bateria, mantendo sempre uma pequena quantidade de carga disponível e evitando que a mesma se esgote por completo);

A bateria possui dimensões de 34mm x 10mm x 50mm e pesa aproximadamente 32 gramas, ela possui conector *PCB JST-PHR2* e suporta uma tensão máxima para recarga de até 4.2V (Figura 8).

Para a Fase 1 do projeto foi utilizada a bateria recarregável com 1800 mAh, já para a Fase 2 do projeto foi utilizada a bateria de mesmas especificações, porém com 3600 mAh.

**Figura 8** – Bateria do tipo polímero de íon de lítio 3,7V – 1800 mAh.



### 3.1.5 Programador UartsBee v4

O *UartSBee* v4, produzido e comercializado pela *SeedStudio*, é um adaptador USB para Serial e pode ser usado para programação da *firmware* nos rádios ou se comunicar com aplicações. Ele é compacto e equipado com o soquete próprio para os rádios utilizados na plataforma *Radiuino* (20 pinos 2,0 mm). Ele possui o chip *FT232* integrado e funciona nas tensões 3.3V e 5V, tensão que a maioria dos módulos-rádio trabalham. (Figura 9).

**Figura 9** – Programador *UartsBee* v4.



**Fonte:** *SeedStudio* (2017).

### 3.1.6 Antena mono polo $\frac{1}{4} \lambda$ (lambda).

O módulo *BE990* opera em 915MHz e nele é possível acoplar uma antena mono polo de  $\frac{1}{4} \lambda$  (Figura 10) de 8 cm que corresponde a antena do tipo *wire* fornecido junto ao *BE990*.

**Figura 10** – Antena mono polo  $\frac{1}{4} \lambda$ .



### 3.1.7 Estrutura de suporte do NSSF.

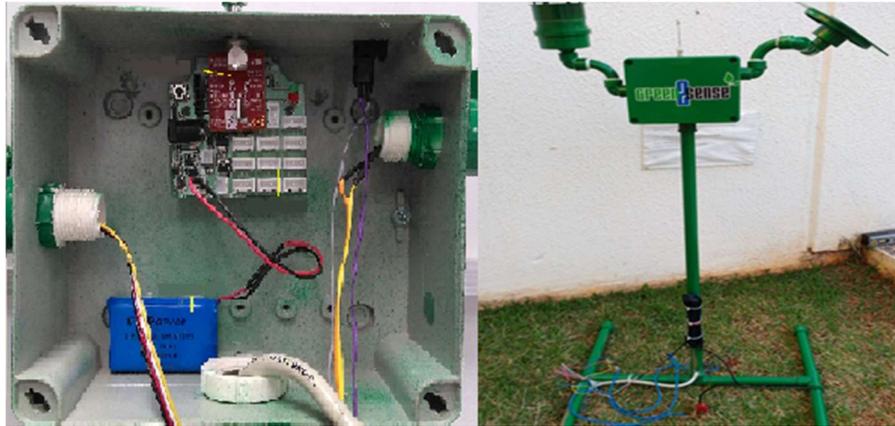
O projeto optou desde o princípio pela utilização de materiais de baixo custo, em especial para o desenvolvimento da estrutura que sustentará o dispositivo. Foi escolhido o material plástico PVC para a estrutura, devido seu custo acessível e fácil manipulação (ou manutenção). Os tubos de PVC foram preenchidos com cimento e argamassa afim de proporcionar maior peso e melhor sustentação, vide Figura 11.

**Figura 11** – Preenchimento da estrutura em PVC com cimento e argamassa.



Uma caixa hermética, também plástica, foi adaptada para acoplar o circuito, rádio e bateria, sendo que todos os orifícios foram vedados para garantir que a umidade do ar ou chuvas não danifiquem o sistema. A cor verde-floresta foi adotada no intuito de camuflar o dispositivo nas aplicações em ambientes externos da Fase 2 e foi utilizada cola adesiva plástica para tubos de PVC rígido para colar as conexões (Figura 12).

**Figura 12** - Caixa hermética aberta mostrando o circuito interno e, a direita, estrutura montada.



### 3.1.8 Plataforma Arduino e Firmware

*Arduino* é uma plataforma de prototipagem de código aberto, baseado em *software* e *hardware* “easy-to-use”. A plataforma *Arduino* conta com uma IDE (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento - em português) para desenvolvimento da *firmware*.

A *firmware* é um *software* embarcado ou um conjunto de instruções operacionais programadas diretamente no *hardware* de um dispositivo eletrônico. A *firmware* é carregada no módulo rádio utilizando um programador *UartSBee v4*.

### 3.1.9 Plataforma Radiuino

A plataforma *Radiuino* (criada em 2011) é uma extensão do projeto *Arduino* com o intuito de agregar a este as funcionalidades de uma RSSF através da inclusão de módulos-rádio para o envio dos dados processados no microcontrolador.

Tal plataforma funciona com o envio de pacotes via radiofrequência em 915 MHz através de módulos-rádio. Esses pacotes sempre contém 52 *bytes*, sendo que cada *byte* possui uma função e uma relação direta com os pinos da placa onde

o microcontrolador está alocado. É nesse pacote também que colocamos os dados de identificação da base e do NSSF. Dessa forma é possível programar através da IDE *Arduino* um conjunto de instruções e enviá-las via radiofrequência para automação de ações, identificando o NSSF e a base. Os *bytes* de 16 a 33 estão associados às entradas/saídas analógicas do NSSF. Os *bytes* de 34 a 51 estão associados às entradas/saídas digitais. A Figura 13 ilustra o mapeamento do pacote de 52 *bytes* do *Radiuino* (*Radiuino*, 2017).

**Figura 13** – Mapa do pacote de 52 *bytes* da plataforma *Radiuino*.

Cabeçalhos das camadas	PhyHdr[#]				MACHdr[#]				NetHdr[#]				TranspHdr[#]					
Posição na variável do cabeçalho	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3		
Função de cada posição de cada cabeçalho	RSSI_D LINK	LQI_D LINK	RSSI_ULINK	LQI_UPL INK	Tempo Sleep	Tempo Sleep1**	Tempo Sleep2*	TBD	DST_ID Quem Recebe		SRC_ID Quem manda	SRC_NI D	COUNT	TBD	TBD	TBD		
Posição dos bytes no pacote	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Conversores AD	AD0[#]			AD1[#]			AD2[#]			AD3[#]			AD4[#]			AD5[#]		
Posição na variável dos AD	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
Função de cada posição de cada AD	AD0[0]	AD0[1]	AD0[2]	AD1[0]	AD1[1]	AD1[2]	AD2[0]	AD2[1]	AD2[2]	AD3[0]	AD3[1]	AD3[2]	AD4[0]	AD4[1]	AD4[2]	AD5[0]	AD5[1]	AD5[2]
Posição dos bytes no pacote	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
Entradas e saídas digitais	IO0[#]			IO1[#]			IO2[#]			IO3[#]			IO4[#]			IO5[#]		
Posição na variável dos IO	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
Função de cada posição de cada IO	IO0[0]	IO0[1]	IO0[2]	IO1[0]	IO1[1]	IO1[2]	IO2[0]	IO2[1]	IO2[2]	IO3[0]	IO3[1]	IO3[2]	IO4[0]	IO4[1]	IO4[2]	IO5[0]	IO5[1]	IO5[2]
Posição dos bytes no pacote	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51

**Fonte:** *Radiuino* (2017).

### 3.1.10 Função *SleepMode*

Os módulos rádio possuem três estados de funcionamento: recebendo, transmitindo ou dormindo, sendo o último estado pouco utilizado em aplicações envolvendo radiofrequência - RF.

A função *SleepMode* é um processo de dormiência do microcontrolador que desativa alguns dos processos internos do módulo rádio. Existem diversos níveis de dormiência que impactam diretamente na diminuição do consumo de energia.

As operações básicas desligadas no *SleepMode* são: conversão de analógico para digital, saídas digitais e *interfaces* externas. Uma base de temporização continua funcionando no módulo rádio durante o tempo de

dormência, para que a operação normal possa retornar ao estado ativo após um tempo determinado.

O acionamento da função *SleepMode* se dá através de uma avaliação do *byte 4* no estado 0 ou 1. O estado 1 induz o módulo radio a dormência e o estado 0 determina a ativação ou “acordar”. Na plataforma *Radiuino*, a função *SleepMode* está disponível em duas versões: *SleepMode* e *SleepMode2*.

Na função *SleepMode*, quando o *byte 4* está no estado 1, são lidos os *bytes 5 e 6* e convertidos para segundos, definindo assim o tempo de dormência do módulo rádio. O NSSF opera através de comandos da estação base.

A Figura 14 representa o consumo de dois módulos RF num ciclo de funcionamento do módulo RF em 900 000 milissegundos, onde entende-se:

- RX = Estado recebendo;
- TX = Estado transmitindo;
- *SleepMode* = Estado dormindo.

**Figura 14** – Representação dos estados de funcionamento do módulo RF.



### 3.1.11 Radiuino Test

O *RadiuinoTest* é um *software* de mapeamento e simulação de pacotes da plataforma *Radiuino*. Com ele é possível visualizar *byte por byte* e suas respectivas funções, simular um pacote e enviá-lo para um NSSF através uma base conectada a um microcomputador via USB e verificar se os pacotes estão sendo transmitidos e recebidos sem perda. O *RadiuinoTest* é *open source*, a Figura 15 mostra a interface de trabalho do *software*.

Figura 15 – Interface de trabalho do software *RadiuinoTest*.



### 3.1.12 Python

O *setup* padrão de testes funcionará sob o controle de um *software* gerenciador, que administra o ciclo de funcionamento do NSSF e é programado em *Python*. *Python* é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por *Guido van Rossum* em 1991.

Optou-se pela linguagem *Python* (versão 2.7.13) devido a sua plataforma ser extremamente robusta para aplicações que envolvem RSSF e por ser *open source software*. Mantém como prioridade a legibilidade do código acima da velocidade ou expressividade. Mistura uma sintaxe concisa e clara com recursos robustos de sua biblioteca padrão e por módulos e *frameworks* desenvolvidos por desenvolvedores ao redor do mundo (*Python Brasil*, 2017).

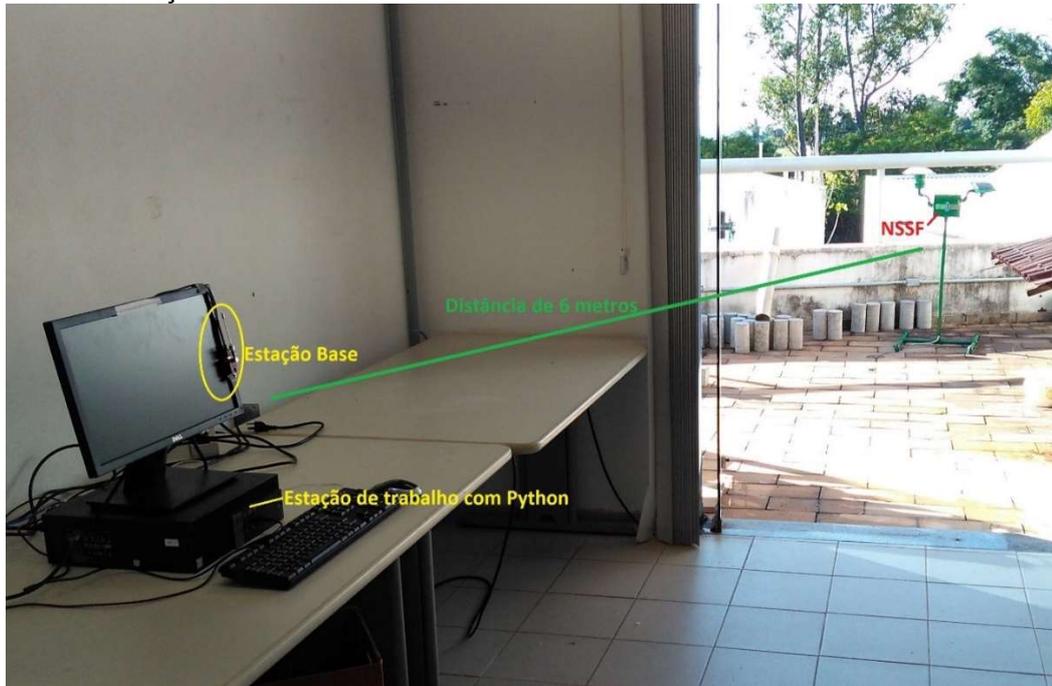
### 3.1.13 Testes em ambiente protegido

Uma das etapas da Fase 1 destina-se à utilização e comprovação de eficiência da função *SleepMode* da plataforma *Radiuino*. Para tanto, definiram-se os padrões do *setup* e em especial da tensão de referência (*Volts*) inicial da bateria utilizada, pois a comprovação de eficiência será dada através do esgotamento de carga da bateria.

Dois ciclos de testes foram executados, ambos com o *setup* físico e cenário idênticos, porém em um dos ciclos de testes foi utilizada a função *SleepMode* na *firmware* e no outro apenas a *firmware* padrão de comunicação,

objetivando a comparação entre os dois NSSF. Os ciclos iniciaram-se com zero segundos e terminaram somente quando a bateria apresentou carga insuficiente para alimentar o NSSF. A Figura 16 representa o cenário da infraestrutura utilizada nos testes.

**Figura 16** – Ilustração do cenário de testes.



As características definidas englobam tensão de referência inicial e final da bateria de íon de lítio, canal de comunicação, *offset* dos rádios, distância entre estação base e NSSF e número de medidas, conforme pode ser visualizado na Tabela 9.

**Tabela 9** – Padrões de configuração para o cenário de testes.

Características	NSSF	NSSF +
		<i>SleepMode</i>
Frequência	915,4 MHz	915,4 MHz
Canal	2	2
OFFSET Base	12	12
OFFSET NSSF	12	12
Distância entre base x NSSF	6 m	6 m
Tensão inicial da bateria	3.90 V	3.90 V
Tensão final da bateria	2.80 V	2.80 V
Tempo <i>SleepMode</i>	0 s	40 s
Medidas Solicitadas	10000	10000

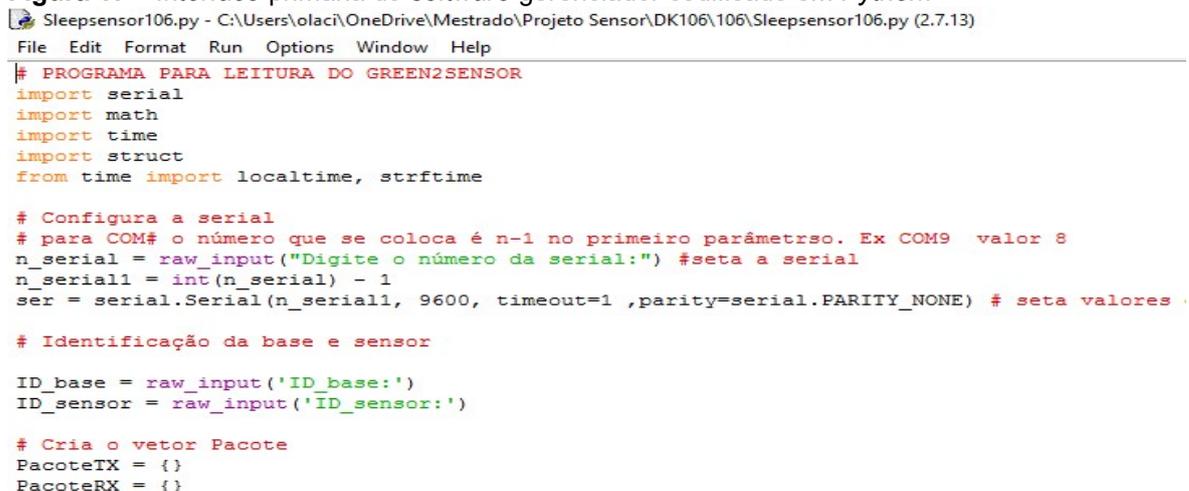
Optou-se por 40 segundos pelo tempo de dormência em *SleepMode* para que os testes fossem finalizados num prazo de tempo aceitável.

Primeiramente iniciou-se a bateria de testes de esgotamento de carga com o NSSF padrão, ou seja, sem a aplicação do *SleepMode*. Conectou-se a estação base à estação de trabalho e o NSSF foi posicionado a seis metros da estação base. Foi utilizada uma fonte de alimentação externa para carregamento da bateria de lítio do NSSF até a tensão desejada, no caso 3,90V.

Como anunciado anteriormente, todo o controle da aplicação é feito pelo *software* gerenciador codificado em *Python*. Esse código foi desenvolvido para trabalhar com o *setup* descrito e possibilita a escolha entre as opções de medida comum, *SleepMode* e *SleepMode 2*.

A Figura 17 possibilita um melhor entendimento da *interface* inicial da aplicação em *Python*.

**Figura 17** – *Interface* primária do *software* gerenciador codificado em *Python*.



```
Sleepsensor106.py - C:\Users\olaci\OneDrive\Mestrado\Projeto Sensor\DK106\106\Sleepsensor106.py (2.7.13)
File Edit Format Run Options Window Help
# PROGRAMA PARA LEITURA DO GREEN2SENSOR
import serial
import math
import time
import struct
from time import localtime, strftime

# Configura a serial
# para COM# o número que se coloca é n-1 no primeiro parâmetro. Ex COM9 valor 8
n_serial = raw_input("Digite o número da serial:") #seta a serial
n_serial1 = int(n_serial) - 1
ser = serial.Serial(n_serial1, 9600, timeout=1, parity=serial.PARITY_NONE) # seta valores

# Identificação da base e sensor

ID_base = raw_input('ID_base:')
ID_sensor = raw_input('ID_sensor:')

# Cria o vetor Pacote
PacoteTX = {}
PacoteRX = {}
```

A Figura 17 ilustra todo o ambiente de desenvolvimento do código *Python*. Nessa janela é possível realizar edições ou acréscimos de novas funcionalidades nos dispositivos. Com o *Python* é possível gerenciar todos os dispositivos, plataformas e ferramentas utilizadas numa RSSF. Para que a aplicação seja executada, é pressionada a tecla F5, onde uma nova janela (chamada *Shell*) para a execução da aplicação é aberta, conforme Figura 18.

**Figura 18** – Interface Shell de execução da aplicação em *Python*.

```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afaf1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\olaci\OneDrive\Mestrado\Projeto Sensor\DK106\106\Sleepsensor106.py 4
Digite o número da serial:4
ID_base:0
ID_sensor:1
Medição de grandezas do sensor autonomo:
0 - Mede pacote normal:
1 - Mede sleep nativo:
2 - Mede sleep2:
s - Para sair:
Comando:0
Entre com o número de medidas = 10000
Arquivo de log: Coleta_de_dados_2017_06_01_11-12-36.txt
Thu Jun 01 11:12:36 2017 0 | Tensão da Bateria = 3.90 V | Temperatura = 0.00 °C | Umidade =
: -61.50 DBM
Thu Jun 01 11:12:36 2017 1 | Tensão da Bateria = 3.90 V | Temperatura = 0.00 °C | Umidade =
: -64.50 DBM
Thu Jun 01 11:12:37 2017 2 | Tensão da Bateria = 3.90 V | Temperatura = 0.00 °C | Umidade =
: -64.00 DBM
```

Analisando a Figura 18 entende-se que:

- A área 1 (em vermelho) é destinada à definição do número da porta serial a que a estação base está conectada ao computador, o número de identificação da estação base e o número de identificação do NSSF na rede. No momento há apenas um NSSF, ele será por padrão rotulado como 1 e a estação base como 0.
- A área 2 (em verde) corresponde a seleção do modo de medida a ser executada. O código *Python* disponibiliza para essa aplicação três opções de medidas, além da opção “Sair”. Neste trabalho serão abordadas apenas as opções 0 e 1 do respectivo código. A opção 0 – Mede pacote normal ativa o modo de medição padrão da plataforma *Radiuino* que é baseada em uma medição (pacote) por segundo. A opção 1 – Mede *sleep* nativo ativa a função *SleepMode* e quando selecionado disponibiliza um campo para determinar o tempo (em segundos) de dormência do módulo rádio nos intervalos entre o envio dos pacotes.
- A área 3 (em amarelo) é reservada para estimarmos o número de medidas que serão realizados pelo NSSF.
- A área 4 (em azul) mostra o nome do arquivo de *log* gerado. O arquivo de *log* armazena em formato *.txt* todos os dados mensurados pelos sensores, vindos dos pacotes enviados pelo NSSF. O código *Python* foi desenvolvido para nomear

automaticamente os arquivos de *log* com a data e hora do início das leituras. Ele coleta as informações sobre data e hora do computador local ou estação de trabalho, dessa forma é necessário verificar a data e hora do sistema operacional antes do início da aplicação. Os arquivos de *log* são salvos na pasta raiz onde a aplicação *.py* está armazenada.

O código *Python* também foi pensado para utilizar as funções específicas da placa *DK106*, como o cálculo da tensão de referência atual e disponível da bateria. Essa variável servirá como base para representar a bateria carregada e descarregada. Entende-se como descarregada o estado da bateria em que não há mais energia disponível para alimentar a transmissão dos pacotes do módulo rádio.

Para definir o estado descarregado da bateria, aplicaram-se três sequências de testes, que enviavam pacotes pelo NSSF a cada segundo, e foi observado que em ambas as vezes o NSSF parou de responder, quando a bateria atingiu 2,80V na medição fornecida pela placa *DK106*.

Para recarga da bateria utilizou-se uma fonte externa onde o carregamento completo representava 3,90V na informação fornecida pelo *DK 106*. Dessa forma, foram padronizados os valores máximo e mínimo, ou seja, bateria carregada (3,90V) e bateria descarregada (2,80V) baseando-se nos testes descritos.

O *log* de texto gerado a cada execução da aplicação fornece, além da data (dia da semana, mês, ano e horário início da aplicação), uma sequência padronizada em linha de variáveis selecionadas, exemplificada no recorte de texto a seguir, extraído de uma simulação teste:

```
"Tue May 16 11:53:29 2017 8 | Tensão da Bateria = 3.93 V |  
Temperatura = 0.00 °C | Umidade = 1.00 % | VG = 805.00 | VWC = 305.70 %  
| RSSID: -60.50 DBM | RSSIU: -52.50 DBM"
```

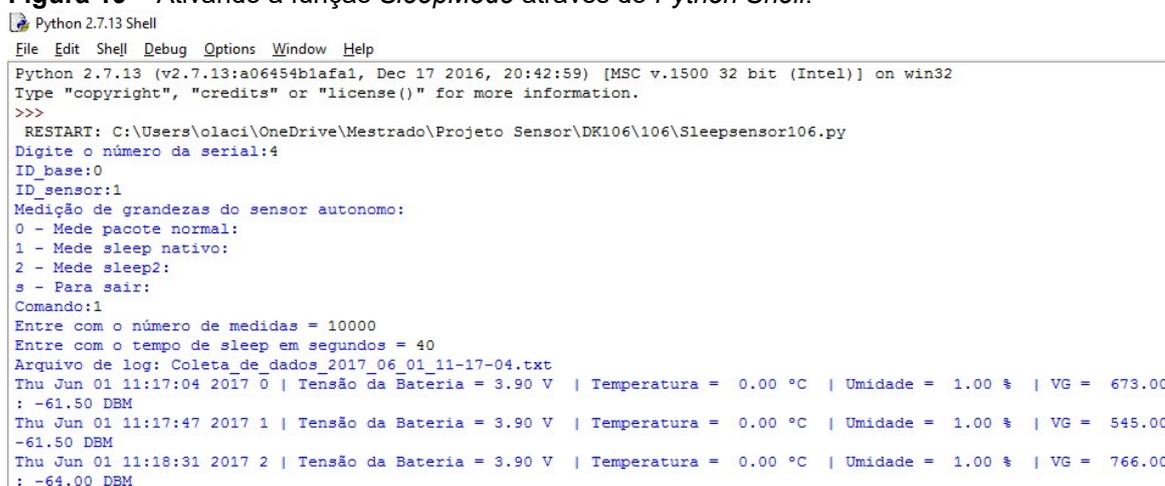
O *software* gerenciador foi programado para suportar as duas etapas do projeto. Na primeira etapa foi usada apenas o primeiro elemento da sequência (tensão da bateria). As variáveis temperatura, umidade, *VG*, *VWC* (*Volumetric*

*Water Content* ou Coeficiente Volumétrico de Água), *RSSID (RSSI Download)* e *RSSIU (RSSI Upload)* correspondem a segunda etapa do projeto proposto.

Após a definição das informações no campo 1, selecionou-se a opção 0 no campo 2, e ajustou-se o número de medidas em 10000 para simular o ciclo de testes de esgotamento de bateria do NSSF sem a função *SleepMode*, como mostrado na Figura 18.

Todo o procedimento foi repetido alterando as opções do *software Python* para ativar a função *SleepMode* no NSSF, conforme mostra a Figura 19.

**Figura 19** – Ativando a função *SleepMode* através do *Python Shell*.



```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\olaci\OneDrive\Mestrado\Projeto Sensor\DK106\106\Sleepsensor106.py
Digite o número da serial:4
ID_base:0
ID_sensor:1
Medição de grandezas do sensor autonomo:
0 - Mede pacote normal:
1 - Mede sleep nativo:
2 - Mede sleep2:
s - Para sair:
Comando:1
Entre com o número de medidas = 10000
Entre com o tempo de sleep em segundos = 40
Arquivo de log: Coleta_de_dados_2017_06_01_11-17-04.txt
Thu Jun 01 11:17:04 2017 0 | Tensão da Bateria = 3.90 V | Temperatura = 0.00 °C | Umidade = 1.00 % | VG = 673.00
: -61.50 DBM
Thu Jun 01 11:17:47 2017 1 | Tensão da Bateria = 3.90 V | Temperatura = 0.00 °C | Umidade = 1.00 % | VG = 545.00
-61.50 DBM
Thu Jun 01 11:18:31 2017 2 | Tensão da Bateria = 3.90 V | Temperatura = 0.00 °C | Umidade = 1.00 % | VG = 766.00
: -64.00 DBM
```

Analisando a Figura 19, seleciona-se a opção 1 – *Mede sleep nativo*, onde seguimos o padrão de dez mil medidas e definimos o tempo de *SleepMode* em 40 segundos.

Todos os ciclos foram aplicados nas dependências do laboratório de Pesquisa em Sistemas de Radiofrequência da PUC Campinas, mantendo sempre o mesmo padrão de *setup*, configurações e distância.

### 3.2 Fase 2

Após a finalização da Fase 1, deu-se início à Fase 2 do projeto, onde a mesma engloba a utilização de parte do *Kit IoT*, além da plataforma *Zabbix*.

Serão descritos os componentes e programas que sofreram alterações em relação à Fase 1, detalhamento da função *SleepMode 2* e posteriormente os desafios encontrados na execução dos testes *outdoor*.

Quase todas as especificações de *hardware* foram mantidas para a execução dos testes em ambientes externos. Alterou-se a bateria de lítio (de 1,8 Ah para 3,6 Ah) e acrescentaram-se os sensores de umidade de solo e nível de água (*Vegetronix VH400*) e umidade e temperatura do ar (*DHT22*).

Para a monitoração inicial da área (Área 1) de interesse, foram selecionadas as grandezas de umidade do ar, temperatura do ar e umidade do solo. O sistema ainda permite acoplar a ele mais sensores, possibilitando a mensuração de um número maior de grandezas, conforme o tipo e a necessidade da monitoração.

O gerenciamento de rede nesta fase também consiste em um algoritmo programado em *Python*, que solicita todas as requisições necessárias à RSSF e também realiza o *upload* dos dados para o servidor de aplicação.

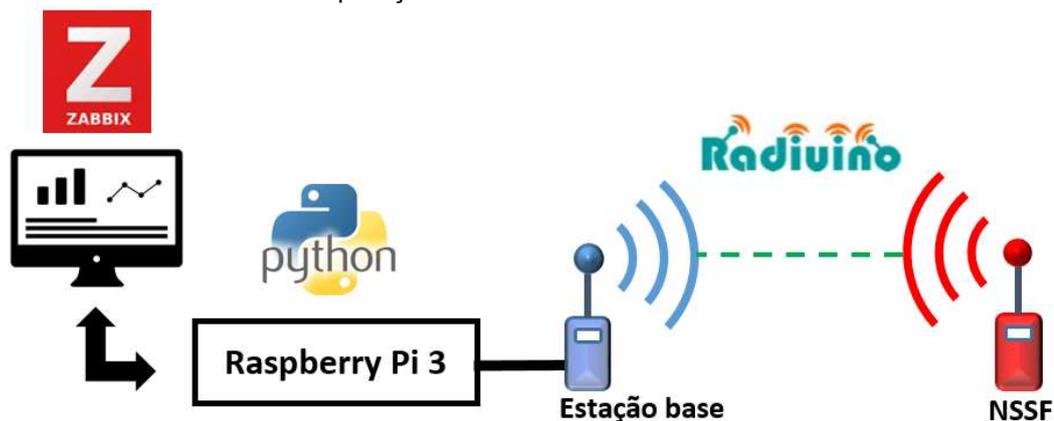
Para que os dados coletados pelo sistema de monitoramento não apresente valores incoerentes ou incorretos devido a problemas relacionados ao mau funcionamentos dos sensores, foi definido um tratamento dos dados coletados.

O *Python* só mostrará resultados finais das grandezas monitoradas que estiverem dentro do padrão (entre 2° C e 60° C para temperatura do ar, por exemplo), sobrescrevendo assim medidas que demonstrarem valores exacerbados, tanto para mais, quanto para menos.

O servidor de aplicação escolhido foi o *Zabbix*, onde nele foram disponibilizadas as informações de interesse e seus respectivos gráficos gerados em tempo real.

Pode-se definir o ciclo de funcionamento do *setup* do sistema na Fase 2 baseado na Figura 20.

Figura 20 – Visão sistêmica da aplicação na Fase 2.



Na Figura 20:

- 1- O script em *Python* envia uma solicitação para a estação rádio base;
- 2- A estação rádio base transmite o pacote de informação por radiofrequência;
- 3- O *NSSF* recebe a solicitação, acrescenta os dados dos sensores e retransmite o pacote de informação;
- 4- A estação rádio base recebe o pacote de informação e envia para o algoritmo em *Python*;
- 5- O algoritmo faz tratativas de dados e faz *upload* para o servidor *Zabbix*;
- 6- As informações são armazenadas no banco de dados do *Zabbix*; os usuários podem ter acesso por meio da *interface* gráfica (ou até mesmo por consulta ao banco de dados, caso o usuário necessite).

Para a Fase 2 do projeto, utilizou-se o *Kit IoT*, fornecido pela empresa *IoTtools*. Foi escolhido o *Kit IoT* devido à sua total compatibilidade com as plataformas utilizadas e também pela semelhança dos componentes que integram o *NSSF*. A Figura 21 disponibiliza uma visão completa do *Kit IoT*.

Figura 21 – Kit IoT.



Fonte: *IoTools* (2017).

O *Kit IoT* é composto por sensores de temperatura, *Raspberry Pi 3*, NSSF (rádio *BE990*, placa *DK106*, bateria de lítio 3.7 V / 3600 mAh, antena mono polo  $\frac{1}{4}$  de  $\lambda$ ), estação base (programador *Uartsbee*, módulo rádio *BE990*), roteador *Wi-Fi*, fontes de alimentação e suportes tri-pé.

Para a Fase 2, foram utilizados do *Kit IoT* o *Raspberry Pi 3* e toda a plataforma *Zabbix*, mantendo os componentes do NSSF da Fase 1 (exceto a bateria), já que os mesmos são semelhantes.

As principais vantagens da utilização da solução fornecida pela *IoTools* é a substituição da estação de trabalho convencional (computador comum) pelo *Raspberry Pi 3* (portátil) e pela integração do sistema físico com a plataforma *Zabbix*.

### 3.2.1 *Zabbix*

*Zabbix*, ferramenta *open source* e multiplataforma, utilizada para gerenciamento das infraestruturas de redes de computadores e também aplicações.

Sua principal vantagem é a facilidade de manipulação de objetos e ativos, onde todo o trabalho do dia a dia é agilizado pela não necessidade de repetição de processos que muitas vezes são necessários em outras aplicações.

Com ele é possível monitorar utilizando *templates* personalizados de monitoração e assim replicá-los para os fins necessários.

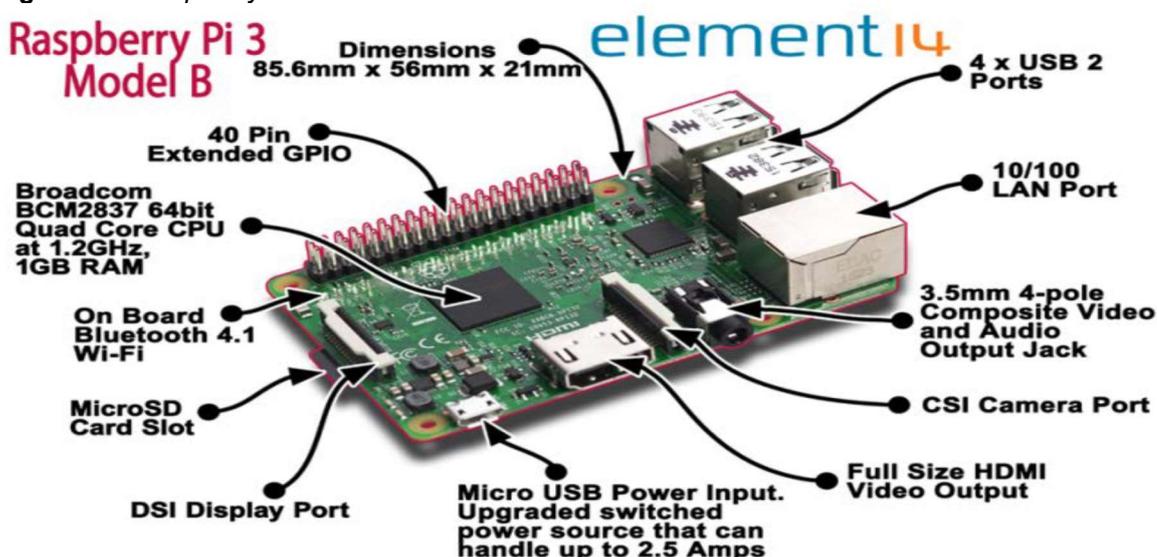
O *Zabbix* atuará como um agente de gerenciamento de todo o sistema, funcionando em ligação com o *Python*, ambos hospedados no *Raspberry Pi 3*.

### 3.2.2 Raspberry Pi 3

*Raspberry Pi* é um mini computador de placa única, possui entradas e saídas para que seja possível a utilização com periféricos, assimilando-se à um computador comum. Criado pela Fundação *Raspberry*, este microcomputador é composto em uma única placa onde todos os seus componentes são integrados. Na Figura 22 são mostrados os principais recursos da placa.

Tanto o servidor de aplicação quanto o algoritmo em *Python* foram hospedados no *Raspberry*, que funciona com sistema operacional *Linux* versão *Raspbian* de 64 bits.

Figura 22 - *Raspberry Pi 3*.



Fonte: *Raspberry* (2017).

O *Raspberry Pi 3 Model B* contém módulos de conexão *Wi-Fi* e *Bluetooth*, além de porta *HDMI*, 4 portas *USB 2.0* e processador de 1.2 GHz com 1 GB (*Gigabyte*) de memória *RAM* (*Random Access Memory*). Por não ter memória de armazenamento interno o microcomputador possui entrada para cartão *SD* expansível para até 128 GB.

### 3.2.3 Vegetronix VH400

O sensor *Vegetronix VH400* (Figura 23) é uma das melhores opções atualmente no mercado de sensores para medida de umidade e nível de água. Além de funcionar em diversas plataformas e linguagens, o *VH400* mede a constante dielétrica do solo utilizando a técnica de linhas de transmissão, sendo resistente a salinidade da água e não corroe ao longo do tempo, mesmo que submerso por longos períodos.

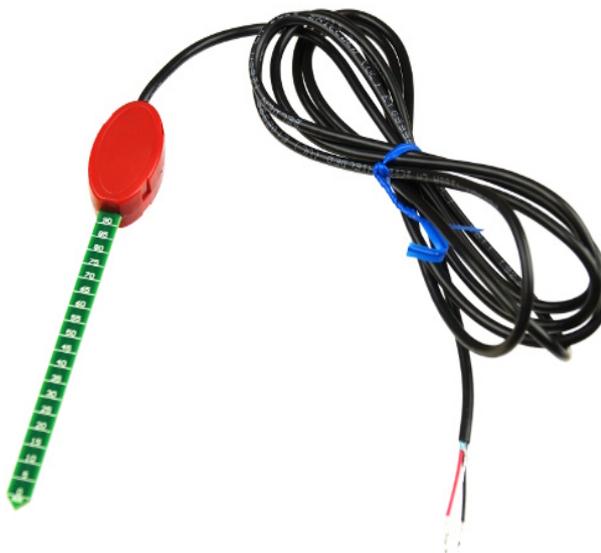
O fabricante aconselha que o sensor seja colocado no solo respeitando a profundidade mínima de 18 cm para monitoração de umidade.

Para a aferição exata do sensor, foi utilizado um medidor de bolso USB *VG Meter 200 Hand (A)*. O medidor em questão é um *datalogger* portátil que funciona com um *Vegetronix VH400* sem a necessidade de um computador ou microcontrolador, garantindo uma segunda base de consulta às medidas, bem como uma calibração mais exata do sensor.

Nesta fase do projeto, o *Vegetronix VH400* é o sensor que realiza as medidas das variáveis *VWC* e “Resposta analógica *VH400*” (valor de leitura puro do sinal analógico que, após um cálculo da *firmware*, converte a resposta analógica em *VWC*).

As especificações técnicas do sensor estão disponíveis na Tabela 9 e seus aspectos físicos na Figura 23.

**Figura 23** – Sensor de umidade desolo e nível de água *Vegetronix VH400*.



Fonte: *Vegetronix* (2017).

**Tabela 9** – Especificações técnicas *Vegetronix VH400*.

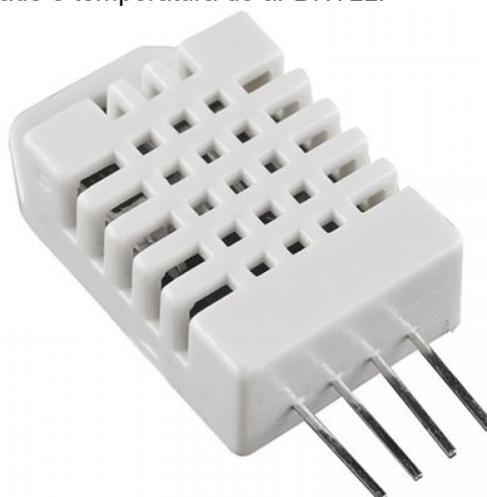
VH400 Sensor	
Power consumption	< 13mA
Supply Voltage	3.5V to 20 VDC.
Dimensions	See drawing below.
Power on to Output stable	400 ms
Output Impedance	10K ohms
Operational Temperature	-40°C to 85°C
Accuracy at 25°C	2%
Output	0 to 3V related to moisture content
Shell Color	Red
Voltage Output Curves	<a href="#">Curves</a> , <a href="#">Piecewise linear equations</a>
Certifications	<a href="#">CE Declaration of Conformity</a>

**Fonte:** *Vegetronix* (2017).

### 3.2.4 DHT22

O *DHT22* é um sensor de temperatura e umidade que possui 1 pino com saída digital e permite fazer leituras de temperaturas entre -40 a +80 graus *Celsius* e umidade entre 0 a 100%, compatível com *Arduino*, *Raspberry* e outros microcontroladores. A Figura 24 disponibiliza a visão do sensor e a Tabela 10 as especificações técnicas.

**Figura 24** – Sensor de umidade e temperatura do ar *DHT22*.



**Fonte:** *Sparkfun* (2017).

**Tabela 10** – Especificações técnicas do *DHT22*.

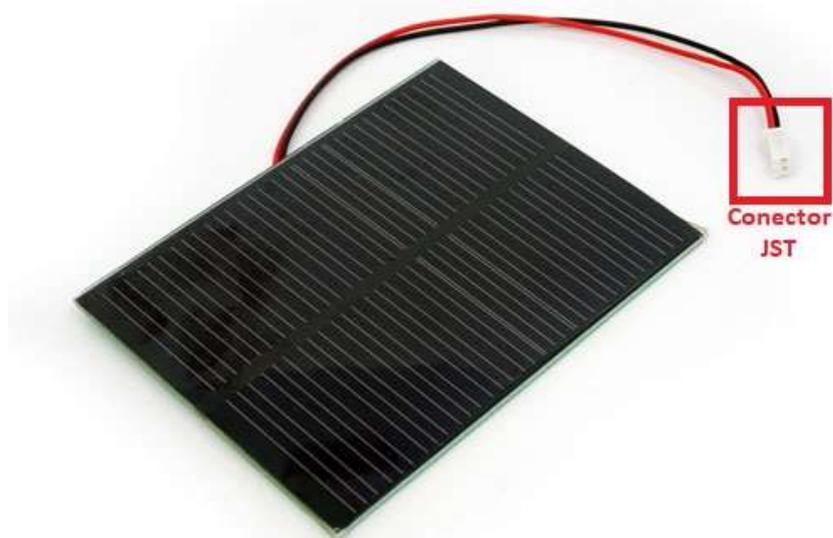
Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+/-0.3%RH
Long-term Stability	+/-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

Fonte: *Sparkfun* (2017).

### 3.2.5 Painel fotovoltaico 1W

O painel solar utilizado no projeto é feito de material monocristalino com eficiência em torno de 17%. Possui uma superfície fina de resina e traseiras resistentes, adequadas para ambientes ao ar livre. Um conector JST de 2mm é ligado ao painel, o que o torna perfeito para conectar-se à maioria dos dispositivos que fazem uso de energia fotovoltaica como fonte de alimentação. É possível a visualização do painel na Figura 25 e suas especificações na Tabela 12.

**Figura 25** – Painel fotovoltaico 1 W.



Fonte: *Seedstudio* (2017).

**Tabela 11** – Especificações técnicas do painel fotovoltaico de 1 W.

- • Dimensions: 100x80x2.5(±0.2) mm
  
- • Typical voltage: 5.5V
  
- • Typical current: 170mA
  
- • Open-circuit voltage: 8.2 V
  
- • Maximum load voltage: 6.4V

**Fonte:** Seedstudio (2017).

### 3.2.6 Função *SleepMode 2*

A função *SleepMode 2* foi desenvolvida objetivando uma autonomia maior do NSSF em uma RSSF. Na versão *SleepMode 2* os *bytes* 5 e 6 são convertidos em múltiplos de 15 segundos.

As mudanças no código foram feitas de modo a permitir a retro compatibilidade com códigos anteriores, ou seja, continua existindo o modo de *SleepMode* utilizado anteriormente na Fase 1, tendo sido mantidas tanto o código de acionamento como o código de resposta do mesmo.

O funcionamento do modo *SleepMode 2* acontece da seguinte maneira:

- No campo MACHdr[0] (*Byte* 4) deve ser enviado o valor “3” significando *sleepmode2*;
- No campo MACHdr[1] (*Byte* 5) deve ser mandado o tempo de *sleep* (T1) sendo ele em múltiplos de 15 segundos. Onde T1\*15 segundos (tempo máximo de dormência é de 63,75 minutos);
- No campo MACHdr[2] (*Byte* 6) deve ser enviado o tempo de espera em escuta ativa (T2) também em intervalos de 15 segundos;
- No campo MACHdr[3] (*Byte* 7) deve ser mandado o tempo de *sleep* caso não receba nenhum pacote na escuta ativa (T3), também em intervalos de 15 segundos.

Quando o NSSF recebe um pacote do tipo *SleepMode 2* o sensor irá responder com um ACK de *sleep2*, valor “4”, no campo MACHdr[0]. Em seguida o sensor entra em modo *sleepmode2*, da mesma forma que o *SleepMode* tradicional, pelo intervalo determinado em MACHdr[1] ou (T1).

Após finalizado o tempo de *sleep* (T1), o sensor permanecerá ligado em modo de recepção esperando por uma comunicação pelo tempo determinado em MACHdr[2] (T2).

Caso nenhuma comunicação seja recebida durante o tempo em modo de recepção (T2) o sensor entrará novamente em modo de *sleep* pelo período determinado em MACHdr[3] (T3).

Após o período de dormência (T3) o sensor volta novamente ao modo de recepção pelo período (T2).

O sensor ficará alternando entre os modos de recepção com período (T2) e dormência com período (T3) até o momento em que receber uma comunicação durante o modo de recepção. O período de dormência (T1) ocorre somente no início do processo de funcionamento do NSSF.

Se a comunicação enviada para o sensor for um pacote de requisição/acionamento normal (MACHdr[0] = "0") ou uma comunicação de *sleepmode* padrão (MACHdr[0] = "1") ele sairá do modo *SleepMode 2* e não voltará a dormir automaticamente até que seja requisitado que ele entre no modo *sleepmode2* novamente, requisição feita pelo código *Python*.

Para implementar o modo *SleepMode 2* foram executadas pequenas modificações no *firmware* dos módulos rádio que trabalham na plataforma *Radiuino*, onde todas podem ser visualizadas no Apêndice A.

### 3.2.7 Condições emergenciais

Visando uma aplicação futura para monitoração e mitigação de impactos causados por desastres naturais, agregou-se ao sistema de monitoramento a capacidade de reagir com base em condições hierárquicas definidas pelo usuário, tais condições podem estar relacionadas à qualquer variável que possa ser monitorada.

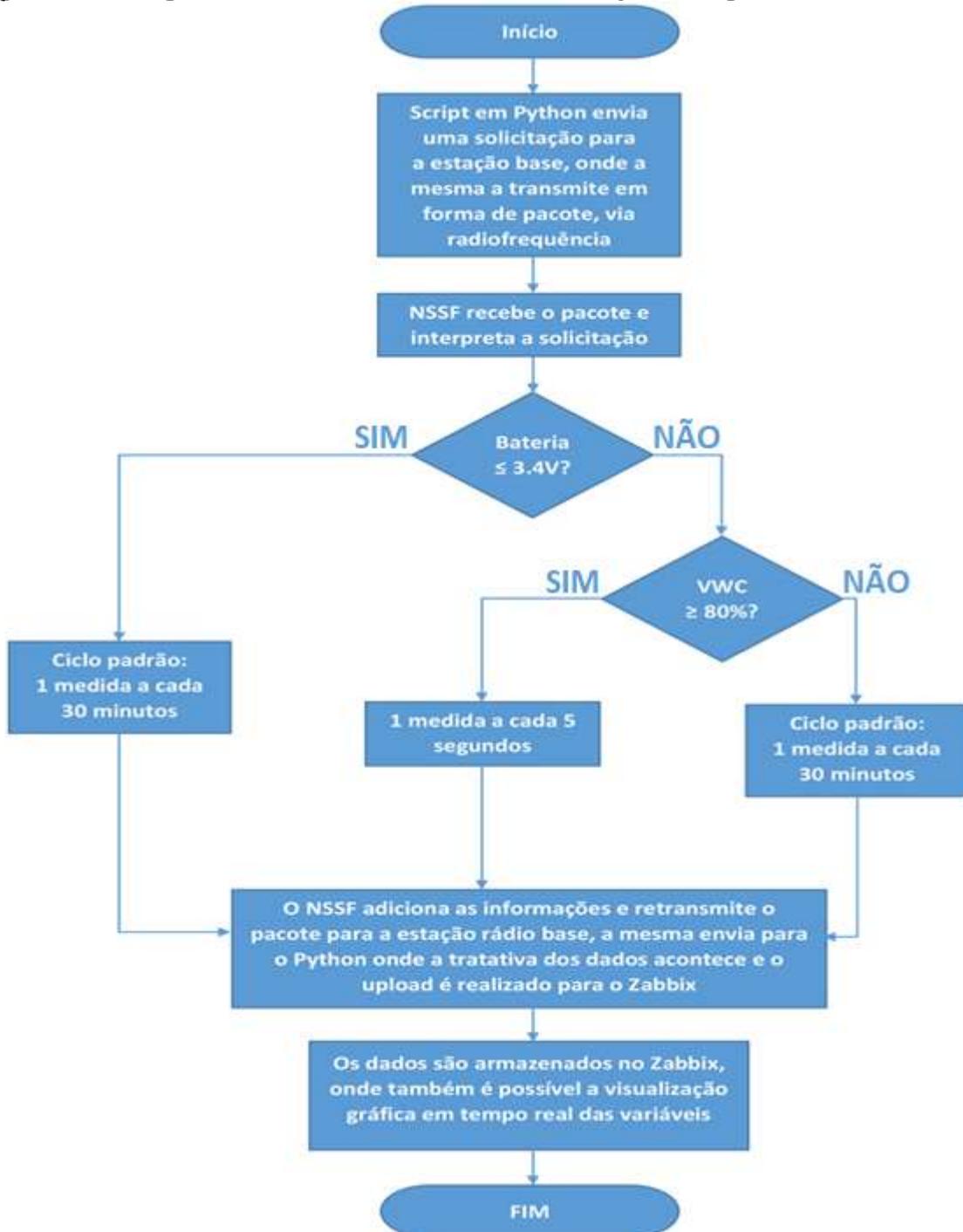
A princípio o foco constitui-se no coeficiente volumétrico que representa a porcentagem de umidade no solo (*VWC*), definindo que o sistema manteria seu ciclo normal de funcionamento (1 medida a cada 30 minutos) até o momento em que o *VWC* atingisse um valor superior a 80%, simulando um solo próximo a saturar devido ao excesso de umidade, fator de risco para deslizamentos e desabamentos. Caso o valor do *VWC* for igual ou superior a 80%, a função *SleepMode 2* é desativada, e o NSSF passa a fornecer 1 medida (ou 1 pacote) a cada 5 segundos.

Considerando o consumo de energia como fator crítico, estabeleceu-se que o valor de tensão de referência da bateria atingisse um valor menor ou igual a

3.4V, o NSSF assumiria o ciclo padrão com o *SleepMode 2*, fornecendo 1 medida a cada 30 minutos, independentemente do valor do VWC, definindo assim a hierarquia das condições emergenciais no sistema.

A Figura 26 disponibiliza um diagrama em blocos (fluxograma) do funcionamento das condições emergenciais no sistema.

**Figura 26** – Fluxograma do ciclo de funcionamento das condições emergenciais.



### 3.2.8 Testes em ambiente externo

Definiram-se, como áreas de interesse, dois pontos do bosque da PUC Campinas, onde um dos pontos (Área 1) estava próximo a uma nascente, com ambiente segregado e arborizado, e o outro ponto (Área 2) estava próximo ao estacionamento do CCHSA (Centro de Ciências Humanas e Sociais Aplicadas).

Foram estipulados 14 dias de testes ininterruptos, onde o NSSF monitorou a Área 1 por 7 dias e a Área 2 por mais 7 dias. As características técnicas do *setup* padrão para ambos os pontos englobam tensão de referência inicial (4.2 V) e final (2.8V) da bateria de lítio, distância entre estação base e os pontos das Áreas 1 e 2 (NSSF) e tempo de dormência, apresentados na Tabela 12:

**Tabela 12** – Características técnicas do *setup* de testes *outdoor*.

Características	Valores
Tensão inicial da bateria	4.20 V
Tensão final da bateria	2.80 V
Tempo de dormência	30 min.
Distância entre Área 1 e base	129,9 m
Distância entre Área 2 e base	190,2 m

O teste *outdoor* foi planejado para que as medidas de umidade e temperatura do ar e umidade do solo fossem coletadas a cada 30 minutos, e após cada coleta o NSSF fosse induzido ao estado de dormência pelo função *SleepMode 2*.

O projeto não possui um objetivo comparativo entre as grandezas medidas nos 2 locais, nem com o tratamento posterior dos dados coletados, já que o principal intuito é o funcionamento geral do sistema de monitoramento.

Na Figura 27 é mostrada a imagem das Áreas de interesse via satélite e nas Figura 28 sua respectiva distância entre NSSF e estação base (*Raspberry Pi 3*).

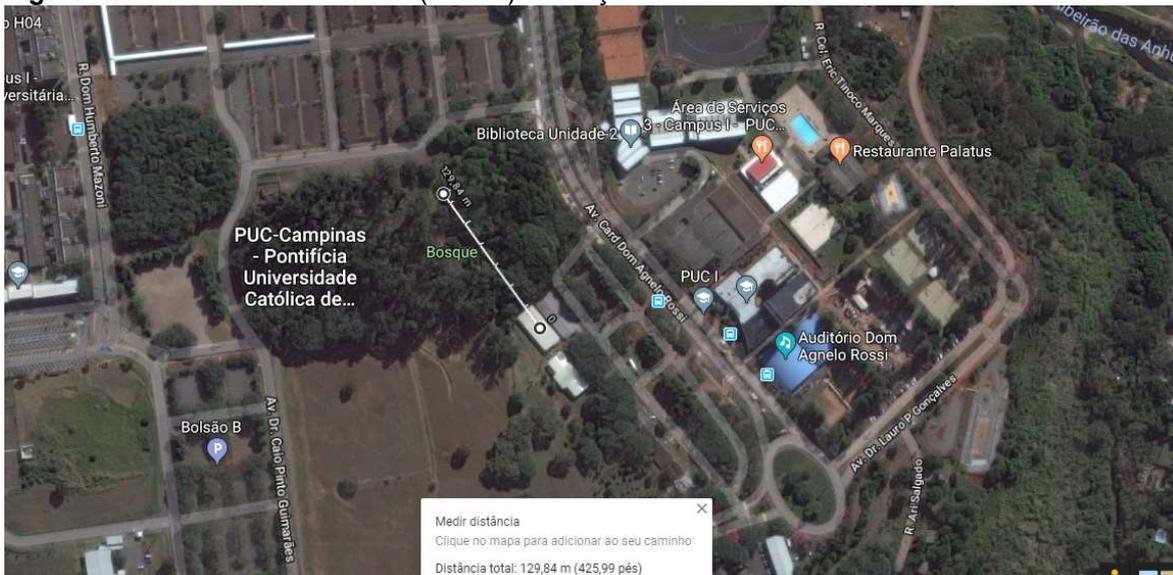
**Figura 27** – Imagem via satélite das áreas de interesse.



Fonte: Adaptado de *Google Maps* (2018).

A Área 1 estava aproximadamente à 130 metros de distância da estação base, o terreno estava plano entre os dois pontos nesse estágio, dessa forma somente algumas árvores e animais foram classificados como “obstáculos” na coleta e transmissão dos dados. A Figura 28 disponibiliza uma visão aérea da distância entre os pontos:

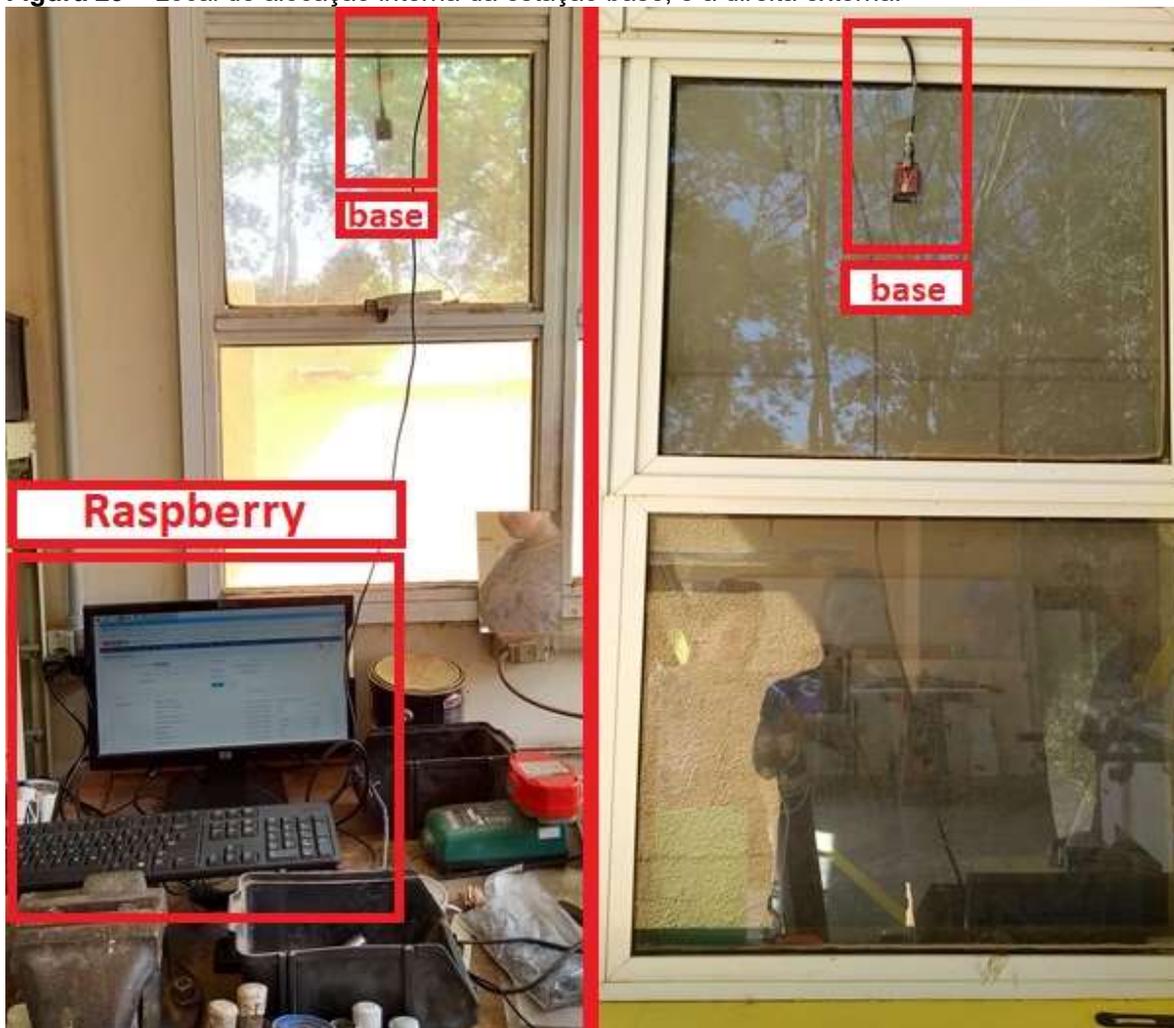
**Figura 28** – Distância entre NSSF (Área 1) e estação base.



Fonte: *Google Maps* (2018).

O *Raspberry Pi 3*, juntamente com todos os outros dispositivos que compõem a estação base, ficaram alocados no interior de uma oficina de reparos no Departamento de Manutenção da PUC Campinas, conforme a Figura 29:

**Figura 29** – Local de alocação interna da estação base, e à direita externa.



O local escolhido para abrigar a estação base foi estratégico pois situa-se próximo aos pontos monitorados, além de oferecer maior proteção aos equipamentos contra furtos ou danos ocasionados por chuva ou intervenções humanas.

Nas Figuras 30, 31, 32 e 33 é possível a visualização real da área de interesse (Área 1) com o NSSF já fixado no local.

**Figura 30** – Área de interesse (Área 1).



**Figura 31** – Área de interesse com NSSF já fixado (Área 1).



**Figura 32** – Sensor de umidade do solo *VH400* à 18 cm da superfície (Área 1).



**Figura 33** – Sensor de umidade do solo *VH400* instalado (Área 1).



O sensor de temperatura do ar (*DHT22*) possui materiais metálicos em sua composição, fazendo com que a incidência de radiação solar direta possa influenciar, para mais, as medidas de temperatura. Para isso, foi adicionado um prato plástico para proteção do sol, conforme visto na Figura 34.

**Figura 34** – Prato para proteção da radiação solar direta no sensor *DHT22* (Área 1).



Após 2 dias de testes foi constatada através dos valores de *RSSID* e *RSSIU*, uma diminuição na intensidade do sinal. Foi verificado no local que a estrutura que sustentava o NSSF havia cedido, fato ocorrido devido aos fortes ventos ou impacto de algum animal silvestre. Com a queda do NSSF, a antena direcional perdeu parte de seu alinhamento com a estação base, resultando em uma atenuação de sinal na transmissão.

Foi realizada a troca do suporte por um tripé, fixado com barras de ferro de aproximadamente 10 cm de comprimento, introduzidas no solo através de cavidades nas extremidades de cada pé (Figura 35).

**Figura 35** – Suporte cedido e, à direita, tripé substituído e já fixado (Área 1).



A configuração final do NSSF (também apelidado de *Green to Sense*) é mostrada na Figura 36, onde é possível também a visualização dos locais onde os dispositivos que compõem o NSSF estão alocados.

**Figura 36** – NSSF e seu aspecto final em campo (Área 1).



A Área 2 estava aproximadamente a 190 metros de distância da estação base, o terreno possui um declínio entre os dois pontos nesse estágio, dessa forma a diferença de altura entre os pontos e a grade de metal que cerca o estacionamento, em forma circular, foram classificados como os “obstáculos” na transmissão dos dados nesse ponto. Na Figura 37 é possível uma visualização aérea da distância entre a Área 2 e a estação base e, a Figura 38 disponibiliza uma visão dos “obstáculos” citados encontrados na Fase 2:

**Figura 37** – Distância entre NSSF (Área 2) e estação base.



Fonte: Google Maps (2018).

**Figura 38** – Grade de metal circular e declínio do solo na Área 2.



O ponto escolhido na Área 2 para alocação do NSSF pode ser visualizado na Figura 39:

Figura 39 – NSSF fixado na Área 2.



O teste teve início em 16/11/2017 (às 09h00) e término em 30/11/2017 (às 12h33), o tempo permaneceu nublado e com intensas pancadas de chuva durante 13 dos 14 dias de testes da Fase 2. O Sol apareceu por algumas horas apenas no dia de início do monitoramento. O painel fotovoltaico do NSSF foi alinhado em 23° ao Norte.

Todo o processo de monitoramento foi acompanhado utilizando a interface do Zabbix. Assim que o acesso é feito pelo *browser* e o *login* efetuado no endereço 127.0.0.1/zabbix, o menu de monitoramento é selecionado automaticamente e a aba *Dashboards* é aberta, onde é possível a visualização geral do sistema em atividade, o adionamento de itens favoritos e o *status* dos parâmetros gerais da aplicação, conforme a Figura 40.

Figura 40 – Interface inicial do Zabbix.

The screenshot shows the Zabbix web interface. The browser address bar displays '127.0.0.1/zabbix/zabbix.php?action=dashboard.view'. The navigation menu includes 'Monitoramento', 'Inventário', 'Relatórios', 'Configuração', 'Administração', and 'Extras'. The main dashboard area is titled 'Dashboard' and contains several sections:

- Gráficos favoritos:** Nenhum gráfico adicionado.
- Telas favoritas:** Nenhuma tela adicionada.
- Mapas favoritos:** Nenhum mapa adicionado.
- Status do Zabbix:** A table showing system parameters and their values.

Parâmetro	Valor	Detalhes
Zabbix está rodando	Sim	localhost:10051
Quantidade de hosts (habilitados/desabilitados/templates)	45	3 / 0 / 42
Quantidade de itens (habilitados/desabilitados/não suportados)	87	82 / 0 / 5
Quantidade de triggers (habilitadas/desabilitadas [incidente/ok])	45	45 / 0 [2 / 43]
Número de usuários (online)	3	1
Desempenho requerido do servidor, novos valores por segundo	0.88	

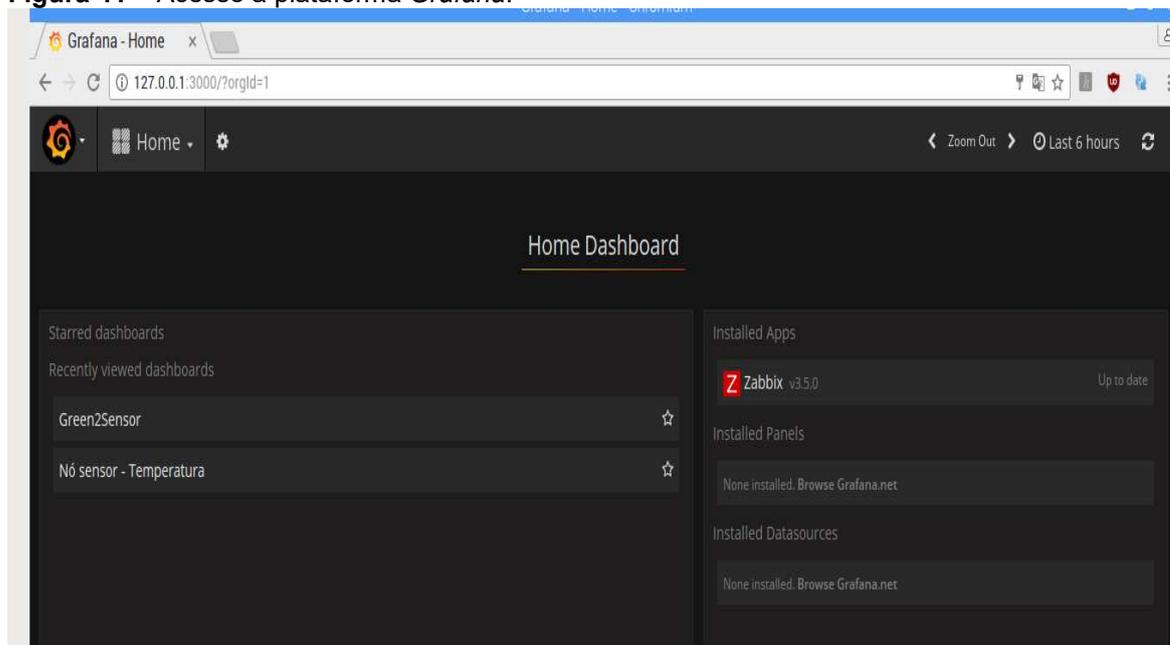
Atualizado: 09:51:05

Além da visualização dos dados através de números, foi escolhida a plataforma *Grafana*, que funciona como um *plugin* de *interface* gráfica mais completa, onde estão disponíveis diversos tipos de gráficos temporais e indicadores animados.

Os indicadores animados e gráficos temporais gerados no *Grafana* utilizam os dados fornecidos pelo servidor *Zabbix*. O indicadores animados mostram em tempo real as medidas das variáveis, onde também é possível determinar valores mínimos e máximos das variáveis e, caso um dos valores de mínimo ou máximo seja atingido, é possível ativar alertas visuais.

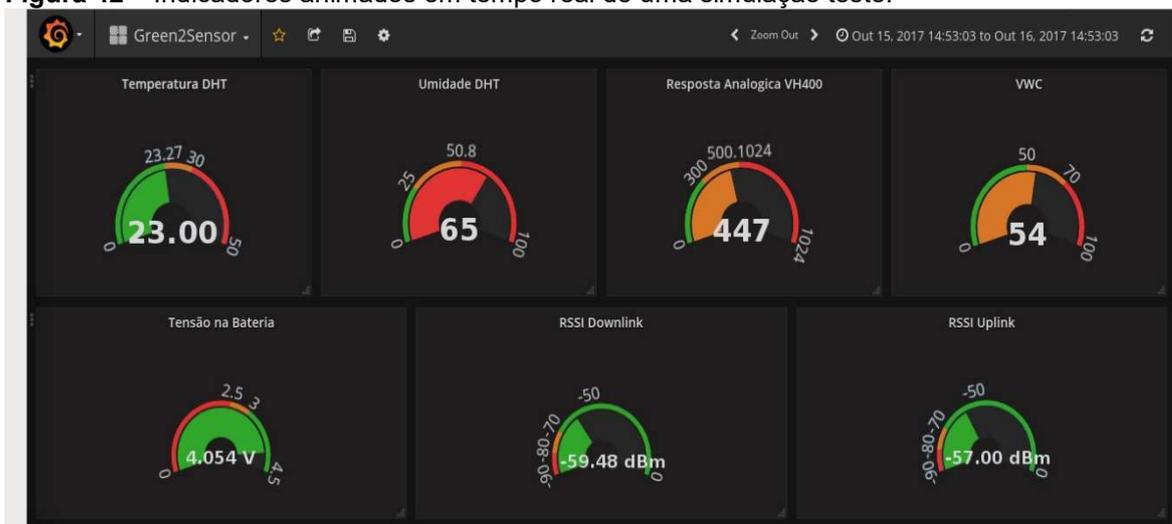
O acesso ao *Grafana* (Figura 41) é feito utilizando o endereço 127.0.0.1/3000, após efetuado o *login* a tela inicial é aberta, onde os *dashboards* extraídos do *Zabbix* ficam disponíveis. No projeto o *dashboard* selecionado é o *Green2Sensor*.

**Figura 41** – Acesso à plataforma *Grafana*.



Após selecionar o *dashboard* de interesse, são exibidos os indicadores animados padrão do *Grafana*, referentes aos valores das variáveis extraídas do servidor *Zabbix*. A Figura 42 disponibiliza uma visualização das variáveis em tempo real de uma simulação teste.

**Figura 42** – Indicadores animados em tempo real de uma simulação teste.



Dentre as dificuldades encontradas no decorrer da execução do projeto, destacam-se os testes embrionários iniciais em laboratório com um tipo de aplicação pioneira para o sistema de monitoramento, toda a logística e instalação do NSSF em campo e o extremo cuidado com a preservação da natureza durante todo o projeto.

Todos os dados coletados poderão ser disponibilizados na *internet*, utilizando um *modem 3G/4G* ou através da própria conexão local com a *internet*. A conexão local não foi possível devido aos padrões e políticas de segurança digital da PUC Campinas.

## 4 RESULTADOS

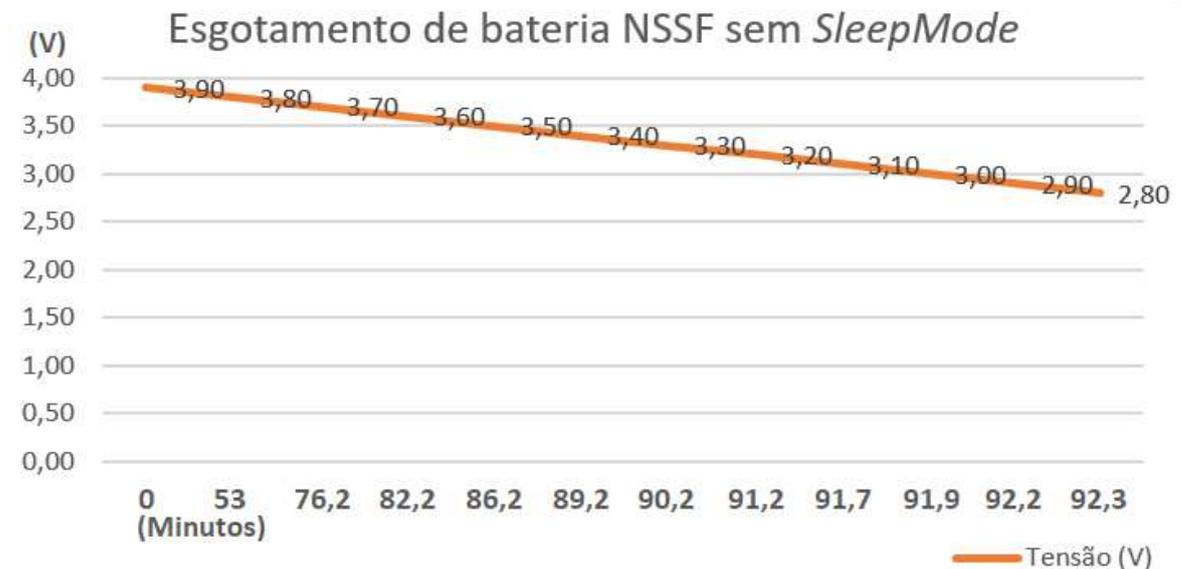
### 4.1 Fase 1

O objetivo da primeira fase de testes é a avaliação de funcionamento da RSSF com a função *SleepMode*, objetivando uma melhoria energética no consumo do NSSF para um aumento significativo do *time to live*.

As medidas de tensão de bateria ocorreram junto com o envio de cada pacote, porém para a análise experimental foram selecionados apenas os dados obtidos a cada 0,10V de queda. Considerando que a bateria forneceu um total de 1800 mAh em 92,2 minutos (ou 1 hora e 54 minutos), entende-se que o consumo médio de energia do NSSF foi de 19,49 mA, fornecendo um total de 5539 medidas.

A Figura 43 mostra o gráfico de esgotamento da bateria.

**Figura 43** - Gráfico do processo de descarga da bateria do NSSF sem a função *SleepMode*.



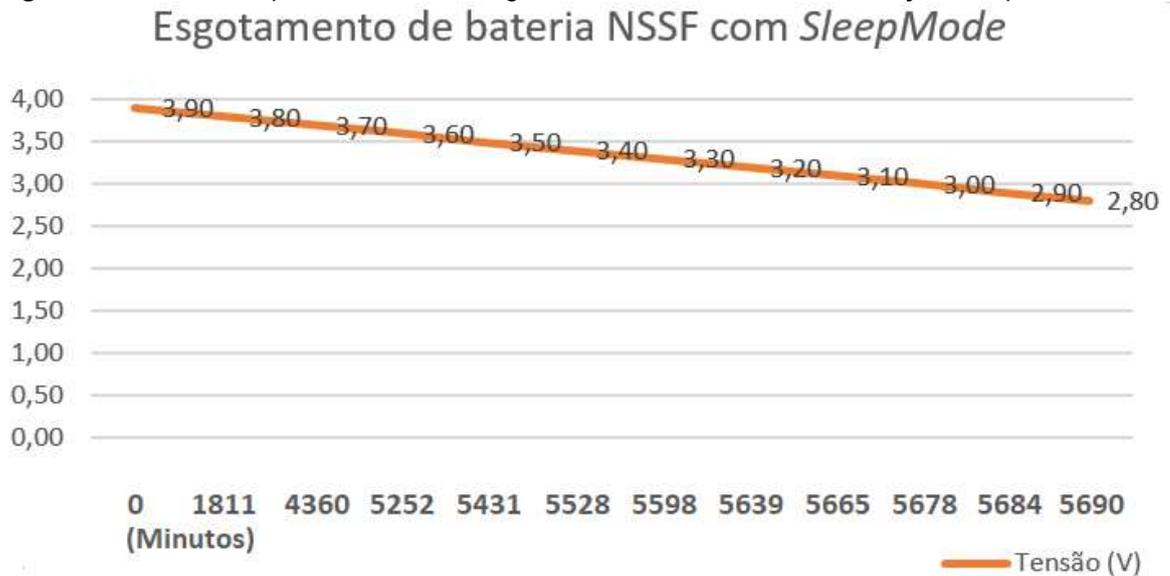
Para o NSSF com *SleepMode*, os valores colhidos totalizaram 8490 medidas em um prazo de 5690 minutos ininterruptos, ou 94 horas e 43 minutos.

As medidas de tensão de bateria também ocorreram junto com o envio de cada pacote, e selecionamos também apenas os dados a cada 0,10V de queda. A bateria forneceu um total de 1800 mAh em 5690 minutos (ou 94 horas e 43

minutos), considera-se que o consumo médio de corrente do NSSF foi de 0,316 mA.

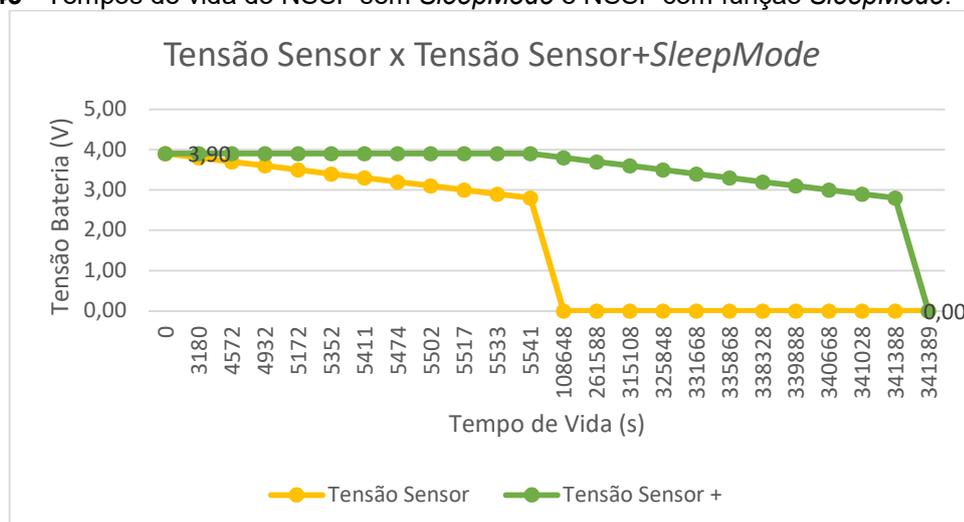
O esgotamento da energia elétrica disponível na bateria do NSSF com *SleepMode* pode ser visualizado na Figura 44.

**Figura 44** - Gráfico do processo de descarga da bateria do NSSF com a função *SleepMode*.



Para que fosse possível a conclusão prévia dos resultado, foi realizada uma comparação entre as variáveis dos dois *setups*, começando pelos valores de tensão *versus* tempo de vida, ou *time to live*. A Figura 45 mostra a ilustração gráfica do processo de descarga dos dois NSSF, em segundos, para maiores detalhes.

**Figura 45** - Tempos de vida do NSSF sem *SleepMode* e NSSF com função *SleepMode*.

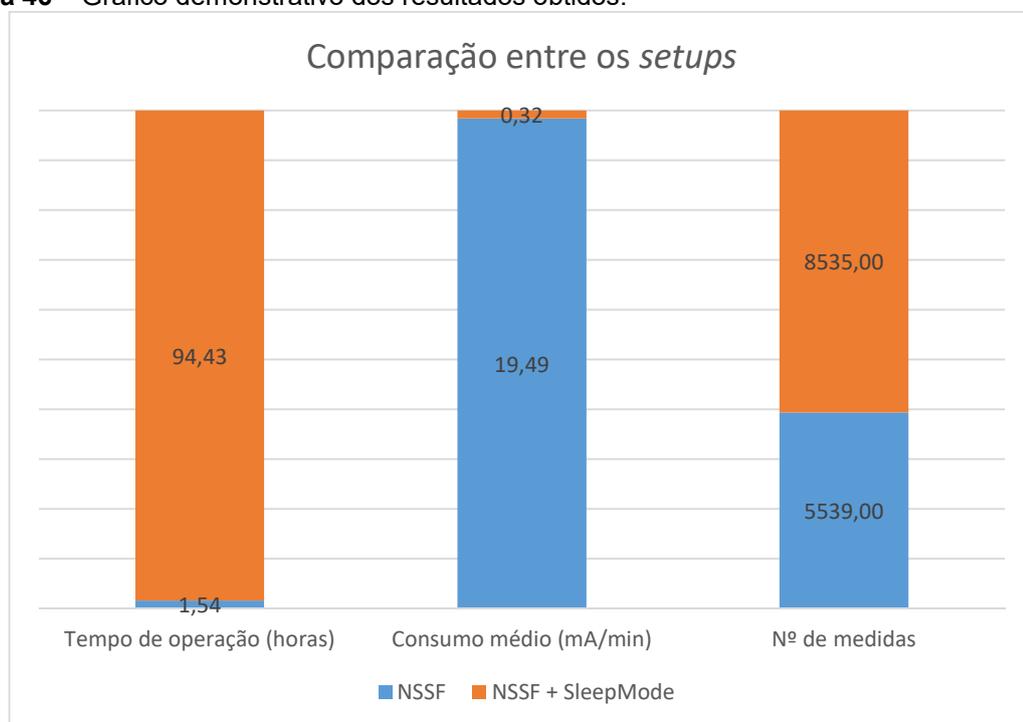


Os dados coletados podem ser visualizados, de forma numericamente resumida, na Tabela 13 e a comparação entre eles é disponibilizada na Figura 46.

**Tabela 13** – Comparação entre os dois *setups* dos resultados obtidos.

	NSSF	NSSF + <i>SleepMode</i>
Tempo de operação	01H54	94H43
Consumo corrente	19,49 mA / min	0,316 mA / min
Nº de medidas	5539 medidas	8490 medidas

**Figura 46** – Gráfico demonstrativo dos resultados obtidos.



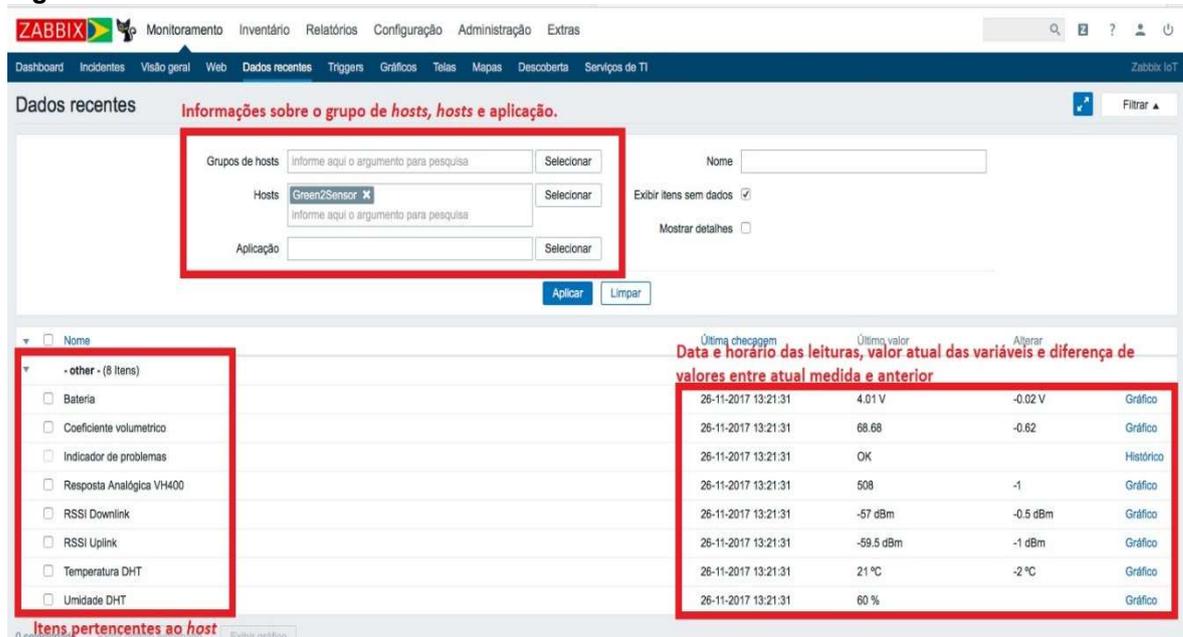
## 4.2 Fase 2

A segunda fase do projeto destinou-se à aplicação em campo do sistema de monitoramento, utilização da função *Sleepmode 2* e uso das condições emergenciais.

Através da aba *Dados recentes* foi possível a visualização em tempo real das variáveis monitoradas, bem como o detalhamento dos valores em forma de gráfico. Nessa aba também é possível a visualização do grupo de *hosts* (servidor *Zabbix*), os *hosts* (no projeto refere-se ao NSSF) e abaixo os itens (valores que

envolvem as variáveis) que pertencem ao *host* selecionado. A *interface* é detalhada na Figura 47.

**Figura 47 – Detalhamento da *Interface* dos dados.**



As Figuras 48, 48 e 50 mostram o exemplo de detalhamento de 2 variáveis: bateria (últimos 3 meses), *zoom* aplicado na variável bateria (últimos 28 dias) e coeficiente volumétrico - *VVC* (últimos 3 meses).

**Figura 48 – Detalhamento gráfico da variável bateria nos últimos 3 meses.**



Figura 49 – Zoom de 28 dias da variável bateria.

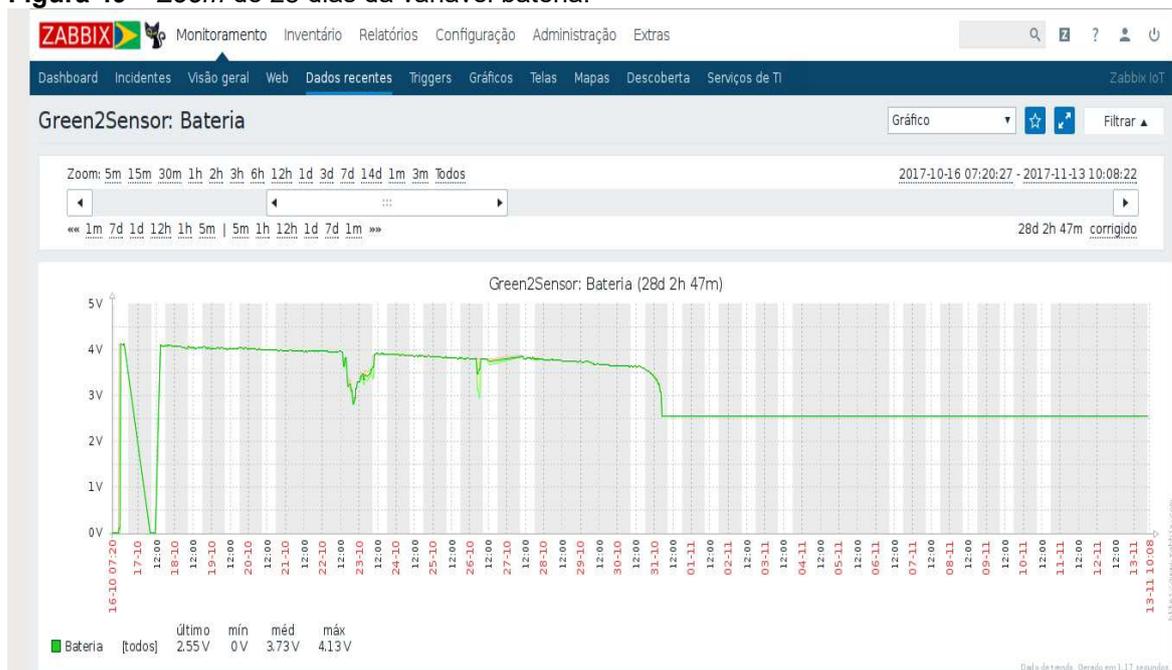
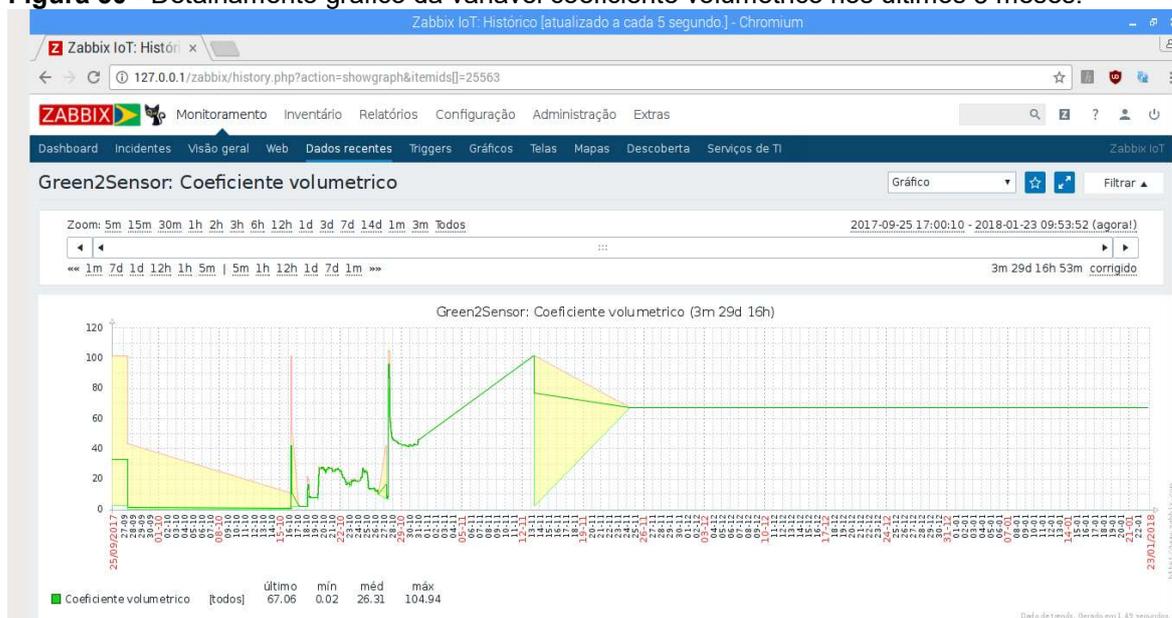


Figura 50 - Detalhamento gráfico da variável coeficiente volumétrico nos últimos 3 meses.



A visualização gráfica temporal de uma variável é feita selecionando um dos indicadores animados em tempo real, onde o *Grafana* consulta o banco de dados do *Zabbix* para a confecção do gráfico, conforme a quantidade de horas ou dias selecionados, é possível também sobrepor um gráfico temporal sobre outro, selecionando dois ou mais indicadores animados, conforme a Figura 51.

**Figura 51** – Gráficos temporais sobrepostos das variáveis temperatura e umidade.



Caso necessário, a mixagem de indicador animado e gráfico temporal pode ser selecionada para visualização conjunta, vide Figuras 52 e 53.

**Figura 52** – Indicador animado e gráfico temporal sobrepostos em tempo real.



**Figura 53** – Indicador animado e gráfico temporal sobrepostos em tempo real.



Um total de 4.564 medidas foram computadas, num prazo de 336 horas ininterruptas de operação do sistema.

Devido às fortes chuvas ocorridas durante o ensaio em ambiente externo, a condição emergencial (referente a variável VWC) foi ativada às 06h23 do dia 27/11/2017, momento em que a umidade do solo excedeu o limite estabelecido de 80% e a mesma continuou ativa por aproximadas 5 horas, sendo sobrescrita pela condição emergencial (referente a variável tensão da bateria) às 11h18 de 28/11/2017, quando o valor de referência da bateria ficou inferior a 3.4V.

As medidas de tensão de bateria também ocorreram junto com o envio de cada pacote. Considerando que a bateria forneceu um total de 3600 mAh em 20160 minutos (336 horas), entende-se que o consumo médio de energia do NSSF foi de 0,178 mA.

A Tabela 14 mostra uma visão geral dos dados sobre o sistema de monitoramento, onde são descritos os tempos de operação e seus respectivos processos e o número de medidas.

**Tabela 14** – Descrição dos processos e seus números.

Processo	Nº de Medidas	Tempo de Operação
Padrão (1 medida/30 minutos)	617	308 horas
Condição Emergencial (VWC - 1 medida/5 segundos)	3900	5 horas
Condição Emergencial (Bateria - 1 medida/30 minutos)	47	23 horas
<b>TOTAL</b>	<b>4564</b>	<b>336 horas</b>

A Tabela 15 mostra uma comparação entre os dois *setups* utilizados na Fase 1 e o *setup* da Fase 2.

**Tabela 15** – Comparação entre os *setups* utilizados nas fases do projeto.

	NSSF	NSSF + <i>SleepMode</i>	NSSF + <i>SleepMode 2</i>
Tempo de operação	01H54	94H43	336H
Consumo médio	19,49 mA	0,316 mA	0,178 mA
Nº de medidas	5539 medidas	8490 medidas	4564 medidas

## 5 CONCLUSÕES

Os resultados obtidos na Fase 1 atenderam às expectativas iniciais do projeto proposto.

Um importante ponto observado foi a descarga da bateria. Ambos os *setups* apresentaram descarga da bateria gradativa e registraram um declínio acelerado, e proporcionais, de carga entre os 3,50V e 2,80V, mostrando que a bateria manteve um padrão de descarga similar em ambos *setups*.

Partindo da análise dos dados mostrados e do princípio de comparação entre os *setups*, remata-se que houve:

- Acréscimo de 54% no número de medidas;
- Aumento do *time to live* em 6.161%;
- Queda de 1.642% no consumo médio de energia do NSSF.

Nota-se que na Fase 1, a função *SleepMode* atuou efetiva e diretamente, de forma positiva, no ponto mais crítico de funcionamento das RSSF: a energia. A opção para definição do tempo de dormência da função *SleepMode* no *Python* propiciou uma melhor adequação às necessidades da aplicação e possibilitando um melhor dimensionamento do número de medidas.

Com base nos resultados obtidos na Fase 2 do projeto, conclui-se que a função *Sleepmode 2* possibilitou uma redução no consumo de energia do NSSF para um significativo aumento do *time to live*, mesmo considerando o prazo de vários dias sem iluminação solar, iluminação da qual garantiria o carregamento da bateria.

As condições emergenciais funcionaram conforme designadas, agregando mais funções inteligentes ao sistema de monitoramento. O valor de *VWC* acima de 80% ativou a condição emergencial por 5 horas, fornecendo um total de 3.900 medidas instantâneas, onde a mesma condição só foi interrompida devido ao baixo nível de tensão da bateria, no momento em que este ficou inferior a 3.4V.

Partindo da análise dos dados mostrados na Fase 2 e, do princípio de comparação entre os *setups* que utilizaram o *SleepMode* (nativo e 2), entende-se que houve:

- Queda de 46,2% no número de medidas;

- Aumento do *time to live* em 355,9%;
- Queda de 56,32% no consumo médio de energia do NSSF.

A queda no número de medidas está ligada diretamente ao tipo de monitoração desejada, ou seja, no caso do trabalho em questão a monitoração das grandezas eram necessárias a cada 30 minutos e essa leitura sofreu interrupção apenas no momento crítico de excessiva umidade do solo, passando a ser uma monitoração quase instantânea após a variável VWC atingir 80% ou mais.

O consumo médio de energia em 0,178 mA mostra melhorias em eficiência energética utilizando o *SleepMode 2*, em comparação com o *SleepMode*. Ambas técnicas de dormência são válidas, fazendo com que o usuário decida qual utilizar com base na sua disponibilidade de *hardware/software* e tipo de monitoração desejada.

A mais notável contribuição do projeto foi conectar e adaptar diversas tecnologias objetivando a consolidação de um sistema estável, robusto e confiável para o monitoramento ambiental. Além disso, as funções *SleepMode* e *SleepMode 2* poderão ser utilizadas em diversas aplicações de RSSF que necessitam de uso inteligente de energia para um bom funcionamento e que trabalham com a plataforma *Radiuino*.

O sistema correspondeu ao esperado. Conclui-se então que o projeto é viável e funcional, beneficiando o refinamento em processos de monitoração ambiental para áreas de interesse que envolvam riscos provenientes de condições climáticas emergenciais e fornecendo melhorias em eficiência energética para RSSF.

## 6 TRABALHOS FUTUROS

Os trabalhos futuros poderão abranger novas formas de monitoramento, envolvendo diversos outros tipos de grandezas ambientais e, conseqüentemente, novas condições emergenciais.

Os futuros testes com o sistema de monitoramento também englobarão:

- Reprodução do NSSF para mais pontos de monitoramento;
- Dimensionamento de um painel fotovoltaico maior;
- Troca da bateria por outra com mais capacidade energética;
- Aprimoramento e melhoria das condições emergenciais;
- Testes *outdoor* de longa duração em recortes de mata presentes na Região Metropolitana de Campinas.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

1WSOLAR Panel 80X100. Disponível em: <<https://www.seeedstudio.com/1W-Solar-Panel-80X100-p-633.html>>. Acesso em: 10 jul. 2017.

ALDABÓ, Ricardo. **Energia Solar**. São Paulo: Artliber, 2002.

AKYILDIZ, Ian F. et al. A Survey on Sensor Networks. **Ieee Communications Magazine**, [S. l.], v. 40, n. 8, p.102-114, ago. 2002. Disponível em: <<http://dx.doi.org/10.1109/MCOM.2002.1024422>>. Acesso em: 01 jul. 2017.

ANASTASI, Giuseppe et al. Energy conservation in wireless sensor networks: A survey. **Ad Hoc Networks**, [S. l.], v. 7, n. 3, p.537-568, maio 2009. Disponível em: <<https://doi.org/10.1016/j.adhoc.2008.06.003>>. Acesso em: 10 jan. 2016.

ASSUMPÇÃO, Raphael M.. **Avaliação do impacto em rede de sensores sem fio com utilização de sistema de rádio sobre fibra**. 2011. 98 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Ceatec – Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas, Campinas, 2011. Disponível em: <<http://tede.bibliotecadigital.puc-campinas.edu.br:8080/jspui/handle/tede/521>>. Acesso em: 5 out. 2017.

BARROS, Marcelo F. et al. Avaliação do desempenho da rede de sensores sem fio eKo em uma cultura de milho. In: CONGRESSO BRASILEIRO DE AGRICULTURA DE PRECISÃO- CONBAP, 2012, Ribeirão Preto. **Anais...** Ribeirão Preto: Sbea, 2012. p. 1 - 8. Disponível em: <<http://www.agriculturadeprecisao.org.br/publicacoes/categoria/7/252/avaliacao-do-desempenho-da-rede-de-sensores-sem-fio-eko-em-uma-cultura-de-milho-barrosmarcelo-f--silva-luiz-o-l-a--cugnasca-carlos-e--molin-jose-p--conbap-2012>>. Acesso em: 10 out. 2017.

BRUNDTLAND, Gru et al. **Our common future** ('brundtland report'). 1987.

CAMARGO, Daniel S.; MIERS, Charles C.. Sensoriamento climático em sala de servidores utilizando soluções de software e hardware livre. In: COMPUTER ON THE BEACH, 2015, Itajaí. **Anais...** . Itajaí: Universidade do Vale do Itajaí - Univali, 2015. p. 389 - 391. Disponível em: <<https://siaiap32.univali.br/seer/index.php/acotb/article/view/7055>>. Acesso em: 13 out. 2017.

CASTRO, Rui M. G.. **Energias Renováveis e Produção Descentralizada: Introdução à energia fotovoltaica**. Lisboa: Deec - Secção de Energia, 2002. 51 p. Disponível em: <[http://www.troquedeenergia.com/Produtos/LogosDocumentos/Introducao\\_a\\_Energia\\_Fotovoltaica.pdf](http://www.troquedeenergia.com/Produtos/LogosDocumentos/Introducao_a_Energia_Fotovoltaica.pdf)>. Acesso em: 22 mai. 2017.

CUNHA, Felipe D. et al. Uma Nova Abordagem para Acesso ao Meio em Redes de Sensores Sem Fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 31., 2013, Brasília. **Anais...**

. Brasília: Sbrc, 2013. p. 3 - 16.

FUNDAÇÃO ECOLÓGICA NACIONAL. **Fontes alternativas de energia:** [S. l.]: Fen, 2013. 45 slides, color. Disponível em: <<http://fen.org.br/wp-content/uploads/2013/05/Fontes-Alternativas-de-Energia.pdf>>. Acesso em: 4 mai. 2017.

FUNG, Carol J.; LIU, Yanni E.. Lifetime Estimation of Large IEEE 802.15.4 Compliant Wireless Sensor Networks. In: IEEE INTERNATIONAL SYMPOSIUM ON MODELING, ANALYSIS AND SIMULATION OF COMPUTERS AND TELECOMMUNICATION SYSTEMS, 2008, Baltimore. **Proceedings...** . Baltimore, Md, Usa: IEEE, 2008. p. 1 - 4. Disponível em: <<http://dx.doi.org/10.1109/mascot.2008.4770588>>. Acesso em: 10 out. 2017.

GARCIA, Fernando P.; SOUZA, José N.; ANDRADE, Rossana M. C.. Sistemas de Monitoramento Passivo para RSSF – Soluções Existentes e uma Nova Proposta Energeticamente Eficiente. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS – SBRC, 31., 2013, Brasília. **Anais...** . Brasília: Sbc, 2013. p. 179 - 192. Disponível em: <<http://sbrc2013.unb.br/files/anais/trilha-principal/artigos/artigo-13.pdf>>. Acesso em: 14 nov. 2017.

GOULART, Cleyton et al. Monitoramento do Potencial Risco de Deriva na Pulverização Agrícola Utilizando Rede de Sensores Sem Fio. In: COMPUTER ON THE BEACH, 8., 2017, Florianópolis. **Anais...** . Florianópolis: Universidade do Vale do Itajaí, 2017. p. 1 - 3. Disponível em: <<https://siaiap32.univali.br/seer/index.php/acotb/article/view/10523>>. Acesso em: 2 out. 2017.

GROVE - Temperature and Humidity Sensor Pro. Disponível em: <[http://wiki.seeed.cc/Grove-Temperature\\_and\\_Humidity\\_Sensor\\_Pro/](http://wiki.seeed.cc/Grove-Temperature_and_Humidity_Sensor_Pro/)>. Acesso em: 17 nov. 2017.

INTERGOVERNMENTAL PANEL ON CLIMATE CHANGE. **What is the IPCC?** Disponível em: <[http://www.ipcc.ch/news\\_and\\_events/30years.shtml](http://www.ipcc.ch/news_and_events/30years.shtml)>. Acesso em: 21 mai. 2016.

INTERNATIONAL ENERGY AGENCY. **World Energy Outlook.** Paris: IEA, 2009. 698 p.

ISMAEL, Fernanda C. M. et al. Identificação e avaliação dos impactos ambientais resultantes da erosão do solo na área do Câmpus da UFCG em Pombal - PB. **Revista Verde de Agroecologia e Desenvolvimento Sustentável**, Mossoró, v. 8, n. 4, p.87-96, out-dez 2013. Disponível em: <<http://www.gvaa.com.br/revista/index.php/RVADS/article/view/2483>>. Acesso em: 10 out. 2017.

KYOTO Protocol. Disponível em: <[http://unfccc.int/kyoto\\_protocol/items/2830.php](http://unfccc.int/kyoto_protocol/items/2830.php)>. Acesso em: 12 fev. 2016.

LIMA, Raffael C. et al. Wireless Sensor Networks application on remote

Healthcare Monitoring. **Revista Interdisciplinar de Tecnologias e Educação [RInTE]**, Boituva, v. 1, n. 1, p.1-5, jun. 2016.

LION FILHO, Carlos A. P. Q.. **Construção e análise de desempenho de um fogão solar à concentração utilizando dois focos para cozimento direto**. 2007. 92 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade Federal do Rio Grande do Norte, Natal, 2007. Disponível em: <<http://repositorio.ufrn.br:8080/jspui/handle/123456789/15523>>. Acesso em: 6 nov. 2017.

LUQUETA, Gerson R. et al. Wireless Sensor Network to Monitoring an Ozone Sterilizer. **IEEE Latin America Transactions**, [s.l.], v. 14, n. 5, p.2167-2174, maio 2016. Disponível em: <<http://dx.doi.org/10.1109/tla.2016.7530410>>. Acesso em: 10 out. 2017.

MARENGO, José A.. **Mudanças Climáticas Globais e seus Efeitos sobre a Biodiversidade**: Caracterização do Clima Atual e Definição das Alterações Climáticas para o Território Brasileiro ao Longo do Século XXI. 2. ed. Brasília: Ministério do Meio Ambiente - MMA, 2007. 212 p.

MELLO, Alexandre J. T. S.; LÜDERS, Ricardo; NEVES JUNIOR, Flávio. Uma rede de sensores para monitoração da proteção catódica em dutos. In: CONGRESSO BRASILEIRO DE PESQUISA E DESENVOLVIMENTO EM PETRÓLEO E GÁS, 4., 2007, Campinas. **Anais...** . Campinas: Associação Brasileira de Pesquisa e Desenvolvimento em Petróleo e Gás, 2007. p. 1 - 9. Disponível em: <<http://www.portalabpg.org.br/PDPetro/4/listaresumos.htm>>. Acesso em: 12 out. 2017.

MOLIN, Jose P. **Agricultura de precisão**: situação atual e perspectivas. Piracicaba: Usp, 2004. 9 p. Disponível em: <<http://www.agencia.cnptia.embrapa.br/gestor/cana-de-acucar/catalogo/REC000fkl0ctof02wyiv80sq98yqwqrhrez.html>>. Acesso em: 10 dez. 2016.

MONCKS, Paulo C. S. et al. Soluções de redução de consumo energético para redes de sensores sem fio (RSSFs) aplicadas à ambientes florestais. **Tecnológica**, [s.l.], v. 20, n. 2, p.87-96, 6 jul. 2016. Disponível em: <<http://dx.doi.org/10.17058/tecnolog.v20i2.7063>>. Acesso em: 14 mai. 2016.

MONZONI, Mario et al. **Tendências e oportunidades na economia verde**: energias renováveis. [S. l.]: Fgv Eaesp - Gvces, 2010. 72 p. Disponível em: <<http://hdl.handle.net/10438/15423>>. Acesso em: 10 nov. 2016.

PESSOA, Cláudio R. M.; JAMIL, George L.; BRANCO JÚNIOR, Manuel R. F.. A internet das coisas: conceitos, aplicações, desafios e tendências. In: CONTECSI - INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT, 13., 2016, São Paulo. **Anais...** . São Paulo: Contecsi Usp, 2016. p. 129 - 149. Disponível em: <<http://www.contecsi.fea.usp.br/envio/index.php/contecsi/13CONTECSI/paper/vie/w/3654>>. Acesso em: 19 out. 2017.

PODELESKI, Fabiana da Silva. **Rede sensor sem fio para monitoramento de umidade e temperatura de solo**. 2013. 67 f. TCC (Graduação) - Curso de Engenharia Elétrica, Pontifícia Universidade Católica de Campinas, Campinas, 2013.

THE PEW CHARITABLE TRUSTS. **Who's Winning the Clean Energy Race?** Filadélfia: The Pew Charitable Trusts, 2014.

RADIOIT. Disponível em: <<https://www.radioit.com.br/produtos.html>>. Acesso em: 14 out. 2016.

RADIUINO. Disponível em: <<http://radiuino.cc/>>. Acesso em: 14 out. 2016.

REIS, João Bosco Coura dos; CORDEIRO, Thiago Lobão; LOPES, Eymar Silva Sampaio. **Utilização do sistema de monitoramento e alerta de desastres naturais aplicado a situações de escorregamentos**: caso de Angra dos Reis. [S. l.]: Terrama<sup>2</sup>, [201-].

REN21. **Renewables 2016**: Global status report. Paris: Ren21, 2016. Disponível em: <<http://www.ren21.net/gsr-2016/>>. Acesso em: 12 nov. 2016.

ROSELINE, R. A.; SUMATHI, P.. Solar power for Wireless Sensor Networks in Environment Monitoring Applications-A Review. **International Journal Of Scientific And Research Publications**, [S. l.], v. 4, n. 8, p.1-3, ago. 2017. Disponível em: <<http://www.ijsrp.org/research-paper-0814.php?rp=P323045>>. Acesso em: 5 fev. 2016.

SILVA, Jorge J. B. da. **Análise técnica e econômica de um sistema fotovoltaico como fonte de energia para agricultura familiar**. 2014. 59 f. Monografia (Especialização) - Curso de Formas Alternativas de Energia, Universidade Federal de Lavras, Lavras, 2014. Disponível em: <<http://repositorio.ufla.br/jspui/handle/1/4557>>. Acesso em: 3 out. 2017.

TERUEL, Barbara et al. Desempenho de rede de sensores sem fio em casa de vegetação. **Revista Brasileira de Engenharia Agrícola e Ambiental**, [s.l.], v. 16, n. 12, p.1374-1380, dez. 2012. Disponível em: <<http://dx.doi.org/10.1590/s1415-43662012001200016>>. Acesso em: 10 out. 2017.

VH400 Soil Moisture Sensor Probes. Disponível em: <<https://www.vegetronix.com/Products/VH400/>>. Acesso em: 14 out. 2016.

WANG, Qinghua; BALASINGHAM, Ilangko. Wireless Sensor Networks: An Introduction. In: MERRETT, Geoff V; TAN, Yen Kheng (Ed.). **Wireless Sensor Networks: Application - Centric Design**. [S. l.]: Intech, 2010. p. 1-13. Disponível em: <<https://www.intechopen.com/books/wireless-sensor-networks-application-centric-design>>. Acesso em: 12 out. 2017.

WINKLER, Michael et al. Theoretical and practical aspects of military wireless sensor networks. **Journal Of Telecommunications And Information Technology**, [S. l.], v. 2008, n. 2, p.37-45, fev. 2008. Disponível em:

<<http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-article-BATA-0001-0025>>. Acesso em: 4 fev. 2017.

ZHAO, Gang. Wireless Sensor Networks for Industrial Process Monitoring and Control: A Survey. **Network Protocols And Algorithms**, [s.l.], v. 3, n. 1, p.46-63, 5 abr. 2011. Disponível em: <<http://dx.doi.org/10.5296/npa.v3i1.580>>. Acesso em: 13 mai. 2017.

## 8 APÊNDICES

### 8.1 Código Python

Segue o código *Python* utilizado no projeto, com o funcionamento do *SleepMode 2* e interação com o *Zabbix*.

```
#!/usr/bin/env python
# -*- coding: latin-1 -*-
# Link de acesso ao MapBox https://www.mapbox.com/studio/

# PROGRAMA PARA MULTIPLOS SALTOS
import serial
import math
import time
import struct
import os.path
from time import localtime, strftime
from datetime import datetime
from datetime import timedelta
#from tago import Tago
from pyzabbix import ZabbixSender,ZabbixMetric
from zabbix_api import ZabbixAPI
#import requests

#===== configurações de rede =====
'''
#Inserindo os Tokens dos dispositivos para upload de dados na Tago.io
MY_DEVICE_TOKEN_1 = ""
MY_DEVICE_TOKEN_2 = ""
MY_DEVICE_TOKEN_3 = ""
MY_DEVICE_TOKEN_4 = ""
MY_DEVICE_TOKEN_S = ""
my_device_1 = Tago(MY_DEVICE_TOKEN_1).device
my_device_2 = Tago(MY_DEVICE_TOKEN_2).device
my_device_3 = Tago(MY_DEVICE_TOKEN_3).device
my_device_4 = Tago(MY_DEVICE_TOKEN_4).device
my_device_S = Tago(MY_DEVICE_TOKEN_S).device
'''

'''
#Inserindo as informações para upload de dados no ThingSpeak
API_KEY = 'api key'
API_URL = 'https://api.thingspeak.com/update'
'''

server = "http://127.0.0.1/zabbix"
username = "admin"
password = "zabbix"

zapi = ZabbixAPI(server = server, path="", log_level = 6)
zapi.login(username, password)

#
```

```

item = zapi.item.get({"output":"extend","hostids":"10117" , "search":{"key_": "vwc"},
"sortfield":"name"})

for x in item:
    valor = x["lastvalue"]
    #print "VALOR ", valor

#=====liga o pino de do GPIO (DTR)=====
##os.system('sudo sh -c "echo -n "1" > /sys/class/gpio/gpio87/value"')

#=====IP do Zabbix Server=====
#IP = raw_input ("Digite o IP do Servidor Zabbix = ") #seta o IP do Servidor Zabbix
IP = '127.0.0.1'

#=====CONFIGURA SERIAL 1 (USB)=====
# para COM# o numero que se coloca eh n-1 no primeiro parametro. Ex COM9 valor 8
#n_serial = raw_input("Digite o numero da serial = ") #seta a serial
#n_serial1 = "/dev/ttyUSB0"
n_serial1 = "/dev/ttyUSB0"

#int(n_serial) - 1
ser = serial.Serial(n_serial1, 9600, timeout=0.5,parity=serial.PARITY_NONE) # seta valores da
serial

### ===== reseta o BE900 =====
## comentado porque a usb já reinicia a serial pelo DTR quando abro a serial
##ser.setDTR(False) #send DTR Pin to High
##time.sleep(0.2)
##ser.setDTR(True) #send DTR Pin to low

time.sleep(2)

#===== VARIÁVEIS UTILIZADAS =====

#ITEMP1 = TEMP2 = LUMI1 = LUMI2 = RSSId1 = RSSlu1 = RSSId2 = RSSlu2 = 0

#===== TEMPOS DE REPOSTA E POLLING =====
TEMPO1 = 0.5 # Tempo de Resposta = é o tempo entre o envio do pacote da base para sensor e
da devolução do pacote do sensor para base
TEMPO2 = 1 # Tempo entre transmissões
TEMPO3 = 1 # Tempo de espera entre as transmissoes, caso nao haja perda de pacotes

#===== Identificação das bases=====
#ID_base = raw_input('ID_base = ')
ID_base1 = 0

#===== Identificação do sensor a ser acessado =====
# Identificação do sensor a ser acessado
#ID_sensor1 = raw_input('ID do sensor 1 = ')
ID_sensor1 = 1

#Definicao dos tempos de Sleep
t1 = 120 #1800 segundos (1800/15 = 120)
t2 = 8 #120 segundos (120/15 = 8)
t3 = 120 #1800 segundos (1800/15 = 120)

#===== CRIAÇÃO DOS PACOTES PARA CADA SENSOR=====
Pacote1 = {}

```

```

# Cria Pacote
de 52 bytes com valor zero em todas as posições.
for i in range(0,52): # faz um array com 52 bytes
    Pacote1[i] = 0

#===== Criação dos diretórios e dos nomes dos arquivos para armazenamento de log
=====

timenow = datetime.now()
data = (timenow.strftime("%H")) # %H arquivo por hora / %M arquivo por minuto
Diretorio_dia="/opt/radiuino/dados/logs/"+strftime("%Y_%m_%d")+ "/"
if not os.path.exists(Diretorio_dia):
    os.makedirs(Diretorio_dia)
Diretorio_dia_hora="/opt/radiuino/dados/logs/"+strftime("%Y_%m_%d")+ "/" + data + "/"
if not os.path.exists(Diretorio_dia_hora):
    os.makedirs(Diretorio_dia_hora)

arq_medidas1 = Diretorio_dia_hora + "Sensor_" + str(ID_sensor1) + strftime("_%Y_%m_%d_")+
data + strftime("_hrs.txt")

arqlog = "/opt/radiuino/dados/logs/log.txt" #log para debug

#===== Execução das medidas =====

#===== configuração do pacote de envio do sensor 1 - SHT75 =====

Pacote1[8] = int(ID_sensor1) #endereço de destino do pacote
Pacote1[10] = int(ID_base1) #endereço de origem do pacote

if float(valor) >= 80.0:
    Pacote1[4] = 0
    #print "entrou no if"
else:
    Pacote1[4] = 3
    #print "entrou no else"

Pacote1[5] = int(t1)
Pacote1[6] = int(t2)
Pacote1[7] = int(t3)

ser.flushInput() # Limpa o buffer da serial

#===== COMUNICAÇÃO DA BASE COM O SENSOR 1 =====
# ==== Transmissão do pacote de leitura ====
for k in range(0,52):
    TXbyte = chr(Pacote1[k])
    ser.write(TXbyte)

# ==== Aguarda a resposta do sensor ====
time.sleep(int(TEMPO1))

line = ser.read(52) # faz a leitura de 52 bytes do buffer que recebe da serial pela COM
if len(line) == 52:

    rssid = ord(line[0]) # RSSI_DownLink
    rssiu = ord(line[2]) # RSSI_UpLink

    #RSSI Downlink
    if rssid > 128:
        RSSId=((rssid-256)/2.0)-81

```

```

else:
    RSSId=(rssid/2.0)-81
#RSSI Uplink
if rssi > 128:
    RSSlu=((rssi-256)/2.0)-81

else:
    RSSlu=(rssi/2.0)-81

# Temperatura - DHT22
ad0t = ord(line[16]) # tipo de sensor - no caso está medindo temperatura
ad0h = ord(line[17]) # alto
ad0l = ord(line[18]) # baixo
AD0 = ad0h * 256 + ad0l
AD0 = AD0/100.0

# Umidade - DHT22
ad1t = ord(line[19]) # tipo de sensor - no caso está medindo LDR
ad1h = ord(line[20]) # alto
ad1l = ord(line[21]) # baixo
AD1 = ad1h * 256 + ad1l
AD1 = AD1/100.0

# VG
ad2t = ord(line[22]) # tipo de sensor - no caso está medindo LDR
ad2h = ord(line[23]) # alto
ad2l = ord(line[24]) # baixo
AD2 = ad2h * 256 + ad2l

# VWC
ad3t = ord(line[25]) # tipo de sensor - no caso está medindo LDR
ad3h = ord(line[26]) # alto
ad3l = ord(line[27]) # baixo
AD3 = ad3h * 256 + ad3l
AD3 = AD3/100.0

# Bateria
ad4t = ord(line[28]) # tipo de sensor - no caso está medindo LDR
ad4h = ord(line[29]) # alto
ad4l = ord(line[30]) # baixo
AD4 = ad4h * 256 + ad4l
AD4 = AD4*(5.0/1023.0)

# ===== Escrita no arquivo de LOG =====
timenow = datetime.now()
arq1 = open(arq_medidas1, "a+")
print >>arq1,timenow.strftime("%Y/%m/%d %H:%M:%S.%f"),'; Temperarura DHT ; ',AD0,';C;', '
Umidade DHT ;',AD1,';%;', ' ;VG;',AD2,'; VWC ;', AD3,'; Bateria;',AD4,';V;', ' ; RSSId;',RSSId,';dbm;', '
RSSlu;',RSSlu,';dbm;'
arq1.close()

'''
# ===== Publicando informações no Tago.io =====
payload1=
[{"variable": "temperatura", "value": str(temperatura), "unit": "°C"}, {"variable": "tmax", "value": str(tmax), "unit": "°C"}, {"variable": "tmin", "value": str(tmin), "unit": "°C"}, {"variable": "RSSIup", "value": str(RSSlu), "unit": "dBm"}, {"variable": "RSSIdown", "value": str(RSSId), "unit": "dBm"}]
response_1 = my_device_1.insert(payload1)
time.sleep(int(TEMPO2))

```

```

'''
'''
# ===== Publicando informações no ThingSpeak =====
data = {'api_key': API_KEY, 'field1':temperatura, 'field2':tmax, 'field3':tmin, 'field4':RSSId1,
'field5':RSSId1};
result = requests.post(API_URL, params=data)
print result.status_code

# if result.status_code == 200:
#     print "Success!! Thingspeak"
# else:
#     print "Fail!! Thingspeak"

time.sleep(int(TEMPO2))
'''

# if ad0t == 0:

problema = "OK"
# Define as chaves dos itens no Zabbix
L1=["temperatura", "umidade", "vg", "vwc", "bateria", "RSSId", "RSSId"]

# Define as variáveis do Script que alimentarão os itens
L2=[AD0, AD1, AD2, AD3, AD4, RSSId, RSSId]

# Faz um loop para enviar cada valor de métrica para cada item no Zabbix
x = 0
while x<len(L1):
    # Envia Temperatura para o Zabbix
    metrics = []
    m = ZabbixMetric('Green2Sensor', L1[x], L2[x])
    metrics.append(m)
    zbx = ZabbixSender(zabbix_server=IP, zabbix_port=10051, use_config=None)
    zbx.send(metrics)
    x+=1
# else:
#     problema = "Problema de medida"
#     #L1=["problema"]
#     #L2=[problema]

# x=0
# while x<len(L1):
#     # Envia Temperatura para o Zabbix
#     metrics = []
#     m = ZabbixMetric('Sensor', L1[x], L2[x])
#     metrics.append(m)
#     zbx = ZabbixSender(zabbix_server=IP, zabbix_port=10051, use_config=None)
#     zbx.send(metrics)
#     x+=1

else:
    # ===== Limpa o Buffer da serial de entrada =====
    ser.flushInput()

# ===== Aguarda tempo entre transmissões =====
#time.sleep(int(TEMPO2))

# ===== Fecha as portas seriais =====

```

```
ser.close()
time.sleep(0.5)
#time.sleep(int(TEMPO3))
time.sleep(int(TEMPO3))

# ==== reseta o BE900 ====
##os.system('sudo sh -c "echo -n "0" > /sys/class/gpio/gpio87/value"')
##time.sleep(0.1)
##os.system('sudo sh -c "echo -n "1" > /sys/class/gpio/gpio87/value"')
```

## 9 ANEXOS

### 9.1 Anexo A – Esquema elétrico da placa DK106

