

Universidade Federal do Espírito Santo - UFES
Centro de Ciências Exatas, Naturais e da Saúde - CCENS
Bacharelado em Sistemas de Informação

Nélio Gonçalves Godoi

UVLoRA: Monitoramento de Radiação UV com IoT

Alegre, ES
2019

Nélio Gonçalves Godoi

UVLoRA: Monitoramento de Radiação UV com IoT

Trabalho de Conclusão de Curso apresentado
ao Departamento de Computação do Centro
de Ciências Exatas, Naturais e da Saúde da
Universidade Federal do Espírito Santo como
requisito parcial para obtenção do grau de
Bacharel em Sistemas de Informação.

Orientador: Prof.: Dr. Roberto Colistete Jr.

Alegre, ES
2019

Nélio Gonçalves Godoi

UVLoRA: Monitoramento de Radiação UV com IoT

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof.: Dr. Roberto Colistete Jr.

Trabalho apresentado em 12 de Dezembro de 2019:

COMISSÃO EXAMINADORA

Prof^a. Me^a. Valéria Alves Silva
Professora da disciplina

Prof. Dr. Roberto Colistete Junior
Orientador

Prof. Dr. Helder de Amorim Mendes
Coorientador

Agradecimentos

A todos que de alguma forma contribuíram para a realização deste trabalho. Ao Professor Roberto Colistete Júnior idealizador e orientador deste trabalho. Aos Professores Helder Amorim Mendes e a Valéria Alves Silva pela disponibilidade.

Dedico este trabalho a todos que quando crianças, sonharam em tornar o mundo melhor por meio do conhecimento e da ciência, e principalmente, a todos os professores que acreditaram e incentivaram tal sonho!

Resumo

UVLoRa trata-se de uma estação de monitoramento de radiação ultravioleta (UV) com dados publicados em tempo real em servidor *IoTCloud*. Um exemplo de ciência cidadã a ser levada a sociedade, sendo um instrumento de melhoraria da saúde pública já que pode gerar uma conscientização da população sobre os riscos do câncer de pele e danos aos olhos causados ou agravados pela UV. Seu principal objetivo é o monitoramento e conscientização sobre os riscos da exposição exagerada. UVLoRa utiliza dispositivos inteligentes programados, sensores e tecnologia de transmissão sem fio LoRa para comunicação em longas distâncias, alcançando mais de 4 Km. Dois protótipos de aparelhos foram desenvolvidos um para nó e outro para gateway onde é possível comunicação bidirecional.

Palavras Chave: IoT, UVI, LoRa, Melanoma, Micropython

Abstract

UVLoRa is a monitoring station for ultraviolet radiation (UV) with data published in real time on server IoTCloud. An example of citizen science being brought to society as a tool for improving public health as it can raise awareness of the risks of skin cancer and eye damage caused or aggravated by UV. Its main objective is the monitoring and awareness of the risks of overexposure. UVLoRa utilizes smart programmed devices, sensors and LoRa wireless transmission technology for long distance communication reaching over 4 km. Two handset prototypes have been developed one for node and one for gateway where two-way communication is possible.

Keywords: IoT, UVI, LoRa, Melanoma, Micropython

Listas de ilustrações

Figura 1 – Ilustração da divisão do espectro de luz solar que chega a superfície terrestre.	11
Figura 2 – Arquitetura LoRaWAN	25
Figura 3 – Diagrama da estrutura da Rede UVLoRa.	30
Figura 4 – LoPy4	31
Figura 5 – Adafruit VEML6075	32
Figura 6 – PAN TILT adquirido para testes	33
Figura 7 – Exemplo de domo óptico	34
Figura 8 – Preços para a assinatura da plataforma Ubidots.	36
Figura 9 – Apresentação da Ubidots STEM, versão gratuita da plataforma para pesquisa e educação.	37
Figura 10 – Custos de assinatura do Adafruit IO.	37
Figura 11 – Protótipo de UVLoRa Nó	39
Figura 12 – Visão interna da caixa de proteção do prototípico UVLoRa.	40
Figura 13 – Itens básicos que compõem um protótipo de nó	40
Figura 14 – Protótipo de nano gateway UVLoRa.	41
Figura 15 – Itens básicos que compõem um protótipo de nano gateway UVLoRa. .	41
Figura 16 – Log do <i>script</i> via serial/USB.	43
Figura 17 – Log do <i>script</i> via serial/USB.	44
Figura 18 – Dashboard UVLoRa no <i>IoTCloud</i> Ubidots.	45
Figura 19 – SMSs enviados pela plataforma Ubidots, recurso de alerta.	46
Figura 20 – Painel UVLoRa criado utilizando a plataforma Adafruit IO.	47
Figura 21 – Mapa com marcações do teste de alcance LoRa realizado.	47
Figura 22 – Visada do ponto 2 para o ponto 3	48
Figura 23 – Visada do ponto 2 para o ponto 1	49
Figura 24 – Autonomia da bateria dos dispositivos UVLoRa Nó.	49
Figura 25 – Autonomia da bateria dos dispositivos UVLoRa Nó.	50
Figura 26 – Protótipo de placa desenvolvida para o UVLoRa por Eduardo Amorim Benincá, formado em Física na UFES-Alegre e bolsista de Desenvolvimento Tecnológico Industrial (DTI-B) da FAPES, atuando na UFES-Vitória em projetos de IoT e sensores do CPID (Centro de Pesquisa, Inovação e Desenvolvimento) do ES.	53
Figura 27 – Pytrack.	53

Lista de tabelas

Tabela 1 – Recomendações de fotoproteção da WHO	19
Tabela 2 – Parcada dos dados do arquivo de log gerado pelo Nô UVLoRa	43
Tabela 3 – Parcada dos dados do arquivo de log gerado pelo Nano Gateway UVLoRa.	43

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	13
1.1.1	Objetivos Específicos	13
2	REFERENCIAL TEÓRICO	14
2.1	Radiação ultravioleta (UV)	14
2.1.1	Efeitos sobre a saúde humana	15
2.1.2	Fatores de Risco	17
2.1.3	Proteção	18
2.1.3.1	UV Index	18
2.2	Internet das coisas (IoT)	20
2.2.1	Hardwares	20
2.2.2	Softwares	21
2.2.3	Tecnologias de comunicação	23
2.2.4	LoRa	24
2.2.4.1	LoRaWAN	24
2.2.4.2	LoRaMESH	26
2.2.4.3	LoRaMAC	26
2.2.5	Aplicações	27
2.2.5.1	Cidades mais inteligentes	27
2.2.5.2	Limpeza do ar e da água	28
2.2.5.3	Agricultura mais eficiente	28
2.2.5.4	Combatendo o câncer de mama	29
3	METODOLOGIA	30
3.1	Classificação e delineamento da pesquisa	30
3.2	A dinâmica de funcionamento do UVLoRa	30
3.3	Dispositivos de Hardwares	31
3.4	Softwares	35
3.5	Comunicação	38
3.6	Consumo de Energia	38
4	RESULTADOS E AVALIAÇÃO	39
4.1	Dispositivos UVLoRa	39
4.1.1	Nó	39
4.1.2	<i>Script</i> do Nó	39

4.1.3	Nano Gateway	41
4.1.4	<i>Script do Nano Gateway</i>	41
4.2	Driver VEML6075	42
4.3	Caso de Uso	42
4.4	Testes UVLoRa	44
4.5	Consumo de energia e autonomia de bateria no N�o	45
5	CONCLUSO	51
5.1	Trabalhos Futuros	52
5.1.1	Testes de acur�cia e preciso� de sensores UV	52
5.1.2	Utiliza�o de domo de luz para protec�o do sensor	52
5.1.3	Otimiza�o do sistema para economia de energia	52
5.1.4	Mais testes de alcance LoRa	52
5.1.5	Utiliza�o de PyTrack em dispositivos UVLoRa	52
REFERÊNCIAS		54
AP�NDICE A – <i>SCRIPT UVLORA N�O</i>		60
AP�NDICE B – <i>SCRIPT UVLORA NANO GATEWAY</i>		65
AP�NDICE C – <i>DRIVER MICROPYTHON PARA VEML6075</i> . . .		70

1 Introdução

A luz solar no seu sentido mais amplo é o espectro total da radiação eletromagnética fornecida pelo Sol. Na Terra, a luz solar é filtrada pela atmosfera, e parte dessa radiação é visível como a luz do dia quando o Sol está acima do horizonte.

No entanto, do espectro solar que atinge a superfície terrestre: 5% é de radiação ultravioleta com comprimento de onda que varia entre 100nm e 400nm; 39% é de luz visível indo de 400nm a 800nm de comprimento de onda; e outros 56% de infravermelhos que vão acima de 800nm de comprimento de onda (BALOGH et al., 2011), como é ilustrado na figura 1.

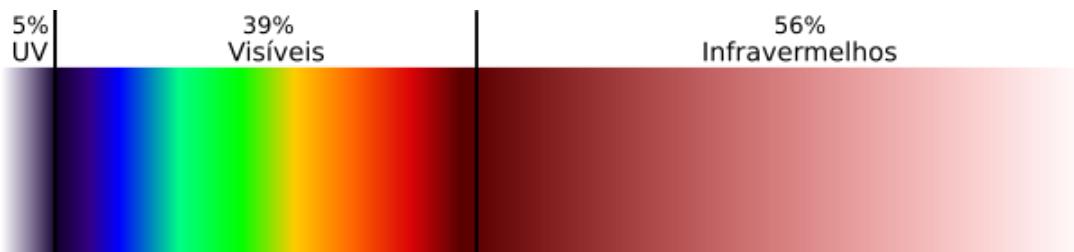


Figura 1 – Ilustração da divisão do espectro de luz solar que chega a superfície terrestre.

Fonte: O Autor.

O organismo humano percebe a presença destas radiações de formas distintas. A radiação infravermelha (IV) é percebida sob a forma de calor, a radiação visível (Vis) através das diferentes cores detectadas pelo sistema óptico e a radiação ultravioleta (UV) através de reações fotoquímicas, ou seja, reações de interação de átomos e moléculas com a radiação eletromagnética.

Preocupada em incentivar as pessoas a reduzirem à sua exposição ao sol e assim diminuir os efeitos nocivos à saúde e consequentemente os custos com tratamentos de problemas de saúde relacionados a UV, a WHO et al. (1979) desenvolveu o Índice UV (UVI) unidade de medida do nível de radiação ultravioleta. A escala foi adotada e padronizada em conjunto com a Organização das Nações Unidas (ONU) e a Organização Meteorológica Mundial (OMM) em 1994. O UVI dá uma indicação do potencial perigo de exposição ao sol e isso pode ajudar os indivíduos a fazerem escolhas saudáveis. Este índice pode ser usado para ajustar planos diários de cada pessoa ao escolher roupas e acessórios para proteção contra os raios UV. Em muitos países, o UVI é relatado juntamente com a previsão do tempo nos jornais, na TV e no rádio; No entanto, isso é feito somente durante os meses de verão geralmente. O problema é que em países próximos a linha do equador, como o Brasil, onde o UVI no verão chega a 16, e mesmo durante o inverno continua alto e portanto

as pessoas deveriam estar quase sempre protegidas, o UVI é por vezes desconhecido ou ignorado.

A Internet das Coisas, ou simplesmente IoT, do inglês *Internet of Things*, retrata a próxima evolução da internet, e já é uma realidade, representando um grande avanço na coleta, distribuição e análise de dados. Resumindo a IoT cria uma melhor qualidade de vida ao conectar "Coisas", que podem ser pessoas, objetos, equipamentos diversos e sistemas, e integrá-los usando a conectividade e tecnologia da web (KAMALANATHAN et al., 2013).

A internet em si, é definida como uma rede de computadores que interconecta milhares de dispositivos, enquanto a Web é definida como uma infraestrutura que provê serviços e aplicações distribuídas (KUROSE; ROSS; ZUCCHI, 2010). Cientes dessas definições, podemos observar que IoT envolve plataformas de hardware e software para permitir que dados sejam coletados, distribuídos, processados e então analisados.

Combinando os diversos tipos de microcontroladores, sensores, atuadores, protocolos e aplicações software disponíveis atualmente é possível criar impensáveis possibilidades de automações, abrangendo praticamente todas as áreas do conhecimento indo desde as lavouras na produção de alimentos aos hospitais no monitoramento de pacientes, passando por outras atividades de saúde, pesquisa, ensino, comércio, indústria, esporte, lazer etc.

Embora existam dispositivos capazes de medir UVI, eles são muito pouco utilizados devido a sua não popularização e dificuldade no manuseio do aparelho durante as medições. Alguns sites, mostram uma previsão do UVI baseadas em imagens de satélites via irradiação solar, nuvens e altitude em cada local, mas estes valores representam apenas estimativas.

Como o índice varia muito com os diversos fatores que podem influenciar nos níveis de UV, o ideal seria cada região ter uma estação de monitoramento de UVI, idealmente com dados publicados na Internet para permitir que ele seja usado por uma população para ajustar os planos diários e permitir escolher roupas e acessórios. Analogamente à escala de temperatura, o UVI dá uma indicação do nível de radiação UV e consequentemente do potencial perigo à exposição ajudando-os a fazer escolhas mais saudáveis ao realizar tarefas do dia a dia onde é indispensável à exposição ao sol.

Há uma lacuna ainda existente de aparelhos de monitoramento de radiação UV. E assim nasce o projeto UVLoRa que é então uma estação de monitoramento de radiação ultravioleta com Internet das coisas (IoT) via LoRa – tecnologia de comunicação sem fio de longa distância, cujo nome é na realidade uma abreviação para *Long Range*, que significa “longo alcance” em inglês – e ou Redes Móveis com foco em ciência cidadã. Trabalhos no preenchimento desta lacuna incentivam a adoção do UVI no planejamento de atividades ao ar livre e consequentemente preservam a saúde da população que tem acesso à tal informação já que acaba por minimizar os índices de câncer de pele e outros problemas de

saúde causados ou agravados pela UV.

1.1 Objetivos

Objetiva-se o desenvolvimento do sistema UVLoRa em parte ou em sua totalidade, a ser realizado a fim de comprovar a eficácia da utilização da IoT na coleta e visualização de dados, com foco na prevenção de doenças causadas ou agravadas por radiação UV.

1.1.1 Objetivos Específicos

Mais especificamente os objetivos do UVLoRa, visam principalmente o monitoramento dos níveis de radiação UV por meio dos aparelhos UVLoRa Node a serem desenvolvidos.

Visa-se também cobrir localidades sem cobertura de redes móveis por meio dos aparelhos UVLoRa Nano Gateway a serem desenvolvidos.

Desenvolver um sistema para permitir o envio de informações úteis (Exemplo: Previsão do clima para a sua localização, Datas de vacinação, Mutirões, etc.) a comunidades remotas através dos aparelhos UVLoRa Node caso ele esteja nesta localização.

Busca-se ainda equipamentos robustos e autônomos em energia por longos períodos de tempo, que sejam simples de instalar, que tenham manutenção reduzida, que sejam fáceis de usar e principalmente que seja fácil de interpretar os dados.

2 Referencial Teórico

Este capítulo está divido em duas sessões devido a distinção dos assuntos abordados, radiação ultravioleta e internet das coisas. Isto ocorre devido a fato desta pesquisa ser caráter exploratório e realizar análises nestas áreas em busca de um objetivo em comum na qual há poucas informações sistematizadas.

2.1 Radiação ultravioleta (UV)

A região UV é dividida em três bandas: UVA (315-400 nm), UVB (280-315 nm), UVC (100-280 nm). As de bandas de menor energia penetram mais profundamente na pele, no entanto segundo à Organização Mundial de Saúde (WHO do inglês *World Health Organization*) à medida que a luz solar passa pela atmosfera terrestre, todos os raios UVC e aproximadamente 90% da raios UVB são absorvidos pela camada ozônio e outros compostos da atmosfera. Os raios UVA são os menos absorvidos e portanto, a radiação UV que atinge a superfície terrestre é amplamente composta por UVA com um pequeno percentual de UVB (WHO, 2018).

Além da atmosfera, os níveis de UV são influenciados por fatores ambientais, como por exemplo (WHO, 2018):

- **Elevação do Sol:** quanto mais elevado o Sol no céu, maior o nível de radiação UV, logo por volta do meio dia solar ocorre o máximo.
- **Estação do ano:** no inverno a radiação ultravioleta é menos intensa, no verão é mais intensa;
- **Latitude:** Quanto mais próximo do equador, maiores os níveis de radiação UV.
- **Cobertura da nuvem:** Mesmo com a cobertura da nuvem, os níveis de radiação UV podem ser altos devido à dispersão da radiação UV por moléculas de água e partículas finas na atmosfera.
- **Altitude:** Em altitudes mais elevadas, uma atmosfera mais fina filtra menos radiação UV. A cada 1000 metros de altitude, os níveis de UV aumentam de 10% para 12%.
- **Ozônio:** A camada de ozônio absorve parte das radiações UV que de outra forma atingiriam a superfície da Terra. Os níveis de ozônio variam ao longo do ano e até ao longo do dia.

- **Reflexão do solo:** A radiação UV é refletida ou dispersa em diferentes extensões por diferentes superfícies, por exemplo, a neve pode refletir tanto quanto 80% da radiação UV, areia seca da praia cerca de 15% e espuma do mar em cerca de 25%.

2.1.1 Efeitos sobre a saúde humana

Ainda em 1979 a WHO et al. (1979) já publicava estudos sobre a UV alertando sobre o seu efeito biológicos sobre nós humanos, animais e plantas. Onde esclarece que um efeito benéfico completamente estabelecido da UV na pele é a síntese da Vitamina D3, no entanto a sua produção é limitada e necessita apenas de pequenas exposições para à sintetizar a vitamina, e que a excessiva exposição a UV pode trazer malefícios se feita sem a proteção adequada onde o câncer de pele e a catarata são exemplos.

Confirmando o que a WHO propunha a mais de 30 anos antes Norval e Halliday (2011) reafirmam: "A exposição à luz solar tem resultados positivos e negativos sobre saúde humana". concluem ainda que, ambos UVA e UVB são imunossupressores¹ em humanos e este é um evento que os torna vulneráveis a todos os tipos de câncer de pele e particularmente em países com altos níveis de exposição ao sol esta é a mais importante consequência a saúde dessa exposição (NORVAL; HALLIDAY, 2011).

A diminuição do nível de ozônio na atmosfera provoca o aumento na quantidade de radiação UV que atinge a Terra sendo assim à medida que a camada de ozônio torna-se mais fina, o filtro de proteção fornecido pela atmosfera é progressivamente reduzido. Estimando que para cada 1% de diminuição do nível de ozônio, exista o aumento de 1 a 2% na quantidade de radiação UVB que atinge a superfície terrestre. Sabendo que aumento da incidência de câncer de pele é dado em função do aumento da radiação UVB é possível verificar que para cada 1% de diminuição do nível de ozônio o risco de desenvolver câncer de pele aumenta em cerca de 3% (BALOGH et al., 2011). Modelos computacionais preveem que uma diminuição de 10% na camada de ozônio pode causar um adicional de 300 000 casos câncer de pele não-melanoma e 4 500 de melanoma e entre 1,6 e 1,75 milhões de casos de catarata a nível mundial. Estes são alguns dos problemas mais comuns dentre aproximadamente 40 tipos de doenças que são causadas ou agravadas pela exposição a UV (JUCHEM et al., 2001).

O melanoma é um tipo de câncer de pele que tem origem nos melanócitos (células produtoras de melanina, substância que determina a cor da pele) e tem predominância em adultos brancos. Embora o câncer não melanoma seja o mais frequente no Brasil e corresponda a 30% de todos os tumores malignos registrados no país, o melanoma representa apenas 3% das neoplasias malignas de pele - maior órgão do corpo humano -, apesar de ser o mais grave devido à sua alta possibilidade de metástase (INCA, 2018b). O

¹ Dicionário: imunossupressor adjetivo substantivo masculino - que suprime ou reduz as reações imunológicas específicas do organismo contra uma substância ou tratamento

não melanoma apresenta altos percentuais de cura, se for detectado precocemente. Entre os tumores de pele, é o de maior incidência e mais baixa mortalidade. São mais comuns em pessoas com mais de 40 anos de pele clara, ou com doenças cutâneas prévias, sendo relativamente raro em crianças e negros, exceto aqueles já portadores de doenças cutâneas anteriores pois são mais sensíveis à ação dos raios solares. Como a pele é heterogênea, o câncer de pele não-melanoma pode apresentar tumores de diferentes linhagens (INCA, 2018c).

Uma pessoa morre a cada 3 horas por câncer de pele: O tumor de pele matou 3.316 brasileiros somente em 2013... 4,5 milhões de brasileiros já tiveram câncer de pele... e em dez anos, o número de mortes por câncer de pele cresceu 55% no país. (EXAME, 2016).

Sobre um dos problemas de saúde consequentes ou agravados pela exposição exagerada a UV Taylor (1999) esclarece, que a epidemiologia² mostra que a etiologia³ da catarata é complicada e envolve muitos fatores de risco, entre os quais alguns muito proeminentes, como abuso de álcool, tabagismo pesado, diabéticos e uso crônico de esteroides. Ao considerar todos os fatores de risco, conclui que as únicas intervenções preventivas eficazes parecem ser parar de fumar e reduzir a exposição ocular à radiação UVB.

Para o Brasil, estima-se 85.170 casos novos de câncer de pele não melanoma entre homens e 80.410 nas mulheres para cada ano do biênio 2018-2019. Esses valores correspondem a um risco estimado de 82,53 casos novos a cada 100 mil homens e 75,84 para cada 100 mil mulheres. É o mais incidente em ambos os sexos. O câncer de pele não melanoma é o mais incidente em homens nas Regiões Sul (160,08/100 mil), Sudeste (89,80/100 mil) e Centro-Oeste (69,27/100 mil). Nas demais Regiões, Nordeste (53,75/100 mil) e Norte (23,74/100 mil), encontra-se na segunda posição. Entre as mulheres, é o mais incidente em todas as Regiões do país, com um risco estimado de 97,46/100 mil na Região Sul; 95,16/100 mil na Sudeste; 92,66/100 mil na Região Centro-Oeste; 45,59/100 mil na Região Nordeste; e 27,71/100 mil na Região Norte. Quanto ao melanoma, sua letalidade é elevada, porém sua incidência é baixa (2.920 casos novos em homens e 3.340 casos novos em mulheres). As maiores taxas estimadas em homens e mulheres encontram-se na Região Sul (INCA, 2018a).

Rezende, Corrêa e Duarte (2009) demonstram em sua pesquisa feita somente para o estado de São Paulo que grande parte dos custos com o tipo mais grave de câncer de pele, mais de 95%, é decorrente de tratamentos de estágios mais avançados da doença, que demandam medicamentos caros cujo custo do tratamento de um paciente pode cobrir o

² Ramo da medicina que estuda os diferentes fatores que intervêm na difusão e propagação de doenças, sua frequência, seu modo de distribuição, sua evolução e a colocação dos meios necessários a sua prevenção

³ Ramo do conhecimento cujo objeto é a pesquisa e a determinação das causas e origens de um determinado fenômeno

tratamento de vários dos estágios iniciais da doença. Demonstrando então, que a prevenção é importante tanto para salvar vidas quanto por questões financeiras. Ressalta ainda que o diagnóstico em estágios da doença tem também esse efeito, gerando considerável economia de recursos tanto para o sistema público quanto para o sistema privado de saúde e que esse recurso adicional poderia ser aplicado em programas de prevenção e diagnóstico precoce do câncer de pele ou, ainda, investido em pesquisas de novas técnicas e medicamentos para o tratamento desses pacientes.

2.1.2 Fatores de Risco

Vários autores vêm se dedicando à condução de pesquisas com o objetivo de elucidar a etiologia do melanoma maligno de pele. Rhodes et al. (1987) afirmam que os fatores de risco mais importantes, em ordem decrescente, seriam:

- Presença de nevo¹ transformado ou em transformação;
- Idade adulta;
- Existência de lesões pigmentares irregulares (inclusive nevo displásico e lentigo maligno);
- Existência de um nevo congênito;
- Ser de raça caucasiana;
- História de melanoma cutâneo prévio;
- Relato de melanoma cutâneo na família;
- Imunodepressão
- Sensibilidade ao sol; e
- Exposição solar excessiva.

Ao considerar que tanto as características individuais para cada tipo de pele como a exposição a UV têm implicações na etiologia do melanoma, a caracterização e a forma de interação destes dois fatores de risco no Brasil merecem uma análise cuidadosa. Não havendo dúvidas de que os indivíduos de origem europeia residentes em clima tropical ou temperado apresentam um risco elevado de desenvolver câncer cutâneo. Grande parte dos imigrantes europeus e seus descendentes que se estabeleceram no Brasil se dedicaram ao trabalho rural. Nas regiões sudeste e sul existem comunidades agrícolas com grande

¹ Nevos são pequenas manchas marrons regulares na pele, salientes ou não. São popularmente conhecidos por pintas e sinais.

concentração de pessoas de pele clara que se expõem ao sol continuadamente, muitas vezes desde a primeira infância. Além disto, a exposição solar decorrente de atividades esportivas de lazer é bastante comum em todos os extratos sociais de todo o território nacional. Dessa forma o risco para câncer de pele está presente na realidade brasileira o que justifica a tomada de medidas eficazes que possam prevenir e incentivar o diagnóstico e tratamento precoce dos problemas de saúde oriundos da exposição a UV (MENDONÇA et al., 1992).

2.1.3 Proteção

Segundo González, Fernández-Lorente e Gilaberte-Calzada (2008), fotoproteção é um elemento profilático e terapêutico frente aos efeitos danosos da UV. Esta abordagem é realizada por meio do uso de protetores solares, vestimentas protetoras e exposição restrita à luz solar. A primeira linha de defesa contra estes efeitos nocivos é a utilização dos fotoprotetores, também denominados protetores solares.

A necessidade da fotoproteção é uma realidade irrefutável, seja pela diminuição da incidência de câncer de pele ou por outros fatores. A longo dos anos, a evolução no desenvolvimento de fotoprotetores, visando à obtenção de formulações seguras e eficazes, capazes de fornecer proteção UV ampla cresceu, mas estudos aprofundados sobre segurança, eficácia e absorção sistêmica ainda são importantes para o completo entendimento das interações envolvidas com o uso dos protetores solares, ferramentas essenciais e indispensáveis diante dos danos provocados pela radiação UV (BALOGH et al., 2011).

2.1.3.1 UV Index

O UVI é formulado usando a Comissão Internacional de Iluminação (CIE) espectro de ação de referência para eritema induzido por UV na pele humana. É uma medida da radiação UV que é relevante e definida para uma instalação horizontal superfície. O UVI é uma quantidade sem unidade definida pela fórmula (WHO; ICNIRP et al., 1995):

$$I_{UV} = K_{er} * \int_{250nm}^{400nm} E_\lambda * s_{er}(\lambda) d\lambda$$

Onde E_λ é a irradiância espectral solar expresso em $W/(m^2 * nm)$ no comprimento de onda λ e $d\lambda$ é o intervalo de comprimento de onda usado no somatório. $s_{er}(\lambda)$ é a referência de eritema espectro de ação, e K_{er} é uma constante igual a $40m^2/W$.

A determinação do UVI pode ser feita através de medições ou cálculos de modelo. Duas abordagens de medição podem ser adotadas: A primeira consiste em usar um espectrorradiômetro e calcular o UVI usando a fórmula acima. O segundo é usar um detector banda larga que foi calibrado e programado para que dê o valor de UVI diretamente (WHO; ICNIRP et al., 1995).

O UVI descreve o nível de radiação UV solar na superfície da Terra. Os valores do índice variam de zero para cima - quanto maior o valor do índice, maior o potencial de danos à pele e aos olhos e menos tempo leva para que ocorram danos. Enquanto estudos de conscientização como o UVLoRa acontecem a WHO faz algumas recomendações de proteção, associando os vários tipos de fotoproteção . Tabulando esta informação, obtém-se o resultado mostrado na tabela 1:

Tabela 1 – Recomendações de fotoproteção da WHO

UVI	FORÇA	RECOMENDAÇÕES	TEMPO DE EXPOSIÇÃO
0.0–2.9	Baixo	Usar um chapéu é proteção suficiente.	Mais de 1 hora
3.0–5.0	Moderado	Um chapéu e um protetor solar com FPS 15 é recomendado.	Cerca de 30 Min.
5.1–7.9	Alto	Vestindo um chapéu, um protetor solar com FPS 30 e ficar na sombra é recomendado.	Cerca de 20 min.
8.0–10.9	Muito Alto	Além das precauções recomendadas acima, recomenda-se ficar em casa entre 10:00h e 16:00h	Menos de 15 min.
11.0+	Extremo	Além das precauções recomendadas acima, é aconselhado a ficar em casa se possível.	Menos de 5 min.

Fonte: (WHO, 2018).

“Orientações farmacêuticas na luta contra o câncer de pele”, foi uma campanha realizada no mês de novembro de 2015 e março de 2016, na Universidade Federal de Alfenas, Minas Gerais/Brasil. Em estudo descritivo, realizado com base nas informações coletadas durante a campanha: Os autores do estudo observam a insuficiência de informação mesmo em comunidades acadêmicas e a falta de adesão ao uso dos fotoprotetores, desconsiderando o alto índice de câncer de pele existente no Brasil, fato que motiva a realização de outras campanhas com o objetivo de divulgar informações a respeito dos riscos da exposição desprotegida ao sol. Concluem que abordagens precoces sobre a fotoproteção podem contribuir para a conscientização relacionada ao uso regular e adequado dos fotoprotetores, visando à proteção e prevenção do fotoenvelhecimento, de cataratas e, principalmente, do câncer de pele (SILVA et al., 2017).

2.2 Internet das coisas (IoT)

A internet ganhou o mundo à algum tempo, e hoje alcança lugares nunca antes imaginados, claramente é um dos maiores feitos da humanidade considerando-se o impacto que ela causou na educação, nos negócios, na própria humanidade e principalmente nas formas de comunicação. A Internet das Coisas, ou simplesmente IoT (do inglês Internet of Things) representa a próxima evolução da internet, e já é uma realidade, representando um grande avanço na coleta, distribuição e análise de dados (EVANS, 2011).

Segundo a Cisco Systems², a IoT “surgiu” entre 2008 e 2009, quando o número estimado de dispositivos conectados à internet, cerca de 12,5 bilhões, passou a ser maior que o número de pessoas no planeta, que na mesma época contabilizava cerca de 6,8 bilhões, uma média próxima de 1,8 dispositivos por pessoa. Desde então, tem avançado a passos lentos, se comparado ao que era esperado. Para o futuro a Cisco estima que 50 bilhões de dispositivos estejam conectados em 2020 (CISCO, 2011).

Segundo a International Business Machines Corporation (IBM) produzimos mais de 2,5 quintilhões de bytes de dados todos os dias, no entanto 80% deles são desestruturados, portanto invisíveis para a tecnologia atual, como diz a empresa, estamos iniciando Era Cognitiva com a IoT (WATSON, 2017). Varias empresas como a IBM tem projetos envolvendo IoT em desenvolvimento, e prometem revolucionar o modo como a humanidade se relaciona com as informações, estas empresas no entanto tem um interesse, muitas vezes econômicos, e até que seus produtos e ou serviços cheguem à maioria da população um bom tempo é demandado.

KUROSE, ROSS e ZUCCHI (2010) definem internet como “uma rede de computadores que interconecta milhares de dispositivos ao redor do mundo.” enquanto a Web é definida como uma infraestrutura que provê serviços e aplicações distribuídas. Esta informação é importante para entendermos que IoT envolve plataformas de hardware e software para permitir que dados sejam coletados, distribuídos, processados e então analisados.

2.2.1 Hardwares

O hardware está associado à parte física do sistema, ou seja, os circuitos e dispositivos que coletam e suportam o processamento das informações.

Os recursos e projetos para a criações de IoT, inclusive open-sources³ vem crescendo e se popularizando, como o é o caso do Arduíno lançado em 2005, que trata-se de uma plataforma de hardware e software para microcontroladores. Os microcontroladores, por

² Cisco Systems - Companhia transnacional estadunidense do ramo de tecnologia em soluções para redes e comunicações

³ Open Source Iniciative - OSI (2019), Disponível em: <<https://opensource.org/>>

sua vez, são pequenos, de baixo custo e possuem placa única de circuito com Processador, Memória RAM, Memória ROM além de entradas e saídas analógicas e digitais (ARDUINO, 2015).

Existem diversos tipos de placas Arduíno oficiais, cada uma com suas particularidades, e ainda diversos outros tipos placas não oficiais. A esses dispositivos é possível conectar sensores e atuadores, que estendem as funcionalidades do dispositivo, tornando-o capaz de coletar (sensores) dados do mundo real como temperatura, umidade, pressão, aceleração, campo magnético, luz visível, infravermelhos, radiação ultravioleta, etc. ou atuar (atuadores) sobre o meio em que está inserido exibindo mensagens, acionando motores, interruptores, etc (MCROBERTS, 2015) (ARDUINO, 2015). O LoPy4 é uma placa de desenvolvimento compatível com Micropython de com conectividade sem fio quádrupla LoRa, Sigfox⁴, WiFi e Bluetooth. Com chipset Espressif ESP32⁵.

Assim como os microcontroladores, os sensores constituem uma vasta coleção onde é possível encontrar itens de diversos tipos e finalidades. O SI1145 é um exemplo, este sensor óptico foi desenvolvido pela Silicon Labs (2019) capaz de ler valores de Vis, IR e UV. Ele pode ser facilmente conectado a um microcontrolador via conexão I2C⁶. Ele extrapola⁷ a medida do UVI com base nas suas leituras de Vis e IR, portanto pode gerar falsos positivos, de acordo com ambiente que se encontra. Outros sensores do mesmo tipo como: ML8511 e GUVA-S12SD que são sensores analógicos⁸ de UV.

Em um ambiente profissional, as características dos microcontroladores fazem deles uma excelente ferramenta de prototipação rápida e de projeto simplificado. Por outro lado, em um ambiente acadêmico, ele pode ser perfeitamente utilizado como ferramenta educacional ou de pesquisa, uma vez que não requer do usuário conhecimentos profundos de eletrônica digital nem da programação de dispositivos digitais específicos (RENNNA et al., 2013).

2.2.2 Softwares

Os softwares para IoT correspondem ao conjunto de programas responsáveis pela execução das tarefas. E dentro deste tópico pode-se destacar: os **drivers**, que funções a serem integradas a um sistema operacional para controlar um determinado periférico,

⁴ SigFox - SigFox (2019). Disponível em: <<https://www.sigfox.com/en>>

⁵ ESP32 - Espressif (2019). Disponível em: <<https://www.espressif.com/en/products/hardware/esp32>>

⁶ O protocolo I2C descreve o funcionamento de um barramento de comunicação serial que utiliza apenas dois fios, inventado pela Philips no início da década de 90, este protocolo é muito utilizado para conectar periféricos de baixa velocidade a placas-mãe, microcontroladores e afins.

⁷ Dicionário: generalizar com base em dados parciais ou reduzidos; estender a validade de uma afirmação ou conclusão além dos limites em que ela é comprovável; exceder.

⁸ Como a UV é uma grandeza analógica já que pode assumir "infinitos" valores de amplitude dentro de uma faixa de valores. Dessa forma, um sensor analógico emite por sua saída uma quantidade energia que representa proporcionalmente, o valor real da grandeza, enquanto que sensores digitais já emitem um valor que por eles é entendido como real.

para o uso dos diversos tipos *shields* mais complexos; os **códigos de programação dos microcontroladores** que podem ser comparados aos sistemas operacionais, já que as dão as instruções ao hardware e disponibilizam uma interface padrão de comunicação; e as **plataformas de gerenciamento destes dispositivos**, pois normalmente os mesmos se encontram em grupos de dispositivos onde cada um tem uma função própria e todos juntos um objetivo único.

A programação dos microcontroladores atualmente, pode ser feita usando linguagens de programação adaptadas a este meio, como é caso da linguagem própria da plataforma Arduíno, dispensando o uso da linguagem de baixo nível Assembly que até então facilitou significativamente o desenvolvimento de *firmwares*, uma classe específica de software que fornece controle de baixo nível para o hardware específico do dispositivo. Ela nada mais é que uma versão legível da linguagem de máquina e portanto é orientada para a mesma ou melhor, para o processador, e assim sendo, seria necessário conhecer a estrutura do processador para poder programá-lo com Assembly. Essa linguagem utiliza instruções de baixo nível que operam diretamente com registradores e memórias, ou seja, as instruções são diretamente executadas pelo processador (SEBESTA, 2018).

Micropython vem com a proposta de deixar a programação de microcontroladores ainda mais simples usando o poder e a simplicidade da linguagem de programação Python, trata-se de uma nova linguagem de programação para microcontroladores, baseada em Python 3 e criada pelo Físico inglês Damien George lançada em 2013. A linguagem é muito mais simples se comparada a linguagem da plataforma Arduíno, que é baseada em Wiring que por sua vez é baseada em C/C++ e portanto se assemelha muito as suas descendentes em alguns pontos. Alguns dispositivos já trazem Micropython embarcado e há ainda diversos outros compatíveis tanto com Arduíno quanto com Micropython e outras linguagens (MICROPYTHON, 2018) (GEORGES, 2015).

Atualmente com o claro avanço da IoT, diversos *IoT Clouds*, sugiram e surgem, e muitos morrem pelo caminho. Dentre diversos disponíveis alguns se destacam seja pela abordagem, seja pela qualidade de serviço oferecido. No entanto as plataformas que não requerem uma hospedagem ou configuração local para funcionar, quando oferecem um plano gratuito ele é muito limitado. Geralmente, se fala *IoT Cloud* para indicar um novo tipo de sistema distribuído que consiste em um conjunto de dispositivos inteligentes interconectados com uma infraestrutura, plataforma ou software na nuvem remota por meio da Internet e capazes de fornecer IoT como um serviço. Como é o caso da plataforma do Google que está entre as melhores plataformas disponíveis atualmente. Seu foco principal é tornar as coisas fáceis e rápidas. No entanto o serviço é cobrado e seu preço é feito por minuto, o que o torna mais barato que outras plataformas. Seguindo essa ideia, a IBM, Microsoft, Cisco, Amazon dentre diversas empresas no ramo de tecnologia, têm plataformas similares de *IoT Cloud*, cada uma com suas particularidades (RANA, 2018).

Plataformas como o Thingspeak, permitem coletar e armazenar dados de sensores na nuvem. Fornecendo uma aplicação para analisar e visualizar os dados no Matlab. Permite a criação um canal separado para cada dado que deseja armazenar, permite ter canais públicos e privados além de ter hospedagem grátis para alguns canais, visualização fácil e recursos adicionais para Ruby, Node.js e Python. API⁹ RESTful¹⁰ e MQTT¹¹ para comunicação (RANA, 2018) (THINGSPEAK, 2019). Outros como Ubidots¹² oferecem uma aplicação muito simples e completa, no entanto gratuitamente só permite o uso por 30 dias.

2.2.3 Tecnologias de comunicação

A diferença essencial entre “a Internet” e “a Internet das coisas” é que na IoT, há apenas "menos de tudo" disponível em um determinado dispositivo: menos memória, menos capacidade de processamento, menos largura de banda, etc.; e, claro, menos energia disponível. Isto é porque as “coisas” são normalmente impulsionadas por uma bateria e a maximização do tempo de vida é uma prioridade. Esse impulso para "fazer mais com menos" leva a restrições que limitam a aplicabilidade das redes celulares tradicionais, bem como de tecnologias, como o WiFi, devido aos requisitos de energia e escalabilidade (AUGUSTIN et al., 2016). Mas o fato é que devido ao crescimento exponencial no domínio de semicondutores que resultou em uma explosão de padrões de uso de sistemas de processamento econômicos e esses sistemas quando se fortalecem com tecnologias avançadas de comunicação como por exemplo, Bluetooth, WiFi, 3G, etc. entre outras, convergem para uma forma emergente de domínio tecnológico - a Internet das Coisas (RAY, 2016).

Então outra gama de protocolos e tecnologias surgiu para atender aos requisitos de comunicação da IoT: As Redes de Área Ampla e Baixo Consumo (LPWAN, do inglês *Low Power Wide Area Network*). Coloquialmente falando, um LPWAN deveria ser para a IoT o que o WiFi é para as redes de consumidores. Ele deve oferecer cobertura de rádio sobre uma área grande através de estações base e adaptar taxas de transmissão, potência de transmissão, modulação, ciclos de trabalho, etc. de tal forma que os dispositivos finais incorrem em um consumo de energia baixo devido à sua conexão. LoRa é um desses protocolos LPWAN e tem como alvo implantações onde os dispositivos finais têm energia limitada e não precisam transmitir mais do que alguns bytes por vez (AUGUSTIN et al., 2016).

⁹ API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos" (TECH, 2019).

¹⁰ RESTful: capacidade de determinado sistema aplicar os princípios de REST (RICHARDSON; RUBY, 2008).

¹¹ MQTT — A publish/subscribe protocol for Wireless Sensor Networks (HUNKELER; TRUONG; STANFORD-CLARK, 2008).

¹² Ubidots (UBIDOTS, 2019) - <https://ubidots.com/>

A tecnologia LoRa permite comunicações em longas distâncias, gastando muito pouco de energia. O nome da tecnologia foi criado pela LoRa Alliance, um consócio criado para incentivar o uso da tecnologia. Usa uma potência muito baixa, da ordem de 20dbm ou 100mW. Além disso, por possuir uma especificação de camada lógica mais simples (forma de organizar o envio da informação), a tecnologia LoRa se torna ideal para o universo da internet das coisas pois ela é mais flexível (SINHA; WEI; HWANG, 2017).

2.2.4 LoRa

LoRa opera em uma banda não licenciada abaixo de 1 GHz para operação de link de comunicação de longo alcance. É um esquema proprietário de modulação de espectro de dispersão que é derivado da modulação de espectro de dispersão chirp¹³ (CSS, do inglês: *chirp spread spectrum*) e que troca taxa de envio de dados por sensibilidade dentro de uma largura de banda de um canal fixo. O CSS, desenvolvido na década de 1940, era tradicionalmente usado em aplicações militares por causa de suas longas distâncias de comunicação e robustez anti interferência. O LoRa é sua primeira implementação de baixo custo para uso comercial (SINHA; WEI; HWANG, 2017), pois usa chirps com uma variação linear ao longo do tempo para codificar informações. Devido à linearidade dos pulsos de chirp, os desvios de frequência entre o receptor e o transmissor são equivalente a deslocamentos de tempo, e dessa forma facilmente eliminados no decodificador. Isso também torna essa modulação imune ao efeito Doppler¹⁴, equivalente a um deslocamento de frequência. O deslocamento de frequência entre o transmissor e o receptor pode atingir 20% da largura de banda sem impactar desempenho de decodificação, isso ajuda a reduzir o preço dos transmissores LoRa, pois os cristais embutidos nos transmissores não precisam serem fabricados com extrema precisão. Os receptores LoRa são capazes de bloquear os sons de frequência recebidos, oferecendo uma sensibilidade na ordem de -130 dBm.

LoRa possui 3 possibilidades de formas de uso atualmente: O LoRa-MAC, que é onde o rádio é usado diretamente e os dados enviados não são formatados ou criptografados de nenhuma forma e nenhuma informação de endereçamento é adicionada ao pacote; o LoRa-WAN de modo contrário ao LoRa-MAC, possui criptografia, endereçamento e outras funcionalidades; e existe ainda o LoRa-MESH uma implementação de rede Mesh utilizando LoRa.

2.2.4.1 LoRaWAN

LoRaWAN™ é o nome dado a um protocolo que define a arquitetura do sistema bem como os parâmetros de comunicação usando a tecnologia LoRa®. O protocolo

¹³ Um chirp é um sinal no qual a frequência aumenta (up-chirp) ou diminui (down-chirp) com o tempo (PURI, 2019)

¹⁴ O que é o efeito Doppler? (BRASILESCOLA, 2019) - <https://brasilescola.uol.com.br/o-que-e/fisica/o-que-e-efeito-doppler.htm>

LoRaWAN™ implementa os detalhes de funcionamento, segurança, qualidade do serviço, ajustes de potência visando maximizar a duração da bateria dos módulos, e os tipos de aplicações tanto do lado do módulo quanto do servidor. A arquitetura da rede LoRaWAN é composta pelos seguintes itens em uma abordagem de baixo para cima, ou seja, dos itens mais externos da rede para os mais internos (AUGUSTIN et al., 2016). A figura 2 ilustra esta dinâmica.

1. **Módulos ou dispositivos finais:** São os elementos básicos da rede, e normalmente dispositivos responsáveis por sensores de temperatura, movimento, interruptores, leitores de consumo de energia, de água, gás, etc.
2. **Gateways:** São os elementos de conexão entre os módulos e os servidores de rede. Um só gateway pode receber os dados de vários dispositivos e encaminha-los para o servidor de rede. Dependendo das condições de topologia do local, um único gateway pode cobrir um raio desde 2Km até 15 Km.
3. **Servidores de rede:** Os servidores de rede são os responsáveis pelo gerenciamento das informações enviadas pelos gateways.
4. **Servidores de aplicações:** Os servidores de aplicações são programas específicos que recebem os pacotes dos servidores de rede e de acordo com a informação executam uma ou mais ações específicas.

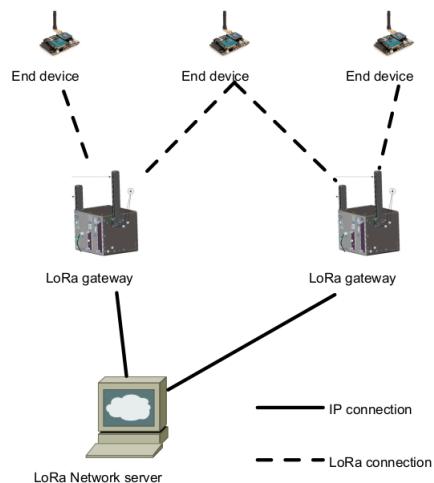


Figura 2 – Arquitetura LoRaWAN

Fonte: (AUGUSTIN et al., 2016).

2.2.4.2 LoRaMESH

O LoRa-MESH por sua vez é uma implementação de Rede Mesh que são redes em malha sem fio auto-configuráveis e de crescimento orgânico utilizando LoRa. Recentemente vêm sendo consideradas para a criação de infra-estrutura de baixo custo para a construção de redes de acesso comunitárias e de cidades digitais, mas ainda é um recurso pouco maduro, no entanto seria a forma ideal para comunicação entre dispositivos UVLoRa. O uso desta forma foi descartado devido a sua pouca maturidade e a sua não possibilidade deixar os aparelhos em modo de consumo mínimo de energia em versão de código aberto. A versão que permite este tipo de ação já está disponível no entanto não possui o código aberto (PYCOM, 2019f).

2.2.4.3 LoRaMAC

Optou-se então, por utilizar LoRa-MAC devido a sua extrema flexibilidade que em contra partida necessita o desenvolvimento de toda a lógica e protocolos de comunicação necessários entre gateways e nós. UVLoRa utiliza protocolo ALOHA desenvolvido na década de 1970 por Norman Abramson na Universidade do Havaí para transmissão de dados via LoRa. O protocolo ALOHA permite que um host utilize o meio de transmissão sempre que tiver necessidade de enviar um dado pela rede. Uma vez que o meio é compartilhado podem ocorrer colisões e o dado é perdido. O ALOHA é portanto um melhor serviço, e não garante que os dados cheguem ao recipiente remoto sem estar corrompido. Uma rede ALOHA somente trabalha bem quando o meio tem uma baixa utilização como na UVLoRa, tendo consequentemente uma baixa probabilidade de colisão e uma menor probabilidade de que os dados estejam corrompidos (TANNENBAUM, 1997).

Para se atingir uma comunicação eficiente e confiável, vários parâmetros LoRa podem ser configurados como:

- SF (do inglês, *Spreading Factor*): o fator de espalhamento, que permite os possíveis valores 7, 8, 9 ,10, 11 e 12.
- BW (do inglês, *Bandwidth*): a largura de banda, sendo possível 125 kHz ou 500 kHz.
- CR: o *Code Rate*, cujos valores podem ser 4/5, 4/6, 4/7 ou 4/8.
- TP (do inglês, *Transmission Power*): a potência de transmissão, podendo ser um valor entre 0 e 20 dBm.
- F: A frequência utilizada.

A frequência escolhida não possui um impacto direto na qualidade da transmissão, sendo mais relevante em locais com diversos dispositivos LoRa operando, onde a escolha de uma frequência menos congestionada torna-se importante. Na UVLoRa foi selecionada

a frequência 925.5 MHz. Definida a frequência, ainda é necessário decidir qual SF, BW, CR e TP serão utilizados dentre centenas de combinações possíveis, tornando impraticável testar cada uma. É preciso determinar a configuração que minimiza o custo de energia de transmissão e atenda ao desempenho e especificações de comunicação necessários e para isso é necessário entender o impacto da seleção de parâmetros da transmissão.

A TP tem um impacto direto no consumo de energia já que quanto maior a potência maior o consumo de energia. Outro fator que determina o consumo de energia é o tempo necessário para transmitir um pacote (*AirTime*) do Nô ao Nano Gateway, que depende da combinação de CR, BW e SF selecionados. A influência desses parâmetros no *AirTime* do pacote segue a da seguinte forma: a diminuição do SF resulta em um *AirTime* menor, enquanto o aumento do SF resulta em um *AirTime* maior; a diminuição da BW resulta em um *AirTime* maior, enquanto que o aumento da BW resulta em um *AirTime* menor; um CR igual a 4/5 possui um menor *AirTime* que um CR de 4/8.

Para a UVLoRa, utilizou-se o software de LoRa AirTime Calculator (JúNIOR, 2019), para a realização de tal cálculo, levando em consideração que as regulamentações da Agência Nacional de Telecomunicações (ANATEL) para utilização da tecnologia no Brasil. SF=10, BW = 125 kHz, CR = 4/8 é a configuração de alcance máximo para LoRa no Brasil respeitando os 400 ms de *AirTime* da ANATEL, o que resulta em uma taxa de envio de dados de 610,35 bps e um *AirTime* 0,29696s para pacote com 9 bytes, utilizado pela UVLoRa. Tal configuração suporta pacote de até 14 bytes sem ultrapassar os 400 ms para uso do mesmo canal, dessa forma ainda restam 5 bytes como margem de crescimento para enviar mais informações no pacote o que permite no futuro adicionar outros sensor por exemplo. A escolha de um CR = 4/5 diminui um pouco o alcance, aumenta a taxa de envio de dados para 976,56 bps e o pacote máxima aumenta para 24 bytes o que pode ser útil para o recebimento de mensagens de texto pelo Nô.

2.2.5 Aplicações

Já é possível ver aplicações práticas da internet das coisas na organização do trânsito, na agilização de tratamentos médicos e também na preservação do meio ambiente mundo a fora, sempre condicionada à capacidade humana de analisar os dados que os dispositivos conectados geram. A pouco tempo atrás o Fórum Mundial Econômico (WEF, do inglês: *World Economic Forum*) listou seis áreas nas quais a IoT já faz toda a diferença (WEF, 2018).

2.2.5.1 Cidades mais inteligentes

Hoje, mais da metade da população mundial já vive em ambientes urbanos. Em 2050, a previsão da ONU é que a proporção suba para dois terços. Por isso, é fundamental

cuidar para que as cidades sejam lugares sustentáveis e bem organizados, que suportem o peso das mudanças climáticas e a chegada de mais milhões de habitantes (ONU, 2014).

A internet das coisas vem ajudando várias cidades a cumprir esse objetivo. Em Barcelona, na Espanha, o uso de água para irrigação em jardins e fontes públicas já é controlado digitalmente, evitando desperdícios. O mesmo acontece com o sistema de iluminação pública, que tem postes dotados de sensores de presença, usados como roteadores para conexão Wi-Fi (ADLER, 2016).

Também em Barcelona, um sistema implantado nas vias públicas avisa os motoristas sobre lugares disponíveis para estacionar seus carros. Por meio de sensores no asfalto, sinais são emitidos para um aplicativo, ajudando o motorista a estacionar rapidamente, o que reduz o trânsito e as emissões de gases pelos veículos (ADLER, 2016).

2.2.5.2 Limpeza do ar e da água

Cidades que sofrem muito com a poluição têm direcionado esforços para melhorar a qualidade do ar e da água. Em Londres, onde 9 mil pessoas morrem anualmente em função de problemas respiratórios, a Drayson Technologies¹⁵ está distribuindo para os cidadãos pequenos aparelhos que medem o nível de poluição do ar. Eles podem ser plugados em carros e bicicletas, circulando junto com os veículos pela cidade. Os sensores transmitem as informações para o aplicativo da empresa. O aplicativo, por sua vez, consolida as informações num único servidor, permitindo aos moradores da cidade conferir um mapa digital da qualidade do ar em cada ponto da cidade.

Uma ideia semelhante foi levada a Oakland, na Califórnia, pela *startup* Aclima¹⁶, em parceria com o Google. Nesse caso, os sensores foram distribuídos pelos carros do Google Street View, e as informações ficarão disponíveis para que os especialistas trabalhem em ações para reduzir a poluição no ar (ACLIMA, 2018).

2.2.5.3 Agricultura mais eficiente

O campo também se beneficia da internet das coisas. Drones que fazem imagens aéreas e sensores de qualidade do solo ajudam os produtores a identificar os melhores locais para plantar as novas safras. Esses recursos já estão presentes também no Brasil. *startups* como a Agrosmart¹⁷ instalam junto às plantações sensores meteorológicos que identificam indicadores como a radiação solar, direção do vento, pressão barométrica, etc.

¹⁵ Empresa de desenvolvimento de hardware e software. A empresa se concentra no desenvolvimento de dispositivos de energia sem fio e redes de sensores sem fio, em Londres, Reino Unido. <https://www.getfreevolt.com/>

¹⁶ A Aclima fornece dados e informações hiperlocais da qualidade do ar para melhorar a saúde humana e planetária. Sediada em São Francisco, California (ACLIMA, 2018). <https://aclima.io/contact/>

¹⁷ Startup de tecnologia agropecuária (AGROSMART, 2019). <https://agrosmart.com.br/>

O mapeamento aéreo com o uso de drones¹⁸ também já é usado por aqui, assim como tecnologias para máquinas semeadeiras, que mostram em tempo real aos controladores se toda a extensão do solo está sendo usada de forma adequada¹⁹.

2.2.5.4 Combatendo o câncer de mama

Com previsão de 59,7 mil novos casos entre as mulheres brasileiras no biênio 2018-2019, segundo o Inca (2019), o câncer de mama já é alvo de diversas campanhas de conscientização no programa Outubro Rosa. Mas o combate pode ser potencializado pela internet das coisas.

A mamografia tradicional pode falhar em identificar a doença nos estágios iniciais. Para resolver o problema, a Cyrcadia Health²⁰ desenvolveu a ITBra. O equipamento consiste em um top com microssensores que identificam mínimas variações de temperatura na região dos seios. Ao transmitir as informações para o smartphone da usuária ou para o médico, os dispositivos ajudam os profissionais da saúde a identificar padrões que possam representar um perigo para a saúde da mulher. A Cyrcadia está testando a solução na Ásia, onde questões culturais impedem uma conscientização mais ampla e tornam o câncer de mama ainda mais letal. Espera-se que, em breve, a empresa leve seu produto para outros países (CYRCADIA, 2019) .

¹⁸ Caso da capixaba Imaginair, empresa da cidade de Castelo especializada em imagens aéreas e mapeamentos (IMAGINAIR, 2019). <http://www.imaginair.com.br>

¹⁹ A Trimble, empresa multinacional no ramo maquinaria agrícola inteligente, tem vários revendedores no Brasil (TRIMBLE, 2019). <https://www.trimble.com.br/intl/br/encontre-um-revendor.aspx>

3 Metodologia

3.1 Classificação e delineamento da pesquisa

Esta pesquisa, classificada como exploratória e aplicada e porquanto realiza análises em áreas na qual há poucas informações sistematizadas e visa produzir um conhecimento que possa ser efetivamente aplicado na vida real, ajudando a alterar uma situação, fenômeno ou sistema. A principal fonte de dados será o estudo de caso a ser realizado.

3.2 A dinâmica de funcionamento do UVLoRa

Para a construção deste projeto são necessários basicamente microcontroladores com conectividade LoRa e WiFi, sensores de UV. Um microcontrolador com um sensor UV associado, coleta e envia o UVI a um gateway por uma conexão LoRa. Este gateway, nada mais é que outro microcontrolador conectado a internet via WiFi, que tanto disponibiliza os dados recebidos dos nós na internet, quanto envia informações úteis aos nós. O Infográfico na figura 3 ilustra essa estrutura que possibilita esta dinâmica.

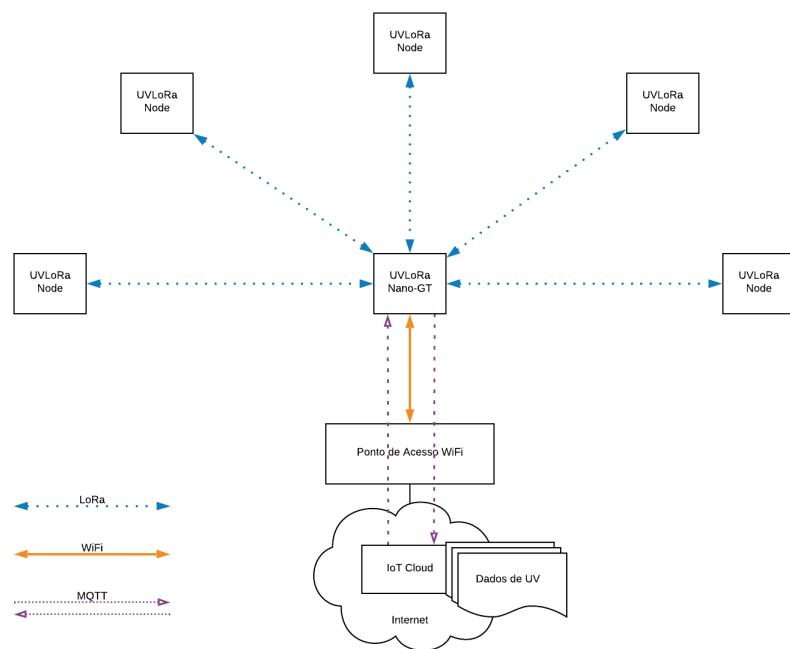


Figura 3 – Diagrama da estrutura da Rede UVLoRa.

Fonte: O Autor.

Portanto são necessários no mínimo 2 microcontroladores, um será nano gateway

e outro o nó. Um nano gateway LoRa(RAW) pode atender em topologia estrela até aproximadamente 100 nós com sensores dentro de uma região de cobertura de vários quilômetros. Se necessário mais alcance, um nano gateway pode se conectar a um outro nano gateway, até que se atinja uma região com cobertura WiFi ou 3G e energia elétrica externa disponível. E um nó com um sensor UV e uma tela para a visualização das medições, equipado com uma bateria que permita o seu carregamento via porta micro USB.

3.3 Dispositivos de Hardwares

Segundo a dinâmica apresentada, é possível com a tecnologia disponível atual combinar diversos tipos de hardwares e obter-se o mesmo resultado, dessa forma este projeto pode ser adaptado de diversas formas, para ser mais barato, mais preciso ou simplificado, dependo dos dispositivos escolhidos para formar o arranjo.

Optando pela simplicidade no desenvolvimento, o LoPy4 é uma ótima opção de microcontrolador, já que oferece uma combinação simpática e flexível e pode atuar como um nano gateway LoRa, ou seja, pode interfacear uma rede de nós conectados a ela com outra rede que usa protocolos diferentes (PYCOM, 2019a). No entanto ele é apenas um exemplo de hardware deste tipo, e de forma assim como ele existem outros capazes de cumprir esse mesmo desafio. A figura 4 é uma fotografia do dispositivo.



Figura 4 – LoPy4

Fonte: O Autor.

O sensor de UV pode ser um dos diversos disponíveis, cada um com sua particularidade, uns mais precisos e mais caros, outros menos precisos e mais baratos, digitais ou analógicos¹. O VEML6075² desenvolvido pela Vishay³ também utiliza conexão I2C e

¹ Um vasta lista de opções é retornada em sites de vendas como Aliexpress (ALIEXPRESS, 2019a). <http://encurtador.com.br/cgtZ5>

² Adafruit VEML6075 (ADAFRUIT, 2019a) - <https://www.adafruit.com/product/2899>

³ A Vishay Intertechnology foi fundada em 1962 pelo Dr. Felix Zandman, com um empréstimo de seu primo Alfred P. Slaner. A empresa recebeu o nome da vila ancestral do Dr. Zandman na Lituânia, em memória dos membros da família que morreram no Holocausto. O portfólio inicial de produtos da Companhia

mede o UVA e o UVB, além de fornecer o valor de UVI diretamente. A Figura 5 mostra uma fotografia da placa desenvolvida para o VEML6075.



Figura 5 – Adafruit VEML6075

Fonte: O Autor.

Este projeto visa o baixo custo, a fim de se tornar popular. Como a disponibilidade de sensores é grande, uma análise deve ser realizada a fim de encontrar um dispositivo de melhor custo benefício, isto é, que traga boa precisão pelo menor custo. Uma versão básica deste projeto pode custar até US\$300 dólares, aproximadamente R\$1.150 reais com taxas de importação utilizando os exemplos microcontrolador e sensor de UV, que podem ser considerados os mais relevantes hardwares neste projeto. Esta versão mais básica contém os seguintes componentes:

- 2 placas LoPy⁴, custando cerca de R\$160,00 reais cada;
- 2 placas Expansion Board⁵, uma para cada LoPy custando cerca de R\$70,00 reais cada;
- 1 placa com sensor ultravioleta VEML6075⁶ custando cerca de R\$ 25,00 reais;
- 2 caixas PyCase⁷ custando cerca R\$125,00 reais cada;
- 2 baterias 3,7V e 2200 mAh⁸ custando cerca de R\$ 25,00 reais cada no Brasil;
- 1 carregador de energia USB custando cerca de R\$25,00 reais no Brasil;
- fios custando cerca de R\$20,00 reais 50 peças ou mais.

consistia em resistores de folha e extensômetros de resistência à folha. Em 1985, tendo crescido de uma *start-up* para o fabricante líder mundial desses produtos, a Companhia iniciou uma série de aquisições estratégicas para se tornar uma fabricante de linha ampla de componentes eletrônicos. Hoje, a Vishay Intertechnology é um dos maiores fabricantes mundiais de semicondutores discretos e componentes eletrônicos passivos (INTERTECHNOLOGY, 2019). <http://www.vishay.com/company/about/>

⁴ LoPy4 (PYCOM, 2019c) - <https://pycom.io/product/lopy4/>

⁵ Expansion Board v3.0 (PYCOM, 2019d) - <https://pycom.io/product/expansion-board-3-0/>

⁶ Adafruit VEML6075 (ADAFRUIT, 2019a) - <https://www.adafruit.com/product/2899>

⁷ PyCase (PYCOM, 2019e) - <https://pycom.io/product/pycase-clear/>

⁸ Baterias 3,7V 2200mah (MERCADOLIVRE, 2019) - <http://abre.ai/atXU>

Um nó UVLoRa é responsável por coletar os dados de radiação UV e repassa-los ao nano gateway. Por tanto a ele é conectado o sensor de UV a ser usado. Como este dispositivo foi desenvolvido com o intuito de ficar em lugares remotos e ou de difícil acesso e nenhuma disponibilidade de energia, ele precisa ser munido de uma bateria. Nesta versão do dispositivo o sensor de UV fica posicionado na horizontal sobre a caixa de proteção do equipamento e sem proteção alguma, no entanto, deveria estar apontado para o sol para garantir uma medição mais exata do UVI ou ser protegido por um domo óptico que concentraria os raios rebeldos num único ponto. O apontamento do sensor seria possível utilizando *PAN TILT*⁹, confira figura 6, que são plataformas móveis para robótica, normalmente projetadas para a instalação de pequenas câmeras ou sensores, e neste caso a proteção do sensor poderia ser feita usando uma placa plana de material transparente a UV¹⁰, esta opção é logo descartada pela UVLoRa, já que os Nós precisariam alimentar este periférico que consome muita energia. A outra solução seria a o uso de um domo óptico transparentes a UV¹¹, conforme a Figura 7. Esta no entanto é uma solução bastante cara mas possível de ser utilizada pela UVLoRa. A ainda a possibilidade de uso de domos mais baratos, e possivelmente menos ou nada transparentes a UV, e neste caso uma compensação do UVI deve ser calculada, algo que exige muitos estudos e testes porém possível de ser feita.



Figura 6 – PAN TILT adquirido para testes

Fonte: O Autor.

Como cada nó foi projetado para ser instalado em um local de difícil acesso, alimentado por uma bateria, o código-fonte precisou ser adaptado para que se tenha

⁹ Um exemplo é o Suporte Pan/Tilt para Câmera Raspberry Pi Sparkfun (FILIPEFLOP, 2019) - <https://www.filipeflop.com/produto/suporte-pan-tilt-para-camera-raspberry-pi-sparkfun/>

¹⁰ Fused Silica Window Quartz Glass (ALIEXPRESS, 2019b) - <https://pt.aliexpress.com/item/32919984062.html?spm=2114.12010612.8148356.10.4a574834IR8HIV>

¹¹ BK7 & Quartz Optical Domes (PHOTONICS, 2019) - https://www.photonics.com/Products/BK7_Quartz_Optical_Domes/pr62965



Figura 7 – Exemplo de domo óptico

Fonte: (PHOTONICS, 2019)

o menor consumo de energia, alta confiabilidade e seja capaz de funcionar por longos períodos de tempo com a mínima manutenção. Dessa forma ele utiliza um recurso do LoPy4 chamado “Deep Sleep”, um modo de economia de energia (PYCOM, 2019b). Nesse modo, as CPUs, a maior parte da RAM e todos os periféricos digitais estão desligados. Permanecendo apenas o controle do relógio de tempo real (RTC, do inglês *Real Time Clock*) e seus periféricos. Temos várias maneiras de despertar o LoPy4 quando está adormecido, sendo que as fontes de despertar podem ser configuradas a qualquer momento antes de entrar no modo Deep Sleep, no UVLoRa utilizou-se um cronômetro (*timer*) para despertar o dispositivo (PYCOM, 2019h). Para facilitar o *debug* um *log* via conexão serial foi implementado em seu código, o recurso pode ser desabilitado via *software* por meio da *flag* ‘`_SERIAL_DEBUG`’ que recebe um valor booleano. Este recurso gera um saída que pode ser acompanhada via comando ‘`repl`’ do RShell. Outro recurso de *debug* implementado no código, é um *log* via arquivo que registra, dados de pacotes enviados e recebidos. Este recurso gera como saída um arquivo gravado na memória interna do LoPy4 caso não exista um cartão microSD conectado, com o comando ‘`cp`’ do RShell é possível resgatar este arquivo do dispositivo.

Um nano gateway UVLoRa é responsável por coletar os dados de radiação UV repassados pelos nós e enviá-los ao *IoTCloud*. Para fazer isso, recebe os dados de UV vindos dos nós via LoRa e por uma rede WiFi se conecta a internet e utiliza protocolo MQTT para se comunicar com o *IoTCloud* e envia-lo os dados recebidos. Diferentemente dos Nós, os Nano Gateways UVLoRa, não precisaram se preocupar com o consumo de energia, já que devem ficar em locais com disponibilidade da mesma e ainda redes WiFi próximas. Este código faz o uso de paralelismo para executar algumas funções, já que precisa interfacear a conexão do Nós com o servidor *IoTCloud*. A *thread* principal do programa fica responsável

por receber os pacotes LoRa, enquanto que os dados recebidos deste pacote, são repassados para o *IoTCloud* por outra *thread* utilizando MQTT, e o dispositivo ainda fica subscrito em tópico MQTT no mesmo servidor em uma terceira *thread*. Assim como no código dos nós, para facilitar o *debug* um *log* via conexão serial também foi implementado neste código, e da mesma forma o recurso pode ser desabilitado via *software* por meio da *flag* '`_SERIAL_DEBUG`' que recebe um valor booleano. Este recurso gera um saída que pode ser acompanhada via comando '`repl`' do RShell e cujo exemplo pode ser visto na figura 17. Também foi implementado o recurso de *log* via arquivo e da mesma forma gera como saída um arquivo gravado na memória interna do LoPy4 caso não exista um cartão microSD conectado, com o comando '`cp`' do RShell é possível resgatar este arquivo do dispositivo. Vale ressaltar que este código foi inspirado no código de Nano Gateway disponibilizado pela Pycom (PYCOM, 2019a).

3.4 Softwares

Para desenvolvimento deste projeto serão necessários alguns softwares de apoio, como editores de texto e de comunicação serial com as placas, para envio de arquivos de códigos, resgatar arquivos de log, etc. Além dos softwares de apoio, dois *script* serão produzido a fim de controlar nós e gateways UVLoRa.

Sobre o software editor de texto, ele pode ser qualquer um dentre os diversos disponíveis, cada um com suas particularidades, aqui utilizou-se Visual Studio Code¹² ferramenta muito versátil e de código aberto da gigante Microsoft. Para comunicação serial utilizou-se RShell¹³, uma ferramenta de shell simples que roda no host e usa o REPL¹⁴ do MicroPython para enviar trechos de código para o placa, a fim de obter informações do sistema de arquivos e copiar arquivos de e para o sistema de arquivos do MicroPython.

Quanto aos softwares de *IoTCloud* utilizou-se para testes o Ubidots que permite montar e iniciar rapidamente aplicações de IoT gratuitamente por 30 dias e depois deste prazo o Adafruit IO passou a ser utilizado, e de forma similar permite de forma simples montar aplicações e painéis de visualização de dados com diversos recursos interessantes.

Ubidots foi utilizado a fim de analisar uma plataforma com todos os seus recursos disponíveis, sua assinatura é um tanto quanto cara mas reflete a qualidade dos serviços

¹² Visual Studio Code (MICROSOFT, 2019) - <https://code.visualstudio.com/>

¹³ RShell (HYLANDS, 2019) - <https://github.com/dhylands/rshell>

¹⁴ Um loop de leitura-avaliação-impressão (REPL do inglês: *read–eval–print loop*), também denominado shell de nível superior ou de idioma interativo, é um ambiente de programação de computador simples e interativo que recebe entradas de usuário único (ou seja, expressões únicas), avalia (executa) e retorna o resultado para o usuário; um programa gravado em um ambiente REPL é executado por partes. O termo é geralmente usado para se referir a interfaces de programação semelhantes ao ambiente interativo da máquina Lisp clássica. Exemplos comuns incluem shells de linha de comando e ambientes semelhantes para linguagens de programação, e a técnica é muito característica do script (FREECODECAMP, 2019).

prestados, a plataforma oferece um teste grátis por 30 dias com todos os recursos disponíveis sem limite de uso. Apresenta interface muito simples e simpática, oferecendo muitos recursos, como a possibilidade de criação de gráficos variados, mapas, tabelas, botões, indicadores, etc. Os preços para assinatura da plataforma podem ser vistos na figura 8.

	IoT Entrepreneur	Professional	Industrial	Scale
	\$49 /month	\$199 /month	\$499 /month	\$1,799 /month
	REQUEST A FREE TRIAL	REQUEST A FREE TRIAL	REQUEST A FREE TRIAL	REQUEST A FREE TRIAL
Devices 1 device = 20 variables	25	200	1,000	4,000
Data Ingestion	2 Million dots per month	15 Million dots per month	50 Million dots per month	200 Million dots per month
Data Extraction	2 Million dots per month	15 Million dots per month	50 Million dots per month	200 Million dots per month
End Users	-	50	200	800
Organizations	-	Unlimited	Unlimited	Unlimited
Additional devices	N/A	\$50 per block of 50 devices (\$1 per device)	\$25 per block of 50 devices (\$0.50 per device)	\$20 per block of 50 devices (\$0.40 per device)
All plans include				
<ul style="list-style-type: none"> • 1,000 Ubifunctions executions per month • 1,000 events executions per month • 100 Email and telegram alerts per month • 10 SMS and 1 Voice call per month 				

Figura 8 – Preços para a assinatura da plataforma Ubidots.

Fonte: (UBIDOTS, 2019)

O Ubidots dedica um orçamento anual para manter e dar suporte a 60.000 contas gratuitas na plataforma Ubidots STEM, e também para patrocinar comunidades técnicas, aceleradores de IoT, eventos, encontros, escolas e universidades ao redor o mundo. Para isso emitem códigos de cupom a fim de espalhar a disponibilidade e acessibilidade da IoT, transformando idéias teóricas em aplicativos práticos de IoT. O autor deste projeto ganhou um destes cupons para realizar pesquisas do UVLoRa (SEPÚLVEDA, 2019), que agora conta com uma versão totalmente gratuita porém limitada da plataforma, todavia é muito útil para o futuro da realização desta pesquisa.

A Versão STEM do Ubidots, permite ao usuário principalmente, confira figura 9:

- Somente uso não comercial;
- 3 dispositivos com até 10 variáveis cada;
- Os dados ficam retidos por 1 mês;
- 10 SMS;

O Adafruit IO, outro servidor *IoTCloud* testado neste trabalho tem uma abordagem diferente para as contas gratuitas. Nele, todas as contas são inicialmente gratuitas, bastante

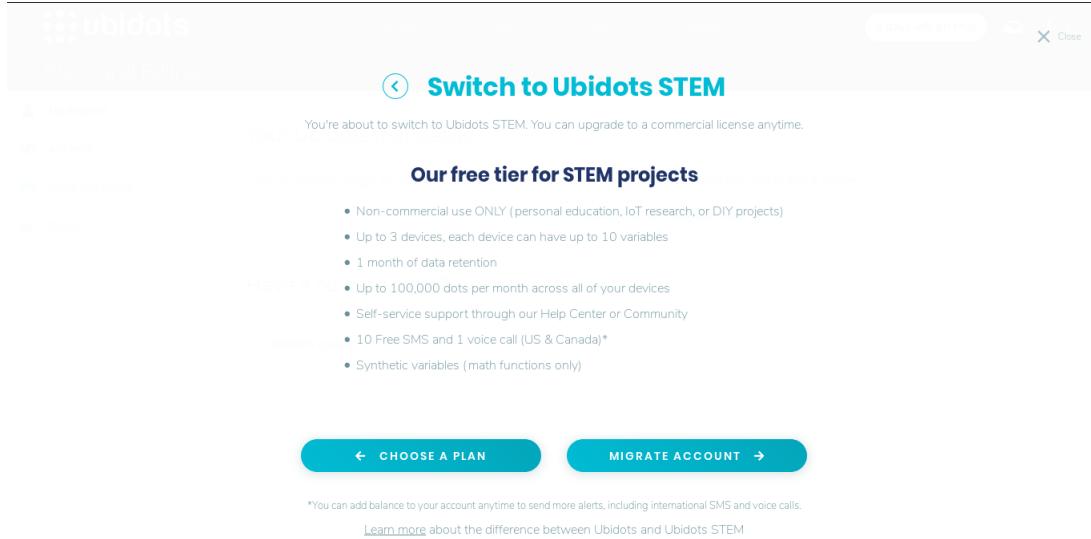


Figura 9 – Apresentação da Ubidots STEM, versão gratuita da plataforma para pesquisa e educação.

Fonte: (UBIDOTS, 2019)

limitada em questão de numero de variáveis, ou seja dados diferentes em cada dispositivo, mas suficiente para realização de testes e pesquisas. A plataforma da Adafruit possui um interface bastante simples de ser usada, porém não muito intuitiva e a disponibilidade de recursos é menor, todavia ainda sim supriu muito bem as necessidades do momento da UVLoRa. A Figura 10 mostra os custos de assinatura da plataforma.

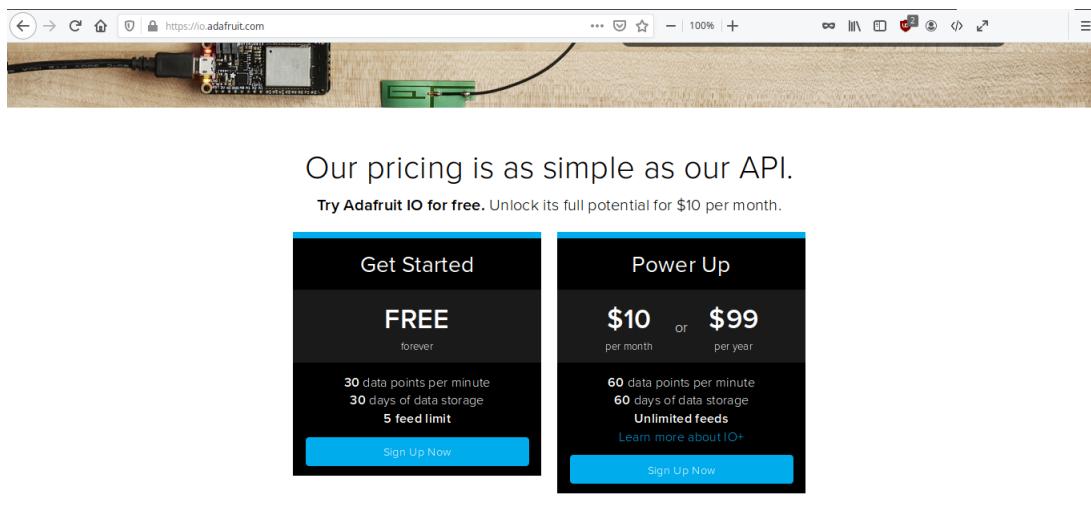


Figura 10 – Custos de assinatura do Adafruit IO.

Fonte: O Autor.

3.5 Comunicação

Para comunicação entre as várias partes que compõem o sistema UVLoRa, foram utilizadas varias tecnologias de comunicação entre enlaces de rede sem fio e protocolos de comunicação. Especificamente, entre os nós (UVLoRa Node) e gateway (UVLoRa Nano-GT) utiliza-se rede sem fio LoRa com protocolo próprio. Os UVLoRa Nano-GT e utilizam rede sem fio WiFi para se conectar ao ponto de acesso próximo e utiliza-se protocolo MQTT para troca de informações com o *IoTCloud*.

3.6 Consumo de Energia

Um desafio grande para a UVLoRa é manter grande a autonomia da bateria, para que seja necessário o mínimo de intervenção possível ao dispositivos, dessa forma, otimizações de hardware e de software foram feitas a fim de garantir o máximo alcance na comunicação via LoRa. Testes de consumo realizados com os dispositivos seguiram a seguinte formula:

$$A_{bat} = \frac{C_{bat}}{\left(\frac{i_a * t_a + i_d * (t_d - t_a)}{t_d} \right)}$$

,

Onde: (i_d) é a corrente modo *deepsleep*; (i_a) é a corrente no modo ativo; (t_d) é tempo em modo *deepsleep*; (t_a) é o tempo em modo ativo; (C_{bat}) é a capacidade da bateria utilizada; e (A_{bat}) é a autonomia da bateria em horas.

4 Resultados e Avaliação

4.1 Dispositivos UVLoRa

4.1.1 Nô

As Figuras 11, 12 e 13 mostram tal dispositivo.



Figura 11 – Protótipo de UVLoRa Nô

Fonte: O Autor.

4.1.2 Script do Nô

O *software* em questão tem um algoritmo similar ao que segue:

```

1  acorda
2  se prepara
3  le o sensor
4  cria pacote lora
5  envia o pacote
6  enquanto o downlink nao terminar:
7      se receber comando em downlink:
8          executa comando
9      se receber mensagem em downlink:
10         grava mensagem no arquivo
11      se nao:
12          continua
13  grava os dados de log no arquivo
14  dorme
15

```



Figura 12 – Visão interna da caixa de proteção do prototípico UVLoRa.

Fonte: O Autor.

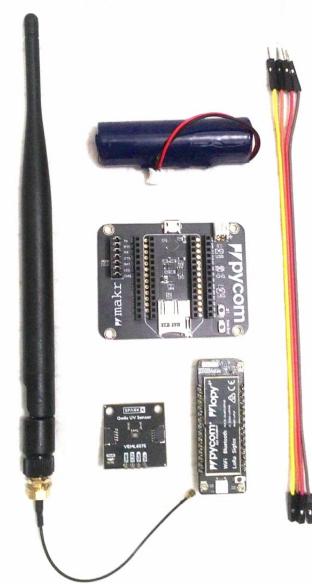


Figura 13 – Itens básicos que compõem um protótipo de nó

Fonte: O Autor.

Este algorítimo é executado pelo nó a cada intervalo de tempo pré definido, que durante os testes foi configurado como 1 minuto, mas em operação pode ser algo entre 10 e 15 minutos já que a UV não varia tão rapidamente, salvo em caso de nuvens passando.

4.1.3 Nano Gateway

As Figuras 14 e 15 mostram tal dispositivo e os itens que o compõem.



Figura 14 – Protótipo de nano gateway UVLoRa.

Fonte: O Autor.



Figura 15 – Itens básicos que compõem um protótipo de nano gateway UVLoRa.

Fonte: O Autor.

4.1.4 Script do Nano Gateway

O *script* em questão, que pode ser encontrado no Apêndice B possui um algoritmo similar ao que segue:

```

1      inicia
2      se prepara
3      enquanto verdade:
4          espera por pacotes lora
5          espera mensagem de servidor iot
6          se receber uma mensagem do servidor iot:
7              extrai dados da mensagem
8              guarda dados para envio posterior ao no correspondente
9          se nao:
10             continua
11         se receber um pacote lora:
12             extrai dados do pacote
13             verifica o id do no que enviou este pacote
14             se houverem mensagens para este no:
15                 envia a mensagem
16             se nao:
17                 continua
18             registra logs no arquivo
19         se nao:
20             continua
21

```

4.2 Driver VEML6075

Durante o desenvolvimento deste trabalho, sentiu-se a necessidade de testar outros sensores de UV diferentes do SI1145, um forte candidato era o VEML6075 devido as suas características e custo benefício, no entanto, não havia um *driver* na linguagem MicroPython para o sensor, e diante dessa situação optou-se por desenvolve-lo pra que tal sensor pudesse ser utilizado, tal *driver* apresenta funcionamento satisfatório, mas mais testes precisam ser feitos antes da disponibilização do *software* pra a comunidade MicroPython.

Tal código esta em disponível no Apêndice C deste documento e foi baseado no trabalho de Ladyada (2018)¹, desenvolvido para CircuitPython derivação da linguagem MicroPython mantida pela Adafruit Industries²(ADAFRUIT, 2019b).

4.3 Caso de Uso

Neste caso uso os *scripts* MicroPython que comandam os dispositivos e o driver do sensor VEML6075 que foi desenvolvido para uso na UVLoRa, foram instalados e testados nos dispositivos. Durante os testes, logs como o apresentado na Figura 16 foram gerados pelo Nô UVLoRa.

¹ Adafruit Driver CircuitPython para sensor VEML6075 (LADYADA, 2018) - https://github.com/adafruit/Adafruit_CircuitPython_VEML6075

² CircuitPython (ADAFRUIT, 2019b) - <https://circuitpython.org/>

```
[      ] UVLoRa Node Starting...
[ :D ] Hello! I'm node 2.
[ : ) ] SDCard mounted - 144.765GB free
[ :| ] Testing msgid file exist
[ :D ] File exists, opening...
[ :| ] Preparing to ship Lora package...
[ :| ] Prepared. sending...
[ :D ] Sended!
[ :| ] On downlink window. waiting 500 ms...
[ :| ] Updating message ID
[ :C ] Fineshed. Bye!
```

Figura 16 – Log do *script* via serial/USB.

Fonte: O Autor.

Na Tabela 2 pode-se conferir um exemplo do conteúdo do arquivo log gerado no caso.

Tabela 2 – Parcera dos dados do arquivo de log gerado pelo Nô UVLoRa

msg_id	wrc	bat	uvi	uva	uvb
200	0	241	0	0	0
201	3	254	0	0	0
202	3	243	0	0	0
203	3	246	0	0	0
204	3	254	0	0	0
...

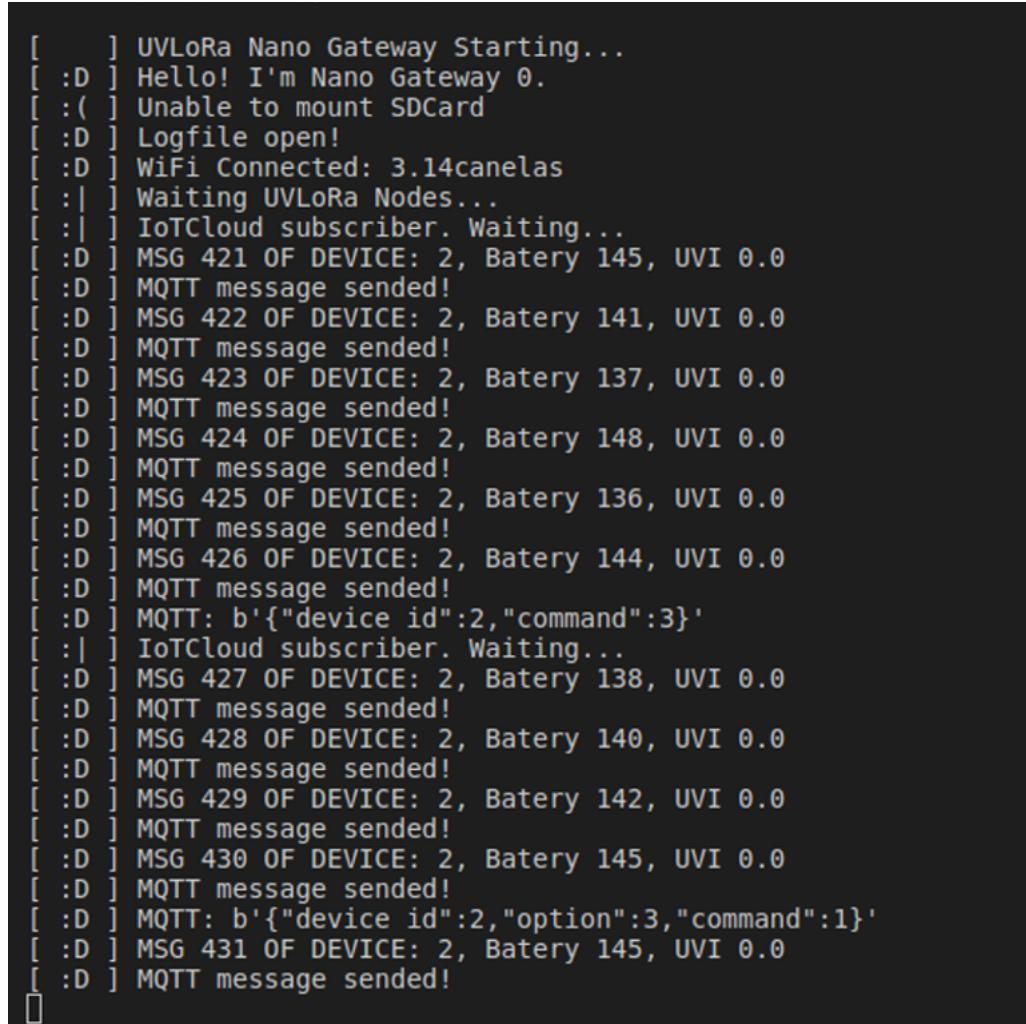
A Figura 17 mostra um exemplo do conteúdo do arquivo de log gerado pelo Nano Gateway UVLoRa no caso.

A seguir na Tabela 3, pode-se conferir um exemplo do conteúdo do arquivo de log gerado pelo código do protótipo do Nano Gateway UVLoRa

Tabela 3 – Parcera dos dados do arquivo de log gerado pelo Nano Gateway UVLoRa.

datatime	id	rssi	msg_id	uvi	uva	uvb	bat	wrc
(2019, 11, 17, 11, 29, 37, 424556)	2	-19	200	0	0	0	241	0
(2019, 11, 17, 11, 30, 12, 826523)	2	-19	201	0	0	0	254	3
(2019, 11, 17, 11, 30, 48, 238517)	2	-18	202	0	0	0	243	3
(2019, 11, 17, 11, 31, 23, 639520)	2	-19	203	0	0	0	246	3
...

Para a realização deste caso de uso foram utilizados os servidores *IoTCloud*, Ubidots e Adafruit IO, ambos de forma gratuita.



```
[     ] UVLoRa Nano Gateway Starting...
[ :D ] Hello! I'm Nano Gateway 0.
[ :( ] Unable to mount SDCard
[ :D ] Logfile open!
[ :D ] WiFi Connected: 3.14canelas
[ :| ] Waiting UVLoRa Nodes...
[ :| ] IoTCloud subscriber. Waiting...
[ :D ] MSG 421 OF DEVICE: 2, Batery 145, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 422 OF DEVICE: 2, Batery 141, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 423 OF DEVICE: 2, Batery 137, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 424 OF DEVICE: 2, Batery 148, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 425 OF DEVICE: 2, Batery 136, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 426 OF DEVICE: 2, Batery 144, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MQTT: b'{"device id":2,"command":3}'
[ :| ] IoTCloud subscriber. Waiting...
[ :D ] MSG 427 OF DEVICE: 2, Batery 138, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 428 OF DEVICE: 2, Batery 140, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 429 OF DEVICE: 2, Batery 142, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MSG 430 OF DEVICE: 2, Batery 145, UVI 0.0
[ :D ] MQTT message sended!
[ :D ] MQTT: b'{"device id":2,"option":3,"command":1}'
[ :D ] MSG 431 OF DEVICE: 2, Batery 145, UVI 0.0
[ :D ] MQTT message sended!
```

Figura 17 – Log do *script* via serial/USB.

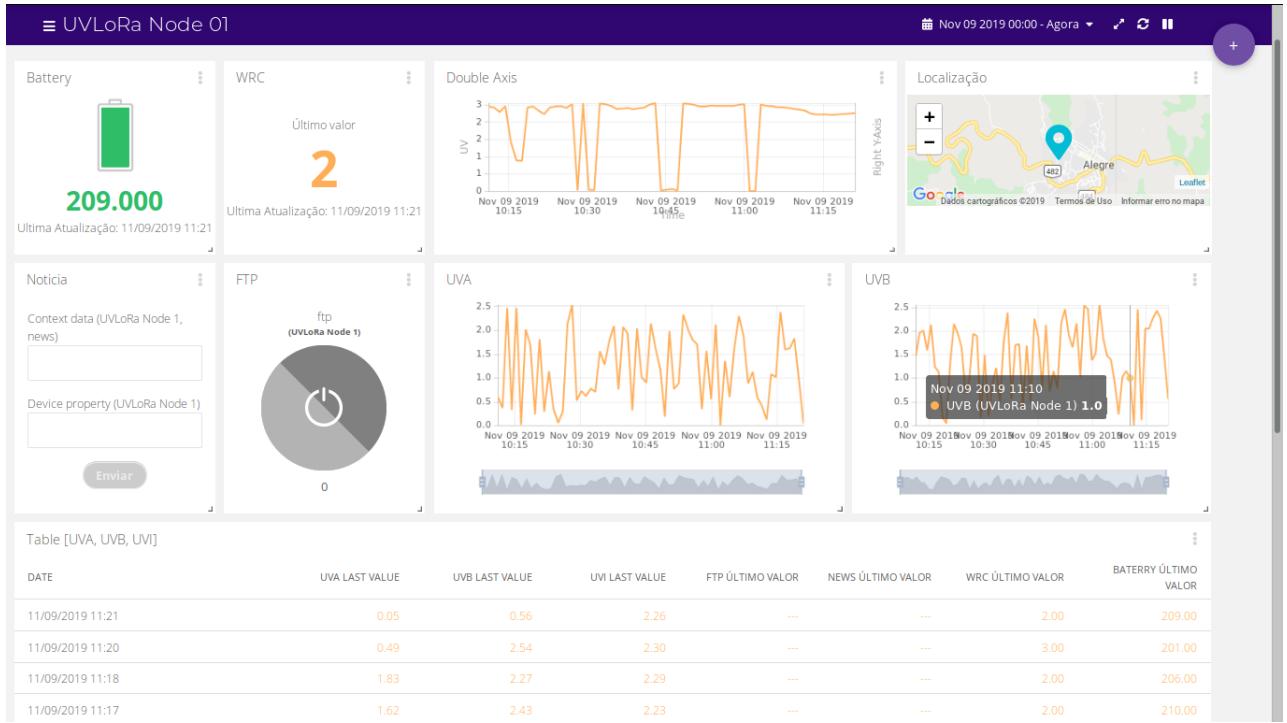
Fonte: O Autor.

A Figura 18 mostra parte dos recursos da plataforma utilizados no painel do UVLoRa, enquanto a Figura 19 mostra alertas enviados pela plataforma na ocorrência de um evento, neste caso, quando o nó não envia nada por um período de 10 minutos.

A figura 20 mostra o painel do UVLoRa criado com a plataforma Adafruit IO.

4.4 Testes UVLoRa

Testes realizados mostraram uma capacidade de alcance do LoRa superior a 4 Km com capacidade para ir ainda mais longe, confira figura 21. Ao todo neste teste a distância do Nô UVLoRa até cidade foi cerca de 8,27 Km, pois o concentrador UVLoRa ficou num ponto favorável onde ainda era possível se conectar a uma rede móvel 3G, e desse forma utilizou-se um celular para gerar uma rede WiFi onde o Nano Gateway UVLoRa se conectou.

Figura 18 – Dashboard UVLoRa no *IoTCloud* Ubidots.

Fonte: O Autor.

Neste teste, os dados de UV eram coletados pelo UVLoRa Nó no ponto 3 do mapa a cerca de 8,27 Km da cidade de Alegre e em seguida enviados ao UVLoRa Nano Gateway no ponto 2 do mapa a cerca de 4,25 Km de distância da cidade de Alegre, o UVLoRa Nano Gateway por sua vez enviava os dados ao servidor IoT Ubidots via uma conexão WiFi, criado por um celular que faz a ponte entre o WiFi gerado e a rede 3G, cuja antena mais próxima localizada no ponto 1, ja na cidade.

4.5 Consumo de energia e autonomia de bateria no Nó

Testes realizados com os dispositivos UVLoRa Nó mostram um consumo médio de 0,021 miliampere (i_d) em modo *deepsleep* na qual irá passar a maior parte do tempo e consumo de 50 miliampere ($\text{mA } i_a$) no modo ativo. Em cada ciclo, tais dispositivos ficam em média 10 minutos ($607,5 + \text{ou} - 7,5$ segundos, t_d) em modo *deepsleep*, e em média 2,5 segundos (t_a) modo ativo. De posse de tais dados além da capacidade da bateria que para os testes foi de 2600 miliampere-hora ($\text{mAh } C_{bat}$), é possível prever a autonomia da bateria A_{bat} em horas com a seguinte formula:

$$A_{bat} = \frac{C_{bat}}{\left(\frac{i_a * t_a + i_d * (t_d - t_a)}{t_d} \right)}$$

Substituindo os valores na equação pelos valores obtidos da UVLoRa temos:



Figura 19 – SMSs enviados pela plataforma Ubidots, recurso de alerta.

Fonte: (UBIDOTS, 2019)

$$A_{bat} = \frac{2600mAh}{\left(\frac{50mA*2,5s + 0,021mA*(600s - 2,5s)}{600s} \right)}$$

$$A_{bat} = \frac{2600mAh}{\left(\frac{50mA*2,5s + 0,021mA*597,5s}{600s} \right)}$$

$$A_{bat} = \frac{2600mAh}{\left(\frac{125mA + 12,55mA}{600s} \right)}$$

$$A_{bat} = \frac{2600mAh}{\left(\frac{137,55mA}{600s} \right)}$$



Figura 20 – Painel UVLoRa criado utilizando a plataforma Adafruit IO.

Fonte: O Autor.

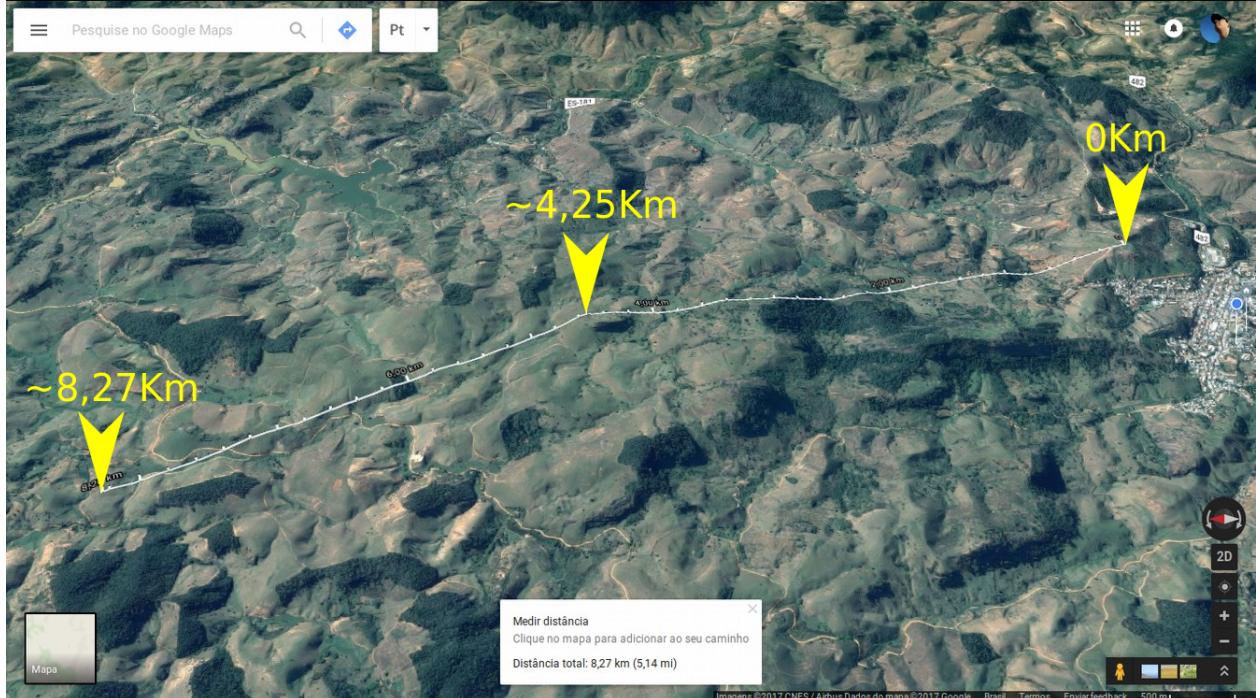


Figura 21 – Mapa com marcações do teste de alcance LoRa realizado.

Fonte: O Autor.

$$A_{bat} = \frac{2600mAh}{0,23mA}$$



Figura 22 – Visada do ponto 2 para o ponto 3

Fonte: O Autor.

$$A_{bat} = 11304,35h$$

Convertendo tal valor para dias temos: $11304,35/24 = 471,01$ dias, que por sua vez é aproximadamente 1 ano e 3 meses de autonomia com as condições atuais de configuração dos dispositivos UVLoRa Nô. Tal tempo pode ser elevado em 5 vezes utilizando-se baterias do tipo $LiSOCl_2$ que são consideravelmente maiores que as outras baterias primárias comercialmente disponíveis. A corrente de fuga dessas baterias é extremamente baixa, o que pode suportar longos períodos de armazenamento e atingir uma vida útil de 10 a 20 anos (POWERTECH, 2018) a depender da forma de utilização. No caso da UVLoRa, utilizando-se uma bateria de 19000mAh, a sua autonomia seria na casa dos 9 anos.

Dado que o ciclo de ativação do dispositivo em porcentagem é cerca de: $D = \frac{2.5}{2.5+600} \times 100\% \simeq 0.415\%$, podemos analisar no gráfico da Figura 24 que a autonomia da bateria, está em função do ciclo de ativação que é de 10 minutos.

Tendendo o valor em porcentagem do ciclo de ativação a zero, ou seja aumentando o tempo entre ativações, tal autônoma cresce consideravelmente confira Figura 25. Como os dados enviados pelos Nós UVLoRa permitem tal variação sem grandes consequências, devido a sua natureza, ciclo entre ativações pode ser ajustado a fim obter-se uma maior autonomia da bateria.



Figura 23 – Visada do ponto 2 para o ponto 1

Fonte: O Autor.

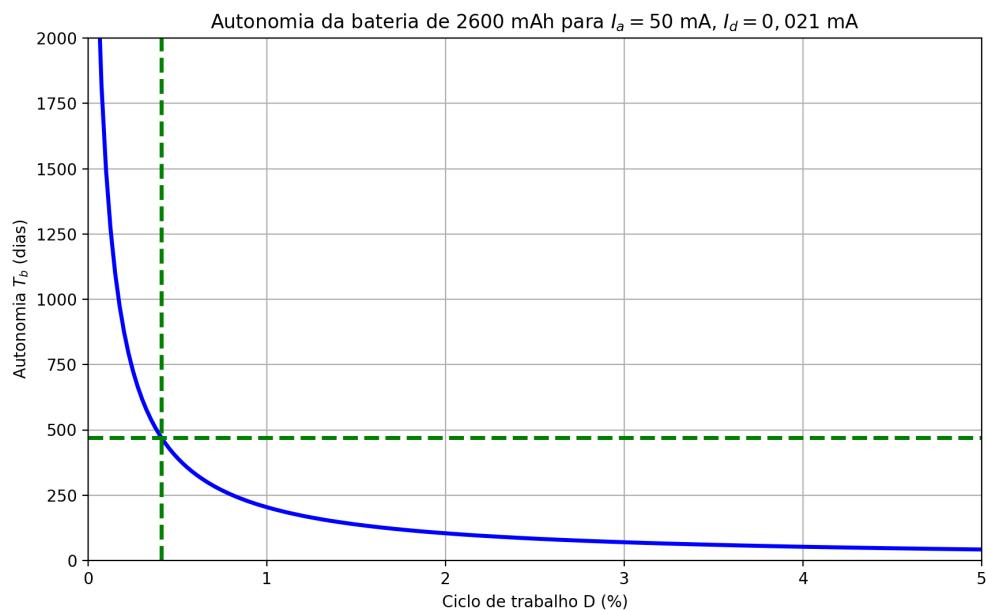


Figura 24 – Autonomia da bateria dos dispositivos UVLoRa Nô.

Fonte: O Autor.

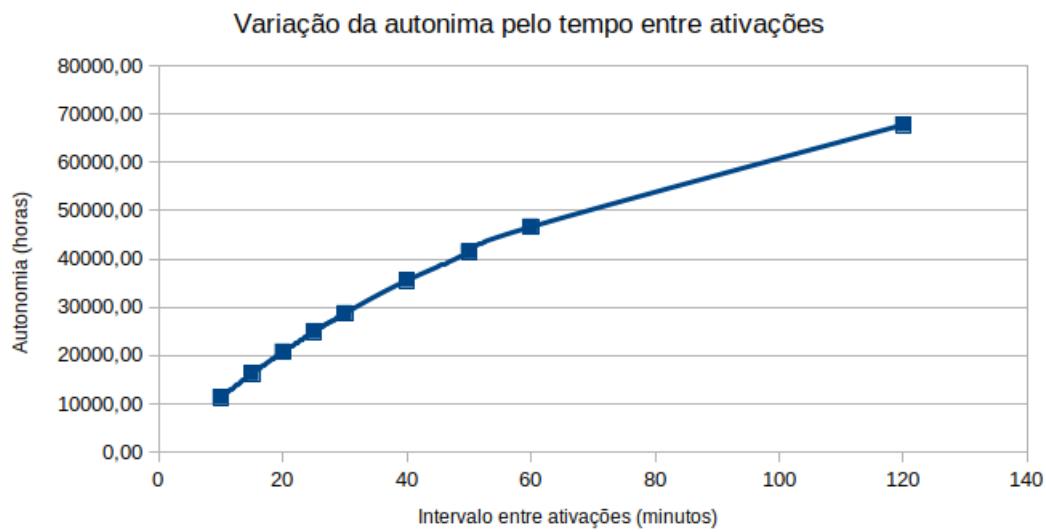


Figura 25 – Autonomia da bateria dos dispositivos UVLoRa Nô.

Fonte: O Autor.

5 Conclusão

A criação da internet revolucionou a comunicação, isso é um fato e atualmente estamos vivenciando sua primeira grande evolução internet com a criação dos dispositivos inteligentes que se comunicam através dela, que passa a ser chamada de Internet das Coisas e possibilita aplicações que anteriormente eram impensáveis.

Obter esses dispositivos inteligentes ficou muito simples e mesmo barato e como isso o numero de aplicações envolvendo Internet das Coisas cresce exponencialmente. Em meio a emergência do assunto está sendo desenvolvida a UVLoRa: uma estação de monitoramento de radiação ultravioleta com dados publicados na internet. UVLoRa é um exemplo de ciência cidadã a ser levada a população, sendo um instrumento de melhoraria da saúde pública já que pode gerar uma conscientização sobre os riscos do câncer de pele e danos aos olhos causados ou agravados pela radiação ultravioleta.

Foi escolhido LoRa como tecnologia de transmissão, onde um UVLoRa Nó consegue transmitir os dados coletados para um nano gateway localizado a alguns quilômetros de distância, conectado a internet. Alguns testes realizados mostram que a tecnologia LoRa atende perfeitamente os propósitos da UVLoRa, conseguindo transmitir dados para um gateway situado a mais de 4 km de distância. O *IoTCloud* possibilitou a criação de gráficos para visualização dos dados coletados, disponibilizando tanto via web quanto mobile, além de permitir a criação de alertas para algumas situações e enviar comandos aos dispositivos.

UVLoRa possui itens diversos utilizados em seu desenvolvimento com custo relativamente baixo e um sistema de comunicação próprio que combina tecnologias como LoRa, WiFi, MQTT e outras. A comunicação via LoRa é feita com protocolo próprio e já permite o envio de mensagens do *IoTCloud* para um nó qualquer. Estas mensagens podem ser comandos para o nó executar ou podem ter conteúdos diversos como notícias por exemplo.

Em caso de desconexão com a internet, os dados não são perdidos, pois podem ser salvos tanto no gateway quanto no nó em um cartão micro SD instalado ou mesmo em sua memória interna. A UVLoRa ainda conta com terceira redundância de dados, pois mesmo se algum dos dispositivos falhar, ou por qualquer motivo o funcionamento do sistema for interrompido, os dados coletados estarão armazenados também na nuvem.

Satisfatoriamente os testes comprovam o tão esperado sucesso na comunicação a grandes distâncias usando LoRa, assim como era esperado para as aferições do UVI e troca de dados. Dessa forma o UVLoRa permite que o UVI seja aferido em localidades remotas e sem o acesso a redes sem fio. Os dados coletados e disponibilizados publicamente pode-se conscientizar a população sobre os perigos da radiação UV, permitindo-as planejar o seu dia, afim ter uma vida mais saudável. Esses resultados sugerem ainda que a utilização de

microcontroladores e sensores de baixo custo e transmissão dos dados via uso de LoRa, é uma solução viável e eficaz.

5.1 Trabalhos Futuros

5.1.1 Testes de acurácia e precisão de sensores UV

A medição de grandeza física é extremamente difícil, se não impossível, de ser realizada, ou seja, de obter-se o "valor verdadeiro" do mensurando.

5.1.2 Utilização de domo de luz para proteção do sensor

Os circuitos eletrônicos são elementos essenciais para funcionamento da UVLoRa necessitando cuidados especiais, principalmente quando instaladas em ambientes agressivos sujeitos a vibração, umidade, temperaturas críticas ou partículas em suspensão no ar.

5.1.3 Otimização do sistema para economia de energia

Um desafio grande para a UVLoRa é manter grande a autonomia da bateria, para seja necessário o mínimo de intervenção possível, para isso uma placa de circuitos impressos esta sendo desenvolvida, a fim de garantir a autônima esperada para os nós. Da mesma forma que o hardware, os softwares também devem sofrer ajustes para alcançar tal objetivo. A figura 26 mostra o protótipo em questão.

5.1.4 Mais testes de alcance LoRa

Seria desejável explorar o alcance da LoRa em ambientes mais extremos, como em florestas e montanhas, ou qualquer coisa que possa ocasionar um nível de interferência maior.

5.1.5 Utilização de PyTrack em dispositivos UVLoRa

PyTrack trata-se de uma placa de expansão pra LoPy4, veja figura 27. As principais características da Pytrack são: GPS super preciso, acelerômetro de 3 eixos e 12 bits, porta USB com acesso serial, carregador de bateria LiPo, compatibilidade com cartão MicroSD, operação com energia ultra baixa (1uA em deepsleep). E ainda suporta vários serviços de localização como GPS, GLONASS, Galileu e QZSS (PYCOM, 2019g).

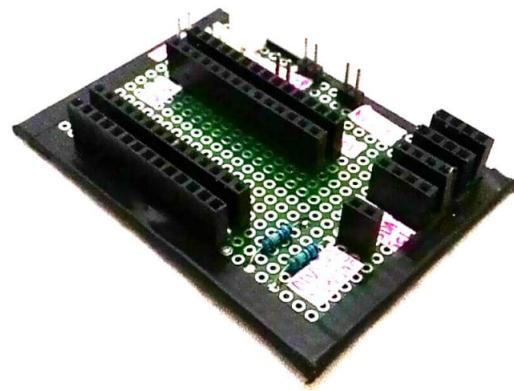


Figura 26 – Protótipo de placa desenvolvida para o UVLoRa por Eduardo Amorim Benincá, formado em Física na UFES-Alegre e bolsista de Desenvolvimento Tecnológico Industrial (DTI-B) da FAPES, atuando na UFES-Vitória em projetos de IoT e sensores do CPID (Centro de Pesquisa, Inovação e Desenvolvimento) do ES.

Fonte: O Autor.

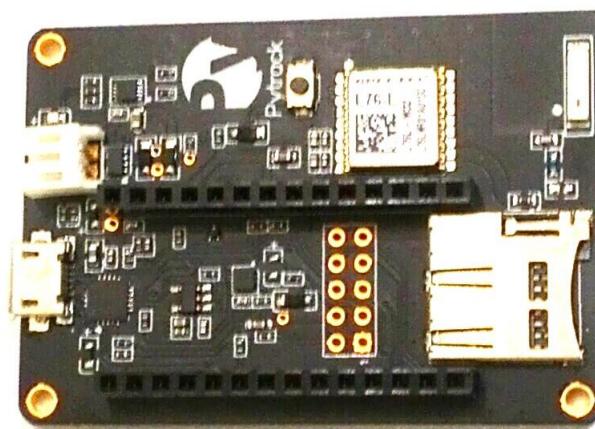


Figura 27 – Pytrack.

Fonte: O Autor.

Referências

- ACLIMA. *Aclima and Google Scale Air Quality Mapping to More Places Around the World*. 2018. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://aclima.io/press/aclima-google-scale-air-quality-mapping-to-more-places-around-the-world/>>. Citado na página 28.
- ADAFRUIT. *Adafruit VEML6075*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://pycom.io/product/lopy4/>>. Citado 2 vezes nas páginas 31 e 32.
- ADAFRUIT, I. *CircuitPython*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://circuitpython.org/>>. Citado na página 42.
- ADLER, H. K. S. by L. *How Smart City Barcelona Brought the Internet of Things to Life*. 2016. [Online; Acesso em: 29 de Outubro de 2019]. Disponível em: <<https://datasmart.ash.harvard.edu/news/article/how-smart-city-barcelona-brought-the-internet-of-things-to-life-789>>. Citado na página 28.
- AGROSMART. *Agrosmart*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://agrosmart.com.br/>>. Citado na página 28.
- ALIEXPRESS. *Busca no Aliexpress*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <https://pt.aliexpress.com/wholesale?catId=0&initiative_id=SB_20191117142534&SearchText=Sensorer+UV>. Citado na página 31.
- ALIEXPRESS. *Fused Silica Window Quartz Glass*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://pt.aliexpress.com/item/32919984062.html?spm=2114.12010612.8148356.10.4a574834IR8HIV>>. Citado na página 33.
- ARDUINO, S. A. Arduino. *Arduino LLC*, 2015. Citado na página 21.
- AUGUSTIN, A. et al. A study of lora: Long range & low power networks for the internet of things. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 16, n. 9, p. 1466, 2016. Citado 2 vezes nas páginas 23 e 25.
- BALOGH, T. S. et al. Proteção à radiação ultravioleta: recursos disponíveis na atualidade em fotoproteção. *Anais Brasileiros de Dermatologia*, Sociedade Brasileira de Dermatologia, v. 86, n. 4, p. 732–742, 2011. Citado 3 vezes nas páginas 11, 15 e 18.
- BRASILESCOLA. *O que é efeito Doppler?* 2019. [Online; Acesso em: 30 de Outubro de 2019]. Disponível em: <<https://brasilescola.uol.com.br/o-que-e/fisica/o-que-e-efeito-doppler.htm>>. Citado na página 24.
- CISCO, I. B. S. G. *Beyond the New Normal How Retailers Can Thrive in a Rapidly Changing World*. 2011. [Online; Acesso em: 01 de Outubro de 2018]. Disponível em: <http://www.cisco.com/c/dam/en_us/about/ac79/docs/retail/Beyond-the-New-Normal_IBSG_051211-FINAL.pdf>. Citado na página 20.
- CYRCADIA. *The iTbra*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://cycadia.asia/#section-itbra>>. Citado na página 29.

- ESPRESSIF. *ESP32 Overview*. 2019. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <<https://www.espressif.com/en/products/hardware/esp32>>. Citado na página 21.
- EVANS, D. The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, v. 1, n. 2011, p. 1–11, 2011. Citado na página 20.
- EXAME, F. C. *Uma pessoa morre a cada 3 horas por câncer de pele*. 2016. [Online; Acesso em: 30 de Setembro de 2018]. Disponível em: <<https://exame.abril.com.br/brasil/morte-por-cancer-de-pele-cresce-55-em-10-anos/>>. Citado na página 16.
- FILIPEFLOP. *Suporte Pan/Tilt para Câmera Raspberry Pi Sparkfun*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://www.filipeflop.com/produto/suporte-pan-tilt-para-camera-raspberry-pi-sparkfun/>>. Citado na página 33.
- FREECODECAMP. *This is why your read-eval-print-loop is so amazing*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://www.freecodecamp.org/news>this-is-why-your-read-eval-print-loop-is-so-amazing-cf0362003983/>>. Citado na página 35.
- GEORGES, D. Micropython. 2015. Citado na página 22.
- GONZÁLEZ, S.; FERNÁNDEZ-LORENTE, M.; GILABERTE-CALZADA, Y. The latest on skin photoprotection. *Clinics in dermatology*, Elsevier, v. 26, n. 6, p. 614–626, 2008. Citado na página 18.
- HUNKELER, U.; TRUONG, H. L.; STANFORD-CLARK, A. Mqtt-s—a publish/subscribe protocol for wireless sensor networks. In: IEEE. *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*. [S.I.], 2008. p. 791–798. Citado na página 23.
- HYLANDS, D. *RShell*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://github.com/dhylands/rshell>>. Citado na página 35.
- IMAGINAIR. *Imaginair*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<http://www.imaginair.com.br>>. Citado na página 29.
- INCA, I. N. de Câncer José Alencar Gomes da S. Estimativa 2018: incidência de câncer no brasil. 2018. Citado na página 16.
- INCA, I. N. de Câncer José Alencar Gomes da S. *Pele Melanoma*. 2018. [Online; Acesso em: 11 de Junho de 2019]. Disponível em: <<https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-melanoma>>. Citado na página 15.
- INCA, I. N. de Câncer José Alencar Gomes da S. *Pele Melanoma*. 2018. [Online; Acesso em: 11 de Junho de 2019]. Disponível em: <<https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-nao-melanoma>>. Citado na página 16.
- INCA, I. N. do Câncer José Alencar Gomes da S. *Câncer de mama*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama>>. Citado na página 29.
- INTERTECHNOLOGY, V. *About Vishay Intertechnology*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<http://www.vishay.com/company/about/>>. Citado na página 32.

- JUCHEM, P. et al. Riscos à saúde da radiação ultravioleta. *Revista Brasileira de Cirurgia Plástica*, Revista Brasileira de Cirurgia Plástica, v. 13, n. 2, p. 31–60, 2001. Citado na página 15.
- JÚNIOR, R. C. *LoRa AirTime Calculator Software*. 2019. [Online; Acesso em: 04 de Dezembro de 2019]. Disponível em: <<https://gitlab.com/rcolistete/lora-lorawan/tree/master/LoRaAirTimeCalculator>>. Citado na página 27.
- KAMALANATHAN, N. A. et al. Improving the patient discharge planning process through knowledge management by using the internet of things. *Advances in Internet of Things*, Scientific Research Publishing, v. 3, n. 02, p. 16–26, 2013. Citado na página 12.
- KUROSE, J. F.; ROSS, K. W.; ZUCCHI, W. L. *Redes de Computadores e a Internet: Uma abordagem top-down*. [S.l.]: Pearson, 2010. Citado 2 vezes nas páginas 12 e 20.
- LABS, S. *Datasheet SI1145, sensor de óptico de Luz Visível, Infravermelha e UV*. 2019. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <<https://www.silabs.com/documents/public/data-sheets/Si1145-46-47.pdf>>. Citado na página 21.
- LADYADA, A. I. by. *Adafruit CircuitPython VEML6075*. 2018. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <https://github.com/adafruit/Adafruit_CircuitPython_VEML6075>. Citado na página 42.
- MICROBERTS, M. *Arduino Básico-2ª edição: Tudo sobre o popular microcontrolador Arduino*. [S.l.]: Novatec Editora, 2015. Citado na página 21.
- MENDONÇA, G. A. et al. Risco crescente de melanoma de pele no brasil. *Revista de saúde pública*, SciELO Public Health, v. 26, p. 290–294, 1992. Citado na página 18.
- MERCADOLIVRE. *Baterias*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<http://abre.ai/atXU>>. Citado na página 32.
- MICROPYTHON. *MicroPython*. 2018. [Online; Acesso em: 23 de Novembro de 2018]. Disponível em: <<https://micropython.org/>>. Citado na página 22.
- MICROSOFT. *Visual Studio Code*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://code.visualstudio.com/>>. Citado na página 35.
- NORVAL, M.; HALLIDAY, G. M. The consequences of uv-induced immunosuppression for human health. *Photochemistry and Photobiology*, Wiley Online Library, v. 87, n. 5, p. 965–977, 2011. Citado na página 15.
- ONU, O. das N. U. *World's population increasingly urban with more than half living in urban areas*. 2014. [Online; Acesso em: 29 de Outubro de 2019]. Disponível em: <<https://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html>>. Citado na página 28.
- OSI, O. S. I. *Open Source*. 2019. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <<https://opensource.org/>>. Citado na página 20.
- PHOTONICS. *BK7 & Quartz Optical Domes*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <https://www.photonics.com/Products/BK7_Quartz_Optical_Domes/pr62965>. Citado 2 vezes nas páginas 33 e 34.

- POWERTECH. *Primary cells Lithium Thionyl Chloride (Li-SOCl₂)*. 2018. [Online; Acesso em: 30 de Outubro de 2019]. Disponível em: <<https://www.powertechsystems.eu/home/products/primary-cells-lithium-thionyl-chloride-li-socl2/>>. Citado na página 48.
- PURI, D. *IoT devices communicate by Chirping*. 2019. [Online; Acesso em: 30 de Outubro de 2019]. Disponível em: <<https://www.networkworld.com/article/3241725/iot-devices-communicate-by-chirping.html>>. Citado na página 24.
- PYCOM. *Datasheet LoPy4*. 2019. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <https://docs.pycom.io/.gitbook/assets/specsheets/Pycom_002_Specsheets_LoPy_v2.pdf>. Citado 2 vezes nas páginas 31 e 35.
- PYCOM. *Deep Sleep API*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://docs.pycom.io/datasheets/boards/deepsleep/api/>>. Citado na página 34.
- PYCOM. *Expansion Board v3.0*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://pycom.io/product/expansion-board-3-0/>>. Citado na página 32.
- PYCOM. *Expansion Board v3.0*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://pycom.io/product/expansion-board-3-0/>>. Citado na página 32.
- PYCOM. *PyCase*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://pycom.io/product/pycase-clear/>>. Citado na página 32.
- PYCOM. *Pymesh updates*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://forum.pycom.io/topic/4449/pymesh-updates/14>>. Citado na página 26.
- PYCOM. *Pytrack*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://pycom.io/product/pytrack/>>. Citado na página 52.
- PYCOM. *RTC*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://docs.pycom.io/firmwareapi/pycom/machine/rtc/>>. Citado na página 34.
- RANA, D. by D. *Top 11 Cloud Platforms for Internet of Things (IoT)*. 2018. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <<https://dzone.com/articles/10-cloud-platforms-for-internet-of-things-iot>>. Citado 2 vezes nas páginas 22 e 23.
- RAY, P. P. A survey of iot cloud platforms. *Future Computing and Informatics Journal*, Elsevier, v. 1, n. 1-2, p. 35–46, 2016. Citado na página 23.
- RENNNA, R. B. D. et al. Introdução ao kit de desenvolvimento arduino. *UFF*, 2013. Citado na página 21.
- REZENDE, M. L.; CORRÊA, M. de P.; DUARTE, E. M. Estimativa do custo do tratamento de câncer de pele tipo melanoma no estado de são paulo-brasil. *An Bras Dermatol*, v. 84, n. 3, p. 237–43, 2009. Citado na página 16.
- RHODES, A. R. et al. Risk factors for cutaneous melanoma: a practical method of recognizing predisposed individuals. *Jama*, American Medical Association, v. 258, n. 21, p. 3146–3154, 1987. Citado na página 17.

- RICHARDSON, L.; RUBY, S. *RESTful web services*. [S.l.]: "O'Reilly Media, Inc.", 2008. Citado na página 23.
- SEBESTA, R. W. *Conceitos de Linguagens de Programação-11*. [S.l.]: Bookman Editora, 2018. Citado na página 22.
- SEPÚLVEDA, U. by D. *How to redeem a coupon code*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://help.ubidots.com/en/articles/1571154-how-to-redeem-a-coupon-code>>. Citado na página 36.
- SIGFOX. *SigFox*. 2019. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <<https://www.sigfox.com/en>>. Citado na página 21.
- SILVA, R. d. O. A. et al. Quem se ama, protege sua pele: orientações farmacêuticas na prevenção contra o câncer de pele. *Revista Conexão UEPG*, Pró-Reitoria de Extensão e Assuntos Culturais. Universidade Estadual de ..., v. 13, n. 2, p. 306–315, 2017. Citado na página 19.
- SINHA, R. S.; WEI, Y.; HWANG, S.-H. A survey on lpwa technology: Lora and nb-iot. *Ict Express*, Elsevier, v. 3, n. 1, p. 14–21, 2017. Citado na página 24.
- TANNENBAUM, A. S. *Redes de computadoras*. 3. [S.l.]: Pearson Educación, 1997. Citado na página 26.
- TAYLOR, H. R. Epidemiology of age-related cataract. *Eye*, Nature Publishing Group, v. 13, n. 3b, p. 445, 1999. Citado na página 16.
- TECH, S. C. *O que é API?* 2019. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <<https://canaltech.com.br/software/o-que-e-api/>>. Citado na página 23.
- THINGSPEAK. *IoT Analytics*. 2019. [Online; Acesso em: 15 de Junho de 2019]. Disponível em: <<https://thingspeak.com/>>. Citado na página 23.
- TRIMBLE. *Trimble*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://www.trimble.com.br/intl/br/encontre-um-revendor.aspx>>. Citado na página 29.
- UBIDOTS. *Ubidots*. 2019. [Online; Acesso em: 28 de Outubro de 2019]. Disponível em: <<https://ubidots.com/>>. Citado 4 vezes nas páginas 23, 36, 37 e 46.
- WATSON, C. E. I. *O uso da computação cognitiva pelos profissionais de Marketing*. 2017. [Online; Acesso em: 15 de Novembro de 2018]. Disponível em: <<https://www.ibm.com/blogs/watson-customer-engagement/br-pt/2017/05/04/o-uso-da-computacao-cognitiva-pelos-profissionais-de-marketing/>>. Citado na página 20.
- WEF, W. E. F. *Ultraviolet radiation and health*. 2018. [Online; Acesso em: 29 de Outubro de 2019]. Disponível em: <<https://www.weforum.org/agenda/2018/01/6-ways-the-internet-of-things-is-improving-our-lives/>>. Citado na página 27.
- WHO, W. H. O. *Ultraviolet radiation and health*. 2018. [Online; Acesso em: 29 de Setembro de 2018]. Disponível em: <http://www.who.int/uv/sun_protection/en/>. Citado 2 vezes nas páginas 14 e 19.

WHO, W. H. O.; ICNIRP, I. C. on N.-I. R. P. et al. *Global solar UV index: a practical guide.* [S.l.]: WHO, 1995. Citado na página 18.

WHO, W. H. O. et al. *Environmental Health Criteria 14. Ultraviolet radiation.* [S.l.]: United Nations Environment Programme/WHO, Geneva, Switzerland., 1979. Citado 2 vezes nas páginas 11 e 15.

APÊNDICE A – *Script* UVLoRa Nô

```

1 """
2     Version: 0.5
3         Author: Nelio Goncalves Godoi
4             E-mail: neliogodoi@yahoo.com.br
5 Last Update: 04/12/2019
6         Title: UVLoRa Node
7 """
8
9 _SERIAL_DEBUG      = False
10
11 if(_SERIAL_DEBUG):
12     print("\n[     ] UVLoRa Node Starting...")
13 from uos import urandom, mount
14 from network import LoRa, WLAN
15 from machine import I2C, deepsleep, ADC, wake_reason, reset_cause, SD, RTC
16 import socket
17 import time
18 import struct
19 import veml6075
20 import pycom
21
22 _DEVICE_ID        = 2
23
24 if(_SERIAL_DEBUG):
25     print("[ :D ] Hello! I'm Node {}".format(_DEVICE_ID))
26
27 # Usando a limitaao [ SF=10, BW=125kHz, CR=4/8, que e a configuracao LoRa-MAC uplink
28 # com maior alcance possivel seguindo a normas da ANATEL (< 400 ms de airtime),
29 # o pacote LoRa maximo e 14 bytes
30
31
32 _LORA_PKG          = "!BBHBBBBB"           # device_id, pkg_len, msg_id, wrc, batt, uvi, uva
33     , uvb
34 _LORA_PKG_MSG       = "!BBBBBBBBBBBB"        # device_id, pkg_len, msg[12] # == MAX PKG SIZE
35     ==
36 _PKG_SIZE           = struct.calcsize(_LORA_PKG)
37 _DOWNLINK_WINDOW    = 750                  # ms, 297 ms for uplink + 400 ms for downlink + 53 ms
38     gateway processing margin
39 _SDCARD_EXIST       = False
40 _SAVE_ON_FLASH      = True
41 _SDCARD_DEBUG        = False
42 _DPSLP_TIMEOUT      = (1000*30)      # 600 s = 10 minutos
43 _FTP_STATUS          = 0                   # 0 - Desativado | 1 - Ativado
44 _FTP_TIMEOUT         = (1000*60)*10 # 10 min
45 _ADC                 = ADC()
46 _I2C                 = I2C(0, I2C.MASTER)
47
48 lora = LoRa(mode      = LoRa.LORA,
49             frequency = 925500000,
50             tx_power   = 20,
51             sf         = 10,

```

```

52         bandwidth    = LoRa.BW_125KHZ,
53         coding_rate = LoRa.CODING_4_8,
54         tx_iq        = True)
55 lora_sock = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
56 lora_sock.setblocking(False)
57
58 # Variaveis extras
59 sensor = veml6075.VEML6075(i2c = _I2C)
60 opt = -1
61 ap = None
62 location = '/flash'
63
64 # Monstagem do Cartao SD se houver
65 try:
66     _SD = SD()
67     location = '/sd'
68     mount(_SD, location)
69     _SDCARD_EXIST = True
70     if(_SERIAL_DEBUG):
71         print("[ : ) ] SDCard - {:.3f}GB Free".format(os.getfree(location)/104857))
72 except:
73     if(_SERIAL_DEBUG):
74         print('[ :( ] Unable to mount SDCard')
75
76 # Abertura do arquivo msgid.conf
77 try:
78     if(_SERIAL_DEBUG):
79         print("[ :| ] Testing msgid file exist")
80     msg_id_file = open("/flash/msgid.conf", "x")
81     if(_SERIAL_DEBUG):
82         print("[ :/ ] The file does not exist, but I will create it now!")
83     msg_id_file.close()
84     msg_id_file = open("/flash/msgid.conf", "w")
85     msg_id_file.write("0")
86     if(_SERIAL_DEBUG):
87         print("[ :D ] Msgid file created sucessfull, opening...")
88     msg_id_file.close()
89 except:
90     if(_SERIAL_DEBUG):
91         print("[ :D ] File exists, opening...")
92 finally:
93     msg_id_file = open("/flash/msgid.conf", "r")
94     msg_id = int(msg_id_file.read())
95     msg_id_file.close()
96
97 @property
98 def ftp_on():
99     global ap
100    ssid = "UVLoRa-Node-{}".format(_DEVICE_ID)
101    auth = (WLAN.WPA2, 'uvlorा')
102    ap   = WLAN.init(mode= WLAN.AP, ssid= ssid, auth= auth, channel=7, antenna=None)
103    if(_SERIAL_DEBUG):
104        print("[ : ) ] FTP Mode ON!\n\nWifi Credentials:\nSSID: {}\nPASS: {}".format(ssid, auth[1]))
105        print("[ :| ] Use this credentials:\n url: ftp://192.168.4.1\nuser: micro\npass: python")
106        print("[ :| ] This section ends in 10 minutes!")
107    initial_time = time.ticks_ms()
108    while _FTP_STATUS and ((time.ticks_ms() - initial_time) < _FTP_TIMEOUT):
109        pass

```

```

110     if(_SERIAL_DEBUG):
111         print("[ : ) ] FTP Mode OFF by TimeOut")
112
113 @property
114 def ftp_off():
115     global ap
116     ap.deinit()
117     if(_SERIAL_DEBUG):
118         print("[ : ) ] FTP Mode OFF by command of IoTCloud")
119
120 def check_timeout(initial_time):
121     return ((time.ticks_ms() - initial_time) < _DOWNLINK_WINDOW )
122
123 datalogfilename = '{}/datalog.txt'.format(location)
124 try:
125     logfile = open(datalogfilename, 'x')
126     logfile.close()
127     logfile = open(datalogfilename, 'a')
128     logfile.write('msg_id, wrc, batt, uvi, uva, uvb')
129     logfile.flush()
130 except:
131     logfile = open(datalogfilename, 'a')
132
133 # RTC reading
134 rtc = RTC()    # internal RTC of ESP32
135 if rtc.now()[0] < 2000:
136     rtc.init((2019, 12, 5, 0, 00, 0, 0))
137 print("[ : ) ] RTC date hour tuple : {}".format(rtc.now()))
138
139 if(_SERIAL_DEBUG):
140     print("[ :| ] Preparing to send Lora package...")
141 pycom.rgbled(0x7f7f00)
142
143 # Read VEML6075
144 uvi = int(sensor.uv_index * 100)
145 uva = int(sensor.uva * 100)
146 uvb = int(sensor.uvb * 100)
147
148 # Read the battery voltage usind the ADC on pin P16 with voltage divider
149 _ADC.vref(1126)  # put here ESP32 VRef measured in mV with a voltmeter
150 # Expansion Board v3.x (or equivalent) has voltage divider (1M + 1M) converting to
151     # integer voltage, [0, 6600] mV
152 batt_mV = 2*(_ADC.channel(pin='P16', attn=ADC.ATTN_11DB).voltage())//1
153 # batt_mV_byte in 1 byte, [0-255], in scale (2.60-5.15)V by using (Vbat - 2.6V)*100.
154 batt_mV_byte = min(max(0, (batt_mV - 2600)//10), 255)
155
156 # Wake and Reset Reason
157 wake_code = wake_reason()[0]    # machine.PWRON_WAKE/PIN_WAKE/RTC_WAKE/ULP_WAKE
158 reset_code= reset_cause()      # machine.PWRON_RESET/HARD_RESET/WDT_RESET/DEEPSLEEP_RESET
159     /SOFT_RESET/BROWN_OUT_RESET
160
161 if(_SERIAL_DEBUG):
162     print("[ :D ] Message #{},".format(msg_id))
163     print("[ :D ] Wake and Reset code : 0x{:02x}.".format(wrc))
164     print("[ :D ] Battery : {} mV, byte format : {}.".format(batt_mV, batt_mV_byte))
165     print("[ :D ] VEML6075 UVI : {}, UVA : {}, UVB : {}.".format(uvi, uva, uvb))
166
167 pkg = struct.pack(_LORA_PKG, _DEVICE_ID, _PKG_SIZE, msg_id, wrc, batt_mV_byte, uvi, uva,
168 uvb)

```

```

167
168 if(_SERIAL_DEBUG):
169     print("[ :I ] Prepared. sending...")
170 lora_sock.send(pkg)
171 if(_SERIAL_DEBUG):
172     print("[ :D ] Sent!")
173
174 # downlink
175 if(_SERIAL_DEBUG):
176     print("[ :I ] On uplink & downlink window. waiting {} ms...".format(_DLINK_WINDOW))
177
178 initial_time = time.ticks_ms()
179 while (check_timeout(initial_time)):
180     recv_pkg = lora_sock.recv(512)
181     time.sleep(0.2)
182     if (len(recv_pkg) > 2):
183         len_recv_pkg = recv_pkg[1]
184         if(_SERIAL_DEBUG):
185             print("[ : ) ] Received package! ")
186             # print(recv_pkg)
187             if( len_recv_pkg == len(recv_pkg) ):
188                 device_id, pkg_len, msg = struct.unpack(_LORA_PKG_MSG, recv_pkg)
189                 if( device_id == _DEVICE_ID ):
190                     if( opt == -1): #
191                         opt = msg
192                         if(_SERIAL_DEBUG):
193                             print("[ :D ] Received valid package ! ")
194                             logfile.write("option,{},,,,\n".format(msg))
195                             logfile.flush()
196                     else:
197                         if(opt != -1):
198                             if(opt == 0): # FTP
199                                 if(msg == 0):
200                                     ftp_on
201                                     logfile.write("command,ftp,on,,,,\n".format(msg))
202                                     if(_SERIAL_DEBUG):
203                                         print("[ : ) ] Command -0 - FTP")
204                                     elif(msg == 1):
205                                         logfile.write("command,ftp,off,,,,\n".format(msg))
206                                         if(_SERIAL_DEBUG):
207                                             print("[ : ) ] Command -0 - FTP")
208                                     elif(msg == 1):
209                                         logfile.write("message,{},,,,\n".format(msg))
210                                         if(_SERIAL_DEBUG):
211                                             print("[ : ) ] Command 1 - RECEIVE NEWS")
212                                     else:
213                                         if(_SERIAL_DEBUG):
214                                             print("[ :( ] Command not found")
215
216 pycom.rgbled(0xFF0000)
217 if(_SERIAL_DEBUG):
218     print("[ :I ] Updating message ID")
219 logfile.write("{} ,{} ,{} ,{} ,{} ,{} \n".format(msg_id, wrc, batt_mV_byte, uvi, uva, uvb))
220 logfile.flush()
221 msg_id_file = open("/flash/msgid.conf", "w")
222 if (msg_id + 1 < 512):
223     msg_id_file.write(str(msg_id + 1))
224 else:
225     msg_id_file.write("0")
226 msg_id_file.close()
227 logfile.close()

```

```
226
227 deepsleep_time_ms = 1000
228 #deepsleep_time_ms = _DPSLP_TIMEOUT + 300*(urandom(1)[0] % 50)    # De 0 ate 15 s em 50
229     passos, estrategia anti-colisao!
230 if(_SERIAL_DEBUG):
231     print("[ :I ] Deep sleep time : {} ms".format(deepsleep_time_ms))
232     print("[ :C ] Finished. Bye!")
233     time.sleep(0.001)
234 deepsleep(deepsleep_time_ms)
```

APÊNDICE B – Script UVLoRa Nano Gateway

```

1 """
2     Version: 0.4.3
3     Author: Nelio Goncalves Godoi
4     E-mail: neliogodoi@yahoo.com.br
5 Last Update: 04/12/2019
6     Title: UVLoRa Nano Gateway
7 """
8
9 print("\n[    ] UVLoRa Nano Gateway Starting...")
10 import socket
11 import gc
12 import struct
13 from json import dumps, loads
14 from time import sleep
15 from pycom import heartbeat, rgbled
16 from _thread import start_new_thread
17 from machine import SD, RTC
18 from os import mount, getfree
19 from simple import MQTTClient
20 from network import LoRa, WLAN
21
22 _DEVICE_ID      = 0
23
24 _SERVER          = 'io.adafruit.com'
25 _PORT            = 0
26 _USER             = 'neliogodoi'
27 _PASSWORD        = '92caff99dd304298b13110e28e18f540'
28 _CLIENTE_ID     = 'nano-gt-1'
29
30 _WIFI_SSID       = '3.14canelas'
31 _WIFI_PASS        = 'murtaqgeme2'
32
33 # Usando a limitacao [ SF=10, BW=125kHz, CR=4/8, que e a configuracao
34 # LoRa-MAC uplink com maior alcance possivel seguindo a normas da ANATEL,
35 # o pacote LoRa maximo e 14 bytes
36
37 _LORA_PKG_FORMAT = "!BBHBBBBB"           # device_id, pkg_len, msg_id, wrc, batt, uvi, uva
38 , uvb
38 _LORA_PKG_MSG     = "!BBBBBBBBBBBB"      # device_id, pkg_len, msg[12] # == MAX PKG SIZE
39 ==
40
40 _NTP_DELTA        = 3155673600
41 _NTP_PKT_LEN      = 48
42 _NTP_QUERY         = '\x1b' + (_NTP_PKT_LEN-1) * '\0'
43
44 _SDCARD_EXIST     = False
45 _SAVE_ON_FLASH    = True
46 _SERIAL_DEBUG      = True
47
48 heartbeat(False)
49 rtc = RTC()

```

```

50 rtc.ntp_sync('pool.ntp.org')
51
52 lora = LoRa(mode      = LoRa.LORA ,
53             frequency = 925500000 ,
54             tx_power   = 20 ,
55             sf         = 10 ,
56             bandwidth  = LoRa.BW_125KHZ ,
57             coding_rate = LoRa.CODING_4_8 ,
58             rx_iq      = True)
59 lora_sock = socket.socket(socket.AF_LORA , socket.SOCK_RAW)
60 lora_sock.setblocking(False)
61 print("[ :D ] Hello! I'm Nano Gateway {}".format(_DEVICE_ID))
62
63 location = '/flash'
64 try:
65     sd = SD()
66     location = '/sd'
67     mount(sd, location)
68     _SDCARD_EXIST = True
69     if(_SERIAL_DEBUG):
70         print("[ :] SDCard - {:.3f}GB Free".format(getfree(location)/104857))
71 except:
72     if(_SERIAL_DEBUG):
73         print('[ :() ] Unable to mount SDCard')
74 datalogfilename = '{}/datalog.csv'.format(location)
75 try:    # Primeira abertura do arquivo
76     logfile = open(datalogfilename , 'x')
77     logfile.close()
78     logfile = open(datalogfilename , 'a')
79     logfile.write('Datetime,DeviceID,RSSI,MsgID,UVI,UVA,UVB,Battery,WRC\n')
80     logfile.flush()
81     if(_SERIAL_DEBUG):
82         print('[ :D ] Logfile created!')
83 except: # Abre arquivo para escrita
84     logfile = open(datalogfilename , 'a')
85     if(_SERIAL_DEBUG):
86         print('[ :D ] Logfile open!')
87
88 rgbled(0xFFFF00)
89
90 wlan = WLAN()
91 wlan.init(mode = WLAN.STA)
92 wlan.connect(_WIFI_SSID , auth = (WLAN.WPA2 , _WIFI_PASS) , timeout = 5000 )
93 while not wlan.isconnected():
94     rgbled(0xFF0000)
95     sleep(.5)
96     rgbled(0x000000)
97     sleep(.5)
98 if(_SERIAL_DEBUG):
99     print('[ :D ] WiFi Connected: {}'.format(_WIFI_SSID))
100 rgbled(0x00FF00)
101 sleep(1)
102 rgbled(0x000000)
103
104 def iotcloud_sub_command():
105     global mqtt
106     while True:
107         mqtt.set_callback(subscribe_callback)
108         try:
109             mqtt.connect()

```

```

110     mqtt.subscribe(b"neliogodoi/feeds/nano-gt-1.command")
111     if(_SERIAL_DEBUG):
112         print('[ :| ] IoTCloud subscriber. Waiting... ')
113     while True:
114         try:
115             mqtt.wait_msg()
116         except:
117             break
118     except:
119         try:
120             mqtt.disconnect()
121         except:
122             continue
123
124 def mqtt_publish(topic, msg):
125     global mqtt
126     global logfile
127     try:
128         mqtt.publish(topic, msg)
129         if(_SERIAL_DEBUG):
130             print("[ :D ] MQTT message sent!")
131     except:
132         logfile.write('MQTT CONNECTION LOST, , , , , \n')
133         logfile.flush()
134         if(_SERIAL_DEBUG):
135             print("[ :( ] MQTT message not sent!")
136
137 def send_command(device_id_recv):
138     global downlink_list
139     global lora_sock
140     # print(downlink_list)
141     if( len(downlink_list) > 0 ):
142         for index in downlink_list:
143             if( device_id_recv == index["device_id"]):
144                 try:
145                     pkg = struct.pack(_LORA_PKG_MSG % len(index["option"]),
146                                         index["device_id"],
147                                         len(index["option"]),
148                                         0,
149                                         index["option"])
150                     lora_sock.send(pkg)
151                 except:
152                     pkg = struct.pack(_LORA_PKG_MSG % len(index["option"]),
153                                         index["device_id"],
154                                         len(index["option"]),
155                                         0,
156                                         index["option"])
157                     lora_sock.send(pkg)
158                     sleep(0.5)
159                     pkg = struct.pack(_LORA_PKG_MSG % len(index["command"]),
160                                         index["device_id"],
161                                         len(index["command"]),
162                                         0,
163                                         index["command"])
164                     lora_sock.send(pkg)
165                     break
166
167
168 def subscribe_callback(topic, msg):
169     global downlink_list

```



```
230                     'wrc'      :wrc}{})
231             # topic = 'neliogodoi/groups/node-{}'.format(device_id) # Ubidots
232             topic = '/v1.6/devices/{}'.format(device_id)           # Adafruit IO
233             start_new_thread(mqtt_publish, (topic, msg))
234             rgbled(0x000000)
235     except:
236         if(_SERIAL_DEBUG):
237             print('[ :C ] Unexpected error')
238     finally:
239         logfile.close()
```

APÊNDICE C – *Driver MicroPython para VEML6075*

```

1 """
2 Version: 0.0.2
3 Author: Nelio Goncalves Godoi
4 E-mail: neliogodoi@yahoo.com.br
5 Last Update: 28/11/2019
6 Based on the work by author ladyada (2018) for Adafruit Industries
7 (https://github.com/adafruit/Adafruit_CircuitPython_VEML6075)
8 """
9
10 import time
11 from ustruct import unpack
12
13 _VEML6075_ADDR = const(0x10)
14
15 _REG_CONF      = const(0x00)
16 _REG_UVA       = const(0x07)
17 _REG_DARK      = const(0x08)  # check is true?
18 _REG_UVB       = const(0x09)
19 _REG_UVCOMP1   = const(0x0A)
20 _REG_UVCOMP2   = const(0x0B)
21 _REV_ID        = const(0x0C)
22
23 # Valid constants for UV Integration Time
24 _VEML6075_UV_IT = { 50: 0x00, 100: 0x01, 200: 0x02, 400: 0x03, 800: 0x04 }
25
26 class VEML6075(object):
27
28     def __init__(self,
29                  i2c,
30                  integration_time= 50,
31                  high_dynamic= True,
32                  uva_a_coef= 2.22,
33                  uva_b_coef= 1.33,
34                  uvb_c_coef= 2.95,
35                  uvb_d_coef= 1.74,
36                  uva_response= 0.001461,
37                  uvb_response= 0.002591 ):
38
39         # Set coefficients
40         self._addr = _VEML6075_ADDR
41         self._a = uva_a_coef
42         self._b = uva_b_coef
43         self._c = uvb_c_coef
44         self._d = uvb_d_coef
45         self._uvaresp = uva_response
46         self._uvbresp = uvb_response
47         self._uvacalc = self._uvbcalc = None
48
49         # Init I2C
50         self._i2c = i2c
51         # read ID!

```

```

52     veml_id = self._read_register(_REV_ID)
53     if veml_id != 0x26:
54         raise RuntimeError("Incorrect VEML6075 ID 0x%02X" % veml_id)
55     # shut down
56     self._write_register(_REG_CONF, 0x01)
57     # Set integration time
58     self.integration_time = integration_time
59     # enable
60     conf = self._read_register(_REG_CONF)
61     if high_dynamic:
62         conf |= 0x08
63     conf &= ~0x01 # Power on
64     self._write_register(_REG_CONF, conf)
65
66     def _take_reading(self):
67         """Perform a full reading and calculation of all UV calibrated values"""
68         conf = self._read_register(_REG_CONF)
69         conf &= ~0x01                      # Power on
70         self._write_register(_REG_CONF, conf)
71         time.sleep(0.1)
72         uva = self._read_register(_REG_UVA)
73         uvb = self._read_register(_REG_UVB)
74         #dark = self._read_register(_REG_DARK)
75         uvcomp1 = self._read_register(_REG_UVCOMP1)
76         uvcomp2 = self._read_register(_REG_UVCOMP2)
77         # Equation 1 & 2 in App note, without 'golden sample' calibration
78         self._uvacalc = uva - (self._a * uvcomp1) - (self._b * uvcomp2)
79         self._uvbcalc = uvb - (self._c * uvcomp1) - (self._d * uvcomp2)
80         #print("UVA = %d, UVB = %d, UVcomp1 = %d, UVcomp2 = %d, Dark = %d" %
81         #      (uva, uvb, uvcomp1, uvcomp2, dark))
82         conf |= 0x01 # Power off
83         self._write_register(_REG_CONF, conf)
84
85     @property
86     def uva(self):
87         """The calibrated UVA reading, in 'counts' over the sample period"""
88         self._take_reading()
89         return self._uvacalc
90
91     @property
92     def uvb(self):
93         """The calibrated UVB reading, in 'counts' over the sample period"""
94         self._take_reading()
95         return self._uvbcalc
96
97     @property
98     def uv_index(self):
99         """The calculated UV Index"""
100        self._take_reading()
101        return ((self._uvacalc * self._uvaresp) + (self._uvbcalc * self._uvbresp)) / 2
102
103    @property
104    def integration_time(self):
105        """The amount of time the VEML is sampling data for, in millis.
106        Valid times are 50, 100, 200, 400 or 800ms"""
107        key = (self._read_register(_REG_CONF) >> 4) & 0x7
108        for k, val in enumerate(_VEML6075_UV_IT):
109            if key == k:
110                return val
111        raise RuntimeError("Invalid integration time")

```

```
112
113     @integration_time.setter
114     def integration_time(self, val):
115         if not val in _VEML6075_UV_IT.keys():
116             raise RuntimeError("Invalid integration time")
117         conf = self._read_register(_REG_CONF)
118         conf &= ~ 0b01110000 # mask off bits 4:6
119         conf |= _VEML6075_UV_IT[val] << 4
120         self._write_register(_REG_CONF, conf)
121
122     def _read_register(self, register):
123         """Read a 16-bit value from the 'register' location"""
124         result = unpack('BB', self._i2c.readfrom_mem(self._addr, register, 2))
125         return (result[1] << 8) | result[0]
126
127     def _write_register(self, register, value):
128         """Write a 16-bit value to the 'register' location"""
129         self._i2c.writeto_mem(self._addr, register, bytes([value, value >> 8]))
```