

UNIVERSIDADE FEDERAL DO PARANÁ  
SETOR DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

EDUARDO SEITI NISHIKAWA TABA  
EVERTON LUIZ RUBIO

**DESENVOLVIMENTO DE UMA SOLUÇÃO IOT PARA A  
APLICAÇÃO EM SISTEMAS DE ILUMINAÇÃO PÚBLICA**

CURITIBA  
2019

EDUARDO SEITI NISHIKAWA TABA  
EVERTON LUIZ RUBIO

**DESENVOLVIMENTO DE UMA SOLUÇÃO IOT PARA A  
APLICAÇÃO EM SISTEMAS DE ILUMINAÇÃO PÚBLICA**

Trabalho de conclusão de curso de graduação apresentado como requisito parcial à obtenção do grau de Engenheiro Eletricista ao Curso de Engenharia Elétrica do Setor de Tecnologia da Universidade Federal do Paraná.

Orientador: André Augusto Mariano, Ph. D.

CURITIBA  
2019

# FOLHA DE APROVAÇÃO

EDUARDO SEITI NISHIKAWA TABA  
EVERTON LUIZ RUBIO

## DESENVOLVIMENTO DE UMA SOLUÇÃO IOT PARA A APLICAÇÃO EM SISTEMAS DE ILUMINAÇÃO PÚBLICA

Trabalho de conclusão aprovado como requisito parcial à obtenção do grau de Engenheiro Eletricista ao Curso de Engenharia Elétrica do Setor de Tecnologia da Universidade Federal do Paraná, pela seguinte banca examinadora:

---

**Prof. André Augusto Mariano, Ph. D.**  
Orientador - Departamento de Engenharia Elétrica, UFPR

---

**Prof. Dr. Evelio Martín García Fernández**  
Departamento de Engenharia Elétrica, UFPR

---

**Prof. Tiribíça Krueger Moreira**  
Departamento de Engenharia Elétrica, UFPR

Curitiba, 05 de dezembro de 2019

## AGRADECIMENTOS

Os agradecimentos principais são direcionados às nossas famílias e amigos por terem dado todo o amor, carinho, confiança, paciência e incentivo ao longo de nossas vidas. Sem eles nada disso seria possível.

Agradecimentos também vão para todas as pessoas envolvidas no Departamento de Engenharia Elétrica da UFPR pela dedicação ao trabalho e na formação de futuros engenheiros eletricitas.

Agradecimentos especiais para o nosso professor orientador, tutor e amigo, André Augusto Mariano, pela oportunidade, tempo, apoio e ensinamentos para o desenvolvimento deste trabalho.

## RESUMO

TABA, Eduardo; RUBIO, Everton. DESENVOLVIMENTO DE UMA SOLUÇÃO IOT PARA A APLICAÇÃO EM SISTEMAS DE ILUMINAÇÃO PÚBLICA. 2019. 75 f. Trabalho de Conclusão de Curso – Setor de Tecnologia, Universidade Federal do Paraná. Curitiba, 2019.

Este trabalho apresenta um estudo, projeto e teste de viabilidade de uma solução IoT para ser aplicada em sistemas de iluminação pública, sendo o principal objetivo o desenvolvimento de uma rede de comunicação de dados para o tráfego de informações entre os pontos de iluminação e um servidor. Assim, obtém-se um sistema inteligente de gerenciamento e monitoramento e suas principais vantagens são a coleta de dados sobre iluminação e controle remoto da luminosidade dos postes. As funcionalidades e o tráfego de dados na rede não precisam estar limitados a valores de intensidade luminosa e comandos do servidor, por isso sensores foram adicionados a cada ponto de iluminação e seus dados coletados são enviados para o servidor periodicamente. O trabalho começa com uma pesquisa de mercado sobre soluções já disponíveis para controle de iluminação pública e seu foco principal é analisar quais protocolos de comunicação costumam ser utilizados para este tipo de aplicação. O protocolo de comunicação está diretamente relacionado a escolha do *hardware* que será utilizado e por isso esta é a primeira etapa do projeto. Uma vez definido o protocolo de comunicação inicia-se o levantamento de requisitos para o projeto que são eles: a escolha do *hardware*, desenvolvimento da rede para comunicação dos dados, desenvolvimento dos circuitos de condicionamento de sinal para os periféricos e desenvolvimento do código do sistema embarcado e de um servidor para controle dos nós da rede. Ao final da fase de desenvolvimento chega-se a um protótipo que pode fazer parte de uma rede de comunicação de dados que se forma e se recupera em caso de falhas de forma autônoma. Um teste de viabilidade e de funcionamento da rede de comunicação é feito para que se possa avaliar diferentes cenários que poderiam acontecer em um caso real. As métricas de avaliação de desempenho da rede foram selecionadas de tal forma que o seu funcionamento pudesse ser generalizado para um aplicação real. Ao final do projeto uma rede com as seguintes características foi obtida: tempo de formação e auto-recuperação em caso de falhas na ordem de minutos, latência na ordem de dezenas de milissegundos e alcance do *link* de comunicação entre os nós da rede que pode chegar até a 100 metros. Todos esses resultados combinados viabilizam o projeto para sua aplicação em sistemas de iluminação pública.

**Palavras-chave:** Redes *Mesh*, Iluminação Pública Inteligente, Gerenciamento Remoto.

## ABSTRACT

TABA, Eduardo; RUBIO, Everton. DEVELOPMENT OF A IOT SOLUTION FOR APPLICATION IN STREET LIGHT SYSTEMS. 2019. 75 f. Trabalho de Conclusão de Curso – Setor de Tecnologia, Universidade Federal do Paraná. Curitiba, 2019.

This work presents a study, design and feasibility test of an IoT solution to be applied in public lighting systems. The main goal is to develop a communication network for information traffic between lighting points and a server. Thus, an intelligent management and monitoring system is obtained and its main advantages are data collection of lighting and remote control of lamp brightness. Data traffic on the network need not to be limited to lighting values and server commands, therefore sensors have been added to each lighting point and their collected data are sent to the server from time to time. The work begins with market research on solutions already available for street lighting control. The main focus is to analyze which communication protocols are commonly used for this type of application. The communication protocol is directly related to the hardware's choice and therefore this is the first step of the project. Once the communication protocol has been chosen, the requirement gathering for the project's development followed this order: hardware choice, communication network development, signal conditioning circuits design for the peripherals and embedded system and server code. At the end of the development phase, the final prototype designed can be part of a data communication network that is self-organizing and self-healing, meaning the network can be built and maintained autonomously. A feasibility and functioning test of the communication network is made to evaluate different scenarios that could happen in a real case. The network performance metrics were selected such that the conclusions drawn from them could be generalized to a real application. At the end of the project, a communication network with the following characteristics was obtained: formation and self-recovery with time in the order of minutes, latency with time in the order of tens of milliseconds and a communication link between nodes that can reach up to 100 meters. All of these results combined make the project feasible for its application in street lighting systems.

**Keywords:** Mesh Networks, Smart Street Lighting, Remote Management.

## LISTA DE FIGURAS

Figura 1 – Rede de sensores sem fio. . . . .	18
Figura 2 – Rede Wi-Fi tradicional. . . . .	19
Figura 3 – Arquitetura de rede ESP-MESH. . . . .	20
Figura 4 – Topologia em árvore ESP-MESH. . . . .	21
Figura 5 – Tipos de nós ESP-MESH. . . . .	22
Figura 6 – Efeitos do limite de RSSI. . . . .	23
Figura 7 – Seleção de nó pai preferido. . . . .	24
Figura 8 – Tabelas de roteamento ESP-MESH. . . . .	24
Figura 9 – Processo de construção da rede ESP-MESH. . . . .	25
Figura 10 – Exemplo de eleição do nó raiz. . . . .	26
Figura 11 – Exemplo de designação do nó raiz. . . . .	27
Figura 12 – Exemplo de criação da rede com inicialização assíncrona. . . . .	29
Figura 13 – Auto-reparação de falha do nó raiz. . . . .	30
Figura 14 – Auto reparação em caso de perda de pai intermediário. . . . .	31
Figura 15 – Exemplo de troca do nó raiz. . . . .	32
Figura 16 – Pacote ESP-MESH. . . . .	33
Figura 17 – Representação gráfica do conceito PWM. . . . .	35
Figura 18 – Esquemático do circuito amplificador operacional não inversor. . . . .	36
Figura 19 – Diagrama de blocos para a execução do projeto. . . . .	37
Figura 20 – ESP32 WROOM32 DevKit v1. . . . .	40
Figura 21 – Vista superior do local para o teste de alcançabilidade da comunicação Wi-Fi. . . . .	42
Figura 22 – Ilustração, símbolo elétrico e curva característica do LDR. . . . .	44
Figura 23 – DHT22. . . . .	45
Figura 24 – Diagrama de blocos para o projeto dos circuitos auxiliares. . . . .	46
Figura 25 – Circuitos de atuação dos LEDs e sensoriamento de corrente elétrica. . . . .	47
Figura 26 – Circuitos de sensoriamento de luminosidade e temperatura/umidade. . . . .	48
Figura 27 – Circuito para alimentação do <i>hardware</i> e dos periféricos. . . . .	48
Figura 28 – Gráfico da curva característica do LDR. . . . .	49
Figura 29 – Gráfico linearizado da curva característica do LDR. . . . .	50
Figura 30 – Interface gráfica do servidor. . . . .	54
Figura 31 – Exemplo de conexão entre rede <i>mesh</i> e servidor. . . . .	55
Figura 32 – Vista superior do sistema eletrônico projetado. . . . .	55
Figura 33 – Gabinete plástico para o suporte do sistema eletrônico projetado. . . . .	56
Figura 34 – As 10 unidades participantes dos testes. . . . .	57
Figura 35 – Vista panorâmica do local escolhido para os testes. . . . .	58

Figura 36 – Diagrama com a disposição dos pontos de iluminação do local de testes.	59
Figura 37 – A instalação do sistema eletrônico no poste. . . . .	60
Figura 38 – Primeiro cenário: formação da rede com 9 nós participantes. . . . .	62
Figura 39 – Segundo cenário: inserção de um novo nó à rede. . . . .	64
Figura 40 – Terceiro cenário: falha do nó raiz. . . . .	66
Figura 41 – Quarto cenário: retorno do nó falho do cenário anterior. . . . .	67
Figura 42 – Quinto cenário: falha de um nó pai intermediário. . . . .	68
Figura 43 – Sexto cenário: falha de um segundo nó pai intermediário. . . . .	69
Figura 44 – Monitoração de consumo de corrente elétrica pelos LEDs. . . . .	72



## LISTA DE QUADROS

Quadro 1 – Quadro comparativo entre as opções de <i>hardware</i> para o projeto. . . .	39
--	----

## LISTA DE TABELAS

Tabela 1 – Especificações técnicas do ESP32. . . . .	41
Tabela 2 – Resultados obtidos do teste de alcançabilidade da comunicação Wi-Fi. . . . .	43
Tabela 3 – Resultados dos tempos de recuperação em caso de perda de um nó pai intermediário. . . . .	70
Tabela 4 – Resultados dos tempos de ida e volta de pacotes entre o servidor e diferentes camadas da rede. . . . .	71

## LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog to Digital Converter
ANEEL	Agência Nacional de Energia Elétrica
CC	Corrente Contínua
DELT	Departamento de Engenharia Elétrica
ESP-IDF	Espressif IoT Development Framework
ESP-MDF	Espressif Mesh Development Framework
GPIO	General Purpose Input/Output
IoT	Internet of Things
LED	Light Emitting Diode
LDR	Light Dependent Resistor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
PWM	Pulse Width Modulation
RSSI	Received Signal Strength Indication
TIC	Tecnologias de Informação e Comunicação
UFPR	Universidade Federal do Paraná

## SUMÁRIO

<b>1 – INTRODUÇÃO</b>	<b>14</b>
1.1 CONTEXTO	14
1.2 OBJETIVOS	15
1.2.1 Objetivo Geral	15
1.2.2 Objetivos Específicos	15
1.3 JUSTIFICATIVA	15
1.4 ESTRUTURA DO TRABALHO	16
<b>2 – FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1 REDES INTELIGENTES	17
2.1.1 Rede de Sensores Sem Fio	18
2.2 PROTOCOLO ESP-MESH	19
2.2.1 Conceitos	20
2.2.1.1 Arquitetura em Árvore	20
2.2.1.2 Tipos de Nós	21
2.2.1.3 Quadros de Sinalização e Limite de RSSI	22
2.2.1.4 Nó Pai Preferido	22
2.2.1.5 Tabelas de Roteamento	23
2.2.2 Construção da Rede	25
2.2.2.1 Processo Geral	25
2.2.2.2 Seleção Automática do Nó Raiz	26
2.2.2.3 Nó Raiz Designado Pelo Usuário	27
2.2.2.4 Seleção de Nó Pai	28
2.2.2.5 Inicialização Assíncrona	28
2.2.2.6 Prevenção de <i>Loopback</i>	29
2.2.3 Gerenciamento da Rede	30
2.2.3.1 Falha do Nó Raiz	30
2.2.3.2 Falha de Nó Pai Intermediário	31
2.2.3.3 Troca de Nó Raiz	31
2.2.3.4 Troca de Nó Pai	33
2.2.4 Transmissão de Dados	33
2.2.4.1 Pacotes ESP-MESH	33
2.2.4.2 Controle de Fluxo <i>Upstream</i>	34
2.3 DIMERIZAÇÃO	34
2.4 MODULAÇÃO POR LARGURA DE PULSO	34
2.5 MOSFET OPERANDO COMO CHAVE	35

2.6	AMPLIFICADOR OPERACIONAL NÃO INVERSOR . . . . .	36
<b>3</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>37</b>
3.1	LEVANTAMENTO DE REQUISITOS . . . . .	38
3.1.1	Requisitos do Projeto . . . . .	39
3.1.2	Seleção do <i>Hardware</i> . . . . .	39
3.2	<i>HARDWARE</i> . . . . .	40
3.2.1	Principais Características . . . . .	40
3.2.2	Especificações Técnicas . . . . .	41
3.2.3	Teste de Alcançabilidade da Comunicação Wi-Fi . . . . .	42
3.3	<i>MESH</i> . . . . .	43
3.4	PERIFÉRICOS . . . . .	44
3.4.1	Sensores Escolhidos . . . . .	44
3.4.1.1	Luminosidade - LDR . . . . .	44
3.4.1.2	Corrente - Resistor <i>Shunt</i> . . . . .	45
3.4.1.3	Temperatura e Umidade - DHT22 . . . . .	45
3.4.2	Atuação Escolhida - LED . . . . .	45
3.4.3	Fonte de Alimentação Escolhida - Associação de Pilhas AA . . . . .	46
3.4.4	Circuitos de Condicionamento de Sinais . . . . .	46
3.4.4.1	Atuação dos LEDs e Sensor de Corrente Elétrica . . . . .	46
3.4.4.2	Sensor de Luminosidade e Temperatura/Umidade . . . . .	47
3.4.4.3	Fonte de Alimentação . . . . .	48
3.4.5	Calibração das medidas . . . . .	48
3.4.5.1	LDR . . . . .	48
3.4.5.2	ADC do ESP32 . . . . .	49
3.5	APLICAÇÃO . . . . .	51
3.5.1	Programação do ESP32 . . . . .	51
3.5.1.1	Programação do Servidor . . . . .	53
3.5.1.2	Conexão Rede Mesh-Servidor . . . . .	54
3.5.2	Aspectos Construtivos . . . . .	55
<b>4</b>	<b>TESTES . . . . .</b>	<b>57</b>
4.1	DESCRIÇÃO GERAL . . . . .	57
4.1.1	Do Local . . . . .	58
4.1.2	Da Instalação . . . . .	59
4.2	CENÁRIOS A SEREM ANALISADOS . . . . .	60
4.3	MÉTRICAS A SEREM AVALIADAS . . . . .	61
<b>5</b>	<b>RESULTADOS E DISCUSSÕES . . . . .</b>	<b>62</b>
5.1	DOS CENÁRIOS . . . . .	62

5.1.1	Cenário 1 . . . . .	62
5.1.2	Cenário 2 . . . . .	64
5.1.3	Cenário 3 . . . . .	65
5.1.4	Cenário 4 . . . . .	65
5.1.5	Cenário 5 . . . . .	67
5.1.6	Cenário 6 . . . . .	68
5.2	DAS MÉTRICAS . . . . .	69
5.2.1	Quadros de Sinalização Wi-Fi . . . . .	69
5.2.2	Tempo de Recuperação . . . . .	70
5.2.3	Latência na Rede . . . . .	70
5.2.4	Alcance para o Link entre Dois Nós . . . . .	71
5.3	DAS APLICABILIDADES . . . . .	72
5.3.1	Atuação dos LEDs . . . . .	72
5.3.2	Consumo de Corrente Elétrica e Detecção de Falhas . . . . .	72
<b>6 – CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>		<b>73</b>
<b>Referências . . . . .</b>		<b>74</b>

# 1 INTRODUÇÃO

## 1.1 CONTEXTO

A humanidade está vivenciando novas e constantes mudanças em seus processos produtivos nos últimos anos, caracterizadas pelo uso de tecnologias que permitem a fusão entre os meios físico, digital e biológico. Esse movimento de cunho global é denominado indústria 4.0 ou quarta revolução industrial. Originado no ano de 2011, o que era apenas para ser um plano estratégico anunciado na Feira Industrial de Hannover, na Alemanha, como uma forma de incentivo à informatização da manufatura, atualmente ela passa a ter princípios bem definidos e já com diversas aplicações (SCHWAB, 2016).

Enquanto as revoluções anteriores trouxeram grandes avanços: a primeira, marcada pelo uso da energia mecânica e motores a vapor; a segunda, caracterizada pelo emprego de energia elétrica e motores a combustão, além da consolidação de linhas de produção; e a terceira, com o desenvolvimento da eletrônica e da robótica, juntamente com o emprego da automação; a quarta revolução industrial terá um impacto mais profundo, complexo e exponencial (FREITAS, 2018). Porque ela permitirá a integração de todos os avanços conquistados e supervisionados por uma inteligência artificial, tornando os processos contínuos, mais eficientes, virtuais, descentralizados, flexíveis e autônomos.

Um dos pilares responsáveis para a viabilização da indústria 4.0 e o cerne deste trabalho é a Internet das Coisas (*Internet of Things* - IoT). Ela pode ser descrita como uma rede de comunicação instantânea e inteligente de tecnologias que estão presentes no cotidiano das pessoas, buscando proporcionar a conectividade entre os meios físico e digital (SOUZA; JUNIOR; NETO, 2017).

Estimulada pela crescente quantidade de Tecnologias de Informação e Comunicação (TIC), uma série de tendências começou a surgir com a aplicabilidade da IoT. Uma delas é o *Smart Lighting*. Empregado já em algumas cidades pelo mundo, é um conceito que procura estabelecer alternativas para o controle e a supervisão de sistemas de iluminação pública, proporcionando reduções adicionais nos custos de operação e manutenção, como uma forma de incentivo na busca pela eficiência energética (ANEEL, 2019).

De maneira geral, *Smart Lighting* caracteriza-se pelo emprego de uma "inteligência" em cada ponto de iluminação pública, que por sua vez, comunicará com seus adjacentes via rede *wireless*, criando uma rede integrada capaz de processar comandos como também dispor informações de operação para uma central de controle. O padrão de rede *wireless* é o *mesh*, por ser o mais adequado para esse tipo de aplicação, pois a sua arquitetura, em malha ou árvore, possibilita várias formações de uma rede que se adaptam as características geográficas das instalações existentes em iluminação públicas, além de ser tolerantes a falhas de comunicação (BUENO; NALON, 2016).

Por se tratar de uma tendência bem atrativa para órgãos responsáveis pela prestação do serviço de iluminação pública, projetos em *Smart Lighting* vêm sendo concebidos em algumas cidades pelo mundo com resultados satisfatórios, o que abre oportunidades para a implementação em outras. Este trabalho visa apresentar uma solução que poderá ter o potencial de futuramente ser implementado em cidades brasileiras.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Desenvolver e avaliar uma solução IoT de baixo custo e de baixo consumo, utilizando o conceito de redes *wireless mesh*, com a possibilidade futura de aplicação em sistemas de iluminação pública, visando a promoção de uma rede integrada e inteligente.

### 1.2.2 Objetivos Específicos

- Realizar uma pesquisa de mercado e seleção de um *hardware* disponível para a implementação de redes *mesh*, além de averiguar a sua capacidade para o projeto.
- Efetuar um estudo teórico relacionado aos conceitos de infraestrutura e roteamento de redes *wireless mesh*.
- Implementar um modelo de rede *wireless mesh*.
- Selecionar tecnologias, sensores e atuadores, que serão integradas à rede, além de projetar circuitos auxiliares para a interface com o sistema eletrônico escolhido.
- Promover a solução IoT, para uma situação exemplo de aplicação, com a implementação da funcionalidade de uma rede integrada e inteligente.
- Realizar testes e analisar os respectivos resultados obtidos na situação exemplo criada, bem como avaliar a viabilidade do projeto.

## 1.3 JUSTIFICATIVA

Este trabalho visa apresentar uma solução de baixo custo e de baixo consumo para um momento que está sendo incorporado novas tendências tecnológicas quando o assunto é a IoT, como o *Smart Lighting* em sistemas de iluminação pública.

Além disso, segundo The World Bank, no relatório "Iluminando Cidades Brasileiras", as redes de iluminação pública constituem mais de 4% do consumo total de energia no país, e o custo de energia para iluminação pública já representa o segundo maior item orçamentário de grande parte de municípios. Assim, projetos de eficiência energética para esse setor têm sido bastante reconhecidos nos últimos anos. E pequenos passos já foram dados, como o emprego de luminárias que utilizam a tecnologia de LEDs (*Light Emitting Diodes*) em algumas cidades brasileiras. (BANK, 2016). Agora, há uma necessidade de tecnologias que permitam a promoção de sistemas inteligentes de gestão e controle para



supervisionar essa nova infraestrutura de luminárias eficientes. O que motiva a realização deste trabalho.

#### 1.4 ESTRUTURA DO TRABALHO

O trabalho está organizado em seis capítulos:

- O [Capítulo 1](#), Introdução, apresenta o contexto, os objetivos gerais e específicos a serem alcançados e a justificativa para a realização do trabalho.
- O [Capítulo 2](#), Fundamentação Teórica, discorre sobre os principais conceitos envolvidos para o desenvolvimento do trabalho.
- O [Capítulo 3](#), Materiais e Métodos, apresenta os procedimentos efetuados e os materiais utilizados para atingir os objetivos propostos.
- O [Capítulo 4](#), Testes, descreve as condições gerais, os cenários a serem analisados e as métricas avaliadas para a análise das funcionalidades da solução IoT desenvolvida.
- O [Capítulo 5](#), Resultados e Discussões, relata os resultados obtidos dos testes, assim como avalia a viabilidade do projeto.
- O [Capítulo 6](#), Conclusão e Trabalhos Futuros, encerra com uma análise geral do trabalho e propõe a perspectiva de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 REDES INTELIGENTES

A conectividade entre dispositivos, e a conexões dos dispositivos a uma rede de transmissão de dados, tem criado novos horizontes para o desenvolvimento de cidades inteligentes, sistemas de transporte inteligentes e a Internet das Coisas (IoT). Porém, para que essas tendências tenham sucesso, é necessário uma conexão confiável, estável e segura entre os dispositivos e a rede. Isso requer, acima de tudo, uma infraestrutura de redes de comunicação bem estruturada.

Para que se possa usufruir de todos os benefícios que uma rede com uma grande quantidade de dispositivos conectados tem a oferecer, alguns pontos merecem atenção. Um desses pontos é que deve ser levado em consideração a conectividade ponta a ponta, desde o sistema final até os nós. Isso porque uma rede com dispositivos que podem se conectar ou desconectar a qualquer momento está sujeita a um crescimento exponencial de dispositivos transmitindo dados na rede, e isso pode levar a um engarrafamento dos dados, grandes atrasos de transmissão e conseqüentemente um queda de performance da rede. Analisar e pensar sobre este problema pode aumentar a escalabilidade da rede e diminuir o tempo de troca de dados entre os dispositivos e seu destino final. Para tanto, elementos da rede devem oferecer, além de uma boa segurança, um bom gerenciamento do tráfego de dados. Isso pode ser atingido com roteadores e *switches* que possuem um análise de dados integrada (FARONI, 2019).

Existem algumas vantagens na utilização de *analytics* e *big data* para análise do tráfego de dados em uma rede. Isso porque, baseado em informações concretas sobre a rede, é possível tomar melhores decisões sobre o encaminhamento de pacotes ou sobre como otimizar a estrutura da rede. Isso é possível através de funções que reportam informações e fazem uma análise preditiva sobre o tráfego de dados.

Quando se há uma grande quantidade de dispositivos conectados a rede, uma das topologias mais adequadas é uma do tipo convergente (FARONI, 2019). Desta forma ocorre uma centralização do fluxo de dados da rede, facilita o caminho da transferência de dados na rede (que tem como efeito uma diminuição dos problemas de compatibilidade), facilita o gerenciamento da rede e diminui custos (BOYD, 2018).

Um outro aspecto muito importante sobre uma grande quantidade de dispositivos conectados à internet é a segurança das informações. Com o crescimento de dispositivos conectados à rede o problema da cibersegurança tende a aumentar também, isso por que cada dispositivo conectado torna-se uma porta de entrada para programas maliciosos (WHALEY, 2019) e atualmente a maior parte dos dispositivos IoT são comercializados com sistemas de segurança muito fracos (GERSHWIN, 2019). O problema da cibersegurança em

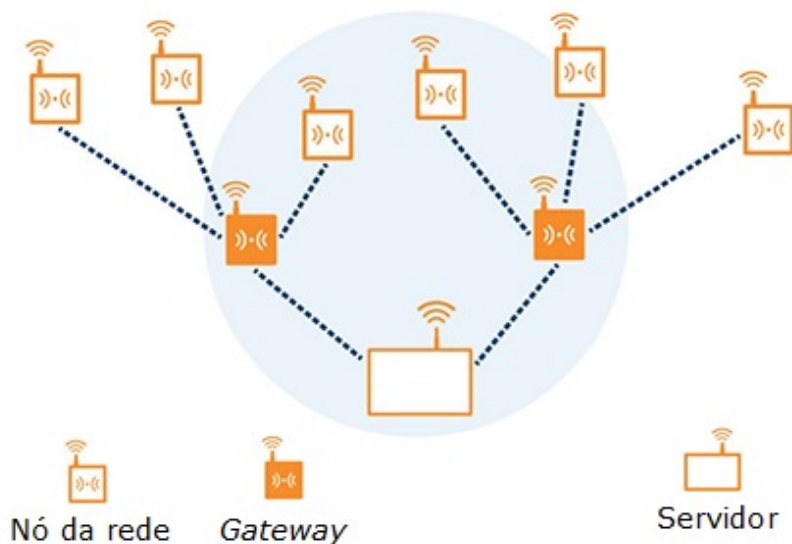
IoT ainda requer bastante desenvolvimento, porém algumas medidas podem ser tomadas para melhorar a segurança da rede como: monitoramento dos pacotes que são enviados dos dispositivos para a nuvem, encriptação do tráfego de dados e utilizar credenciais e senhas fortes para autenticação.

### 2.1.1 Rede de Sensores Sem Fio

Redes de sensores sem fio é uma rede cujo principal objetivo é captar e armazenar informações sobre alguma característica do ambiente como por exemplo: temperatura, umidade, níveis sonoros, níveis de poluição, intensidade luminosa, entre outros. Os dispositivos fazem a coleta dos dados através de sensores instalados e enviam esses dados usando algum protocolo de comunicação (WiFi, ZigBee, Bluetooth) para o servidor, onde é feita a armazenagem e processamento dos dados.

Os principais elementos da rede são: os nós, os concentradores do fluxo de dados na rede, que normalmente estão conectados uma rede externa (*gateway*), e o servidor que coleta e gerencia as informações. As redes mais modernas são bidirecionais, permitindo também o controle da atividade do sensor ou controle de algum atuador presente no nó (DEVICES, 2019). A Figura 1 ilustra um exemplo de conexão entre todos os elementos da rede de sensores sem fio.

Figura 1 – Rede de sensores sem fio.



Fonte: Adaptado *ANALOG DEVICES* (2019).

A estrutura básica de um nó se resume: a bateria para alimentação de todos os periféricos, sensores para coleta do dados, rádio para comunicação sem fio e um microcontrolador que coordena a coleta de dados e a comunicação com outros periféricos.

As principais características de um rede de sensores sem fio são:

- Capacidade de lidar com a falha de algum nó. Isso torna a rede mais resiliente.
- Escalabilidade para que possa haver uma grande quantidade de nós conectados a rede.
- Permitir a mobilidade dos nós sem perda de conexão com a rede.
- Capacidade de suportar condições ambientais adversas.
- Baixo consumo de energia, isso por que os dispositivos funcionam de maneira autônoma e por muitas vezes em locais remotos, o que dificulta a troca de baterias.

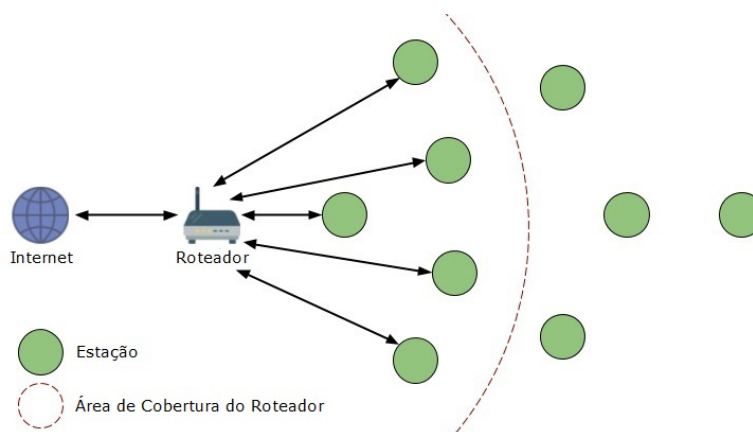
## 2.2 PROTOCOLO ESP-MESH

O protocolo ESP-MESH é um manual de informações elaborado pelos engenheiros da empresa *Espressif Systems* para a concepção de redes *wireless mesh* em sua linha de *hardware* ESP. Os conceitos e as funcionalidades do protocolo descritas a seguir foram desenvolvidas a partir da consulta desse manual (SYSTEMS, 2019b).

O protocolo ESP-MESH é construído com auxílio do protocolo Wi-Fi (IEEE 802.11 b/g/n). Cada um dos dispositivos (aqui chamados de "nós") criam uma rede Wi-Fi em que outros nós podem se conectar, criando assim uma rede local sem fio. O protocolo é auto-organizável e auto-reparável, assim a rede é criada e mantida de forma autônoma.

A estrutura tradicional de uma rede local sem fio é do tipo ponto a multiponto. Nesta topologia, caso algum dispositivo queira fazer parte da rede, ele deve obrigatoriamente conectar-se a um nó conhecido como ponto de acesso. Como consequência disso, a rede acaba sofrendo com dois fatores limitantes. O primeiro deles é que se um dispositivo quiser se conectar a rede, ele deve estar dentro da área de cobertura do ponto de acesso. O segundo é que existe um número máximo de dispositivos que podem se conectar a rede está limitada a capacidade máxima do ponto de acesso. A Figura 2 ilustra uma arquitetura tradicional de Wi-Fi e exemplifica os dois problemas aqui citados.

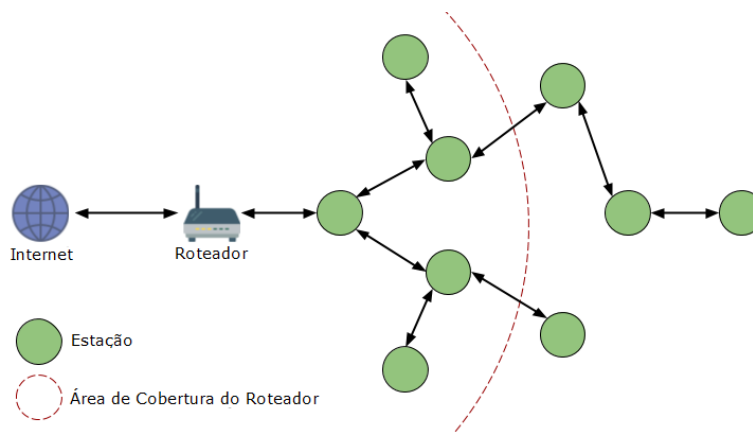
Figura 2 – Rede Wi-Fi tradicional.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

Na topologia ESP-MESH estes dois problemas são resolvidos, isso por que os nós não precisam necessariamente se conectar a um ponto de acesso, mas podem conectar-se entre si. Isso amplia a área de cobertura da rede local sem fio e espalha entre todos os nós da rede a carga de dispositivos conectados. A Figura 3 ilustra a arquitetura de rede ESP-MESH.

Figura 3 – Arquitetura de rede ESP-MESH.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

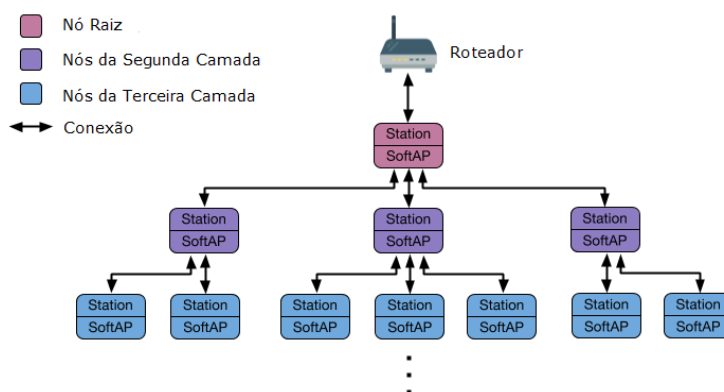
## 2.2.1 Conceitos

### 2.2.1.1 Arquitetura em Árvore

O protocolo ESP-MESH implementa uma arquitetura em árvore para formação da rede. Cada um dos nós cria uma rede Wi-Fi individual na qual outros nós podem se conectar. O nó que cria a rede Wi-Fi é aqui referido como nó "pai". Aquele que se conecta a rede Wi-Fi do nó pai é aqui referido como nó "filho". Assim que um nó filho se conecta a rede Wi-Fi do nó pai, ele cria a sua própria rede Wi-Fi, se tornando uma opção de conexão para novos nós que venham a fazer parte da rede. A combinação destas redes Wi-Fi individuais dão origem a uma única rede local sem fio.

No protocolo Wi-Fi, as estações são limitadas à uma única conexão com um ponto de acesso. No entanto um ponto de acesso pode se conectar simultaneamente à várias estações. O protocolo ESP-MESH permite que os nós atuem simultaneamente como uma estação e como um ponto de acesso. Portanto, um nó no ESP-MESH pode receber várias conexões quando usado como ponto de acesso (um pai pode ter vários filhos), enquanto simultaneamente possuir uma única conexão usando sua interface de estação (um filho só pode ter um pai). Isso resulta em uma topologia de rede em árvore com uma hierarquia pai-filho que consiste em várias camadas. A Figura 4 exemplifica a arquitetura em árvore do ESP-MESH.

Figura 4 – Topologia em árvore ESP-MESH.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

No ESP-MESH qualquer par de nós da rede podem se comunicar entre si, mesmo que não estejam conectados diretamente na forma pai-filho. Por isso, os nós no ESP-MESH não apenas transmitem seus próprios pacotes, mas podem ser usados simultaneamente como passagem para pacotes de outros nós. Assim, a rede opera de forma de múltiplos saltos, o que significa que os nós podem transmitir pacotes para outros nós na rede através de um ou mais saltos sem fio.

Quando um nó pai envia pacotes para um nó filho, esta transmissão é chamada de *downstream*. Quando um nó filho envia pacotes para o nó pai, esta transmissão é chamada de *upstream*.

#### 2.2.1.2 Tipos de Nós

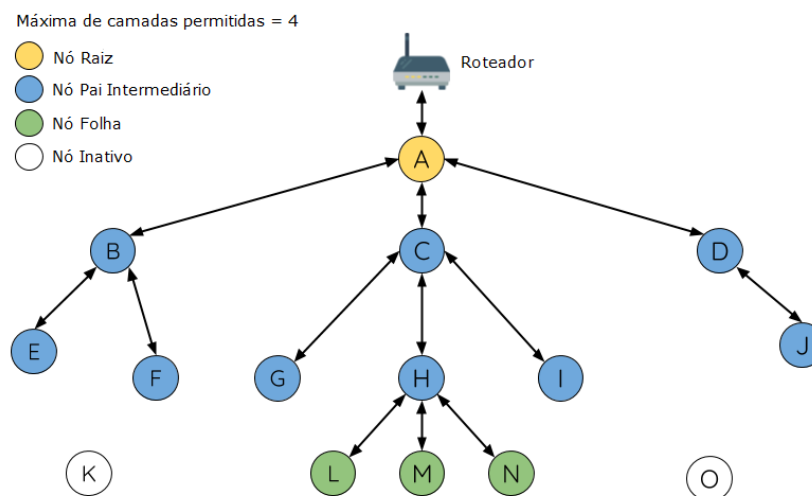
**Nó Raiz:** o nó raiz é o nó superior da rede e serve como a única interface entre a rede ESP-MESH e uma rede IP externa. O nó raiz está conectado a um roteador Wi-Fi convencional e retransmite pacotes para / da rede IP externa para nós na rede ESP-MESH. Só pode haver um nó raiz em uma rede ESP-MESH. A Figura 5 ilustra os 4 tipos de nós da rede ESP-MESH.

**Nós Folha:** um nó folha é um nó que não tem permissão para ter nós filhos. Portanto, um nó folha pode apenas transmitir ou receber seus próprios pacotes, mas não pode encaminhar os pacotes de outros nós. Se um nó estiver situado na camada máxima permitida da rede, ele será atribuído como um nó folha.

**Nós pai intermediários:** nós conectados que não são o nó raiz ou um nó folha são nós pais intermediários. Um nó pai intermediário deve ter uma única conexão *upstream* (um único nó pai), mas pode ter de zero a várias conexões *downstream* (zero a vários nós filhos). Portanto, um nó pai intermediário pode transmitir e receber pacotes, mas também encaminhar pacotes enviados de suas conexões *upstream* e *downstream*.

**Nós inativos:** os nós que ainda não ingressaram na rede são atribuídos como nós

Figura 5 – Tipos de nós ESP-MESH.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

inativos. Os nós inativos tentam formar uma conexão *upstream* com um nó pai intermediário ou tentam se tornar o nó raiz nas circunstâncias corretas.

### 2.2.1.3 Quadros de Sinalização e Limite de RSSI

Todo nó no ESP-MESH que funciona como ponto de acesso transmite periodicamente quadros de sinalização Wi-Fi. Um nó usa quadros de sinalização para permitir que outros nós detectem sua presença e conheçam seu status. Os nós inativos ouvirão os quadros de sinalização para gerar uma lista de possíveis nós pais, um dos quais o nó inativo escolherá para formar uma conexão *upstream*.

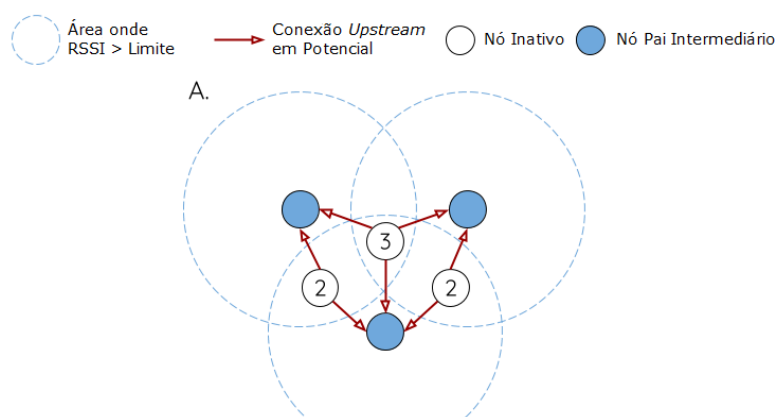
Para impedir que os nós formem uma conexão *upstream* fraca, o ESP-MESH implementa um mecanismo de limite de RSSI (Indicação de intensidade do sinal recebido) para quadros de sinalização. Se um nó detectar um quadro de sinalização com um RSSI abaixo de um limite pré-configurado, o nó que o transmitiu será desconsiderado durante a escolha de um nó pai. A Figura 6 demonstra como o limite do RSSI afeta o número de candidatos ao nó pai que um nó inativo possui.

### 2.2.1.4 Nó Pai Preferido

Quando um nó inativo possui vários candidatos a nó pai, o nó inativo formará uma conexão *upstream* com o nó pai preferido. O nó pai preferido é determinado com base nos seguintes critérios:

- Em qual camada o candidato do nó pai está situado.
- O número de conexões *downstream* que o candidato a nó pai possui atualmente.

Figura 6 – Efeitos do limite de RSSI.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

A seleção do nó pai preferido sempre priorizará o candidato na camada mais rasa da rede (incluindo o nó raiz). Isso ajuda a minimizar o número total de camadas na rede ESP-MESH. Por exemplo, dado um nó da segunda camada e um nó da terceira camada, o nó da segunda camada sempre será preferido.

Se houver vários candidatos ao nó pai na mesma camada, o candidato ao nó pai com o menor número de filhos será o preferido. Este critério tem o efeito de equilibrar o número de conexões *dowstream* entre os nós da mesma camada.

O painel A da [Figura 7](#) demonstra um exemplo de como o nó inativo G seleciona um nó pai preferido, dados os cinco candidatos a nó pai B / C / D / E / F. Os nós na camada mais rasa são preferidos, portanto os nós B / C são priorizados. O nó C é selecionado como o nó pai preferido porque possui menos nós filhos quando comparado com o nó B.

O painel B da [Figura 7](#) demonstra o caso em que o nó raiz está dentro do alcance do nó inativo G. O nó raiz é sempre o nó mais raso em uma rede ESP-MESH, portanto, sempre é o nó pai preferido independente de quantos candidatos a nó pai existam.

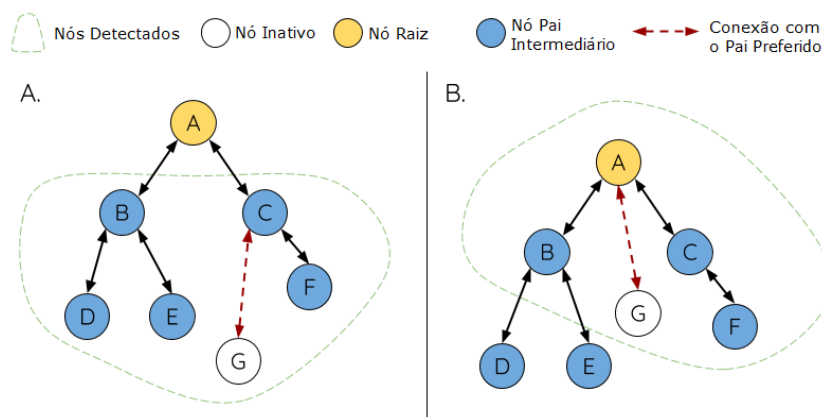
#### 2.2.1.5 Tabelas de Roteamento

Cada nó em uma rede ESP-MESH manterá sua tabela de roteamento individual usada para rotar os pacotes até o nó de destino correto. A tabela de roteamento de um nó específico consistirá nos endereços MAC de todos os nós na sub-rede do nó específico (incluindo o endereço MAC do próprio nó específico). A tabela de roteamento do nó é particionada internamente em várias subtabelas, com cada subtabela correspondente à sub-rede de cada nó filho.

Usando a [Figura 8](#) como exemplo, a tabela de roteamento do nó B consistiria nos endereços MAC dos nós B a I (ou seja, equivalente à sub-rede do nó B). A tabela de roteamento do nó B é particionada internamente em duas subtabelas. Uma delas contém o



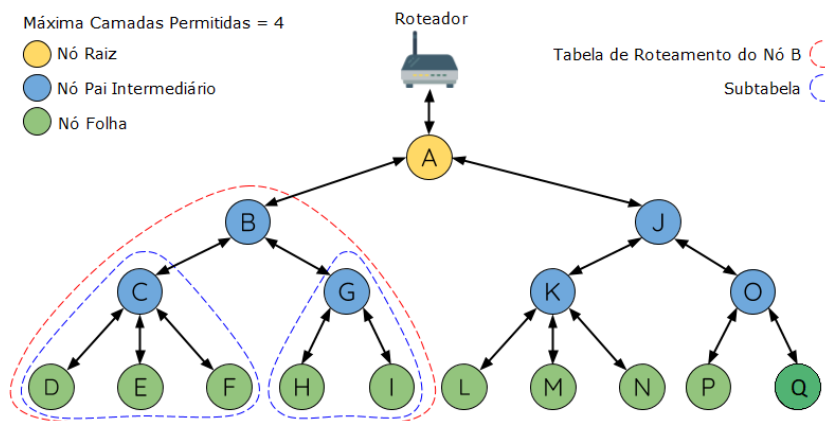
Figura 7 – Seleção de nó pai preferido.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

endereço MAC dos nós de C a F e a outra o endereço MAC dos nós de G a I (sub-redes dos nós C e G, respectivamente).

Figura 8 – Tabelas de roteamento ESP-MESH.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

O ESP-MESH utiliza tabelas de roteamento para determinar se um pacote deve ser encaminhado a uma conexão *upstream* ou *downstream* com base nas seguintes regras:

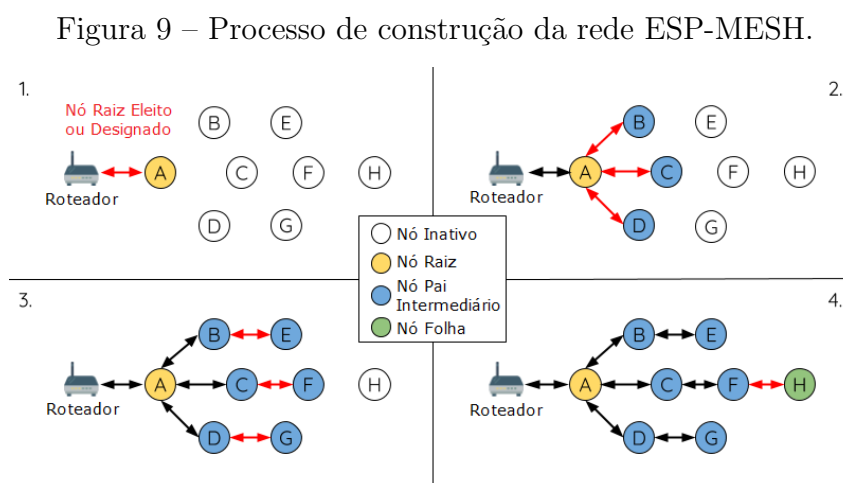
- Se o endereço MAC de destino do pacote estiver na tabela de roteamento do nó atual e não for o nó atual, selecione a subtabela que contém o endereço MAC de destino e encaminhe o pacote para o nó filho correspondente à subtabela.

- Se o endereço MAC de destino não estiver na tabela de roteamento do nó atual, encaminhe o pacote de dados para o nó pai do nó atual. Fazer isso repetidamente fará com que o pacote chegue ao nó raiz, onde a tabela de roteamento deve conter todos os nós na rede.

## 2.2.2 Construção da Rede

### 2.2.2.1 Processo Geral

Um processo de construção de rede ESP-MESH envolve a seleção de um nó raiz e a formação de conexões camada por camada até que todos os nós se unam à rede. O processo de construção da rede ESP-MESH, como ilustrado na [Figura 9](#), pode ser generalizado nas seguintes etapas:



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

1. Seleção de nó raiz: o nó raiz pode ser designado durante a configuração (apenas um dispositivo específico pode ser designado com nó raiz) ou eleito com base na intensidade do sinal entre cada nó e o roteador. Uma vez selecionado, o nó raiz se conectará ao roteador e começará a permitir que conexões *downstream* se formem. Referindo-se à [Figura 9](#), o nó A é selecionado para ser o nó raiz, portanto, o nó A forma uma conexão *upstream* com o roteador.

2. Formação da segunda camada: após o nó raiz estiver conectado ao roteador, os nós inativos no alcance do sinal do nó raiz começarão a se conectar ao nó raiz, formando assim a segunda camada da rede. Uma vez conectados, os nós da segunda camada tornam-se nós pai intermediários. Referindo-se à [Figura 9](#), os nós B a D estão no intervalo do nó raiz. Portanto, os nós B a D formam conexões com o nó raiz e se tornam nós pai intermediários.

3. Formação das camadas restantes: os nós inativos restantes se conectarão aos nós pai intermediários que estiverem dentro do seu alcance, formando assim uma nova camada na rede. Depois de conectados, os nós inativos tornam-se nós pai intermediários ou nós folha. Esta etapa é repetida até que não haja mais nós inativos na rede ou até que a camada máxima permitida da rede seja atingida. Referindo-se à [Figura 9](#), os nós E / F / G se conectam aos nós B / C / D, respectivamente, e se tornam os nós pais intermediários.

4. Limitar a profundidade da árvore: para impedir que a rede exceda o número

máximo permitido de camadas, os nós na camada máxima se tornarão automaticamente nós folha após a conexão. Isso evita que qualquer outro nó ocioso se conecte a eles, impedindo assim a formação de uma nova camada. Assim, se um nó inativo não tiver outro nó pai em potencial, ele permanecerá inativo indefinidamente. Referindo-se à Figura 9, as camadas máximas permitidas da rede são definidas para quatro. Portanto, quando o nó H se conecta, ele se torna um nó folha.

### 2.2.2.2 Seleção Automática do Nó Raiz

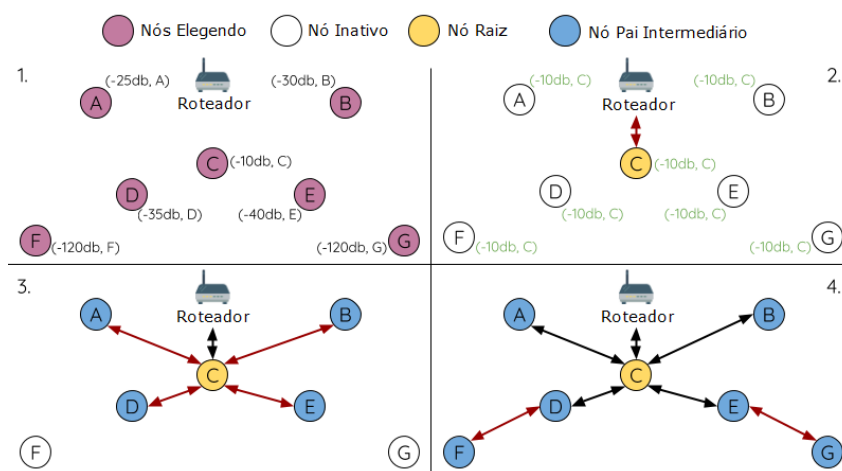
A seleção automática do nó raiz envolve um processo de eleição entre todos os nós inativos, tendo como base a intensidade do sinal (RSSI) que cada um recebe do roteador. Cada nó ocioso transmitirá seus endereços MAC e valores de RSSI do roteador por meio de quadros de sinalização Wi-Fi.

Cada nó procurará simultaneamente os quadros de sinalização de outros nós inativos. Se um nó detectar um quadro de sinalização com um RSSI do roteador mais forte, o nó começará a transmitir o conteúdo desse quadro de farol (ou seja, votando no nó com o RSSI do roteador mais forte). O processo de transmissão e varredura será repetido para um número mínimo pré-configurado de iterações (10 iterações por padrão) e resultará no quadro de sinalização com o RSSI do roteador mais forte sendo propagado por toda a rede.

Após todas as iterações, cada nó verificará individualmente sua porcentagem de votos para determinar se deve se tornar o nó raiz. Se um nó tiver uma porcentagem de votos maior que um limite pré-configurado (90% por padrão), o nó se tornará um nó raiz.

A Figura 10 demonstra como uma rede ESP-MESH é construída quando o nó raiz é selecionado automaticamente.

Figura 10 – Exemplo de eleição do nó raiz.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

1. Na inicialização, cada nó começa a transmitir quadros de sinalização consistindo de seu endereço MAC e do RSSI do roteador.

2. Em várias iterações de transmissão e varredura, o quadro de sinalização com o RSSI do roteador mais forte é propagado por toda a rede. O nó C possui o RSSI do roteador mais forte (-10 dBm), portanto, seu quadro de sinalização é propagado por toda a rede. Todos os nós que participam da eleição votam no nó C, dando ao nó C uma porcentagem de votos de 100%. Portanto, o nó C se torna o nó raiz e se conecta ao roteador.

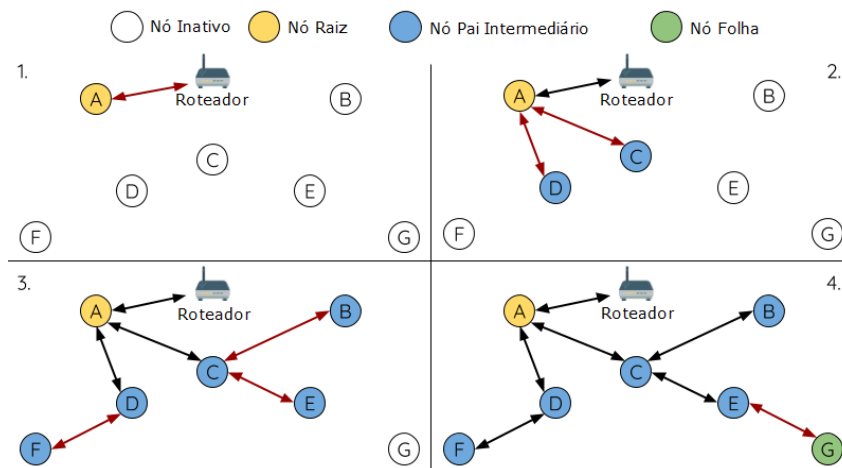
3. Depois que o Nó C se conectar ao roteador, os nós A / B / D / E se conectam ao nó C, pois é o nó pai preferido.

4. Os nós F e G se conectam aos nós D e E, respectivamente, e o processo de criação da rede é concluído.

### 2.2.2.3 Nó Raiz Designado Pelo Usuário

O nó raiz também pode ser designado pelo usuário, isso implica que o nó raiz designado se conecta diretamente ao roteador, renunciando assim ao processo de eleição. Quando um nó raiz é designado, todos os outros nós da rede também renunciam ao processo de eleição para evitar um conflito na escolha do nó raiz. A Figura 11 demonstra como a rede é construída quando o nó raiz é designado.

Figura 11 – Exemplo de designação do nó raiz.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

1. O nó A é designado como nó raiz pelo usuário, portanto, se conecta diretamente ao roteador. Todos os outros nós renunciam ao processo de eleição de um nó raiz.

2. Os nós C / D se conectam ao nó A como seu nó pai preferido. Ambos os nós formam a segunda camada da rede.

3. Da mesma forma, os nós B / E se conectam ao nó C e o nó F se conecta ao nó D. Os nós B / E / F formam a terceira camada da rede.

4. O nó G se conecta ao nó E, formando a quarta camada da rede. No entanto, o número máximo de camadas permitido nesta rede é configurado para quatro. Assim, o nó G se torna um nó folha para impedir a formação de novas camadas.

#### 2.2.2.4 Seleção de Nó Pai

Por padrão, a rede ESP-MESH é auto-organizável, o que significa que cada nó selecionará automaticamente um de seus nós pai em potencial para formar uma conexão *upstream*. O nó pai selecionado é conhecido como o nó pai preferido. Os critérios usados para selecionar o nó pai preferido são projetados para reduzir o número de camadas na rede e equilibrar o número de filhos entre os nós pais.

No entanto, a rede ESP-MESH também permite que os usuários desabilitem o comportamento de auto-organização, o que permite que os usuários definam seus próprios critérios para a seleção de nó pai ou configurem os nós para terem nós pai designados.

#### 2.2.2.5 Inicialização Assíncrona

A construção da rede ESP-MESH pode ser afetada pela ordem em que os nós são ligados. Se alguns nós na rede forem ligados de forma assíncrona (separados por alguns minutos), a estrutura final da rede poderá diferir do caso ideal em que todos os nós são ligados de forma síncrona. Os nós com atraso na inicialização seguirão as seguintes regras:

Regra 1: se um nó raiz já existir na rede, o nó atrasado não tentará eleger um novo nó raiz, mesmo que tenha um RSSI mais forte com o roteador. O nó atrasado ingressará na rede como qualquer outro nó inativo, conectando-se a um nó pai preferido. Se o nó atrasado for o nó raiz designado, todos os outros nós da rede permanecerão inativos até que o nó atrasado seja ligado.

Regra 2: quando um nó atrasado formar uma conexão *upstream* e se tornar um nó pai intermediário, ele também poderá se tornar o novo pai preferido de outros nós (ou seja, ser um nó mais raso). Isso fará com que os outros nós alternem suas conexões *upstream* para se conectarem com o nó atrasado.

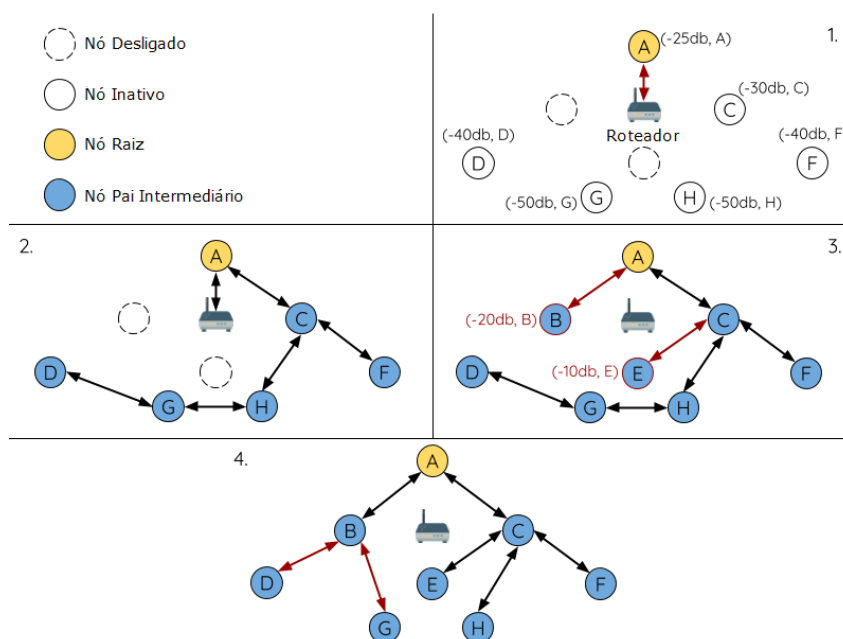
Regra 3: se um nó inativo tiver um nó pai designado com atraso na inicialização, o nó inativo não tentará formar nenhuma conexão *upstream* na ausência de seu nó pai designado. O nó inativo permanecerá inativo indefinidamente até que o nó pai designado seja ligado.

A Figura 12 ilustra um exemplo para demonstrar os efeitos da inicialização assíncrona em relação à construção da rede.

1. Os nós A / C / D / F / G / H são ligados de forma síncrona e iniciam o processo de eleição do nó raiz transmitindo seus endereços MAC e RSSIs do roteador. O nó A é eleito como o nó raiz, pois possui o RSSI mais forte.

2. Depois que o nó A se torna o nó raiz, os nós restantes começam a formar conexões *upstream* camada por camada com seus nós pais preferidos. O resultado é uma

Figura 12 – Exemplo de criação da rede com inicialização assíncrona.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

rede com cinco camadas.

3. Os nós B / E estão atrasados na inicialização, mas nenhum tenta se tornar o nó raiz, mesmo tendo RSSIs de roteador mais fortes (-20 dBm e -10 dBm) em comparação com o nó A. Em vez disso, ambos os nós atrasados formam conexões *Upstream* com seu pai preferido, nós A e C, respectivamente. Ambos os nós B / E tornam-se nós pai intermediários após a conexão.

4. Os nós D / G alternam suas conexões a montante, pois o nó B se torna o novo nó pai preferido por estar em uma camada mais rasa (nó da segunda camada). Devido a troca, a rede resultante possui três camadas, em vez das cinco camadas originais.

#### 2.2.2.6 Prevenção de *Loopback*

Um *loopback* é a situação em que um nó específico forma uma conexão *upstream* com um de seus nós descendentes. Isso resulta em um caminho de conexão circular, quebrando a topologia em árvore. O ESP-MESH impede o *loopback* durante a seleção do nó pai, excluindo os nós já presentes na tabela de roteamento do nó que está fazendo a seleção, impedindo assim que um nó tente se conectar a qualquer outro nó dentro de sua sub-rede.

O ESP-MESH utiliza um mecanismo de verificação de caminho para detectar qualquer ocorrência de *loopback*. Caso algum seja detectado, o nó pai da conexão *upstream* que causou o *loopback* informará o nó filho da situação e iniciará uma desconexão.

### 2.2.3 Gerenciamento da Rede

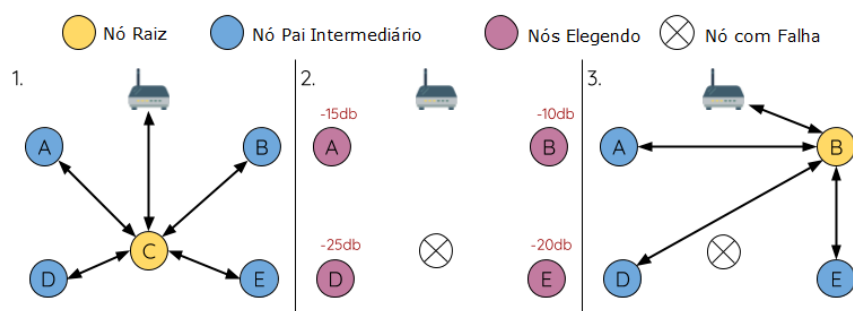
O ESP-MESH é uma rede auto-reparável, o que significa que ele pode detectar e corrigir falhas no roteamento da rede. As falhas ocorrem quando um nó pai é interrompido ou quando a conexão entre um nó pai e seus nós filhos se torna instável. Os nós filhos no ESP-MESH selecionarão automaticamente um novo nó pai e formarão uma conexão *upstream* com ele para manter-se na rede. O ESP-MESH pode manipular tanto falhas de nó raiz quanto falhas de nó pais intermediários.

#### 2.2.3.1 Falha do Nó Raiz

Se o nó raiz for desligado, os nós conectados à ele (nós da segunda camada) detectarão imediatamente sua ausência. Os nós da segunda camada tentarão inicialmente se reconectar com o nó raiz. No entanto, após várias tentativas com falha, os nós da segunda camada inicializarão uma nova rodada de eleição para decidir quem será o novo nó raiz. O nó da segunda camada com o maior RSSI do roteador será eleito como nó raiz, enquanto os nós restantes da segunda camada formarão uma conexão *upstream* com o novo nó raiz (ou um nó pai vizinho, se não estiver dentro do alcance).

Se o nó raiz e várias camadas quebrarem simultaneamente (por exemplo, nó raiz, segunda e terceira camada), a camada mais rasa que ainda estiver funcionando inicializará a eleição do novo nó raiz. A Figura 13 ilustra um exemplo de auto-reparação de uma perda de nó raiz.

Figura 13 – Auto-reparação de falha do nó raiz.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

1. O nó C é o nó raiz da rede. Os nós A / B / D / E são nós da segunda camada conectados ao nó C.

2. O nó C é desligado. Após várias tentativas fracassadas de reconectar, os nós da segunda camada iniciam o processo de eleição transmitindo os RSSIs do roteador. O nó B tem o maior RSSI do roteador.

3. O nó B é eleito como nó raiz e começa a aceitar conexões. Os nós restantes da segunda camada A / D / E formam conexões com o nó B, portanto, a rede é reparada e

pode continuar operando normalmente.

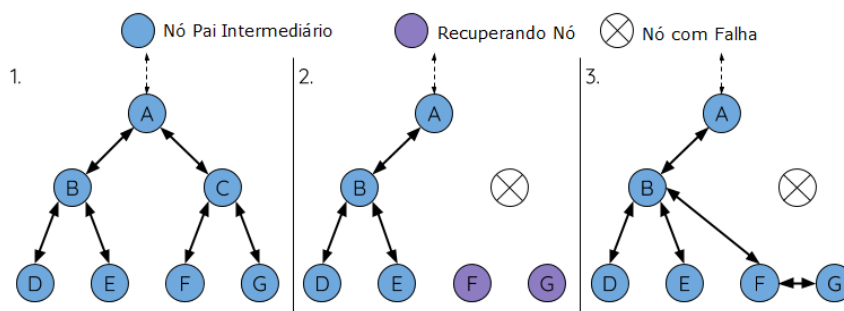
### 2.2.3.2 Falha de Nó Pai Intermediário

Se um nó pai intermediário for desligado, os nós filhos que estavam conectados à ele tentarão inicialmente se reconectar com o nó pai. Após várias tentativas fracassadas de reconectar, cada nó filho começará a procurar possíveis nós pai.

Se outros nós pais potenciais estiverem disponíveis, cada nó filho selecionará individualmente um novo nó pai preferido e formará uma conexão *upstream* com ele. Se não houver outros nós pais em potencial para um nó filho em particular, ele permanecerá ocioso indefinidamente.

A Figura 14 ilustra um exemplo de auto-reparação de uma perda de nó pai intermediário.

Figura 14 – Auto reparação em caso de perda de pai intermediário.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

1. O seguinte diagrama ilustra um rede já formada.

2. O nó C é desligado. Os nós F / G detectam a falha e tentam se reconectar com o nó C. Após várias tentativas sem sucesso de reconexão, os nós F / G começam a selecionar um novo nó pai preferido.

3. O nó G está fora do alcance de qualquer outro nó pai, portanto, permanece inativo por enquanto. O nó F está no alcance dos nós B / E, mas o nó B é selecionado por ser o nó mais raso. O nó F se torna um nó pai intermediário após a conexão com o nó B, assim o nó G pode se conectar com o nó F, que está ao seu alcance.

### 2.2.3.3 Troca de Nó Raiz

O ESP-MESH não troca automaticamente o nó raiz, a menos que o nó raiz seja desligado. Mesmo que o RSSI do roteador no nó raiz se degrade até o ponto de desconexão, o nó raiz permanecerá inalterado. Porém, nos casos em que a RSSI do roteador esteja muito baixa, o nó raiz pode acionar uma nova eleição para verificar se algum outro nó possui condições melhores de conexão com o roteador. O nó raiz atual sinalizará todos

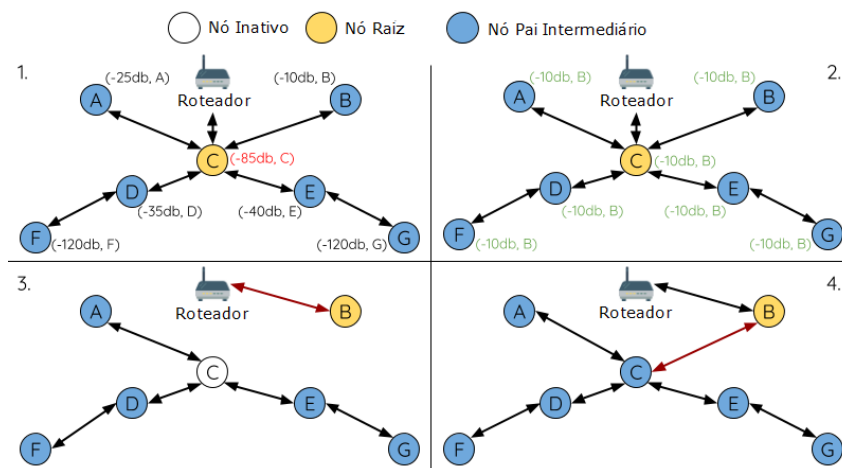


os nós na rede para começar a transmitir e procurar quadros de sinalização. Se outro nó receber mais votos do que o nó raiz atual, uma troca de nó raiz será iniciada. O nó raiz permanecerá inalterado caso contrário.

Um nó raiz recém-eleito envia uma solicitação de comutação para o nó raiz atual que, por sua vez, responderá com uma confirmação, indicando que os dois nós estão prontos para alternar. Depois que a confirmação é recebida, o nó raiz recém-eleito se desconectará do pai e imediatamente formará uma conexão *upstream* com o roteador, tornando-se o novo nó raiz da rede. O nó raiz anterior será desconectado do roteador enquanto mantém todas as suas conexões *downstream* e entrará no estado ocioso. O nó raiz anterior começará a verificar possíveis nós pai e a selecionar um pai preferido.

A Figura 15 ilustra um exemplo de uma eleição para escolha de um novo nó raiz.

Figura 15 – Exemplo de troca do nó raiz.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

1. O nó C é o nó raiz atual, mas a RRSI do roteador diminuiu para -85 dBm. O nó C dispara uma nova eleição e todos os nós começam a transmitir e a procurar quadros de sinalização.

2. Após várias rodadas de transmissão e varredura, o nó B é eleito como o novo nó raiz. O nó B envia ao nó C uma solicitação de comutação e o nó C responde com uma confirmação.

3. O nó B se desconecta do pai e se conecta ao roteador, tornando-se o novo nó raiz da rede. O nó C se desconecta do roteador, entra no estado ocioso e então procura e seleciona um novo nó pai. O nó C mantém todas as suas conexões *downstream* durante todo esse processo .

4. O nó C seleciona o nó B como seu nó pai preferido, forma uma conexão *upstream* e se torna um nó da segunda camada. O layout da rede é semelhante após a comutação, pois o nó C ainda mantém a mesma sub-rede. No entanto, cada nó na sub-rede do nó C

foi colocado uma camada mais funda como resultado da troca de nó raiz.

#### 2.2.3.4 Troca de Nó Pai

A troca do nó pai implica que um nó filho alterne sua conexão *upstream* para outro nó pai de uma camada mais rasa. A troca de nó pai ocorre de forma autônoma, o que significa que um nó filho mudará sua conexão *upstream* automaticamente se um nó pai em potencial de uma camada mais rasa se tornar disponível (devido a uma inicialização assíncrona, por exemplo).

Todos os nós pais em potencial transmitem periodicamente quadros de sinalização, permitindo que um nó filho verifique a disponibilidade de um nó pai mais raso. O intuito principal da comutação de um nó pai é poder ajustar dinamicamente o layout da rede e garantir que cada conexão tenha uma boa RSSI e que o número de camadas na rede seja minimizado.

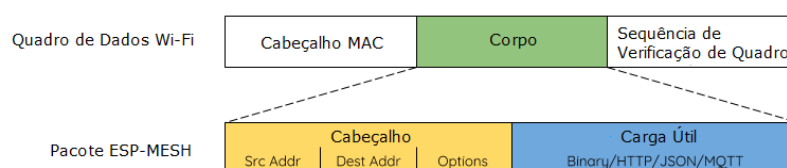
### 2.2.4 Transmissão de Dados

#### 2.2.4.1 Pacotes ESP-MESH

As transmissões de dados na rede usam pacotes ESP-MESH e estes estão totalmente contidos no corpo de um quadro de dados Wi-Fi. Uma transmissão de dados multi-saltos envolverá um único pacote ESP-MESH sendo transportado, a cada salto, por um quadro de dados Wi-Fi diferente.

A [Figura 16](#) mostra a estrutura de um pacote ESP-MESH e sua relação com um quadro de dados Wi-Fi.

Figura 16 – Pacote ESP-MESH.



Fonte: Adaptado ESPRESSIF SYSTEMS (2019).

O cabeçalho de um pacote ESP-MESH contém os endereços MAC dos nós de origem e destino. O campo de opções contém informações referentes aos tipos especiais de pacotes ESP-MESH, como uma transmissão em grupo ou um pacote originário da rede IP externa. A carga útil de um pacote ESP-MESH contém os dados reais do aplicativo.

#### 2.2.4.2 Controle de Fluxo *Upstream*

O ESP-MESH depende do nó pai para controlar o recebimento de dados *upstream* de seus nós filho imediatos. Para impedir que o *buffer* de mensagens de um nó pai transborde devido a uma sobrecarga de transmissões *upstream*, um nó pai alocará uma cota para transmissões *upstream*, conhecidas como janela de recebimento, para cada um de seus nós filho. Cada nó filho deve solicitar uma janela de recebimento antes de poder transmitir dados ao nó pai. O nó filho transmite o pacote de dados de acordo com o tamanho da janela especificado pelo nó pai. Se o nó filho esgotar sua janela de recebimento, ele deverá obter outras janelas de recebimento enviando uma solicitação antes de poder continuar transmitindo.

Devido ao fato de o nó raiz agir como a única interface com uma rede externa, é essencial que os nós *downstream* estejam cientes do status da conexão do nó raiz com a rede externa. Caso contrário, isso pode fazer com que os nós tentem passar dados *upstream* para o nó raiz enquanto ele estiver desconectado da rede externa. Isso resulta em transmissões desnecessárias e perda de pacotes. O ESP-MESH soluciona esse problema fornecendo um mecanismo para estabilizar a taxa de transferência de dados de saída com base no status da conexão entre o nó raiz e a rede IP externa. O nó raiz pode transmitir seu status de conexão de rede externa para todos os outros nós.

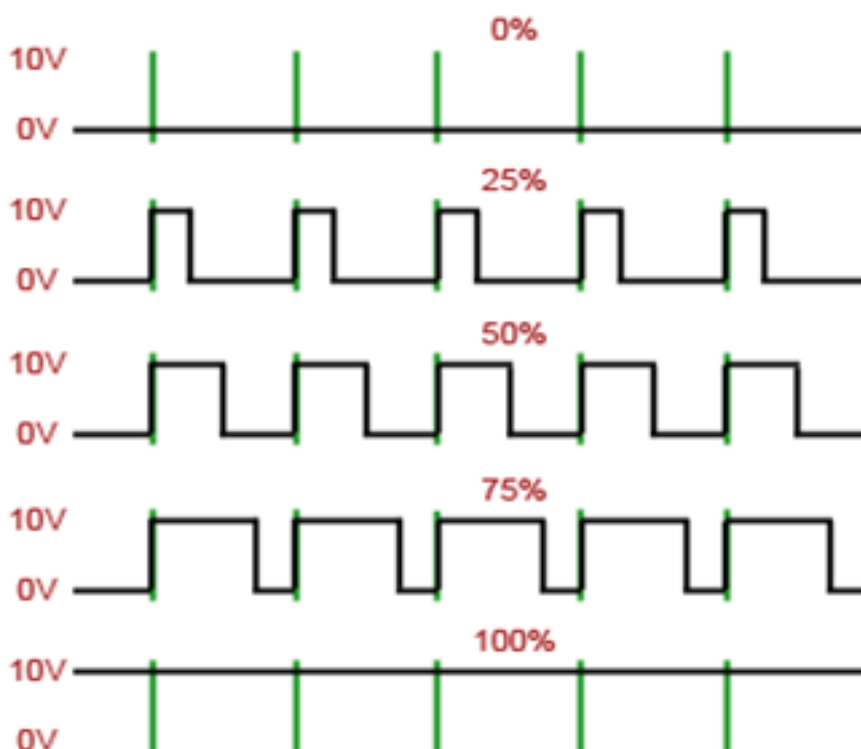
### 2.3 DIMERIZAÇÃO

Processo de controle da intensidade luminosa de uma determinada fonte de luz, por meio da variação de tensão ou corrente elétrica sobre a mesma. Os LEDs, utilizadas neste trabalho, têm a capacidade de serem dimerizáveis, todavia necessitam a conexão de *drivers* (circuitos de acionamento) para dispor tensões e correntes dentro dos limites estabelecidos pelo fabricante. A dimerização pode ser realizada através da modulação de largura de pulso (*Pulse Width Modulation* - PWM) com frequências de modulação elevadas, reprimindo o efeito *flicker*, um fenômeno também conhecido como pisca-pisca (MONTEIRO; CARVALHO; NOGUEIRA, 2014).

### 2.4 MODULAÇÃO POR LARGURA DE PULSO

Denominada também como PWM (*Pulse Width Modulation*), é uma técnica de tratamento do sinal que consiste em manter a amplitude dos pulsos constantes e variar a largura dos pulsos, permitindo o controle da tensão média entregue para uma carga, e por consequência a potência, como uma luminária LED, ou a velocidade de um motor CC (BARR, 2001). Por exemplo: se a largura dos pulsos é definida como 50%, então a tensão média e a potência aplicada sobre a carga correspondem à metade do máximo permitido. Na Figura 17 é representado o conceito PWM para diferentes larguras de pulso.

Figura 17 – Representação gráfica do conceito PWM.



Fonte: BARR (2001).

## 2.5 MOSFET OPERANDO COMO CHAVE

Transistor de efeito de campo é um dispositivo eletrônico muito utilizado em circuitos, digitais e analógicos, para fins de amplificação ou chaveamento de sinais pelo fato de dissipar baixíssima potência. Sua operação é classificada em três diferentes regiões, conforme as tensões aplicadas sobre seus terminais: dreno ( $V_D$ ), fonte ( $V_S$ ), porta ( $V_G$ ) e substrato ( $V_B$ ). Normalmente, o terminal substrato é ligado com o terminal fonte (SEDRA; SMITH, 2007).

Na configuração como chave, o MOSFET deverá respeitar as condições de uma das seguintes regiões de operação: corte ou triodo. Para um MOSFET de canal n:

- **Região de corte**

Condição:

$$V_{GS} < V_t \quad (1)$$

onde:  $V_{GS}$  é tensão entre os terminais porta e fonte; e  $V_t$ , ou tensão limiar (*threshold*), corresponde ao mínimo valor de  $V_{GS}$  para a formação do canal n. Nessa região de operação, o transistor permanece inativo, e portanto, não há fluxo de corrente entre o dreno e a fonte. Caracteriza o estado desligado de chaveamento.

- **Região de triodo**

Condições:

$$V_{GS} > V_t \text{ e } V_{DS} < V_{GS} - V_t \quad (2)$$

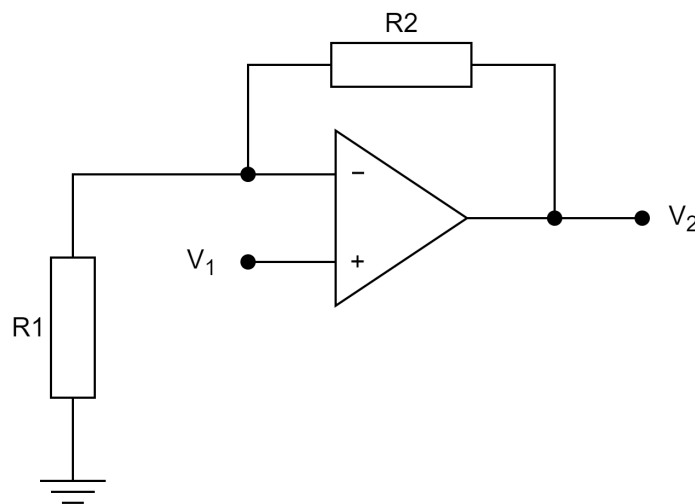
onde:  $V_{DS}$  é a tensão entre os terminais entre dreno e fonte. Nessa região de operação, o transistor é ativo. O canal n criado permite o fluxo de corrente entre o dreno e a fonte. Caracteriza o estado ligado de chaveamento.

## 2.6 AMPLIFICADOR OPERACIONAL NÃO INVERSOR

Corresponde à um circuito para amplificação de sinal, sem a mudança de fase, utilizando-se do amplificador operacional. Este dispositivo, com inúmeras aplicações nos campos da eletrônica, possui características próximas de um amplificador ideal, como: impedância de entrada infinita e impedância de saída nula. O ganho de tensão (G) desse circuito, como mostra a [Equação \(3\)](#), é determinado pelas resistências que compõem a realimentação positiva do amplificador operacional ([WENDLING, 2010a](#)). Na [Figura 18](#) está apresentado o esquemático de seu circuito.

$$G = \frac{V_2}{V_1} = \left(1 + \frac{R_2}{R_1}\right) \quad (3)$$

Figura 18 – Esquemático do circuito amplificador operacional não inversor.



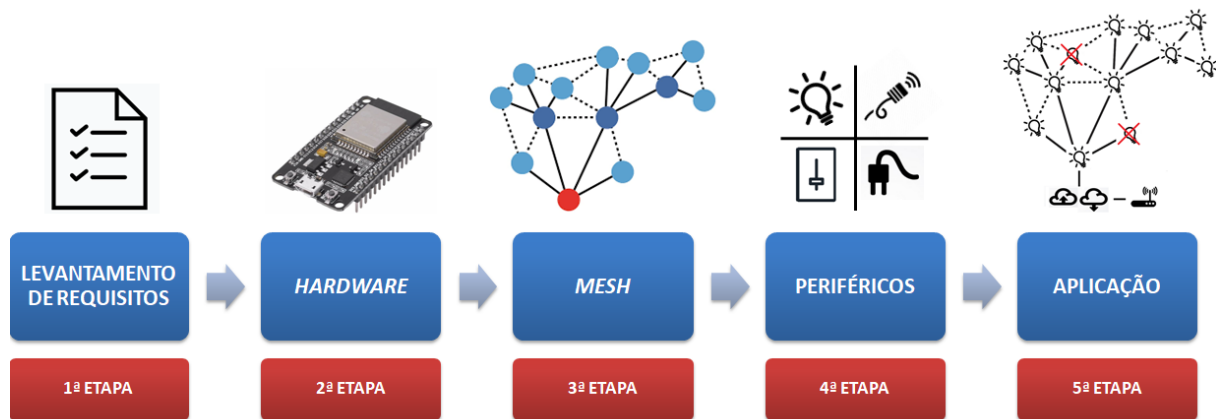
Fonte: Os autores (2019).

### 3 MATERIAIS E MÉTODOS

O foco deste trabalho é desenvolver um sistema eletrônico capaz de comunicar com seus semelhantes, utilizando o padrão de rede *wireless mesh*, com a finalidade de criar uma rede que permita a interconectividade com todos os ativos. Assim, em termos de aplicação, um ponto da rede (ou nó) poderá ter a capacidade de comunicar-se com um outro mesmo que esteja fora da sua área de cobertura de sinal, pois os nós vizinhos poderão retransmitir as transmissões até chegar ao seu verdadeiro destino. É claro, para isso acontecer deve existir, ao menos, nós com área de cobertura entre os locais de origem e de destino.

A execução deste trabalho está estruturado por um processo de cinco etapas, como pode ser visualizado na [Figura 19](#). Inicialmente, serão realizados respectivamente: uma pesquisa do funcionamento e da disposição dos sistemas de iluminação pública brasileiros; e um estudo teórico relacionado à infraestrutura e roteamento de redes *wireless mesh* (já apresentado na [Seção 2.2](#)). A partir das informações coletadas, serão levantados os requisitos como também a seleção de um *hardware* que permita a concepção do projeto.

Figura 19 – Diagrama de blocos para a execução do projeto.



Fonte: Os autores (2019).

Na segunda etapa será efetuada uma análise das principais características do *hardware* escolhido, verificando as suas especificações técnicas, além de comprovar a sua capacidade para a aplicação futura em sistemas de iluminação pública. Na terceira etapa será o desenvolvimento da rede de comunicação. Para isso, primeiramente, deverá ser compreendido o funcionamento do protocolo a ser utilizado e definido como a rede *mesh* deverá estar configurada visando o seu melhor desempenho. Após, será então criado um ambiente de desenvolvimento Linux para a edição e a compilação dos protocolos.

Na quarta etapa serão selecionados dispositivos eletrônicos de sensoriamento e de atuação que irão participar na rede criada, e simultaneamente, darão a aplicabilidade do projeto. Em seguida, serão projetados os circuitos auxiliares dos periféricos para a interface com o hardware escolhido, um processo que envolve o esquemático geral, o roteamento (placa de circuito impresso) e a calibração de medidas.

Na etapa de aplicação será integrado tudo que foi construído até então. Além disso, será criado um servidor *web* (*webserver*) para a comunicação com a rede *mesh*, e assim, poder concentrar todos os dados recebidos da aplicação como também enviar comandos para cada ponto da rede. Ao mesmo tempo, serão desenvolvidos as rotinas destinadas ao processamento de dados e confeccionadas as placas de circuito impresso da etapa anterior. Por fim, será realizado a montagem do sistema eletrônico para a fase de testes.

A seguir, nas Seções 3.1 a 3.5, serão detalhados os processos que foram realizados em cada uma das cinco etapas, assim como a descrição dos materiais utilizados para a consolidação deste trabalho.

### 3.1 LEVANTAMENTO DE REQUISITOS

A prestação de serviços de iluminação pública no Brasil, segundo a resolução normativa ANEEL (Agência Nacional de Energia Elétrica) número 414 - artigo 218, de 9 de setembro de 2010, é de competência dos órgãos que administram os municípios, ou seja, as prefeituras. A resolução ainda estabelece os deveres de proporcionar visibilidade para a segurança do tráfego de pedestres e veículos, proporcionado conforto e segurança. Todavia, não regulamenta padrões da disposição que os sistemas de iluminação pública devem cumprir, ficando a cargo de cada prefeitura, em conjunto com as concessionárias que administram o fornecimento de energia elétrica local, da distribuição dos pontos de iluminação em seus respectivos municípios.

Normalmente, o arranjo dos pontos de iluminação pública pode ser unilateral ou bilateral, dependendo do tipo da via urbana assim como da necessidade de iluminar a região (CEMIG, 2012). Em vias locais, também chamados de ruas, a distribuição é unilateral, em geral, e cada ponto distanciado entre 30 a 40 metros de seu semelhante. Este valor varia de acordo com a tecnologia e as especificações técnicas da lâmpadas utilizadas.

Sendo assim, para a viabilização deste trabalho, é importante estabelecer que o *hardware* escolhido para a construção da rede *wireless mesh* tenha a capacidade de comunicar com seu semelhante na faixa de distâncias entre 60 a 80 metros. Pois, por exemplo, em caso de uma falha de seu dispositivo, os seus vizinhos adjacentes poderão se comunicar, e conseqüentemente, manter o funcionamento da rede. Além disso, deve-se considerar que a arquitetura *mesh* tenha a capacidade de se adaptar conforme as características geográficas das instalações existentes em iluminação pública, além de suprir o tráfego de dados e ser tolerantes as falhas de comunicação.

Um outro requisito é o estabelecimento de um ponto concentrador na rede desenvolvida que irá realizar a troca de dados diretamente com um servidor externo, configurando um sistema de gerenciamento e controle da aplicação.

### 3.1.1 Requisitos do Projeto

- Escolha de um *hardware* disposto de:
  - Comunicação *wireless* com alcance entre 60 a 80 metros.
  - Baixo custo no mercado.
  - Baixo consumo de energia elétrica.
  - Capacidade de processamento.
  - Interface para periféricos.
- Desenvolvimento de uma rede *wireless mesh* com um nó concentrador destinado para a conexão com um roteador, permitindo a troca de dados com um servidor externo.
- Concepção de um servidor externo para o gerenciamento e controle da aplicação.

### 3.1.2 Seleção do *Hardware*

Após várias pesquisas no mercado, avaliando as atuais tecnologias existentes, chegaram-se em três opções viáveis para o projeto e que cumprem os requisitos estabelecidos anteriormente (na [Subseção 3.1.1](#)). Para efetuar a melhor escolha do *hardware* entre os candidatos, foi construído um quadro comparativo ([Quadro 1](#)), avaliando os critérios estabelecidos de acordo com as grandezas que os candidatos mutuamente possuem. As cores de sombreamento realçam o nível de significância, indicando: verde, acima do desejável; laranja, desejável; e vermelho, aceitável.

Quadro 1 – Quadro comparativo entre as opções de *hardware* para o projeto.

	ESP32 (WROOM32)	NRF24L01	XBEE PRO
<b>Alcance de comunicação <i>wireless</i></b>	Moderado	Alto	Alto
<b>Consumo de energia elétrica</b>	Moderado	Moderado	Baixo
<b>Taxa de transmissão</b>	Alto	Alto	Baixo
<b>Facilidade de uso / Integração</b>	Alto	Baixo	Baixo
<b>Preço</b>	Baixo	Moderado	Alto

Fonte: Os autores (2019).

Como pode-se notar, o *hardware* ESP32 WROOM-32 foi o que mais se destacou positivamente em relação aos outros dois candidatos. E portanto, é o escolhido para a concepção deste trabalho.

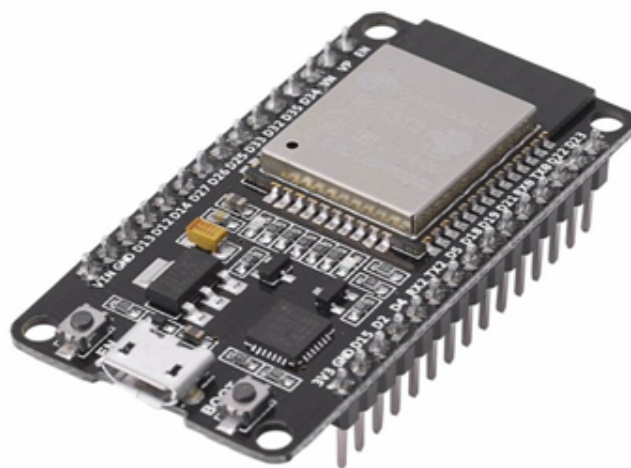


## 3.2 HARDWARE

O ESP32 WROOM32 é um módulo eletrônico desenvolvido e fabricado pela empresa chinesa *Espressif Systems*, destinado para ser capaz de atender as especificações de projetos relacionados a IoT. Lançado oficialmente no ano de 2016, possui um microcontrolador de alta performance e baixo consumo de energia elétrica, suporte Wi-Fi, Bluetooth e Ethernet em um único encapsulamento.

Considerado acessível pelo seu baixo custo, o módulo possui dimensões pequenas (18 mm x 25.5 mm x 3.1 mm), o que torna interessante para o desenvolvimento de projetos comerciais em larga escala. Na [Figura 20](#) está apresentada o ESP32 WROOM32 DevKit v1, uma plataforma *open source* com suporte ao desenvolvimento do microcontrolador ESP32.

Figura 20 – ESP32 WROOM32 DevKit v1.



Fonte: ESPRESSIF SYSTEMS (2019).

### 3.2.1 Principais Características

- Processador: Xtensa Dual-Core 32-Bit LX36.
- Clock: 80 a 240 MHz (ajustável).
- Memória ROM: 448 KB.
- Memória RAM: 520 kB.
- Memória Flash: 4 MB.
- Comunicação *wireless*:
  - Wi-Fi: 802.11 b/g/n em 2,4 a 2,5 GHz.
  - Bluetooth: v4.2 BR / EDR e BLE.

- Interfaces:
  - 12-bit SAR-ADC.
  - 8-bit DAC.
  - PWM.
  - SPI / I2C / I2S / UART / CAN.

### 3.2.2 Especificações Técnicas

As especificações técnicas do *hardware* escolhido, o ESP32, estão disponíveis na [Tabela 1](#). Estas informações, retiradas do *datasheet* fornecido pelo fabricante, são essenciais para compreender a capacidade de sua utilização neste trabalho ([SYSTEMS, 2019a](#)).

Tabela 1 – Especificações técnicas do ESP32.

Parâmetro	Valor	Unidade
Tensão de alimentação de entrada	7 ~ 12	V
Tensão de alimentação operacional - microcontrolador	2,3 ~ 3,6	V
Corrente fornecida por fonte de alimentação externa	0,5	A
Temperatura de operação	-40 ~ 125	°C
Wi-Fi - Tx 802.11b, 1 Mbps, POUT = + 19,5 dBm	240	mA
Wi-Fi - Tx 802.11g, 54 Mbps, POUT = + 16,5 dBm	190	mA
Wi-Fi - Tx 802.11n, MCS7, POUT = + 14 dBm	180	mA
Wi-Fi - Rx 802.11b, -98 dBm	95 ~ 100	mA
Wi-Fi - Rx 802.11g, -75 dBm	95 ~ 100	mA
Wi-Fi - Rx 802.11n, MCS7, -70 dBm	95 ~ 100	mA
Bluetooth - Tx, POUT = 0 dBm	130	mA
Bluetooth - Rx, -94 dBm	95 ~ 100	mA

Fonte: ESPRESSIF SYSTEMS (2019).

Embora o ESP32 tenha recursos bastante atrativos, é importante ponderar o seu consumo de energia elétrica em pleno funcionamento, pois ao se utilizar baterias como forma de alimentação, certamente, haverá necessidade de intervenção em vários momentos para a manutenção. Entretanto, o módulo eletrônico possui modos que desabilitam os recursos não utilizáveis, e assim, proporcionar uma redução significativa no consumo.

Em uma aplicação futura em sistemas de iluminação pública é interessante fazer o proveito da mesma rede elétrica que alimenta os postes, para a alimentação do módulo eletrônico. O que exclui a necessidade do uso de baterias e torna o trabalho mais economicamente viável e atrativo.

Quanto às opções de comunicação *wireless* disponíveis, pode-se descartar de imediato o padrão Bluetooth para a viabilização deste trabalho. Pois, apesar de ser bastante atrativo em função do seu baixo consumo de energia comparado ao Wi-Fi, não possui uma potência de transmissão (dBm) suficiente para que o sinal alcance as distâncias

mínimas estabelecidas (60 a 80 metros). Em contrapartida, o padrão Wi-Fi satisfaz, uma vez que o próprio fabricante do ESP32 confirma a possibilidade do uso deste padrão nas distâncias consideradas.

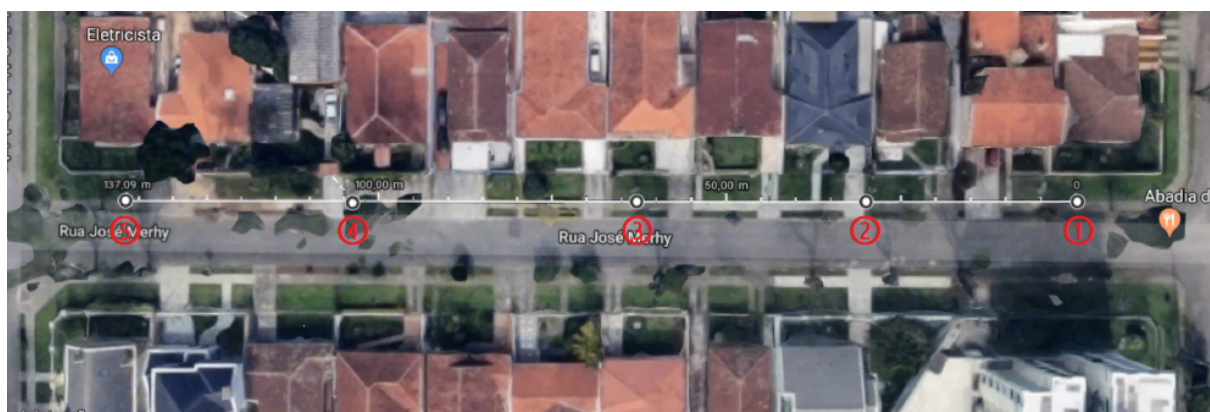
### 3.2.3 Teste de Alcançabilidade da Comunicação Wi-Fi

Para comprovar o alcance da comunicação Wi-Fi, foi realizado um teste prático utilizando dois exemplares do ESP32 em uma via local (rua). A aplicação consistiu fazer com que um módulo, denominado estação, criasse uma rede Wi-Fi para que o outro, ou o cliente, possa se conectar e efetuar o envio de um caractere aleatório. Assim que recebido a mensagem, a estação confirma ao cliente e ambos acionarão um respectivo LED do módulo, emitindo uma forma de luz intermitente para indicar a ocorrência da comunicação *wireless*. Agora, caso o cliente esteja fora da área de cobertura do sinal ou perca a conectividade com a estação, ambos irão desabilitar a atuação do LED.

O teste foi realizado sob as seguintes condições:

- Ambos os módulos estavam sob uma altura de um metro em relação ao chão, simulando um pior caso de desvanecimento do sinal Wi-Fi. Todavia, entre a estação e o cliente havia um livre caminho local, ou seja, uma linha de visada.
- O módulo estação foi fixado na posição nº 1, como mostra a [Figura 21](#), onde situa-se o primeiro poste de iluminação do quarteirão.
- O módulo cliente foi posicionado inicialmente na posição nº 2, seguidos na posição nº 3, 4 e 5. Em cada um dos pontos, onde encontra-se um poste de iluminação, foi verificada a atuação do LED, confirmando ou não a conectividade entre os módulos.

Figura 21 – Vista superior do local para o teste de alcançabilidade da comunicação Wi-Fi.



Fonte: Os autores (2019).

A distância entre os pontos nº 1 e nº 5 totaliza aproximadamente 137 metros, compreendendo uma distância média de 34,25 metros entre os postes. Estas medidas foram obtidas com auxílio do *Google Maps*. Para acompanhar o teste, foi utilizado um aplicativo

Android chamado Wi-Fi *Analyzer* de modo quantificar a potência do sinal recebida da estação nos pontos situados pelo cliente. Os valores resultantes encontram-se na [Tabela 2](#).

Entre os pontos (1 - 2) e (1 - 3) foi verificado a comunicação íntegra entre a estação e o cliente, através da constante intermitência da luz emitida pelo LED indicador. Entre os pontos (1 - 4) notou-se a perda da conectividade em alguns momentos. Já entre os pontos (1 - 5), foi constatado que o cliente não estava mais na área de cobertura do sinal.

Tabela 2 – Resultados obtidos do teste de alcançabilidade da comunicação Wi-Fi.

Posição da estação	Posição do cliente	Potência medida [dBm]
1	2	-60 ~ -70
1	3	-70 ~ -80
1	4	-80 ~ -95

Fonte: Os autores (2019).

Diante dos resultados, comprova-se que o ESP32 WROOM 32 DevKit v1 tem um alcance para a comunicação Wi-Fi dentro dos critérios estabelecidos (na [Subseção 3.1.1](#)).

### 3.3 MESH

A rede *mesh* utilizada no projeto foi a ESP-MESH, que pode ser encontrada no *Github* do *Espressif Systems* com o nome ESP-MDF. É um código aberto desenvolvido pelos engenheiros da *Espressif Systems* e voltado para o ESP32 que implementa todas as funcionalidade do ESP-MESH descritas na [Seção 2.2](#). A implementação do ESP-MESH é feita com auxílio de um outro *framework* também desenvolvido pelos engenheiros da *Espressif Systems*, o ESP-IDF, que é voltado para aplicações IoT utilizando o ESP32. Ambos são implementados utilizando o sistema operacional *FreeRTOS*.

Além da rede ESP-MESH, algumas outras funcionalidades que seriam úteis para este projeto já estão implementadas no ESP-MDF, logo estas funções foram reutilizadas. Uma dessas funções é a de criação de *sockets* para comunicação com um servidor externo (somente o nó raiz utiliza esta função). Já as funções que cuidam da transmissão e recepção de dados também foram reutilizadas, mas precisaram ser adaptadas ao projeto. À função de recebimento foi adicionada um decodificador dos comandos recebidos do servidor. A função de transmissão, que antes era um tarefa executada em paralelo com todo o resto do programa, tornou-se uma função que só é executada após a coleta dos dados dos sensores.

O sistema operacional utilizado para o desenvolvimento do código foi o *Kali Linux* e antes de iniciar o desenvolvimento é preciso instalar algumas bibliotecas adicionais ao sistema, pois são necessárias para a compilação dos códigos do ESP-MESH. As bibliotecas instaladas foram: *gcc*, *git*, *wget*, *make*, *libncurses-dev*, *flex*, *bison*, *gperf*, *python*, *python-pip*,

*python-setuptools*, *python-serial*, *python-cryptography*, *python-future*, *python-pyparsing*. O compilador utilizado foi o *xtensa-esp32* (versão 1.22.0-80-g6c4433a-5.2.0).

A configuração de parâmetros da rede *mesh* (além daquelas referentes a conexão com roteador e um com servidor externo) é feita antes da compilação dos códigos usando o comando *make menuconfig* no terminal linux. A compilação do código é feita utilizando o comando *make* e a gravação no ESP32 é feita através do comando *make flash*.

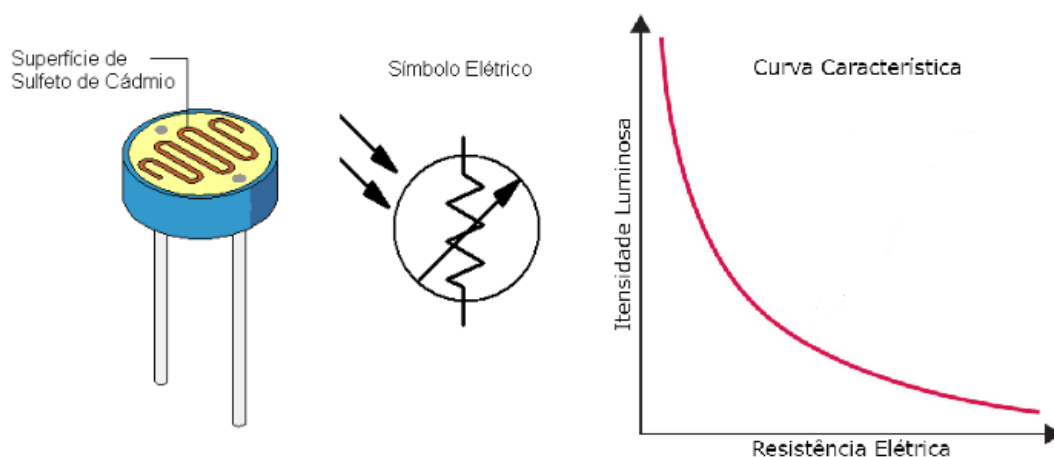
## 3.4 PERIFÉRICOS

### 3.4.1 Sensores Escolhidos

#### 3.4.1.1 Luminosidade - LDR

O LDR, acrônimo de *Light Dependent Resistor*, é um dispositivo semiconductor de Sulfeto de Cádmio (CdS) não polarizável com dois terminais. Classificado como um sensor fotoelétrico, possui o comportamento de variar sua resistência elétrica conforme a quantidade de luz incidente sobre a sua superfície (WENDLING, 2010b). Todavia, a curva característica (Figura 22) mostra que a relação é não-linear, na qual a resistência diminui à medida que a intensidade da luz aumenta.

Figura 22 – Ilustração, símbolo elétrico e curva característica do LDR.



Fonte: Adaptado WENDLING (2010).

Este dispositivo possui uma sensibilidade que se aproxima do olho humana, o que permite a operação com fontes de luz convencionais de luz, como a luz ambiente. A única desvantagem está na sua velocidade de resposta, não operando corretamente em altas frequências, o que não será problema para a realização deste trabalho.

### 3.4.1.2 Corrente - Resistor *Shunt*

Consiste em um resistor de precisão utilizado para medir a corrente elétrica que circula em uma determinada carga posicionada em série, através da mensuração da queda de tensão sobre a sua resistência elétrica. Assim, pela lei de Ohm, conhecidos os valores de tensão e resistência, pode-se determinar o valor da corrente.

Trata-se de um dispositivo invasivo, e portanto, deve possuir um valor de resistência elétrica baixíssima de modo não afetar o funcionamento da carga. Consequentemente, a queda de tensão medida sobre o resistor *shunt* também é baixa, necessitando de um estágio de amplificação para fazer a leitura de sua grandeza.

### 3.4.1.3 Temperatura e Umidade - DHT22

O DHT22, apresentado na [Figura 23](#), é um dispositivo eletrônico destinado para o sensoriamento de temperatura entre  $-40\text{ }^{\circ}\text{C}$  a  $80\text{ }^{\circ}\text{C}$  e de umidade entre 0 a 100%, com precisão de  $0,5^{\circ}\text{C}$  e 2% respectivamente. Possui uma interface para a comunicação de dados direta e compatível com o hardware escolhido e um tempo de resposta de 2 segundos. Formado por um sensor de umidade capacitivo e um termistor, a sua alimentação pode ser efetuada diretamente do ESP32, através do pino de saída 3,3 V ([AOSONG, 2019](#)).

Figura 23 – DHT22.



Fonte: Adaptado AOSONG (2019).

### 3.4.2 Atuação Escolhida - LED

O LED, acrônimo de *Light Emitting Diode*, é um componente semiconductor polarizável que tem a propriedade de transformar a tensão elétrica, aplicada corretamente em seus terminais, em uma forma de emissão de luz. Atualmente, é muito utilizável em aplicações do cotidiano devido ao seu baixo consumo de energia elétrica. Neste trabalho, foram usados LEDs de alto brilho (14000 mcd) com tensão nominal de polarização de 3 V e corrente máxima de 20 mA - especificações do fabricante.

### 3.4.3 Fonte de Alimentação Escolhida - Associação de Pilhas AA

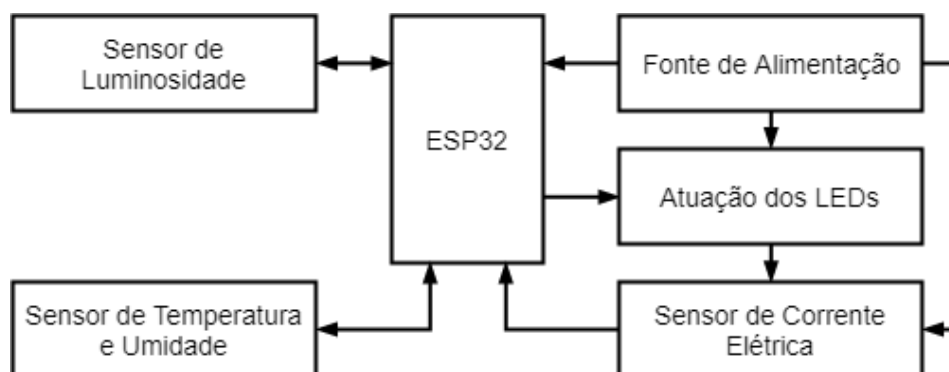
Para promover a alimentação dos dispositivos presentes no projeto, e respeitando as especificações técnicas de cada componente, foi utilizado uma associação em série de quatro pilhas alcalinas AA de tensão nominal de 1,5 V, totalizando uma diferença de potencial de 6 V. Optou-se pelo uso desse tipo de pilha devido à sua capacidade para o fornecimento de corrente elétrica (valores entre 2100 a 2800 mAh), garantindo o pleno funcionamento do hardware e dos periféricos durante a fase de testes da aplicação, que terão algumas horas de duração.

Por via das dúvidas, realizou-se um teste prático para verificar quanto tempo a associação conseguia manter o suporte de alimentação ao projeto. E os resultados foram bem satisfatórios, com um rendimento acima de 10 horas, o que é suficiente para viabilizar os testes da aplicação.

### 3.4.4 Circuitos de Condicionamento de Sinais

O projeto dos circuitos auxiliares, visando a interface do hardware com os periféricos, pode ser estruturado através de um diagrama de blocos (apresentado na [Figura 24](#)) para a melhor compreensão do leitor. Cada bloco constitui de circuitos distintos que serão desenvolvidos a seguir.

Figura 24 – Diagrama de blocos para o projeto dos circuitos auxiliares.



Fonte: Os autores (2019).

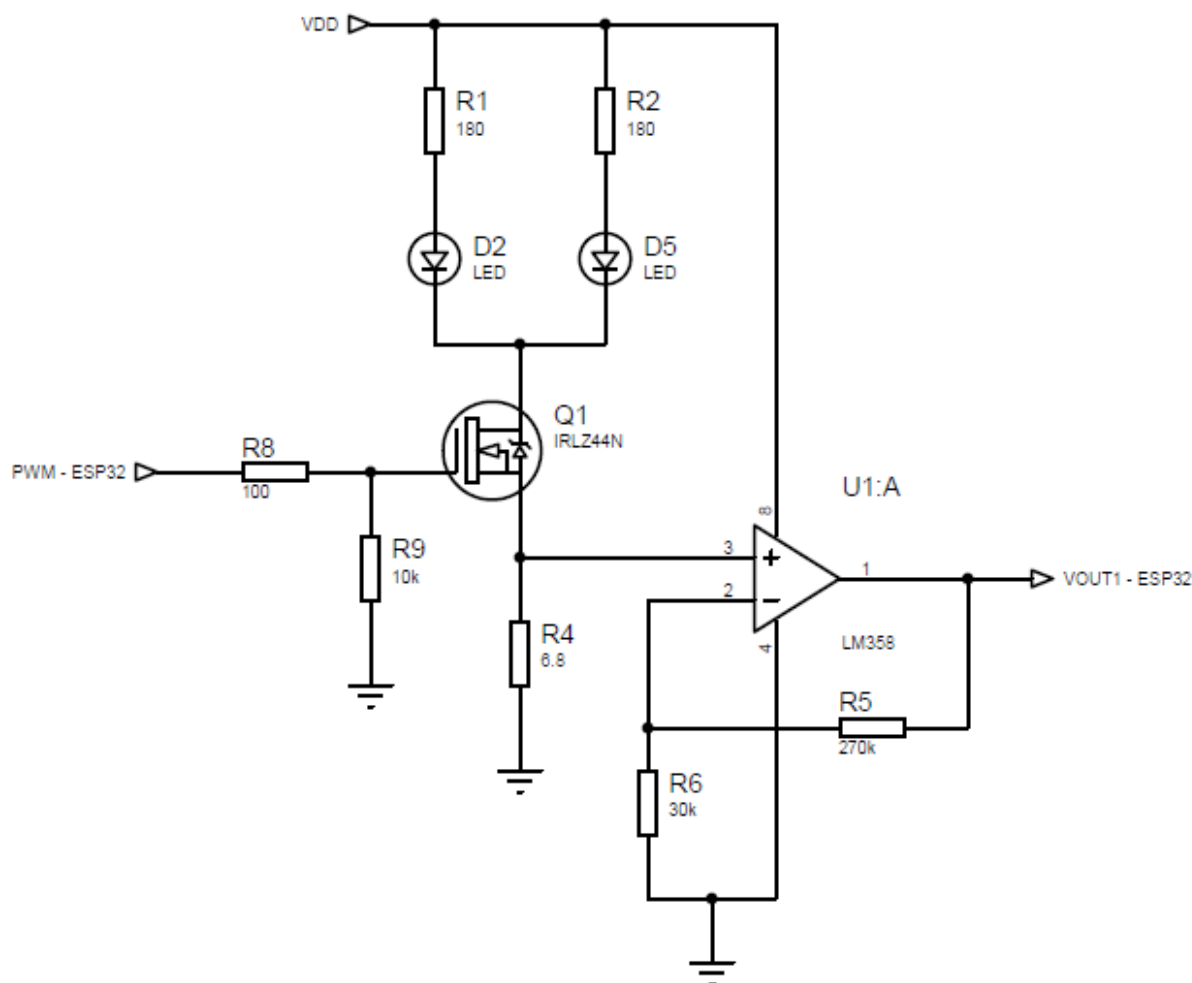
#### 3.4.4.1 Atuação dos LEDs e Sensor de Corrente Elétrica

Fazendo uso dos conceitos apresentados na [Seção 2.5](#), projetou-se um circuito de potência que acionará a carga, no caso dois LEDs, através do sinal de chaveamento PWM gerado pelo hardware (0 ou 3,3 V). O MOSFET utilizado possui uma tensão limiar situado entre 1 e 2 V, dentro das condições para permitir a entrada do resistor *shunt* no terminal

fonte do IRLZ44N. Assim, pode-se indiretamente mensurar a corrente elétrica que está sendo consumida pelos LEDs. Todavia, antes é necessário um estágio de amplificação.

O circuito de amplificação operacional não inversora, revisado na [Seção 2.6](#), foi projetada com um ganho constante de 10, visando que o valor da queda de tensão sobre o resistor *shunt* seja legível e dentro dos limites permitidos pelo ADC do ESP32 (0 a 3,3 V).

Figura 25 – Circuitos de atuação dos LEDs e sensoriamento de corrente elétrica.



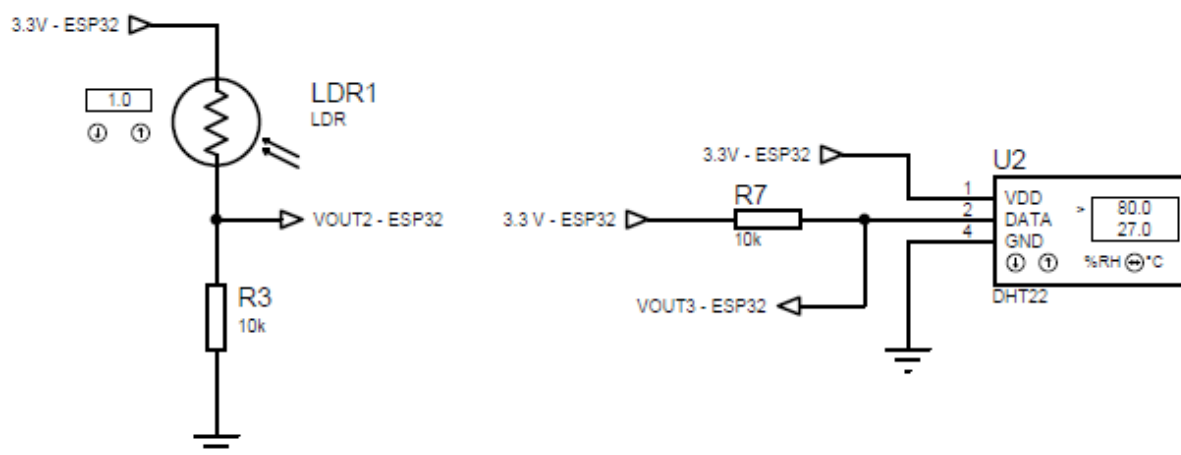
Fonte: Os autores (2019).

#### 3.4.4.2 Sensor de Luminosidade e Temperatura/Umididade

Os circuitos de condicionamento para o LDR e o DHT22 constituíram-se de um divisor de tensão e um resistor de *pull-up*, respectivamente. Para o primeiro caso, a disposição de seu circuito permitiu que a tensão lida pelo ADC do hardware varie diretamente de acordo com a quantidade de luz incidente sobre a superfície do LDR.



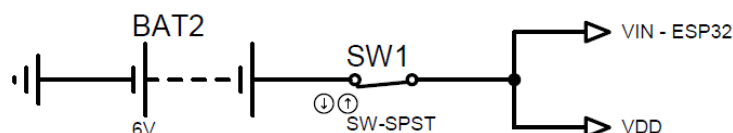
Figura 26 – Circuitos de sensoriamento de luminosidade e temperatura/umidade.



Fonte: Os autores (2019).

#### 3.4.4.3 Fonte de Alimentação

O circuito de condicionamento para a alimentação consistiu apenas na inserção de um interruptor para fazer o acionamento ou não do sistema eletrônico projetado.

Figura 27 – Circuito para alimentação do *hardware* e dos periféricos.

Fonte: Os autores (2019).

#### 3.4.5 Calibração das medidas

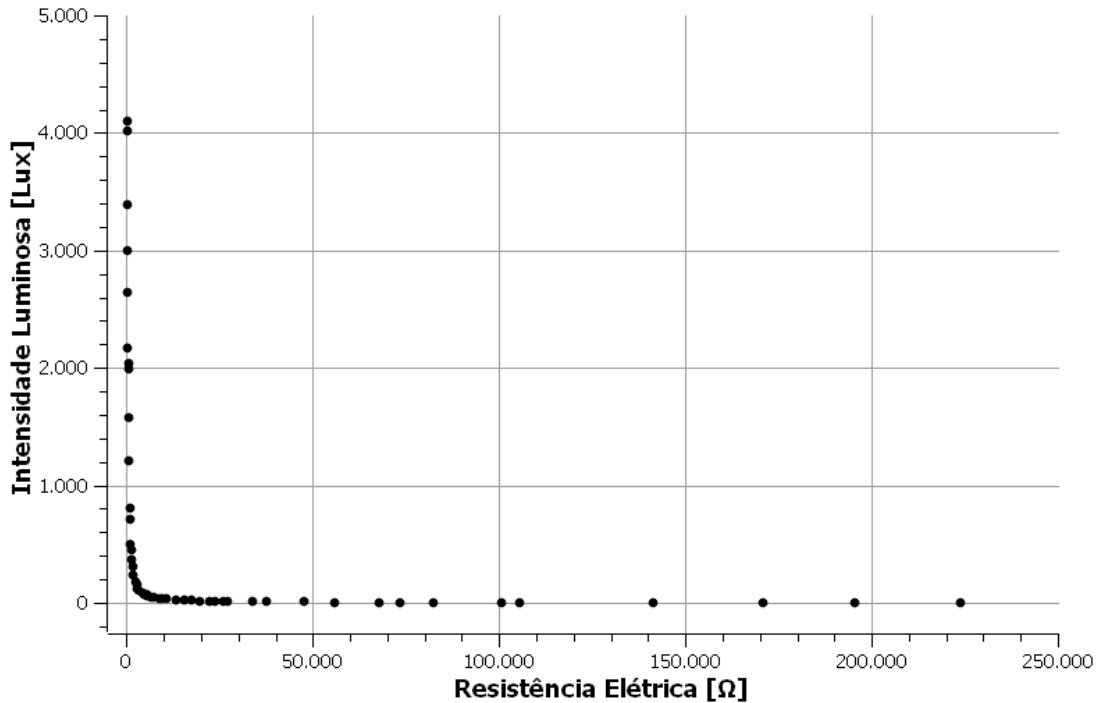
Como em qualquer projeto, o processo de calibração é imprescindível para que os valores mensurados tenham as menores variações possíveis e dentro de uma faixa de incertezas, além de padronizar os resultados de medições. Para este trabalho, o dispositivo LDR e o ADC do *hardware* foram os necessitados.

##### 3.4.5.1 LDR

A calibração do LDR resultou de um processo de três etapas. Primeiramente, foi necessário realizar uma amostragem de 50 dados referentes ao valor da resistência elétrica juntamente da quantidade de luz incidente sobre a sua superfície. Para isso, foram utilizados um multímetro e um luxímetro, respectivamente. Em seguida, com auxílio do *software SciDavis*, plotou-se o gráfico da intensidade luminosa em função da resistência,

visualizado na [Figura 28](#).

Figura 28 – Gráfico da curva característica do LDR.



Fonte: Os autores (2019).

Como esperado, a curva característica se assemelhou ao modelo representado na [Figura 22](#), com um comportamento exponencial. Todavia, por padrão, é mais interessante obter uma equação de 1º grau entre as grandezas de modo facilitar a análise de dados. Assim, na [Equação \(4\)](#) está exibido a relação que lineariza o comportamento do LDR, sendo:  $I_L$  a intensidade luminosa,  $R_{LDR}$  a resistência elétrica do LDR, M o coeficiente angular e B o coeficiente linear.

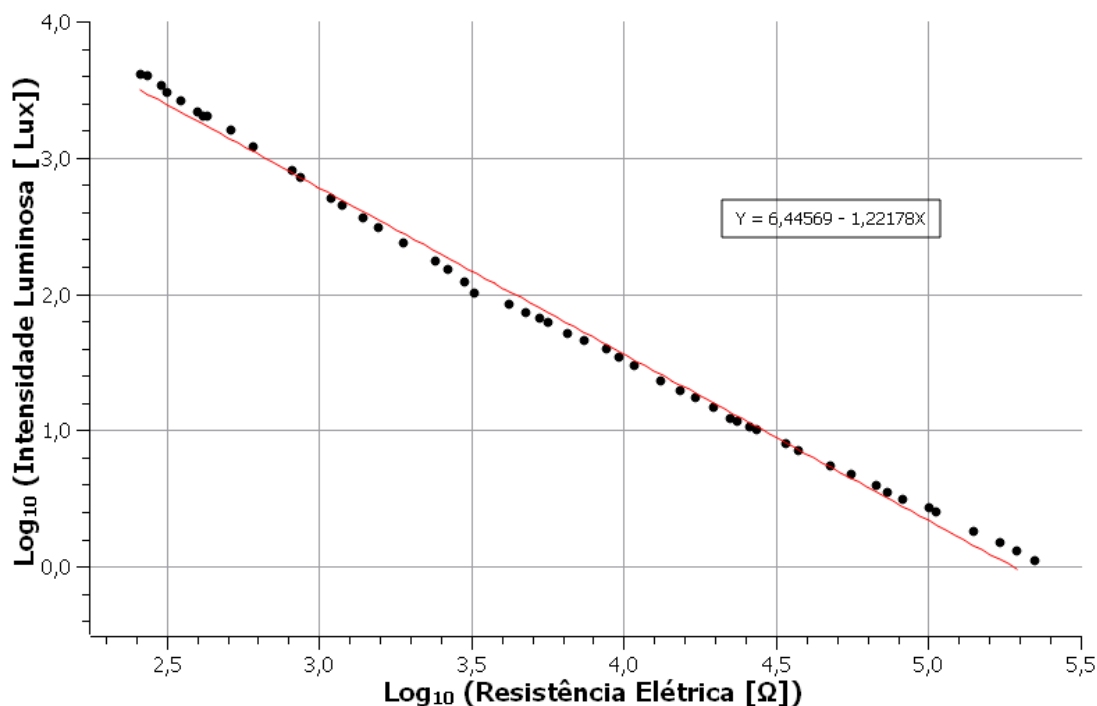
$$I_L [Lux] = R_{LDR}^M \cdot 10^B [\Omega] \quad (4)$$

A última etapa consistiu em linearizar o gráfico inicial com aplicação do logaritmo em cada eixo. Utilizando o método de regressão linear simples, foi determinado a equação de reta que melhor descreve o novo comportamento do gráfico apresentado na [Figura 29](#). E uma vez conhecidos os valores dos coeficientes M e B (-1,22178 e 6,44569, respectivamente) pode-se então definir a equação de calibração do LDR.

#### 3.4.5.2 ADC do ESP32

Devido a variações no processo de fabricação do ESP32, a tensão de referência do conversor que idealmente seria de 1,1 V acaba variando de 1 V a 1,2 V entre diferentes dispositivos. Para contornar este problema é gravado em uma posição específica da memória

Figura 29 – Gráfico linearizado da curva característica do LDR.



Fonte: Os autores (2019).

do ESP32 o valor real da tensão de referência, valor este que deve ser utilizado para os cálculos após o processo de conversão.

Com uma tensão e referência de 1,1 V, tensões de entrada no ADC maiores do que 1,1 V fazem com que o conversor sature, distorcendo o sinal amostrado. Como neste projeto os sinais analógicos podem excursionar para um valor de tensão maior do que 1,1 V, isso acaba tornando-se um problema. Porém, o ESP32 possui um modo de atenuação sinais de entrada e isso faz com que os sinais fiquem dentro do limite de 1,1 V do conversor. Existem valores pre-definidos para esta atenuação e neste projeto o valor utilizado foi de 11 dB. Utilizando esta atenuação, o limite para o valor de entrada do conversor analógico digital se torna o próprio limite de entrada dos pinos do ESP32, que é de 3,3 V.

Tanto a variação da tensão de referência quando a atenuação do sinal de entrada influenciam no valor final lido pelo ADC. Por isso, ao final da amostragem, uma calibração deve ser aplicada ao ADC e esse processo de atenuação deve ser revertido digitalmente. Para tanto, características do ADC são extraídas (levando em consideração o valor real da tensão de referência e atenuação) e são utilizadas, junto com o valor digital amostrado, para chegar-se ao valor real da tensão na entrada do ADC. Uma função implementada no *ESP-IDF* usa como argumentos de entrada o valor amostrado e as características extraídas do ADC e retorna o valor amostrado em escala de mili volts (*esp\_adc\_cal\_raw\_to\_voltage()*).

### 3.5 APLICAÇÃO

O código de aplicação deste projeto pode ser dividido em duas partes principais. A primeira delas é a programação dos dispositivos ESP32, que formarão a rede *mesh*. O código para os dispositivos é programado utilizando a linguagem C. A segunda é o programa servidor, que roda em um computador e simula um sistema final que recebe os dados de cada um dos nós da rede, salva-os em um arquivo externo e pode enviar comandos para os nós. O servidor foi programado utilizando a linguagem de programação *Python*.

#### 3.5.1 Programação do ESP32

Uma vez que a rede *mesh* já está implementada (ESP-MESH) a programação do ESP32 se resume aos seguintes itens:

- Configuração do PWM.
- Configuração dos conversores analógicos digitais.
- Decodificador de comandos recebidos do servidor.
- Função local responsável por alterar a intensidade luminosa dos LEDs
- Função responsável pela coleta dos dados dos sensores e envio para o servidor.

O PWM neste projeto é responsável por definir a intensidade luminosa dos LEDs. Neste caso, os LEDs são usados para mostrar a possibilidade de controle de intensidade luminosa oferecida pela projeto, seja esse controle feito por uma função local no dispositivo ou um comando vindo de um servidor externo. As configurações utilizadas no PWM foram:

- Frequência do PWM igual a 5000 Hz.
- Resolução de 13 bits. Assim o ciclo de trabalho pode variar de 0 a 8191 ( $2^{13} - 1$ ).
- Utilizado o Canal 0 do Timer 0.
- Utilizado o *GPIO 27*.

Os conversores analógico digital neste projeto são responsáveis pela coleta de dados dos sensores. Um deles é o LDR, responsável por captar a intensidade luminosa do ambiente e outro é um resistor *shunt*, responsável por medir o consumo de corrente elétrica nos LEDs. Para ambos os casos foi utilizado o *ADC1* do ESP32, porém utilizando canais diferentes para cada um dos sensores. As configurações utilizadas foram:

- Resolução de 12 bits. Assim o valor amostrado pode variar de 0 a 4095 ( $2^{12} - 1$ ).
- O canal utilizado para o Resistor *shunt* foi o 0 (*GPIO 36*) e para o LDR foi o canal 6 (*GPIO 34*).
- A atenuação para ambos os canais foi de 11 dB.

A leitura do valor da intensidade luminosa do ambiente começa com a amostragem do valor da tensão sobre o resistor conectado em série com o LDR ( $10\text{ k}\Omega$ ). Em seguida, o valor amostrado é processado pela função que aplica a calibração e reverte a atenuação utilizada. Ao final, tem-se o valor de tensão sobre o resistor de  $10\text{ k}\Omega$  e é possível encontrar

a resistência do LDR a partir da [Equação \(5\)](#).

$$R_{LDR} = \frac{3,3 - V_{10k}}{V_{10k}} \cdot 10k \quad (5)$$

Com o valor da resistência do LDR conhecida, pode-se através da [Equação \(4\)](#) encontrar a intensidade luminosa medida pelo LDR em unidade lux.

Já a amostragem do sinal do resistor *shunt* requer uma maior atenção. Isso por que o sinal amostrado é o de um PWM (o mesmo do acionamento dos LEDs). Assim, em dado momento o valor amostrado terá valor zero e em outros o valor máximo. Por isso, quando chega o momento de fazer a amostragem do consumo de corrente nos LEDs, é feita um número de amostras igual a 250 e destas o maior valor amostrado é usado para o resto dos cálculos. Escolher o maior valor evita que um valor igual a zero seja usado nos cálculos seguintes. O valor amostrado é processado pela função que aplica a calibração e reverte a atenuação e a soma das correntes nos LEDs pode ser encontrado a partir da [Equação \(6\)](#), sendo:  $V_{Max}$  o valor máximo amostrado pelo ADC e  $CT$  o ciclo de trabalho do PWM.

$$I_{LED} = \frac{V_{Max}}{68} \cdot CT \quad (6)$$

Existem duas formas de alterar a intensidade luminosa dos LEDs. Uma delas é baseada no valor amostrado pelo LDR e a outra é a partir de comandos recebidos do servidor. No código, uma *flag* serve para indicar se o valor do PWM este sendo definido por um controle local ou por comandos do servidor.

Assim que o valor do LDR é amostrado o valor do PWM que controla os LEDs é alterado baseado na seguinte configuração (desde que a flag esteja definida para processamento local do PWM):

- Luminosidade menor do que 20 lux, então PWM em 100%.
- Luminosidade entre 20 e 60 lux, então PWM em 75%.
- Luminosidade entre 60 e 120 lux, então PWM em 50%.
- Luminosidade entre 120 e 180 lux, então PWM em 25%.
- Caso contrário, PWM desligado.

Para alteração do PWM a partir de comandos recebidos do servidor, os valores que que interpretados como válidos para controle do PWM estão entre 0 e 8191. Uma vez enviado um comando para alteração do valor do PWM pelo servidor o sistema não volta mais para o controle local de intensidade luminosa dos LEDs, a menos que um comando específico seja enviado pelo servidor.

A função que faz a decodificação dos dados recebidos do servidor é a responsável por interpretar os comandos recebidos e tomar a decisão adequada. Como o valor máximo que pode ser recebido do servidor para a alteração do PWM é 8191, a decodificação foi estruturada da seguinte forma:

- Menor do que 0: Ignora o comando.
- Entre 0 e 8191: Altera o PWM para o valor recebido.

- Igual a 8192: Faz com que o PWM seja definido pela função local de controle.
- Igual a 8193: Faz com que o nó envie imediatamente uma mensagem de "ping" para o servidor (é utilizada para medição de atraso de ida e volta de pacotes).

A última função a ser desenvolvida foi a de coleta de dados. Como o código está implementado usando um sistema operacional, esta função foi criada como uma tarefa que roda a cada nove segundos. O que a função faz é a coleta dos dados dos ADCs e, caso a *flag* esteja definida para controle local, chama a função que faz o controle local do PWM. Depois de coletados os dados, eles são imediatamente enviados para o servidor. Os dados seguem o seguinte padrão:

- Etiqueta - Cada ESP32 tem um etiqueta que o identifica unicamente.
- MAC - Seu endereço MAC. Endereço destino para o qual se envia comandos a partir do servidor.
- Camada na Rede - Em qual camada o nó está presente.
- Nó Pai - Para saber quem é seu nó pai e assim seja possível estruturar as conexões formadas pela rede.
- RSSI - RSSI com relação ao nó pai.
- Luminosidade - Valor da luminosidade local.
- Consumo - Valor do consumo de corrente nos LEDs.
- Temperatura - Valor da temperatura local.
- Umidade - Valor da umidade local.

#### 3.5.1.1 Programação do Servidor

O servidor é um programa feito para receber as informações da rede, armazená-los em um arquivo externo e para ser possível a partir dele enviar comandos para cada um dos nós da rede.

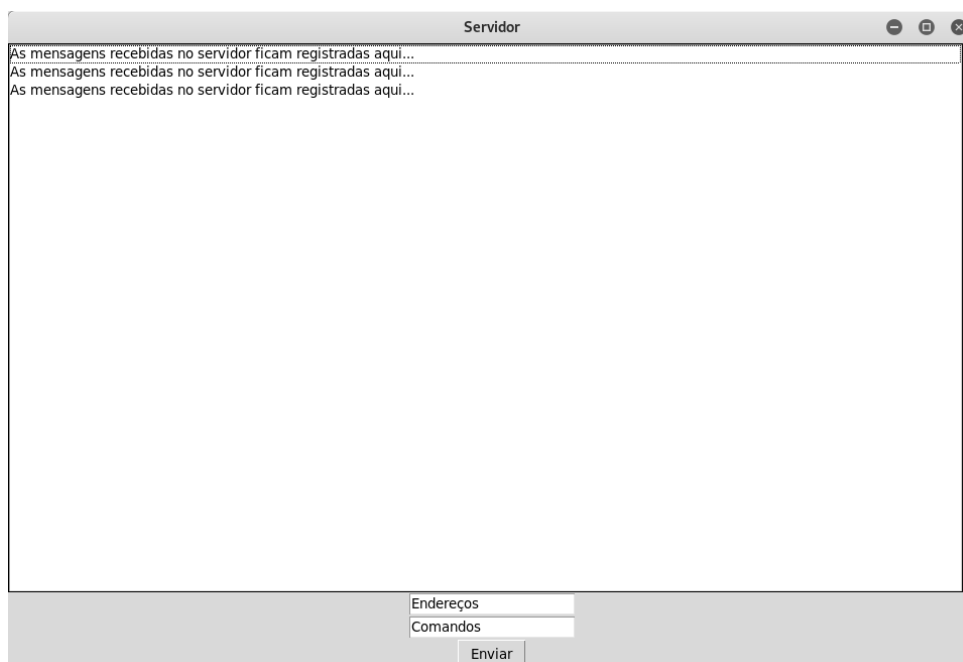
Para receber as informações dos nós, foi feita uma programação *socket* que cria um servidor em um endereço de IP local (neste caso 192.168.0.3) na porta 5000. Assim que o programa se inicia o *socket* é criado e fica esperando pela conexão do nó raiz da rede ESP-MESH.

Foi criada uma interface gráfica responsável por mostrar para o usuário os dados recebidos de cada um dos nós e também poder interagir com cada um deles através de comandos. A interface gráfica que foi programada utilizando a biblioteca *tkinter* é ilustrada na [Figura 30](#).

A interação do usuário com o servidor é feita através dos seguintes itens:

- Lista das mensagens onde ficam os dados recebidos dos nós.
- Campo de mensagem para endereço do ESP32 destino. Nele é escrito o endereço MAC do nó destino (Exemplo de endereço MAC: "a4:cf:12:75:00:20". Caso o endereço seja "ff:ff:ff:ff:ff:ff", todos os nós da rede receberão o comando).
- Campo de mensagem para comandos. Nele é escrito o comando a ser enviado para o

Figura 30 – Interface gráfica do servidor.



Fonte: Os autores (2019).

nó destino. Estes comandos podem ser:

- Valores entre  $0.0$  e  $100.0$ . Comando para alteração do PWM. É feito um cálculo no servidor que define qual deve ser o valor enviado para o nó de tal forma que o ciclo de trabalho do PWM seja igual a porcentagem definida pelo usuário. O valor final enviado do servidor para o nó ( $V_E$ ) é calculado através da [Equação \(7\)](#), sendo  $V_A$  o valor do PWM a ser enviado para o nó.

$$V_E = \frac{V_A}{100} \cdot (2^{13} - 1) \quad (7)$$

- *Local*. Comando enviado para nó que faz com que o PWM não seja mais controlado a partir do servidor, mas sim por um processamento local feito baseado na intensidade do LDR.
- *Ping*. Comando que é utilizado para medir o tempo de ida e volta de um pacote (do servidor para o nó e de volta do nó para o servidor). Assim que o servidor envia este comando um cronômetro é iniciado, sendo parado somente quando uma mensagem "ping" de resposta é recebida do nó.

### 3.5.1.2 Conexão Rede Mesh-Servidor

Sabendo o IP e a porta em que o servidor estão conectados é necessário adicionar esta informação a configuração de cada um dos ESP32, desta forma o nó raiz da rede saberá em qual IP e porta se conectar assim que for eleito. Porém, antes que a conexão com o servidor seja feita, é necessário que o nó raiz conecte-se também a um roteador, o

qual será o responsável pelo gerenciamento de pacotes nas transmissões entre o servidor e a rede *mesh*. O nome da rede Wi-Fi criada pelo roteador e senha para conectar-se a ela também devem ser adicionadas a configuração de cada um dos ESP32. Na Figura 31 é ilustrada a comunicação entre o servidor os nós da rede *mesh* tendo o roteador como intermediário.

Figura 31 – Exemplo de conexão entre rede *mesh* e servidor.

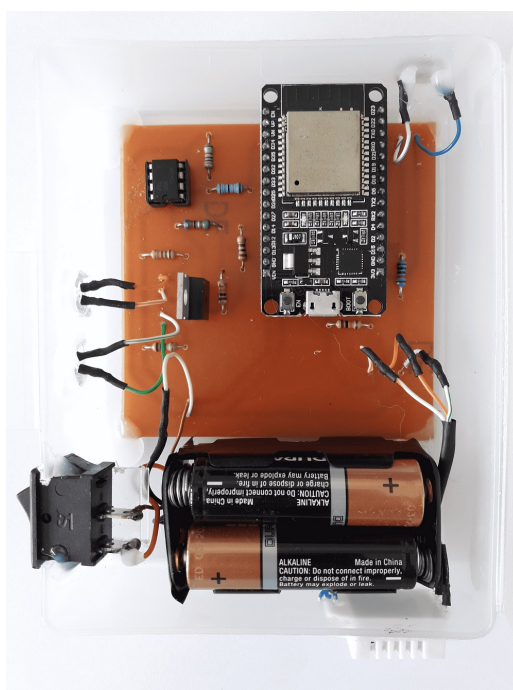


Fonte: Adaptado *ESPRESSIF SYSTEMS* (2019).

### 3.5.2 Aspectos Construtivos

Após desenvolvidos os circuitos auxiliares dos periféricos para a interface com o *hardware*, realizaram-se então as atividades de produção e montagem das placas de circuito impresso contendo o *layout* dos circuitos. Na Figura 32 está apresentado o resultado do processo de uma das unidades confeccionadas neste trabalho, com uma foto da vista superior da disposição dos componentes eletrônicos.

Figura 32 – Vista superior do sistema eletrônico projetado.



Fonte: Os autores (2019).



Além disso, foram elaborados gabinetes plásticos para comportar os componentes eletrônicos e as placas de circuito de impresso, visando protegê-los das condições ambientes assim como facilitar a sua instalação no local de teste. Na [Figura 33](#) pode-se visualizar como ficaram os seus aspectos construtivos de uma unidade confeccionada.

Figura 33 – Gabinete plástico para o suporte do sistema eletrônico projetado.



Fonte: Os autores (2019).

## 4 TESTES

### 4.1 DESCRIÇÃO GERAL

Para verificar as funcionalidades do sistema eletrônico projetado, foram realizados um conjunto de testes em campo envolvendo 10 de suas unidades. Esta quantidade é mais que suficiente para formar uma rede *wireless mesh* de modo proporcionar o teste de diferentes cenários possíveis, assim como avaliar o seu desempenho e analisar as suas aplicabilidades. Na [Figura 34](#) está exibido uma foto das 10 unidades participantes dos testes.

Figura 34 – As 10 unidades participantes dos testes.



Fonte: Os autores (2019).

Os testes foram realizados sob as seguintes condições:

- Todas as unidades foram identificadas por uma etiqueta contendo um número natural e distinto de 1 a 10. Esta etiqueta é de grande importância para a localização de uma determinada unidade na rede *wireless mesh* depois de criada.
- Todas as unidades têm obrigatoriamente, ao menos, as seguintes funcionalidades: sensoriamento da luminosidade local; atuação dos LEDs; sensoriamento do consumo de corrente elétrica pelos LEDs. E apenas três possuem a funcionalidade extra de sensoriamento de temperatura e umidade ambiente (etiquetas nº 1, 5 e 10).
- Todas as unidades foram configuradas para limitar o nº máximo de conexões filhas:
  - Etiquetas nº 1, 2 e 3 limitado com 2 conexões filhas (*downstream*).
  - Etiquetas nº 4, 5, 6, 7, 8 e 9 limitado com 1 conexão filho (*downstream*).
  - Etiqueta nº 10 limitado com 4 conexões filhas (*downstream*).

#### 4.1.1 Do Local

O local escolhido foi o estacionamento reservado aos docentes e discentes do curso de Engenharia Elétrica, localizado ao lado do Bloco PK - Centro Politécnico - UFPR, devido a disposição regular dos seus pontos de iluminação que favorece o teste de diferentes cenários. Além disso, esse local reproduz as condições semelhantes das encontradas em sistemas de iluminação pública, tais como a presença de arvoredos e o constante fluxo de pessoas e veículo, assegurando que os resultados a serem obtidos podem ser generalizados para uma aplicação real. Na [Figura 35](#) é apresentado uma foto com a vista panorâmica do local em questão.

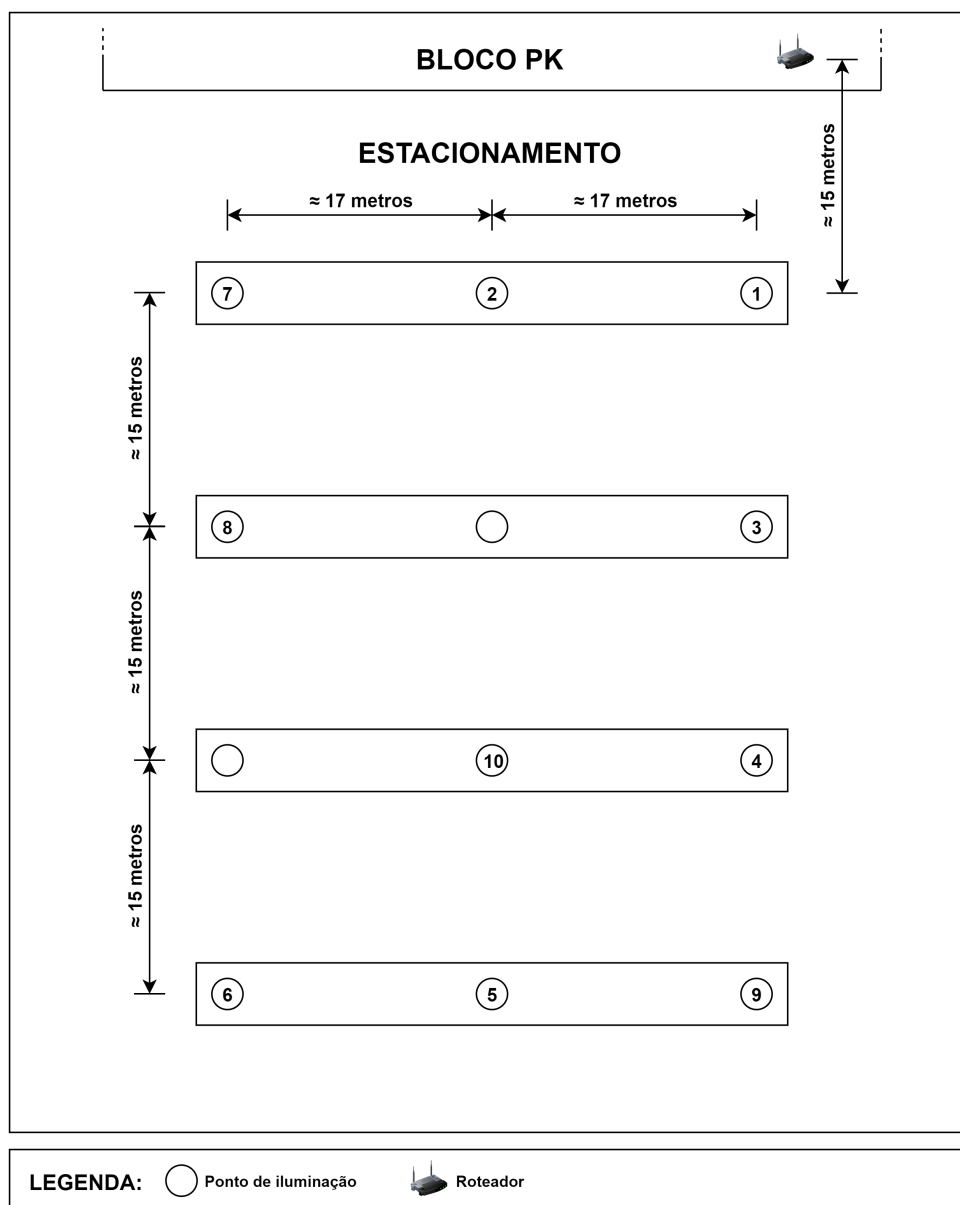
Figura 35 – Vista panorâmica do local escolhido para os testes.



Fonte: Os autores (2019).

Para o melhor entendimento do leitor, foi construído um diagrama com uma representação do arranjo dos pontos de iluminação encontrado no estacionamento ([Figura 36](#)). Como pode-se observar, há um total de 12 postes que estão distribuídos igualmente em 4 fileiras com distâncias regulares de separação. As medidas presentes no diagrama foram estimadas com o auxílio do Google Maps.

Figura 36 – Diagrama com a disposição dos pontos de iluminação do local de testes.



Fonte: Os autores (2019).

#### 4.1.2 Da Instalação

Dos 12 pontos de iluminação disponíveis, foram selecionados 10 para receber o sistema eletrônico projetado. A instalação de cada unidade, como mostra a [Figura 37](#), foi feita na base principal do poste sob uma altura aproximada de 2 metros em relação ao chão. O reconhecimento de como ficou distribuída as 10 unidades, após a instalação, está representado no diagrama da [Figura 36](#), onde em cada ponto de iluminação instalado está identificado com o número da etiqueta da respectiva unidade.

O roteador, responsável para a comunicação com o nó raiz da rede *mesh* criada, foi instalado no gabinete do professor orientador deste trabalho. No diagrama da [Figura 36](#) está representado, a grosso modo, sua localização em relação ao estacionamento.

Figura 37 – A instalação do sistema eletrônico no poste.



Fonte: Os autores (2019).

#### 4.2 CENÁRIOS A SEREM ANALISADOS

- **Cenário 1:** Formação da rede *mesh* com 9 nós participantes (etiquetas nº 1 a 9).
- **Cenário 2:** Inserção de um novo nó à rede (etiqueta nº 10).
- **Cenário 3:** Recuperação da rede após o egresso do nó identificado como raiz.
- **Cenário 4:** Reorganização da rede após o retorno do nó falho do cenário anterior.
- **Cenário 5:** Recuperação da rede após o egresso de um nó identificado como pai intermediário (etiqueta nº 8).
- **Cenário 6:** Recuperação da rede após o egresso de um segundo nó identificado como pai intermediário (etiqueta nº 3).

### 4.3 MÉTRICAS A SEREM AVALIADAS

- **Quadros de sinalização Wi-Fi (RSSI):** indicação relativa de intensidade do sinal Wi-Fi recebida por um nó, do seu respectivo pai, para a transmissão de dados.
- **Tempo de recuperação:** intervalo de tempo necessário para a rede detectar o egresso de um nó e executar ações apropriadas para recuperar a funcionalidade da rede (como gerar um nó raiz ou formar novas conexões).
- **Latência na rede:** o atraso da transmissão de dados entre o servidor e um nó de destino, ou vice-versa.
- **Alcance para o link entre dois nós:** a máxima distância, em metros, que um nó poderá estar afastado de seu vizinho sem que haja a sua saída da rede.

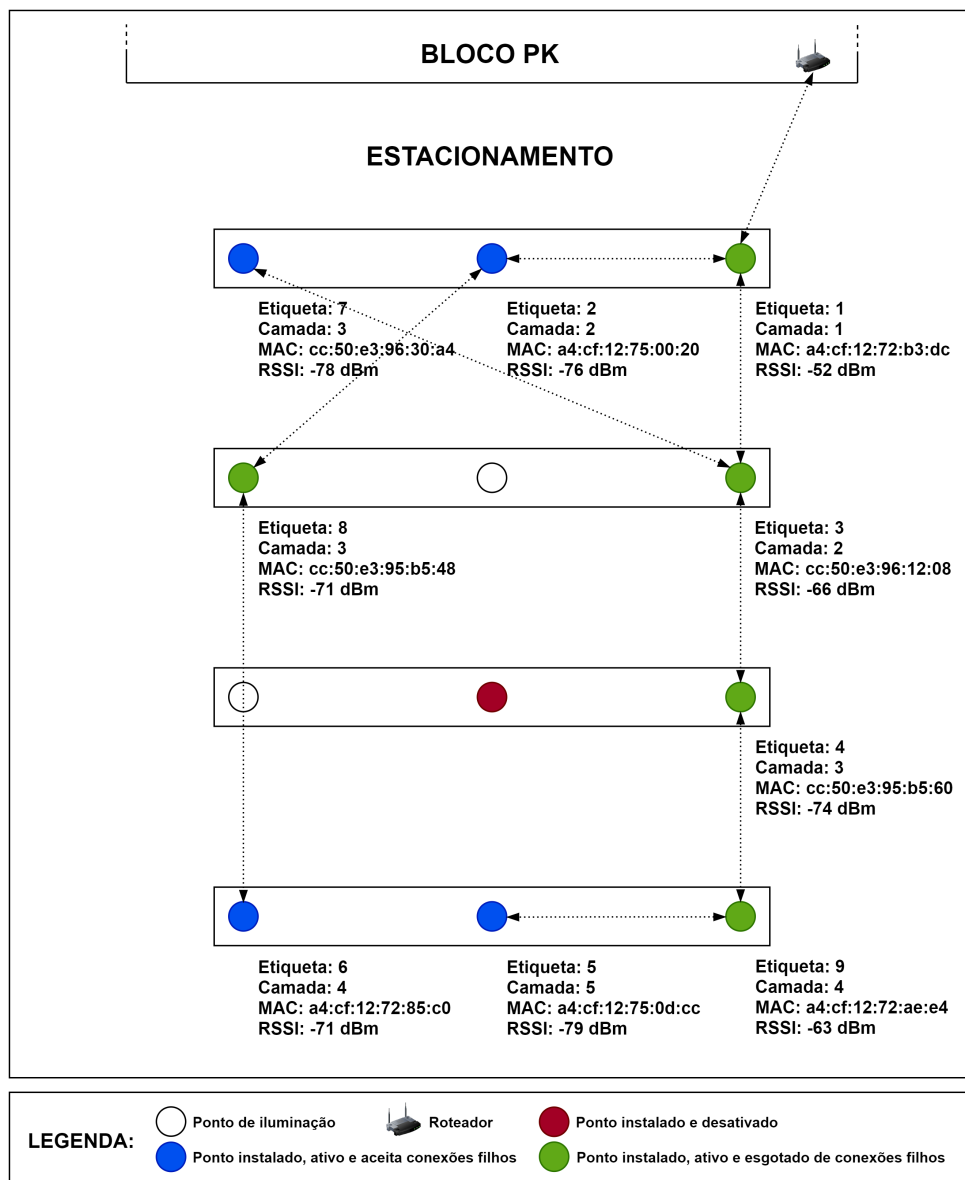
## 5 RESULTADOS E DISCUSSÕES

### 5.1 DOS CENÁRIOS

#### 5.1.1 Cenário 1

A Figura 38 ilustra o estado final do primeiro cenário dos testes, que é a formação da rede *mesh* com 9 nós participantes. Inicialmente todos os nós estavam desligados e foram sendo ligados um a um de forma assíncrona.

Figura 38 – Primeiro cenário: formação da rede com 9 nós participantes.



Fonte: Os autores (2019).

O primeiro a ser ligado foi o nó "1". Como ele era o único nó da rede até o momento ele elegeru-se nó raiz e então conectou-se ao roteador. Em seguida foram ligados, ao mesmo tempo, os nós com etiquetas "2" e "3". Como estes nós estão dentro da área de cobertura do nó "1", e ele pode aceitar até 2 conexões *downstream*, os nós "2" e "3" conectaram-se a ele, formando assim a segunda camada da rede *mesh*.

Os nós seguintes a serem ligados foram aqueles com as etiquetas "4", "7" e "8". Para o nó "7", das opções disponíveis que eram os nós "2" e "3", ele conectou-se ao nó "3". Em nenhum dos cenários que serão aqui descrito o nó 7 conectou-se aos nós "1" e "2", que estão na mesma fileira que ele. Isso ocorreu porque possivelmente existe alguma obstrução entre ele e esses nós, fazendo assim com que ele tivesse que buscar conexão com nós de outras fileiras.

As opções de conexão para o nó "8" eram as mesmas que aquelas apresentadas para o nó "7", ou seja, poderia formar uma conexão *upstream* com os nós "2" ou "3", mas acabou conectando-se com o nó "2". Já o nó "4", devido a maior proximidade com o nó "3" do que com o nó "2", fez sua conexão *upstream* com ele.

Dos três nós que foram adicionados, dois deles fizeram conexão *upstream* com o nó "3" e um deles com o nó "2". Isso é um resultado esperado uma vez que a rede é formada de tal forma a equilibrar a quantidade filhos para evitar sobrecarga em algum nó. A partir deste ponto os nós "1" e "3" não podem mais fazer conexão *downstream*, isso por que o número máximo de conexões (que é igual a dois) já foram feitas.

Os dispositivos identificados como "9", "5" e "6" foram ligados em sequência. O nó "9" quando foi ligado possuía como opção de conexão os nós "4", "2", "7", "8". O nó mais raso da rede é o nó "2", porém, assim como os nós "7" e "8", ele está muito distante do nó "9", não possuindo uma RSSI suficiente para ser considerado um pai potencial. Por isso, o nó "9" escolheu fazer sua conexão *upstream* com o nó "4", entrando assim na camada quatro da rede. Quando o nó "5" foi ligado as suas opções de conexão eram semelhantes às aquelas apresentadas para o nó "9". Como o nó "5" está próximo ao "9", e conseqüentemente longe dos nós "2", "7" e "8", a premissa para seleção de nó pai foi a mesma que foi seguida pelo nó com etiqueta "9". Assim a conexão do nó "5" foi feita com o nó "9", formando a camada cinco da rede. Por último, o nó "6" foi ligado. Suas opções de conexão eram: "2", "5", "7" e "8". Os nós "2" e "7" estão muito longe do nó "6", por isso o RSSI era o suficiente para que fossem considerados como nó pai potencial, restando assim os nós "8" e "5". Ambos possuem um RSSI suficiente para se tornarem nó pai, porém o nó "8" está em uma camada mais rasa da rede quando com parado com o nó "5", por isso o nó "8" foi escolhido como nó pai para formação da conexão *upstream*.

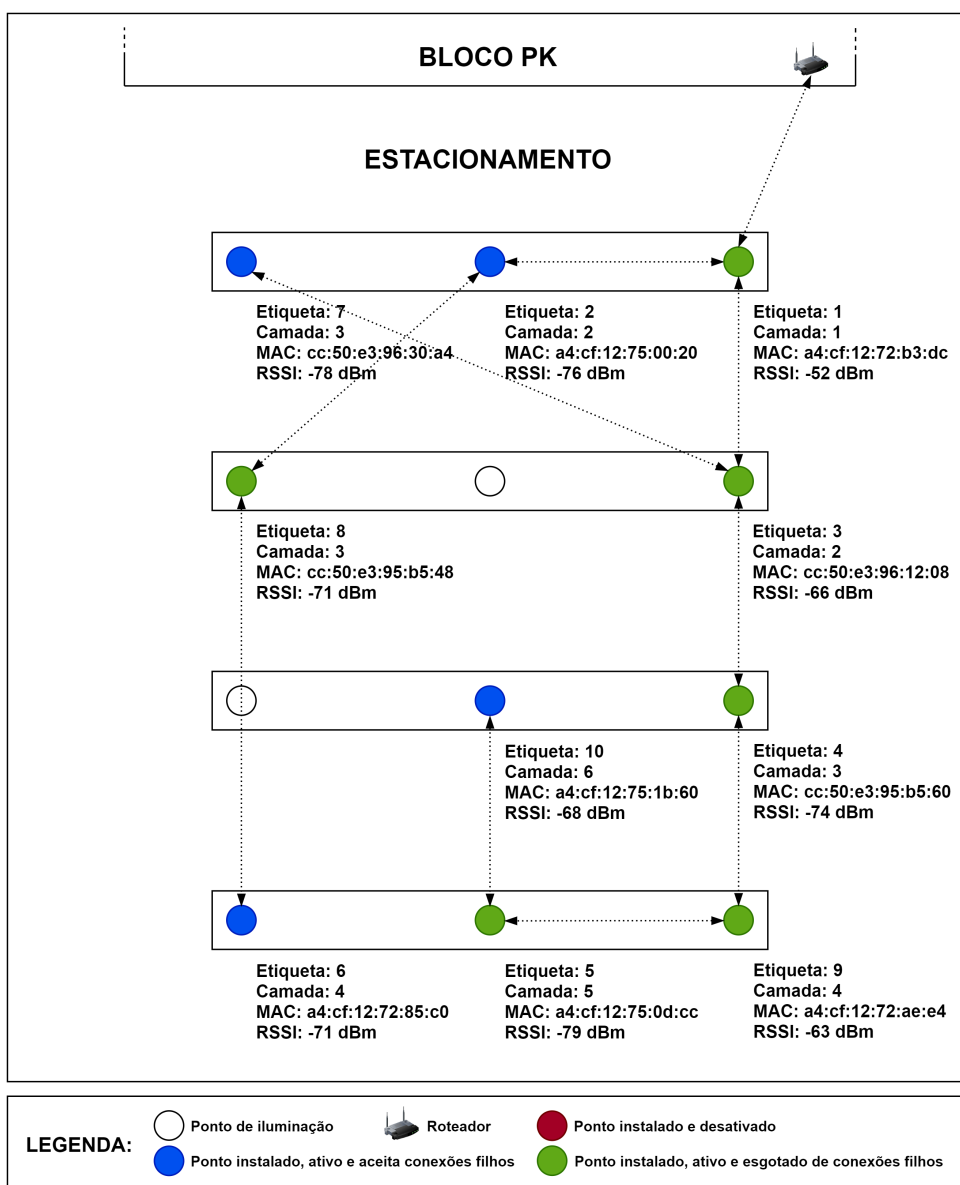
Este último cenário de conexão (nó "6" fazendo conexão com nó "8") está de acordo com o esperado segundo a descrição do protocolo ESP-MESH, ou seja, quando dois pais potenciais estão próximos de um nó, aquele que estiver em uma camada mais rasa da rede será o escolhido como nó pai para formar a conexão *upstream*.



5.1.2 Cenário 2

Após a formação da rede com os 9 nós participantes, foi ativado o nó identificado pela etiqueta "10". Como pode-se observar pela [Figura 39](#), o nó "5" tornou-se seu pai, o que era um dos possíveis resultados esperados. Pois dos nós ativos, apenas quatro eram possíveis para se tornarem pai (os nós "2", "5", "6" e "7") em consequência do número limite de conexões filhos pré-estabelecidos. Dos candidatos, esperava-se que o pai preferido fosse o nó "2" por estar em uma camada inferior comparados aos outros. Contudo, a possível existência de obstruções entre os nós devem ter diminuído o seu valor de RSSI, impossibilitando a conexão. Causa que também é considerada válida para o nó "7".

Figura 39 – Segundo cenário: inserção de um novo nó à rede.



Fonte: Os autores (2019).

Dos dois pais em potencial restantes, verificou-se que o nó "10" optou a conexão com o nó de melhor indicativo RSSI e o mais próximo geograficamente, embora este esteja situado em uma camada maior em comparação ao outro. Por fim, assim que a conexão com seu pai foi instituída, o nó "10" passou a integrar uma nova e maior camada da rede.

### 5.1.3 Cenário 3

A [Figura 40](#) ilustra o terceiro cenário executado na prática que é a falha do nó raiz. Para a execução deste caso particular a rede apresentada no cenário dois foi utilizada como base, sendo a única diferença que o nó raiz (nó com etiqueta "1") foi desligado. Assim que o nó raiz é desligado, os nós "2" e "3" (que estão presente na camada dois) iniciam uma votação entre eles para eleger um novo nó raiz. Será eleito aquele que tiver o maior valor de RSSI do roteador.

O nó "2" é eleito como novo nó raiz e então faz um conexão *upstream* com o roteador. Como o nó "3", que estava conectado ao nó "1", agora está sem um nó pai, ele passa a procurar um nó para se conectar. As opções disponíveis são: "2", "10" e "6". O nó "2" é o que está mais próximo do nó "3" e também em uma camada mais rasa da rede, logo cumpre os dois requisitos para se tornar o nó pai preferido, assim o nó "3" forma uma conexão *upstream* com ele. O nó "8" que anteriormente estava conectado ao nó "2", manteve esta conexão após a eleição do nó "2" como nó raiz, assim o nó 8 passou da camada três para a camada dois.

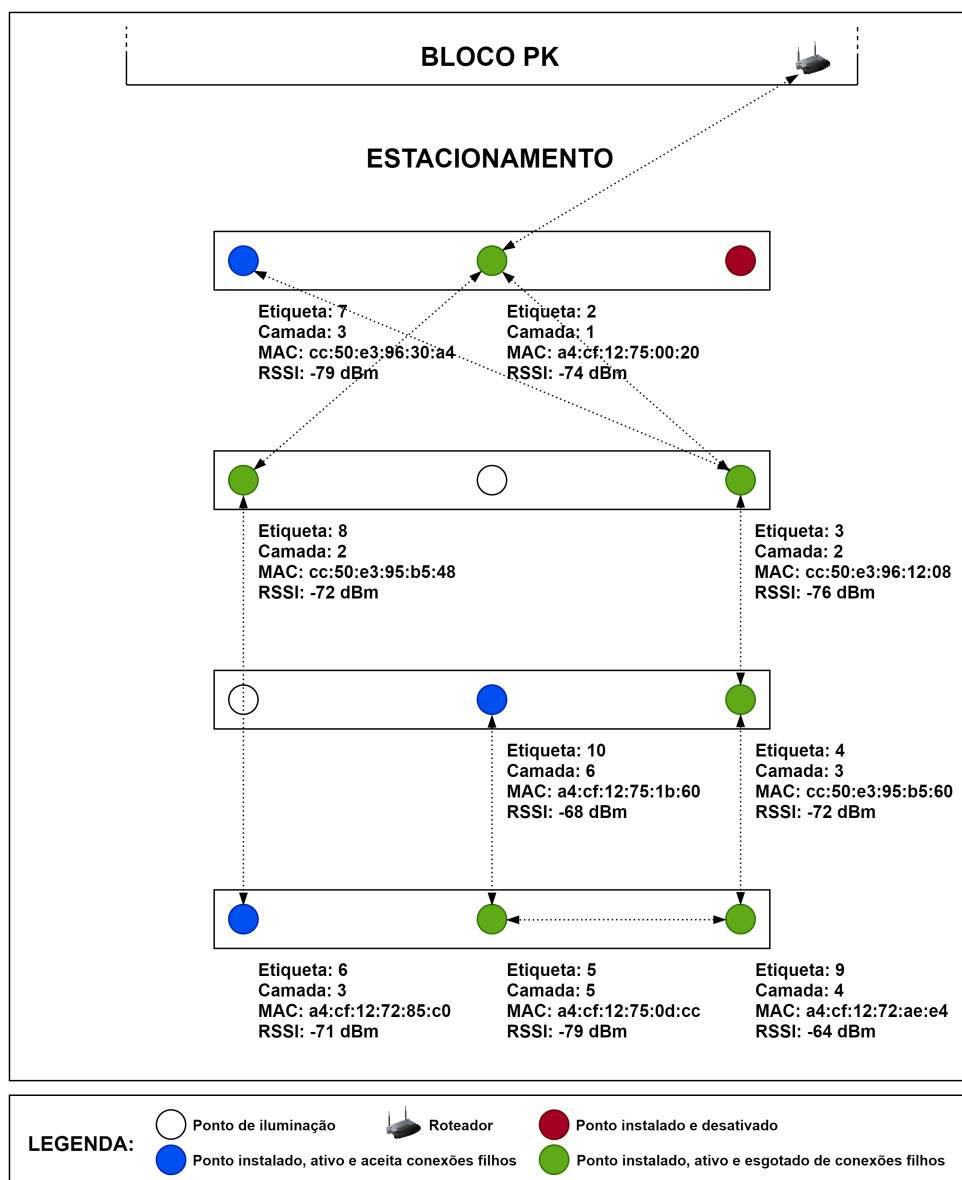
Um nó da camada dois foi eleito como novo nó raiz e o nó "3", assim que ficou sem um nó pai, escolheu dentre as opções possíveis um que, além de ter um bom RSSI, estivesse na camada mais rasa da rede. Esses dois resultados são esperados dada a descrição do protocolo ESP-MESH na [Seção 2.2](#). Além da troca do nó raiz, a estrutura da rede em camadas inferiores foi mantida. Isso também é um resultado esperado uma vez que a troca geográfica do nó raiz não foi significativa e seria custoso (com relação a tempo de formação e banda devido ao tráfego de dados para reorganização) para a rede refazer todas as camadas já criadas.

### 5.1.4 Cenário 4

Após a recuperação da rede no cenário anterior, foi ativado o nó falho para permitir o seu reingresso. Contudo, como o resultado ilustrado na [Figura 41](#), o nó "1" não retornou como nó da primeira camada. O que era de fato esperado, pois segundo as definições do protocolo ESP-MESH, o nó raiz tem como uma das características de não alternar mesmo que surja um nó em mais potencial ou o RSSI do roteador se degrade sem que haja a desconexão. A troca automática do nó raiz apenas ocorre em caso provado de seu egresso.

O motivo do nó raiz não alternar, mesmo em condições desfavoráveis, é explicado pela prioridade em manter preservado as conexões seguintes, de modo não reestruturar novamente do zero a rede, pois há um custo significativo de tempo para recuperação.

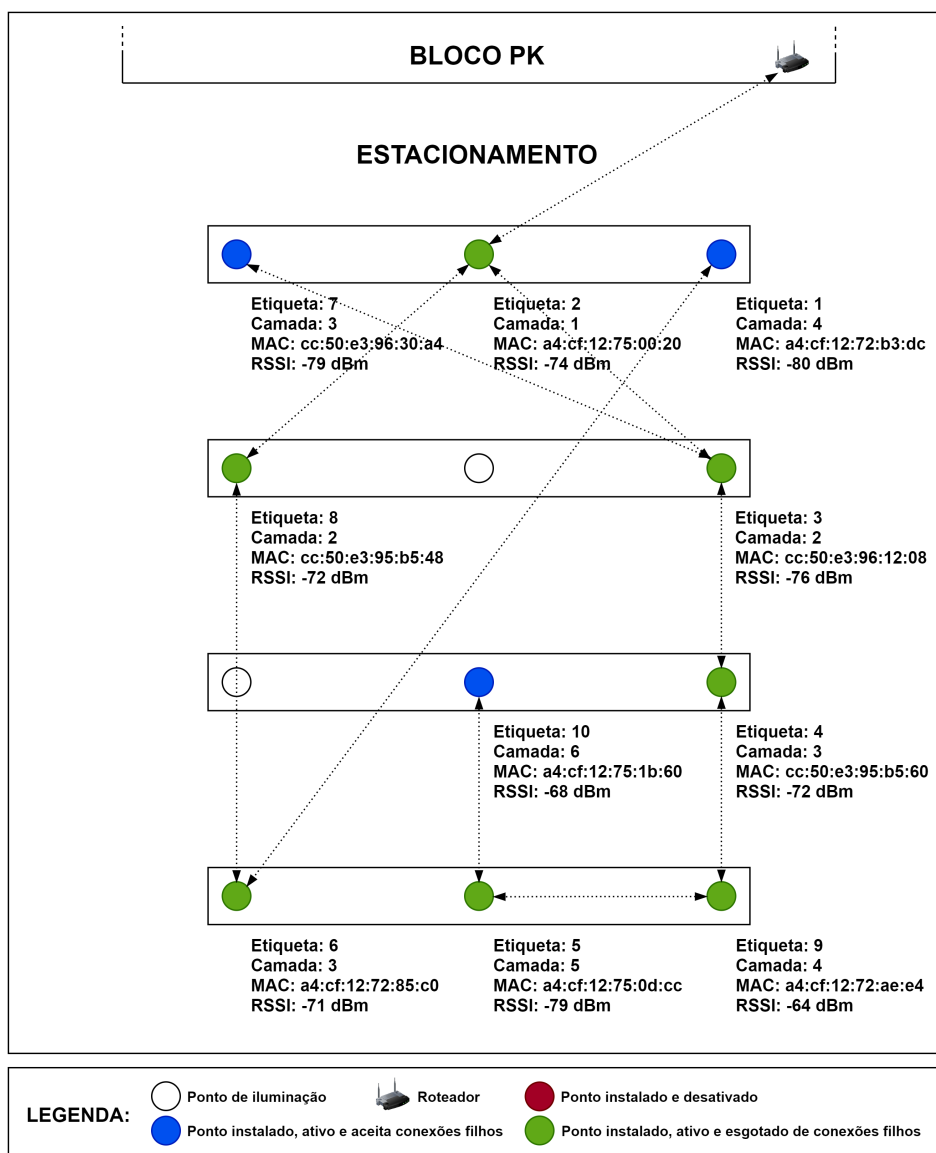
Figura 40 – Terceiro cenário: falha do nó raiz.



Fonte: Os autores (2019).

Assim, o nó "1" passará a ser tratado como um novo nó que foi inserido a rede, idem ao que aconteceu no cenário 2. Dos nós ativos, três são candidatos possíveis para serem pais - nós "6", "7" e "10". Destes, dois são pais preferidos por estarem na mesma e menor camada em comparação ao terceiro - os nós "6" e "7". Por fim, a escolha entre os dois fica então por critério de RSSI. E como pode observar pela [Figura 41](#), as condições ambientes concederam a conexão com o nó "6", permitindo um link com uma distância aproximada de 56 metros entre os nós.

Figura 41 – Quarto cenário: retorno do nó falho do cenário anterior.



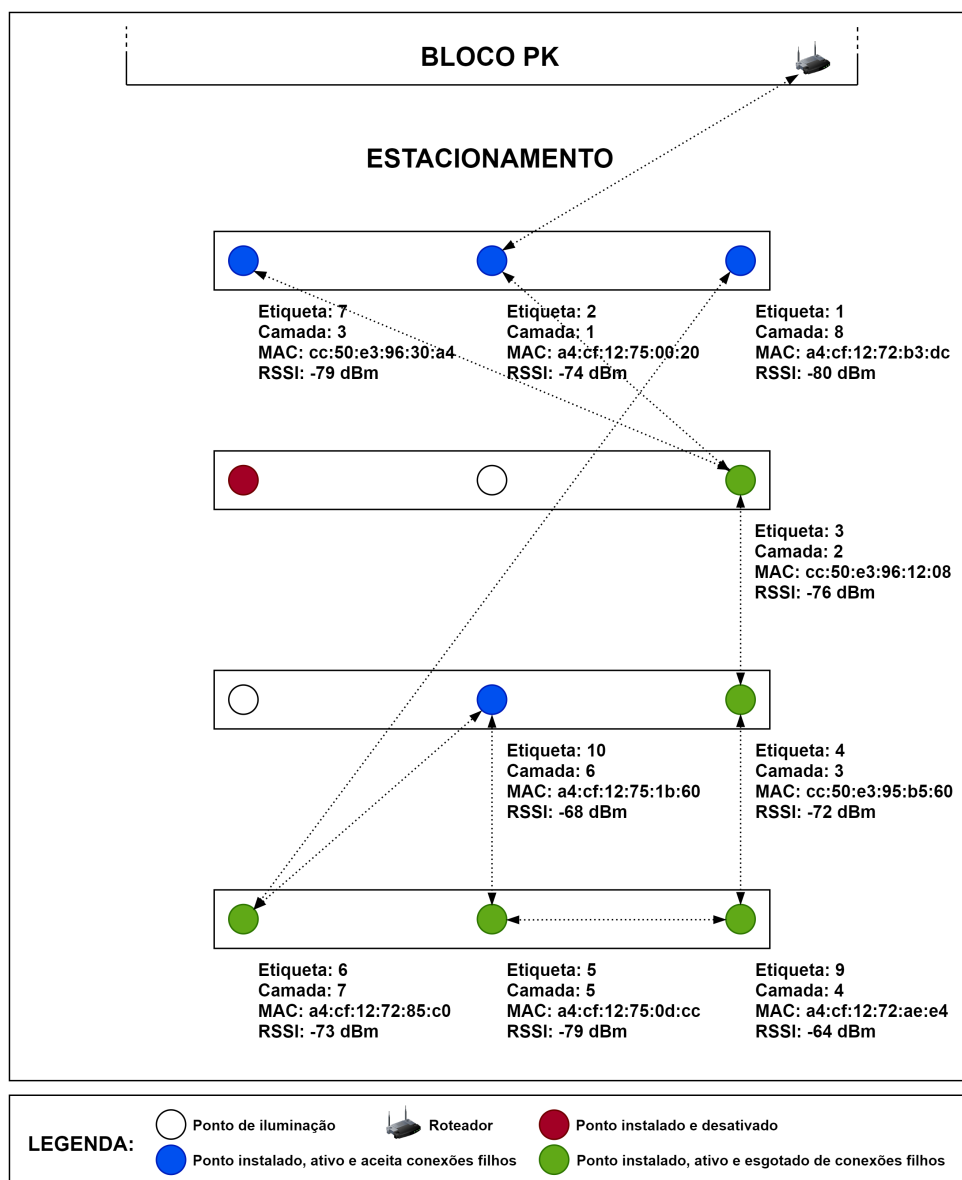
Fonte: Os autores (2019).

### 5.1.5 Cenário 5

Neste cenário foi avaliado o comportamento da rede quando acontece a falha de um nó pai intermediário. O cenário quatro foi utilizado como base e o nó que foi desligado é o nó com etiqueta "8". Apenas um outro nó está conectado ao nó "8", o nó "6". Assim que ele perde seu nó pai, ele começa a avaliar os outros nós da rede que ainda podem fazer conexões *downstream* e são esses os nós: "2", "7" e "10". Os nós "2" e "7" estão em uma camada mais rasa da rede, mas também a uma maior distância do nó "6", conseqüentemente não possuem um RSSI suficiente para serem considerados como nó pai em potencial. Desta forma o nó "6" cria uma conexão *upstream* com o nó "10".

Além da nova conexão formada entre os nós "6" e "10", nenhuma outra alteração aconteceu na rede. Nem mesmo o nó "1" que estava conectado ao nó "6" sofreu alguma

Figura 42 – Quinto cenário: falha de um nó pai intermediário.



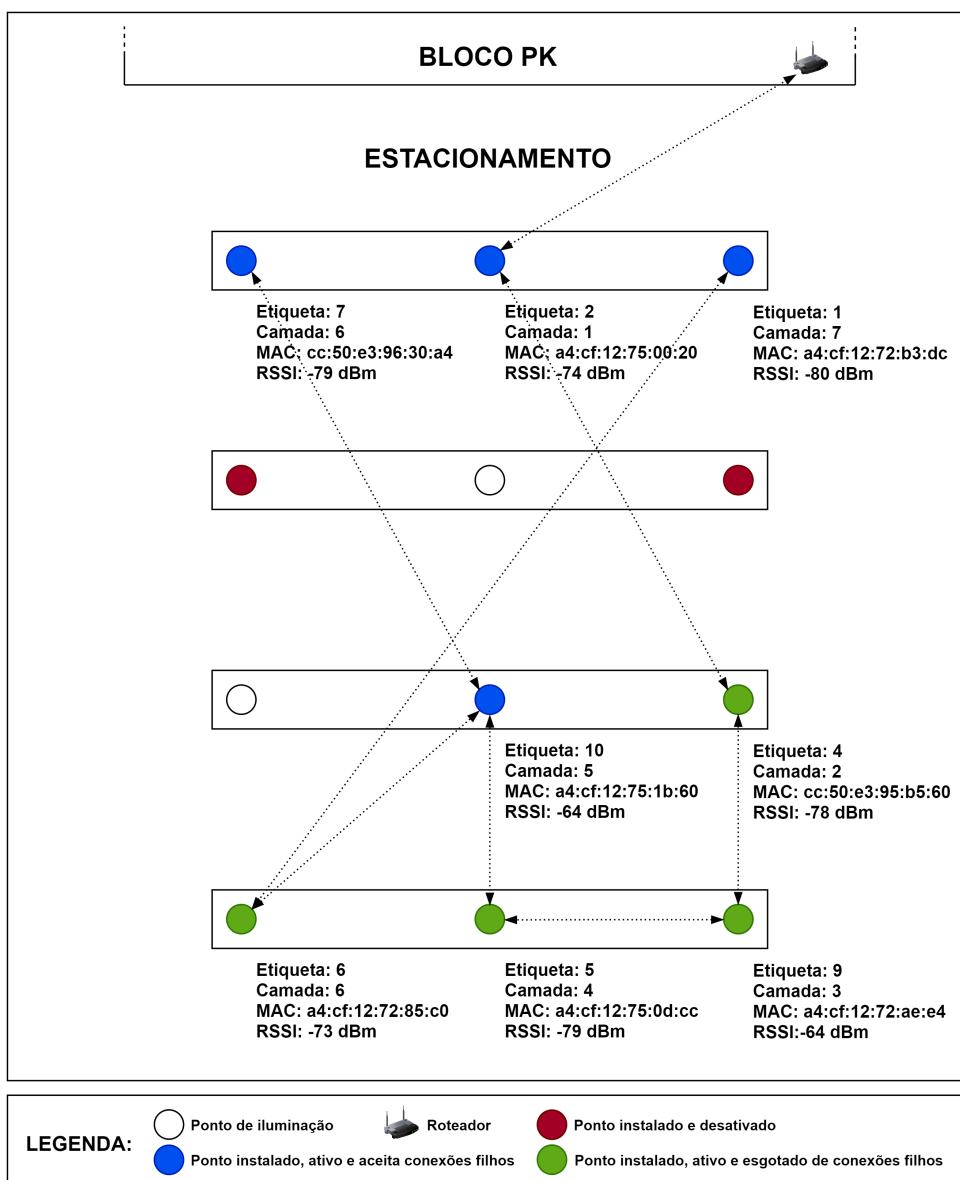
Fonte: Os autores (2019).

alteração de conexão. E isso acaba sendo um resultado esperado dada a descrição de custo de reorganização da rede feita em cenários anteriores.

### 5.1.6 Cenário 6

No último cenário avaliado, por fim, verificou-se a recuperação da rede após a falha de um segundo nó pai intermediário e considerado vital, pois ele é o único nó da camada 2 que sustenta a conexão dos nós restantes depois de seu pai (nó raiz). E como pode-se observar no resultado ilustrado na [Figura 43](#), a rede novamente seguiu os mesmos conceitos vistos nos cenários anteriores, mantendo as conexões pré-existentes e formando novas para as conexões falhas.

Figura 43 – Sexto cenário: falha de um segundo nó pai intermediário.



Fonte: Os autores (2019).

## 5.2 DAS MÉTRICAS

### 5.2.1 Quadros de Sinalização Wi-Fi

O estacionamento do Bloco PK onde os testes foram realizados possui uma grande quantidade de árvores, não só próxima aos postes onde as caixas foram instaladas, mas também espalhadas ao longo de todo o local. No momento do teste, o estacionamento estava lotado de carros nas vagas destinadas a eles. Todos estes elementos podem funcionar como obstruções para a propagação do sinal Wi-Fi de cada um dos ESP32. Além disso, o sinal de Wi-Fi da UFPR (UFPR sem fio) está na mesma frequência que a do ESP32. Isso também pode prejudicar a conectividade entre os dispositivos devido a interferência.

Mesmo com todos estes impedimentos para a formação da rede, em nenhum momento qualquer dispositivo ficou desconectado por não encontrar algum nó pai com que se conectar. Apenas uma conexão que era esperada durante os testes acabou não ocorrendo devido a grande obstrução causada pelas árvores. Esta era a conexão do nó "7" com o nó "2" durante a formação da rede no cenário um.

Os valores de RSSI configurados para que um pai seja considerado um pai forte em potencial é de -78 dBm e mesmo para a conexão mais distante formada, que é a conexão do nó "1" com o nó "6" no quarto cenário, o valor de RSSI está muito próximo de formar uma conexão bastante estável.

### 5.2.2 Tempo de Recuperação

O tempo de recuperação para os cenários acima apresentados foi satisfatório para o funcionamento da rede. Tanto a recuperação de falha do nó raiz, quanto a recuperação por perda de um pai intermediário estão na faixa de minutos. Com relação a perda do nó raiz, os nós da segunda camada tentam algumas reconexões antes de descartar o nó raiz antigo e iniciar uma nova rodada de votação. Todo esse processo de tentar algumas reconexões, eleição de novo nó raiz e conexão do mesmo com o roteador levou 50 segundos.

Já para o caso de recuperação de perda de um nó pai intermediário, assim como no cenário com a perda do nó, os filhos tentam se reconectar com o nó pai por um número de vezes. Caso não consigam, passam então a procurar um novo nó pai. O tempo necessário para que todos os filhos de um nó pai intermediário voltassem a se conectar à rede foi em média de 1 minutos e 45 segundos como pode ser observado na [Tabela 3](#).

Tabela 3 – Resultados dos tempos de recuperação em caso de perda de um nó pai intermediário.

Cenário	Tempo de Recuperação
5	1:30 min
6	2:00 min

Fonte: Os autores (2019).

Estes valores podem ser muito bem aceitados em uma aplicação real quando instalados em postes de iluminação pública, uma vez que caso ocorra alguma falha no sistema eletrônico de algum poste e este saia da rede, o tempo necessário para que a rede se recupere é de apenas alguns minutos.

### 5.2.3 Latência na Rede

Durante a execução do cenários 3 foram também feitos alguns testes de latência entre o servidor e os nós da rede. Este teste da latência consistia em medir o tempo de ida

e volta de pacotes, ou seja, do servidor até um nó específico e de volta do nó até o servidor. O teste foi feito para um nó de cada camada e o resultado pode se visto na [Tabela 4](#).

Tabela 4 – Resultados dos tempos de ida e volta de pacotes entre o servidor e diferentes camadas da rede.

Camada	Etiqueta do Nó	Tempo de Ida e Volta [ms]
1	2	24,74
2	3	37,18
3	6	33,99
4	9	41,20
5	5	52,73
6	10	66,29

Fonte: Os autores (2019).

É possível notar que quanto maior o número da camada, maior o atraso de ida e volta dos pacotes. O que é um resultado esperado uma vez que quanto maior a camada, maior será a quantidade de saltos do pacote até chegar ao destino. Apenas a camada 3 não seguiu a tendência de aumento dos tempos, mas cada dispositivo na rede não cuida exclusivamente do encaminhamento de pacotes. Existem tarefas que rodam em paralelo a isso como amostragem nos ADCs, alteração do valor do PWM e atividades do sistemas operacional. Esta pode ser uma das explicações para o tempo ter ficado abaixo daquele da camada dois. Porém, os resultados do teste de atraso foram bastante satisfatórios e mesmo para a camada 6, última da rede, o atraso de ida e volta foi de 66,29 ms.

#### 5.2.4 Alcance para o Link entre Dois Nós

Um teste de distância foi feito nos ESP32 para verificar se seria possível utilizá-los em um caso real levando em consideração a distância entre os postes, procedimento este que foi descrito na [Subseção 3.2.3](#). Porém agora o ESP32 está dentro de uma caixa de plástico e com vários circuitos periféricos em seu entorno. Por isso um novo teste de distância foi feito para medir até qual distância seria possível conseguir um *link* de comunicação dadas essas novas condições.

Este novo teste foi feito também no centro politécnico e a distância máxima alcançada foi aproximadamente de 100 metros (medida estimada pelo *Google Maps*). Um resultado muito satisfatório, pois uma vez que a distância entre os postes varia de 30 a 40 metros, mesmo que um poste perca comunicação com a rede, ainda assim é possível que um dispositivo se comunique com outro a dois postes de distância.



### 5.3 DAS APLICABILIDADES

Durante a realização dos testes de cenários aproveitou-se também para verificar o funcionamento das aplicabilidades propostas neste trabalho, exceto do dispositivo DHT22 pois a sua implementação não ficou pronta até os testes finais. Nas subseções seguintes serão analisados os principais resultados obtidos.

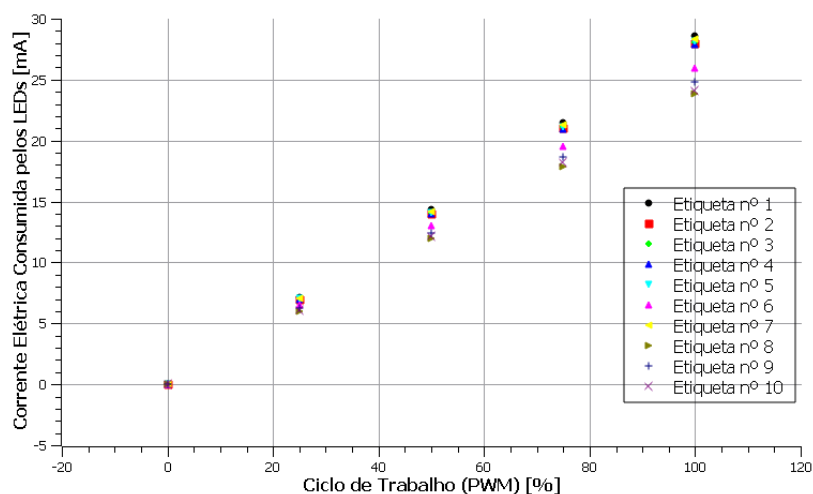
#### 5.3.1 Atuação dos LEDs

O controle da intensidade luminosa dos LEDs foi testada em todos os nós ativos de cada cenário. De modo geral, verificou-se uma rápida resposta para sua atuação, tanto por controle local quanto por comando vindo do servidor externo. Em ambos os casos, a dimerização dos LEDs foi analisada com sucesso, em campo, pelas suas variações de intensidade luminosa, assim como pelos seus dados de consumo de corrente elétrica enviados por cada nó ao servidor (por exemplo, na Figura 44).

#### 5.3.2 Consumo de Corrente Elétrica e Detecção de Falhas

Para averiguar a funcionalidade do consumo de corrente elétrica dos LEDs, foram executados comandos de valores PWM (0%, 25%, 50% e 100%) no servidor em um cenário que todos os nós estivessem ativos. A ideia é observar, de modo geral, a variação relativa de consumo entre nós para estabelecer médias e desvios, e assim, poder detectar maus funcionamentos como também falhas dos LEDs. Na Figura 44 está apresentada o resultado obtido desse processo. E como pode-se observar, todos os nós participantes tiveram um nível de consumo de corrente elétrica padrão, com uma variação máxima de 5 mA para PWM 100%, certificando que todos os nós estavam com LEDs atuando corretamente.

Figura 44 – Monitoração de consumo de corrente elétrica pelos LEDs.



Fonte: Os autores (2019).

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Considerando a proposta do trabalho, e tendo em vista as diferentes etapas do projeto, pode-se concluir que o objetivo do trabalho foi satisfeito. Foi possível testar todos os cenários necessários para validar que a solução funcionaria em uma aplicação real. Isto é, foram verificados: a formação da rede *mesh*, a coleta de informações através de sensores, o controle na atuação na luminosidade baseado no valor da luminosidade do local e o controle de luminosidade remoto com comandos provindos de um servidor. Exemplos de aplicações que podem possibilitar a redução adicionais de custos na operação e manutenção em sistemas de iluminação pública, e conseqüentemente, uma eficiência energética.

Isso foi possível porque todas as etapas do projeto foram concluídas com êxito, desde a análise de mercado para escolha do *hardware*, a concepção dos circuitos eletrônicos, a utilização da biblioteca ESP-MESH e a programação do ESP32 / servidor. Todas as partes foram testadas ao máximo individualmente, tanto em *hardware* quanto em *software* de tal forma que no momento da integração de todas as partes o sistema operasse da maneira esperada. Dos sensores que foram utilizados, o LDR e o resistor *shunt* funcionaram de maneira adequada e serviram como fonte de informação tanto para o tráfego de dados na rede quanto para a atuação dos LEDs, promovendo a aplicabilidade no projeto.

O teste final no politécnico cumpriu sua missão de avaliar a rede *mesh* em diferentes cenários, cobrindo as possíveis situações que podem ser encontradas em uma aplicação real. E como relatado no [Capítulo 5](#), a formação da rede e a adição de novos nós, além do comportamento em casos adversos como falha do nó raiz e falha(s) de nó pai intermediário, funcionaram de acordo com o protocolo ESP-MESH. Com relação as métricas que foram estabelecidas para avaliar o desempenho da rede, verificou-se um tempo máximo de 2 minutos para recuperação em casos de falha(s), um tempo de atraso entre saltos na faixa de milissegundos (ms) e um alcance da comunicação Wi-Fi em torno de 100 metros. Valores de medidas que podem ser considerados muito satisfatórios para o caso da aplicação em sistemas de iluminação pública.

O *hardware* escolhido mostrou-se capacitado para atender os requisitos estabelecidos no projeto, além de ser bastante atrativo em função de suas particularidades como alta performance, facilidade de uso, baixo consumo de energia elétrica, possibilidade de comunicação *wireless* Wi-Fi e, sobretudo, baixo preço (na faixa de 15 a 20 reais no mercado externo). O que torna o projeto bastante interessante para as prefeituras, os órgãos responsáveis pela prestação de serviços de iluminação pública no Brasil.

Em termos de trabalhos futuros, recomenda-se uma melhoria das funcionalidades do atual estado do projeto, como: a otimização do uso do *hardware*, aperfeiçoamento dos códigos de programação do *hardware* / servidor, a implementação de novas aplicabilidades e comunicação com outras tecnologias. Para finalmente empreender uma aplicação real.

## Referências

- ANEEL. **ANEEL Abre Consulta sobre o Fornecimento de Energia para o Serviço de Iluminação Pública**. 2019. Disponível em: <<http://www.aneel.gov.br>>. Citado na página 14.
- AOSONG. **DHT22 Datasheet**. 2019. Disponível em: <<https://akizukidenshi.com/download/ds/aosong/AM2302.pdf>>. Citado na página 45.
- BANK, T. W. **Iluminando Cidades Brasileiras**. 2016. Disponível em: <<https://multimidia.fnp.org.br>>. Citado na página 15.
- BARR, M. **Introduction to Pulse Width Modulation**. 2001. Disponível em: <<https://www.embedded.com/introduction-to-pulse-width-modulation/>>. Citado na página 34.
- BOYD, N. **The Main Difference Between Converged and Hyperconverged Infrastructures**. 2018. Disponível em: <<https://www.sdxcentral.com/data-center/hyperconverged/definitions/difference-converged-hyperconverged/>>. Citado na página 17.
- BUENO, E. M.; NALON, J. A. Iluminação pública inteligente utilizando rede mesh. **Revista Ciência e Tecnologia**, v. 19, n. 35, p. 71–81, 2016. Disponível em: <<http://www.revista.unisal.br/sj/index.php/123/article/view/540>>. Citado na página 14.
- CEMIG. **Projetos de Iluminação Pública**. 2012. Disponível em: <<http://www.cemig.com.br/pt-br/atendimento/documents/nd-3-4p.pdf>>. Citado na página 38.
- DEVICES, A. **Wireless Sensor Networks**. 2019. Disponível em: <<https://www.analog.com/en/products/rf-microwave/wireless-sensor-networks.html#>>. Citado na página 18.
- FARONI. **Smart Network Paving Way Iot Adoption**. 2019. Disponível em: <<https://www.al-enterprise.com/pt-br/blog/smart-network-paving-way-iot-adoption>>. Citado na página 17.
- FREITAS, A. D. P. P. **Análise Bibliométrica da Produção Científica da Indústria 4.0**. 2018. Trabalho de Conclusão de Curso apresentado para obtenção do diploma de Bacharel em Administração da Faculdade de Gestão e Negócios - FAGEN, da Universidade Federal de Uberlândia - UFU, 2018. Citado na página 14.
- GERSHWIN, A. **IoT Time: Cybersecurity Challenges for IoT Users**. 2019. Disponível em: <<https://k2partnering.com/pt-br/cyber-security-pt-br/how-iot-has-escalated-the-cybersecurity-issue/>>. Citado na página 17.
- MONTEIRO, R. V. A.; CARVALHO, B. C.; NOGUEIRA, F. Drivers de lâmpadas de led: Topologias, aplicações e desempenho. **Engineering and Science**, 2014. Disponível em: <<http://periodicoscientificos.ufmt.br/ojs/index.php/eng/article/view/1916>>. Citado na página 34.
- SCHWAB, K. M. **A Quarta Revolução Industrial**. 1. ed. [S.l.]: Edipro, 2016. Citado na página 14.
- SEDRA, A.; SMITH, K. C. **Microeletrônica**. 5. ed. [S.l.]: Pearson, 2007. 141-160 p. Citado na página 35.

- SOUZA, P. H. M. de; JUNIOR, S. J. C.; NETO, G. G. D. Indústria 4.0: Contribuições para setor produtivo moderno. **XXXVII Encontro Nacional de Engenharia de Produção**, 2017. Disponível em: <[http://www.abepro.org.br/biblioteca/TN\\_WIC\\_238\\_384\\_34537.pdf](http://www.abepro.org.br/biblioteca/TN_WIC_238_384_34537.pdf)>. Citado na página 14.
- SYSTEMS, E. **ESP 32 Series - Datasheet**. 2019. Disponível em: <[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)>. Citado na página 41.
- SYSTEMS, E. **ESP-MESH**. 2019. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/mesh.html>>. Citado na página 19.
- WENDLING, M. **Amplificadores Operacionais**. 2010. Disponível em: <<https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/3---amplificadores-operacionais-v2.0.pdf>>. Citado na página 36.
- WENDLING, M. **Sensores**. 2010. Disponível em: <<https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>>. Citado na página 44.
- WHALEY, D. **How IoT has Escalated the Cybersecurity Issue**. 2019. Disponível em: <<https://k2partnering.com/pt-br/cyber-security-pt-br/how-iot-has-escalated-the-cybersecurity-issue/>>. Citado na página 17.