

**UNIVERSIDADE ESTADUAL DO MARANHÃO**  
**CENTRO DE CIÊNCIAS TECNOLÓGICAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E**  
**SISTEMAS**  
**MESTRADO PROFISSIONAL EM ENGENHARIA DA COMPUTAÇÃO E**  
**SISTEMAS**

**LEANDRO LIMA DO NASCIMENTO**

**SISTEMA DE GERENCIAMENTO E MONITORAMENTO DE RACK'S**  
**OUTDOORS DE TELECOMUNICAÇÕES, BASEADO EM *INTERNET OF***  
***THINGS***

**São Luís – MA**

**2022**

**LEANDRO LIMA DO NASCIMENTO**

**SISTEMA DE GERENCIAMENTO E MONITORAMENTO DE RACK'S  
OUTDOORS DE TELECOMUNICAÇÕES, BASEADO EM INTERNET OF THINGS**

Dissertação de Mestrado apresentado por Leandro Lima do Nascimento ao Programa do Mestrado Profissional de Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão.

Orientador: Prof. Dr. Leonardo Henrique Gonsioroski Furtado da Silva

**São Luís - MA**

**2022**

Nascimento, Leandro Lima do.

Sistemas de gerenciamento e monitoramento de rack's outdoors de telecomunicações, baseado em internet of things / Leandro Lima do Nascimento. – São Luís, 2022.

52 f

Dissertação (Programa de Pós-Graduação em Engenharia de Computação e Sistemas) - Universidade Estadual do Maranhão, 2022.

Orientador: Prof. Dr. Leonardo Henrique Gonsioroski Furtado da Silva.

Coorientador: Prof. Dr. Rogério Moreira Lima.

1.IoT - Internet das coisas. 2.Telecomunicações. 3.Sistema de monitoramento inteligente. 4.ESP32. 5.5G. I.Título.

CDU: 621.39:004.77

**Elaborado por Giselle Frazão Tavares - CRB 13/665**



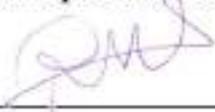
**LEANDRO LIMA DO NASCIMENTO**

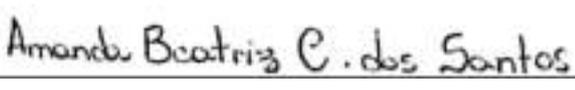
**SISTEMA DE GERENCIAMENTO E MONITORAMENTO DE RACK'S  
OUTDOORS DE TELECOMUNICAÇÕES, BASEADO EM INTERNET OF THINGS**

Dissertação de Mestrado apresentado por Leandro Lima do Nascimento ao Programa do Mestrado Profissional de Engenharia da Computação e Sistemas da Universidade Estadual do Maranhão, orientado pelo Prof. Dr. Leonardo Henrique Gonsioroski Furtado da Silva. Banca Examinadora formada pelos professores:

Aprovado em: 31 de outubro de 2022.

  
\_\_\_\_\_  
**Prof. Dr. Leonardo Henrique Gonsioroski Furtado da Silva (UEMA)**

  
\_\_\_\_\_  
**Prof. Dr. Rogério Moreira Lima (UEMA)**

  
\_\_\_\_\_  
**Amanda Beatriz C. dos Santos**

**Prof.(a) Msc.(a) Amanda Beatriz Cunha Dos Santos (PUC-RIO)**

  
\_\_\_\_\_  
**Prof. Dr. Luis Carlos Costa Fonseca (UEMA)**

  
\_\_\_\_\_  
**Prof. Dr. Mauro Sergio Silva Pinto (UEMA)**

\_\_\_\_\_  
**Prof. Msc. Marcos Tadeu Rezende De Araujo (UFMA)**

## **DEDICATÓRIA**

Dedico esta dissertação para meus familiares, que me ajudaram realizar o sonho de finalizar o Mestrado em Engenharia de Computação e Sistemas. Com todo o apoio e carinho creditado. Agradeço a minha esposa Stephany Beatriz Almeida, pelas sugestões e aperfeiçoamentos do projeto.

Remerço também aos meus grandes orientadores Leonardo Gonsioroski e Rogerio Moreira Lima pela formação profissional, intelectual e tecnológica fornecido pelo notório saber nas áreas de engenharia e telecomunicações que contribuíram para o amadurecimento e realização da Unidade Remota de Gerenciamento para telecomunicações.

*“A imaginação é mais importante que o conhecimento.”*

Albert Einstein

NASCIMENTO, Leandro Lima. **Sistema de gerenciamento e monitoramento de rack's outdoors de telecomunicações, baseado em Internet Of Things**. 2019. 33. Trabalho de Conclusão de Curso (Mestrado Profissional em Engenharia de Computação Sistemas) – Universidade Estadual do Maranhão, São Luís, 2022.

## RESUMO

O presente trabalho visa correlacionar conceitos, tecnologias e aplicação de uma solução baseada em sistemas embarcados, *Internet Of Things* e 5G, para monitorar e garantir a persistência de dados nas empresas de telecomunicações. Primeiramente, realiza-se um levantamento do atual cenário da indústria de Telecomunicações. Além disso, apresentam problemas intrínsecos, nas quais os consumidores são os mais afetados, como a falta de confiabilidade e interatividade do produto oferecido pelas concessionárias de energia. A pesquisa bibliográfica foi através de uma revisão literária baseada em artigos, livros e trabalhos de diversos autores do período de 2019 a 2022. O levantamento de informações sobre o tema ocorreu através das bases de dados: Google Scholar e IEEE. Dessa forma, apresentou-se informações conceituais das tecnologias como ESP32, 5G, sensoriamento e especialmente o protocolo MQTT, para soluções de longa distância, que pertencem ao conceito de Internet das Coisas (IoT). Por isso, propõe-se o desenvolvimento de um dispositivo capaz de monitorar e auxiliar empresas de telecomunicações no gerenciamento para manter seus ativos externos.

**Palavras-chave:** IoT; Internet das coisas; Telecomunicações; Sistema de monitoramento Inteligente; ESP32, 5G.

NASCIMENTO, Leandro Lima. **Management and monitoring system for telecommunications outdoors rack's, based on Internet Of Things**. 2019. 33. Course Completion Work (Professional Master's in Computer Systems Engineering) – State University of Maranhão, São Luís, 2022.

## **ABSTRACT**

The present work aims to correlate concepts, technologies and application of a solution based on embedded systems, Internet Of Things and 5G, to monitor and ensure data persistence in telecommunications companies. First, a survey of the current scenario of the Telecommunications industry is carried out. In addition, they present intrinsic problems, in which consumers are the most affected, such as the lack of reliability and interactivity of the product offered by energy concessionaires. The bibliographic research was through a literary review based on articles, books and works by several authors from the period 2019 to 2022. The collection of information on the topic took place through the databases: Google Scholar and IEEE. In this way, conceptual information was presented on technologies such as ESP32, 5G, sensing and especially the MQTT protocol, for long-distance solutions, which belong to the concept of Internet of Things (IoT). Therefore, we propose the development of a device capable of monitoring and assisting telecommunications companies in the management to maintain their external assets.

**Keywords:** IoT; Internet of Things; Telecommunications; Intelligent monitoring system; ESP32, 5G.

## LISTA DE FIGURAS

Figura 1 – Arquitetura com suporte à conexão LTE .....	15
Figura 2 – Tendências do 5G de acordo com o pilar da sociedade .....	16
Figura 3 – Estrutura 5G para multicamadas. ....	17
Figura 4 - Arquitetura de 3 camadas para IOT. ....	18
Figura 5 – Classificação das tecnologias wireless em relação a velocidade, potência e distância. ....	19
Figura 6 – Cenário Operacional para o NB-IoT. ....	21
Figura 7 - Exemplo da arquitetura do MQTT. ....	22
Figura 8 – Especificações SIM7070G. ....	23
Figura 9 – Especificações Esp32 TTGO T-PCI-E. ....	25
Figura 10 – Vista real do Esp32 TTGO T-PCI-E. ....	27
Figura 11 – Afastamentos mínimos entre condutores da rede de telecomunicação e rede elétrica ao longo do vão. ....	29
Figura 12 – Diagrama em blocos em 3D da parte interna de um rack. ....	30
Figura 13 – Diagrama em blocos das conexões eletroeletrônica dos elementos. ....	32
Figura 14 – Diagrama em blocos do firmware. ....	34
Figura 15 – Diagrama em blocos da aplicação. ....	35
Figura 17 – Estrutura do código fonte do firmware. ....	36
Figura 18 – Condigo fonte para utilização do Portal de Configuração. ....	37
Figura 19 – Utilização do Portal de Configuração. ....	38
Figura 20 – Utilização do Portal de Configuração. ....	38
Figura 21 – Estrutura do endereçamento de rotas MQTT. ....	39
Figura 22 – Criação de uma Instancia MQTT. ....	40
Figura 23 – Funções para retornar rotas (endereçamentos). ....	40
Figura 24 – Função de resposta quando houver dados na subscrição do MQTT. ....	41
Figura 25 – Função para tratar reconexões no MQTT. ....	41
Figura 26 – Estrutura do endereçamento de rotas MQTT. ....	42
Figura 27 – Programação do sensor de Tensão AC com <i>EmonLib</i> . ....	42
Figura 28 – Captura de dados, conversão para JSON e envio via MQTT. ....	43
Figura 29 – Função de loop para verificação de conectividade MQTT. ....	43
Figura 30 – Função loop para envio de dados MQTT. ....	44
Figura 31 – Captura de dados, conversão para JSON e envio via MQTT. ....	44
Figura 32 – Circuito central do controlador, ESP32 na placa de circuito Impresso. ....	45
Figura 33 – Circuito das Portas Auxiliares. ....	46
Figura 34 – Circuito de recepção do sinal analógico do sensor de Tensão AC para o ESP32. ....	46
Figura 35 – Circuito de entrada de energia e regulação em níveis de 3.3 e 5Volts até 1A. ....	47
Figura 36 – Vista virtual de topo e fundo da placa de circuito impresso do dispositivo. ....	47
Figura 37 – Vista de topo e fundo da placa de circuito impresso do dispositivo. ....	48

## LISTA DE ABREVIATURAS

3GPP - 3rd Generation Partnership Project  
ANATEL - Agência Nacional de Telecomunicações  
ADC – *Analog to Digital Converter*  
AP – *Access Point*  
API - *Application Programming Interface*  
CoAP - *Constrained Application Protocol*  
DAC - *Digital to Analog Converter*  
eMBB — *enhanced Mobile Broadband*.  
FDD - *Frequency Division Duplex*  
FreeRTOS - *Real-Time Operating System kernel for microcontroller*  
FTP - *File Transfer Protocol*  
GIOP - *General Input Output Port*  
GSM - *Global System for Mobile Communications*  
HTTP - *Hypertext Transfer Protocol*  
INPI - Instituto Nacional de Propriedade Industrial  
ITU - *International Telecommunication Union*  
IoT – *Internet Of Things*  
JSON – *JavaScript Object Notation*  
LAN - *Local Area Network*  
Li-Ion – *Lithium Ion*  
LPWAN - *Low Power Wide Area Networks*  
LTE - *Long Term Evolution*  
MCom - Ministério das Comunicações  
MQTT - *Message Queuing Telemetry Transport*  
mMTC - *Massive Machine Type Communication*  
M2M – *Machine to Machine*  
NSA - *Non-Standalone*  
NB-IoT - *Narrowband Internet of Things*  
OLT - *Optical Line Terminal*  
OFDM - *Orthogonal Frequency Division Multiplexing*  
PAN - *Personal Area Network*  
RFID - *Radio-Frequency Identification*  
TDD - *Time Division Duplex*  
TSMC - *Taiwan Semiconductor Manufacturing Company*  
ULP - *Ultra Low Power*  
URLLC - *Ultra-Reliable Low Latency Communication*  
VSCode - *Visual Studio Code*  
WAN - *Wide Area Network*

## SUMÁRIO

<b>1</b>	<b>Introdução .....</b>	<b>12</b>
<b>2</b>	<b>Justificativa .....</b>	<b>13</b>
<b>3</b>	<b>Objetivos.....</b>	<b>14</b>
<b>3.1</b>	<b>Objetivo Geral .....</b>	<b>14</b>
<b>3.2</b>	<b>Objetivos Específicos.....</b>	<b>14</b>
<b>4</b>	<b>Fundamentação Teórica.....</b>	<b>15</b>
<b>4.1</b>	<b>Infraestrutura de Telecomunicações durante a evolução para o 5G.....</b>	<b>15</b>
<b>4.2</b>	<b>5G e a Internet das Coisas - IOT .....</b>	<b>18</b>
4.2.1	LPWAN.....	19
4.2.2	NB-IoT/CAT-M - LTE.....	20
4.2.3	MQTT.....	22
<b>4.6</b>	<b>Módulo LPWAN – SIM 7070G .....</b>	<b>23</b>
<b>4.8</b>	<b>ESP32 TTGO PCI-E .....</b>	<b>24</b>
<b>5</b>	<b>Trabalho Proposto.....</b>	<b>28</b>
<b>5.1</b>	<b>Metodologia.....</b>	<b>28</b>
5.1.1	A arquitetura do Sistema .....	28
5.1.2	O Hardware.....	31
5.1.3	O Algoritmo.....	33
5.1.3.1	Hardware .....	33
5.1.3.1	Aplicativo de gestão .....	34
<b>6</b>	<b>Desenvolvimento do Produto Mínimo Viável (MVP).....</b>	<b>36</b>
<b>6.1</b>	<b>Camada de Percepção .....</b>	<b>36</b>
6.1.1	Software.....	36
6.1.2	Hardware .....	45
<b>7</b>	<b>Considerações Finais .....</b>	<b>49</b>
	<b>REFERÊNCIAS .....</b>	<b>50</b>

## 1 Introdução

A revolução 4.0 propiciou avanços como a modernização da indústria e a digitalização da sociedade (SANTOS, ANDRADE, *et al.*, 2020). A humanidade busca mais conectividade, qualidade e rapidez na transmissão de dados. Dessa forma, as pequenas e grandes empresas de telecomunicações devem prestar serviços de forma contínua e ininterrupta, pois muitos serviços essenciais estão migrando para um modelo informatizado e conectado, seja por micro serviços ou por dados em tempo real.

O Brasil é a quinta maior população mundial, essa alta densidade demográfica é um desafio para prestadores de serviços de telecomunicações. Existem cerca de 17 mil entidades autorizadas ou não autorizadas, sendo em sua maioria prestadores de serviço de internet e/ ou tv por assinatura que utilizam cabeamento e transmissão por rádio (IBGE, 2021).

Nesse contexto, a pesquisa realizada em 2020 pela Agência Nacional de Telecomunicações – ANATEL, que visava quantificar a satisfação dos consumidores de serviços de telecomunicação, relatou que no quesito de funcionamento a média da nota atribuída pelos consumidores era de 6,45. Além disso, em 6 anos foram poucas as empresas que conseguiram nota acima de 7. Outro dado importante, é a capacidade de manter a conexão, onde mais de 60% dos pesquisados atribuem notas igual ou inferior a 7 (ANATEL, 2020). Por isso, é evidente a dificuldade dessas empresas na prestação de serviços de qualidade.

Um dos fatores determinantes para uma empresa prestadora de serviço de conexão à internet é sua escalabilidade outdoor, com lançamentos de fibras, cabos ou antenas para que seu serviço possa chegar ao maior número de clientes. Os rack's outdoors são ambientes físicos indispensáveis para as operadoras, pois é nele que equipamentos necessários para prestação de serviço ficam armazenados, tais como switch 's e *Optical Line Terminal* - OLT. Esses ambientes estão fora da matriz da operadora e servem como ambiente para garantir segurança contra eventos climáticos, furtos e outros eventos atípicos.

Portanto, é importante conhecer dentro os projetos de provedores: a topologia aplicada e o *layout* de serviços para o 5G. Com a entrada dessa nova tecnologia, devido ao surgimento do pico, micro e macro células de redes sem fio, a camada física das operadoras serão ainda mais descentralizadas da central (ITU, 2015).

Dessa forma, este estudo propõe o desenvolvimento de dois produtos: um hardware de aquisição e gerenciamento, aplicado à rack's outdoors, e um software de gestão centralizado para provedores de comunicação. Por fim, para realização utiliza-se técnicas de *Internet of Things* - IoT e recomendações da *International Telecommunication Union* - ITU.

## 2 Justificativa

A chegada da tecnologia 5G e o massivo consumo de dados que desencadeia uma série de exigências do mercado para que as operadoras melhorem sua infraestrutura, geralmente optando por comunicações via fibra ótica e rede sem fio, ocasiona o aumento do uso de equipamentos externos à matriz da operadora. Portanto, é de extrema importância garantir a persistência de sinais, serviços e segurança de dados transmitidos pelos canais e sistemas destas operadoras.

Atualmente os switch's e OLT's presentes na parte interna dos rack's outdoors, possuem gerenciamento externo de suas funções e dados de persistência de sinal. Ao pesquisar pelos termos “sistemas de controle e eficiência de rack's externos”, “dispositivos monitoramento de racks externos” e “gerenciamento de racks externos” no IEE Explorer, Google Scholar, Google Patents e no Instituto Nacional de Propriedade Industrial - INPI não foi possível obter resultados satisfatórios em relação a gestão térmica, temperatura e segurança destes rack's. Porém, existem tecnologias e técnicas que quando somadas poderão satisfazer a eficiência térmica, térmica e garantia de segurança a nível físico e isto posto, são necessários mais estudos que corroborem para essa temática.

Este trabalho vai utilizar estudos e aplicações de sensoriamento com técnicas de IoT para desenvolver um hardware e um software que visa gerenciar os rack's externos para obter eficiência e segurança dos equipamentos in loco de forma a garantir a prestação de serviços de comunicações para as operadoras, melhorar a manutenção e mão de obra técnica.

## **3 Objetivos**

### **3.1 Objetivo Geral**

Desenvolver um hardware e um software de gestão térmica e segurança de rack externo aplicando técnicas e tecnologias de IoT e IMT - Recommendation 2083-0.

### **3.2 Objetivos Específicos**

- Conhecer a Internet das Coisas e algumas tecnologias acessíveis;
- Desenvolver hardware de gestão de eficiência térmica, térmica e sistema de segurança física;
- Desenvolver Software de gestão e monitoramento centralizado de rack externo.

## 4 Fundamentação Teórica

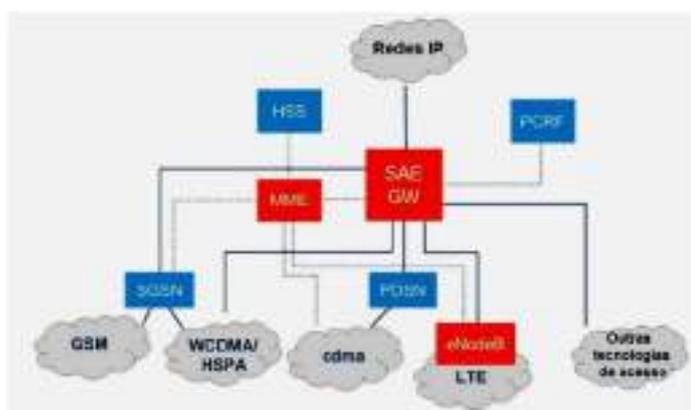
Este capítulo apresenta os principais conceitos relacionados a este trabalho, e distribui-se da seguinte forma: Infraestrutura de Telecomunicações durante a evolução para o 5G, Internet das Coisas, *Message Queuing Telemetry Transport* - MQTT, Microcontrolador ESP32, modulo SIM7070G, Sensores de Toque, Presença e Tensão e Corrente afim de almejar um equipamento voltado à segurança e controle de rack's.

### 4.1 Infraestrutura de Telecomunicações durante a evolução para o 5G

A infraestrutura padrão de um serviço de comunicações aplicado ao *Long Term Evolution* - LTE foi inicialmente desenhado em 2004 com o objetivo de migrar para a próxima fase do *3rd Generation Partnership Project* - 3GPP que visa a padronização de envio, recebimento e reprodução de arquivos multimídia para celulares e outros aparelhos que utilizam o *Global System for Mobile Communications* – GSM. (ALVES DE BRITO , 2011).

Segundo (ALVES DE BRITO , 2011) e (GOMES MOTA, CARVALHO, *et al.*, 2019) o 4G-LTE operam na tecnologia *Orthogonal Frequency Division Multiplexing* - OFDM e em modos *Frequency Division Duplex* – FDD ou *Time Division Duplex* - TDD. A figura 1 abaixo representa a arquitetura de conexão e suporte ao LTE.

**Figura 1 – Arquitetura com suporte à conexão LTE**



Fonte: (ALVES DE BRITO , 2011).

O LTE atualmente é a tecnologia de comunicação mais utilizadas nos dias de hoje com suporte à todos os aparelhos de telefonia móvel fabricados nos dias de hoje. E foi concebido como uma evolução do 3G, para proporcionar melhor desempenho, estabilidade, confiabilidade e eficácia nas conexões e suprir a demanda necessitada.

Como forma de evolução pensando no usuário (demanda e tendências) a ITU, maior autoridade mundial no campo das comunicações, elaborou uma série de recomendações desde 2012 com a visão para o “5G: recomendações e suas tendências” que estabeleceu um marco e uma largada para a evolução da telecomunicação como vimos hoje.

Tais tendências visam agregar e unificar a forma de se comunicar, não só por usuários humanos, mas também com máquinas (ITU, 2015). Os principais pilares dessas recomendações são:

- eMBB (*Enhanced Mobile Broadband* ou Banda larga móvel aprimorada) — Serviços de alta largura de banda para conectividade sem fio.
- URLLC (*Ultra-Reliable Low Latency Communication*, Comunicação de baixa latência ultraconfiável) — Comunicação ultraconfiável e de baixa latência para requisitos críticos.
- mMTC (*Massive Machine Type Communication* ou Comunicação do tipo de máquina massiva) — Comunicação confiável para bilhões de sensores e dispositivos de monitoramento.

Tais pilares foram desenhados de acordo com a característica de uso das principais áreas econômicas da sociedade, conforme a figura 2.

**Figura 2 – Tendências do 5G de acordo com o pilar da sociedade**



Fonte: Jupiter.net

Segundo (TAUFIQUE, JABER, *et al.*, 2017) a tendência da estrutura da rede de comunicações móveis serão projetadas ou são parcialmente projetadas para serem



## 4.2 5G e a Internet das Coisas - IOT

Em duas análises, a de Santos *et al.*(2016) e Veloso *et al.*(2017), conclui-se que a internet das Coisas (IoT) pode ser considerada uma extensão da atual internet com a ideia de conectar objetos do cotidiano às pessoas. Logo, o objetivo é de permitir uma interação simples e rápida entre objetos e o homem, por meio de sensores e cálculos computacionais para passar determinados estados ou resultados para o usuário.

O termo “coisas” trata-se de dispositivos ou tecnologias de *softwares* e *hardwares*, capazes de utilizar interfaces na qual possuem identidades e características virtuais próprias (MARTINS e ZEM, 2015). Ainda nesse conceito, observa-se que também estão integradas às redes telemáticas que criam um núcleo de interface para que o homem possa interagir com as máquinas e integrados de forma (SANTOS, SILVA, *et al.*, 2016).

O início da IoT é um tanto controverso, mas observa-se que, conforme o conceito abordado anteriormente, a IoT só é possível devido a descoberta da internet, pelos militares e universidades, em meados dos anos de 1950, durante a guerra fria (MARTINS e ZEM, 2015). A primeira menção da IoT se deu em meados de 1990 com a publicação de trabalho sobre *Radio-Frequency IDentification* - RFID, mas ganhou a atenção posterior com trabalhos relacionados a Redes Sem Fio (SANTOS, SILVA, *et al.*, 2016).

É preciso adotar uma arquitetura de três camadas para definir a IoT, pois de acordo com (MARTINS e ZEM, 2015) “Internet das coisas encontra-se atualmente em um plano mais conceitual e não existe uma arquitetura totalmente definida”, por não haver uma regra para área, o referido autor sugere duas arquiteturas a de três camadas e a de cinco camada (figura 1).

**Figura 4 - Arquitetura de 3 camadas para IOT.**



Fonte: (MARTINS e ZEM, 2015).

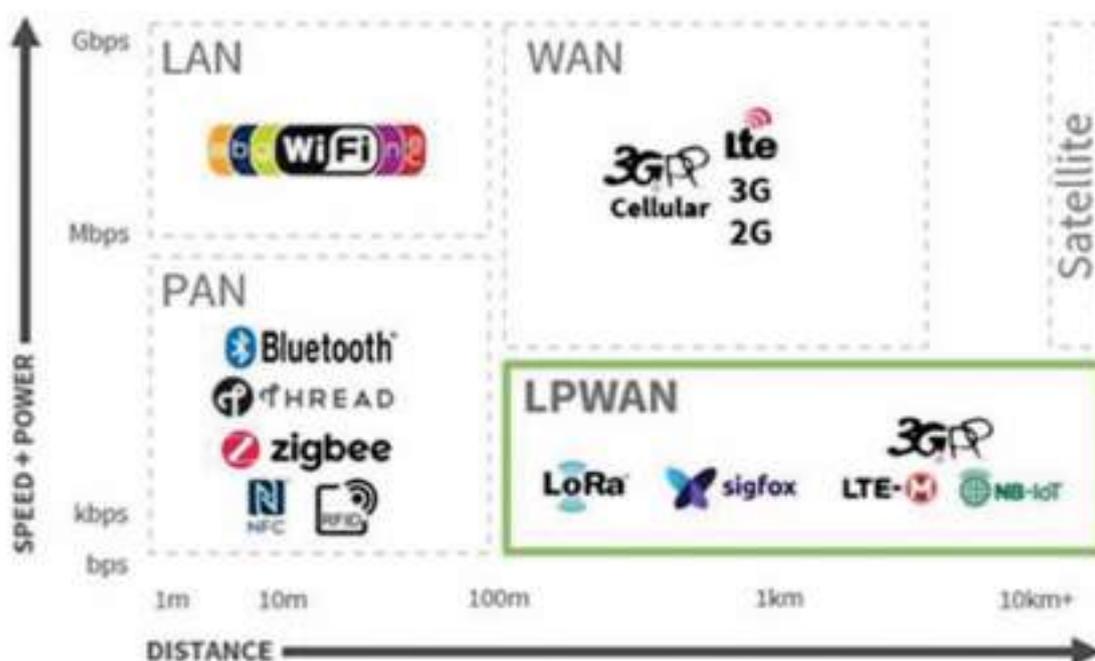
Parafrazeando Martins (2015) e Gerber (2017), a arquitetura de 3 camadas consiste: a camada de percepção, que está relacionada às tecnologias de sensores e hardwares de controle; a rede, que é a camada relacionada às tecnologias de transmissão de dados da camada de percepção pela rede de computadores; a camada de aplicação, que envolve todas as tecnologias de softwares (incluindo as de desenvolvimento para auxiliar nos projetos); e as

interfaces e soluções, que visam a interatividade com o homem. Seguindo este conceito da arquitetura da IoT será abordado alguns estudos de tecnologias atualmente utilizadas. Agora veremos algumas tecnologias de rede para a IOT-5G.

#### 4.2.1 LPWAN

Primeiramente, antes de abordarmos algumas tecnologias *Low Power Wide Area Networks* – LPWAN, devemos nos situar em quais são as atuais tecnologias sem-fio - *Wireless* para a comunicação de dados.

**Figura 5 – Classificação das tecnologias wireless em relação a velocidade, potência e distância.**



Fonte: (Y. MANE, 2021)

Como observado pela Y. Mane (2021), as tecnologias wireless são agrupadas em:

- *Local Area Network* – LAN: engloba a maioria das tecnologias sem fio para regiões de curto alcance porém de acesso compartilhado entre outros dispositivos, essas áreas geralmente são residenciais, corporativas, comerciais e industriais com velocidades que podem ser superiores a 1Mbps.
- *Personal Area Network* – PAN: é considerado um grupo de tecnologias pessoais, as taxas podem ser relativamente inferiores a 1Mbps e um alcance de menos de 100 metros.

- *Wide Area Network* – WAN: tecnologia com uma rede densa de dispositivos que por muitas vezes supera o acesso simultâneo de 256 dispositivos de longo alcance, este grupo comporta a maioria das atuais redes moveis (GSM, 3GPP e LTE).
- *Low Power Wide Area Networks* – LPWAN: comporta-se as tecnologias voltadas à uso massivo de *Machine to Machine* – M2M e tecnologias IoT. O principal foco de estudos e de uso para este trabalho.

Nos dias de hoje, umas das principais tecnologias aplicadas à IoT é a tecnologia LPWAN. Esta tecnologia, fomenta os requisitos de longo alcance, baixo consumo energético, capacidade do sinal RF de superar barreiras físicas com o mais baixo nível de atenuação possível, baixo espalhamento e baixo custo (RANGEL GARCIA e KLEINSCHMIDI, 2017), requisitos que as outras tecnologias não abrangem.

As primeiras estimativas são de custos de dispositivos variando entre 3 a 7 dólares, com pacotes entre 10 a 1000 bytes com uplink máximo de 200 kbps, baterias com duração de mais de 10 anos, área de cobertura entre 2 a 1000 km com suporte há mais de 100 mil dispositivos em uma rede operando em RF ISM (CHAUDHARI, ZENNARO e BOKAR, 2020). No Brasil, opera-se entre 902-907.5 e 915-928 MHz (HANES, 2017). No próximo tópico percorreremos em alguns conceitos sobre o NB-IoT.

#### **4.2.2 NB-IoT/CAT-M - LTE**

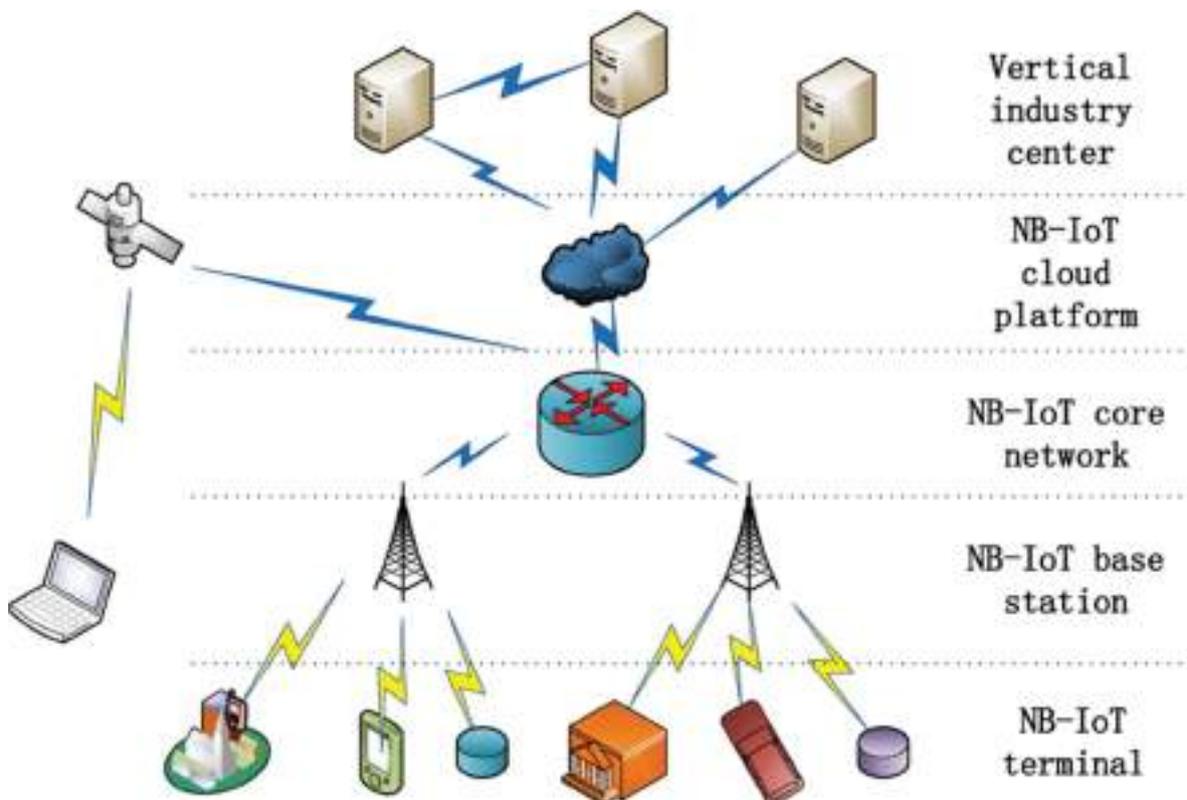
A *3rd Generation Partnership Project* - 3GPP está adaptando tecnologias para o mercado IoT e M2M. Essas oferecem coexistência com as tecnologias *Global System for Mobile* - GSM e LTE nas faixas licenciadas de 700 MHz, 800 MHz e 900 MHz (CHAUDHARI, ZENNARO e BOKAR, 2020) que conectam até 50 mil dispositivos, oferecendo 180 KHz de largura de banda (CHAUDHARI, ZENNARO e BOKAR, 2020). Porém, essa tecnologia não é compatível com o 3G (RAZA, KULKARNI e SOORIYABANDARA, 2020).

Devido às grandes necessidades para soluções de longo alcance e otimização do consumo de energia, o LPWAN tornou-se uma das tecnologias de comunicações mais bem vistas, devido ao seu potencial sugerido por (CHEN, LI, *et al.*, 2019). Com isso, houve abertura de novas tecnologias para esse processo de implementação.

Para o contexto do 5G e IoT, assunto abordado neste trabalho, surge uma tecnologia de transição das tecnologias de redes convencionais para o 5G, o chamado: *Narrow-Band Narrowband Internet of Things* (NB-IoT) .Segundo (CHEN, LI, *et al.*, 2019), pode ser

considerado uma rede definida por software com um sistema de banda estreita em LTE, sendo uma tecnologia emergente de acesso a rádio e que funciona em bandas de frequência autorizadas com taxas de transmissão *uplink* e *downlink* dentro de 256 kbits e com largura de banda limitada a 180 KHz. Isso conferi uma alta confiabilidade e forte anti-interferência que são principais características ausentes nas maiorias das tecnologias LPWAN.

**Figura 6 – Cenário Operacional para o NB-IoT.**



Fonte: (Chen, et al., 2019).

Operacionalmente o NB-IoT é dividido cinco grupos: o grupo usuário (*Terminal*), destinado ao dispositivo do consumidor final; as estações bases, responsáveis principalmente pela cobertura territorial; o Núcleo de Rede, responsável principalmente pela gestão de acesso e demanda; e a plataforma de nuvem, responsável por alguns protocolos de comunicação e o centro de acesso vertical industrial.

Vale ressaltar NB-IoT é mais aplicável a tecnologias sem fio de forma fixa, para comunicações móveis sugere-se utilizar o CAT-M, também da mesma grade familiar a LTE, sendo os principais benefícios da tecnologia a cobertura abrangente, baixo consumo de energia desde o terminal até ao serviço de nuvem, desenvolvido para conexões massivas de energia ultrapassando décadas, e até 50 mil conexões por célula.

### 4.2.3 MQTT

Entende-se que quando há uma comunicação de múltiplas máquinas computadores por exemplo, mais especificamente os microcontroladores, são necessárias regras (protocolo ou “linguagem global”) para que esses microcontroladores possam conversar de forma a amenizar problemas de comunicação. Alguns exemplos de protocolos amplamente utilizados são o FTP e HTTP que necessitam de um elevado poder de processamento, por isso eles são nativos em computadores. (MARTINS e ZEM, 2015; NIKOLOV, 2020).

O MQTT foi desenvolvido em meados da década de 90 pela IBM, para dispositivos mencionados anteriormente (microcontroladores), ao contrário dos protocolos nativos de computadores, o referido protocolo funciona por meio de uma arquitetura de *Publish/Subscribe* (publica/inscreve) voltado para redes inseguras de alta latência (lentidão) e baixa largura de banda. (YUAN, 2017; MARTINS e ZEM, 2015).

Figura 7 - Exemplo da arquitetura do MQTT.



Fonte: Brasizza (2018).

Existem 3 atores ou estados principais: o *Publisher* que envia uma informação, o *Subscribe* que recebe a informação e o *Broker* considerado o servidor do protocolo (figura 3). Eles são responsáveis por fazer todo o redirecionamento das informações (YUAN, 2017; LIMA, 2018). E o MQTT utiliza um sistema de árvores para publicar e inscrever em tópicos (NIKOLOV, 2020).

Um outro fator determinante, ainda observando na figura 7, é que as informações são categorizadas por *Topic*, ou seja, se um dispositivo envia uma informação para um *Topic* 1 por exemplo, somente aqueles que se inscreverem no referido *Topic* poderão receber tais informações (LIMA, 2018; MARTINS e ZEM, 2015).

#### 4.6 Módulo LPWAN – SIM 7070G

A série Sim70xx é uma tecnologia em LPWAN, desenvolvido pela *SIMCom Wireless Solutions Limited* – SimCom que atua para fornecer soluções *Machine to Machine* - M2M, módulos para comunicação sem fio, desenvolvido diretamente para aplicações em IoT, que necessitam de baixa latência, baixas taxas de transferência na comunicação e tenham suporte às tecnologias de protocolos desenvolvidos para M2M (SIMCOM, 2020).

O quadro a seguir retrata às principais características e níveis de desempenho aplicados ao módulo e requisitadas para soluções com base na IoT (ITU, 2015; SIMCOM, 2020; GERBER e KANSAL, 2017).

**Figura 8 – Especificações SIM7070G.**

<b>Grupo Recurso</b>	<b>Detalhes</b>	
<b>Bandas de Frequência</b>	CAT-M	B1/B2/B3/B4/B5/B8/B12/B13/ B14/B18/B19/B20/B25/B26/ B27/B28/B66/B85
	CAT-NB	B1/B2/B3/B4/B5/B8/B12/B13/ B18/B19/B20/B25/B26/B28/ B66/B71/B85
	GSM	850/900/1800/1900MHz
<b>Transferências (Kbps)</b>		
<b>CAT-M (up/down)</b>	589/1119	
<b>CAT-NB (up/down)</b>	127/158,5	
<b>EGPRS (up/down)</b>	296/236,8	
<b>Característica Elétrica</b>		
<b>Tensão de Operação</b>	3,0~ 4,6 V	Op: 30mW ~ 46mW Hib: 2,4mW ~ 3,6mW
<b>Corrente de consumo</b>	Operando: 10 mA Hibernação: 0,8 mA	
<b>Recursos</b>		

<b>Grupo Recurso</b>	<b>Detalhes</b>
<b>Protocolos</b>	TCP/UDP/HTTP/HTTPS/FTP/TLS/ DTLS/PING/LWM2M/ COAP/MQTT
<b>FOTA</b>	Presente
<b>AT incorporado</b>	Presente
<b>Atualizações de firmware</b>	USB/FOTA
<b>Interfaces</b>	
Cartão Sim, UART, USB, PCM/I2S, ADC, SPI, I2C, SIM, INCORPORADO	
<b>Certificações</b>	
<b>Regulatório</b>	FCC/IC/CE/RCM/JATE/ TELEC/NCC/ANATEL*/ GCF/PTCRB/RoHS/REACH

Fonte: (SIMCOM, 2020)

De acordo com o Spectrum-E (ANATEL) o Brasil opera para o 4G nas bandas B1, B3, B5, B7, B28 além das bandas para o GSM. A tecnologia Sim7070, considera também as certificações globais e nacionais é mais que suficiente para aplicações e operação no Brasil.

#### 4.8 ESP32 TTGO PCI-E

Para aplicações de IoT e as soluções em controle e monitoramento, torna-se necessário o uso de microcontroladores. Esses dispositivos têm a principal função de serem programáveis de acordo com o projeto ou solução almejada.

O ESP32 é um microcontrolador fabricado pela *Espressif Systems* - SPRESSIF, uma multinacional chinesa, fundada em 2008 com escritório no Brasil. A empresa é focada em tecnologias de dispositivos (Circuitos Integrados) voltados para IoT, o seu principal produto são os microcontroladores ESP32 fabricado por uma das maiores empresas mundiais de manufatura de chips a *Taiwan Semiconductor Manufacturing Company* - TSMC.

O microcontrolador ESP32 é indicado para aplicações IoT, pois de acordo com a figura 5, o mesmo possui as características e exigências da IoT e hoje é mundialmente utilizado no comércio de equipamentos inteligentes e em prototipação de projetos.

**Figura 9 – Especificações Esp32 TTGO T-PCI-E.**

<b>Hardware Specifications</b>	
<b>Chipset</b>	ESP32-WROVER-B, AXP192 Power Management Chip
<b>FLASH</b>	4MB
<b>PSRAM</b>	8MB
<b>SRAM</b>	520 kB SRAM
<b>Button</b>	PWR Power Switch, Reset
<b>USB to TTL</b>	CP2104
<b>On-board clock</b>	40MHz crystal oscillator
<b>Peripherals</b>	SIM Nano Card Slot, Mini PCIE Card Slot
<b>Antenna</b>	Onboard antenna (Default)
<b>Working voltage</b>	2.7V-3.6V
<b>Working current</b>	
<b>Sleep current</b>	About 700uA
<b>Working temperature range</b>	-40°C ~ +85°C
<b>Size</b>	75*33*10.5mm
<b>Weight</b>	14.5g
<b>Power Supply Specifications</b>	
<b>Power Supply Input</b>	USB 5V/1A
<b>Charging current</b>	500mA
<b>Battery Input</b>	3.7-4.2V
<b>Charge IC</b>	AXP192

<b>JST Connector</b>	2Pin 2.0mm
<b>USB</b>	Type-C
<b>Wi-Fi</b>	
<b>Standard</b>	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC (ESP32 Chip)
<b>Protocol</b>	802.11 b/g/n(802.11n, speed up to150Mbps)A-MPDU and A-MSDU polymerization, support 0.4μS Protection interval
<b>Frequency range</b>	2.4GHz~2.5GHz(2400M~2483.5M)
<b>Transmit Power</b>	22dBm
<b>Communication distance</b>	300m
<b>Bluetooth</b>	
<b>Protocol</b>	Meet bluetooth v4.2BR/EDR and BLE standard
<b>Radio frequency</b>	With -97dBm sensitivity NZIF receiver Class-1,Class-2&Class-3 emitter AFH
<b>Audio frequency</b>	CVSD&SBC audio frequency
<b>Software specification</b>	
<b>Wi-Fi Mode</b>	Station/SoftAP/SoftAP+Station/P2P
<b>Security mechanism</b>	WPA/WPA2/WPA2-Enterprise/WPS
<b>Encryption Type</b>	AES/RSA/ECC/SHA
<b>Firmware upgrade</b>	UART download/OTA (Through network/host to download and write firmware)
<b>Software Development</b>	Support cloud server development /SDK for user firmware development

<b>Networking protocol</b>	IPv4、IPv6、SSL、TCP/UDP/HTTP/FTP/MQTT
<b>User Configuration</b>	AT + Instruction set, cloud server, android/iOSapp
<b>OS</b>	FreeRTOS

Fonte: (LILYGO, 2022).

Uma das vantagens da presente placa de desenvolvimento é a possibilidade de ser mesclada em placa de aplicação através da soldagem de seus pinos, e a possibilidade de upgrade ou *downgrade* da placa de interface através do socket *mini-pci express* presente no topo da placa conforme figura 10.

**Figura 10 – Vista real do Esp32 TTGO T-PCI-E.**



Fonte: (LILYGO, 2022).

É possível observar dentre suas especificações e suas características, ser um componente de notória capacidade de auxílio no desenvolvimento de novas tecnologias de IoT. Dessa forma, contém especialmente recursos como uma camada de abstração de programação para linguagens mais utilizadas na área como o C/C++, Arduino e JavaScript, para criação de aplicações internas do microcontrolador.

Possuem também, camadas de abstração que podem agilizar, facilitar e melhorar funcionalidades e processos do trabalho proposto, sendo também comuns na área de desenvolvimento de softwares e firmwares.

## **5 Trabalho Proposto**

Tendo em vista as dificuldades das operadoras em garantir a persistência das comunicações devido a problemas de sobreaquecimento, quedas térmicas e rack inseguro, devido à localização nas ruas. Esse trabalho propõe o desenvolvimento de um dispositivo composto por sensores de temperatura, tensão, corrente e um sistema digital de trava que se comunica via protocolo M2M de Internet das Coisas com uma aplicação de servidor na matriz.

### **5.1 Metodologia**

Através da pesquisa documental e referenciais bibliográficos é possível selecionar as tecnologias adequadas para aplicação na presente proposta, tais como o uso de uma placa de desenvolvimento embarcado em uma placa aplicada; o uso de tecnologias do MQTT em conjunto com uma *Application Programming Interface* - API facilitando assim a interface com uma aplicação web, além de utilizar as camadas de Aplicação e Percepção de Internet das Coisas. O trabalho visa também teste de campo em algumas empresas parceiras regionais.

Em cada camada, será aplicado os critérios de Interatividade, capacidade de adaptação, previsibilidade, otimização, integração e por fim segurança de dados. Por fim, como forma de validação será utilizado dados de feedback das operadoras parceiras.

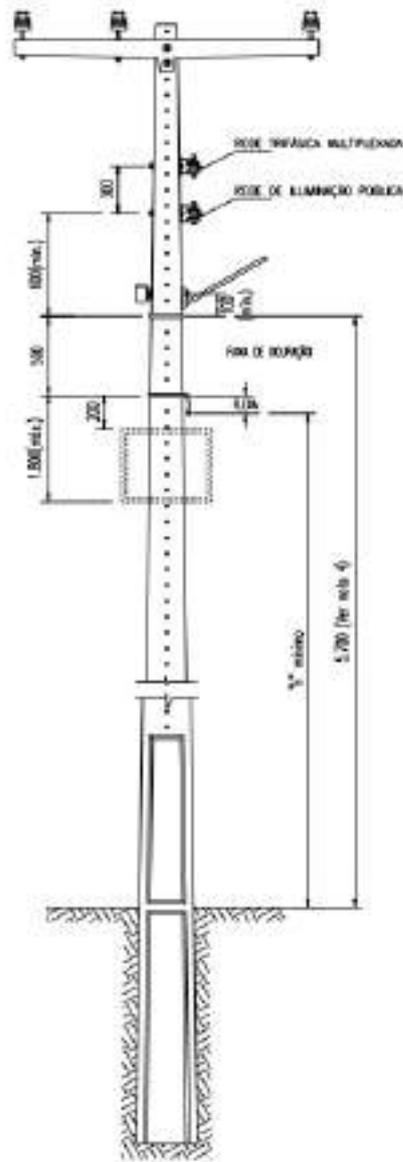
O desenvolvimento da tecnologia se divide em Arquitetura do Sistema, Arquitetura de hardware e arquitetura de Software. Cada etapa será melhor descrita e explicada ao decorrer do presente trabalho.

#### **5.1.1 A arquitetura do Sistema**

Nessa etapa identifica-se o ecossistema da tecnologia, quais elementos serão os principais e quais os elementos eletivos.

Os racks de rede ou gabinete encontram-se em sua maioria em postes de transmissão de rede elétrica a uma distância de pelo menos 1,3m entre o topo do rack e a rede de iluminação ou 0,2m a partir da flecha, sendo melhor observado na figura 11 (EQUATORIAL ENERGIA, 2022).

**Figura 11 – Afastamentos mínimos entre condutores da rede de telecomunicação e rede elétrica ao longo do vão.**



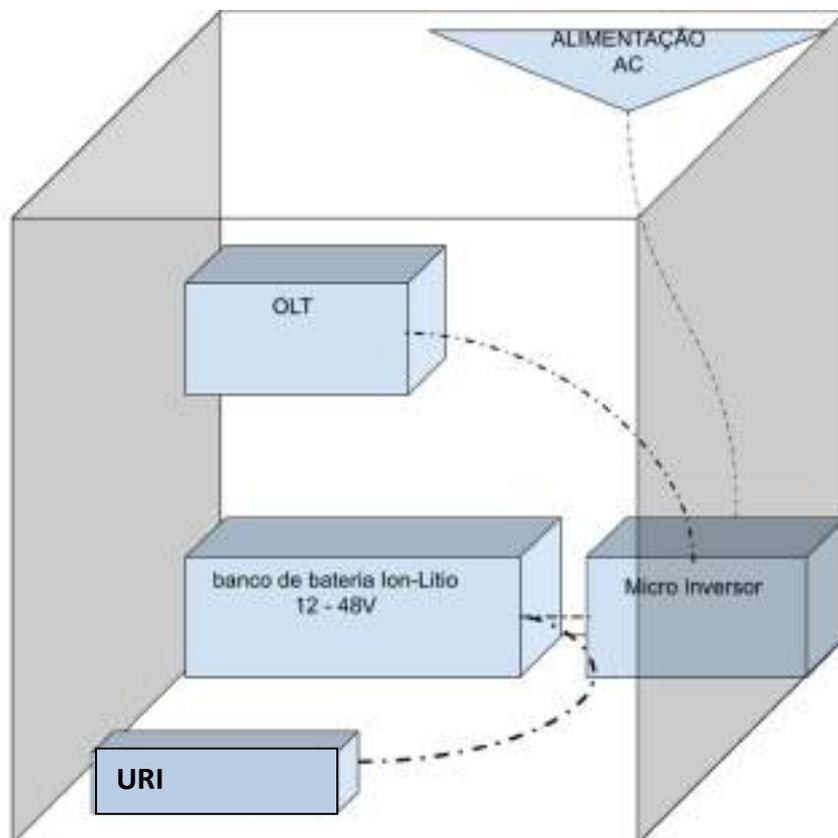
**Fonte:** (EQUATORIAL ENERGIA, 2022).

É possível observar que estes racks ficam expostos no ambiente, sendo acessível com apenas uma escada, é importante ressaltar também que é comum um rack possuir ao menos:

- Um ponto de alimentação AC;
- Um equipamento de transmissão/recepção de dados, por exemplo *Optical Line Terminal* – OLT;
- Um multiplexador de rede, por exemplo switch de rede;
- Uma ventoinha para resfriamento.

A figura 12 apresenta um diagrama em blocos em 3D possibilitando uma visão da localidade e interface entre a tecnologia e o trabalho proposto.

**Figura 12 – Diagrama em blocos em 3D da parte interna de um rack.**



Fonte: Autor.

Conforme a figura 7 e também em levantamento in loco, é possível observar que há além dos elementos comuns e de suas interligações elétricas, que algumas empresas da região de São Luís – MA usam como uma forma de solução rápida para situações de falta de energia um conjunto de banco de baterias (em sua maioria baterias de Chumbo-Ácido) e um micro inversor. Esses, são usados para atuar como banco de energia de backup, convertendo a corrente contínua da bateria em corrente alternada através do micro inversor de energia, os mesmos presentes em sistemas elétricos fotovoltaicos e veiculares.

Entretanto, nota-se a falta de uma padronização para essa conduta que fica a cargo de um técnico efetuar configuração e parametrização desse elemento em específico. Além disso, foi possível notar que as baterias de ácido-chumbo, as mesmas presentes em veículos e nobreaks, não são eficientes para cargas de trabalho acima de 3 horas, tempo esse médio entre a recepção de um chamado técnico e sua resolução de acordo com o painel de dados (ANATEL, 2020). Além disso, há relação entre a capacidade de carga e o tamanho ou peso

excessivo da célula de bateria, já que o rack é um espaço limitado e com um limite de carga definido no projeto e na instalação no poste de transmissão.

O backup de energia ou banco de energia recomendado para esse tipo de aplicação são os mesmos usados em veículos elétricos, as células de Li-Ion, validados nestes veículos, através autonomia de carga e descarga. Para a aplicação de alimentação no contexto desses entaves, o uso desse tipo de bateria poderia garantir autonomia desses pontos de distribuição ou recepção de dados por mais tempo, sendo aplicado até em empresas que atuam em múltiplos municípios com interligações via extraterritorial, onde o tempo de atendimento para resolução nestes pontos pode ultrapassar 24 horas. (FAN, LI, *et al.*, 2020)

Já a solução do presente projeto pode ser instalada na parte interna do rack, local definido pelo projeto da empresa. O monitoramento mais direto é com sistema de alimentação externa e através de leituras de energia (corrente, tensão, potência) do backup de energia, já que a energia é de extrema importância para estes equipamentos estarem online.

As funcionalidades previstas mais importantes são:

- Monitoramento de temperatura (através de sensoriamentos e controle de Fan's);
- Leitura de tensão e corrente tanto da alimentação externa quanto de backup de energia;

Com essas finalidades, o equipamento se comunica diretamente com uma central de operação através de um aplicativo centralizador para multiequipamento de telemetria intercomunicação realizada através de protocolos M2M (MQTT, API ou CoAP).

Nos próximos capítulos trataremos especificamente do hardware (elementos internos e suas interfaces) e do software (lógica de programação tanto para o hardware quanto para aplicação web).

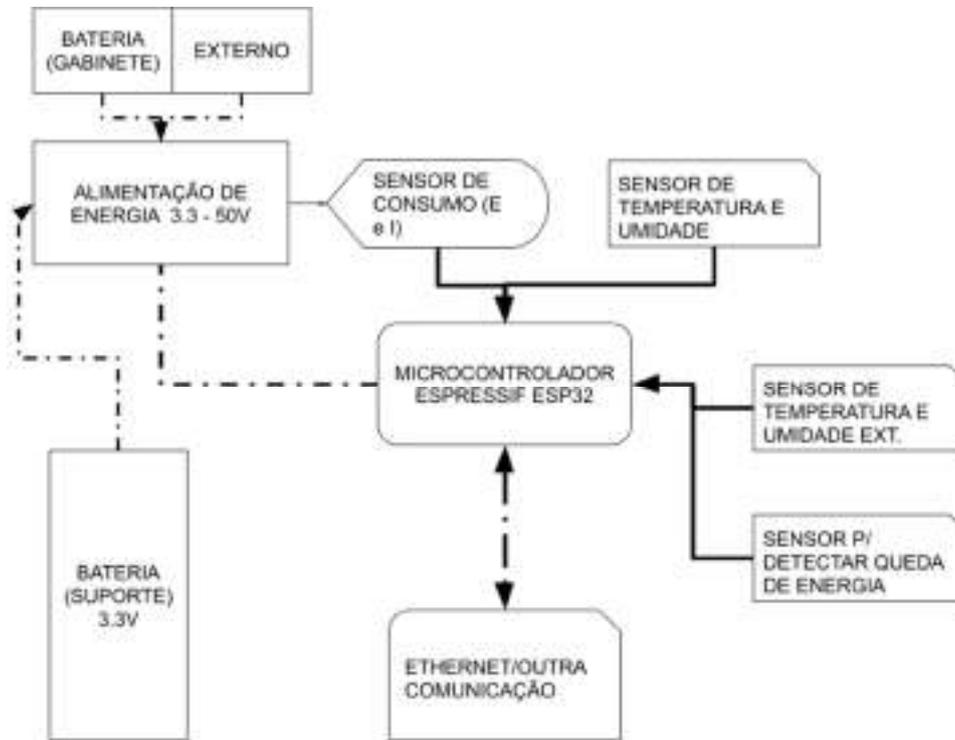
### **5.1.2 O Hardware**

Neste capítulo é descrito e analisado os elementos de hardware proposto, onde servirá como equipamento de monitoramento telemétrico para normatização e resolver os problemas notados anteriormente.

Na figura 13 é possível observar setores e elementos que dispõe do equipamento proposto em um modelo padrão já aplicável em circuitos de microcontroladores. Eles se apresentam com estes elementos o uso de um setor destinado à alimentação capaz de absorver tensões entre 3.3 e 50V além de um controle independente para tão somente a energia, também

agregasse sensoriamentos para energia de backup do rack, alimentação AC (externo) e um backup de bateria dedicado ao microcontrolador, podendo enviar informações de tensão e corrente para o microcontrolador, sendo informações cruciais para vitalidade do ecossistema do rack.

**Figura 13 – Diagrama em blocos das conexões eletroeletrônica dos elementos.**



**Fonte: Autor.**

Outro principal elemento é o microcontrolador, o Espressif ESP32 de 2 núcleos de 250Mhz robusto, com consumo médio em 224mW e com recursos *Ultra Low Power* – ULP que faz o microcontrolador operar em modo *Deep-Sleep* (sono profundo) com um co-processor usado como despertador para gatilhos externos. Ou seja, quando há uma alteração nas entradas digitais com o uso desse recurso o microcontrolador ganha  $\pm 400\%$  de autonomia operando em  $495 \mu W$ , tornando-se ideal para aplicações de baixo contato humano ou para regiões longínquas que há escarces de energia elétrica. (LILYGO, 2022; VELOSO, SOUSA, *et al.*, 2017)

O equipamento dispõe do uso de sensores de corrente DC ACS712 baseado em corrente linear com efeito hall e isolamento de tensão de 2,1 KVRms, para as baterias de backup do rack do próprio equipamento e proteção do equipamento, o uso dessa informação é recepcionada pelo microcontrolador através de sinal analógico que destina-se à vitalidade dos backups e informações necessárias para o operador e suas tomadas de decisões ou planejamento.

Um sensor utilizado pelo referente equipamento para análise do ambiente é o sensor de temperatura DHT22 que possui uma faixa de alimentação de 3,3 à 6V e opera a 2,5mA, realizando uma breve conversão para consumo estima-se a 8,25 mW. Este sensor possui a capacidade de leitura em temperaturas que variam de -40 a 80°C, precisão de  $\pm 0,5^\circ$  e uma taxa de amostragem de 0,5Hz. Há também um sensor extra (EXT) que serve como auxiliar uma medição externa de temperatura.

O sensor de queda de energia é um circuito simples de comutação de relé, na qual permanecerá ativo enquanto energizado externamente (AC), na falta de presença de energia comutará para o estado normalmente aberto, sendo tratado pelo microcontrolador, informando a central de operações e reencaminhando o uso da energia para o backup de baterias. O relé pode ser simplesmente alterado para um circuito com optoacoplador e tríodo de corrente alternada - TRIAC.

A seguir veremos a algoritmo lógico por trás da aplicação do hardware e do software.

### **5.1.3 O Algoritmo**

Este capítulo é destinado à técnica aplicada em dois tópicos, um para a lógica de elaboração do firmware que será responsável pelas tarefas do microcontrolador e outra para lógica de elaboração da aplicação web destinada às funcionalidades da central de operação e monitoramento técnico.

#### **5.1.3.1 Hardware**

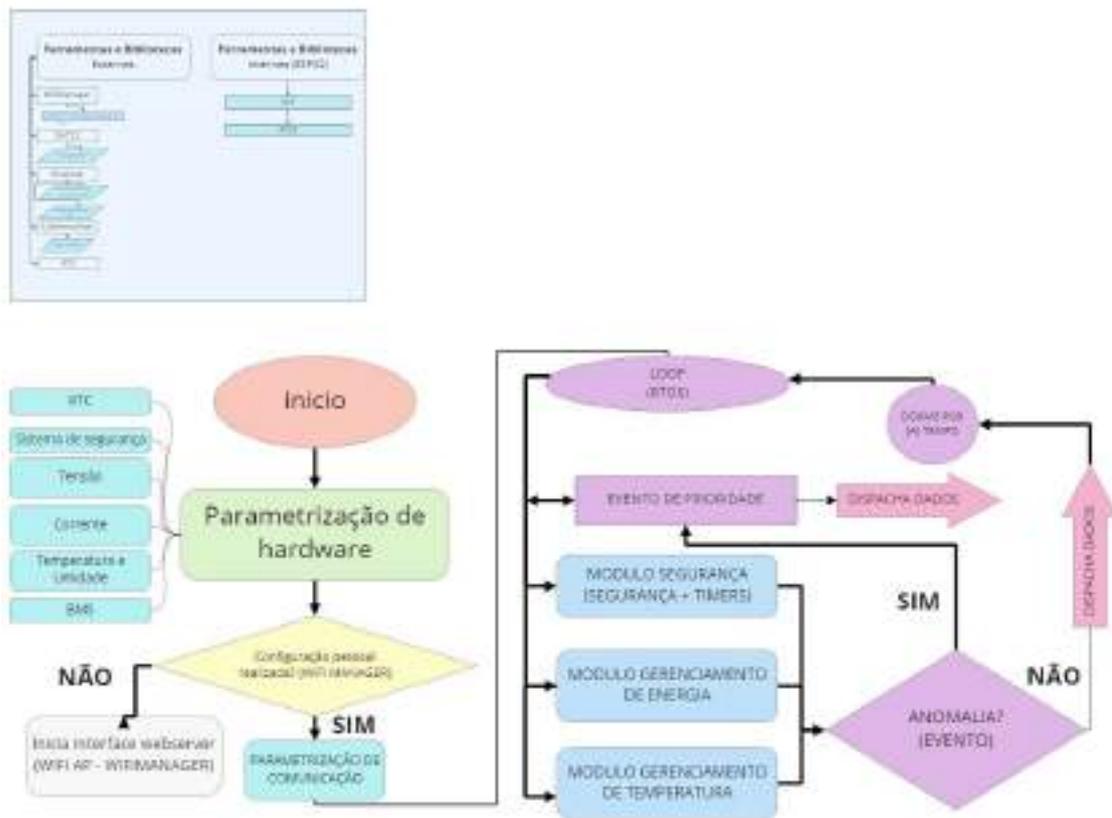
O algoritmo do hardware é definido intrinsecamente como layout de gatilhos, onde o microprocessador estará operando em modo ULP, ou seja, dormindo e a cada tempo definido pela empresa ele acordará para realizar checagens de seus periféricos, ou até que um dos periféricos ofereça algum sinal alterado fazendo com que acorde emergencialmente para tratar gatilhos, seja para uma ação de notificação seja para ação de reações como ligar fans.

O modo ULP, utiliza-se da técnica de uso de um RTC presente interno do microprocessador e independente e um núcleo de baixo poder de processamento, sendo reativo a reações das portas digitais (ADC).

Para facilitar as configurações e parametrização do equipamento e afim de deixá-lo eficiente e com poucos elementos periféricos, usa-se como interface o uso do websockets presente em uma biblioteca para o ESP32 chamada WIFIManager que cria um serviço local web, que pode-se utilizar como interface rápida para configuração.

Definindo aspectos de sensores presentes, tarefas a serem efetuadas, configuração de identificação do servidor, aparelho e tempos de atividades e até mesmo definir parâmetros de ULP, além de parâmetros de gatilhos para despachos de dados figura 14.

**Figura 14 – Diagrama em blocos do firmware.**



Fonte: Autor.

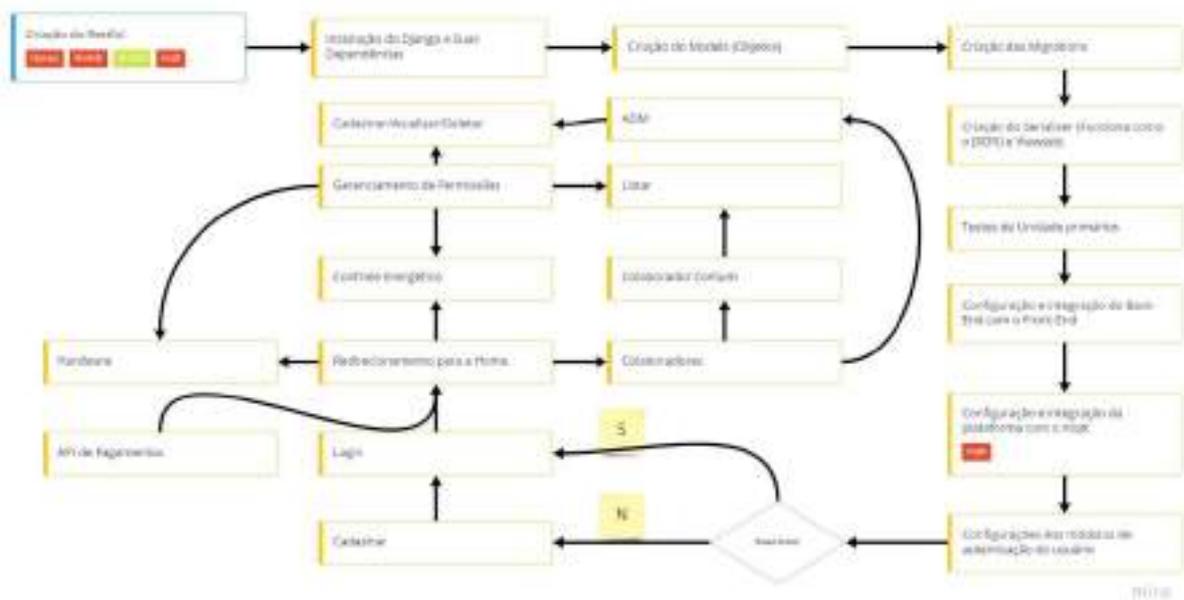
Para despacho de dados, pode-se utilizar pelo menos dois protocolos de comunicação: o MQTT ou WebSockets (GET, POST) enviando dados de pacotes em formato JSON, um padrão já definido na área web. Esses dados serão obtidos em um serviço de armazenamento em massa e injetados em uma aplicação web, visto a seguir.

### 5.1.3.1 Aplicativo de gestão

Este tópico define a camada de aplicação da solução, nesta camada é definido por um sistema web de gestão ou dashboard para monitoramento e controle em massa desta presente solução. Essa aplicação é definida por micro serviços tais como:

- Um serviço de banco de dados SQL para registros e consultas de dados da própria aplicação
- Um serviço de banco de dados NoSQL para armazenamento de dados massivos em tempo real dos equipamentos que operam via broker MQTT ou API.
- Um serviço Broker ou API, broker em caso de uso massivo via MQTT, ou API em requisitos de Web Sockets sendo definido pelo usuário (empresa).
- Um serviço web onde destina-se à interface de dashboard.

**Figura 15 – Diagrama em blocos da aplicação.**



**Fonte: Autor.**

Conforme Figura 15, a aplicação contém todas as regras de comunicação máquina-homem. Contendo regras de autenticação Web já padronizada OAuth, além de gestão de rotinas e módulos, contendo uma área panorâmica geral e outra para visualização de dados de um só equipamento.



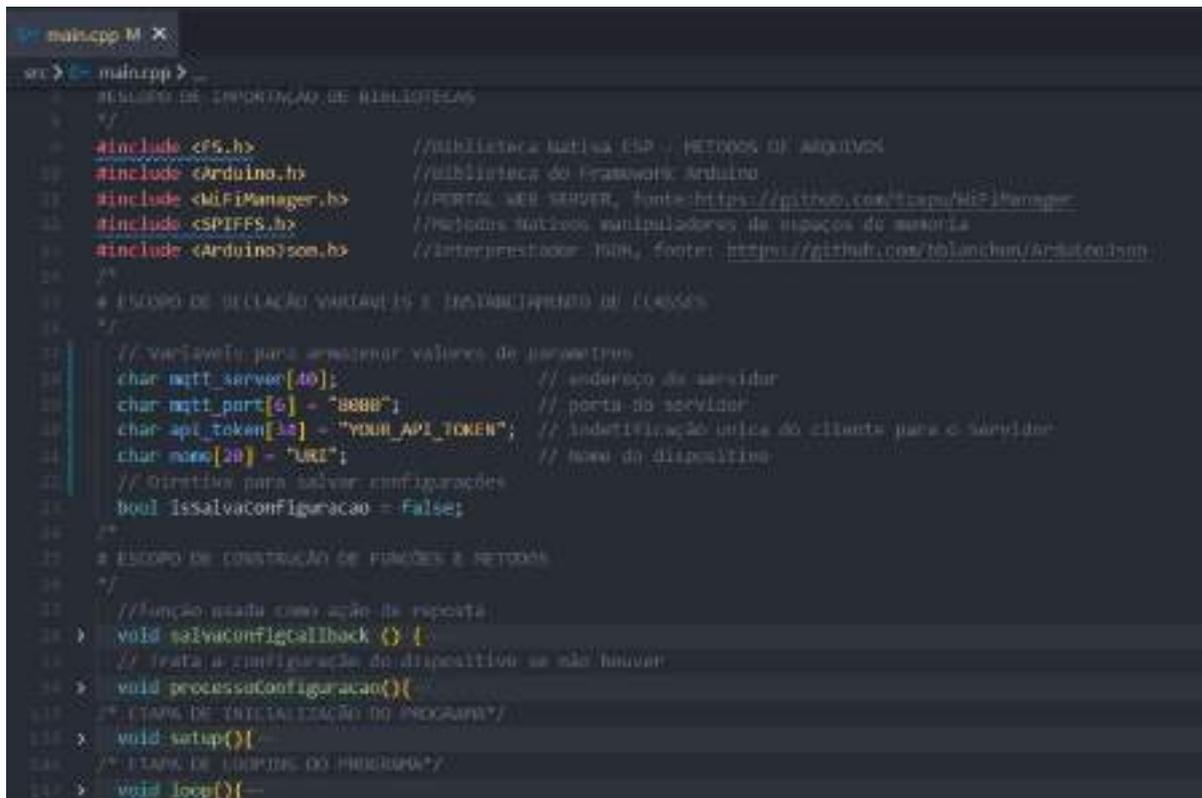
**Fonte: Autor.**

O primeiro recurso que este dispositivo deverá ter é uma interface e para isso podemos usar um recurso nativo do microcontrolador ESP32, a de criar uma interface via Web Server Local no próprio controlador. Já que este possui WIFI em modo AP (ponto de acesso), podemos conectar nossos dispositivos (celulares) sem precisar de fio ou conexão extra. Assim, dispensaremos periféricos como teclado e display.

Vale ressaltar que essa configuração deve ser realizada ao inicializar o dispositivo pela primeira vez ou quando resetar em seu padrão de fábrica.

Na programação podemos utilizar a biblioteca WifiManager que é um gerenciador de conexão WIFI com portal de configuração web para ESP32 e ESP8266. Na raiz da aplicação após instalação da biblioteca via *Platformio*. É necessário importá-lo para nosso arquivo raiz, que neste caso é chamado de “main.ccp” e chamar as seguintes instruções:

**Figura 18 – Código fonte para utilização do Portal de Configuração.**



```
main.cpp M X
src > main.cpp >
// ESCOPO DE IMPORTAÇÃO DE BIBLIOTECAS
//
1 #include <ESP.h> //Biblioteca nativa ESP - MÉTODOS DE ARQUIVOS
2 #include <Arduino.h> //Biblioteca do Framework Arduino
3 #include <WifiManager.h> //PORTAL WEB SERVER, fonte:https://github.com/tzapu/WifiManager
4 #include <SPIFFS.h> //Métodos nativos manipuladores de espaços de memória
5 #include <ArduinoJson.h> //Interpretador JSON, fonte: https://github.com/tblachau/ArduinoJson
//
6
7 # ESCOPO DE DECLARAÇÃO DE VARIÁVEIS E INSTANCIAMENTO DE CLASSES
8
9 // Variáveis para armazenar valores de parâmetros
10 char mqtt_server[40]; // endereço do servidor
11 char mqtt_port[5] = "8888"; // porta do servidor
12 char api_token[32] = "YOUR_API_TOKEN"; // identificação única do cliente para o servidor
13 char name[20] = "URL"; // nome do dispositivo
14 // Diretiva para salvar configurações
15 bool salvaconfiguracao = false;
16
17 # ESCOPO DE CONSTRUÇÃO DE FUNÇÕES E RETORNOS
18
19 //Função usada conexão de resposta
20 void salvaconfigcallback () {
21 // trata a configuração do dispositivo se não houver
22 void processconfiguracao(){
23 /* ETAPA DE INICIALIZAÇÃO DO PROGRAMA?
24 void setup()
25 /* ETAPA DE LOOPING DO PROGRAMA?
26 void loop(){--
```

**Fonte: Autor.**

Como observado na figura 18, precisaremos acrescentar algumas variáveis para armazenamento dos dados que desejamos, como por exemplo dados de conexão com Broker MQTT e o nome do dispositivo. Além das variáveis, precisamos de duas funções uma, chamada de “salvarConfigCallback” usada para chamada toda vez que executar uma configuração e passar a informação (booleana) para realizar o armazenamento ou leitura dos dados em um arquivo JSON, e outra chamada “processoConfiguração”, que instrui todo o

procedimento de leitura, armazenamento e apresentação do portal de configuração, toda vez que o controlador for inicializado, de acordo com a figura 19. Como resultado, o ESP32 estará configurando seu modo de operação em Ponto de acesso, para que outro dispositivo de preferência possa se conectar e realizar as devidas parametrizações.

**Figura 19 – Utilização do Portal de Configuração.**



Fonte: Autor.

Ao conectar podemos assim acessar o portal, como demonstra a figura 20. Vale ressaltar que nos computadores e smartphones em sua maioria, possuem um redirecionamento automático para conectar com o ponto de acesso. E ao clicar em Configure WiFi, abrirá uma tela com os parâmetros disponíveis para personalização do técnico.

**Figura 20 – Utilização do Portal de Configuração.**



Fonte: Autor.

Com essas funcionalidades iniciais, já temos um nível de personalização para dá ao usuário final liberdade de customização e confiabilidade, uma vez que a arquitetura de comunicação MQTT estará com um endereçamento raiz encriptado. Isso ocorre através do conteúdo Cliente Token, sem seguida do endereço definido pelo nome do dispositivo que pode ser também uma nomenclatura única. Dessa maneira, define-se assim uma rota MQTT de publicação ou subscrição, como observado na figura 21.

**Figura 21 – Estrutura do endereçamento de rotas MQTT.**

```
// endereço raiz,
"api_token/none/"

// para publicação de dados
"api_token/none/enviar/dados" //-> conteúdo recebido JSON em txt

// para publicação "Estou vivo!"
"api_token/none/status" //-> Envia ao menos uma vez por dia, um sinal de está vivo

// para subscrição (Recepção de dados)
"api_token/none/receber/dados" //-> endereço de subscrição de dados (de onde irá receber dados)

//Exemplo
"YOUR_API_TOKEN/URI/status" // -> Para saber se o dispositivo está conectado e operante
```

Fonte: Autor.

Com essa estrutura de endereçamento, tornamos os dados rastreáveis, previsíveis e flexíveis para implementação e uso pelo cliente. Por exemplo, se houver um segundo cliente, em meu broker e no serviço de aplicação, ele terá uma raiz “api\_token” único só raiz para cada empresa (cliente), calculando uma estimativa limite de clientes através da formula de combinação composta:

$$C_k^n = \frac{(n + k - 1)!}{k! (n - 1)!}$$

Onde  $n$  é o número máximo de elementos para a variável *api\_token*, definido em sua declaração no código como sendo de tamanho 34. E  $k$  a quantidade de caracteres possíveis, considerando apenas o conjunto alfanumérico, sem levar em conta distinção de maiúsculo e minúsculo, temos um total de 36 caracteres.

$$C_{36}^{34} = \frac{(34 + 36 - 1)!}{36! (34 - 1)!}$$

Assim teremos como resultado entorno de  $5,3 * 10^{19}$  de clientes únicos, baseado no número de tokens possíveis. E teremos  $2,8 * 10^{14}$  de dispositivos únicos por cliente, ao recalcular baseado  $n$  equivalente a 20, que corresponde ao tamanho máximo da variável *nome*.

Outro recurso necessário para desenvolvimento desta solução, é o uso da biblioteca *pubSubClient* que possui todos os métodos necessários para utilização do protocolo MQTT. Vale ressaltar que esta classe, herda uma outra classe, com padrões de conexão internet de baixa camada.

**Figura 22 – Criação de uma Instancia MQTT.**

```
#include <PubSubClient.h> //CLASSE MQTT

WiFiClient wifiCliente; // cria nova classe wifi client
PubSubClient mqtt(wifiCliente); //herda classe wifi
```

Fonte: Autor.

Como estruturado na figura 21, podemos usar funções que retornem tipo *string* para facilitar o reuso durante o processo de codificação, e efetuar as publicações e subscrições para os locais corretos (figura 23).

**Figura 23 – Funções para retornar rotas (endereçamentos).**

```
// função para padronizar e retornar endereçamento correto
String rotaEnviar(){
    return ""+String(api_token)+"/"+ String(nome)+"/enviar/dados";
}
// função para padronizar e retornar endereçamento correto
String rotaReceber(){
    return ""+String(api_token)+"/"+ String(nome)+"/receber/dados";
}
```

Fonte: Autor.

No desenvolvimento de códigos para trabalhar com envio e recebimento de dados, é muito comum o uso de funções conhecidas como *callback*, métodos de funções que retornam como resposta outra função, contendo uma serie de tarefas e procedimentos. Esse tipo de recursividade é útil para tratar cargas uteis e condicionar respostas.

Como visto na figura 24, a função *callback* para o MQTT irá concatenar a carga útil (mensagem) recebida, nos tópicos subscritos, e como no exemplo quando a carga útil for do tópico de recebimento de dados, o controlador mostrará uma mensagem na interface seria, mas poderíamos facilmente realizar um novo tratamento condicional para acionar um relé de uma ventoinha por exemplo.

**Figura 24 – Função de resposta quando houver dados na subscrição do MQTT.**

```
// função usada como ação de resposta para mqtt
void callbackMQTT(char* topic, byte* message, unsigned int length) {
  Serial.print("Mensagem Recebida: ");
  Serial.print(topic);
  Serial.print(". Conteúdo: ");
  String messageTemp;

  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp += (char)message[i];
  }
  Serial.println();
  //SE RECEBER O CORRESPONDENTE
  String rota = rotaReceber();
  if (String(topic) == rota) {
    //recebeu dados, criar conversão de JSON
    Serial.println("OK");
  }
}
```

Fonte: Autor.

Além de um *callback* de resposta, precisa-se lembrar que por situações adversas pode ocasionar percas de conexão entre o dispositivo e o broker, uma vez que estes dependem da conexão com a internet. Portanto define-se uma função para realizar tentativas de reconexões com o broker, afim de garantir a persistência de dados.

**Figura 25 – Função para tratar reconexões no MQTT.**

```
//Reconectar MQTT
void reconectarMQTT() {
  // cria um loop de verificação de estabilidade da conexão
  // para reconectar quando for necessário
  while (!mqtt.connected()) {
    Serial.print("Tentando conecta-se ao broker...");
    // Tentando conexão
    if (mqtt.connect((char*)nome)) {
      Serial.println("Conectado!");
      // inscreve na rota, para receber dados
      String rota = rotaReceber();
      mqtt.subscribe((char*)rota.c_str());
    } else {
      Serial.print("Falhou, re-");
      Serial.print(mqtt.state());
      Serial.println(" Tentando novamente em 5 segundos");
      delay(5000);
    }
  }
}
```

Fonte: Autor.

Para finalizar os métodos e processos do uso do protocolo MQTT, realiza-se a inicialização no escopo de execução do controlador, definindo o servidor e *callback* na inicialização e no laço de execução, verifica-se a conexão e executa o próprio loop do MQTT.

**Figura 26 – Estrutura do endereçamento de rotas MQTT.**

```
void setup(){
  //configuração anterior [...]

  // inicializa
  mqtt.setServer(mqtt_server, 1883);
  mqtt.setCallback(callbackMQTT);
}

void loop(){
  if (!mqtt.connected()) {
    reconectarMQTT();
  }

  mqtt.loop();

  //mqtt.publish(String(mfahrenheit()), mqtt);
}
```

Fonte: Autor.

Assim conclui-se a programação da comunicação para o MVP, abrindo o espaço necessário para a programação das leituras de sensores, para serem encaminhados via protocolo MQTT.

No MVP selecionaremos o sensor de tensão AC ZMPT101K que realiza a leitura de tensão AC entre 0 – 250V, para monitoramento do sistema de energia do rak. Podemos facilitar a sua aplicação, utilizando uma biblioteca de métodos chamada *EmonLib*, que também pode ser usada para realizar leituras de corrente, também e m AC.

**Figura 27 – Programação do sensor de Tensão AC com *EmonLib*.**

```
#include "EmonLib.h" // include emon library

#define PIR_SW_IN
//VALOR DE CALIBRAÇÃO
//DEVE SER AJUSTADO DE ACORDO COM SEU PIR (PIRADO)
#define VOLT_CAL 211.6

EnergyMonitor str; // cria uma instância de sensor

void setup()
{
  Serial.begin(9600);

  str.voltage(PIR_SW, WKT_CAL, 1.0); // voltagem: PIR, ENTRADA, e CALIBRAÇÃO, DESLOCAMENTO DE FASE
}

void loop(){
  str.calcV[17,2000]; //FUNÇÃO DE CÁLCULO (SEmpreculos, TEMPO LIBRE PARA FAZER A MEDIÇÃO)
  float tensao = str.Vrms; //VARIÁVEL RECEBE O VALOR DE TENSÃO IRMS LIDO
  Serial.print("Tensão medida na rede AC: " + String(tensao) + " V AC"); // IMPRIME O TEXTO NA SERIAL
  delay(1000);
}
```

Fonte: Autor.

Como é possível obter o valor de tensão através da variável *tensao*, podemos então serializar em formato JSON, para ser enviado como carga útil na publicação do MQTT, como disposto na figura 28.

**Figura 28 – Captura de dados, conversão para JSON e envio via MQTT.**

```
#include <ArduinoJson.h>

StaticJsonDocument<200> metadata; // instancia objeto json

void setup() {
  //...
}

void loop() {
  //...
  float calCV(17, 2000); //FUNÇÃO DE CÁLCULO (17 SERVIDORES, TEMPO LIMITE PARA FAZER A MEDIÇÃO)
  metadata["sensor_tensao"] = str.Wms; // obtém valor esticado de tensão

  String jsonString; // cria uma variável string, para converter o conteúdo de json string

  serializeJson(metadata, jsonString); // serializa o objeto em uma string
  // de seguida envia carga de dados
  String enviarDados = rotaEnviar();
  mqtt.publish((char*)"enviarDados.c_str()", (char*)"jsonstring.c_str()");
}
}
```

Fonte: Autor.

O modelo acima, funcional, porém não conseguirá trabalhar com multitarefas, pois os microcontroladores executam tarefa por tarefa, aumentando o tempo de resposta. E o presente trabalho propõe uma estrutura além de flexível, persistente, confiável e para isso é necessário realizar utilizar uma serie de métodos, nativos do framework e do pacote de ferramentas do Esp32.

Esses métodos, constituem a tecnologia FreeRTOS adicionado à camada de programação do framework Arduino para operar e multitarefas. O FreeRTOS é um micro sistema operação feito para ser compatível com microcontroladores de baixo poder de processamento.

Para efeitos alcançar a meta de multitarefa, precisa-se separar blocos de tarefas do loop padrão do framework Arduino, chamado de *void loop()* . Em uma análise computacional as tarefas executadas até o momento são: persistência do MQTT, leitura de sensores e Envio de dados, como visto na figura 29 e 30.

**Figura 29 – Função de loop para verificação de conectividade MQTT.**

```
// protocolo MQTT
void MQTTinterupta(void * parameter){
  while(1){
    if (!mqtt.isConnected()) {
      reconnectMQTT();
    }
    mqtt.loop();
  }
}

TaskHandle_t MQTT_interrupte = NULL; // variável para interromper tarefa caso necessário
// processo leitura de sensores
void leituraSensores(void * parameter){
  while(1){
    sensor_t.c.calCV(17, 2000); //FUNÇÃO DE CÁLCULO (17 SERVIDORES, TEMPO LIMITE PARA FAZER A MEDIÇÃO)
    metadata["sensor_tc"] = sensor_t.c.Wms; //obtem o valor de tensão em Ohms
    vTaskDelay(1000 / portTICK_PERIOD_MS); // intervalo para realizar o processo novamente
  }
}

TaskHandle_t Sensores_interrupte = NULL; // variável para interromper tarefa caso necessário
```

Fonte: Autor.

**Figura 30 – Função loop para envio de dados MQTT.**

```
// processo envio de dados
int tempoParaEnvio = 60000; //intervalo tempo padrão envio de dados (ms)
void enviodeDados(void * parameter){
    while(1){
        String jsonString; // cria uma variavel string, para converter o conteúdo em json string
        serializeJson(metaDado, jsonString); // serializa o objeto em uma string
        String enviarDados = rotaEnviar();
        //Envia carga de dados via mqtt
        mqtt.publish((char*)enviarDados.c_str(), (char*)jsonString.c_str());
        vTaskDelay(tempoParaEnvio / portTICK_PERIOD_MS); // intervalo para realizar o processo novo
    }
}
TaskHandle_t enviodeDados_interruptor = NULL; // variavel para interromper tarefa caso necessario
```

Fonte: Autor.

Separando cada bloco de tarefas em novas funções, encapsuladas com novos loops infinitos através de diretiva *While*. E na função de inicialização *Setup()* cria-se uma multitarefa usando a função *xTaskCreate* passando a função da tarefa a ser executada, um nome, tamanho alocado, parâmetros, prioridade e uma variável para interrupção dada como opcional para executar sem interrupções basta trocar por *NULL*.

**Figura 31 – Captura de dados, conversão para JSON e envio via MQTT.**

```
void setup(){
    // Cria processo de persistência de conexão do mqtt
    xTaskCreate(
        MQTTPersistente, // Função a ser chamada
        "MQTT_Task", // Nome da tarefa
        1000, // Tamanho (bytes)
        NULL, // Parametro a ser passado
        1, // Prioridade da tarefa
        MQTT_interruptor // task handle
    );
    xTaskCreate(leituraSensores, "leitura_sensores", 1000, NULL, 1, Sensores_interruptor);
    xTaskCreate(enviodeDados, "envio_dados", 1000, NULL, 1, enviodeDados_interruptor);
}

void loop(){
    // Inicializado devido a criação de loops sem bloqueios
}
```

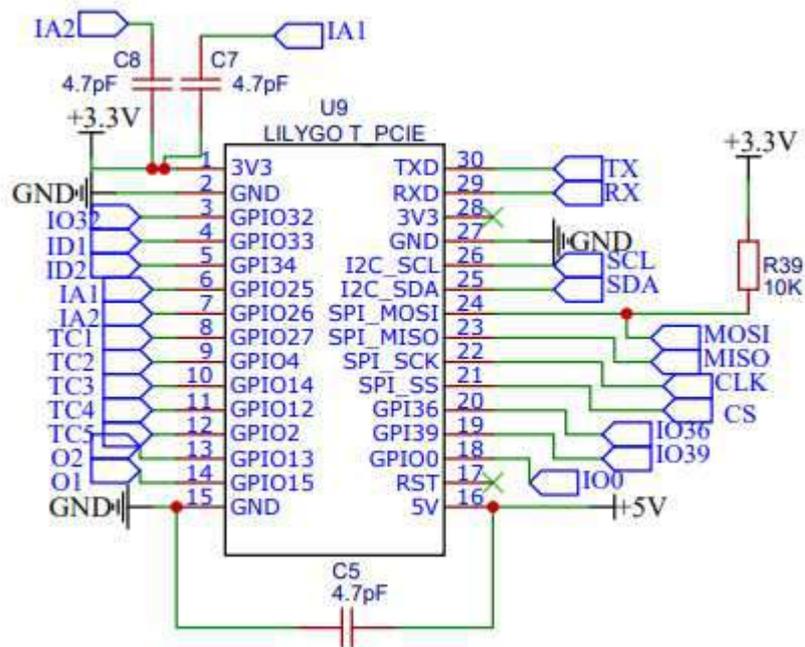
Fonte: Autor.

Com essas técnicas dotadas, criou-se um ambiente flexível de parametrização, um processo persistência de dados que o torna o dispositivo confiável, fazendo a leitura de sensores e enviado para uma plataforma compatível com protocolo MQTT.

### 6.1.2 Hardware

Neste tópico veremos o desenvolvimento do circuito físico do dispositivo, conforme elaborado na figura 13, que constituiu os setores eletroeletrônico. O primeiro passo é mapear os pontos de circuitos, e a utilização de elementos básicos como capacitores e resistores, sugeridos pelo fabricante, afim de garantir a qualidade e eficiência do microcontrolador, já que este depende de uma energia sem ruídos ou surtos.

**Figura 32 – Circuito central do controlador, ESP32 na placa de circuito Impresso.**



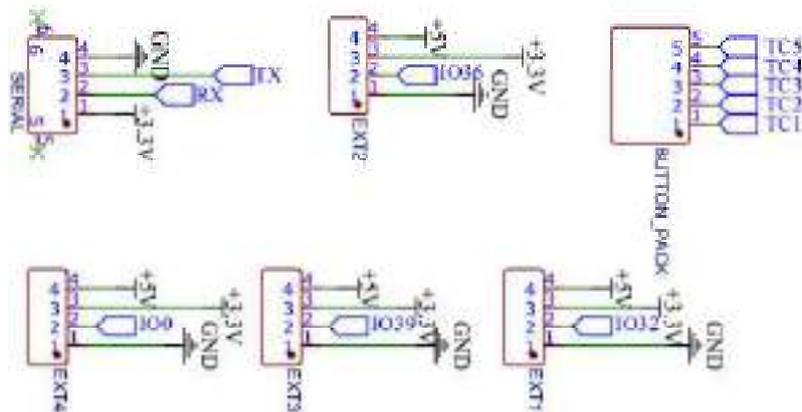
Fonte: Autor.

A figura 32 apresenta o circuito necessário para operar a placa de desenvolvimento ESP32 T-PCIE. Pode-se observar que o sensor de voltagem, definido no código do controlador, está alocado na GPIO25. Uma observação importante a ser considerado é que o ESP32 possui 2 Conversores Analógicos Digitais - ADC e o WIFI para operar. Ao utilizar o ADC 2 ao compilar para o controlador irá se deparar com o erro, para resolver basta usar as portas do ADC1 que corresponde às portas GPIO 32 à 39.

Portas MISO, MOSI, CLK e CS destina-se à comunicação SPI, sendo está o uso de periférico SIM7070G. A placa de transmissão de dados móveis em nosso modelo, está sendo representado pelas configurações WIFI já que ambas trabalham com protocolos internet. O acréscimo é somente ao uso da biblioteca correta TinyGSM, e instanciando o Client Internet na classe MQTT.

A figura 33 apresenta os Bornes de conexão de entrada e saída ou de comunicação serial consideradas auxiliares. Pois, ao depender do nível de recursos que o cliente possa desejar, elas podem ser utilizadas para definir uma funcionalidade. Por exemplo, usar os sensores touch para botões, alocados na matriz de entradas BUTTON\_PACK ou Usar a EXT1 para acionar um mini ar-condicionado para resfriamento do rak.

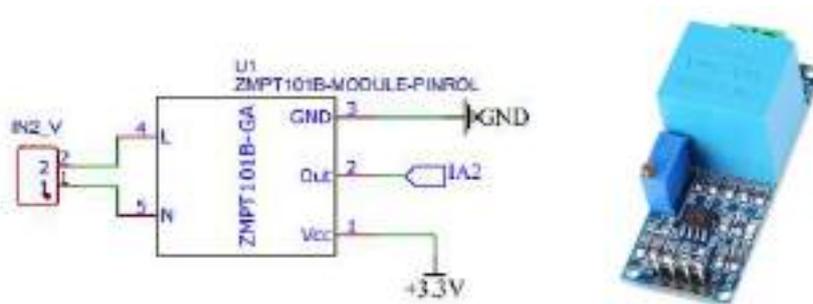
**Figura 33 – Circuito das Portas Auxiliares.**



Fonte: Autor.

O sensor de Tensão, conforme observado na figura 34 e pelo grupo (OPENENERGYMONITOR), desenvolve de ferramentas *Open Source* pra leitura e monitoramento de energia elétrica, diz que o sensor necessita de calibração e a maneira rápida e eficiente. Para isso, é a utilização de um osciloscópio para deslocar através de um *trimpot* o sinal de saída ficar equivalente ao sinal de entrada, assim a margem de erro ou a defasagem das fases não afetariam drasticamente às leituras (LIN, KHAN e DAUD, 2021).

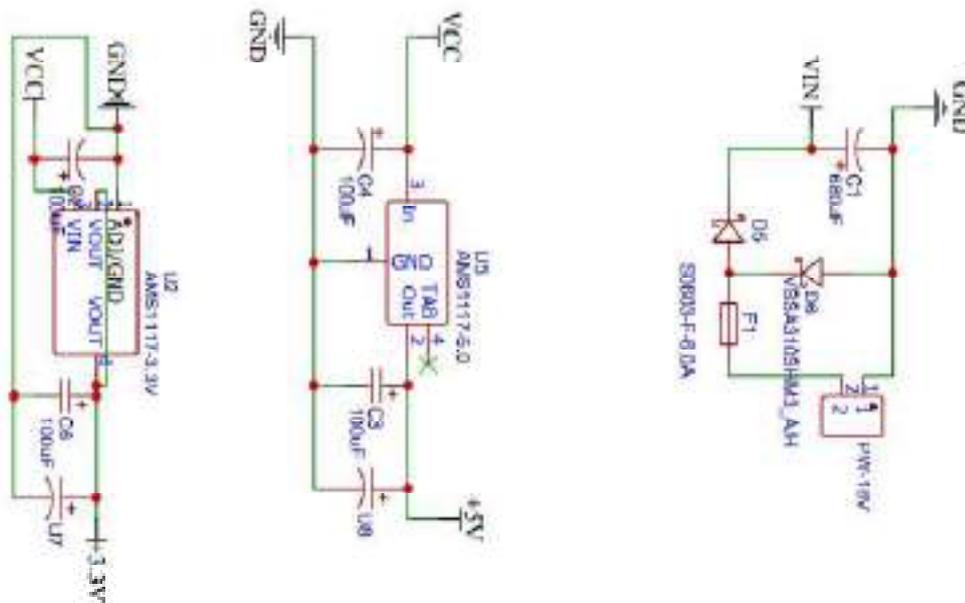
**Figura 34 – Circuito de recepção do sinal analógico do sensor de Tensão AC para o ESP32.**



Fonte: Autor.

O setor de alimentação possui três seções principais, duas regulações de tensão através de um CI regulador de tensão AMS1117 de 3.3Vcc com capacidade de fornecer até 800mA de corrente para periféricos que operam nessa faixa de tensão (NATH, HEMEL e SHIKHA, 2021). Um regulador AMS1117 de 5Vcc também destinado à alimentação e qualidade de energia para periféricos, além de carregar o backup de energia do controlador e alimentá-lo também.

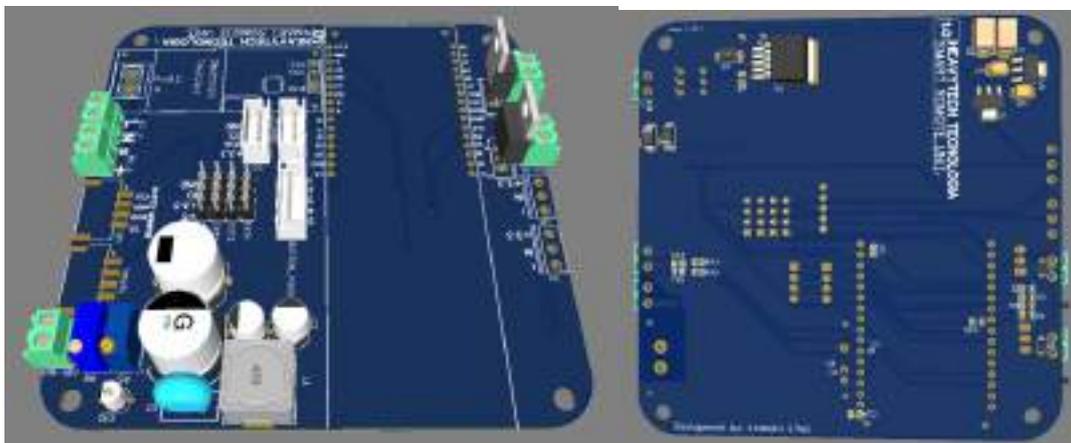
**Figura 35 – Circuito de entrada de energia e regulação em níveis de 3.3 e 5Volts até 1A.**



Fonte: Autor.

Após a elaboração do circuito, realizado na plataforma *EasyEda* de desenvolvimento de circuitos eletrônicos contendo também uma gama de recursos de peças, layouts e até mesmo fabricação direta a baixo custo, podemos chegar ao seguinte resultado visualmente apresentável em 3D na plataforma.

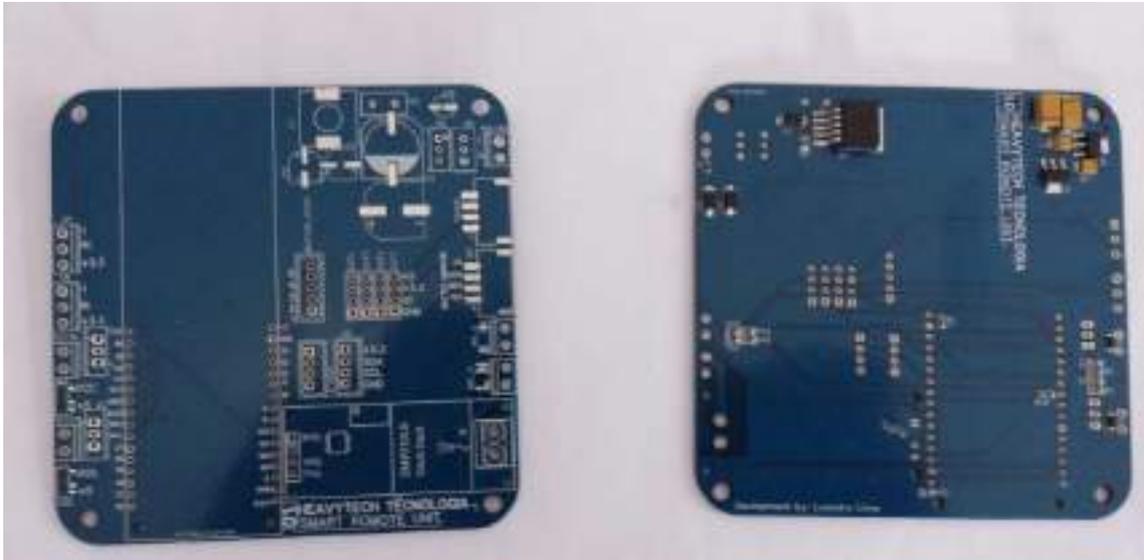
**Figura 36 – Vista virtual de topo e fundo da placa de circuito impresso do dispositivo.**



Fonte: Autor.

Com essa plataforma de desenvolvimento, podemos solicitar a confecção do circuito em uma fabricante parceira, desde que realize o pedido acima de 10 unidades de tamanhos até 10x10cm de forma rápido e simples. Conforme resultado na figura 37.

**Figura 37 – Vista de topo e fundo da placa de circuito impresso do dispositivo.**



Fonte: Autor.

Como observado na imagem 3D, a placa física já confeccionada saiu com a qualidade equivalente ao apresentado na simulação, restando apenas a soldagem dos equipamentos, o que não será abordado neste trabalho.

## 7 Considerações Finais

O presente projeto, em todo o seu desenvolvimento, obteve êxito em boa parte dos seus objetivos propostos conforme definidos anteriormente. Quanto a IoT, foi observado que em sua concepção e sua destinação se dá para a automação, interconexão com outros objetos e ao ser humano.

Dentre a IoT, e suas tecnologias tanto na aplicação, percepção e rede. Com o estudo de recursos tecnológicos e componentes muito importantes para realizar soluções rápidas, pontuais e uma gama de ferramentas e facilidades na sua programação. Quanto aos microcontroladores ESP32, entende-se como um equipamento elaborado e destinado para aplicação na IoT possui com um amplo poder de processamento, memória e portas disponíveis. E o protocolo MQTT, talvez o maior aliado para a rede, por ser um protocolo que objetiva conexões às longas distâncias, demonstrou ser um protocolo simples, aplicável e escalável para praticamente qualquer hardware de automação, com sua baixa banda de comunicação e velocidade.

Quanto à inserção das tecnologias referidas e à aplicação na área de segurança de telecomunicações como um meio de garantir e melhorar as técnicas de manutenção e qualidade de serviços de internet e TV, tornam-se aptas. Pois, cumprem seus requisitos, tais como a simplicidade de aplicação, precisão e confiabilidade de tais ferramentas, seu baixo custo de desenvolvimento e investimento, sua aptidão para aplicações de automação e monitoramento de pontos longos na malha da indústria de telecomunicações.

Como parte de bons resultados o projeto o proporcionou a criação de uma StartUp voltada à área de gestão e monitoramento de recursos empresariais. Além de um pedido de patente de modelo de utilidade sob número BR 20 2022 021692 0, intitulado DISPOSITIVO DE MONITORAMENTO E CONTROLE DE RACKS DE TELECOMUNICAÇÕES advindo de incentivos de pesquisa e incubação do Edital Centelha 22/2019.

Para os próximos trabalhos, será realizado o desenvolvimento da plataforma de gestão bem como testes de desempenho e indicação de qualidade em campo com empresas parceiras. E contribuir com o desenvolvimento social e tecnológico da sociedade brasileira.

## REFERÊNCIAS

**ALVES DE BRITO , Rubens J.** Rede 4G - Implementação. Americana - São Paulo. 2011.

**ANATEL. Painéis de Dados.** Agência Nacional de Telecomunicações, **Dezembro 2020.**  
Disponível em: <<https://www.anatel.gov.br/paineis/consumidor>>.

**ANATEL. Spectrum-E.** Licenciados. ISSN  
<https://sistemas.anatel.gov.br/se/public/view/b/licenciamento.php>. Acesso em: 01 jul. 2022.

**ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS.** Rede de distribuição de energia elétrica — Compartilhamento de infra-estrutura com redes de telecomunicações. **Rio de Janeiro. 2016. (NBR 15214).**

**CAMARNEIRO, Daniel et al.** Desenvolvimento de serviços REST para monitorização dos consumos de energia em chão de fábrica, baseado nas plataformas Eclipse IoT: Bosch e SCoT. Aveiro, Portugal. 2021.

**CHAUDHARI, Bharat; ZENNARO, Marco; BOKAR, Suresh.** LPWAN technologies: Emerging application characteristics, requirements, and design considerations. Future Internet, **12, 2020.**

**CHEN, Xin et al.** Performance Analysis and Uplink Scheduling for QoS-Aware NB-IoT Networks in Mobile Computing. [S.l.]. 2019. (10.1109/ACCESS.2019.2908985).

**EQUATORIAL ENERGIA.** Compartilhamento de Infraestrutura de Rede de Distribuição Aérea. Equatorial Energia. São Luís, p. 31-46. 2022. (NT.016.EQTL.Normas e Qualidade).

**FAN, Ersha et al.** Sustainable Recycling Technology for Li-Ion Batteries and Beyond: Challenges and Future Prospects. Chemical Reviews, v. **14**, p. 7070-7063, 2020.

**GERBER, Anna; KANSAL, Satwik.** Simplify the development of your IoT solutions with IoT architectures. IBM Developer, 7 out. 2017. Disponível em:  
<<https://developer.ibm.com/technologies/iot/articles/iot-lp201-iot-architectures/>>.  
Acesso em: 10 dez. 2020.

**GOMES MOTA, Vitor et al.** Evolução da tecnologia de telefonia móvel e estudo e caracterização de um sistema móvel 5G de quinta geração. ENGEVISTA, Rio de Janeiro, v. **21**, n. 1, p. 154-175, Fevereiro 2019. ISSN online: 2317-6717.

**HANES, David.** IoT fundamentals: Networking technologies, protocols, and use cases for the internet of things. Cisco Press, 2017.

**IBGE. Países.** Instituto Brasileiro de Geografia e Estatística, **Março 2021.** Disponível em: <<https://pais.es.ibge.gov.br/#/mapa/ranking/brasil?indicador=77849&tema=5&ano=2019>>.

**ITU.** IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond. **International Telecommunication Union. Genebra. 2015.**

**LILYGO.** LILYGO® TTGO T-PCIE ESP32-WROVER-B AXP192 Chip WIFI Bluetooth Nano Card SIM Series Composable Development Board Hardware. lilygo, **01 jan. 2022.** Disponível em: <[http://www.lilygo.cn/prod\\_view.aspx?TypeId=50063&Id=1313&FId=t3:50063:3](http://www.lilygo.cn/prod_view.aspx?TypeId=50063&Id=1313&FId=t3:50063:3)>.

**LIMA, João C. D.** Simulação de aplicações utilizando protocolos 6LOWPAN, RPL, MQTT e COAP em Smart Cities. [S.l.]. **2018.**

**LIN, Tan Rui; KHAN, Nur Hanisah Omar; DAUD, Muhamad Zalani.** Arduino based appliance monitoring system using SCT-013 current and ZMPT101B voltage sensors. *Przełąd Elektrotechniczny*, **97**, **2021.**

**MARTINS, Ismael R.; ZEM, José L.** Estudo dos protocolos de comunicação MQTT e COaP para aplicações machine-to-machine e Internet das coisas. *Revista Tecnológica da Fatec Americana, Americana*, v. **3**, n. **1**, p. **64-87**, **março 2015.**

**MINISTÉRIO DAS COMUNICAÇÕES.** 5G, Um novo Brasil Para Você. Governo Federal - Ministério das Comunicações, **01 mar. 2022.** Disponível em: <<https://www.gov.br/mcom/pt-br/assuntos/5g>>. Acesso em: **01 mar. 2022.**

**NATH, Subroto K. D.; HEMEL, Hasibur ; SHIKHA, Zubaida A.** A Low-Cost Non-Isolated AC/DC Power Supply Design and Analysis for Embedded Systems. 2021 International Conference on Electronics, Communications and Information Technology (ICECIT), **2021.**

**NIKOLOV, Neven.** Research of MQTT, CoAP, HTTP and XMPP IoT. 2020 XXIX International Scientific Conference Electronics (ET), **2020.**

**OPENENERGYMONITOR.** Learn OpenEnergyMonitor. OpenEnergyMonitor. Disponível em: <<https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory/introduction>>. Acesso em: **01 jan. 2022.**

**RANGEL GARCIA, Paulo ; KLEINSCHMIDI, João H.** Tecnologias emergentes de conectividade na iot: Estudo de redes lpwan. XXXV Simósio Brasileiro de Telecomunicações e Processamento de Sinais-SBrT2017, **2017.**

**RAZA, Usma; KULKARNI, Parag; SOORIYABANDARA, Suresh.** LPWAN technologies: Emerging application characteristics, requirements, and design considerations. *Future Intenert*, **12**, **2020.**

**SANTOS, Bruno P. et al. Internet das Coisas: da Teoria à Prática. In: COUTINHO, Antonio A. T. R. Livro Texto Minicurso. Belo Horizonte: Sociedade Brasileira de Computação, 2016. Cap. 1. Disponível em: <<https://bps90.github.io/papers/Internet-das-coisas-da-teroria-a-pratica/#>>. Acesso em: 10 dez. 2020.**

**SANTOS, Jadir P. D. et al. Industry 4.0 - Efforts to adjust man the Revolution 4.0. RESEARCH, SOCIETY AND DEVELOPMENT, 9, 2020.**

**SIMCOM. SIM7070 Series Hardware Design. Shanghai. 2020. (Versão 1.03).**

**TAUFIQUE, Azar et al. Planning Wireless Cellular Networks of Future: Outlook, Challenges and Opportunities. IEEE Access, 5, 2017.**

**VELOSO, Artur F. S. et al. Prototipação com nodeMCU para Internet das Coisas em Smart Cities: Uma pesquisa. III Escola Regional de Informática do Piauí, Piauí, 2017. 105-110.**

**Y. MANE, Sarika. LPWAN's – Overview, Market Scenario and Performance Analysis of Lora, Sigfox Using NB-Fi Range Calculator. 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 2021.**

**YUAN, Michael. Conhecendo o MQTT: Por que o MQTT é um dos melhores protocolos de rede para a Internet das Coisas? IBM Developer, 04 out. 2017. Disponível em: <<https://developer.ibm.com/br/articles/iot-mqtt-why-good-for-iot/>>. Acesso em: 10 dez. 2020.**

**YUAN, Quan et al. Research on Mixed Planning Method of 5G and LTE. 2020 3rd International Conference on Information and Computer Technologies (ICICT), 2020.**