



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Interface de rede para monitoramento em tempo real aplicado em uma placa solar

Trabalho de Conclusão de Curso

Paulo Vitor Aragão Silva



São Cristóvão – Sergipe

2021

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Paulo Vitor Aragão Silva

**Interface de rede para monitoramento em tempo real
aplicado em uma placa solar**

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a): Prof. Dr. Edward David Moreno
Coorientador(a): Prof. Dr. Douglas Bressan Riffel

São Cristóvão – Sergipe

2021

Seja extraordinário!

Agradecimentos

Em primeiro lugar, gostaria de agradecer a toda natureza e astros que abençoou e fez com que meus objetivos fossem alcançados independente do percurso percorrido. Aos amigos e familiares em especial, minha mãe, meu pai, meu irmão Gustavo, irmã Kelly, minha segunda mãe Maria, Vanessa minha melhor amiga que sempre me apoiou, deu puxões de orelha mas nunca e jamais me abandonou mesmo em momentos tão difíceis da graduação e da vida pessoal, Jonathan pelo conhecimento sempre compartilhado, experiência e ensinamentos com um coração gigante que com certeza carrego muito conhecimento técnico por causa dele, Marden que com tão pouco tempo vem me ensinando em ser uma pessoa melhor e um profissional incrível além de muitos ensinamentos entusiamos e apoio sendo mais que um líder de time, Henrique e Lucas em especial parceiros de matérias, sonos perdidos e amigos incríveis que tenho orgulho de levar eles pra vida, Túlio, Adriel, Bruno Rodrigues, Carol, Bruno Cavalcante meu vizinho que sempre compartilhávamos o sonho de formação da gente, Reneilson o cara mais incrível com coração gigante que conheci, Michel por todo conhecimento compartilhado e amizade, ao grupo Burgueses (Tallis, Rafael, Matheus Macena, Camilo, Matheus Barreto, Giuseppe) por sempre estarem juntos como uma família mesmo distante, Bruno Anunciação um cara incrível com uma família linda que sempre se preocupou e chegou junto com apoio da minha formação, ao antigo LEIA(Laboratório de Pesquisa da BK Telecom) em especial para Leones um dos amigos mais incríveis em eletrônica que pude conhecer aprender e a trabalhar, Ingrid por sempre acreditar em meu potencial e me acolher em momentos tão difíceis, Larissa por ter compartilhado momentos difíceis em Eletrônica I, Família Colibri (Alê, Deyse, George, Berenice, Gabriel, Vitória e Jesney) por terem uma energia única e me ensinarem sobre ciclos e acreditarem no meu crescimento, todos aqueles que me desculpa se não consegui lembrar de todos que acreditaram e me apoiaram como amigo e formação acadêmica. Sem vocês eu não seria o Paulo de hoje. Gostaria de agradecer aos professores Edward Moreno e Douglas Bressan por terem sido meus orientadores e desempenhado tal função com dedicação, preocupação com a minha formação e pela amizade formada. A todos aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

"Tudo está conectado!" - Douglas Adams

Resumo

A solução de interface de rede desenvolvida nesta proposta de conclusão de curso propõe um estudo de caso genérico IoT com foco na transmissão de dados coletados para nuvem resolvendo o problema do armazenamento local devido a condições hostis do microcontrolador em campo e da perda de dados durante o armazenamento em um cartão SD. Visando uma solução de baixo custo comparada a soluções prontas como módulos de *datalogers* comerciais, que são importados e de alto custo e com controle e armazenamento de dados via *softwares* proprietários. Esta solução comercial apresenta baixa escalabilidade de sensores, baixa flexibilidade de expansão, manutenção restritiva ao fabricante e além de distância via cabo para comunicação para a internet. A solução desenvolvida nesse trabalho tem como foco a interface de comunicação para o servidor que são é baseado toda em uma solução *open-source*. O mesmo consiste em uma solução online de uso gratuito e se comunica a um servidor na nuvem de forma sem fio via WiFi com alta escalabilidade de microcontroladores que podem ser acoplados e sensores. Além disso, por se tratar de uma solução ao usuário final web ela é multiplataforma (Linux, Windows, MAC OSX e Android) o qual permite maior facilidade de acesso ao usuário final para consultas de relatórios, acompanhamento de dados em tempo real e criação de *triggers* em caso de necessidade de alerta de sensores no monitoramento. A proposta permitiu realizar a configuração de um servidor em um *Raspberry Pi* utilizando soluções *open-source* como MQTT, InfluxDB e Grafana, além de realizar a programação de microcontroladores que serão responsáveis pela transmissão de dados para a internet o ESP32. Como resultado obtido, foi possível realizar a coleta de dados de sensores no estudo de caso e realizado o envio de dados via ESP32 validando a solução proposta.

Palavras-chave: Internet das Coisas, MQTT, I2C, Interface de Rede, Grafana, InfluxDB, ESP32, Arduíno e Raspberry Pi.

Abstract

The network interface solution developed in this work proposes a generic IoT case study focusing on transmitting collected data to the cloud solving the local storage problem due to hostile microcontroller conditions in the field and loss of data while storing to an SD card. Aiming for a low-cost solution compared to ready-made solutions such as commercial datalogger modules, which are imported, costly and with data control and storage done via proprietary software. This commercial solution features low sensor scalability, low expansion flexibility, maintenance is restricted to the manufacturer and long cable distance needed for internet communication. The solution developed in this work focuses on a communication interface for the server which is based on an open source solution. It is also a free online solution with wireless communication to a cloud server via WiFi, highly scalable in regards to pluggable microcontrollers and sensors. Additionally, since it is a solution for the web end user, it is cross-platform (Linux, Windows, MAC OSX and Android) which allows easier end-user access to report queries, real-time data tracking and creation of triggers for the monitoring sensors. This proposal allowed for the configuration of a server on a Raspberry Pi using open source solutions like MQTT, InfluxDB and Grafana, in addition to carrying out the programming of microcontrollers responsible for the data transmission to the internet or ESP32. As a result, it was possible to carry out data collection from sensors and send data via ESP32 which validated the proposed solution.

Keywords: Internet of Things, MQTT, I2C, Network Interface, Grafana, InfluxDB, ESP32, Arduino and Raspberry Pi.

Lista de ilustrações

Figura 1 – Matriz elétrica brasileira em maio de 2017 de acordo com dados disponíveis no Banco de Informações de Geração da Agência Nacional de Energia Elétrica.	15
Figura 2 – Raspberry Pi Model 3B	19
Figura 3 – Raspberry Pi Model 3B	19
Figura 4 – Microcontrolador ESP32	20
Figura 5 – Especificação ESP32 Geral	20
Figura 6 – Diagrama de Entradas e Saídas do ESP32 DEVKIT V01	21
Figura 7 – Diagrama de Blocos do ESP32	21
Figura 8 – Microcontrolador Arduino Uno R3	22
Figura 9 – Saídas e Entradas do Arduino Uno R3	22
Figura 10 – Conversor Digital Bidirecional 5v para 3.3v	23
Figura 11 – Especificação do parâmetro elétrico de um termistor	24
Figura 12 – Formas de distribuição da mensagem pelo protocolo MQTT	25
Figura 13 – Arquitetura de referência do I2C	25
Figura 14 – Estrutura de bandas de energia em (a) condutores, (b) semicondutores e (c) isolantes	27
Figura 15 – Eficiência das células fotovoltaicas fabricadas em laboratórios até 2019 segundo	28
Figura 16 – Associação em série de células solares.	30
Figura 17 – Associação em paralelo de células solares.	31
Figura 18 – Influência da variação da irradiância solar na curva característica I-V de uma célula fotovoltaica de silício cristalino em condições STC.	31
Figura 19 – Influência da temperatura da célula fotovoltaica na curva I-V (para irradiância de 1.000 W/m ² , espectro AM1,5	32
Figura 20 – Diagrama de Blocos Simplificado da Solução.	41
Figura 21 – Diagrama de Blocos em campo	42
Figura 22 – Diagrama da Interface de Rede	43
Figura 23 – Fluxograma Código da Interface de Rede	43
Figura 24 – Diagrama de Blocos do servidor	44
Figura 25 – Interface de Acesso do Grafana	45
Figura 26 – Estrutura de Simulação de Coleta de Dados via ESP32 para o Servidor	46
Figura 27 – Bancada de teste de Coleta de Dados via ESP32 para o Servidor	46
Figura 28 – Gráfico de visualização de dados simulados no Grafana	46
Figura 29 – Fluxograma Código da Interface de Rede	47
Figura 30 – Microcontrolador para coleta dos dados	47

Figura 31 – Distribuição em vermelho de termistores em uma placa solar	47
Figura 32 – Termistores aplicado no modulo solar	48
Figura 33 – Termistores aplicado no modulo solar	48
Figura 34 – Apresentação de dados do Grafana coletados da interface de rede	49

Lista de abreviaturas e siglas

MQTT	<i>Message Queuing Telemetry Transport</i> (Protocolo de mensagens leve)
TCP	<i>Transmission Control Protocol</i> (Protocolo de Controle de Transmissão)
CoAP	<i>Constrained Application Protocol</i> (Protocolo de Aplicativo Restrito é um Protocolo de Aplicativo da Internet)
XMPP	<i>Extensible Messaging and Presence Protocol</i> (Protocolo aberto, extensível, baseado em XML)
SNMP	<i>Simple Network Management Protocol</i> (Protocolo Simples de Gerência de Rede)
WAMP	<i>Web Application Messaging Protocol</i> (Protocolo do WebSocket)
IBM	<i>International Business Machines Corporation</i>
M2M	<i>Machine to machine</i>
USB	<i>Universal Serial Bus</i>
RS232	Protocolo para troca série de dados binários entre um DTE (terminal de dados, de <i>Data Terminal equipment</i>)
PIC	Microcontrolador da <i>Microchip</i>
Zigbee	Protocolo de comunicação
I-V	Curva Corrente por Tensão
IOT	Internet das Coisas
STC	Condições normais de Ensaio
PIB	Produto Interno Bruto
ANEL	Agência Nacional de Engenharia Elétrica
I2C	Inter Integrated Circuit
SDA	<i>Serial Data</i>
SCL	<i>Serial Clock</i>
FV	Fotovoltaico

SD	<i>Secure Digital</i>
DATS	<i>Data Acquisition and Transmission System</i>
IOE	<i>Internet of Energy</i>
IaaS	Infraestrutura como Serviço
SaaS	Software como Serviço
PaaS	Plataforma como Serviço
RAM	<i>Random Access Memory</i>
HDMI	<i>High-Definition Multimedia Interface</i>
ROM	<i>Read-Only Memory</i>
IDE	<i>Integrated Development Environments</i>
SDK	<i>Software Development Kit</i>
ICSP	<i>In-Circuito Serial Programing</i>
NTC	(Negative Temperature Coeficien
HTML	<i>HyperText Markup Language</i>
PC	Personal Computer

Lista de símbolos

K	Kelvin
C	Celsius
A	Amperes
V	Volts
Hz	Hertz
MHz	Mega Hertz
J	Joules
W	Watts
m ²	Metros Cuadrados
Gb	Gigabytes
Kb	Kilobytes

Sumário

1	Introdução	14
1.1	Objetivos	15
1.1.1	Geral	15
1.1.2	Específico	16
1.2	Estrutura do Documento	16
2	Fundamentação Teórica	17
2.1	Internet das Coisas(IoT)	17
2.2	Microcontroladores	18
2.2.1	Raspberry Pi	18
2.2.2	ESP32	19
2.2.3	Arduíno	21
2.3	Sensores	23
2.3.1	Conversor Digital 5V para 3.3V	23
2.3.2	Termistor	23
2.4	Servidor	24
2.4.1	Backend	24
2.4.1.1	Protocolo de comunicação	24
2.4.1.1.1	MQTT	24
2.4.1.1.2	I2C	25
2.4.1.2	Banco de Dados	25
2.4.1.2.1	InfluxDB	26
2.4.2	Front-end	26
2.4.2.1	Grafana	26
2.5	Célula Solar	26
2.6	Propriedades de uma Célula Solar	28
2.6.1	Curva I-V	28
2.6.2	Associação de células solares em um módulo fotovoltaico	29
2.6.2.1	Associação em Série	29
2.6.2.2	Associação em Paralelo	30
2.6.3	Influência da Irradiação Solar	30
2.6.4	Influência da Temperatura	32
3	Revisão de trabalhos correlatos	33
3.1	Revisão de trabalhos acadêmicos	33
3.2	Considerações Sobre os Trabalhos Relacionados	34

4	Método e Materiais Usados	38
4.1	Método	38
4.2	Material	39
4.2.1	Sistemas Embarcados	39
4.2.1.1	Arduíno	39
4.2.1.2	ESP32	39
4.2.2	Raspberry PI	39
4.3	Sensores	39
4.3.1	Abaixador de Tensão	39
4.3.2	Sensor de Temperatura	39
5	Desenvolvimento da Interface de rede para monitoramento em tempo real	40
5.1	Aplicação da Metodologia	40
5.2	Diagrama de Blocos do Projeto	41
5.3	Desenvolvimento da Interface de Rede	43
5.4	Desenvolvimento do Servidor	44
5.5	Implementação da solução	45
5.5.1	Envio de dados via ESP32 para o Servidor	45
5.5.2	Envio de dados via Arduino para o Servidor	46
5.6	Aplicação de Estudo de Caso em uma Placa Solar	47
6	Conclusão	50
	Referências	52

1

Introdução

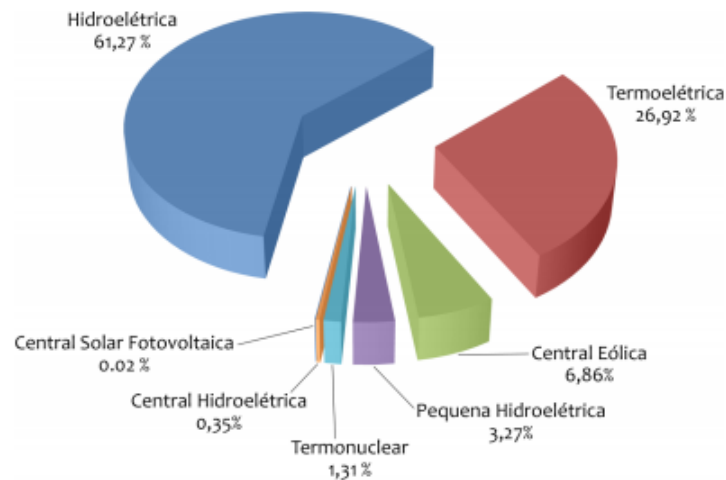
Com o passar dos anos, a evolução das redes de comunicação mudaram a forma como pessoas e organizações trocam informações e organizam as suas atividades. Mecanismos distribuídos e microssores estão cada vez mais presentes no nosso dia a dia. Eles são capazes de captar dados, conectar-se à internet e se comunicar sem a necessidade de fios.

Esse fenômeno introduziu um novo conceito de interconexão digital: a Internet das Coisas (IoT), um tema de interesse que associa a Computação a áreas como Medicina, Geologia, Tráfego Urbano, dentre outros. A Internet das Coisas gerou uma ampla gama de aplicações práticas, com estudos que cada vez mais buscam por soluções de baixo custo e consumo de energia. (PEREIRA, 2018)

Hoje, porém, um dos problemas enfrentados pelas soluções IoT é a gestão de rede, o que envolve transmissão, proteção e armazenamento de dados. Uma das soluções de particular interesse da área de IoT está no sensoriamento de placas solares. Para entendimento melhor da aplicação e o impacto do IoT na produção elétrica solar, iremos primeiro visualizar um panorama geral da malha energética brasileira.

A produção elétrica, uma das áreas da aplicabilidade da Internet das Coisas, possui uma heterogeneidade em sua produção, sendo elas: marítima, hídrica, térmica, luminosa, tectônica, nuclear, entre outras. Atualmente, a matriz energética brasileira é composta principalmente pela produção hídrica, que corresponde a aproximadamente 61%. O restante dos 39% corresponde às demais produções, como podemos identificar na Figura 1. Contudo, a malha energética hídrica é muito dependente das condições climáticas e possui um alto impacto ambiental durante a sua produção.

Figura 1 – Matriz elétrica brasileira em maio de 2017 de acordo com dados disponíveis no Banco de Informações de Geração da Agência Nacional de Energia Elétrica.



Fonte: Agência Nacional de Energia Elétrica

Uma fonte de energia pouco explorada no Brasil é a energia solar. Diferentemente da hídrica, ela tem alta disposição devido às condições geográficas, sol em abundância em regiões com sol em boa parte do ano como nas regiões Norte e Nordeste. Mesmo havendo tanta disparidade e diferenças de energias produzidas no Brasil, segundo (PEREIRA et al., 2006), a energia solar é uma das alternativas energéticas mais promissoras devido à sua inesgotável fonte quando levada em consideração a escala de tempo dos seres humanos.

Porém, durante a produção de energia elétrica, o módulo solar é influenciado por diversos fatores, sendo eles: temperatura da célula solar fotovoltaica, radiação solar, distribuição espectral e tipo de tecnologia empregada na fabricação do módulo solar (GASPARIN, 2009).

Este trabalho tem como proposta apresentar um sistema de aquisição de dados genérico aplicando os conceitos da Internet das Coisas em uma placa solar. Ele permite resolver um problema identificado no desenvolvimento de dispositivos de captação de dados no laboratório de conversão de energia durante o armazenamento local utilizando cartão SD. Esse sistema pode ser visualizado em tempo real via interface gráfica, no intuito de levantar parâmetros para futuros estudos na área de energia solar. Além disso, essa solução pode servir de base para outras aplicações de sensoriamento associadas à Internet das Coisas.

1.1 Objetivos

1.1.1 Geral

Desenvolver uma interface de rede aplicada a um estudo de caso aplicado a uma placa solar para monitorar em tempo real os dados de temperatura por meio de uma interface gráfica.

1.1.2 Específico

Garantir o envio de dados de sensores em uma placa solar retirando do armazenamento local via cartão SD para nuvem.

1.2 Estrutura do Documento

Este trabalho está estruturado em 6 capítulos, da seguinte forma:

- Capítulo 1 - Introdução: contém uma breve introdução sobre a motivação do presente trabalho, apresentando a Internet das Coisas e o panorama da malha energética brasileira;
- Capítulo 2 - Fundamentação Teórica: aborda toda a revisão da literatura especializada, reunindo os conceitos que foram fundamentais para a elaboração deste trabalho.
- Capítulo 3 - Materiais e Métodos: mostra todos os materiais utilizados para o desenvolvimento do trabalho e o método empregado.
- Capítulo 4 - Trabalho Correlatos: realiza um estudo de soluções correlatas, apresentando as semelhanças e as diferenças entre elas e a solução proposta.
- Capítulo 5 - Desenvolvimento da Interface de rede para monitoramento em tempo real: descreve a estratégia utilizada para o desenvolvimento da solução, apresentando o funcionamento da solução em etapas até alcançar o objetivo proposto.
- Capítulo 6 - Considerações Finais e Trabalhos Propostos: sintetiza os pontos positivos e os pontos negativos da solução em relação ao problema proposto pelo trabalho.

2

Fundamentação Teórica

Este capítulo aborda as ideias e os principais conceitos sobre o tema pesquisado, sintetizando os conceitos já publicados.

São abordados de forma encadeada, conceito de microcontroladores (*Raspberry Pi*, ESP32 e Arduíno), sensores (Conversor Digital e Termistores), protocolos de transmissão (MQTT e I2C), banco de dados (InfluxDB) e interface monitoramento de dados online (Grafana). No mercado atual existem soluções com os conceitos listados são chamados de sistemas de aquisição de dados (do inglês, *Data Acquisition and Transmission System- DATS*) porém devido ao seu alto custo e limitação de uso não será abordado nesse trabalho. Além disso, segundo (PEREIRA, 2018) um volume menor de artigos sobre redes de monitoramento IoT aplicadas a planta fotovoltaica foi encontrado na literatura.

2.1 Internet das Coisas(IoT)

A comunicação entre dispositivos na rede sem intervenção de humanos, segundo (SINGER, 2012) é a base do conceito de IoT. Tais dispositivos possuem diversas integrações sendo elas: comunicação via rádio, Bluetooth, Wi-Fi/3GPP, entradas analógicas, digitais entre outros. (ASHTON et al., 2009)

Esses dispositivos de alta integração e baixo consumo de energia vem inovando demandas e sendo inserido cada vez mais nos setores comerciais, industrias e residenciais. Segundo (MOHAMMADZADEH et al., 2018) sistemas IoT são a base para a sustentabilidade quando se refere ao termo Internet da Energia (do inglês, *Internet of Energy-IOE*), o qual consiste no monitoramento e controle do consumo de energias dos dispositivos.

Para a implementação de uma solução IoT é necessário utilizar um banco de dados para garantir o armazenamento. Para a aplicação na rede do IoT esse armazenamento é feito por um conjunto compartilhado de recursos computacionais remoto e gerenciáveis conhecidos como

computação em nuvem, em inglês *Cloud Computing*. Este modelo simplifica a complexidade do armazenamento local dos dados aumentando a eficácia e confiabilidade dos dados além de diminuir a complexidade e custos dos dispositivos em campo.

Modelos de serviços para computação em nuvem pode possuir modelo de desenvolvimento público, privado, comunitário ou híbrido. Além disso, modelos de serviços como Infraestrutura como Serviço (IaaS), Software como Serviço (SaaS) ou Plataforma como Serviço (PaaS). Nesse trabalho abordarmos e desenvolvemos uma solução privada com a plataforma PaaS, o que consiste em uma ferramenta, recursos e serviços focado no usuário final.

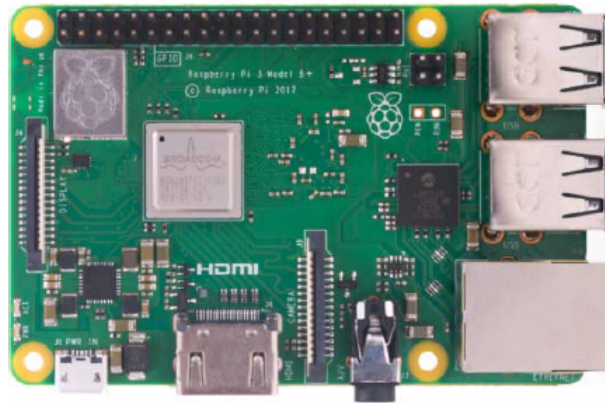
2.2 Microcontroladores

2.2.1 Raspberry Pi

Com a população da IoT, alguns sistemas operacionais baseados em Linux como o Debian foram adaptados e desenvolvidos pela comunidade formada por empresas, acadêmicos e armadores, distribuídos em todo mundo para integrar um sistema operacional a um sistema embarcado ([SANTOS et al., 2016](#)). Essa plataforma hoje executa operações em 8, 16 ou 32 bits, suporta linguagens como C, C++, Python, Shell e aplicações de computação em nuvem. Na Figura 2 podemos observar o microcontrolador que é equipado por:

- PU é de 64 bits com 1 GB de RAM (acesso aleatório memória);
- Contém o *chipset* Broadcom BCM2837B0;
- Vem com Quad-Core ARM Cortex-A53, 4 de 1,4 GHz núcleos;
- Consiste em um cabeçalho de 40 pinos (26 GPIOs);
- Áudio estéreo e vídeo composto são suportados por Conector jack de 3,5 mm;
- 4 portas USB 2.0;
- PoE (*Power Over Ethernet*) recurso importante incorporado a partir desse modelo;
- *Socket* Micro SD usado para aumentar a memória;
- Conector de alimentação micro USB;
- Saída de vídeo via HDMI;
- Interface de Câmera CSI;
- Recursos de conexão sem fio como Bluetooth e Wi-Fi integrado;

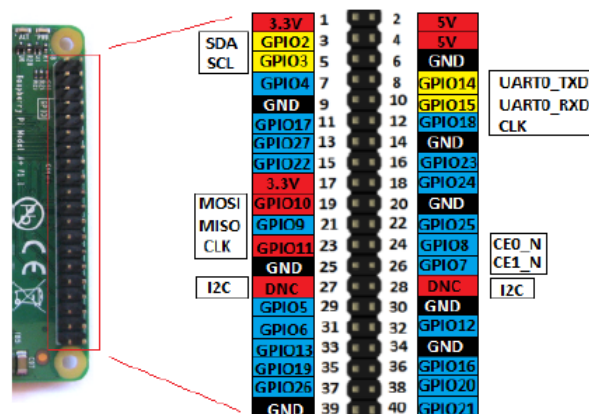
Figura 2 – Raspberry Pi Model 3B



Fonte: (VALOV; VALOVA, 2019)

Descrição de saídas do Raspberry Pi para interconexão com sensores, atuadores e comunicação podemos observar na Figura 3 podendo trabalhar com entradas de 5V ou 3.3V dependendo da entrada a ser trabalhada.

Figura 3 – Raspberry Pi Model 3B



Fonte: (VALOV; VALOVA, 2019)

Além de integrações com sensores e atuadores, uma das diversas aplicações que podemos aplicar este sistema embarcado é a hospedagem de serviços como *servers*. No presente trabalho, a Raspberry Pi será utilizada como concentrador armazenando os dados e apresentando ao usuário final os dados que estão sendo coleados via rede.

2.2.2 ESP32

O microcontrolador ESP32 foi projetado pela empresa *Espressif System* como podemos observar na Figura 4, lançado no ano de 2016 e hoje é considerável um dos mais robustos e notórios controladores do mercado. Tendo como uma forte característica velocidade de processamento, acessibilidade e conectividade, principalmente da sua conexão via Wi-Fi. (KOLBAN, 2017)

O processador constituído no microcontrolador é um *single* ou *dual-core* de 32-bit (com dois núcleos físicos de processamento) podendo trabalhar até na frequência de 240 MHz. (KOLBAN, 2018)

Figura 4 – Microcontrolador ESP32



Fonte: (KOLBAN, 2018)

A conectividade e integração do ESP32 é incorporada em seu chip transmissão sem fio, através de ondas de rádio, representadas por protocolos bluetooth e Wi-Fi. Possui uma conectividade em suas entradas digitais e analógicas de 3.3V, uma memória RAM de 520 Kb e 34 entradas GPIOs como podemos observar na Figura 5.

Figura 5 – Especificação ESP32 Geral

Especificações	Detalhes
Tensão	3.3 V
Memória RAM	520 Kb
Processador	single/dual-core 32 bit
GPIOs	34
Entradas conversoras Analógico/Digital	7

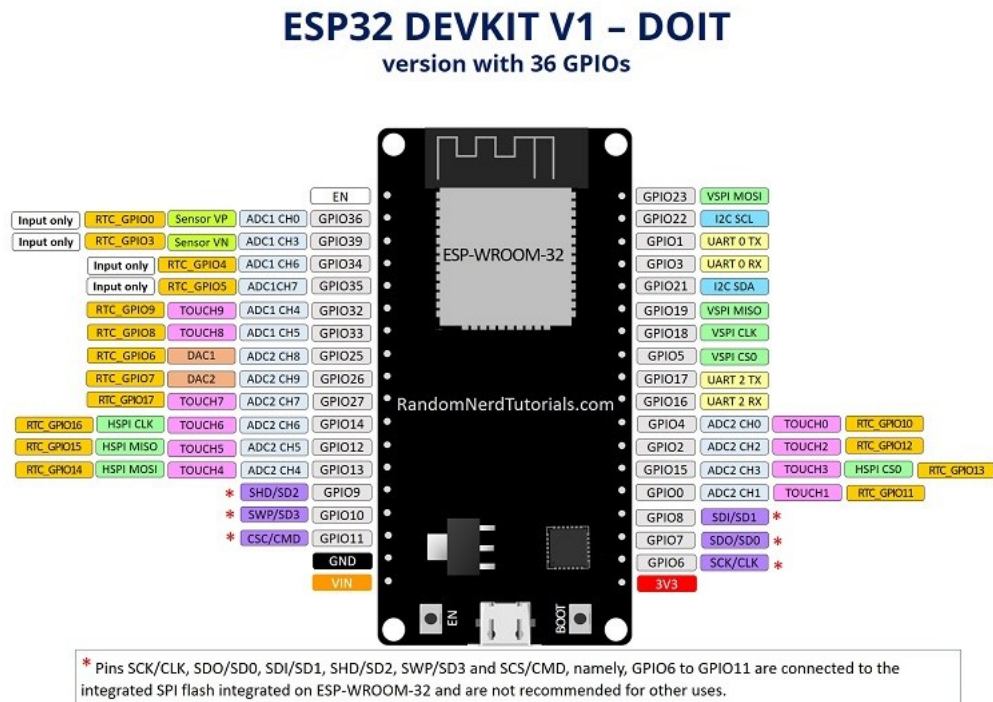
Fonte: (KOLBAN, 2018)

Com 34 portas de entradas e saídas o microcontrolador como podemos observar na Figura 6 é compatível com diversos protocolos de comunicação, sensores e atuadores. Tendo integração com diversos sensores e atuadores.

Além, do seu diferencial para seu principal concorrente o ATmel238 da Arduino, o qual possui uma memória ROM maior o qual pode armazenar dados e não se perder ao ser desligado. Tornando um diferencial para os demais chips em seu lançamento, na Figura 8 podemos observar o diagrama de blocos estrutural do microcontrolador.

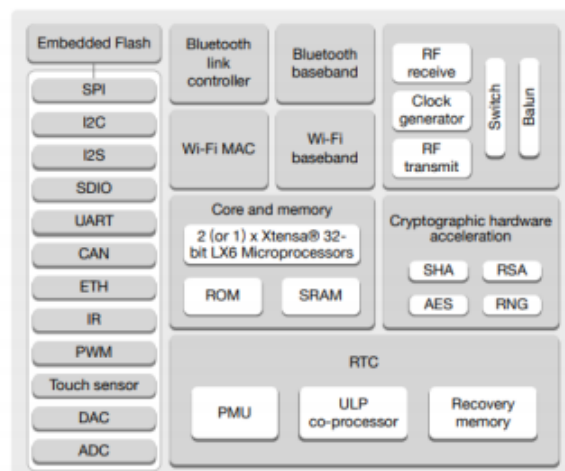
A programação do microcontrolador ESP32 pode ser desenvolvida através de vários softwares utilizando como base a linguagem C/C++. Utilizando o *software development kit* (SDK) fornecido pela própria desenvolvedora ou utilizando interfaces de programação (IDE) como Arduino IDE, Microsoft Code ou Eclipse. Podendo até realizar programação remota com bibliotecas desenvolvidas pela comunidade, por exemplo compilação de código para o microcontrolador via WiFi.

Figura 6 – Diagrama de Entradas e Saídas do ESP32 DEVKIT V01



Fonte: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

Figura 7 – Diagrama de Blocos do ESP32



Fonte: (KOLBAN, 2018)

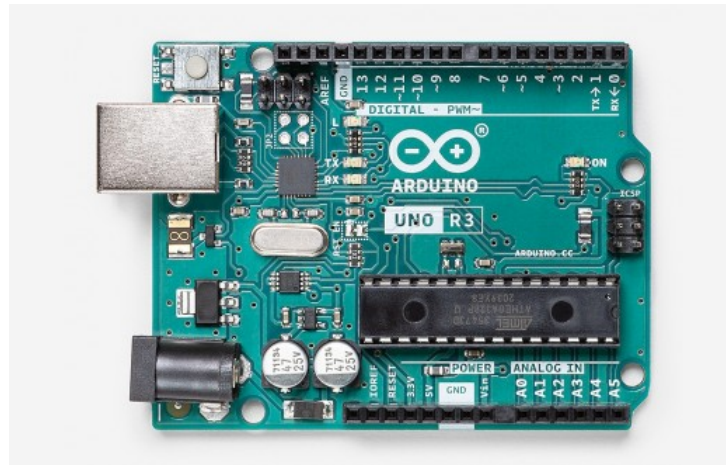
2.2.3 Arduíno

O microcontrolador ATmega328 integrado na placa do Arduíno constitui uma placa de desenvolvimento criada a princípio para fins acadêmicos lançada em 2005. Possuindo uma arquitetura aberta tanto de *hardware* e *software* sendo alimentada pela comunidade *open-source*.

Atualmente na versão do Arduíno Uno versão R3 possui em sua estrutura 14 entradas/saídas digitais, 6 saídas analógicas, conector *Universal Serial Bus*(USB), *In-Circuito Serial Programming*(ICSP). Como podemos observar na Figura (MONK, 2016) a placa descrita com

suas saídas e entradas.

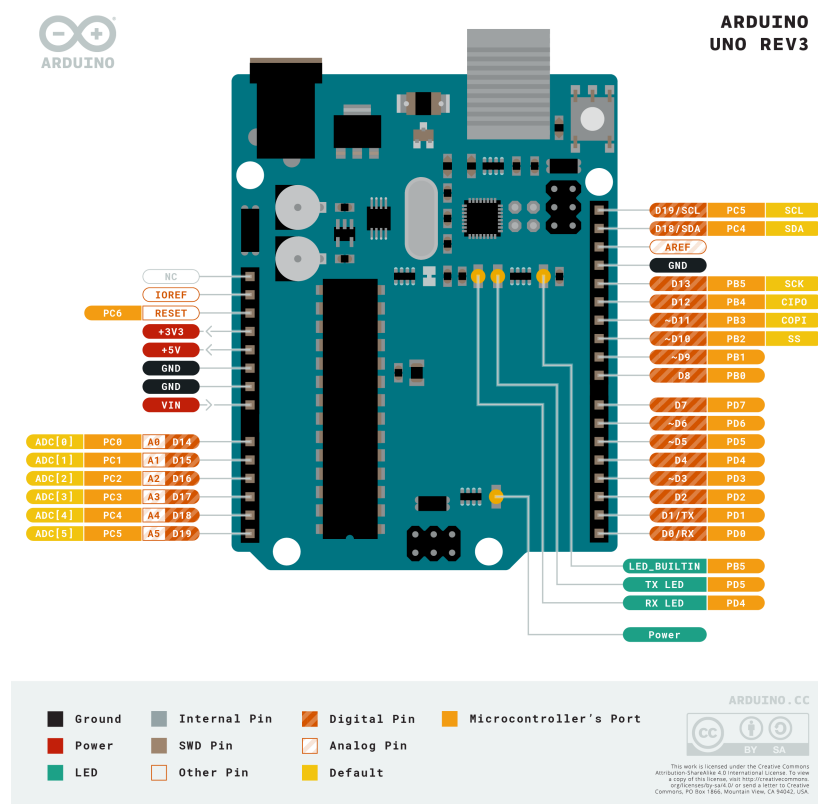
Figura 8 – Microcontrolador Arduino Uno R3



Fonte: (MONK, 2016)

Descrição física das saídas do Arduino podem ser observadas na Figura 9 com conjunto de entradas e saídas analógicas e digitais.

Figura 9 – Saídas e Entradas do Arduino Uno R3



Fonte: <https://store.arduino.cc/usa/arduino-uno-rev3>

Esse controlador pode ser programado *integrated development environment* (IDE) disponível no site da fabricante utilizando a linguagem base C/C++. Diferente de outros microcontroladores mais baixo nível como PIC o microcontrolador ATmega328 possui instalado em seu

firmware uma pré-gravação para um *bootloader* que permite o *upload* de novos códigos sem o uso de *hardware* externo. Se comunicando assim utilizando o protocolo STK500. (SANTOS; JUNIOR et al., 2019)

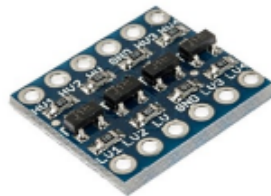
2.3 Sensores

2.3.1 Conversor Digital 5V para 3.3V

O módulo de conversor nível lógico bidirecional, é uma placa capaz de realizar um *step-down* de sinais de 5V para 3,3V e *step-up* de sinais de 3,3V para 5V. Contendo ao todo 2 canais bidirecionais e 2 canais omnidirecional. (OLIVEIRA, 2017)

Para realizar a comunicação entre os microcontroladores ESP32 com tensão de funcionamento 3.3V e o Arduíno com tensão de funcionamento de 5V será utilizado um conversor lógico bidirecional para ajustar os valores de tensão na comunicação e evitar danos físicos. Para que o conversor da Figura 10 funcione será necessário alimentar ambas as tensões *low volt* (LV) com 3.3V e *high volt* (HV) com 5V. Em seguida somente conectar as portas L1 com o ESP32 e a porta H1 com o Arduino assim sucessivamente para cada porta caso haja necessidade de comunicação.

Figura 10 – Conversor Digital Bidirecional 5v para 3.3v



Fonte: Elaborada pelo Autor.

2.3.2 Termistor

O sensor de temperatura NTC também conhecido como termistor é um semicondutor sensível a temperatura. Capaz de exibir uma grande alteração de resistência proporcional a pequenas variação de temperatura. Geralmente essa proporção apresenta coeficientes de temperatura negativa, o que significa que a resistência do termistor diminui com a queda de temperatura aumenta. (PATSKO, 2006)

Na Figura 11 podemos observar as especificações do termistor, nesse trabalho iremos utilizar o NTC 10D-9 o qual está destacado. Sua precisão é de aproximadamente 0,2 °C podendo operar entre 0°C a 100°C.

Figura 11 – Especificação do parâmetro elétrico de um termistor

Electronic parameter Specification					
Part No.	R25°C (Ω)	Max.Steady State Current (A)	Approx.R @ Max.Cur (Ω)	Dissipation factor (δ)(mW/°C)	Thermal time constant (s)
NTC-5D5	5	1	0.353	6	20
NTC-10D5	10	0.7	0.771	6	20
NTC-60D5	60	0.5	1.878	6	18
NTC-200D5	200	0.1	6.259	6	18
NTC-5D7	5	2	0.283	10	30
NTC-8D7	8	1	0.539	9	28
NTC-10D7	10	1	0.616	9	27
NTC-12D7	12	1	0.816	9	27
NTC-16D7	16	0.7	1.003	9	27
NTC-22D7	22	0.6	1.108	9	27
NTC-33D7	33	0.5	1.485	10	28
NTC-200D7	200	0.2	6.233	11	28
NTC-3D9	3	4	0.120	11	35
NTC-4D9	4	3	0.190	11	35
NTC-5D9	5	3	0.210	11	34
NTC-6D9	6	2	0.315	11	34
NTC-8D9	8	2	0.400	11	32
NTC-10D9	10	2	0.458	11	32
NTC-12D9	12	1	0.652	11	32
NTC-16D9	16	1	0.802	11	31

Fonte: Datasheet Exsense - PT Series.

2.4 Servidor

2.4.1 Backend

2.4.1.1 Protocolo de comunicação

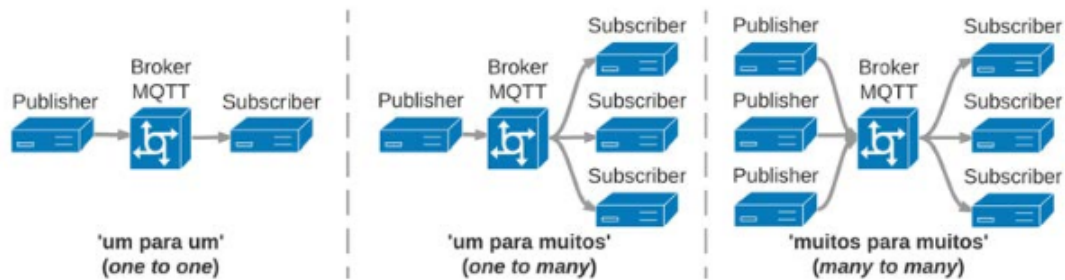
A comunicação entre IoT está sempre em uma constante evolução e o mesmo ainda se encontra em aberto qual protocolo irá se tornar padrão no mercado. (TORRES; ROCHA; SOUZA, 2020) No mercado atual existem soluções com MQTT, CoAP, XMPP, SNMP, WAMP, I2C e entre outros. Todos os protocolos podem ser utilizados para implementações IoT, porém nesse trabalho o protocolo utilizado será o MQTT e I2C.

2.4.1.1.1 MQTT

Desenvolvido em 1999 pela IBM, MQTT (*MQ Telemetry Transport*), é um protocolo aberto de mensagens projetado para a M2M, o qual suporta alta latência, instabilidade na comunicação e baixa largura de banda. (TORRES; ROCHA; SOUZA, 2020)

O protocolo MQTT usa como base TCP e o padrão de mensagens *publisher/subscriber* (publicador/assinante), onde todos os dados são enviados para um intermediário chamado de *broker* que é responsável por receber e entregar as mensagem aos destinatários. Essa estrutura permite modularizar o produtor do cliente podendo ter várias abordagens como de um para um (*one-to-one*), um para muitos (*one-to-many*) ou muitos para muitos (*many-to-many*). Como podemos observar na Figura 12.

Figura 12 – Formas de distribuição da mensagem pelo protocolo MQTT



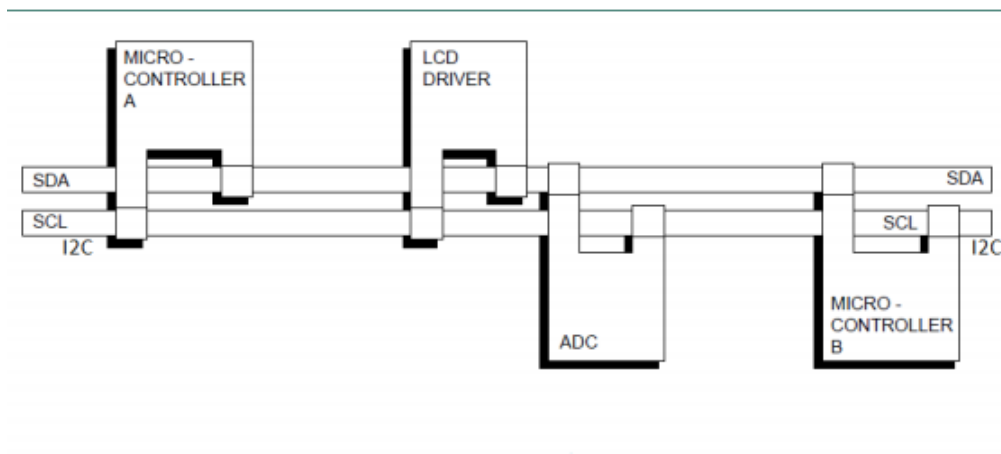
Fonte: (TORRES; ROCHA; SOUZA, 2020)

Neste trabalho iremos utilizar a estrutura *one to one* (um para um) onde o *publisher* será nossa interface de rede e o *subscriber* será nosso servidor onde irá armazenar os dados e apresentar para o usuário final.

2.4.1.1.2 I2C

Desenvolvido pela Phillips o I2C (Inter-Integrated Circuit) é utilizado para conectar dispositivos via serial com baixa velocidade a um sistema embarcado com funcionamento de multi dispositivos como primário e secundário como pode-se observar na Figura 13. Utilizando duas linhas de barramentos bidirecional um responsável pelo envio de dados e outro pelo sinal de *clock* gerado pelo primário sendo eles respectivamente: SDA (*Serial Data*) e SCL(*Serial Clock*). (BEZZON et al., 2018)

Figura 13 – Arquitetura de referência do I2C



Fonte: Modificado I2C-bus specification and user manual

2.4.1.2 Banco de Dados

Banco de dados são conjuntos de informações sobre um domínio específicos e agrupados eles podem ser do tipo relacional e série temporal. Os bancos de dados relacionais são apontados

como pouco escalonáveis e isso acontece devida a sua propriedade ACID (Atomicidade, Consistência, Isolamento e Durabilidade) o que resulta uma maior complexidade em suas instruções. (JUNGES et al., 2017)

É possível alcançar escalabilidade desse modelo de banco porém há um custo elevado e uma maior complexidade para execução. Devido aos problemas de conciliar uma demanda de maior escalabilidade, melhor desempenho e possuir menor tolerância.

A solução mais utilizada é de bancos de dados de séries temporais os quais principalmente resolvem problemas de escalabilidade segundo (JUNGES et al., 2017). Nesse trabalho iremos utilizar o InfluxDB como banco não relacional.

2.4.1.2.1 InfluxDB

O InfluxDB é um banco de dados *open-source* do tipo série temporal designado a trabalhar com alto volumes de escritas e consultas. Ele pode ser gerenciadas pelas ferramentas como Telegraf, Chronograf e Kapacitor. (CASTRO et al.,)

Devido à sua alta capacidade de demanda de leitura e escrita no banco umas das maiores aplicações desse banco de dados é em monitoramento em tempo real. Dessa forma performando muito melhor que banco de dados relacionais. (JUNGES et al., 2017)

2.4.2 Front-end

Camada de apresentação disponibiliza uma interface Web para visualização dos dados por meio de um *dashboard*. Os dados poderão ser visualizados em forma de gráficos, planilhas ou alertas de incidentes. Nesse trabalho iremos utilizar o Grafana como sugestão de acesso pela documentação do próprio projeto do InfluxDB.

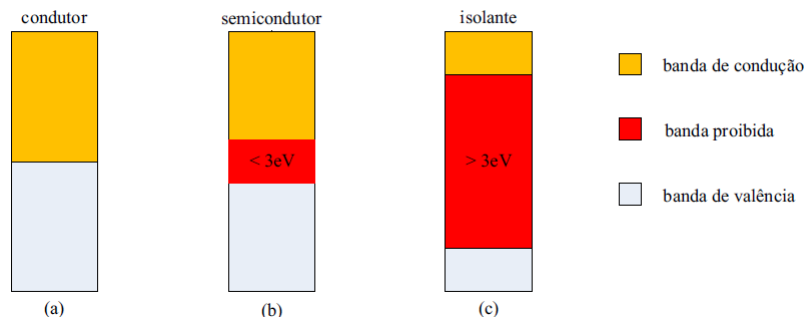
2.4.2.1 Grafana

O Grafana é uma interface web do tipo *open-source* multiplataforma e de visualização interativa. Fornece tabelas, alertas, gráficos e a extensões a diversas fontes suportadas. (MACEDO et al., 2021)

2.5 Célula Solar

Célula solar ou célula fotovoltaica é um dispositivo sensível a luz capaz de converter luz do sol diretamente em energia elétrica por meio de um efeito fotovoltaico. O conjunto de células solares formam unidades chamadas de módulos solares ou placas solares. Cada conjunto de célula solar é formado por material semicondutores estruturado como podemos perceber na Figura 14. (PINHO; GALDINO, 2014)

Figura 14 – Estrutura de bandas de energia em (a) condutores, (b) semicondutores e (c) isolantes



Fonte: (PINHO; GALDINO, 2014)

Devido a uma característica da banda de energia nos semicondutores o efeito fotovoltaico se dá com a condutividade elétrica com a temperatura proporciona uma excitação térmica na banda de valência com isso dependendo da excitação o mesmo por conduzir ou isolar. Para valores de silício (Si) a concentração de portadores intrínsecos (n_i) encontrada o qual é citado na literatura (PINHO; GALDINO, 2014) é da dimensão de $1,45 * 10^{10} / cm^3$ na temperatura de 300 K.

Com isso em uma célula solar pode ser produzida com materiais diferentes e com dopagem distintas. Sendo elas de acordo com a dosagem ou material usado como podemos observar na Figura 15.

Figura 15 – Eficiência das células fotovoltaicas fabricadas em laboratórios até 2019 segundo

Classification	Efficiency, %	Area, cm ²	V _{oc} , V	J _{sc} , mA/cm ²	Fill Factor, %	Test Centre (date)	Description
Silicon							
Si {crystalline cell}	26.7 ± 0.5	79.0 {da}	0.738	42.65 ^a	84.9	AIST {3/17}	Kaneka, n-type rear IBC ⁴
Si {multicrystalline cell}	22.3 ± 0.4 ^b	3.923 {ap}	0.6742	41.08 ^c	80.5	FhG-ISE {8/17}	FhG-ISE, n-type ⁵
Si {thin transfer submodule}	21.2 ± 0.4	239.7 {ap}	0.687 ^d	38.50 ^{d,e}	80.3	NREL {4/14}	Solexel {35 µm thick} ⁶
Si {thin film minimodule}	10.5 ± 0.3	94.0 {ap}	0.492 ^d	29.7 ^{d,f}	72.1	FhG-ISE {8/07}	CSG Solar {<2 µm on glass} ⁷
III-V cells							
GaAs {thin film cell}	29.1 ± 0.6	0.998 {ap}	1.1272	29.78 ^a	86.7	FhG-ISE {10/18}	Alta Devices ⁸
GaAs {multicrystalline}	18.4 ± 0.5	4.011 {t}	0.994	23.2	79.7	NREL {11/95}	RTI, Ge substrate ⁹
InP {crystalline cell}	24.2 ± 0.5 ^b	1.008 {ap}	0.939	31.15 ^a	82.6	NREL {3/13}	NREL ¹⁰
Thin film chalcogenide							
CIGS {cell}	22.9 ± 0.5	1.041 {da}	0.744	38.77 ^h	79.5	AIST {11/17}	Solar Frontier ^{11,12}
CdTe {cell}	21.0 ± 0.4	1.0623 {ap}	0.8759	30.25 ^e	79.4	Newport {8/14}	First Solar, on glass ¹³
CZTSSe {cell}	11.3 ± 0.3	1.1761 {da}	0.5333	33.57 ^g	63.0	Newport {10/18}	DGIST, Korea ¹⁴
CZTS {cell}	10.0 ± 0.2	1.113 {da}	0.7083	21.77 ^a	65.1	NREL {3/17}	UNSW ¹⁵
Amorphous/microcrystalline							
Si {amorphous cell}	10.2 ± 0.3 ^{ib}	1.001 {da}	0.896	16.36 ^e	69.8	AIST {7/14}	AIST ¹⁶
Si {microcrystalline cell}	11.9 ± 0.3 ^b	1.044 {da}	0.550	29.72 ^a	75.0	AIST {2/17}	AIST ¹⁶
Perovskite							
Perovskite {cell}	20.9 ± 0.7 ^{ij}	0.991 {da}	1.125	24.92 ^c	74.5	Newport {7/17}	KRICT ¹⁷
Perovskite {minimodule}	17.25 ± 0.6 ^{il}	17.277 {da}	1.070 ^d	20.66 ^{d,h}	78.1	Newport {5/18}	Microquanta, 7 serial cells ¹⁸
Perovskite {submodule}	11.7 ± 0.4 ⁱ	703 {da}	1.073 ^d	14.36 ^{d,h}	75.8	AIST {3/18}	Toshiba, 44 serial cells ¹⁹
Dye sensitised							
Dye {cell}	11.9 ± 0.4 ^{ik}	1.005 {da}	0.744	22.47 ⁿ	71.2	AIST {9/12}	Sharp ²⁰
Dye {minimodule}	10.7 ± 0.4 ^{il}	26.55 {da}	0.754 ^d	20.19 ^{d,o}	69.9	AIST {2/15}	Sharp, 7 serial cells ²¹
Dye {submodule}	8.8 ± 0.3 ⁱ	398.8 {da}	0.697 ^d	18.42 ^{d,o}	68.7	AIST {9/12}	Sharp, 26 serial cells ²²
Organic							
Organic {cell}	11.2 ± 0.3 ^a	0.992 {da}	0.780	19.30 ^e	74.2	AIST {10/15}	Toshiba ²³
Organic {minimodule}	9.7 ± 0.3 ^a	26.14 {da}	0.806 ^d	16.47 ^{d,o}	73.2	AIST {2/15}	Toshiba {8 series cells} ²³

Fonte: (GREEN et al.,)

Cada tipo de módulo solar a depender da dopagem e material de semiconductor terá uma eficácia de produção específica. Porém, independente de sua tecnologia as propriedades de uma célula solar não se altera.

2.6 Propriedades de uma Célula Solar

2.6.1 Curva I-V

A curva corrente e tensão de uma célula solar pode ser obtida diante a soma da corrente da junção *pn* no escuro com a corrente gerada pelos fótons. (PINHO; GALDINO, 2014) A descrição dada entre a função de corrente e tensão, pode ser denominada também de curva característica semelhante a curva de um diodo semiconductor. Com isso, por se tratar de uma junção *pn* podemos descrever a Curva I-V utilizando a Equação de Shockley, o qual representa um diodo ideal:

$$I = I_L + I_0 \cdot [e^{\frac{qV}{nkT}} - 1] \quad (2.1)$$

No qual temos que as variáveis da Equação 2.1 descritas por: I_L - corrente fotogerada (A), I_0 -

corrente de saturação reversa do diodo (A), n - fator de idealidade do diodo, q – carga do elétron $1,6 \times 10^{-19} C$, k – constante de Boltzmann $1,38 \times 10^{-23} J/K$ e T – temperatura absoluta (K).

Por verificação da equação da Equação de Shockley, temos que $I_L = 0$ quando estiver em um ambiente sem luminosidade, o qual irá se comportar similar a um díodo. Já no caso da corrente reversa I_0 , o mesmo é dependente de vários fatores como propriedades do material de construção, dosagem do material e detalhes de construção. Porém, a corrente reversa pode ser obtida experimentalmente através de uma tensão de circuito aberto (célula sem estar ligada em nada), com isso por inspeção da Equação de Shockley podemos obter o valor experimental da corrente reversa. (PINHO; GALDINO, 2014)

$$I_0 = q \cdot A \cdot n_i^2 \cdot \left(\frac{D_p}{L_p \cdot N_d} + \frac{D_n}{L_n \cdot N_a} \right) \quad (2.2)$$

No qual temos que as variáveis da Equação 2.2: I_0 - corrente de saturação reversa do diodo (A), A – área da seção reta da junção (área da célula), n_i - concentração de portadores intrínsecos no material, N_d, N_a – Concentração dos dopantes tipo n e tipo p, respectivamente, D_p, D_n – Coeficientes de difusão de lacunas e elétrons, L_p, L_n – Comprimentos de difusão de lacunas e de elétrons e q – carga do elétron ($1,6 \times 10^{-19} C$). Em uma curva I-V ou curva característica podemos identificar dados como: I_{sc} : Corrente de curto circuito, V_{oc} : Tensão de circuito aberto, P_{MP} : Potência Máxima ou de pico, I_{MP}, V_{MP} : Tensão de corrente no ponto de máxima potência.

2.6.2 Associação de células solares em um módulo fotovoltaico

2.6.2.1 Associação em Série

A associação em série de células solares é quando o terminal negativo de um módulo está conectado no positivo do módulo seguinte. Possuindo somente um ponto de conexão entre eles. Tal combinação desses módulos de gerarão uma consequência semelhante como é explicado com as associações de resistores.

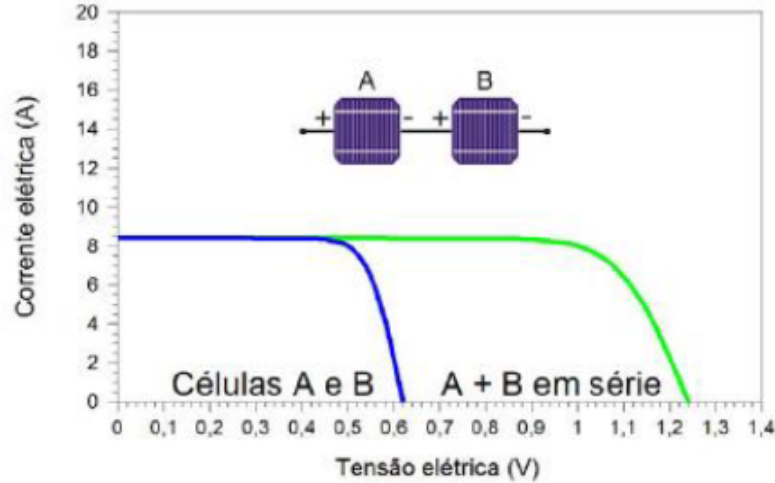
A associação em série implicará em uma mesma corrente passante nas células solares, porém a tensão total será a soma das tensões em cada módulo. Como podemos observar nas equações a seguir:

$$V = V_1 + V_2 + V_3 + \dots + V_n \quad (2.3)$$

$$I = I_1 = I_2 = I_3 = \dots = I_n \quad (2.4)$$

A influência da associação em série na curva corrente vs tensão de uma associação de células solares pode ser identificada na Figura 16.

Figura 16 – Associação em série de células solares.



Fonte: (PINHO; GALDINO, 2014)

2.6.2.2 Associação em Paralelo

Como acontece na associação em paralelo de resistores, a associação de células solares em paralelo é quando o polo positivo é conectado no mesmo polo positivo da seguinte célula. O mesmo acontece com os polos negativos (PINHO; GALDINO, 2014). A propriedade de corrente e tensão se assemelha como no fenômeno dos resistores, como podemos observar nas equações a seguir:

$$V = V_1 = V_2 = V_3 = \dots = V_n \quad (2.5)$$

$$I = I_1 + I_2 + I_3 = \dots = I_n \quad (2.6)$$

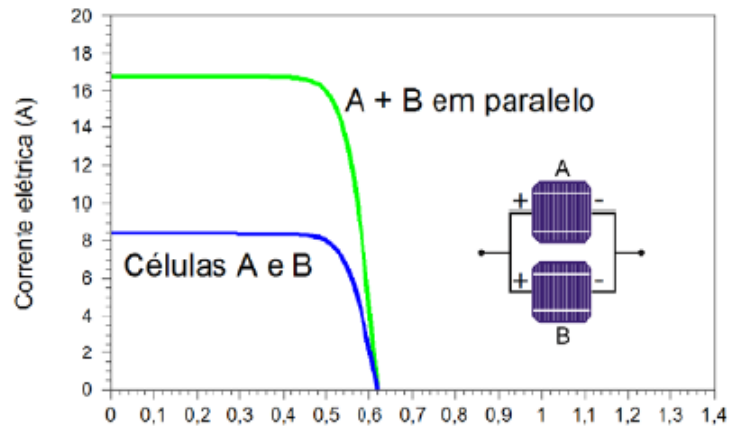
Como se pode observar a associação em paralelo de células solares comporta como um divisor de corrente mantendo a mesma tensão nos terminais. O comportamento dessa associação em paralelo de células solares na curva corrente vs tensão pode ser observado na Figura 17.

2.6.3 Influência da Irradiação Solar

A influência da irradiação solar em uma célula solar está relacionada linearmente a geração de corrente elétrica. (PINHO; GALDINO, 2014) A relação de uma célula ou de um módulo da corrente de curto circuito com a irradiação pode ser obtida de acordo a Equação 2.7.

$$I_{sc} = I_{sc_{stc}} * \frac{G}{1000} \quad (2.7)$$

Figura 17 – Associação em paralelo de células solares.

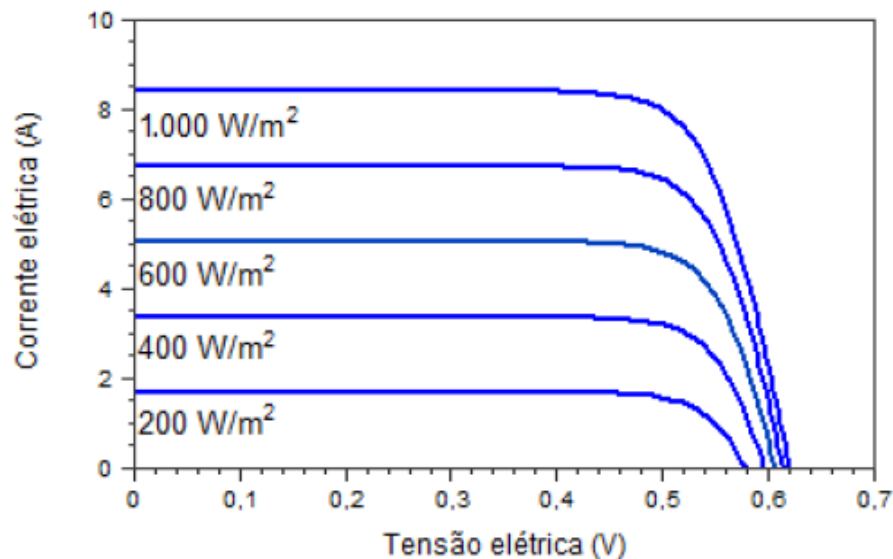


Fonte: (PINHO; GALDINO, 2014)

No qual temos que as variáveis da Equação 2.7: $I_{sc}(A)$ - Corrente de curto-circuito do módulo, para irradiância G e uma temperatura de $25\text{ }^{\circ}\text{C}$, $I_{sc_{stc}}(A)$ - Corrente de curto circuito do modulo nas STC, $G(W/m^2)$ - Irradiância incidente sobre o módulo, $1000(W/m^2)$ – Irradiância nas STC.

Pode-se observar a influência da irradiação solar na Figura 18 quando maior W/m^2 maior a corrente.

Figura 18 – Influência da variação da irradiância solar na curva característica I-V de uma célula fotovoltaica de silício cristalino em condições STC.

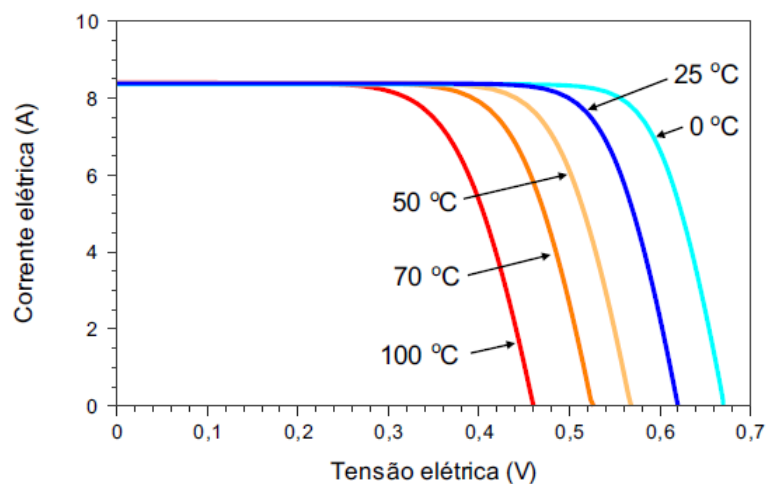


Fonte: (PINHO; GALDINO, 2014)

2.6.4 Influência da Temperatura

A temperatura influencia diretamente o desempenho da célula solar. Pois, dependendo da temperatura a qual o semicondutor esteja exposto o mesmo haverá uma variação de concentração de portadores intrínsecos com isso variando a tensão que é gerada no semicondutor. No *datasheets* das células solares as condições ideais exposto dos módulos é de 25°C, nesse caso influenciará diretamente na tensão do módulo. Pode-se perceber que quanto maior a temperatura exposta no módulo solar menor será a tensão produzido, sendo assim uma relação diretamente proporcional (PINHO; GALDINO, 2014). Como podemos observar na Figura 19 representada pelo Gráfico I-V.

Figura 19 – Influência da temperatura da célula fotovoltaica na curva I-V (para irradiância de 1.000 W/m², espectro AM1,5)



Fonte: (PINHO; GALDINO, 2014)

Como pode se observar os coeficientes térmicos irão influenciar diretamente na produção da célula solar, porém esse fator depende de diversos parâmetros como tipo de célula solar, região onde está a célula solar, local onde está instalado e tamanho da placa solar. Com isso, há uma extrema importância no monitoramento da temperatura das placas solares.

3

Revisão de trabalhos correlatos

3.1 Revisão de trabalhos acadêmicos

- O sistema de aquisição de (COELHO¹ et al., 2013) foi utilizado um microcontrolador PIC16F874A/877A, que se comunica via interface USB por meio de RS232 comunicação serial com um computador para o armazenamento local. Ele é capaz de monitorar corrente, tensão e temperatura.
- O sistema baseado em LabVIEW™ para simulação e monitoramento fotovoltaico descrito por (SILVESTRE; CHOUDER; KARATEPE, 2013) fornece um sistema rápido e confiável em tempo real para análise de desempenho de FV porém a solução por ser comercial há um custo alto de investimento e com baixa escalabilidade de aplicação de sensores. Com armazenamento local comunicando com um computador dedicado para coletas e consultas dos dados local.
- O trabalho proposto por (DZUNG et al., 2014), utiliza os inversores de uma planta FV se comunicando via ZigBee para um sistema central, que envia os dados monitorados via cabo Ethernet para um computador central. O monitoramento é feito localmente via HTTP em uma página HTML do próprio fabricante do embarcado instalado no inversor o STM32F407.
- O sistema de monitoramento proposto por (GAD; GAD, 2015) é baseado em Arduíno com controle local e armazenamento em um cartão SD. Suportando até 16 sensores, para medição de temperatura aplicada a sistemas solares se comunicando de forma serial sem transmissão de dados via internet.
- A implementação desenvolvida pelo (CHOI et al., 2018) de um transmissor com Arduino e um receptor com Raspberry Pi utilizando antena Lora embarcada para o monitoramento

de dados de tensão, corrente, temperatura e baterias. Focando em sua primeira versão 2015 em uma comunicação via Lora, atualizando para uma versão com maior estabilidade na comunicação em 2017 utilizando o microcontrolador somente a Raspberry Pi e comunicação focada no protocolo MQTT. Porém todo seu armazenamento permaneceu local.

- O artigo desenvolvido ([PRAMONO; SARI; MAULANA, 2017](#)) um sistema de monitoramento de corrente e tensão para obtenção da potência máxima gerada por módulos FV. Utilizando sistema embarcado microcontrolador ESP8266, que se comunica diretamente via internet com WiFi embarcado. Para leitura dos sensores é utilizado o Arduino que se comunica com ESP8266 via serial para transmissão dos dados e o mesmo realiza o armazenamento local dos dados via cartão SD. O banco de dados e interface web utilizado foram *Thingspeak* o qual existe uma limitação até uma quantidade máximas de sensores a serem alocados de forma gratuita.
- O trabalho em ([PULIPAKA et al., 2017](#)), implementa uma solução com Raspberry Pi para leitura de dados no próprio inversor. Se comunicando via Bluetooth com Raspberry Pi e enviando os dados via RS-485. Sua visualização via web localmente, com armazenamento de dados na memória do inversor ou no banco de dados local no Raspberry Pi.
- O projeto proposto ([ZAGO; FRUETT, 2017](#)) consiste em um sistema de baixo custo para medir tensão e corrente geradas por um módulo FV. No qual vários nós de sensores se comunicam entre si até que a informação chegue no centralizador que envia os dados para nuvem. O centralizador é um microcontrolador Raspberry Pi 3, que se comunica com os nós via Xbee Zigbee que nesse caso são Arduinos Uno R3 coletando os dados dos sensores. Toda a solução é baseada no padrão IEEE 802.15.4 com distância de comunicação em linha reta de até 50m.
- O sistema proposto por ([ERRAISSI et al., 2018](#)) monitora os dados meteorológicos de um inversor. O microcontrolador se baseia em um PcDuino e armazena os dados em cartão SD card localmente e envia os dados através da internet. A combinação em Arduino e Raspberry Pi opera com uma maior escalabilidade de sensores no Arduino e uma solução mais robusta e maior poder de processamento devido o sistema operacional Linux no Raspberry Pi.

3.2 Considerações Sobre os Trabalhos Relacionados

Os trabalhos de comparação de DATs e de um sistema de monitoramento IoT a partir da revisão bibliográfica percebe-se que muito deles realizam sensoriamento de várias propriedades de células solares como temperatura, tensão, corrente, irradiância e entre outros. Porém, dentro de todas as soluções percebeu-se uma possível fonte de problema identificada no laboratório durante o armazenamento local como pode se observar na tabela.

O diferencial desse trabalho é o foco na descentralização da informação localmente, sendo enviado para um servidor remoto para que o usuário possa acessar os dados remotamente sem se preocupar com a perda de dados durante o processo de armazenamento físico.

ASPECTOS CONSIDERADOS	(COELHO <i>et al.</i> , 2013)	(CHOUDEUR <i>et al.</i> , 2013)	(DZUNG <i>et al.</i> , 2014)	(GAD; GAD, 2015)	(CHOI <i>et al.</i> , 2015)	(CHOI <i>et al.</i> , 2017)	(PRAMANO, 2017)	(ZAGO; FRUETT, 2017)	(SANTOS <i>et al.</i> , 2017)	(ERRAÏSSI <i>et al.</i> , 2018)	(SOLUÇÃO PROPOSTA)
Rede de Monitoramento	Sim	Não	Sim	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim
Aquisição de dados	PIC16F874A/877A	Datalogger Agilent	Inversores FV	Arduino	Raspberry Pi	Raspberry Pi	Inversores FV	Arduino Uno	Arduino Mega	Inversores FV e PcDuino (Arduino + Raspberry Pi)	Arduino Uno + ESP32
Processamento de dados	Computador Dedicado	Computador Dedicado	STM32F407 com coordenador CC2530ZNP	PIC	Arduino	Raspberry Pi	Raspberry Pi	Raspberry Pi	Arduino Mega	PcDuino	Raspberry Pi (Servidor Remoto)
Comunicação entre os sistemas embarcados	RS232	GPIO bus (IEEE-488)	Zigbee	Serial	LoRa	SUN (Smart Utility Network)	Bluetooth ou RS 485 ou Ethernet	Xbee Zigbee	Zigbee	Bluetooth	Local (I2C) Remoto (MQTT)
Quantidade de Sensores Adicionados	3	8	4 por nó	16	4	2 por nó	Nenhum	2 por nó	16	4 por nó	8 por dispositivos 112 dispositivos por barramento
Variáveis de medidas	Tensão e Corrente CC e CA, Temperatura	Temperatura Ambiente, Irradiância, Tensão e Corrente CC e CA;	Tensão e Corrente CC e CA	Temperatura	Tensão, corrente e Temperatura	Tensão e Corrente CC da String FV	Tensão e Corrente CC da String FV	Tensão e Corrente CC do Módulo FV	Tensão e Corrente CC e CA, Temperatura e Irradiância Solar	Temperatura, irradiância, velocidade e direção do vento, tensão e corrente CC e CA	Temperatura, porém é possível escalar outros sensores
Intervalo de Medida	Não é citado	1 minuto	Não é citado	5 minutos	Não é citado	Não é citado	Não é citado	10 segundos	Não é citado	5 minutos	1 minuto
Armazenamento de Dados	PC Dedicado	PC Dedicado	PC Dedicado	Local em Cartão SD	Banco de dados MongoDB	Local em Cartão SD e Banco de dados ThingSpeak	Memória local do inversor e banco de dados local	Banco de dados local	Banco de Dados Local	Cartão SD e Banco de Dados Local	Armazenamento Remoto (Banco de dados InfluxDB)
Transmissão dos dados à Internet	Não	Não	Ethernet	Não	MQTT via WiFi	WiFi	Ethernet	Não	WiFi	Ethernet	WiFi
Visualização dos dados	Excel em PC Dedicado	LabVIEW™ em PC dedicado	Servidor Local no STM32	Excel em PC Dedicado	Android App	Página ThingSpeak na Web	Servidor Web local	Página HTML embarcada no RPi	Servidor Local	Servidor Local embarcado no PcDuino	Servidor Remoto Web no Grafana

Tabela 1 – Levantamento de Requisitos de DATs aplicados em placas fotovoltaicas. Fonte: Elaborada pelo Autor.

Soluções comerciais apresentam desvantagens como alto custo, pouca memória e necessidade de uma conexão de um PC localmente aumentando a complexidade de uma aplicação em campo. Além de algumas soluções citadas na Tabela 1 possui baixa flexibilidade na programação, exigindo software proprietários para interface, poucas entradas de sensores a solução proposta nesse trabalho tem como foco apresentar uma maior escalabilidade de sensores, descentralização do armazenamento de dados e acesso remoto aos dados monitorados.

4

Método e Materiais Usados

4.1 Método

A abordagem utilizada para evitar erros durante o desenvolvimento do projeto foi a abordagem sistemática proposta por Jain (1991), a qual foi aplicada nesse trabalho. A abordagem é separada em passos os quais é necessário segui-los.

Primeiro passo é a resposta a um problema, o qual deve ser claro, pois ele influencia diretamente a definição dos objetos e do sistema que será proposto.

Segundo passo é o desenvolvimento de uma diagrama abstrato da solução que poderá ser validado mais a frente.

Terceiro passo são as métricas de critério para desempenho os quais segundo Jain (1991), neste caso a validação da comunicação entre o server e o client remoto para validação do envio remoto de dados.

Quarto passo é a elaboração de passos para incrementar na solução remota os sensores localmente.

Quinto passo, trata-se da escolha de fatores para estudo e parâmetros os quais podem sofrer variações durante a pesquisa. Recomenda-se iniciar o mesmo com poucos fatores e aumentar a lista conforma a necessidade.

Sexto passo é a seleção de técnica de avaliação. O qual existe três técnicas: a simulação, a modelagem analítica e a implementação. A técnica dependerá do tempo e dos recursos que disponíveis para solução do problema.

Sétimo passo, escolha da carga, o qual consiste no caso do projeto desenvolvido a lista de sensores que no caso serão termistores aplicado a um estudo de casos que serão enviados via MQTT para o servidor.

Oitavo passo, planejar os experimentos com a lista de fatores e níveis, deve-se estabelecer uma sequência para o experimento o qual para obter o máximo de informações sobre a solução com mínimo de esforço realizado.

Nono passo, consiste na interpretação dos dados o qual pode-se ser aplicado conceitos de verificar se o problema central foi resolvido.

Décimo passo, e último consiste na apresentação do resultado. Na mesma etapa deve-se atentar à apresentação final da avaliação.

4.2 Material

4.2.1 Sistemas Embarcados

4.2.1.1 Arduíno

Módulo receptor dos dados dos sensores e responsável para a transmissão via serial para o ESP32.

4.2.1.2 ESP32

Módulo transmissor de dados para o servidor utilizando o protocolo MQTT.

4.2.2 Raspberry PI

Módulo servidor onde estará alocado o servidor para recebimento, armazenamento e apresentação dos dados.

4.3 Sensores

4.3.1 Abaixador de Tensão

Sensor step-down de 5V para 3.3V bidirecional para comunicação entre os microcontroladores ESP32 e Arduíno.

4.3.2 Sensor de Temperatura

Sensor termistor NTC 10D-9.

5

Desenvolvimento da Interface de rede para monitoramento em tempo real

Neste capítulo apresenta-se a prototipação da solução referente a interface de rede para soluções *IoT* e a aplicação de um estudo de caso em um rastreador solar.

5.1 Aplicação da Metodologia

Para que a aplicação da metodologia seja efetiva é necessário definir o objetivo da solução assim como o escopo do sistema, serviços oferecidos e a técnica de avaliação. Como serão mostrados a seguir:

Objetivo:

- Desenvolvimento de uma interface de comunicação local entre os microcontroladores;
- Envio dos dados para servidor remoto;
- Monitoramento e apresentação dos dados coletados;
- Armazenamento de dados em banco de dados remoto;

Sistema:

- O sistema corresponde a um *Software* e *Hardware* os quais respectivamente:
 - Envia para a nuvem para coletar os dados dos sensores que estão sendo monitorados via *Grafana* o qual interage com o usuário via interface web.
 - Realiza a comunicação local via serial utilizando o protocolo I2C.

Serviço:

- Monitoramento dos dados coletados em tempo real;
- Alertas em caso de dados críticos dos sensores;
- Histórico de dados dos sensores;

Técnica de avaliação:

- Medição de disponibilidade das coletas realizadas;

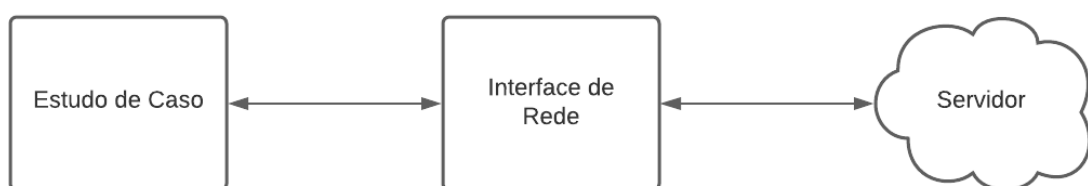
Etapas do Desenvolvimento:

- Diagrama de Blocos do Projeto;
- Desenvolvimento da Interface de Rede;
- Desenvolvimento do Servidor;
- Simulação de Coleta de Dados;
- Aplicação de Estudo de Caso em Rastreador Solar;

5.2 Diagrama de Blocos do Projeto

Para o desenvolvimento e entendimento da solução foi projetado um diagrama de bloco simplificado. O diagrama foi separado em três blocos sendo eles: servidor, interface de rede e estudo de caso. O servidor é responsável pela recepção dos dados de sensoriamento, armazenamento e apresentação dos dados, interface de rede é uma solução de sensoriamento via porta serial com a comunicação via protocolo i2c localmente e via Wi-Fi para o servidor via protocolo MQTT. Já o estudo de caso pode ser uma aplicação genérica de sensoriamento utilizando um microcontrolador que trabalhe com uma tensão de 5V ou 3.3V como podemos observar na Figura 20 a estrutura da solução. O foco desse trabalho é apresentar uma solução e estrutura para monitoramento de soluções de internet das coisas resolvendo o problema com armazenamento local.

Figura 20 – Diagrama de Blocos Simplificado da Solução.

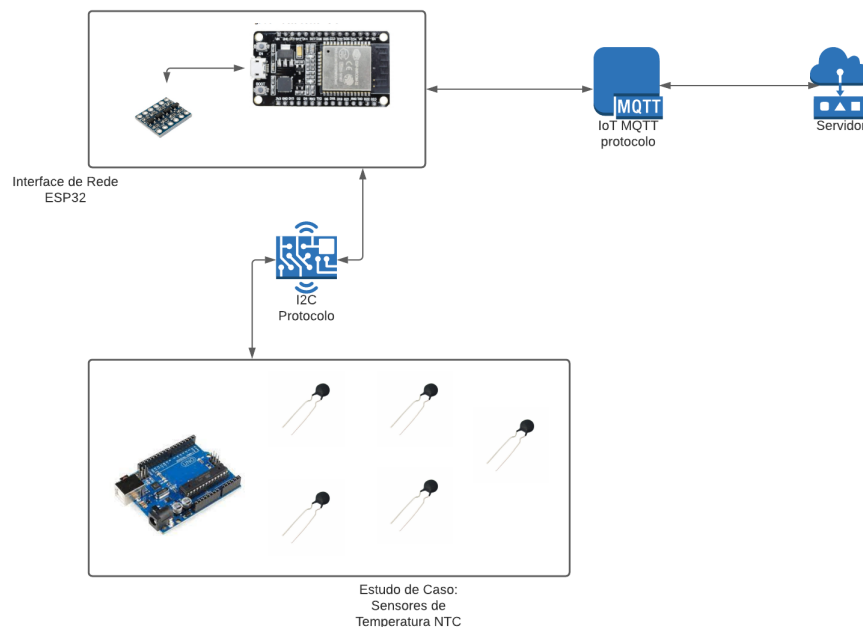


Fonte: Elaborada pelo autor.

No caso do estudo de caso esse trabalho será aplicado em uma placa solar monitorando os dados de temperatura durante a exposição solar. Simplificando uma das suas aplicações que seriam em um rastreador solar com diversos sensores e assim validando a solução proposta. Esse armazenamento no microcontrolador escolhido é o Arduino não é possível realizar o acompanhamento dos dados em tempo real, verificar os armazenados sem interromper as coletas e além disso o calor em campo aumenta os riscos de problemas físicos no armazenamento como o aquecimento do cartão SD. Esses problemas motivou o desenvolvimento de uma interface de rede para acompanhamento dos dados em tempo real, um armazenamento descentralizado do local experimental e uma interface visual com gráficos e tabela para estudos. Para simplificar as coletas serão coletados três dados de sensores do rastreamento solar como irradiância difusa, temperatura e tensão da placa solar no rastreador. No entanto, vale ressaltar que a solução pode ser escalável para mais sensores.

Para o entendimento do estudo de caso e interface de rede podemos observar na Figura 21 que a Interface de Rede será composto por um ESP32 e um conversor abaixador de tensão *step-down* para receber os dados e enviar para o servidor já no estudo de caso será utilizado um Arduino para coletar os dados de sensores NTC distribuidos em uma placa solar e enviar via protocolo I2C para o primário.

Figura 21 – Diagrama de Blocos em campo



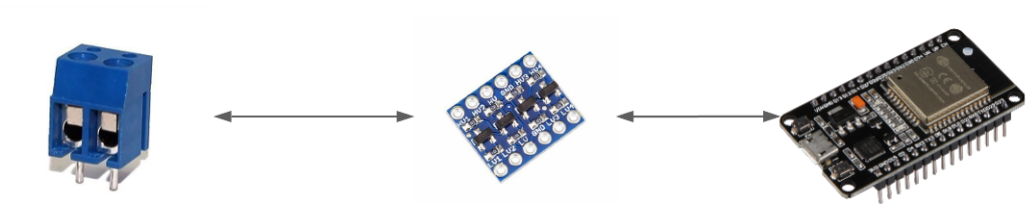
Fonte: Elaborada pelo autor.

Quando a informação chega no servidor, ele é armazenado no banco de dados *InfluxDB* que por sua vez ele é consultado por um código em *Python* que realizara a ponte entre o MQTT e o banco de dados. Depois disso, a interface web do Grafana no *front-end* irá realizar consultas no banco e apresentar em forma de gráficos ou tabelas em tempo real com *delay* de 30 segundos. Cada bloco será desenvolvido em sessões nesse trabalho para melhor entendimento.

5.3 Desenvolvimento da Interface de Rede

Durante o desenvolvimento da interface de Rede foi abordado em três partes: Física, Programação e Protocolos. A conexão física como pode se observar na Figura 22 consiste em um abaixador de tensão de 5V para 3.3V que realiza a conexão físicas entre outros dispositivos IoT para recebimento dos dados de sensores.

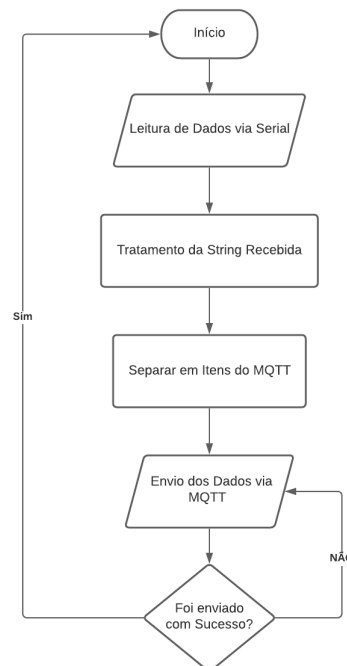
Figura 22 – Diagrama da Interface de Rede



Fonte: Elaborada pelo autor.

A programação como podemos observar na Figura 23 consiste em receber os dados via porta serial do tipo *string* e realizar a quebra e tratamento desses dados para serem enviados para o servidor.

Figura 23 – Fluxograma Código da Interface de Rede



Fonte: Elaborada pelo autor.

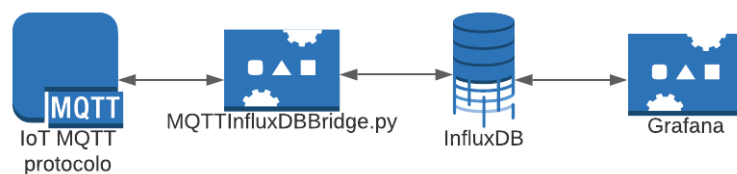
A comunicação via protocolo como pode-se observar na Figura 21 consiste localmente via serial utilizando o protocolo I2C. Essa comunicação será responsável pelo envio dos dados

de sensores localmente. Já a comunicação com servidor ou remota será utilizado o protocolo MQTT que utilizará a conexão WiFi disposta no local do experimento.

5.4 Desenvolvimento do Servidor

O servidor configurado em uma *raspberry pi* possui uma estrutura de serviços dispostas como pode-se observar na Figura 24. O qual será responsável pelo recebimento dos dados da interface de rede via protocolo *MQTT*.

Figura 24 – Diagrama de Blocos do servidor

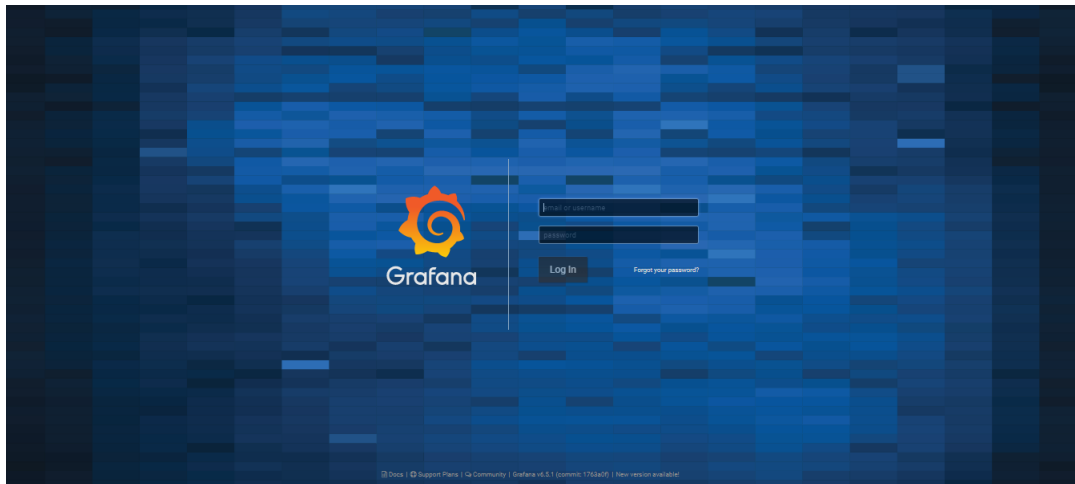


Fonte: Elaborada pelo autor.

Esses dados proveniente da nuvem serão publicados em um tópico definido na interface de rede para o host server para broker do MQTT. Porém, o MQTT por ser um protocolo leve de mensagens e simples não possui mecanismos de armazenamento de dados em caso de desligamento do servidor. Com isso, foi desenvolvido um script em python que realizará a ponte entre os tópicos de dados de sensores em MQTT para o banco de dados InfluxDB de série temporal para armazenamento dos dados. Esse script está rodando como serviço no servidor e monitorado pelo daemon de serviços do Linux para caso haja parada de serviço e não perca dados que chegam nos tópicos do MQTT.

Por fim, a consulta dos dados, visualização e acompanhamento dos dados em tempo real foi implementado uma solução *open-source* o *Grafana* como podemos observar na Figura 25. O qual irá apresentar os dados para o usuário final de forma multiplataforma sem a necessidade de está no local do experimento e em tempo real com até um minuto de delay a depender das coletas definidas na comunicação entre a interface de rede.

Figura 25 – Interface de Acesso do Grafana



Fonte: Elaborada pelo autor.

5.5 Implementação da solução

Para realização da implementação como pode observar os blocos na Figura 20 a solução foi dividida em três partes. O primeiro bloco consiste na configuração do servidor e sua documentação. Englobando tarefas como configuração de rede, protocolos, interface web, banco de dados, scripts e serviços do servidor. Toda a solução foi documentada em repositório Github para futuras replicações.

Sobre os demais dois blocos, por se tratar de situações de hardware e software foi dividido os problemas para validação da programação, comunicação via protocolos e sensoramento.

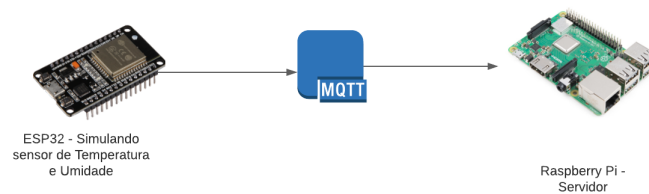
A primeira implementação foi a validação da interface de rede com o envio de dados simulados via ESP32 utilizando o protocolo MQTT com o intuito de validar a comunicação via rede para o servidor. Após validado, foi implementado a comunicação via serial entre o Arduino e ESP32 com foco de validar em código e a comunicação do protocolo I2C.

E por fim, foi aplicado no estudo de caso proposto nesse trabalho sensores de temperatura NTC em uma placa solar para coleta de sensores durante a exposição de uma placa solar ao sol e validação da solução.

5.5.1 Envio de dados via ESP32 para o Servidor

Para realização dos testes simulado os dados de sensores temperatura e umidade no ESP32 (Figura 26) foram simulados partindo de 20 e 80 respectivamente e sendo incrementado 0.01 a cada 30 segundos. Para que fosse possível validar a coleta e apresentação de dados no *Grafana*.

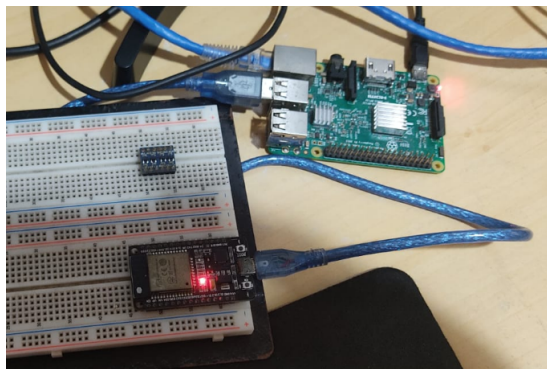
Figura 26 – Estrutura de Simulação de Coleta de Dados via ESP32 para o Servidor



Fonte: Elaborada pelo autor.

A estrutura física foi montada em bancada local para validação com podemos observar na Figura 27 realizando toda comunicação via Wi-Fi.

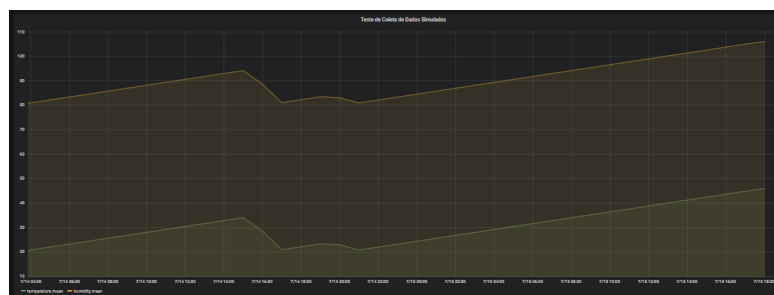
Figura 27 – Bancada de teste de Coleta de Dados via ESP32 para o Servidor



Fonte: Elaborada pelo autor.

O gráfico da coleta dos dados e da simulação podem ser observados na Figura 28, vale ressaltar que os dados são simulados de sensores, portanto, não são reais nesse teste.

Figura 28 – Gráfico de visualização de dados simulados no Grafana

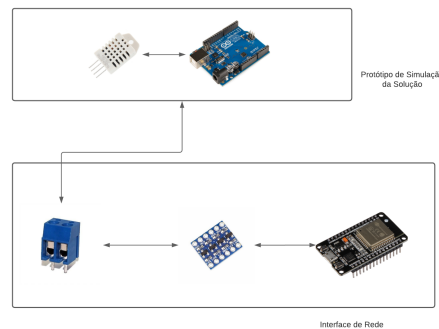


Fonte: Elaborada pelo autor.

5.5.2 Envio de dados via Arduino para o Servidor

Para a simulação da coleta de dados foi colocado um DHT22 em um Arduino em bancada para homologação da solução para testar a comunicação entre os microcontroladores e a comunicação e coleta do servidor como podemos observar na Figura 29.

Figura 29 – Fluxograma Código da Interface de Rede

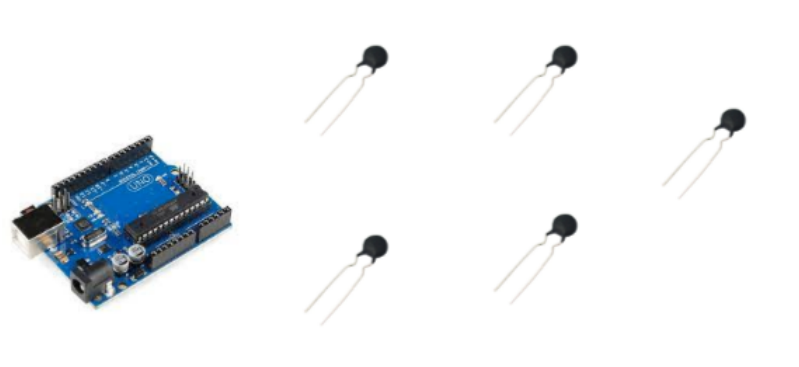


Fonte: Elaborada pelo autor.

5.6 Aplicação de Estudo de Caso em uma Placa Solar

Para implementação do estudo de caso foi utilizado um arduino uno com cinco sensores de temperatura NTC disposto como pode observar na Figura 30.

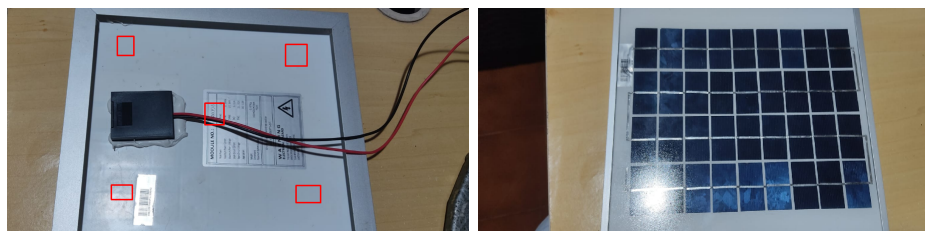
Figura 30 – Microcontrolador para coleta dos dados



Fonte: Elaborada pelo autor.

Para coleta de dados esses sensores serão distribuídos como se observa na Figura 31 para uma maior cobertura da placa solar. Durante a exposição solar os valores de temperatura distribuídos na placa serão coletados pelo microcontrolador e enviados para o servidor pela solução proposta.

Figura 31 – Distribuição em vermelho de termistores em uma placa solar



Fonte: Elaborada pelo autor.

Para melhor aderência dos sensores no módulo solar foi utilizado cola quente nos termistores isolando as pernas dos sensores como pode-se observar na Figura 32. Para melhor

observar o comportamento térmico iremos utilizar a placa em circuito aberto para que seu ponto de operação possua um maior valor em temperatura.

Figura 32 – Termistores aplicado no modulo solar



Fonte: Elaborada pelo autor.

Após a aplicação dos termistores foi integrado a solução na interface de rede para monitoramento validado por esse trabalho, como pode-se observar na Figura 33 o módulo fotovoltaico a esquerda e a direita a solução de monitoramento de rede local Arduino + EsP32.

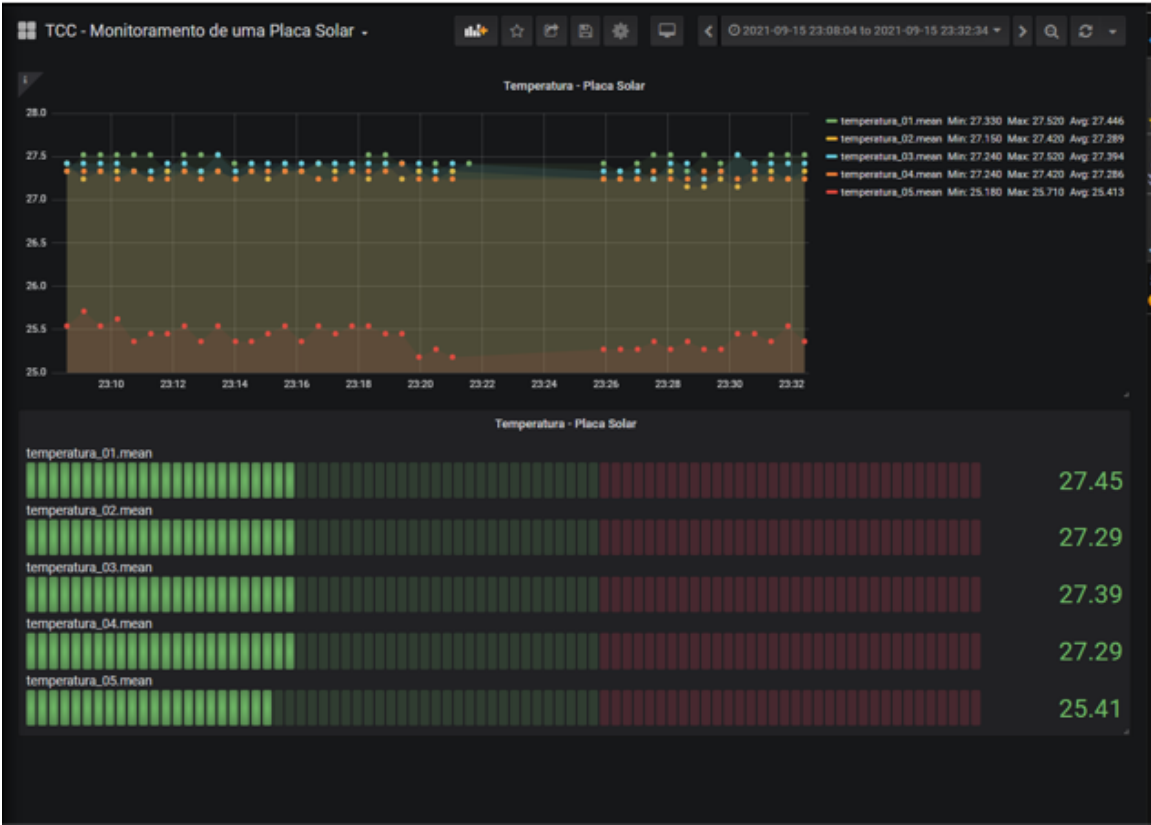
Figura 33 – Termistores aplicado no modulo solar



Fonte: Elaborada pelo autor.

Os dados coletados pela interface de monitoramento podem ser observadas na Figura 34 remotamente com gráfico plotado, assim validando a solução em hardware e software.

Figura 34 – Apresentação de dados do Grafana coletados da interface de rede



Fonte: Elaborada pelo autor.

6

Conclusão

O sistema desenvolvido neste Trabalho de Conclusão de Curso para monitoramento IoT, substituindo o armazenamento local, consiste primeiro na estruturação da solução e depois a aplicação de um estudo de caso em uma placa monitorando os dados de temperatura. Isso permitiu montar um sistema de baixo custo escalonável para diversos outros tipos de sensores baseada em software livre e com distribuição *online*, bem como disponibilizar a sua utilização de forma gratuita para diversos projetos e pesquisas. A gravação *online* de dados é feita on-line por meio dos microcontroladores ESP32 e Arduíno, com servidor remoto baseado em Linux embarcado na *Raspberry Pi*.

Este é um sistema escalonável que também pode ser expandido para coleta de parâmetros provenientes de outros tipos de sensores, sejam eles analógicos ou digitais. Para a implementação da solução, é necessário conhecimento básico em linguagem de programação C (para o ESP32 e Arduino) e Linux (para o *Raspberry Pi* para a implementação e modificações, caso necessário).

A interface web desenvolvida utilizando o Grafana é efetiva para consultas de dados *on-line* dos dados em tempo real. Por ser uma solução web de código aberto, possui acessibilidade em multiplataformas independentemente do sistema operacional do usuário e podendo até ser acessado via celular em um navegador.

A rede de monitoramento foi construída em bancada e estruturada em pequenas partes, integrando a comunicação dos microcontroladores com sensores simulados para validação da solução. Como segunda etapa, foram colocados sensores de temperatura em uma placa solar para verificar o comportamento e a coleta de dados dos sensores.

O envio dos dados do Arduíno e ESP32 foi realizado de maneira serial utilizando o protocolo I2C, uma comunicação serial baseada na unidade primária responsável pelo envio dos dados para o servidor (nesse caso, ESP32), e a secundária, responsável pela coleta de dados dos sensores (nesse caso, Arduíno). O envio de dados para servidor foi realizado via conexão Wi-Fi para um servidor remoto.

Nos resultados obtidos, foi possível averiguar que os dados dos sensores foi enviado com sucesso na interface web. Porém, é necessário na solução proposta uma estabilidade na conexão devido à falta de armazenamento local para sincronização de dados em caso de falta de conexão. Um problema comum nas soluções verificadas no estudo de caso de trabalhos correlatos. Devido ao fechamento do escopo, a proposta tem o foco de substituir o armazenamento local devido aos problemas de temperatura e perda de dados durante o armazenamento de dados.

Portanto a solução proposta neste trabalho é possível resolver a necessidade e o problema no armazenamento do grupo de pesquisa, Realizando a substituição do armazenamento local via cartão SD combinado com o envio de dados para rede e monitoramento dos experimentos remotamente. Como proposta para trabalhos futuros, é sugerido melhorias para problemas novos levantados devido a essa abordagem de transmissão de dados remotamente que foram levantadas pelo próprio grupo de pesquisa e avaliados na revisão de trabalhos correlatos, sendo elas:

- Implementação de armazenamento local na EPROM do ESP32 para futura sincronização em caso de perda de conexão com a internet para o servidor;
- Comunicação *full duplex* entre *server* e o microcontrolador secundário, retirando toda a base de cálculos para calibração caso seja necessário, e centralização no microcontrolador com maior base de processamento, diminuindo o consumo de energia em campo;
- Implementação de regras de segurança no protocolo MQTT e criptografia de dados para maior segurança.
- Armazenamento local via cartão SD no Arduíno e transmissão de dados via I2C no ESP32

Referências

ASHTON, K. et al. That ‘internet of things’ thing. *RFID journal*, v. 22, n. 7, p. 97–114, 2009. Citado na página 17.

BEZZON, G. et al. Desenvolvimento de um sistema de automação e controle envolvendo dispositivos embarcados com comunicação em rede de protocolo i2c para o ensino técnico em mecatrônica. *Revista dos Trabalhos de Iniciação Científica da UNICAMP*, n. 26, 2018. Citado na página 25.

CASTRO, L. F. S. de et al. Integrando sistemas multi-agentes embarcados, simulaç ao urbana e aplicaç oes de iot. Citado na página 26.

CHOI, C.-S. et al. Lora based renewable energy monitoring system with open iot platform. In: IEEE. *2018 international conference on Electronics, Information, and Communication (ICEIC)*. [S.l.], 2018. p. 1–2. Citado na página 33.

COELHO¹, F. et al. Desenvolvimento de um sistema de aquisição de dados de baixo custo para monitoramento de sistemas de geração fotovoltaica. In: *Congresso Norte Nordeste de Pesquisa e Inovação*. [S.l.: s.n.], 2013. p. 1. Citado na página 33.

DZUNG, P. Q. et al. Real-time communication network solution based on zigbee and ethernet for photovoltaic systems. In: IEEE. *2014 9th IEEE Conference on Industrial Electronics and Applications*. [S.l.], 2014. p. 197–202. Citado na página 33.

ERRAISSI, N. et al. Implementation of a low-cost data acquisition system for “propre. ma” project. *Measurement*, Elsevier, v. 117, p. 21–40, 2018. Citado na página 34.

GAD, H.; GAD, H. E. Development of a new temperature data acquisition system for solar energy applications. *Renewable energy*, Elsevier, v. 74, p. 337–343, 2015. Citado na página 33.

GASPARIN, F. P. Desenvolvimento de um traçador de curvas características de módulos fotovoltaicos. 2009. Citado na página 15.

GREEN, M. A. et al. Solar cell efficiency tables (version 53). *Progress in Photovoltaics: Research and Applications*, v. 27, n. 1, p. 3–12. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/pip.3102>>. Citado na página 28.

JUNGES, L. W. et al. Estudo sobre os bancos de dados para séries temporais no contexto de iot. Blumenau, SC, 2017. Citado na página 26.

KOLBAN, N. Kolban’s book on esp32. USA: *Leanpub*, 2017. Citado na página 19.

KOLBAN, N. Kolban’s book on esp32, september 2018. Texas, USA, 2018. Citado 2 vezes nas páginas 20 e 21.

MACEDO, G. M. de et al. Solução de iot para monitoramento elétrico de ups em ambiente de data center. *Brazilian Journal of Development*, v. 7, n. 3, p. 29691–29708, 2021. Citado na página 26.

- MOHAMMADZADEH, A. K. et al. A fuzzy analytic network process (fanp) approach for prioritizing internet of things challenges in iran. *Technology in Society*, Elsevier, v. 53, p. 124–134, 2018. Citado na página 17.
- MONK, S. *Programming Arduino: getting started with sketches*. [S.l.]: McGraw-Hill Education, 2016. Citado 2 vezes nas páginas 21 e 22.
- OLIVEIRA, R. R. Uso do microcontrolador esp8266 para automação residencial. *Rio de Janeiro: UFRJ Escola Politécnica*, 2017. Citado na página 23.
- PATSKO, L. F. Tutorial: Aplicações, funcionamento e utilização de sensores. *Maxwell Bohr: Instrumentação eletrônica*, p. 41, 2006. Citado na página 23.
- PEREIRA, E. B. et al. *Atlas brasileiro de energia solar*. [S.l.]: Inpe São José dos Campos, 2006. v. 1. Citado na página 15.
- PEREIRA, R. I. S. Rede de monitoramento iot aplicada em plantas de microgeração fotovoltaica. 2018. Citado 2 vezes nas páginas 14 e 17.
- PINHO, J. T.; GALDINO, M. A. Manual de engenharia para sistemas fotovoltaicos. *Rio de Janeiro*, v. 1, p. 47–499, 2014. Citado 7 vezes nas páginas 26, 27, 28, 29, 30, 31 e 32.
- PRAMONO, S. H.; SARI, S. N.; MAULANA, E. Internet-based monitoring and protection on pv smart grid system. In: IEEE. *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*. [S.l.], 2017. p. 448–453. Citado na página 34.
- PULIPAKA, S. et al. Non-intrusive real-time monitoring of pv generation at inverters using internet of photovoltaics. *Electronics Letters*, IET, v. 53, n. 16, p. 1137–1138, 2017. Citado na página 34.
- SANTOS, B. P. et al. Internet das coisas: da teoria à prática. 2016. Citado na página 18.
- SANTOS, J. W.; JUNIOR, L. et al. *Sistema de automatização residencial de baixo custo controlado pelo microcontrolador ESP32 e monitorado via smartphone*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2019. Citado na página 23.
- SILVESTRE, S.; CHOUDER, A.; KARATEPE, E. Automatic fault detection in grid connected pv systems. *Solar energy*, Elsevier, v. 94, p. 119–127, 2013. Citado na página 33.
- SINGER, T. Tudo conectado: conceitos e representações da internet das coisas. *Simpósio em tecnologias digitais e sociabilidade*, v. 2, p. 1–15, 2012. Citado na página 17.
- TORRES, A. B.; ROCHA, A. R.; SOUZA, J. N. de. Análise de desempenho de brokers mqtt em sistema de baixo custo. In: SBC. *Anais do XV Workshop em Desempenho de Sistemas Computacionais e de Comunicação*. [S.l.], 2020. p. 47–58. Citado 2 vezes nas páginas 24 e 25.
- VALOV, N.; VALOVA, I. Raspberry pi as a tool to combine different courses part of university education. In: IEEE. *2019 18th International Conference on Information Technology Based Higher Education and Training (ITHET)*. [S.l.], 2019. p. 1–5. Citado na página 19.
- ZAGO, R. M.; FRUETT, F. A low-cost solar generation monitoring system suitable for internet of things. In: IEEE. *2017 2nd International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*. [S.l.], 2017. p. 1–6. Citado na página 34.