



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA - DEEL

Curso: ENGENHARIA ELECTRÓNICA-Laboral

Estagio Profissional

Proposta de um sistema de monitoramento e controle do consumo de energia eléctrica com base em IoT para o Instituto Medio de Ciências e Gestão

Autor:

João Júnior Massingue

Supervisor:

Eng^o Frederico Constantino Zile

Maputo, Dezembro de 2024



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA - DEEL

Curso: ENGENHARIA ELECTRÓNICA-Laboral

Estagio Profissional

Proposta de um sistema de monitoramento e controle do consumo de energia eléctrica com base em IoT para o Instituto Medio de Ciências e Gestão

Autor:

João Júnior Massingue

Supervisor:

Engº Frederico Constantino Zile

Maputo, Dezembro de 2024



GUIA DE AVALIAÇÃO DO RELATÓRIO ESCRITO

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

F1 - GUIA DE AVALIAÇÃO DO RELATÓRIO ESCRITO

Nome do estudante: _____

Referência do tema: _____ Data: ___/___/___

Título do tema: Desenvolvimento de sistema de controle automático de iluminação e tomadas de uma residência

1. Resumo					
1.1. Apresentação dos pontos chaves no resumo (clareza, organização, correlação com o apresentado)	1	2	3	4	5
Secção 1 subtotal (max: 5)					

2. Organização (estrutura) e explanação										
2.1. Objectivos	1	2	3	4	5					
2.2. Introdução, antecedentes e pesquisa Bibliográfica	1	2	3	4	5	6	7	8	9	10
2.3. Metodologias	1	2	3	4	5	6	7	8	9	10
2.4. Resultados, sua análise e discussão	1	2	3	4	5	6	7	8	9	10
2.5. Conclusões e aplicação dos resultados (recomendações)	1	2	3	4	5	6	7	8	9	10
Secção 2 subtotal(max: 45)										

3. Argumentação										
3.1. Criatividade e originalidade	1	2	3	4	5					
3.2. Rigor	1	2	3	4	5					
3.3. Análise crítica, evidência e lógica	1	2	3	4	5	6	7	8	9	10
3.4. Relação objectivos/ métodos/ resultados/conclusões	1	2	3	4	5					
3.5. Relevância	1	2	3	4	5					
Secção 3 subtotal(max: 30)										

4. Apresentação e estilo da escrita					
4.1. Legibilidade e organização	1	2	3	4	5
4.2. Ilustração e qualidade das figuras e tabelas	1	2	3	4	5
4.3. Estilo da escrita (fluência do texto, uso da língua e gramática)	1	2	3	4	5
4.4. Fontes bibliográficas (citação correcta, referências, etc)	1	2	3	4	5
Secção 4 subtotal(max: 20)					

Total de pontos (max:)

Nota (=Total*0,2)

Nota: quando exista a componente gráfica (desenhos técnicos), a nota acima é multiplicada por 0,8 cabendo os restantes 20% do peso à referida parte gráfica.

ANEXO 11.
GUIA DE AVALIAÇÃO DA APRESENTAÇÃO ORAL E DEFESA
(PELO JÚRI)



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
 DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

F2 – GUIA DE AVALIAÇÃO DA APRESENTAÇÃO ORAL E DEFESA

Nome do estudante: _____
 Referencia do tema: _____ Data: ___ / ___ / ___

1. Introdução										
1.1. Apresentação dos pontos chaves na introdução (Contexto e importância do trabalho)	1	2	3	4	5	6	7	8	9	10
Secção 1 subtotal(max: 10)										

2. Organização e explanação										
2.1. Objectivos	1	2	3							
2.3. Metodologia	1	2	3	4						
2.4. Resultados, sua análise e discussão	1	2	3	4	5	6	7	8	9	10
2.5. Conclusões e aplicação dos resultados (recomendações)	1	2	3	4	5	6	7	8		
Secção 2 subtotal(max: 25)										

3. Estilo da apresentação										
3.1. Uso efectivo do tempo	1	2	3	4	5					
3.2. Clareza, tom, vivacidade e entusiasmo	1	2	3	4	5					
3.3. Uso e qualidade dos audio-visuais	1	2	3	4	5					
Secção 3 subtotal(max: 15)										

4. Defesa										
4.1. Exactidão nas respostas	1	2	3	4	5	6	7	8	9	10
4.2. Domínio dos conceitos	1	2	3	4	5	6	7	8	9	10
4.3. Confiança e domínio do trabalho realizado	1	2	3	4	5	6	7	8	9	10
4.4. Domínio do significado e aplicação dos Resultados	1	2	3	4	5	6	7	8	9	10
4.5. Segurança nas intervenções	1	2	3	4	5	6	7	8	9	10
Secção 3 subtotal(max: 50)										

Total de pontos (max: 100)		Nota (=Total*0,2)	
-----------------------------------	--	--------------------------	--



TERMO DE ENTREGA DO RELATÓRIO
UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉNICA

TERMO DE ENTREGA DE RELATÓRIO DO ESTÁGIO PROFISSIONAL

Declaro que o estudante: João Júnior Massingue, entregou no dia ___/___/20__ as 3 cópias do relatório do seu Projecto do Curso com a referência: _____ intitulado: **Proposta de um sistema de monitoramento e controle do consumo de energia electrica com base em IoT para o Instituto Medio de Ciências e Gestão.**

Maputo, ____ de _____ de 20__

O(a) Chefe de Secretaria



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉNICA

AVALIAÇÃO DOS SUPERVISORES

Autor: João Júnior Massingue

Proposta de um sistema de monitoramento e controle do consumo de energia eléctrica com base em IoT para o Instituto Medio de Ciências e Gestão

Supervisor da Faculdade

Nota

(Eng. Frederico Zile)

Supervisor da Instituição

Nota

(Leonardo Pascoal)

DEDICATÓRIA

Dedico este trabalho a todos aqueles que me apoiaram e contribuíram na minha educação desde a minha infância, em especial aos meus pais e colegas que tiveram um papel muito importante na minha vida académica.

AGRADECIMENTOS

Em primeiro lugar, quero a Deus pela dádiva da vida, agradecer à minha família, pelo apoio que sempre me foi dado. Principalmente aos meus pais e irmãos, pela confiança que sempre depositaram em mim, pelo carinho em todos os momentos, fáceis e difíceis.

Agradeço ao curso de Engenharia Electrónica da Faculdade de Engenharia da Universidade Eduardo Mondlane, seus docentes e funcionários técnicos-administrativos, que proporcionaram o alicerce académico para alcançar este objectivo.

Agradeço em especial ao meu supervisor da faculdade Eng^o Frederico Zile, pela sua orientação segura, sua dedicação, pelas sugestões e correções que foram de suma importância para a realização deste trabalho.

Agradecer a todos os meus amigos e colegas pelo apoio em todos os momentos, pelos inúmeros conselhos e palavras de incentivo, e por estarem sempre ao meu lado.

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de monitoramento e controle do consumo de energia elétrica baseado em IoT, com foco no Instituto Médio de Ciências e Gestão. O objetivo principal foi criar uma solução prática e eficiente que pudesse reduzir desperdícios energéticos, otimizar custos operacionais e promover a sustentabilidade em um ambiente escolar.

A metodologia adotada incluiu a análise de requisitos técnicos, revisão bibliográfica sobre eficiência energética e tecnologias IoT, desenvolvimento de um protótipo funcional e realização de testes. O sistema proposto utiliza sensores de corrente e tensão, microcontroladores ESP32, relés inteligentes e uma plataforma de gerenciamento remoto via protocolo MQTT. A arquitetura foi projetada para coletar dados de consumo em tempo real, permitindo o desligamento de equipamentos em períodos de inatividade e o controle remoto de dispositivos elétricos.

Os resultados obtidos demonstraram uma redução significativa no consumo energético dos equipamentos monitorados, destacando a viabilidade técnica e econômica do sistema. Além disso, o projeto promoveu a conscientização ambiental e a adoção de práticas mais responsáveis entre os usuários. O sistema desenvolvido é escalável e pode ser adaptado para outras instituições educacionais ou contextos similares, tornando-se uma ferramenta eficiente para alcançar metas de eficiência energética e sustentabilidade.

Palavras Chaves: Eficiência energética, Internet das Coisas (IoT), Monitoramento de energia.

ABSTRACT

This work presents the development of an IoT-based monitoring and control system for electrical energy consumption, focusing on the Instituto Médio de Ciências e Gestão. The main objective was to create a practical and efficient solution that could reduce energy waste, optimize operational costs and promote sustainability in a school environment.

The methodology adopted included the analysis of technical requirements, literature review on energy efficiency and IoT technologies, development of a functional prototype and carrying out tests. The proposed system uses current and voltage sensors, ESP32 microcontrollers, smart relays and a remote management platform via the MQTT protocol. The architecture was designed to collect consumption data in real time, allowing the shutdown of equipment during periods of inactivity and the remote control of electrical devices.

The results obtained demonstrated a significant reduction in the energy consumption of the monitored equipment, highlighting the technical and economic viability of the system. Furthermore, the project promoted environmental awareness and the adoption of more responsible practices among users. The developed system is scalable and can be adapted to other educational institutions or similar contexts, becoming an efficient tool for achieving energy efficiency and sustainability goals.

Keywords: Energy efficiency, Internet of Things (IoT), Energy monitoring.

ÍNDICE

DEDICATÓRIA	IV
AGRADECIMENTOS.....	V
RESUMO	VI
ABSTRACT.....	VII
1. CAPÍTULO 1: INTRODUÇÃO	1
1.1. Introdução.....	1
1.2. Definição do Problema.....	2
1.3. Relevância da pesquisa	2
1.4. Objectivos.....	3
1.4.1. Objectivo geral.....	3
1.4.2. Objectivos específicos	3
1.5. Metodologia.....	3
2. CAPÍTULO 2: REVISÃO TEORICA.....	4
2.1. Potência Eléctrica.....	4
2.2. Medidores de energia e redes eléctricas inteligentes.....	4
2.3. Métodos convencionais de medição de energia eléctrica.....	6
2.4. Sensor de Corrente	6
2.5. Módulo relé	8
2.6. IOT.....	Erro! Marcador não definido.
2.7. Padrões de comunicação sem fio	11
2.7.1. Wireless Fidelity (WiFi).....	11
2.7.2. ZigBee	12
2.7.3. Message Queuing Telemetry Transport - MQTT	12
2.8. Microcontroladores.....	15
2.8.1. EPS8266	16
2.8.2. NodeMCU	17
2.9. Banco de dados MySQL.....	20
2.10. Arduíno.....	21
2.10.1. Ferramenta de desenvolvimento	21
2.10.2. Programação	22
2.11. Servidor web.....	23
3. Capítulo 3: Metodologia.....	25
3.1. Descrição das funcionalidades do sistema	25
3.1.1. Equipamentos do Sistema	25
3.1.2. Sensores de Energia.....	25
3.1.3. Atuadores	25

3.1.4.	Plataforma IoT	25
3.1.5.	Rede de Comunicação	25
3.2.	<i>Funcionalidades do Sistema</i>	25
3.2.1.	Monitoramento em Tempo Real	25
3.2.2.	Controle dos equipamentos	26
3.3.	<i>Benefícios do Sistema</i>	26
3.3.1.	Redução de Custos	26
3.3.2.	Melhoria na Eficiência Energética	26
3.3.3.	Sustentabilidade	26
3.3.4.	Conforto e Produtividade	26
3.4.	<i>Arquitetura do Sistema</i>	26
3.5.	<i>Sensor de Corrente</i>	27
•	<i>Custo-Benefício</i>	30
•	<i>Suporte e Documentação</i>	30
3.6.	<i>Sensor de Tensão</i>	30
3.7.	<i>Microcontrolador</i>	31
3.7.1.	Factores que contribuíram para a escolha do microcontrolador	32
3.8.	<i>Desenho do Circuito para o protótipo</i>	34
3.8.1.	Componentes necessários	34
3.9.	<i>Descrição do circuito</i>	34
3.9.1.	Monitoramento de Corrente	34
3.9.2.	Controle de Dispositivos	35
3.9.3.	Comunicação e Processamento	35
3.9.4.	Monitoramento de Tensão (Opcional)	35
3.10.	<i>Dimensionamento da fonte de alimentação</i>	35
3.11.	<i>Desenho do circuito do protótipo</i>	36
4.	CAPÍTULO 4: TESTE E RESULTADOS	37
4.1.	<i>Testes e Resultados</i>	37
4.1.1.	Montagem do protótipo	38
4.1.2.	Custos do Prototipo	39
4.2.	<i>Interface do usuário</i>	40
4.3.	<i>Resultados</i>	41
5.	CAPÍTULO 5: CONCLUSÕES E RECOMENDAÇÕES	43
5.1.	<i>Conclusão</i>	43
6.	CAPÍTULO 7: BIBLIOGRAFIA	44
6.1.	<i>Bibliografia</i>	44
7.	ANEXO 1: CÓDIGO	45
8.	ANEXO II	63

ÍNDICE DE FIGURAS

Figura 1: Sensor de Corrente ACS712	7
Figura 2: Elementos de relé eletromecânico.....	8
Figura 3: Arquitetura de camadas para IoT	10
Figura 4: Modelo de Publicação e assinatura do protocolo MQTT	13
Figura 5: Componentes de Microcontroladores	15
Figura 6: Diversos modelos de esp8266.....	16
Figura 7: pinagem do NodeMCU esp12-f	18
Figura 8: Funcionamento de um servidor web básico	24
Figura 9: Arquitetura do sistema proposto	27
Figura 10: Módulo do sensor de corrente ACS712.....	28
Figura 11: Circuito Sensor de corrente ACS712.....	28
Figura 12: Circuito para medir a Tensão AC	31
Figura 13: Microcontrolador Esp32.....	32
Figura 14: Esquemas em 2D do protótipo	36
Figura 15: Circuito do protótipo	37
Figura 16: Protótipo.....	39
Figura 17: Interface de Controle com Computador	41
Figura 18: Interface de controle com Smartphone	41
Figura 19: Informação no display de controle.....	41

ÍNDICE DE TABELAS

Tabela 1: Modelos do Sensor ACS712	7
Tabela 2: Especificações do esp8266	17
Tabela 3: Dimensionamento da fonte de alimentação	36
Tabela 4: Principais dispositivos de consumo	38
Tabela 5: Preços e quantidade dos componentes usados	39

LISTA DE ABREVIATURAS

- **IoT** - Internet of Things (Internet das Coisas)
- **IMCG** – Instituto Médio de Ciências e Gestão
- **ESP32** - Microcontrolador da série ESP da Espressif Systems
- **MQTT** - Message Queuing Telemetry Transport (Protocolo de transporte de mensagens)
- **HVAC** - Heating, Ventilation, and Air Conditioning (Aquecimento, Ventilação e Ar Condicionado)
- **Wi-Fi** - Wireless Fidelity (Fidelidade Sem Fio)
- **ADC** - Analog-to-Digital Converter (Conversor Analógico-Digital)
- **PCB** - Printed Circuit Board (Placa de Circuito Impresso)
- **VAR** - Volt-Ampere Reactive (Potência Reativa)
- **LCD** - Liquid Crystal Display (Display de Cristal Líquido)
- **GPIO** - General Purpose Input/Output (Entradas e Saídas de Uso Geral)
- **CPU** - Central Processing Unit (Unidade Central de Processamento)
- **TLS/SSL** - Transport Layer Security/Secure Sockets Layer (Protocolos de Segurança para Comunicação)

- **IDE** - Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
- **C++** - Linguagem de Programação baseada em C com suporte a Orientação a Objetos
- **AC** - Alternating Current (Corrente Alternada)
- **DC** - Direct Current (Corrente Contínua)
- **IHM** - Interface Homem-Máquina
- **UART** - Universal Asynchronous Receiver-Transmitter (Transmissor/Receptor Assíncrono Universal)
- **IC** - Inter-Integrated Circuit (Circuito Interintegrado, protocolo de comunicação)
- **kWh** - Kilowatt Hour (Quilowatt-hora, unidade de energia elétrica)
- **CPE** - Customer Premises Equipment (Equipamento do Cliente, termo comum em redes)
- **QoS** - Quality of Service (Qualidade de Serviço, parâmetro de rede)
- **HTTP** - Hypertext Transfer Protocol (Protocolo de Transferência de Hipertext)

1. CAPÍTULO 1: INTRODUÇÃO

1.1. Introdução

A gestão eficiente de recursos energéticos tem se tornado uma prioridade global, especialmente em instituições educacionais, onde o consumo de energia elétrica representa uma parcela significativa dos custos operacionais. Além de impactar diretamente o orçamento escolar, o desperdício energético compromete os esforços de sustentabilidade e a criação de um ambiente propício ao ensino e aprendizado. Nesse contexto, as tecnologias emergentes, como a Internet das Coisas (IoT), oferecem soluções inovadoras para monitorar e controlar o consumo de energia de forma eficaz e sustentável.

Este projeto se propõe a desenvolver um sistema de monitoramento e controle do consumo energético baseado em IoT para o Instituto Médio de Ciências e Gestão. A solução contempla a instalação de sensores, microcontroladores e sistemas de comunicação que permitem a coleta, análise e gerenciamento remoto de dados em tempo real.

O foco está no controle de dispositivos elétricos, como iluminação e sistemas de climatização, integrando funcionalidades que promovem economia de energia, redução de custos e aumento da eficiência. Além disso, o projeto busca atender às demandas de modernização tecnológica no setor educacional, alinhando-se às metas globais de sustentabilidade e inovação.

A relevância desta pesquisa vai além do impacto financeiro, proporcionando benefícios ambientais e conscientizando a comunidade acadêmica sobre práticas mais sustentáveis no uso de energia elétrica. Ao abordar um problema prático e atual, esta iniciativa pretende servir como modelo para outras instituições, demonstrando o potencial da IoT em promover mudanças significativas em contextos educacionais.

1.2. Definição do Problema

A gestão eficaz e sustentável dos recursos energéticos em ambientes escolares representa um desafio pertinente no contexto atual da educação. Escolas em todo o mundo enfrentam pressões crescentes para reduzir seus custos operacionais e criar ambientes propícios ao aprendizado. No entanto, a falta de ferramentas de monitoramento em tempo real e estratégias de gestão eficientes tem levado a um uso inadequado de energia, resultando em desperdício, maiores despesas.

O problema central reside na ausência de um sistema abrangente que permita às escolas monitorar, analisar e otimizar o consumo de energia de forma eficaz e sustentável. Segundo a Lyceum (2021) a aplicação da Internet das Coisas (IoT) oferece um potencial significativo para abordar essa lacuna, mas a falta de pesquisas específicas sobre sua implementação em ambientes escolares dificulta a compreensão das melhores práticas, dos desafios técnicos e das implicações financeiras.

Portanto, o problema a ser abordado é como aplicar a IoT de maneira eficaz para monitorar e controlar os sistemas de climatização, iluminação e outros dispositivos em escolas, a fim de melhorar a eficiência energética, reduzir custos operacionais e promover a sustentabilidade ambiental, enquanto ainda proporciona um ambiente adequado para o ensino e a aprendizagem.

1.3. Relevância da pesquisa

1.3.1.1. Redução de Custos Operacionais:

A implementação eficaz da IoT para a gestão da eficiência energética nas escolas tem o potencial de reduzir significativamente os custos operacionais, liberando recursos financeiros que podem ser direcionados para a melhoria da educação e do ambiente de aprendizado. Portanto, o desenvolvimento de tecnologia que visam na redução de consumo de energia reduzem a demanda por eletricidade, o que conseqüentemente diminui a necessidade de queima de combustíveis fósseis e contribui para a preservação do meio ambiente.

1.4. Objectivos

1.4.1. Objectivo geral

- Elaborar uma proposta de um sistema de monitoramento e controle do consumo de energia eléctrica com base em IoT para o Instituto Médio de Ciências e Gestão.

1.4.2. Objectivos específicos

- Fazer o levantamento dos requisitos do sistema;
- Identificar as tecnologias IoT disponíveis para monitorar e controlar sistemas de energia, incluindo dispositivos de sensoriamento e plataformas de gerenciamento;
- Analisar os custos associados e os benefícios à implementação do sistema;
- Realizar simulação e testes do protótipo.

1.5. Metodologia

A metodologia para a pesquisa do trabalho vai seguir as seguintes etapas:

- **Revisão da Literatura:**

Realizar uma revisão abrangente da literatura relacionada à eficiência energética em ambientes escolares, incluindo o uso da IoT para esse fim.

Identificar as melhores práticas, tendências e desafios na aplicação da IoT para a gestão de recursos energéticos em escolas.

- **Coleta de Dados Iniciais:**

Coletar dados iniciais sobre o consumo de energia, sistemas de climatização, iluminação e outros dispositivos no Instituto.

- **Design e Implementação da Solução IoT:**

Desenvolver sistemas de monitoramento IoT personalizados, incluindo sensores de consumo de energia.

- **Simulações e testes**

Fazer simulações e testes do protótipo do sistema e avaliar os benefícios do mesmo.

2. CAPÍTULO 2: REVISÃO TEORICA

2.1. Potência Eléctrica

Conceitos de potência eléctrica O conceito de energia é relativamente abstrato e, no caso de dispositivos eléctricos, pode ser estabelecida como sendo o trabalho requerido pelo dispositivo para realizar tarefas, a finalidade básica de um sistema eléctrico é suprir energia aos consumidores a ele conectados (BARTHOLD; REPEEN; HEDMAN; 1983, ORSINI; CONSONNI, 1995). Como a energia absorvida por um equipamento depende do tempo de funcionamento, o conceito de potência possui o intuito incorporar estas duas grandezas. Desta forma, potência é a razão entre a energia fornecida ou absorvida por um determinado equipamento/dispositivo ao longo do tempo (BARTHOLD; REPEEN; HEDMAN; 1983).

Potência eléctrica absorvida por um dispositivo é dada por (1), onde $p(t)$ é a potência eléctrica instantânea, $v(t)$ é a tensão instantânea entre os terminais da carga e $i(t)$ é a corrente instantânea entrando e saindo do dispositivo (SADIKU; ALEXANDER; MUSA, 2014).

$$p(t) = v(t) * i(t) \quad (1)$$

Quando a potência instantânea é avaliada por um período, tem-se o conceito de potência média. Dependendo da carga (resistiva ou reativa) esta potência pode ser absorvida por todo o tempo ou absorver nenhuma potência média (SADIKU; ALEXANDER; MUSA, 2014). Neste sentido, denomina-se potência ativa, medida em W , como sendo o valor médio de $p(t)$ ao longo do tempo, e representa a potência transformada em calor ou em trabalho por um dispositivo eléctrico, ou seja, a parcela que realmente é consumida. Esta componente de potência é proporcional à parcela resistiva do circuito. Já a potência reativa, medida em VAR, está relacionada à energia necessária para manter o campo magnético em cargas indutivas e/ou o campo eléctrico em cargas capacitivas (SADIKU; ALEXANDER; MUSA, 2014, ORSINI; CONSONNI, 1995).

2.2. Medidores de energia e redes eléctricas inteligentes

Com o crescimento de metodologias e sistemas tecnológicos para geração e distribuição de energia elétrica, as concessionárias de energia necessitam de meios para realizar a mensuração dos insumos que são solicitados a ela, uma vez que a energia elétrica é fornecida às unidades consumidoras. Em virtude deste cenário, as empresas administradoras adotaram como padrão a utilização de medidores de energia para aferir os valores de energia usufruído pelos clientes, de forma a realizar o cálculo médio da tarifa a ser instaurada para o consumidor.

Segundo a ABB (2021), medidor de energia é um equipamento que tem por objetivo realizar a medição dos valores de tensão, corrente e potência de uma rede de energia. Através desse equipamento é efetuada a medição de energia elétrica em sistemas alternados (AC).

Mattede (2022) acrescenta que os medidores de energia possuem a importante característica de medir os valores consumidos de energia elétrica pelos clientes e realizar o detalhamento do consumo em kWh para as distribuidoras de energia. Com isso, a possibilidade de analisar o consumo real de energia de cada cliente torna-se uma ação mais palpável, como também funcional e permitindo ao usuário final analisar o quanto foi gasto ao longo do período de uso.

O medidor é um equipamento muito utilizado por diversas classes de consumidores, que por sua vez, analisam os dados das grandezas elétricas medidas e, fazendo uso do seu sistema de controle interno, realiza os cálculos dos valores de potência de toda a unidade consumidora.

Segundo Ferreira (2012), os smart meters, ou medidores inteligentes, são dispositivos medidores com capacidade de processar, armazenar e realizar comunicação com demais equipamentos e sistemas, que podem ir muito além do consumo propriamente dito. Neste caso, eles permitem a troca de informações em tempo real, bidirecionalmente entre cliente final e distribuidor de energia, permitindo o monitoramento das grandezas elétricas, bem como mensurar a qualidade de energia.

As ações de mensuração do consumo de energia elétrica, bem como a modernização dos sistemas de energia, permitindo a sua interligação entre diversos elementos possibilita a conectividade entre dispositivos, facilitando a análise e o controle dos sistemas onde os medidores são aplicados. Com essa modernização e adequação dos sistemas elétricos aos

novos elementos de conectividade, torna-se possível a ampliação de redes inteligentes, ou smart grids.

Rivera (2013), detalha que o conceito de smart grid veio para revolucionar as características da concepção do sistema de controle para energia, trazendo esta área ao novo mundo de conectividade e integração de dispositivos a internet. Isso permite que o consumidor final tenha em mãos um relatório detalhado do consumo de energia elétrica em seu período de uso, bem como propor ações em automatizar seus sistemas de aplicação, programar ações que tragam eficiência, em termos financeiros, bem como a produtividade da operação.

Fracari (2015), afirma que a eficiência gerada por redes inteligentes excede as expectativas de seus objetivos traçados, trazendo relatórios eficientes de consumo e horários, maior direcionamento do fluxo de energia para regiões que necessitam de maior demanda, possibilidade de detectar fraudes em sistemas elétricos, rastreando a unidade consumidora que está praticando alguma ligação ilegal, como também programar ações de manutenção em pontos que são identificados problemas causados por intervenção humana, ações da natureza, paradas programadas e etc.

2.3. Métodos convencionais de medição de energia elétrica

A energia elétrica é uma forma de potência eficaz da diferença de potencial elétrico entre dois pontos, que permite criar uma corrente elétrica (ALEXANDER; SADIKU, 2014). Todos os dispositivos elétricos industriais e domésticos possuem um potencial nominal que indica a potência exigida no equipamento.

Apesar de no Sistema Internacional de Unidades (S.I.) a unidade de energia ser o Joule (J), se tratando de consumo da energia elétrica é utilizado o *kilowatt hour* (KWh) e o seu cálculo pode ser realizado mediante a Equação 2.7:

Substituindo a $E = P\Delta t/1000$ (2.7)

Equação 2.6 na Equação 2.7, tem-se:

$$E = V I \Delta t / 1000 \quad (2.8)$$

Onde, E é a energia elétrica em KWh, V a tensão aplicada no equipamento em Volt (V), I a corrente do equipamento em Ampère (A) e Δt a variação do tempo em horas.

2.4. Sensor de Corrente

Para a medição da corrente elétrica foi escolhido o sensor de corrente invasivo ACS712 (Figura 1). Fabricado pela Allegro MicroSystems, é um sensor preciso e econômico capaz de medir tanto corrente alternada quanto corrente contínua. O *datasheet* do sensor informa a CC suportada, logo para encontrar qual é Corrente Alternada (CA) máxima é necessário dividir este valor pela raiz quadrada de 2 que corresponde a grandeza utilizada para encontrar o valor eficaz de uma onda senoidal (FILIFELOP, 2019).

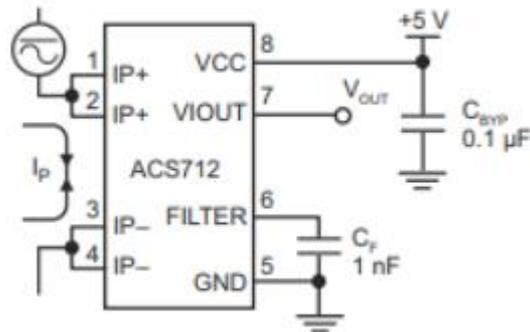


Figura 1: Sensor de Corrente ACS712

Fonte: Microsystems, 2007

Existem 3 modelos deste sensor com limites de medição de corrente e sensibilidades distintas, como demonstrado na Tabela 1

Tabela 1: Modelos do Sensor ACS712

Modelo	Corrente DC (A)	Corrente AC (A)	Sensibilidade (mV/A)
ACS712ELCTR-05B-T	-5 a 5	-3,54 a + 3,54	185
ACS712ELCTR-20A-T	-20 a 20	-14,14 a +14,14	100
ACS712ELCTR-30A-T	-30 a 30	-21,21 a +21,21	66

O funcionamento do sensor se baseia no efeito Hall que é quando a corrente que flui por um caminho de cobre (pinos 1 e 2 aos pinos 3 e 4) gera um campo magnético e este induz uma tensão proporcional a este campo (pino 7) (MICROSYSTEMS, 2007).

Os terminais do condutor de cobre apresentam um isolamento elétrico dos condutores do sensor (pinos 5 a 8), permitindo que possa ser empregado em circuitos sem o uso de um isolamento externo (MICROSYSTEMS, 2007).

2.5. Módulo relé

O relé é um componente elétrico que através da aplicação de uma corrente gera uma indução magnética realizando a movimentação de um contato, servindo na maior parte das vezes como um interruptor a distância, podendo realizar a integração de circuitos de alta potência com circuitos eletrônicos de correntes e tensão inúmeras vezes menores. (MORIMOTO, 2016).

O relé é um componente eletromecânico, ou seja, ele conta com uma parte mecânica de contato e o acionamento ocorre através da corrente elétrica em uma bobina. Na Figura 2 é possível visualizar todos os componentes de um relé eletromecânico, que são:

- Bobina
- Induzido
- Núcleo
- Contatos da bobina
- Contatos do relé

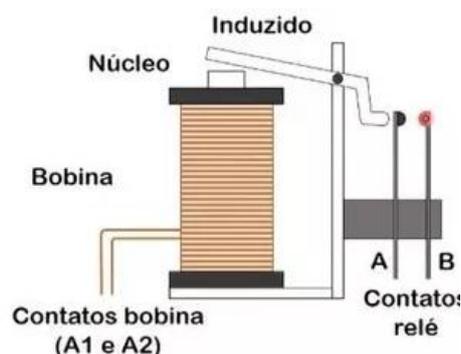


Figura 2: Elementos de relé eletromecânico

Fonte: Marimoto (2016)

Espécie de eletroímã. No momento em que você acionar a bobina, o núcleo irá atrair o induzido que ao mesmo tempo que se desloca em direção a bobina, também irá empurrar o contato A do relé na direção do contato B, fechando este contato.

No presente projeto o relé foi usado para controlar as lâmpadas presente no protótipo o que nos possibilita também o controle através de Microcontroladores.

A Internet das Coisas, ou IoT, refere-se aos milhares de milhões de dispositivos físicos em todo o mundo que estão agora ligados à Internet, todos eles colecionando e compartilhando dados. Através da utilização de chips de computador mais baratos e da ubiquidade das redes sem fios é possível transformar qualquer coisa, desde algo tão pequeno como um comprimido, até algo tão grande como um avião, numa parte da Internet das Coisas. Ligar todos estes diferentes objetos e adicionar-lhes sensores acrescenta um nível de inteligência digital a dispositivos que de outra forma seriam “ignorantes”, permitindo-lhes comunicar dados em tempo real sem envolver um ser humano. A Internet das Coisas está a tornar o tecido do mundo à nossa volta mais inteligente e mais reativo, fundindo os universos digital e físico (Ranger, 2020).

O conceito da internet das coisas, derivado do termo em inglês IoT, teve origem em uma apresentação do pesquisador Kevin Ashton em 1999. De acordo com o cientista, se os computadores pudessem coletar dados sem a ajuda de seres humanos, seria possível rastrear e coletar informações de todas as formas. Saberíamos quando algo precisa ser trocado ou consertado e assim, reduziríamos custos e desperdícios, acreditando que a *Internet of Things* tem o potencial de transformar o mundo semelhantemente da internet (ASHTON, 2009).

Atualmente, é possível observar objetos conectados a rede interagindo entre si, com uma central ou com pessoas. Esses dispositivos são capazes de coletar informações que podem ser disponibilizadas de forma gráfica para outros indivíduos analisarem. Além disso, alguns desses objetos podem ser controlados de forma remota, ou ainda realizar ações automaticamente tendo como base os dados coletados.

Atualmente o *Institute of Electrical and Electronics Engineers* (IEEE) está considerando a arquitetura, demonstrada na Figura 4, como padrão para dispositivos IoT

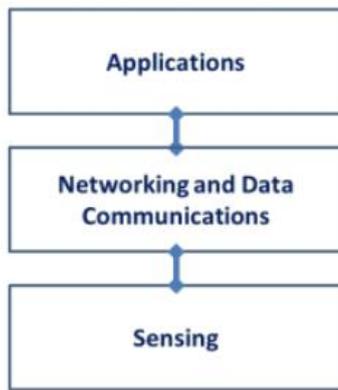


Figura 3: Arquitetura de camadas para IoT

Fonte: Chebudie; Minerva; Rotondi, 2014

Oracle (2022) destaca que a Internet das Coisas (IoT) objetiva-se na integração em rede de objetos físicos incorporados a sensores, softwares e outras tecnologias que visam conectar e trocar dados com quaisquer dispositivos e sistemas pela internet, sendo que estes podem variar desde objetos domiciliares básicos, até mesmo a equipamentos industriais complexos.

Além disso, a Internet das Coisas também pode incluir pessoas, animais e demais objetos com sensores incorporados, não se submetendo somente conectividade de objetos simples e sem independência, concedendo a possibilidade de conectividade de entre todos os envolvidos nos mais variados setores de utilização, seja eles urbanos, rurais, industriais e etc.

Além da possibilidade de proporcionar interconectividade entre pessoas, dispositivos e variadas redes de sistemas, a IoT disponibiliza melhor gestão de tempo para ações de comando em sistemas automatizados de controle de processos e análise de dados, bem como a eficiência de recursos e tomada de decisões mais precisa, utilizando maior quantidade de dados e concedendo melhor tempo de resposta.

Amazon (2022) aponta alguns de muitos benefícios na aplicação dos recursos de Internet das Coisas para os ambientes de aplicação industrial, comercial, energia e demais ramos de negócios, sendo estes:

- Aumento da segurança de informação, possibilitando o monitoramento do tráfego de dados e montagem de arquitetura para proteção das informações;
- Desenvolvimento de recursos para programação de sistemas autônomos de controle de processos industriais, comerciais;

- Elaboração de programas de Inteligência Artificial e Aprendizado de Máquina para desenvolvimento de atividades diárias em processos industriais, agropecuários, hidroenergéticos e etc;
- Proporcionar a inovação de sistemas de automação de atividades e tarefas que resultem em maior eficiência e resultados satisfatórios.

Oracle (2022) aponta que a Internet das Coisas é uma das tecnologias mais importantes do presente século, uma vez que é possível conectar objetos do cotidiano, como eletrodomésticos, eletroeletrônicos, automóveis, bem como a integração de edifícios e outros estabelecimentos comerciais ou residenciais à internet, usando equipamentos tragam esta conectividade, possibilitando o monitoramento de dados e a gestão dos sistemas de controle para proporcionar maior eficiência dos sistemas.

A padronização da definição das tecnologias baseadas em IoT permite o seu avanço, viabilizando troca de informações entre vários dispositivos conectados à internet, além da interação com diversos usuários. Diante disso, umas gamas de possibilidades surgem, como por exemplo: cidades inteligentes, saúde e casas inteligentes. Entretanto, novos desafios também surgem como: regulamentações, padronizações e segurança (SANTOS et al., 2016).

2.6. Padrões de comunicação sem fio

Atualmente, a comunicação sem fio está altamente presente no ambiente, visto que a população possui diversos dispositivos conectados entre si sem a necessidade de cabos. Existem diferentes padrões de comunicação sem fio, tais como *Bluetooth*, WiFi e ZigBee. O padrão WiFi é uma das mais utilizadas para aplicações envolvendo internet das coisas

2.6.1. Wireless Fidelity (WiFi)

A conexão WiFi é um protocolo de comunicação baseado no padrão IEEE 802.11 que possui uma diversidade de capacidade e cobertura de dispositivos além de ter um custo baixo. Isto possibilitou a sua vasta utilização em diversos cenários que engloba redes 2G/3G e a rede residencial (RUBINSTEIN; REZENDE, 2002).

A Wi-fi *Alliance*® é uma organização mundial que valoriza e torna a tecnologia WiFi a mais utilizada do mundo. É responsável por definir tecnologias e programas WiFi inovadores defendendo globalmente regras de uso justo. Hoje em dia encontram-se mais dispositivos

WiFi em uso do que a população, além de que mais da metade do tráfego da Internet é realizado por redes WiFi (ALLIANCE, 2018).

Este padrão de comunicação sem fio é flexível, suportar diversos dispositivos conectados, possui segurança e controle de acesso. Assim, o WiFi foi escolhido para este trabalho por apresentar essas características citadas, além de estar presente na maioria das residências. Além disso, o microcontrolador ESP32 possui conectividade WiFi.

2.6.2. ZigBee

O ZigBee foi desenvolvido para disponibilizar uma rede de baixo consumo, wireless, de monitoramento e controle baseado no padrão IEEE 802.15.4. Tem como principais características o baixo consumo de energia, baixo custo e vazão. Opera na frequência 2.4GHz, porém é capaz de operar em outras duas, 868MHz e 915Mhz (SANTOS et al., 2016).

Esta tecnologia é projetada para uso em aplicações embarcadas que requerem baixa taxa de dados e consumo de energia, tais como plantas industriais, automação predial e a área de saúde.

2.6.3. Message Queuing Telemetry Transport - MQTT

Dispositivos IoT precisam trabalhar entre si e com banco de dados, sendo então a conexão a Internet um requisito. Pensando nisso, no final da década de 90 foi desenvolvido pela IBM um protocolo de troca de mensagens entre partes de forma assíncrona, ou seja, comunicação não simultânea (YUAN, 2017).

Esse protocolo de mensagens é leve e flexível projetado para dispositivos que utilizam largura de banda baixa, com alta latência e com requisitos de hardware simples. Esses fundamentos o tornam ideal para dispositivos IoT ou Machine-to-Machine (M2M) conseguindo garantir entrega e confiabilidade (MQTT.ORG, 2019).

2.6.3.1. Funcionamento

O MQTT implementa um modelo de publicação e assinatura entre um agente central e infinitos clientes, sendo o primeiro um servidor responsável por receber todas as mensagens dos clientes e, posteriormente destiná-las a clientes finais. Basicamente, funciona da seguinte forma: Clientes conectam-se ao *broker* e podem publicar ou receber atualizações

de um tópico. O cliente publica informações em um tópico e este as envia para o *broker*. Seguidamente, o agente central encaminha as mensagens do tópico para todos os clientes que o assinam (YUAN, 2017). A Figura 6 ilustra um modelo de publicação e assinatura do protocolo MQTT.

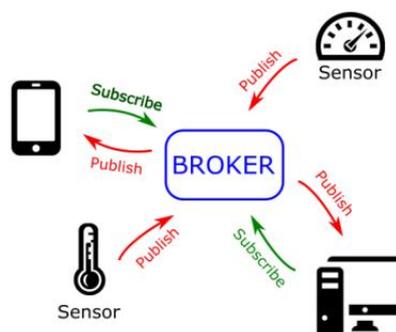


Figura 4: Modelo de Publicação e assinatura do protocolo MQTT

Fonte: Yuan 2017

De acordo com a IBM (2018), MQTT (Message Queuing Telemetry Transport) trata-se de um protocolo de comunicação, baseado no padrão TCP/IP, utilizado para trocar mensagens entre pequenos dispositivos em uma pequena largura de banda ou conexões elevadas, de forma a enviar mensagens da maneira mais confiável.

Neri (2019) ressalta que um sistema MQTT se fundamenta na comunicação entre cliente e servidor, sendo que o primeiro elemento pode realizar inserção de mensagens de dados, bem como a captação das informações cedidas por outro cliente e o segundo elemento administra os dados recebidos e os que serão enviados.

A origem deste protocolo é oriunda da necessidade promover um protocolo simples e que possua a capacidade de comunicar diversos equipamentos entre si, sendo tal capacidade de comunicação provisionada por microcontroladores para obtenção de dados em uma frequência de transmissão leve entre máquinas e sensores, algo que não é visto com tal eficiência no protocolo HTTP.

Santos (2022) destaca que no protocolo MQTT, a publicação e o recebimento de dados são efetuados mediante a um servidor chamado de Broker, como também será acompanhado de um cliente que possuirá o papel de Publisher (Publicador), que fará a transmissão da mensagem proposta, tendo um tópico de destino e junto do conteúdo da mensagem, classificado como Payload. Tal mensagem será gerida pelo Broker, que encaminhará ao Subscriber aquilo que foi previamente detalhado no tópico da mensagem.

Tendo tal linha de comunicação entre os elementos, é possível que o cliente se torne um Subscriber, desde que apenas encaminhe uma solicitação ao Broker que será capaz de estabelecer uma ligação entre cliente e o tópico propriamente dito. A Figura 1.6 mostra um exemplo de arquitetura entre dispositivos, utilizando o protocolo MQTT.

De maneira simplificada, tal comunicação pode ser dividida nas seguintes nomenclaturas:

- Subscriber (Subscrito): Personagem inscrito no tópico de mensagens, responsável por ser o receptor de dados;
- Publisher (Publicador): Trata-se do emissor que fará o envio de dados para um tópico pré-determinado;
- Broker: Gerenciador de informações entre Publisher e Subscriber para o envio de mensagens entre as partes;
- Client (Cliente): Elemento que pode enviar e receber dados oriundos do Broker;
- Mensagem (Message): Refere-se aos dados que são trocados entre os clientes presentes e o Broker existente;
- Payload: Conteúdo presente na mensagem enviada.

Neri (2019) aponta que no processo de envio de informações entre um cliente e um Broker, existem procedimentos de verificação de mensagens de ponta a ponta classificados como Qualidade de Serviço (QoS), que tem como objetivo realizar a interceptação das mensagens enviadas entre um cliente MQTT e um broker, de forma a verificar se o payload enviado chegou ao receptor, apresentando uma mensagem de confirmação. As Qualidades de serviço do protocolo MQTT são classificadas de três maneiras, sendo elas:

- QoS 0 (No máximo uma vez): A mensagem é enviada somente uma vez e não há passos subsequentes, tampouco existe um retorno para confirmar se chegou ao destinatário;
- QoS 1 (Ao menos uma vez): A mensagem é enviada ao Broker ao menos uma vez, de maneira que o cliente aguarda a confirmação da chegada da mensagem ao seu receptor;
- QoS 2 (Precisamente uma vez): A mensagem é entregue precisamente uma vez, tendo a necessidade que a mensagem enviada seja armazenada no emissor e no receptor de destino até que seja processado tal dado.

MQTT.ORG (2022) ressalta grandes vantagens na utilização do protocolo MQTT, sendo elas:

- Comunicação bidirecional entre equipamentos em rede;
- Criptografia de dados e certificações de autenticação para aumentar a segurança da informação;
- Apresenta suporte para trabalhar em redes instáveis e inseguras;
- Conectividade com muitos dispositivos voltados para Internet das Coisas

2.7. Microcontroladores

Segundo (Carvalho & Penido, 2013) Microcontroladores São circuitos integrados que possuem em seu interior todos os componentes necessário ao seu funcionamento dependendo unicamente da fonte de alimentação externa. Pode-se dizer que os microcontroladores são computadores de um único chip. A Figura 5 Ilustra os componentes de um microcontrolador típico.

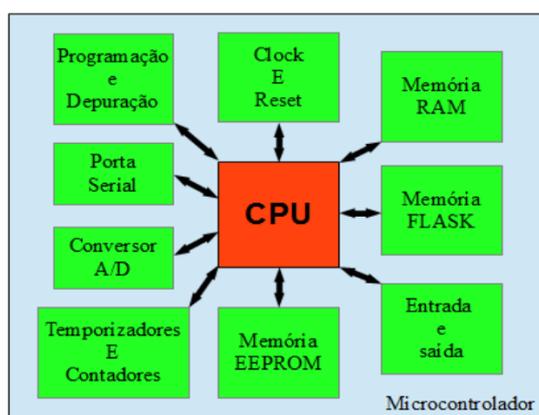


Figura 5: Componentes de Microcotroladores

Fonte: Carvalho & Penido, 2013

Um sistema microprocessado é composto por uma unidade central de processamento CPU e um conjunto de periféricos necessários para o seu funcionamento. Dentre estes periféricos podemos destacar a memória de dados, a memória de programa e o circuito de clock. Os microcontroladores diferem dos sistemas tradicionais por já integrarem os seus periféricos dentro do próprio componente.

Esta integração é uma das principais vantagens dos microcontroladores, pois contendo todos os periféricos no mesmo componente faz com que sua utilização seja mais fácil e mais barata. Sistemas microcontrolados não necessitam de muitos componentes, o que torna mais simples a construção das placas de circuito e diminui o custo dos componentes e da produção.

O tipo da embalagem, a velocidade de processamento, a quantidade de memória e os tipos de periféricos variam de modelo para modelo e também entre fabricantes, pois cada microcontrolador é desenvolvido para um tipo de operação.

Os microcontroladores são muito utilizados pela sua versatilidade, pois seu comportamento depende principalmente do software que nele é gravado. Assim um mesmo microcontrolador pode ser utilizado para uma infinidade de aplicações bastando apenas mudar o seu software.

Outra vantagem é a possibilidade de atualização de um produto através da atualização do software do microcontrolador, o que não é possível com circuitos analógicos ou digitais tradicionais.

Na escolha do microcontrolador é importante observar se os recursos que ele oferece suportam o objetivo do projeto, por exemplo, se for necessário usar a internet é necessário que o microcontrolador seja dotado de wi-fi ou que seja possível integrar um módulo wi-fi.

2.7.1. EPS8266

EPS8266 é o nome de um sistema embarcado projetado pela Espressif Systems. O ESP8266 se define como uma solução de redes Wi-fi autossuficiente se oferecendo como uma ponte entre um microcontrolador pré-existente e a rede com sinal Wi-fi, e que também é capaz de executar aplicações de maneira independente. (KOLBAN, 2015).

Existem vários tipos de modelos de placas disponíveis, porém, é importante notar que existe apenas um processador ESP8266 e ele é encontrado em todas as suas variações. O que diferencia uma placa de outra é o número de pinos GPIO expostos, a quantidade de memória flash disponível, o estilo de pinos conectores e várias outras considerações relacionadas a sua construção. De um ponto de vista de programação, eles são todos iguais. (KOLBAN, 2015)

Os diversos tipos e modelos do microcontrolador Esp8266 podem ser observados na figura 5.



Figura 6: Diversos modelos de esp8266

A tabela 3 exibe as especificações do ESP8266 fornecidas pelo livro Kolban's Book on ESP8266.

Tabela 2: Especificações do esp8266

Tensão	3.3V.
Consumo de corrente	10uA – 170mA.
Memória Flash	16MB max (512K normal).
Processador	Tensilica L106 32 bits.
Velocidade do Processador	80-160MHz.
RAM	32K + 80K.
GPIOs	17 (multiplexada com outras funções).
Conversor A/D	1 entrada com 1024 passos de resolução.
Suporte 802.11	b/g/n/d/e/i/k/r.
Máximo de conexões TCP concorrente	5

2.7.2. NodeMCU

Módulo NodeMCU é uma placa que foi criada para facilitar o desenvolvimento de aplicações para o módulo ESP8266 ESP-12.

Como é possível ver na foto, existe um módulo ESP-12 soldado na placa. Nessa placa já existem todos os circuitos necessários para fazer o ESP-12 funcionar interface Serial-USB, regulador de tensão, leds indicadores, botões de controle (Reset e Flash) e barramentos de pinos para permitir o uso em Protoboards.

A grande vantagem dessa placa NodeMCU é que ela funciona como se fosse um Arduino. Você conecta o cabo USB, e com a IDE Arduino você carrega seus programas na placa. Nem precisa pressionar os botões!

Para a alimentação do ESP-12, existe um regulador de 3,3V AMS1117 (corrente max 1A).

Considerando que o consumo máximo de um ESP-12 é de aproximadamente 200 mA, sobra uma corrente disponível de 800 mA. Mas se for usar a tensão de 3,3V do próprio regulador para alimentar um outro dispositivo externo, recomendo que nunca ultrapasse os 500 mA.

A alimentação dessa placa pode ser feita através do próprio conector USB (5,0V) ou então através do pino VIN (EXT), com uma alimentação regulada de 5,0V .

Apesar do Regulador AMS1117 aceitar tensões de até 9V na entrada, recomendo que se for alimentar a placa através desse pino, use sempre uma fonte regulada de 5V, pois assim nunca sobreaquecerá o regulador! Evitando um possível defeito no mesmo.

Não recomendo que use o pino 3,3V REG para a entrada de alimentação da placa. Esse pino é a saída do regulador e não a entrada. Mas no site da ESPRESSIF, tem a informação de que a alimentação possa ser também através do pino 3,3V REG.

- **Uma observação importante:** as opções de alimentação são mutuamente exclusivas, isto é, somente poderá usar uma opção (USB, 5V ou 3,3V) . Não se usa mais de uma opção, pois poderá danificar algum componente da placa. Não se esqueça de conectar o GND da fonte no GND da placa.

O Chip da interface Serial-USB é o **CP2102** da Silicon Labs. Ele suporta USB 2.0. (Não suporta USB 3.0). Baud Rates até 1 Mbps. Na placa NodeMCU, a alimentação 3,3V do CP2102 é feita através do regulador AMS1117.

O consumo máximo de corrente desse chip é de apenas 26 mA. O CP2102 possui um regulador interno de 3,3V, mas nessa placa NodeMCU, esse regulador não é usado.

Na figura 6 está apresentada a pinagem do NodeMCU esp12-f.

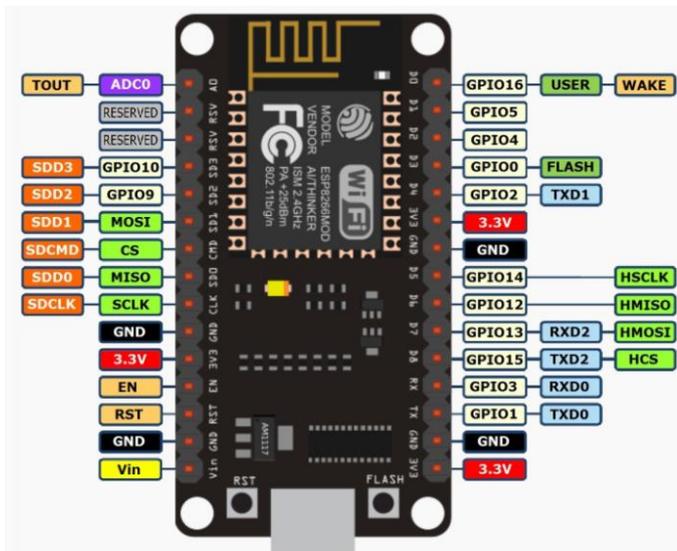


Figura 7: pinagem do NodeMCU esp12-f

Funções dos pinos

- **VIN** – Esse é o pino de alimentação externa (recomendo 5,0V / 1A). Pode usar até 9V, mas o regulador da placa deverá esquentar. Não use-o se estiver usando a USB.
- **GND**– Esse é o terra da placa. Não se esqueça de conectá-lo ao terra de outros dispositivos.
- **RST** – Reset do módulo ESP-12. Nível LOW(0V) dá um reboot na placa.
- **EN** – (Enable) ativa o módulo ESP-12 quando o nível for HIGH(3,3V).

- **3.3V** – saída do regulador interno 3,3V – Para alimentar outro dispositivo, não use mais do que 500 mA de corrente.
- **CLK** – interface SPI (clock) – pino SCLK (GPIO_6)
- **SD0** – interface SPI (master in serial out) – pino MISO (GPIO_7)
- **CMD** – interface SPI (chip select) – pino CS (GPIO_11)
- **SD1** – interface SPI (master out serial in) – pino MOSI (GPIO_8)
- **SD2** – pino GPIO_9 pode ser usado também para comunicação com SD Card (SDD2)
- **SD3** – pino GPIO_10 – pode ser usado também para comunicação com SD Card (SDD3)
- **RSV** – reservado (não use).
- **ADC0**– pino de entrada do conversor analógico digital ADC de 10 bits. Tensão máxima de 1,1V (variação do valor digital – 0 a 1024).
- **D0** – pino GPIO_16 pode ser usado para acordar (WAKE UP) o ESP8266 em modo sono profundo (Deep sleep mode).
- **D1** – pino GPIO_5 – entrada ou saída.
- **D2** – pino GPIO_4 – entrada ou saída.
- **D3** – pino GPIO_0 é usado também para controlar o upload do programa na memória Flash. Está conectado no botão FLASH.
- **D4** – pino GPIO_2 – UART_TXD1 quando carregando o programa na memória FLASH
- **D5** – pino GPIO_14 pode ser usado em SPI de alta velocidade (HSPI-SCLK)
- **D6** – pino GPIO_12 pode ser usado em SPI de alta velocidade (HSPI-MISO)
- **D7** – pino GPIO_13 pode ser usado em SPI de alta velocidade (HSPI-MOSI) ou UART0_CTS.
- **D8** – pino GPIO_15 pode ser usado em SPI de alta velocidade (HSPI-CS) ou UART0_RTS.
- **RX** – pino GPIO_3 – U0RXD quando carregando o programa na memória FLASH.

- **TX** – pino GIPO_1 – U0TXD quando carregando o programa na memória FLASH.

Conforme Kalita, Gogoi, Berzboruah (2010), a maior parte das organizações usam computadores, que é algo essencial para implementar a tecnologia *web* em suas atividades. Os serviços baseados na *Web* são muito importantes em vários tipos de transmissão e recepção de informações. Grande parte das instituições possuem suas próprias aplicações *web*, que facilitam o fornecimento de informações relacionadas ao usuário e a instituição no mundo inteiro, tornando-se algo benéfico para ambos.

O avanço da tecnologia é algo constante nos últimos tempos, com isso a Internet evoluiu muito, tornando as páginas *web* cada vez mais atraentes e cheias de recursos. As páginas *web* referem-se a todo site ou aplicação que fiquem hospedados em servidores na internet, onde o usuário não precisa nada mais do que um navegador (*browser*) para poder utilizar, sem precisar instalar nenhum software adicional (FARIA, 2013).

2.8. Banco de dados MySQL

Para o correto funcionamento de um sistema, independente do porte dele, é necessário um gerenciamento de dados confiável, que, de modo geral, será utilizado para armazenar os dados em um banco de dados, e posteriormente, recuperá-los por meio de linguagem de programação. O uso de banco de dados relacional como repositório de dados é algo muito comum entre os desenvolvedores de aplicação de hoje em dia, a tecnologia encontrada em banco de dados relacional está muito bem fundamentada, facilitando a flexibilidade que ela nos fornece (LUCKOW E MELO, 2010).

Além da necessidade de um banco de dados, é necessário o uso de uma conexão com esse repositório, para realizar as operações desejadas. Uma das maneiras de realizar essa conexão é com a utilização de um *driver Java Database Connectivity* (JDBC). Este *drive* é fornecido pelas principais empresas desenvolvedoras de banco de dados em extensão Java Application Resource (JAR). Na Figura 24 pode-se perceber a arquitetura utilizada no JDBC, em comunicação com a aplicação (LUCKOW E MELO, 2010).

A utilização do JDBC utiliza a classe *DriveManager* e o método *getConnection* para fazer a operação. É informada uma *String* com a URL com informações para a conexão com o banco, onde cada banco de dados possui sua URL padrão para realizar a conexão de

maneira correta, por isso é de extrema importância obter o driver direto com o fabricante do banco de dados (LUCKOW E MELO, 2010).

Segundo Col e Nesello (2013), o MySQL é um sistema com a função de gerenciar um banco de dados, que utiliza a linguagem Structured Query Language (SQL) como interface de comunicação. O MySQL é uma tecnologia *open source*, para diversos tipos de aplicações. A MySQL pertence ao ORACLE, e tornou-se conhecida pelos desenvolvedores pela sua facilidade de integração, portabilidade, compatibilidade, pouca exigência de hardware, desempenho e facilidade de uso.

2.9. Arduíno

O Arduíno é uma plataforma hardware, que tem seu projeto criado com base no microcontrolador Atmel AVR. Ele possui linguagem de programação própria, mas muito parecida com C/C++, facilitando no desenvolvimento de projetos eletrônicos. O desenvolvimento do Arduino teve o objetivo de aprimorar os projetos que era preciso comunicação direta entre o software e o hardware, utilizando meios de controlar sensores e atuadores, para assim facilitar a percepção e interação do projeto com o ambiente real. Na placa do Arduino pode-se perceber a semelhança com um computador de pequeno porte, devido suas características, como o microcontrolador, memória RAM, memória *flash*, *clock*, entre outros (OLIVEIRA E ZANETTI, 2016).

O Arduino UNO é um dos principais modelos de Arduino, nele pode-se perceber várias características importantes, sendo elas: 14 pinos de saída ou entradas digitais, possui um microprocessador ATmega328, memória RAM de 2 KB, memória flash de 32 KB e *clock* de 16 MHz. Na Figura 25 pode-se observar o formato do Arduino UNO e algumas especificações dos pinos e configurações (MONK, 2009).

2.9.1. Ferramenta de desenvolvimento

A plataforma Arduino como qualquer outra plataforma programável, utiliza uma linguagem de programação, um compilador para verificar o código descrito e uma maneira de executar a transmissão do código para o dispositivo. Para facilitar este processo é preciso

utilizar um *Integrated Development Environment* (IDE) que é um ambiente integrado de programação. Quando é criado um programa para Arduino, ele é denominado sketch, neste

ambiente de trabalho pode-se encontrar inúmeras ferramentas facilitadoras, como a transferência do código desenvolvido para o hardware, pode-se selecionar o modelo de Arduino que precisasse desenvolver, entre outras funcionalidades. Na Figura 26 pode-se visualizar a IDE Arduino no seu ambiente de desenvolvimento, sendo o primeiro passo para o desenvolvimento de um projeto (OLIVEIRA E ZANETTI, 2016).

A instalação do IDE pode ser feita de maneira simples, pode-se encontrar várias versões para plataformas diferentes, possui compatibilidades com os principais sistemas operacionais e seu download pode ser feito de maneira gratuita no site oficial do Arduino (MONK, 2009).

Toda ferramenta de desenvolvimento precisa seguir os seus próprios padrões, sendo de grande importância o conhecimento das variáveis suportadas pelo IDE de desenvolvimento. As principais variáveis do IDE Arduino estão representadas na Tabela 3, ajudando a facilitar o desenvolvimento da aplicação (OLIVEIRA E ZANETTI, 2016).

2.9.2. Programação

Segundo (PEREIRA, 2015) a programação é tipificada por ser uma proposta que expresse um determinado ofício através de uma lógica, sendo dividida em várias ascendências sequenciais que possam ser aplicados a um equipamento ou máquina.

Para a transformação de um problema em programação é necessário esmiuçar as causas, para então buscar as resolver de maneira concisa, organizando as causas e soluções de forma proveitosa, por fim traduzir todas as chaves encontradas para uma linguagem que seja compreendida pela máquina aonde se vai atuar.

A programação é fragmentada em diversos tipos de linguagens que possuem suas peculiaridades, bem como vantagens e dispêndios, a precursor da programação foi o Assembly, atualmente existem variadas línguas programáticas, como por exemplo: Javascript, Java, Python, C/C++ entre outras, através disso SEBESTA (2011) destaca que é importante buscar a melhor forma para expressar de maneira concisa e bem concebida as ideias de resolução, sendo necessário para isso o amplo conhecimento dos ramos da programação, o que fará com que se poupe tempo, memória e processamento dos controladores.

A linguagem de programação C++ é o resultado da implementação da linguagem C, e vem se tornando uma das mais propagadas no meio dos microcontroladores, devido sua

possibilidade de trabalhar com altos e baixos níveis e principalmente por apresentar um bom grau didático, bem como a Javascript que representa uma evolução baseada na sua estrutura, que tornou o desenvolvimento Web tangível através do modo inovador de apresentar conteúdos interativos e animados.

2.9.2.1. Linguagem de programação C++

A linguagem de programação C++ teve sua fundamentação na linguagem C, que por sua vez é fruto da transformação das anteriores principalmente da Basic Combined Programming Language (BCPL), que foi concebida na década de 80 através de estudos e aperfeiçoamentos empregados por Stroustrup, é baseada na programação orientada a objetos, que possibilita através do recursos, conhecidos como objetos a representação e modelação do mundo real para a programação, facilitando a criação, aprimoramento e correção das programações.

Segundo (DEITEL, 2006) os programas em linguagem C++ além da linguagem propriamente dita, ainda possuem duas estruturas essenciais que os tornam inerentes, conhecidas como classes e funções, que estão comportadas nas bibliotecas, esses utensílios quando usados como fundamentação reduzem o tempo necessário para aprendizagem que se levaria em outras linguagens.

2.9.2.2. Javascript

Javascript é a linguagem de programação que tem como principal função prover a simplificação de processos, possibilitando a criação de programas embarcados no próprio código das páginas HTML, viabilizando maiores níveis de interação das páginas com o usuário, através do processamento e verificação de dados no computador do cliente, sem a necessidade de acesso constante ao servidor (FLANAGAN, 2011).

Devido á relevância desta linguagem para a criação de páginas web, sua evolução acompanhou o desenvolvimento tecnológico, sendo muito utilizada atualmente, podendo ser observada principalmente na criação de aplicativos para dispositivos móveis, assegurando programas mais dinâmicos e interativos através do controle das animações por exemplo.

2.10. Servidor web

Servidor web (*web server*)" pode referir ao hardware ou ao software, ou ambos trabalhando juntos. (mozilla.org).

1. Referente ao hardware, um servidor web é um computador que armazena arquivos que compõem os sites (por exemplo, documentos HTML, imagens, folhas de estilo, e arquivos JavaScript) e os entrega para o dispositivo do usuário final. Está conectado a Internet e pode ser acessado através do seu nome de domínio (DNS), como por exemplo mozilla.org.
2. Referente ao software, um servidor web inclui diversos componentes que controlam como os usuários acessam os arquivos hospedados (armazenados para disponibilização), no mínimo um *servidor HTTP*. Um servidor HTTP é um software que compreende URLs (endereços web) e HTTP (o protocolo que seu navegador utiliza para visualizar páginas web).

Em um nível mais básico, o navegador fará uma requisição utilizando o protocolo HTTP sempre que necessitar de um arquivo hospedado em um servidor web. Quando a requisição alcançar o servidor web correto (hardware), o *servidor HTTP* (software) enviará o documento requerido, também via HTTP. A figura 3 mostra esse processo.

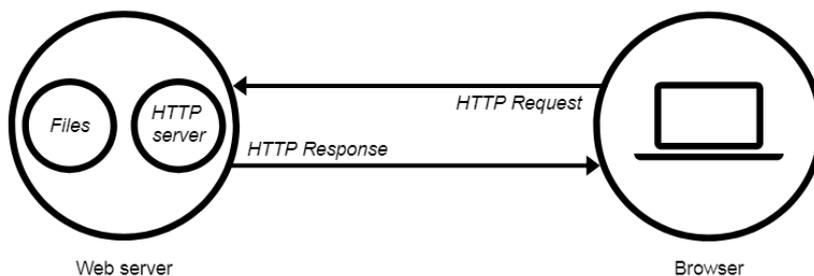


Figura 8: Funcionamento de um servidor web básico

Fonte: Mozilla.org

3. Capítulo 3: Metodologia

3.1. Descrição das funcionalidades do sistema

O sistema proposto tem o objectivo de monitorar e controlar o consumo de energia, assim sendo, deve responder os requisitos descritos abaixo.

3.1.1. Equipamentos do Sistema

3.1.2. Sensores de Energia

Sensores de Corrente e Tensão: Utilizados em salas de aula, laboratórios, bibliotecas e outros espaços para medir o consumo específico de equipamentos.

3.1.3. Atuadores

Relés Inteligentes: Controlam o desligamento e ligamento de equipamentos com base em horários programados ou presença de pessoas.

3.1.4. Plataforma IoT

Gateways: Coletam dados dos sensores e os enviam para a nuvem ou um servidor local.

Servidores de Dados: Armazenam e processam dados recebidos dos sensores e atuadores.

Software de Gerenciamento de Energia: Interface web e móvel para visualização, análise e controle do consumo de energia.

3.1.5. Rede de Comunicação

Wi-Fi/Ethernet: Usado para conectar dispositivos IoT à rede local e à internet.

Protocolos de Comunicação: MQTT, Zigbee, Z-Wave, entre outros, garantem comunicação eficiente entre dispositivos.

3.2. Funcionalidades do Sistema

O Sistema proposto deve ser capaz de responder aos seguintes requisitos abaixo mencionados:

3.2.1. Monitoramento em Tempo Real

Visualização do consumo de energia em tempo real para diferentes setores do instituto (salas de aula, laboratórios, etc.).

3.2.2. Controle dos equipamentos

Desligamento automático de equipamentos fora do horário de uso, quando o ambiente está desocupado ou quando o equipamento apresenta

3.3. Benefícios do Sistema

O sistema deve ser capaz de trazer os benefícios que se seguem:

3.3.1. Redução de Custos

Identificação de desperdícios de energia e implementação de medidas corretivas. Redução do consumo de energia através de controles automáticos e eficientes.

3.3.2. Melhoria na Eficiência Energética

Otimização do uso de HVAC (Heating, Ventilation, and Air Conditioning) iluminação com base em dados reais. Melhoria na gestão de equipamentos de alto consumo energético.

3.3.3. Sustentabilidade

Promoção de práticas sustentáveis e conscientização sobre o uso eficiente de energia.

3.3.4. Conforto e Produtividade

Manutenção de um ambiente confortável e adequado para estudantes e funcionários. Redução de interrupções causadas por falhas de equipamentos.

3.4. Arquitetura do Sistema

O sistema proposto neste trabalho é baseado em IoT. Basicamente, é composto por sensores que captam as informações do ambiente, como a medição de corrente e de tensão e um microcontrolador capaz de processar os dados e enviar para a aplicação o servidor web e para uma IHM (Interface Homem Máquina) e atuadores para poder ligar e desligar a carga caso seja necessário.. Os testes do sistema serão realizados com dois dispositivos,

um conectado em uma Lâmpada e outro em uma tomada. A Figura 9 ilustra uma arquitetura do sistema.

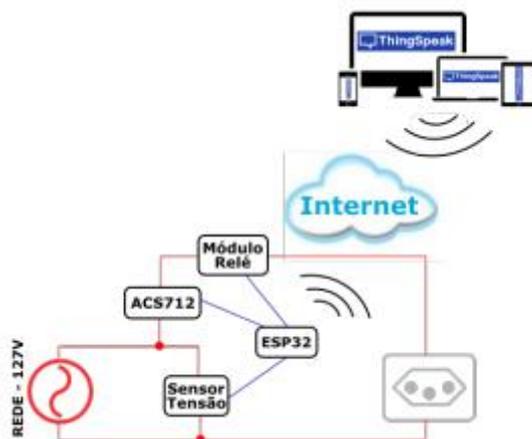


Figura 9: Arquitetura do sistema proposto

Fonte: Autor

Em relação ao sensoriamento da corrente, por se tratar de um sensor de corrente invasivo, o ACS712 é conectado em série com a carga para que a medição ocorra. Depois, é conectado em série o módulo relé que é responsável por realizar o controle Liga/Desliga do dispositivo. O microcontrolador EPS32 é ligado a todos esses periféricos sendo responsável por receber comandos, tratar os dados dos sensores e enviá-los para o Servidor. Os dados armazenados na nuvem, podem ser visualizados e exportados. Também, por meio dessa aplicação, é realizada o acionamento remoto.

3.5. Sensor de Corrente

Neste trabalho foi utilizado o módulo do sensor ACS712 com a faixa de leitura $\pm 5A$ (Figura 10) que simplificada representa uma placa de circuito impresso contendo o sensor (1), capacitores necessários para o seu funcionamento (2), bornes para a ligação da corrente a ser medida (3) e por fim, um LED e um resistor que indica que está alimentado por um tensão 5V (4).

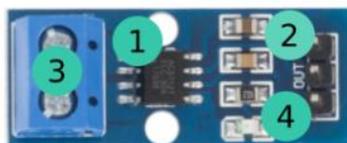


Figura 10: Módulo do sensor de corrente ACS712

Fonte: Autor

Por se tratar de um sensor de corrente invasivo, é necessário colocá-lo em série com a carga. A Figura 11 ilustra a configuração do circuito.

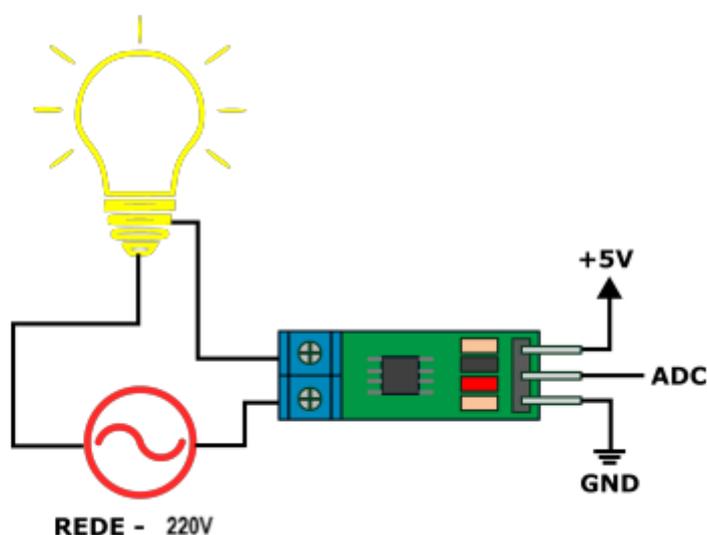


Figura 11: Circuito Sensor de corrente ACS712

é capturado 1920 amostras durante 1 segundo e então calcula-se o valor eficaz da tensão da saída do sensor. Porém, de acordo com o seu *datasheet* a tensão de saída é proporcional à corrente de entrada e possui um *offset* correspondente ao zero do sinal senoidal da corrente. Esta medida é igual a metade da alimentação. Diante disso, para encontrar o valor correspondendo ao *offset*, foi medido o valor ADC sem carga conectada ao sensor. O valor médio encontrado foi igual a 3000. Então, este valor é extraído da leitura ADC.

Por fim, a medida é dividida por 0,185 que representa a sensibilidade do sensor (185mV/A). A parte principal do código utilizado pode ser observado a seguir:

```

1  for (int i = 0; i <= 1920; i++){
2      leitura=analogRead(PinoCorrente); //aquisição da leitura do ADC
3      soma+=leitura*leitura - 3000; //somatório do quadrado da leitura
4      delayMicroseconds(390); //aguarda 390ms para próxima leitura

5  }
6  valorADC = sqrt(soma/1920); //faz o calculo do valor eficaz
7  valorADC = (3.30*valorADC)/4096; //Calcula a tensão da saída do sensor
8  corrente = valorADC/0.185; //Converte a tensão da saída em saída

```

A escolha do sensor ACS712 para o projeto é justificada com base em suas características técnicas, facilidade de integração e custo-benefício. Este sensor de corrente baseado no efeito Hall oferece uma solução prática e eficiente para medir correntes AC e DC, sendo amplamente utilizado em aplicações de monitoramento energético. A seguir, são apresentados os principais fatores que motivaram sua seleção:

- **Capacidade de Medição de Corrente AC e DC**

O ACS712 é capaz de medir correntes alternadas (AC) e contínuas (DC), tornando-o uma opção versátil para aplicações que requerem monitoramento de diferentes tipos de sinais. Essa característica é especialmente útil em sistemas híbridos ou em aplicações que envolvam tanto cargas AC quanto fontes DC, como inversores ou baterias.

- **Faixas de Corrente Disponíveis**

O ACS712 é oferecido em diferentes modelos, com faixas de medição de $\pm 5A$, $\pm 20A$ e $\pm 30A$, permitindo sua utilização em diversas aplicações, desde sistemas residenciais de baixa potência até ambientes industriais. A escolha da faixa adequada possibilita medições precisas e confiáveis sem sacrificar a resolução ou a sensibilidade.

- **Isolamento Galvânico**

Por ser baseado na tecnologia de efeito Hall, o ACS712 oferece isolamento elétrico entre a carga e o circuito de controle. Isso aumenta significativamente a segurança do sistema, protegendo o microcontrolador e outros componentes sensíveis contra tensões perigosas, especialmente em sistemas de alta potência.

- **Saída Analógica Proporcional**

O sensor fornece uma saída analógica proporcional à corrente medida, o que simplifica a integração com microcontroladores como o ESP32. A interface analógica facilita a aquisição

de dados e o processamento por meio de conversores ADC integrados no microcontrolador, reduzindo a necessidade de circuitos adicionais.

- **Precisão e Estabilidade**

O ACS712 apresenta boa precisão para medições de corrente, sendo adequado para sistemas de monitoramento de consumo energético onde é importante detectar variações no consumo. Além disso, sua estabilidade térmica garante que o desempenho não seja significativamente afetado por mudanças de temperatura no ambiente.

- **Facilidade de Integração**

O sensor ACS712 é compacto e fácil de implementar em placas de circuito impresso (PCBs) ou sistemas de prototipagem. Sua compatibilidade com tensões de alimentação de 5V e a simplicidade do circuito de interface o tornam uma escolha prática para projetos de monitoramento.

- **Custo-Benefício**

O ACS712 combina um preço acessível com recursos avançados, proporcionando uma excelente relação custo-benefício. Isso o torna uma solução viável para projetos que necessitam de múltiplos sensores sem comprometer o orçamento total.

- **Suporte e Documentação**

O ACS712 é amplamente utilizado na comunidade de desenvolvedores, o que resulta em grande disponibilidade de tutoriais, bibliotecas e exemplos de código. Isso reduz o tempo de desenvolvimento e facilita a implementação do sensor no projeto.

A escolha do sensor ACS712 para o projeto de monitoramento e controle de energia é fundamentada em sua versatilidade, precisão e facilidade de uso. Sua capacidade de medir correntes AC e DC com isolamento galvânico e uma saída analógica simples o tornam uma solução eficiente e segura. Além disso, seu custo acessível e a ampla disponibilidade de suporte técnico contribuem para a criação de um sistema confiável e econômico. Essas características fazem do ACS712 a opção ideal para atender às necessidades do projeto.

3.6. Sensor de Tensão

A tensão do dispositivo é medida por meio da utilização do circuito ilustrado na

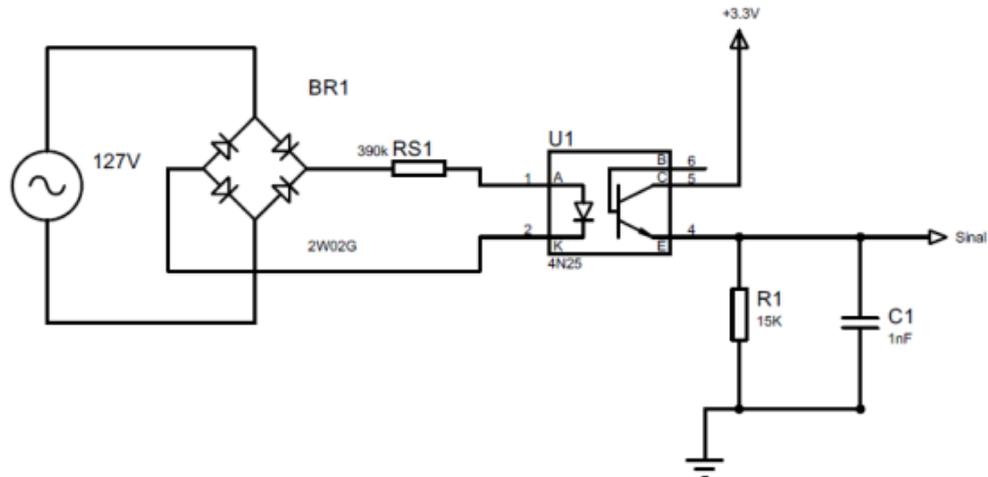


Figura 12: Circuito para medir a Tensão AC

Fonte: Autor

Primeiramente uma ponte retificadora de onda completa é utilizada para converter o meio ciclo negativo da tensão CA em positivo. Depois é utilizado um resistor para limitar a corrente que segue para o optoacoplador 4N25. Este último, é responsável por isolar o circuito de alta tensão para o de baixa tensão. Por fim, é utilizado um resistor para que o ADC do microcontrolador consiga medir o sinal e também um capacitor para filtrá-lo. Os valores lidos pelo microcontrolador são calculados por um algoritmo que retorna o valor RMS da tensão.

Este circuito para medição de tensão foi escolhido, pois o sinal correspondente a tensão aplicada é dada em forma de onda senoidal, tornando possível o cálculo do valor eficaz.

3.7. Microcontrolador

Escolher o microcontrolador ideal para um projeto é um processo que depende de vários fatores, relacionados tanto às necessidades técnicas do projeto quanto às restrições de tempo, custo e recursos disponíveis. Para o presente projecto, foi usado o microcontrolador ESP32 indicado na figura. Abaixo estão os critérios mais importantes que se considerou para a escolha.

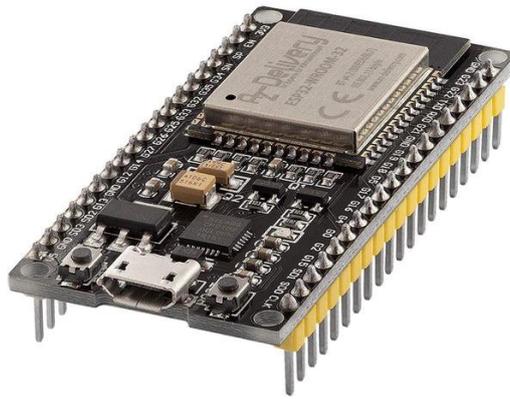


Figura 13: Microcontrolador Esp32

A escolha do ESP32 como microcontrolador para o projeto de monitoramento e controle de energia é fundamentada em suas características técnicas, versatilidade e custo-benefício. Este dispositivo se destaca como uma solução robusta e eficiente para atender às demandas do projeto. A seguir, são detalhados os principais fatores que justificam essa decisão:

3.7.1. Factores que contribuíram para a escolha do microcontrolador

- **Conectividade Avançada**

O ESP32 oferece conectividade Wi-Fi e Bluetooth integradas, permitindo que o sistema de monitoramento envie dados em tempo real para a nuvem ou dispositivos móveis. Isso é essencial para aplicações de IoT, onde a transmissão contínua de informações sobre consumo energético é uma exigência. O suporte ao protocolo MQTT e outros protocolos de comunicação facilita a integração com plataformas de gerenciamento e controle remoto.

- **Desempenho e Eficiência**

Com um processador dual-core de 32 bits (arquitetura Xtensa LX6) e uma frequência de operação de até 240 MHz, o ESP32 oferece desempenho mais do que suficiente para processamento rápido de dados, como leituras de sensores de corrente e tensão. Sua capacidade de realizar cálculos em tempo real permite o monitoramento preciso do consumo energético, além de possibilitar a execução de algoritmos de controle.

- **Recursos de Periféricos**

O ESP32 possui múltiplos periféricos integrados, como ADCs de 12 bits para leitura analógica, PWMs para controle de dispositivos e interfaces digitais como I2C, SPI e UART. Esses recursos são fundamentais para conectar sensores de corrente (ex.: sensores ACS712) e atuadores como relés ou contactores, garantindo a versatilidade necessária para o projeto.

- **Consumo de Energia Ajustável**

Embora o projeto envolva monitoramento de energia, o ESP32 apresenta modos de economia de energia que o tornam ideal para aplicações onde a eficiência energética do próprio sistema é importante. Ele suporta modos de baixo consumo, como "deep sleep", permitindo uma operação eficiente em aplicações alimentadas por baterias.

- **Suporte à Criptografia e Segurança**

O ESP32 inclui recursos de segurança de hardware, como aceleração de criptografia e suporte a TLS/SSL, que garantem a proteção dos dados transmitidos. Isso é crucial para o monitoramento de energia, onde a confidencialidade e integridade dos dados são importantes, especialmente em ambientes industriais ou residenciais conectadas.

- **Custo-Benefício**

O ESP32 combina alto desempenho e recursos avançados com um custo relativamente baixo, tornando-o uma escolha econômica para projetos de monitoramento e controle de energia. Sua ampla adoção no mercado e o suporte por uma grande comunidade de desenvolvedores garantem acesso a bibliotecas, exemplos de código e placas de desenvolvimento prontas, reduzindo o tempo e os custos de desenvolvimento.

- **Facilidade de Desenvolvimento**

O suporte a plataformas de prototipagem, como Arduino IDE, PlatformIO e ESP-IDF, facilita o desenvolvimento e a integração do firmware. A disponibilidade de placas de desenvolvimento amplamente difundidas, como o ESP32 DevKit, permite uma rápida validação das soluções, além de simplificar o processo de prototipagem e testes.

O ESP32 é uma escolha estratégica para o projeto de monitoramento e controle de energia devido à sua conectividade robusta, alto desempenho, recursos avançados de periféricos e segurança. Sua flexibilidade permite a implementação de um sistema eficiente, escalável e

integrado a ambientes modernos de IoT, garantindo uma solução confiável e econômica para o monitoramento energético.

3.8. Desenho do Circuito para o protótipo

Segue uma descrição geral de um circuito para **monitoramento e controle de energia** utilizando o **ESP32**, o sensor de corrente **ACS712**, e outros componentes necessários. Este projeto inclui monitoramento de corrente e controle de dispositivos com relés.

3.8.1. Componentes necessários

- **ESP32**: Microcontrolador para processamento e comunicação.
- **ACS712**: Sensor de corrente para medir o consumo.
- **Sensor de Tensão**: Para monitorar a tensão da rede (opcional, dependendo da necessidade de monitorar tensão).
- **Relé (módulo)**: Para controle de dispositivos.
- **Fonte de alimentação**: Para alimentar o ESP32 e o sistema.
- **Conectores ou bornes**: Para conexão com a carga e a rede elétrica.

3.9. Descrição do circuito

O circuito é dividido em três partes principais:

- **Monitoramento de corrente com ACS712.**
- **Controle de dispositivos com relé.**
- **Comunicação e processamento com ESP32.**

3.9.1. Monitoramento de Corrente

O sensor **ACS712** é conectado em série com a carga para medir a corrente que atravessa o circuito. O pino de saída do sensor fornece uma tensão proporcional à corrente, que será lida pelo pino ADC do ESP32.

- **Conexões do ACS712:**

- **VCC:** Conectado ao pino **3.3V ou 5V** (dependendo da versão do sensor).
- **GND:** Conectado ao **GND** do ESP32.
- **OUT:** Conectado a um pino ADC do ESP32 (ex.: GPIO34).

3.9.2. Controle de Dispositivos

Um **módulo relé** é usado para ligar/desligar dispositivos conectados à rede elétrica. O ESP32 controla o relé por meio de um pino digital.

- **Conexões do Relé:**

- **VCC:** Conectado ao **5V** do ESP32.
- **GND:** Conectado ao **GND** do ESP32.
- **IN:** Conectado a um pino digital do ESP32 (ex.: GPIO27).
- **Carga:** Conectada ao lado comutado do relé (Normalmente Aberto - NO).

3.9.3. Comunicação e Processamento

O ESP32 processa os dados coletados pelo sensor ACS712 e pode transmitir informações para a nuvem (via Wi-Fi) ou um aplicativo móvel. Ele também recebe comandos para controlar o relé.

3.9.4. Monitoramento de Tensão (Opcional)

Se necessário, um sensor de tensão pode ser usado para monitorar a tensão da rede elétrica (apenas com precaução em circuitos de alta tensão).

3.10. Dimensionamento da fonte de alimentação

A fonte de alimentação será responsável por fornecer alimentação para os microcontroladores, o circuito de comando das cargas, , os sensores de corrente e tensão e o display.

entrada da alimentação do dispositivo será USB tipo C, pós essa alimentação já vem embutido no microcontrolador esp32 e é fácil de encontrar no mercado. cujas folhas dados se encontram em anexo.

Abaixo na tabelas 3 faz-se o levantamento das cargas a que fonte terá de alimentar por forma a determinar a corrente da fonte total que o dispositivo ira consumir.

Tabela 3: Dimecionamento da fonte de alimentação

Ordem	Carga	IMax	VNominal
1	Microcontrolador Esp32	240mA	5V
2	Sensor de Corrente	20mA (x4)	5V
3	Sensor de Tensão	15mA	5V
4	Display LCD 20x5	40mA	5V
5	Módulo Relém	100mA (x6)	5V
	Total:	475mA	

O consumo total do sistema é de no máximo 0,5A, então para minimizar o trabalho e custos, essas fontes podem ser adquiridas no mercado por um valor de 50mt-200mt no máximo.

3.11. Desenho do circuito do protótipo

Depois de se seleccionar todos os componentes necessário, desenhou-se um circuito para a prototipagem da proposta e para se seguir a fase de testes. Na figura 14 esta apresentado um esquema eletrónico para o protótipo do projecto.

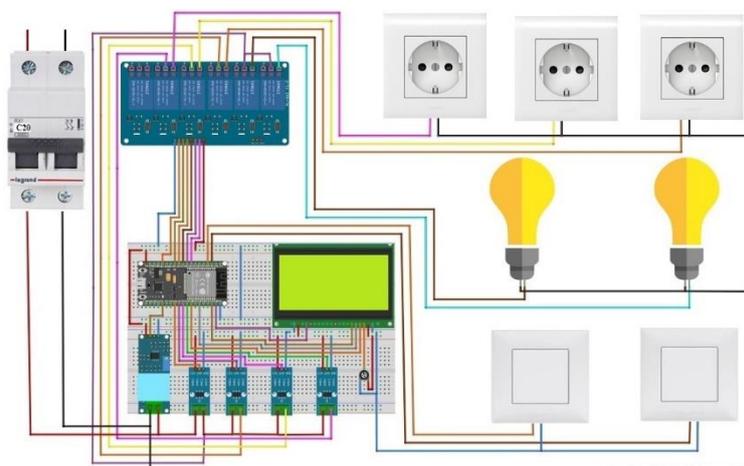


Figura 14: Esquemas em 2D do protótipo

Nesse esquema vai se controlar duas lâmpadas e três tomadas para simular os dispositivos como AC's e outros dispositivos electronicos. No protótipo, foi adicionado um Display LCD que permite ver o consumo também através do mesmo.

No esquema que se segue, estão representadas as ligações feitas no protótipo.

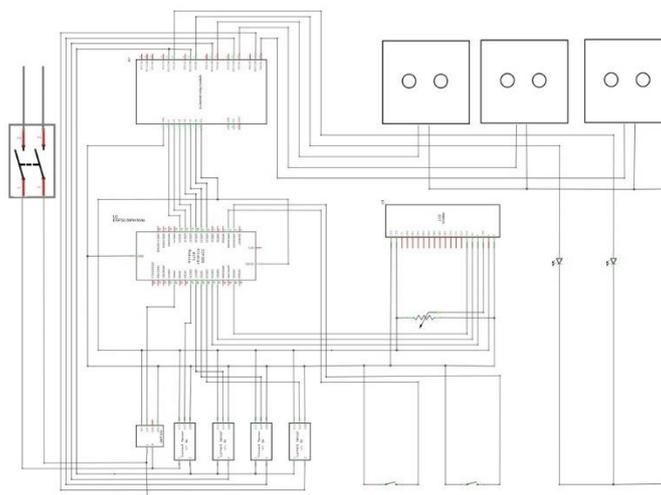


Figura 15: Circuito do protótipo

4. CAPÍTULO 4: TESTE E RESULTADOS

4.1. Testes e Resultados

O projeto em questão terá como estudo de caso o Instituto Médio de Ciências e Gestão (IMCG). Este instituto abrange uma diversidade de espaços, incluindo salas de aula, laboratórios, salas de informática, biblioteca, gabinetes dos diretores e professores, além da secretaria, entre outros. Em quase todos espaços, é comum encontrar equipamentos de grande consumo energético, como os ar condicionados, iluminação e outros equipamentos.

O laboratório é um espaço com mais necessidade de ter o controle dos equipamentos que consomem energia eléctrica, devido a variedade de equipamentos que la se encontram, algum com um alto consumo de energia e outros que devem ser mais cuidadosos ao utilizar

para evitar que de alguma forma possam ser danificados devido ao consumo de energia. Os equipamentos que podemos encontrar no laboratório.

Na tabela abaixo, esta apresentada uma tabela de consumo de energia por hora de todos os equipamentos principais que o sistema pode controlar no instituto.

Tabela 4: Principais dispositivos de consumo

Local	Equipamento	Quantidade	Consumo (W)	Horas de uso/dia	Consumo/dia (KWh)
Sala de aulas	Ar condicionados	16	1800	14	403.2
	Lâmpadas	64	20	14	17.92
Laboratório	Ar condicionados	1	1800	14	25.2
	Lâmpadas	6	20	14	1.68
	Autoclaves	1	1200	14	16.8
	Lâmpada de exames	2	200	9	3.6
Outros	Bomba de agua	1	2200	6	13.2
	Lâmpadas	20	20	14	5.6
	Ar condicionados	7	1800	14	176.4
				Total	663.6

A tabela mostra que o consumo diário de energia pode chegar a 663,6KWh o que significa que segundo a tabela do anexo II, o consumo diário em Mt pode ser obtida multiplicando esse valor a 14.6Mt, o que resulta em 9699,56Mt nos dias de maior consumo.

4.1.1. Montagem do protótipo

O protótipo foi montado numa placa de madeira, onde tem 2 interruptores que controlam duas lâmpadas para simular a iluminação do instituto e também tem 3 tomadas para poder conectar as cargas que desejas ser controlados. A figura 16 mostra o protótipo depois de montado.

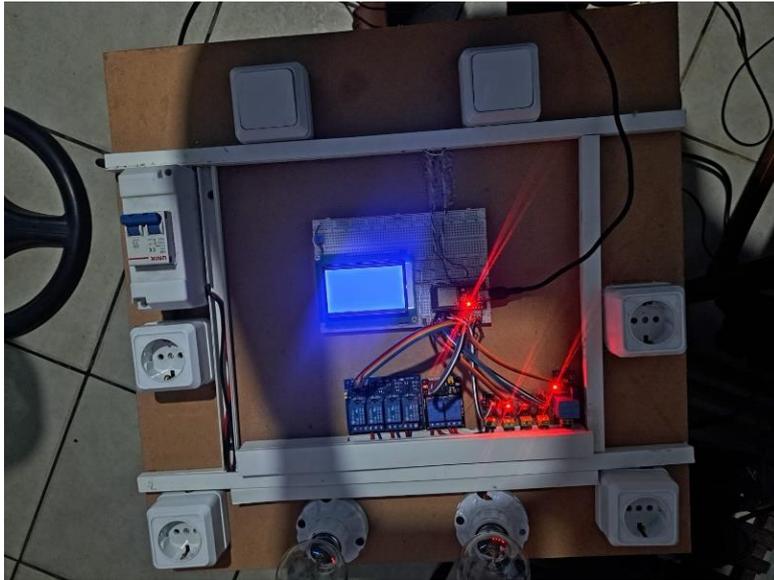


Figura 16: Protótipo

4.1.2. Custos do Protótipo

Para fazer o protótipo foi usado o material listado na tabela abaixo e os respectivos custos.

Tabela 5: Preços e quantidade dos componentes usados

Componentes	Quantidades	Valor unitário (MT)	Preço total (MT)
Node MCU esp32	1	1650	1650
Módulo Relé de 6 canais	1	1100	1100
Lâmpadas	2	30	60
Cabos	1	700	700
Fonte de 5V	1	200	50
Protoboard	1	300	300
Tomadas	3	80	240
Bocal	2	25	50
Sensor de Corrente	4	650	2600
Sensor de Tensão	1	650	650
Capacitor Variavel	1	40	40
Interuptor	2	80	160
Digitor duplo	1	800	800
Tabua	1	400	400
Calha	2	100	200
		Total	9450

Com a tabela acima, tem-se uma ideia de quanto custaria a montagem desse sistema, faltando no protótipo acima a placa de circuito impresso, que será desenhado com auxílio

do software proteus e impresso na china, o que não alteraria muito no custo final do dispositivo.

Com o projeto foi possível entender escopos e ferramentas que auxiliam muito na integração de ideias em soluções de hardware e software, criando melhorias de análise dos estudos e soluções para desenvolver da melhor forma possível a criação de um projeto que entregasse de forma factível um projeto IOT (internet of things). sendo plausível com ambientes de código aberto, protocolos que criam regras e meios de comunicação para um determinado segmento de trocas de dados.

A instalação do hardware ESP832, combinado com sensores de corrente e tensão, e conectado a um servidor MQTT. os resultados forneceram uma ilustração convincente de como a integração de várias tecnologias em desenvolvimento pode produzir um sistema de monitoramento e gerenciamento de energia mais eficaz e potente.

Com seus poderosos recursos de processamento e comunicação, o hardware ESP32 acabou sendo uma excelente escolha para o nosso projeto. o uso do ESP32, equipado com sensores de corrente, tensão, temperatura e umidade. Permitiu avaliar com precisão o consumo de energia em tempo real e transmitir dados pela rede com eficiência, design leve de publicação/assinatura do protocolo MQTT, o sistema foi capaz de se conectar com o servidor de forma eficaz, usando menos largura de banda.

O sistema de monitoramento de energia se mostrou confiável, eficaz e altamente adaptável. A iniciativa é uma evidência da força e adaptabilidade das tecnologias IoT (Internet of Things) e, sem dúvida, estimulará mais pesquisas no futuro.

4.2. Interface do usuário

A interface é a estrutura responsável por possibilitar a interação e controle e monitoramento de todo o sistema autônomo, através de representações gráfica com o usuário, tendo como foco a funcionalidade e integração da planta, de maneira simples, apresentando comandos que possam ser manipulados, compreendidos e acessíveis por qualquer pessoa autorizada.

A página desenvolvida possui 5 botoes, cada um com suas respectivas funções, são eles que enviam os comandos para o ESP32.

As figuras abaixo ilustram dois possíveis cenários da página, a figura 18 mostra uma interface de um utilizador que usa smartphone para o controle, e a figura 17 mostra a interface do utilizador que usa um computador.

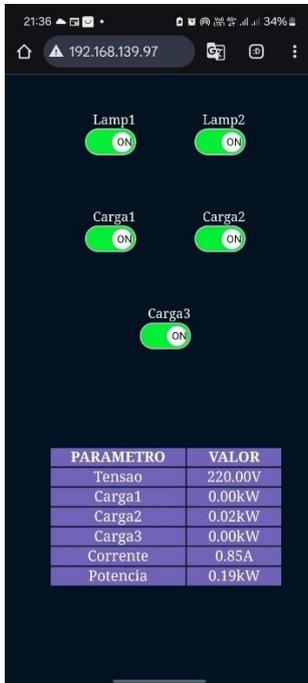


Figura 18: Interface de controle com Smartphone



Figura 17: Interface de Controle com Computador

Também é possível verificar os dados através do display de controle instalado no protótipo como mostra a figura 19.



Figura 19: Informação no display de controle

4.3. Resultados

Depois dos testes feitos no Institutos, identificou-se que os equipamentos como Impressoras, Computadores, projetores, entre outro, não precisam de monitoramento, pós, quando estão no modo *Sleep*, não consome energia.

Com o projecto, espera-se reduzir o consumo diário dos principais equipamentos que consomem energia como ar condicionados e lâmpadas. Com o uso desse sistema, é possível monitorar o consumo e desligar os aparelhos quando forem ligados sem necessidade. Com isso, espera-se reduzir o consumo de energia em até 50%.

O microcontrolador usado no sistema possui 6 entradas analógicas, com isso, é possível controlar até 6 cargas ao mesmo tempo, para a sua implementação no instituto, seriam necessários multiplicadores de portas, o que alteraria o custo de desenvolvimento do sistema.

5. CAPÍTULO 5: CONCLUSÕES E RECOMENDAÇÕES

5.1. Conclusão

O projeto realizado alcançou os objetivos propostos ao desenvolver um sistema eficiente de monitoramento e controle do consumo de energia elétrica baseado em IoT. Através da integração de sensores, microcontroladores ESP32, relés inteligentes e uma interface de gerenciamento remoto, foi possível criar um protótipo funcional que demonstrou sua eficácia em reduzir desperdícios energéticos em ambientes escolares.

Os testes realizados no Instituto Médio de Ciências e Gestão comprovaram que a solução é capaz de monitorar em tempo real o consumo de energia de equipamentos como iluminação e sistemas de climatização, desligando-os em períodos de inatividade. Essa funcionalidade não apenas gera economia financeira para a instituição, mas também contribuiu para a promoção de práticas sustentáveis e para a conscientização sobre o uso eficiente de recursos energéticos.

Além do impacto direto na redução de custos operacionais, o projeto se destacou pela sua flexibilidade e escalabilidade. A utilização de tecnologias amplamente disponíveis no mercado, como o microcontrolador ESP32 e o protocolo MQTT, garante que o sistema possa ser facilmente replicado ou adaptado para atender diferentes demandas.

Como perspectiva futura, recomenda-se a integração de algoritmos de inteligência artificial para prever padrões de consumo e otimizar ainda mais o desempenho do sistema. Além disso, a implementação de funcionalidades adicionais, como monitoramento de outros recursos (água e gás, por exemplo), pode expandir os benefícios do sistema. Este trabalho reforça o papel transformador da IoT na modernização de instituições educacionais, contribuindo para uma gestão mais eficiente e sustentável dos recursos.

6. CAPITULO 7: BIBLIOGRAFIA

6.1. Bibliografia

- BARTHOLD, L. O.; REPPEN, N. D.; HEDMAN, D. E. Análise de circuitos de sistemas de potência. 2. ed. Santa Maria Universidade Federal de Santa Maria,. **1983**
- BOLZANI, C. A. M. Residências Inteligentes. [S.I.]: Livraria da Física, 2004;
- SADIKU; ALEXANDER; MUSA, Fundamentos de Circuitos Elétricos, 2014
- MATTEDE, H. Medidor de energia elétrica, tipos e características. Disponível em: <<https://www.mundodaeletrica.com.br/medidor-de-energia-eletrica-tipos-caracteristicas/>>. Acesso em 10 agosto 2024.
- FOROUZAN, Behrouz A. Comunicação de dados e redes de computadores. 3a. ed. Porto Alegre: Bookman, 2006 BR/docs/Learn /Common_questions/ What_is_a_web_server
- Carvalho, É. d., & Penido, C. (2013). Microcontroladores. Minas Gerais.
- KOLBAN, Neil. Kolban's Book on ESP8266. Texas, USA. 2015.
- PEREIRA, Silvio do Lago. Estruturas de Dados em C. Uma Abordagem Didática.
- FLANAGAN, David. JavaScript: The Definitive Guide. [S.I.]: O'Reilly Media, 2011
- DEITEL, Harvey M. C++: Como Programar. [S.I.]: Pearson Universidades, 2006
- OGATA, K. Engenharia de Controle Moderno. [S.I.]: Prentice-Hall, 2003. v. 4ª edição.
- PANESI, André R. Quinteros. Fundamentos de Eficiência Energética. São Paulo, SP: Ensino Profissional, 2006
- LAKATOS, Eva Maria; MARCONI, Marina de Andrade. Fundamentos de metodologia científica. 5. ed. São Paulo: Atlas, 2003
- OLIVEIRA, P.C. Uso do microcontrolador ESP8266 para automação residencial. Disponível em: <<http://repositorio.poli.ufrj.br/monografias/monopoli10019583.pdf>>. Acesso em 12 abr 2022
- Ranger, S. (2023, 03). What is the IoT? Everything you need to know about the Internet of things right now. Retrieved July 20, 2024, from <https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/> Referências Bibliográficas

7. ANEXO 1: CÓDIGO

```
/*
UNIVERSIDADE EDUARDO MONDLANE
SISTEMA DE MONITORAMENTO E CONTROLE DE CARGAS
AUTOR: JOAO JUNIOR MASSINGUE
DATA: DEZEMBRO DE 2024
*/

// BIBLIOTECAS
#include <U8g2lib.h>
#include "EmonLib.h"
#define CURRENT_CAL 0.740
#define VOLT_CAL 99

//DEFINICAO DOS SENSORES
EnergyMonitor corrent1;
EnergyMonitor corrent2;
EnergyMonitor corrent3;
EnergyMonitor corrent4;
EnergyMonitor tensao;

bool track1=false;
bool track2=false;

float supplyVoltage;
double currentDraw1;
double currentDraw2;
double currentDraw3;
double currentDraw4;

//DEFINICAO DOS PINOS
#define sensor1_pino37 5
#define sensor2_pino31 18

#define sensor1_corrente 36
#define sensor2_corrente 39
#define sensor3_corrente 34
#define sensor4_corrente 35
#define sensor_tensao 32

#define relay1_pino33 14
#define relay2_pino25 13

#define relay3_pino26 25
#define relay4_pino27 33
#define relay5_pino16 26
#define relay6_pino17 27
```

```

//DECLARACAO DAS VARIAVES
bool lamp1=true;
bool lamp2=true;
bool lamp3=true;
bool lamp4=true;
bool lamp5=true;
bool lamp6=true;

String Sensor1="";
String Sensor2="";
String JSONtxt;
float carga1;
float carga2;
float carga3;
float carga4;
float potencia;

String Tensao;
String Carga1;
String Carga2;
String Carga3;
String Carga4;
String Potencia;

U8G2_ST7920_128X64_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 22, /* data=*/ 21, /* CS=*/
19); // ESP32
char* dado;

//INCLUSAO DA BIBLIOTECA DO SERVIDOR WEB E DECLARACAO DAS VARIAVEIS DE WIFI E SANHA
#include <WiFi.h>
#include <WebServer.h>
#include <WebSocketsServer.h>
const char* ssid = "JONNY";
const char* password = "123456789";

//INICIAR O WEB SERVER
WebServer server(80);
WebSocketsServer webSocket = WebSocketsServer(81);

//PAGINA HTML DO WEB SERVER
char webpageCont[] PROGMEM =
  R"====(
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```
<title>Automacao</title>
<style>

    body{
        background: #031321;
        width: 100%;
        display: flex;
        text-align: center;
        justify-content: center;
        align-items: center;
        flex-direction: column;
    }
    .container_principal{

        width: 70%;
        column-count: 2;
        column-gap: 3px;
        flex-wrap: wrap;

    }
    .container{
        min-height: 15vh;
        display: flex;
        align-items: center;
        justify-content: center;

    }
    .contain{
        height: 1px;
        position: relative;
    }

    .container .contain span{
        color: white;
        position: absolute;
        left: 10px;
        top: -20px;
    }
    .slider1{
        position: relative;
        background: white;
        width: 60px;
        height: 30px;
        border: 2px solid #bbb;
        border-radius: 20px;
        transition: all 300ms ease-in-out;
    }
    .slider2{
```

```
    position: relative;
    background: rgb(4, 240, 55);
    width: 60px;
    height: 30px;
    border: 2px solid #bbb;
    border-radius: 20px;
    transition: all 300ms ease-in-out;
}
```

```
.slider3{
    position: relative;
    background: rgb(4, 240, 55);
    width: 60px;
    height: 30px;
    border: 2px solid #bbb;
    border-radius: 20px;
    transition: all 300ms ease-in-out;
}
```

```
.slider4{
    position: relative;
    background: white;
    width: 60px;
    height: 30px;
    border: 2px solid #bbb;
    border-radius: 20px;
    transition: all 300ms ease-in-out;
}
```

```
.slider5{
    position: relative;
    background: rgb(4, 240, 55);
    width: 60px;
    height: 30px;
    border: 2px solid #bbb;
    border-radius: 20px;
    transition: all 300ms ease-in-out;
}
```

```
.button1{
    background: rgb(4, 240, 55);
    border: none;
    height: 25px;
    width: 25px;
    border-radius: 50%;
    pointer-events: none;
    top: 3px;
    left: 3px;
    position: absolute;
}
```

```
    transition: all 300ms ease-in-out;
}
```

```
.button2{
    background: white;
    border: none;
    height: 25px;
    width: 25px;
    border-radius: 50%;
    pointer-events: none;
    top: 3px;
    left: 32px;
    position: absolute;
    transition: all 300ms ease-in-out;
}
```

```
.button3{
    background: white;
    border: none;
    height: 25px;
    width: 25px;
    border-radius: 50%;
    pointer-events: none;
    top: 3px;
    left: 32px;
    position: absolute;
    transition: all 300ms ease-in-out;
}
```

```
.button4{
    background: rgb(4, 240, 55);
    border: none;
    height: 25px;
    width: 25px;
    border-radius: 50%;
    pointer-events: none;
    top: 3px;
    left: 3px;
    position: absolute;
    transition: all 300ms ease-in-out;
}
```

```
.button5{
    background: white;
    border: none;
    height: 25px;
    width: 25px;
    border-radius: 50%;
```

```

        pointer-events: none;
        top: 3px;
        left: 32px;
        position: absolute;
        transition: all 300ms ease-in-out;
    }

.tabelas{
    display: flex;
    justify-content:center;
    align-items: center;
    margin-bottom: 10px;
    flex-direction: column;
    height: 45vh;

}
.detalhes{
    width: 380px;
    height: 190px;
    background-color: rgb(178, 178, 212);
}
.tabela {

display:flex;
justify-content: center;
}
table{
    display:flex;
    text-align: center;
    justify-content: center;
    width: 100%;
    color: white;
    font-size: 18px;
}
th{
    background: rgba(113, 99, 186, 255);
    text-transform: uppercase;
}
td{

    background: rgba(113, 99, 186, 255);

}
th, td{
    padding-left: 25px;
    padding-right: 25px;
}
.label{

```

```

width: 65px;
height: 35px;
z-index: 30;
}
</style>
</head>
<body>
<div class="container_principal">
  <div class="container">

    <div class="contain">

      <div class="label" onclick="myFuncao1()">
        <div class="slider1">
          <button class="button1">OFF</button>
        </div>
      </div>
      <span id="myID1">Lamp1</span>
    </div>
  </div>

  <div class="container">

    <div class="contain">

      <div class="label" onclick="myFuncao2()">
        <div class="slider2">
          <button class="button2">ON</button>
        </div>
      </div>
      <span>Carga1</span>
    </div>
  </div>

  <div class="container">

    <div class="contain">

      <div class="label" onclick="myFuncao4()">
        <div class="slider4">
          <button class="button4">OFF</button>
        </div>
      </div>
      <span id="myID2">Lamp2</span>
    </div>
  </div>

  <div class="container">

```

```

    <div class="contain">

    <div class="label" onclick="myFuncao5()">
    <div class="slider5">
        <button class="button5">ON</button>
    </div>
</div>
    <span>Carga2</span>
</div>
</div>
</div>
</div>

```

```

<div class="container">

    <div class="contain">

    <div class="label" onclick="myFuncao3()">
    <div class="slider3">
        <button class="button3">ON</button>
    </div>
</div>
    <span>Carga3</span>
</div>
</div>
    <div class="tabelas">

    <div class="tabela">
        <table>
            <tr>
                <th>parametro </th>
                <th>valor</th>
            </tr>
            <tr>
                <td>Tensao</td>
                <td id="tensao">0</td>
            </tr>
            <tr>
                <td >Carga1</td>
                <td id="carga1">0</td>
            </tr>
            <tr>
                <td>Carga2</td>
                <td id="carga2">0</td>
            </tr>
            <tr>
                <td>Carga3</td>
                <td id="carga3">0</td>
            </tr>
        </table>
    </div>

```

```

        </tr>
        <tr>
            <td>Corrente</td>
            <td id="carga4">0</td>
        </tr>
        <tr>
            <td>Potencia</td>
            <td id="potencia">0</td>
        </tr>
    </table>
</div>
</div>
<script>
    InitWebSocket()
    function InitWebSocket(){

```

```
Socket = new WebSocket('ws://' + window.location.hostname + ':81/');
```

```

var cont1=false;
var cont2=true;
var cont3=true;
var cont4=false;
var cont5=true;
var cont6=true;
const button1= document.querySelector('.button1');
const slider1=document.querySelector(".slider1");

```

```

const button2= document.querySelector('.button2');
const slider2=document.querySelector(".slider2");

```

```

const button3= document.querySelector('.button3');
const slider3=document.querySelector(".slider3");

```

```

const button4= document.querySelector('.button4');
const slider4=document.querySelector(".slider4");

```

```

const button5= document.querySelector('.button5');
const slider5=document.querySelector(".slider5");

```

```

const button6= document.querySelector('.button6');
const slider6=document.querySelector(".slider6");

```

```

const carga1=document.getElementById("carga1");
const carga2=document.getElementById("carga2");

```

```

const carga3=document.getElementById("carga3");
const carga4=document.getElementById("carga4");
const tensao=document.getElementById("tensao");
const potencia=document.getElementById("potencia");

Socket.onmessage=function(evt){
JSONObj = JSON.parse(evt.data);
dado1=JSONObj.SNS1;
dado2=JSONObj.SNS2;
tens=JSONObj.TENSAO;
carg1=JSONObj.CARGA1;
carg2=JSONObj.CARGA2;
carg3=JSONObj.CARGA3;
carg4=JSONObj.CARGA4;
potenc=JSONObj.POTENCIA;

carga1.innerText=carg1+"kW";
carga2.innerText=carg2+"kW";
carga3.innerText=carg3+"kW";
carga4.innerText=carg4+"A";
tensao.innerText=tens+"V";
potencia.innerText=potenc+"kW";

    if(dado1==1){
        button1.style.left=32+"px";
        button1.style.left=32+"px";
        button1.style.background="white"
        button1.innerText="ON"
        slider1.style.background="rgb(4, 240, 55)";
    }
    else{
        button1.style.left=3+"px";
        button1.style.background="rgb(4, 240, 55)"
        button1.innerText="OFF"
        slider1.style.background="white";
    }

    if(dado2==1){
        button4.style.left=32+"px";
        button4.style.left=32+"px";
        button4.style.background="white"
        button4.innerText="ON"
        slider4.style.background="rgb(4, 240, 55)";
    }
    else{
        button4.style.left=3+"px";
        button4.style.background="rgb(4, 240, 55)"

```

```

        button4.innerText="OFF"
        slider4.style.background="white";
    }

    console.log(String(cont1)+String(cont2)+String(cont3)+String(cont4))
}

function myFuncao1(){
    Socket.send("led10n");
    cont1=!cont1;

    if(cont1==true){
        button1.style.left=32+"px";
        button1.style.left=32+"px";
        button1.style.background="white"
        button1.innerText="ON"
        slider1.style.background="rgb(4, 240, 55)";
        Socket.send("led10n");
    }else{

        button1.style.left=3+"px";
        button1.style.background="rgb(4, 240, 55)"
        button1.innerText="OFF"
        slider1.style.background="white";
        Socket.send("led10f");

    }

}

function myFuncao2(){
    cont2=!cont2;
    if(cont2==true){
        button2.style.left=32+"px";
        button2.style.left=32+"px";
        button2.style.background="white"
        button2.innerText="ON"
        slider2.style.background="rgb(4, 240, 55)";
        Socket.send("led40n");
    }else{
        button2.style.left=3+"px";
        button2.style.background="rgb(4, 240, 55)"
        button2.innerText="OFF"
        slider2.style.background="white";
        Socket.send("led40f");
    }
}

```

```

}

function myFuncao3(){

    cont3=!cont3;
    if(cont3==true){
        button3.style.left=32+"px";
        button3.style.left=32+"px";
        button3.style.background="white"
        button3.innerText="ON"
        slider3.style.background="rgb(4, 240, 55)";
        Socket.send("led3On");
    }else{
        button3.style.left=3+"px";
        button3.style.background="rgb(4, 240, 55)"
        button3.innerText="OFF"
        slider3.style.background="white";
        Socket.send("led3Of");
    }
}
}

```

```

function myFuncao4(){

    cont4=!cont4;
    if(cont4==true){
        button4.style.left=32+"px";
        button4.style.left=32+"px";
        button4.style.background="white"
        button4.innerText="ON"
        slider4.style.background="rgb(4, 240, 55)";
        Socket.send("led2On");
    }else{
        button4.style.left=3+"px";
        button4.style.background="rgb(4, 240, 55)"
        button4.innerText="OFF"
        slider4.style.background="white";
        Socket.send("led2Of");
    }
}
}

```

```

function myFuncao5(){

    cont5=!cont5;

    if(cont5==true){
        button5.style.left=32+"px";
        button5.style.left=32+"px";
    }
}

```

```

        button5.style.background="white"
        button5.innerText="ON"
        slider5.style.background="rgb(4, 240, 55)";
        Socket.send("led50n");
    }else{
        button5.style.left=3+"px";
        button5.style.background="rgb(4, 240, 55)"
        button5.innerText="OFF"
        slider5.style.background="white";
        Socket.send("led50f");
    }
}

```

```

function myFuncao6(){

    cont6=!cont6;
    console.log("Clicado");

    if(cont6==true){
        button6.style.left=32+"px";
        button6.style.left=32+"px";
        button6.style.background="white"
        button6.innerText="ON"
        slider6.style.background="rgb(4, 240, 55)";
        Socket.send("led60n");
    }else{
        button6.style.left=3+"px";
        button6.style.background="rgb(4, 240, 55)"
        button6.innerText="OFF"
        slider6.style.background="white";
        Socket.send("led60f");
    }
}

```

</script>

</body>
 </html>
)=====";

```

void u8g2_prepare(void) {
    u8g2.setFont(u8g2_font_helvr08_tr);
    u8g2.setFontRefHeightExtendedText();
    u8g2.setDrawColor(1);
    u8g2.setFontPosTop();
    u8g2.setFontDirection(0);
}

```

```

void draw(void) {
  u8g2_prepare();
  u8g2.drawStr( 0, 0, "REDE");
  u8g2.setCursor(0, 15);
  u8g2.print(String("U:")+String(supplyVoltage)+String("V"));

  u8g2.setCursor(0, 35);
  u8g2.print(String("I:")+String(currentDraw2)+String("A"));

  u8g2.setCursor(0, 55);
  u8g2.print(String("P:")+String(potencia)+String("kW"));

  u8g2.drawStr( 60, 0, "CARGAS");

  u8g2.setCursor(60, 15);
  u8g2.print(String("CRG1:")+String(carga1)+String("kW"));
  u8g2.setCursor(60, 35);
  u8g2.print(String("CRG2:")+String(carga4)+String("kW"));
  u8g2.setCursor(60, 55);
  u8g2.print(String("CRG3:")+String(carga3)+String("kW"));

}

void setup(void) {
  Serial.begin(9600);
  u8g2.begin();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print("."); delay(500);
  }
  WiFi.mode(WIFI_STA);
  Serial.print(" Local IP: ");
  Serial.println(WiFi.localIP());

  server.on("/", handleRoot);
  server.begin(); websocket.begin();
  websocket.onEvent(webSocketEvent);

  pinMode( sensor1_pino37, INPUT_PULLUP);
  pinMode( sensor2_pino31, INPUT_PULLUP);
  pinMode( sensor1_corrente, INPUT);
  pinMode( sensor2_corrente, INPUT);

```

```

pinMode( sensor3_corrente, INPUT);
pinMode( sensor4_corrente, INPUT);

tensao.voltage(sensor_tensao, VOLT_CAL, 1.7);
corrent1.current(sensor1_corrente , CURRENT_CAL);
corrent2.current(sensor2_corrente , CURRENT_CAL);
corrent3.current(sensor3_corrente , CURRENT_CAL);
corrent4.current(sensor4_corrente , CURRENT_CAL);

pinMode(relay1_pino33 , OUTPUT);
pinMode(relay2_pino25 , OUTPUT);
pinMode(relay3_pino26 , OUTPUT);
pinMode(relay4_pino27 , OUTPUT);
pinMode(relay5_pino16 , OUTPUT);
pinMode(relay6_pino17 , OUTPUT);

digitalWrite(relay1_pino33 , HIGH);
digitalWrite(relay2_pino25, HIGH);
digitalWrite(relay3_pino26 , LOW);
digitalWrite(relay4_pino27 , LOW);
digitalWrite(relay5_pino16 , LOW);
digitalWrite(relay6_pino17 , LOW);
}

void loop(void) {
    tensao.calcVI(17,2000);
    corrent1.calcVI(17, 100);
    corrent2.calcVI(17, 100);
    corrent3.calcVI(17, 100);
    corrent4.calcVI(17, 100);
    supplyVoltage = tensao.Vrms;
    if(supplyVoltage<=110){
        supplyVoltage=0.00;
    }else if(supplyVoltage>110&&supplyVoltage<=240){
        supplyVoltage=220.00;
    }else {
        supplyVoltage=224.88;
    }
    currentDraw1 = corrent1.Irms;
    currentDraw2 = corrent2.Irms;
    currentDraw3 = corrent3.Irms;
    currentDraw4 = corrent4.Irms;
    Serial.println(String("corrent1 =")+String(currentDraw1)+String("  corrent2 =
")+String(currentDraw2)+String("  corrent3
")+String(currentDraw3)+String("  corrent4 =")+String(currentDraw4));
    if(currentDraw1<=0.05){
        currentDraw1=0.00;
    }
}

```

```

}
if(currentDraw2<=0.05){
  currentDraw2=0.00;
}
if(currentDraw3<=0.05){
  currentDraw3=0.00;
}
if(currentDraw4<=0.05){
  currentDraw4=0.00;
}
carga1=supplyVoltage*currentDraw1;
carga3=supplyVoltage*currentDraw3;
carga4=supplyVoltage*currentDraw4;
potencia=supplyVoltage*currentDraw2;

server.handleClient();
websocket.loop();
u8g2.clearBuffer();
draw();
u8g2.sendBuffer();
bool sensor1=!digitalRead(sensor1_pino37);
bool sensor2=!digitalRead(sensor2_pino31);
if(sensor1){
  if(track1==false){
    lamp1=!lamp1;
    track1=true;
  }

  }else{
  if(track1==true){
    lamp1=!lamp1;
    track1=false;
  }
  }

if(sensor2){
  if(track2==false){
    lamp2=!lamp2;
    track2=true;
  }

  }else{
  if(track2==true){
    lamp2=!lamp2;
    track2=false;
  }
  }
Sensor1=String(lamp1);

```

```

Sensor2=String(lamp2);

Tensao=String(supplyVoltage);
String corent =String(currentDraw2);
Serial.println(Tensao);
Carga1=String(carga1);
Carga2=String(carga2);
Carga3=String(carga3);
Carga4=String(carga4);
Potencia=String(potencia);
digitalWrite(relay1_pino33 , !lamp1);
digitalWrite(relay2_pino25, !lamp2);
digitalWrite(relay3_pino26 , !lamp3);
digitalWrite(relay4_pino27 , !lamp4);

digitalWrite(relay5_pino16 , !lamp5);
digitalWrite(relay6_pino17 , !lamp6);

    JSONtxt = "{\"TENSAO\":\""+Tensao+"\"";
    JSONtxt += "\"SNS1\":\""+Sensor1+"\"";
    JSONtxt += "\"SNS2\":\""+Sensor2+"\"";
    JSONtxt += "\"CARGA1\":\""+Carga1+"\"";
    JSONtxt += "\"CARGA2\":\""+Carga4+ "\"";
    JSONtxt += "\"CARGA3\":\""+Carga3+ "\"";
    JSONtxt += "\"CARGA4\":\""+corent+"\"";
    JSONtxt += "\"POTENCIA\":\""+Potencia+"\"}";

    websocket.broadcastTXT(JSONtxt);
}
void websocketEvent(uint8_t num, WStype_t type, uint8_t * payload, size_t length)
{
    switch (type) {
        case WStype_DISCONNECTED:
            Serial.println("Client " + String(num) + " disconnected");
            break;
        case WStype_CONNECTED:
            Serial.println("Client " + String(num) + " connected");
            break;
        case WStype_TEXT:

            String msg=String((char*)(payload));
            Serial.println(msg);
            if(msg=="led1On"){
                lamp1=1;
                msg="";
            }else if(msg=="led2On"){

```

```

    lamp2=1;

    msg="";
}else if(msg=="led30n"){
    lamp3=1;
    msg="";
}else if(msg=="led40n"){
    lamp4=1;
    msg="";
}else if(msg=="led50n"){
    lamp5=1;
    msg="";
}else if(msg=="led60n"){
    lamp6=1;
    msg="";
}else if(msg=="led10f"){
    lamp1=0;
    msg="";
}else if(msg=="led20f"){
    lamp2=0;
    msg="";
}else if(msg=="led30f"){
    lamp3=0;
    msg="";
}else if(msg=="led40f"){
    lamp4=0;
    msg="";
}else if(msg=="led50f"){
    lamp5=0;
    msg="";
}else if(msg=="led60f"){
    lamp6=0;
    msg="";
}
break;
}
}
void handleRoot()
{
    server.send(200,"text/html",webpageCont);
}

```

8. ANEXO II



Tarifário de Energia Eléctrica

CATEGORIAS TARIFÁRIAS: Social, Doméstica, Agrícola e Geral (Baixa Tensão)

Consumos Registados (kWh)	PREÇO DE VENDA POR CATEGORIA TARIFÁRIA				Taxa fixa (Mt)
	Tarifa Social (MT/kWh)	Tarifa Doméstica (MT/kWh)	Tarifa Agrícola (Mt/kW)	Tarifa Geral (Mt/kW)	
De 0 a 125	0.97				
De 0 a 300		6.00	3.69	9.32	233.37
De 301 a 500		8.49	5.26	13.31	233.37
Superior a 500		8.91	5.75	14.56	233.37
Pré-pagamento	0.97	7.64	5.11	13.34	