

UNIVERSIDADE ESTÁCIO DE SÁ
ENGENHARIA ELÉTRICA
CAMPUS NORTE SHOPPING

CARLOS ANDRÉ DA COSTA FONSECA
DAVID CARDOSO FERNANDES DA SILVA

**SISTEMA DE BAIXO CUSTO PARA MONITORAMENTO DO
CONSUMO DE ENERGIA POR PADRÃO DE CARGAS UTILIZANDO
APRENDIZADO DE MÁQUINA**

Rio de Janeiro - RJ
2020

CARLOS ANDRÉ DA COSTA FONSECA
DAVID CARDOSO FERNANDES DA SILVA

**SISTEMA DE BAIXO CUSTO PARA MONITORAMENTO DO
CONSUMO DE ENERGIA POR PADRÃO DE CARGAS UTILIZANDO
APRENDIZADO DE MÁQUINA**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia
Elétrica da Universidade Estácio de Sá,
como requisito parcial para obtenção do
Título de Bacharel em Engenharia
Elétrica.

Orientador: Prof. Tiago Motta Quirino

Rio de Janeiro – RJ
2020

CARLOS ANDRÉ DA COSTA FONSECA
DAVID CARDOSO FERNANDES DA SILVA

**SISTEMA DE BAIXO CUSTO PARA MONITORAMENTO DO
CONSUMO DE ENERGIA POR PADRÃO DE CARGAS UTILIZANDO
APRENDIZADO DE MÁQUINA**

Trabalho de Conclusão de Curso defendido e aprovado em: 23 de Junho de 2020.

Banca examinadora:

Prof. Tiago Motta Quirino, M.Sc.
Orientador
Universidade Estácio de Sá

Prof. Carlos Henrique da Silva Rocha, M.Sc.
Universidade Estácio de Sá

Prof. Natália Corrêa Lecques Ruiz
Universidade Estácio de Sá

José Rodrigo de Castro Silva, M.Sc
Transpetro

“Não existem métodos fáceis para resolver problemas difíceis. “

René Descartes

RESUMO

O presente trabalho tem como objetivo construir um sistema simples de baixo custo capaz de monitorar o consumo de determinada carga (equipamento), visando fornecer informações para proporcionar a economia de energia. Propõe-se a utilização do microcontrolador ESP32 para coletar dados do sensor de corrente não intrusivo e alimentar um banco de dados com a relação de consumo instantâneo. O projeto proposto tem como base a aplicação de Aprendizado de Máquina aliado a algoritmos capazes de aprender padrões e realizar previsões minimizando assim, o custo elevado de projetos presentes no qual envolve um sensor de corrente por equipamento ligado na rede para monitoramento do consumo de energia. Com isso, o consumidor pode ter um *feedback* detalhado sobre o consumo de energia e com esses dados ele poderá reduzir o tempo de funcionamento de um determinado equipamento em uso excessivo ou até mesmo levar a substituição de um equipamento antigo por um mais moderno e econômico.

Palavras-Chave: Não intrusivo, microcontrolador ESP32, Eficiência Energética, Aprendizado de Máquina.

ABSTRACT

The present work has as objective to build a simple low-cost system capable of monitoring the consumption of a given workload (equipment), aiming to provide information to provide energy savings. It is proposed to use the ESP32 microcontroller to collect data from the current sense for non-invasive and feed a database with the instantaneous consumption. The proposed project has as its basis the application of Machine Learning ally the algorithms that are able to learn patterns and make predictions thus minimizing, the high cost of projects present in which involves a current probe by equipment connected on the network for monitoring the energy consumption. With this, the consumer may have a detailed feedback on energy consumption and with these data it can reduce the operating time of a particular product in use excessive or even lead to replacement of an old equipment for a more modern and economical.

Keywords: Non-Invasive, ESP32 microcontroller, Energy Efficiency, Machine Learning.

LISTA DE FIGURAS

Figura 1 – Arquitetura do ESP32	17
Figura 2 – WEMOS LOLIN32	19
Figura 3 - Arduino Mega	20
Figura 4 – Protocolo I2c	21
Figura 5 - Aprendizado de Máquinas	23
Figura 6 – Classificação de dados por KNN	26
Figura 7 – Sensor de corrente não invasivo	29
Figura 8 – Esquema resistor de BURDEN	30
Figura 9 - Sinal de corrente no secundário do sensor de corrente não intrusivo	31
Figura 10 - Conversor A/D	32
Figura 11 - Conversor A/D tipo delta-sigma	32
Figura 12 - Diagrama de blocos do funcionamento do sistema desenvolvido	38
Figura 13 - Visão geral do protótipo	39
Figura 14 - Esquema do sistema desenvolvido	40
Figura 15 - Ligação do resistor de burden no TC	41
Figura 16 - Vsaída CKT	42
Figura 17 - Offset DC	43
Figura 18 - Circuito final sensor + offset DC	43
Figura 19 - Formas de Onda	44
Figura 20 - Tela de preferência da IDE do Arduino	45
Figura 21 - Gerenciador de placas Arduino IDE	46
Figura 22 - Definição das variáveis de programação	47
Figura 23 - Fluxograma da rotina de aquisição de dados	47
Figura 24 - ADS1115	50
Figura 25 - Representação gráfica do sensor de corrente + Offset – A0 do Esp32	51
Figura 26 - Representação gráfica do sensor de corrente + Offset – A0 do ADS1115	52
Figura 27 - Dados de treino	53
Figura 28 - Gráfico do dataset de treino	54
Figura 29 - Erro para valores de K	55
Figura 30 - Matriz de confusão K=1	56
Figura 31 - Métricas de avaliação K=1	56
Figura 32 - Matriz de confusão K=3	57
Figura 33 - Métricas de avaliação K=3	57
Figura 34 - Condição desligado/ligado	58
Figura 35 - Fluxograma de comparação entre sistema existente x sistema apresentado	59

LISTA DE TABELAS

Tabela 1 – ESP32 x Arduino	18
Tabela 2 - Exemplo de uma matriz de confusão	27
Tabela 3 - Tabela de configuração do amplificador de ganho programável do ADS1115	50
Tabela 4 - Custo total do projeto existente x apresentado	60

LISTA DE ABREVIATURAS E SIGLAS

A/D	Analógico Digital
CLP	Controlador Lógico Programável
DC	<i>Direct current</i>
FN	<i>False Negative</i>
FP	<i>False Positive</i>
HVAC	<i>Heating, Ventilating and Air Conditioning</i>
I2C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
IEF	Corrente Eficaz
IoT	Internet das Coisas
IP	Corrente de Pico
I/O	Entrada e Saída
KNN	<i>K — Nearest Neighbors</i>
LED	<i>Light-emitting Diode</i>
MLP	<i>Multi Layer Perceptron</i>
MME	Ministério de Minas e Energia
PNEE	Plano Nacional de Eficiência Energética
TC	Transformador de Corrente
TN	<i>True Negative</i>
TP	<i>True Positive</i>
URL	<i>Uniform Resource Locator</i>
Wi-Fi	<i>Wireless Fidelity</i>

SUMÁRIO

CAPÍTULO 1 - CONSIDERAÇÕES INICIAIS	11
1.1 INTRODUÇÃO	11
1.2 CENÁRIO	12
1.3 PROBLEMA	12
1.4 JUSTIFICATIVA	13
1.5 OBJETIVOS	14
1.5.1 OBJETIVO GERAL.....	14
1.5.2 OBJETIVOS ESPECÍFICOS	14
1.6 METODOLOGIA.....	15
CAPÍTULO 2 - REFERENCIAL TEÓRICO	16
2.1 MICRONTROLADOR	16
2.1.1 ESP32	17
2.1.2 ARDUINO	20
2.1.3 PROTOCOLO I2C	21
2.2 PROGRAMAÇÃO	22
2.2.1 LINGUAGEM DE PROGRAMAÇÃO C/C++	22
2.2.2 LINGUAGEM DE PROGRAMAÇÃO PYTHON	22
2.3 APRENDIZADO DE MÁQUINA	23
2.3.1 ALGORITMO - VIZINHOS MAIS PRÓXIMOS (K-NEAREST NEIGHBORS)....	24
2.3.2 MATRIZ DE CONFUSÃO	26
2.4 TRANSFORMADORES DE CORRENTE.....	28
2.4.1 RESISTOR SHUNT OU BURDEN	29
2.5 CONVERSOR A/D	31
CAPÍTULO 3 - ESTADO DA ARTE	34
3.1 FORNAZARI, BORGES (2016)	34

3.2 QUIRINO T., QUIRINO J., AMARAL, NEPOMUCENO e SILVA (2019).....	34
3.3 MELO, CAMPOS e SILVA (2019)	35
CAPÍTULO 4 - APLICAÇÃO.....	37
4.1 IMPLEMENTAÇÃO DO SISTEMA DESENVOLVIDO	37
4.1.1 VISÃO GERAL DO PROTÓTIPO.....	38
4.1.2 ESQUEMA ELETRÔNICO DO SISTEMA DESENVOLVIDO	39
4.2 APLICAÇÃO DO SENSOR DE CORRENTE (TC)	41
4.3 PROGRAMAÇÃO DO MÓDULO LOLIN32	44
4.4 APLICAÇÃO DO CONVERSOR A/D	48
4.5 APLICAÇÃO DO ALGORITMO DE MACHINE LEARNING	53
4.5.1 RESULTADOS DO KNN – MATRIZ DE CONFUSÃO.....	55
4.6 INTERFACE AO USUÁRIO.....	58
4.7 VIABILIDADE ECONÔMICA	59
5 CONSIDERAÇÕES FINAIS	61
REFERÊNCIAS.....	62

CAPÍTULO 1 - CONSIDERAÇÕES INICIAIS

1.1 INTRODUÇÃO

No cenário econômico atual, onde ainda persistem os efeitos da crise econômica que o Brasil passou nos últimos anos, as empresas ainda enfrentam grandes dificuldades para se recuperarem. É visto que a redução de custos de produção é uma das principais iniciativas tomada pelos gestores para manterem a empresa em atividade.

O custo de energia elétrica vem aumentando consideravelmente em todo o país devido à escassez de chuvas e quedas no nível dos reservatórios das usinas hidrelétricas que compõem a maior parte da produção de energia elétrica do Brasil. Por conta disso, são utilizadas usinas termelétricas que possuem custos elevados e afetam negativamente o meio ambiente.

A conta de energia elétrica para as pequenas e médias indústrias pode representar até 40% do custo de produção (FIRJAN, 2017). Portanto, o monitoramento e gerenciamento de cargas tem sido um recurso estratégico de grande importância para as empresas e qualquer atividade que possa reduzir o consumo sem influenciar a produtividade.

A eficiência energética tem sido assunto em debate nos centros de pesquisa em eficiência energética e meios de comunicação. De acordo com o Plano Nacional de Eficiência Energética (PNEE), criado pelo Ministério de Minas e Energia (MME) a eficiência energética se refere às diversas ações que objetivam gerar a mesma quantidade de energia com menos recursos naturais ou obter o mesmo serviço (“realizar trabalho”) com menos energia para atender a demanda da sociedade.

Na abordagem de eficiência energética é comumente utilizado para esse feito a instalação de luminárias de *LED*, uso de automação para monitorar e gerenciar o sistema de carga, a troca de motores antigos por motores modernos etc.

A automação tem sido um grande aliado à eficiência energética. Hoje já podemos encontrar sistemas capazes de controlar o ligar e o desligar de luzes, controlar eletrodomésticos, HVAC (aquecimento, ventilação e ar condicionado) e outros sistemas residenciais e industriais.

Os sistemas de automação presentes hoje no mercado, visam efetuar o monitoramento de cargas através de inúmeros sensores de correntes ligados

individualmente a cada circuito para aferir a corrente circulada e calcular o consumo de energia tornando assim um projeto de alto custo já que os sensores de correntes presentes no mercado são de custo muito elevado.

Portanto, indaga-se: Como monitorar diariamente o consumo de energia utilizando um sistema de baixo custo?

Então, o objetivo geral deste projeto é construir um sistema simples de baixo custo capaz de monitorar o consumo discriminado de cargas, visando fornecer informações para proporcionar a economia de energia. Com os dados de consumo, além de identificar os dispositivos com consumo elevado de energia elétrica, pode-se tomar algumas medidas de eficiência energética.

1.2. CENÁRIO

O projeto pode ser aplicado em residências, condomínios, centro comercial, indústrias ou qualquer outro meio em que se vê a necessidade de aferir o consumo de energia de forma detalhada e com baixo custo.

1.3 PROBLEMA

O consumo energético no Brasil vem se destacando por ser um dos temas mais debatidos na atualidade. Isto se faz devido a vários fatores como o consumo irracional por parte da população, gerando uma alta demanda na produção e o constante aumento do custo desta energia no país. O consumidor brasileiro só pode acompanhar o consumo de energia elétrica de seus equipamentos através dos medidores eletrônicos ou digitais fornecidos pela concessionária. Os medidores apresentam muitas informações técnicas desconhecidas pelo consumidor, o que dificulta na interação entre o consumidor e o dispositivo.

Como monitorar diariamente o consumo de energia utilizando um sistema simples de baixo custo e passar essas informações de uma forma clara ao consumidor para então gerenciar as cargas e consequentemente proporcionar uma maior eficiência energética?

1.4 JUSTIFICATIVA

Nos últimos anos, devido aos problemas de disponibilidade de energia elétrica, as questões referentes ao uso eficiente da energia alcançaram ainda mais destaque.

A eficiência energética cada vez mais se ressalta no cenário brasileiro nos dias de hoje, devido a sua grande importância na diminuição do consumo de energia elétrica, reduzindo os impactos ambientais e sociais provenientes da carência de novos projetos para produção e geração de energia. A implantação de ações de eficiência energética busca a obtenção do menor índice de custo em relação à energia utilizada em seu processo. Podemos destacar que para melhorar a eficiência energética, faz-se necessário o monitoramento, gerenciamento e o levantamento dos níveis de eficiência energética encontrados nos estabelecimentos com a utilização da automação como um aliado.

Para a medição do consumo mensal de energia elétrica são utilizados equipamentos de medição totalizadores de consumo, que registram a diferença do consumo total do mês anterior e o mês corrente, obtendo, assim, o consumo atribuído ao mês corrente, medido em kWh (LAPA et al., 2010).

Para produzir mais com menos energia, grandes empresas usam e abusam dos avanços tecnológicos, a fim de terem um gerenciamento mais eficiente de sua utilização. Novidades como *Cloud computing*, *Big Data*, *Analytics*, *Inteligência Artificial* e *Internet das Coisas (IoT)* começaram a ser aplicadas no monitoramento e controle dos principais pontos de consumo energético nas organizações (AALOK, 2019).

A automação é capaz de executar comandos, obter dados, regular parâmetros e controlar funções automaticamente, sem a intervenção humana. Segundo Pinheiro (2004), através da automação é possível realizar a função mais simples até a mais complexa, ou seja, a integração permite que um dispositivo seja controlado de modo inteligente, tanto individualmente quanto em conjunto, visando um maior conforto, informação e segurança.

Considerada a etapa mais avançada, as técnicas de inteligência artificial, como aprendizado de máquina, podem ainda ser aplicadas para a identificação de padrões de consumo, detecção de falhas e previsão de consumo que destaquem oportunidades de economia de energia. Além

disso, o sistema pode ser integrado a uma solução mais completa de automação, enviando comandos e parâmetros para controlar a operação da empresa de forma automática (AALOK, 2019).

O equipamento proposto neste presente trabalho irá permitir ao usuário monitorar os equipamentos ligados utilizando a técnica de aprendizado de máquina aliada a automação, com um custo ínfimo de produção obtendo as informações sobre potência consumida dividida individualmente pelos equipamentos ligados.

1.5 OBJETIVOS

1.5.1 Objetivo Geral

Construir um sistema simples de baixo custo capaz de monitorar diariamente o consumo de energia por padrão de cargas utilizando aprendizado de máquina e passar essas informações de uma forma clara ao consumidor para então proporcionar a economia de energia, seja reduzindo o tempo de funcionamento de um determinado dispositivo em uso excessivo, ou até mesmo substituindo um equipamento antigo por um mais moderno e econômico.

1.5.2 Objetivos Específicos

- Montar um quadro com cargas e equipamentos necessários para a aferição e controle do sistema (transformador de corrente, microcontrolador);
- Desenvolvimento do firmware para programação do microcontrolador;
- Desenvolvimento do algoritmo responsável para controle do sistema;
- Analisar o funcionamento do sistema independente e coletar os dados de consumo de energia;
- Realização de um experimento em laboratório para atestar a viabilidade econômica do sistema;
- Tratamento dos dados e avaliação dos resultados obtidos;

1.6 METODOLOGIA

Segundo Gil (2011), este trabalho possui natureza de pesquisa aplicada na qual envolve a aplicação prática da ciência na solução de problemas específicos. Para a elaboração deste trabalho, foram consultados diversos trabalhos científicos, entre eles: livros, artigos, dissertações, monografias e consultas a sites relacionados com o tema.

Após a realização das consultas, será realizado um experimento em laboratório para verificar viabilidade do sistema e possíveis impedimentos de aplicação.

Os procedimentos para a realização deste trabalho são:

- Desenvolvimento do referencial teórico para aumentar os conhecimentos sobre o objeto de estudo;
- Conceituar eficiência energética;
- Conceituar aprendizado de máquina;
- Pesquisa literária sobre o microcontrolador *open source* Esp32 e verificar suas possibilidades de utilização conforme os objetivos propostos;
- Pesquisa bibliográfica para analisar o estado da arte da automação;
- Desenvolvimento de um protótipo para atestar a viabilidade do sistema e alcançar os resultados pretendidos;

CAPÍTULO 2 - REFERENCIAL TEÓRICO

2.1 MICROCONTROLADOR

O microcontrolador é um circuito integrado compacto projetado para governar uma operação específica em um sistema incorporado. Um microcontrolador típico possui um processador, memória e portas de entrada e saída (I/O) em um único chip.

Os microcontroladores são encontrados em veículos, robôs, máquinas de escritório, dispositivos médicos, transmissores RF, máquinas de cartão de crédito, entre outros dispositivos. Eles são praticamente uma miniatura de um computador pessoal projetados para controlar pequenos recursos de um componente dedicado sem um sistema operacional complexo.

Um microcontrolador é incorporado dentro de um sistema para controlar uma função singular em um dispositivo. Ele faz isso interpretando os dados que recebe em suas portas de entrada e saída (I/O) usando seu processador central. As informações temporárias que o microcontrolador recebe são armazenadas em sua memória de dados, onde o processador as acessa e usa instruções armazenadas em sua memória de programa para decifrar e aplicar os dados recebidos. Em seguida, ele usa suas portas de entrada e saída (I/O) para se comunicar e executar a ação apropriada.

Microcontroladores são usados em uma ampla variedade de sistemas e dispositivos. Os dispositivos geralmente utilizam vários microcontroladores que trabalham juntos no sistema para lidar com suas respectivas tarefas. Por exemplo, um carro pode ter muitos microcontroladores que controlam vários sistemas individuais, como o sistema de travagem autobloqueio, controle de tração, injeção de combustível ou controle de suspensão. Todos os microcontroladores se comunicam para informar as ações corretas. Alguns podem se comunicar com um computador central mais complexo dentro do carro, enquanto outros podem se comunicar apenas com outros microcontroladores. Eles enviam e recebem dados usando suas portas de entrada e saída (I/O) e processam esses dados para executar as tarefas designadas.

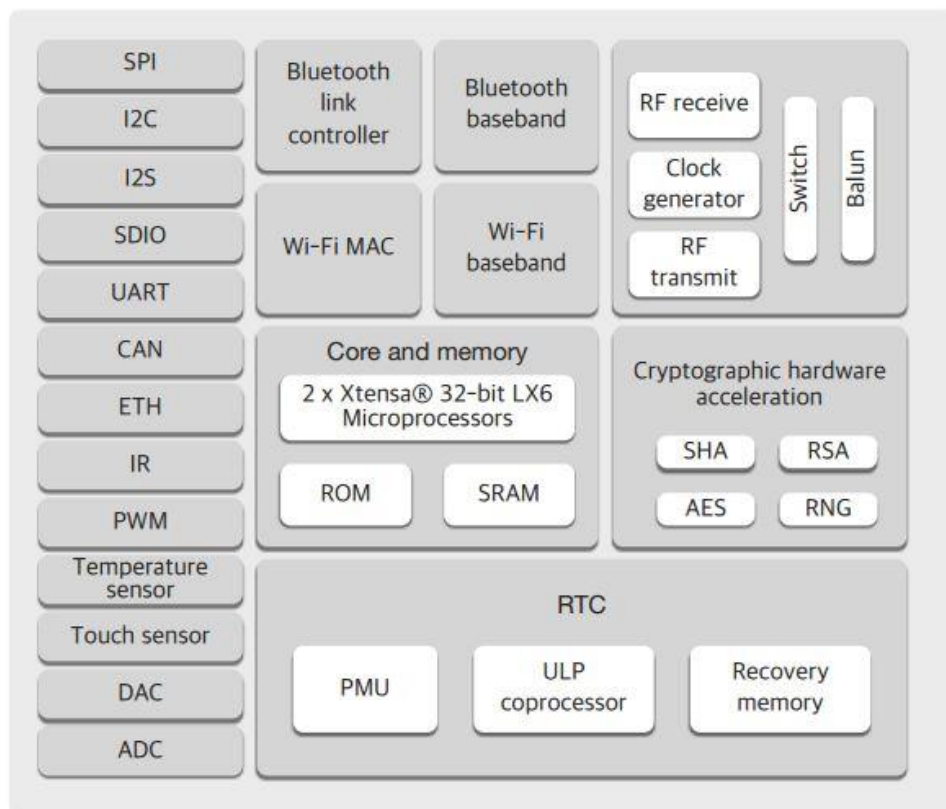
2.1.1 ESP32

O Esp32 é um microcontrolador *open source* para desenvolvimento, foi criado com o objetivo de oferecer fácil comunicação entre *software* mais *hardware* e o usuário.

Este microcontrolador de baixo custo foi criado pela empresa *Espressif Systems* e foi lançado recentemente no mercado onde já vem sendo considerado como um dos mais robustos e notórios controladores da atualidade, tendo fortes características sua velocidade de processamento, acessibilidade e conectividade, evidenciando essa última principalmente pela sua integração com os módulos *Wi-Fi* e *bluetooth* internamente junto ao processador (SANTOS e JUNIOR, 2019, apud KOLBAN, 2017).

O ESP32 é constituído por um processador robusto que foi projetado para ser *single* ou *dual-core* de 32-bit (com dois núcleos físicos de processamento) que pode chegar a trabalhar com frequências de *clock* de 80MHz até 240 MHz. A Figura 1 apresenta em forma de diagrama de blocos a estrutura do ESP32.

Figura 1 – Arquitetura do ESP32



Fonte: SANTOS (2019)

Além de contar com uma vantagem enorme com relação a sua capacidade de armazenamento sendo exponencialmente maior se comparada com a dos já consagrados microcontroladores Arduino, podendo chegar a ser o dobro. Tomando como base sua memória flash de 4MB comparada com o modelo ATmega 2560 de 256KB.

Tabela 1 – ESP32 x Arduino

Descrição	Arduino UNO	Arduino MEGA 2560	ESP32
Alimentação	5V	5V	2,2V ~ 3,3V DC
Bluetooth	Não possui	Não possui	Bluetooth Low Energy v4.2 (BLE)
Wi-Fi	Somente com Shields	Somente com Shields	2,4 GHz, 802.11 b/g/n/e/i (802.11n até 150 Mbps)
Conversores ADC (Analogico para Digital)	6 ADC com 10-bit de resolução (1024 bits)	16 ADC com 10-bit de resolução (1024 bits)	18 ADC com 12-bit de resolução (4096 bits)
Conversores DAC (Digital para Analógico)	Nenhum	Nenhum	2 DAC com 8-bit de resolução (256 bits)
Corrente de Consumo	Média de 15mA	Média de 70mA	Média de 80mA
Frequência de Operação	0 ~ 16 MHz	0 ~ 16 MHz	80MHz ~ 240MHz
Memória FLASH	32KB	256 KB	4MB
Memória RAM/SRAM	2KB	8 KB	520KB
Memória ROM/EEPROM	4KB	4 KB	448KB
Pinos de I/O	23 pinos com 6 PWM	54 pinos com 14 PWM	34 pinos com 16 PWM
Processador	AVR® 8-bit RISC	ATmega2560 RISC com até 16 MIPS	Xtensa® Dual-Core 32-bit LX6
Preço Médio (Modelo Original)	U\$22,00 (Exterior) ou R\$69,99 (Brasil)	U\$38.50 (Exterior) ou R\$99,99 (Brasil)	U\$14,95 (Exterior) ou R\$38,90 (Brasil)

Fonte: Autoria própria

Na Tabela 1 podemos observar algumas especificações do microcontrolador ESP32, comprovando sua superioridade em relação ao Arduino.

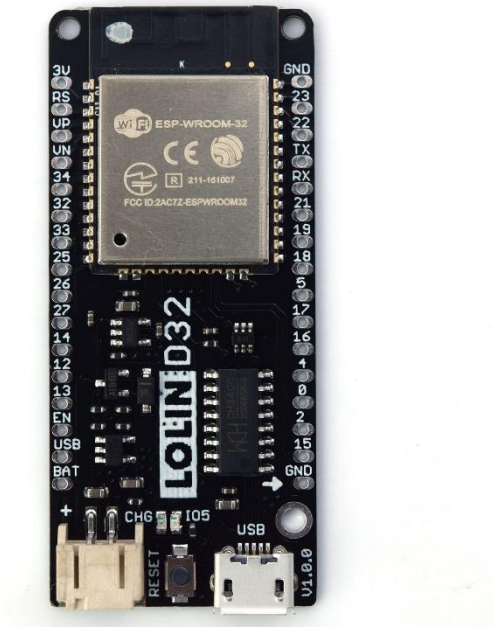
Como podemos ver, o Esp32 se torna superior na maioria das características citadas na tabela anterior e com um preço muito reduzido em relação aos concorrentes.

Diferente de alguns fabricantes de microcontroladores, a Espressif (desenvolvedora do ESP32) não disponibiliza nenhuma IDE de desenvolvimento junto com a distribuição do conjunto de programas (*toolchains – SDK-IDF*) para o ESP32. (SIMPLICIO, 2018)

A fácil programação do ESP32 é factível através dos diversos softwares compatíveis, evidenciando-se a linguagem de programação C/C++, que pode ser desenvolvida através do programa *Software Development Kit* (SDK) fornecido pela própria desenvolvedora do microcontrolador ou então pelo mais célebre, Arduino IDE, ademais por apresentar uma estruturada conectividade sem fio é possível realizar a programação remota, utilizando por exemplo o *Wi-Fi*. (SANTOS e JUNIOR, 2019)

Na Figura 2 encontra-se um módulo com o Esp32 utilizado neste projeto, modelo *LOLIN32* da empresa *WEMOS*.

Figura 2 – WEMOS LOLIN32



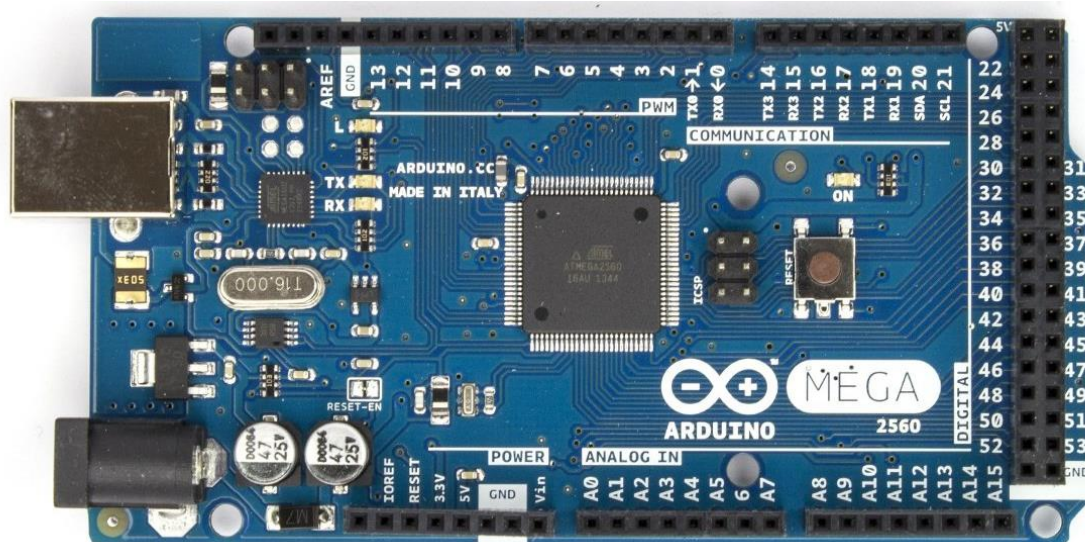
Fonte: WEMOS (2020)

2.1.2 ARDUINO

O Arduino é uma plataforma *open source* (código aberto) muito utilizado atualmente na construção de projetos eletrônicos. O Arduino consiste em um módulo no qual se tem integrado um microcontrolador Atmega acoplado junto com reguladores e chips de conversão serial para realizar a comunicação entre o módulo e o seu Ambiente de Desenvolvimento Integrado chamado *IDE* que pode ser executado em qualquer computador usado para escrever e fazer *upload* do código produzido para o módulo.

A plataforma Arduino tornou-se bastante popular entre as pessoas que estão começando no setor eletrônico e por boas razões. Diferentemente da maioria das placas de circuito programáveis antigas, o Arduino não precisa de um *hardware* separado (programador) para carregar novo código na placa, você simplesmente pode usar um cabo. Além disso, o Arduino *IDE* usa uma versão simplificada do C++, facilitando assim o aprendizado do programa. É apresentado na Figura 3 a placa do Arduino Mega uma das placas mais poderosas atualmente.

Figura 3 - Arduino Mega

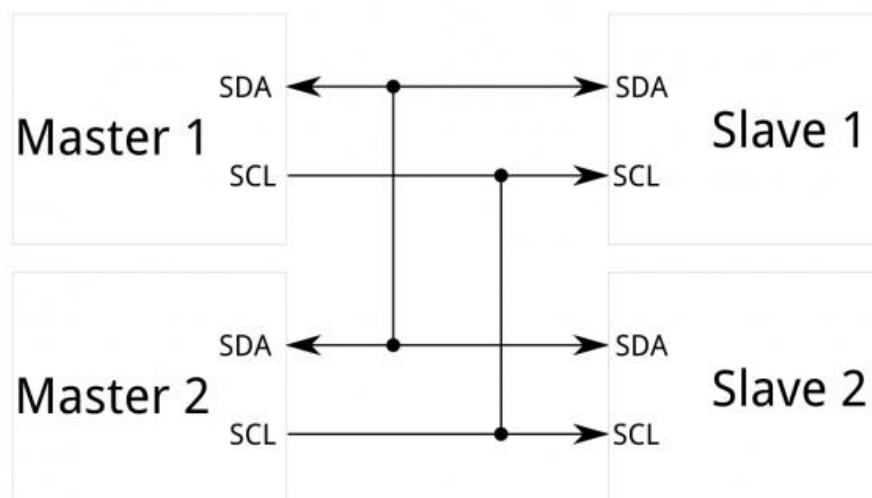


Fonte: Mlstatic (2020)

2.1.3 PROTOCOLO I2C

O protocolo I2C é um protocolo destinado a permitir que múltiplos circuitos integrados digitais “escravos” (*slave*) se comuniquem com um ou mais chips “mestre” (*master*). Criado originalmente em 1982 pela Philips para ser utilizado entre chips da própria empresa. A especificação original permitia apenas comunicação a 100KHz e fornecia apenas endereços de 7bits limitando o número de dispositivos no barramento para 112. Em 1992, a primeira especificação pública foi publicada, adicionando um modo rápido de 400KHz e um espaço de endereçamento expandido de 10bits. A vantagem de usar I2C é que requer apenas dois fios para comunicação e estes podem suportar até 1008 dispositivos escravos. Além disso o I2C pode suportar um sistema multimestre permitindo que mais de um mestre se comunique com todos os dispositivos no barramento (embora os dispositivos “mestre” não possam se comunicar pelo barramento e se revezem). Hoje o protocolo I2C pode se comunicar a uma velocidade de até 3.4MHz. Na maior parte dos microcontroladores atuais o I2C consiste em dois sinais: SCL e SDA. O SCL é o sinal do relógio (*clock*) e o SDA é o sinal de dados. O sinal do *clock* é sempre gerado pelo atual mestre do barramento, mas alguns dispositivos escravos podem forçar o *clock* algumas vezes para atrasar o envio de mais dados pelo mestre, isso é chamado de “alongamento do *clock*”. Na Figura 4 temos um diagrama de ligação do protocolo I2C:

Figura 4 - Protocolo I2C



Fonte: Robocore (2020)

2.2 PROGRAMAÇÃO

Uma linguagem de programação é um vocabulário e um conjunto de regras gramaticais para instruir um computador ou dispositivo de computação para executar tarefas específicas. O termo linguagem de programação geralmente se refere a linguagens de alto nível, como BASIC, C, C++, Python, COBOL, Java, FORTRAN, Ada e Pascal.

Cada linguagem de programação possui um conjunto exclusivo de palavras-chave (palavras que entende) e uma sintaxe especial para organizar as instruções do programa. (SANTOS W., 2018)

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 Python que de uns anos para cá sua popularidade explodiu.

2.2.1 LINGUAGEM DE PROGRAMAÇÃO C / C++

O C++ foi inicialmente desenvolvido por Bjarne Stroustrup da *Bell Labs* durante a década de 1980 (originalmente com o nome *C with Classes*, como um adicional à linguagem C) com o objetivo de melhorar a linguagem de programação C ainda que mantendo máxima compatibilidade. (GARBADE, 2017)

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.

2.2.2 LINGUAGEM DE PROGRAMAÇÃO PYTHON

Python não é uma linguagem nova. Foi criada em 1991, mas somente de uns anos para cá é que sua popularidade explodiu. Um dos motivos é o aumento do desenvolvimento de aplicações web, pois Python possui um framework poderoso e robusto que permite um desenvolvimento rápido de websites.

O principal objetivo da linguagem Python é estimular produtividade e legibilidade de códigos. Ela é conhecida por sua velocidade e pelo seu estilo compacto de escrever. Com poucas linhas de código é possível criar muitas funcionalidades.

A linguagem faz uso de baixa quantidade de palavras reservadas no código, uso de indentação para marcar blocos de comandos e possui um coletor de lixo automático que gerencia a memória para os programadores. Todas as suas funcionalidades são desenvolvidas para facilitar a vida do desenvolvedor. (IMPACTA, 2018)

2.3 APRENDIZADO DE MÁQUINA (*MACHINE LEARNING*)

Machine Learning (Figura 5) do português aprendizado de máquina, é um tipo de inteligência artificial que permite aos computadores tomarem decisões com ajuda de algoritmos que reconhecem padrões e se tornam capazes de fazer previsões.

Sua principal característica, porém, é não precisar ter as rotinas implantadas a mão: o próprio sistema tem a habilidade de aprender com a análise de dados e executar tarefa com uma precisão cada vez maior. Um bom exemplo é o filtro de spams dos e-mails. Sem a necessidade de configuração, ele se aprimora de forma automática e, com o tempo, se torna cada vez mais eficiente para bloquear mensagens indesejadas na caixa de entrada. Para ser um sistema inteligente, com capacidade de aprender com o histórico de dados, o aprendizado de máquina utiliza muitos algoritmos e métodos diferenciados para alcançar os objetivos do sistema (TIAGO, 2018).

Figura 5 - Aprendizado de Máquina



Fonte: Verhaert (2020)

Uma das bases para o funcionamento do *Machine Learning* são os algoritmos que aprendem por meio de exemplos. Os dois modos mais conhecidos de aprendizagem de algoritmos são a supervisionada e a não supervisionada.

Na modalidade supervisionada, o algoritmo aprende recebendo dados que contêm a resposta certa. Ou seja, há uma relação entre a entrada e a saída e ela é usada para treinar o algoritmo. Por exemplo, na contratação de um empréstimo realizada em um banco, os dados analisados serão do histórico de crédito do cliente. As informações que entram no sistema já estão rotuladas como positivos ou negativos para a concessão do crédito. Já na não supervisionada, os dados recebidos não carregam rótulos, ou seja, os efeitos das variáveis não são previstos. Com isso, os resultados estão relacionados com os padrões encontrados nos dados. Na busca em um banco de artigos científicos, por exemplo, eles são agrupados de acordo com determinadas variáveis (número de páginas, frequência de palavras e outros). O filtro se aprimora conforme o sistema encontra novos padrões entre os artigos selecionados (TIAGO, 2018).

2.3.1 ALGORITMO - VIZINHOS MAIS PRÓXIMOS (*K-NEAREST NEIGHBORS*)

KNN (*K — Nearest Neighbors*) é um dos muitos algoritmos (de aprendizagem supervisionada) usado no campo de *data mining* e *machine learning*, ele é um classificador onde o aprendizado é baseado “no quão similar” é um dado (ou um vetor) do outro. O treinamento é formado por vetores de n dimensões (JOSÉ, 2018).

O KNN captura a ideia de similaridade (às vezes chamada de distância ou proximidade) como o desenvolvimento matemático aprendida na infância dos docentes - calculando a distância entre os pontos de um gráfico. Existem várias maneiras de calcular a distância, e uma delas pode ser preferível, dependendo do problema que estamos resolvendo. No entanto, a distância em linha reta (também chamada de distância euclidiana) é uma escolha popular e familiar.

A seguir, temos as etapas de um algoritmo KNN:

1. Recebe os dados não classificados;
2. Mede a distância (Euclidiana, Manhattan, Minkowski ou Ponderada) do novo; dado com todos os outros dados que já estão classificados;
3. Obtém as X (no caso essa variável X é o parâmetro K) menores distâncias;
4. Verifica a classe de cada um dos dados que tiveram a menor distância e conta a quantidade de cada classe que aparece;

5. Toma como resultado a classe que mais apareceu dentre os dados que tiveram as menores distâncias;
6. Classifica o novo dado com a classe tomada como resultado da classificação;

Para calcular a distância entre dois pontos (sua nova amostra e todos os outros dados que temos no banco de dados) é muito simples, como dito anteriormente, há várias formas de obter esse valor, neste projeto utiliza-se a distância euclidiana, pois é uma das mais usadas.

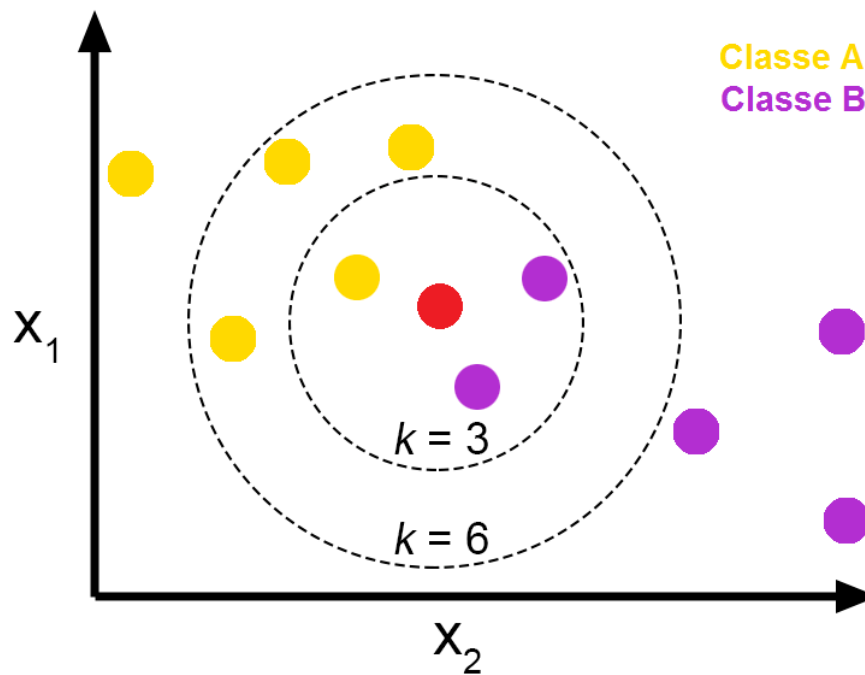
A fórmula da distância euclidiana se encontra na equação 1.

$$d = \sqrt{\sum_{k=1}^n (p_{ik} - p_{jk})^2} \quad (1)$$

É com essa fórmula que vamos verificar a distância entre 1 ponto (sua amostra não classificada) e 1 outro ponto do nosso banco de dados (1 outro dado já classificado) para então ver a similaridade dos dois, quanto menor for o resultado da expressão, maior será a similaridade entre esses dois dados. Deseja-se rodar estes cálculos várias vezes, até ter rodados em todos os dados. Feito isso têm-se uma matriz. Com o resultado de cada cálculo da distância (no caso, a similaridade) do dado não classificado para um outro dado já classificado.

Logo abaixo, há uma imagem com esse processo (Figura 6). Um dado não classificado (em vermelho) e todos os seus outros dados já classificados (amarelo e roxo) cada um com sua classe (A e B). Então calcula-se a distância do seu novo dado com todos os outros pra saber quais estão mais próximos (que possuem as menores distâncias), feito isso tomando-se 3 (ou 6) dos dados mais próximos se verifica qual é a classe mais próxima, no caso da Figura 6, os dados mais próximos do novo dado são aqueles que estão dentro do primeiro círculo (de dentro pra fora), e ali dentro há 3 outros dados (já classificados), com isso pode-se verificar qual é a classe predominante ali dentro. Observando a Figura 6, pode-se concluir que é o roxo, pois há 2 bolinhas roxas e apenas 1 amarela, logo esse novo dado que antes não estava classificado, agora ele será classificado como roxo.

Figura 6 – Classificação de dados por KNN



Fonte: José (2018)

2.3.2 MATRIZ DE CONFUSÃO

A matriz de confusão é um resumo dos resultados da previsão em um problema de classificação do aprendizado de máquina, ou seja, a matriz de confusão mostra as maneiras pelas quais seu modelo de classificação é confuso quando faz previsões. O número de previsões corretas e incorretas é resumido com valores de contagem e dividido por cada classe. É uma tabela com 4 ou mais combinações diferentes de valores previstos e reais que mostra as frequências de classificação para cada classe do modelo.

Estas frequências podem ser classificadas como:

- Verdadeiro Positivo (TP): ocorre quando no conjunto real, a classe que estamos buscando foi prevista corretamente. Por exemplo, quando a mulher está grávida e o modelo previu corretamente que ela está grávida.
- Verdadeiro Negativo (TN): ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista corretamente. Exemplo: a mulher não estava grávida, e o modelo previu corretamente que ela não está.

- Falso Positivo (FP): ocorre quando no conjunto real, a classe que estamos buscando prever foi prevista incorretamente. Exemplo: a mulher não está grávida, mas o modelo disse que ela está.
- Falso Negativo (FN): ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente. Por exemplo, quando a mulher está grávida e o modelo previu incorretamente que ela não está grávida.

Tabela 2 – Exemplo de uma matriz de confusão

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Rodrigues (2019)

Na Tabela 2 pode-se observar um exemplo de uma matriz de confusão.

Ao ser feita a contagem de todos esses termos e obter a matriz de confusão, é possível calcular métricas de avaliação para a classificação. Essas métricas podem ser divididas entre:

- Acurácia: indica uma performance geral do modelo. Dentre todas as classificações, quantas o modelo classificou corretamente; Demonstra-se o cálculo da acurácia a partir da equação 2.

$$Acurácia = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

- Precisão: dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas;

Demonstra-se o cálculo da precisão a partir da equação 3.

$$Precisão = \frac{TP}{TP+FP} \quad (3)$$

- Sensibilidade (*Recall*): dentre todas as situações de classe Positivo como valor esperado, quantas estão corretas;

Demonstra-se o cálculo da sensibilidade a partir da equação 4.

$$Sensibilidade = \frac{TP}{TP+FN} \quad (4)$$

- *F1-Score*: média harmônica entre precisão e recall.

Demonstra-se o cálculo da *F1-Score* a partir da equação 5.

$$F1 - Score = \frac{2*Precisão*Sensibilidade}{Precisão+Sensibilidade} \quad (5)$$

2.4 TRANSFORMADORES DE CORRENTE

O transformador de corrente é basicamente utilizado para medição, monitoramento e proteção de equipamentos. Segatto e Coury (2006, p. 425) definem que a função do transformador de corrente (TC) é “transformar as correntes oriundas do sistema de potência em valores de baixa magnitude e fornecer isolação galvânica entre a rede elétrica e os relés e outros instrumentos conectados ao enrolamento secundário do TC”.

Existem 2 tipos de sensores de corrente, os intrusivos e os não-intrusivos. No método intrusivo o sensor é conectado diretamente a alimentação. No método não-intrusivo a conexão direta a alimentação é evitada.

O sensor de corrente que será utilizado no projeto é do tipo não-intrusivo, o qual se adapta melhor ao projeto devido a sua facilidade de instalação. Seu método de funcionamento é como um alicate amperímetro onde o núcleo envolve um fio que

conduz a corrente elétrica. A Figura 7 ilustra o sensor PZCT-02 a ser utilizado, ele trabalha com corrente de entrada entre 0A e 100A e corrente de saída entre 0A e 100mA.

Figura 7 – Sensor de corrente não intrusivo



Fonte: Solear (2020)

2.4.1 RESISTOR SHUNT OU *BURDEN*

Como o microcontrolador (*ESP32*) faz leitura de tensão em suas portas analógicas, é necessário ligar um resistor em paralelo aos terminais de saída do sensor de corrente. Este resistor é chamado de resistor *BURDEN* (Resistor de carga).

O *ESP32* realiza a leitura de tensão entre 0V e 3,3V, então devemos escolher um resistor que forneça no máximo um valor de 3,3v na saída. Para isso deve-se calcular a resistência de carga.

Sabe-se que o sensor mede corrente alternada máxima de 100A AC. Esse valor de 100A é o valor RMS, que também é chamado de valor eficaz. O valor RMS é calculado pela equação 6:

$$I(RMS) = \frac{I_{\text{máx}}(\text{corrente de pico})}{\sqrt{2}} \quad (6)$$

A corrente de pico máxima medida é dada por:

$$I_{\text{máx}}(\text{corrente de pico}) = \sqrt{2} * I(\text{RMS}) \gg \sqrt{2} * 100 = 141,4A$$

Se uma corrente de 100A na entrada produz uma corrente de 100mA na saída, basta usar a relação de transformação para encontrar o número de espiras no secundário. Pode-se destacar o cálculo da relação de transformação na equação 7:

$$\frac{N_1}{N_2} = \frac{I_2}{I_1} \gg \frac{1}{N_2} = \frac{100mA}{100A} \gg N_2 = 1000 \quad (7)$$

Onde:

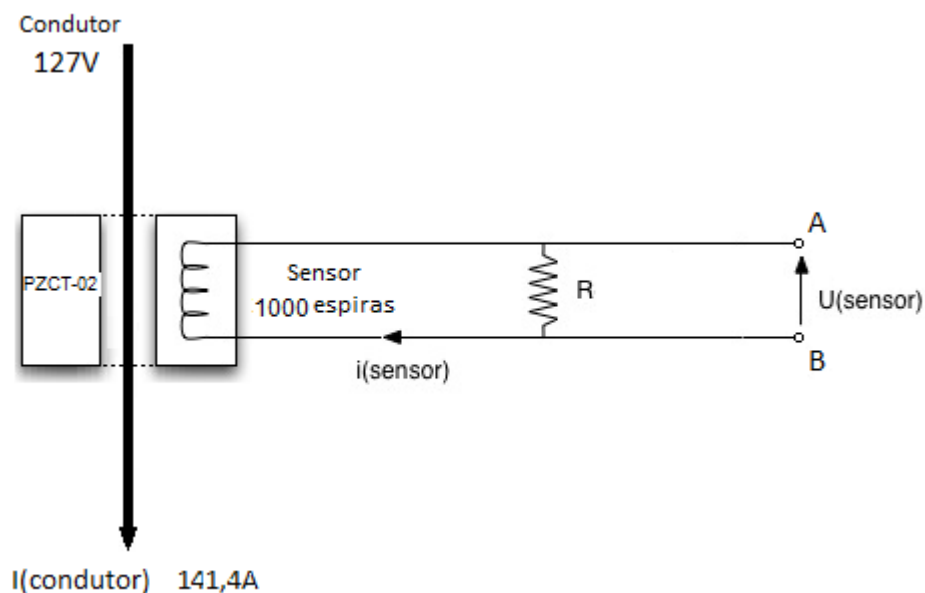
N1 = número de espiras no primário (neste caso = 1)

N2 = número de espiras no secundário

I1 = corrente no primário (corrente a ser medida)

I2 = corrente no secundário

Figura 8 – Esquema resistor de BURDEN



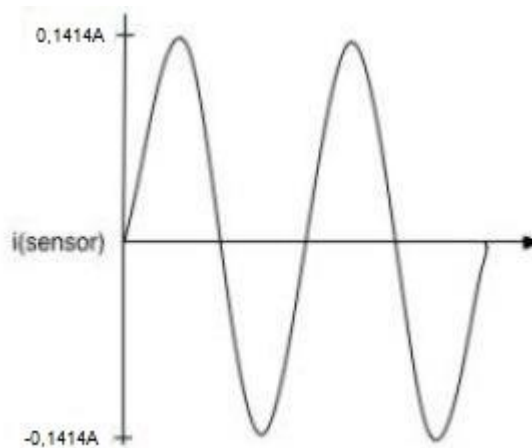
Fonte: Autoria própria

A corrente de saída do sensor é inversamente proporcional ao número de espiras. Pode-se destacar na equação 8 o cálculo da corrente no secundário do sensor:

$$I(sensor) = \frac{I(medido)}{N2} \gg \frac{141,4}{1000} = 0,1414 \quad (8)$$

Têm-se um sinal da corrente de saída do sensor conforme Figura 9.

Figura 9 – Sinal da corrente no secundário do sensor de corrente não intrusivo



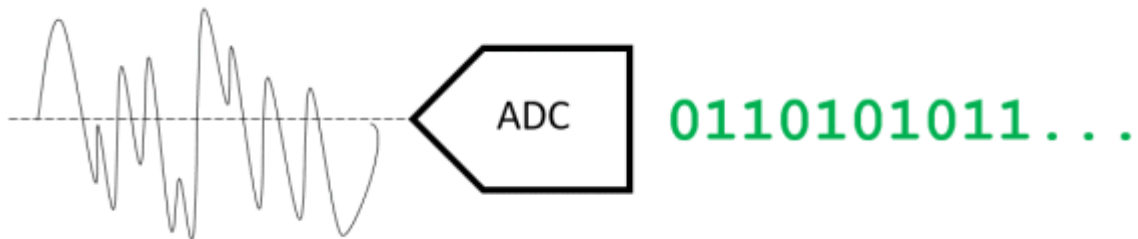
Fonte: Autoria própria

Sinal de saída do PZCT-02 para um valor de corrente eficaz de 100A.

2.5 CONVERSOR A/D

O principal objetivo dos conversores A/D (Figura 10) dentro de um sistema de aquisição de dados é converter sinais analógicos condicionados em um fluxo de dados digitais para que o sistema de aquisição de dados possa processá-los para exibição, armazenamento e análise.

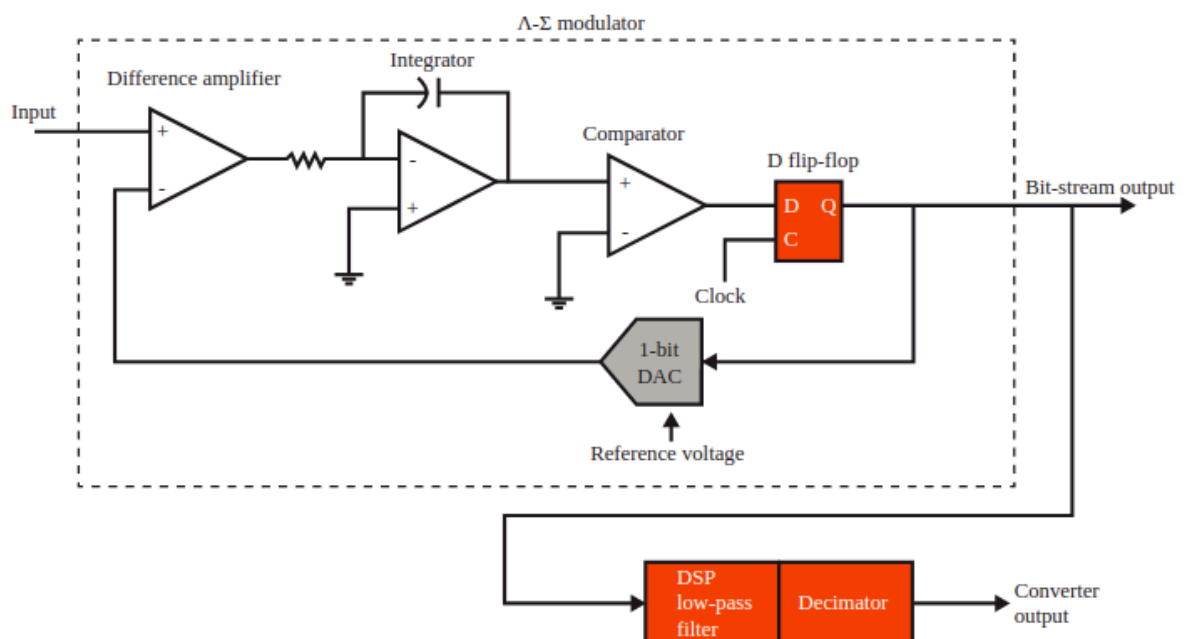
Figura 10 - Conversor A/D



Fonte: Luiz Bittencourt (2016)

O A/D utilizado no projeto é do tipo *delta-sigma*. Este tipo de A/D possui um *design* complexo e poderoso os tornando ideais para aplicações dinâmicas que requerem a maior resolução possível do eixo de amplitude. É por isso que eles são comumente encontrados em áudio, som, vibração e uma ampla variedade de projetos de aquisição de dados da última geração. Na Figura 11 é exibido um diagrama de blocos de um *delta-sigma*:

Figura 11 - Conversor A/D tipo *delta-sigma*



Fonte: Gstatic (2020)

Um filtro passa-baixa implementado em um *delta-sigma* elimina virtualmente o ruído de quantização, resultando em um excelente desempenho sinal-ruído.

A implementação destes chips para projetos que possuem aquisição de dados geralmente envolve uma forte concentração de filtro *anti-aliasing*, tornando praticamente impossível digitalizar sinais falsos.

CAPÍTULO 3 – ESTADO DA ARTE

Neste capítulo serão abordados trabalhos e pesquisas de outros autores que possuem relação com o tema deste trabalho acadêmico. A pesquisa bibliográfica é importante para nos proporcionar fontes para o embasamento teórico do assunto a ser tratado e nos ajudar a responder questões referentes aos nossos objetivos.

3.1 FORNAZARI, BORGES (2016)

Este trabalho de conclusão de curso de alunos da Universidade Tecnológica Federal do Paraná tem como objetivo propor um sistema de monitoramento e controle de consumo de energia nas salas de aula da Universidade usando CLP e sistema supervisório buscando a eficiência energética.

Na construção do sistema proposto, foi utilizado um CLP, um relé de interface para proteção de entradas e saídas do CLP, um sensor de corrente e dois sensores de presença infravermelho para detecção de fontes de calor presentes no ambiente (pessoas). Todos estes componentes aliados a um sistema supervisório.

O sistema desenvolvido neste trabalho, apresentou resultados satisfatórios para o objetivo de monitorar e controlar os circuitos elétricos. Mas, por outro lado, verificou-se que durante os intervalos de aula onde o ambiente ficava praticamente desocupado, com poucos alunos, o sistema não foi capaz de identificá-los quando os mesmos ficavam em regiões de pouco alcance dos sensores, fundo da sala, pois a área de detecção dos sensores priorizava o espaço do professor visando identifica-lo durante a aula. Um aumento da cobertura da área da sala por mais sensores, resolveria o problema.

Como trabalho futuro, propuseram outras formas de controle, não apenas a ocupação das salas de aula, mas também o uso da iluminação natural e a viabilidade da troca das lâmpadas fluorescentes por lâmpadas LED.

3.2 QUIRINO T., QUIRINO J., AMARAL, NEPOMUCENO e SILVA (2019)

O artigo apresentado pelos autores na Revista Mundi Engenharia tem o objetivo apresentar a pesquisa e desenvolvimento de um medidor de vazão termal

não intrusivo e inteligente para baixas vazões de ar utilizando o princípio da distribuição de temperaturas na parede de um tubo.

Os pesquisadores optaram por escolher as Redes Neurais Artificiais como método para inferir a relação entre a variação térmica na parede de um tubo e a vazão, pois as Redes Neurais Artificiais possuem a capacidade de compensar erros sistemáticos e realizar aproximação universal de funções por reconhecer padrões entre variáveis mesmo havendo um bando de dados incompleto, ambíguo e parcialmente incorreto.

A Rede Neural utilizada foi a rede MLP (*Multi Layer Perceptron*) que é uma rede neural com uma ou mais camadas ocultas com um número indeterminado de neurônios.

Para que a rede tenha uma boa capacidade de generalização, utilizou-se o treinamento com validação cruzada no qual os dados são divididos em conjunto de treinamento e teste, onde foi implementada as temperaturas do conjunto de treinamento na Rede Neural Artificial em relação as respectivas vazões, para que a RNA aprenda o padrão entre as temperaturas medidas e a vazão. Posteriormente as temperaturas dos conjuntos de dados de teste são implementadas a rede para determinação do erro, ou seja, com dados não utilizados anteriormente.

Foi concluído que a implementação de sistemas inteligentes, em especial as redes neurais artificiais na instrumentação, colabora com novas possibilidades de configuração de sensores pois permitem a modelagem empírica do sistema considerando e corrigindo erros e incertezas quando treinadas.

3.3 MELO, CAMPOS e SILVA (2019)

Este trabalho de conclusão de curso de alunos da Universidade Estácio de Sá teve como objetivo a criação de um dispositivo capaz de monitorar o consumo de energia elétrica e oferecer ao consumidor final a possibilidade de controle e economia em seu consumo de energia elétrica residencial através de um dispositivo de monitoramento acoplado a rede elétrica, porém não intrusivo e que utiliza a tecnologia Arduino.

O sistema desenvolvido pelos alunos possuía sensores de corrente, Arduino Mega, Shield ESP8266. Todo o *hardware* foi utilizado em conjunto com o aplicativo *Blynk* para apresentação dos dados ao consumidor através de aplicativo.

Os autores concluíram que todos os objetivos do sistema automatizado foram satisfatoriamente atendidos. Chegaram também a conclusão de que a implementação do aplicativo *Blynk* interagindo com o sistema proposto, proporcionou maior conhecimento e conforto ao consumidor, despertando sua curiosidade sobre o quanto está sendo consumido em sua residência e consequentemente sua conscientização.

Ficou sugerido pelos autores que fossem elaborados futuramente uma otimização do sistema proposto para que ocupe menos espaço possível e implementação de mais funções ao dispositivo com o propósito de fornecer ao consumidor um monitoramento mais detalhado do seu consumo.

CAPÍTULO 4 – DESENVOLVIMENTO

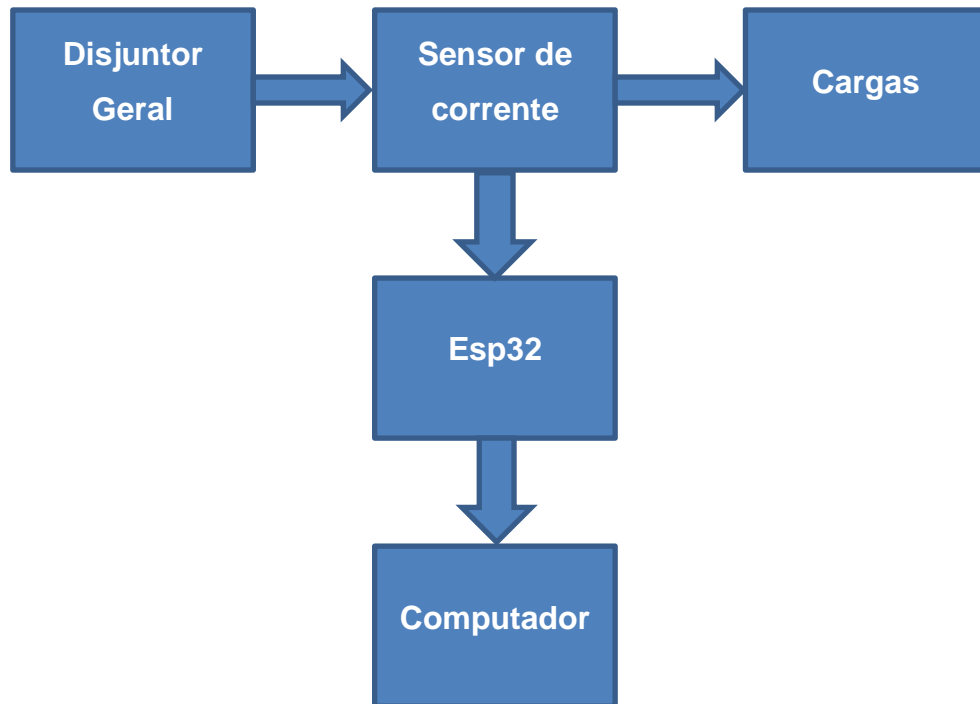
4.1 IMPLEMENTAÇÃO DO SISTEMA DESENVOLVIDO

A implementação do sistema de baixo custo para monitoramento do consumo de energia foi realizada através do *Lolin32*, pois este módulo microcontrolador possui as melhores características que satisfazem as necessidades da prática a ser elaborada.

Para efeito de teste, este sistema foi implementado em um quadro de energia de autoria própria com cargas ligadas na rede monofásica. O sistema é apresentado através de uma interface ao usuário por um programa criado utilizando a linguagem *Python* onde apresenta qual carga estará ligada e seu consumo atual.

A confecção do sistema se divide em algumas partes, uma delas é a aquisição de dados (corrente) de cada carga instalada no sistema feita por um sensor de corrente não intrusivo (*TC*) interligado ao *Lolin32* para então alimentar uma planilha do Excel com dados de corrente para serem tratados posteriormente como um banco de dados de treino do sistema. O sensor de corrente ficará posicionado logo após o disjuntor geral de entrada onde será analisada a corrente total que passa pelo circuito de cargas e o microcontrolador *Esp32* ficará em uma *proto board* e precisará de uma alimentação de 5V, podendo ser pilha, porta *USB* ou qualquer carregador de 5V de pelo menos 500mA. Embora o módulo *Lolin32* seja alimentado com 5V, ele trabalha junto a um regulador de 3V3 acoplado ao seu módulo para alimentação do microcontrolador *Esp32*. A partir dos dados de treino presentes no banco de dados, o computador com a ajuda de um algoritmo de aprendizado de máquinas, reconhece os padrões de cargas ligadas em tempo real e se torna capaz de realizar previsões tornando assim um sistema inteligente de monitoramento de consumo. A Figura 12 mostra como é feito o funcionamento do sistema proposto em forma de diagrama de blocos.

Figura 12 – Diagrama de blocos do funcionamento do sistema desenvolvido

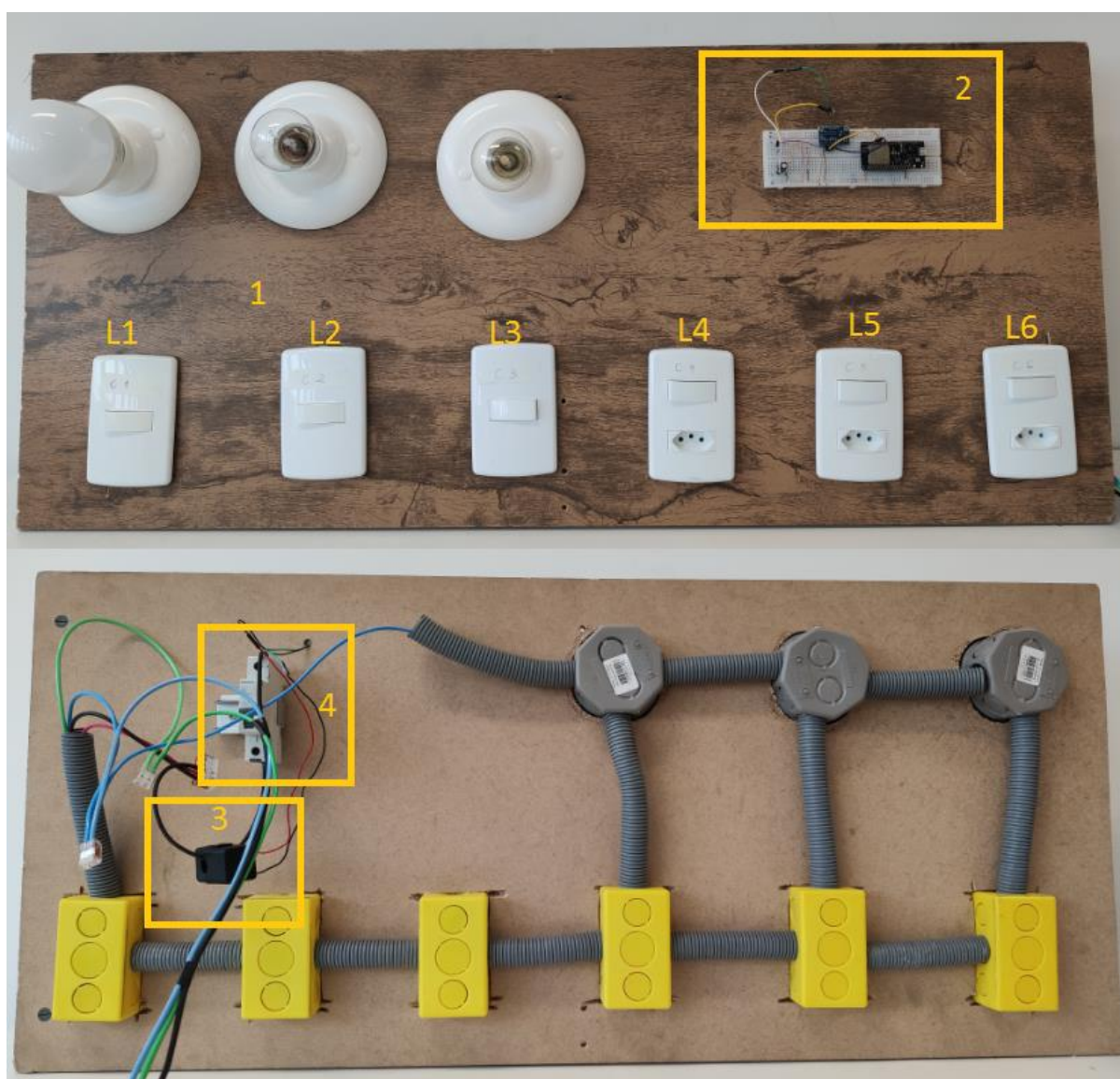


Fonte: Autoria própria

4.1.1 VISÃO GERAL DO PROTÓTIPO DESENVOLVIDO

A Figura 13 mostra a visão geral do sistema desenvolvido. O número 1 da Figura, mostra as disposições das cargas instaladas de L1 a L6. As cargas são de 20W, 40W, 60W, 120W, 320W e 490W, respectivamente. O número 2 da Figura 13 representa o sistema eletrônico desenvolvido para aquisição de dados do sensor de corrente, que é composto por: microcontrolador, conversor A/D externo, resistores e capacitor montados em uma *proto board*. No número 3 da Figura 13 representa-se o sensor de corrente não intrusivo colocado na fase do sistema desenvolvido. Finalmente, o número 4 da Figura 13 representa o disjuntor geral.

Figura 13 – Visão geral do protótipo desenvolvido

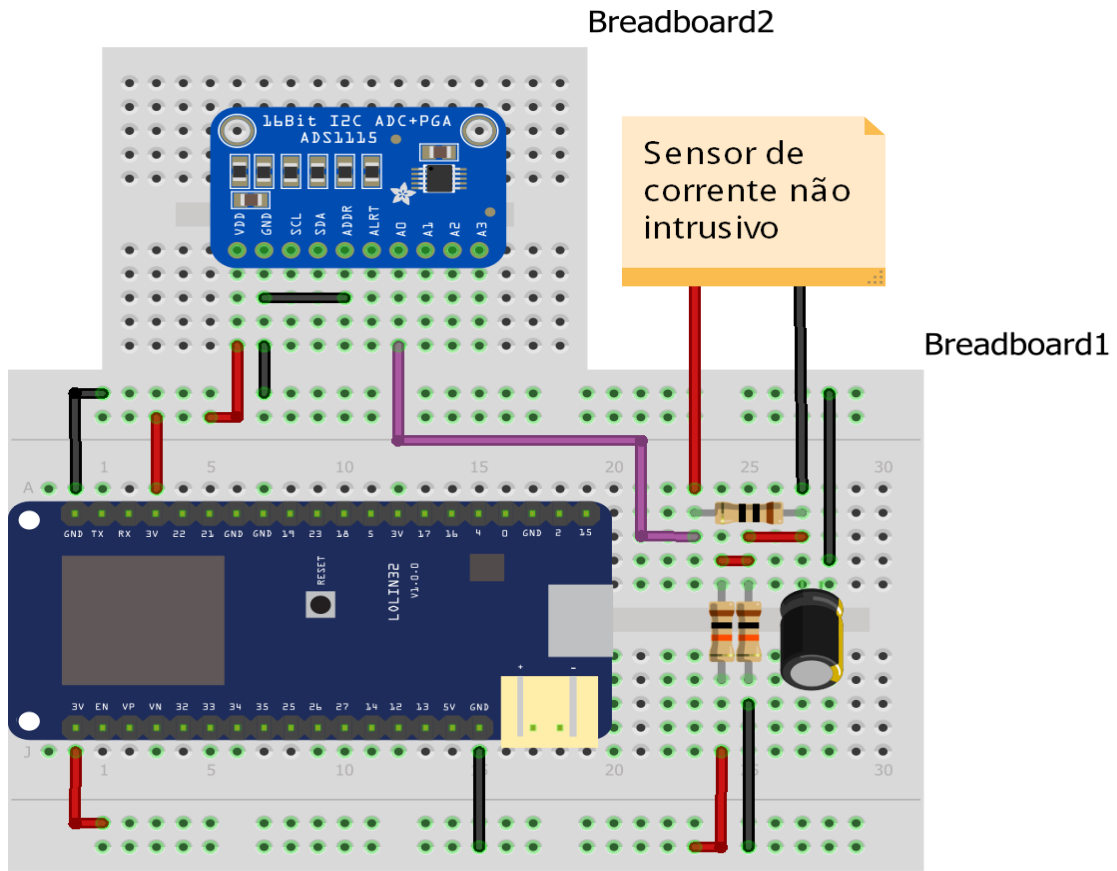


Fonte: Autoria própria

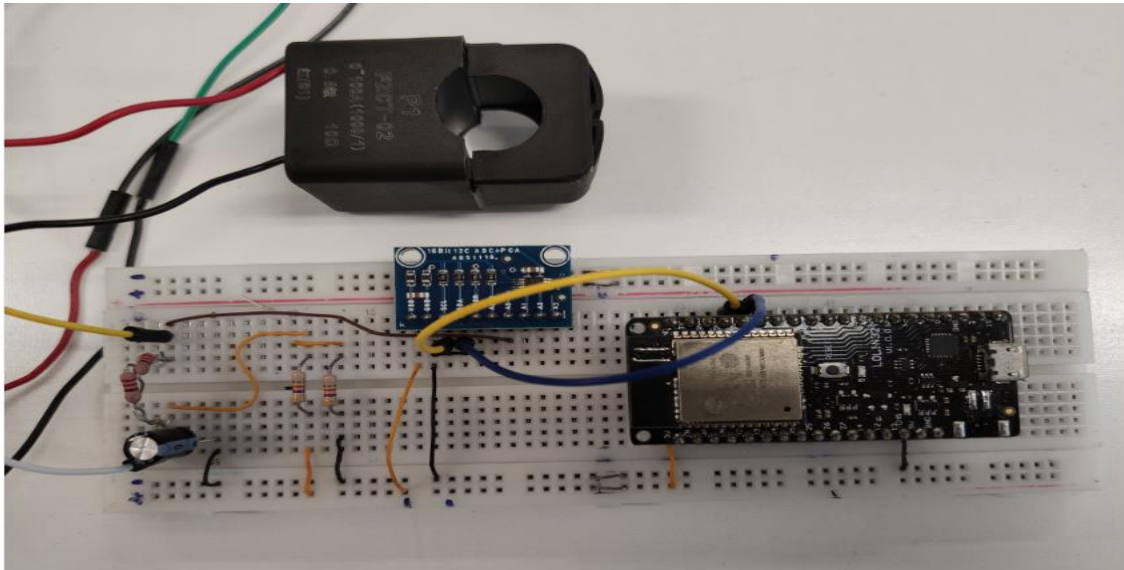
4.1.2 ESQUEMA ELETRÔNICO DO SISTEMA DESENVOLVIDO

Na Figura 14 apresenta-se o esquema do circuito de aferição do consumo de energia.

Figura 14 – Esquema do sistema desenvolvido



fritzing



Fonte: Autoria própria

4.2 APLICAÇÃO DO SENSOR DE CORRENTE (TC)

O sensor de corrente apresenta em seu circuito secundário, a corrente que circula em seu enrolamento primário com sua posição vetorial substancialmente mantida em uma proporção definida e conhecida como relação de transformação.

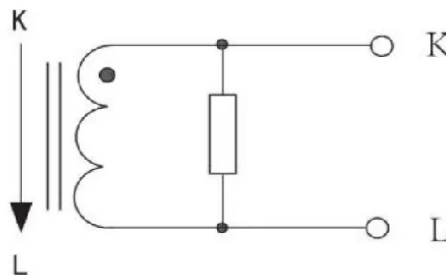
Para que se possa fazer a leitura da corrente com o *Esp32*, projeta-se um circuito para conversão das leituras de corrente em tensão pois o pino analógico do *Esp32* não realiza leitura de correntes.

- Em primeiro lugar, calcula-se a corrente máxima (corrente de pico) de entrada do circuito pela equação 9. Sabendo-se que a corrente eficaz (I_{ef}) máxima a ser medida pelo sensor é de 100A. Logo, encontra-se a corrente de pico (I_p) de acordo com a equação 9:

$$I_p = I_{ef} * \sqrt{2} \quad (9)$$

- Como a especificação de relação de transformação do TC é de 1:1000, ou seja, para uma corrente de 100A teremos na saída uma corrente de 100mA, podemos então com esse dado calcular o resistor de *burden* (ou de carga) que ficará em paralelo com a bobina do sensor para obtermos uma tensão na saída deste circuito como apresentado na Figura 15:

Figura 15 – Ligação do resistor de *burden* no TC



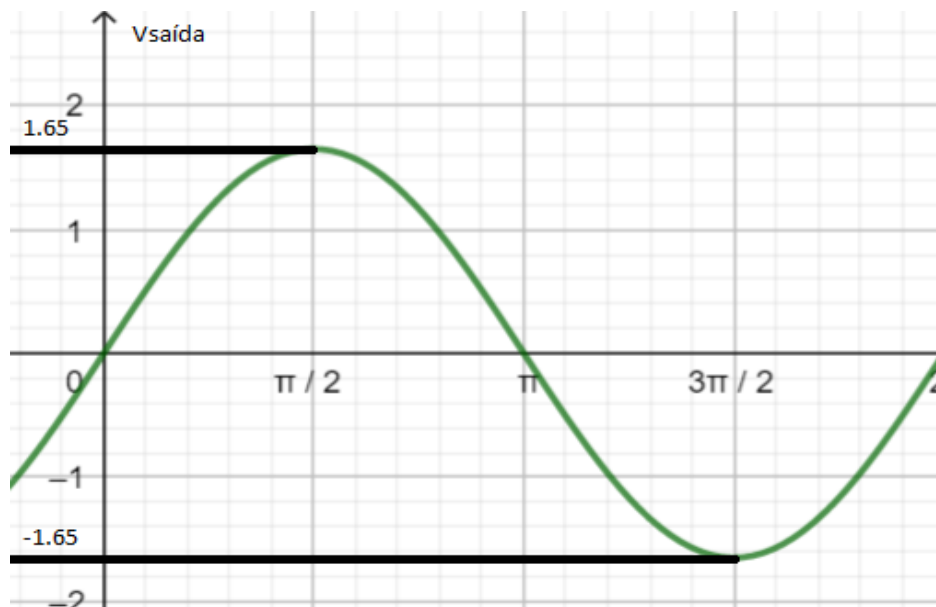
Fonte: Autoria própria

- O resistor de *burden* será calculado de acordo com a metade da tensão máxima de entrada no pino analógico do *Esp32*. Demonstra-se o cálculo do resistor a partir da equação 10.

$$R = \frac{V_{m\acute{a}x}}{I_{pSec}} = \frac{1.65}{0.1414} = 11.67\Omega \text{ ----- Valor comercial: } 10\Omega \quad (10)$$

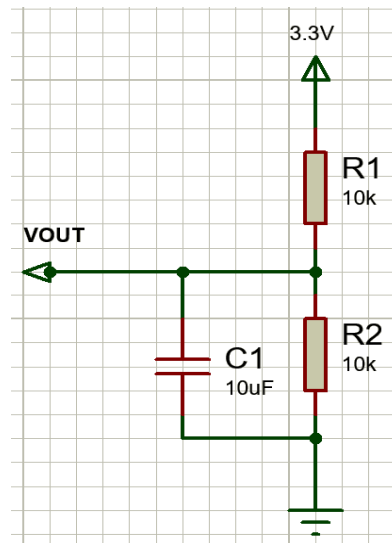
- Utilizando este resistor calculado, tem-se na saída (*Vsaída*) do circuito a seguinte forma de onda demonstrada na Figura 16:

Figura 16 – Vsaída CKT



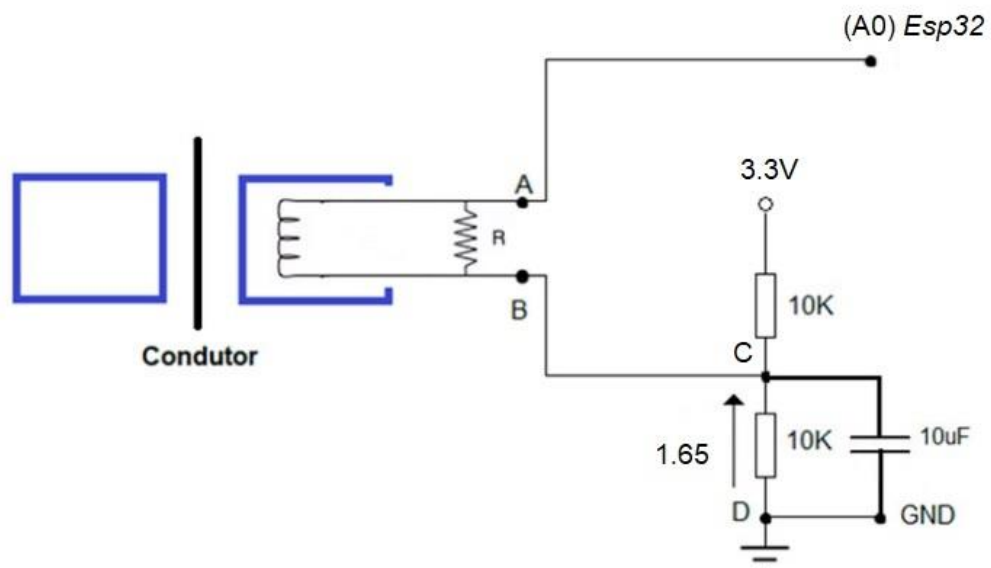
Fonte: Autoria própria

- Como o *Esp32* não lê valores de tensões negativas, então foi montado um circuito divisor de tensão (Figura 17) para adicionar 1.65Vdc de *offset* na saída do sensor de corrente (TC), deslocando assim a forma de onda para a escala positiva com o zero em 1.65Vdc.

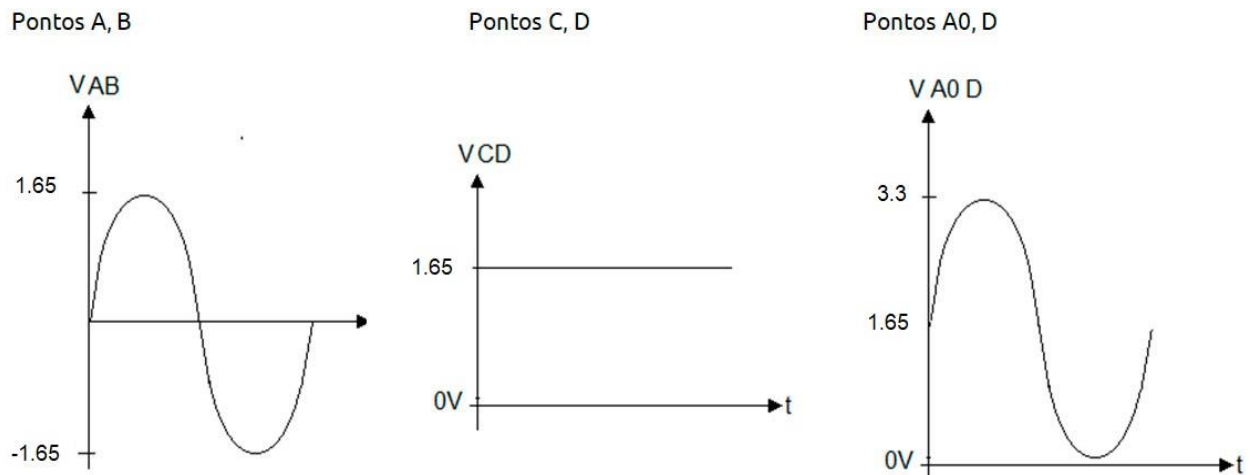
Figura 17 – Offset DC

Fonte: Autoria própria

- O capacitor está sendo aplicado em paralelo ao circuito divisor de tensão para que não tenha variação de tensão. O circuito final está sendo apresentado na Figura 18 e suas formas de onda na Figura 19.

Figura 18 – Circuito final AC sensor + offset DC

Fonte: Autoria própria

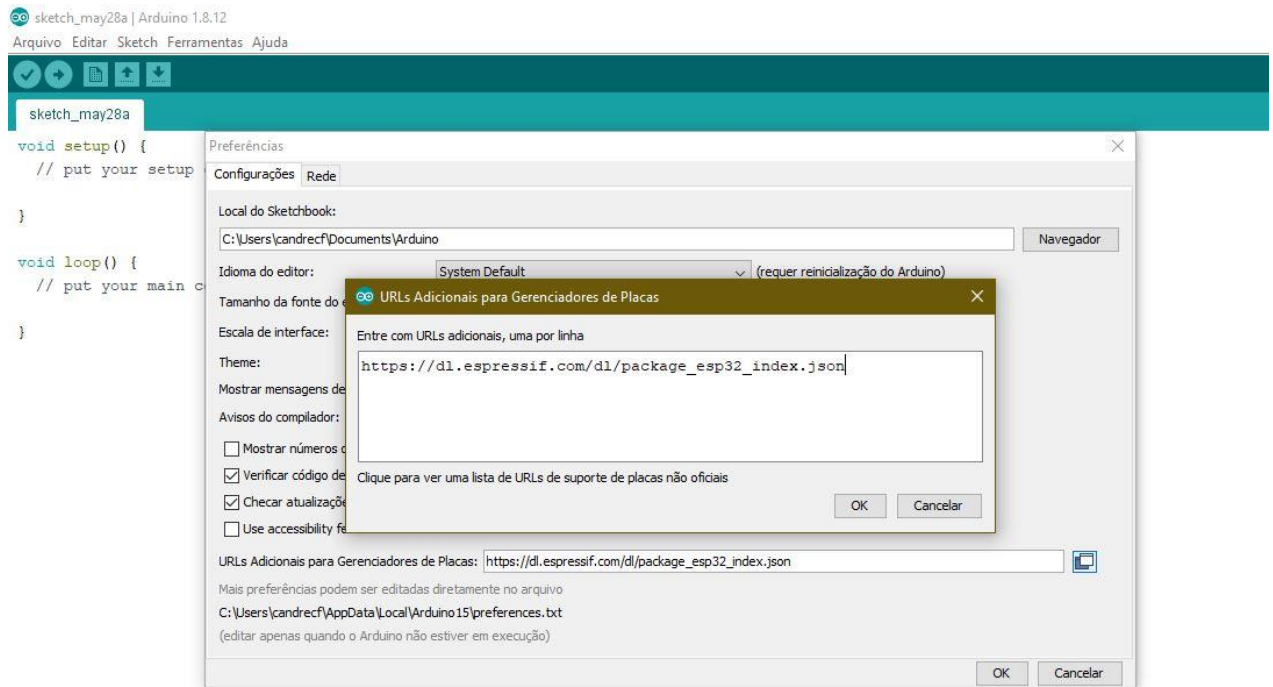
Figura 19 – Formas de Onda

Fonte: Autoria própria

4.3 PROGRAMAÇÃO DO MÓDULO LOLIN32

A programação do *Esp32* foi desenvolvida utilizando o *software* de programação do Arduino, a sua IDE. Para usar e programar o *Esp32* no *software* do Arduino, foi necessário alterar algumas configurações de preferência em sua IDE. Uma dessas alterações, foi disponibilizar um endereço *url* para que o Arduino IDE acesse uma pequena “base de dados” no formato. json que contém a configuração de inúmeras placas inclusive a que foi utilizada. Com esta *url*, a IDE passa a disponibilizar para *download* as configurações de placa do *Esp32*. A Figura 20, mostra a forma de inserção da *url*:

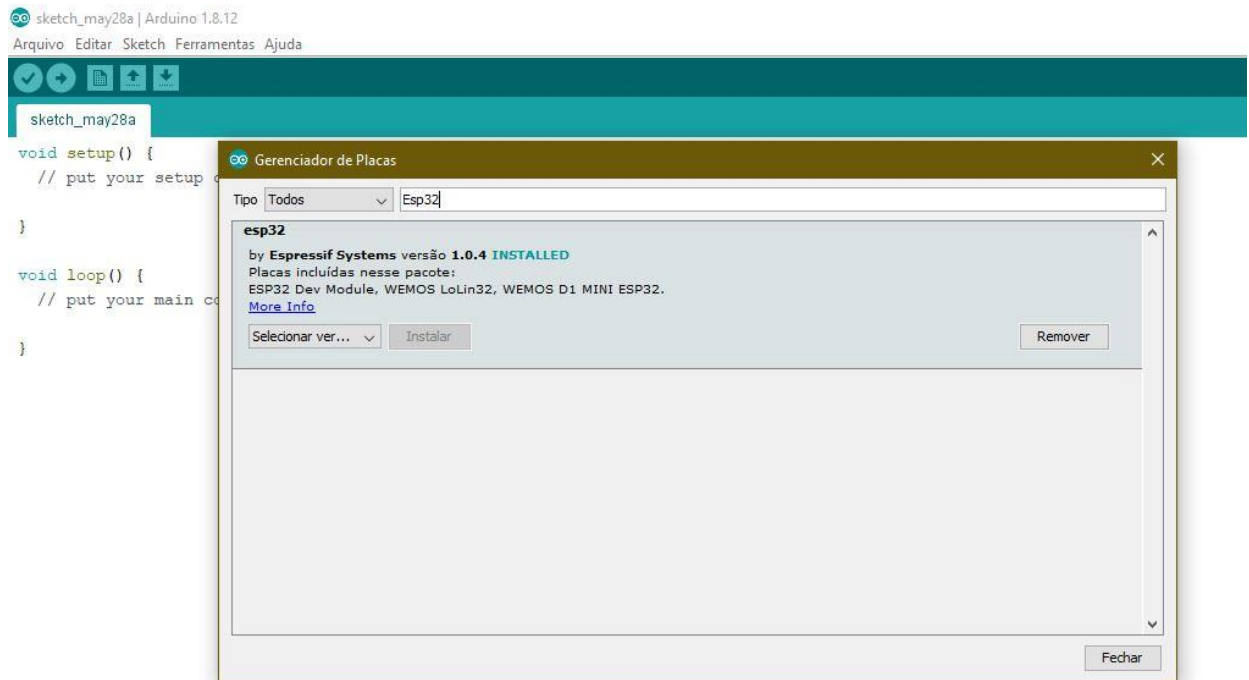
Figura 20 – Tela de preferência da IDE do Arduino



Fonte: Autoria própria

Após a inserção da *url* adicional para gerenciamento de placas nas configurações da IDE, pôde-se então seguir para a ferramenta de gerenciamento de placas pesquisar e instalar a versão mais recente do *driver* do microcontrolador *Esp32*, conforme ilustrado na Figura 21.


Figura 21 – Gerenciador de placas Arduino IDE



Fonte: Autoria própria

Após a instalação do *driver* mais recente do *Esp32*, já é possível realizar a programação do microcontrolador selecionado.

A programação do *Esp32* é realizada utilizando a mesma linguagem de programação do Arduino. O código de programação inicia com criações de variáveis, importação de bibliotecas prontas para a comunicação e a configuração do modo operacional do conversor analógico digital externo usado neste projeto com finalidade de aumentar a quantidade de amostras coletadas provindas do sensor de corrente. Podemos encontrar o começo do código na Figura 22.

Figura 22 – Definição das variáveis de programação


```

final_esp32
#include <Wire.h>
#include <Adafruit_ADS1015.h>

Adafruit_ADS1115 ads(0x48);
float Voltage = 0.0;
float tensao = 0.0;
float potencia = 0.0;
int espiras = 1000;
int burden = 110;
float v_rede = 127.0;

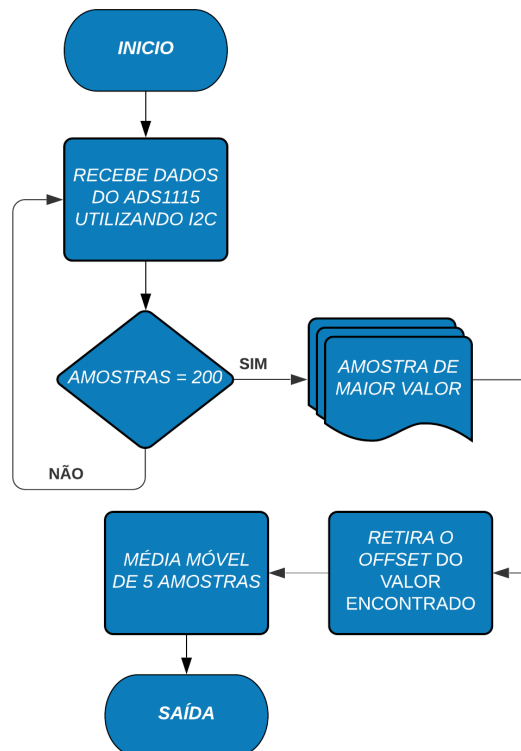
int vetCorrente[200];
int maior_Valor = 0;

float tensao2 = 0.0;
int n = 5;
int m = 0;
int j = 0;
float soma = 0.0;
float result = 0.0;

```

Fonte: Autoria própria

A rotina do programa em C++ desenvolvido na *IDE* do Arduino para aquisição de dados vindos do sensor de corrente está sendo representada no fluxograma presente na Figura 23.

Figura 23 - Fluxograma da rotina de aquisição de dados**Fonte:** Autoria própria

4.4 APLICAÇÃO DO CONVERSOR A/D

Foi criado uma rotina de armazenamento da tensão de pico proveniente do secundário do sensor de corrente não intrusivo (TC) em paralelo ao resistor de *burden*. Como foi observado na Figura 22, o vetor responsável pelo armazenamento destes valores é o *vetCorrente[200]* onde ele registra todos os valores de gradações do conversor digital analógico do *Esp32*. Com este vetor de 200 amostras, foi possível capturar a maior gradação (pico) da corrente no secundário do sensor de corrente. Como a frequência da rede elétrica é de 60Hz, então sabe-se que o valor de pico da rede varia entre pico positivo a pico negativo 60 vezes por segundo. Para calcular a corrente de pico durante um ciclo, é necessário calcular o período da onda que nada mais é do que o tempo em que se repete um ciclo completo de repetições (equação 11).

$$T = \frac{1}{f} = \frac{1}{60} = 16,67ms \quad (11)$$

O *Esp32* possui um *clock* de 240MHz. Demonstra-se então o cálculo do período a partir da equação 12

$$T = \frac{1}{f} = \frac{1}{240Mhz} = 4,1667us \quad (12)$$

De acordo com a equação 12, o *Esp32* tem seu período de repetição tão rápido que podemos desconsiderar este tempo no cálculo de aferição do pico da corrente durante um ciclo.

Para aferir sempre o valor de pico da senóide, foi criado este *vetorCorrente[200]* de 200 amostras com um *delay* de 100us totalizando assim um ciclo de captura de:

$$200 * 100us = 20000us \text{ ou } 20ms$$

O *Esp32* possui um conversor analógico digital integrado com resolução de 12bits, ou seja, 2^{12} tornando a faixa de leitura de 0 a 4095 gradações. Como a

alimentação do *Esp32* é de 3.3V, o valor referente a cada gradação, está sendo demonstrado na equação 13:

$$\left(\frac{V_{cc}}{\text{Resolução}}\right) * ADC = \left(\frac{3,3}{2^{12}}\right) * ADC = 0,80mV * ADC \quad (13)$$

Pode-se observar uma resolução de 0,80mV por gradação, ou seja, no mínimo teríamos que ter uma tensão no secundário do sensor de corrente (TC) que fosse maior que 0,80mV para que o conversor analógico digital conseguisse ler. Levando em consideração a utilização de uma carga de 10W, verifica-se uma tensão no secundário do sensor de corrente de acordo com a equação 14:

$$V_{Sec} = \frac{10W}{127Vac} = \frac{0,07874A}{1000 (esp)} = 0,07874mA * 10 (burden) = 0,7874mV \quad (14)$$

Com o valor de tensão do secundário do sensor de corrente calculado na equação 12, pode-se notar que se fosse necessário a aferição de cargas de baixo consumo como a do exemplo, o conversor analógico digital não conseguiria ler, ele interpretaria como sendo 0V. Para uma melhor precisão dos dados optou-se por usar um conversor analógico digital externo de 16bits, o ADS1115.

O conversor analógico digital ADS1115 (Figura 24) é um A/D com resolução de 16bits e com comunicação I2C muito utilizado em projetos que demandam mais resolução e exatidão. Sua alimentação pode ser de 2V a 5,5V, além disso possui uma amostragem de 8 a 860 amostras por segundo e possui um amplificador de ganho programável interno capaz de amplificar o sinal das entradas analógicas para aproveitar ao máximo o *range* do A/D. Este controle do amplificador de ganho programável é feito digitalmente via registradores internos acessados pela I2C. Configurando o seu amplificador ganho programável, podemos trabalhar com sinais entre +/-256mV até +/-6,144V (*range* padrão). Na Tabela 3 observa-se uma tabela de configuração do amplificador de ganho programável deste A/D demonstrada em seu *datasheet*:

Figura 24 - ADS1115



Fonte: FilipeFlop (2020)

Tabela 3 - Tabela de configuração do amplificador de ganho programável do ADS1115

FSR	LSB SIZE
$\pm 6.144 \text{ V}^{(1)}$	187.5 μV
$\pm 4.096 \text{ V}^{(1)}$	125 μV
$\pm 2.048 \text{ V}$	62.5 μV
$\pm 1.024 \text{ V}$	31.25 μV
$\pm 0.512 \text{ V}$	15.625 μV
$\pm 0.256 \text{ V}$	7.8125 μV

Fonte: Texas Instruments (2020)

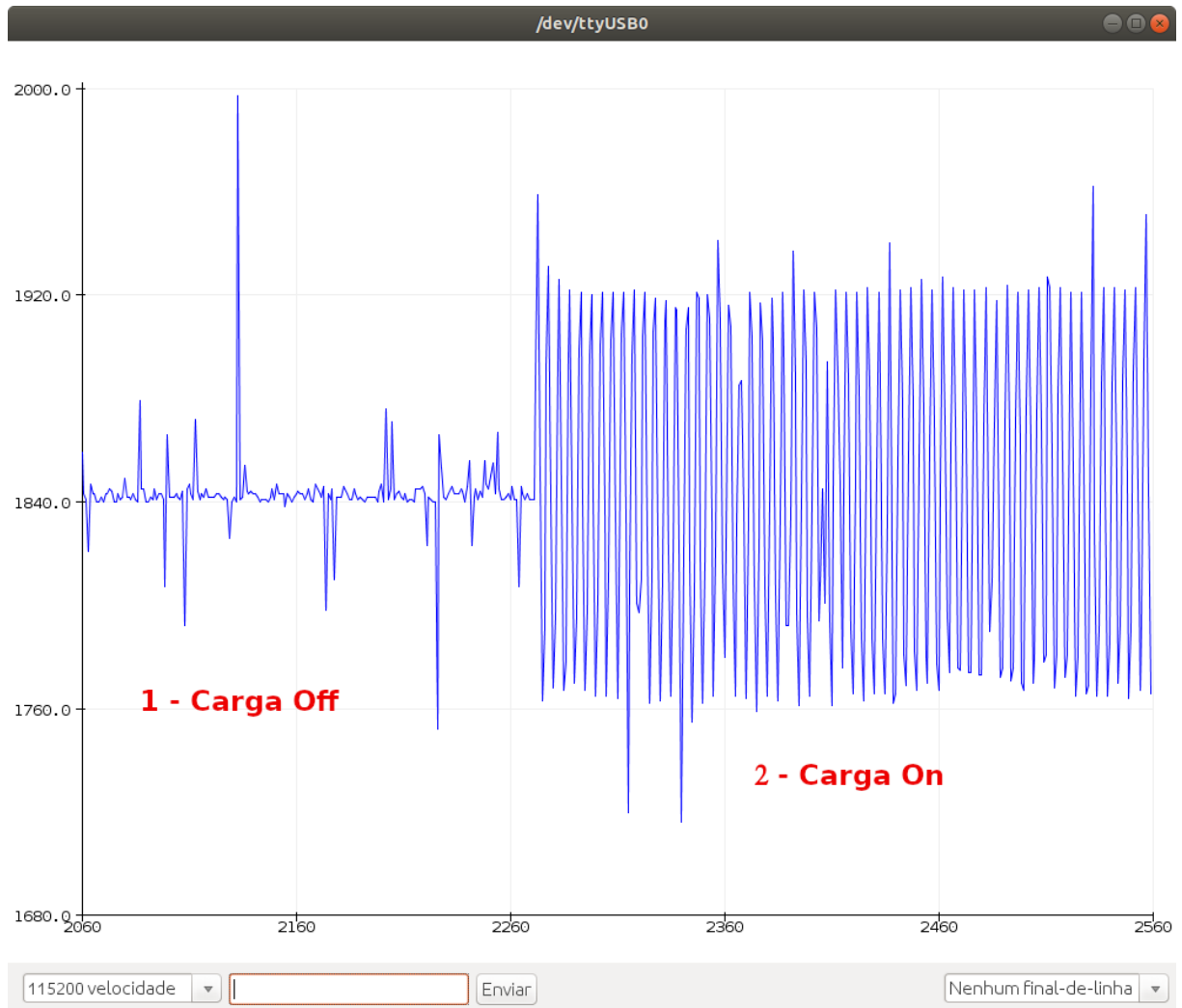
Como padrão, utiliza-se a configuração de $\pm 6,144\text{V}$, ou seja, 2/3 do ganho. Observa-se que quanto menor for a escolha da tensão na configuração, maior vai ser o ganho, como apresentado na Tabela 2, os ganhos variam entre 2/3x a 16x de ganho total. Quando maior o ganho, maior processamento e mais lento fica o sistema. Com o ganho do projeto setado para 2/3x do ganho total, calculamos então o valor de volts por gradação de acordo com a equação 15:

$$\left(\frac{V_{cc}}{\text{Resolução}} \right) * ADC = \left(\frac{6,144}{2^{15}} \right) * ADC = 0,1875\text{mV} * ADC \quad (15)$$

Com o valor de 0,1875mV por gradação, pode-se aferir e interpretar valores de corrente até mesmo de baixíssimas cargas. Este foi um dos motivos a optar por a usar este conversor para este projeto. O segundo motivo a usar este conversor, foi a realização de experimentos para verificar o sinal de entrada no conversor analógico digital do *Esp32*. Foi montado o circuito completo com o resistor de *burden* (carga) e o circuito divisor de tensão para inserir um *Offset* DC de 1.65V junto a saída do

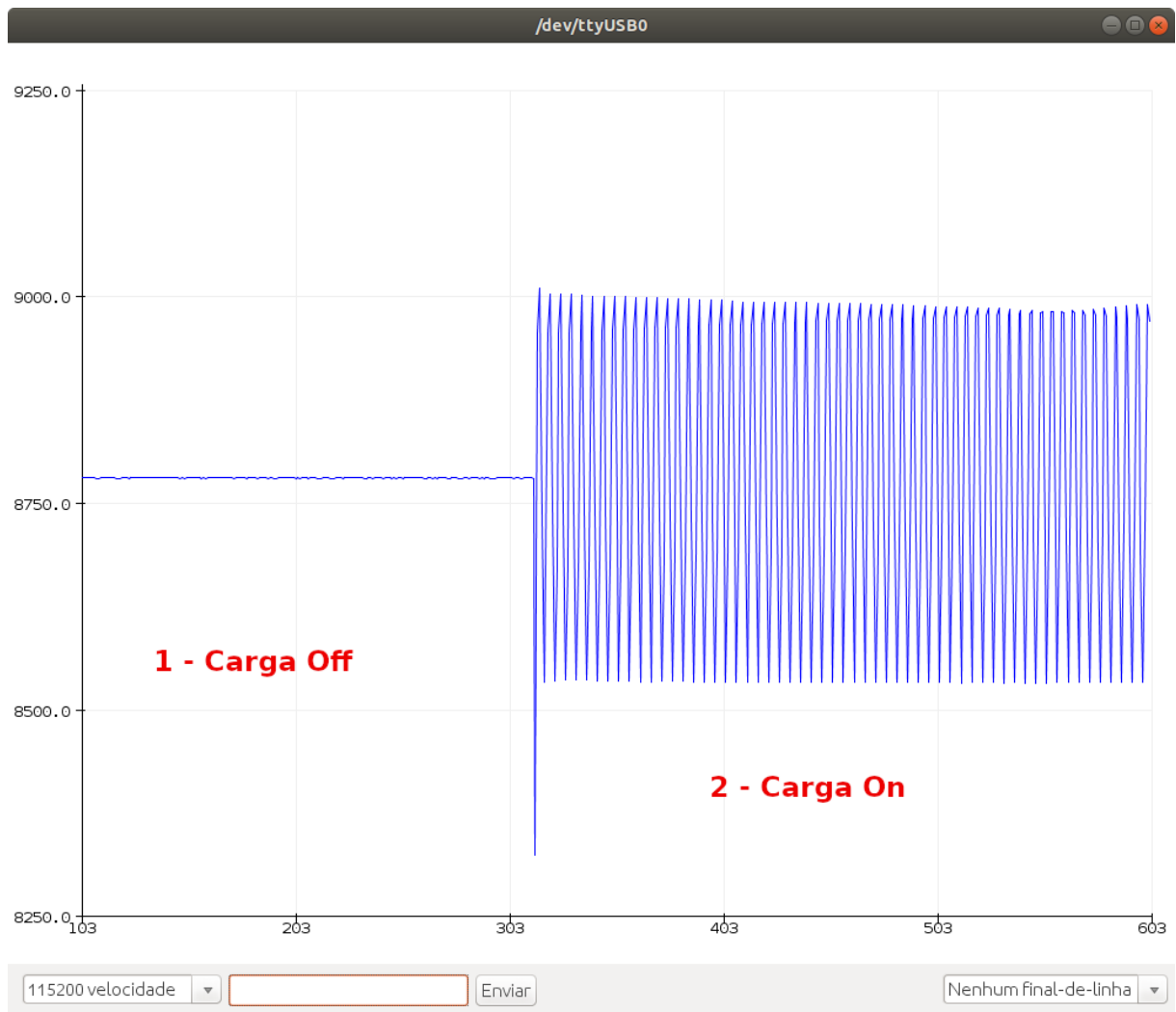
sensor de corrente do (TC). Então foi conectado este circuito completo a entrada analógica A0 do *Esp32*, ligamos uma carga qualquer e obteve-se o seguinte resultado apresentado na Figura 25 em relação a carga desligada e ligada:

Figura 25 - Representação gráfica do sensor de corrente + *Offset* - A0 do *Esp32*



Fonte: Autoria própria

Foi então realizado o mesmo experimento utilizando o conversor analógico digital externo, o ADS1115. A Figura 26 apresenta o mesmo experimento com a mesma carga, desligada e ligada:

Figura 26 - Representação gráfica do sensor de corrente + Offset - A0 do ADS1115

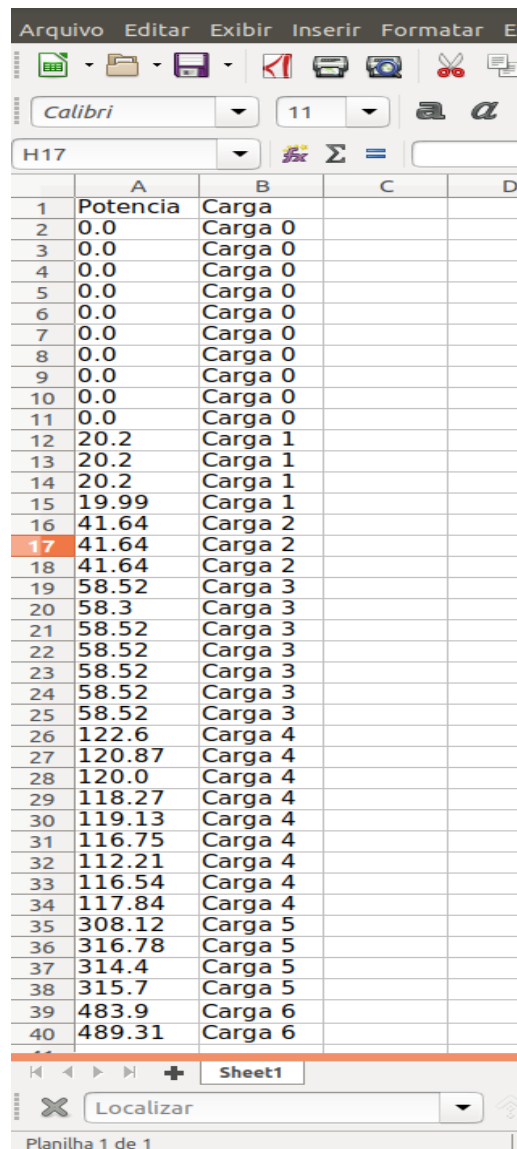
Fonte: Autoria própria

Foi observado então uma superioridade na Figura 26 (ADS1115) em relação ao A/D interno do *Esp32*, como foi apresentado na Figura 25, o conversor A/D do *Esp32* nos apresentou ruídos em sua entrada até mesmo com a carga desligada. Isso poderia ser resolvido com filtros analógicos ou digitais, mas optou-se por além de ter uma solução para este problema, mais resolução na construção do nosso projeto já que pode-se trabalhar com qualquer tipo de carga, tanto baixa como alta.

4.5 APLICAÇÃO DO ALGORITMO DE MACHINE LEARNING

Com a coleta dos valores de corrente proveniente do secundário do sensor de corrente não intrusivo (TC), o microcontrolador *Esp32* fica responsável por armazenar os dados em uma planilha para servir como banco de dados (*dataset*) de treino do sistema. Estes dados de treino serão apresentados ao algoritmo de *machine learning* para criar um modelo que ajudará na previsão de dados não conhecidos. Na figura 27 podemos observar uma planilha alimentada com dados de treino.

Figura 27 - Dados de treino

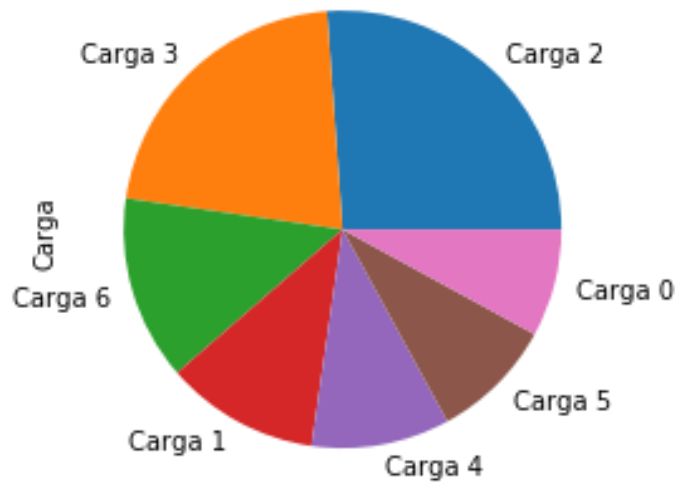


	A	B	C	D
1	Potencia	Carga		
2	0.0	Carga 0		
3	0.0	Carga 0		
4	0.0	Carga 0		
5	0.0	Carga 0		
6	0.0	Carga 0		
7	0.0	Carga 0		
8	0.0	Carga 0		
9	0.0	Carga 0		
10	0.0	Carga 0		
11	0.0	Carga 0		
12	20.2	Carga 1		
13	20.2	Carga 1		
14	20.2	Carga 1		
15	19.99	Carga 1		
16	41.64	Carga 2		
17	41.64	Carga 2		
18	41.64	Carga 2		
19	58.52	Carga 3		
20	58.3	Carga 3		
21	58.52	Carga 3		
22	58.52	Carga 3		
23	58.52	Carga 3		
24	58.52	Carga 3		
25	58.52	Carga 3		
26	122.6	Carga 4		
27	120.87	Carga 4		
28	120.0	Carga 4		
29	118.27	Carga 4		
30	119.13	Carga 4		
31	116.75	Carga 4		
32	112.21	Carga 4		
33	116.54	Carga 4		
34	117.84	Carga 4		
35	308.12	Carga 5		
36	316.78	Carga 5		
37	314.4	Carga 5		
38	315.7	Carga 5		
39	483.9	Carga 6		
40	489.31	Carga 6		

Fonte: Autoria própria

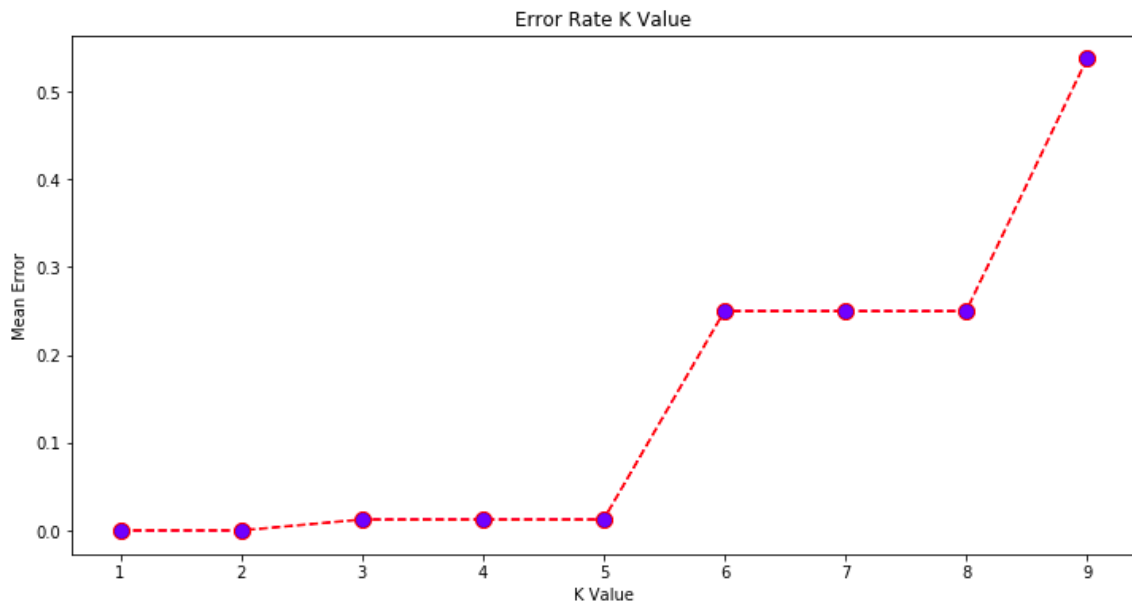
Com o *dataset* de treino criado, foi implementado um algoritmo para que fosse possível realizar a criação do modelo a partir destes dados e realizar previsão de todos dados futuros não conhecidos. Na Figura 28 podemos ver um gráfico em forma de pizza do *dataset* de treino apresentando a quantidade de aferição de consumo de cada carga foi armazenado.

Figura 28 – Gráfico do *dataset* de treino



Fonte: Autoria própria

Foi utilizado no projeto o algoritmo *KNN* conhecido como sendo “Algoritmo de Vizinhos Mais Próximos”, este algoritmo classifica um dado de acordo com a distância entre pontos em um espaço euclidiano (instâncias) e os dados mais frequentes dentro de K amostras são utilizados então para um esquema de votação. O valor da constante K , é o número de vizinhos mais próximos do dado que você quer que o modelo preveja, se K for muito pequeno, a classificação fica sensível a pontos de ruído. Por outro lado, se K for muito grande, a “vizinhança” pode incluir elementos de outras classes levando assim a um erro na previsão do modelo. Foi feito então alguns testes para K onde aplica-se valores de K de 1 até 10 para que seja encontrado um melhor valor para o algoritmo, ou seja, menor erro. A Figura 29 apresenta um gráfico do erro esperado para cada valor de K entre 1 e 10.

Figura 29 - Erro para valores de K

Fonte: Autoria própria

Foi observado que quanto maior for o valor da constante K maior será o erro na previsão do modelo, isso significa que deve-se escolher qualquer valor de K em que o erro se aproxime de 0. Foi escolhido então o valor de $K = 1$. Um dos motivos para a utilização do KNN neste projeto é porque este algoritmo possui uma boa precisão na maioria dos casos aplicados, além disso ele possui um tempo para efetuar treinamento dos dados muito diferenciado, sendo um dos mais rápidos para esta atividade.

Com o modelo de previsão pronto, foi apresentado a ele os dados de teste no qual são os dados em tempo real (consumo atual) permitindo assim realizar previsões em tempo real das cargas instaladas.

4.5.1 RESULTADOS DO KNN – MATRIZ DE CONFUSÃO

Para avaliar e comparar as previsões, foi aplicado o processo de matriz de confusão em um banco de dados com 80 dados armazenados para um teste experimental. A matriz de confusão foi gerada a partir de uma biblioteca disponível para *python*. Os dados gerados pela matriz de confusão são apresentados em forma de uma relação entre os dados esperados e os dados apresentados pela previsão. A Figura 30 ilustra a aplicação da matriz de confusão no sistema proposto.

Figura 30 – Matriz de Confusão K=1

```
In [15]: print(confusion_matrix(y_teste, knn.predict(x_teste)))
```

```
[[ 1  0  0  0  0  0  0]
 [ 0  5  0  0  0  0  0]
 [ 0  0 23  0  0  0  0]
 [ 0  0  0  4  0  0  0]
 [ 0  0  0  0 18  0  0]
 [ 0  0  0  0  0 14  0]
 [ 0  0  0  0  0  0 15]]
```

```
In [16]: print(pd.crosstab(y_teste, knn.predict(x_teste), rownames=['Esp.'], colnames=['Apresen.'], margins = True))
```

Apresen.	Carga 0	Carga 1	Carga 2	Carga 3	Carga 4	Carga 5	Carga 6	All
Esp.								
Carga 0	1	0	0	0	0	0	0	1
Carga 1	0	5	0	0	0	0	0	5
Carga 2	0	0	23	0	0	0	0	23
Carga 3	0	0	0	4	0	0	0	4
Carga 4	0	0	0	0	18	0	0	18
Carga 5	0	0	0	0	0	14	0	14
Carga 6	0	0	0	0	0	0	15	15
All	1	5	23	4	18	14	15	80

Fonte: Autoria própria

Pode-se notar que assumindo um valor de K=1 nosso sistema não apresentou nenhum erro na classificação (previsão). Com a matriz de confusão apresentada é possível calcular métricas de avaliação para a classificação (previsão). Pode-se observar na Figura 31 a aplicação destas métricas de avaliação neste mesmo banco de dados de teste experimental.

Figura 31 – Métricas de avaliação K=1

```
In [17]: print(classification_report(y_teste, pred))
```

	precision	recall	f1-score	support
Carga 0	1.00	1.00	1.00	1
Carga 1	1.00	1.00	1.00	5
Carga 2	1.00	1.00	1.00	23
Carga 3	1.00	1.00	1.00	4
Carga 4	1.00	1.00	1.00	18
Carga 5	1.00	1.00	1.00	14
Carga 6	1.00	1.00	1.00	15
accuracy			1.00	80
macro avg	1.00	1.00	1.00	80
weighted avg	1.00	1.00	1.00	80

Fonte: Autoria própria

Observa-se uma precisão, sensibilidade, acurácia e *F1-score* de 100% para todas as classes da classificação (previsão).

Para realizar uma prova real sobre a influência do valor da constante K sobre a classificação (previsão), assumiu-se então um valor de $K = 3$ e foi aplicado novamente o processo de matriz de confusão e a aplicação das métricas de avaliação para verificar como a classificação (previsão) se comporta. As Figuras 32 e 33 representam os novos resultados.

Figura 32 – Matriz de confusão K=3

```
In [15]: print(confusion_matrix(y_teste, knn.predict(x_teste)))
```

```
[[ 0  1  0  0  0  0  0]
 [ 0  5  0  0  0  0  0]
 [ 0  0 23  0  0  0  0]
 [ 0  0  0  4  0  0  0]
 [ 0  0  0  0 18  0  0]
 [ 0  0  0  0  0 14  0]
 [ 0  0  0  0  0  0 15]]
```

```
In [16]: print(pd.crosstab(y_teste, knn.predict(x_teste), rownames=['Esp.'], colnames=['Apresen.'], margins = True))
```

Apresen.	Carga 1	Carga 2	Carga 3	Carga 4	Carga 5	Carga 6	All
Esp.							
Carga 0	1	0	0	0	0	0	1
Carga 1	5	0	0	0	0	0	5
Carga 2	0	23	0	0	0	0	23
Carga 3	0	0	4	0	0	0	4
Carga 4	0	0	0	18	0	0	18
Carga 5	0	0	0	0	14	0	14
Carga 6	0	0	0	0	0	15	15
All	6	23	4	18	14	15	80

Fonte: Autoria própria

Figura 33 – Métricas de avaliação K=3

```
In [17]: print(classification_report(y_teste, pred))
```

	precision	recall	f1-score	support
Carga 0	0.00	0.00	0.00	1
Carga 1	0.83	1.00	0.91	5
Carga 2	1.00	1.00	1.00	23
Carga 3	1.00	1.00	1.00	4
Carga 4	1.00	1.00	1.00	18
Carga 5	1.00	1.00	1.00	14
Carga 6	1.00	1.00	1.00	15
accuracy			0.99	80
macro avg	0.83	0.86	0.84	80
weighted avg	0.98	0.99	0.98	80

Fonte: Autoria própria

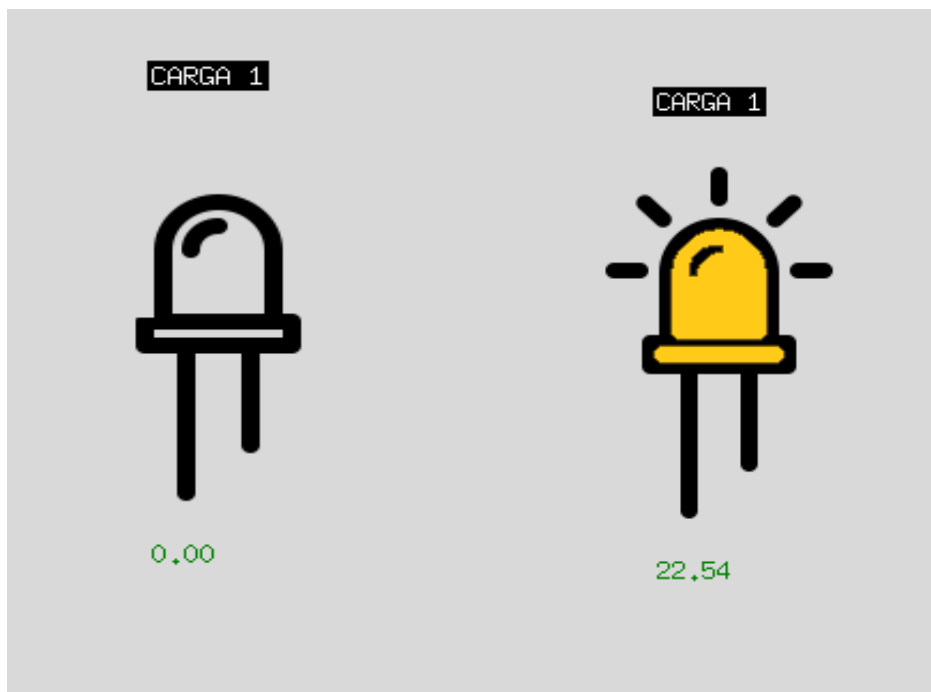
Pode-se observar que para um valor de $K=3$ neste algoritmo aplicado, a classificação (previsão) apresenta erro.

4.6 INTERFACE AO USUÁRIO

A interface ao usuário é responsável por possibilitar uma apresentação mais objetiva dos dados obtidos. Nesse projeto, a interface gráfica ao usuário apresentará o estado atual (ligado ou desligado) e a potência ativa individualmente e do conjunto das cargas instaladas no sistema.

O programa desenvolvido com a ajuda da linguagem *python*, possui 6 imagens com formato de um *LED*. A cada previsão feita pelo sistema se a carga estiver ligada ou desligada, a imagem do *LED* muda de acordo com a condição operacional. Logo embaixo das imagens que representam as cargas, é exibido o seu consumo individualmente. A Figura 34 ilustra a interface com os dois possíveis cenários, desligado e ligado.

Figura 34 – Condição desligado/ligado

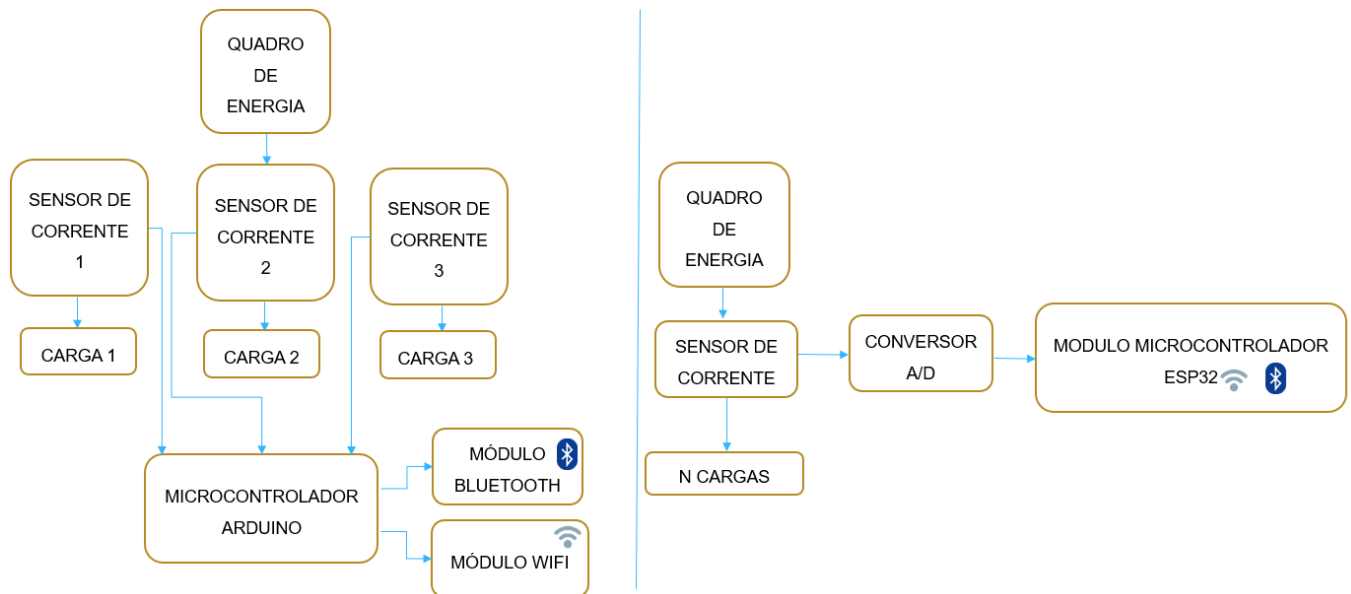


Fonte: Autoria própria

4.7 VIABILIDADE ECONÔMICA

A Figura 35 apresenta um fluxograma de comparação entre o sistema existente de monitoramento de consumo de energia e o sistema apresentado neste trabalho.

Figura 35 – Fluxograma de comparação entre sistema existente x sistema apresentado





Fonte: Autoria própria

Observa-se através do fluxograma da Figura 35 uma redução na quantidade de componentes do sistema existente. No sistema existente, o projetista costuma usar um sensor de corrente individual para cada carga, um módulo microcontrolador (Arduino) e os módulos externos de *bluetooth* e *Wi-Fi* enquanto o sistema apresentado neste trabalho possui apenas um sensor de corrente, um módulo microcontrolador (Esp32) e um conversor A/D para melhorar a precisão do sistema. Destaca-se que quanto mais quantidades de cargas aferidas no sistema existente, mais sensor de corrente deverá ser utilizado encarecendo então mais ainda o custo do projeto. Já no sistema apresentado não importando a quantidade de cargas o custo sempre será o mesmo em relação a quantidade de componentes do projeto.

Foi efetuado alguns cálculos de comparação em relação ao custo efetivo do projeto existente e o apresentado. Podemos conferir na Tabela 4:

Tabela 4 - Custo total do projeto existente x apresentado

CUSTOS				CUSTOS		
Quant.	Componente	Valor R\$		Quant.	Componente	Valor R\$
1	ARDUINO MEGA	100,00		1	WEMOS LOLIN ESP32	49,00
3	SENSORES DE CORRENTE	42,00		1	SENSOR DE CORRENTE PZCT-2 100A	42,00
1	MODULO WIFI ESP8266	30,00		1	CONVERSOR A/D – ADS1115	17,00
1	MODULO BLUETOOTH	30,00		TOTAL		108,00
TOTAL		286,00				

Fonte: Autoria própria

Observa-se uma economia de 62% equivalente a R\$178,00. Devemos ressaltar que quanto mais sensores de corrente utilizados no sistema existente, mais caro o projeto ficará e se tornará cada vez mais viável economicamente o sistema apresentado neste trabalho.

5 – CONSIDERAÇÕES FINAIS

A instalação deste sistema em uma residência, comércio ou indústria, permite ao consumidor um controle real sobre a energia elétrica consumida, tendo uma visão ampla de como, onde e quando a energia é utilizada, e assim poder tomar decisões para um consumo mais eficiente de energia.

De acordo com o que foi proposto, projetado e apresentado o trabalho atingiu o objetivo esperado ao prever quais cargas foram acionadas no sistema de forma precisa e clara ao usuário, indicando o consumo individual e total das cargas.

Vale ressaltar que o baixo custo, relativo aos equipamentos, para implementar o protótipo é praticamente fixo, pois o protótipo utilizará o mesmo número de equipamentos independentemente do número de circuitos ou equipamentos que o usuário queira medir. Deste modo o valor será alterado apenas de acordo com o modelo do TC utilizado pois ele será dimensionado de acordo com a faixa de medição de corrente.

Considerando os trabalhos existentes neste mesmo sentido e fazendo uma comparação rápida é possível considerar que a solução descrita neste projeto é completa e entrega ao usuário uma ótima alternativa para monitoramento do consumo de energia.

Diante do exposto, constata-se a relevância e importância do presente projeto para a sociedade, ao contribuir para a redução do consumo e utilização de forma consciente e eficiente da energia.

REFERÊNCIAS

- AALOK *et al.* **Como a IA, IoT e a análise de dados podem garantir a eficiência energética nas empresas.** [S. l.], 20/08/2019. Disponível em: <https://aalok.com.br/blog/analise-dados-eficiencia-energetica/>. Acesso em: 3 mar. 2020.
- BITENCOURT, L. **Conversor A/D.** 2016. Fotografia. Disponível em: <https://luizbitencourt.files.wordpress.com/2016/09/adc.png?w=577&h=163>. Acesso em: 3 mar. 2020.
- DEITEL, H. M. **C++ - Como Programar.** [S. l.: s. n.], 2006. 26 p. Disponível em: https://moodle.ufsc.br/pluginfile.php/2377663/mod_resource/content/0/Como%20Programar%20C%2B%2B%20Deitel%205ed.pdf. Acesso em: 3 mar. 2020.
- FILIPEFLOP. **ADS1115.** 2020. Fotografia. Disponível em: <https://uploads.filipeflop.com/2017/11/ads1115.jpg>. Acesso em: 3 mar. 2020.
- FIRJAN *et al.* **Energia Elétrica: Custo e qualidade para a competitividade da indústria nacional.** [S. l.], 2017. Disponível em: <https://www.firjan.com.br/firjan/empresas/competitividade-empresarial/temas-em-destaque/energiaeletrica/custo/default-7.htm>. Acesso em: 3 mar. 2020.
- GARBADE, M. J. **Prós e contras da sintaxe alternativa do C++.** [S. l.], 26 jan. 2017. Disponível em: <https://blog.education-ecosystem.com/pros-contras-sintaxe-alternativa-c/>. Acesso em: 3 mar. 2020.
- GIL, A. C. *et al.* **Métodos e técnicas de pesquisa social.** [S. l.: s. n.], 2011. Disponível em: <https://ayanrafael.files.wordpress.com/2011/08/gil-a-c-mc3a9todos-e-tc3a9cnicas-de-pesquisa-social.pdf>. Acesso em: 3 mar. 2020.
- GSTATIC. **Conversor A/D tipo delta-sigma.** 2020. Fotografia. Disponível em: <https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcSFyGkmuzm9Hq9bBlwi1N2yWzGfLI44bSbk8tojzdHPpCNcKmkP&usqp=CAU>. Acesso em: 3 mar. 2020.
- IMPACTA. **Conheça a Linguagem Python e saiba como usá-la!** [S. l.], 2018. Disponível em: <https://www.impacta.com.br/blog/2018/09/17/conheca-a-linguagem-python-e-saiba-como-e-possivel-usa-la/>. Acesso em: 3 mar. 2020.
- JOSÉ, I. **KNN (K-Nearest Neighbors) #1: Como funciona?.** [S. l.], 1 jul. 2018. Disponível em: <https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>. Acesso em: 3 mar. 2020.
- KOLBAN, N. **Kolban's Book on Esp32.** USA: [s. n.], 2017. 785 p. v. 1. Disponível em: <https://www.robolinkmarket.com/Data/EditorFiles/datasheet/kolban-ESP32.pdf>. Acesso em: 3 mar. 2020.
- LAPA, C. *et al.* **Importância da medição de energia para a eficiência energética.** [S. l.: s. n.], 2010. p. 1. Disponível em: <https://docplayer.com.br/9698165->

Importancia-da-medicao-de-energia-para-a-eficiencia-energetica.html. Acesso em: 3 mar. 2020.

MLSTATIC. **Arduino Mega**. 2020. Fotografia. Disponível em: https://http2.mlstatic.com/arduino-mega-2560-r3-cabo-usb-D_NQ_NP_798452-MLB41920604298_052020-F.webp. Acesso em: 3 mar. 2020.

PINHEIRO, J. M. S. Redes de proteção. **Sistemas de Automação**, [S. l.], p. 1, 13 jun. 2004. Disponível em: https://www.projetoderedes.com.br/artigos/artigo_sistemas_automacao.php. Acesso em: 3 mar. 2020.

ROBOCORE. **Protocolo I2C**. 2020. Fotografia. Disponível em: https://www.robocore.net/upload/tutoriais/50_img_4_H.png?519. Acesso em: 3 mar. 2020.

RODRIGUES, V. **Exemplo de matriz de confusão**. 2020. Fotografia. Disponível em: https://miro.medium.com/max/1400/1*s7VB26Cfo1LdVZcLou-e0g.png. Acesso em: 3 mar. 2020.

SANTOS, J. W.; JUNIOR, R. C. L. **Sistema de automatização residencial de baixo custo controlado pelo microcontrolador esp32 e monitorado via smartphone**. 2019. Trabalho de conclusão de curso (Tecnólogo em Automação Industrial) - Universidade Tecnológica Federal do Paraná, [S. l.], 2019. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/12133/1/PG_COAUT_2019_1_02.pdf. Acesso em: 3 mar. 2020.

SANTOS, W. **O que é Linguagem de Programação?**. [S. l.], 6 set. 2018. Disponível em: <https://www.ymda.com.br/business/linguagem-de-programacao/>. Acesso em: 3 mar. 2020.

SEGATTO, E. C.; COURY, D. V. A differential relay for power transformers using intelligent tools. **IEEE Transactions on Power Systems**, [S. l.], v. 21, 31 jul. 2006. 3, p. 1154- 1162. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1664950>. Acesso em: 3 mar. 2020.

SIMPLICIO, F. **ESP32 – Primeiros Passos com SDK-IDF (Programação Nativa em C/C++)**. [S. l.], 10 jul. 2018. Disponível em: <https://www.universidadeiot.com.br/esp32-primeiros-passos-com-sdk-idf-programacao-nativa-em-cc/>. Acesso em: 3 mar. 2020.

SOLEAR. **Sensor de corrente não intrusivo**. 2020. Fotografia. Disponível em: https://www.solear.net.br/loja/957-large_default/1.jpg. Acesso em: 3 mar. 2020.

TIAGO, M. **O que é machine learning e como funciona?**. [S. l.], 29 jan. 2018. Disponível em: <https://transformacaodigital.com/dados/o-que-e-machine-learning-e-como-funciona/>. Acesso em: 3 mar. 2020.

VERHAERT. **Aprendizado de Máquina**. 2020. Fotografia. Disponível em: <https://verhaert.com/wp-content/uploads/2018/05/Machine-vs-Deep-Learning-InputOutput.jpg>. Acesso em: 3 mar. 2020.

WEMOS. **WEMOS LOLIN32**. 2020. Fotografia. Disponível em: https://docs.wemos.cc/en/latest/_static/boards/d32_v1.0.0_1_16x16.jpg. Acesso em: 3 mar. 2020.