



Universidade Federal de Uberlândia

Faculdade de Engenharia Elétrica

KAIKE ALVES SILVA

**MONITOR DE CONSUMO DE ENERGIA ELÉTRICA COM CONEXÃO À
INTERNET**

Uberlândia

2018

Kaike Alves Silva

**MONITOR DE CONSUMO DE ENERGIA ELÉTRICA COM CONEXÃO À
INTERNET**

Trabalho apresentado como requisito parcial de avaliação na disciplina Trabalho de Conclusão de Curso do Curso de Engenharia Elétrica da Universidade Federal de Uberlândia.

Orientador: Ernane Antônio Alves Coelho

Assinatura do Orientador

Uberlândia

2018

AGRADECIMENTOS

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Elétrica, por me proporcionarem um ensino de alto nível e acesso a diversas oportunidades de crescimento pessoal e profissional.

Ao professor Ernane Coelho, pela orientação e suporte.

A Jardel Vieira, meu supervisor em meu primeiro estágio, pela oportunidade de mostrar meu trabalho, pelos inúmeros ensinamentos e por desenvolver em mim uma paixão ainda maior pela tecnologia, que culminou na ideia deste trabalho.

Aos colegas do NUPEP, pela disponibilidade e suporte durante o desenvolvimento do trabalho.

A minha namorada Flávia, pela inspiração, companheirismo e apoio constantes.

E finalmente a meus pais, sem os quais não teria atingido nenhum de meus objetivos, pela confiança e amor inigualáveis.

RESUMO

O bom domínio das próprias características de consumo de energia elétrica tem se tornado cada vez mais interessante aos consumidores residenciais. Entretanto, ferramentas de análise de energia elétrica são geralmente encontradas no mercado a valores altos e oferecem poucas opções de personalização. Visando propor uma solução a este problema, este trabalho apresenta uma ideia de sistema *IoT* (Internet das Coisas) de baixo custo para a medição de grandezas elétricas, baseado no uso de uma placa de aquisição e condicionamento de sinais, de uma placa de desenvolvimento da série *ESP32*, do banco de dados *InfluxDB* e da plataforma de monitoramento *Grafana*.

Palavras-chave: Economia de energia; consumo de energia; Internet das Coisas; processamento de sinais; *ESP32*; *InfluxDB*; *Grafana*.

ABSTRACT

A good knowledge of one's own characteristics of electrical energy consumption is becoming increasingly beneficial for residential consumers. Nonetheless, tools for analyzing electrical energy are usually found in the market at high prices and offer few personalization options. Aiming to propose a solution to this problem, this work presents an idea of a low-cost Internet of Things system for measuring electrical quantities, based on the use of a signal acquisition and conditioning board, an ESP32 development board, the database InfluxDB and the monitoring and analytics platform Grafana.

Keywords: Energy savings; electrical consumption; Internet of Things; signal processing; ESP32; InfluxDB; Grafana.

LISTA DE ILUSTRAÇÕES

- Figura 1** - Número de dispositivos conectados à internet no mundo
- Figura 2** - Entidades interessadas na Internet das Coisas
- Figura 3** - Exemplo de amostragem digital de sinal
- Figura 4** - Placa de desenvolvimento *DOIT ESP32 Dev Kit V1*
- Figura 5** - Mapa dos pinos de IO da placa de desenvolvimento *Dev Kit 1*
- Figura 6** - Placa de aquisição e condicionamento e fonte de alimentação
- Figura 7** - Estágio de aquisição de corrente e filtragem
- Figura 8** - Estágio de aquisição de tensão e filtragem
- Figura 9** - Estágio somador da placa de aquisição e condicionamento
- Figura 10** - Estágio de alimentação da placa de aquisição e condicionamento
- Figura 11** - Estágio de geração da tensão de referência da placa de aquisição e condicionamento
- Figura 12** - Esquemático de calibração
- Figura 13** - Reta gerada pela função *polyfit* para a relação entre tensão real e palavras digitais
- Figura 14** - Reta gerada pela função *polyfit* para a relação entre corrente real e palavras digitais
- Figura 15** - Faixa de transição em filtros reais
- Figura 16** - Ferramenta ngrok operando um túnel para a rede local
- Figura 17** - Configuração da alimentação de gráfico na plataforma *Grafana*
- Figura 18** - Configuração de alerta por meio do aplicativo Telegram na plataforma *Grafana*
- Figura 19** - Configuração de alerta para medida monitorada pela plataforma *Grafana*
- Figura 20** - Esquemático da configuração utilizada para análise do sistema
- Figura 21** - Curva de potência ativa da carga resistiva gerada pelo *Grafana*
- Figura 22** - Alertas gerados pelo *Grafana* e recebidos pelo *Telegram*
- Figura 23** - Curvas de potência da carga capacitiva gerada pelo *Grafana*
- Figura 24** - Curva do fator de potência da carga capacitiva gerada pelo *Grafana*
- Figura 25** - Curvas de potência do motor a vazio gerada pelo *Grafana*
- Figura 26** - Curva do fator de potência do motor a vazio gerada pelo *Grafana*

Figura 27 - Curva de corrente eficaz da fonte de alimentação de laptop gerada pelo Grafana

Figura 28 - Curva de potência ativa da fonte de alimentação de laptop gerada pelo Grafana

Figura 29 - Esquemático completo da placa de aquisição e condicionamento

Figura 30 - Dashboard durante monitoramento de carga resistiva

Figura 31 - Dashboard durante monitoramento de motor elétrico a vazio

Figura 32 - Dashboard durante monitoramento de fonte de alimentação

Figura 33 - Dashboard durante monitoramento de carga capacitiva

LISTA DE QUADROS E TABELAS

Tabela 1 - Relação entre valores reais de tensão e palavras digitais

Tabela 2 - Relação entre valores reais de corrente e palavras digitais

Quadro 1 - Valores de tarifação do Grupo B1 residencial da CEMIG

Quadro 3 - Exemplo de organização de dados em bancos de dados do tipo *SQL*

Quadro 4 - Exemplo de organização de dados no *InfluxDB*

Quadro 5 - Resultados para carga resistiva

Quadro 6 - Resultados para carga capacitiva

Quadro 7 - Resultados para motor a vazio

Quadro 8 - Resultados para fonte de alimentação de laptop

LISTA DE ABREVIATURAS E SIGLAS

ANEEL – Agência Nacional de Energia Elétrica

CEMIG - Companhia Energética de Minas Gerais

IEEE - *Institute of Electrical and Electronics Engineers* (Instituto de Engenheiros Eletricistas e Eletrônicos)

ADC – *Analog to Digital Converter* (Conversor Analógico-Digital)

DAC – *Digital to Analog Converter* (Conversor Digital-Analógico)

AD – Analógico-Digital

CPU – Central Processing Unit

IOT – *Internet of Things* (Internet das coisas)

SOC – *System on Chip* (Sistema em um chip)

IDE – *Integrated Development Environment* (Ambiente Integrado de Desenvolvimento)

IP – *Internet Protocol*

NUPEP – Núcleo de Eletrônica de Potência

ADC – *Analog to Digital Converter* (Conversor Analógico-Digital)

DAC – *Digital to Analog Converter* (Conversor Digital-Analógico)

CPU – *Central Processing Unit* (Unidade Central de Processamento)

API – *Application Programming Interface* – Interface de programação de aplicações

TSDB – *Time series database* (Banco de dados de Séries Temporais)

OSS – *Open Source Software* (Software de código aberto)

IDE – *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado)

RMS – *Root mean square* (valor eficaz)

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivo.....	13
2 DESENVOLVIMENTO.....	14
2.1 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1.1 Internet das coisas (IoT).....	14
2.1.2 Análise de grandezas elétricas por meio digital	17
2.2 METODOLOGIA	19
2.2.1 Microcontrolador – ESP 32	20
2.2.2 Placa de condicionamento de sinais	23
2.2.3 Conversão de valores digitais para unidades elétricas	28
2.2.4 Configuração da amostragem analógico-digital	32
2.2.5 Banco de dados <i>InfluxDB</i>	34
2.2.6 Software	38
2.2.7 Grafana – Plataforma de visualização dos dados	42
2.3 RESULTADOS E DISCUSSÕES	44
2.3.1 Carga resistiva	45
2.3.2 Carga resistiva em paralelo com carga capacitiva	48
2.3.3 Motor a vazio.....	49
2.3.4 Fonte alimentando <i>laptop</i>	51
3 CONCLUSÃO.....	54
4 REFERÊNCIAS.....	55
ANEXO A.....	58
APÊNDICE A.....	59
APÊNDICE B.....	61
APÊNDICE C	64

1 INTRODUÇÃO

No cenário brasileiro atual, torna-se cada vez mais interessante aos consumidores de energia elétrica o bom conhecimento das próprias características de consumo, o que é evidenciado por diferentes medidas adotadas por órgãos reguladores e por concessionárias de energia elétrica, que começam a oferecer modalidades diferentes de tarifação de acordo com o período e a quantidade de energia consumida. Além disso, com o desenvolvimento da geração distribuída de energia e a diminuição do valor das instalações de microgeração de energia solar, passa a ser vantajosa ao consumidor comum a aquisição de um sistema próprio de geração, o que requer um bom conhecimento da curva de consumo de sua residência para que seja feito um dimensionamento adequado do sistema.

Em relação à tarifação pelas concessionárias, a Resolução Normativa Nº 733 da ANEEL, publicada em 06 de setembro de 2016, estabeleceu condições para a aplicação da chamada tarifa horária branca, definida na Resolução Normativa Nº 414, de 9 de setembro de 2010, que representa uma nova modalidade tarifária onde o valor cobrado é diferenciado conforme o período de consumo elétrico. O objetivo desta nova tarifa é fornecer aos consumidores residenciais valores reduzidos pelo consumo de energia elétrica em períodos onde as distribuidoras de energia apresentam menores demandas (período fora de ponta), em contrapartida a um aumento do valor da tarifa para períodos de demandas intermediárias e altas (período intermediário e período de ponta). Logo, nos períodos de menor demanda por parte da concessionária, o valor do consumo de energia elétrica é mais barato ao consumidor do que o valor cobrado na tarifa convencional (CEMIG, 2018).

Desta maneira, do ponto de vista do consumidor, a tarifa é vantajosa caso seja possível ao mesmo a distribuição do consumo de energia elétrica na residência em períodos fora de ponta, evitando o uso de equipamentos de alto consumo, como chuveiros, ferros de passar roupa, aquecedores e ar condicionado durante o período de ponta. Já para as distribuidoras de energia elétrica, a tarifa pode ser vantajosa, pois, com uma grande adesão à nova modalidade pelos consumidores, a curva de carga apresentaria menores picos nos horários de ponta, deslocando o consumo para períodos onde a rede de transmissão opera com capacidade ociosa (CEMIG, 2018).

Os valores da tarifa branca, assim como os horários de ponta e horários fora de ponta, são definidos pelas concessionárias de cada região. No contexto da área de concessão da CEMIG, o período de ponta é definido como o horário entre 17h e 20h, portanto o intermediário é delimitado pelo horário entre 16h e 17h e entre 20h e 21h. Os valores de cada período para consumidores residenciais pode ser verificado no Quadro 1, onde são comparados os valores da tarifa residencial B1 convencional e da tarifa residencial B1 branca estabelecidos pela concessionária, em R\$/kWh.

Quadro 1 - Valores de tarifação do Grupo B1 residencial da CEMIG

Modalidades	Bandeira verde	Bandeira amarela	Bandeira vermelha 1	Bandeira vermelha 2
Convencional	0,58684	0,59684	0,61684	0,63684
Branca ponta	1,13617	1,14617	1,16617	1,18617
Branca intermediária	0,73035	0,74035	0,76035	0,78035
Branca fora ponta	0,47923	0,48923	0,50923	0,52923

Fonte: Adaptado de CEMIG (2018)

Como citado acima, os valores apresentados na tabela evidenciam os objetivos da Resolução 733, onde é possível notar um valor reduzido de tarifação para períodos fora da ponta e valores elevados para períodos de ponta na tarifa branca.

Identificam-se também oportunidades relacionadas à própria geração de energia pelos próprios consumidores. A geração distribuída de energia elétrica pode ser caracterizada com a geração de energia a partir de pequenas centrais geradoras, a partir de fontes renováveis ou mesmo fósseis, próxima aos centros de consumo (ANEEL, 2016). Com a geração própria, o consumidor pode produzir a energia necessária para alimentar sua residência e “emprestar” o excedente à sua concessionária de energia elétrica.

O consumidor brasileiro passou a poder gerar sua própria energia elétrica a partir de geração própria com a entrada em vigor na Resolução Normativa N°482 da ANEEL, de 2012, que estabelece condições gerais para a instalação de sistemas de microgeração e minigeração distribuída de energia elétrica. A resolução estabelece um sistema de compensação de energia elétrica, onde a energia ativa injetada na rede pelo consumidor é cedida à distribuidora local em troca de um crédito em

quantidade de energia ativa (kWh), a ser utilizado para abater o consumo em outro posto tarifário (ANEEL, 2016).

Caso a intenção da instalação de micro ou minigeração própria seja suprir a sua demanda, verifica-se a conveniência do domínio da curva de carga da residência com o propósito de se alcançar uma boa relação entre o gerado pela instalação e o consumido pela residência.

Com base nestas opções, constata-se a utilidade do bom domínio do consumidor das próprias características de consumo, com o propósito do mesmo poder usufruir das novas possibilidades oferecidas pelo mercado de energia elétrica. Entretanto, a medição de consumo nas residências é geralmente feita por equipamentos isolados, que tem como função apenas a medição do consumo de energia em determinados períodos de tempo, o que impede um levantamento preciso da curva de potência da residência. Para análises mais profundas, há a opção da utilização de medidores inteligentes, mas estes são encontrados a preços elevados no mercado, o que os tornam de difícil acesso para o consumidor geral (valores pesquisados em lojas virtuais).

1.1 Objetivo

Diante do apresentado, este trabalho propõe o desenvolvimento de um medidor de potência ativa, de baixo custo, que possibilite ao usuário o acompanhamento da curva de potência de uma carga, ou conjunto de cargas, durante longos períodos de tempo. Com a intenção de ser uma ferramenta de análise mais profunda, acrescenta-se ao objetivo principal a medição de outras grandezas elétricas, como a tensão eficaz fornecida pela concessionária, a corrente eficaz injetada nas cargas observadas, o fator de potência, a potência aparente e a potência reativa de cargas.

Requisita-se também que o sistema ofereça ao usuário um fácil acesso à informação gerada, através da internet e por meio de representações gráficas, de modo a poder ser considerado um dispositivo parte da *IoT*.

2 DESENVOLVIMENTO

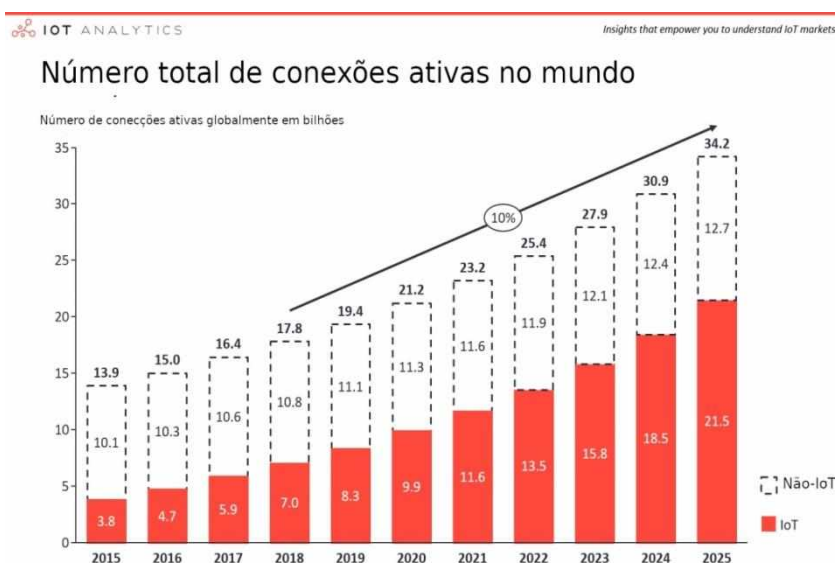
Nesta parte do trabalho serão apresentadas as atividades desempenhadas durante o desenvolvimento do mesmo. Será feita uma exposição de fundamentos teóricos dos temas pertinentes à construção do sistema, as ferramentas empregadas para se alcançar o objetivo proposto e os resultados obtidos pelo monitor de consumo.

2.1 FUNDAMENTAÇÃO TEÓRICA

As principais características do sistema proposto neste trabalho são a sua capacidade de se comunicar via *internet* e de medir e calcular grandezas elétricas. Desta maneira, esta seção apresentará de modo geral o que se entende por *Internet das Coisas*, assim como dará uma contextualização do seu papel atual no mundo. Será feita em seguida uma conceituação técnica acerca da aquisição digital de sinais, de modo a explorar seus requisitos, características e sua aplicação no cálculo de grandezas elétricas.

2.1.1 Internet das coisas (IoT)

A *Internet das Coisas*, também chamada de *Internet de tudo* ou *Internet Industrial*, é reconhecida atualmente como umas das áreas mais importantes da tecnologia e vem ganhando atenção em uma grande variedade de mercados (LEE, LEE, 2015). A presença dos dispositivos chamados *IoT* têm aumentado e tende a crescer ainda mais nos próximos anos. O número destes dispositivos, ativos e conectados à rede, já ultrapassa 7 bilhões no mundo, não incluindo celulares, tablets e linhas de telefone, sendo esperado que este número possa alcançar 22 bilhões em 2025. No gráfico da Figura 1 pode-se observar a tendência de expansão do número de dispositivos conectados à internet nos próximos anos, onde nota-se uma taxa de crescimento estimada em 10% ao ano.

Figura 1 - Número de dispositivos conectados à internet no mundo

Fonte: Adaptado de LUETH (2018)

Este crescimento traduz-se também na economia, uma vez que se estima que o mercado de sistemas *IoT*, que em 2018 ultrapassa 151 bilhões de dólares, pode chegar a alcançar 1567 bilhões de dólares em 2025 (LUETH, 2018).

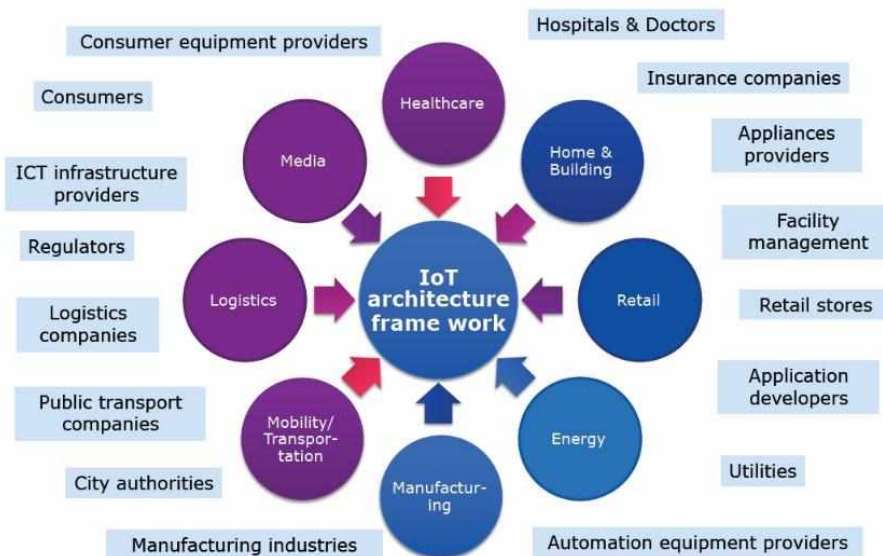
Apesar de ser presente e largamente utilizado pelo mercado e em pesquisas e estudos, o termo *Internet das Coisas* ainda não possui um consenso quanto a sua definição precisa, devido ao fato de ser um conceito relativamente recente e em constante atualização. A IEEE descreve a *Internet das Coisas* como “Uma rede de itens – cada um com sensores embutidos – que são conectados à internet”. (IEEE, 2014). A empresa HP defende que “A Internet das Coisas se refere à identificação e *Internetização* de objetos cotidianos, permitindo interação humana e controle destas “coisas”, de qualquer lugar no mundo, assim como a interação entre dispositivos sem a necessidade de envolvimento humano” (MIESSLER, 2014). Mattern e Floerkemeier (2010) conceituam a *IoT* de forma ainda mais ampla, ao defenderem que “a Internet das Coisas representa uma visão onde a internet se estende ao mundo real, abrangendo objetos cotidianos”.

De acordo com estas diferentes definições, podemos considerar como parte da *Internet das Coisas* qualquer item cotidiano que não tenha como propósito original a conexão com outros objetos ou a interação humana por meio da internet, mas que passa a ter esta capacidade por meio da adição de sensores e de

comunicação com a internet. Desta maneira, pode-se imaginar que qualquer objeto pode ser modificado, a fim de possibilitar a interação do mesmo a outros objetos ou pessoas por meio da internet, abrindo inúmeras possibilidades de controle e de captação de informações do dia a dia de modo automatizado.

Aplicações para a *IoT* podem ser encontradas em diferentes indústrias e atividades. LEE e LEE (2015) identificam três categorias principais para a aplicação da Internet das Coisas: monitoramento e controle, *big data* e análises de mercado, e compartilhamento e colaboração de informações. A primeira categoria, monitoramento e controle, representa a coleta de dados em equipamentos para a análise de desempenho, de consumo de energia e das condições ambientais que o equipamento está inserido, possibilitando a tomada de decisão pelo usuário ou pelo próprio sistema, por meio de atuadores. A segunda categoria, *big data* e análises de mercado, faz uso dos dispositivos *IoT* e de máquinas equipadas com sensores para obter grandes quantidades de dados sobre consumidores, alimentando ferramentas de análise com o objetivo de descobrir e resolver problemas do produto ou negócio. A terceira categoria, compartilhamento de informações, diz respeito ao possível compartilhamento de informações entre o sistema *IoT* e pessoas, ou entre o sistema e outros objetos, a fim de comparar comportamentos esperados com os comportamentos reais, obtidos por sensores.

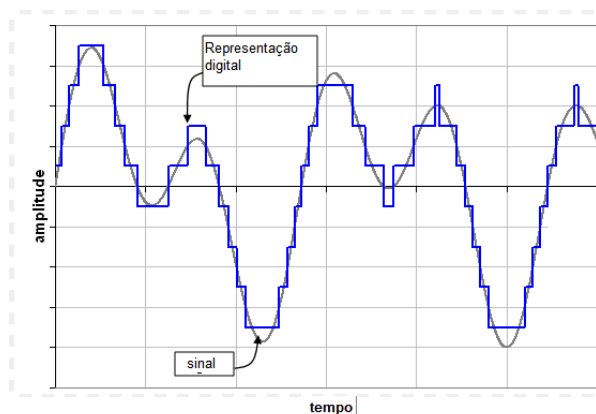
Com base nestas categorias, observa-se que a conexão de objetos à internet tem aplicações em quase qualquer campo do cotidiano e da indústria. O IEEE evidencia em seu relatório *Towards a definition of the Internet of Things (IoT)* (2015) esta constatação, considerando como entidades interessadas na Internet das Coisas, chamados de *stakeholders*, desde hospitais até autoridades públicas, conforme mostrado na Figura 2, que consta no relatório.

Figura 2 - Entidades interessadas na Internet das Coisas

Fonte: IEEE (2015)

2.1.2 Análise de grandezas elétricas por meio digital

Através da medição digital da corrente e da tensão na entrada de uma carga, pode-se obter diferentes grandezas elétricas, como os valores eficazes da corrente e da tensão, as potências ativa, reativa e aparente e o fator de potência. A medição digital destes valores dá-se através da amostragem de valores instantâneos em intervalos regulares, que são então quantizados em valores discretos. Na figura 3 é mostrado o exemplo de um processo de amostragem e quantização, onde o sinal original, analógico, é representado pela linha preta e os valores discretos pela linha azul.

Figura 3 - Exemplo de amostragem digital de sinal

Fonte: Adaptada do website Quantization (2018)

Se seguido o teorema de *Nyquist–Shannon*, que determina que a frequência de amostragem deve ser superior a duas vezes a frequência do sinal de entrada, o sinal original pode ser perfeitamente recuperado após a amostragem (RNP, 2002).

A obtenção e cálculo de grandezas elétricas por meio da amostragem digital possui algumas vantagens em relação à medida analógica, sendo a maior delas a possibilidade de cálculos precisos de potência e outras grandezas mesmo em condições de não-linearidade, o que permite uma análise mais profunda das diferentes frequências harmônicas presentes no sinal. Além disso, destacam-se outras vantagens como a facilidade de calibração do sistema de medição, que pode ser feita através da alteração de parâmetros pelo usuário do sistema, e a precisão da multiplicação digital, que não é afetada por efeitos de linearidade encontrados em multiplicadores analógicos (STEFAN, 1999). A amostragem digital por meio da conversão AD de sinais foi impulsionada nos últimos anos pelo avanço de técnicas de conversão, pela melhora no processamento e pela diminuição do preço dos sistemas de aquisição ligados aos conversores.

Apesar das vantagens apresentadas, deve-se atentar para a existência de algumas fontes de erros ao se utilizar a amostragem digital de sinais, aplicando-se correção onde for possível. Algumas dessas fontes de erro têm origem nos circuitos analógicos de aquisição de sinais, no processo de amostragem, na conversão AD e nos cálculos digitais (FUCHS, 2006).

No circuito analógico de aquisição, encontram-se erros devido à imprecisão dos sensores de corrente e de tensão e de ruídos nos sinais, que podem ser corrigidos através da presença de filtros, da calibração do circuito e da correção de parâmetros nos cálculos da parte digital. No processo de amostragem, pode haver a presença de erros causados pela discretização do sinal, devendo-se atentar para características como o número de amostragens por janela e a frequência de amostragem, de modo a buscar uma amostra coerente com os sinais de entrada. Erros na conversão AD, causados por imprecisão do conversor e pelo deslocamento de tempo (*time-shift*) entre amostras podem ser amenizados pela utilização de conversores de boa qualidade e com uma alta taxa de amostragem (FUCHS, 2006).

No que diz respeito ao cálculo de grandezas elétricas com base em amostras digitais, deve-se discretizar as equações utilizadas, já que a aquisição dos sinais digitais se dá por meio da amostragem instantânea e discreta de valores, enquanto as equações originais são relativas ao cálculo de sinais contínuos. O cálculo da

corrente e da tensão eficazes do sistema, de forma discretizada, podem ser observadas nas equações 1 e 2. Na equação 3 é exibida a equação para o cálculo da potência ativa (FUCHS, 2006).

$$V_{rms} = \sqrt{\frac{1}{N} \sum_{k=1}^N V_k^2} \quad (1)$$

$$I_{rms} = \sqrt{\frac{1}{N} \sum_{k=1}^N I_k^2} \quad (2)$$

$$P = \frac{1}{N} \sum_{k=1}^N V_k I_k \quad (3)$$

Onde V_k e I_k representam as amostras dos sinais de tensão e corrente, N corresponde ao número de amostragens da janela de amostragem, V_{rms} e I_{rms} correspondem aos valores de tensão e corrente eficazes e P ao valor de potência ativa.

Com os valores de tensão e correntes eficazes e de potência ativa, é possível calcular os valores de potência reativa e aparente e o fator de potência do sistema. As equações para a obtenção dessas grandezas são mostradas nas equações 4, 5 e 6.

$$S = V_{rms} I_{rms} \quad (4)$$

$$Q = \sqrt{(S^2 - P^2)} \quad (5)$$

$$fp = \frac{P}{S} \quad (6)$$

Onde S , Q e fp correspondem, respectivamente, à potência ativa, potência reativa e fator de potência.

2.2 METODOLOGIA

O sistema desenvolvido por este trabalho é composto por diferentes ferramentas e dispositivos, que juntos buscam alcançar o objetivo de monitorar o consumo de energia de uma carga. Desta maneira, o processo de construção do

sistema foi dividido em etapas, que serão expostas individualmente nos tópicos seguintes. Serão apresentados: uma visão geral do microcontrolador utilizado no trabalho, o ESP32, assim como suas características técnicas relevante; a placa de aquisição e condicionamento de sinais utilizados para a conversão dos valores de tensão e corrente para níveis adequados à leitura do microcontrolador; a metodologia empregada para conversão dos valores digitais obtidos para valores reais, assim como as calibrações necessárias aplicadas aos valores obtidos; a definição de valores relacionados à amostragem dos sinais pelo microcontrolador; uma apresentação do *InfluxDB*, banco de dados utilizado para o armazenamento dos dados, e da configuração necessária para sua operação; uma visão geral do *software* desenvolvido para a aquisição, processamento e armazenamento de sinais, carregado no microcontrolador; e uma apresentação da plataforma gráfica utilizada para a análise dos dados pelo usuários, o *Grafana*.

2.2.1 Microcontrolador – ESP 32

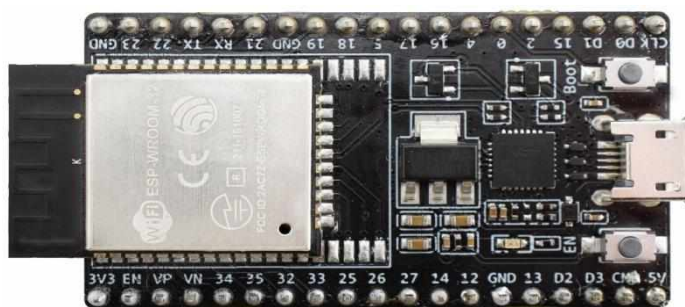
Visto que o objetivo do trabalho é o desenvolvimento de um sistema de medição de baixo custo, com conexão à internet e que seja facilmente replicável, buscou-se um microcontrolador de preço acessível, com boa disponibilidade no mercado e que, se possível, possuísse em um único *chip* os periféricos necessários para o trabalho (conversor AD e *Wi-Fi*). De forma geral, o microcontrolador escolhido deveria também ser capaz de realizar as tarefas de conversão digital-analógica dos sinais, de cálculo das grandezas de interesse e de comunicação com o banco de dados responsável por armazenar as grandezas calculadas.

Por atender a todos esses requisitos, foi escolhido um microcontrolador da série *ESP32*, levando também em consideração sua grande comunidade de usuários, assim como a possibilidade de integração à *IDE Arduino*, o que proporciona ao usuário um ambiente intuitivo de programação e comunicação, além do acesso a um grande número de bibliotecas.

O *ESP32* é uma série de microcontroladores do tipo sistema-em-um-chip (*SoC – System on a chip*), desenvolvidos pela *Espressif Systems*, que têm como objetivo serem uma solução integrada para aplicações de *IoT* baseadas em *Wi-Fi* ou *Bluetooth* (ESPRESSIF, 2018). O sistema reúne em um mesmo *chip* um

microprocessador *dual-core Xtensa LX6*, *Wi-Fi 2.4GHz*, *Bluetooth 4.2* e conversores AD de 12 bits, dentre outros periféricos (ESPRESSIF, c2016). Com a utilização do ESP32, excluiu-se a necessidade da utilização de outros módulos externos para conversão analógico-digital dos sinais e para a comunicação com o banco de dados.

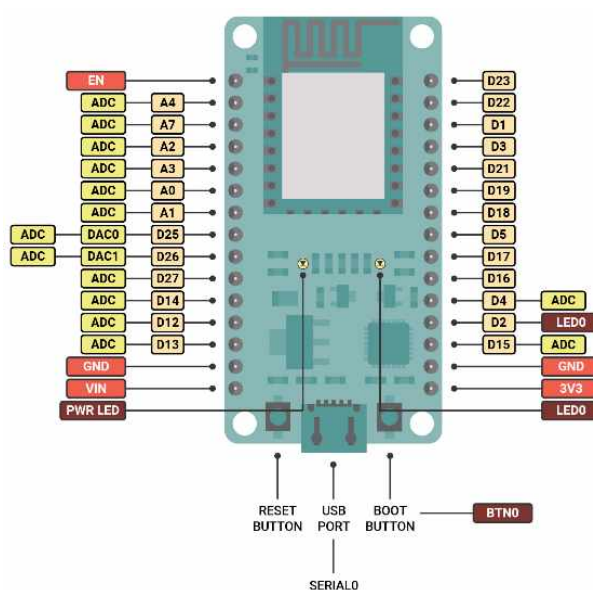
Figura 4 - Placa de desenvolvimento DOIT ESP32 Dev Kit V1



Fonte: ESPRESSIF, 2018

Foi definido para o trabalho a placa de desenvolvimento *DOIT ESP32 Dev Kit V1*, modelo da série *ESP32* facilmente encontrado em lojas brasileiras em valores próximos a 50 reais. Uma foto do *kit* pode ser vista na Figura 4. O esquemático, o mapa dos pinos (Figura 5) e demais informações sobre a placa de desenvolvimento são disponibilizados gratuitamente na *internet* através do *website* oficial da *Espressif*.

Figura 5 - Mapa dos pinos de IO da placa de desenvolvimento Dev Kit 1



Fonte: ESPRESSIF, 2018

No Quadro 2 são apresentadas as principais características técnicas da série, obtidas em seu *datasheet*.

Quadro 2 - Características técnicas da série ESP32

ESP32	
CPU	Microprocessadores Xtensa® Dual-Core 32-bit LX6 Clock de 160MHz - até 240MHz 448 KB ROM 512 KB graváveis de SRAM
Periféricos	Bluetooth v4.2/BLE ADC SAR de 12-Bit 2 x DAC 8-Bit 10 Sensores Capacitivos de Toque Sensor de temperatura 4 x barramento SPI 2 x I2S 2 x I2C 3 x UART SDIO Ethernet MAC com suporte IEEE CAN 2.0 Controle Remoto IR Motor PWM LED PWM Sensor Hall Pré-Amplificador de Baixo Consumo 36 Pinos GPIO
Tensão de Operação	2.8-3.3V
Intervalo de Temperatura de Operação	-40-125°C
Frequencia do Wireless	2.4 GHz
Protocolos Wireless	802.11 b/g/n/e/i HT40
Segurança Wireless	WFA/WPA/WPA2/WAPI
Encriptação	AES/HASH(SHA-2)/RSA/ECC/RNG
Dimensões	26.0mm x 18.0mm

Fonte: *Datasheet* da série, ESP32 (2018)

Definida a placa, foram utilizados os pinos 34 e 35, referentes a dois canais do ADC1, para a aquisição dos sinais analógicos recebidos da placa de aquisição e condicionamento de sinais, onde são lidos os valores das medidas de tensão e corrente, respectivamente. A alimentação da placa foi feita através da entrada USB.

A partir deste ponto, o termo *ESP32* será utilizado neste trabalho para representar a placa de desenvolvimento utilizada no sistema.

2.2.2 Placa de condicionamento de sinais

Como os níveis de tensão e de corrente da rede são muito elevados para serem lidos diretamente pelos *inputs* do microcontrolador, faz-se necessário a conversão destas grandezas para um nível aceitável. Para isto, foi utilizado um módulo de aquisição e condicionamento de sinais projetado por Marcelo Rodrigues S. Brito, Henrique J. Avelar e Ernane A. A. Coelho e disponibilizada pelo NUPEP, que tem como função, além de converter a tensão e corrente para níveis baixos de tensão (0 V – 3.3 V), filtrar altas frequências a fim de evitar ruídos e efeitos de *aliasing*. O conjunto placa de aquisição-fonte pode ser visto na Figura 6. A alimentação do módulo é feita em 220V e o mesmo possui duas entradas de tomada, uma destinada à alimentação da carga e uma para a conexão com a rede.

Figura 6 - Placa de aquisição e condicionamento e fonte de alimentação



Fonte: Autoria própria (2018)

A configuração do módulo pode ser vista como uma composição de diferentes estágios, onde cada um é responsável por um objetivo, sendo eles a aquisição de sinais, a filtragem de sinais, a soma de níveis contínuos de tensão e a alimentação do circuito.

O estágio de aquisição do módulo é responsável pela conversão dos valores de tensão e corrente, que é feita através de dois sensores do tipo *Hall*, sendo um

transdutor de corrente LA 55-P/SP1 e um transdutor de tensão LV 25-P/SP2 para a medição de corrente e tensão, respectivamente.

O transdutor de corrente converte diretamente a corrente nas espiras de seu circuito primário em uma corrente reduzida no circuito secundário, com uma relação K_n de 1:2000. Como a corrente eficaz máxima suportada pelo transdutor é de 50A, valor elevado se comparado à corrente injetada nas cargas a serem testadas pelo protótipo, foram feitas 5 voltas com o cabo de alimentação da carga no circuito primário, multiplicando assim proporcionalmente o campo magnético e a corrente gerada no secundário, de maneira a se definir um valor máximo de 10A para o dispositivo. Desta maneira, aumenta-se a resolução da conversão analógico-digital, já que se mantém o número de *bits* (12) das palavras digitais, diminuindo a escala de corrente.

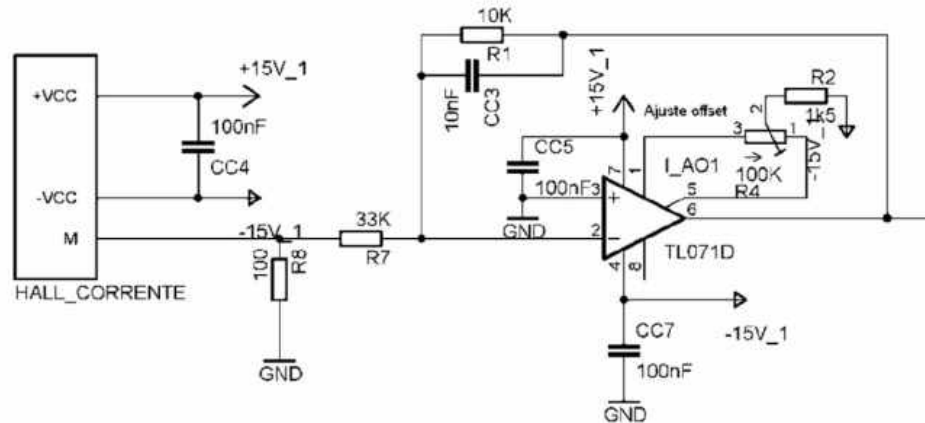
O transdutor de tensão, apesar de ser responsável por converter valores de tensão de níveis elevados em níveis aceitáveis pelo microprocessador, também é um sensor de corrente. Portanto, a tensão de entrada deve ser aplicada em um resistor de medição, responsável por converter a tensão em uma corrente a ser injetada no primário do transdutor. Após a queda de tensão no resistor de medição, a corrente injetada no primário do transdutor é convertida para o secundário com uma relação de K_n de 2500:1000.

Tendo que a entrada do primário do transdutor de tensão suporta correntes eficazes de até 10mA, a definição da tensão máxima admitida pelo estágio de aquisição do módulo foi dada através da escolha do resistor de medição. Como o resistor utilizado no módulo tem valor de 50k Ω , pode-se deduzir através da lei de Ohm que a tensão máxima admitida pelo módulo é de 500V.

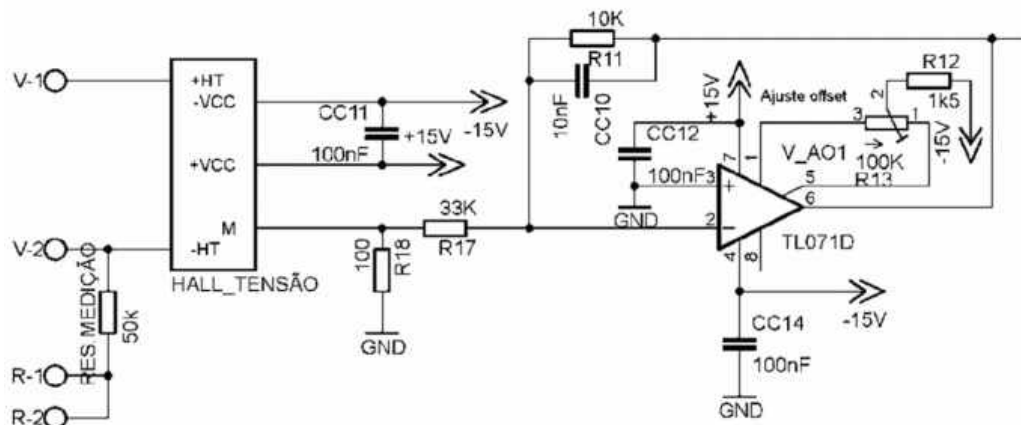
$$V = I \times R = 10mA \times 50k\Omega = 500V \quad (7)$$

Realizada a conversão dos valores de tensão e de corrente, os sinais são tratados em um primeiro estágio de condicionamento que contém um filtro de passa-baixa ativo de um pólo com o objetivo de filtrar frequências elevadas em relação às frequências de interesse do sinal da entrada, evitando assim efeitos de *aliasing*. O filtro possui um AOP TL071D, capacitores e resistores.

Os estágios contendo a aquisição dos sinais e o filtro passa-baixa é exibido nas Figuras 7 e 8.

Figura 7 - Estágio de aquisição de corrente e filtragem

Fonte: Disponibilizada pelo NUPEP

Figura 8 - Estágio de aquisição de tensão e filtragem

Fonte: Disponibilizada pelo NUPEP

Para esta configuração de filtro passa-baixa ativo, tem-se a função de transferência exibida na equação 8. (NILSSON, RIEDEL, 2014).

$$H(s) = -K \times \frac{w_c}{s + w_c} \quad (8)$$

Onde é possível constatar que para frequências superiores à frequência de corte (w_c) o sinal é atenuado.

O valor da frequência de corte do filtro, necessária para a definição das características de amostragem do sistema, pode ser calculado conforme equação 9.

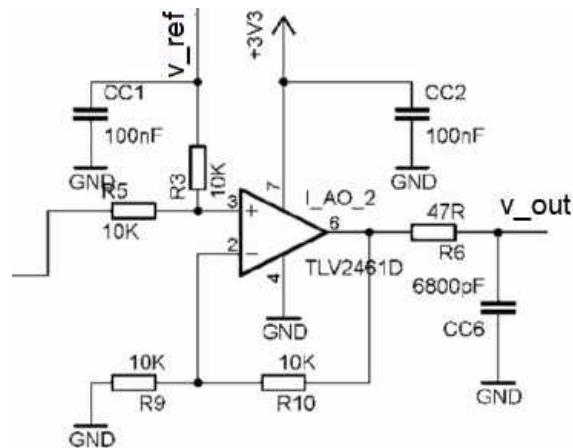
$$f_c = \frac{1}{2\pi \times R_2 \times C} = \frac{1}{2\pi * 10k\Omega * 10nF} = 1591,5Hz \quad (9)$$

Onde f_c representa a frequência de corte em unidades de Hertz.

No esquemático do estágio de filtragem nota-se também a existência de *trimpots* para o ajuste do *offset* dos AOPs, que possibilitam a calibração dos valores de tensão e corrente adquiridos pelo estágio de aquisição, a fim de garantir que os valores da saída do mesmo sejam centrados em 0V.

Após o filtro *anti-aliasing*, os sinais são tratados por estágios somadores onde são somados a um nível contínuo de tensão. Como os sinais medidos no primeiro estágio variam entre valores positivos e negativos, o estágio somador é necessário para que os sinais sejam centrados em um valor positivo, garantindo que a saída apresente sempre tensão positiva, visto que o conversor AD do microprocessador aceita somente valores desta natureza.

Figura 9 - Estágio somador da placa de aquisição e condicionamento



Fonte: Disponibilizado pelo NUPEP

O estágio somador é composto por um (01) AOP TLV2461, resistores e capacitores e foi realizado conforme o esquemático apresentado na Figura 9. Através da análise do circuito e considerando ideais condições, podemos deduzir que o valor da tensão de saída equivale de fato ao sinal de entrada somado ao valor de referência conforme equações 10, 11 e 12.

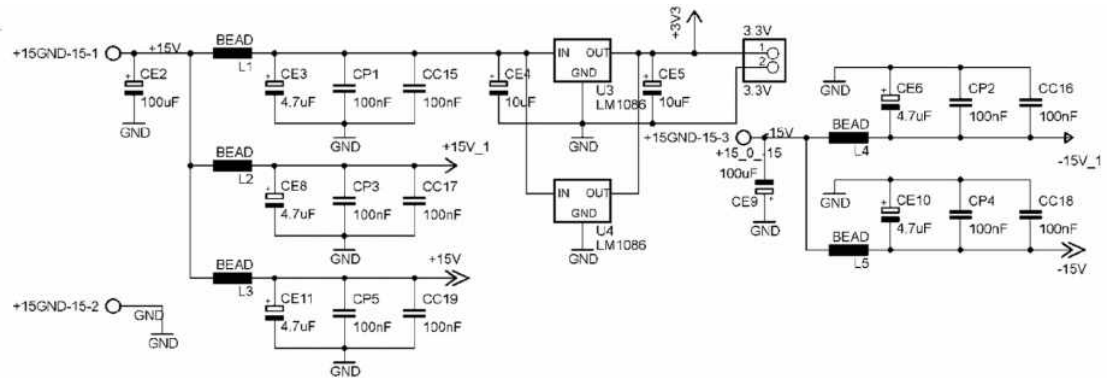
$$\frac{v_i(t) - \frac{1}{2}v_{out}(t)}{R_5} = \frac{\frac{1}{2}v_{out}(t) - V_{ref}}{R_3} \quad (10)$$

$$\frac{v_i(t) - \frac{1}{2}v_{out}(t)}{10k\Omega} = \frac{\frac{1}{2}v_{out}(t) - V_{ref}}{10k\Omega} \quad (11)$$

$$v_{out}(t) = v_i(t) + V_{ref} \quad (12)$$

O estágio de filtro e de alimentação do circuito, responsável por suprir a alimentação dos circuitos de condicionamento de sinais e por filtrar eventuais ruídos, é composto por capacitores, *beads* de ferrite, reguladores de tensão e um AOP. O estágio deve ser alimentado com uma tensão contínua de 15V e -15V e fornece aos circuitos de condicionamento e aquisição níveis de 3.3V, +15V e -15V, além do valor de referência de tensão contínua ao estágio somador. O esquemático do estágio pode ser visualizado na Figura 10.

Figura 10 - Estágio de alimentação da placa de aquisição e condicionamento

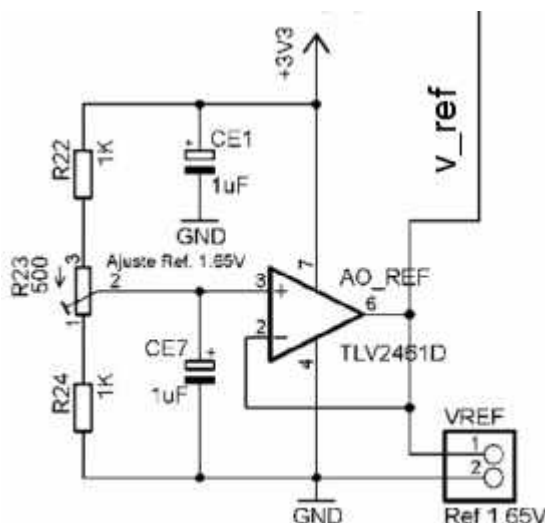


Fonte: Disponibilizado pelo NUPEP

Na entrada do estágio, a tensão de alimentação de +15V é filtrada através dos *beads* de ferrite e dos capacitores e então convertida a um nível de 3.3V através de um regulador de tensão LM1086 de saída 3.3V. A tensão de -15V é também filtrada através de *beads* de ferrite e de capacitores.

A última etapa do estágio de alimentação é o circuito responsável por fornecer a tensão de referência V_{ref} ao estágio de condicionamento, e é composto por um AOP TLV2451, resistores, capacitores e um *trimpot*. O esquemático deste estágio pode ser verificado na Figura 11. Através da variação do *trimpot* é possível ajustar o valor de referência conforme o adequado para o conversor AD do microcontrolador, sendo o valor usual 1.65V.

Figura 11 - Estágio de geração da tensão de referência da placa de aquisição e condicionamento



Fonte: Disponibilizado pelo NUPEP

O módulo de aquisição e condicionamento de sinais foi calibrado manualmente, através da variação dos *trimpots* e verificação dos valores de tensão em multímetros de alta precisão. O valor de referência foi ajustado em 1.65V, valor que representa o centro da escala de tensão do módulo. O esquemático completo do módulo pode ser observado no Anexo A deste documento.

As saídas de tensão e corrente convertidas foram conectadas aos pinos 34 e 35 do microcontrolador ESP32, respectivamente.

2.2.3 Conversão de valores digitais para unidades elétricas

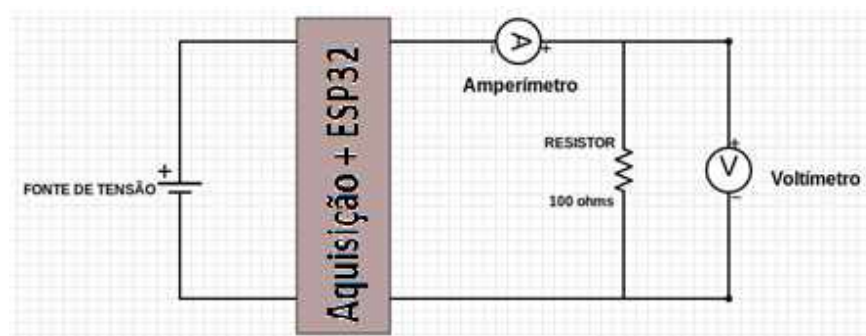
O conversor AD do ESP32, responsável pela amostragem dos sinais tratados pela placa de aquisição e condicionamento de sinais, lê os sinais como palavras digitais de 12-bits, ou seja, de 0 a 4096 em números decimais. Desta maneira, é necessário que seja feita a conversão dos valores digitais para valores reais, em unidades de *volts* e *ampères*. Esta conversão é feita através da subtração de um valor de *offset*, tendo em vista que o valor digital não é centrado em zero, e da multiplicação de um ganho (responsável pela conversão da amplitude do sinal) aos valores digitais.

Além da necessidade da conversão, é importante que a relação entre valores digitais e reais seja obtida para que seja feita uma boa calibração dos valores em

relação ao conversor AD do microcontrolador. No *datasheet* da plataforma é informado que entre *chips* diferentes da série *ESP32* pode haver uma diferença de $\pm 6\%$ nos valores lidos pelos conversores AD, o que resultaria em erros de conversão se os valores digitais não forem devidamente calibrados (ESPRESSIF, 2018).

Com a finalidade de obter a relação entre os valores digitais e reais, foram impostos diferentes valores de tensão e corrente contínua a uma carga resistiva, enquanto os respectivos valores digitais foram colhidos pelo conversor AD do microcontrolador, conforme esquemático visto na Figura 12. Para cada valor de tensão e corrente imposto à carga foi anotado a medida da grandeza real através de um amperímetro e de um voltímetro.

Figura 12 - Esquemático de calibração



Fonte: Autoria própria (2018)

Feitas as ligações, foi carregado o código desenvolvido para a aquisição e cálculo da média das palavras digitais. O código completo pode ser encontrado no Apêndice A deste trabalho. Foi obtida a média aritmética de 1000 amostras de valores digitais para cada valor de tensão e corrente aplicados à carga, de modo à amenizar possíveis erros individuais de medição. Nas Tabelas 1 e 2 podem ser examinados valores reais de tensão e corrente e as médias dos valores de suas respectivas palavras digitais.

Tabela 1 - Relação entre valores reais de tensão e palavras digitais

Tensão real (V)	Tensão digital – média de 1000 amostras	Tensão real (V)	Tensão digital – média de 1000 amostras	Tensão real (V)	Tensão digital – média de 1000 amostras
-225	1405.81	-60	1693.37	105	1976.13
-210	1430.34	-45	1718.01	120	2001.63
-195	1456.50	-30	1744.38	135	2029.69
-180	1483.62	-15	1771.99	150	2058.84
-165	1508.43	0	1796.94	165	2083.72
-150	1533.63	15	1821.53	180	2109.63
-135	1564.41	30	1848.79	195	2132.57
-120	1590.95	45	1870.93	210	2159.96
-105	1615.93	60	1901.53	225	2184.67
-90	1643.40	75	1924.84		
-75	1667.20	90	1950.25		

Tabela 2 - Relação entre valores reais de corrente e palavras digitais

Corrente real (A)	Corrente digital – média de 1000 amostras	Corrente real (A)	Corrente digital – média de 1000 amostras
-2.50	1355.19	0.25	1846.20
-2.25	1399.04	0.5	1889.85
-2.00	1443.60	0.75	1934.75
-1.75	1487.56	1.00	1977.77
-1.50	1531.89	1.25	2022.08
-1.25	1582.86	1.50	2069.71
-1.00	1624.41	1.75	2114.05
-0.75	1669.11	2.00	2158.83
-0.5	1713.32	2.25	2201.11
-0.25	1758.19	2.50	2245.89
0	1800.76		

Uma vez lidos os valores resultantes da conversão analógico-digital, foi utilizado o *software* de computações matemáticas *GNU Octave* para a análise dos dados. Para a obtenção dos valores de ganho e *offset*, que relacionam os valores

digitais e analógicos, foi utilizada a função *polyfit* do *software*, que tem como finalidade a construção de curvas que se ajustam a uma série de pontos por meio do método dos mínimos quadrados (OCTAVE, c1996). A função toma como argumentos dois vetores de amostras e o grau da curva de ajuste e retorna os coeficientes do polinômio que representa a curva resultante. Como desejado um valor de ganho e um de *offset*, foi gerada uma reta, ou seja, um polinômio de grau um (1) para os pontos de tensão e corrente. As retas resultantes podem ser examinadas nas equações 12 e 13.

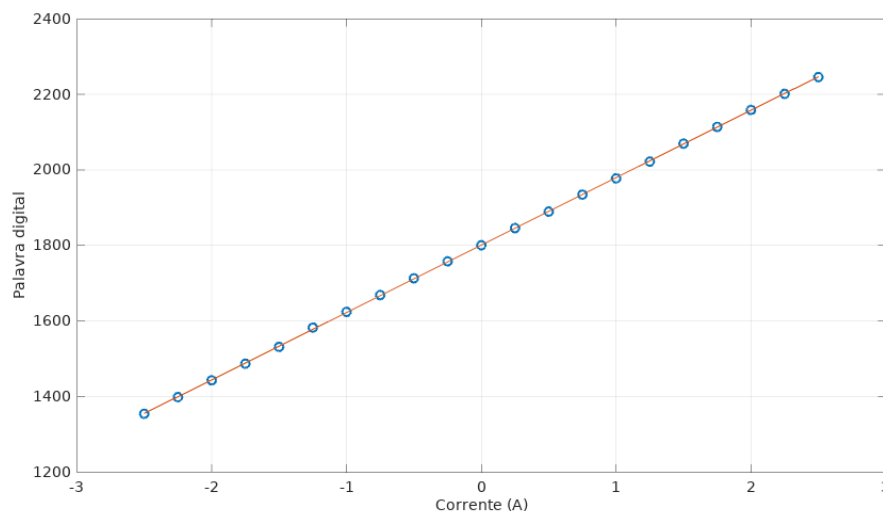
$$V_{real} = 0.5771V_{digital} - 1796.1168 \quad (12)$$

$$I_{real} = 5.6110 \cdot 10^{-3} \cdot I_{digital} - 1801.2 \quad (13)$$

Onde $V_{digital}$ e $I_{digital}$ são os valores das palavras digitais a serem convertidas, sendo V_{real} e I_{real} equivalentes aos valores reais correspondentes. Desta maneira, os valores de ganho e *offset* a serem utilizados no *software* do microcontrolador equivalem a 0.5771 e 1796.1168 para a conversão da tensão e $5.6110 \cdot 10^{-3}$ e 1801.25 para a corrente.

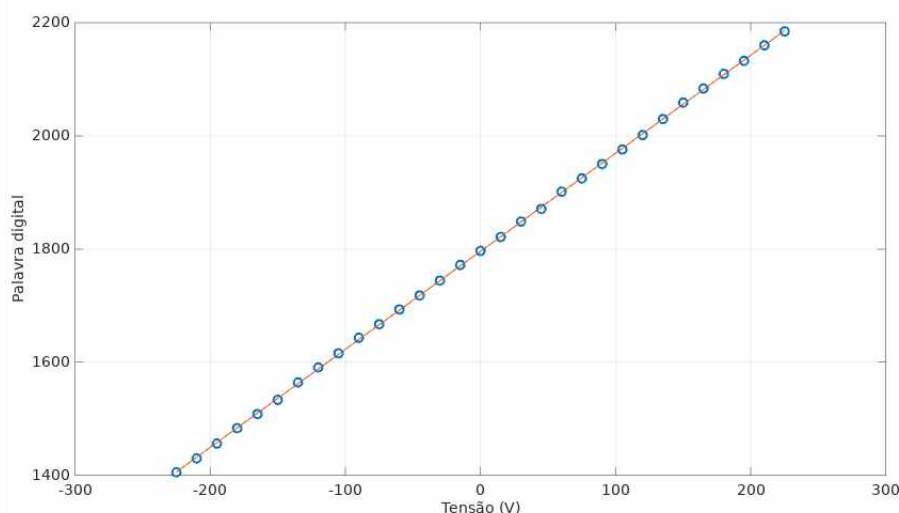
A fins de análise foram gerados gráficos para a visualização das retas em comparação com os pontos medidos. Os gráficos podem ser visualizados nas Figuras 13 e 14, onde os círculos vermelhos representam os valores obtidos experimentalmente e a reta azul equivale ao polinômio e grau 1 gerado pela função *polyfit*. Nota-se que o conversor AD teve comportamento consideravelmente linear para os valores medidos.

Figura 13 - Reto gerada pela função *polyfit* para a relação entre tensão real e palavras digitais



Fonte: Autoria própria (2018)

Figura 14 - Reto gerada pela função *polyfit* para a relação entre corrente real e palavras digitais



Fonte: Autoria própria (2018)

2.2.4 Configuração da amostragem analógico-digital

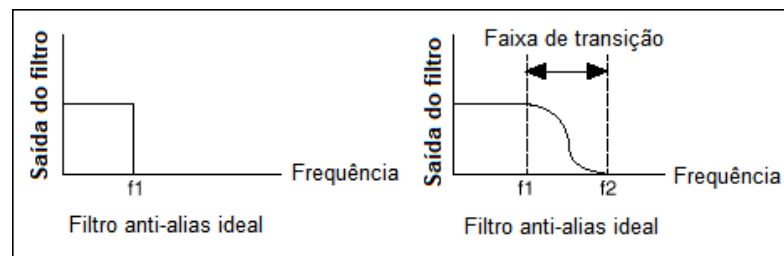
A amostragem dos sinais pelo microcontrolador, feita através do conversor AD, deve seguir algumas diretrizes em relação à definição de parâmetros como frequência de amostragem, número de amostras e número de períodos coletados para que os resultados obtidos sejam otimizados. Para a definição destes parâmetros, deve-se levar em consideração informações como a frequência do sinal analógico amostrado, a existência ou não de filtros analógicos no estágio de aquisição e a resolução desejada.

Conforme apresentado anteriormente, a placa de aquisição e condicionamento de sinais possui um filtro *anti-aliasing* do tipo passa-baixa, responsável por filtrar frequências superiores à frequência de corte do filtro. Com isto, tem-se que a taxa de amostragem utilizada pelo conversor AD deve ter valor superior a duas vezes a frequência de corte do filtro, conforme posto pelo *Teorema de Amostragem de Nyquist-Shannon*, a fim de que seja possível recuperar o sinal original.

Dado que o filtro tem um pólo sintonizado em 1591,5 Hz, idealmente a frequência de amostragem deve ser superior a duas vezes este valor, ou seja, 3183 Hz. Entretanto, como deve ser levada em consideração a natureza real do filtro, deve-se também observar a existência da faixa de transição após a frequência de

corte, onde o sinal é atenuado mas ainda tem valor relevante, conforme ilustrado pela Figura 15. Recomenda-se então que a taxa de amostragem seja superior ao valor ideal, de modo a evitar efeitos de *aliasing* de frequências superiores à frequência de corte (NATIONAL INSTRUMENTS, 2018). Desta maneira, definiu-se como taxa mínima de amostragem um valor superior a três vezes a frequência de corte, ou 4774,5 Hz.

Figura 15 - Faixa de transição em filtros reais



Fonte: Adaptado de NATIONAL INSTRUMENTS (2018)

Para que o sistema seja preparado para a realização de análises em frequência, foi definido o número de 1024 amostras por janela, já que é um requisito da *FFT* que o número de amostras seja uma potência de 2, ficando o sistema preparado para futura implementação deste algoritmo. Para definir-se a frequência de amostragem e o número de amostras a ser realizado por janela de amostragem, levou-se em consideração a técnica de amostragem coerente (*coherent sampling*), onde o número de amostras obtidas na janela equivale a um número inteiro de ciclos do sinal original. Na equação 14 pode-se analisar a relação entre as frequências do sinal e de amostragem e os números de amostras e ciclos. Recomenda-se também que o número de ciclos na janela de amostragem seja ímpar ou primo, de modo a eliminar fatores comuns com o número de amostras da janela (MAXIM INTEGRATED, 2002).

$$\frac{f_{\text{sinal}}}{f_s} = \frac{M}{N_s} \quad (14)$$

Onde f_{sinal} , f_s , M e N representam a frequência do sinal, a frequência de amostragem, o número de ciclos do sinal de entrada por janela e o número de amostras por janela.

Tendo que os sinais de tensão e corrente tem frequência fundamental de 60Hz e harmônicas múltiplas deste valor, levou-se em consideração esta frequência

de sinal. Assim, visto que o número de amostras por janela foi definido como 1024, as relações vistas nas equações 15 e 16 foram obtidas.

$$\frac{60}{f_s} = \frac{M}{1024} \quad (15)$$

$$M \times f_s = 61440 \quad (16)$$

Determinou-se então de maneira iterativa uma taxa de amostragem de 7680Hz, primeiro valor superior a 4774,5 Hz que satisfaz as condições citadas.

2.2.5 Banco de dados *InfluxDB*

Entre os objetivos do trabalho desenvolvido está a necessidade dos dados coletados serem armazenados com segurança em um servidor e serem acessíveis pela rede de *internet*, possibilitando assim ao usuário o monitoramento do sistema e a operação dos dados remotamente. O armazenamento dos dados é também um requisito devido à natureza do projeto, que visa levantar o consumo de uma ou mais cargas durante longos períodos de tempo. Por estas razões, se faz necessária a utilização de um banco de dados para este fim e que, idealmente, as informações acessíveis pelo usuário através de outras redes ou computadores além da rede local do servidor. Desta maneira, o sistema se torna independente de um servidor físico presente na mesma rede a que está conectado, possibilitando mobilidade ao dispositivo de aquisição dos dados.

De acordo com os requisitos citados no parágrafo anterior e levando em consideração a natureza dos dados obtidos sistema, que são armazenados na forma de séries temporais, buscou-se um banco de dados do tipo TSDB (*Time Series Database* – Banco de dados de Séries Temporais). Os bancos de dados do tipo TSDB são otimizados para o tratamento de dados baseados em séries temporais, que tem como características gerais a imutabilidade da informação (novos dados são sempre acrescentados, não substituídos), a mudança dos valores durante o tempo e o alto volume de dados (STEPHENS, 2018).

Após pesquisa das opções de bancos de dados do tipo TSDB disponíveis, definiu-se o *InfluxDB* para o armazenamento dos dados, por ser uma ferramenta gratuita *opensource*, devido à sua alta performance e ao suporte a plataformas de

análise e monitoramento (BERMAN, 2017). O *InfluxDB*, disponibilizado pela empresa *InfluxData*, é definida como “um banco de dados de alta performance desenvolvido especialmente para informações baseadas em séries temporais” (INFLUXDATA, 2018a). O uso do *InfluxDB* é recomendado portanto para o armazenamento de dados que tenham como referência índices temporais, ou seja, organizados de acordo de maneira “*time-stamped*” (carimbados com a data).

A organização dos dados no banco *InfluxDB* possui diferenças em relação aos bancos de dados usuais do tipo SQL, principalmente pelo fato dos dados serem indexados em relação ao tempo. Enquanto em banco de dados SQL são organizados em tabelas, colunas indexadas, colunas e linhas, os dados do *InfluxDB* são divididos em medidas (*measurements*), etiquetas (*tags*), campos (*fields*) e pontos (*points*) (INFLUXDATA, 2018b).

A título de exemplo, são apresentados nos Quadros 3 e 4 a maneira como seriam organizados os mesmos dados nos dois modelos, onde o primeiro representa uma tabela SQL e o segundo a mesma organização na estrutura InfluxDB.

Quadro 3 - Exemplo de organização de dados em bancos de dados do tipo SQL

carga_id	carga	tempo	#_Pot_ativa
1	lampadas	1429185600000000000	300
1	lampadas	1429185601000000000	450
1	lampadas	1429185602000000000	223
1	lampadas	1429185603000000000	128
2	freezer	1429185600000000000	1050
2	freezer	1429185601000000000	940
2	freezer	1429185602000000000	1110

Fonte: Adaptado de INFLUXDATA (2018)

Quadro 4 - Exemplo de organização de dados no InfluxDB

name: monitor_potencia	
tags: carga_id=1, carga=lampadas	
time	#_Pot_ativa
2015-04-16T12:00:00Z	300
2015-04-16T12:00:01Z	450
2015-04-16T12:00:02Z	223
2015-04-16T12:00:03Z	128
name: monitor_potencia	
tags: park_id=2, carga=freezer	
time	#_Pot_ativa
2015-04-16T12:00:00Z	1050
2015-04-16T12:00:01Z	940
2015-04-16T12:00:02Z	110

Fonte: Adaptado de INFLUXDATA (2018)

Nos casos acima, pode-se inferir que medidas, etiquetas, campos e pontos do InfluxDB equivalem, respectivamente, a tabelas, colunas indexadas, colunas não-indexadas e linhas SQL (INFLUXDATA, 2018b). Nota-se também que a organização do InfluxDB é mais adequada ao monitoramento feito no trabalho, por se basear no tempo como índice.

A linguagem de acesso e *query* do *InfluxDB* é, no entanto, similar a usada por bancos de dados do tipo SQL. A declaração abaixo, usada para selecionar dados em um banco InfluxDB demonstra esta similaridade, já que utiliza as mesmas *keywords* (*SELECT*, *FROM*, *WHERE*) e estrutura das declarações do tipo SQL.

SELECT <stuff> FROM <measurement_name> WHERE <some_conditions>

Em relação à inserção de dados no banco, o InfluxDB utiliza o chamado *protocolo de linha* (*line protocol*) para adicionar novos pontos, onde uma única linha representa um ponto de dados no banco, informando a medida, etiqueta, campo e

timestamp (estampa de tempo). Um exemplo de linha para inserção de dados no exemplo acima pode ser observado na linha abaixo.

```
monitor_potencia,carga_id=1,carga=lampadas#_Pot_ativa=350 1465839830100400200
```

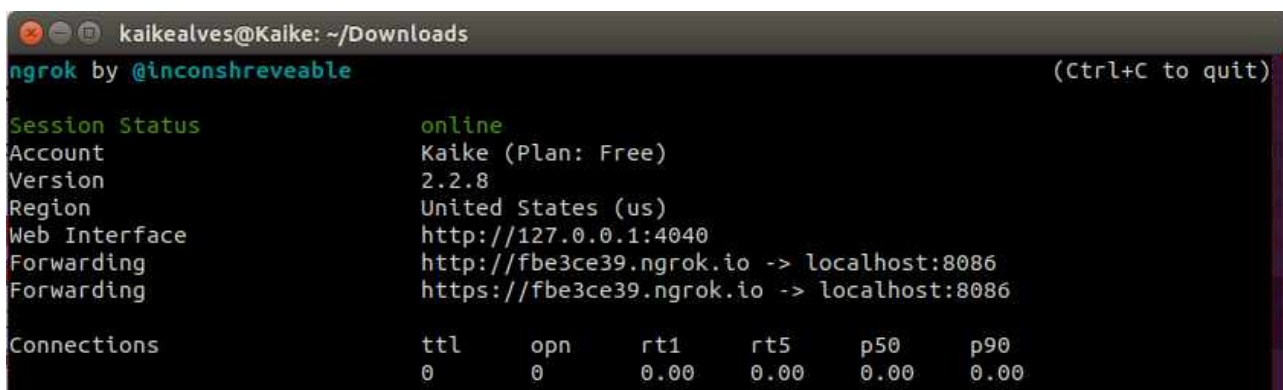
Neste exemplo, *monitor_potencia* corresponde ao nome do banco de dados, *carga_id=1* e *carga=lampadas* ao nome das etiquetas, *#_Pot_ativa=350* ao campo e *1465839830100400200* à *timestamp* (carimbo de tempo). Ressalta-se que, na ausência de um valor de *timestamp*, é utilizado o tempo de referência do banco de dados.

O OSS *InfluxDB* está disponível para os sistemas operacionais Linux e macOS, sendo o primeiro o utilizado neste trabalho, na distribuição *Ubuntu 18.04 LTS*. Optou-se pela versão 1.6.3 do *InfluxDB*, por ser a versão mais atual no momento da instalação (INFLUXDATA, 2018b).

Uma vez instalado e iniciado, o banco de dados está preparado para ser operado e receber novos dados através da *API HTTP*. Entretanto, como é de interesse do projeto conceder ao usuário o acesso ao banco de dados remotamente, e não somente através da rede local, é necessário expor o servidor a acessos externos. Para isso, utilizou-se a ferramenta *ngrok*, responsável por criar um “túnel” entre o servidor operando na rede local e a *internet*. A ferramenta é disponibilizada gratuitamente no *website ngrok.com*.

Na Figura 16 pode ser visualizada a execução da ferramenta, que fornece ao usuário informações como o estado atual do túnel, a *url* para acesso da porta e o número de conexões à porta desde o momento da ativação.

Figura 16 - Ferramenta *ngrok* operando um túnel para a rede local



```

kaikealves@Kaike: ~/Downloads
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             Kaike (Plan: Free)
Version             2.2.8
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://fbe3ce39.ngrok.io -> localhost:8086
Forwarding           https://fbe3ce39.ngrok.io -> localhost:8086

Connections
ttl    opn    rt1    rt5    p50    p90
0      0      0.00   0.00   0.00   0.00

```

Fonte: Autoria própria (2018)

2.2.6 Software

O *software* a ser carregado no microcontrolador ESP32 tem como funções a amostragem dos sinais tratados pela placa de aquisição e condicionamento de sinais, o cálculo das grandezas elétricas de interesse (potência ativa, reativa e aparente, fator de potência, valor eficaz da tensão e da corrente) e o envio dos valores calculados para o banco de dados. O *software*, que foi desenvolvido em linguagem C++ no ambiente de programação da IDE *Arduino*, será apresentado de forma geral a seguir. Ressalta-se que as bibliotecas utilizadas no trabalho são mantidas pela comunidade *Arduino* e possuem código aberto, podendo ser obtidas gratuitamente pela internet. O código completo pode ser encontrado no Apêndice B deste documento.

Ao ser executado, a primeira ação do *software* é a tentativa de estabelecimento de conexão do ESP32 com a rede *Wi-Fi* desejada. Para tal, utiliza-se a biblioteca *Wi-Fi*, disponibilizada gratuitamente pela *Espressif*. A função *Wi-Fi.begin*, que toma como argumentos o nome e a senha da rede, é responsável pela tentativa de estabelecimento da conexão, enquanto a função *WiFi.status* é utilizada para verificar se a conexão foi corretamente estabelecida. O *software* só segue para as próximas ações se houver sucesso na conexão.

Com a conexão *Wi-Fi* estabelecida, o *software* configura os parâmetros para a utilização das interrupções de tempo do microcontrolador e para o envio de dados para o banco de dados *InfluxDB*. Para a interação com o banco de dados o *software* utiliza funções da biblioteca *InfluxArduino*, desenvolvida e disponibilizada por Thomas Bruen. A função *Influx.configure*, que toma como argumentos o nome do banco de dados *InfluxDB*, o IP e a porta, é chamada então para a configuração das *strings* utilizadas para a alimentação do banco de dados.

A amostragem dos sinais, feita através da conversão dos sinais lidos pelo conversor AD, é regida por interrupções de tempo (*timer interrupts*), portanto é necessária a configuração do *timer* do microcontrolador, que é feita através de funções de bibliotecas nativas da plataforma *Arduino*. Os requisitos para a configuração das interrupções de tempo são a definição do *timer* a ser utilizado, o número chamado de *prescaler*, o tipo de contagem (crescente ou decrescente) e o valor de gatilho da interrupção (TECHTUTORIALSX, 2017)

A plataforma *ESP32* possui quatro *timers* de propósito geral embutidos, com contadores de 64 bits baseados em *prescalers* de 16 bits, onde a contagem pode ser configurada de maneira crescente ou decrescente (ESPRESSIF, 2018). O *timer* utilizado no trabalho foi o de número 1, já que o de número 0 é empregado por outras funções utilizadas no código. Optou-se, por conveniência, pela contagem crescente.

O número *prescaler*, responsável por dividir a frequência do sinal de base do microcontrolador, que no caso do *ESP32* corresponde a 80 MHz, altera a taxa de contagem do *timer*, possibilitando um maior controle sobre o valor de gatilho das interrupções. Como o *prescaler* do *ESP32* é representado em 16 bits, a taxa de contagem pode ser dividida por qualquer valor no intervalo entre 2 a 65536. Decidiu-se por um valor de *prescaler* de 80, de modo obter-se uma frequência de contagem de 1 MHz, ou seja, com tempo de incremento de um micro-segundo.

A função *timerBegin* é então chamada com os argumentos definidos, sendo o número do *timer* 0, o número do *prescaler* 80 e o modo de contagem como *true*, equivalente a contagem do tipo crescente. Em seguida, é chamada a função de *handle timerAttachInterrupt*. Com o *timer* configurado e iniciado, configura-se a interrupção de tempo desejada. A interrupção é responsável por alertar ao *software* caso a contagem feita pelo *timer* atinja um valor de gatilho, sendo importante destacar que a contagem do *timer*, assim como a verificação se o valor de gatilho foi atingido, acontece paralelamente à execução do código (EARL, 2014).

Para o ajuste do valor de gatilho, deve-se primeiro definir a taxa de amostragem desejada, já que o gatilho é obtido através da divisão do período de amostragem pelo período de contagem do *timer* (TECHTUTORIALSX, 2017). Como a taxa de amostragem definida para o trabalho foi de 7680 Hz, tem-se como período de amostragem ideal 130,21 μ s, conforme cálculo abaixo.

$$T_{ideal} = \frac{1}{f_{amostragem}} = \frac{1}{7680Hz} = 130,21\mu s \quad (17)$$

Onde T_{ideal} e f_{ideal} representam o período de amostragem ideal e a taxa de amostragem ideal, respectivamente.

Entretanto, tendo que a contagem do *timer* incrementa a cada 1 μ s, deve-se tomar um valor múltiplo do mesmo. Desta maneira, tomou-se o tempo de amostragem inteiro mais próximo de 130,21 μ s, ou seja, 130 μ s. Conforme mostrado

pelo cálculo abaixo (equação 18), verifica-se que este período de amostragem equivale a uma taxa de amostragem de 7692,3 Hz.

$$f_{real} = \frac{1}{T_{real}} = \frac{1}{130\mu s} = 7692,3Hz \quad (18)$$

Onde T_{real} e f_{real} equivalem ao período de amostragem e à taxa de amostragem utilizados no *software*. Desta forma, definiu-se o valor de gatilho como 130.

Após serem definidos os valores da interrupção de tempo, a função *timerAlarmWrite*, responsável por configurar a interrupção é chamada com os argumentos *timer*, que corresponde ao objeto criado pela função *timerBegin*, o valor de gatilho de 130, e a opção de reiniciar a contagem após atingir o valor de gatilho como *true*. Em seguida, é chamada a função *timerAlarmEnable*, responsável por ativar a contagem e os alertas.

Uma vez declaradas as configurações gerais, o *software* entra no laço de execução, onde são feitas as amostragens dos sinais, os cálculos das grandezas de interesse e o envio das mesmas para o banco de dados. No início do *loop* global, entra-se em um laço onde um vetor *buffer* é preenchido com amostras dos sinais de tensão e de corrente lidas a cada alerta do *timer* (130μs), até que seja atingido o número de amostras desejadas, que no caso do trabalho corresponde a 1024.

Como os *buffers* são preenchidos com os valores digitais lidos no conversor AD, que variam de 0 a 2^{12} (0 a 4096), a próxima etapa no código é a conversão destes para valores reais de tensão e corrente, em unidades de *volts* e *ampères*. A conversão é feita através da multiplicação dos valores por um ganho e da adição de um valor de *offset*, definidos experimentalmente e apresentados anteriormente. A equação de conversão utilizada, aplicada em todos os pontos da amostra, pode ser observada nas equações 19 e 20.

$$V_{real}[k] = G_v \times V_{digital}[k] + offset_v \quad (19)$$

$$I_{real}[k] = G_i \times I_{digital}[k] + offset_i \quad (20)$$

Onde $V_{digital}$ e $I_{digital}$ correspondem aos vetores que contêm as amostras em valores digitais, G_v e G_i representam os valores de ganho para tensão e corrente, $offset_v$ e $offset_i$ representam os valores de *offset* de tensão e corrente e V_{real} e I_{real}

são os vetores resultantes, que contêm as amostras em valores reais, em *volts* e *ampères*.

Uma vez convertidos os valores digitais para valores reais, a próxima etapa executada pelo *software* é o cálculo das grandezas elétricas de interesse. Para isto, são chamadas funções de cálculo, criadas pelo autor. O cálculo da tensão e correntes eficazes são feitos pela função *calcRMS*, que tomam como argumento o vetor com as amostras de tensão ou corrente e o número de amostras do vetor. Os valores eficazes são calculados conforme equação 21, que corresponde ao cálculo de valores eficazes com valores discretos.

$$T_{rms} = \sqrt{\frac{\sum_{k=0}^{N_s} T[k]^2}{N_s}} \quad (21)$$

Onde $T[k]$ são os pontos do vetor de amostra, N_s representa o número de pontos na amostra e T_{rms} é o valor eficaz resultante.

Em seguida são calculados os valores da potência ativa, reativa e aparente. A função *calcActPower*, responsável por calcular a potência ativa, recebe como argumentos os vetores com as amostras de tensão e corrente e o número de amostras dos vetores, e retorna um valor de potência conforme equação 22, que corresponde ao cálculo de potência ativa com valores discretos.

$$P = \frac{\sum_{k=0}^{N_s} I[k] \times V[k]}{N_s} \quad (22)$$

Onde $I[k]$ e $V[k]$ correspondem os pontos dos vetores de amostras de tensão e corrente reais, N_s representa o número de amostras no vetor e P à potência ativa resultante. Desconsideraram-se efeitos de *time-shift* (diferença temporal) entre as amostras, devido à alta taxa de amostragem utilizada.

A potência aparente, a potência reativa e o fator de potência são então calculados conforme equações 22, 23 e 24.

$$S = V_{rms} \times I_{rms} \quad (23)$$

$$Q = \sqrt{S^2 - P^2} \quad (24)$$

$$fp = \frac{P}{S} \quad (25)$$

Tendo as grandezas elétricas calculadas, a próxima etapa do *software* é o envio dos dados para o banco de dados *InfluxDB*. O envio é feito através da função *influx.write*, que recebe como argumentos o valor da medida e seu nome, a etiqueta e o campo. As medidas são enviadas ao banco de dados com a mesma etiqueta (*tag*), e com os campos respectivos à grandeza elétrica (*p*, *q*, *s*, *fp*, *v_rms*, *i_rms*). Tendo o dispositivo alimentado pelo USB, é possível monitorar o envio das informações pelo *serial monitor* da IDE do *Arduino*, onde é retornado um erro de valor -1 caso haja falha no envio.

Com o envio dos dados, o *software* é colocado em um laço vazio durante um período estipulado nas definições do mesmo. Quando o período é atingido, o *loop* global é reiniciado.

2.2.7 Grafana – Plataforma de visualização dos dados

Uma vez feitas as aquisições, os cálculos e o armazenamento dos dados colhidos pelo sistema, faz-se necessária a apresentação das informações ao usuário, de modo que este possa monitorar o sistema em tempo real, assim como analisar o histórico das medidas. Desta maneira, buscou-se uma plataforma que oferecesse um serviço de apresentação de gráficos com integração ao banco de dados, de modo a possibilitar a construção de *dashboards* (painéis de gráficos) com a exibição das informações armazenadas de maneira clara e organizada, com a opção de visualização de dados em tempo real ou em períodos passados.

Levando em consideração os requisitos e a utilização do banco de dados *InfluxDB* pelo trabalho, optou-se pela plataforma *Grafana*, uma ferramenta *open-source* que permite ações de *query*, visualização, alerta e compreensão de métricas armazenadas em bancos de dados. A plataforma pode ser instalada nos principais sistemas operacionais, mas oferece também um serviço de *cloud* (armazenamento na nuvem), com uma opção gratuita que inclui um serviço de até cinco *dashboards*. Com a utilização deste serviço, é possível acessar as informações do sistema a partir de qualquer rede a qualquer momento, sem a necessidade de instalação da plataforma em um servidor fixo.

Em relação à alimentação de dados à plataforma, a ferramenta permite a integração com mais de 30 bancos de dados diferentes, dentre eles o utilizado neste

trabalho, o *InfluxDB* (GRAFANA, 2018). O suporte ao *InfluxDB* pela plataforma inclui diversas funções, como editor de *query* com complementação das medidas, conforme utilizadas pelo banco de dados, manipulação de grupos de acordo com suas *timestamps*, sistema de alertas e criação de painéis (GRAFANA, 2018).

A plataforma foi configurada para ser alimentada pelos dados contidos no banco *InfluxDB* e uma *dashboard* foi criada para o monitoramento das medidas, contendo cinco (5) gráficos, sendo cada um para o monitoramento das seguintes grandezas:

- Potência ativa (W)
- Tensão eficaz (V)
- Corrente eficaz (A)
- Potência aparente, ativa e reativa (VA, W, Var)
- Fator de potência

A configuração de cada gráfico é feita através da seleção da *query* de acesso ao banco de dados, como exibidas na Figura 17.

Figura 17 - Configuração da alimentação de gráfico na plataforma *Grafana*

The image shows the Grafana query editor interface. At the top, the 'Data Source' is set to 'Power_Analysis'. Below this, the query editor is displayed with the following configuration:

FROM	default	p	WHERE	load	=	simulated	+
SELECT	field (value)	mean ()	+				
GROUP BY	time (\$_interval)	fill (null)	+				
FORMAT AS	Time series						
ALIAS BY	Naming pattern						

Fonte: Autoria própria (2018)

Lançando mão do sistema de alertas da plataforma, foi criado um alerta para o monitoramento da curva de potência ativa. Assim, quando o valor monitorado excede um limite estabelecido, uma mensagem é enviada ao usuário por meio de um canal de comunicação escolhido, que tem como opções o envio por *e-mail*, *Slack*, *Telegram*, entre outros. Para fim de testes, foi configurado um alerta para o monitoramento de potência ativa, onde se definiu um valor limite de 600W e uma tolerância de 1 minuto.

Para o recebimento dos alertas, foi escolhida o aplicativo de mensagens *Telegram*, por ser um aplicativo gratuito de mensagens instantâneas largamente utilizado. A configuração do alerta na plataforma *Grafana* pode ser visto nas Figuras 18 e 19.

Figura 18 - Configuração de alerta por meio do aplicativo *Telegram* na plataforma *Grafana*

Edit Notification Channel	
Name	Alerta_telegram
Type	Telegram
Send on all alerts	☐
Include image	☒
Send reminders	☐
Telegram API settings	
BOT API Token	728633819:AAHu2djv3rWekmq5rAjx4xucMW9Lge7BwjC
Chat ID	-308174347

Fonte: Autoria própria (2018)

Figura 19 - Configuração de alerta para medida monitorada pela plataforma *Grafana*

Alert Config						
Name	Potência ativa alerta					
Evaluate every	60s					
Conditions						
WHEN	avg ()	OF	query (A, 30s, now)	IS ABOVE	600	
+						

Fonte: Autoria própria (2018)

2.3 RESULTADOS E DISCUSSÕES

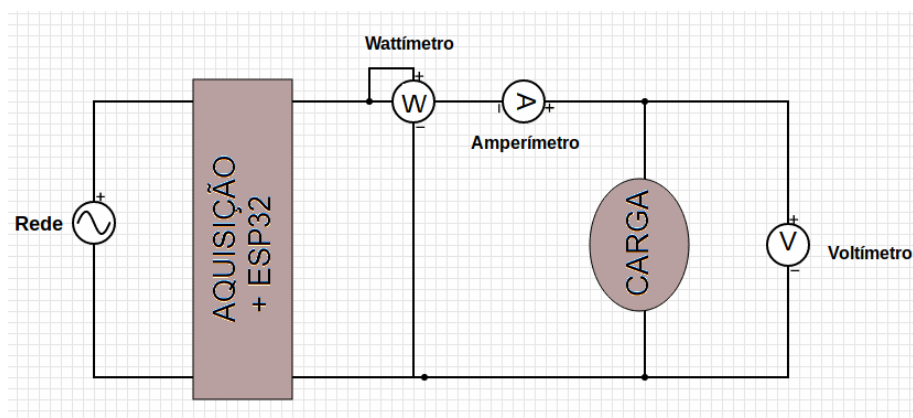
Com todas as etapas e ferramentas desenvolvidas e configuradas, o sistema de medição de consumo foi avaliado em diferentes cenários, onde as características elétricas de algumas cargas foram examinadas. Foram escolhidas cargas com características diferentes, a fim de se observar o comportamento do medidor em situações diversas. Para cada carga investigada, foram calculadas pelo sistema valores das grandezas elétricas desejadas, que foram então enviadas e armazenadas no banco de dados e visualizadas na ferramenta de apresentação de dados *Grafana*. Foram adquiridos também valores resultantes de aferições de outros

equipamentos de medição, através do uso de um voltímetro digital, um amperímetro digital e um *wattímetro*, conectados na entrada das cargas, a fim de analisar a precisão das medidas feitas pelo sistema. Já os valores de potência aparente, potência reativa e fator de potência foram calculados com base nos valores de corrente eficaz, tensão eficaz e potência ativa aferidos pelos equipamentos de medição.

Assim, examinaram-se as medidas obtidas pelo sistema para as seguintes cargas: um resistor variável de 100Ω ; um resistor variável de 100Ω em paralelo com um capacitor de $22\mu\text{F}$; uma fonte de energia conectada a um *laptop*; e um motor operando a vazio.

Pode-se visualizar na Figura 20 o esquemático padrão utilizado para as aferições.

Figura 20 - Esquemático da configuração utilizada para análise do sistema



Fonte: Autoria própria (2018)

A *dashboard* com todas as medidas resultantes do monitoramento das cargas podem ser analisada no Apêndice C deste trabalho.

2.3.1 Carga resistiva

A primeira carga investigada foi o resistor variável de 100Ω , por ser uma carga com característica resistiva, alimentada na rede em 220V. O resistor foi ajustado para resistências intermediárias enquanto os medidores e os gráficos na plataforma foram analisados.

No Quadro 5 são apresentados os resultados para a carga resistiva.

Quadro 5 - Resultados para carga resistiva

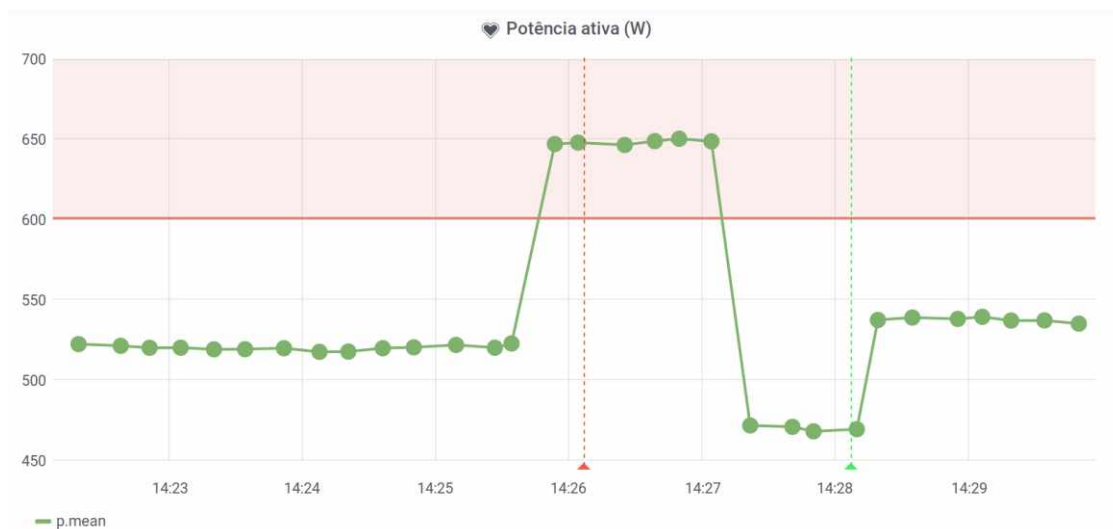
	Monitor (Grafana)	Equipamentos de aferição
Tensão Eficaz	215,417 V	213,4 V
Corrente Eficaz	2,145 A	2,137 A
Potência ativa	461,888 W	455 W
Potência aparente	462,069 VA	456,0 VA
Potência reativa	12,932 VAR	30,7 VAR
Fator de potência	0,9996	0,998

Fonte: Autoria própria (2018)

Nota-se pelos resultados que os valores de corrente eficaz, tensão eficaz e de potência ativa obtidos pelo sistema foram muito próximos aos lidos pelos equipamentos de aferição, o que mostra uma boa precisão nas medidas. Percebeu-se também uma medida de fator de potência com valor aproximadamente unitário, coerente com a realidade, já que a carga analisada tem características quase puramente resistivas.

Em contrapartida, verificou-se que o valor de potência reativa calculada pelo sistema apresentou um erro considerável em relação ao calculado com base nos valores aferidos, o que pode ser causado pelas pequenas discrepâncias nas leituras de tensão e potência ativa.

Depois de colhidos estes resultados, a carga resistiva foi monitorada por alguns minutos com o objetivo de se adquirir as curvas das grandezas elétricas em relação ao tempo. O resistor variável teve seus valores alterados durante o monitoramento a fim de se analisar a resposta do sistema. Na Figura 21 pode ser visualizado o gráfico da curva de potência ativa gerado durante alguns minutos, onde podem ser observadas mudanças na potência injetada de acordo com a mudança da resistência.

Figura 21 - Curva de potência ativa da carga resistiva gerada pelo *Grafana*

Fonte: Autoria própria (2018)

Constatou-se que durante um período de tempo a potência ativa superou o limite estabelecido de 600W, gerando um alerta pela plataforma após confirmar a conservação do nível de potência pelo período de tolerância de 1 (um) minuto. A mensagem foi recebida pelo aplicativo *Telegram*. Após ter seu valor reduzido abaixo do limite durante outro período de tolerância, uma segunda mensagem foi enviada pela plataforma comunicando o fim do alerta. A mensagem recebida pode ser visualizada na figura, onde a ativação do alerta e a saída do mesmo são sinalizadas pelas tags *[Alerting]* e *[OK]*. As mensagens recebidas pelo aplicativo podem ser visualizadas na Figura 22.

Figura 22 - Alertas gerados pelo *Grafana* e recebidos pelo *Telegram*

Fonte: Autoria própria (2018)

2.3.2 Carga resistiva em paralelo com carga capacitiva

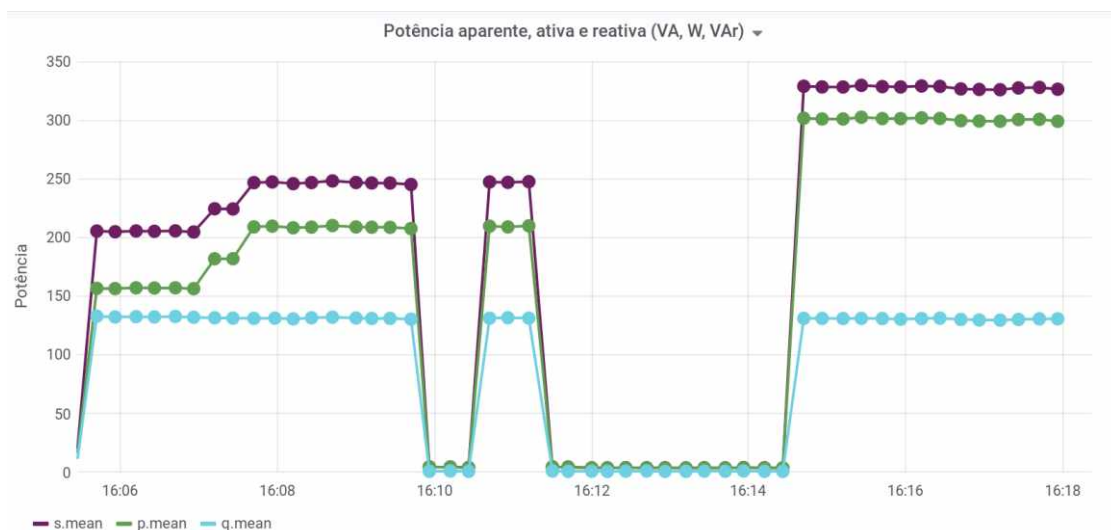
O resistor variável de 100Ω foi então ligado em paralelo a um capacitor de $22\mu\text{F}$, alimentada na rede em 127V, com a finalidade de estudar-se o comportamento do sistema quando medindo uma carga capacitiva. Os resultados podem ser analisados no Quadro 7.

Quadro 6 - Resultados para carga capacitiva

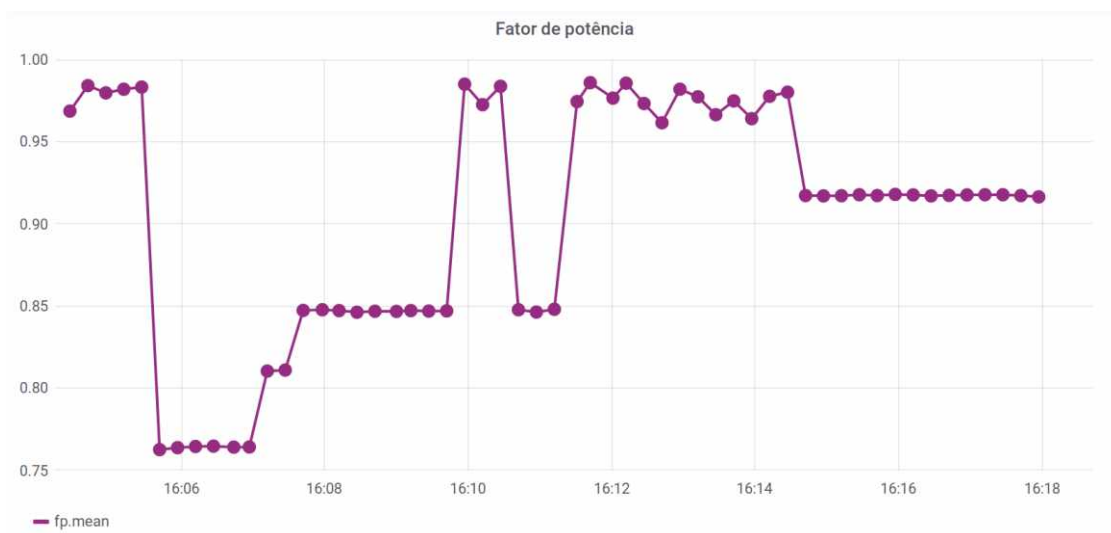
	Monitor (Grafana)	Equipamentos de aferição
Tensão Eficaz	126,000 V	123.6 V
Corrente Eficaz	1,958 A	1,952 A
Potência ativa	208.242 W	205 W
Potência aparente	245.822 VA	241,3 VA
Potência reativa	130.630 VAr	127,2 VAr
Fator de potência	0,847	0,850

Fonte: Autoria própria (2018)

Verifica-se nos resultados uma boa precisão do sistema neste cenário. Na Figura 23 podem ser analisados os valores de potências ativa, reativa e aparente calculados pelo sistema, enquanto na Figura 24 pode ser visualizado o fator de potência correspondente. Nota-se em alguns períodos uma mudança na potência ativa, causada pela alteração do valor do resistor, enquanto a potência reativa mantém o mesmo valor. Este comportamento é condizente com a realidade, já que as cargas estão em paralelo e o capacitor tem valor fixo.

Figura 23 - Curvas de potência da carga capacitiva gerada pelo *Grafana*

Fonte: Autoria própria (2018)

Figura 24 – Curva do fator de potência da carga capacitiva gerada pelo *Grafana*

Fonte: Autoria própria (2018)

2.3.3 Motor a vazio

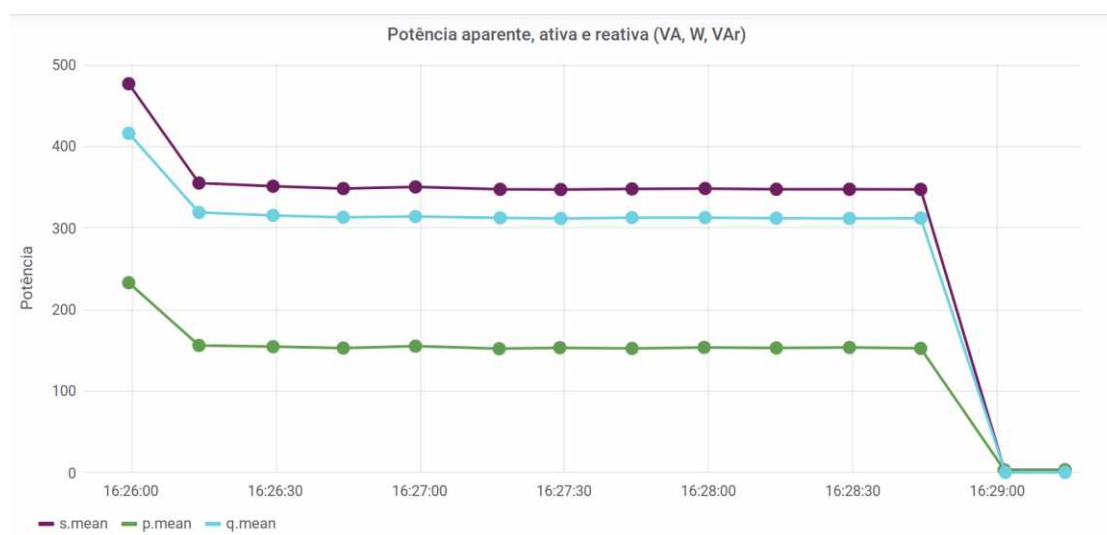
Em seguida, estudou-se o comportamento do sistema quando feita a alimentação de um motor elétrico operando a vazio em 127V, com a finalidade de se verificar o cenário onde a carga a ser medida é do tipo indutiva. Os resultados obtidos são exibidos no Quadro 7.

Quadro 7 - Resultados para motor a vazio

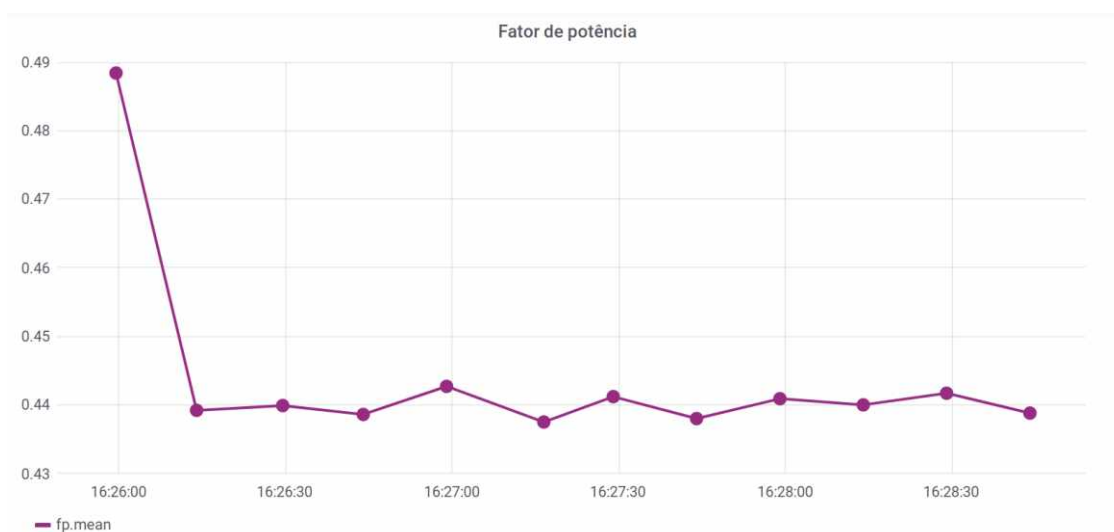
	Monitor (<i>Grafana</i>)	Equipamentos de aferição
Tensão Eficaz	125.702 V	123,7 V
Corrente Eficaz	2.824 A	2,883 A
Potência ativa	155.905 W	150 W
Potência aparente	354.976 VA	356,6 VA
Potência reativa	318.907 VAR	323,5 VAR
Fator de potência	0.439	0,421

Fonte: Autoria própria (2018)

Os resultados apresentaram uma boa precisão para este cenário para todas as grandezas elétricas observadas. Na Figura 25 pode ser analisado o gráfico das curvas das potências ativa, reativa e aparente, onde pode ser verificado a característica indutiva do motor a vazio, sendo a potência reativa, inclusive, superior à ativa. Pode ser visto na Figura 26 o fator de potência da carga.

Figura 25 - Curvas de potência do motor a vazio gerada pelo Grafana

Fonte: Autoria própria (2018)

Figura 26 - Curva do fator de potência do motor a vazio gerada pelo Grafana

Fonte: Autoria própria (2018)

2.3.4 Fonte alimentando *laptop*

A última análise foi feita com fonte de alimentação conectada a um *laptop*. Com o objetivo de investigar o desempenho do sistema em um cenário onde há baixo consumo, variação de corrente e não-linearidade. Os resultados podem ser verificados no Quadro 8.

Quadro 8 - Resultados para fonte de alimentação de *laptop*

	Monitor (<i>Grafana</i>)	Equipamentos de aferição
Tensão Eficaz	218,988 V	216,544 V
Corrente Eficaz	0,298 A	0,310 A
Potência ativa	42,738 W	40 W
Potência aparente	65,23 VA	67,129 VA
Potência reativa	-	-
Fator de potência	0,655	0,560

Fonte: Autoria própria (2018)

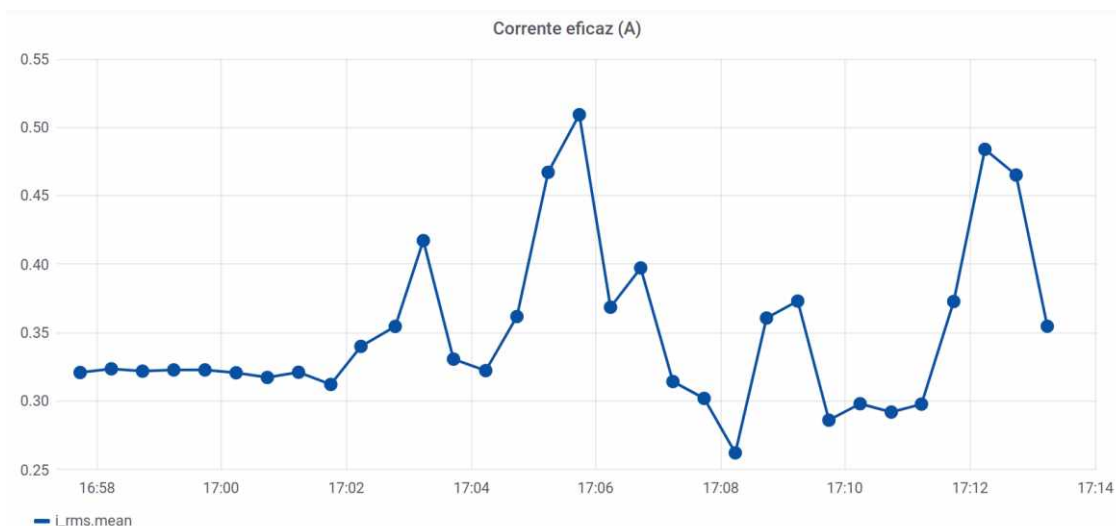
Os resultados mostraram boa precisão para as medidas de tensão, corrente e potência ativa. Entretanto, os erros encontrados são maiores do que os obtidos para outras cargas, o que pode ser explicado pelo baixo consumo da fonte de

alimentação e pela rápida variação da corrente, conforme exigida pelo processamento do *laptop*. Dependendo do momento da leitura, estas características podem levar a obtenção de dados diferentes entre o sistema e os equipamentos de aferição.

Ressalta-se que para a fonte de alimentação não foi considerado o cálculo de potência aparente, uma vez que a entrada da fonte, composta por um retificador, permite apenas o fluxo unidirecional de potência. Desta maneira, não há sentido em falar-se em potência reativa, já que não há troca de energia entre a rede e a carga. Observa-se, entretanto, um baixo fator de potência, causado pela distorção harmônica da corrente.

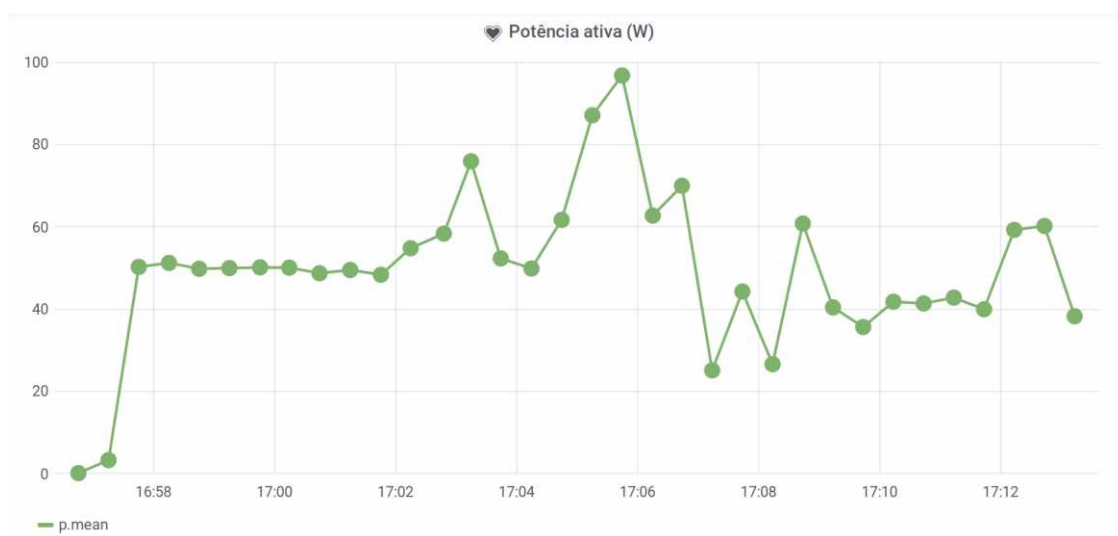
Na Figura 27 pode ser visualizada a curva de corrente adquirida durante a alimentação da fonte. Nota-se uma alta variação, condizente com atividades realizadas no *laptop*, como a mudança de brilho da tela ou a execução de *softwares* que exigem menor ou maior processamento. Na Figura 28 pode ser examinada a potência ativa requerida pela fonte, que mostra uma grande variação e um baixo consumo, não ultrapassando 100W.

Figura 27 - Curva de corrente eficaz da fonte de alimentação de laptop gerada pelo *Grafana*



Fonte: Autoria própria (2018)

Figura 28 - Curva de potência ativa da fonte de alimentação de laptop gerada pelo Grafana



Fonte: Autoria própria (2018)

3 CONCLUSÃO

O sistema de monitoramento de grandezas elétricas desenvolvido foi capaz de atender aos requisitos do trabalho, apresentando considerável precisão quando comparado a medidores comerciais, mesmo para cargas com características diversas. Desta maneira, é possível considerar que há potencial nesta ideia de sistema para ambientes residenciais.

Do ponto de vista da *Internet das Coisas*, o sistema se mostrou satisfatório, com suas ferramentas apresentando boa adequação à proposta, sendo acessíveis de diferentes dispositivos com conexão à internet, como computadores e *smartphones*. O banco de dados utilizado, *InfluxDB*, exibiu um bom desempenho, recebendo dados com rapidez, armazenando-os de maneira adequada e oferecendo suporte a diferentes plataformas. A plataforma de análise *Grafana* demonstrou ser apropriada para o monitoramento das grandezas, possibilitando a construção de *dashboards* graficamente agradáveis para o acompanhamento em tempo real do consumo de energia elétrica.

O microcontrolador utilizado no trabalho, da série ESP32, apresentou bom desempenho, mesmo com o emprego apenas de seus periféricos embutidos. Entretanto, considera-se que resultados ainda mais precisos podem ser obtidos com conversores AD externos, que possuam maior quantidade de *bits* e suporte a taxas de amostragens superiores à utilizada.

Para trabalhos futuros, sugere-se a construção de um protótipo finalizado com todos os componentes (fonte, placa e microcontrolador) embutidos em um mesmo dispositivo, o que facilitaria testes em períodos mais longos. No que tange ao estudo das características elétricas das cargas, recomenda-se a inclusão de ferramentas de análise de qualidade de energia elétrica à montagem, através do uso de *FFTs*, uma vez que a amostragem utilizada no *software* suporta esta análise, o que possibilitaria investigações no domínio da frequência.

4 REFERÊNCIAS

ANEEL. **Cadernos Temáticos ANEEL: Micro e Minigeração Distribuída – Sistemas de Compensação de Energia Elétrica**. 2. ed. Brasília: [s.n.], 2016. 34 p.

BERMAN, D. **InfluxDB vs. Elasticsearch for Time Series Analysis**. 2017. Disponível em: <<https://logz.io/blog/influxdb-vs-elasticsearch/>>. Acesso em: 25 nov. 2018.

CEMIG. **FAQ Tarifas**. 2018. Disponível em: <https://www.cemig.com.br/pt-br/atendimento/Paginas/FAQ_Tarifa.aspx>. Acesso em 25 nov. 2018. Chalmers University Of Technology, Sweden, 1999. Disponível em: <<https://www.sp.se/en/index/services/DSWM/Documents/SvenssonStefanPhD.pdf>>. Acesso em: 25 nov. 2018.

EARL, B. **Timer Interrupts**. 2014. Disponível em: <<https://learn.adafruit.com/multi-tasking-the-arduino-part-2/timers>>. Acesso em: 25 nov. 2018.

ESPRESSIF. **ESP32 Series Datasheet**. 2018. Disponível em <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 25 nov. 2018.

ESPRESSIF. **Get started: ESP32**. c2016. Disponível em: <<https://docs.espressif.com/projects/esp-idf/en/latest/get-started/index.html>>. Acesso em: 25 nov. 2018.

FUCHS, P. **Digital Power and Energy Meterwith DSP TMS320C6711**. 2006. Disponível em: <<http://www.ti.com/lit/ml/sprp504/sprp504.pdf>>. Acesso em: 25 nov. 2018.

IEEE, **Towards a definition of the Internet of Things (IoT)**, 2015. Disponível em <https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revison1_27MAY15.pdf>. Acesso em 25 nov. 2018.

INFLUXDATA.**InfluxDB**. 2018. Disponível em: <<https://www.influxdata.com/time-series-platform/influxdb/>>. Acesso em: 25 nov. 2018.

INFLUXDATA. **InfluxDB 1.6 documentation**. 2018. Disponível em: <<https://docs.influxdata.com/influxdb/v1.6/>>. Acesso em: 25 nov. 2018.

GRAFANA. **Grafana**: The leading open source software for time series analytics2018. Disponível em: <<https://grafana.com/grafana>>. Acesso em: 25 nov. 2018.

LEE, I; LEE, K. **The Internet of Things (IoT)**: Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431–440. doi:10.1016/j.bushor.2015.03.008

LUETH, K. **State of the IoT 2018**: Number of IoT devices now at 7B.2018. Disponível em: <<https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>>. Acesso em 25 nov. 2018.

MATTERN, F; FLOERKEMEIER, C. **From the internet of computers to the internet of things, From active data management to event-based systems and more**: papers in honor of Alejandro Buchmann on the occasion of his 60th birthday. 2010. Disponível em: <<http://www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf>>. Acesso em: 25 nov. 2018.

MAXIM INTEGRATED. **Coherent Sampling vs. Window Sampling**. 2002. Disponível em: <<https://www.maximintegrated.com/en/app-notes/index.mvp/id/1040>>. Acesso em: 25 nov. 2018.

MESSLER, D. **HP Security and the Internet of Things**. 2014. Disponível em:<http://h30499.www3.hp.com/t5/Fortify-Application-Security/HP-Security-and-The-Internet-of-Things/ba-p/6450208.U9_M6dQsL2s>. Acesso em: 25 nov. 2018.

NATIONAL INSTRUMENTS. **Anti-Aliasing Filters and Their Usage Explained**. 2018. Disponível em: <<http://www.ni.com/white-paper/54448/en/>>. Acesso em: 25 nov. 2018.

NILSSON, J W.; RIEDEL, S. **Electric Circuits**. 10. ed. USA: Pearson, 2014. 816 p.

OCTAVE. **Polynomial Interpolation**. c1996. Disponível em: <<https://octave.org/doc/v4.4.1/Polynomial-Interpolation.html>>. Acesso em: 25 nov. 2018.

RNP. **Noções básicas sobre áudio**. 2002. Disponível em <<http://penta3.ufrgs.br/RNP/cap3/3.2%20Audio/>>. Acesso em: 25 nov. 2018.

STEPHENS, R. **The State of the Time Series Database Market**. 2018. Disponível em: <<https://redmonk.com/rstephens/2018/04/03/the-state-of-the-time-series-database-market/>>. Acesso em: 25 nov. 2018.

SVENSSON, S. **Power measurement techniques for non-sinusoidal conditions: The significance of harmonics for the measurement of power and other AC quantities**. 159 p. Tese (Doutorado) – Dep. ent of Electric Power Engineering,

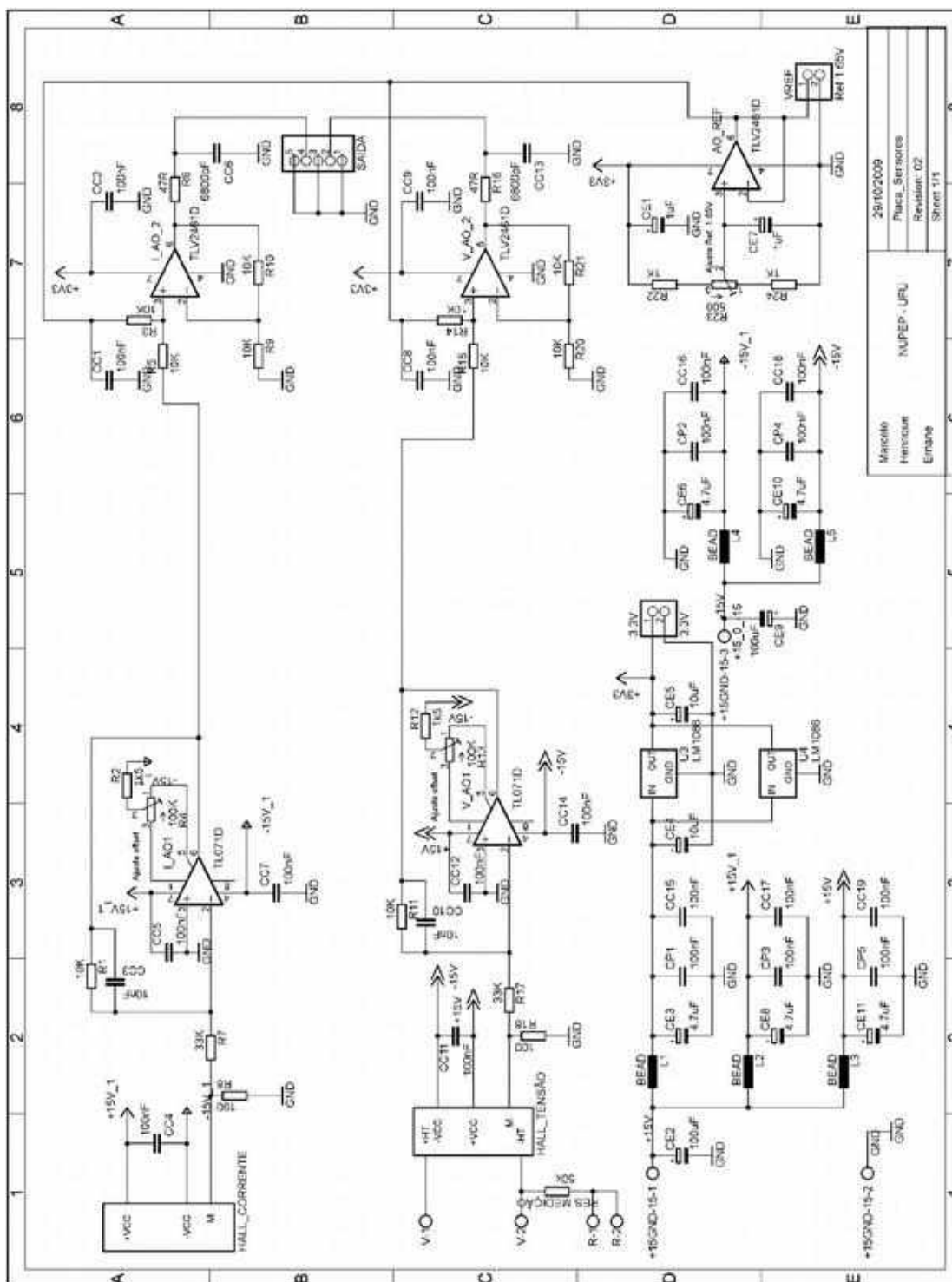
TECHTUTORIALSX. **ESP32 Arduino: Timer interrupts**. 2017. Disponível em: <<https://techtutorialsx.com/2017/10/07/esp32-arduino-timer-interrupts/>>. Acesso em: 25 nov. 2018.

TRENTCLASSASSIGNMENT. **Quantization**. 2017. Disponível em: <<https://trentsclassassignment.weebly.com/>>. Acesso em: 25 nov. 2018.

ANEXO A

Na Figura 29 é exibido o esquemático da placa de aquisição e condicionamento de sinais, fornecido pelo NUPEP.

Figura 29 - Esquemático completo da placa de aquisição e condicionamento



Fonte: Disponibilizado pelo NUPEP

APÊNDICE A

É apresentado neste apêndice o código carregado no microcontrolador ESP32 para a aquisição das palavras digitais equivalentes aos sinais de tensão e corrente colhidos pela placa de aquisição e condicionamento de sinais.

```
#define PRESCALER 80    // o clock de 80MHz sera dividido pelo valor PRESCALER
#define TIMERCOUNT 130 // 130us -> 7680 Hz
#define HWTIMER 1      // numero do clock de 0 a 4
#define COUNT_UP true  // true -> contagem crescente
#define COUNT_DOWN false // false -> contagem decrescente
#define EDGE true      // necessario para conf. do timer
#define LEVEL false    // necessario para conf. do timer
#define RELOAD_ON true  // contagem reinicia quando atinge limite
#define RELOAD_OFF false // contagem nao reinicia quando atinge limite
#define I_PIN 35        // pino AD conectado á saida de corrente da placa de cond.
#define V_PIN 34        // pino AD conectado á saida de tensao da placa de cond.
#define N_SAMPLES 1000 // numero de amostras de palavras digitais

int V_dig[N_samples], I_dig[N_samples];
double V_dig_avg = 0;
double I_dig_avg = 0;
volatile int interruptCounter; // contagem usada pelo interrupt
int k = 0;

hw_timer_t * timer = NULL; // ponteiro para variavel hw_timer_t
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED; // usado na sincronizacao do loop
com o interrupt

// funcao de interrupcao chamada quando timer atinge o valor TIMERCOUNT
void IRAM_ATTR onTimer() {
    portENTER_CRITICAL_ISR(&timerMux);
    interruptCounter++; // incrementa quando ocorre interrupcao
    portEXIT_CRITICAL_ISR(&timerMux);
}

void setup(){

    Serial.begin(115200); // inicia a comunicacao serial

    timer = timerBegin(HWTIMER, PRESCALER, COUNT_UP); // inicializacao do timer
    timerAttachInterrupt(timer, &onTimer, EDGE); // funcao de manipulacao do timer
    timerAlarmWrite(timer, TIMERCOUNT, RELOAD_ON); // especifica a contagem que gera a inter.
    timerAlarmEnable(timer); // ativa o timer
}

void loop()
{

    unsigned long startTime = micros(); //used for timing when to send data next.
    interruptCounter = 0;
```

```

/***** AQUISICAO DAS AMOSTRAS *****/
while(k < N_samples){ // loop para aquisicao das amostras das palavras digitais
  if (interruptCounter > 0 && k < N_samples) {

    portENTER_CRITICAL(&timerMux);
    V_dig[k] = analogRead( V_PIN );
    I_dig[k] = analogRead( I_PIN );
    k++;
    interruptCounter--;
    portEXIT_CRITICAL(&timerMux);

    totalInterruptCounter++;
  }
}
/*****/
k = 0;

// calcula a media de 1000 amostras de palavras digitais e imprime na saida serial do ESP32
for( int j = 0; j < N_samples; j++){
  V_dig_avg = V_dig_avg + V_dig[j];
  I_dig_avg = I_dig_avg + I_dig[j];
}
V_dig_avg = V_dig_avg / N_samples;
I_dig_avg = I_dig_avg / N_samples;
Serial.print("Media V: ");
Serial.println(V_dig_avg);
Serial.print("Media I: ");
Serial.println(I_dig_avg);
/*****/
delay(5000);
V_dig_avg = 0;
I_dig_avg = 0;

}

```

APÊNDICE B

É apresentado neste apêndice o código carregado no microcontrolador ESP32 para a aquisição dos sinais, cálculo das grandezas elétricas, conexão à rede WiFi e comunicação com o banco de dados.

```
#include <WiFi.h>
#include "InfluxArduino.hpp"

/***** TIMER AND SAMPLING *****/
#define PRESCALER 80 // o clock de 80MHz sera dividido pelo valor PRESCALER
#define TIMERCOUNT 130 // 130us -> 7680 Hz
#define HWTIMER 1 // numero do clock de 0 a 4
#define COUNT_UP true // true -> contagem crescente
#define COUNT_DOWN false // false -> contagem decrescente
#define EDGE true
#define LEVEL false
#define RELOAD_ON true // contagem reinicia quando atinge limite
#define RELOAD_OFF false // contagem nao reinicia quando atinge limite
#define I_PIN 35 // pino AD conectado á saida de corrente da placa de cond.
#define V_PIN 34 // pino AD conectado á saida de tensao da placa de cond.
#define N_SAMPLES 1024 // numero de amostras por janela

volatile int interruptCounter; // contagem usada pelo interrupt
int V_dig[1024], I_dig[1024];
double V_real[1024], I_real[1024];
int k = 0;
double V_gain = 0.5771;
double V_offset = 1796.1168;
double I_gain = 0.005611;
double I_offset = 1801.2462;

hw_timer_t * timer = NULL; // ponteiro para variavel hw_timer_t
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED; // usado na sincronizacao
// funcao de interrupcao chamada quando timer atinge o valor TIMERCOUNT
void IRAM_ATTR onTimer() {
    portENTER_CRITICAL_ISR(&timerMux);
    interruptCounter++; // incrementa quando ocorre interrupcao
    portEXIT_CRITICAL_ISR(&timerMux);
}

// dados para conexao a rede WiFi
const char WIFI_NAME[] = "NUPEP"; // nome da rede wifi
const char WIFI_PASS[] = " "; // senha da rede wifi
const char INFLUX_DATABASE[] = "mydb"; // nome da base de dados
const char INFLUX_IP[] = "10.1.2.168"; // ip do servidor

// dados para o envio de dados para o BD influxDB
InfluxArduino influx;
const char INFLUX_MEASUREMENT_P[] = "p"; // medida p a ser criada no BD
const char INFLUX_MEASUREMENT_V_RMS[] = "v_rms"; // medida v_rms a ser criada no BD
const char INFLUX_MEASUREMENT_I_RMS[] = "i_rms"; // medida i_rms a ser criada no BD
const char INFLUX_MEASUREMENT_S[] = "s"; // medida s a ser criada no BD
const char INFLUX_MEASUREMENT_Q[] = "q"; // medida q a ser criada no BD
const char INFLUX_MEASUREMENT_FP[] = "fp"; // medida fp a ser criada no BD

unsigned long DELAY_TIME = 15 * 1000 * 1000; // periodo de espera em microsegundos
```

```

void setup()
{
    Serial.begin(115200);

    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) { // aguarda conexao a rede WiFi
        delay(500);
    }
    Serial.println("WiFi connected!");
    influx.configure(INFLUX_DATABASE, INFLUX_IP); // configuracao do influxDB (porta 8086)
    Serial.print("Using HTTPS: ");

    timer = timerBegin(HWTIMER, PRESCALER, COUNT_UP); // inicializacao do timer
    timerAttachInterrupt(timer, &onTimer, EDGE); // funcao de manipulacao do timer
    timerAlarmWrite(timer, TIMERCOUNT, RELOAD_ON); // configura o valor que gera a interrupcao
    timerAlarmEnable(timer); // ativa o timer
}

void loop()
{
    unsigned long startTime = micros(); // usado para definir o periodo de espera ao fim do loop
    bool writeSuccessful = false; // flag usada para verificacao de envio de dados para BD
    interruptCounter = 0;

    /***** AQUISICAO DE DADOS *****/
    while(k < N_SAMPLES){ // loop para aquisicao das amostras de tensao e corrente
        if (interruptCounter > 0 && k < N_SAMPLES) {

            portENTER_CRITICAL(&timerMux);
            V_dig[k] = analogRead( V_PIN );
            I_dig[k] = analogRead( I_PIN );

            k++;
            interruptCounter--;
            portEXIT_CRITICAL(&timerMux);

        }
    }
    k = 0;

    for( int j = 0; j < N_SAMPLES; j++){
        V_real[j] = ( (double)V_dig[j] - V_offset ) * V_gain;
        I_real[j] = ( (double)I_dig[j] - I_offset ) * I_gain;
    }
    /***** CALCULOS *****/
    float Vrms, Irms, Pact, S, Q, fp, Vp, Ip;
    Vrms = calcRMS(V_real, N_SAMPLES); // calculo da tensao eficaz
    Irms = calcRMS(I_real, N_SAMPLES); // calculo da corrente eficaz
    Pact = calcActPower(V_real, I_real, N_SAMPLES); // calculo da potencia ativa
    S = Vrms * Irms; // calculo da potencia aparente
    Q = sqrt(S*S - Pact*Pact); // calculo da potencia reativa
    fp = Pact/S; // calculo do fator de potencia
    /***** INFLUXDB *****/
    char tags[32];
    char fields[32];

```

```

sprintf(tags,"load=simulated"); // seleciona a tag chamada carga

sprintf(fields,"value=%0.4f", Vrms);
writeSuccessful = influx.write(INFLUX_MEASUREMENT_V_RMS,tags,fields); // envia v_rms
if(!writeSuccessful) Serial.println(influx.getResponse());

sprintf(fields,"value=%0.4f", Irms); //write one fieldd: Irms
writeSuccessful = influx.write(INFLUX_MEASUREMENT_I_RMS,tags,fields); // envia i_rms
if(!writeSuccessful) Serial.println(influx.getResponse());

sprintf(fields,"value=%0.4f", Pact);
writeSuccessful = influx.write(INFLUX_MEASUREMENT_P,tags,fields); // envia p
if(!writeSuccessful) Serial.println(influx.getResponse());

sprintf(fields,"value=%0.4f", S);
writeSuccessful = influx.write(INFLUX_MEASUREMENT_S,tags,fields); // envia s
if(!writeSuccessful) Serial.println(influx.getResponse());

sprintf(fields,"value=%0.4f", Q);
writeSuccessful = influx.write(INFLUX_MEASUREMENT_Q,tags,fields); // envia q
if(!writeSuccessful) Serial.println(influx.getResponse());

sprintf(fields,"value=%0.4f", fp);
writeSuccessful = influx.write(INFLUX_MEASUREMENT_FP,tags,fields); // envia fp
if(!writeSuccessful) Serial.println(influx.getResponse());

while ((micros() - startTime) < DELAY_TIME) {} // espera ate que o tempo DELAY_TIME

}
/* funcao calcRMs
objetivo: calculo de valor eficaz de janela de amostragem
parametros: vec[] - amostras
            Nsamples - numero de amostras consideradas
retorno: RMSval - valor eficaz das amostras
*/
float calcRMS(double vec[], int Nsamples){
float RMSval = 0;
for(int j = 0; j < Nsamples; j++){
RMSval = RMSval + vec[j] * vec[j];
}
RMSval = sqrt(RMSval / Nsamples);
return RMSval;
}

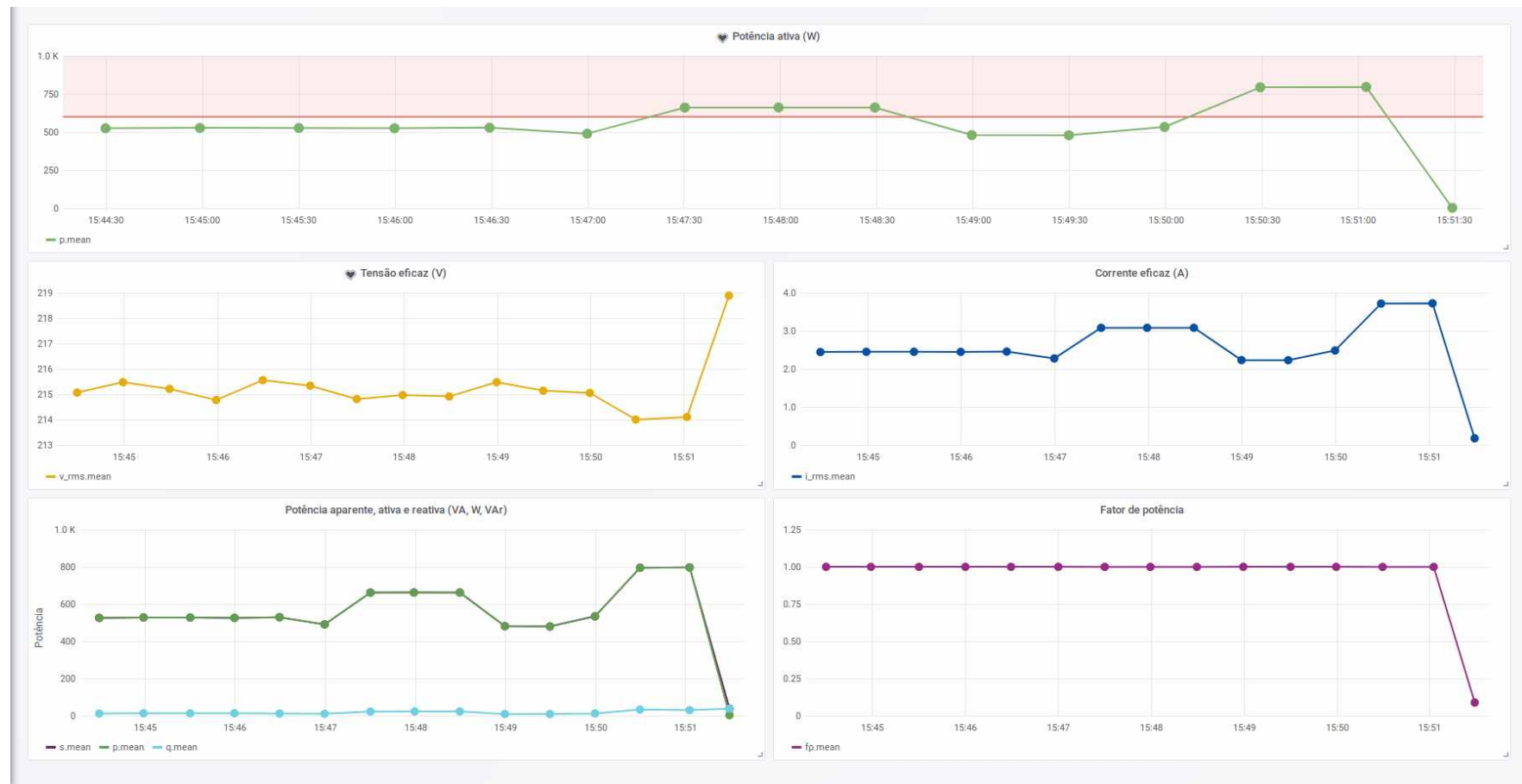
/* funcao calcRMs
objetivo: calculo de potencia ativa
parametros: V[ ] - amostras de tensao
            I[ ] - amostras de corrente
            Nsamples - numero de amostras consideradas
retorno: RMSval - valor eficaz das amostras
*/
float calcActPower(double V[], double I[], int Nsamples){
float actPower = 0;
for(int j = 0; j < Nsamples; j++){
actPower = actPower + V[j] * I[j];
}
actPower = actPower / Nsamples;
return actPower;
}

```

APÊNDICE C

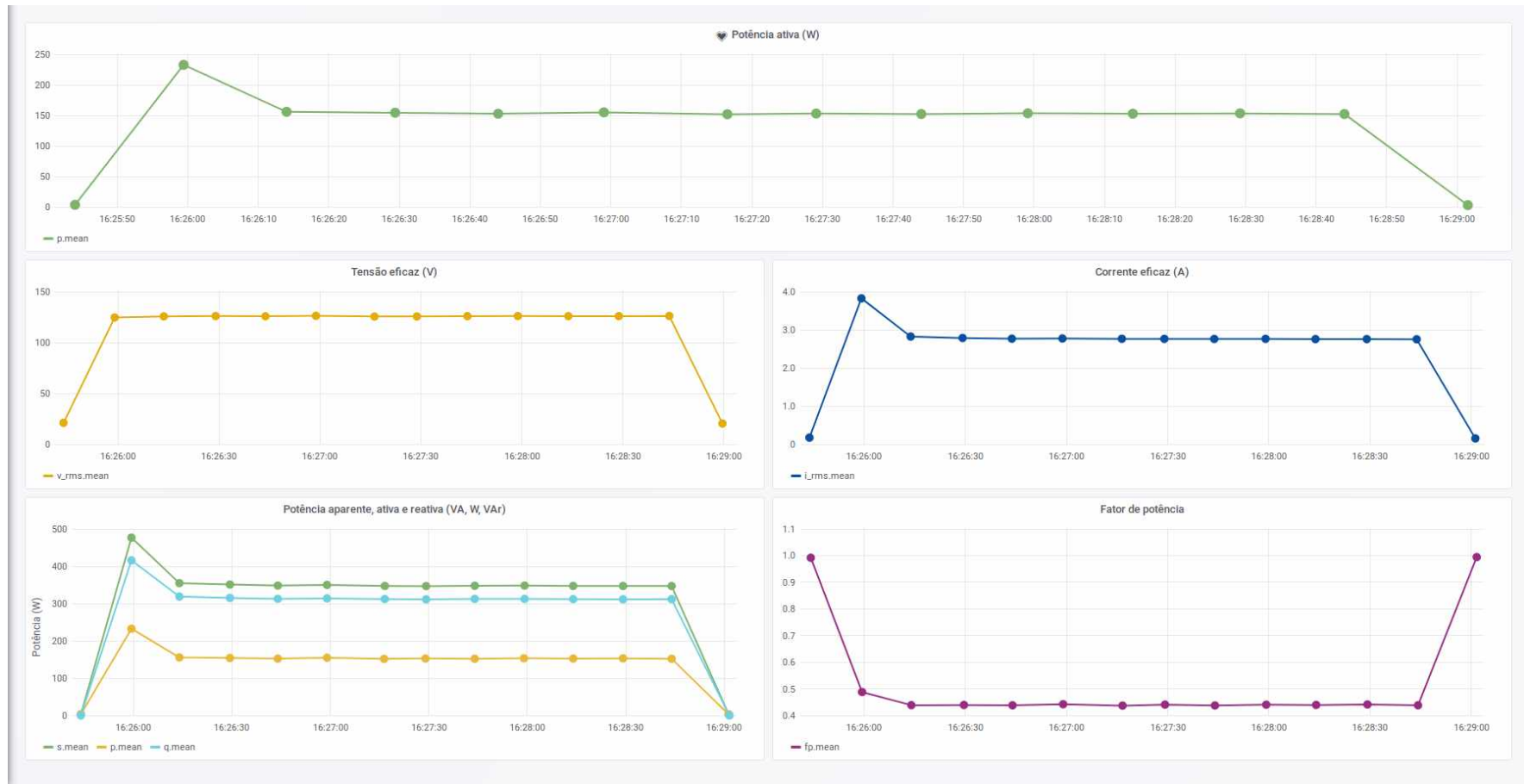
Neste apêndice são apresentadas capturas de tela das *Dashboards* na plataforma *Grafana* durante o monitoramento das cargas.

Figura 30 - *Dashboard* durante monitoramento de carga resistiva



Fonte: Autoria própria (2018)

Figura 31 - Dashboard durante monitoramento de motor elétrico a vazio



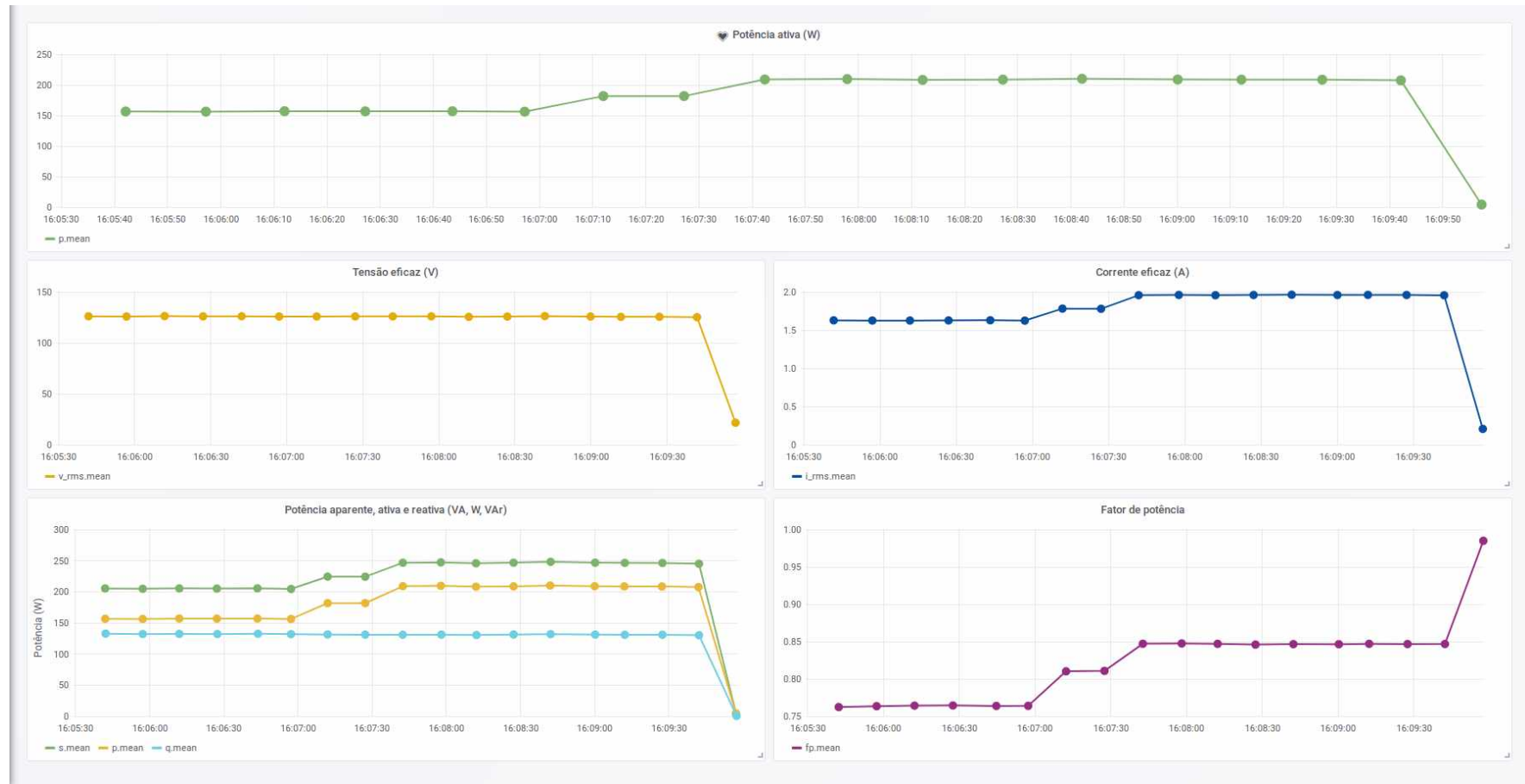
Fonte: Autoria própria (2018)

Figura 32 - Dashboard durante monitoramento de fonte de alimentação



Fonte: Autoria própria (2018)

Figura 33 - Dashboard durante monitoramento de carga capacitiva



Fonte: Autoria própria (2018)