

ANDRÉ LUIZ SIMONAGIO GRANA

A aplicação do conceito de *Internet of Things* ao monitoramento de fontes de alimentação do sistema de sinalização do Metrô de São Paulo

São Paulo
2021

ANDRÉ LUIZ SIMONAGIO GRANA

A aplicação do conceito de *Internet of Things* ao monitoramento de fontes de alimentação do sistema de sinalização do Metrô de São Paulo

Versão Original

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em *Internet of Things*.

Área de Concentração: *Internet of Things*

Orientador: Prof. Dr. Vidal Augusto Zapparoli Castro Melo

São Paulo
2021

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo-na-publicação

Grana, André Luiz Simonagio

A aplicação do conceito de Internet of Things ao monitoramento de fontes de alimentação do sistema de sinalização do Metrô de São Paulo / A. L. S. Grana - São Paulo, 2021.

89 p.

Monografia (MBA em Internet of Things) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1.MANUTENÇÃO PREDITIVA 2.PROCESSAMENTO DE DADOS
3.CIRCUITOS ELETRÔNICOS 4.APRESENTAÇÃO DE INFORMAÇÃO
I.Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

Nome: GRANA, André L. S.

Título: A aplicação do conceito de *Internet of Things* ao monitoramento de fontes de alimentação do sistema de sinalização do Metrô de São Paulo

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em *Internet of Things*.

Aprovado em: 09/02/2021

Banca Examinadora

Prof(a). Dr(a). VIDAL AUGUSTO ZAPPAROLI CASTRO MERO

Instituição: PECE / POLI / USP

Julgamento: APROVADO

Prof(a). Dr(a). Carlos Eduardo Cugnasca



Instituição: Departamento de Enga. de Computação e Sistemas Digitais - EPUSP

Julgamento: Aprovado

Prof(a). Dr(a). Eduardo Ferreira Franco



Instituição: PCS/POLI - USP

Julgamento: Aprovado

AGRADECIMENTOS

À Companhia do Metropolitano de São Paulo pelo incentivo financeiro para realização do MBA em *Internet of Things* e aos gestores e colegas Alexandre de Maio Parpinelli, Jean Marco Rodrigues Fini, Fernando Serafim e Antônio Marcio Barros Silva que apoiaram a iniciativa e a tornaram possível.

Aos amigos Eng.^o Carlos Alfredo Santos e Eng.^o Francisco Carlos Lopes Mathias pelas valiosas informações técnicas concedidas.

Aos colegas da Coordenadoria de Monitoramento de Ativos, pela troca de informações cotidiana que enriqueceu o desenvolvimento realizado e amplia o meu conhecimento diariamente.

Ao Professor Dr. Vidal Augusto Zapparoli Castro Melo, que orientou a elaboração dessa monografia, pela dedicação e contribuições que melhoraram esse trabalho de forma transformadora.

Aos colegas e professores do MBA em IoT que, pelos últimos dois anos, estiveram juntos explorando os temas que alicerçaram a construção desse trabalho.

RESUMO

GRANA, A. L. S. **A aplicação do conceito de *Internet of Things* ao monitoramento de fontes de alimentação do sistema de sinalização do Metrô de São Paulo**. 2020. 90 páginas. Monografia (MBA em *Internet of Things*). Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo. São Paulo. 2021.

Os sistemas de controle do transporte metroferroviário dependem de fontes de alimentação para fornecer energia à diversos subsistemas, sendo consideradas essenciais para a disponibilização correta e segura do serviço de transporte. O presente trabalho propõe o desenvolvimento de uma solução que empregue os conceitos de Internet das Coisas para oferecer um monitoramento constante da tensão fornecida pelas fontes, com a hipótese de permitir a extração de dados distribuídos e conseqüente disponibilização de analíticos de forma centralizada para viabilizar procedimentos de manutenção preditiva. Os requisitos e características para a proposta foram listados a partir da experiência de profissionais do setor de manutenção do Metrô de São Paulo, de roteiros internos de manutenção, de bibliografia sobre arquiteturas de Internet das Coisas, da documentação de fontes de alimentação e da documentação disponível da plataforma de Internet das Coisas denominada FIWARE, utilizada no trabalho. Um protótipo foi desenvolvido e implementado nos níveis de *hardware*, *software* e analíticos centralizados e demonstrou a viabilidade de utilização dos dados coletados como base de inteligência para processos de manutenção preditiva.

Palavras-chave: Internet das Coisas. Monitoramento. Manutenção Preditiva. Fonte de Alimentação.

ABSTRACT

GRANA, A. L. S. **Internet of Things concept applied to Metrô de São Paulo signaling system's power supplies monitoring**. 2020. 90 pages. Monografia (MBA em *Internet of Things*). Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo. São Paulo. 2021.

The energy supply of several subway control systems depends on power supplies, these are considered essential for the correct and safe availability of the transport service. This work proposes the development of a solution that applies the Internet of Things concepts to offer constant voltage monitoring of the power supplies, with the hypothesis of allowing the distributed data extraction and centralized analytics in order to enable predictive maintenance procedures. The requirements and characteristics for the proposal were based on the experience of maintenance professionals of Metrô de São Paulo, on internal maintenance procedures, on Internet of Things architectures bibliography, on power supplies documentation and on the documentation available from the FIWARE Internet of Things platform, which is used in this work. A prototype was developed and implemented at the hardware, software and centralized analytics levels and the feasibility of the use of the data collected as an intelligence base for predictive maintenance processes was demonstrated.

Keywords: Internet of Things. Monitoring. Predictive Maintenance. Power Supply.

LISTA DE ILUSTRAÇÕES

Figura 1: Diagrama de blocos de uma fonte de alimentação.	18
Figura 2: Tensão no ponto “A” da Figura 1.....	19
Figura 3: Tensão no ponto “B” da Figura 1.....	20
Figura 4: Tensão no ponto “C” da Figura 1.	20
Figura 5: Tensão no ponto “D” da Figura 1.	21
Figura 6: Detalhe da tensão de <i>ripple</i>	22
Figura 7: Ilustração de sistema de controle de trens baseado em bloco fixo.	23
Figura 8: Ilustração de um sistema de controle de trens de bloco móvel.	23
Figura 9: Modelo de referência para IoT.	27
Figura 10: Exemplo de arquitetura de aplicação IoT utilizando a plataforma FIWARE.	31
Figura 11: "A cadeia de valor do dado".	37
Figura 12: Arquitetura planejada para o monitoramento de fontes.....	38
Figura 13: Forma de onda de um sinal digitalizado.	41
Figura 14: Sinal completo e sinais separados em tensão de corrente contínua e tensão de corrente alternada por meio de filtros.	43
Figura 15: Representação do optoacoplador 4N25.....	43
Figura 16: Diagrama de blocos do circuito de adequação do sinal.	44
Figura 17: Diagrama de conexões da comunicação SPI de um <i>Master</i> com 3 <i>Slaves</i>	46
Figura 18: Exemplo do paradigma <i>publish-subscribe</i>	47
Figura 19: Arquitetura dos <i>Generic Enablers</i> , módulos e serviços desenvolvida nesse trabalho.	50
Figura 20: Esquemático do circuito projetado para adequação do sinal e isolamento.	52
Figura 21: Diagrama de conexões entre a fonte monitorada e o Microcontrolador com função de conversão AD.	53
Figura 22: Montagem do circuito de adequação do sinal em <i>protoboard</i>	53
Figura 23: Montagem realizada em bancada para teste do módulo de adequação. .	54
Figura 24: Troca de mensagens sobre o canal SPI.	55
Figura 25: Mensagem de solicitação enviada pelo <i>gateway</i> IoT ao conversor AD....	55

Figura 26: Mensagem de resposta enviada pelo conversor AD ao <i>gateway</i> IoT.....	56
Figura 27: Fluxograma do <i>software</i> embarcado no microcontrolador para conversão AD e envio das informações digitalizadas ao <i>gateway</i>	57
Figura 28: Conexões entre o <i>gateway</i> (Raspberry Pi) e o conversor AD (microcontrolador STM32F103).....	58
Figura 29: Detalhe da conexão SPI entre a Raspberry Pi e o microcontrolador STM32F103.....	58
Figura 30: Fluxograma do programa sendo executado no <i>gateway</i> IoT para solicitação dos dados ao conversor AD e envio ao Orion.	59
Figura 31: Configuração do Grafana para conexão com a base de dados CrateDB.	63
Figura 32: <i>Dashboard</i> para teste de visualização dos dados coletados de uma fonte.	63
Figura 33: Alteração visual causada por limiares ultrapassados pelos valores lidos.	64
Figura 34: Filtro passa-alta.....	79
Figura 35: Resposta em frequência do filtro passa-alta.	80
Figura 36: Corrente no coletor do optoacoplador 4N25 em função da corrente no LED e da temperatura ambiente ($V_{ce}=10V$).	81
Figura 37: Diagrama de circuito amplificador diferencial.....	82
Figura 38: Esquemático do circuito de amplificação da componente AC (ripple+ruído).....	83
Figura 39: Variação de CTR pela corrente no LED do optoacoplador (normalizada).	85
Figura 40: Circuito de aquisição simulado no software LTSpice.	86

LISTA DE ABREVIATURAS E SIGLAS

IoT	<i>Internet of Things</i>
SQL	<i>Structured Query Language</i>
GE	<i>Generic Enabler</i>
RMSP	Região Metropolitana de São Paulo
RC	Resistor-Capacitor
CBTC	<i>Communications-Based Train Control</i>
CCSA	<i>China Communications Standards Association</i>
ITU-T	<i>International Telecommunication Union - Telecommunication Standardization Sector</i>
CASAGRAS	<i>Coordination And Support Action for Global RFID-related Activities and Standardisation</i>
IETF	<i>Internet Engineering Task Force</i>
AAA	<i>Authentication, Authorization and Accounting</i>
API	<i>Application Programming Interface</i>
NGSI	<i>Next Generation Services Interface</i>
IEC	<i>International Electrotechnical Commission</i>
EN	<i>European Standard</i>
AD	Analógico-Digital
LED	<i>Light-Emitting Diode</i>
SPI	<i>Serial Peripheral Interface</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo	14
1.2	Justificativa	15
1.3	Método de Pesquisa	15
1.4	Estrutura do Trabalho	15
2	EXPLANAÇÃO DO PROBLEMA	17
2.1	Fontes de Alimentação	18
2.2	Sistema de Sinalização Metroferroviário	22
2.3	Considerações do Capítulo	24
3	CONTEXTUALIZAÇÃO DE IOT	25
3.1	Arquitetura de IoT	27
3.2	Plataforma IoT	28
3.3	Considerações do Capítulo	31
4	LEVANTAMENTO DE REQUISITOS E SOLUÇÕES DE MERCADO	32
4.1	Requisitos	32
4.2	Soluções de Mercado	33
4.3	Análise de aderência	35
4.4	Considerações do Capítulo	36
5	DESENVOLVIMENTO	37
5.1	Digitalização	39
5.2	Adequação do Sinal	42
5.3	Tradução SPI para NCSI	44
5.4	Gerenciador de contexto (<i>Broker</i>)	47
5.5	Armazenamento histórico	48
5.6	Apresentação	49
5.7	Considerações do Capítulo	50

6	EXPERIMENTAÇÃO	51
6.1	Módulo de adequação do sinal e módulo de digitalização	51
6.2	Gateway	57
6.3	Concentrador IoT	61
6.4	Apresentação	62
6.5	Considerações do Capítulo	65
7	CONSIDERAÇÕES FINAIS	67
7.1	Limitações	67
7.2	Conclusão	68
7.3	Contribuições do Trabalho	69
7.4	Trabalhos Futuros	69
	REFERÊNCIAS	71
	GLOSSÁRIO	74
	APÊNDICE A – Colaboração de especialistas para levantamento de requisitos	75
	APÊNDICE B – Projeto do circuito de adequação do sinal	79
	APÊNDICE C – Software desenvolvido para o gateway	87

1 INTRODUÇÃO

A mobilidade na Região Metropolitana de São Paulo (RMSP) possui um componente fundamental na composição das viagens dos milhões de passageiros que se deslocam diariamente por diversos modais, o sistema metroviário.

De acordo com COMPANHIA DO METROPOLITANO DE SÃO PAULO (2019):

O transporte sobre trilhos contribuiu com 2,4 bilhões de viagens, representando 37% do total de viagens motorizadas realizadas na RMSP. Desse total, a rede metroviária operada pelo Metrô, ViaQuatro e ViaMobilidade, com 101,0 km de extensão, seis linhas e 89 estações, foi responsável por 23,4% das viagens na RMSP.

Por consequência desse papel relevante no traslado de tantas pessoas, qualquer evento de indisponibilidade ou atraso na oferta de trens gera prejuízo direto aos passageiros, seja na aglomeração nas plataformas ou no aumento da duração total da viagem.

Nesse contexto, a Companhia do Metropolitano de São Paulo (comumente chamada de Metrô de São Paulo), apresenta, entre as iniciativas de sua estratégia corporativa, a redução de custos, a aplicação de novas tecnologias e modernização do parque instalado.

Essas iniciativas vão ao encontro dos valores apresentados pela empresa, de prover “rapidez, confiabilidade, inovação, economia, segurança, sustentabilidade e mobilidade, preservando o interesse público” (CIA. DO METROPOLITANO DE SÃO PAULO, 2019).

A manutenção eficiente do sistema representa, portanto, uma premissa para atingir os objetivos corporativos apresentados, mantendo a viabilidade econômica da empresa e oferecendo serviço de qualidade à população.

Um dos sistemas vitais para a alta disponibilidade e alta frequência na oferta de trens do Metrô é o sistema de sinalização, que tem como função o controle centralizado dos trens, possibilitando segurança, sincronismo e automatismo. Um defeito que cause

indisponibilidade nesse sistema exige que a operação passe para um controle manual pelos operadores de trens, passando a depender fundamentalmente da comunicação dos operadores com o Centro de Controle, e, como consequência, é necessário aumentar o intervalo entre as composições e reduzir a velocidade de circulação.

Dos diversos componentes do sistema, destacam-se as fontes de alimentação, responsáveis por converter e entregar energia para os equipamentos de um trecho correspondente a duas ou três estações.

Em operação há décadas, uma falha nesse componente pode representar um grande transtorno operacional. Por isso, exigem atenção frequente e demandam inspeções periódicas das equipes de manutenção preventiva para monitorar o funcionamento e tentar agir antecipadamente.

Um sistema crítico, como explicado por RUSHBY (1994), é aquele que, caso apresente uma falha, pode levar a uma consequência inaceitável, seja por causar prejuízo à missão ou por colocar vidas em risco. Por isso os componentes que compõem tais sistemas necessitam possuir confiabilidade elevada aliada a condições de contorno para evitar tais consequências.

1.1 Objetivo

O objetivo deste trabalho é analisar, projetar e construir um protótipo de solução, com base nos conceitos de Internet das Coisas, para monitoramento de fontes de alimentação do sistema de sinalização do Metrô de São Paulo. A solução abrange a aquisição, transformação e transmissão dos dados de tensão de fontes de alimentação para disponibilização de analíticos de maneira centralizada, com o objetivo de permitir a visualização, análise de dados e a implementação de processos de manutenção preditiva.

1.2 Justificativa

O monitoramento remoto e a realização de análises que adicionem inteligência na manutenção de um sistema crítico possibilita maior disponibilidade e melhor gestão de processos de manutenção, com o direcionamento das equipes para atividades essencialmente presenciais.

A necessidade de se obter um monitoramento com custo baixo em um grande volume de equipamentos e a possibilidade de extrair valor dos dados coletados a partir do sensoriamento remoto, demonstra a aderência das demandas aos conceitos de Internet das Coisas.

O desenvolvimento de um projeto de IoT que viabilize a coleta e monitoramento de valores de tensão das fontes pode ser aplicado também a outros sistemas que possuam grandezas similares, tanto em componentes mais modernos e digitais quanto em componentes legados e analógicos.

1.3 Método de Pesquisa

O trabalho utilizou como métodos de pesquisa principais a pesquisa aplicada e a pesquisa experimental, por propor e prototipar uma aplicação prática de tecnologias de ponta para solução de problema específico, com definição exata do problema e da hipótese. A pesquisa aplicada foi apoiada pela pesquisa exploratória, baseada no levantamento de referências teóricas e conversa com especialistas da área.

1.4 Estrutura do Trabalho

O capítulo 1 INTRODUÇÃO apresenta as motivações, o objetivo, a justificativa e a estrutura do trabalho.

O capítulo 2 EXPLANAÇÃO DO PROBLEMA apresenta o problema abordado e o ambiente no qual está inserido, considerando suas particularidades.

O capítulo 3 CONTEXTUALIZAÇÃO DE IOT traz os conceitos de IoT e o embasamento para construção de uma solução IoT para ser utilizada em ambientes de missão crítica.

O capítulo 4 LEVANTAMENTO DE REQUISITOS E SOLUÇÕES DE MERCADO apresenta um levantamento de alternativas ao desenvolvimento proposto que sejam amplamente disponíveis no mercado, suas características e limitações.

O capítulo 5 DESENVOLVIMENTO mostra o estudo desenvolvido para obtenção de uma solução completa onde são aplicados os conceitos de IoT ao problema analisado, e aborda-se desde o desenho de uma arquitetura básica até o planejamento dos módulos necessários à solução.

O capítulo 6 EXPERIMENTAÇÃO demonstra as etapas de construção de um protótipo de acordo com o desenvolvimento realizado e as funções obtidas com cada módulo.

O capítulo 7 CONSIDERAÇÕES FINAIS delimita os aspectos do objetivo que foram atingidos com a solução proposta, as contribuições do desenvolvimento realizado e possibilidades de continuidade do trabalho.

Em REFERÊNCIAS são apresentados os materiais citados no texto e utilizados como fontes consultadas para o desenvolvimento desse trabalho.

No APÊNDICE A são apresentadas as colaborações de profissionais do ramo de manutenção que atuam no Metrô de São Paulo para subsidiar o levantamento de requisitos.

O APÊNDICE B contém o detalhamento do projeto de um circuito de adequação de sinal desenvolvido no trabalho.

No APÊNDICE C é apresentado o código de um *software* construído no desenvolvimento do protótipo.

2 EXPLANAÇÃO DO PROBLEMA

A manutenção no Metrô de São Paulo, como na maioria dos sistemas, subdivide-se em três tipos, preventiva, corretiva e preditiva. A preventiva é baseada em planos de manutenção periódicos, ou seja, com base em intervalos de tempo ou com base no número de ciclos de operação determinados por análises estatísticas do histórico de falhas e por especificações dos fabricantes. A corretiva é acionada apenas quando o sistema apresenta algum defeito ou indisponibilidade, ou seja, quando ocorrem falhas durante a operação. Por fim, a preditiva é baseada no monitoramento de sinais coletados de forma a auxiliar o planejamento do melhor momento para atuar em um equipamento, no sentido de evitar falhas, ou seja, "... utilizar a condição real do equipamento para buscar uma melhor operação" e obter um menor custo total do ciclo de vida de um sistema (MOBLEY, 2002).

Portanto, com vistas a se obter maior disponibilidade e atuações mais assertivas, é desejável que seja aplicada a manutenção preditiva aos equipamentos, o que depende da coleta de dados e informações para entender o estado de um equipamento.

Ao analisar aspectos da manutenção do sistema de sinalização do Metrô de São Paulo, constata-se que as fontes de alimentação de equipamentos mais antigos não possuem nenhum sensoriamento para coleta de dados. Esses dispositivos apresentam papel fundamental na operação do Metrô, exigem inspeções e atuações frequentes para mitigar possíveis problemas e suas construções atuais permitem somente as técnicas preventivas e corretivas de manutenção.

De forma a detalhar esses componentes e onde se inserem nos sistemas do Metrô, são apresentados nos itens a seguir os conceitos de Fontes de Alimentação e do Sistema de Sinalização do Metrô de São Paulo.

2.1 Fontes de Alimentação

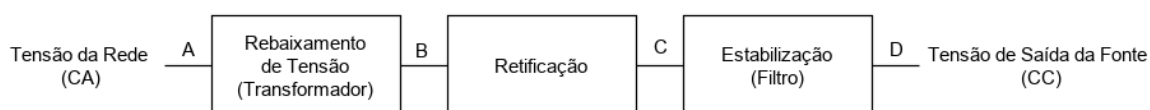
As fontes de alimentação exercem função de conversão de energia em sistemas elétricos e eletrônicos. Em uso residencial, a maioria apresenta a característica de conversão de tensão de fornecimento das concessionárias, em corrente alternada, para tensão compatível com equipamentos eletrônicos em corrente contínua.

Podem ser divididas em fontes chaveadas ou lineares. No primeiro tipo, um circuito eletrônico utiliza um pulso de alta frequência em conjunto com filtros para obter a tensão desejada na saída da fonte. Já no segundo, utiliza-se um transformador e um circuito para retificação e estabilização do sinal (ELLIOTT, 2001).

Tipos diferentes de fontes possuem indicadores de desempenho diferentes. Nesse trabalho, a solução proposta foca em fontes lineares retificadas, por representarem a maior parte das fontes presentes no sistema de sinalização do Metrô de São Paulo.

A análise de funcionamento de uma fonte linear retificada permite desenhar de maneira simplificada o diagrama de blocos da Figura 1, onde a energia elétrica distribuída pela rede em corrente alternada passa por três etapas de conversão até ser fornecida em corrente contínua na saída da fonte: o rebaixamento de tensão, a retificação e a estabilização do sinal de saída.

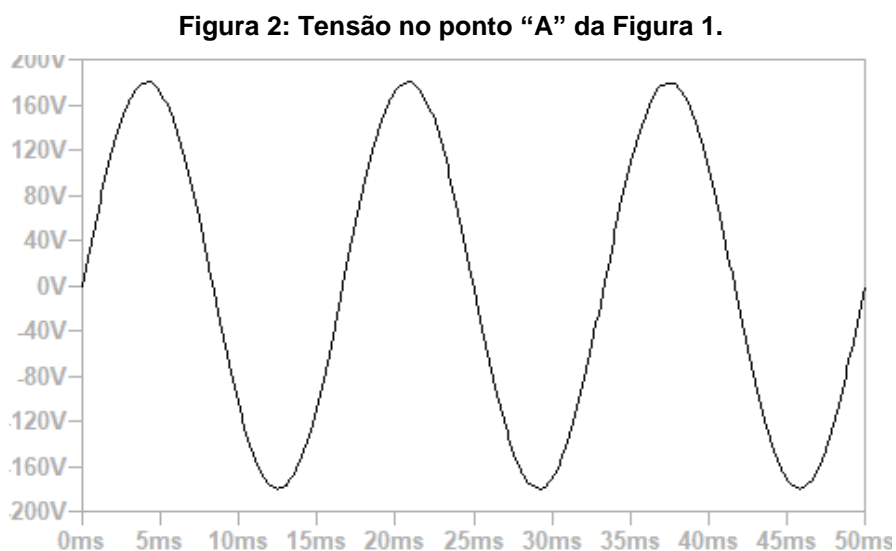
Figura 1: Diagrama de blocos de uma fonte de alimentação.



Fonte: Próprio autor.

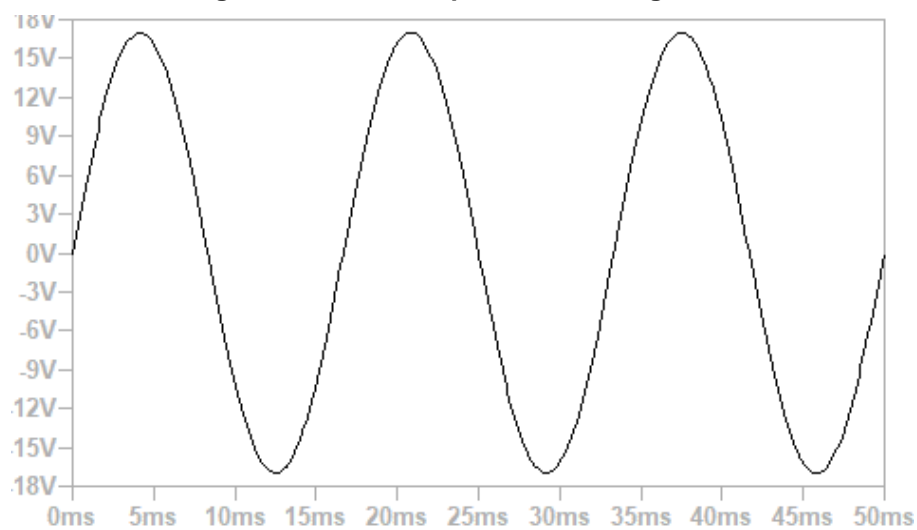
Para exemplificar o funcionamento das fontes desse tipo, serão apresentados gráficos da variação da tensão, em Volts, em relação ao tempo. Os gráficos são divididos para cada etapa do processo de conversão de energia realizado pela fonte e representados pelas letras "A", "B", "C" e "D" da Figura 1, desde a entrada da tensão alternada fornecida pela rede até a saída com tensão contínua necessária para operação dos dispositivos.

O gráfico que representa a variação da tensão no tempo observada no ponto "A" é representado na Figura 2. Observa-se que a tensão geralmente fornecida pelas concessionárias de energia possui a forma senoidal e oscila em uma frequência de 60Hz, ou seja, 60 ciclos por segundo, totalizando 16,67 ms por ciclo.



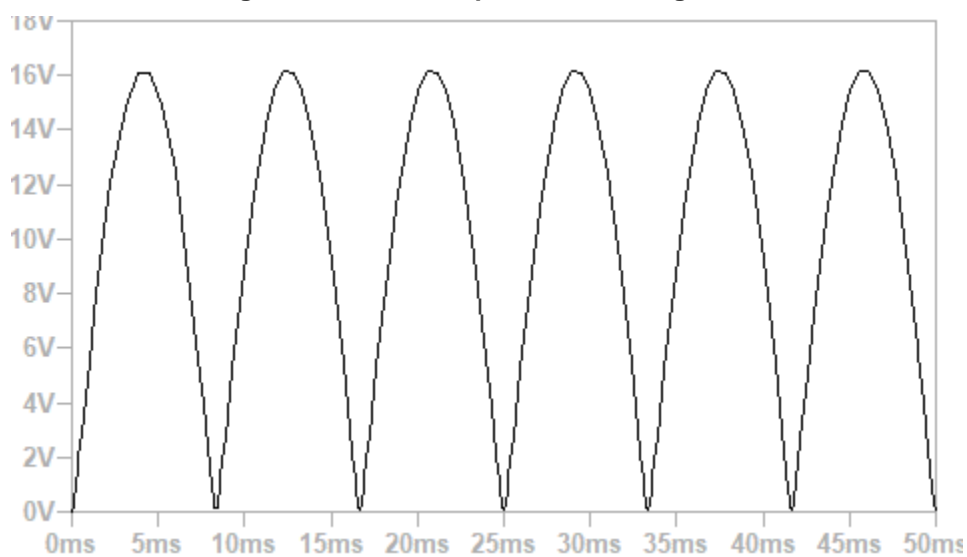
Fonte: Próprio autor.

A Figura 3 apresenta uma observação no ponto "B", após a primeira etapa da conversão da fonte linear, caracterizada pelo rebaixamento da tensão. Nessa etapa é utilizado um transformador para levar a tensão da rede para um valor próximo ao desejado na saída da fonte. No exemplo, a tensão que estava oscilando entre 180 V e -180 V foi rebaixada e agora oscila entre +17 V e -17 V.

Figura 3: Tensão no ponto "B" da Figura 1.

Fonte: Próprio autor.

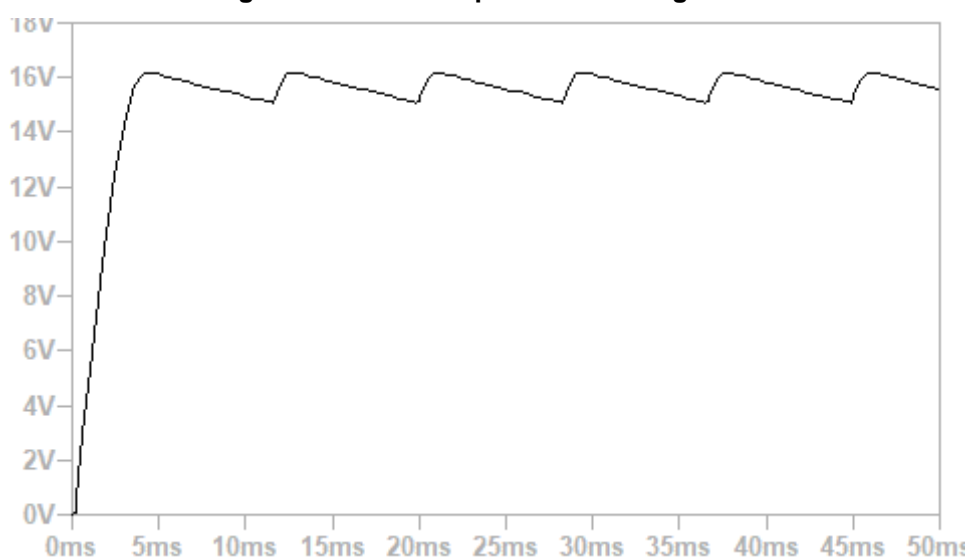
A próxima etapa do processo é chamada de retificação. Essa transformação utiliza diodos, que são semicondutores capazes de bloquear o fluxo de corrente em um sentido. Com uma associação de diodos, chamada de ponte de diodos, pode-se direcionar a parte negativa do ciclo da tensão alternada para que fique positiva. A forma de onda observada após a retificação, e caracterizada pelo ponto "C", está ilustrada na Figura 4.

Figura 4: Tensão no ponto "C" da Figura 1.

Fonte: Próprio autor.

Por fim, para obtenção da forma de onda desejada, que, para esse exemplo, é uma tensão constante de 15 V, utiliza-se um filtro capacitivo, geralmente composto de um capacitor associado em paralelo com a carga, responsável por armazenar a energia e entregá-la para a carga nos períodos onde a tensão fica mais baixa. A Figura 5 apresenta a forma de onda que pode ser observada após a estabilização, o que caracteriza o ponto "D".

Figura 5: Tensão no ponto "D" da Figura 1.

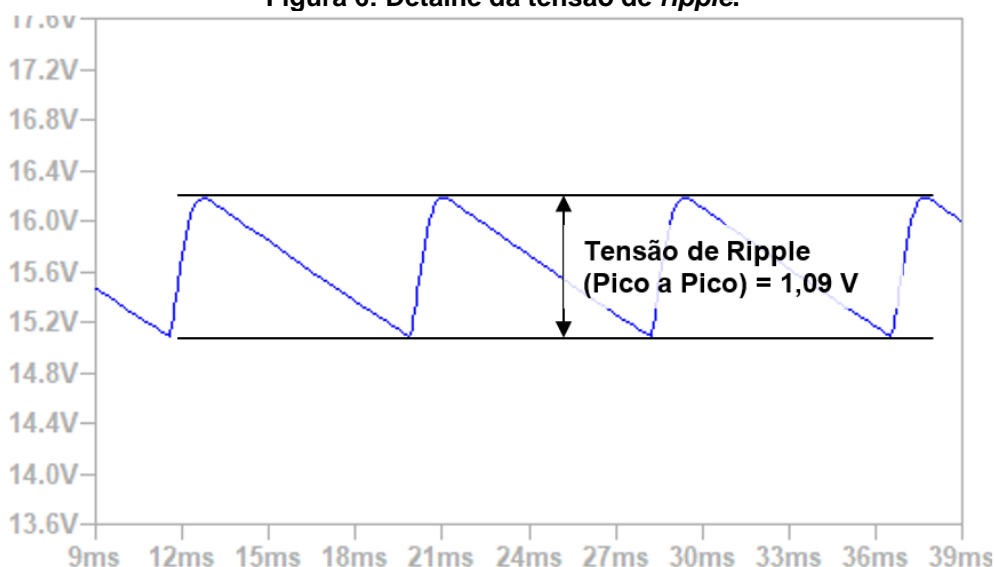


Fonte: Próprio autor.

Para complementar o entendimento do processo, a Figura 6 destaca o *ripple*, parâmetro utilizado para explicar o quanto a tensão está oscilando em torno do valor desejado. O sinal apresentado é uma ampliação do apresentado na Figura 5 após o tempo de 5 milissegundos. Esse comportamento é resultante da estabilização da fonte com a utilização de capacitores e sua intensidade é resultante de uma relação entre a quantidade de energia que o capacitor consegue armazenar e a quantidade de energia consumida pela carga.

O capacitor é um componente que pode sofrer degradação por diferentes fatores que causam a alteração na quantidade de energia capaz de ser armazenada, ou seja, redução na capacitância do componente, resultando em alterações no comportamento original da fonte (GUPTA, YADAV, DEVOTO, & MAJOR, 2018).

Figura 6: Detalhe da tensão de ripple.



Fonte: Próprio autor.

2.2 Sistema de Sinalização Metroferroviário

As principais funções de um sistema de sinalização em um sistema de transporte ferroviário são: controlar a velocidade e posicionamento dos trens de forma eficiente e segura; controlar paradas para embarque e desembarque; e comandar os equipamentos de mudança de via para atender às necessidades operacionais.

Os sistemas de sinalização automáticos se baseiam principalmente em dois tipos: com bloco fixo e com bloco móvel. O bloco é um trecho de via que pode estar ocupado ou não por um trem ou outro veículo de via, e a esse bloco é designado uma velocidade máxima (informação também chamada de código de velocidade) que é enviada para os trens.

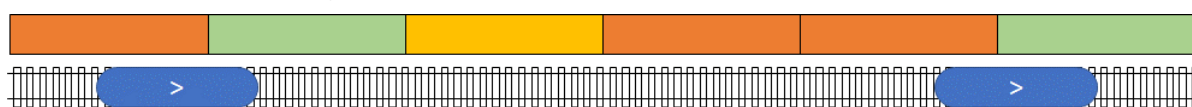
Em um sistema com bloco fixo, os trechos de via são alocados conforme a instalação do sistema e no Metrô de São Paulo os blocos fixos possuem tamanhos variados entre 200 metros e 400 metros. Os equipamentos do sistema de sinalização utilizam modulação em frequência para enviar, através dos trilhos, a informação para os trens, que decodificam o sinal e, ao mesmo tempo, representam uma impedância naquele trecho de via que é percebida pelo sistema como a presença de um veículo naquele local.

Conforme apresenta MENEZES & ALVES (2015):

O sistema de sinalização do Metrô de São Paulo foi desenvolvido na década de 70 pela Westinghouse para a linha 1, tendo o Metrô adquirido, além dos equipamentos, o projeto desse sistema. Ao longo do tempo esse sistema foi sendo aprimorado pelo Metrô e pela Westinghouse e suas sucessoras, as empresas CMW e Alstom, operando atualmente esse sistema nas linhas 1, 2, 3 e 5, permitindo hoje intervalo entre trens inferiores a 120 s.

Na Figura 7 estão ilustrados os blocos em relação à ocupação de trens em um sistema de bloco fixo. A cor vermelha indica proibição para avançar, amarela um código de velocidade intermediário para frenagem e verde um código de velocidade que permite avançar.

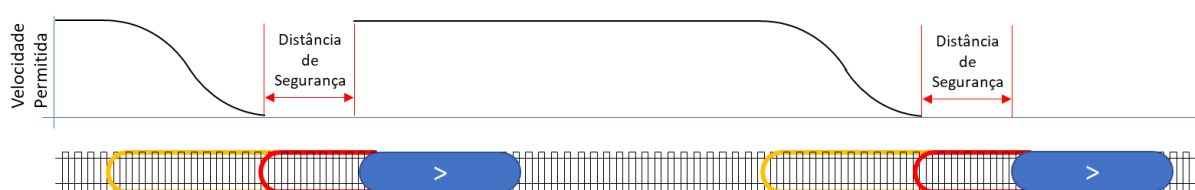
Figura 7: Ilustração de sistema de controle de trens baseado em bloco fixo.



Fonte: Adaptado de MENEZES & ALVES (2015).

No caso de blocos móveis, também chamado de *Communications-Based Train Control* (CBTC) ou sistema de controle de trens baseado em comunicação, o sistema mantém a localização de cada trem e troca informações de velocidade por meio de redes de dados. Conforme ilustrado na Figura 8, o sistema cria blocos virtuais que se deslocam em conjunto com o trem, formando bolhas de ocupação, representadas na figura pelos contornos vermelhos e amarelos. Dessa forma, a eficiência de utilização da via pode ser maior se comparada a um sistema com o bloco fixo, ou seja, os trens podem ficar mais próximos uns aos outros mantendo as premissas de segurança.

Figura 8: Ilustração de um sistema de controle de trens de bloco móvel.



Fonte: Adaptado de MENEZES & ALVES (2015).

Hoje o Metrô de São Paulo utiliza os dois tipos de sistemas, sendo fundamental para a alta disponibilidade desse meio de transporte a correta manutenção de todos seus componentes. Tanto os sistemas de blocos fixos baseados em modulação de sinais quanto os sistemas de blocos móveis baseados em comunicação são compostos por equipamentos eletrônicos e, conseqüentemente, possuem fontes de alimentação elétrica para funcionar. Em especial para os sistemas baseados em modulação em frequência, é fundamental que as fontes de alimentação apresentem um sinal estável, sem ruídos e sem interferências.

2.3 Considerações do Capítulo

Esse capítulo contextualizou o cenário no qual o presente trabalho foi elaborado e introduz os conceitos do sistema e do componente que serão alvo do trabalho. As principais vantagens da aplicação do conceito de IoT no monitoramento das fontes de alimentação são a possibilidade de uma análise integrada e comparativa dos dados coletados e o monitoramento contínuo, permitindo uma interface facilitada com os técnicos de manutenção. Com a existência de uma base de dados histórica, abre-se ainda a possibilidade de aplicação de inteligência artificial para criação de valor sobre o monitoramento existente.

3 CONTEXTUALIZAÇÃO DE IOT

O termo Indústria 4.0 foi originalmente introduzido pelo governo alemão, na forma de uma iniciativa estratégica para manter a competitividade da indústria. Essa iniciativa teve como objetivo a aplicação de novas tecnologias e a digitização na indústria. Um dos habilitadores para a Indústria 4.0 é a aplicação do conceito IoT na indústria (KAGERMANN, WAHLSTER, & HELBIG, 2013).

Para analisar a aplicabilidade do conceito IoT na indústria, algumas definições do termo *Internet of Things* publicadas por diferentes órgãos de regulamentação e padronização referências mundiais são apresentadas a seguir.

A *China Communications Standards Association* (CCSA), uma associação chinesa para desenvolvimento de padrões para telecomunicações, define IoT como “Uma rede que pode coletar informações do mundo físico e controlar o mundo físico por meio de diversos dispositivos instalados com capacidade de percepção, processamento, execução e comunicação, e suportar comunicações entre humanos e coisas ou entre coisas através da transmissão, classificação e processamento de informações” (CHEN, XU, LIU, HU, & WANG, 2014, tradução nossa).

O *International Telecommunication Union - Telecommunication Standardization Sector* (ITU-T), setor ligado às Nações Unidas para produção de padrões para todos os campos das telecomunicações e tecnologias de informação e comunicação, define IoT como “uma infraestrutura global para a sociedade da informação, possibilitando serviços avançados por meio da interconexão (física e virtual) das coisas baseada nas tecnologias interoperáveis de informação e comunicação existentes e em desenvolvimento” (CHEN, XU, LIU, HU, & WANG, 2014, tradução nossa).

A *Coordination And Support Action for Global RFID-related Activities and Standardisation* (CASAGRAS), projeto filiado à União Europeia com a participação de organizações da China, Japão, Coréia e EUA, define IoT como “uma infraestrutura global de rede, conectando objetos físicos e virtuais explorando captura de dados e capacidade de comunicação” (CHEN, XU, LIU, HU, & WANG, 2014, tradução nossa).

A *Internet Engineering Task Force* (IETF) aborda o conceito IoT como “uma rede mundial de objetos interconectados e individualmente endereçáveis baseada em protocolos de comunicação padrões” (CHEN, XU, LIU, HU, & WANG, 2014, tradução nossa).

Apesar das nuances nas definições do termo *Internet of Things* apresentadas, é comum o contexto de existência de conexão entre o mundo físico e o mundo virtual; de um grande volume de dispositivos; da agregação de capacidades de localização, sensoriamento, controle, conectividade e integração de dados; com aplicações em diferentes verticais incluindo a indústria.

Para guiar os desenvolvimentos com base nesse conceito, as organizações internacionais de padronização criaram modelos de referência que apresentam, de forma macro, funções básicas para guiar a construção de soluções. Os modelos apresentam uma arquitetura a ser seguida sem restringir como devem ser os relacionamentos entre os componentes que farão parte de uma solução.

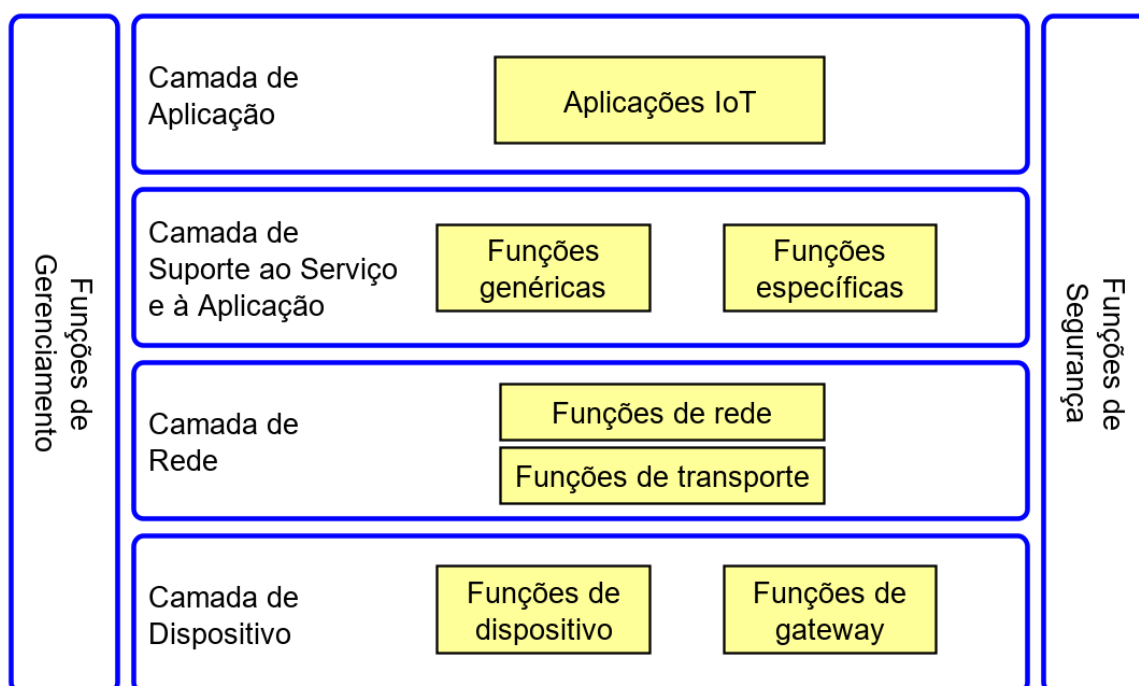
Com o sucesso da aplicação dos conceitos e modelos de referência, surgiram iniciativas para detalhar a interoperabilidade entre soluções e componentes de diferentes fabricantes e plataformas que especificam e implementam a comunicação e o fluxo da informação dentro de uma solução. Esses desenvolvimentos e disponibilizações são úteis para facilitar uso e agilizar os desenvolvimentos pretendidos, pois já implementam pontos fundamentais que são necessários na maioria das soluções.

No item 3.1 será apresentado o modelo de referência para IoT criado pela *International Telecommunication Union - Telecommunication Standardization Sector* e no item 3.2. é apresentada a plataforma FIWARE de interoperabilidade entre componentes, utilizados, respectivamente, como base de arquitetura e de implementação para este trabalho.

3.1 Arquitetura de IoT

Para obtenção de uma solução IoT de forma escalável e segura, é recomendada a utilização de um modelo de referência como base para criação de uma arquitetura, a fim de que todos os componentes respeitem requisitos consolidados e um padrão de implantação. O modelo de referência detalhado em ITU-T (2012) e o diagrama proposto nesse documento está reproduzido na Figura 9.

Figura 9: Modelo de referência para IoT.



Fonte: Adaptado de ITU-T (2012, pp.7, tradução nossa).

De cima para baixo na figura, a "Camada de Aplicação" contempla o resultado final disponibilizado pelo sistema, ou seja, a própria aplicação que será suportada por todas as outras camadas.

Em seguida, a "Camada de Suporte ao Serviço e à Aplicação" contempla recursos como conversão, processamento e armazenamento de dados, sejam eles genéricos e aplicáveis a diversas soluções, ou específicos da aplicação fim do sistema.

Logo abaixo, a "Camada de Rede" deve conter capacidades para prover conectividade, incluindo autenticação, autorização e registro, também conhecidos pelo termo em inglês *Authentication, Authorization and Accounting* (AAA).

Em seguida, a "Camada de Dispositivo" deve implementar o acesso do dispositivo à infraestrutura IoT, seja enviando dados diretamente para a rede ou exercendo a função de *gateway* para integrar interfaces da Camada do Dispositivo com a Camada de Rede.

Ainda de acordo com o modelo de referência analisado, a função de segurança deve estar presente e ser transversal entre as demais camadas:

- na Camada de Aplicação: com autorização, autenticação, confidencialidade e integridade dos dados; proteção à privacidade; auditoria; e antivírus;
- na Camada de Rede: com autorização, autenticação, confidencialidade e proteção à integridade das sinalizações de rede;
- na Camada de Dispositivo: com autenticação, autorização, confidencialidade e integridade dos dados e dos dispositivos.

O modelo de referência também apresenta funções de gerenciamento, contempla gestão de dispositivos e da infraestrutura de rede e provê facilidades de ativação, atualização e alocação de recursos.

Modelos de referência são, portanto, guias para a criação de uma arquitetura e nem sempre todas as funções e camadas indicadas precisam estar presentes nos desenvolvimentos.

3.2 Plataforma IoT

Para convencionar o fluxo dos dados dentro de uma solução de IoT e permitir uma cooperação abrangente no desenvolvimento de módulos que exercem diferentes

funções, foram desenvolvidas plataformas como o FIWARE¹, o OpenIoT² e o dojot³, criados por diferentes instituições.

Para este trabalho foi escolhido o FIWARE, por já existir conhecimento do autor na sua utilização e por existir diversos desenvolvimentos sobre licenças *open-source* disponíveis para aproveitamento.

Conforme descrito pela FIWARE Foundation (2020, tradução nossa):

A plataforma FIWARE é um conjunto de componentes de código aberto que podem ser conectados entre si e com componentes de plataformas terceiras para construir soluções inteligentes de forma mais rápida, fácil e barata. Uma API poderosa, ainda que simples (FIWARE NGSI) possibilita a integração e provém a base para a interoperabilidade e replicação (portabilidade) de soluções inteligentes.

A plataforma FIWARE é composta por dois elementos principais:

- os componentes: também chamados de *generic enablers*, exercem funções como armazenamento, gerenciamento de informações, segurança, interface com dispositivos, análise e visualização;
- a API, do termo inglês *Application Programming Interface*, caracterizada por um conjunto de regras para delimitar como cada módulo poderá ser acionado e utilizado.

Cada componente pode ser considerado como um *software* capaz de receber e enviar requisições a outros componentes para executar uma função específica. Os componentes podem estar espalhados nas diversas camadas da arquitetura. Por exemplo:

- um componente na Camada de Dispositivo faz a tradução entre as requisições da plataforma e o mundo físico;

¹ Informações disponíveis em FIWARE Foundation (2020).

² Projeto *open-source* financiado pela União Europeia, com detalhes em OpenIoT FP7 Project (2021), disponível em <https://github.com/OpenIoT/Organisation>.

³ Mais informações à disposição na página oficial do projeto, conduzido pelo CPQD, DOJOT (2017), disponível em "<http://www.dojot.com.br/>".

- um componente na Camada de Suporte ao Serviço e à Aplicação realiza armazenamento ou transformação de dados;
- um componente na Camada de Aplicação executa consolidação e análise de dados.

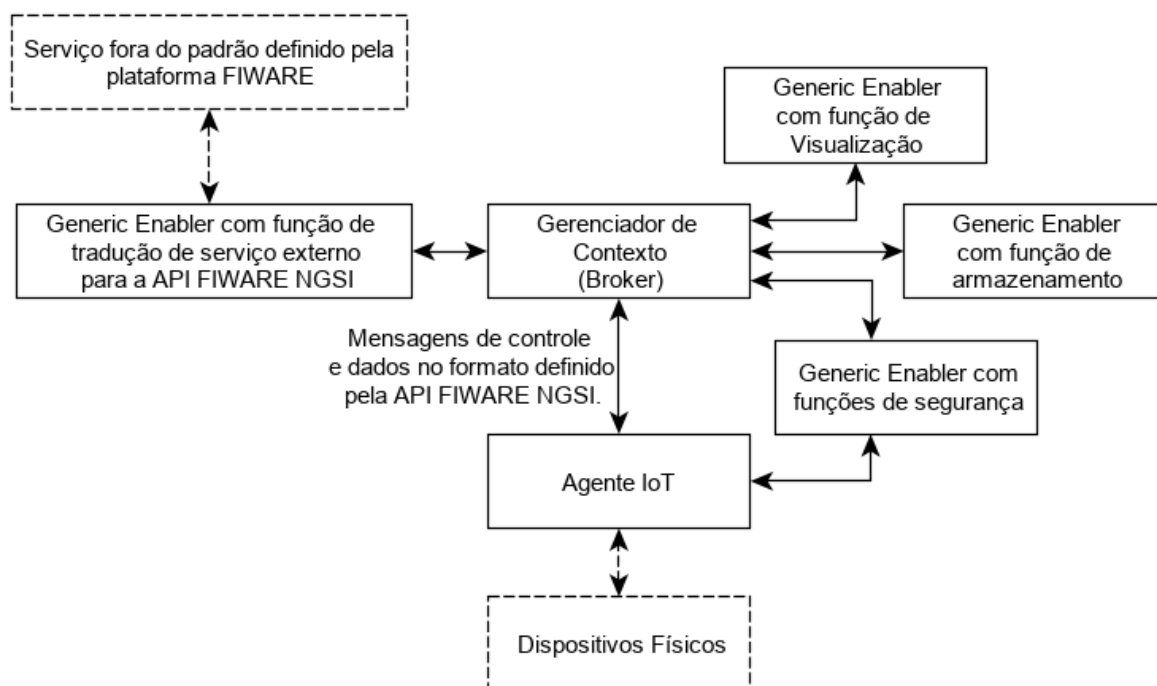
Obrigatoriamente, em uma aplicação construída sobre a plataforma FIWARE, deve existir um componente com a função de gerenciamento de contexto, denominado pela Plataforma de *broker*, que disponibilizará a troca de mensagens entre os demais componentes.

Com o objetivo de facilitar a compreensão, a Figura 10 apresenta um diagrama que exemplifica a relação entre componentes de uma solução baseada em FIWARE. Os componentes e conexões tracejados são partes da arquitetura que não são definidas pela FIWARE.

Todas as mensagens entre os componentes, sejam elas de dados coletados de sensores, comandos enviados para atuadores ou controle entre os componentes, devem ser realizadas com a API FIWARE NGSI.

Componentes com função de tradução e com função de interface com os dispositivos físicos são necessários para possibilitar a troca de informações de uma solução construída sobre essa plataforma com um elemento externo não compatível diretamente com a interface.

Figura 10: Exemplo de arquitetura de aplicação IoT utilizando a plataforma FIWARE.



Fonte: Próprio autor.

3.3 Considerações do Capítulo

Nesse capítulo, conceituou-se o termo IoT e foi apresentado um modelo de referência a ser seguido para construção de uma solução. A disponibilização, em um ambiente centralizado, dos dados das fontes do sistema de sinalização do Metrô vai ao encontro do conceito de conexão entre o mundo físico e virtual relativo à Internet das Coisas, com um grande número de dispositivos a serem interconectados. O modelo de referência habilita a solução a ser expansível de forma a obter o melhor resultado possível a partir dos dispositivos conectados e a plataforma FIWARE permite a prototipação rápida para validação de conceitos e evolução de uma proposta em busca de um produto.

4 LEVANTAMENTO DE REQUISITOS E SOLUÇÕES DE MERCADO

Antes do início dos desenvolvimentos objeto do trabalho, foi realizada uma etapa de levantamento de requisitos técnicos necessários ao cenário das fontes de alimentação do Metrô de São Paulo e das soluções disponíveis no mercado. O trabalho de levantamento possibilitou uma análise de aderência entre os requisitos e as funcionalidades das soluções já existentes.

O levantamento de requisitos, das soluções de mercado e a análise são apresentados de forma separada nos subitens a seguir.

4.1 Requisitos

A consolidação dos requisitos da solução foi feita a partir da análise de documentos de manutenção preventiva internos do Metrô e baseada em informações fornecidas pelo Engenheiro Francisco Carlos Lopes Mathias, responsável pelo planejamento da manutenção do sistema de sinalização de bloco fixo do Metrô de São Paulo, e pelo Supervisor de Manutenção Carlos Alfredo dos Santos, que possui experiência prática na manutenção dos ativos tratados nesse trabalho. As colaborações estão contidas no Apêndice A desse documento.

Os principais requisitos levantados foram:

1. **valor médio da tensão contínua sendo fornecida pelas fontes:** considerada a principal variável que pode indicar a existência de um problema em um componente;
2. **tensão de ripple:** frequentemente se observa uma alteração dessa característica em sistemas em degradação, e os roteiros de inspeção periódica sempre orientam sua aferição. O sistema deve ser capaz de perceber variações do ripple na ordem de 20 mV, conforme limiares encontrados no roteiro de manutenção interno do Metrô (CIA. DO METROPOLITANO DE SÃO PAULO, 2020).

3. **certificações de segurança:** deve ser realizada a devida análise para que o dispositivo conectado não coloque em risco as condições de segurança do sistema que está sendo alimentado pela fonte;
4. **isolamento:** garantir a não interferência nos circuitos existentes e minimizar a quantidade de componentes conectados eletricamente à fonte, o que pode aumentar o risco de uma falha no dispositivo de coleta adicionar um sinal inesperado no ponto de alimentação do sistema de sinalização;
5. **acesso aos locais onde estão instalados os componentes a serem monitorados:** em cada local a ser monitorado podem existir mais de dez pontos de tensão a serem medidos, com tensão nominal de 6, 12 ou 28 V, positivos e negativos, é necessário que o equipamento de leitura possua a capilaridade para conexão com todos esses sinais;
6. **não demandar intervenções frequentes:** por estarem espalhados em diversos locais e possuírem acesso restrito em horários fora da operação comercial do Metrô;
7. **criação de alarmes:** para permitir rápida atuação das equipes de manutenção em caso de falhas.

4.2 Soluções de Mercado

Com o objetivo de avaliar possíveis soluções análogas ao objetivo proposto no projeto, este capítulo apresenta um levantamento soluções amplamente disponíveis no mercado, bem como suas características e limitações.

Para que sejam compatíveis com a aplicação abordada no trabalho, buscou-se equipamentos com certificações definidas em normas internacionais. Para as fontes de alimentação de uso industrial, podem ser citadas as normas IEC 62368 e IEC 61558, que dizem respeito à segurança elétrica do equipamento. Para questões que abordam o aspecto de imunidade em relação à interferência e emissividade eletromagnética, as normas de referência são as EN 61000, EN 55024 e EN 55032.

Portanto, a pesquisa por soluções de fontes de alimentação comerciais com características de Internet das Coisas definiu como premissa o atendimento às normas citadas ou equivalentes e foi realizada em dois distribuidores de componentes eletrônicos mundialmente conhecidos (Digi-Key Electronics e Mouser Electronics) e nos catálogos das fabricantes Mean Well e TDK Lambda.⁴

A Tabela 1 apresenta um resumo das fontes encontradas com as características descritas.

Tabela 1: Fontes de alimentação analisadas.

Fabricante	Modelo	Comunicação	Certificações⁵
MEAN WELL	PHP-3500	PMBus	IEC 62368
		CANBus	IEC 55032
MEAN WELL	HEP-1000	PMBus	IEC 62368
		CANBus	IEC 55032
TDK Lambda	TPS4000	PMBus	IEC 62368
			IEC 55032
Delta Electronics	IMA-S600-12-ZYPLI	PMBus	IEC 62368
			IEC 55032
ABB	MPR0712TE	PMBus	IEC 60950
			IEC 55032

Fonte: Elaborada pelo autor com base em ABB (2020), Delta Electronics (2016), TDK-Lambda (2020) e MEAN WELL (2020) .

A fabricante TDK-Lambda descreve na documentação do modelo TPS4000 as seguintes informações disponíveis para leitura através do protocolo PMBus: tensão de saída, corrente de saída, temperatura interna, estado e alarmes de funcionamento

⁴ Os fabricantes Mean Well e TDK Lambda foram escolhidos pois figuram no entre as 10 maiores empresas do setor segundo a publicação da EPSNEWS (2014) disponível em "<https://epsnews.com/2014/05/12/top-10-power-supply-rankings-shift/>". A publicação aponta para outras referências, porém não foi possível encontrar informações que contribuíssem para o trabalho.

⁵ Escolhida uma certificação sobre segurança elétrica e uma sobre o tema de emissividade eletromagnética. Todos os modelos possuem mais certificações.

e informações de fabricação. Entretanto, não está especificado no documento informação sobre a taxa de amostragem das leituras (TDK-LAMBDA, 2020).

A MEAN WELL informa no manual de ambos modelos pesquisados as seguintes características disponíveis por PMBus: tensão de entrada, tensão de saída, corrente de saída, temperatura interna, estado e alarmes de funcionamento e informações de fabricação. A precisão da leitura da tensão de saída informada é de ± 240 mV e apesar da velocidade de comunicação ser especificada em 100 kHz, a taxa de amostragem não é especificada (MEAN WELL, 2020).

Na folha de dados do equipamento fabricado pela ABB, estão listadas as seguintes variáveis para leitura são: tensão de entrada, tensão de saída, corrente de entrada, corrente de saída, temperatura interna, velocidade de rotação da ventilação forçada do módulo, estado e alarmes de funcionamento e informações de fabricação. A velocidade de comunicação é especificada entre 100 e 400 kHz, configurável, porém a precisão e taxa de amostragem não são descritas (ABB, 2020).

As informações públicas relacionadas à implementação do protocolo pela fabricante Delta Electronics não trazem detalhes das variáveis disponíveis.

4.3 Análise de aderência

As fontes encontradas que possuem recurso de comunicação para comando remoto e informação de estado de funcionamento e falhas utilizam os protocolos PMBus ou CANBus, sendo o primeiro comum entre todas as fontes com monitoramento embutido encontradas. Essa adoção do protocolo PMBus permite que uma solução que o utilize para sensoriamento das fontes seja compatível através de diferentes fabricantes.

Entretanto, a documentação encontrada não detalha informações como precisão dos valores lidos e taxa de amostragem, o que pode limitar as informações extraídas a partir do monitoramento e impossibilita o atendimento ao requisito 2 (leitura de tensão de *ripple*) listado no item 4.1.

Todos os outros requisitos: monitoramento da tensão média, certificados de segurança e isolamento, integração com os pontos monitorados, alta disponibilidade e criação de alarmes podem ser atendidos com a utilização das fontes listadas em uma plataforma IoT.

A utilização das fontes comerciais encontradas depende da modernização do parque instalado, composto por centenas de fontes de alimentação. O trabalho desenvolvido representa uma alternativa à essa substituição, pois possibilita o monitoramento de fontes existentes e mantém a compatibilidade em caso de troca por outros modelos, independente do fabricante.

4.4 Considerações do Capítulo

Foram apresentados os requisitos necessários ao projeto e as alternativas existentes ao desenvolvimento que foi realizado nesse trabalho, com ressalvas em relação à taxa de amostragem e à precisão dos dados fornecidos pelas fontes comerciais, o que impossibilita a medida do *ripple* sem desenvolvimento adicional.

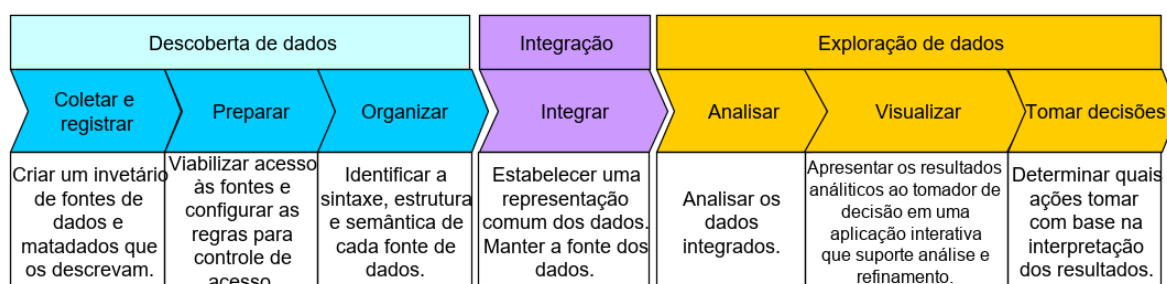
O fator custo não foi avaliado pela dificuldade em obter precificação desses componentes no mercado nacional, pois possuem aplicação específica e poucos clientes.

5 DESENVOLVIMENTO

Para criação de uma proposta de solução para suprir a necessidade de manutenção preditiva das fontes de alimentação dos sistemas de sinalização com base nos conceitos de IoT e que atendam aos requisitos levantados, optou-se pelo desenvolvimento de um dispositivo que possa ser conectado às fontes de forma a medir os sinais que representem seu estado de funcionamento e que possua conexão com uma plataforma capaz de armazenar, apresentar e disponibilizar os dados para análises.

Em Miller & Mork (2013) é apresentada uma cadeia de valor para melhor compreensão de como os dados geram resultados. Essa cadeia de valor está apresentada na Figura 11, onde são propostas as etapas pelas quais os dados coletados precisam passar até que se transformem em decisões e, conseqüentemente, resultem em valor. Esse trabalho propõe uma solução para tomada de decisão de manutenção baseada em dados de monitoramento, portanto, o desenvolvimento foi realizado de forma a ser compatível com um contexto mais amplo de análise de dados da organização.

Figura 11: "A cadeia de valor do dado".



Fonte: (Miller & Mork, 2013, tradução nossa).

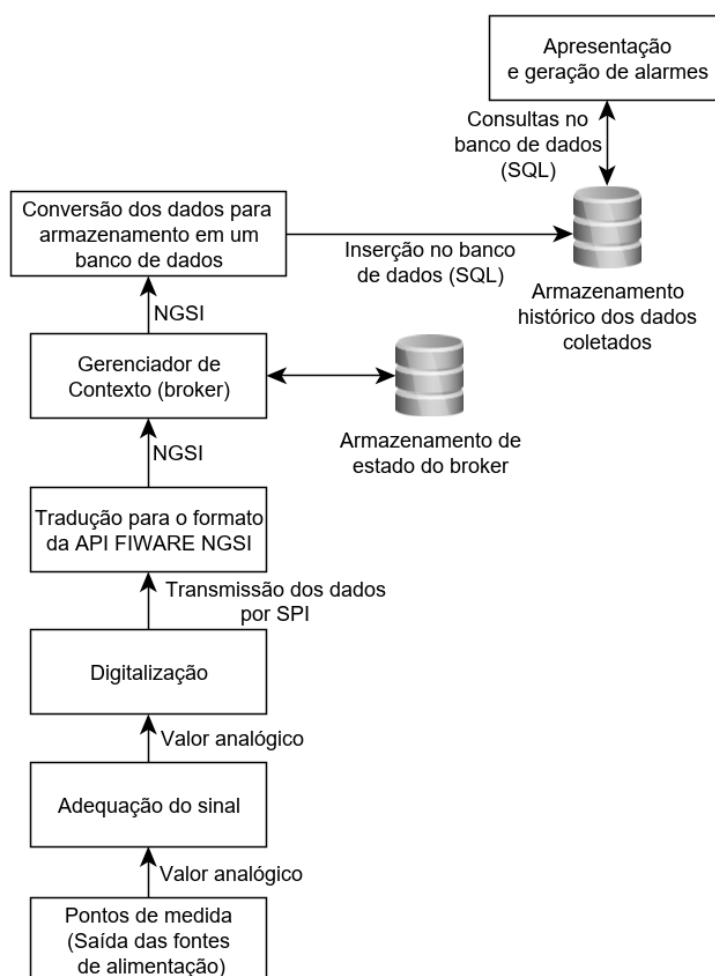
O desenvolvimento apresenta resultados, de acordo com o modelo de referência, nas "Camada de Dispositivo", "Camada de Suporte ao Serviço e à Aplicação" e "Camada de Aplicação".

Para a "Camada de Rede", foi utilizada a infraestrutura já existente, baseada em no padrão *Ethernet*. Portanto, essa camada não será abordada no trabalho.

A arquitetura proposta para essa solução foi construída considerando particularidades da aplicação, como alimentação elétrica contínua e uma infraestrutura de rede de dados segura de alta velocidade para comunicação entre os dispositivos.

O diagrama de blocos da Figura 12 representa a arquitetura planejada, baseada na plataforma FIWARE. Sua construção, em torno do *broker*, contempla: a área de armazenamento das variáveis de contexto; blocos responsáveis por traduzir a informação proveniente dos pontos monitorados para a API FIWARE NGSI e a partir desse formato para armazenamento em um banco de dados persistente; e exibição dos valores armazenados nesse banco de dados. Os blocos serão detalhados nos subitens deste capítulo.

Figura 12: Arquitetura planejada para o monitoramento de fontes.



Fonte: Próprio autor.

Em paralelo com o modelo de referência da ITU-T, a adequação e digitalização do sinal analógico das fontes e a tradução para a API NGSi estão na "Camada de Dispositivo". O *broker*, o armazenamento de estado do *broker*, o serviço para conversão dos dados em dados estruturados e o armazenamento histórico ficam na "Camada de Suporte à Aplicação e ao Serviço". Já o módulo de apresentação e a geração de alarmes ficam na "Camada de Aplicação".

Os subitens abaixo representam os diagramas da figura em uma ordem lógica para melhor compreensão.

5.1 Digitalização

Para que seja possível ler a saída das fontes de alimentação é necessário codificar a tensão e processá-la em sistemas digitais. O sinal da fonte é uma grandeza com intensidade variável e os computadores e sistemas digitais operam apenas com sinais binários.

Para obter um valor binário a partir de um valor analógico é feita uma codificação da intensidade das grandezas de forma que possam ser representadas em zeros e uns. Esse conceito se aplica em todas as nossas interações com o mundo digital, seja na intensidade das cores de cada pixel de um monitor ou na captação de sons com um microfone para gerar um arquivo de áudio no computador.

Um tipo comum de codificação é a transformação proporcional da intensidade do sinal no tempo em um número inteiro dentro de uma faixa, que depende da quantidade de *bits*, ou seja, de zeros e uns, que usaremos para codificar cada valor em cada instante de tempo.

Por exemplo, se utilizado um conversor de 8 bits, tem-se um intervalo de números inteiros entre 0 e 255 (11111111 em notação decimal). Será considerada uma grandeza analógica que pode assumir valores entre 0 e 10, com todos os valores reais possíveis. Em uma conversão direta a grandeza analógica 0,00 será representada pelo código 00000000, a grandeza 10,00 pelo código 11111111, correspondente ao

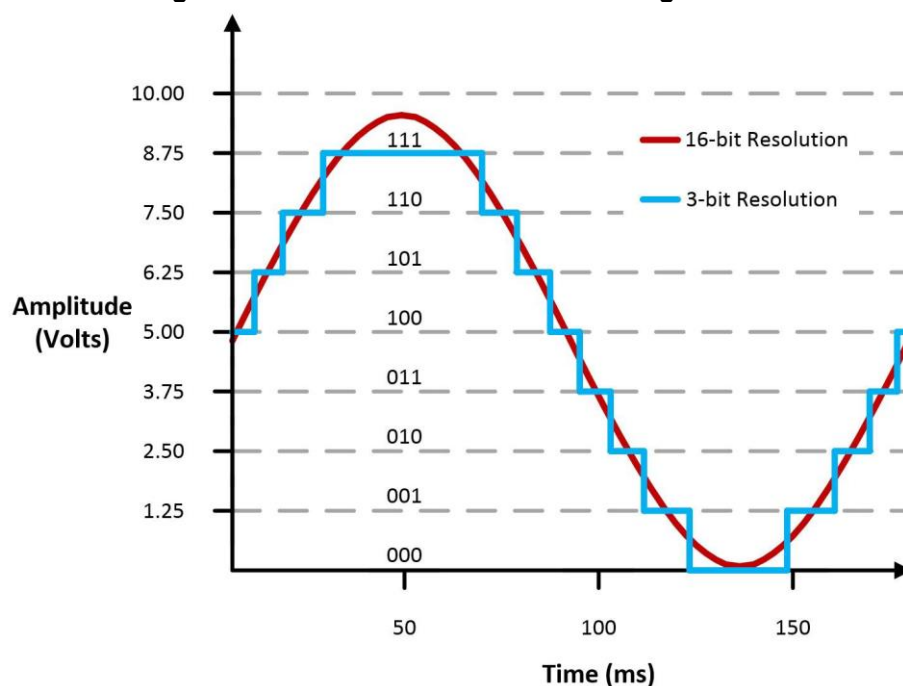
número decimal 255. As variações entre elas serão representadas de forma proporcional, 2,51 pode ser representado por 01000000, por exemplo.

Com essa transformação, a menor variação dessa grandeza que pode ser representada no código é correspondente a variação do *bit* menos significativo. No exemplo, uma variação de um bit corresponde a variação de 0,039215 do valor analógico convertido. A quantidade de *bits* que um conversor pode usar para representar um valor analógico é chamada de resolução.

Outro parâmetro importante na conversão de sinais analógicos em digitais é a taxa de conversão, medido em número de amostras por segundo. A codificação das grandezas analógicas demora um intervalo de tempo devido a características da tecnologia de conversão e dos componentes utilizados. Supondo que um conversor leve 100 ms para concluir o processo e apresentar um código correspondente ao valor analógico, somente teremos outra amostra do sinal analógico no instante 0,1 s, portanto, todas as variações de intensidade que ocorrerem entre 0 s e 0,1 s são perdidas na codificação.

Para ilustrar, é apresentada na Figura 13 a forma de onda de um sinal convertido com resoluções diferentes, pode-se observar que um número menor de *bits* traz menos detalhes do valor analógico.

Figura 13: Forma de onda de um sinal digitalizado.



Fonte: NATIONAL INSTRUMENTS CORP. (2019).

O dispositivo responsável pela a digitalização do sinal analógico é o conversor AD (*analog-to-digital*). As características que diferem entre os conversores são: resolução, taxa de amostragem, número de canais de leitura, circuitos auxiliares necessários, tensão de alimentação, formatos de saída dos valores digitais, entre outras.

Para atender os requisitos e manter o menor custo para o projeto será utilizado para digitalização do sinal um circuito integrado programável que possui integrado em um mesmo encapsulamento um ou mais conversores AD, chamado microcontrolador. O microcontrolador STMF103C8T6 fabricado pela STMicroelectronics foi aplicado nessa solução (STMICROELECTRONICS, 2015).

Porém, para compatibilizar o sinal das fontes às características elétricas da entrada do microcontrolador será necessário adequar o sinal.

5.2 Adequação do Sinal

Com intuito de atender os requisitos de coleta tensão de *ripple* e de coleta de tensão média das fontes, ambas com isolamento conforme requisitado, é necessário o desenvolvimento de um circuito eletrônico para realizar a interface entre o ponto leitura conectado na fonte e a entrada do conversor AD.

Dentre as estratégias possíveis para se obter esses valores com uma resolução que permita a análise, optou-se por decompor o sinal da saída das fontes em dois: o primeiro sinal terá a informação da tensão média, chamada também de tensão de corrente contínua (VCC); e o segundo sinal a variação de tensão em torno do valor médio, incluindo a tensão de *ripple*, chamada de tensão de corrente alternada (VCA).

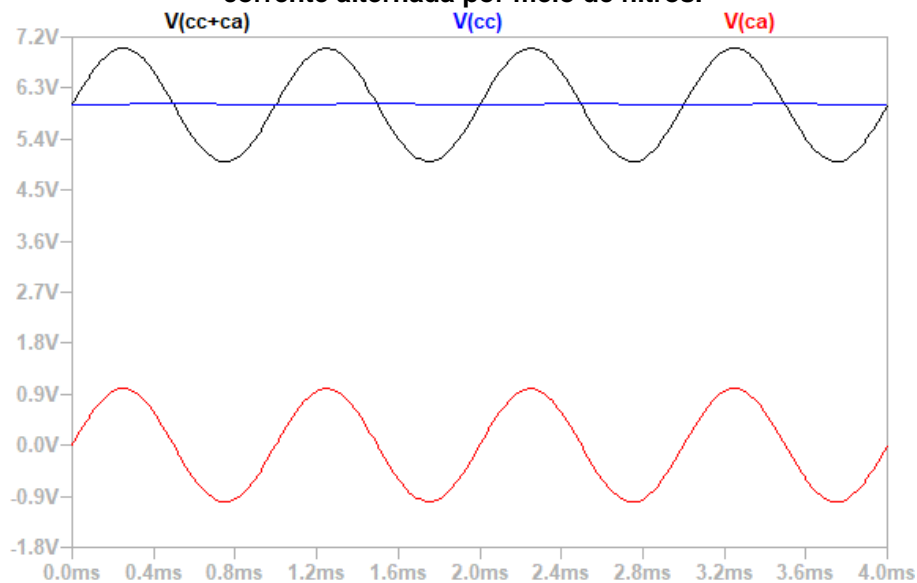
Uma fonte de alimentação de corrente contínua ideal possui tensão de saída constante, estável e próxima ao valor de saída nominal da fonte. É desejável que qualquer outro sinal somado à tensão de saída seja nulo, portanto, espera-se que a tensão de corrente alternada somada à saída da fonte possua valores de tensão muito pequenos, na ordem de mV.

A separação dessas componentes é realizada com a utilização de filtros. Filtros são circuitos projetados para manipular os sinais e destacar ou atenuar as características desejadas. Oppenheim, Willsky, & Nawab (2010) definem a filtragem em seu livro sobre sinais e sistemas:

Em diversas aplicações, é interessante mudar as amplitudes relativas dos componentes em frequência de um sinal ou talvez eliminar por completo alguns componentes em frequência, tal processo é conhecido como filtragem. Os sistemas lineares invariantes no tempo, que mudam a forma do espectro, são conhecidos como filtros conformadores de frequência. Os sistemas que são projetados para deixar passar algumas frequências essencialmente não distorcidas e que atenuam significativamente ou eliminam outras são chamados filtros seletivos em frequência.

O gráfico da Figura 14 apresenta o resultado de uma simulação da utilização de filtros para separar as tensões que compõem um sinal ilustrativo.

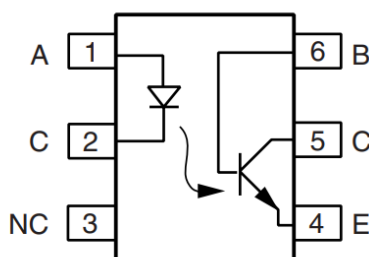
Figura 14: Sinal completo e sinais separados em tensão de corrente contínua e tensão de corrente alternada por meio de filtros.



Fonte: Próprio autor.

De forma a atender ao requisito de isolamento entre entradas, optou-se pela utilização de optoacopladores entre a saída das fontes que serão monitoradas e a entrada do conversor. O optoacoplador é um componente que utiliza o sinal de entrada para acionar um LED e, por luz, excitar um fototransistor na saída, de forma que não há contato elétrico entre entrada e saída. A representação do componente está reproduzida na Figura 15.

Figura 15: Representação do optoacoplador 4N25.



Fonte: VISHAY (2019).

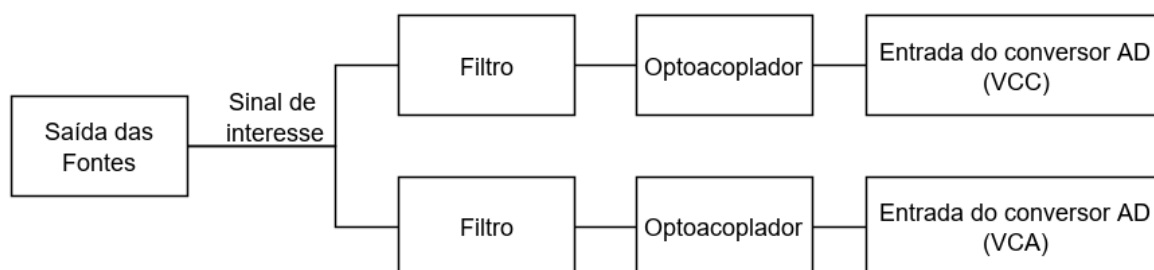
A utilização desse componente em conjunto com um circuito que adapte o sinal de saída da fonte à faixa de entrada do conversor AD possibilita a extração da informação das fontes de forma isolada. Porém, uma característica do optoacoplador é a não-

linearidade em toda a faixa de operação, ou seja, o sinal medido na saída não será diretamente proporcional ao da entrada.

Essa não-linearidade causa uma dificuldade na transformação do sinal recebido pelo microcontrolador em um valor digital que represente corretamente o estado da fonte. É necessário considerar essa característica no desenvolvimento tanto no circuito do filtro quanto no circuito para conexão do optoacoplador ao microcontrolador.

De forma abrangente, a adequação do sinal deverá seguir, portanto, um diagrama conforme apresentado na Figura 16, caracterizado pela segregação da tensão monitorada entre os sinais relativos ao sinal de corrente contínua e ao sinal de corrente alternada, com isolamento elétrico entre o ponto monitorado e a entrada do conversor AD.

Figura 16: Diagrama de blocos do circuito de adequação do sinal.



Fonte: Próprio autor.

5.3 Tradução SPI para NGSi

Com a utilização de um microcontrolador para digitalização dos valores lidos, podem ser adotadas diversas estratégias de comunicação entre o microcontrolador e o dispositivo que fará a tradução para o formato especificado na plataforma FIWARE, o

NGSI. O microcontrolador escolhido para esse projeto possui as seguintes interfaces para comunicação: UART⁶, I²C⁷, SPI⁸, CAN⁹ e USB¹⁰.

Nesse trabalho adotou-se o SPI, por possuir alta taxa de transmissão e possibilitar comunicação com múltiplos dispositivos, permitindo aumentar a quantidade de pontos de coleta com a utilização de um mesmo dispositivo de tradução.

Essa comunicação funciona com quatro fios mais uma referência, são eles: SCK, MOSI, MISO, CE, e o terra como referência. O SCK é o *clock* da comunicação, esse sinal é gerado pelo dispositivo *master* para sincronismo; o MOSI é o sinal que o dispositivo *master* envia os dados aos *slaves*; o MISO é o sinal enviado pelos *slaves* ao *Master*; o CE é o sinal de seleção que permite a utilização de vários *slaves* conectados em um único *master*.

No diagrama apresentado na Figura 17, está ilustrada a conexão de um dispositivo *Master* SPI com três *Slaves*, e os sinais necessários para ocorrer a comunicação. Nesse trabalho, o *Master* será o dispositivo de tradução e o *Slave* o microcontrolador responsável pela digitalização.

6 UART é a comunicação serial assíncrona, que utiliza um canal para transmissão e um para recepção e não utiliza um sinal que sincronize o início e o fim das transmissões entre os dispositivos participantes da comunicação, é de simples utilização, porém indicado para baixas velocidades de transmissão e pode implicar em perda de dados (SEEDSTUDIO, 2020).

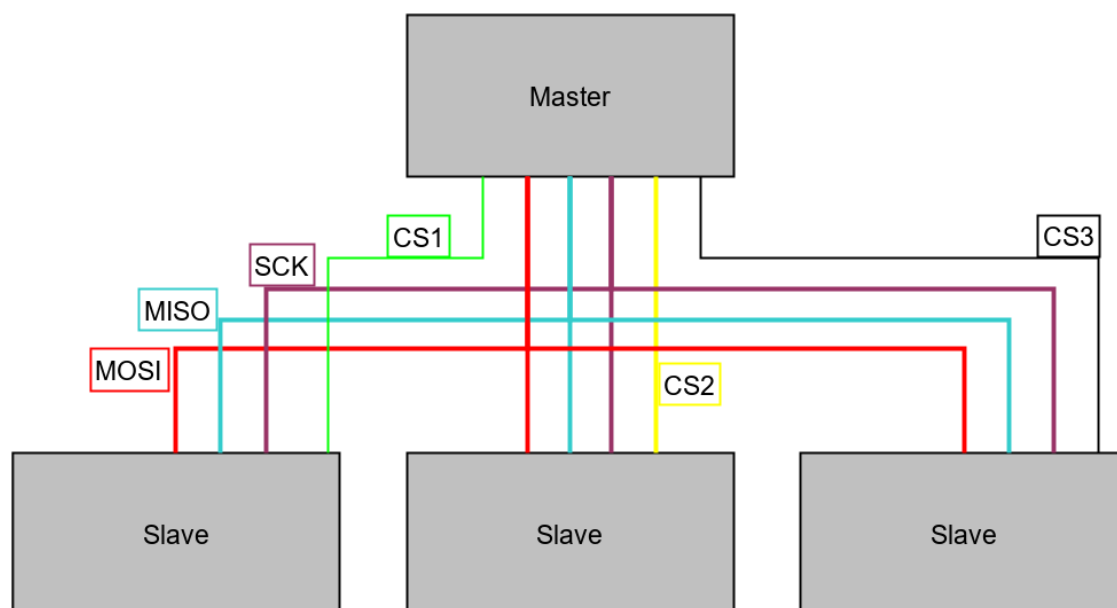
7 I²C é um protocolo serial desenvolvido pela Philips, utilizado para conectar periféricos com simplicidade de conexão e que utiliza endereçamento lógico. É compatível com multi-mestres e multi-escravos, porém as velocidades de transmissão são menores que do SPI (SEEDSTUDIO, 2020).

8 SPI é um protocolo serial para comunicação síncrona que funciona com três (unidirecional ou não-simultânea) ou quatro fios (bidirecional simultânea). Utilizada para curtas distâncias em sistemas embarcados pode atingir dezenas de megabits por segundo de velocidade de transmissão e opera com um *master* e n *slaves* endereçados por *hardware* (SEEDSTUDIO, 2020).

9 As redes CAN possuem verificação de erros, controle de priorização de mensagens e também podem ser implantadas com poucos fios de conexão entre os dispositivos, com endereçamento lógico (NATIONAL INSTRUMENTS, 2020).

10 USB é um padrão para comunicação serial, geralmente ponto a ponto, funciona com quatro fios, exige cuidados em hardware e software para o correto funcionamento (STMICROELECTRONICS, 2015).

Figura 17: Diagrama de conexões da comunicação SPI de um *Master* com 3 *Slaves*.



Fonte: Próprio autor.

O dispositivo *Master*, exercendo a função de interface entre a rede no padrão Ethernet e os dispositivos físicos é chamado de *gateway* IoT. No desenvolvimento desse projeto utilizou-se o *single-board computer* Raspberry Pi para esse módulo.

A utilização de um *gateway* possibilita, além da tradução para o padrão NGSI, embutir tratamento de dados na camada de dispositivo, o que desonera o processamento central, ou em nuvem, dos dados coletados.

Essa funcionalidade reduz a latência e permite análise relativa ao contexto no qual o *gateway* está instalado em ciclos de tempo menores do que utilizando arquitetura onde os dados são transmitidos diretamente para um banco de dados central. Bonomi, Milito, Zhu, & Addepalli (2012) apresentam essas características como o conceito de *fog computing* e afirmam: “Muitas aplicações requerem tanto a localização da *fog*, quanto a centralização da *Cloud*, particularmente para *analytics* e *Big Data*” (tradução nossa).

Nesse contexto, o módulo utilizado para tradução dos dados recebidos por SPI do microcontrolador para o formato NGSI permite que sejam realizadas centenas de leituras de tensão por segundo, calcular valores que resumam o comportamento

observado com essas leituras e enviá-los de maneira consolidada aos outros componentes da arquitetura proposta.

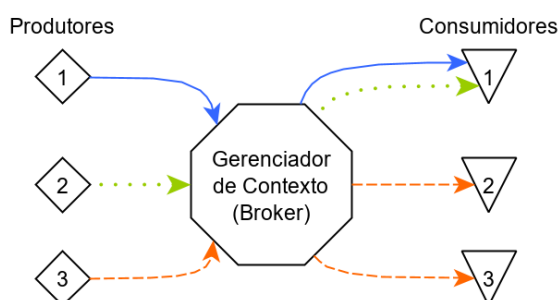
5.4 Gerenciador de contexto (*Broker*)

Como mencionado anteriormente, o Gerenciador de Contexto é um módulo obrigatório em uma solução construída sobre a plataforma FIWARE. Esse módulo centraliza o fluxo de dados e utiliza o paradigma *publish-subscribe* para organizar a comunicação entre os módulos que geram os dados e os módulos que os acessam.

No exemplo apresentado na Figura 18, existem três elementos que geram e três elementos que utilizam os dados. O consumidor 1 utiliza os dados dos produtores 1 e 2. Os consumidores 2 e 3 utilizam somente os dados gerados pelo produtor 3.

Para que isso ocorra, é preciso registrar no *broker* cada fluxo de dados a ser enviado do *broker* para os consumidores. Com esse registro, também chamado de subscrição, o *broker* notificará o consumidor assim que receber uma atualização do dado de interesse gerado pelo produtor.

Figura 18: Exemplo do paradigma *publish-subscribe*.



Fonte: Próprio autor.

O *componente* utilizado para atuar como gerenciador de contexto nesse projeto foi o Orion, indicado na documentação da FIWARE FOUNDATION (2020).

Para funcionamento do Orion é obrigatória a instalação do MongoDB. Esse sistema de gerenciamento de banco de dados é responsável por armazenar o estado das variáveis que estão sendo publicadas e lidas.

5.5 Armazenamento histórico

Para armazenamento dos dados históricos coletados das fontes foi utilizado o componente QuantumLeap em conjunto com o banco de dados CrateDB. O QuantumLeap é um serviço que recebe dados temporais no formato NGSI v2, os converte em dados semiestruturados e os armazena em um banco de dados temporal, o CrateDB (MARTEL INNOVATE, 2020).

Para que o Orion envie ao QuantumLeap as atualizações recebidas pelas fontes, é necessário executar uma subscrição. Essa subscrição é realizada pelo próprio QuantumLeap com a seguinte requisição:

```
curl -X POST \  
  
'http://172.17.0.2:8668/v2/subscribe?orionUrl=http://172.17.0.1:1026/v2&quantumleapUrl=http://172.17.0.2:8668/v2&entityType=PSmonitor'
```

Na requisição apresentada, o QuantumLeap está executando no endereço 172.17.0.2, na porta 8668 e o Orion no endereço 172.17.0.1, na porta 1026. O valor “PSmonitor” enviado ao parâmetro *entityType* restringe para que apenas as mensagens com o tipo “PSmonitor” sejam enviadas pelo Orion ao QuantumLeap. Isso garante que a mesma instância do Orion possa ser utilizada para outras aplicações de outros tipos com outros componentes associados.

Dessa forma os dados identificados com o tipo “PSmonitor” são gravados no banco de dados CrateDB conectado ao QuantumLeap, essa conexão deve ser feita na instalação.

5.6 Apresentação

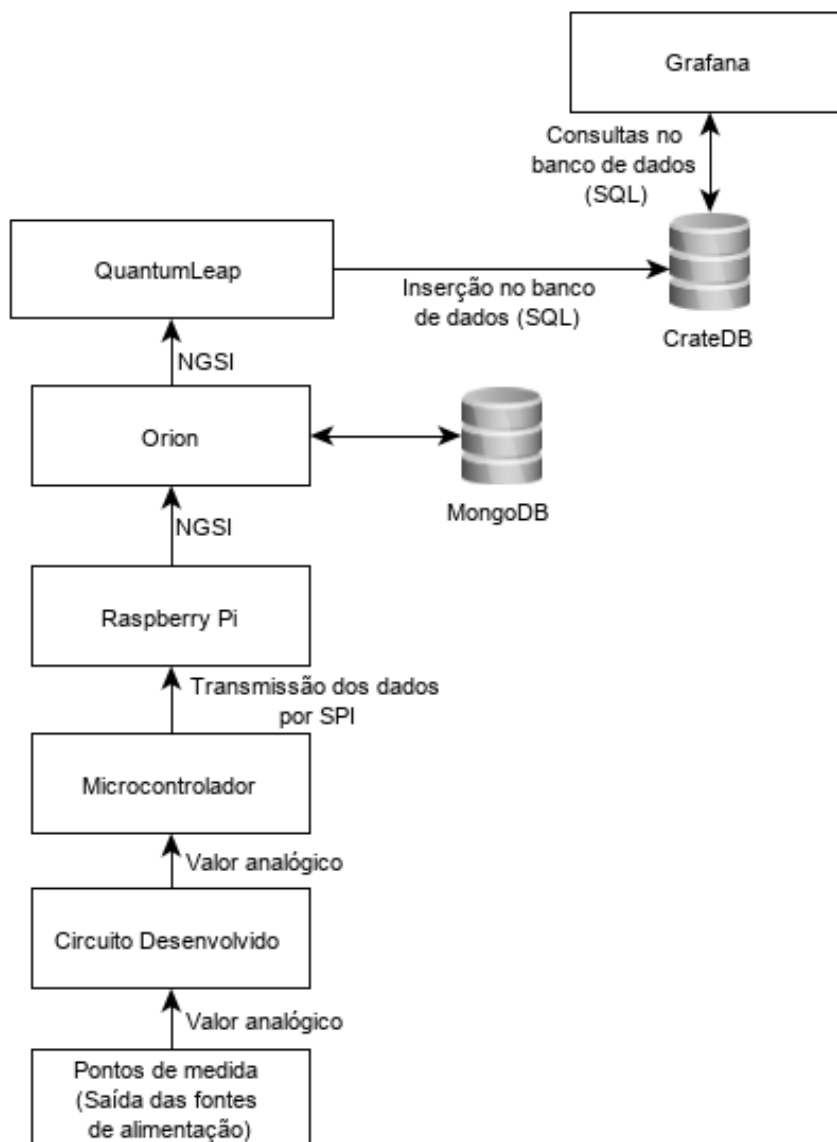
A apresentação dos dados gravados foi construída no software Grafana, que possibilita a criação facilitada de *dashboards* com a concentração de dados a partir de diversas fontes.

As consultas que o Grafana executa no CrateDB utilizam a linguagem de manipulação de banco de dados SQL, comum entre diversos sistemas gerenciadores de banco de dados.

Com as facilidades presentes na ferramenta, podem ser criados gráficos e mostradores com indicativos de cores que permitem a visualização do estado das fontes em tempo real, além de consulta para análise dos dados históricos.

Finalmente, a arquitetura dos módulos implementados está apresentada na Figura 19, onde se apresentam os elementos que serão utilizados para implementação da arquitetura planejada apresentada na Figura 12. O bloco de adequação do sinal é representado pelo circuito desenvolvido; a digitalização pelo microcontrolador programado para executar essa tarefa; a tradução para a API FIWARE NGSI realizada por software desenvolvido em Python executando na Raspberry Pi; o Orion como gerenciador de contexto, associado com o MongoDB para armazenamento das variáveis de estado; o QuantumLeap realizando a interface entre o Orion e o banco de dados persistente, no caso o CrateDB; e o Grafana para apresentação dos dados.

Figura 19: Arquitetura dos *Generic Enablers*, módulos e serviços desenvolvida nesse trabalho.



Fonte: Próprio autor.

5.7 Considerações do Capítulo

Este capítulo apresentou o projeto de uma solução que permite prover dados para análise, com foco em coleta e registro e possibilidade de escalabilidade e integração com outras fontes de dados.

6 EXPERIMENTAÇÃO

Com base no projeto sobre cada etapa necessária para obtenção de uma coleta de dados que viabilize a manutenção preditiva de fontes, um protótipo foi construído e testado em bancada, contemplando todos os módulos apresentados, da adequação do sinal analógico à exibição dos valores.

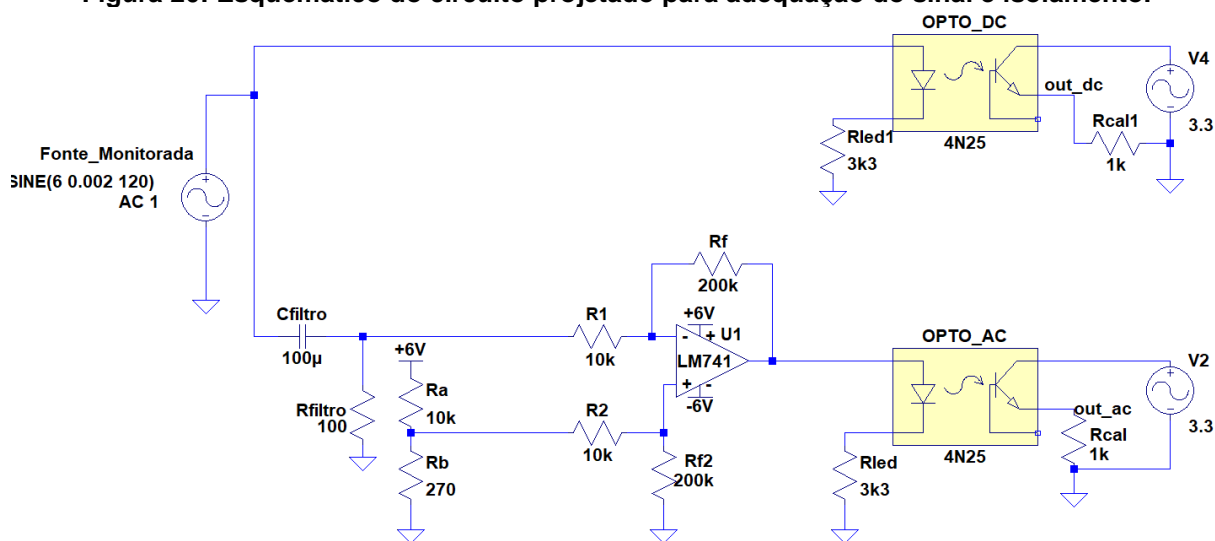
Este capítulo apresenta uma síntese da construção desses módulos e informações complementares estão disponíveis nos apêndices.

6.1 Módulo de adequação do sinal e módulo de digitalização

O circuito de adequação do sinal apresentado no item 5.2 foi projetado, simulado e montado em uma placa de prototipagem. Maiores detalhes do projeto estão apresentados no Apêndice B, e o esquemático do circuito resultante está reproduzido na Figura 20.

Para aplicação em uma fonte do sistema de sinalização do Metrô, que é simétrica, é necessário replicar o circuito para o sinal positivo e para o sinal negativo de cada fonte.

Figura 20: Esquemático do circuito projetado para adequação do sinal e isolamento.



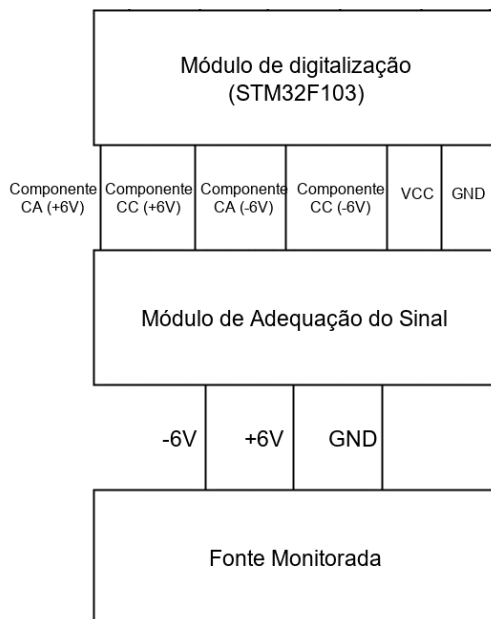
Fonte: Próprio autor.

Dessa forma, são necessárias quatro entradas do microcontrolador para digitalização dos sinais provenientes de cada fonte, sendo eles: sinal positivo, *ripple* do sinal positivo, sinal negativo e *ripple* do sinal negativo, sempre em relação a uma referência comum.

Para a comunicação entre o microcontrolador e o *gateway*, foi desenvolvido um protocolo de troca de mensagens de forma que seja possível o envio dos dados desses quatro canais.

A montagem em laboratório para monitoramento de uma fonte com duas saídas (+6 V e -6 V) foi realizada conforme o diagrama apresentado na Figura 21.

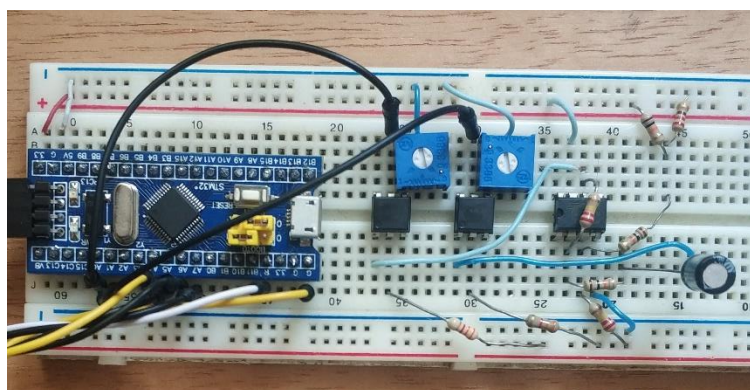
Figura 21: Diagrama de conexões entre a fonte monitorada e o Microcontrolador com função de conversão AD.



Fonte: Próprio autor.

Para utilização desse microcontrolador foi utilizada a placa de desenvolvimento *Blue Pill* desenvolvida pelo projeto STM32-base (2019). Uma montagem foi realizada em *protoboard* para verificação prática dos resultados obtidos nas simulações, essa montagem está apresentada na Figura 22.

Figura 22: Montagem do circuito de adequação do sinal em *protoboard*.

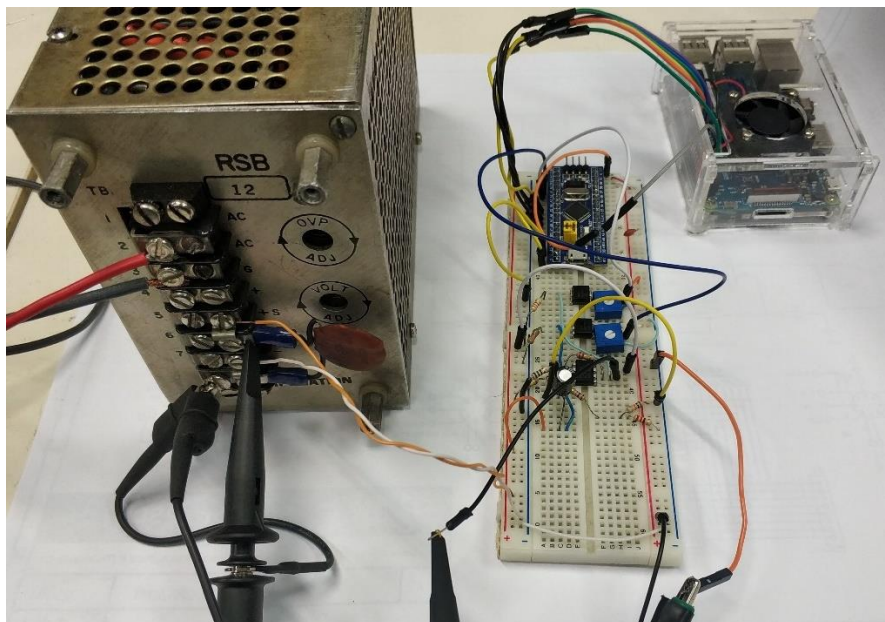


Fonte: Próprio autor.

Para validação desse módulo, foram medidos, utilizando um osciloscópio, os valores de tensão contínua e alternada (*ripple*) do sinal, diretamente nas saídas fonte, conforme montagem apresentada na Figura 23. Em seguida, os valores foram

comparados com os sinais obtidos após o circuito de adequação, que realiza amplificação da tensão alternada e acoplamento ótico entre a entrada e a saída.

Figura 23: Montagem realizada em bancada para teste do módulo de adequação.



Fonte: Próprio autor.

Por último, após a calibração do resistor de saída do optacoplador e da equação de conversão do *software* com uma fonte de precisão de 6 V, os valores lidos com os instrumentos foram comparados com os valores digitalizados e os resultados apresentados na Tabela 2.

Tabela 2: Leitura de uma fonte de 6V para validação do módulo de adequação e digitalização.

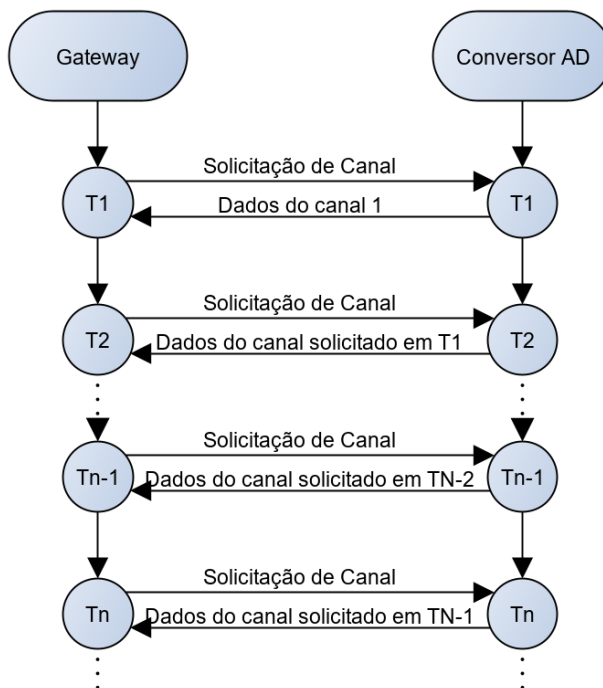
Sinal	Saída da Fonte	Saída do Circuito de Adequação	Valor Digitalizado (após calibração)
Valor médio	6,02 V	1,20 V	6,10 V
Valor RMS do <i>ripple</i>	3 mV	25 mV	4 mV

Fonte: Próprio autor.

Para o envio dos dados digitalizados pelo canal SPI *full-duplex*, foi desenvolvido um protocolo de comunicação entre a Raspberry e o microcontrolador para possibilitar seleção do canal de leitura. A troca de mensagens foi organizada conforme a Figura

24, sendo que T1 representa a primeira troca de mensagens, T2 a segunda e Tn a enésima.

Figura 24: Troca de mensagens sobre o canal SPI.



Fonte: Próprio autor.

Foram utilizados 4 bytes por comunicação por SPI, sendo que as mensagens de solicitação de leitura com o formato no diagrama da Figura 25.

Figura 25: Mensagem de solicitação enviada pelo gateway IoT ao conversor AD.

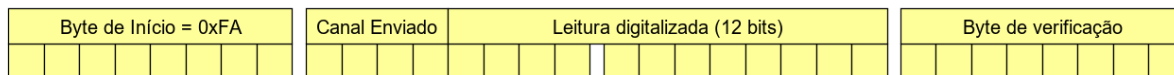


Fonte: Próprio autor.

Por ser uma comunicação *full-duplex*, mesmo que não seja necessário, é obrigatório que sejam enviados na solicitação de leitura o mesmo número de bytes que se deseja ler do dispositivo *slave*. Para obter a precisão necessária, os dados digitalizados são enviados com 12 bits, por isso, 12 bits da mensagem de solicitação são sempre preenchidos com zeros.

As mensagens com os dados lidos foram montadas conforme diagrama da Figura 26.

Figura 26: Mensagem de resposta enviada pelo conversor AD ao *gateway* IoT.

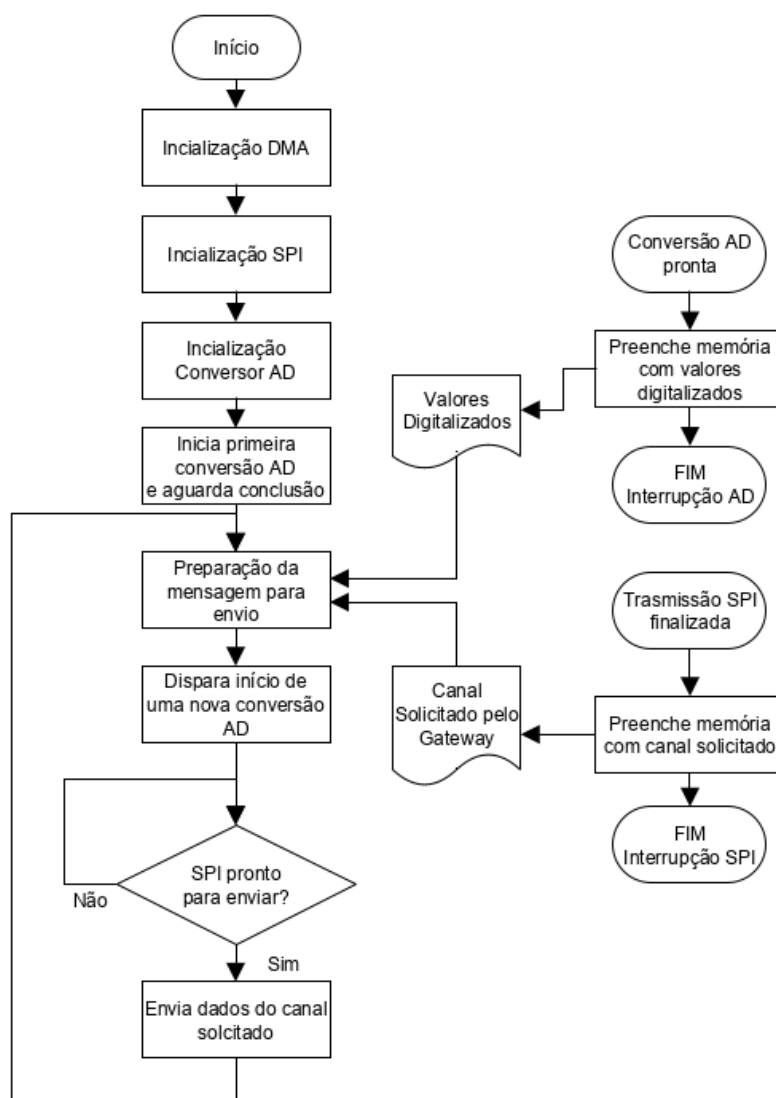


Fonte: Próprio autor.

Um byte de início foi definido para sincronização da comunicação SPI, de forma a prover uma verificação simples de que emissor e receptor estão enviando as mensagens com a mesma sequência de quadros. O byte de verificação utilizado em ambas mensagens é formado por uma operação lógica XOR entre os outros 3 bytes, de forma que a mensagem seja descartada caso ocorra uma inversão de bits na comunicação.

O software desenvolvido para o microcontrolador STM32F103 está representado no fluxograma da Figura 27.

Figura 27: Fluxograma do software embarcado no microcontrolador para conversão AD e envio das informações digitalizadas ao gateway.



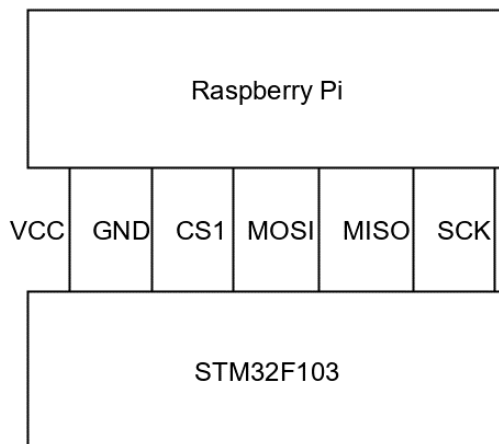
Fonte: Próprio autor.

6.2 Gateway

Conforme análise realizada no item 5.3, optou-se pela utilização de um *single-board computer* como *gateway* IoT, no caso a Raspberry Pi. Esse módulo realiza a interface entre o conversor analógico-digital e a camada de rede *Ethernet* que transporta os dados coletados pelos sensores até o ambiente de processamento, armazenamento e apresentação dos dados coletados.

Foram feitas as conexões ilustradas na Figura 28 para conectar, através do protocolo SPI, a Raspberry Pi com o microcontrolador STM32F103.

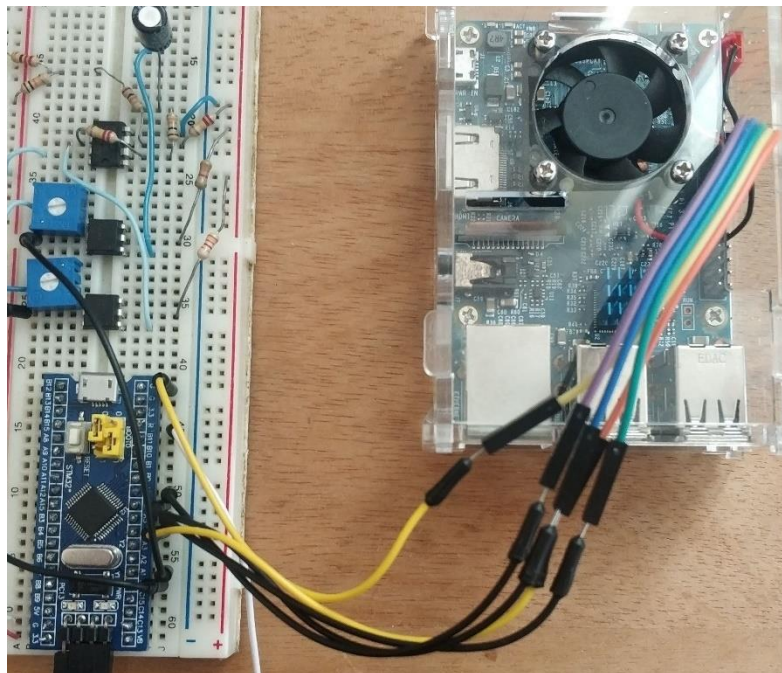
Figura 28: Conexões entre o gateway (Raspberry Pi) e o conversor AD (microcontrolador STM32F103).



Fonte: Próprio autor.

Seguindo o diagrama apresentado, a conexão apresentada na Figura 29 foi realizada.

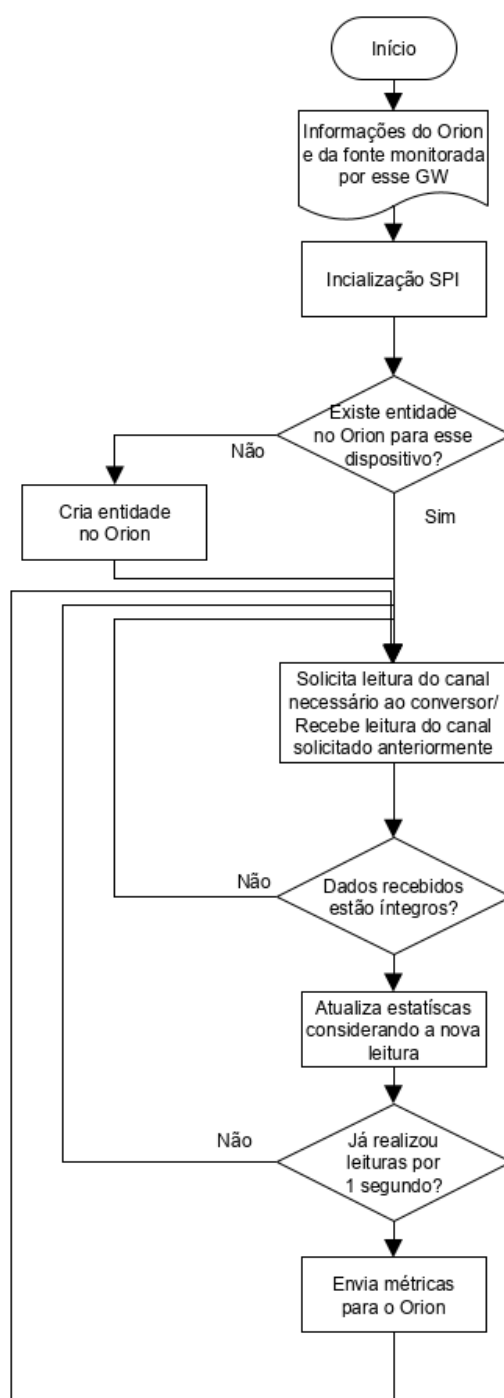
Figura 29: Detalhe da conexão SPI entre a Raspberry Pi e o microcontrolador STM32F103.



Fonte: Próprio autor.

Para execução da função de *gateway*, foi desenvolvido um software na linguagem Python para coletar o valor dos canais SPI e criar e enviar uma requisição HTTP para o concentrador IoT, no caso, o Orion. O fluxograma desse software está apresentado na Figura 30.

Figura 30: Fluxograma do programa sendo executado no *gateway* IoT para solicitação dos dados ao conversor AD e envio ao Orion.



Para o controle do periférico SPI presente na Raspberry Pi e abstração dos comandos para configuração de velocidade, modo de operação, e para transferência de dados, foi utilizada a biblioteca *python-spi*, desenvolvida por Tom Stokes (STOKES, 2015).

A requisição HTTP necessária para criar e atualizar o valor de um atributo de um objeto criada no Orion é bem definida na documentação e foi utilizada nesse mesmo software para o envio do dado ao concentrador. A biblioteca “*json*”, para manipulação de dados no formato JSON (PYTHON SOFTWARE FOUNDATION, 2020), e a biblioteca “*requests*” (REITZ, 2016), para facilitar o envio de requisições HTTP, foram utilizadas.

O trecho de código para o envio das informações ao Orion está reproduzido abaixo. São utilizados alguns parâmetros predefinidos pela plataforma FIWARE, como *id* e *type*, e outros específicos para a aplicação, como *rms_ripple*, *mean*, *max* e *min*.

```

payload = {
    "id": id,
    "type": type,
    "rms_ripple": {
        "value": rms_ripple,
        "type": "Float"
    },
    "mean": {
        "value": media_ripple,
        "type": "Float"
    },
    "max": {
        "value": valor_max,
        "type": "Float"
    },
    "min": {
        "value": valor_min,
        "type": "Float"
    }
}
r = requests.post('http://'+IP_Orion+:1026/v2/entities', data = json.dumps(payload),
headers = headers)

```

O código completo do software desenvolvido está apresentado no Apêndice C.

6.3 Concentrador IoT

A utilização dos componentes, chamados de *generic enablers*, compatíveis com o *framework* FIWARE traz uma infraestrutura organizada e modular ao projeto, permitindo expansão tanto em quantidade de pontos coletados, quanto em tipos de variáveis lidas sem que seja necessário modificar os serviços de processamento, armazenamento e exibição.

Por ser uma arquitetura baseada em serviços web, a migração desses módulos é possível, seja entre *hardware* local ou para um ambiente de nuvem, e caracteriza a versatilidade desse tipo de ambiente.

Um exemplo dessa versatilidade foi observado com a implantação do serviço do QuantumLeap, que foi realizada utilizando-se um *container* disponibilizado em SmartSDK (2019). Esse serviço, conectado ao banco de dados CrateDB, instalado e rodando no *host* com endereço IP 192.168.1.20, pôde ser instalado e iniciado com o seguinte comando.

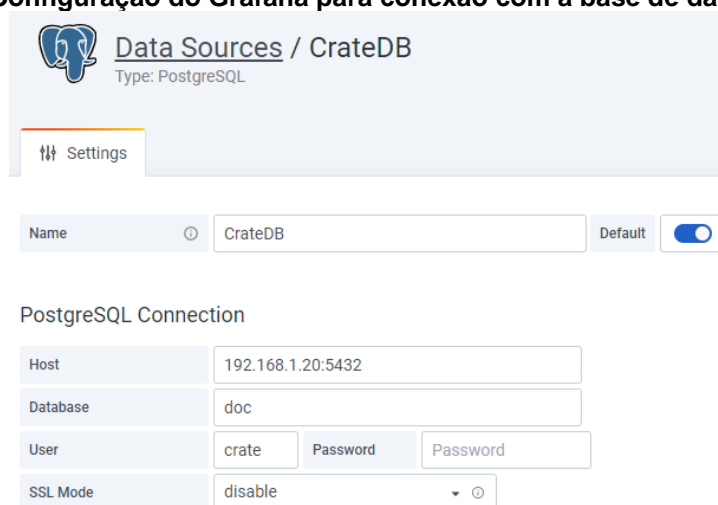
```
docker run -d -p 8668:8668 -e "CRATE_HOST=http://192.168.1.20"
smartsdk/quantumleap
```

Esse serviço abstrai a interação com o banco de dados, de forma que, com a subscrição apresentada no capítulo 5.5, todos os dados do tipo de entidade determinado, são armazenados corretamente.

6.4 Apresentação

A conexão do Grafana ao CrateDB, para visualização dos dados armazenados, foi realizada com comandos compatíveis com o banco de dados PostgreSQL, em um conector existente no CrateDB. A configuração da fonte de dados no Grafana foi feita conforme publicação do *blog* oficial do CrateDB (SLATER, 2019), e está reproduzida na Figura 31.

Figura 31: Configuração do Grafana para conexão com a base de dados CrateDB.



Data Sources / CrateDB
Type: PostgreSQL

Settings

Name: Default:

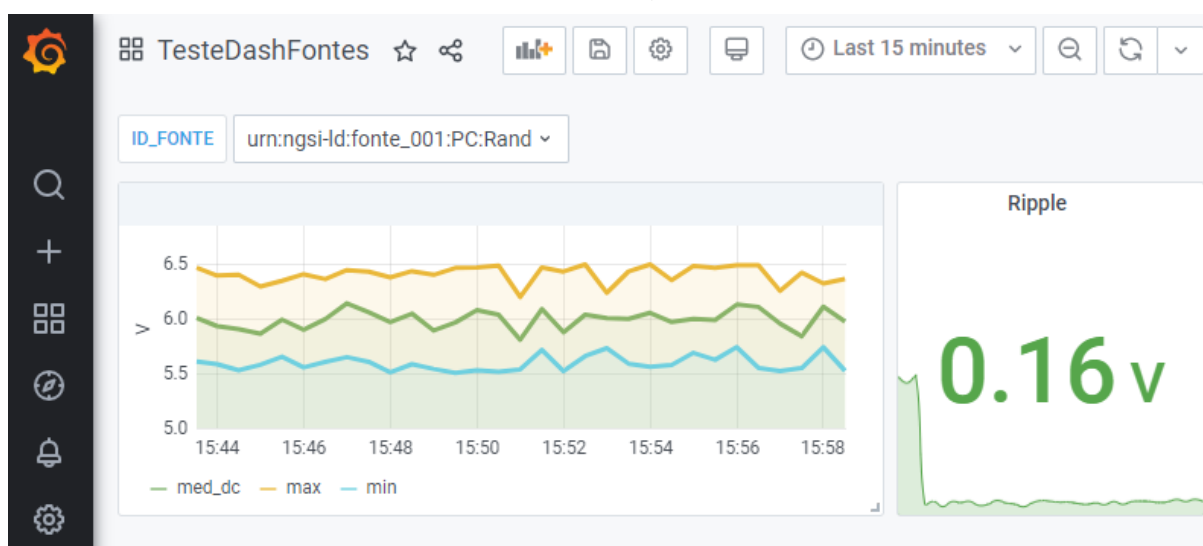
PostgreSQL Connection

Host	<input type="text" value="192.168.1.20:5432"/>		
Database	<input type="text" value="doc"/>		
User	<input type="text" value="crate"/>	Password	<input type="text" value="Password"/>
SSL Mode	<input type="text" value="disable"/>		

Fonte: Próprio autor.

Para ilustração das possibilidades de visualização dos valores lidos foi criado um dashboard com dois objetos no Grafana, chamados *plugins*, conforme apresentado na Figura 32. O *plugin* chamado *Graph* é caracterizado pelo gráfico com o valor da tensão no tempo e utiliza uma requisição SQL para buscar no banco de dados que está sendo alimentado pelo QuantumLeap os dados a serem exibidos.

Figura 32: Dashboard para teste de visualização dos dados coletados de uma fonte.

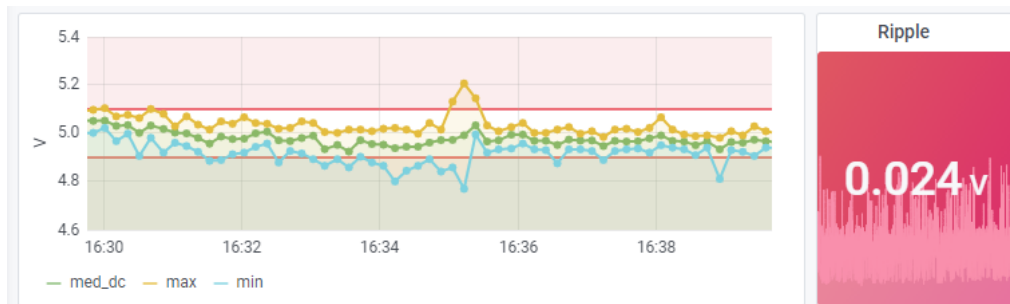


Fonte: Próprio autor.

No Grafana, também é possível criar limites para os valores coletados, de forma que caracterizem de forma visual um problema em um elemento do sistema, através da

alteração da cor do elemento e da apresentação de linhas representando os limites. A Figura 33 apresenta uma demonstração da alteração visual causada por valores lidos fora do intervalo aceitável.

Figura 33: Alteração visual causada por limiares ultrapassados pelos valores lidos.



Fonte: Próprio autor.

O *plugin Graph* utiliza variáveis para desenhar o gráfico apresentado, uma chamada obrigatoriamente *time*, que deve conter o horário da leitura do valor no formato de data “*Unix Timestamp*”, e outras relativas aos valores a serem apresentados no gráfico.

Convenientemente, ao realizar a subscrição do QuantumLeap no Orion, o metadado contendo o horário da leitura é recebido e armazenado na coluna *time_index*. A requisição feita para obter o gráfico da Figura 32 está apresentada a seguir.

```
SELECT
  time_index AS time,
  med_dc,
  max,
  min
FROM etfonte_001
```

```
WHERE entity_id='$ID_FONTE'
ORDER BY 1,2
```

*Comando para buscar as informações.
Horário renomeado para “time”.
Valor médio de saída da fonte.
Valor máximo coletado nessa amostra.
Valor mínimo coletado nessa amostra.
Tabela de origem, no caso de entidades do tipo “fonte_001”.
Seleção de qual fonte será apresentada.
Ordenação padrão do Grafana.*

Nessa requisição, destaca-se o termo *entity_id = '\$ID_FONTE'*, que serve para que, no *dashboard* do Grafana, seja possível selecionar a fonte a ser monitorada. Esse ID deve ser único por fonte monitorada e será o mesmo indicado ao enviar os dados a partir da Raspberry Pi para o Orion.

6.5 Considerações do Capítulo

Com base no desenvolvimento realizado foi possível implantar uma infraestrutura para armazenamento e disponibilização dos dados, versátil para se integrar a diversas ferramentas.

Para se obter um valor digitalizado atendendo aos requisitos de isolamento e precisão, foi projetado e construído um circuito eletrônico baseado em amplificadores operacionais e optoacopladores. Após a digitalização do dado, a transmissão a partir do microcontrolador até *gateway IoT* foi realizada por meio de um protocolo desenvolvido especificamente para essa aplicação que opera sobre comunicação SPI.

Foi apresentada a utilização da Raspberry Pi como *gateway IoT*, as conexões e softwares necessários para essa aplicação, de forma a realizar a interface do microcontrolador com a plataforma FIWARE.

A utilização do componente QuantumLeap, compatível com a plataforma e conectado ao concentrador Orion, possibilitou a automatização do registro dos valores coletados em um banco de dados.

E por fim, para concluir o ciclo da coleta até a apresentação das tensões das fontes de alimentação, uma ferramenta de construção de *dashboards* foi apresentada nesse capítulo, o Grafana. Esse software apresenta uma extensa gama de componentes disponíveis, por isso, um aprofundamento nas funcionalidades da ferramenta pode levar a uma eficiência maior na visualização dos dados.

Na experimentação da solução em bancada, notou-se dificuldade na gestão dos pontos monitorados e dos dispositivos utilizados para monitoramento. Esse aspecto não representou um problema para um teste em bancada, porém certamente deve ser um ponto de atenção em uma aplicação em larga escala.

7 CONSIDERAÇÕES FINAIS

O trabalho apresenta um protótipo de solução de monitoramento de fontes de alimentação que pode conectar, através de uma plataforma escalável e versátil, diversos dispositivos físicos em um ambiente virtual, e permitir melhorias no processo de manutenção e no acompanhamento do ciclo de vida dos ativos monitorados.

No protótipo construído, foram explorados aspectos de diversas camadas de uma solução IoT, abrangendo desde a interface no nível do dispositivo até a exibição em uma interface amigável.

Pôde-se verificar a importância de uma delimitação dos requisitos dos sinais a serem coletados, principalmente quando existem particularidades nas grandezas observadas. Essas características podem elevar a complexidade da conexão de um dispositivo de coleta a um equipamento físico.

O uso de uma plataforma que permite acesso facilitado aos dados históricos e à condição dos ativos permite expansão da solução, seja para aplicação de modelos estatísticos e de inteligência artificial para obtenção de padrões e previsão de comportamento dos equipamentos, ou apenas para acesso por ferramentas de visualização que permita *insights* sobre os equipamentos.

7.1 Limitações

A conexão do ponto de coleta de dados em um sistema crítico ao dispositivo de monitoramento deve ser avaliada em termos das certificações necessárias quanto à segurança, mesmo quando se utiliza isolamento óptico para leitura da grandeza e tomam-se cuidados de projeto para não causar consequências no equipamento monitorado em caso de falha no dispositivo de leitura.

Nos testes em bancada realizados com o protótipo construído, essas medidas de mitigação de riscos foram levadas em conta, mas no caso de uma aplicação prática

de um sistema de monitoramento é necessário análise pelos engenheiros responsáveis pelo sistema de sinalização do projeto do dispositivo a ser conectado.

Outro aspecto que exige análise de engenheiros responsáveis por esse sistema é a definição de limiares para apresentação de alarmes, sendo necessário considerar o contexto de cada equipamento monitorado.

A utilização de *generic enablers* baseados na plataforma FIWARE trouxe as vantagens de uma arquitetura aberta no sentido de versatilidade, porém a administração das entidades cadastradas mostrou-se um desafio e uma limitação na utilização dessa solução, caso fosse aplicada em larga escala. Deve-se considerar o desenvolvimento ou utilização de uma solução que possua ferramentas para facilitar a alteração, manutenção e gestão dos pontos de coleta e do ambiente como um todo.

7.2 Conclusão

Para se demonstrar uma solução para monitoramento de fontes, foi desenvolvido um protótipo com a análise e implantação das Camadas de Dispositivo, de Suporte ao Serviço e à Aplicação e a de Aplicação.

A Camada de Dispositivo, composta por um módulo de adequação do sinal monitorado, um módulo de digitalização e um *gateway* IoT, coleta os dados do campo e os converte em um formato compatível com uma plataforma IoT.

A Camada de Suporte ao Serviço e à Aplicação, construída com instalação e utilização de componentes da plataforma FIWARE, viabilizou a centralização dos dados coletados através de uma arquitetura escalável, com softwares de uso gratuito e passível de conexão com serviços externos. O *software* Grafana foi utilizado para a camada de aplicação, permitindo a visualização de dados históricos e configuração de alarmes com indicativo de cores, além de possibilitar a utilização de painéis de acordo com a preferência do usuário.

No caminho de se obter processos de manutenção compatíveis com a Indústria 4.0, o desenvolvimento apresentado mostrou diversas etapas necessárias para obtenção

de uma coleta de dados, para a partir daí poder expandir para outros pilares da quarta revolução industrial como aprendizado de máquina e Big Data.

7.3 Contribuições do Trabalho

Foi demonstrada a aplicabilidade de uma solução com conceitos de IoT para o monitoramento de um parque de equipamentos com alta criticidade, apresentando uma alternativa frente aos softwares supervisórios e equipamentos especializados empregados nesse tipo de problema, sem que seja necessário substituir os ativos instalados. Além disso o trabalho descreve tecnicamente o passo a passo empregado para contribuir com a construção de soluções análogas.

7.4 Trabalhos Futuros

Devido à característica da aplicação analisada de possuir uma infraestrutura de rede própria e independente da Internet, um aspecto que não foi aprofundado nesse trabalho foi a segurança na troca de mensagens e no acesso aos serviços. Porém, a realização de uma análise de segurança e aplicação de medidas para implantar os elementos de segurança em todas as camadas da arquitetura são necessárias à aplicação do desenvolvimento realizado aqui.

Os testes foram realizados exclusivamente em fontes lineares retificadas e a aplicação em outros tipos de fontes e demais sinais de tensão aumenta a abrangência do escopo e, conseqüentemente, o número de dispositivos compatíveis com essa solução.

A análise do retorno sobre o investimento realizado com a aplicação em escala da solução desenvolvida é, também, um possível desdobramento desse desenvolvimento.

Por fim, uma última vertente proposta é a aplicação de aprendizado de máquina e inteligência artificial no conjunto de dados armazenado, buscando extrair mais valor dos dados coletados.

REFERÊNCIAS

ABB. **Datasheet: MPR0712TE Power Supply**. 2020.

BONOMI, F., MILITO, R., ZHU, J., & ADDEPALLI, S. **Fog computing and its role in the internet of things**: Proceedings of the first edition of the MCC workshop on Mobile cloud computing (MCC '12). Association for Computing Machinery, New York, NY, USA, 13–16. 2012. Fonte: <https://dl.acm.org/doi/10.1145/2342509.2342513>.

CHEN, S., XU, H., LIU, D., HU, B., & WANG, H. **A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective**. IEEE Internet of Things Journal, Vol.1, No. 4, pp. 349-359. doi:10.1109/JIOT.2014.2337336. Agosto de 2014.

CIA. DO METROPOLITANO DE SÃO PAULO. **Relatório Integrado**, 2019. Disponível em <http://www.metro.sp.gov.br/metro/institucional/pdf/relatorio-integrado-2019.pdf>. Acesso em 22 de Outubro de 2020.

CIA. DO METROPOLITANO DE SÃO PAULO. **GC-1.82.CT.YA/7XX-001: ROTEIRO DE MP - P1 CMT MUX E ER/PCO TUC**. 2020. (Documento Interno)

DELTA ELECTRONICS. **IMA 600 Watts Power Supply Series for medical and industrial applications**: Product data sheet. 2016.

ELLIOTT, R. Elliott Sound Products. **Linear Power Supply Design**, 2001. Disponível em <https://sound-au.com/power-supplies.htm>. Acesso em 12 de julho de 2020.

FIWARE FOUNDATION. **ABOUT US**, 2020. Disponível em <https://www.fiware.org/about-us/>. Acesso em 03 de outubro de 2020.

FIWARE FOUNDATION. **DOCS » IOT AGENTS & ROBOTS » IOT AGENT (JSON)**, 2020. Disponível em <https://fiware-tutorials.readthedocs.io/en/latest/iot-agent-json/index.html>. Acesso em 5 de Julho de 2020.

FIWARE FOUNDATION. **INSTALLING ORION**, 2020. Disponível em <https://fiware-orion.readthedocs.io/en/master/admin/install/index.html>. Acesso em 23 de agosto de 2020.

GUPTA, A., YADAV, O. P., DEVOTO, D., & MAJOR, J. **A Review of Degradation Behavior and Modeling of Capacitors: Preprint**. Golden, CO: National Renewable Energy Laboratory, 2018. Disponível em <https://www.nrel.gov/docs/fy19osti/71386.pdf>. Acesso em 12 de julho de 2020,

ITU-T. **Next Generation Networks – Frameworks and functional architecture models**: Overview of the Internet of things. Series Y: Global Information Infrastructure, Internet Protocol Aspects And Next-Generation Networks. Recommendation Y.2060. 2012.

KAGERMANN, H., WAHLSTER, W., & HELBIG, J. **Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry** (Final Report of the Industrie 4.0 Working Group). acatech -- National Academy of Science and Engineering. 2013.

MARTEL INNOVATE. **QuantumLeap Overview**, 2020. Disponível em <https://quantumleap.readthedocs.io/>. Acesso em 23 de agosto de 2020.

MEAN WELL. **3500W Conduction Cooling with PFC Switching Supply. PHP-3500 Series**, 2020. Disponível em <https://www.meanwell.com/Upload/PDF/PHP-3500/PHP-3500-SPEC.PDF>. Acesso em 05 de julho de 2020.

MEAN WELL. **PHP-3500 User's Manual**. 2020.

MENEZES, W. L., & ALVES, M. R. **Modernização dos ATCs L/O para os trens das frota K e L do Metrô**. In: 21ª Semana de Tecnologia Metroferroviária. 2015.

MILLER, H. G., & MORK, P. (2013). **From Data to Decisions: A Value Chain for Big Data**. IT Professional. n. 15. p. 57-59. 2013.

MOBLEY, R. K. **An Introduction to Predictive Maintenance**. Butterworth-Heinemann, 2002.

NATIONAL INSTRUMENTS. **Controller Area Network (CAN) Overview**, 2020. Disponível em <https://www.ni.com/pt-br/innovations/white-papers/06/controller-area-network--can--overview.html>. Acesso em 10 de Janeiro de 2021.

NATIONAL INSTRUMENTS CORP. **Aquisição de sinais analógicos: largura de banda, teorema de amostragem de Nyquist e aliasing**, 2019. Disponível em <https://www.ni.com/pt-br/innovations/white-papers/06/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html>. Acesso em 04 de outubro de 2020.

OPPENHEIM, A. V., WILLSKY, A. S., & NAWAB, S. **Sinais e Sistemas**. São Paulo: Pearson Prentice Hall, 2010.

PERTENCE, A. J. **Amplificadores Operacionais e Filtros Ativos**. Bookman Editora, 2015.

PYTHON SOFTWARE FOUNDATION. **json — JSON encoder and decoder**, 2020. Disponível em <https://docs.python.org/3/library/json.html>. Acesso em 2020 de Outubro de 18.

REITZ, K. **Requests: HTTP for Humans**, 2016. Disponível em <https://requests.readthedocs.io/en/master/>. Acesso em 18 de Outubro de 2020.

RUSHBY, J., Computer Science Laboratory, SRI International. **Critical System Properties: Survey and Taxonomy**. Fevereiro de 1994.

SEEDSTUDIO. **UART vs I2C vs SPI – Communication Protocols and Uses**, 2020. Disponível em <https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses/>. Acesso em 10 de Janeiro de 2021.

SLATER, N. **Pair CrateDB with Grafana 6.x, an Open Platform for Time Series Data Visualization**, 2019. Disponível em <https://crate.io/a/pair-cratedb-with-grafana-6-x/>. Acesso em 7 de Setembro de 2020.

SMARTSDK. **smartsdk/quantumleap**, 2019. Disponível em <https://hub.docker.com/r/smartsdk/quantumleap/>. Acesso em 17 de Outubro de 2020.

STM32-base. **STM32-base project documentation**, 2019. Disponível em <https://stm32-base.org/>. Acesso em 27 de setembro de 2020.

STMICROELECTRONICS. **Medium-density performance line ARM®-based 32-bit MCU with 64 or 128 KB Flash, USB, CAN, 7 timers, 2 ADCs, 9 com. interfaces**, 2015. Disponível em <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>. Acesso em 06 de agosto de 2020.

STOKES, T. **python-spi**, 2015. Disponível em <https://github.com/tomstokes/python-spi>. Acesso em 18 de Outubro de 2020.

SYSTEM MANAGEMENT INTERFACE FORUM, INC. **PMBus™ Power System Management Protocol. Part II - Command Language**, 2010. Disponível em https://www.pmbus.org/Assets/PDFS/Public/PMBus_Specification_Part_II_Rev_1-2_20100906.pdf. Acesso em 18 de Janeiro de 2021.

TDK-LAMBDA. **TPS4000 Series: 4080W 3-Phase Input Industrial Power Supplies**. 2020. Disponível em https://product.tdk.com/info/en/catalog/datasheets/tps4k_e.pdf. Acesso em 5 de Janeiro de 2021.

VISHAY. **Optocoupler, Phototransistor Output, with Base Connection**. 2019. Disponível em <https://www.vishay.com/docs/83725/4n25.pdf>. Acesso em 7 de agosto de 2020.

GLOSSÁRIO

Microcontrolador	Circuito integrado, que encapsula em um mesmo dispositivo o processador, memória volátil, memória persistente, periféricos e entradas e saídas programáveis.
Raspberry Pi	Computador desenvolvido em uma placa única, contendo interface de rede, de vídeo, de áudio e saídas e entradas digitais, de acordo com as diversas versões disponíveis.
<i>Container</i>	Camada de virtualização em nível de sistema operacional, que garante isolamento entre softwares e atendimento a todas as dependências necessárias ao serviço executado no <i>container</i> .

APÊNDICE A – COLABORAÇÃO DE ESPECIALISTAS PARA LEVANTAMENTO DE REQUISITOS

No trecho a seguir estão reproduzidas as colaborações dos Engenheiros Carlos Alfredo Santos, supervisor de manutenção, e Francisco Carlos Lopes Mathias, engenheiro de manutenção, para levantamento de requisitos para um sistema de monitoramento das fontes de alimentação do sistema de sinalização do Metrô de São Paulo.

“As fontes de alimentação são para alimentação de equipamentos dos sistemas ATC - Automatic Train Control, ATP - Automatic Train Protection (em estações específicas conhecidas como estações mestras) e ATO - Automatic Train Operation (em todas as estações). O ATP é utilizado na linha 1 e 3, a linha 2 utiliza o CBTC.

O ATP é um sistema com a filosofia de falha segura, é responsável por paralisar a circulação de trens no trecho sobre domínio quando qualquer situação de anormalidade de segurança for detectada.

O Interlocking, ou ILK, possui a finalidade de controlar a circulação de trens, selecionando o código de velocidade com base nas condições da via a fim de evitar acidentes. Esse sistema realiza a detecção de presença de trem em circuitos de mudança de via, alinhamento de rota nos trechos de mudança de via e verificação de aproximação entre trens e rotas conflitantes (entre trens na mesma via em direção opostas). Trabalha com as fontes de alimentação de 28 Vdc e 12 Vdc.

O sistema Multiplex, ou MUX, é responsável por gerar o código de velocidade para seleção do ILK. Envia e recebe da via, por onde passam os trens, o código de velocidade para os trens, realiza a comparação da transmissão e recepção e faz a detecção de ocupação de via em subtrechos de via (denominado de circuito de via).

Para a transmissão e recepção do código de velocidade ao trem, existe uma caixa a margem de via (sobre a passarela de emergência) denominada de WaySideBox (WSB) de um lado a transmissora e, a aproximadamente 150 m, a receptora. O MUX compara o código de velocidade que enviou e recebeu, em caso de divergência é gerada a ocupação real do circuito, onde o trem realiza o *shunt* [contato] entre os 2

trilhos ou a falsa ocupação, quando não há realmente um trem sobre o circuito, assim gerando uma falha urgente.

O sistema ATO realiza o controle de parada programada dos trens na plataforma, abertura e fechamento de portas dos trens, de forma automática.

Todos os equipamentos possuem uma priorização nas intervenções, a maior prioridade é da atuação local por operador qualificado e autorizado, depois as intervenções centralizadas, realizadas pelo centro de controle operacional – CCO e, por último, as intervenções automáticas.

As falhas das fontes são por ausência de tensão na saída ou degradação da tensão de saída, ripple alto, tensão baixa e, no MUX, geração de código de velocidade mais alto para o perfil de velocidade de via, quando em sobretensão.

As falhas mais comuns nas fontes do ILK são: ausência de 28 Vdc, o que gera o não alinhamento de rota em trecho de aparelho de mudança de via, devido ao fato de que as máquinas de chave que geram a mudança de via ficam sem comando remoto. A ausência de 12 VDC provoca o não alinhamento de rota, pois os relês responsáveis pela lógica de acionamento não são alimentados.

Para o MUX, onde existe dualidade de alimentação, 12 Vdc, +6 Vdc e -6 Vdc, em caso de falha não gera o código de velocidade, não detecta ocupação e não envia os códigos de velocidade para os trens.

Para o ATO, não realiza a parada programada do trem na plataforma, não realiza abertura e fechamento de porta, e, conseqüentemente, não gera sinalização destas tarefas.

Os dados que considero importantes para coleta e armazenamento são: tensão na saída das fontes, ripple na saída das fontes, tensões na barra bus (onde se distribui para cada gaveta do armário - MUX e ATO) e a temperatura. A corrente e a potência são conhecidas e baixas, não sendo necessário sua análise.

As fontes do ILK são dualizadas (2 de 12 Vdc e 2 de 28 Vdc), o MUX (2 de +6 Vdc, 2 de -6 Vdc e 2 de +6 Vdc para gerar 12Vdc), ATO (1 de 12 Vdc e 1 de 6 Vdc).

Considero que as coletas dos dados da fonte devem ser realizadas a cada a cada 3 minutos, após a verificação de 3 sinais consecutivos, ou seja, após 9 minutos, será apresentada indicação de possível falha.

A manutenção era realizada com base na programação de manutenção elaborada pela Engenharia de Manutenção para as equipes de campo realizarem periodicamente inspeções técnicas, a cada 6 meses, para realizar medições de tensões e verificação de anormalidades para programar atividades futuras para correção.

O sinal era medido à noite, registrado em papel, esperando-se uma confiabilidade na equipe que realizava a inspeção técnica, normalmente composta por dois técnicos, após operação comercial, ou seja, era uma medida estática.

A medição deve ser online e dinâmica, assim possibilitando a manutenção em função da condição do ativo e um melhor rastreamento histórico da vida desses componentes, economizando em mão de obra, deslocamento de equipe até o local, e viabilizando uma manutenção mais assertiva.

Uma equipe de manutenção trata os alarmes centralizados. Em conjunto com as engenheiras e equipes de campos foram definidos critérios de escalonamento e severidades para os níveis de tensão observados nas fontes, assim as atividades podem ser programadas para correções antes dos sistemas falharem, melhorando dessa forma a assertividade da manutenção e disponibilidade operacional para o principal cliente do metrô que é o passageiro.”

Colaboração do Eng.^o Francisco Carlos Lopes Mathias:

“A falha mais comum nas fontes da linha 1 (tipo series) é tensão baixa devido a perda de regulação, este evento é relativamente abrupto, sendo socorrido com a dualidade.

Na linha 3, as fontes (tipo chaveadas primeira geração), ficam sem tensão de saída também abruptamente, devido a atuação da proteção que queima o fusível de saída, geralmente consequência de manobras da parte elétrica que elas não suportam (manobras nas subestações e manutenção em inversores que fornecem energia para

essas fontes), e também por falha no seu circuito de chaveamento, o que é muito crítico.

Somente a coleta e registro da tensão permite realizar rapidamente a substituição de um componente defeituoso antes de perder toda a função de alimentação do sistema, pois, existindo a dualidade, a função ainda é preservada.

Cada conjunto de fontes duais do MUX é composto dos seguintes pontos de coleta:

- 06 pontos por equipamento MUX que necessita de alimentação de +12, +6, -6 Volts;
- 02 Pontos para cada fonte de +12 Volts;
- 02 Pontos para cada fonte de +06 Volts;
- 02 Pontos para cada fonte de - 06 Volts.

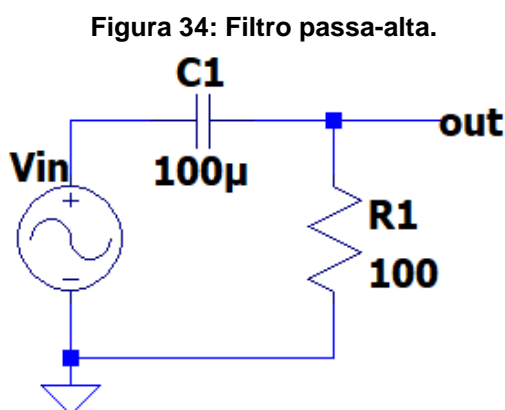
A frequência de leituras pode ser muito baixa, pois antes as inspeções eram realizadas em intervalos de meses. Na linha 3, o ideal é a coleta de dados sempre que houver distúrbios na parte elétrica.

O monitoramento das condições de alimentação elétrica das fontes, além da tensão de saída, pode trazer informações sobre seu comportamento, pois esses distúrbios causados por algum evento externo podem stressá-las. Uma evidência desse parecer é que os equipamentos alimentados por bancos de bateria são mais estáveis.”

APÊNDICE B – PROJETO DO CIRCUITO DE ADEQUAÇÃO DO SINAL

Descreve-se a seguir o projeto do circuito eletrônico responsável pela compatibilização das tensões elétricas monitoradas com o conversor AD mantendo o isolamento do ativo e os requisitos de precisão especificados.

O filtro utilizado para separar a tensão de corrente contínua da tensão de corrente alternada foi um filtro passivo simples RC, conforme ilustrado na Figura 34.



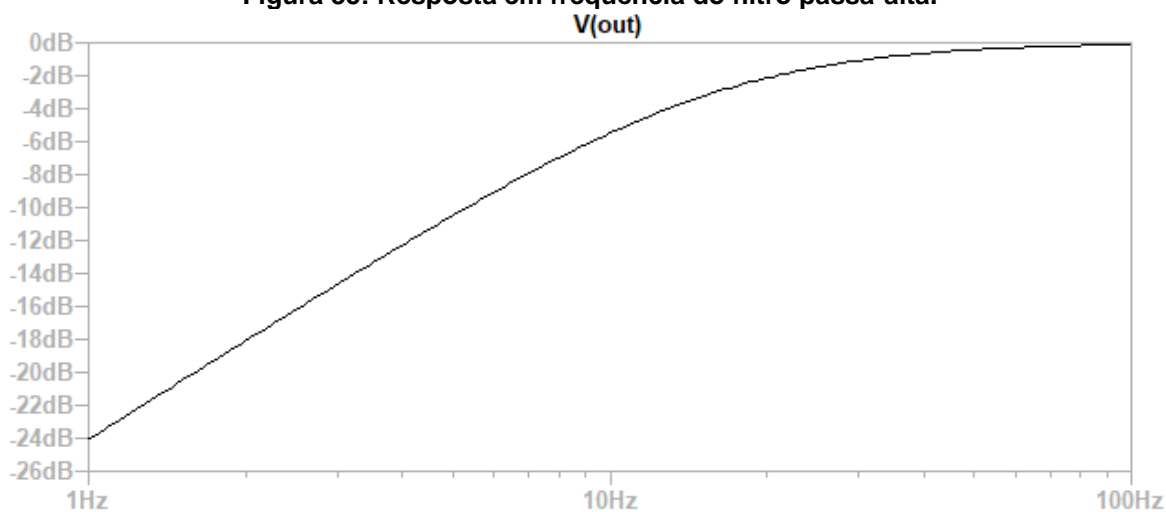
Fonte: Próprio autor.

A frequência de corte do filtro passa-alta é:

$$f_c = \frac{1}{2\pi \cdot R \cdot C}$$

Portanto, a frequência de corte é de aproximadamente 15,9 Hz, e o diagrama de resposta em frequência está reproduzido na Figura 35.

Figura 35: Resposta em frequência do filtro passa-alta.



Fonte: Próprio autor.

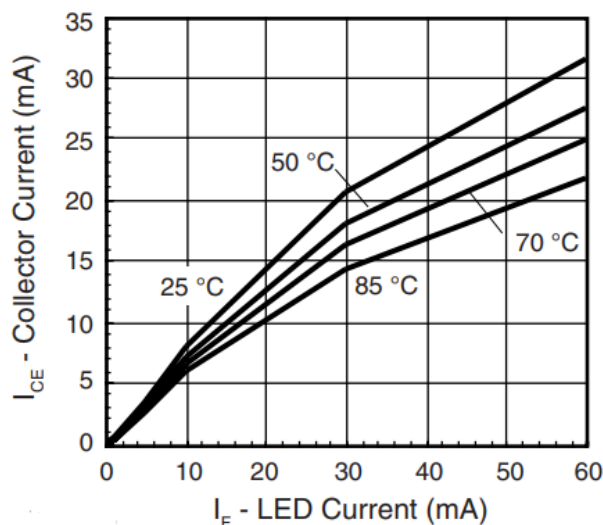
Isso significa que o circuito atenuará consideravelmente os valores do sinal que oscilarem com menos de 15,9 Hz, ou seja, a tensão constante de saída da fonte não fará parte do sinal resultante, permitindo medir com mais detalhes os valores que oscilarem em torno desse sinal constante.

Essa separação do sinal permite que os sinais contínuo e alternado sejam amplificados de forma diferente, uma vez que é esperado que o sinal alternado possua uma escala muito menor.

Para se projetar corretamente o amplificador, deve-se verificar a entrada necessária para o optoacoplador, de forma que esse componente opere em uma região fora de saturação e de uma forma que seja mais linear possível, ou seja, de forma que a saída seja proporcional à entrada.

A corrente do transistor de saída do optoacoplador é uma função da corrente no LED da entrada. Essa relação não é linear para toda a faixa de corrente de entrada suportada pelo componente, conforme pode ser verificado no datasheet de cada fabricante. Utiliza-se para esse projeto o 4N25, fabricado pela VISHAY, sua curva que apresenta a corrente de saída em função da corrente de entrada e da temperatura está reproduzida na Figura 36.

Figura 36: Corrente no coletor do optoacoplador 4N25 em função da corrente no LED e da temperatura ambiente ($V_{ce}=10V$).



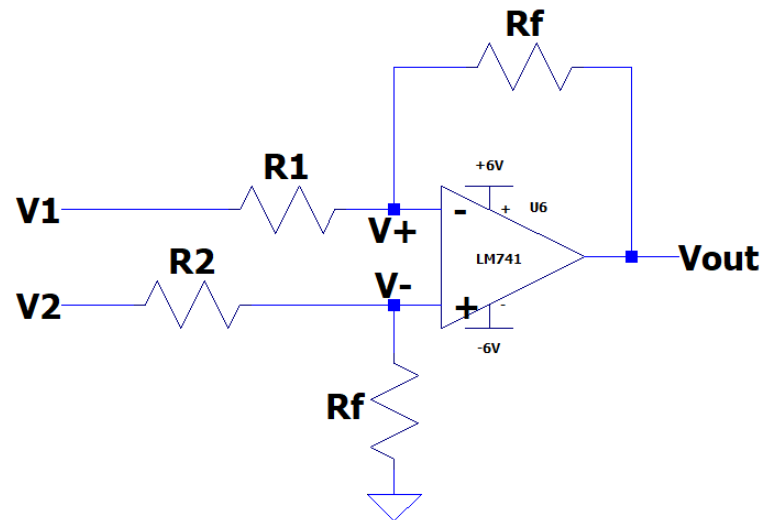
Fonte: (VISHAY, 2019)

Para simplificar o processamento dos valores, deve-se trabalhar em uma região linear de operação do componente, por isso, a corrente de excitação do LED será dimensionada para valores menores que 5 mA.

Para a parte CC do sinal monitorado, consegue-se obter essa corrente apenas adicionando uma carga em série com o optoacoplador, dessa forma, não é necessário amplificar o sinal.

Para a componente CA obtida com a filtragem do sinal, por ser esperado um sinal inferior a 10mV e simétrico (com valores positivos e negativos), será necessário amplificação e ajuste de *offset*, uma vez o que o optoacoplador somente transmite sinais positivos pela característica do LED da entrada. Para isso, foi utilizado um amplificador operacional em ligação do tipo amplificador diferencial, conforme diagrama apresentado na Figura 37.

Figura 37: Diagrama de circuito amplificador diferencial.



Fonte: Adaptado de Pertence (2015).

Considerando $V_+ = V_-$ e $i_+ = i_- = 0$, que são as considerações para um amplificador operacional ideal, é possível deduzir que:

$$V_{out} = V_2 \cdot \left(\frac{R_f}{R_f + R_2} \right) \cdot \left(\frac{R_f + R_1}{R_f \cdot R_1} \right) - \frac{V_1}{R_1}$$

Para $R_1 = R_2$:

$$V_{out} = \frac{R_f}{R_1} \cdot (V_2 - V_1)$$

A corrente que alimentará o optoacoplador deverá variar entre 0 e 4mA, para tanto, descontando a queda de tensão de aproximadamente 1V no LED de entrada do optoacoplador e um resistor de 1 k Ω em série, temos, a tensão máxima de saída do amplificador (V_{out}) necessária dada por:

$$V_{out} - 1V = 4mA \cdot 1k\Omega \therefore V_{out} = 5V$$

Como o optoacoplador somente trabalha com correntes positivas, um *offset* é necessário, ou seja, deve-se somar uma tensão DC positivo à saída. Para uma excursão de 4V com valor inicial de 1V de queda no LED, o valor central, de *offset*, deverá ser 3V, e o ganho necessário a partir da variação inicial de 100mV é de 20, portanto, devemos somar uma tensão 20 vezes menor em V_2 , dada por:

$$V_2 = \frac{3 V}{20} = 0,15 V$$

Um divisor resistivo fornece em V_2 a tensão de referência necessária de 150 mV para somar no amplificador diferencial, na Figura 38 identificado por R_a e R_b .

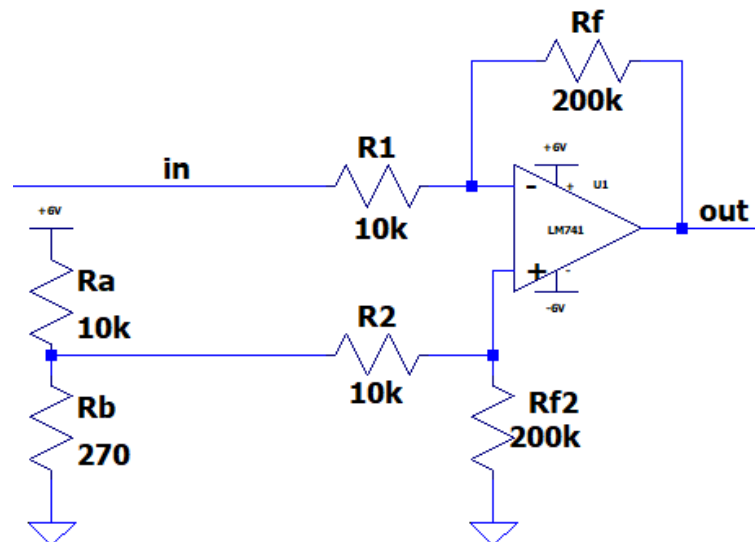
Os valores de R_f , R_1 e R_2 escolhidos para satisfazer os cálculos acima foram 200k Ω , 10k Ω e 10k Ω , respectivamente. Substituindo-os na equação do amplificador diferencial:

$$V_{out} = \frac{200k\Omega}{10k\Omega} \cdot (0,15 V - V_1)$$

Se é esperado que V_1 varie entre -0,1 V e +0,1 V, logo $V_{out(max)} = 5 V$ e $V_{out(min)} = 1 V$.

Dessa forma, o circuito de amplificação para a parte AC fica conforme esquema apresentado na Figura 38.

Figura 38: Esquemático do circuito de amplificação da componente AC (ripple+ruído).



Fonte: Próprio autor.

Para a componente DC, que será obtida conectando o sinal diretamente ao optoacoplador, é necessário dimensionar um resistor para limitar a corrente do LED.

Conforme já abordamos acima, é necessário obter uma corrente máxima entre 4 e 5 mA no LED para que haja espaço para excursão do sinal e leitura do valor pelo conversor analógico-digital.

Descontada a queda de tensão no LED do optoacoplador de aproximadamente 1V, para obter a corrente desejada quando existir tensão de 0 a 15V na fonte, temos:

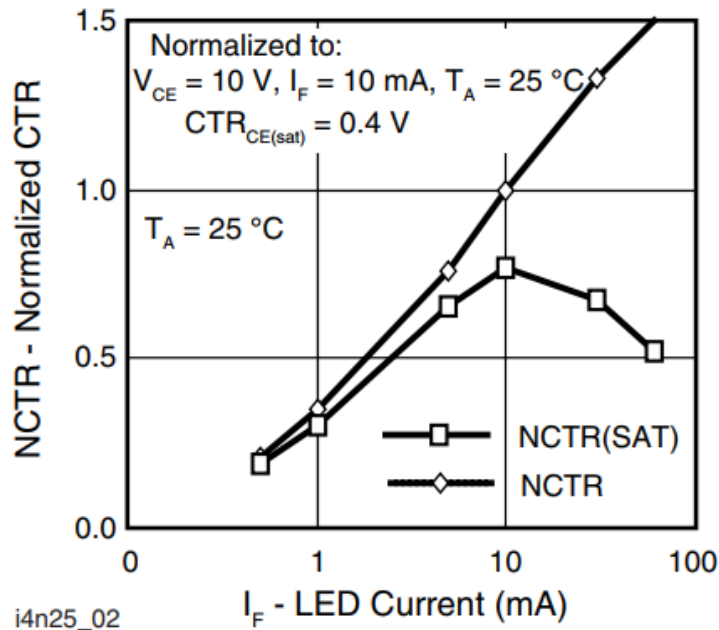
$$R = \frac{(15\text{ V} - 1\text{ V})}{4\text{ mA}} = 3500\ \Omega$$

Arredondando para um valor comercial próximo, um resistor de 3,3 k Ω em série com o LED do optoacoplador resulta em uma corrente de 4,24 mA para 15V.

O optoacoplador possui um parâmetro informado pelo fabricante chamado CTR, do inglês *current transfer ratio*, ou taxa de transferência de corrente. Esse parâmetro indica o comportamento do transistor de saída do componente em função da corrente de excitação do LED de entrada, geralmente indicado por uma porcentagem. Além do seu valor possuir uma faixa muito incerta indicada pelos fabricantes de optoacopladores de uso geral, com valores entre 30% e 80%, o CTR varia de acordo com a corrente de entrada.

No gráfico da Figura 39, reproduzido a partir do *datasheet* do fabricante, é possível verificar que o CTR não é constante para toda a faixa de operação, o que seria ideal para obter a linearidade da leitura mesmo com o acoplamento ótico.

Figura 39: Variação de CTR pela corrente no LED do optoacoplador (normalizada).



Fonte: (VISHAY, 2019)

Para prover uma leitura mais precisa dos valores de tensão, uma alternativa é prever uma forma de calibração individual em cada sensor, por isso, será adicionado um resistor variável em série com o transistor de saída, a ser ajustado com um sinal de referência.

No software de simulação LTSpice, foi verificado para uma tensão de 3,3V que o CTR do modelo do optoacoplador é de 80%, ou seja, a corrente na saída (I_c) é obtida por:

$$I_c = CTR \cdot I_f \rightarrow I_c = 0,8 \cdot I_f$$

Outro parâmetro que deve ser considerado é o V_{CE} , ou seja, a tensão no transistor de saída. Para se obter o maior alcance no sensor, realiza-se o cálculo para o ponto central do alcance do sensor. Para a entrada uma tensão de 7,5 V e para a saída um $V_{CE} = V_{CC}/2$, resultando em 1,65 V para a tensão de 3,3 V de alimentação do conversor AD.

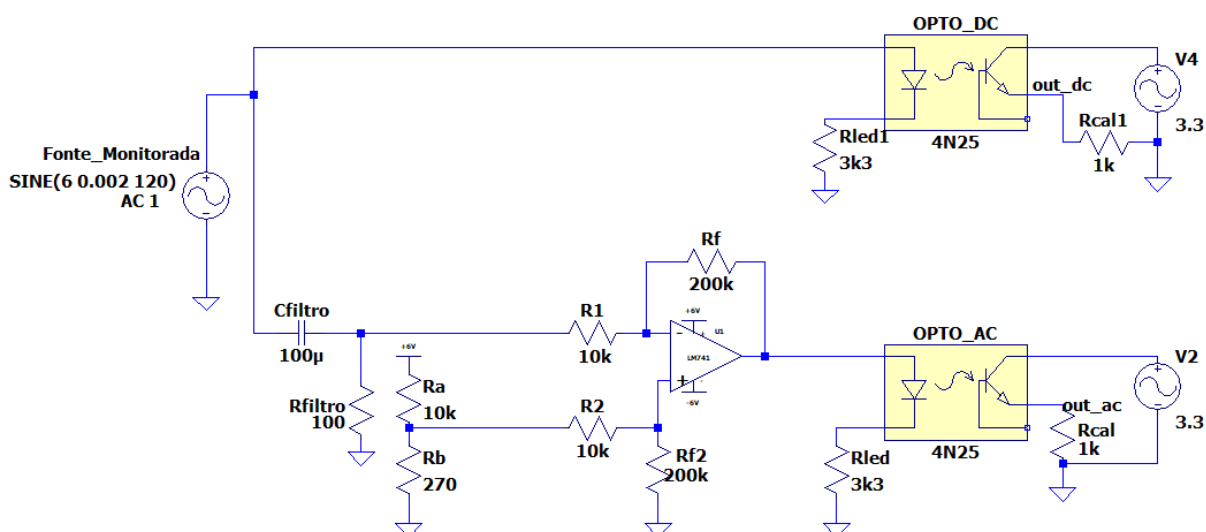
Dessa forma, o valor do resistor a ser conectado em série ao transistor de saída pode ser obtido por:

$$R = \frac{1,65 \text{ V}}{\frac{7,5 \text{ V} - 1 \text{ V}}{3300 \Omega} \cdot 80\%} = \frac{1,65 \text{ V}}{\frac{7,5 \text{ V} - 1 \text{ V}}{3300 \Omega} \cdot 80\%} = \frac{1,65 \text{ V}}{1,97 \text{ mA} \cdot 80\%} = 1046,95 \Omega$$

Aproxima-se para o valor comercial de 1 k Ω , mas ajustável para adequar a diferenças de construção dos optoacopladores.

Antes da construção de um protótipo que possibilite verificar a solução proposta, foi projetado o circuito apresentado na Figura 40 e simulado com utilização do software LTSpice, contemplando tanto a parte DC quanto a parte AC.

Figura 40: Circuito de aquisição simulado no software LTSpice.



Fonte: Próprio autor.

Dessa forma, nos pontos out_dc e out_ac, espera-se a saída entre 0 e 3,3 V compatível com o conversor AD.

APÊNDICE C – SOFTWARE DESENVOLVIDO PARA O GATEWAY

Software desenvolvido na linguagem Python para ser executado na Raspberry Pi de forma a exercer a função de *gateway IoT*.

Por se tratar de uma reprodução do código executado no interpretador Python, os comentários estão sem acentuação, uma vez que os caracteres especiais são incompatíveis com o interpretador.

```
import spi
import requests
import json
import time
import RPi.GPIO as gpio

# Informacoes do servidor Orion
IP_Orion = '192.168.1.15'
intervaloEntreLeituras = 10

# Informacoes sobre a fonte monitorada
type = 'fonte_001'
station = 'JAB' #estacao na qual o monitor esta instalado
psName = 'MUXA' #nome da fonte sendo monitorada, no caso, a tensao positiva da raspberry

# ID conforme nomenclatura FIWARE
id = 'urn:ngsi-ld:'+type+':'+station+':'+psName

# Inicializacao para requisicoes HTTP
headers = {'Content-type': 'application/json'} # cabecalho utilizado para post

# Inicializacao do SPI
pSpi = spi.SPI('/dev/spidev0.0')
pSpi.mode = spi.SPI.MODE_0
pSpi.bits_per_word = 8
pSpi.speed = 5000000 # 5Mbps
gpio.setmode(gpio.BCM)
CS_PIN = 8
gpio.setup(CS_PIN, gpio.OUT)
```

```
#canais e calibracao do medidor
```

```
channel_ac = 0x01
```

```
channel_dc = 0x02
```

```
cal_dc = 0.003412 #V/bit
```

```
cal_ac = 0.000043 #V/bit
```

```
offset = 0.4 #volts
```

```
#Verificar as entidades cadastradas, se nao encontrar o dispositivo, enviar a requisicao de novo cadastro
```

```
r=requests.get('http://'+IP_Orion+':1026/v2/entities')
```

```
entities = r.json()
```

```
idFound = False
```

```
for entity in entities:
```

```
    if(entity['id'] == id):
```

```
        idFound = True #ja encontrou o id, sai do laço de repeticao
```

```
        break
```

```
if(not idFound): #se nao encontrou a entidade, faz a requisicao para cria-la
```

```
    pload = {
```

```
        "id": id,
```

```
        "type": type,
```

```
        "rms_ac": {
```

```
            "type": "Float"
```

```
        },
```

```
        "med_dc": {
```

```
            "type": "Float"
```

```
        },
```

```
        "max": {
```

```
            "type": "Float"
```

```
        },
```

```
        "min": {
```

```
            "type": "Float"
```

```
        }
```

```
    }
```

```
    r = requests.post('http://'+IP_Orion+':1026/v2/entities',data = json.dumps(pload), headers = headers)
```

```
    print('Criou a entidade:\n'+json.dumps(pload, indent=1))
```

```
#realiza nova busca para confirmar a criacao da entidade
```

```
r=requests.get('http://'+IP_Orion+':1026/v2/entities')
```

```
entities = r.json()
```

```

idFound = False
for entity in entities:
    if(entity['id'] == id):
        idFound = True
        break
if(not idFound):
    print ("Erro - nao foi possivel criar a entidade")
    exit (2)

while (True):
#Faz n leituras do valor analogico e calcula max, min, med, rms_ripple
    #inicializacao de variaveis a cada leitura
    sum = 0
    sum_rms_ac = 0
    max = -255
    min = 255
    n_leituras = 100 # numero de leituras de cada amostragem

    #na primeira comunicacao informa quais entradas analogicas serao utilizadas
    gpio.output(CS_PIN, gpio.LOW)
    received = pSpi.transfer([0x11, 0x22, channel_dc, 0x44])
    gpio.output(CS_PIN, gpio.HIGH)

    for i in range(n_leituras):
        gpio.output(CS_PIN, gpio.LOW)

# recebe o canal de interesse o solicita o próximo
    received = pSpi.transfer([0x11, 0x22, channel_ac, 0x44])

    # verificacao de paridade e do canal que esta sendo enviado
    if(((received[0] ^ received[1] ^ received[2]) == received[3]) and (((received[1]>>4) & 0x0F) ==
channel_dc)) :
        reading = (received[1]<<8 & 0x0F00) | (received[2] & 0xFF) #se tiver sucesso une os 12
bits recebidos
        reading = reading * cal_dc #multiplica pelo valor de calibracao (Vrms/bit)
        sum = sum + reading
        if (reading > max) : max = reading
        if (reading < min) : min = reading
    else:
        print('Erro na verificacao do DC')

```

```

# recebe o canal de interesse o solicita o próximo
    received = pSpi.transfer([0x11, 0x22, channel_dc, 0x44])
    gpio.output(CS_PIN, gpio.HIGH)
    # verificacao de paridade e do canal que esta sendo enviado
if((received[0] ^ received[1] ^ received[2] == received[3]) and (((received[1]>>4) & 0x0F) ==
channel_ac)) :
    reading = (received[1]<<8 & 0x0F00) | (received[2] & 0xFF) #se tiver sucesso une os 12
bits recebidos
    reading = reading * cal_ac - offset # multiplica pelo valor de calibracao (V/bit) e subtrai
o offset
    sum_rms_ac = sum_rms_ac + reading ** 2
else:
    print('Erro na verificacao do AC')

# calculo das informações referentes as leituras
mean = sum / n_leituras
rms_ripple = (sum_rms_ac/n_leituras - (mean ** 2)) ** 0.5
print('RMS='+str(rms_ripple)+' Mean='+str(mean)+' Max='+str(max)+' Min='+str(min))
pload= {
    "rms_ac": {
        "value": rms_ripple,
        "type": "Float"
    },
    "med_dc": {
        "value": mean,
        "type": "Float"
    },
    "max": {
        "value": max,
        "type": "Float"
    },
    "min": {
        "value": min,
        "type": "Float"
    }
}
requests.post('http://'+IP_Orion+':1026/v2/entities/'+id+'/attrs',data = json.dumps(pload), headers
= headers)
time.sleep(intervaloEntreLeituras)

```