

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Jeann Carlo Martins Raguzzoni

**PLATAFORMA AGREGADORA PARA RECURSOS ENERGÉTICOS
DISTRIBUÍDOS**

Santa Maria, RS
2022

Jeann Carlo Martins Raguzzoni

**PLATAFORMA AGREGADORA PARA RECURSOS ENERGÉTICOS
DISTRIBUÍDOS**

Dissertação apresentada ao curso de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do título de **Mestre em Engenharia Elétrica**.

Orientadora: Prof.^a Dr.^a Luciane Neves Canha

Santa Maria, RS
2022

Declaro, JEANN CARLO MARTINS RAGUZZONI, para os devidos fins e sob as penas da lei, que a pesquisa constante neste trabalho de conclusão de curso (Dissertação) foi por mim elaborada e que as informações necessárias objeto de consulta em literatura e outras fontes estão devidamente referenciadas. Declaro, ainda, que este trabalho ou parte dele não foi apresentado anteriormente para obtenção de qualquer outro grau acadêmico, estando ciente de que a inveracidade da presente declaração poderá resultar na anulação da titulação pela Universidade, entre outras consequências legais.

Ficha gerada com os dados fornecidos pelo autor

Raguzzoni, Jeann Carlo Martins
Implementação Prática de uma plataforma agregadora para recursos energéticos distribuídos / Jeann Carlo Martins
Raguzzoni.-2022.
54 p.; 30cm

Orientadora: Luciane Neves Canha
Dissertação (mestrado) – Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-graduação em Engenharia Elétrica, RS, 2022.

1. Recursos energéticos distribuídos 2. Smart Grid 3. Plataforma integradora 3. Agregadores 4. Middleware I. Canha, Luciane Neves II. Implementação Prática de uma plataforma agregadora para recursos energéticos distribuídos.



Este trabalho está licenciado pela [Creative Commons Attribution-ShareAlike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

This work is licensed under the [Creative Commons Attribution-ShareAlike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

Jeann Carlo Martins Raguzzoni

**PLATAFORMA AGREGADORA PARA RECURSOS ENERGÉTICOS
DISTRIBUÍDOS**

Dissertação apresentada ao curso de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do título de **Mestre em Engenharia Elétrica**.

Aprovado em 28 de janeiro de 2022:

Luciane Neves Canha, Dra. (UFSM)
(Presidente/Orientadora)

Rafael Gressler Milbradt, Dr. (UFSM)

Paulo Ricardo da Silva Pereira, Dr. (UNISINOS)

Santa Maria, RS
2022

AGRADECIMENTOS

A minha orientadora, Professora Luciane Neves Canha, pela oportunidade de orientação, dedicação, conhecimento transmitido e, sobretudo, confiança e motivação, acreditando em meu potencial para desenvolver este trabalho.

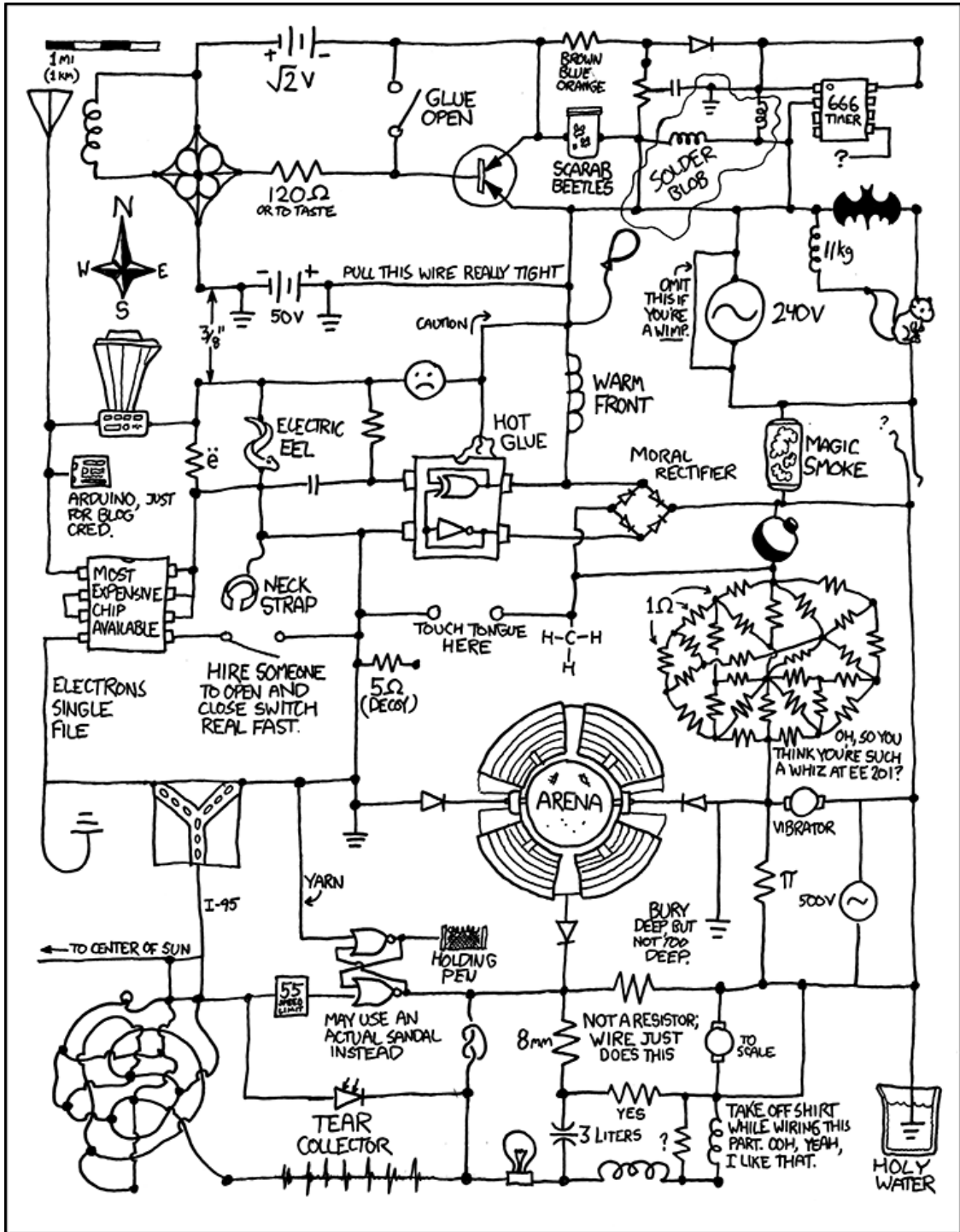
Ao professor Rafael Gressler Milbradt, pelo auxílio fornecido no desenvolvimento de partes mais técnicas.

À Universidade Federal de Santa Maria, ao Programa de Pós-Graduação em Engenharia Elétrica, ao Núcleo de Pesquisa e Desenvolvimento de Engenharia Elétrica pelo suporte no desenvolvimento do trabalho e toda infraestrutura cedida.

Ao apoio financeiro do Programa de Pesquisa e Desenvolvimento ANEEL e Copel-Dis através do projeto de P&D PD 2866/0462-2016, à equipe da NHS por proporcionar o inversor híbrido para os testes e desenvolvimentos.

Aos meu setor (NUPEDDEE) e colegas de trabalho pela compreensão, incentivo e disponibilidade do horário especial.

A minha namorada Luizi, pela compreensão e força nos momentos difíceis.



RESUMO

IMPLEMENTAÇÃO PRÁTICA DE UMA PLATAFORMA AGREGADORA PARA RECURSOS ENERGÉTICOS DISTRIBUÍDOS

AUTOR: Jeann Carlo Martins Raguzzoni

ORIENTADORA: Luciane Neves Canha

Há muitos anos os sistemas de distribuição de energia elétrica têm operado com poucas mudanças, especialmente quando se trata do consumidor. Com a chegada de equipamentos de medição avançada, instalação de sistemas de geração de energia renováveis e outros recursos energéticos distribuídos, uma significativa alteração começa a acontecer, exigindo que as empresas, clientes e operadores dos sistemas de distribuição tenham de se adaptar. As mudanças têm produzido impactos em diferentes áreas, seja na forma das cobranças, atualização de equipamentos, legislação, registro e análise de dados, dentre outras. Os recursos energéticos distribuídos unidos aos consumidores, que agora passam a ser chamados de prosumidores, trazem um novo paradigma para o sistema de distribuição e exigem uma maior acurácia em relação aos dados produzidos e registrados. Os recursos energéticos distribuídos são diversos, tais como módulos fotovoltaicos, sistemas de armazenamento em baterias, veículos elétricos e podem provocar fluxos de energia bidirecionais. Toda essa gama de mudanças provoca diferentes tipos de impactos sobre as redes elétricas e conseqüentemente sobre os tipos de dados que são gerados. Esta dissertação busca abordar as dificuldades e necessidades ao se conectarem diferentes recursos energéticos distribuídos e equipamentos a uma rede elétrica inteligente. O trabalho propõe uma solução que permita agregar os recursos de carga, geração e armazenamento que estão no lado do consumidor à operação das redes de distribuição, criando ao fim uma solução de *software* e *hardware*. Ao final é desenvolvida uma plataforma agregadora para recursos energéticos distribuídos de forma a fazer a integração de distintos equipamentos em um só sistema, para que se comuniquem entre si, possibilitando diversas aplicações nas redes elétricas inteligentes, como ações de resposta à demanda e contratos inteligentes.

Palavras-chave: Recursos energéticos distribuídos. Smart Grid. Plataforma integradora. Agregadores. *Middleware*.

ABSTRACT

PRACTICAL IMPLEMENTATION OF AN AGGREGATION PLATFORM FOR DISTRIBUTED ENERGY RESOURCES

AUTHOR: Jeann Carlo Martins Raguzzoni

ADVISOR: Luciane Neves Canha

For many years our energy distribution system structure has been virtually the same, particularly for the consumer. With the arrival of advanced metering equipment, renewable energy implementations and other distributed energy resources, a sudden change begins to happen, requiring that both consumer and the provider sides to adapt. Those changes have an impact in many areas, be them be related to how the consumers are charged, new equipment to replace old ones, legislation, collection and data analysis, among other changes. The distributed energy resources alongside consumers that can now be called prosumers now bring a new paradigm for the energy distribution system and require better accuracy in relation to the data that is being generated and stored. These new energy resources are diverse, such as photovoltaic panels, battery storage systems, electric vehicles and may imply on a bidirectional flow of energy. All of those changes cause different kinds of impact over the electric grid and consequentially over the kind of data that is being generated. This master thesis looks to address the difficulties and necessities that exist on connecting different kind of distributed energy resources to a smart grid. This work proposes a solution that allows the integration of many load, generation and storage resources located on the consumer side to the operation side of the *smart grid*, resulting in a *software* and *hardware* implementation. Finally, at the end an aggregator platform for distributed energy resources is developed to integrate distinct equipment in a single place, so that they can communicate transparently with each other, allowing for many others kind of applications on the smart grid, such as demand response algorithms and smart contracts.

Keywords: Distributed Energy Sources. Smart grid. Integration Platform. Aggregators. Middleware.

LISTA DE FIGURAS

Figura 1 - Ilustração identificando o que está por trás do medidor	16
Figura 2 - Dashboard Desenvolvido no Sistema de Monitoramento	19
Figura 3 - Arquitetura do Supervisório de Nitulescu e Korodi	20
Figura 4 - Fluxo e planejamento de Execução	22
Figura 5 - Idealização do Middleware	24
Figura 6 - Middleware e suas conexões com dispositivos de uma smart home	25
Figura 7 - Raspberry pi 3 Model B+ utilizado no projeto	26
Figura 8 - Medidor LandisGyr E34A	28
Figura 9 - Software da LandisGyr para leitura e análise	29
Figura 10 - Exemplo de comunicação já com alguns dados decodificados (em vermelho, dados que batiam com os medidos externamente, apenas a Fase A havia sido conectada)	30
Figura 11 - Tela inicial do Software OLIMPO da Eletra	32
Figura 12 - Dados Sendo Lidos no software da Eletra	32
Figura 13 - Destaque dos Dados DLMS/COSEM no software OLIMPO	33
Figura 14 - Ficha Técnica do Inversor Híbrido NHS de 3kW	34
Figura 15 - Inversor NHS instalado em laboratório da UFSM com 2 bancos de bateria de chumbo-ácido	35
Figura 16 - Porção de uma das tabelas de dados no modelo Sunspec do inversor	35
Figura 17 - Dados da geração solar (W) já armazenados no InfluxDB	36
Figura 18 - Fluxograma de funcionamento geral do software agregador para equipamentos passivos (medidores inteligentes, sensores)	37
Figura 19 - Fluxograma de funcionamento do software agregador para o Inversor	38
Figura 20 - Um dos dashboards criados no Grafana para visualização geral dos dados	39
Figura 21 - Consulta personalizada de valores diretamente da base de dados (potência solar)	39
Figura 22 - Painel para visualização em tabela e exportação dos dados	40
Figura 23 - Fluxo do Node-RED para controle via API JSON	42
Figura 24 - Dados reais do Sistema com atuação do algoritmo externo	43
Figura 25 - Fluxo do dashboard	44
Figura 26 - Dashboard de controle do Inversor via Node-RED	44
Figura 27 - Raspberry pi utilizado e roteador	46
Figura 28 - Visão geral da instalação no NUPEDDEE	46
Figura 29 - Raspberry pi protegido e escondido atrás da TV	47
Figura 30 - Visão geral da instalação no Container do Carport	47
Figura 31 - Raspberry pi com Home Assistant em caixa própria	48

LISTA DE TABELAS

Tabela 1 - Tabela de Comandos Sunspec Inversor NHS.....	41
---	----

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ANEEL	Agência Nacional de Energia Elétrica
API	Application Protocol Interface
ARM	Advanced RISC Machine
CLP	Controlador Lógico Programável
COSEM	Companion Specification for Energy Metering
CSV	Comma Separated Values
DB	Database
DER	Distributed Energy Resource
DLMS	Device Language Message Specification
IOT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
NAT	Network Address Translation
NBR	Norma Brasileira
NUPEDDEE	Núcleo de Pesquisa e Desenvolvimento em Engenharia Elétrica
RED	Recurso Energético Distribuído
RN	Resolução Normativa
SSH	Secure Shell
TCP	Transmission Control Protocol
UFSM	Universidade Federal de Santa Maria
USB	Universal Serial bus

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MOTIVAÇÃO E JUSTIFICATIVA	12
1.2	PRINCIPAIS CONTRIBUIÇÕES	13
1.3	OBJETIVOS.....	14
1.3.1	Objetivo geral	14
1.3.2	Objetivos específicos	14
1.4	DISTRIBUIÇÃO DO ASSUNTO.....	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	RECURSOS ENERGÉTICOS DISTRIBUÍDOS (REDS)	15
2.2	MEDIDORES INTELIGENTES E CONSUMIDOR	15
2.3	INVERSORES CONVENCIONAIS E HÍBRIDOS	16
2.4	PROTOCOLOS <i>MODBUS</i> E <i>SUNSPEC</i>	16
2.5	CONCLUSÃO DO CAPÍTULO	17
3	REVISÃO BIBLIOGRÁFICA	18
3.1	AÇÕES DE GERENCIAMENTO PELO LADO DA DEMANDA, RESPOSTA A DEMANDA, GERENCIAMENTO DE REDS.....	18
3.2	INFLUX E GRAFANA.....	18
3.3	NODE-RED.....	19
3.4	MIDDLEWARE MODULAR	20
3.5	RESUMO DO CAPÍTULO.....	21
4	METODOLOGIA	22
4.1	PLANEJAMENTO DE EXECUÇÃO	22
4.2	IDEALIZAÇÃO DO SISTEMA	23
4.2.1	Apresentação dos Requisitos do Sistema	24
4.3	IDEIA GERAL E ESCOLHA DE PLATAFORMAS	24
5	DESENVOLVIMENTO PRÁTICO E RESULTADOS	28
5.1	ESTUDOS DOS EQUIPAMENTOS	28
5.1.1	Medidor Inteligente Landisgyr	28
5.1.2	Medidor Inteligente Eletra	31
5.1.3	Inversor Híbrido NHS	34
5.2	DESENVOLVIMENTO DE <i>SOFTWARE</i> E PROGRAMAÇÃO	37
5.2.1	<i>Softwares</i> Agregadores	37
5.2.2	Grafana	38
5.2.3	Controle de Controle do Inversor	40
5.2.4	Node-RED	42
5.2.4.1	Controle Automático via Algoritmo Externo Modo API.....	42

5.2.4.2	Controle Manual via Dashboard.....	43
5.3	<i>HARDWARE</i> UTILIZADO	45
5.4	INSTALAÇÕES DO SISTEMA.....	45
5.4.1	Instalação no NUPEDEE	45
5.4.2	Instalação no <i>Carport</i>	47
5.4.3	Instalação no Container COPEL	48
5.5	NOTAS FINAIS DO CAPITULO	49
6	CONCLUSÃO	50
	REFERÊNCIAS	51

1 INTRODUÇÃO

Mesmo após mais de 120 anos de existência a forma como os sistemas de energia elétrica são estruturados continua bastante similar, com a geração de energia ocorrendo longe de áreas residenciais, sendo transmitida e convertida antes de chegar no consumidor final. Para fazer um paralelo, é possível citar alguns outros serviços oferecidos aos consumidores, como na telefonia celular, onde o cliente é visto apenas como o próprio nome sugere: um utilizador, um dependente – alguém que apenas consome recursos e não é capaz de retribuir o mesmo serviço.

Nos sistemas de distribuição de energia elétrica, o padrão também costuma ser este, o consumidor é visto como apenas outra carga na rede, a qual a concessionária deve ser capaz de lidar. Recentemente presenciou-se o surgimento de um novo paradigma para nossos sistemas de distribuição de energia elétrica, onde a mesma deixa de ser tratada como um bem unidirecional para o consumidor mediano, e começa a vê-lo não mais como um ente a ser atendido, mas sim como uma potencial fonte de fornecimento de energia. Este fato deve-se à crescente popularização dos carros elétricos, armazenamento residencial de energia em baterias, por exemplo, e também às fontes de energias renováveis residenciais, como os módulos fotovoltaicos. Considerando-se a rápida difusão destes elementos e o novo paradigma dos sistemas de energia, percebe-se a necessidade de novas ferramentas que contribuam com a integração dos prosumidores e recursos energéticos distribuídos (REDs).

O desenvolvimento desta dissertação deve-se à importância de agregar os recursos de carga, geração e armazenamento presentes no lado do consumidor ou atrás do medidor, termos conhecido em inglês como *behind the meter* à operação das redes de distribuição. O trabalho se propõe a analisar os aspectos acadêmicos e práticos associados ao desenvolvimento de uma plataforma integradora e deixa como legado um framework integrador que pode servir de base para diferentes aplicações nas redes elétricas inteligentes e é elemento chave para quaisquer ações de resposta a demanda ou *smart contracts* que envolvam múltiplos REDs.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

Na última década, em especial no Brasil, os sistemas de distribuição têm registrado alguns avanços em direção à maior digitalização, descentralização e descarbonização, termo conhecido como 3D's. Entretanto, é a partir dos últimos cinco anos que se verificam ações maiores e mais efetivas por parte das distribuidoras, órgão regulamentador e consumidores.

Podem-se citar a instalação de medidores inteligentes, alterações na resolução normativa RN 482/2012 que trata da micro e minigeração, a qual sofreu mudanças presentes na RN 687/2015, lançamento de projetos estratégicos sobre armazenamento e mobilidade elétrica pela Agência Nacional de Energia Elétrica (ANEEL) bem como crescimento da geração distribuída em clientes de baixa tensão e média tensão, especialmente devido à geração solar fotovoltaica.

Em termos tecnológicos é possível enumerar uma série de importantes mudanças e inovações que começaram a disseminar nos sistemas de distribuição, como a mobilidade elétrica, introdução de inversores híbridos, resposta à demanda, dispositivos de internet das coisas IoT, tarifa branca e a aplicação de tarifas em tempo real em muitos países. Toda essa evolução é decorrência da transição energética pela qual o mundo está passando, visando maior eficiência, redução das perdas e substituição da geração a partir de combustíveis fósseis por energias produzidas a partir de fontes renováveis.

A transição energética traz consigo desafios e mudanças profundas na maneira como os agentes se relacionam entre si e com o sistema elétrico, exigindo ferramentas que oportunizem maximizar os ganhos das interações do prossumidor com os operadores do sistema de distribuição e vice-versa. No entanto, há muitas barreiras atuais devido à necessidade de uma série de padronizações, protocolos e até mesmo normativas que possam garantir a atuação confiável dos diferentes dispositivos e respostas que atendam aos contratos estabelecidos.

Esses elementos todos contribuíram para motivar e justificar o desenvolvimento de uma dissertação focada na elaboração de um framework que integre os REDs e ao mesmo tempo aponte uma série de desafios a serem superados para que os avanços possam ser sentidos o mais breve possível e possa, inclusive, colaborar para a elaboração de *sandboxes* regulatórios para o gerenciamento inteligente de REDs.

1.2 PRINCIPAIS CONTRIBUIÇÕES

Um sistema funcional que pode ser utilizado como disponibilizado ou servir de alicerce/inspiração para a construção de um similar mais avançado.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver um *middleware* capaz de integrar dispositivos distintos que compõem uma rede elétrica inteligente, visando proporcionar ações de gerenciamento inteligente para REDs.

1.3.2 Objetivos Específicos

Estudar dispositivos já existentes no mercado dentro do interesse da pesquisa, como medidores inteligentes de energia e inversores híbridos.

Estudar maneiras de como extrair os dados de dispositivos distintos: análise e estudo de protocolos em manuais técnicos, normas e aplicativos

Desenvolver metodologia para especificar (definir requisitos e restrições) uma plataforma (*hardware* + *software*) que atenda aos requisitos necessários para integrar dados dos dispositivos estudados, além de permitir possíveis expansões futuras.

Elaborar o desenvolvimento prático para implementar requisitos definidos e estudar os resultados.

1.4 DISTRIBUIÇÃO DO ASSUNTO

Este trabalho está dividido entre quatro capítulos de desenvolvimento, tratando inicialmente da fundamentação teórica, onde serão introduzidos alguns termos e conceitos utilizados durante o restante do trabalho; uma revisão bibliográfica de sistemas similares já publicados e testados; a descrição da metodologia utilizada para o desenvolvimento; a descrição do que foi desenvolvido na prática, incluindo programação, desenvolvimento e instalação de um sistema protótipo; e a análise de resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão contextualizados os principais conceitos e ferramentas utilizadas durante o desenvolvimento do trabalho, servindo como um apoio ao leitor, importantes para o entendimento dos capítulos subsequentes.

2.1 RECURSOS ENERGÉTICOS DISTRIBUÍDOS (REDS)

Os recursos energéticos distribuídos (REDS) são as gerações conectadas em redes de baixa ou média tensão, como sistemas de geração fotovoltaica, microturbinas, baterias entre outros (AKOREDE, 2010). Os REDs são os elementos chaves deste trabalho, visto que o objetivo geral é buscar integrá-los a outros sistemas inteligentes.

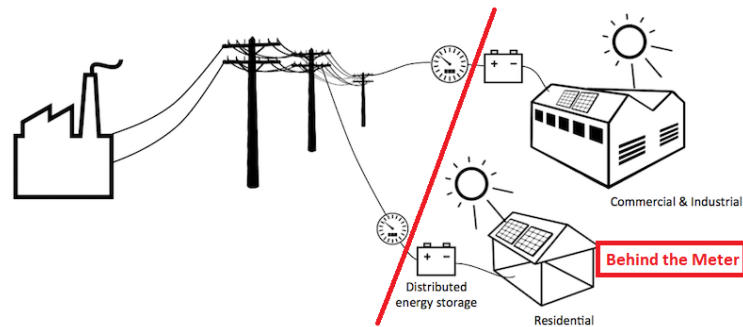
Os REDs utilizados e estudados neste trabalho são a geração solar fotovoltaica e o armazenamento em baterias (lítio, chumbo-ácido) através do inversor híbrido, os quais são gerenciados dentro de uma residência com medidor elétrico inteligente – gerenciamento o qual exige implementações em *hardware* e *software* específicas para cada equipamento, a fim de possibilitar o envio de informações e comandos para e com os mesmos.

2.2 MEDIDORES INTELIGENTES E CONSUMIDOR

Os medidores inteligentes são a classe de medidores de energia eletrônicos capazes de, além de realizar a medição precisa do consumo, receber e enviar dados através de algum meio de comunicação pré-definido, bem como armazenar localmente dados referentes a falhas e relatório de operações. É comum também entre os medidores inteligentes a medição do fluxo reverso de energia, que parte do consumidor em direção à rede da concessionária. Estas funcionalidades permitem que se realize medições remotas do consumo mensal, ações de corte e religação em caso de inadimplência e outros fatores, ações de resposta à demanda e coleta mais avançada de dados referentes ao perfil de consumo do usuário.

No momento em que se começa a realizar a medição e monitoramento também do fluxo reverso, é possível e necessário, para o fornecedor de energia, conhecer o que existe por trás do medidor, conceito conhecido em inglês como *behind-the-meter*, utilizado para descrever os recursos energéticos distribuídos que ficam instalados dentro da propriedade do consumidor, conectados após o medidor (no sentido concessionária – consumidor).

Figura 1 - Ilustração identificando o que está por trás do medidor



Fonte: Adaptado de Maluf (2016).

O posicionamento de DERs atrás do medidor implica em benefícios econômicos para o consumidor, mas também causa um impacto na rede de distribuição. Da mesma maneira em que consumidores gerando sua própria energia e injetando o sobressalente na rede de distribuição pode auxiliar em fatores como regulação de frequência e balanceamento de cargas, também podem ocorrer casos em que os transformadores da rede de média tensão estariam subdimensionados para o fluxo reverso, causando sobrecarga e falhas graves na rede.

2.3 INVERSORES CONVENCIONAIS E HÍBRIDOS

O inversor é um equipamento capaz de converter tensão contínua em tensão alternada, operação que pode ser implementada de várias maneiras eletronicamente e serve um papel importantíssimo no meio das energias renováveis, permitindo a conexão de baterias, painéis solares e outros recursos energéticos de corrente contínua à rede elétrica (PAUL, KAHN, 2013).

O conceito de um inversor híbrido surge quando o integramos a um sistema de baterias, permitindo que as mesmas sejam carregadas pelo recurso de energia distribuída (fotovoltaico, por exemplo) e também pela rede de distribuição. Este mesmo inversor permitirá descarregá-las para suprir o fornecimento de energia a cargas importantes durante períodos de queda, bem como fornecer energia à rede de distribuição, seja por vantagem econômica, programa de resposta à demanda ou outro motivo (BELLINASSO, MICHELS, 2016).

2.4 PROTOCOLOS *MODBUS* E *SUNSPEC*

O protocolo *MODBUS* foi criado para facilitar a comunicação de equipamentos supervisórios e de automação, com foco em ser simples e independente de fabricante. Conforme

descrito em DUTERTRE (2008), a comunicação entre dispositivos acontece, em sua maioria, de forma mestre-escravo, com o mestre enviando uma requisição e aguardando o retorno dos dados. Os comandos da comunicação são relativos à escrita ou leitura de dados, os quais são armazenados em registradores, divididos em quatro tipos *Coils* com apenas 1 bit que aceitam tanto leitura quanto gravação; *Discrete Inputs*, também com 1 bit mas apenas para leitura; *Holding Registers*, com 16 bits que aceitam leitura e gravação; e *Input Registers*, com 16 bits apenas para leitura. Os registradores são endereçados com um código de 16 bits, sendo que não é necessário para o dispositivo implementar acesso a todos endereços possíveis (DUTERTRE, 2008). Neste trabalho, a variante *MODBUS/TCP* (SWALES, 1999) será a utilizada, devido à natureza dos equipamentos estudados, mudando apenas o meio pela qual a comunicação acontece.

Para aproveitar a facilidade de implementação dentro de um protocolo *MODBUS*, foi criado a especificação chamada *Sunspec*, que consiste em padrões que determinam para qual função certas faixas de valores serão utilizadas dentro de dispositivos em sistemas de energia renovável (RANDLE, 2013).

Ao contrário de outras especificações que tentaram criar padrões para inversores e permitiam apenas o monitoramento dos mesmos, a *Sunspec* também inclui em seu modelo de dados operações que permitem controlá-los de forma programada, agendada ou autônoma (RANDLE, 2013).

2.5 CONCLUSÃO DO CAPÍTULO

Os conceitos introduzidos neste capítulo, apesar de apenas brevemente mencionados na introdução do trabalho, são de suma importância para o bom entendimento dos próximos, onde também serão expandidas as explicações sobre os mesmos.

3 REVISÃO BIBLIOGRÁFICA

Buscando uma contextualização melhor da aplicação de alguns conceitos e elementos que se pretende trabalhar neste projeto, foram buscadas publicações referentes a estudos dos mesmos.

3.1 AÇÕES DE GERENCIAMENTO PELO LADO DA DEMANDA, RESPOSTA A DEMANDA, GERENCIAMENTO DE REDS

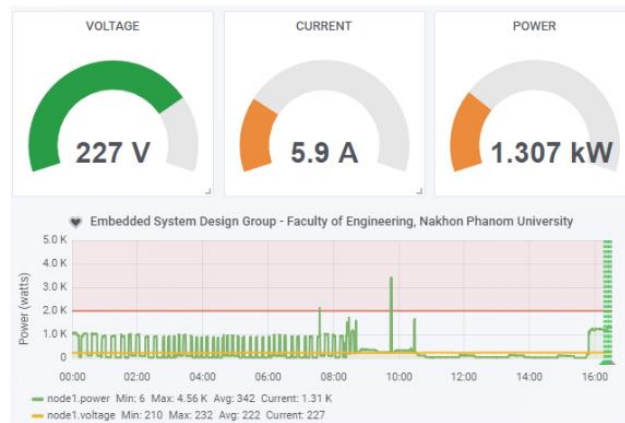
Com o surgimento de novas tecnologias em recursos energéticos, além da evolução e redução de preço das mesmas, painéis fotovoltaicos, turbinas eólicas, microturbinas, baterias e outros geradores são inseridos cada vez mais na rede de distribuição (JIAYI, CHUANWEN, RONG, 2008). Segundo Dominguez-Garcia e Hadjicostis (2011), se dedicarmos esforços ao nível de distribuição, correta coordenação e controle de cargas e recursos energéticos distribuídos, conseguimos providenciar uma maior flexibilidade aos serviços auxiliares, resultando em uma maior eficiência e confiabilidade do sistema como um todo – o controle da carga pode ser feito através de contratos do consumidor com o fornecedor, que pode providenciar preços mais acessíveis em troca deste controle da demanda (controle por resposta a demanda).

No trabalho de Cunha (2016), desenvolveu-se uma metodologia capaz de avaliar a viabilidade técnica e econômica de gerenciamento pelo lado da demanda, levando em conta a tarifa branca e REDs inseridos em baixa tensão. Apesar de na época, 2016, a tarifa horária ainda não ter sido implementada, os estudos mostraram que a união do gerenciamento pelo lado da demanda com REDs incentivaria e viabilizaria o uso da tarifa branca por parte dos consumidores, a fim de reduzir seus gastos com energia elétrica.

3.2 INFLUX E GRAFANA

Em um artigo publicado por Chooruang e Meekul (2018) na *International Conference on ICT and Knowledge Engineering (ICT&KE)* foi apresentada uma solução para um sistema de monitoramento de energia em tempo real, utilizando-se sensores e microcontroladores de baixo custo, os quais enviavam os dados (corrente, tensão, potência ativa e consumo cumulativo) para uma base de dados *Influx*, estes dados, posteriormente, foram apresentados em formas de gráficos em dashboards criados no *software Grafana*.

Figura 2 - Dashboard Desenvolvido no Sistema de Monitoramento



Fonte: Chooruang e Meekul (2018), p. 3.

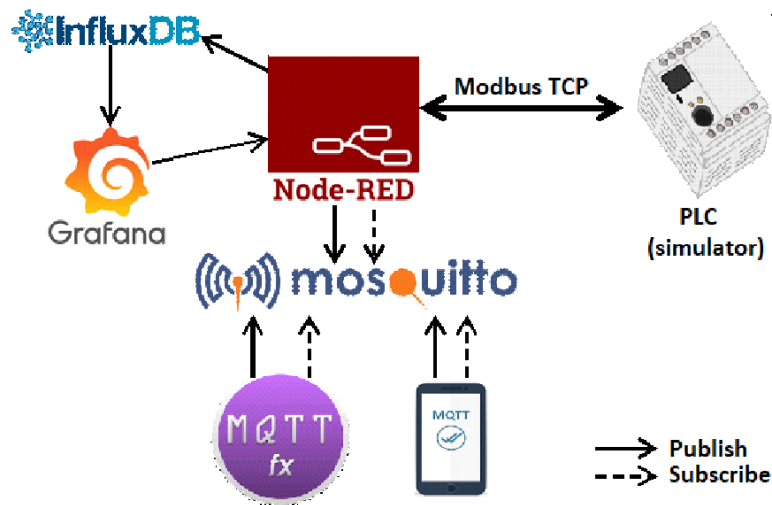
Este trabalho mostrou que o *Influx* é uma boa opção para armazenar dados temporais em grande quantidade. Os resultados dos gráficos criados com o *Grafana* também atendem bem os requisitos necessários neste trabalho.

3.3 NODE-RED

O Node-RED é uma ferramenta de programação baseada em fluxos, composta de uma parte que roda em segundo plano em um servidor e outra parte, interface gráfica, acessível via navegadores web (OpenJS Foundation & Contributors, 2021). A ferramenta, como interface de programação, permite executar uma diversidade muito grande de tarefas, desde o desenvolvimento de páginas para visualização de dados bem como a integração de dispositivos de *hardware* e interfaces de programação de aplicativos (APIs) (BLACKSTOCK, 2014), neste trabalho, a segunda funcionalidade foi a mais explorada.

O trabalho de Nițulescu e Korodi (2020), *Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions*, faz um estudo visando implementar um sistema supervisorio de aquisição de dados utilizando o Node-RED, notando sua rápida evolução para atender os mais novos desafios da (*IoT*). Um dos pontos que chama bastante atenção, similar ao deste trabalho, é o fato de utilizar a comunicação via *Modbus TCP* dentro do Node-RED, mas para comunicação com CLPs, ao contrário deste trabalho que se comunica com um inversor utilizando o mesmo protocolo.

Figura 3 - Arquitetura do Supervisório de Nițulescu e Korodi



Fonte: Nițulescu e Korodi (2020), p. 6.

Outro ponto de interesse é o fato da base de dados *InfluxDB* e o visualizador de dados *Grafana* aparecerem novamente, dessa vez com o próprio *Node-RED* fazendo a coleta dos dados via *Modbus TCP* e inserindo no *InfluxDB*, em contraste com o trabalho atual que a coleta acontece em um *software* próprio.

3.4 MIDDLEWARE MODULAR

A grande variedade de fabricantes, dispositivos, protocolos, meios de comunicação e dentro de uma rede inteligente apresenta-se como um desafio na hora de integrá-la. A comunicação entre dispositivos sendo um dos principais requisitos de uma rede inteligente exige que soluções sejam criadas.

A solução proposta, na maioria dos casos, é a criação de um *Middleware*, que pode ser definido como um *software* que fica entre as camadas de rede mais baixas (dispositivos de *hardware*) e as camadas mais altas em nível de *software* (os aplicativos), tendo como sua principal funcionalidade a abstração das dessemelhanças de dispositivos em uma aplicação distribuída, fornecendo uma maneira transparente e única para que aplicativos os acessem, sem conhecer as particularidades de cada objeto integrante (RODRÍGUEZ-MOLINA, 2016).

Ainda seguindo Rodríguez-Molina (2016), um *middleware* age como a cola que une diversas entidades de *software* e *hardware*, processando informações de maneira a esconder do usuário a complexidade intrínseca de dispositivos utilizados em sistemas de geração distribuídos.

3.5 RESUMO DO CAPÍTULO

Neste capítulo foi apresentada a revisão bibliográfica buscando posicionar o tema da dissertação em relação a publicações que tratam do tema da integração dos REDs com os sistemas de gerenciamento. Com a validação dos *softwares* e conceitos citados nas referências bibliográficas pode-se mostrar a consistência da ferramenta proposta e a importância do tema proposto.

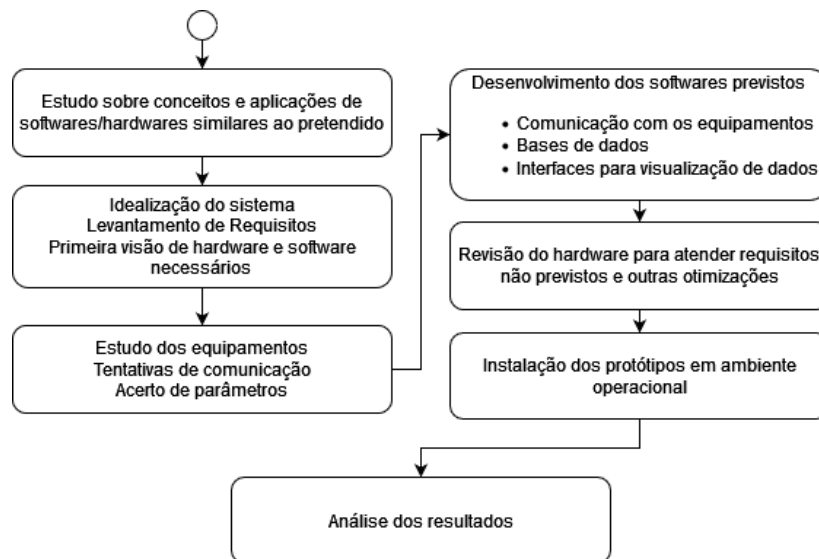
4 METODOLOGIA

Neste capítulo será apresentada a metodologia usada para o desenvolvimento do trabalho, em ordem cronológica. O conteúdo deste capítulo busca organizar o conteúdo de forma coerente, explicando o planejamento e fluxo de ideias durante sua idealização.

4.1 PLANEJAMENTO DE EXECUÇÃO

A fim de facilitar a visualização do que deveria ser implementado e poder auxiliar os leitores quanto ao processo de implementação, a Figura 4 ilustra, resumidamente, o fluxo pretendido para a realização dos próximos passos no projeto.

Figura 4 - Fluxo e planejamento de Execução



Fonte: Autor.

Primeiramente, realizou-se um estudo sobre aplicações similares ao que se pretende fazer neste trabalho, a fim de levantar novas possibilidades de *softwares* a serem utilizados e também confirmar o funcionamento dos softwares já conhecidos. Com o conhecimento do que já é possível e o do que se pode estender, idealiza-se o sistema, primeiramente sem pensar muito em quais ferramentas serão utilizados, mas logo na sequência, com esta primeira visão pronta, já é possível ter uma ideia inicial do que será necessário implementar em *software* e quais serão os *hardwares* necessários.

De maneira a validar se a primeira escolha de recursos foi válida, realiza-se um estudo mais aprofundado nos equipamentos que serão utilizados (medidores, inversores), já implementado pequenos pedaços de *software* para validar a possibilidade de comunicação e evitar problemas futuros – neste passo, também existe a possibilidade de suporte e ajuste de parâmetros, caso necessário, com o fabricante dos equipamentos. Após esta análise, caso a escolha inicial dos recursos não tenha sido adequada, será preciso reavaliá-las, confrontando agora os pontos que a tornaram inadequada.

Com a primeira versão dos *softwares* em mãos, por assim dizer, parte-se para o desenvolvimento e unificação das ferramentas desenvolvidas, a fim de criar um serviço robusto que possa fazer a coleta de dados e envio de comandos para os equipamentos de forma confiável. Cria-se uma base de dados onde serão armazenadas as informações coletadas e, já dentro do mesmo *software*, se implementa a comunicação com o mesmo. Por fim, neste passo, em paralelo se desenvolvem dashboards que irão possibilitar a visualização facilitada dos dados históricos e em tempo real dos equipamentos. O que foi desenvolvido até aqui, é o *middleware*, mencionado já em outros pontos do trabalho.

O *middleware* já implementado permite que sejam realizadas revisões no que foi pretendido em *hardware* para repensar ideias e se certificar de que nenhum ponto necessário nos objetivos específicos e requisitos previamente levantados ficou de fora.

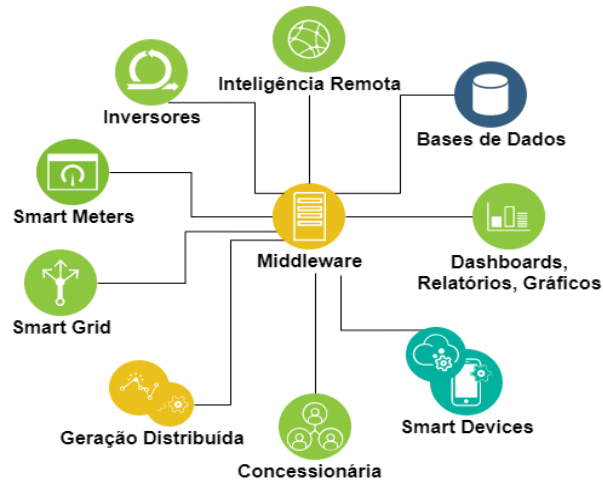
Dentro do possível, ambientes operacionais com alguns dos equipamentos já estão funcionando, permitindo que seja feita a implementação real do protótipo desenvolvido até o momento.

Por fim, com o protótipo já instalado e realizando comunicação em duas vias, parte-se para a análise dos resultados de tudo que foi implementado.

4.2 IDEALIZAÇÃO DO SISTEMA

O sistema busca unir inúmeras e distintas fontes de dados de recursos energéticos distribuídos. O sistema é chamado de *middleware*, pois, de certa maneira, se posiciona ao centro de tudo, recebendo e enviando dados coerentes para cada dispositivo, mas permitindo uma interação com os mesmos de forma transparente e única para o usuário final.

Figura 5 - Idealização do Middleware



Fonte: Autor.

4.2.1 Apresentação dos Requisitos do Sistema

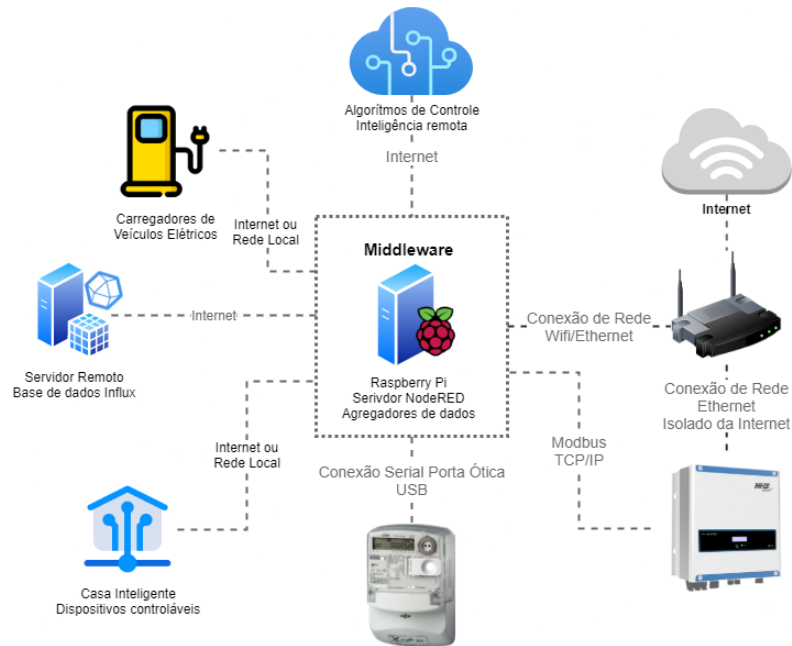
O primeiro medidor inteligente cedido para testes e aprendizado foi o Landis Gyr E34A, com interface de comunicação serial RS232 via porta óptica com cabo magnético e suporte ao protocolo de comunicação ABNT NBR 14522.

4.3 IDEIA GERAL E ESCOLHA DE PLATAFORMAS

Utilizando preferencialmente *softwares* de código aberto e programando novas soluções, desenvolveu-se neste trabalho um *middleware* capaz de fazer a interface com vários tipos de dispositivos que integram uma *smart home*.

As escolhas de mais importantes durante a concepção da plataforma já implementada envolviam principalmente qual seria a base de dados para armazenar o grande fluxo de informações e qual *hardware* de baixo custo, mas que não comprometeria a usabilidade, poderia ficar acoplado aos equipamentos.

Figura 6 - Middleware e suas conexões com dispositivos de uma smart home



Fonte: Autor.

A base de dados *InfluxDB* foi a escolhida por já estar consolidada no mercado, utilizada em vários sistemas de coletas de dados, sua infraestrutura de armazenamento temporal e algoritmos para otimização de consultas por tempo se mostra ideal para o projeto.

O *hardware* embarcado *Raspberry pi* foi o escolhido devido a sua disponibilidade no ambiente de pesquisa, oferece um sistema operacional Linux embarcado (ARM) sem grandes diferenças para uma versão desktop, facilitando o desenvolvimento de aplicações sem implicar em novos conhecimentos de programação. Possui também portas USB e interface serial RS232 em *hardware* para conexão nos equipamentos, ethernet e WiFi 2.4Ghz para conexão com a rede. Seu sistema operacional fica instalado em um cartão de memória microSD, facilitando o upgrade em campo e reprodução para outros *Raspberry*. Por ser amplamente conhecido e utilizado, existem muitas opções de acessórios e bibliotecas de código disponíveis a fácil alcance.

Figura 7 - Raspberry pi 3 Model B+ utilizado no projeto



Fonte: Autor.

A base de dados pode ser instalada tanto no *hardware* embarcado quanto em um servidor externo, para saber qual a opção mais adequada, é necessário calcular o espaço necessário para o armazenamento dos dados durante o período de tempo especificado.

A seguinte equação foi utilizada para estudar o quanto de espaço seria necessário.

$$E_t = \frac{60 \cdot 24 \cdot t_a \cdot T_d \cdot f_d}{10^9}$$

Onde:

E_t = Espaço total, em gigabytes

t_a = tempo de armazenamento, em dias

T_d = tamanho de 1 pacote de dados, em bytes

f_d = frequência de envio dos dados (envios por minuto)

Em um cenário hipotético, estima-se que cada pacote de dados tenha, em média, 1024 bytes, com dados coletados a cada 20 segundos por um período de 365 dias, resultando em um total de aproximadamente 1,6Gb por ano, por equipamento. Em um *Raspberry pi* com cartão

de memória de 16Gb, seria possível, com segurança, manter dados de até 3 equipamentos durante 2 anos, tempo satisfatório para os requisitos do sistema. Apesar da possibilidade de se armazenar os dados de forma local no próprio *hardware* embarcado, se optou também por enviá-los a um servidor externo, proporcionando uma maior segurança e disponibilidade dos mesmos.

5 DESENVOLVIMENTO PRÁTICO E RESULTADOS

Neste capítulo serão abordados e descritos todos os processos que aconteceram durante o desenvolvimento do trabalho, desde o primeiro contato e estudo inicial dos equipamentos, o desenvolvimento dos códigos de programação e as instalações físicas do sistema.

5.1 ESTUDOS DOS EQUIPAMENTOS

Este Capítulo trata dos equipamentos que foram utilizados durante o projeto, sendo eles 2 medidores inteligentes e 1 inversor híbrido, cedidos via projetos de pesquisa que já estavam em andamento.

5.1.1 Medidor Inteligente LandisGyr

O primeiro medidor inteligente cedido para testes e aprendizado foi o Landis Gyr E34A, com interface de comunicação serial RS232 via porta óptica com cabo magnético e suporte ao protocolo de comunicação ABNT NBR 14522.

Figura 8 - Medidor LandisGyr E34A



Fonte: Adaptado de Material de divulgação comercial LandisGyr (2018).

Nas tentativas de comunicação com este medidor utilizando os meios descritos na norma técnica, não foi possível estabelecer uma comunicação. O medidor não parecia seguir exatamente o padrão especificado nas normas técnicas, impossibilitando a obtenção dos dados

pela porta óptica. Em contato com o fabricante, foi possível obter uma cópia do programa de comunicação oficial desenvolvido pela própria LandisGyr, este programa é chamado de PLAWIN.

Figura 9 - Software da LandisGyr para leitura e análise



Fonte: Autor.

Com este *software* foi possível realizar as primeiras leituras de dados do medidor e confirmar que o problema era apenas com o protocolo de comunicação e não com o cabo óptico magnético, porta óptica ou outro problema de *hardware*.

Em comunicação com o setor de suporte da empresa, não se conseguiu estabelecer contato com o setor que havia desenvolvido o mesmo para saber mais sobre o protocolo de comunicação e quais seriam os problemas do método utilizado inicialmente neste trabalho. Para prosseguir o projeto sem maiores atrasos, fez-se necessário a implementação de uma engenharia reversa no *software* e equipamento fornecidos para descobrir onde estava o problema ou desenvolver uma solução personalizada de captura dos dados.

Para agilizar o trabalho de engenharia reversa, foi utilizado um programa chamado “*Serial Port Monitor*” para analisar, de forma transparente, a comunicação que o medidor realizava com o *software* oficial da LandisGyr.

Após inicializado e configurado o programa de monitoramento da porta serial, o *software* da LandisGyr foi iniciado com o cabo conectado ao medidor, logo em seguida foram

realizadas as operações de leitura necessárias, mais precisamente o comando “leitura de valores instantâneos”.

Na saída do programa que monitora a porta serial foram observadas algumas similaridades ao protocolo especificado na norma ABNT, mas com alguns pacotes de dados no começo da mensagem que eram desconhecidos, explicando assim o porquê da impossibilidade de se comunicar com o medidor seguindo apenas o que foi especificado no protocolo padrão. Mudando parâmetros no programa oficial e enviando diferentes comandos, chegou-se à conclusão que os dados extra enviados tem a ver com a segurança do medidor (senha para acesso) e canais de acesso (para possibilitar a extensão de parâmetros em outros medidores, possivelmente).

Com os dados corretos em mãos, foi possível dar continuidade do desenvolvimento da aplicação que capturaria os dados do medidor de forma independente. Para o desenvolvimento deste aplicativo, foi escolhida a linguagem *Python*, que fornece maneiras rápidas e fáceis de trabalhar com comunicação via porta serial, bem como tratamento de dados.

Foi desenvolvido com sucesso um script em *Python* que replica os comandos do *software* oficial da LandisGyr para extrair os dados instantâneos do medidor e guardá-los em uma base de dados externa, tornando possível o processamento desses dados independentes do medidor, da empresa ou qualquer outra fonte externa.

Figura 10 - Exemplo de comunicação já com alguns dados decodificados (em vermelho, dados que batiam com os medidos externamente, apenas a Fase A havia sido conectada)

```

Enviando handshake...
bufferMedidor, 260 bytes:
0x21 0x1 0x44 0x85 0x21 0x15 0x14 0x2 0x29 0x6 0x18 0x6 0x15 0x14 0x0 0x29 0x6 0x18 0x14 0x54 0x26 0x28 0:
0x1 0x2 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x97 0xba 0x5 0x5
----
Enviando pedido de medicao...
bufferMedidor, 260 bytes:
0x14 0x1 0x44 0x85 0x21 0x15 0x14 0x3 0x29 0x6 0x18 0x66 0xe6 0x5b 0x43 0x66 0x66 0x66 0x3f 0x9a 0x99 0x9:
f 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xf:
----
DUMP dados decodificados para debug:
idComando: 0x14                                versaoMedidor: 0x57 |                                fatorPotenciaA: (1.0,)
hora: 0x15                                       indicadorConversaoGrandezas: 0x1                fatorPotenciaB: (0.0,)
minuto: 0x14                                     tensaoFaseA: (219.89999389648438,)              fatorPotenciaC: (0.0,)
segundo: 0x3                                     tensaoFaseB: (0.899999761581421,)              fatorPotenciaTrifasico: (1.0,)
dia: 0x29                                        tensaoFaseC: (0.3000001192092896,)              defasagemCorrenteFaseA: (nan,)
mes: 0x6                                         tensaoLinhaAB: (nan,)                          defasagemCorrenteFaseB: (nan,)
ano: 0x18                                       tensaoLinhaBC: (nan,)                          defasagemCorrenteFaseC: (nan,)
caracteristicaReativaA: 0x4c                   tensaoLinhaCA: (nan,)                          temperaturaInterna: (nan,)
caracteristicaReativaB: 0x4c                   correnteFaseA: (0.1899999976158142,)            frequenciaRede: (59.5,)
caracteristicaReativaC: 0x4c                   correnteFaseB: (0.03999999910593033,)          anguloFaseA: (0.0,)
caracteristicaReativaTrifasica: 0x4c          correnteFaseC: (0.029999999329447746,)          anguloFaseB: (-1.0,)
tipologacao: 0x0                                correnteNeutro: (0.0,)                          anguloFaseC: (-1.0,)
latitudeGraus: 0xff                             potenciaAtivaA: (17.0,)                         anguloEntreFasesAB: (nan,)
latitudeMinutos: 0xff                           potenciaAtivaB: (0.0,)                         anguloEntreFasesBC: (nan,)
latitudeSegundos: 0xff                          potenciaAtivaC: (0.0,)                         anguloEntreFasesCA: (nan,)
latitudeNorteSul: 0xff                          potenciaAtivaTrifasica: (17.0,)                 distorcaoHarmonicaTensaoFaseA: (nan,)
longitudeGraus: 0xff                             potenciaReativaA: (0.0,)                       distorcaoHarmonicaTensaoFaseB: (nan,)
longitudeMinutos: 0xff                           potenciaReativaB: (0.0,)                       distorcaoHarmonicaTensaoFaseC: (nan,)
longitudeSegundos: 0xff                          potenciaReativaC: (0.0,)                       distorcaoHarmonicaCorrenteFaseA: (nan,)
longitudeOesteLeste: 0xff                       potenciaReativaTrifasica: (0.0,)                distorcaoHarmonicaCorrenteFaseB: (nan,)
                                                    CRC: 0x1b 0x58

```

Fonte: Autor.

A partir deste ponto deu-se como concluída a tarefa de extrair, de forma independente, os dados do medidor inteligente LandisGyr E34A. O código não é específico para este medidor apenas, pois o *software* da LandisGyr abrange outros modelos. Acredita-se que não seja necessário a realização de modificações no *software* desenvolvido para suportar outros medidores da mesma marca, ou, caso se faça necessário, as modificações seriam simples (como troca do ID de dispositivo, senha, canal, etc.) e boa parte do código pode ser reaproveitado.

A fim de disponibilizar os dados do medidor para pesquisas futuras e outros possíveis algoritmos de controle, o código foi expandido para armazenar os dados extraídos em uma base de dados *Influx* e um servidor MQTT do grupo de pesquisa, o qual utiliza os dados para seus processos de automação residencial.

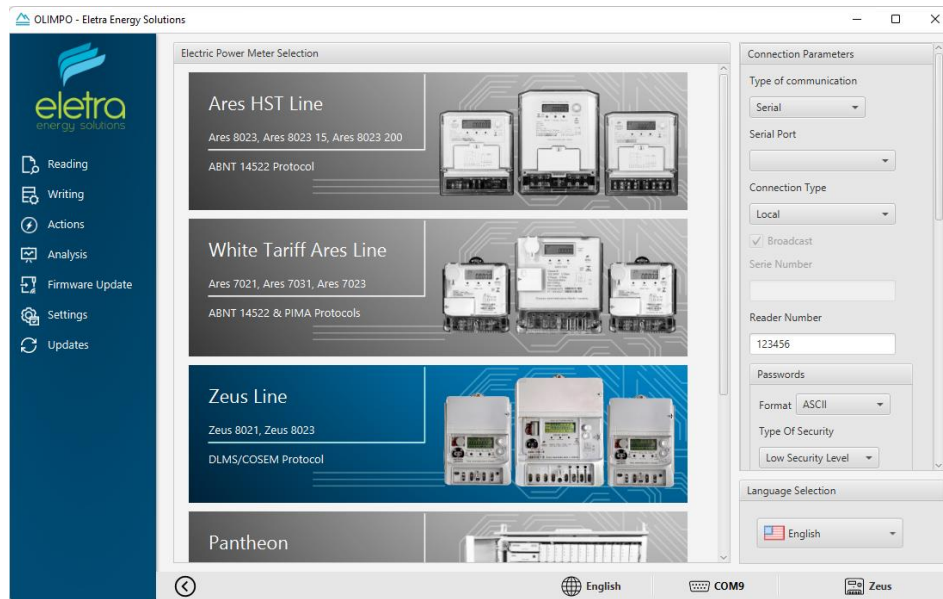
Presume-se que as dificuldades encontradas ao realizar este tipo de comunicação com o medidor se devem ao fato de que o mesmo não ter sido preparado para ser utilizado desta maneira (capturando dados em tempo real). A porta óptica serve para configuração do medidor por técnicos treinados da Landis ou para leitura de dados via agente da concessionária, previamente treinado e com *softwares* oficiais.

5.1.2 Medidor Inteligente Eletra

Dando sequência aos estudos de medidores inteligentes, foi possível também estudar medidores da empresa Eletra, modelo Zeus - em suas versões monofásica e trifásica.

Para acelerar o processo de desenvolvimento, antes de tentar fazer a interface com o medidor como aconteceu com o medidor da LandisGyr, foi realizado um contato inicial com a empresa, solicitando *softwares*, manuais técnicos, esclarecimento de algumas dúvidas e outras dicas, conforme disponibilidade. A empresa Eletra forneceu seu *software* OLIMPO, para que fosse possível extrair os dados do medidor via cabo óptico (ou porta RS232 integrada) sem grandes dificuldades.

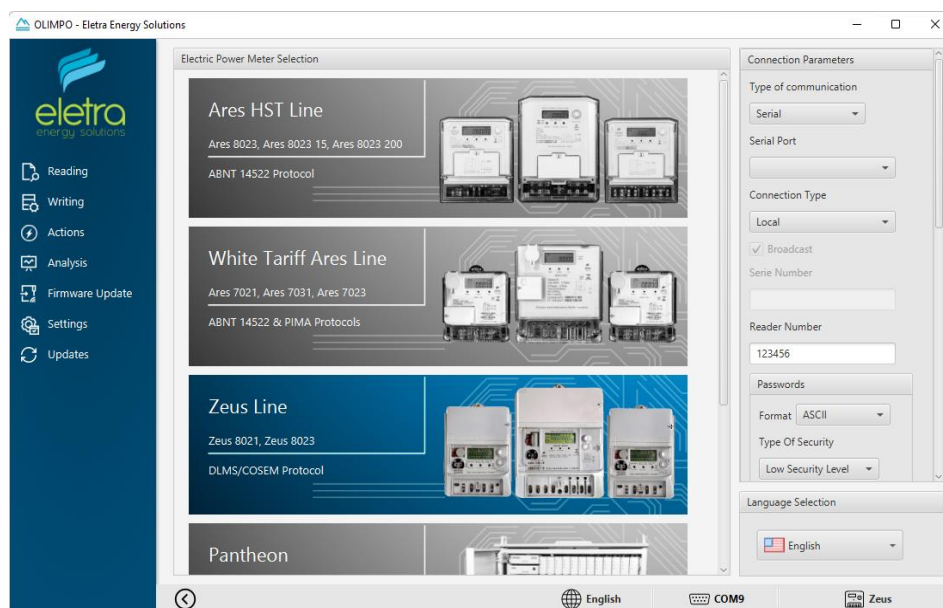
Figura 11 - Tela inicial do Software OLIMPO da Eletra



Fonte: Autor.

Após configurar o *software* com os parâmetros corretos (porta serial, senha), foi possível obter os dados dentro do mesmo, os quais batiam com os medidos externamente na bancada de testes. Com a confirmação de que o medidor estava apto a se comunicar pela porta óptica sem problemas, deu-se início a programação de um script em *Python* para fazer a leitura automatizada, a fim de replicar o processo desenvolvido com o medidor da LandisGyr.

Figura 12 - Dados Sendo Lidos no software da Eletra



Fonte: Autor.

Um ponto de destaque no *software* OLIMPO é a clareza com a qual os dados são apresentados, além de um relatório de comunicação serial detalhado, byte a byte. Isso possibilitou o fácil entendimento do protocolo, para que o mesmo fosse replicado, além de dar uma dica importante - na parte inferior da tela, onde se localiza o relatório, o programa menciona que a comunicação está sendo feita via protocolo DLMS/COSEM.

Figura 13 - Destaque dos Dados DLMS/COSEM no software OLIMPO

```
[Enviado][DLMS/COSEM] - 08/10/2020 16:35:23.887
Dado: [7E A0 19 03 03 32 EC C8 E6 E6 00 C0 01 C1 00 03 01 00 20 07 00 FF 02 00 85 83 7E]

[Recebido][DLMS/COSEM] - 08/10/2020 16:35:23.975
Dado: [7E A0 13 03 03 52 44 77 E6 E7 00 C4 01 01 00 12 08 8F 4B 48 7E]
```

Fonte: Autor.

Antes de prosseguir com o desenvolvimento similar ao que foi feito anteriormente (interceptar os dados durante a comunicação e replicá-los), buscou-se realizar a comunicação com o medidor via protocolo DLMS/COSEM, utilizando bibliotecas já prontas na linguagem *Python*, disponibilizadas gratuitamente em repositórios pessoais do *Github* (Palmlund Wahlgren Innovative Technology AB, 2021). Entretanto, mesmo após diversas tentativas, não foi possível iniciar uma comunicação ou extrair os dados desta maneira. O medidor da Eletra também possui um protocolo de segurança, onde antes de permitir a requisição de dados pelas portas seriais, se faz necessário o envio de uma senha para “desbloquear” a conexão. Nas bibliotecas utilizadas, este tipo de segurança não é mencionado nas documentações, indicando que este é o fator que impossibilitou a conexão.

Para dar sequência ao trabalho descartou-se esta possibilidade e foi utilizado um *software sniffer* (interceptador) da porta serial, assim como no medidor da LandisGyr, copiando-se a comunicação inicial para que o script automatizado enviasse uma cópia dos dados. Unindo este cabeçalho de segurança com os outros dados mostrados no próprio programa OLIMPO (e confirmados no *sniffer serial*), foi possível realizar a extração automática dos dados, assim como aconteceu anteriormente no medidor da LandisGyr.

Por fim, o mesmo código que enviava os dados do medidor da LandisGyr para uma base de dados *Influx* foi adicionado ao script que coleta os dados do medidor Eletra.


Caso mais algum medidor inteligente seja adquirido, o mesmo processo de desenvolvimento poderá ser utilizado, possibilitando uma análise transparente dos dados do mesmo, não importando para o usuário a forma como eles estão sendo extraídos, muito menos a marca/modelo.

5.1.3 Inversor Híbrido NHS

O outro objeto de estudo deste trabalho, podendo se chamar de principal, é um inversor híbrido. Este equipamento, fundamentalmente, deve ser capaz de permitir o fluxo de energia de um sistema de geração e armazenamento acontecer em múltiplos sentidos, diferentemente de um sistema de armazenamento convencional onde as baterias servem apenas para suprir uma queda de energia, em um sistema com inversor híbrido é possível descarregá-las para injetar energia na rede, bem como carregá-las a qualquer momento, não importando a fonte (energia solar, rede elétrica principal ou outra fonte de geração conectada).

O modelo de inversor que foi disponibilizado para o estudo foi desenvolvido pela NHS, empresa brasileira situada em Curitiba - PR, e não possui ainda um nome definido, pois ainda está em fase de testes e desenvolvimento. Este trabalho também visou contribuir com o desenvolvimento do equipamento.

Figura 14 - Ficha Técnica do Inversor Híbrido NHS de 3kW

Inversor Híbrido NHS SOLAR-3K-HSM1 	
Máx. FV - Potência do Gerador	3600 W
Máx. Tensão FV	490 Vdc
Isc FV	30 Adc
Corrente nom. FV	16 Adc
Range de Tensão - MPPT	120 - 480 Vdc
Tensão das baterias	192 Vdc
Tensão operação baterias	163 - 232 Vdc
Corrente máx. carga/desc. bateria	24,5 Adc
Potência máx. carga/desc. bateria	3000 W
Tipo de bateria	Chumbo-Ácido
Faixa de capacidade das baterias	17 - 220 Ah*
Tensão nom. da rede e saída aux.	220 Vac
Corrente nom. da rede e saída aux.	13,7 Aac
Corrente máx. da rede e saída aux.	14,5 Aac
Frequência nom. da rede e saída aux.	60 Hz
Pot. nom. aparente da rede e saída aux.	3000 VA
Pot. max. aparente da rede e saída aux.	3000 VA
Tipo de carga permitida na saída aux.	resistiva, indutiva, capacitiva e não linear.
Topologia	Não Isolado
Eficiência brasileira	93,08%
Range do fator de potência	Padrão >0,99 0,90 cap - 0,90 ind
Temperatura de operação	0 a 50°C
Grau de proteção (IP)	IP20

Fonte: Autor.

Figura 15 - Inversor NHS instalado em laboratório da UFSM com 2 bancos de bateria de chumbo-ácido



Fonte: Autor.

Este inversor foi instalado nos laboratórios do NUPEDDEE dentro da UFSM, acompanhado de um sistema com 10 painéis solares de 300W cada e 2 bancos de bateria, cada um com 16 baterias de chumbo-ácido 12V 70Ah, ligadas em série, totalizando 192V com 26kWh de capacidade. O inversor possui um algoritmo interno de funcionamento onde a prioridade é manter a carga auxiliar sempre alimentada, como um nobreak, além de sempre manter a bateria em 100% de carga, seja utilizando energia solar ou a energia da rede principal. Para comunicação e controle externo, o inversor disponibiliza uma interface *Modbus* sobre TCP/IP e/ou USB, já preparada para rodar sobre o protocolo *Sunspec*.

Figura 16 - Porção de uma das tabelas de dados no modelo Sunspec do inversor

TABLE 103 – INVERTER (THREE PHASE)							DESCRIÇÃO
DESCRIPTION	NAME	OFFSET	ADDR	SIZE	TYPE	RANGE	
Inverter (Three Phase)	TABLE103_ID		191	1	R	101 – 102 – 103	Referência da tabela selecionada de acordo com a quantidade de fases do equipamento
Length	TABLE103_L		192	1	R	50	Tamanho do pacote
Amps	TABLE103_A	0	193	1	R		Somatório das correntes de todas as fases
Amps PhaseA	TABLE103_AphA	1	194	1	R		Corrente na fase A
Amps PhaseB	TABLE103_AphB	2	195	1	R		Corrente na fase B
Amps PhaseC	TABLE103_AphC	3	196	1	R		Corrente na fase C
Amp Scale Factor	TABLE103_A_SF	4	197	1	R	-1	Fator de escala para valores de corrente
Phase Voltage AB	TABLE103_PPVphAB	5	198	1	R		Tensão entre as fases A e B
Phase Voltage BC	TABLE103_PPVphBC	6	199	1	R		Tensão entre as fases B e C
Phase Voltage CA	TABLE103_PPVphCA	7	200	1	R		Tensão entre as fases C e A
Phase Voltage AN	TABLE103_PhVphA	8	201	1	R		Tensão entre a fases A e Neutro
Phase Voltage BN	TABLE103_PhVphB	9	202	1	R		Tensão entre a fases B e Neutro
Phase Voltage CN	TABLE103_PhVphC	10	203	1	R		Tensão entre a fases C e Neutro
Voltage Scale Factor	TABLE103_V_SF	11	204	1	R	-1	Fator de escala para valores de tensão AC
Watts	TABLE103_W	12	205	1	R		Somatório das potências ativas de todas as fases
Watts Scale Factor	TABLE103_W_SF	13	206	1	R	0	Fator de escala para o valor de potência ativa
Hz	TABLE103_Hz	14	207	1	R		Frequência da rede elétrica
Hertz Scale Factor	TABLE103_Hz_SF	15	208	1	R	-1	Fator de escala para o valor de frequência
VA	TABLE103_VA	16	209	1	R		Potência aparente
VA Scale Factor	TABLE103_VA_SF	17	210	1	R	0	Fator de escala para o valor de potência aparente

Fonte: Adaptado do Manual Técnico do produto.

O desenvolvimento da integração com o inversor híbrido da NHS foi auxiliado pela própria empresa, que disponibilizou materiais técnicos, bibliotecas e funcionários para suporte. Os engenheiros responsáveis pelo desenvolvimento do inversor optaram pela comunicação utilizando o protocolo *Sunspec* sobre *Modbus*, que já define um padrão de parâmetros e configurações para controlar e monitorar sistemas de energia distribuídos (fonte). Através da análise dos códigos de exemplo fornecidos pela empresa, desenvolvidos em *Python*, foi possível desenvolver a integração sem grandes dificuldades. Inicialmente, o *software* apenas capturava os dados em tempo real da geração, consumo, baterias, etc. e os armazenava em uma base de dados *Influx*, a mesma utilizada para os medidores anteriormente, mas assim que os dados foram validados, o programa foi modificado para aceitar também a escrita de parâmetros, que visam controlar o fluxo da energia no inversor, como carregar ou descarregar manualmente as baterias, permitir ou impedir injeção de energia solar na rede, ligar e desligar a carga auxiliar, entre outros.

Figura 17 - Dados da geração solar (W) já armazenados no InfluxDB

```
> SELECT "DCW" FROM "inverter" LIMIT 15
name: inverter
time                DCW
----                -
1593019285000000000 1916
1593019513000000000 1852
1593019543000000000 1842
1593019574000000000 1842
1593019605000000000 1822
1593019635000000000 1804
1593019666000000000 1800
1593019697000000000 1819
1593019727000000000 1781
1593019758000000000 1795
1593019789000000000 1798
1593019819000000000 1825
1593019850000000000 1819
1593019881000000000 1812
1593019911000000000 1804
```

Fonte: Autor.

Agora com os dados em tempo real do inversor já disponíveis para acesso transparente, sem passar pelo inversor, bem como a possibilidade de comandá-lo remotamente, a integração com o mesmo se finalizou, possibilitando continuar com os objetivos do projeto.

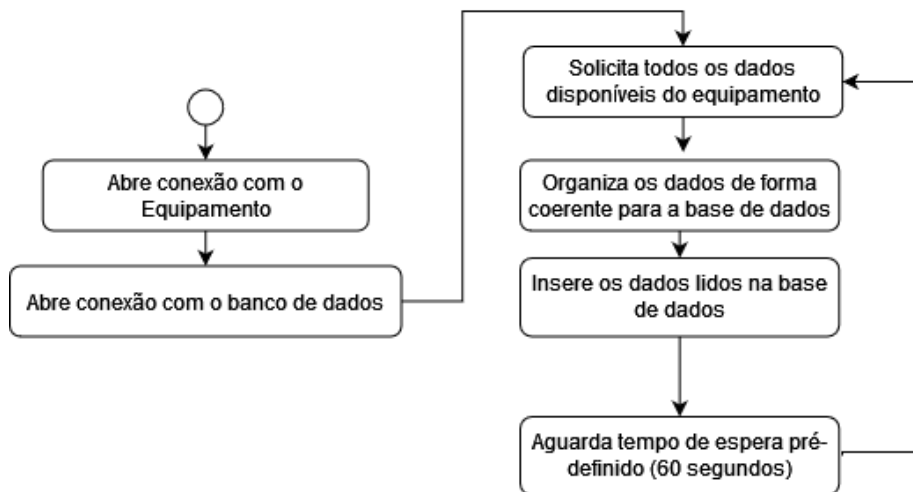
5.2 DESENVOLVIMENTO DE *SOFTWARE* E PROGRAMAÇÃO

Tendo em mente já as capacidades dos equipamentos (*hardware*), dá-se início ao desenvolvimento e/ou aprofundamento dos códigos e programas necessários, neste subcapítulo serão descritos os *softwares* utilizados e desenvolvidos.

5.2.1 *Softwares* Agregadores

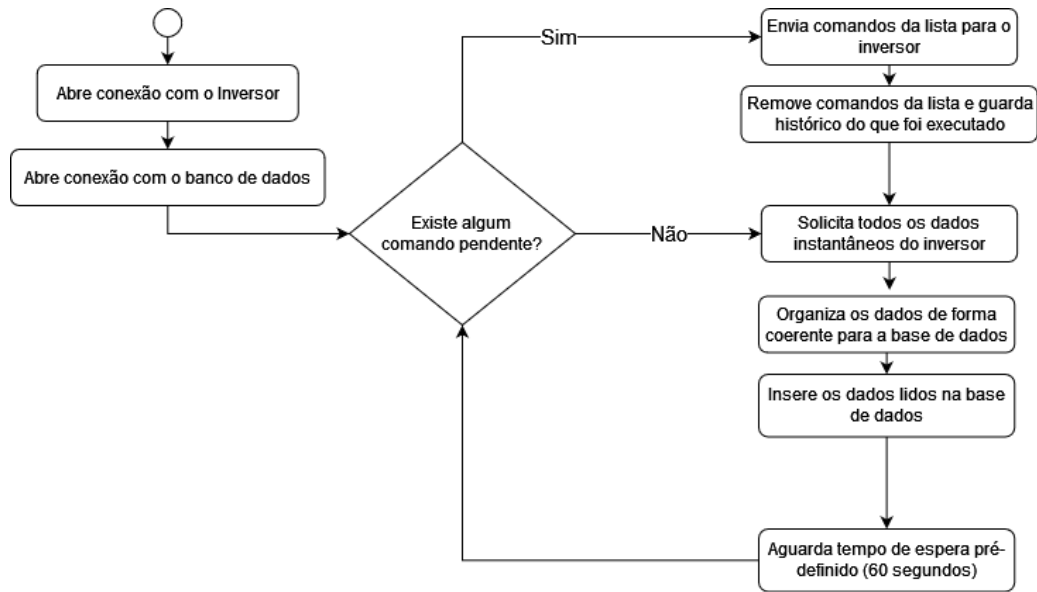
Os *softwares* desenvolvidos até o presente momento buscavam extrair os dados dos equipamentos que usam diferentes formas de comunicação, visando disponibilizá-los em uma única forma que pode ser acessada de maneira simples e transparente, sem precisar entender como cada um deles funciona. Estes *softwares* foram chamados de agregadores, pois visam agregar os dados de diferentes fontes em lugar apenas. Cada dispositivo tem seu próprio *software* agregador, mas todos possuem em comum a interface com a base de dados.

Figura 18 - Fluxograma de funcionamento geral do software agregador para equipamentos passivos (medidores inteligentes, sensores)



Fonte: Autor.

Figura 19 - Fluxograma de funcionamento do software agregador para o Inversor



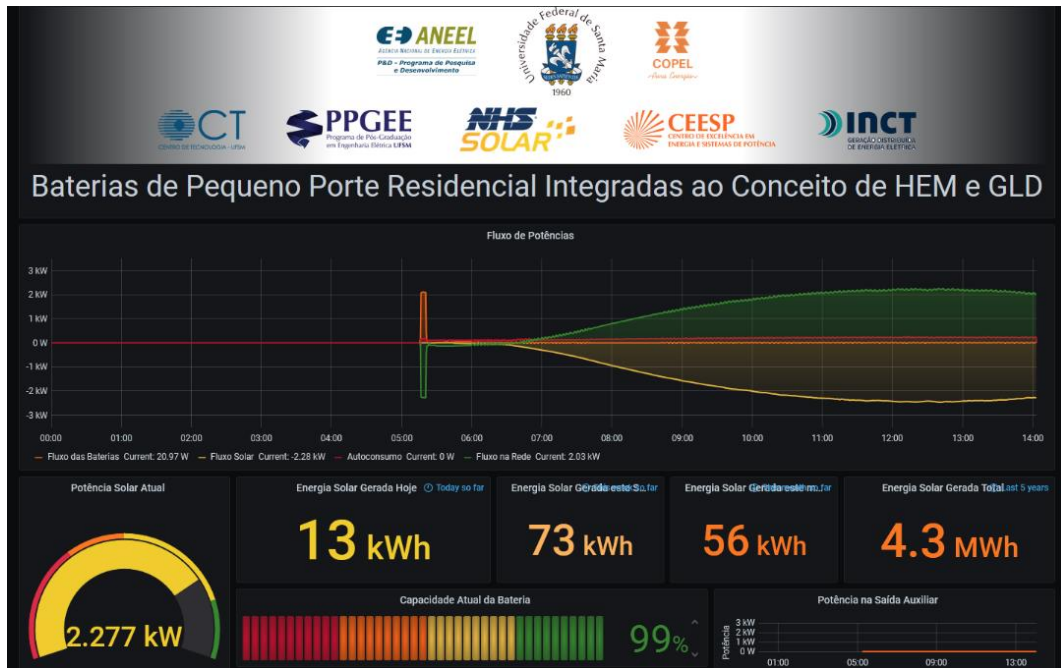
Fonte: Autor.

O *software* agregador desenvolvido para o inversor da NHS, entretanto, necessitava de um módulo extra, capaz de receber comandos externos e repassá-los ao inversor. A forma como o inversor recebe os comandos é através da escrita em certos endereços do protocolo *Sunspec* (registradores *Modbus*), os registradores que podem receber escrita, suas funções e valores aceitos foram especificados nas tabelas fornecidas pela empresa NHS. Este requisito foi implementado de maneira simples, através de um arquivo de texto contendo a fila de comandos a ser executada, um comando por linha (com seus respectivos parâmetros). Este arquivo, por sua vez, é monitorado pelo *software* agregador, que, ao detectar uma nova linha inserida executa o respectivo comando e o remove da fila. Este fluxo de funcionamento pode ser visualizado na figura.

5.2.2 Grafana

A quantidade de dados coletada nos equipamentos é consideravelmente alta - mais de 100 variáveis lidas a cada minuto, o que dificulta o acompanhamento dos mesmos em forma de tabela, além da barreira encontrada ao tentar ler os dados de uma base de dados sem interface gráfica. Para solucionar este problema, optou-se por implementar uma interface web onde os dados podem ser visualizados em forma de gráficos ou tabelas, com consultas manuais ou dashboards configuráveis. Nesta interface é possível também exportar os dados visualizados para outros formatos, como Excel, CSV ou JSON.

Figura 20 - Um dos dashboards criados no Grafana para visualização geral dos dados



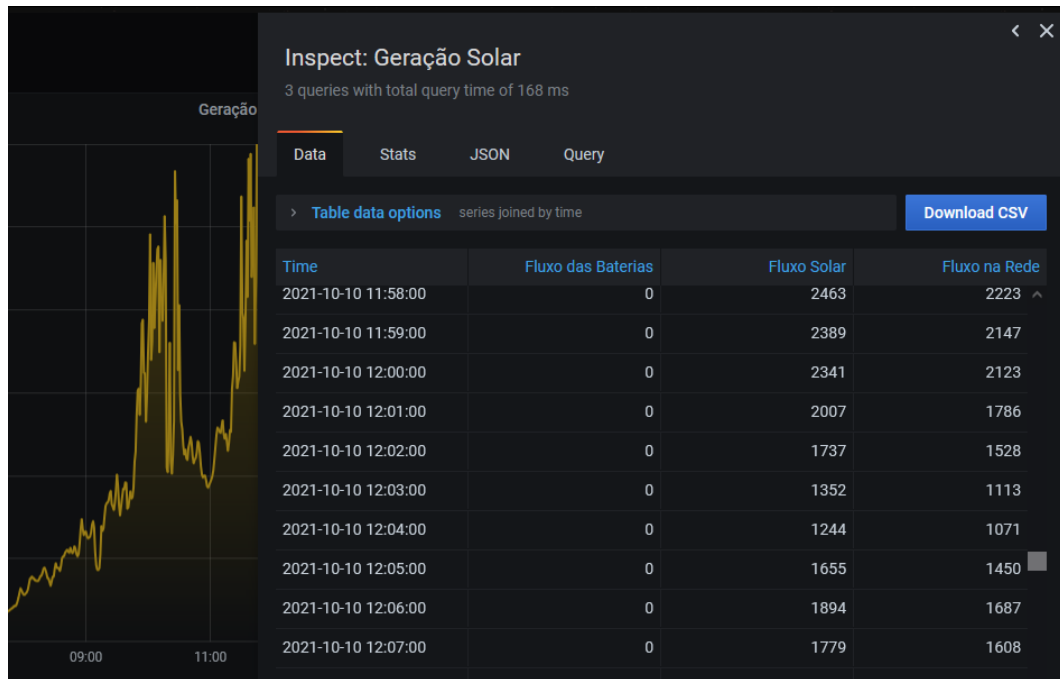
Fonte: Autor.

Figura 21 - Consulta personalizada de valores diretamente da base de dados (potência solar)



Fonte: Autor.

Figura 22 - Painel para visualização em tabela e exportação dos dados



Fonte: Autor.

5.2.3 Controle de controle do Inversor

Neste momento, com os dados disponíveis em tempo real e fácil visualização, o próximo passo foi testar os comandos de controle do inversor, a fim de validar o código de controle do *software* agregador bem como testar e estudar o comportamento do inversor. As variáveis de controles disponíveis para controle do inversor não são muitas, mas possibilitaram atender quase todos os requisitos de controle iniciais. Na Tabela 1 é possível visualizar a relação dos campos utilizados no controle:

Tabela 1 - Tabela de Comandos Sunspec Inversor NHS

Comando	Valores	Função
Habilitar Comandos Remotos	0 ou 1	0 = Comandos externos desativados. Neste modo o inversor opera com seu algoritmo interno que prioriza a carga auxiliar e bateria em 100% acima de tudo; 1 = Comandos Externos ativados. Modo de Gerenciamento de Energia. Neste modo o inversor vai atender o que for especificado nos comandos “Potência de Descarga” e “Profundidade de Descarga”
Profundidade de Descarga	0 a 100 (inteiros)	Representa qual o valor mínimo em de carga, em porcentagem, que a bateria pode atingir.
Potência de Carga ou Descarga	-1000 a 1000 (inteiros)	0 = Zera a corrente nas baterias, impedindo que elas carreguem ou descarreguem (o descarregamento intrínseco das baterias continuará, ela não será mantida) -1000 a -1 = Define a porcentagem da potência nominal (Valor * 0,1%) do banco que será utilizada para descarregá-lo, injetando energia na rede. Em nosso inversor de potência nominal 3000W, um valor de -400 representaria uma descarga da bateria em 1200W. Este valor será ignorado caso a bateria tenha atingido a profundidade especificada no registro anterior. 1 a 1000 = Define a porcentagem da potência nominal (Valor * 0,1%) do banco que será utilizada para carregá-lo, injetando energia na rede. Em nosso inversor de potência nominal 3000W, um valor de -400 representaria uma descarga da bateria em 1200W. Este valor será ignorado caso a bateria tenha atingido a capacidade máxima de carga (100%).

Fonte: Adaptado do manual técnico do fabricante.

O inversor possui uma tabela consideravelmente extensa de comandos, mas para atuação direta, os 3 comandos descritos na Tabela 1 são os disponíveis. Os outros comandos tratam da definição de parâmetros como capacidade e tipo da bateria, tensão de operação, limites de corrente de proteção, entre outros.

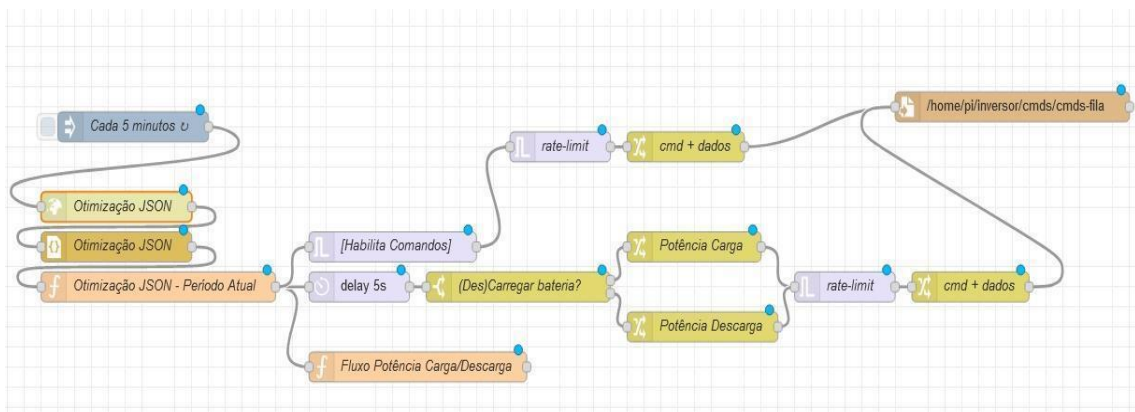
5.2.4 Node-RED

O Node-RED, por ser uma ferramenta com inúmeras possibilidades de conexão, irá permitir que os dados capturados pelos *softwares* agregadores e armazenados sejam lidos, pois bastará realizar a leitura dos mesmos diretamente da base de dados *Influx*. Da mesma maneira, também será possível enviar os comandos de controle para o *software* agregador através da simples escrita de um arquivo, bem como ler instruções de controle fornecidas por algoritmos externos, retransmitindo-as para o inversor.

5.2.4.1 Controle automático via algoritmo externo modo API

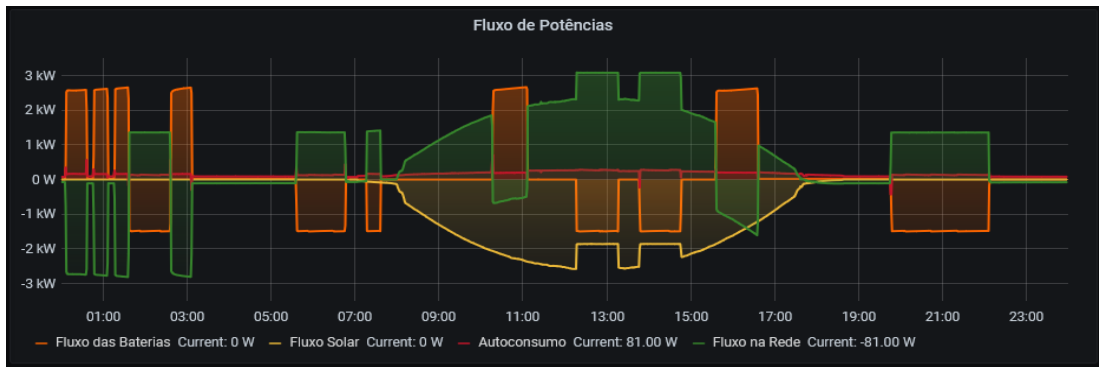
A primeira versão do controle automatizado foi feita via consulta a uma API externa, desenvolvida por um outro trabalho dentro do mesmo projeto. Esta API implementava o controle proposto no trabalho de AZEVEDO (2020), o qual analisava os dados da geração e consumo, disponibilizando saídas de controle para controlar o inversor.

Figura 23 - Fluxo do Node-RED para controle via API JSON



Fonte: Autor.

Figura 24 - Dados reais do Sistema com atuação do algoritmo externo



Fonte: Autor.

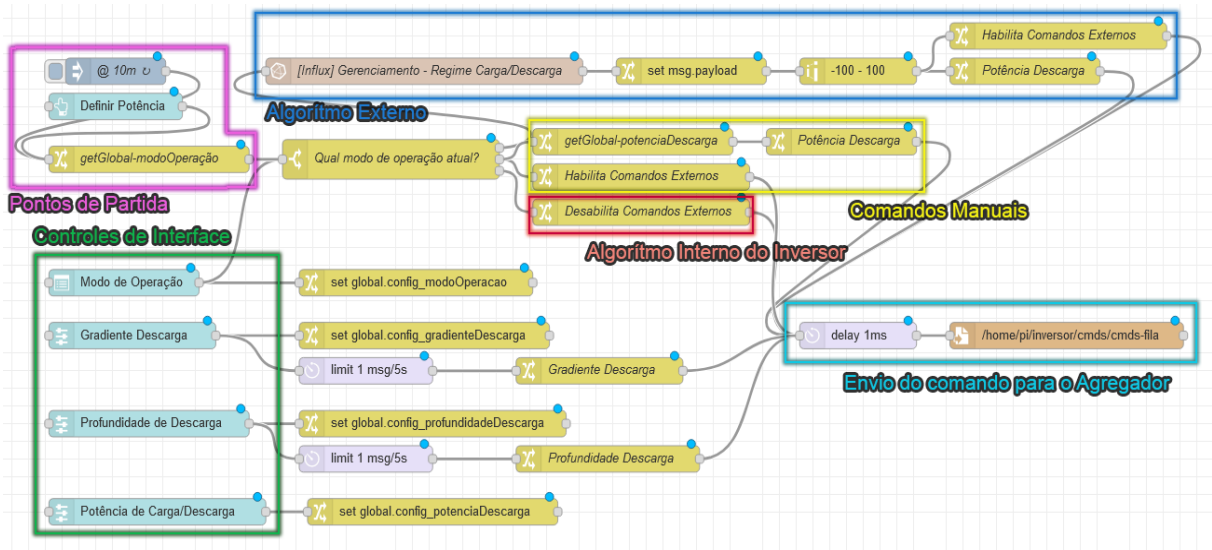
A Figura 23 mostra integralmente o fluxo criado no Node-RED para ler os dados da API JSON via requisição web, convertê-los para objeto, extrair os valores, adaptá-los aos parâmetros do inversor, gerar a linha do comando, e, por fim, escrever no arquivo de fila. O comando será executado pelo *software* agregador, conforme demonstrado na figura 19.

A Figura 24 demonstra as curvas reais do sistema durante 24 horas de atuação do algoritmo externo, comandando a carga a descarga das baterias para injeção na rede. Nota-se que as baterias carregam algumas vezes durante a madrugada (00:00 às 04:00) e começam a injetar no começo da manhã, suprindo alguns picos de demanda durante o resto do dia. Este algoritmo e sistema de controle foi testado durante 3 meses ininterruptos.

5.2.4.2 Controle manual via dashboard

Em paralelo com a criação do fluxo para controle automatizado via entradas externas, também foi desenvolvido um dashboard que possibilita, de maneira visual e descomplicada, escolher entre três modos de operação: Algoritmo Interno do Inversor, Algoritmos Externos e Controle Manual.

Figura 25 - Fluxo do dashboard



Fonte: Autor.

Figura 26 - Dashboard de controle do Inversor via Node-RED



Fonte: Autor.

A Figura 25 mostra o fluxo criado para a geração do dashboard (página web) da Figura 26, nela foram destacadas 6 áreas:

- Rosa – Pontos de partida: A execução do fluxo começa aqui em 2 nodes, o primeiro é um node que faz com que o algoritmo seja executado a cada 5 minutos e o segundo acontece quando o usuário interage com o botão “Definir Potência” no dashboard;

- Verde – Controles da Interface: Os nodes dessa parte do fluxo são os elementos que o usuário interage (Figura 26). Estão ligados a outros nodes que armazenam em memória temporário os valores definidos;
- Vermelho – Algoritmo Interno do Inversor: Esta parte do fluxo define um comando que desabilita os comandos externos no inversor, fazendo com que o mesmo trabalhe com seu algoritmo interno padrão;
- Amarelo – Comandos Manuais: Este bloco irá enviar comandos para o inversor com os parâmetros definidos na interface visual do dashboard
- Azul – Algoritmo Externo: Os nodes neste fluxo estão fazendo um trabalho similar ao que foi descrito no capítulo 4.3.7.1, mas ao invés de buscar seus parâmetros em uma API externa, ele busca os mesmos na base de dados *Influx*, respeitando o método como o novo algoritmo de controle foi escrito;
- Ciano – Envio do comando para o *Software* Agregador: O bloco final deste fluxo é onde todos acabam, contém apenas 2 nodes, um que limita a quantidade de comandos por segurança (caso alguém tente dar muitos comandos manualmente) e outro que escreve no arquivo de fila de comandos do *software* agregador.

5.3 HARDWARE UTILIZADO

O *hardware* utilizado para a implementação se divide em duas partes: O *hardware* local conectado no dispositivo que irá fornecer os dados e/ou ser controlado; e o servidor remoto onde ficam armazenados os dados.

5.4 INSTALAÇÕES DO SISTEMA

O sistema está atualmente instalado em 3 lugares: NUPEDEE, no centro de tecnologia da UFSM, container do carregador de veículos elétricos rápidos da UFSM e no container de *Carsharing* da COPEL em Curitiba.

5.4.1 Instalação no NUPEDEE

O Núcleo de Pesquisa e Desenvolvimento em Engenharia elétrica – NUPEDEE foi o local escolhido para a instalação do primeiro Inversor Híbrido da NHS, por consequência, a primeira versão do sistema foi instalada próxima ao mesmo, em um *Raspberry pi 3 Model B+*.

Figura 27 - Raspberry pi utilizado e roteador



Fonte: Autor.

Para evitar que o inversor e o *Raspberry pi* ficassem públicos na rede da UFSM, um roteador configurado com uma rede NAT (isolada) interna foi utilizada, com acesso externo sob demanda via redirecionamento de portas no firewall do mesmo. A Figura 27 mostra a instalação inicial, com o *Raspberry pi* exposto em cima do roteador.

Figura 28 - Visão geral da instalação no NUPEDDEE



Fonte: Autor.

Figura 29 - Raspberry pi protegido e escondido atrás da TV



Fonte: Autor.

Algum tempo depois, foi utilizado o mesmo *Raspberry pi* para exibir os dashboards criados no *Grafana* em uma TV, instalada ao lado do inversor, finalizando, por enquanto, a instalação do sistema neste ponto, conforme as figuras Figura 28 e Figura 29.

5.4.2 Instalação no Carport

O segundo local de instalação foi no container do *carport* da UFSM, onde se encontra o primeiro eletroposto rápido do Rio grande do Sul. Neste local está instalado o inversor híbrido da NHS em sua versão de 10kW.

Figura 30 - Visão geral da instalação no Container do Carport



Fonte: Autor.

A ligação à rede neste ponto foi feita como na instalação anterior, com um roteador para isolar o inversor e o *Raspberry pi* da rede da UFSM. O *Raspberry pi* também foi conectado, via cabo óptico e extensor USB, no medidor inteligente da Eletra, o mesmo estudado no capítulo 4.3.2.

5.4.3 Instalação no Container Copel

O container na COPEL em Curitiba é o terceiro e final local onde o sistema foi instalado. Diferentemente dos outros dois pontos onde a principal função do *Raspberry pi* era rodar os *softwares* agregadores, nesta instalação o *Raspberry pi* foi utilizado por um outro projeto, que visa fazer a automação de dispositivos como lâmpadas, TVs e aparelhos de ar condicionado, através do sistema operacional *Home Assistant*, o qual não permite acesso direto ao seu sistema de arquivos.

Figura 31 - Raspberry pi com Home Assistant em caixa própria



Fonte: Autor.

Para evitar a instalação de mais um *Raspberry pi*, o *software* agregador foi instalado pela interface gráfica do *Home Assistant*, utilizando um sistema de containers chamado *Docker*, através do *addon Portainer* (NIJHOF, 2021). O *Raspberry pi* está conectado via *wifi* em um mini roteador, no qual também se conecta um cabo de rede, ligado ao inversor híbrido NHS de 3kW, o mesmo instalado no NUPEDDEE. Por motivos de segurança e políticas internas da empresa, em primeiro momento não foi possível conectar o *Raspberry pi* diretamente à rede da

COPEL, então foi instalado um modem 4G USB no mini roteador. O acesso remoto é feito através de redirecionamento de portas remotas via SSH (pacote SSH & Terminal no *Home Assistant*).

5.5 NOTAS FINAIS DO CAPITULO

Durante a implementação prática do sistema foram encontradas certas dificuldades, desde detalhes esquecidos na hora de especificar os requisitos do sistema, equipamentos que não implementam corretamente as especificações, falta de documentação ou documentação incorreta e também problemas físicos em nível eletrônico.

Apesar dos problemas encontrados, os 3 protótipos instalados seguem funcionando corretamente até o fim deste trabalho.

6 CONCLUSÃO

A redes elétricas inteligentes são uma realidade inescapável, causando a informatização de todo e quaisquer tipos de dispositivo dentro da mesma – seja desde um simples fusível com aviso, chaves remotas, transformadores monitorados ou mesmo sistemas de armazenamento de energia, inversores de frequência, e muitos outros. Esta grande gama de dispositivos com capacidade de comunicação, quando unidos, criam possibilidades ainda não muito exploradas, mas para ideias possam ser implementadas efetivamente, surge junto a necessidade de padronização dos métodos de comunicação dos mesmos, algo que ainda não se prevê acontecer.

A contribuição deste trabalho se dá no fato de mostrar que, mesmo com equipamentos diferentes de distintos fabricantes, é possível criar um sistema automatizado coerente e funcional.

Apesar dos objetivos definidos terem sido atingidos com sucesso, ficam abertas muitas possibilidades de trabalhos futuros, como a exploração de mais tipos de dispositivos com protocolos de comunicação diferentes dos estudados, teste de novos sistemas embarcados de *hardware* alternativos ao *Raspberry Pi*, refatoramento dos códigos para facilitar integração de novos equipamentos, interfaces gráficas mais amigáveis.

REFERÊNCIAS

- ANEEL. Resolução Normativa nº 482. Estabelece as condições gerais para o acesso de microgeração e minigeração distribuída aos sistemas de distribuição de energia elétrica, o sistema de compensação de energia elétrica, e dá outras providências. Disponível em: < <http://www2.aneel.gov.br/cedoc/ren2012482.pdf>>. Acesso em: 1 de out. 2021.
- ANEEL. Resolução Normativa nº 687. Altera a Resolução Normativa nº 482, de 17 de abril de 2012, e os Módulos 1 e 3 dos Procedimentos de Distribuição – PRODIST. Disponível em: < <http://www2.aneel.gov.br/cedoc/ren2015687.pdf>>. Acesso em: 1 de out. 2021.
- AKOREDE, Mudathir Funsho; HIZAM, Hashim; POURESMAEIL, Edris. Distributed energy resources and benefits to the environment. *Renewable and sustainable energy reviews*, v. 14, n. 2, p. 724-734, 2010.
- AZEVEDO, Rodrigo Motta de. *Matheurística Dinâmica e Pro-ativa Integrada a Inversores Híbridos para o Gerenciamento Robusto De Recursos Energéticos Distribuídos*. 2020. Teste (Doutorado em Engenharia Elétrica) - Universidade Federal de Santa Maria, Santa Maria: 2020
- BELLINASO, L. V.; MICHELS, L. . Inversores fotovoltaicos híbridos: a integração de inversores conectados à rede e no-breaks. *RBS Magazine - Revista Brasileira de Energia Solar*, p. 17 - 19, 01 mar. 2016.
- BLACKSTOCK, Michael; LEA, Rodger. Toward a distributed data flow platform for the web of things (distributed node-red). In: *Proceedings of the 5th International Workshop on Web of Things*. 2014. p. 34-39.
- CASELLA, G.; BERGER, R. L. *Statistical inference*. 2nd ed ed. Australia ; Pacific Grove, CA: Thomson Learning, 2002.
- CHORUANG, Komkrit; MEEKUL, Kraison. Design of an IoT energy monitoring system. In: *2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE)*. IEEE, 2018. p. 1-4.
- CUNHA, Murilo Vargas da. *ESTRATÉGIAS DE GERENCIAMENTO PELO LADO DA DEMANDA APLICADAS AOS CONSUMIDORES DE BT CONSIDERANDO A TARIFA BRANCA E A GERAÇÃO DISTRIBUÍDA*. UFSM, 2016.
- DIRK, Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*. 2014(239), 2014.
- DOMINGUEZ-GARCIA, Alejandro D.; HADJICOSTIS, Christoforos N. Distributed algorithms for control of demand response and distributed energy resources. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011. p. 27-32.
- JIAYI, Huang; CHUANWEN, Jiang; RONG, Xu. A review on distributed energy resources and MicroGrid. *Renewable and Sustainable Energy Reviews*, v. 12, n. 9, p. 2472-2483, 2008.

- MALUF, Nadim. 95. THE CHANGING NATURE OF THE FUTURE ELECTRIC GRID <<https://qnovo.com/95-changing-nature-future-electric-grid/>>. Publicado em 2016, acesso em 10/10/2021
- NIJHOF, Franck. *Home Assistant* Community Add-on: Portainer. <<https://community.home-assistant.io/t/home-assistant-community-add-on-portainer/68836>>. Acesso em 5 de out. 2021
- NITULESCU, Ioana-Victoria; KORODI, Adrian. Supervisory control and data acquisition approach in node-RED: Application and discussions. *IoT*, v. 1, n. 1, p. 76-91, 2020.
- OLIVEIRA, P. E. S. et al. Nocturnal Intermittent Coupling Between the Interior of a Pine Forest and the Air Above It. *Boundary-Layer Meteorology*, v. 146, n. 1, p. 45–64, jan. 2013.
- OpenJS Foundation & Contributors. About: Node-RED. Disponível em <<https://Node-RED.org/>>. Acesso em 14/10/2021.
- Palmlund Wahlgren Innovative Technology AB. DLMS.dev. Disponível em <<https://dlms.dev/>>. Acesso em 15/10/2021.
- PAUL, Shuva; KHAN, Md Kamrul Alam. Design, fabrication and performance analysis of solar inverter. In: 2013 IEEE Energytech. IEEE, 2013. p. 1-5.
- RANDLE, Bill et al. *Sunspec* Alliance Interoperability Specification-Inverter Controls Model. 2013.
- RODRÍGUEZ-MOLINA, Jesús. The Role of *Middleware* in Distributed Energy Systems Integrated in the *Smart grid*. *Energy Management of Distributed Generation Systems*, p. 133, 2016.
- SWALES, Andy et al. Open *Modbus/tcp* specification. Schneider Electric, v. 29, p. 3-19, 1999.