

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA
ENGENHARIA ELÉTRICA

ALYNNE PENHA DE MENDONÇA SILVA
LUCAS EDUARDO RIBEIRO

**SENSORIAMENTO REMOTO DE POTÊNCIA PARA O SISTEMA
FOTOVOLTAICO CONECTADO À REDE DO ESCRITÓRIO VERDE
DA UTFPR**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2017

ALYNNE PENHA DE MENDONÇA SILVA
LUCAS EDUARDO RIBEIRO

**SENSORIAMENTO REMOTO DE POTÊNCIA PARA O SISTEMA
FOTOVOLTAICO CONECTADO À REDE DO ESCRITÓRIO VERDE
DA UTFPR**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de TCC2, do curso de Engenharia Elétrica do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR) como requisito para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Glauber Gomes de Oliveira Brante

CURITIBA
2017

**Alyne Penha de Mendonça Silva
Lucas Eduardo Ribeiro**

Sensoriamento Remoto de Potência para o Sistema Fotovoltaico Conectado à Rede do Escritório Verde da UTFPR.

Este Trabalho de Conclusão de Curso de Graduação foi julgado e aprovado como requisito parcial para a obtenção do Título de Engenheiro Eletricista, do curso de Engenharia Elétrica do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR).

Curitiba, 30 de novembro de 2017.

Prof. Emerson Rigoni, Dr.
Coordenador de Curso
Engenharia Elétrica

Profa. Annemarien Gehrke Castagna, Mestre
Responsável pelos Trabalhos de Conclusão de Curso
de Engenharia Elétrica do DAELT

ORIENTAÇÃO

Glauber Gomes de Oliveira Brante, Dr.
Universidade Tecnológica Federal do Paraná
Orientador

BANCA EXAMINADORA

Glauber Gomes de Oliveira Brante, Dr.
Universidade Tecnológica Federal do Paraná

Guilherme Luiz Moritz, Dr.
Universidade Tecnológica Federal do Paraná

Jair Urbanetz Junior, Dr.
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia Elétrica.

AGRADECIMENTOS

Gostaríamos de agradecer ao Prof. Dr. Glauber Brante por toda a orientação e apoio dedicado durante o desenvolvimento deste trabalho e por ter nos proporcionado um modelo de orientação comprometida e atenciosa.

Agradecemos ao Prof. Dr. Guilherme Moritz por toda ajuda fornecida, cujas intervenções foram essenciais para conclusão deste projeto, e ao prof. Dr. Jair Urbanetz pelo apoio e contribuição na construção desta pesquisa.

Agradecemos as nossas família que com muito carinho, incentivo e paciência nos acompanharam durante toda esta caminhada, este apoio tornou possível a conclusão desta etapa. E aos amigos que fizeram parte desta caminhada, por todo o incentivo e todos os momentos de aprendizados juntos.

RESUMO

RIBEIRO, Lucas; SILVA, Alynne. Sensoriamento Remoto de Potência para o Sistema Fotovoltaico Conectado à Rede do Escritório Verde da UTFPR. 2017. 47 f. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

O modelo de geração distribuída juntamente com as novas tecnologias voltadas a geração de energia mais limpa e eficiente, como sistemas fotovoltaicos e eólicos, estão conquistando espaço no cenário nacional e mundial. A avaliação desses modelos conectados à rede é de grande importância para o planejamento e desenvolvimento da expansão destas tecnologias. A análise desses sistemas fica condicionada a coleta e armazenamento de dados. Como o Escritório Verde da Universidade Tecnológica Federal do Paraná, centro de referência em sustentabilidade no estado do Paraná, necessitava de um sistema que monitorasse seu consumo e também a produção do Sistema Fotovoltaico Conectado à rede, foi proposto neste trabalho a implementação de um sistema de monitoramento e armazenamento dessas informações em um banco de dados. Com o propósito de corrigir esta deficiência foi implementado no escritório verde um sistema de monitoramento de baixo custo, utilizando um Raspberry PI com o *software* Node-RED e o sistema de medição já instalado no local. Como resultado obteve-se uma aplicação capaz de armazenar e apresentar em um *dashboard* parâmetros de potência, tais como potência ativa, reativa e energia, essenciais para análise de viabilidade, e alguns parâmetros de qualidade como taxa de distorção harmônica da corrente e tensão, além do fator de potência.

Palavras-chave: Sistemas Fotovoltaicos. Medição Remota. Geração Distribuída. Banco de Dados. Node-RED.

ABSTRACT

RIBEIRO, Lucas; SILVA, Alynne. Power Remote Monitoring for the Grid-Connected Photovoltaic System at the Green Office at UTFPR. 2017. 47 f. Trabalho de Conclusão de Curso (Engenharia Elétrica) – Departamento Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

The distributed generation model allied with the new technologies focused on a more efficient and less polluting energy sources, such as photovoltaic systems and eolic power plants have gained importance both in the national and international scenario. The evaluation performance of these systems connected to the grid have great value to the planning and expansion development of these technologies. The analysis of these systems is conditioned to the collecting and storage of his data. Since the Green Office at UTFPR, regional center of reference in sustainability, lacked a monitoring system for the consumption and production of the photovoltaic system connected to its grid, in this paper was proposed to implement a monitoring and storage system for the power data of this building. With the purpose to correct this deficiency was implemented a low cost system using a Raspberry PI with Node-RED software, and the already installed system of measurement. This system is able to give the main power parameters, such as active and reactive power, in addition to quality parameters like total harmonic distortion (THD) and power factor.

Keywords: Photovoltaic Systems. Remote Measurement. Distributed Generation. Database. Node-RED.

LISTA DE FIGURAS

FIGURA 1	– Exemplo de aplicação em IoT	11
FIGURA 2	– Escritório Verde da UTFPR	11
FIGURA 3	– Fluxo de Dados	15
FIGURA 4	– Esquema de Conexões dos elementos do Sistema	16
FIGURA 5	– Sistema Fotovoltaico Conectado à Rede	17
FIGURA 6	– Painel solar e inversor do SFCR do Escritório Verde	18
FIGURA 7	– Medidores instalados no EV	19
FIGURA 8	– Exemplo de leitura dos dados dos medidores	21
FIGURA 9	– Regra dos Trapézios Repetida	22
FIGURA 10	– Tela de trabalho principal do Node-RED	23
FIGURA 11	– Descrição do uso da pilha TCP/IP pelo protocolo Modbus	24
FIGURA 12	– Frame de Requisição/Resposta do Protocolo Modbus	25
FIGURA 13	– Esquema de um Raspberry PI modelo B	27
FIGURA 14	– Sistema de Banco de Dados	28
FIGURA 15	– Mensagem de Requisição medidores PAC3200	30
FIGURA 16	– Diagrama subfluxo medidor	31
FIGURA 17	– Subfluxo de decodificação dos dados	31
FIGURA 18	– Armazenamento de dados com Node-RED	32
FIGURA 19	– Diagrama de Entidades do Banco de Dados	33
FIGURA 20	– Página do consumo	34
FIGURA 21	– Gráfico da potência gerada	34
FIGURA 22	– Troca de potência com o Bloco V	35
FIGURA 23	– Fluxo para correção de falhas	36
FIGURA 24	– Acesso ao Node-RED	41
FIGURA 25	– Tela de autenticação	41
FIGURA 26	– Página inicial Putty	42
FIGURA 27	– Ambiente terminal do Raspberry Pi	42
FIGURA 28	– Arquivo de configuração do Node-RED	43
FIGURA 29	– Página de download MySQL Workbench	44
FIGURA 30	– Escolha do sistema operacional e download do MySQL Workbench.	45
FIGURA 31	– Página inicial MySQL Workbench.	45
FIGURA 32	– Solicitação de senha de acesso para o banco de dados.	46
FIGURA 33	– Menu inicial do MySQL Workbench.	46
FIGURA 34	– Página de Exportação - Seleção das variáveis	47
FIGURA 35	– Página de Exportação - Local para armazenamento do BD.	47

LISTA DE TABELAS

TABELA 1	– Tabela de Registros das Medições do PAC3200 e PAC4200	20
TABELA 2	– MBAP Header do protocolo Modbus	25
TABELA 3	– Códigos de comando Modbus	25
TABELA 4	– Custos do Sistema	36

LISTA DE SIGLAS

ADU	Application Data Unit
APIs	Application programming interface
BD	Banco de Dados
EV	Escritório Verde
FV	Fotovoltaico
GPIO	General Purpose Input Output
IBM	International Business Machines Corporation
IoT	Internet of Things
MBAPH	Modbus Application Protocol Header
PDU	Protocol Data Unit
RSSF	Rede de Sensores Sem Fio
RTC	Real Time Clock
SFCR	Sistema Fotovoltaico Conectado à Rede Elétrica
SGBD	Sistema de Gerenciamento de Banco de Dados
SGBDR	Sistema de Gerenciamento de Banco de Dados Relacional
SIN	Sistema Interligado Nacional
SO	Sistema Operacional
SQL	Structured Query Language
SSH	Secure Shell
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1	INTRODUÇÃO	10
1.1	TEMA	10
1.2	PROBLEMAS E PREMISSAS	12
1.3	OBJETIVOS	13
1.3.1	Objetivo Geral	13
1.3.2	Objetivos Específicos	13
1.4	JUSTIFICATIVA	13
1.5	PROCEDIMENTOS METODOLÓGICOS	14
1.6	ESTRUTURA DO TRABALHO	14
2	ASPECTOS DAS SOLUÇÕES	16
2.1	SISTEMA FOTOVOLTAICO CONECTADO A REDE	17
2.1.1	Escritório Verde - UTFPR	18
2.2	SISTEMA DE AQUISIÇÃO DOS DADOS	19
2.2.1	Medidores	19
2.2.2	Conexão com os Medidores e Tratamento dos Dados	23
2.2.3	Comparação com Outras Soluções	27
2.3	SISTEMA DE ARMAZENAMENTO DOS DADOS	27
2.3.1	Do Raspberry Pi	27
2.3.2	Do Banco de Dados	28
2.4	VISUALIZAÇÃO DOS DADOS	30
3	RESULTADOS	31
3.1	COLETA E TRATAMENTO DOS DADOS	31
3.2	ARMAZENAMENTO DOS DADOS	33
3.3	DASHBOARD LOCAL PARA VISUALIZAÇÃO DOS DADOS	34
3.4	PREVENÇÃO DE FALHAS	36
3.5	CUSTOS DOS EQUIPAMENTOS	37
3.6	DISCUSSÃO DOS RESULTADOS	37
4	CONCLUSÕES	39
	REFERÊNCIAS	40
	Apêndice A – ACESSO AOS DADOS	42
	Apêndice B – DOWNLOAD DOS DADOS	45

1 INTRODUÇÃO

1.1 TEMA

O mundo vive hoje uma tensão entre o consumo crescente de energia elétrica, sendo esta fundamental ao desenvolvimento de uma sociedade tecnológica, e as demandas por novas tecnologias e modelos de gestão energética que respondam tanto a antecipação de uma crise na matriz dominante de origem fóssil, como a questões sócio-ambientais. A situação energética do Brasil se distingue do contexto mundial por ter como característica fundamental sua matriz energética de origem predominantemente renovável, composta principalmente por usinas hidrelétricas, térmicas e em menor escala usinas nucleares, eólicas dentre outras. A participação de fontes renováveis ficou próxima de 81,7% da oferta interna de eletricidade em 2016, sendo a geração hidráulica responsável por 68,1% da oferta de potência, com 96.925MW de capacidade instalada, e a geração termelétrica por cerca de 28,4% de participação (EPE, 2016).

Apesar desse caráter predominantemente renovável, é preciso lembrar que a geração hidráulica no Brasil é garantida por grandes centrais hidrelétrica construídas nos leitos dos rios brasileiros, e inúmeros problemas sócio-ambientais são decorrentes desse modelo. Além disso esta fonte está sujeita às condições hidrológicas do país, o que quer dizer que quando existe alguma redução do potencial hídrico existe a necessidade de compensação da demanda por outra fonte, no caso do Brasil principalmente pela geração termelétrica, uma fonte cara e altamente poluente. Os impactos inerentes à geração térmica e nuclear estão relacionados principalmente ao tipo de combustível utilizado para gerar energia, seja da queima de combustíveis fósseis, ou o problema do descarte do material radioativo e das consequências de eventuais acidentes que possam ocorrer para o homem e o meio ambiente. A este quadro somam-se ainda questões relacionadas com a transmissão e distribuição de energia, posto o extenso território nacional, o grande número de consumidores e o difícil acesso a locais isolados do Sistema Interligado Nacional (SIN), que interliga os sistemas de geração, transmissão e distribuição de energia da maior parte do território brasileiro.

Se considerados os diversos problemas associados a matriz elétrica brasileira, nota-se a necessidade de diversificação dos métodos de geração, transmissão e distribuição de energia, principalmente em um país com grande potencial para produzir e diversificar suas fontes. Neste contexto a geração fotovoltaica é uma opção bastante interessante, pois além do enorme potencial fotovoltaico do país, esta fonte possibilita a integração com a rede de distribuição e

geração junto ao ponto de consumo, facilitando a agregação do conceito de geração. Sistemas fotovoltaicos conectados à rede elétrica (SFCR) tem se mostrado uma opção viável que não só atenuam os aspectos relativos a impactos ambientais, por ser uma fonte limpa e inesgotável, como melhoram consideravelmente o desempenho da rede de distribuição. Apesar do uso de sistemas fotovoltaicos estar em ascensão no cenário mundial esta tecnologia ainda está sendo avaliada no Brasil, com uma participação tímida de 0,01% da matriz energética (EPE, 2016).

Para facilitar a inserção desta fonte torna-se essencial o uso de tecnologias que permitam avaliar o desempenho desses sistemas, como o uso de dispositivos inteligentes que integrem conceitos de gestão de energia, otimização da eficiência, e serviços de medição e controle que permitam a coleta de dados. A necessidade crescente de sistemas inteligentes e capazes de analisar dados em tempo real, processá-los e tomar uma ação baseado em um rol de informações, tem desenvolvido e aprimorado cada vez mais tecnologias, objetos e processos. Neste contexto a internet das coisas (IoT, em inglês *Internet of Things*) tem se popularizado pois segundo Perera et al. (2014a) e Sundmaeker et al. (2010) esta permitirá uma comunicação direta entre qualquer objeto, rede ou pessoa. Assim os objetos inteligentes poderiam interagir trocando dados, informação e conhecimento. De acordo com Perera et al. (2014b) os objetos seriam capazes de analisar nossas preferências e necessidades, e então sem a necessidade de um comando explícito executarem ações baseadas nessa análise. Para isso os objetos inteligentes precisam de um contexto, informações fornecidas por um ou mais sensores capazes de caracterizar a situação.

Tendo em vista a contínua busca por elementos, sistemas e processos mais eficientes, tanto do ponto de vista econômico quanto ambiental, as aplicações envolvendo IoT são um ponto chave e que ainda tem muito campo a ser estudado e desenvolvido. Um exemplo de aplicação com impacto ambiental e econômico envolvendo IoT é o sistema de rotas inteligentes para captação de lixo desenvolvido pela empresa (ENEVO, 2016). A Figura 1 foi retirada do site da empresa e ilustra bem este exemplo. A proposta da aplicação é monitorar o nível de lixo presente em cada lixeira e, ao invés de propor uma rota fixa como normalmente é feito nas cidades, propor uma rota otimizada para retirada do lixo.

Dessa forma, este trabalho propõe o desenvolvimento de um sistema inteligente de medições para o monitoramento de um SFCR. A análise foi feita em uma edificação da Universidade Tecnológica Federal do Paraná (UTFPR), o Escritório Verde (EV), que adotou o sistema fotovoltaico conectado à rede elétrica como geração de energia, onde se trabalha ao mesmo tempo com a ideia de geração distribuída, em que o consumidor gera a energia que consome, e o uso de módulos fotovoltaicos, utilizando-se energia solar como fonte de energia.

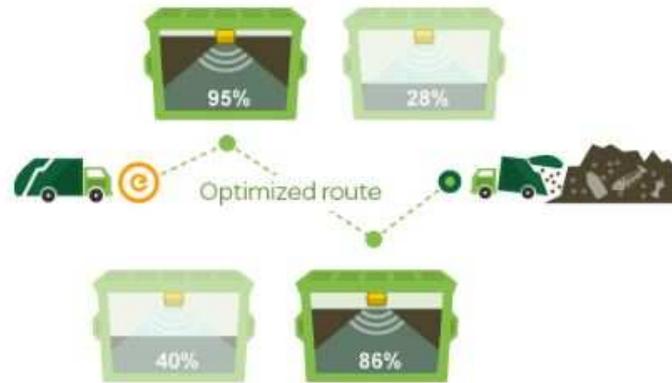


Figura 1: Exemplo de aplicação em IoT: Rota dinâmica para coleta de lixo.

Fonte: Enevo (2016)

O SFGR é composto por um sistema de geração fotovoltaico, que converte energia solar em elétrica, e um inversor, que converte a corrente contínua que sai do painel solar em corrente alternada com valores compatíveis com o sistema elétrico ao qual está conectado (URBANETZ; CASAGRANDE, 2012).



Figura 2: Escritório Verde da UTFPR.

Fonte: Urbanetz e Casagrande (2012)

1.2 PROBLEMAS E PREMISSAS

Dentre os principais problemas enfrentados pela geração convencional estão o congestionamento das linhas de transmissão e os diversos problemas sócio-ambientais relacionados com as fontes geradoras predominantes. A ampliação do uso de energia solar através de sistemas fotovoltaicos é uma solução a esses problemas pois oferece a possibilidade de uma geração menos agressiva e mais distribuída (URBANETZ, 2010).

Tendo-se em conta que a aquisição de dados de potência e a existência de um banco de dados são fatores fundamentais para a análise e implementação de sistemas fotovoltaicos e sabendo-se que o Escritório Verde possui apenas dados de potência na saída do inversor, viu-se a necessidade de coletar outros dados, como a potência consumida pelo escritório e a potência trocada com o bloco V da UTFPR, sede centro em Curitiba, para onde escoo o excedente da produção de energia.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Visando tornar o processo de verificação do desempenho do SFCR instalado no Escritório Verde mais eficiente, propõe-se neste trabalho o desenvolvimento e análise de um sistema inteligente de medições, capaz de monitorar e exibir em um *dashboard* local parâmetros de potência do Escritório Verde tais como potência gerada pelos painéis fotovoltaicos, potência consumida no escritório e potência trocada com o bloco V.

1.3.2 Objetivos Específicos

- Implementar um sistema inteligente de coleta de dados para fazer interface com os medidores de potência já instalados no EV;
- Criar um banco de dados acessível, através de registro, aos coordenadores do EV;
- Desenvolver um *dashboard* capaz de acessar o banco de dados e apresentar os dados coletados em uma interface visual;
- Garantir a estabilidade e confiabilidade da aquisição dos dados.

1.4 JUSTIFICATIVA

A solução proposta neste trabalho se justifica pelo desenvolvimento de um sistema inteligente de medições que contribua para uma melhor eficiência do Sistema Fotovoltaico Conectado à Rede do Escritório Verde da UTFPR, por meio do monitoramento em tempo real dos dados de geração e consumo, permitindo assim o controle do desempenho desses sistemas.

Um outro aspecto decisório para a escolha desta solução foram os fatores custo e baixo consumo de energia. Dessa maneira, a escolha das tecnologias que irão compor esta

aplicação justifica-se na generalidade a qual esta aplicação pode ser utilizada e no baixo custo dos equipamentos. Além disso, o sistema escolhido destaca-se pelo fato de gravar os dados sem a dependência de conexão com a internet e de possibilitar um armazenamento local dos mesmos, o que resulta em um sistema mais seguro e com baixa possibilidade de perda de dados, problema crucial e decisivo nessa aplicação.

1.5 PROCEDIMENTOS METODOLÓGICOS

A fim de escolher a melhor solução, levando em conta aspectos como tempo de implementação, recursos disponíveis, eficácia e usabilidade da aplicação final, a primeira fase do projeto teve como foco principal a revisão bibliográfica, isto é, um estudo do sistema proposto e as suas etapas constitutivas, que pode-se resumir em 3 elementos: um sistema de coleta dos dados, composto por 3 medidores já instalados no SFCR local (2 SENTRON PAC3200 e 1 SENTRON PAC4200) que coletam os dados de potência gerada, consumida e excedente do SFCR; um sistema de armazenamento, composto por um banco de dados desenvolvido para esta aplicação; e finalmente um sistema para a visualização local desses dados.

Em um segundo momento atuou-se na implementação do sistema em si. De modo geral foi utilizada uma unidade agregadora central, o microcomputador Raspberry PI, responsável tanto pela comunicação com os medidores, coleta e tratamento dos dados de potência, quanto pelo armazenamento desses dados em um banco de dados local e o desenvolvimento de um *dashboard*, por onde os dados podem ser visualizados e acessados na rede local do EV. Por fim houve a fase de testes, onde foi analisado se todos os pré-requisitos definidos na proposta e na fase de revisão bibliográfica foram cumpridos.

1.6 ESTRUTURA DO TRABALHO

Este trabalho foi estruturado da seguinte maneira:

- O Capítulo 1 trata da apresentação do trabalho, definição e delimitação do tema a ser abordado, problematização e objetivos.
- O Capítulo 2 apresenta a revisão bibliográfica e fundamentação teórica sobre o tema. Este capítulo expõe quais as soluções e plataformas utilizadas para a resolução do problema proposto além das características destas soluções.

- O Capítulo 3 apresenta e discute os resultados obtidos, avaliando se todos os pré-requisitos foram cumpridos.
- O Capítulo 4 trata das conclusões e sugestões para trabalhos futuros.

2 ASPECTOS DAS SOLUÇÕES

Este Capítulo trata da apresentação das soluções selecionadas e das características referentes a cada uma. O diagrama de blocos da Figura 3 apresenta o esquema com todos os elementos da aplicação bem como o fluxo de dados.

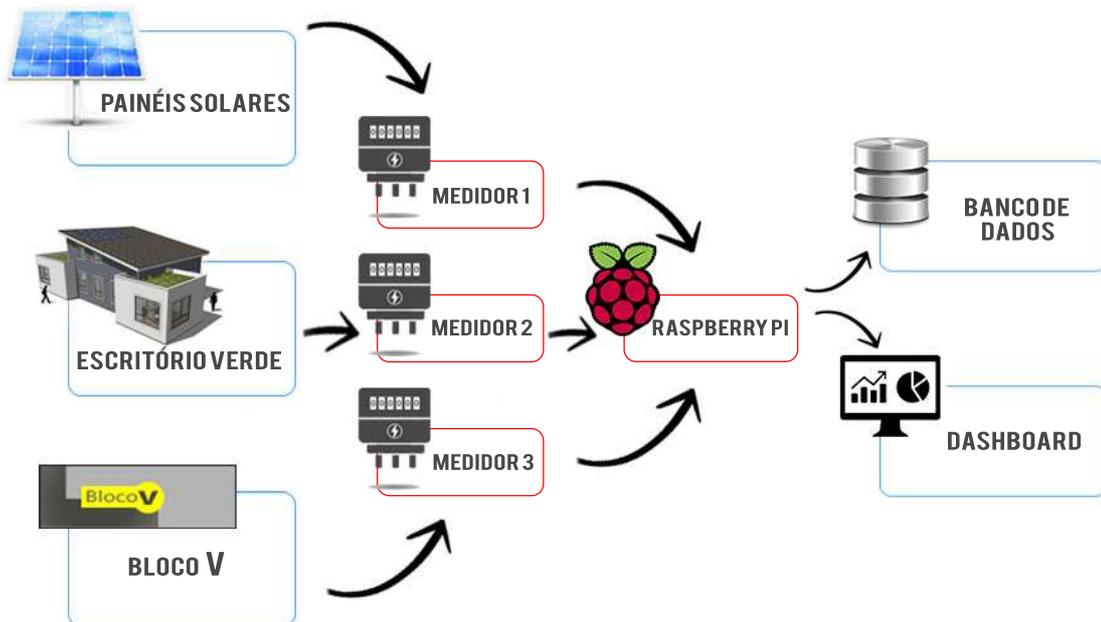


Figura 3: Fluxo de dados no sistema.

Fonte: Autoria própria

No diagrama de blocos da Figura 3 percebe-se como é dado o fluxo de dados através do sistema. De modo geral, a energia gerada pelo sistema fotovoltaico conectado a rede alimenta o Escritório Verde por um sistema de fornecimento bifásico, e quando houver excedente da produção fotovoltaica esta vai escoar para o bloco V, um prédio adjacente ao EV. Após a geração de energia pelo SFCR os dados são coletados nos medidores e seguem até o Raspberry PI onde está instalado o banco de dados. Uma vez que os dados forem salvos no banco de dados eles podem ser acessados por um *dashboard* local onde podem ser visualizados.

Para a melhor compreensão do fluxo de dados no sistema, este capítulo será dividido em quatro partes: O Sistema Fotovoltaico Conectado à rede, que explica de forma resumida o processo de geração de potência no sistema que será analisado. A coleta dos dados que apresenta os principais responsáveis pela coleta e como será realizada a conexão entre os

mesmos. O armazenamento dos dados que explica o banco de dados e o hardware em que este será instalado e a visualização dos dados que apresenta o *dashboard* desenvolvido para a visualização dos dados.

O esquema apresentado na Figura 4 apresenta como será feita a comunicação entre os principais componentes do sistema. Como pode se observar, todo o processo do fluxo de dados, que compreende a coleta dos dados, o armazenamento no BD e o *dashboard* para o acesso ao banco de dados, acontecerá em uma rede interna, o que torna o sistema mais confiável ao garantir que esse fluxo não dependa de uma conexão com a internet para acontecer. Todos essas etapas do fluxo de processamento e os componentes do sistema serão melhor explicados na sequência.

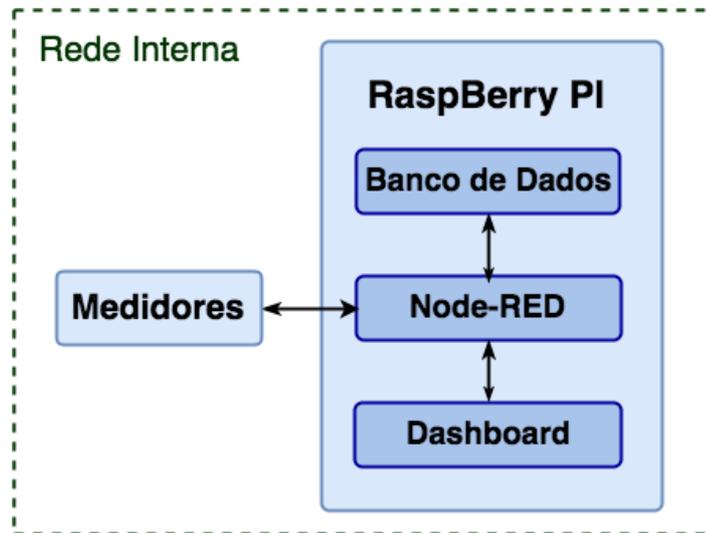


Figura 4: Esquema de Conexão dos elementos do Sistema

Fonte: Autoria Própria

2.1 SISTEMA FOTOVOLTAICO CONECTADO A REDE

O Sol é a grande fonte primária de energia do planeta, sendo responsável direta ou indiretamente pela maior parte da energia disponível na Terra. É possível gerar energia elétrica a partir da radiação solar tanto pelo aproveitamento térmico, que implica na absorção de energia sob a forma de calor, quanto pelo efeito fotovoltaico (FV), a conversão de luz em eletricidade. A energia solar fotovoltaica é a energia obtida através do efeito fotovoltaico, ou seja, a conversão direta de luz em eletricidade realizada através de células fotovoltaicas, a unidade fundamental desse processo de conversão (PINHO; GALDINO, 2014).

As células fotovoltaicas são formadas por um material semicondutor, por exemplo o

silício, que quando exposto a luz tem como propriedade uma diferença de potencial nos seus extremos tornando possível a geração de eletricidade. Estas células são conectadas entre si por arranjos série/paralelo formando o módulo fotovoltaico o qual, por fim, vai se associar com outros módulos para formar o painel fotovoltaico com os níveis de tensão e corrente desejados para o sistema (URBANETZ, 2010).

Os sistemas fotovoltaicos podem ser isolados da rede elétrica, neste caso necessitando de algum elemento armazenador de energia, ou podem ser conectados diretamente na rede sendo então chamados de sistemas fotovoltaicos conectados à rede elétrica. De modo geral esses SFCRs são compostos de um painel FV e um inversor que vai converter a corrente contínua que sai dos módulos em corrente alternada, com tensão e frequência compatíveis com a rede elétrica. Esta configuração pode ser verificada na Figura 5.

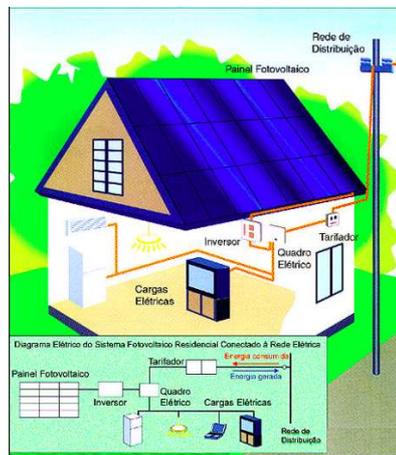


Figura 5: Sistema Fotovoltaico Conectado à Rede.

Fonte: CEPEL (2003)

Urbanetz (2010) aponta que as principais vantagens desse tipo de sistema são a elevada produtividade, ausência de baterias e desligamento automático no caso de falta de energia. Além disso esse tipo de sistema contribui com a disponibilidade de energia junto ao ponto de consumo, sendo vistos então como uma forma de geração distribuída.

2.1.1 Escritório Verde - UTFPR

O Escritório Verde é uma edificação sustentável elaborada pela Universidade Tecnológica Federal do Paraná em Curitiba. Além dos diversos métodos sustentáveis adotados durante a sua construção, foi instalado no EV um SFCR que juntamente da rede da concessionária supre a energia do imóvel.

O EV é alimentado pelo sistema fotovoltaico quando há geração de energia, ou seja, quando existe incidência solar, e no caso de baixa ou nenhuma incidência solar a demanda do EV é suprida pela concessionária. Quando a energia gerada é superior à consumida o sistema fotovoltaico escoar o excedente da produção para o bloco V da UTFPR, um prédio adjacente ao sistema de geração.

O SFCR do EV é composto por 10 módulos KYOCERA de tecnologia de silício policristalino ligados em série e cobrindo uma área de 15m², com uma potência total instalada de 2,1kWp, e um inversor monofásico PVPOWERED modelo PVP2000 em 220V com uma potência nominal de 2kW para a conversão da corrente contínua em alternada (URBANETZ; CASAGRANDE, 2012). Este inversor também possui um sistema de monitoramento que inclui o módulo de monitoramento PVM1010 e o acesso do usuário às informações de desempenho do inversor em um site protegido por senha. O módulo determina a quantidade de energia que está sendo gerada a partir do SFCR a partir da coleta contínua de dados de tensão e corrente do inversor, calculando a produção de energia em tempo real em kWp e a produção de energia acumulada em kWh em vários intervalos de tempo (PVPOWERED, 2009). Os painéis instalados e o inversor que compõem o sistema do EV estão apresentados na Figura 6.



Figura 6: Painel solar e inversor do SFCR do Escritório Verde.

Fonte: Urbanetz e Casagrande (2012)

2.2 SISTEMA DE AQUISIÇÃO DOS DADOS

2.2.1 Medidores

Para a coleta e armazenamento dos dados de geração e consumo da energia gerada pelo sistema fotovoltaico foram instalados medidores de energia no EV. No total são três medidores da Siemens instalados no quadro de medição, dois destes do modelo Sentron PAC3200, sendo

um a medição da geração do SFCR e outro a medição da troca com o bloco V. O terceiro medidor é do modelo Sentron PAC4200 e mede o consumo do EV. A Figura 7 apresenta os medidores instalados.



Figura 7: Medidores instalados no EV.

Fonte: Autoria própria

Os multímetros Sentron PAC permitem visualizar todos os parâmetros de rede relevantes para a distribuição de energias em baixa tensão, disponibilizando uma gama variada de grandezas (SIEMENS, 2007, 2008). Como esses medidores possuem comunicação Modbus TCP e funcionam em uma configuração mestre-escravo, para se ter acesso a esses dados é necessário enviar uma mensagem de solicitação especificando os parâmetros pretendidos. Essas grandezas são identificadas na memória dos medidores por dois registros que devem ser especificados nesta mensagem de requerimento.

Desse modo, a partir das informações contidas no manual dos medidores foram selecionadas os parâmetros considerados mais relevantes para este trabalho, a saber, potência ativa, potência reativa, potência aparente, corrente e tensão eficazes, taxa de distorção harmônica, fator de potência e frequência, sendo todos estes indicadores referentes às linhas 1 e 2 do sistema bifásico de alimentação e consumo (SIEMENS, 2007, 2008). A Tabela 1 apresenta as variáveis selecionadas para este trabalho, dentre as várias fornecidas pelos medidores, bem como o registro correspondente a cada uma dessas grandezas nos multímetros. É importante ressaltar que a quantidade de variáveis mensuradas nos dois modelos dos medidores não são iguais, sendo que o multímetro Sentro PAC4200 registra mais tipos de variáveis que o Sentro

PAC3200. A lista de todos os registros disponibilizados pelos medidores podem ser encontradas nos seus respectivos manuais (SIEMENS, 2007, 2008).

Tabela 1: Tabela de Registros das Medições do PAC3200 e PAC4200

Registro Sentron PAC3200	Registro Sentron PAC4200	Variável
1	1	Tensão da Linha 1 (U1)
3	3	Tensão da Linha 2 (U2)
7	7	Tensão entre as Linhas 1 e 2 (U12)
13	13	Corrente da Linha 1 (I1)
15	15	Corrente da Linha 2 (I2)
19	19	Potência Aparente da Linha 1 (S1)
21	21	Potência Aparente da Linha 2 (S2)
25	25	Potência Ativa da Linha 1 (P1)
27	27	Potência Ativa da Linha 2 (P2)
31	31	Potência Reativa da Linha 1 (Q1)
33	33	Potência Reativa da Linha 2 (Q2)
37	37	Fator de Potência da Linha 1 (FP1)
39	39	Fator de Potência da Linha 2 (FP2)
43	261	THD da Tensão da Linha 1 (TDHU1)
45	263	THD da Tensão da Linha 2 (TDHU2)
49	267	THD da Corrente da Linha 1 (TDHI1)
51	269	THD da Corrente da Linha 2 (TDHI2)
55	55	Frequência (freq)

Para que os dados coletados pelos medidores possam ser utilizados e compreendidos eles precisam antes ser tratados. As grandezas selecionadas como relevantes para esta aplicação são armazenadas em dois registradores, portanto precisam ser agrupados e transformados em dados decifráveis (SIEMENS, 2011). Além de entender como o protocolo que envia os dados dos medidores funciona é necessário saber como os dados são organizados dentro do pacote a fim de que se possam decodificá-los.

Tomando como exemplo a leitura da potência ativa que flui por uma das linhas do medidor PAC4200, sabe-se a partir do manual desse medidor que essa grandeza é armazenada em uma variável de 32 bits, com dois registradores de 16 bits cada (SIEMENS, 2008). Portanto a primeira etapa é juntar os registradores que compõem uma mesma grandeza. A Figura 8 exemplifica esse processo. Após concatenar os registradores deve ser feita uma divisão entre os bits em três campos. O primeiro campo contém apenas um bit e indica o sinal, zero para positivo e um para negativo. O segundo campo, com 8 bits, indica quantos bits do terceiro campo vão compor a parte inteira, para isso é preciso fazer uma relação do número lido no campo dois menos 127. Por fim deve ser adicionado ao início do terceiro campo um bit de valor 1, o terceiro campo mais este bit acrescido comporão o valor em si (SIEMENS, 2011).

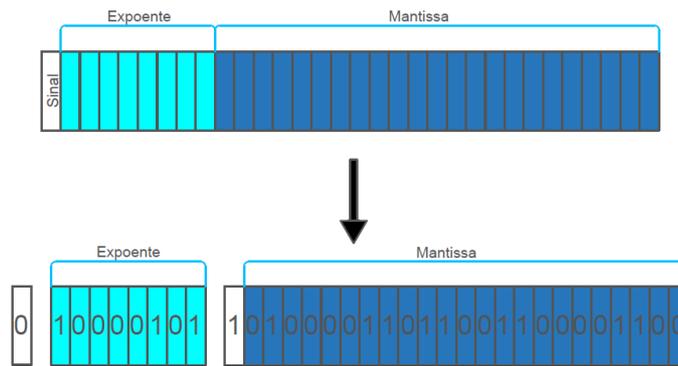


Figura 8: Exemplo de leitura dos dados dos medidores.

Fonte: Autoria própria.

Para o exemplo fornecido na Figura 8 o valor lido seria:

Primeiro campo: 0 \rightarrow Sinal positivo

Segundo campo: $(10000101)_2 - (127)_{10} = 133 - 127 \rightarrow 6$ bits. Isso significa que os 6 primeiros bits do terceiro campo mais o bit adicionado vão compor a parte inteira.

$$\text{Terceiro campo} \begin{cases} \text{Parte inteira} & (1010000)_2 = 80 \\ \text{Parte fracionária} & (11011001100001100)_2 = 111372 \end{cases}$$

Assim para o exemplo dado, a potência ativa lida é de 80,11 Watts.

Para fins computacionais, uma aproximação do valor lido pelo medidor e dada por (SIEMENS, 2011)

$$\text{valor lido} = (-1)^s (1 + m \cdot 2^{-23}) \cdot 2^{d-127}, \quad (1)$$

na qual d corresponde ao valor decimal do expoente, m à mantissa e s o sinal. No exemplo calculado anteriormente, empregando (1) chega-se ao valor de 80,85 Watts de leitura.

Ademais, além dos dados escolhidos dentre as variáveis disponíveis pelos medidores, viu-se a necessidade da obtenção de dados de energia, fator de grande importância para análise econômica e de viabilidade de um projeto de geração distribuída. No entanto uma questão que teve de ser considerada é que os medidores não disponibilizam diretamente dados da energia produzida e consumida pelo SFCR. Optou-se então pelo cálculo da energia a partir da potência utilizando a Regra dos Trapézios Repetida de Newton-Cotes (LOPES; RUGGIERO, 1988), que consiste em aproximar pequenos trechos da curva de uma função pela integral dos segmentos de reta definidos por esses trechos, como pode ser melhor visualizado no gráfico da Figura 9.

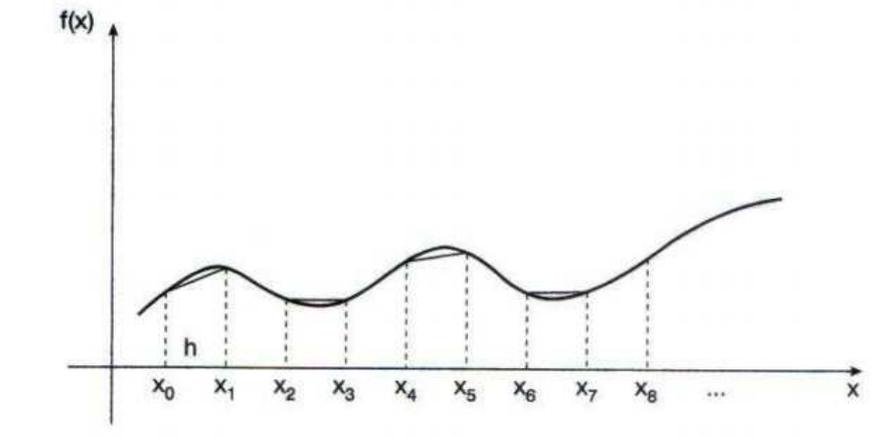


Figura 9: Regra dos Trapézios Repetida.

Fonte: (LOPES; RUGGIERO, 1988)

De acordo com Lopes e Ruggiero (1988) essa regra é equivalente a aproximar a área abaixo da curva dessa função pela soma das áreas dos trapézios nos x_i pontos de subdivisão do intervalo de integração, em que $f(x_i)$ e $f(x_{i+1})$ são as bases do trapézio, e $h = x_{i+1} - x_i$ é a largura, com x_i pontos de subdivisão e $i = 0, 1, \dots, n$. Dessa forma, a Regra Trapezoidal Repetida é definida por:

$$I_{TR} = h/2 \cdot [f(x_0) + 2 \cdot (f(x_1) + f(x_2) + \dots + f(x_{n-1})) + f(x_m)], \quad (2)$$

O cálculo da energia será realizado a partir de uma adaptação desta regra, em que será utilizada a curva da Potência Ativa (P) gerada e consumida pelo sistema em função do tempo de amostragem da coleta nos medidores. Portanto, a energia será calculada pela função:

$$Energia = t/2 \cdot [P(x_0) + 2 \cdot (P(x_1) + P(x_2) + \dots + P(x_{n-1})) + P(x_m)] \quad (3)$$

Sendo "P" a potência resultante da soma das potências da linha 1 e 2 do sistema, e "t" o intervalo de tempo utilizado entre uma medição e outra da potência.

2.2.2 Conexão com os Medidores e Tratamento dos Dados

Para o tratamento do dados e a conexão com os medidores será instalado a ferramenta de programação visual Node-RED no microcomputador Raspberry Pi (este será melhor apresentada na seção 2.3.1). O Node-RED é responsável por tratar e decodificar os dados

recebidos pelos medidores, comunicar os medidores com o servidor de banco de dados instalado no Raspberry Pi e por fim comunicar o banco de dados com um *dashboard* desenvolvido no próprio Node-RED para o acesso aos dados pela rede local.

Este editor é uma ferramenta desenvolvida pela IBM para conectar rapidamente dispositivos IoT de *hardware*, APIs (Interface de programação de aplicativos) e serviços online (JS FOUNDATION, 2017). É um editor de fluxo de código aberto implementado em JavaScript, uma linguagem de programação para a execução de *scripts* em navegadores web do lado do cliente sem que seja necessário que este *script* passe pelo servidor (ZAIDI, 2017). Além disso utiliza o *framework* Node.js, uma plataforma de programação JavaScript do lado do servidor projetado para construir aplicações de rede rápidas e escaláveis (NODE.JS FOUNDATION, 2017).

O Node-RED torna possível a abstração de processos envolvendo IoT a partir da lógica de ligação em blocos, isto porque a maior parte da programação é feita a partir de funções pré-definidas, os "nós", o que reduz programações complexas em diferentes linguagens a uma simples arrastar e soltar desses nós. A interface gráfica principal do Node-RED pode ser vista na Figura 10, onde os blocos são os nós de processamento e o conjunto desses nós formam um fluxo por onde os dados vão sendo manipulados.

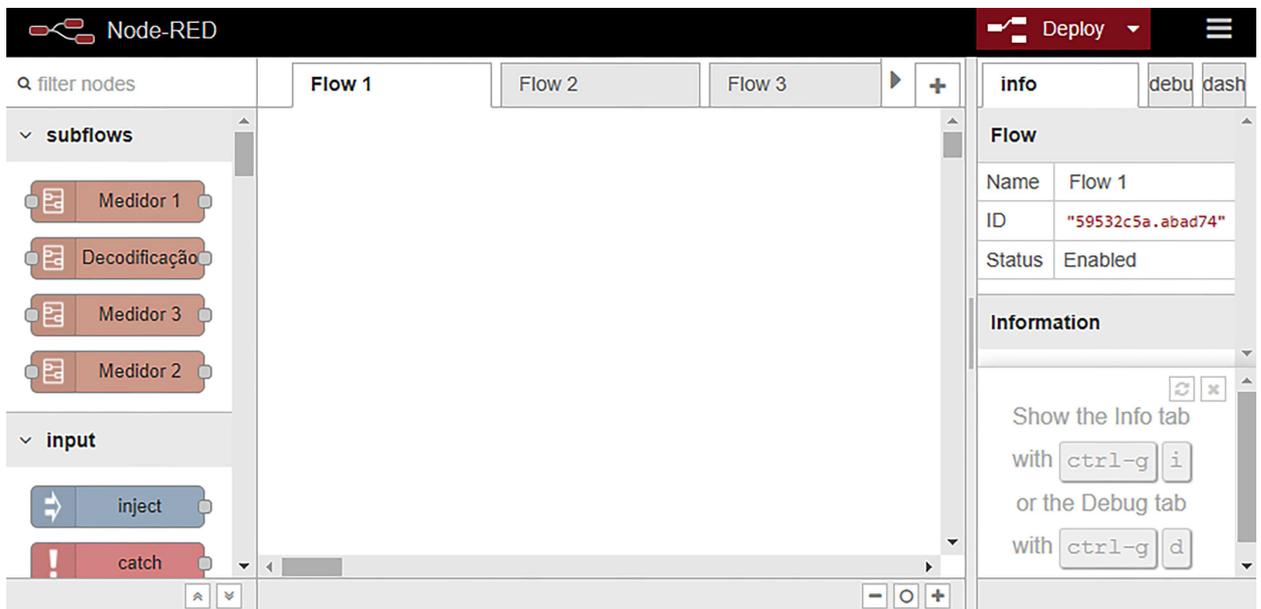


Figura 10: Tela de trabalho principal do Node-RED.

Fonte: Autoria Própria

De modo geral esses nós podem ser divididos em três tipos: os nós de entrada, são aqueles que leem ou recebem valores externos, os nós de transição, de onde os dados são

trabalhados e entregues a outros nós de transição ou de saída e os nós de saída, por onde saem os dados tratados ou não. A partir da combinação desses nós podem ser criados fluxos com variadas funções. Além disso, no caso de não existir alguma função pré-definida, qualquer funcionalidade adicional pode ser inserida em JavaScript.

Para esta aplicação foi criado um fluxo responsável pela coleta, decodificação, tratamento e visualização dos dados. Como os medidores instalados no EV enviam seus dados através do protocolo Modbus TCP, a conexão com os medidores será feita a partir de uma porta ethernet sob o protocolo Modbus TCP, o que implica na elaboração de um fluxo capaz de enviar mensagens nesse padrão.

O Modbus é um protocolo de mensagens da camada de aplicação que possui algumas variações quanto ao meio físico e as mensagens, sendo as mais comuns o Modbus TCP e o Modbus Serial RTU (SIEMENS, 2016). No protocolo Modbus TCP é a pilha TCP/IP que vai proporcionar uma interface para gerenciar conexões e enviar e receber dados, sendo utilizada para a troca de dados em tempo real. A Figura 11 apresenta uma descrição global do uso da pilha TCP/IP pelo protocolo Modbus.

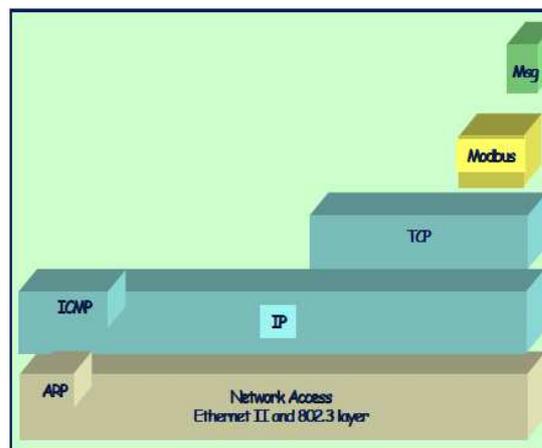


Figura 11: Descrição do uso da pilha TCP/IP pelo protocolo Modbus.

Fonte: Modbus Organization (2004)

Esse protocolo possibilita a comunicação do tipo cliente/servidor entre dispositivos conectados em uma rede ethernet TCP/IP, onde tanto a requisição solicitada pelo cliente quanto a resposta devolvida pelo servidor são encapsulados em frames TCP no formato binário em uma estrutura particionada, como apresentado na Figura 12.

O ADU (*Application Data Unit*) é construído pelo cliente, quem inicia a transação Modbus, sendo que no PDU (*Protocol Data Unit*) estão localizados os dados e código fonte da função. E no MBAPH (*Modbus Application Protocol Header*) que está identificado o ADU, ele

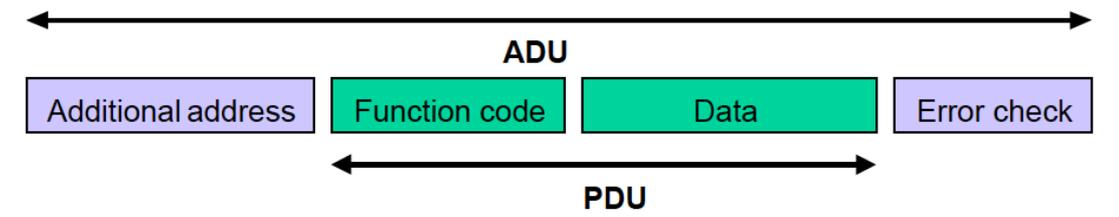


Figura 12: Frame de Requisição/Resposta do Protocolo Modbus

Fonte: Modbus Organization (2004)

é composto por 7 bytes que estão divididos como mostrado na Tabela 2.

Tabela 2: MBAP Header do protocolo Modbus

	Tamanho	Descrição	Cliente	Servidor
Identificador de Transação	2 Bytes	Identificador da operação Requisição/Resposta Modbus	Inicializado pelo cliente	Recopiado pelo servidor da requisição recebida
Identificador de Protocolo	2 Bytes	0 = Protocolo Modbus	Inicializado pelo cliente	Recopiado pelo servidor da requisição recebida
Tamanho	2 Bytes	Contagem dos bytes dos campos subsequentes	Inicializado pelo cliente (requisição)	Inicializado pelo servidor (resposta)
Identificador de Unidade	1 Byte	Identificação do cliente com conexão serial ou outras	Inicializado pelo cliente	Recopiado pelo servidor da requisição recebida

O código da função é codificado em 1 byte que indica para o servidor o tipo de ação que este deve tomar. Existem diferentes códigos de comando no protocolo Modbus, os principais estão apresentados na Tabela 3. Nesta aplicação será utilizado o código 0x03 que lê um número variável de saídas analógicas ou memórias. Esse campo de código da função será utilizado também na resposta do servidor para o cliente para indicar se a resposta é normal ou se ocorreu algum tipo de erro, nesse caso colocando 1 no bit mais significativo.

Tabela 3: Códigos de comando Modbus

Código do Comando Modbus	Descrição
0x01	Lê saídas digitais
0x02	Lê entradas digitais
0x03	Lê saídas analógicas ou memórias
0x04	Lê entradas analógicas
0x05	Altera o estado de uma saída digital
0x06	Altera o estado de uma saída analógica

Nesta aplicação o Raspberry Pi ocupa o lugar de cliente e os Medidores do servidor. O Node-RED vai realizar uma solicitação para os Medidores e estes irão responder com as informações requeridas.

2.2.3 Comparação com Outras Soluções

É importante destacar que no decorrer da pesquisa e da implementação foi possível se defrontar com outras soluções e trajetórias possíveis para resolução do problema proposto. De fato, a proposta inicial deste trabalho tratava do desenvolvimento de um sistema inteligente de medições que utilizasse uma Rede de Sensores Sem Fio (RSSF) para a transmissão de dados. Esta solução se destacava principalmente pelo fato de RSSF serem uma tecnologia bastante abrangente e dinâmica, possuindo a capacidade de monitoramento remoto, realizada através de sensores sem fio, possui flexibilidade de configuração, robustez e dinamismo, características que tornam essa tecnologia uma opção muito interessante para aplicações envolvendo SFRC.

No entanto, a aplicação envolvendo RSSF apresentou algumas desvantagens relacionadas principalmente com o aumento significativo tanto dos custos do sistema, como a aquisição de conversores serial/Ethernet para interface com os sensores, quanto da complexidade da solução. Por conseguinte, apesar de uma solução possível e que atendia os requisitos deste trabalho, optou-se por mudar a solução para um sistema mais simples e mais econômico, que realizasse a transmissão de dados de maneira guiada, ou seja, através de cabos. Esta opção se mostrou mais adequada para o contexto em que este trabalho está inserido, propondo um sistema de monitoramento para o SFRC do EV confiável e de baixo custo.

2.3 SISTEMA DE ARMAZENAMENTO DOS DADOS

2.3.1 Do Raspberry Pi

O Raspberry Pi é um computador desenvolvido para fins acadêmicos do tamanho de um cartão e preço reduzido, aproximadamente \$30 (RASPBERRY PI FOUNDATION, 2017). Além de um processador potente da família ARM, esta CPU possui um conjunto de pinos para entrada e saída de uso geral (GPIO), *Slot* para cartão de memória, porta USB, porta Ethernet, entre outros (RICHARDSON; WALLACE, 2013).

Como esse microcomputador é compatível com sistemas ARMv6, qualquer linguagem que possa ser compilada nesta arquitetura pode ser usada para o desenvolvimento de *softwares*, o que o torna aplicável às mais diversas finalidades. Além disso utiliza como sistema operacional distribuições do Linux, como o *Raspbian* e o *Arch Linux*, o que torna esta tecnologia ainda mais interessante ao trabalhar com uma plataforma de código aberto.

Consideradas as características apontadas o Raspberry Pi foi a tecnologia escolhida para desempenhar um papel central neste trabalho, nele foi instalado o Node-RED e o banco

de dados para o armazenamento dos dados do sistema. Utilizou-se o Raspberry Pi modelo B com 512MB de memória RAM e que abriga processador, entrada ethernet, saída HDMI, portas USB, dentre outros. Visto que o computador não apresenta disco rígido próprio, foi instalado o sistema operacional *Raspbian* em um cartão SD com 16Gb de memória. O modelo utilizado pode ser visualizado na Figura 13.

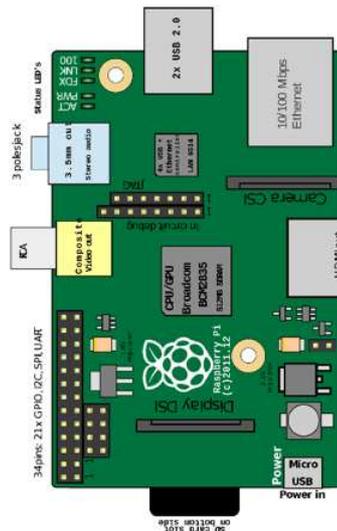


Figura 13: Esquema de um Raspberry PI modelo B.

Fonte: (RASPERRY PI FOUNDATION, 2017)

2.3.2 Do Banco de Dados

No contexto de sistemas inteligentes é substancial a existência de algum mecanismo que possibilite o controle e a disponibilização de informações, principalmente quando considera-se que, quanto mais conexões, mais e novos tipos de dados surgem, aumentando com isso a necessidade de se compreender e atender essa demanda. Sabendo-se disso, vê-se a importância neste projeto da criação de um banco de dados que armazene esses dados e que torne possível a manipulação dos mesmos.

Banco de dados, segundo Date (2000), é um sistema de armazenamento de dados que organiza e relaciona esses dados de forma lógica e coerente, transformando-os em informações. Em outras palavras, é um conjunto de dados organizados, relacionados e com uma finalidade específica. Um sistema de banco de dados é composto pelo banco de dados em si, mais um sistema de gerenciamento (SGBD), que irá tratar todas as solicitações de acesso (DATE, 2000). A Figura 14 ilustra um esquema geral de um sistema de banco de dados.

Um SGBD é basicamente um conjunto de softwares que facilita e possibilita operações

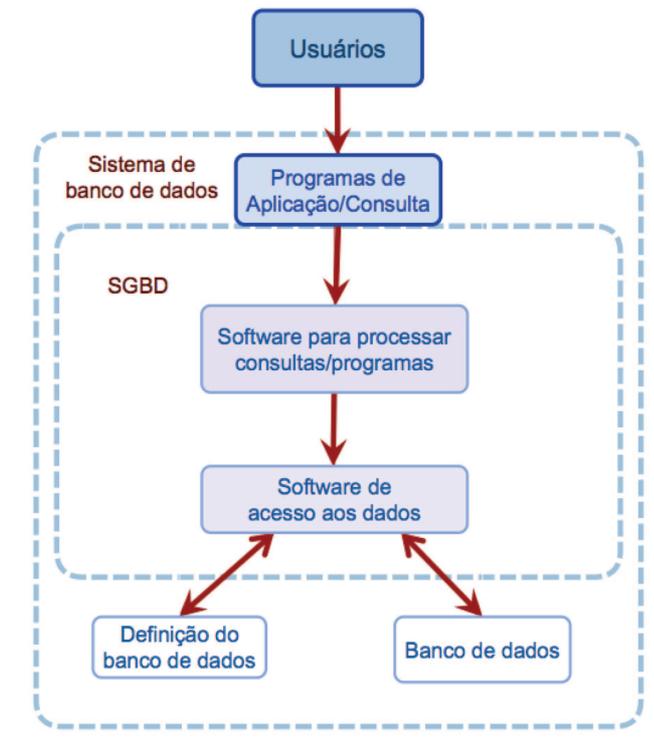


Figura 14: Esquema de um sistema de banco de dados

Fonte: Adaptado de Elmasri e Navathe (2011).

como definição, construção, manipulação, compartilhamento, proteção e manutenção dos dados entre os usuários e aplicações (ELMASRI; NAVATHE, 2011). Além disso, um SGBD disponibiliza uma interface que permite a manipulação dos dados de forma mais prática para o usuário. Um software de gerenciamento que trabalhe com a estrutura relacional recebe o nome de sistema de gerenciamento de banco de dados relacional (SGBDR). A linguagem padrão para acessar sistemas de gerenciamento de banco de dados relacional é o SQL, do inglês *Structured Query Language* ou linguagem de consulta estruturada, usada para manipular, armazenar e recuperar dados a partir de um banco de dados.

O MySQL é um SGBDR relacional que utiliza a linguagem SQL como interface, é um gerenciador de banco de dados *open source* extremamente difundido, isto porque possui consistência, alta performance, confiabilidade, fácil usabilidade, além de, pelo fato de usar a linguagem SQL, ser um gerenciador compatível com a maioria dos sistemas (DATE, 2000).

A escolha neste trabalho de se utilizar o MySQL decorreu principalmente pelo fato de ser um software livre e de código aberto (MYSQL, 2017), além de permitir que os dados sejam acessados remotamente em arquiteturas de servidor/cliente, características que o torna capaz de se adaptar a aplicações de sistemas distribuídos e proporciona maior controle e flexibilidade ao

sistema.

Por fim, outra característica de grande relevância é a presença de ferramentas de segurança de dados, garantindo um banco de dados restrito à pessoas autorizadas, além de ser um gerenciador robusto e possuir tecnologia de controle para múltiplos usuários (MYSQL, 2017).

2.4 VISUALIZAÇÃO DOS DADOS

Foi desenvolvido um *dashboard* capaz de exibir os parâmetros de potência coletados pelos medidores, sendo atualizada continuamente na medida que novos dados são inseridos no banco de dados. Nesta proposta, a interface apresenta os valores de potência monitorados e, a partir desses valores, traça gráficos que facilitam a visualização e possíveis análises desses dados. O *dashboard* foi elaborado no Node-RED para a comunicação com a rede local, propiciando um sistema de armazenamento e visualização dos dados confiável, com baixa possibilidade de perda de dados e que não dependa da intermitência de redes externas.

O *dashboard* do Node-RED é um periférico que permite construir um painel de controle capaz de exibir gráficos, apresentar dados em intervalos de tempo definidos ou em tempo real, além de permitir a manipulação desses dados em uma interface amigável. O Node-RED já fornece um conjunto abrangente de nós para a construção de painéis básicos adequados para sistemas de IoT, disponibilizando gráficos, medidores, controladores, textos, além de possibilitar a confecção de nós e funções personalizadas a partir da linguagem JavaScript (SENSE TECNIC, 2016).

Nesta aplicação, o *dashboard* permite monitorar os dados de potência do sistema e traçar séries temporais a partir dos diversos indicadores de energia coletados e armazenados. A confecção e edição do *dashboard* e a comunicação deste com o banco de dados foram feitas a partir dos nós pré-definidos no programa e funções criadas para o tratamento e exibição dos dados, que foram integrados nos fluxos de armazenamento e coleta dos dados.

3 RESULTADOS

Os resultados da implementação podem ser subdivididos em três grandes áreas: a coleta e tratamento de dados, o armazenamento e a visualização dos dados. Nesses três tópicos citados são abordados aspectos referentes à implementação propriamente dita, além disso são apresentados os custos do sistema implementado e por fim é dedicada uma seção para discussões sobre os resultados obtidos.

3.1 COLETA E TRATAMENTO DOS DADOS

A coleta de dados e o tratamento foram feitas no ambiente de desenvolvimento do Node-RED. Para realizar a comunicação com os medidores foi utilizado o nó pré-programado da paleta de nós do Node-RED, *TCP Request*. Esse nó envia uma requisição TCP através do parâmetro *msg.payload* e aguarda uma resposta, como parâmetros obrigatórios devem ser indicados o endereço IP e a porta ao qual se pretende enviar a requisição.

Para cada medidor, através de um nó de função, foi configurada uma mensagem com o parâmetro *msg.payload* como um *buffer* onde foi montada a requisição para o servidor. Essa requisição segue a estrutura definida pelo protocolo Modbus TCP. Além de conter o MBAPH, este buffer contém o código da função Modbus desejada, neste caso 0x03, o endereço do primeiro registro a ser lido, 1 (que corresponde ao primeiro registro da tensão da fase 1), e a quantidade de registros que devem ser lidos, 56 no caso do PAC3200. A Figura 15 mostra como é montada a mensagem para os medidores do modelo PAC3200.

MBAP Header						FC	1° Registro	N° de Reg.			
0x00	0x01	0x00	0x00	0x00	0x06	0x01	0x03	0x00	0x01	0x00	0x38

Figura 15: Mensagem de Requisição medidores PAC3200.

Fonte: Autoria própria

Como os resultados obtidos a partir do medidor PAC4200 possuem uma diferença em relação aos registros do medidor PAC3200, a requisição para este foi separada em duas partes a fim de que fosse coletada a menor quantidade possível de registros desnecessários. Dessa forma, para o medidor PAC4200, foi feita uma requisição do registro 1 até o 56 e outra do registro 261 até o registro 270.

Visto que para coletar todos os dados ao mesmo tempo o sistema mostrou-se instável e com perdas na conexão, definiu-se pequenos intervalos constantes de 250 ms entre uma requisição e outra. Este problema pode estar relacionado principalmente ao fato de que todas as requisições são direcionadas unicamente para uma porta do Raspberry Pi, a porta 502 que é padrão para comunicação utilizando o protocolo Modbus TCP.

Após enviadas as requisições, as respostas dos servidores são armazenadas em um *buffer* e precisam ser tratadas antes de seguirem para a fase de decodificação dos dados. Para cada medidor foi então elaborado um subfluxo capaz de ler os dados e entregar os registros demandados na requisição. A primeira etapa na fase de tratamento dos dados consiste em remover as partes da mensagem referente ao protocolo Modbus, extrair o dado em si e criar um objeto *msg.topic* para armazenar o modelo do medidor que está enviando a mensagem, este objeto será usado na fase da decodificação.

Através de um nó de função foi desenvolvido um código que seleciona os dados e os envia a um nó que separa cada par de byte do *buffer* em uma nova mensagem. Essas mensagens são direcionadas então a outro nó de função responsável por juntar esses bytes para formar um registro de 16 bits como definido no manual dos medidores. A Figura 16 mostra o diagrama em blocos do subfluxo descrito.

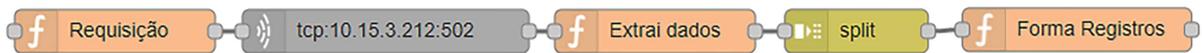


Figura 16: Diagrama subfluxo medidor.

Fonte: Autoria própria

Na fase de decodificação foi construído um subfluxo capaz de decodificar as mensagens tanto dos medidores PAC3200 quanto do Medidor PAC4200, para isso utilizou-se um nó de seleção ou chave, do inglês *switch*, que envia os dados para uma das duas saídas conforme o modelo do medidor.

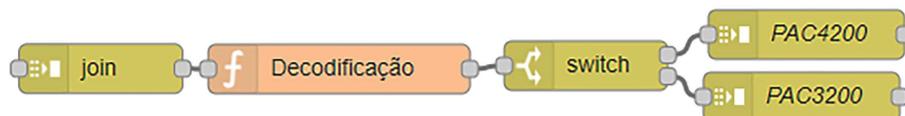


Figura 17: Subfluxo de Decodificação dos dados.

Fonte: Autoria própria

A decodificação propriamente dita é feita utilizando a linguagem JavaScript dentro do

nó de função, neste nó os dois registros referentes a uma grandeza são agrupados e aplicados à equação 1. A Figura 17 mostra o subfluxo desenvolvido para a decodificação dos dados.

3.2 ARMAZENAMENTO DOS DADOS

O banco de dados foi projetado para armazenar localmente as informações de cada medidor de modo a facilitar o acesso, manipulação e consultas aos dados. O gerenciador de banco de dados MySQL foi instalado no Raspberry Pi e a conexão do banco de dados com os medidores foi feita através de fluxos confeccionados no *software* Node-RED como pode ser visto na Figura 18.

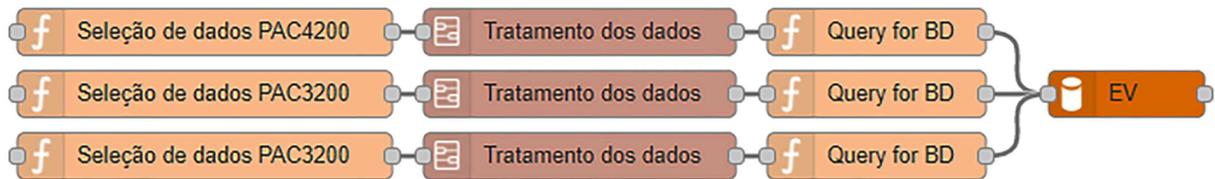


Figura 18: Armazenamento de dados com Node-RED.

Fonte: Autoria própria

Para o sistema de armazenamento de dados deste trabalho foram criadas 3 tabelas no banco de dados do MySQL, uma para cada medidor, e foram gerados campos nas tabelas para cada parâmetro solicitado aos medidores. A estrutura do banco de dados está apresentada no diagrama de entidades da Figura 19. As variáveis selecionadas para esta aplicação já foram descritas na seção 2.2.1, e as grandezas abreviadas no diagrama estão detalhadas na Tabela 1.

Para cada tabela foi definida uma chave primária, o "id", um parâmetro que serve como índice de referência e é responsável por não permitir que valores se repitam em uma mesma tabela, sendo incrementado a cada nova inserção de valores no banco de dados. O campo *timestamp* coleta a data e hora local na Raspberry Pi, armazenando o horário que os dados são recebidos no BD.

Os dados são coletados nos medidores pelo Node-RED a cada 5 segundos, onde é feita a média simples desses valores em um nó de função para o armazenamento no banco de dados de 5 em 5 minutos. Estes valores foram determinados levando-se em conta que diminuindo o tempo de amostragem dos dados se aumenta a capacidade de armazenamento do banco de dados. Além disso, segundo norma regulatória da ANEEL o padrão para o fornecimento de informações para cálculo da curva de carga para fins tarifários e cálculo de perdas deve ser a cada 5 minutos (ANEEL, 2017).

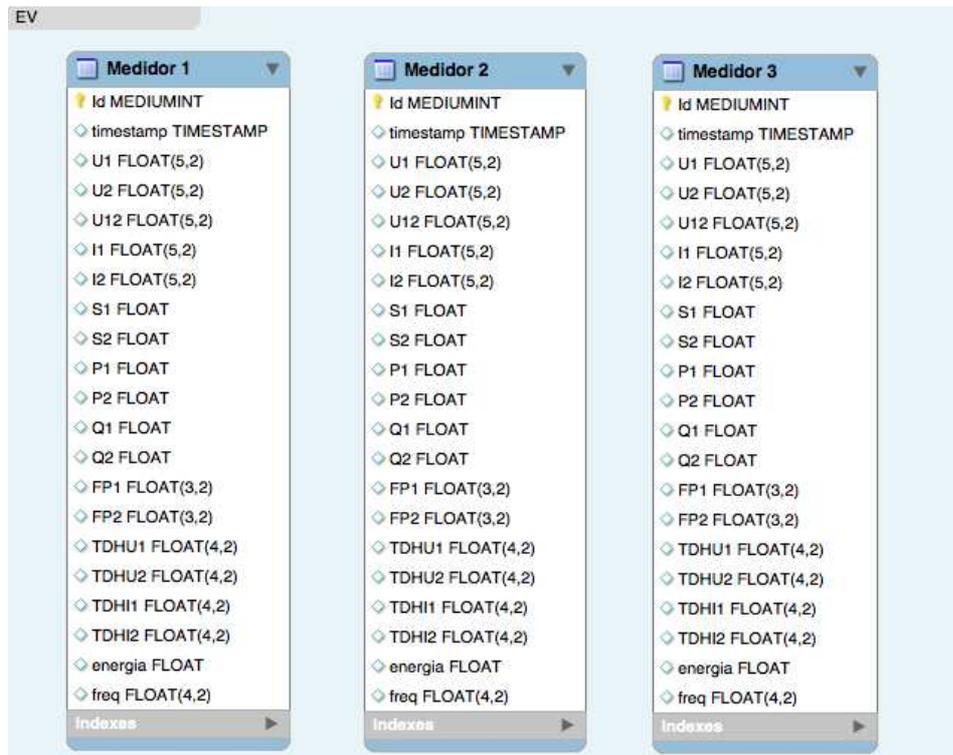


Figura 19: Diagrama de Entidades do Banco de Dados.

Fonte: Autoria própria

O fluxo representado na Figura 18 mostra como é realizada a conexão com o BD e exibe os nós responsáveis pela triagem dos dados, pela formação do *Query* à ser enviado ao BD e o subfluxo onde os dados são tratados para o armazenamento. No subfluxo de tratamento dos dados está implementado o código para a média dos dados no período de 5 min e também o cálculo da energia a partir da potência ativa segundo a regra dos trapézios repetida.

3.3 DASHBOARD LOCAL PARA VISUALIZAÇÃO DOS DADOS

A propósito desse sistema de visualização de dados é disponibilizar as principais informações do SFCR de forma que seja possível uma compreensão global de como se comporta esse sistema fotovoltaico. Para tanto foi desenvolvido um *dashboard* responsável pela exibição local dos dados coletados nos medidores.

A confecção do painel de dados foi realizada através de nós e funções pré definidas do Node-RED. Esse painel foi dividido em 3 páginas, uma para cada medidor, sendo a primeira composta das principais informações da geração do sistema fotovoltaico, a segunda das características do consumo de energia pelo Escritório Verde e a terceira dos dados de potência



Figura 20: Página do consumo.

Fonte: Autoria própria

excedente do sistema, ou seja, a troca com o bloco V. Em cada página são apresentados os parâmetros instantâneos coletados e alguns indicadores gráficos.



Figura 21: Gráfico da potência gerada.

Fonte: Autoria própria

Como a proposta de um *dashboard* é a apresentação das informações mais centrais e pertinentes de um sistema, optou-se por apresentar graficamente somente as informações mais relevantes para a análise do desempenho de um sistema fotovoltaico. Foram selecionadas para serem apresentadas no *dashboard* as curvas de potência ativa gerada, consumida e excedente em Watts ao longo do dia, bem como a tensão de linha em Volts. Esses parâmetros são coletados diretamente do banco de dados e apresentados no *dashboard* através de um nó do tipo *chart* da biblioteca do Node-RED. A Figura 20 mostra a página do *dashboard* referente ao medidor do consumo do EV. Nas figuras 21 e 22 é possível ver , respectivamente, os gráficos da potência

gerada pelo inversor e a troca de potência com o bloco V da UTFPR no dia 30 de novembro de 2017, nesse último, o sinal negativo indica que a potência está sendo absorvida pelo bloco V.

Para a apresentação dos parâmetros instantâneos foi elaborado um fluxo que tem como entrada os dados coletados diretamente dos medidores. Este fluxo vai utilizar os dados logo após estes serem decodificados na fase de coleta e tratamento de dados, e apresenta esses dados através de um nó do tipo texto pré-definido na paleta de nós do *dashboard* do programa. Dessa forma os dados são atualizados assim que cada nova informação é coletada nos medidores, ou seja, de 5 em 5 segundos.



Figura 22: Troca de potência com o Bloco V.

Fonte: Autoria própria

3.4 PREVENÇÃO DE FALHAS

A prevenção das falhas ocorre em basicamente 2 etapas, detecção do erro e execução de uma ação de correção. Por padrão o Node-red possui um nó para detecção de erros nos fluxos, assim quando ocorre algum erro, esta mensagem pode ser utilizada como *trigger*, ou gatilho, para uma ação de solução. No caso de eventuais falhas do sistema, optou-se por reiniciar o sistema inteiro a partir de outro nó pré-instalado do Node-RED, que envia comandos diretamente ao terminal do Linux onde a aplicação está instalada. Dessa forma se alguma falha ocorrer o sistema reiniciará e continuará o processo padrão de execução.

Na Figura 23 pode-se ver o fluxo constituído para tratar os erros. Do ponto de vista de execução do processo de correções de falhas, o fluxo que trata dessa tarefa acontece de forma paralela à execução código principal, independente do resto do processo.

No caso de falta de energia uma falha que pode ocorrer e que não pode ser corrigida

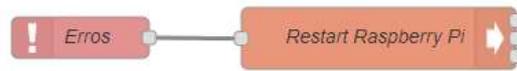


Figura 23: Fluxo para correção de falhas.

Fonte: Autoria própria

pelo sistema de prevenção proposto é a sincronização do horário, fato que pode causar problemas principalmente no que concerne ao registro dos dados no BD, que inclui a data e hora que os mesmos são gravados.

Quando ocorre uma falha deste tipo, assim que a energia for restabelecida o Raspberry Pi tentará se comunicar com a internet para corrigir seu horário, caso não haja conexão com a internet ele permanecerá com o horário errado. Uma solução é implementar um RTC, *Real Time Clock*, para manter o controle do tempo preciso mesmo com o sistema desligado. Também seria possível alimentar o Raspberry Pi através do banco de baterias do sistema fotovoltaico isolado do EV ou a partir de uma bateria power bank.

3.5 CUSTOS DOS EQUIPAMENTOS

Considerou-se importante fornecer um levantamento dos custos envolvidos nesta solução, posto que um dos principais esforços deste trabalho foi fornecer uma solução de baixo custo. A Tabela 4 apresenta uma estimativa aproximada do preço dos principais equipamentos do sistema proposto. Os custos se resumem basicamente a um Raspberry Pi modelo B e um cartão de memória SD de 16Gb. Vale ressaltar que todos os programas e *softwares* instalados e utilizados nesta solução estão disponíveis gratuitamente.

Tabela 4: Custos do Sistema

Equipamento	Quantidade	Preço Unitário (R\$)	Subtotal (R\$)
Raspberry PI modelo B	1	190,00	190,00
Cartão SD - 16 Gb	1	60,00	60,00
			Total = 250,00

3.6 DISCUSSÃO DOS RESULTADOS

Antes do desenvolvimento do código em si, foram necessárias realizar pré-configurações no Raspberry Pi a fim de obter maior segurança e estabilidade. Como configuração inicial foi adicionado à rotina de inicialização do SO (Sistema Operacional) o

comando para iniciar automaticamente o Node-RED. Dentre as ações tomadas para melhorar o desempenho do sistema optou-se, após a realização de alguns testes, pela versão mínima do sistema operacional *Raspbian* que por ser mais simples, sem interface com uma área de trabalho, possibilitou que o sistema fosse executado sem entraves.

Para aumentar a segurança foi adicionada uma senha no usuário padrão do Raspberry Pi, permitindo assim um acesso seguro pelo protocolo SSH ao mesmo. Para o banco de dados foram definidos usuários e senhas de modo que há usuários que têm total acesso ao BD e outros com restrições, com apenas o privilégio de leitura dos dados. O editor do Node-RED por padrão não é seguro, isto é, qualquer um com acesso a rede local poderia acessar o endereço de ip do *host*, no caso o endereço do Raspberry Pi, na porta reservada ao editor, e ter total acesso para fazer alterações no código. Para evitar esse problema foi adicionada uma autenticação através de senha e usuário e criados usuários com permissão total e outros somente com permissão de leitura.

Um ponto crítico para o sistema implementado é a ocorrência de interrupção no fornecimento de energia. Nessa situação o sistema para de funcionar completamente. Durante o reestabelecimento da energia o Raspberry Pi deve estar conectado à rede externa para ser capaz de ajustar a hora e a data e evitar que os dados sejam salvos com esses parâmetros errados no BD. Uma maneira de resolver esse problema seria conectar a alimentação do Raspberry Pi ao banco de baterias do sistema fotovoltaico isolado ou adicionar um relógio RTC, que é capaz de manter o horário atualizado mesmo quando os dispositivos estão desligados.

O Banco de Dados inicialmente deveria salvar os dados fornecidos pelos medidores a cada instante no BD. Contudo, quando realizada uma solicitação para extração dos dados a resposta era muito lenta. Desse modo procurou-se aplicar métodos de otimização na criação do BD visando aumentar a velocidade de recebimento da resposta do servidor. Dentre essas ações está a definição do tamanho de cada variável que seria salva e a limitação do tempo em 5 minutos para cada conjunto de dados salvo, sendo que para uma maior amostragem é necessário buscar outros métodos de otimização para o BD.

As vantagens de se ter um *dashboard* na rede local é que o servidor da aplicação não fica exposto a ameaças externas e este pode ser acessado mesmo quando não existir conexão com a internet. Como a quantidade de dados extraídos é relativamente grande uma aplicação WEB com mais recursos poderia organizar melhor estes dados e permitir maior interação com o usuário, contudo isto exigiria um grau de complexidade em uma área não dominada e fugiria do tema proposto por este trabalho.

4 CONCLUSÕES

A partir da metodologia adotada foi possível atingir os principais objetivos do projeto, obtendo-se no fim um sistema de monitoramento confiável, de baixo custo e baixo consumo de energia, capaz de coletar, armazenar os dados do sistema em um banco de dados, e exibir em um *dashboard* local os parâmetros de potência da geração, consumo e do excedente da produção do Sistema Fotovoltaico Conectado à Rede do Escritório Verde da UTFPR. Para melhor atender aos requisitos do projeto esta aplicação foi dividida em quatro partes principais: um sistemas de coleta, um sistema de armazenamento, um sistema para a visualização dos dados e um sistema para correção e prevenção de erros.

No que se refere a escolha dos elementos do sistema, o Raspberry Pi foi o *hardware* escolhido para assumir um papel central nesta aplicação, nele foi hospedado o banco de dados e instalado o editor visual Node-RED, ferramenta de programação responsável por tratar e decodificar os dados recebidos pelos medidores, além de estabelecer a conexão com o banco de dados. A conexão com os medidores foi feita sob o protocolo Modbus TCP implementado em um fluxo desenvolvido no Node-RED. Foi necessário um processo de decodificação e tratamento desses dados antes do armazenamento. O banco de dados por sua vez foi desenvolvido na plataforma MySQL e foi confeccionado de forma a otimizar o sistema de consulta e armazenamento dos dados, sendo que o seu limite de armazenamento está restrito à capacidade do cartão de memória conectado ao Raspberry Pi. O *dashboard* foi desenvolvido no Node-RED e disponibiliza aos usuários os valores monitorados e gera gráficos a partir desses valores. A fim de aumentar a confiabilidade foi implementado também um sistema de auto-recuperação em caso de erros durante a operação.

Os principais problemas enfrentados decorreram principalmente da grande quantidade de dados a serem armazenados, o que demandou a confecção de um banco de dados mais robusto. Em segundo lugar, a instabilidade da rede de internet do local, o que justificou o desenvolvimento do sistema em uma rede local que garantisse a estabilidade na transmissão dos dados. Um outro ponto crítico que deve ser considerado é a dessincronização do sistema e a possibilidade de perda de dados em caso de falta de energia elétrica.

Levando em conta os parâmetros requeridos pelo sistema, a viabilização de estudos e trabalhos envolvendo geração distribuída e sistema fotovoltaico é imensa. Um exemplo de trabalho que poderia ser realizado é a análise dos impactos da inserção de micro geradores à rede de distribuição. Outra possibilidade de pesquisa é o desenvolvimento de um aplicativo web mais complexo, com acesso à rede externa, tornando os dados mais facilmente acessíveis.

REFERÊNCIAS

- AGENCIA NACIONAL DE ENERGIA ELÉTRICA. **Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST Módulo 5 – Sistemas de Medição**. ANEEL, 2017.
- CEPEL. **Centro de Pesquisas de Energia Elétrica. Energia Solar Fotovoltaica, Estágio Atual e Perspectivas**. 2003. http://www.agg.ufba.br/cier_solarfotovoltaica.pdf. [Online accessed: 19 - Sep - 2017].
- DATE, C. **Introdução a Sistemas de Banco de Dados**. 7a. ed. Rio de Janeiro: Editora Campus, 2000. 900 p. SBN 85-352-1273-6.
- ELMASRI, R.; NAVATHE, S. B. **Sistema de banco de dados**. 6a. ed. São Paulo: Pearson Addison Wesley, 2011. 790 p.
- ENEVO. **Optimising Waste Collection**. 2016. <http://www.enevo.com/>. [Online accessed: 27-September-2016].
- EMPRESA DE PESQUISA ENERGÉTICA - EPE. **Síntese do Relatório Final do Balanço Energético Nacional ano base 2016**. MME, 2016.
- JS FOUNDATION. **Node RED - Flow-based programming for the Internet of Things**. 2017. <https://nodered.org/>. [Online accessed: 01 - Nov -2017].
- LOPES, V. L. d. R.; RUGGIERO, M. A. G. **Cálculo numérico: aspectos teóricos e computacionais**. Sao Paulo: McGraw-Hill, 1988.
- MODBUS ORGANIZATION. **Modbus Messaging on TCP/IP Implementation Guide**. 2004. V1.0a.
- MYSQL. **Manual de referência do MySQL 8.0**. 2017. <https://dev.mysql.com/doc/refman/8.0/en/introduction.html>. [Online accessed: 20 - May -2017].
- NODE.JS FOUNDATION. **About Node.js**. 2017. <https://nodejs.org/en/about/>. [Online accessed: 01 - Nov -2017].
- PERERA, C.; LIU, C. H.; JAYAWARDENA, S.; CHEN, M. A survey on internet of things from industrial market perspective. **IEEE Access**, IEEE, v. 2, p. 1660–1679, 2014.
- PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 16, n. 1, p. 414–454, 2014.
- PINHO, J. T.; GALDINO, M. A. Manual de engenharia para sistemas fotovoltaicos. **Rio de Janeiro: CEPEL**, 2014.

- PVPOWERED. **Performance Monitoring for PV Powered Inverters**. 2009. <http://pvpowered.com.phtemp.com/downloads/PVM1010-Datasheet.pdf>. [Online accessed: 15-November-2017].
- RASPBERRY PI FOUNDATION. **Raspberry PI Foundation**. 2017. <https://www.raspberrypi.org/about/>. [Online accessed: 20 - May -2017].
- RICHARDSON, M.; WALLACE, S. **Primeiros passos com o Raspberry pi**. 1a. ed. São Paulo: Novatec, 2013. 192 p. ISBN : 9788575223451.
- SENSE TECNIC. **Node-RED Programming Guide**. 2016. <http://noderedguide.com/tag/dashboard/>. [Online accessed: 15 - Nov -2017].
- SIEMENS. **Manual multimedidor SENTRON PAC3200**. 2007. Ordernumber: A5E01168664F-01.
- SIEMENS. **Manual multimedidor SENTRON PAC4200**. 2008. Ordernumber: A5E02316180F-01.
- SIEMENS. **Communication and Expansion Modules**. Manual. 2011.
- SIEMENS. **Controladores SIMATIC. Panorama de comunicação MODBUS**. 2016.
- SUNDMAEKER, H.; GUILLEMIN, P.; FRIESS, P.; WOELFFLÉ, S. Vision and challenges for realising the internet of things. **Cluster of European Research Projects on the Internet of Things, European Commision**, 2010.
- URBANETZ, J. Sistemas fotovoltaicos conectados a redes de distribuição urbanas: sua influência na qualidade da energia elétrica e análise dos parâmetros que possam afetar a conectividade. **Universidade Federal de Santa Catarina. Florianópolis**, p. 189, 2010.
- URBANETZ, J.; CASAGRANDE, E. Sistema fotovoltaico conectado à rede elétrica do escritório verde da utfpr. In: **VIII CBPE–Congresso Brasileiro de Planejamento Energético, Curitiba**. 2012.
- ZAIDI, R. Introduction to javascript. In: **JavaScript Essentials for SAP ABAP Developers**. Springer, 2017. p. 1–13.

APÊNDICE A – ACESSO AOS DADOS

Para visualizar os fluxos desenvolvidos e a área de trabalho padrão do Node-RED basta estar conectado na rede local do escritório verde e acessar pelo *browser* de preferência do usuário o endereço IP do Raspberry Pi na porta 1880, como indicado na Figura 24. O *dashboard* para visualização dos dados encontra-se no endereço: 10.15.3.210:1880/ui.

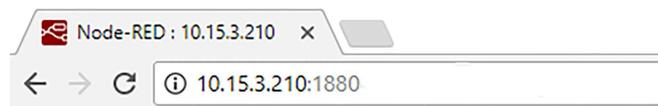


Figura 24: Acesso ao Node-RED.

Fonte: Autoria própria

Quando na ocasião do primeiro acesso será solicitado um usuário e senha como apresentado na Figura 25. Neste caso o *login* é EV-guest e a senha é utfpr. Além deste usuário foi criado um usuário administrador com privilégio total para alterar os fluxos e acessar os dados.

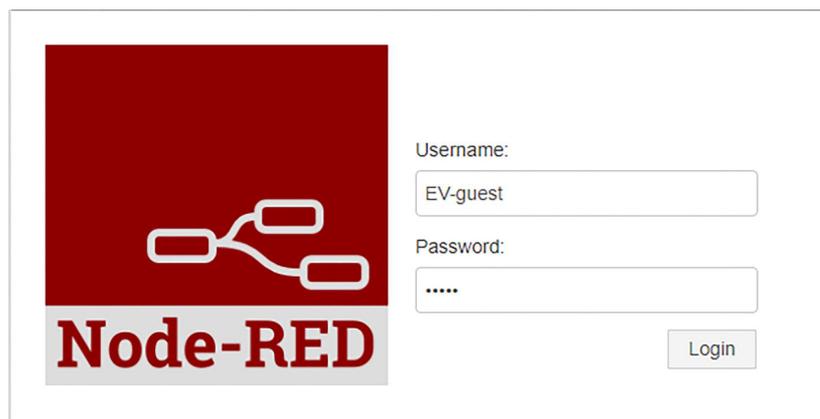


Figura 25: Tela de autenticação.

Fonte: Autoria própria

Para realizar o cadastro de novos usuários, alterar a senha de acesso ou os privilégios de um determinado usuário é necessário o arquivo *settings.js* localizado no diretório de instalação do Node-RED. O acesso deve ser feito através do protocolo SSH, utilizando qualquer programa

de emulação de terminal, como por exemplo o *software* Putty. Na Figura 26 é possível ver a tela de início do Putty já com as configurações para o acesso ao Raspberry Pi pela SSH.

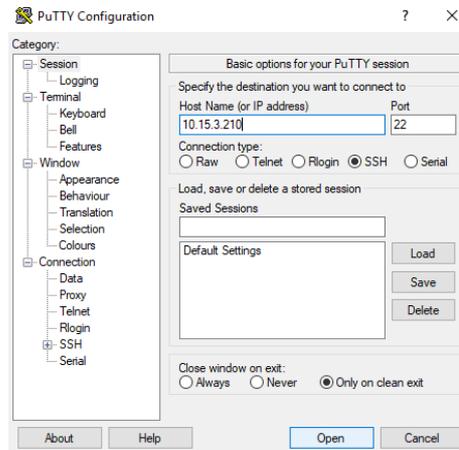


Figura 26: Página inicial Putty.

Fonte: Autoria própria

Realizada a conexão com o Raspberry Pi, deve ser feito o login no usuário *pi*, através da senha *utffprev*. No ambiente terminal do linux acessar o diretório de instalação do Node-RED através do comando `cd ~/.node-red` como indicado na Figura 27.

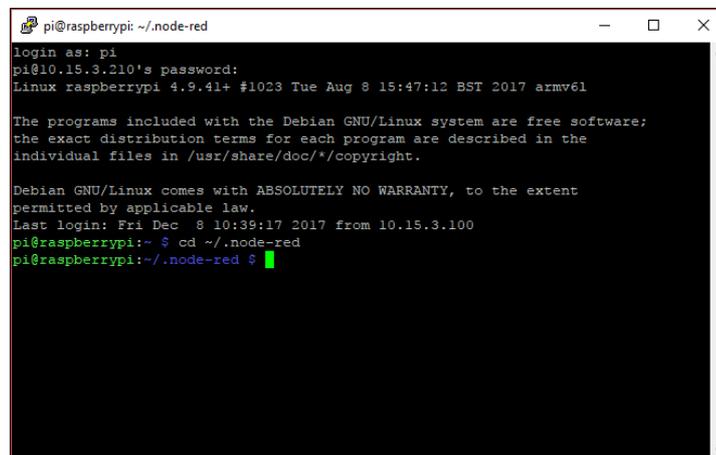


Figura 27: Ambiente terminal do Raspberry Pi.

Fonte: Autoria própria

Em seguida dar o comando `sudo nano settings.js` para editar o arquivo de configuração *settings.js*. Neste arquivo deve serem configurados os usuários, permissões e senhas, como pode ser visto na Figura 28. As senhas utilizam criptografia do tipo Hash. Para gerar uma nova senha basta digitar no terminal `node-red-admin hash-pw`. Será então solicitada uma nova senha

e gerada a senha criptografada, a qual deve ser copiada para o campo correspondente no arquivo *settings.js*.

```
GNU nano 2.7.4 File: settings.js Modified
// -----
// To password protect the Node-RED editor and admin API, the following
// property can be used. See http://nodered.org/docs/security.html for details.
adminAuth: {
  type: "credentials",
  users: [{
    username: "admin",
    password: "$2a$08$zZWtXIja0fBlpzD4sHCMYOCMyz2Z6dNbM6t18sJogENOMcxWV9DNS",
    permissions: "*"
  },
  {
    username: "EV-guest",
    password: "$2a$08$zZWtXIja0fBlpzD4sHCMYOCMyz2Z6dNbM6t18sJogENOMcx$M6S",
    permissions: "read"
  }
  ]
},

// To password protect the node-defined HTTP endpoints (httpNodeRoot), or
// the static content (httpStatic), the following properties can be used.
// The pass field is a bcrypt hash of the password.
// See http://nodered.org/docs/security.html#generating-the-password-hash
httpNodeAuth: {user:"admin",pass:"$2a$08$zZWtXIja0fBlpzD4sHCMYOCMyz2Z6dNbM6t18sJogENOMcxWV9DNS"},
httpNodeAuth: {user:"EV-guest",pass:"$2a$08$zZWtXIja0fBlpzD4sHCMYOCMyz2Z6dNbM6S"}

^G Get Help ^O Write Out ^W Where Is ^H Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^T Uncur Text ^I To Spell ^_ Go To Line
```

Figura 28: Arquivo de configuração do Node-RED.

Fonte: Autoria própria

APÊNDICE B – DOWNLOAD DOS DADOS

Para o download dos dados, o dispositivo utilizado para a transferência dos dados do banco de dados deve estar conectado na mesma rede do Raspberry Pi, ou seja, diretamente no módulo disponibilizado no EV em que tanto o Raspberry Pi quanto os medidores estão conectados.

O download dos dados do sistema pode ser realizado a partir do software MySQL Workbench. Para o download e instalação dessa ferramenta acesse a página do MySQL: <https://dev.mysql.com/>. Em “Developer Zone” vá em “MySQL Downloads” e escolha a opção “MySQL Workbench” como apresentado na Figura 29.

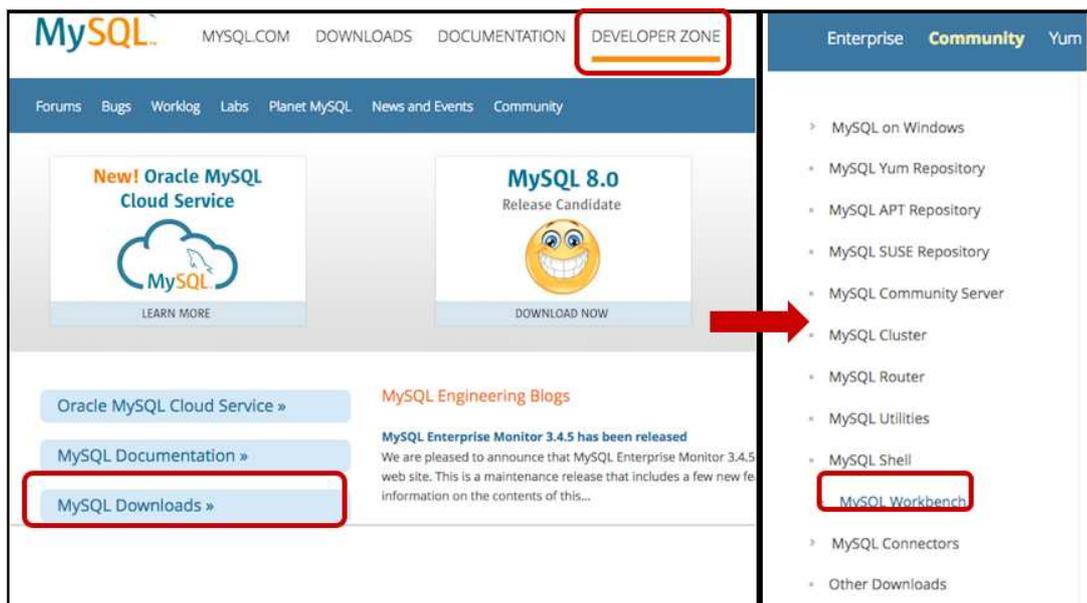


Figura 29: Página de download MySQL Workbench

Fonte: Autoria Própria

Escolha o sistema operacional, faça o download e na sequência instale o programa, como está indicado na Figura 30. Observar que para o caso da versão do MySQL Workbench para Windows é necessário instalar antes alguns programas pré-requisitos. A lista desses programas está disponibilizada na própria página de download.

Após o download e instalação do programa, abra o MySQL Workbench e acrescente uma nova conexão na página inicial do programa. Em “MySQL Connection” clique no símbolo

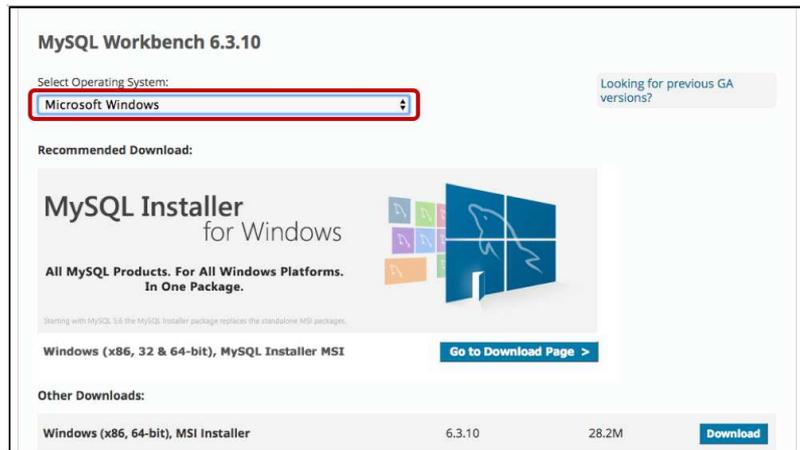


Figura 30: Escolha do sistema operacional e download do MySQL Workbench.

Fonte: Autoria Própria

“+”, na nova janela “Setup New Connection” preencha como apresentado na 31, no espaço “Hostname” digite o IP do Raspberry “10.15.3.210”, em “Port” preencha com “3306”, e o usuário “admin”.

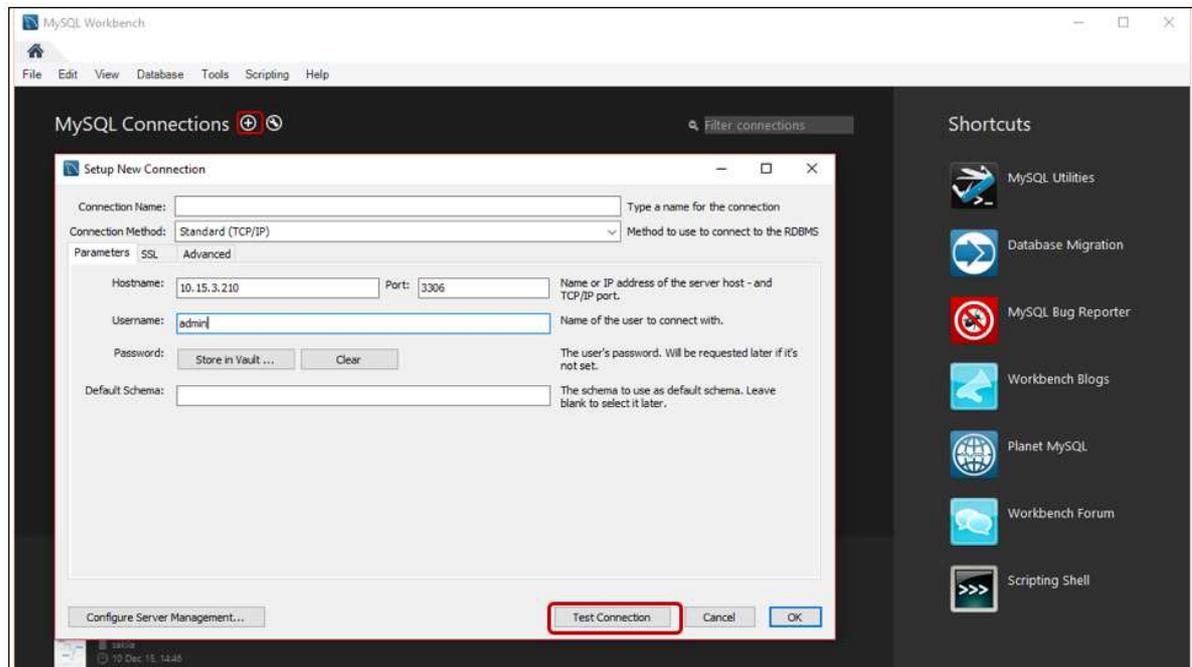


Figura 31: Conexão com o banco de dados na página inicial do MySQL Workbench.

Fonte: Autoria Própria

Em seguida clique em “Test Connection”, uma janela como a apresentada na Figura 32 deverá aparecer solicitando a senha para o acesso ao Banco de dados. Preencha a senha com “escritorioverde” e selecione “Ok”. Caso a conexão não possa ser efetuada, um detalhe

importante que deve ser considerado é o desligamento do Firewall do notebook que em alguns casos pode bloquear este acesso. Em seguida teste novamente a conexão e clique em “OK”.



Figura 32: Solicitação de senha de acesso para o banco de dados.

Fonte: Autoria Própria

Na nova página aberta, vá no menu “Navigator” no canto esquerdo da página. Em “Schemas”, localize e clique no banco de dados “EV”, selecione a tabela que deseja descarregar e clique com o botão direito. Selecione a opção “Table Data Export Wizard”.

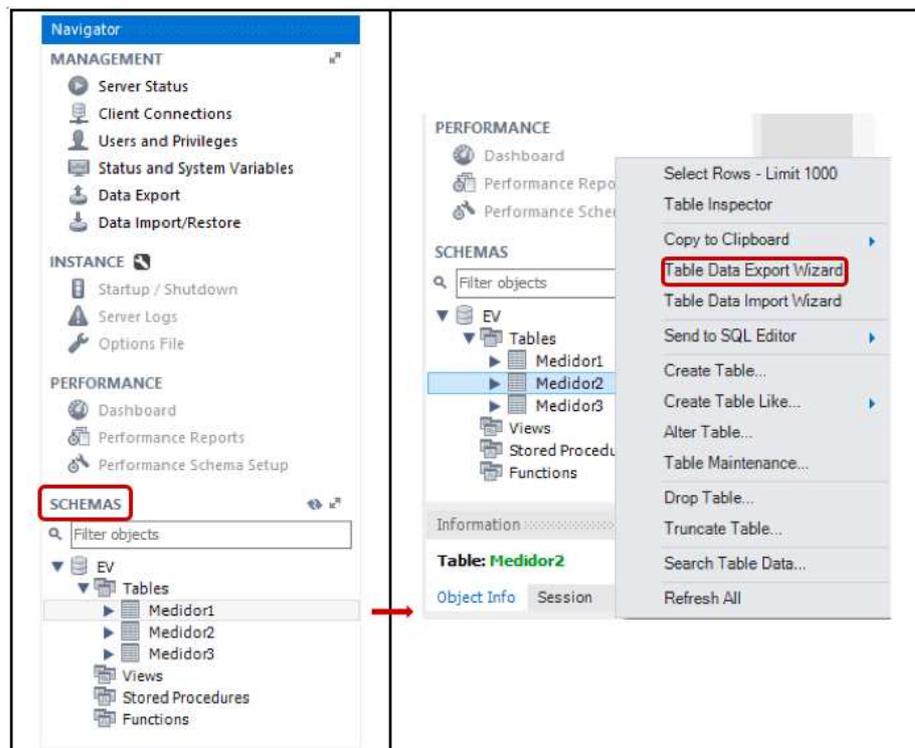


Figura 33: Menu inicial do MySQL Workbench.

Fonte: Autoria Própria

Na página do “Table Data Export”, representada na Figura 34, selecione todas as variáveis que deseja exportar e clique em “Next”.

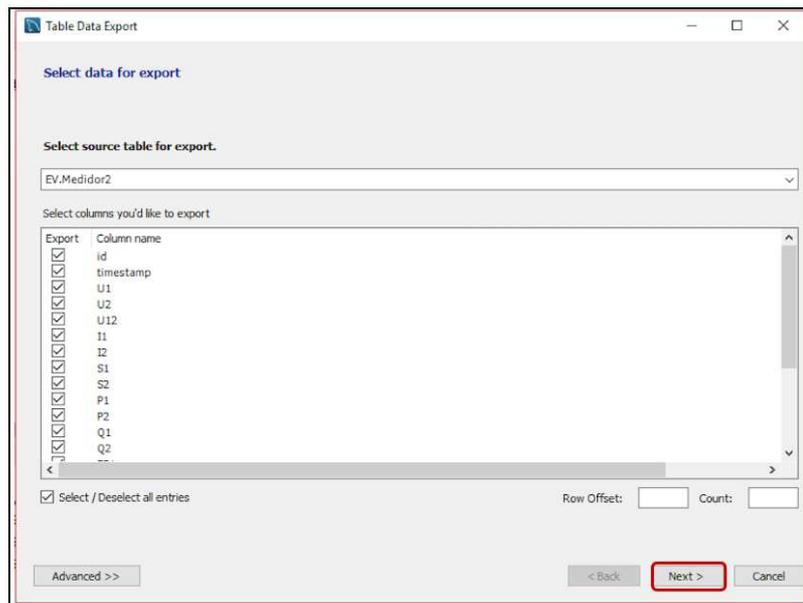


Figura 34: Página de Exportação - Seleção das variáveis.

Fonte: Autoria Própria

Escolha um local para o armazenamento do BD no espaço indicado na Figura 35 e pressione “Next”. Selecione “Next” nas duas próximas páginas e por fim “Finish”. O banco de dados estará salvo no diretório indicado no formato CSV.

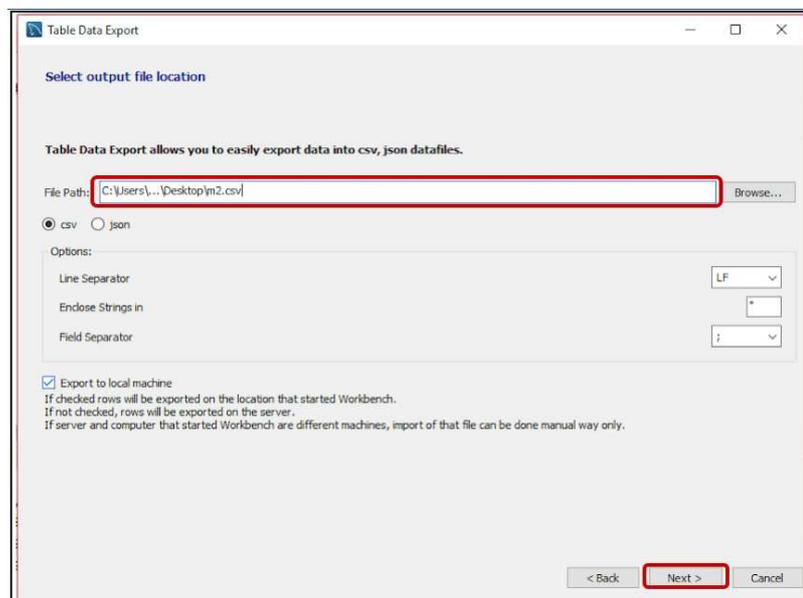


Figura 35: Página de Exportação - Local para armazenamento do BD.

Fonte: Autoria Própria