UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS DE CURITIBA
CURSO DE ENGENHARIA ELÉTRICA

FÁBIO SANSON IHLE

CONTROLE ENERGÉTICO INTELIGENTE DE UM SISTEMA DE AQUECIMENTO SOLAR

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2019

FÁBIO SANSON IHLE

CONTROLE ENERGÉTICO INTELIGENTE DE UM SISTEMA DE AQUECIMENTO SOLAR

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de TCC 2, do curso de Engenharia Elétrica do Departamento Acadêmico de Eletrotécnica (DAELT) da Universidade Tecnológica Federal do Paraná (UTFPR), como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientador (a): Prof. Dr. Adriano Ruseler

CURITIBA

2019

FABIO SANSON IHLE

CONTROLE ENERGÉTICO INTELIGENTE DE UM SISTEMA DE AQUECIMENTO SOLAR

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, 26 de junho de 2019.

_____

Prof. Antonio Carlos Pinho, Dr
Coordenador de Curso
Engenharia Elétrica

_____

Profa. Annemarlen Gehrke Castagna, Ma.
Responsável pelos Trabalhos de Conclusão de Curso
de Engenharia Elétrica do DAELT

ORIENTAÇÃO

_____

Adriano Ruseler, Dr
Universidade Tecnológica Federal do Paraná
Orientador

BANCA EXAMINADORA

_____

Adriano Ruseler, Dr
Universidade Tecnológica Federal do Paraná

_____

Amauri Amorin Assef, Dr
Universidade Tecnológica Federal do Paraná

_____

Winderson Eugenio dos Santos, Dr
Universidade Tecnológica Federal do Paraná

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

RESUMO

IHLE, Fabio Sanson. CONTROLE ENERGÉTICO INTELIGENTE DE UM SISTEMA DE AQUECIMENTO SOLAR. 79 f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

Mundialmente é crescente a busca por soluções para atender a demanda de energia e adequar a matriz energética frente a escassez de recursos fósseis. Os sistemas de aquecimento de água solar desempenham uma importante contribuição para a ascensão da sustentabilidade e geração limpa. Entretanto, existe uma margem para otimização do sistema, tratando-se principalmente da suplementação do aquecimento nos dias em que a insolação é insuficiente. O presente estudo se propôs a desenvolver um sistema de controle inteligente para o sistema de aquecimento solar, a partir da coleta dos valores de temperatura, valores das tarifas das fontes energéticas e atuação via ferramenta de programação visual Node-RED, de modo a garantir o conforto térmico ao banho e proporcionar o uso racional dos recursos energéticos.

Palavras-chave: Aquecimento Solar. Aquecimento Auxiliar. Node-RED. MQTT.

ABSTRACT

IHLE, Fabio Sanson. INTELLIGENT CONTROL OF A SOLAR HEATING SYSTEM. 79f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

Globally is growing the research for solutions to meet the energy demand and to adapt the energy matrix against the scarcity of fossil resources. Solar water heating systems make an important contribution to the rise of sustainability and clean energy generation. However, there is a margin for the system optimization, mainly in supplementation of the heating, on the days when the insolation is not sufficient. The present study proposed to develop an intelligent control system for the solar heating, using solar system temperature information, energy prices and an implementation through the Node-RED visual programming tool, able to guarantee thermal comfort to the bath and to provide rational use of the energy resources.

Keywords: Solar Heating. Auxiliary Heating. Node-RED. MQTT.

LISTA DE FIGURAS

Figura 1: Sistema de aquecimento solar do tipo ativo. 14
Figura 2: Topologia do sistema de controle. 14
Figura 3: Participação das fontes no aquecimento de água para banho nos domicílios. 19
Figura 4: Indicadores sistema de aquecimento solar de água. 19
Figura 5: Circuitos de transferência de energia coletor solar. 20
Figura 6: Conexão Solar + Gás. 22
Figura 7: Esquema de funcionamento de um controlador para automatização do acionamento do sistema de aquecimento auxiliar. 22
Figura 8: Fluxograma de funcionamento sistema proposto. 30
Figura 9: Diagrama de pinos do microcontrolador ESP8266. 36
Figura 10: Estrutura funcionamento protocolo MQTT. 37
Figura 11: Paleta de funções do Node-RED. 38
Figura 12: Armazenamento do valor da temperatura no banco de dados. 39
Figura 13: Operação de consulta ao banco de dados em horário programado. 39
Figura 14: Bloco de configurações. 41
Figura 15: Conjunto blocos de dados de entrada dos sensores. 43
Figura 16: Fluxo de saída em formato JSON. 44
Figura 17: Bloco de entrada e processamento tarifas. 44
Figura 18: Fluxo de saída valor tarifa em JSON. 45
Figura 19: Conexão bloco de função “Custos Fontes” com bloco InfluxDB. 45
Figura 20: Variáveis e funções para cálculo dos custos. 46
Figura 21: Detalhe código para carregamento custos e nomes fontes para array. 46
Figura 22: Definição para as 4 saídas bloco “Custos Fontes”. 47
Figura 23: Visão macro dos blocos para acionamento aquecimento auxiliar. 47
Figura 24: Propriedades *node* “timeswitch”. 48
Figura 25: Detalhamento encaminhamento dos fluxos pelo *node* “switch”. 49
Figura 26: Propriedades *node* “OFF”. 49
Figura 27: Propriedades *node* “Temperatura” 50
Figura 28: Propriedades *node* “Auxiliar”. 50
Figura 29: Detalhe blocos para seleção fonte auxiliar. 51

Figura 30: Propriedades *node* “Operação”. 51
Figura 31: Detalhe *node* para acionamento fonte auxiliar elétrica. 52
Figura 32: Detalhe cálculo tempo acionamento fonte auxiliar. 52
Figura 33: Propriedades *node* “*trigger*”. 53
Figura 34: Representação do bloco de simulação. 54
Figura 35: *Node* “DADOS 2018” – Base histórica temperaturas e tarifas. 54

LISTA DE TABELAS

Tabela 1: Consumo final energético no setor residencial brasileiro em 2012. 18
Tabela 2: Irradiação diária média para o município de Curitiba/PR. 31
Tabela 3: Temperatura média para o município de Curitiba/PR. 32
Tabela 4: Tarifas de referência. 32
Tabela 5: Tarifas de referência. 32
Tabela 6: Energia e excedentes por número de banhos. 34
Tabela 7: Custos por volume de água aquecida. 59

LISTA DE GRÁFICOS

Gráfico 1: Energia térmica para aquecimento água e participação fonte solar. 33
Gráfico 2: Participação fonte auxiliar. 33
Gráfico 3: Custo de operação fontes auxiliares. 34
Gráfico 4: Energia térmica acumulada no período de 2018. 56
Gráfico 5: Temperatura reservatório no período de 2018. 57
Gráfico 6: Energia térmica complementar. 57
Gráfico 7: Custo fontes auxiliares período 2018. 58
Gráfico 8: Custo energia térmica período 2018. 58
Gráfico 9: Custo por volume aquecido. 59

LISTA DE SIGLAS

ANP	Agência Nacional do Petróleo, Gás Natural e Biocombustível
API	Interface de Programação de Aplicações
BEN	Balanço Energético Nacional
COMPAGAS	Companhia Paranaense de Gás
COPEL	Companhia Paranaense de Energia
CPU	Unidade central de processamento
CRESESB	Centro de Referência para Energia Solar e Eólica Sérgio de Salvo Brito
EPE	Empresa de Pesquisa Energética
FBP	*Flow-Based Programming*
GLP	Gás liquefeito de petróleo
GN	Gás natural
GPIO	*General Purpose Input Output*
I^2C	*Inter-Integrated Circuit*
INMET	Instituto Nacional de Meteorologia
IBM	International Business Machines
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
MME	Ministério de Minas e Energia
MQTT	*Message Queuing Telemetry Transport*
NBR	Norma Brasileira
NTC	*Negative Temperature Coefficient*
RAM	*Random Access Memory*
RISC	*Reduced Instruction Set Computer*
RTC	*Real time clock*
SAS	Sistemas de aquecimento solar
SPI	*Serial Peripheral Interface*
TEP	Tonelada equivalente de petróleo
VPS	*Virtual Private Server*

SUMÁRIO

1. INTRODUÇÃO 12
1.1. TEMA 12
1.1.1. Delimitação do Tema 13
1.2. PROBLEMA E PREMISSAS 15
1.3. OBJETIVOS 15
1.3.1. Objetivo Geral 15
1.3.2. Objetivos Específicos 15
1.4. JUSTIFICATIVA 16
1.5. PROCEDIMENTOS METODOLÓGICOS 16
1.6. ESTRUTURA DO TRABALHO 16
2. REVISÃO BIBLIOGRÁFICA 18
2.1. USO DE ENERGIA NAS RESIDÊNCIAS 18
2.2. SISTEMAS DE AQUECIMENTO SOLAR 20
2.2.1. Fontes de aquecimento auxiliar 21
2.2.2. Sistemas de Gerenciamento 22
2.2.3. Consumo Energético 24
3. CONTROLE PROPOSTO 29
3.1. CENÁRIO SIMULADO 30
3.1.1. Fontes auxiliares 32
3.1.2. Volume de água aquecida 34
4. TOPOLOGIA DO SISTEMA 36
4.1. *HARDWARE* ATUADOR 36
4.2. COMUNICAÇÃO 37
4.3. SERVIDOR 37
4.3.1. Node-RED 38

5. IMPLEMENTAÇÃO.....40
5.1. SERVIDOR.....40
5.2. *HARDWARE* ATUADOR.....40
5.3. NODE-RED.....41
5.3.1. Sistema de controle.....41
5.3.2. Simulação.....53
6. RESULTADOS.....56
7. CONCLUSÃO.....60
REFERÊNCIAS.....62
APÊNDICE A.....64
APÊNDICE B.....76

1. INTRODUÇÃO

1.1. TEMA

Mundialmente é crescente a busca por soluções para atender a demanda de energia e adequar a matriz energética frente a escassez de recursos fósseis e sustentabilidade. De acordo com Balanço Energético Nacional (BEN) publicado em 2017 pela Empresa de Pesquisa Energética (EPE), vinculada ao Ministério de Minas e Energia (MME), a participação das fontes renováveis na matriz energética mundial em 2014 foi de 13,5%. No Brasil, para o mesmo período, o índice foi de 39,4%. Neste contexto, a contribuição dos sistemas de aquecimento de água solar possui um papel importante, cooperando de forma renovável com a diversificação da matriz energética.

De acordo com as projeções do estudo de Demanda de Energia 2050, publicado em 2016 pela EPE e MME, ocorrerá uma expansão da fonte solar no aquecimento de água para banho nos domicílios brasileiros, índice que em 2014 foi de 6,0%, para 23,8% em 2050, juntamente com o gás natural, que em 2014 totalizou 4,4% para 47,5% em 2050. Ambas expansões irão contribuir para redução da utilização de energia elétrica para esta finalidade, que em 2014 foi de 83,6% e para 2050 projeta-se 24,2% de participação.

Lima (2003) define que os sistemas de aquecimento solar de água, são compostos por coletores solares, reservatório para armazenar a água quente (*boiler*), fonte auxiliar de aquecimento e rede de distribuição da água aquecida. Lima também classifica os sistemas de acordo com o tipo de circulação do fluido entre os coletores e o reservatório: passivo quando está se dá pela diferença de densidade (termossifão), e ativo caso a circulação da água seja feita por uma bomba. A fonte auxiliar é responsável por complementar o aquecimento em dias de baixa insolação ou que a temperatura da água esteja aquém da desejada. Esta comumente é composta por uma resistência elétrica de imersão dentro do reservatório, ou por circulação através de um aquecedor a gás.

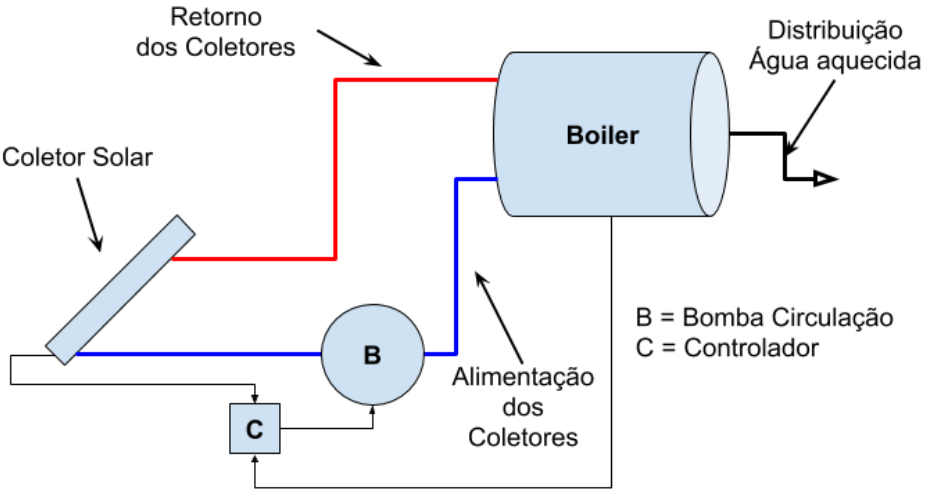
Em sistemas de aquecimento solar de água do tipo ativo, é necessário um controlador para efetuar o acionamento da bomba de circulação. Ayompe (2011) define que o controlador diferencial de temperatura atua acionando a bomba quando a diferença de temperatura entre o reservatório e o coletor atinge um valor previamente configurado. Lima (2003) complementa que o desligamento da bomba ocorre quando a diferença de temperatura se torna pequena ou caso a temperatura configurada para o reservatório seja atingida. Dentre os controladores, alguns modelos oferecem a possibilidade de controle do aquecimento auxiliar, acionando-o em dias e horários configuráveis e, também, proteção anti-congelamento das placas coletoras, recirculando a água entre o reservatório e os coletores.

Sistema de controle pode ser definido como "uma disposição de componentes físicos, conectados ou relacionados de maneira a comandar, dirigir ou regular a si mesmo ou a outros sistemas" (DISTEFANO, 2012, 1). Bauchspiess (2008) caracteriza um sistema como inteligente quando este possui a capacidade de aprendizado ou adaptação em um ambiente desconhecido ou nova situação. O controle energético inteligente pode ser conceituado como a aplicação de um sistema de controle inteligente, atuando no gerenciamento de elementos consumidores de energia dentro de determinado processo, com intuito de minimizar os desperdícios.

1.1.1. Delimitação do Tema

Neste contexto, um sistema de controle inteligente aplicado a aquecimento solar, toma como base sistemas do tipo ativo, conforme esquematizado na Figura 1, substituindo o elemento controlador, baseado no diferencial de temperatura, por um controlador inteligente.

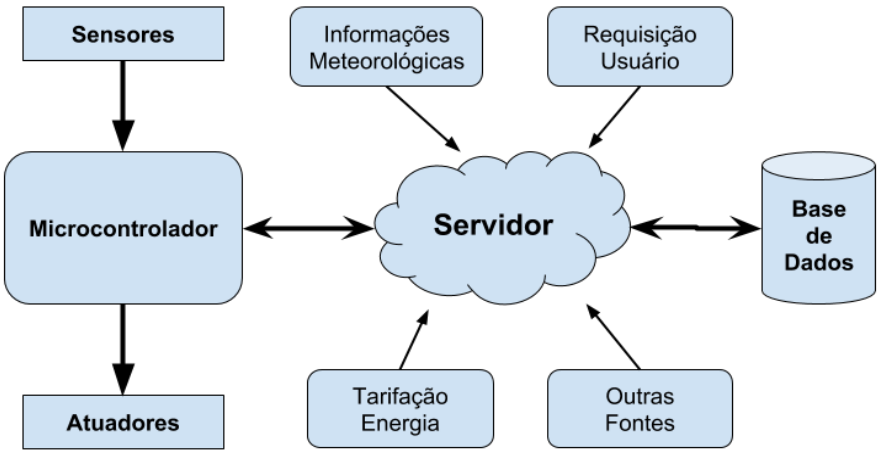
Figura 1: Sistema de aquecimento solar do tipo ativo.



Fonte: Autoria própria.

O sistema de controle inteligente aplicado a aquecimento solar é esquematizado na Figura 2. A parte de comando é composta por sensores para aquisição de dados do aquecedor solar, atuadores para circulação e aquecimento auxiliar, conectados a um microcontrolador que realiza a interface com o servidor. O controle é gerenciado pelo servidor, que recebe dados provenientes de diversas fontes e através do processamento desses, realiza a tomada de decisão.

Figura 2: Topologia do sistema de controle.



Fonte: Autoria própria.

O servidor proposto para o gerenciamento, trata-se de um servidor virtual privado (do inglês *Virtual Private Server* – VPS) executando o sistema operacional de código aberto Ubuntu Server (https://ubuntu.com), em conjunto com a ferramenta Node-RED (https://nodered.org) e banco de dados InfluxDB (https://www.influxdata.com).

1.2. PROBLEMA E PREMISSAS

Os controladores dedicados para sistemas de aquecimento solar existentes no mercado, atuam sobre a bomba de circulação e acionamento do aquecimento auxiliar, com base na temperatura e intervalos de tempo. Decorrente da operação com base em poucas entradas de dados e configurações estáticas, possibilita que se aqueça uma quantidade de água além da necessária para atender a demanda e, desta forma, desperdiça energia.

A premissa deste trabalho é que um sistema de controle inteligente para o sistema de aquecimento solar será capaz de atuar garantindo conforto térmico ao banho e proporcionar o uso racional dos recursos energéticos.

1.3. OBJETIVOS

1.3.1. Objetivo Geral

O presente trabalho tem como objetivo desenvolver um sistema de controle inteligente, para aplicação em sistemas de aquecimento de água por energia solar em ambiente residencial.

1.3.2. Objetivos Específicos

- Levantar as opções de controle para sistemas de aquecimento de água por energia solar disponíveis no mercado;
- Desenvolver um dispositivo para realizar a aquisição de dados e comando;
- Realizar simulação do sistema de controle inteligente;

- Desenvolver e implementar um modelo de controle inteligente através de uma ferramenta de programação baseada em fluxo;
- Avaliar os resultados do sistema implementado com foco no conforto térmico;

1.4. JUSTIFICATIVA

Devido à característica de operação dos controladores para aquecimento solar existentes no mercado, possivelmente se aquece uma quantidade de água maior que a necessária e, desta forma, desperdiça recursos energéticos. O presente trabalho visa o desenvolvimento de um sistema de controle inteligente que propicie o uso racional dos recursos energéticos, com a mínima necessidade de intervenção do usuário.

1.5. PROCEDIMENTOS METODOLÓGICOS

Para execução do presente trabalho, foi realizada uma revisão bibliográfica sobre sistemas de aquecimento solar e estratégias de controle. Posteriormente, realizou-se um estudo teórico para determinar o modelo de controle inteligente a ser aplicado. Determinado o modelo, realizou-se a etapa da simulação para mensurar o desempenho do sistema, e implementação com a prototipagem do sistema.

1.6. ESTRUTURA DO TRABALHO

O presente trabalho será composto por 7 capítulos.

O capítulo 1 apresenta e delimita o tema, estabelecendo os objetivos gerais e específicos, bem como a justificativa para sua execução e a metodologia a ser executada.

O capítulo 2 aborda a fundamentação teórica, buscando trazer o os conceitos de sistemas de aquecimento solar, suas fontes de aquecimento auxiliar, sistemas de controle e seu modelo energético.

No capítulo 3 será apresentado o sistema proposto para o controle.

No capítulo 4 serão apresentadas as ferramentas propostas para implementação do sistema.

No capítulo 5 será apresentada a implementação do sistema de controle utilizando a plataforma Node-RED e detalhamento dos *softwares* auxiliares.

No capítulo 6 serão apresentados e discutidos os resultados obtidos a partir da simulação com a implementação do modelo.

No capítulo 7 serão apresentadas as conclusões.

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo é realizada uma breve apresentação do uso de energia nas edificações residenciais brasileiras, do sistema de aquecimento solar, fontes auxiliares e sistemas de gerenciamento, bem como do equacionamento necessário para modelar o consumo energético do sistema.

2.1. USO DE ENERGIA NAS RESIDÊNCIAS

De acordo com EPE (2013), no âmbito residencial brasileiro as fontes energéticas mais relevantes são a eletricidade, o gás liquefeito de petróleo (GLP) e a lenha, conforme Tabela 1.

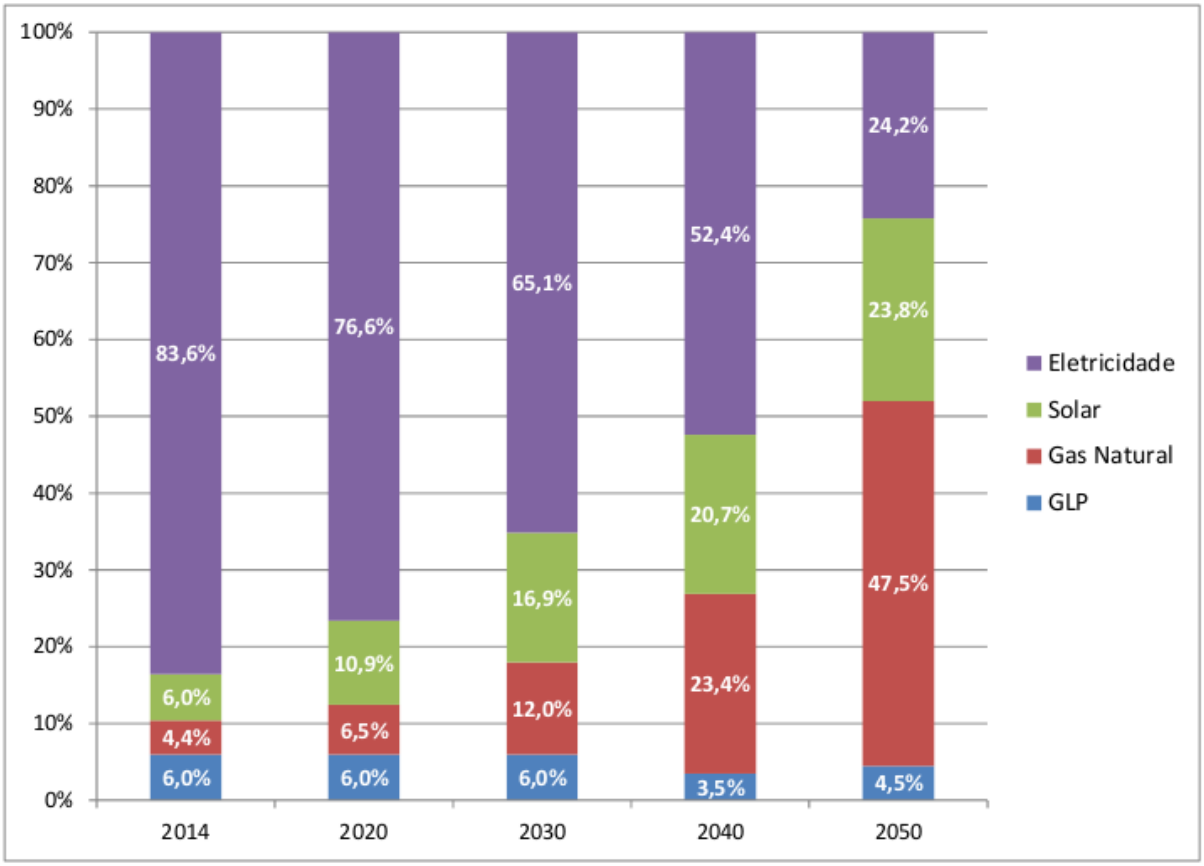
Tabela 1: Consumo final energético no setor residencial brasileiro em 2012.

Fonte	**Consumo (10^3 tep – tonelada equiv. de petróleo)**	**%**
Eletricidade	10118	42,6
Lenha	6472	27,2
GLP	6393	27
Carvão vegetal	478	2
Gás natural	295,5	1,2
Querosene	4,5	0
Total	**23761**	**100**

Fonte: EPE (2013).

De acordo com a nota técnica DEA 13-15 do EPE (2015), tratando-se de aquecimento da água para banho nos domicílios, em 2014 a fonte energética mais relevante foi a eletricidade. Para 2050, conforme ilustrado na Figura 3, projeta-se uma maior penetração do gás natural e da energia solar, reduzindo a participação da eletricidade e GLP.

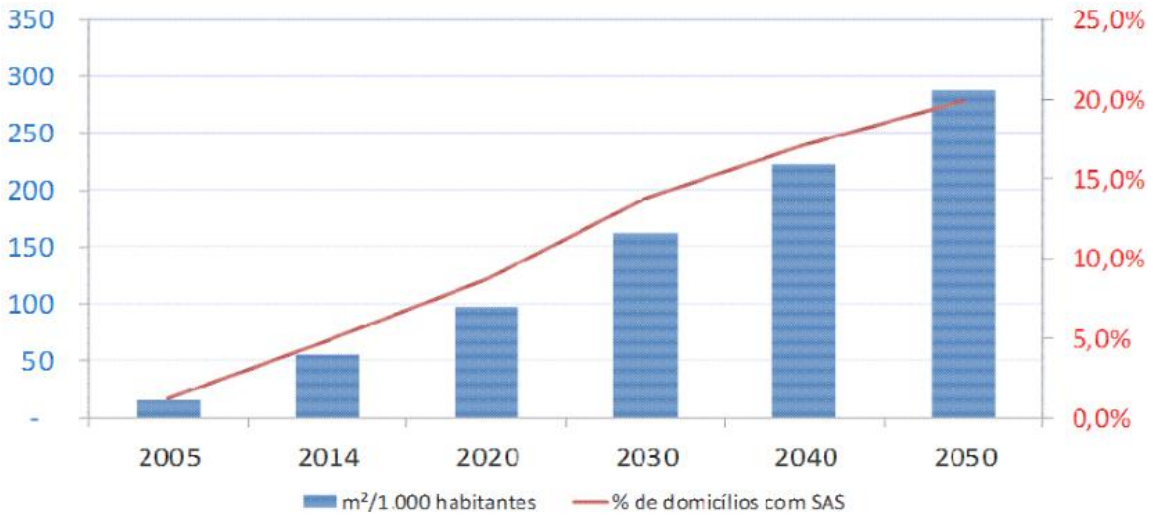
Figura 3: Participação das fontes no aquecimento de água para banho nos domicílios.



Fonte: EPE (2015).

Neste mesmo estudo, o EPE estima que a expansão dos sistemas de aquecimento solar (SAS) continue mantendo os níveis de crescimento atual, conforme ilustrado na Figura 4.

Figura 4: Indicadores sistema de aquecimento solar de água.



Fonte: Elaboração EPE (2015).

2.2. SISTEMAS DE AQUECIMENTO SOLAR

Segundo a NBR 15747:2009, sistema de aquecimento solar (SAS) é um "sistema composto de coletor solar e outros componentes para fornecimento de energia térmica".

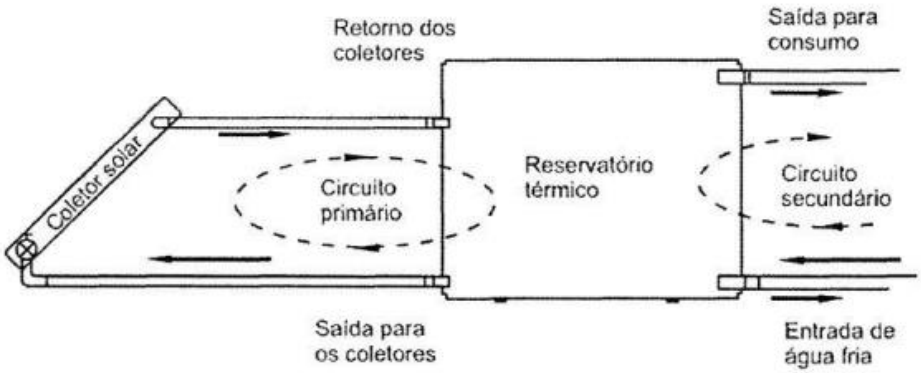
A NBR 15569:2008 classifica como principais elementos de um SAS:

a) coletor(es) solar(es);
b) reservatório térmico;
c) sistema de aquecimento auxiliar.

A norma também o divide, conforme a Figura 5, em dois circuitos de transferência de energia:

a) primário (entre coletores e armazenamento);
b) secundário (abastecimento e distribuição da água).

Figura 5: Circuitos de transferência de energia coletor solar.



Fonte: NBR 15569:2008.

Siqueira (2009) define um coletor solar como o elemento que "promove a conversão da radiação solar, transferindo o fluxo energético proveniente da radiação incidente para o fluído que circula no interior do mesmo".

Reservatório térmico, comumente chamado de *boiler*, é o elemento responsável por armazenar a energia captada. De acordo com a NBR 15569:2008, seu uso se faz necessário no âmbito residencial "em função da não simultaneidade entre consumo e disponibilidade de energia solar".

O sistema de aquecimento auxiliar atua como fonte alternativa de calor para períodos de menor insolação ou de maior consumo. De acordo com Lima (2003), a fonte auxiliar pode ser elétrica, a gás ou gerada por uma bomba de calor.

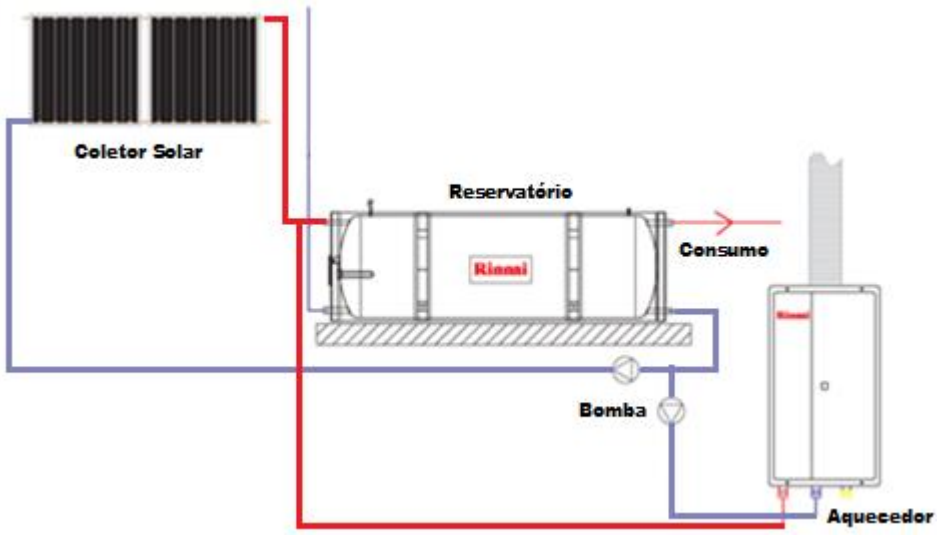
2.2.1. Fontes de aquecimento auxiliar

Segundo Lima (2003), o "sistema de aquecimento solar da água não é projetado para fornecer 100% da demanda da água quente". Portanto, para dias em que a insolação se faz insuficiente, torna-se necessária a utilização de uma fonte de aquecimento auxiliar para suprir a demanda de energia térmica.

A NBR 15569:2008 estabelece que a especificação do sistema de aquecimento auxiliar deve considerar prioridade a fonte solar e a influência que este possa causar no desempenho do SAS. Sua utilização pode ser feita em série ou paralelo com o reservatório térmico.

De acordo Neves (2013), a resistência elétrica imersa no reservatório é a fonte de aquecimento auxiliar mais usual. Alternativamente, pode-se utilizar um aquecedor a gás de passagem, pelo qual ocorrerá a recirculação da água do reservatório conforme esquematizado na Figura 6.

Figura 6: Conexão Solar + Gás.



Fonte: Adaptado de Rinnai (2017).

2.2.2. Sistemas de Gerenciamento

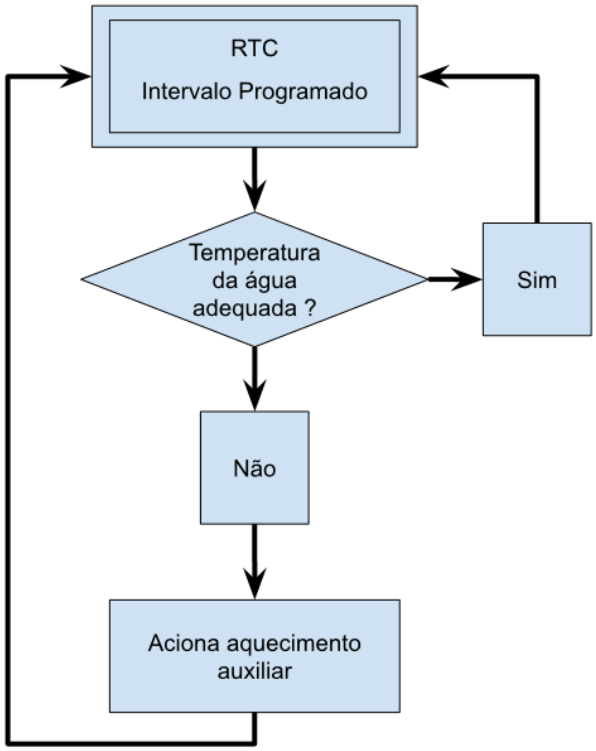
A partir de pesquisa em *sites* de fabricantes de sistemas de aquecimento solar, verificou-se que os controladores voltados para aplicação residencial disponíveis no mercado, têm como principais funcionalidades oferecidas:

- Controle da circulação da água entre os coletores solares e o reservatório térmico (aplicável à circulação forçada);
- Automatização do acionamento do sistema de aquecimento auxiliar.

Para ambas as funcionalidades os controladores permitem programar intervalos de tempo e dias da semana (eventos) nos quais as funções serão executadas, bem como os ajustes de temperatura para acionamento. As temperaturas são coletadas através de sensores do tipo NTC, e os acionamentos realizados através de saídas à relé.

A função de automatização de acionamento do sistema de aquecimento auxiliar pode ser esquematizada conforme a Figura 7. A operação tem início nos intervalos programados, verificados a partir do relógio de tempo real (RTC).

Figura 7: Esquema de funcionamento de um controlador para automatização do acionamento do sistema de aquecimento auxiliar.



Fonte: Autoria própria.

2.2.3. CONSUMO ENERGÉTICO

Nessa seção serão apresentados os equacionamentos necessários para modelar sob aspecto energético o funcionamento do sistema de aquecimento solar e suas fontes auxiliares, que servirão de base para o sistema de controle.

2.2.3.1. Calorimetria

Dada a equação fundamental da calorimetria, pode-se definir a capacidade térmica como (Bertoldo, 2009):

$$C = \frac{Q}{\Delta T} \tag{1}$$

Sendo:
C é a capacidade térmica, expressa em joule por kelvin (J/K) ou joule por graus Celsius (J/°C);
Q é a quantidade de calor, expressa em joule (J). Usualmente é expressa em caloria (cal);
ΔT é a variação de temperatura, expressa em kelvin (K). Usualmente também é expressa graus Celsius (°C);

O calor específico pode ser equacionado como (Bertoldo, 2009):

$$c = \frac{C}{m} \tag{2}$$

Sendo:
c é o calor especifico, expresso em Joules por quilograma e Kelvin (J / kg.K) ou Joules por quilograma e graus Celsius (J / kg.°C), usualmente é expressa em caloria (cal / g.°C);
m é a massa, expressa em quilogramas (kg).

Substituindo-se a Equação (2) na Equação (1), tem se:

$$Q = m \times c \times \Delta T \quad (3)$$

Sabendo que a relação entre massa e volume pode ser equacionada como:

$$m = \rho \times V \quad (4)$$

Sendo:
ρ é a densidade, expressa em quilogramas por litro (kg/l),
V é o volume, expresso em litros (l).

Substituindo-se a Equação (4) na Equação (3), tem-se:

$$Q = \rho \times V \times c \times \Delta T \quad (5)$$

Considerando-se a densidade da água igual a 1 kg/l, calor especifico igual a 4186 J/kg.°C (NBR15569, 2008), e a relação 1 Wh = 3600 Joule, substituindo-se os valores na Equação (5) calcula-se Et, que representa a energia térmica expressa em Watt-hora (Wh):

$$E_T = \frac{1 \times V \times 4186 \times \Delta T}{3600} \quad (6)$$

A partir da Equação (6) pode-se calcular a energia térmica necessária para variação da temperatura (ΔT) de um determinado volume de água (V).

2.2.3.2. Fonte solar

De acordo com a NBR15569, o cálculo de demanda de energia útil pode ser equacionado por:

$$E_{útil} = \frac{V_{armaz} \times \rho \times C_p \times (T_{armaz} - T_{ambiente})}{3600} \quad (7)$$

Sendo:

$E_{útil}$ é a energia útil, expressa em quilowatts hora por dia (kWh/dia);

V_{armaz} é o volume do sistema de armazenamento;

ρ é a massa específica da água igual a 1000, expressa em quilogramas por metros cúbicos (kg/m³);

C_p é o calor específico da água igual a 4,18, expresso em quilojoules por quilograma Kelvin (kJ/kg);

T_{armaz} é a temperatura de armazenamento da água, expressa em graus Celsius (°C);

$T_{ambiente}$ é a temperatura ambiente média, expressa em graus Celsius (°C).

De acordo com a NBR15569, o cálculo da área coletora:

$$A_{coletora} = \frac{(E_{útil} - E_{perdas}) \times FC_{instal} \times 4{,}901}{PMDEE \times I_G} \quad (8)$$

Sendo:

$A_{coletora}$ é a área coletora, expressa em metros quadrados (m²);

I_G é o valor da irradiação global média para o local de instalação, expresso em quilowatts hora por metro quadrado dia (kWh/m². dia);

FC_{instal} é o fator de correção para inclinação e orientação do coletor solar (adimensional);

$E_{útil}$ é a energia útil, expressa em quilowatts hora por dia (kWh/dia);

E_{perdas} é o somatório das perdas térmicas dos circuitos primário e secundário, expresso em quilowatts hora por dia (kWh/dia), calculada pela soma das perdas ou pela Equação (9):

$$E_{perdas} = 0{,}15 \times E_{útil} \quad (9)$$

$PMDEE$ é a produção média diária de energia específica do coletor solar, expressa em quilowatts hora por metro quadrado (kWh/m²), calculada através da Equação (10):

$$PMDEE = 4{,}901 \times (Fr\tau\alpha n - 0{,}0249 \times Fr_{UL}) \quad (10)$$

Sendo:

$Fr\tau\alpha n$ é o coeficiente de ganho do coletor solar (adimensional);

Fr_{UL} é o coeficiente de perdas do coletor solar (adimensional);

A partir da Equação (8) substituindo-se as variáveis expressas na Equações (9) e (10), obtém-se a energia útil gerada pelo coletor:

$$E_{\text{útil}} = \frac{A_{coletora} \times (Fr\tau\alpha - 0{,}0249 \times Fr_{UL}) \times I_G}{1{,}15 \times FC_{instal}} \quad (11)$$

O rendimento pode ser equacionado por:

$$\eta = \frac{E_{\text{útil}}}{E_{fornecida}} \quad (12)$$

2.2.3.3. Fonte auxiliar elétrica

Para fonte auxiliar à energia elétrica (resistência de imersão), a quantidade de energia consumida pode ser calculada diretamente a partir da Equação (12). O custo de operação da fonte é dado pela multiplicação entre quantidade de energia e tarifa, conforme Equação (13).

$$\text{Custo} = E_{fornecida} \times TarifaE \quad (13)$$

Sendo:

Custo é o valor monetário, expresso em Real brasileiro (R$);

$E_{fornecida}$ é a quantidade de energia a ser fornecida, expressa em quilowatts hora (kWh);

$TarifaE$ é o custo cobrado por quantidade de energia elétrica, expressa em Real brasileiro por quilowatts hora (R$/kWh).

2.2.3.4. Fonte auxiliar a gás

Para fonte auxiliar a gás natural (GN) ou gás liquefeito de petróleo (GLP),

calcula-se a quantidade de energia necessária a partir da Equação (12). O custo de operação então é calculado utilizando-se da potência do equipamento, consumo e tarifas, conforme Equação (14) para GN e Equação (15) para GLP.

$$\text{Custo} = \frac{E_{fornecida}}{Pot} \times ConsumoGN \times TarifaGN \quad (14)$$

Sendo:

Pot é a potência da fonte auxiliar, expressa em quilowatt (kW);

$ConsumoGN$ é o consumo de gás natural, expresso em metros cúbicos por hora (m^3/h);

$TarifaGN$ é custo cobrado por quantidade de gás, expresso em Real brasileiro por metros cúbicos ($R\$/m^3$).

$$\text{Custo} = \frac{E_{fornecida}}{Pot} \times ConsumoGLP \times TarifaGLP \quad (15)$$

Sendo:

$ConsumoGLP$ é o consumo de gás liquefeito de petróleo, expresso em quilograma por hora (kg/h);

$TarifaGLP$ é custo cobrado por quantidade de gás, expresso em Real brasileiro por quilograma (R$/kg).

3. CONTROLE PROPOSTO

O sistema de controle proposto tem sua configuração base (instalação), a partir dos parâmetros do reservatório e fontes auxiliares:

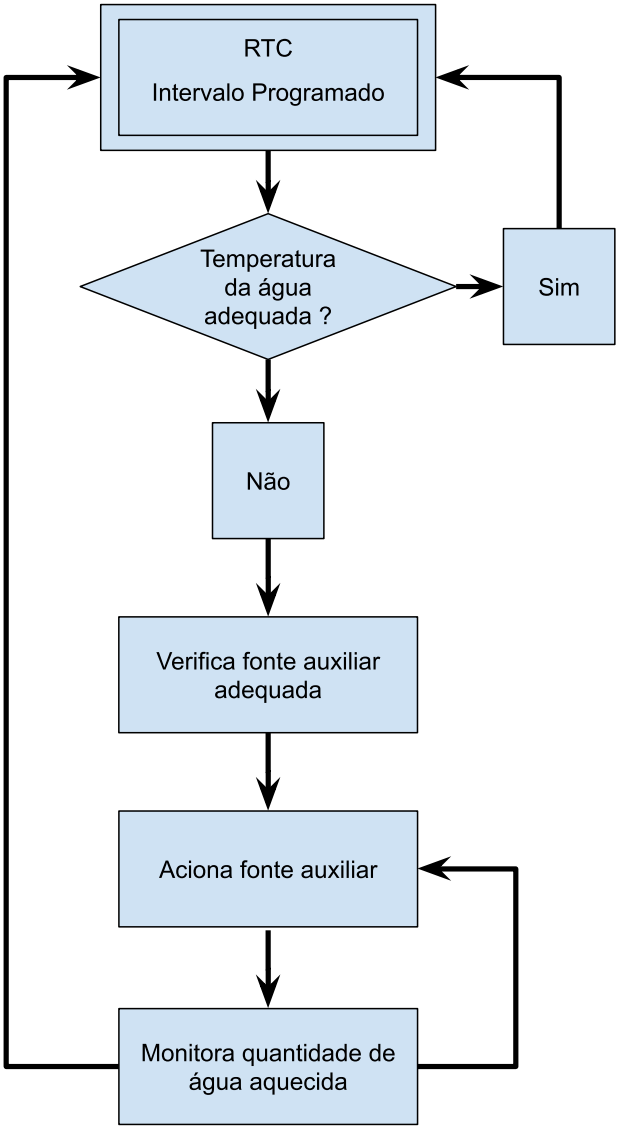
- Capacidade;
- Potência;
- Rendimento;
- Consumo;
- Limites operacionais.

Além disso, são empregados dados provenientes do usuário ou outras bases:

- Tarifas;
- Ajuste de temperatura;
- Perfil de consumo.

No momento em que se faça necessária a utilização de fonte auxiliar para complementar o aquecimento da água, o sistema realizará a tomada de decisão para definir a fonte mais adequada e monitorar a quantidade de água a ser aquecida, conforme ilustrado na Figura 8.

Figura 8: Fluxograma de funcionamento sistema proposto.



Fonte: Autoria própria.

3.1. CENÁRIO SIMULADO

Para estimar o funcionamento do sistema, considerou-se como cenário base um SAS dimensionado para uma residência localizada na cidade de Curitiba/PR com as seguintes premissas:

- Quatro moradores;
- Duas duchas e duas torneiras;
- Temperatura de armazenamento da água no reservatório térmico: 60ºC;
- Temperatura da água no ponto de consumo: 40ºC;

- Posição dos coletores solares sem desvios em relação ao Norte Geográfico;
- Coletores solares instalados em sua inclinação ideal.

Em valores comerciais, o SAS base possui as seguintes características:

- Quatro coletores com área de 1,5 m²/coletor (classificação PBE: A);
- Um reservatório térmico com 400 litros, com apoio elétrico de 2,5 kW integrado (resistência de imersão).

Para o sistema de aquecimento auxiliar a gás, fez-se as seguintes considerações:

- Aquecedor de água a gás do tipo instantâneo (classificação PBE: A);
- Capacidade de vazão: 8 litros/min;
- Potência: 13,5 kW;
- Rendimento: 85% (GN) / 84% (GLP);
- Consumo máximo de gás: 1,22 m³/h (GN) / 0,98 kg/h (GLP).

Para o cálculo da participação da fonte solar, utilizou-se os valores irradiação solar diária média no plano horizontal, apresentados na Tabela 2, com base nos dados do programa SunData versão 3 do Centro de Referência para Energia Solar e Eólica Sérgio de Salvo Brito (CRESESB) - Coordenada geográfica de referência: latitude 25,5° S e longitude 49,249° O.

Tabela 2: Irradiação diária média para o município de Curitiba/PR.

Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Irradiação diária média (kWh / m².dia)	5,34	5,15	4,5	3,79	3,06	2,75	2,9	3,85	3,84	4,41	5,26	5,56

Fonte: CRESESB (2018).

Para os valores de condição inicial (temperatura da água), foi utilizada a temperatura média para o município de Curitiba/PR, a partir de dados do normal climatológico do Brasil 1981-2010 do Instituto Nacional de Meteorologia (INMET),

apresentados na Tabela 3.

Tabela 3: Temperatura média para o município de Curitiba/PR.

Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Temperatura média (ºC)	20,9	21	20,1	18,3	15,1	13,9	13,5	14,6	15,3	17,1	18,9	20,2

Fonte: INMET (2018).

3.1.1. Fontes auxiliares

Para os custos de operação de cada uma das fontes auxiliares, utilizou-se como referência as tarifas do período de set/17 até ago/18, conforme Tabela 4 e Tabela 5. Para energia elétrica, utilizou-se os dados da Companhia Paranaense de Energia (COPEL) referentes à Tarifa Convencional - Subgrupo B1. Para o GLP, botijão 13 kg, utilizou-se o levantamento de preços ao consumidor final da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP). Para o GN, utilizou-se os dados da Companhia Paranaense de Gás (COMPAGAS) referentes à tarifa residencial.

Tabela 4: Tarifas de referência entre os meses de setembro de 2017 à fevereiro de 2018.

Mês Referência	set/17	out/17	nov/17	dez/17	jan/18	fev/18
Custo Energia Elétrica (R$ / kWh)	0,64	0,64	0,64	0,64	0,64	0,64
Custo Gás GLP (R$ / botijão P13)	60,15	63,10	64,92	65,62	66,24	68,52
Custo Gás Natural (R$ / m³)	3,09	3,09	3,09	3,09	3,09	3,09

Fonte: Copel, ANP e Compagas (2018).

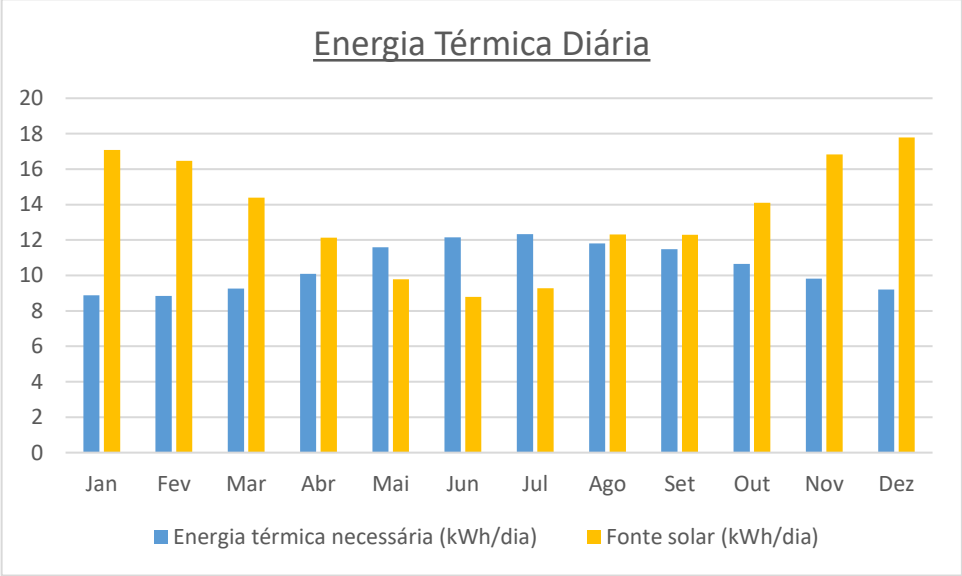
Tabela 5: Tarifas de referência entre os meses de março à agosto de 2018.

Mês Referência	mar/18	abr/18	mai/18	jun/18	jul/18	ago/18
Custo Energia Elétrica (R$ / kWh)	0,64	0,64	0,64	0,77	0,77	0,77
Custo Gás GLP (R$ / botijão P13)	68,64	68,19	68,21	69,36	69,38	69,54
Custo Gás Natural (R$ / m³)	3,09	3,09	3,09	3,09	3,09	3,09

Fonte: Copel, ANP e Compagas (2018).

A partir dos dados de irradiação e temperatura, equacionou-se a energia térmica necessária, por dia, para aquecer o volume de água e a participação da fonte solar. Os resultados obtidos estão descritos na Gráfico 1.

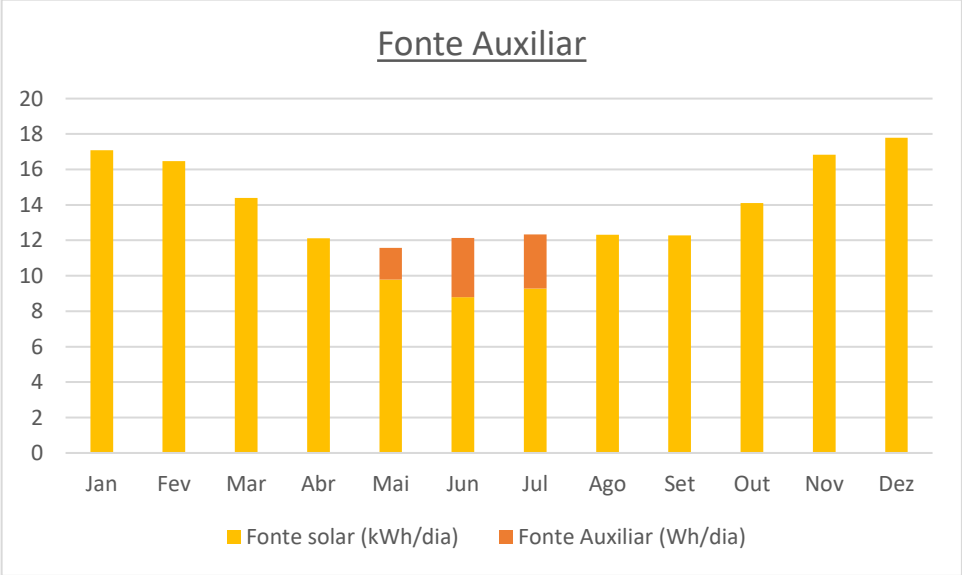
Gráfico 1: Energia térmica necessária para aquecimento da água e a participação energética da fonte solar.



Fonte: Autoria própria.

Nos meses em que a irradiação solar não é suficiente, faz-se necessário o complemento por parte da fonte auxiliar. A participação da fonte auxiliar está descrita no Gráfico 2.

Gráfico 2: Participação fonte auxiliar.

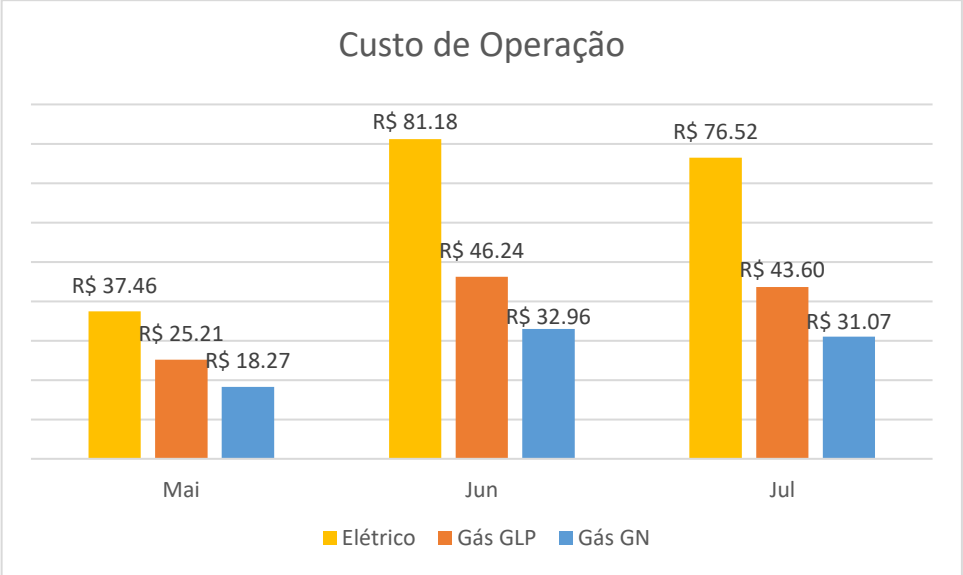


Fonte: Autoria própria.

Nos meses em que se faz necessário suplementar a fonte solar para fornecer

a energia térmica demandada, calculou-se o custo diário e mensal para realizar o complemento com cada uma das fontes em estudo, conforme apresentado no Gráfico 3.

Gráfico 3: Custo de operação fontes auxiliares.



Fonte: Autoria própria.

3.1.2. Volume de água aquecida

Com base no mesmo sistema do cenário anterior, tomando como base o mês de junho (pior média de irradiação diária), a participação da fonte solar é suficiente apenas para elevar em 9,6ºC a temperatura do sistema, fazendo-se necessária uma maior utilização da fonte auxiliar. Admitindo-se que:

- Massa de água por banho: 60 kg;
- Capacidade do reservatório térmico: 5 banhos;
- Temperatura de consumo: 40 ºC.

Tabela 6: Energia e excedentes por número de banhos.

Nº de Banhos	Energia Térmica (Wh)	Excedente (%)
1	1151	80%
2	2303	60%
3	3454	40%
4	4605	20%
5	5757	0%

Fonte: Autoria própria.

Um sistema no qual seja aquecido todo o volume, e o perfil de consumo acabe por não demandar tal oferta, tem-se excedentes que acumulados ao longo dos meses podem gerar uma quantidade significativa de energia desperdiçada.

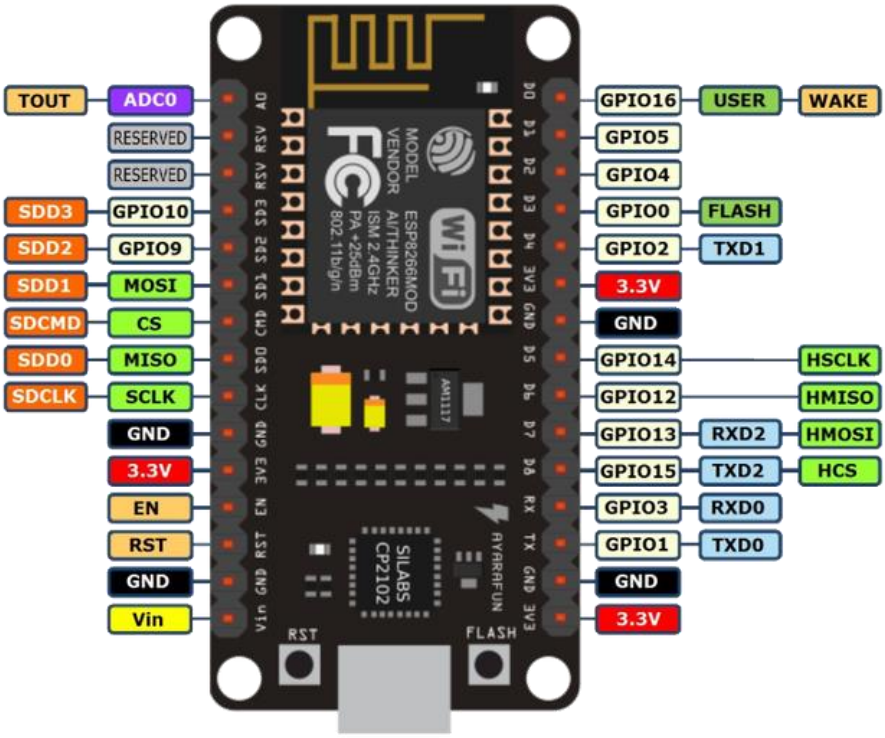
4. TOPOLOGIA DO SISTEMA

Este capítulo discorre sobre o sistema, seus componentes selecionados, características e interconexões.

4.1. *HARDWARE* ATUADOR

Para realizar a interface com o servidor, faz-se necessário um *hardware* atuador atuando sobre o acionamento das bombas e envio das temperaturas. Como premissa deste projeto, selecionou-se o microcontrolador ESP8266 da fabricante Espressif sob plataforma NodeMCU, representado na Figura 9. O NodeMCU ESP8266 conta com CPU 32-bit RISC: Tensilica Xtensa LX106 (clock de 80 MHz), 64 KB de memória RAM, 4 MB de FLASH, 16 pinos de GPIO, SPI, I²C e Wi-Fi IEEE 802.11 b/g/n (Espressif, 2018).

Figura 9: Diagrama de pinos do microcontrolador ESP8266.

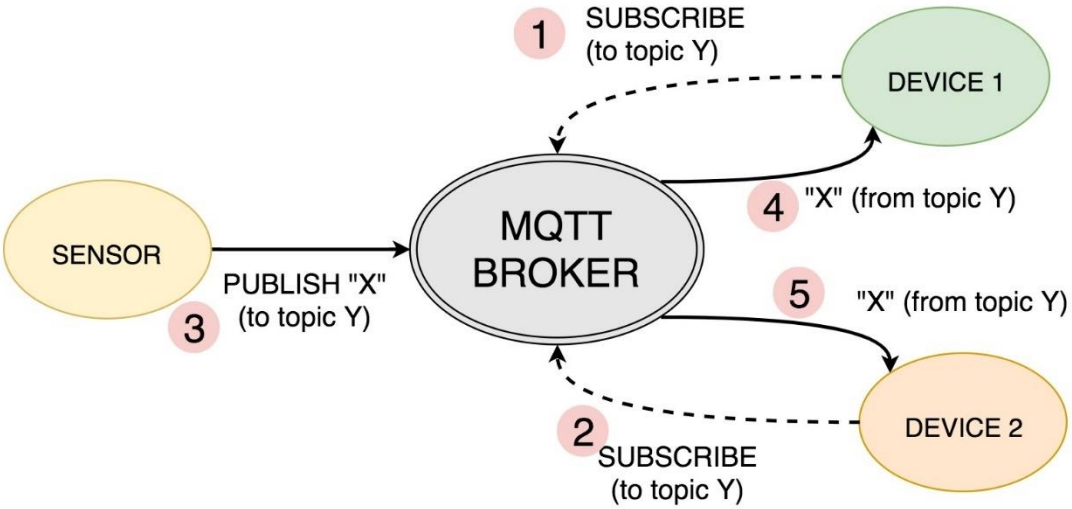


Fonte: IOT Bytes (2016).

4.2. COMUNICAÇÃO

Para comunicação entre o *hardware* atuador e o servidor, selecionou-se o protocolo MQTT (*Message Queuing Telemetry Transport*). Voltado para sensores e pequenos dispositivos móveis, o protocolo MQTT tem como princípios a baixa utilização de largura de banda e consumo reduzido de recursos do dispositivo. A partir da conexão de um ou mais dispositivos a um *broker* (servidor), estes podem publicar informações em tópicos, ou inscrever-se em determinado tópico para receber as informações publicadas, conforme esquematizado na Figura 10.

Figura 10: Estrutura funcionamento protocolo MQTT.



Fonte: Norwegian Creations (2017).

4.3. SERVIDOR

Com intuito de tornar o sistema flexível, optou-se por utilizar um servidor para realizar o gerenciamento do sistema, desta forma reduzindo as atribuições do *hardware* local. Em virtude da extensa compatibilidade e gama de aplicativos, selecionou-se o sistema operacional Ubuntu Server da Canonical. O servidor tem o papel de hospedar: o *broker* MQTT, a ferramenta Node-RED e o banco de dados InfluxDB.

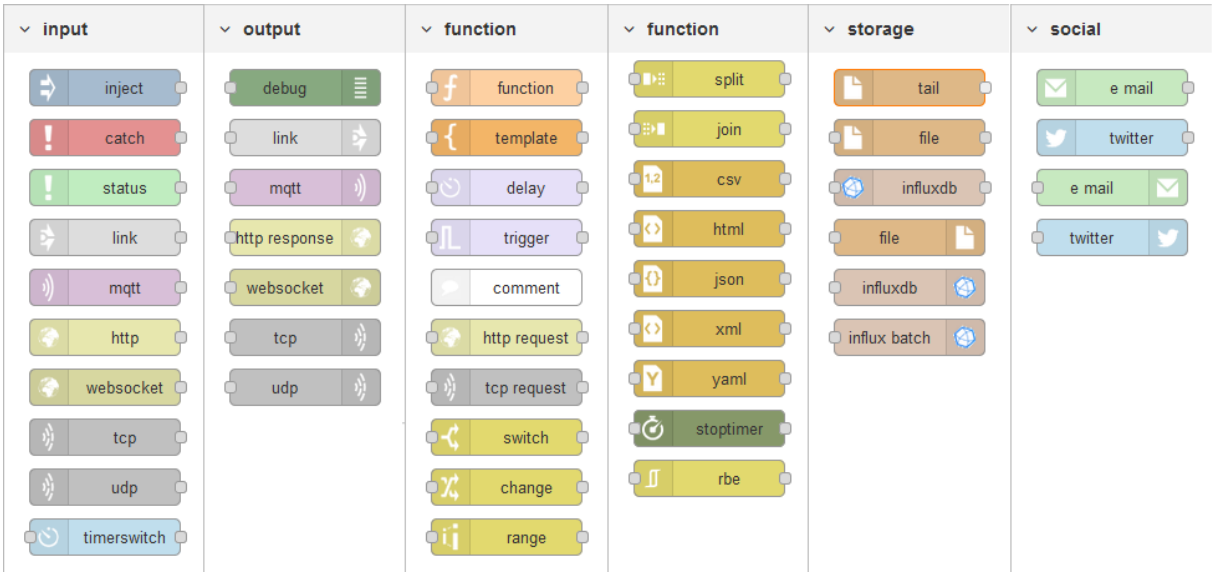
4.3.1. Node-RED

A ferramenta Node-RED se baseia em blocos de funções, chamados de *nodes*, que podem ser conectados de forma a obter o comando desejado. Os blocos se dividem em cinco categorias:

- Entrada – realiza funções de entrada;
- Saída – realiza funções de saída;
- Funções – realiza operações diversas;
- Armazenamento – realiza operações com banco de dados e outras formas de armazenamento;
- Social – realiza a comunicação com plataformas sociais.

A plataforma também conta com uma biblioteca *on-line* de *nodes*, os quais podem ser instalados, adicionando novos blocos com novas funções. A Figura 11 apresenta a paleta de funções do Node-RED.

Figura 11: Paleta de funções do Node-RED.

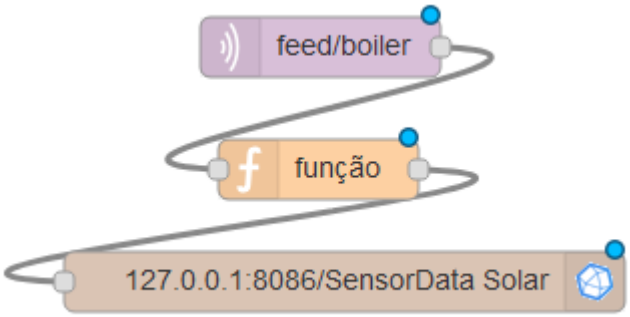


Fonte: Autoria própria.

4.3.1.1. Aplicações da ferramenta

Com o estudo da ferramenta, verificou-se como exemplificado pela Figura 12, uma possibilidade de utilização no sistema proposto. A partir de um *node* do tipo entrada MQTT inscrito em um tópico “feed/boiler”, ao receber uma mensagem, irá condicionar a informação recebida, através do *node* “função” e armazená-la na base de dados “SensorData”, através do *node* de armazenamento InfluxDB. Este irá realizar operação junto ao banco de dados de mesmo nome.

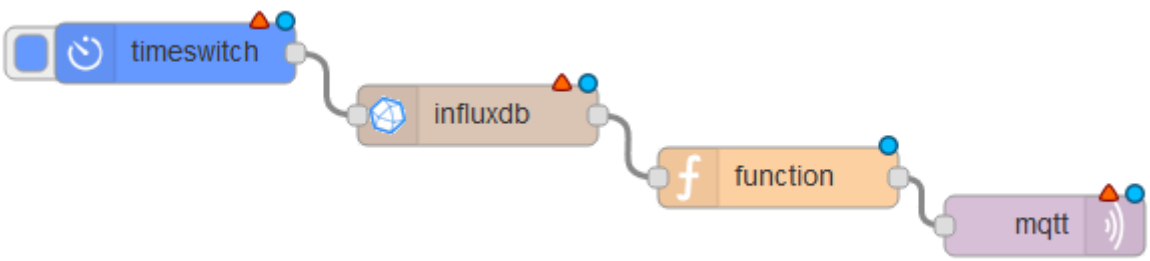
Figura 12: Armazenamento do valor da temperatura no banco de dados.



Fonte: Autoria própria.

Uma segunda possibilidade de aplicação, é realizando uma consulta ao banco de dados, conforme esquematizado na Figura 13. O início da operação se dá a partir do *node* “timeswitch”, o qual executa a função em horário ou intervalo de tempo programado, seguido do *node* “influxdb” que realiza a consulta ao banco de dados, e posteriormente terá sua saída condicionada pelo *node* “function”. Por fim as informações serão transmitidas para um dispositivo através do *node* mqtt.

Figura 13: Operação de consulta ao banco de dados em horário programado.



Fonte: Autoria própria.

5. IMPLEMENTAÇÃO

Neste capítulo são discutidas as implementações realizadas para desenvolvimento e efetivação do projeto proposto. A integra dos códigos está disponível no Apêndices A e B referentes a, respectivamente, *hardware* atuador e Node-RED.

5.1. SERVIDOR

Para hospedar os serviços necessários para execução do projeto proposto, utilizou-se o sistema operacional Ubuntu Server da Canonical, versão 18.04 LTS. O servidor conta com a seguinte especificação:

- Processador: 1x Núcleo @ 3.50+ GHz;
- Memória RAM: 1024 MB;
- Armazenamento: 20 GB SSD.

Adicionalmente, para implementação do sistema, foram instalados os seguintes *softwares*:

- *Broker* MQTT: Mosquitto versão 1.4.15, da Eclipse Foundation;
- Banco de Dados: InfluxDB versão 1.7.6, da InfluxData;
- Node-RED versão 0.20.3, da JS Foundation.

5.2. *HARDWARE* ATUADOR

O *hardware* atuador, concebido para operação do sistema, consiste em um módulo NodeMCU ESP8266 conectado a um conjunto de sensores digitais de temperatura do tipo DS18B20 (telemetria do sistema de aquecimento solar) e saídas a relê (acionamento fontes auxiliares).

O sensor DS18B20, possui a seguinte especificação:

- Faixa de medição: -55°C a +125°C;

- Precisão: ±0.5°C entre -10°C e +85°C;
- Resolução programável de 9 *Bits* a 12 *Bits*;
- Interface de comunicação 1-*Wire*.

A comunicação do *hardware* atuador com o servidor é realizada através do protocolo MQTT. Os valores das temperaturas coletados: reservatório, alimentação, retorno gás e retorno coletores, são enviados a cada 30s para o servidor. O atuador também se subscreve nos tópicos de comando das fontes auxiliares e aciona as saídas conforme os sinais recebidos.

5.3. NODE-RED

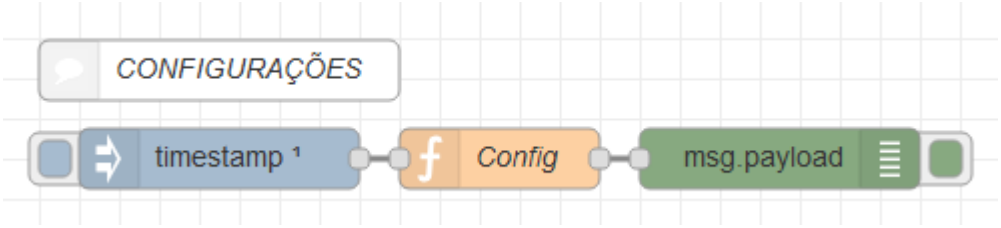
5.3.1. Sistema de controle

O sistema de controle implementado na plataforma Node-RED é composto por 4 blocos: configurações, sensores, tarifas e aquecimento auxiliar. Cada um dos blocos é detalhado nos tópicos subsequentes.

5.3.1.1. Bloco Configurações

O bloco configurações, conforme representação na Figura 14, é responsável por realizar a configuração inicial do sistema e atualizações futuras.

Figura 14: Bloco de configurações.



Fonte: Autoria própria.

O bloco é composto por uma entrada do tipo “timestamp”, que aciona o bloco de função “Config”, que realiza as seguintes operações:

- Define variáveis booleanas para ajustar os sistemas de aquecimento auxiliar disponíveis:
    - “aux_ele” – Sistema auxiliar elétrico;
    - “aux_gn” – Sistema auxiliar a gás natural;
    - “aux_glp” – Sistema auxiliar a gás liquefeito de petróleo.

Cada um destes sistemas pode ser habilitado ou desabilitado, alterando respectivamente, a variável para *true* ou *false*.

- Define variáveis de ajuste de operação:
    - “temperatura_banho” – Define a temperatura desejada para o reservatório;
    - “qtd_banho” – Define a quantidade de banhos a serem considerados;
    - “m_h2o” – Massa (em kg) estimada de água consumida por banho;
    - “operacao” – Define o modo de operação do sistema de controle. Esse pode ser do tipo “quantidade”, operando com base no volume de água a ser aquecido, ou do tipo “temperatura”, no qual opera apenas com base na temperatura desejada (aquece todo o volume disponível no reservatório).

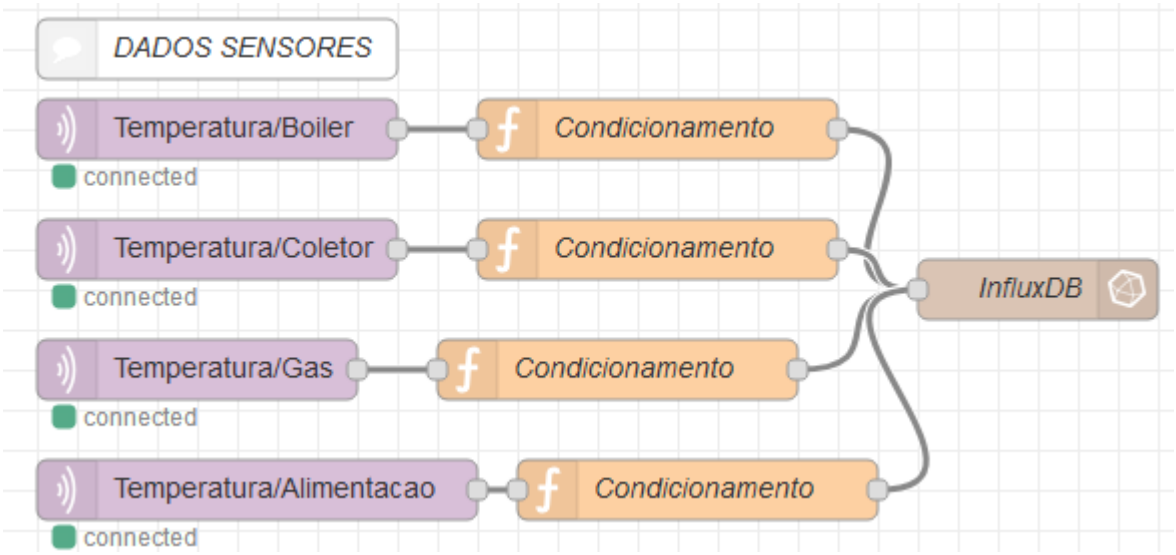
- Define as constantes dos sistemas instalados:
    - “rend_ele” – Rendimento aquecedor auxiliar elétrico;
    - “pot_ele” – Potência aquecedor auxiliar elétrico em Watts;
    - “rend_gn” – Rendimento aquecedor auxiliar a gás natural;
    - “cons_gn” – Consumo gás natural em metros cúbicos por hora;
    - “pot_gn” – Potência aquecedor auxiliar a gás natural em Watts;
    - “rend_glp” – Rendimento aquecedor auxiliar a GLP;
    - “cons_glp” – Consumo GLP em quilograma por hora;
    - “pot_glp” – Potência aquecedor auxiliar a GLP em Watts;
    - “vol_boiler” – Volume do reservatório em litros;
    - “ce_h2o” – Calor especifico da água em Watt-hora por quilograma grau Celsius, aproximadamente 1,163 Wh / kg.ºC.

Após o processamento e inserção das variáveis, a função retorna, para a janela de *debug* (depuração), um *timestamp* (marca temporal) denotando a presente data e hora, ou alerta e ocorrência em caso de erro.

5.3.1.2. Bloco Sensores

O bloco sensores, representado na Figura 15, é o responsável por receber e condicionar os dados de temperatura provenientes do *hardware* atuador.

Figura 15: Conjunto blocos de dados de entrada dos sensores.



Fonte: Autoria própria.

O bloco é composto por entradas do tipo “mqtt”, o qual alimenta os respectivos blocos de função “Condicionamento”. Estes realizam o condicionamento dos dados recebidos e posteriormente os enviam para o bloco de saída do tipo “InfluxDB”.

O bloco de entrada “mqtt” é responsável por realizar a subscrição nos tópicos de interesse do *broker* MQTT. Para as entradas, os tópicos dos sensores são:

- “Temperatura/Boiler” – Temperatura do reservatório em ºC;
- “Temperatura/Coletor” – Temperatura para monitoramento do retorno coletores solar em ºC;
- “Temperatura/Gas” – Temperatura para monitoramento retorno aquecedor a gás em ºC;

- “Temperatura/Alimentacao” – Temperatura para monitoramento da alimentação dos coletores solar em ºC.

O bloco de função para condicionamento dos dados é responsável por converter a variável do tipo *string,* proveniente do bloco “mqtt”, para uma variável do tipo *float*. Adicionalmente a função carrega o valor de temperatura para uma variável global do ambiente Node-RED e formata a saída para “Notação de Objetos JavaScript” (JSON), conforme ilustrado na Figura 16.

Figura 16: Fluxo de saída em formato JSON.

```
msg.payload = {
    temperatura: temp,
    sensor: "boiler",
}
```

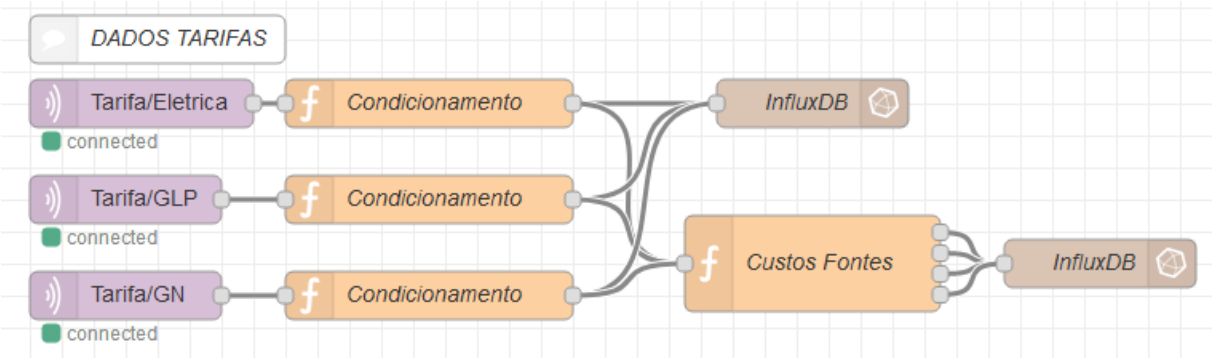
Fonte: Autoria própria.

O bloco InfluxDB recebe o JSON contendo as séries de dados “temperatura” e “sensor”, e as salva na base definida pelo parâmetro “Measurements” do bloco. Para o bloco sensores a base designada é “Temperaturas”.

5.3.1.3. Bloco Tarifas

O bloco tarifas, representado na Figura 17, é responsável por processar os valores de tarifas e definir a fonte com melhor custo benefício.

Figura 17: Bloco de entrada e processamento tarifas.



Fonte: Autoria própria.

A partir de uma entrada através dos tópicos:

- “Tarifa/Eletrica” - tarifa da energia elétrica em reais por quilowatt-hora (R$/kWh);
- “Tarifa/GLP” - tarifa do botijão de GLP modelo P13 em reais por botijão (R$/botijão);
- “Tarifa/GN” - tarifa do gás natural em reais por metro cúbico (R$/m³).

O bloco de função para condicionamento converte a variável do tipo *string,* proveniente do bloco “mqtt”, para uma variável do tipo *float*. Adicionalmente a função carrega o valor da respectiva tarifa para uma variável global do ambiente Node-RED (“tar_ele”, “tar_glp” e “tar_gn”) e formata a saída para JSON, conforme ilustrado na Figura 18.

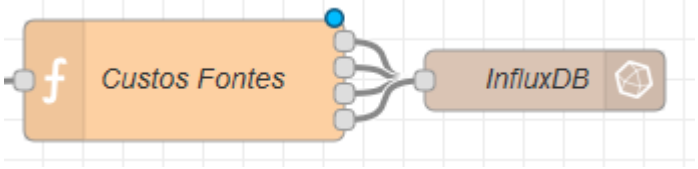
Figura 18: Fluxo de saída valor tarifa em JSON.

```
11 msg.payload = {
12     tarifa: tar,
13     topic: "eletrica",
14 }
15 return msg;
```

Fonte: Autoria própria.

A saída de um dos blocos “Condicionamento”, realiza o direcionamento das funções para o armazenamento na base “Tarifas” através do bloco InfluxDB e aciona o bloco de função “Custos Fontes” para o cálculo do custo benefício de cada uma das fontes.

Figura 19: Conexão bloco de função “Custos Fontes” com bloco InfluxDB.



Fonte: Autoria própria.

Para o cálculo do custo benefício de cada umas das fontes, além do valor das tarifas são utilizados os dados de potência, consumo e rendimento das fontes. O custo

calculado é o equivalente a geração de 1 kW de energia térmica, definido para cada fonte através das variáveis “custo_aux_ele”, “custo_aux_glp” e “custo_aux_gn”, conforme Figura 20.

Figura 20: Variáveis e funções para cálculo dos custos.

```
17  var custo_aux_ele = ((1000/rend_ele)/1000)*tar_ele;
18  var custo_aux_glp = (((1000/rend_glp)/(pot_glp))*cons_glp)*(tar_glp/13);
19  var custo_aux_gn = (((1000/rend_gn)/(pot_gn))*cons_gn)*tar_gn;
```

Fonte: Autoria própria.

Para seleção da fonte, com melhor condição de custo, são definidos 2 *arrays*: “custos” e “nome_fontes”. Estes *arrays* são carregados, através da função *push*, com os custos e nome das fontes que estão habilitadas (conforme bloco configuração). A seleção da melhor opção é realizada pela função “*Math.min.apply”* a qual seleciona o menor valor do *array*. Em caso de duas ou mais fontes com o mesmo custo, a priorização segue a seguinte sequência: Elétrico, Gás Natural e GLP.

Figura 21: Detalhe código para carregamento custos e nomes fontes para *array*.

```
22  var custos = [];
23  var nome_fontes = [];
24
25  if(global.get("aux_ele")===true){
26      custos.push(custo_aux_ele);
27      nome_fontes.push("Elétrico");
28  }
29  if(global.get("aux_gn")===true){
30      custos.push(custo_aux_gn);
31      nome_fontes.push("Gás Natural");
32  }
33  if(global.get("aux_glp")===true){
34      custos.push(custo_aux_glp);
35      nome_fontes.push("GLP");
36  }
```

Fonte: Autoria própria.

O bloco “Custos Fontes” possui 4 saídas (msg1, msg2, msg3 e msg4), as quais exportam os custos de cada uma das fontes e a fonte selecionada para o banco de dados, conforme ilustrado na Figura 22.

Figura 22: Definição para as 4 saídas bloco “Custos Fontes”.

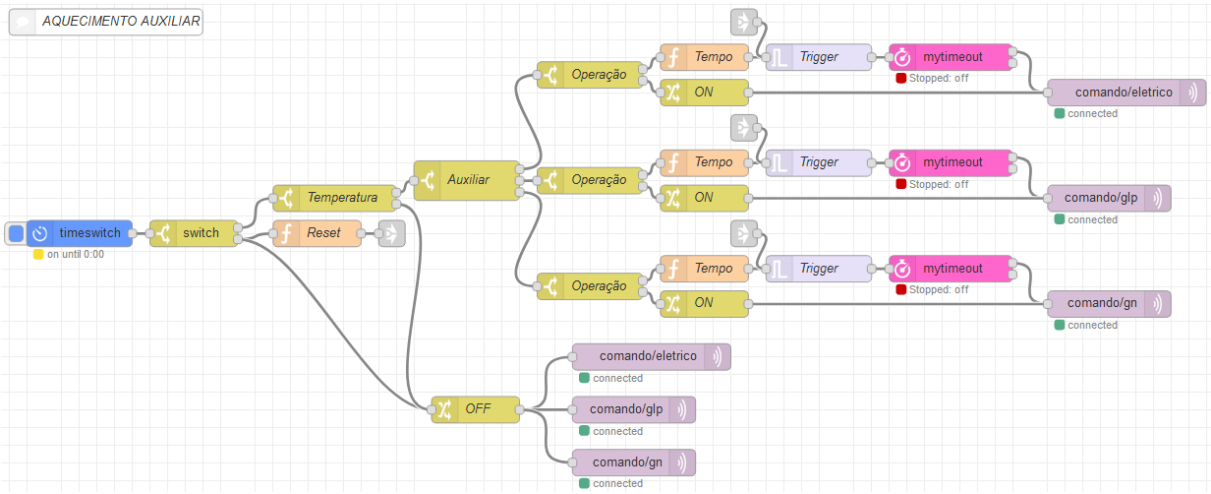
```
46  var msg1 = {payload: {custo:custo_aux_ele,topic:"eletrica"}};
47  var msg2 = {payload: {custo:custo_aux_glp,topic:"glp"}};
48  var msg3 = {payload: {custo:custo_aux_gn,topic:"gn"}};
49  var msg4 = {payload: {fonte:melhor_fonte,topic:"melhor"}};
```

Fonte: Autoria própria.

5.3.1.4. Bloco Aquecimento Auxiliar

O bloco aquecimento auxiliar, ilustrado na Figura 23, é o responsável por gerenciar o acionamento do aquecimento auxiliar selecionado quando necessário, e controlar a quantidade de água a ser aquecida.

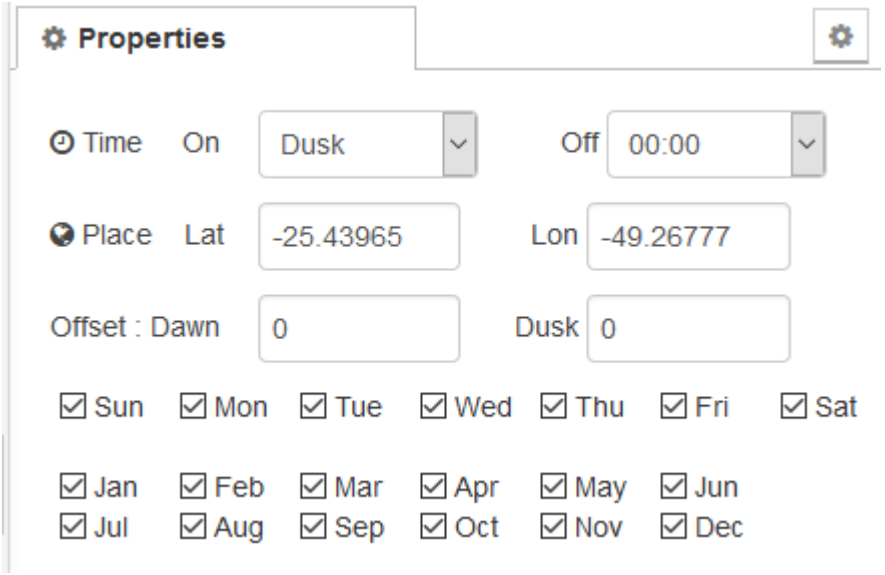
Figura 23: Visão macro dos blocos para acionamento aquecimento auxiliar.



Fonte: Autoria própria.

O funcionamento se dá a partir do *node* “timeswitch” que permite agendar eventos para acionamentos em horários programados. O “timeswitch” tem como parâmetros a latitude e longitude do local da instalação, o horário, meses e dias da semana nos quais se deseja ativá-lo. Para esta implementação considerou-se o acionamento diário do pôr do sol (*dusk*) até às 00:00.

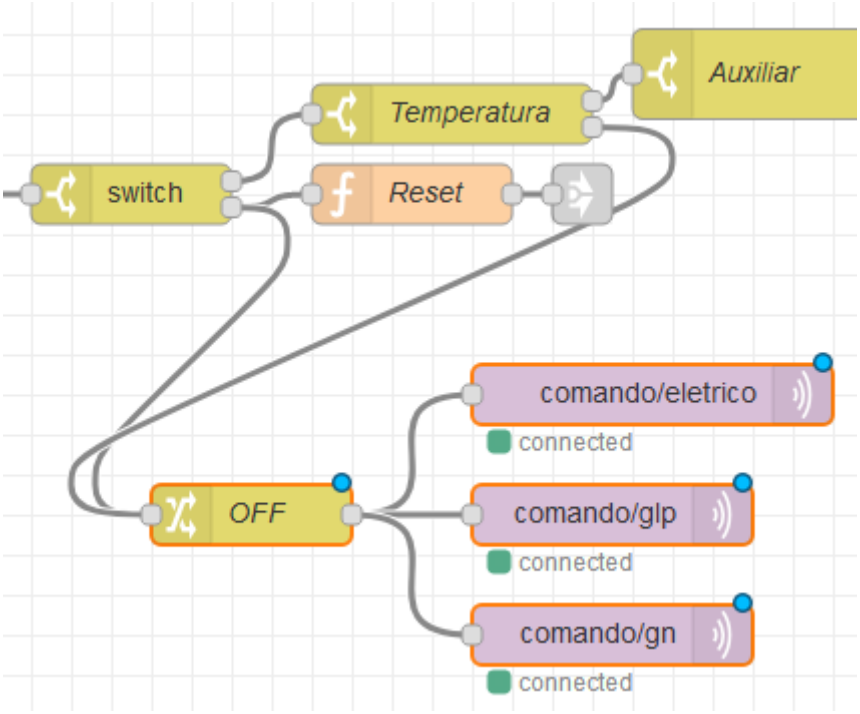
Figura 24: Propriedades *node* “timeswitch”.



Fonte: Autoria própria.

O *node* “*timeswitch*” gera valores de saída “1” para condição ligado (dentro do horário agendado) e “0” para condição desligado (fora do horário agendado). Sua saída alimenta o *node* do tipo “switch” que é responsável por encaminhar o fluxo conforme a condição de entrada. Para a condição desligado, o fluxo será redirecionado a segunda saída. Esta irá executar o *node* de função “Reset” (*reset* do *trigger* modo de operação por quantidade de água aquecida – será explanado mais à frente) e o *node* “OFF” do tipo *change*.

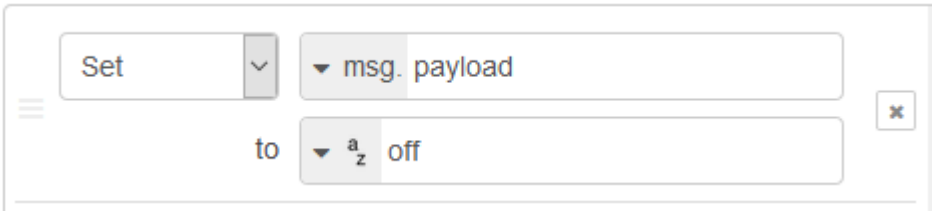
Figura 25: Detalhamento encaminhamento dos fluxos pelo *node* “switch”.



Fonte: Autoria própria.

O *node* “OFF” tem a função de substituir, conforme ilustrado na Figura 26, o conteúdo recebido do fluxo (*payload*) pelo valor “off” e este é encaminhado aos *nodes* de saída do tipo MQTT. Por sua vez, os *nodes* MQTT distribuem a mensagem para os tópicos de cada uma das fontes: “comando/eletrico”, “comando/glp” e “comando/gn”.

Figura 26: Propriedades *node* “OFF”.



Fonte: Autoria própria.

Caso o valor de saída do *node* “*timeswitch*” seja “1” (condição ligado) o *node* “*switch*” encaminha o fluxo para o *node* “Temperatura”. O *node* “Temperatura” também é do tipo *switch*, e é responsável, conforme Figura 27, por comparar o valor das variáveis “temperatura_boiler” (temperatura do reservatório) e “temperatura_banho” (temperatura desejada). Em caso de uma temperatura maior ou igual a desejada, o

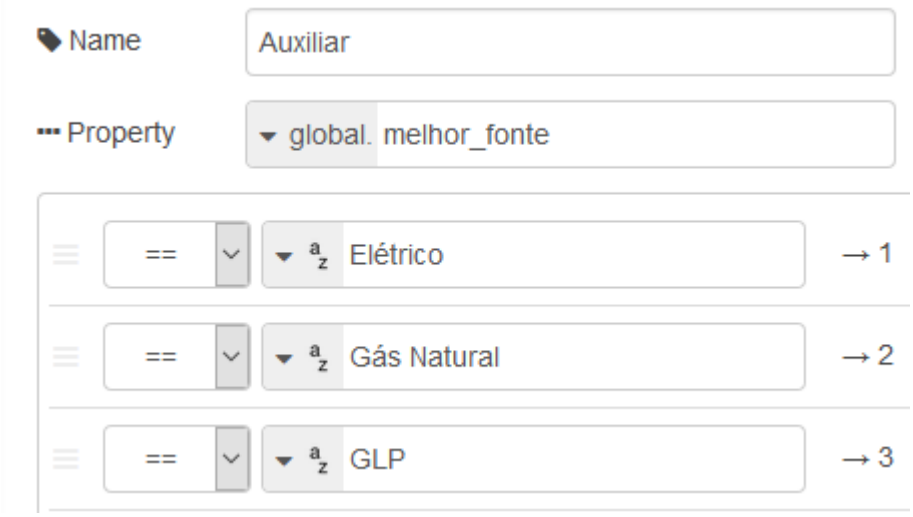
node irá encaminhar o fluxo para o *node* “OFF” (explanado no parágrafo anterior). Caso a temperatura seja menor do que a desejada, o fluxo seguirá para o node “Auxiliar”.

Figura 27: Propriedades *node* “Temperatura”.

Fonte: Autoria própria.

O *node* “Auxiliar” tem a função de realizar o direcionamento para a fonte selecionada através do bloco “Custos Fontes” (previamente definido). Por ser do tipo *switch*, ele realiza esse direcionamento a partir do comparativo do valor da variável “melhor_fonte”. Conforme ilustrado pela Figura 28, caso o valor seja “Elétrico”, “Gás Natural” ou “GLP” o fluxo será direcionado, respectivamente, para as saídas 1, 2 e 3.

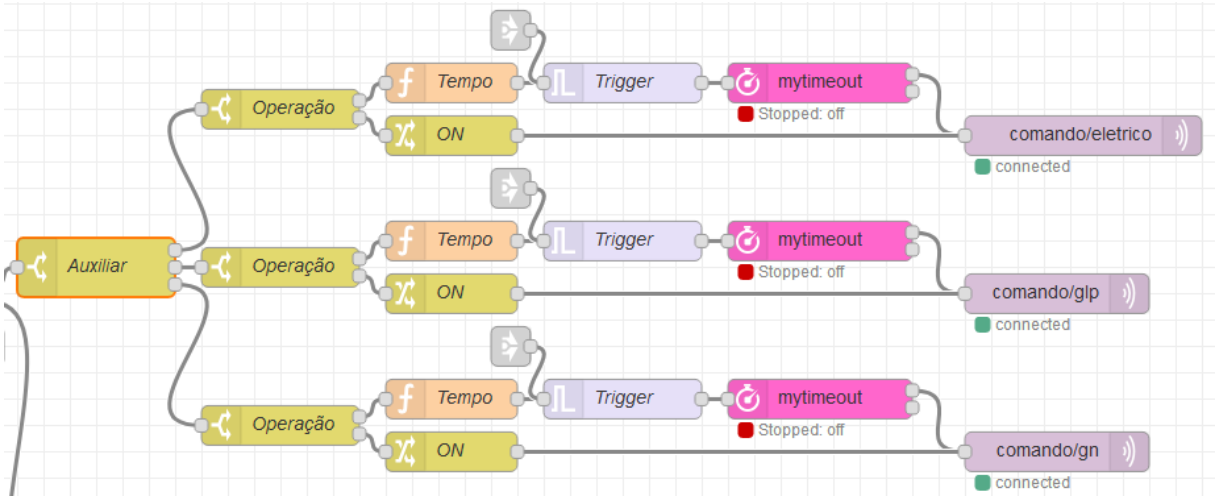
Figura 28: Propriedades *node* “Auxiliar”.



Fonte: Autoria própria.

Cada uma das saídas do *node* “Auxiliar” está ligada, conforme Figura 29, a um conjunto de blocos que realiza o acionamento da respectiva fonte auxiliar.

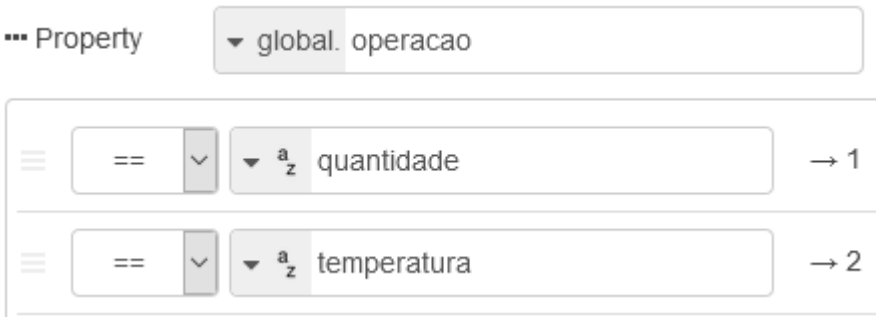
Figura 29: Detalhe blocos para seleção fonte auxiliar.



Fonte: Autoria própria.

O conjunto de blocos para acionamento da fonte auxiliar, inicia-se com o *node* “Operação”. Este redireciona o fluxo conforme modo de operação configurado, acionando a saída 1 para modo “quantidade” (com base no volume de água a ser aquecido), ou a saída 2 para modo “temperatura” (aquece todo o volume disponível no reservatório).

Figura 30: Propriedades *node* “Operação”.

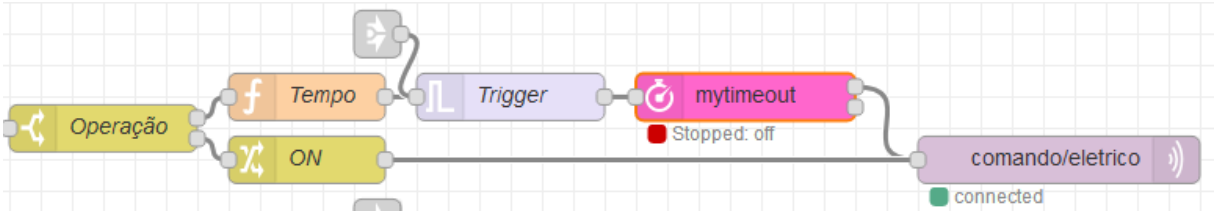


Fonte: Autoria própria.

Para o modo “temperatura” (saída 2), o *node* “ON”, do tipo *change*, realiza a troca do fluxo recebido (*payload*) pelo valor “on” e este é encaminhado ao *node* de

saída do tipo MQTT. Por sua vez, o *node* MQTT distribui a mensagem para os tópicos da respectiva fonte para seu acionamento. A fonte auxiliar só será desativada caso a condição de temperatura ajustada seja atingida ou exceda o horário programado no *node* “timeswitch”.

Figura 31: Detalhe *node* para acionamento fonte auxiliar elétrica.



Fonte: Autoria própria.

Para o modo “quantidade” (saída 1), o *node* do tipo função “Tempo” é acionado e realiza o cálculo da duração do acionamento da respectiva fonte auxiliar, para geração da energia térmica complementar necessária. Para o cálculo, conforme Figura 32, são utilizadas as variáveis:

- “ce_h2o” – Calor especifico da água em Wh / kg.ºC;
- “m_h2o” – Massa de água quente por banho em kg;
- “qtd_banhos” – quantidade de banhos;
- “temperatura_banho” – temperatura desejada para o reservatório em ºC;
- “temperatura_boiler_final” – temperatura reservatório antes do acionamento;
- “rendimento” – rendimento da fonte auxiliar. Está poderá ser “rend_ele”, “rend_gn” ou “rend_glp”;
- “potencia” – potência da fonte auxiliar em W. Está poderá “pot_ele”, “pot_gn” ou “pot_glp”.

Figura 32: Detalhe do cálculo do tempo de acionamento da fonte auxiliar.

```
10 //Calcula quantidade de energia térmica necessária
11 var qtd_ene_nes=(ce_h2o*m_h2o*qtd_banhos*(temperatura_banho-temp_boiler_final))/rendimento;
12 //Calcula tempo em segundos
13 var timer = Math.round((qtd_ene_nes/potencia)*3600);
```

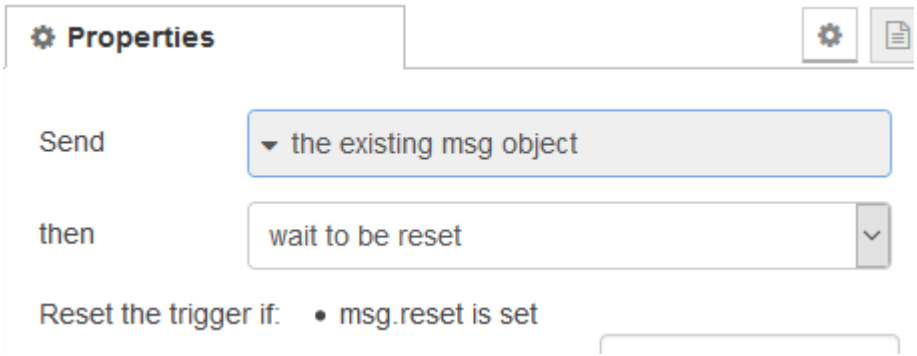
Fonte: Autoria própria.

O valor “timer”, saída da função “Tempo”, será direcionado para o *node*

“mytimeout” que realizará o acionamento da fonte auxiliar durante o tempo programado. O *node* “mytimeout”, que tem sua saída conectada aos *nodes* do tipo MQTT de cada uma das fontes, ao iniciar a contagem envia uma mensagem “on”, e ao término da contagem uma mensagem “off”.

Entre o *node* de função “Tempo” e o *node* “mytimeout” é utilizado o *node* “*trigger*”. Sua função é garantir um único acionamento por intervalo programado para operação. Seu funcionamento se dá permitindo apenas a passagem do primeiro fluxo recebido e bloqueando os subsequentes. A condição que permite novamente a passagem de um fluxo é receber o comando “*reset*”, e este é enviado quando o *node* “timeswitch” envia a condição desligado (fora do horário agendado).

Figura 33: Propriedades *node* “*trigger*”.



Fonte: Autoria própria.

5.3.2. Simulação

Para simulação utilizou-se uma base histórica no formato CSV (*Comma Separated Values*) a qual contém dados reais de temperatura inicial, final e tarifas referentes ao ano de 2018, coletadas a partir de um sistema de aquecimento solar instalado na residência, em Curitiba/PR, do executor do presente trabalho. O bloco da simulação é composto pelo conjunto de *nodes* ilustrados na Figura 34. Optou-se por implementar um só bloco, mas de forma a operar calculando todos os possíveis cenários.

Figura 34: Representação do bloco de simulação.

Fonte: Autoria própria.

Para inicializar o fluxo no bloco utilizou-se o *node* “timestamp”, este aciona o *node* “DADOS 2018” no qual estão armazenados, em CSV, valores de entrada referentes ao ano de 2018. Esse fluxo segue para o *node* “csv”, que será responsável por interpretar os dados de entrada e iterar linha a linha as variáveis para sua saída. Esta que por sua vez está conectada a função “Aquecimento” e realizará todos os cálculos, salvando os resultados na base de dados InfluxDB.

Figura 35: *Node* “DADOS 2018” – Base histórica temperaturas e tarifas.

Template — Syntax Highlight: none

```
1  Data;T Bi;T Bf;Tar_Ene;Tar_GLP;Tar_GN
2  31/12/2018;33.5;53.5;0.76897;66.105;3.749
3  30/12/2018;34.8;52.3;0.76897;66.105;3.749
4  29/12/2018;36.3;55.8;0.76897;66.105;3.749
5  28/12/2018;26.8;54.3;0.76897;66.105;3.749
6  27/12/2018;29.3;43.0;0.76897;66.105;3.749
7  26/12/2018;26.5;46.8;0.76897;66.105;3.749
```

Fonte: Autoria própria.

Por se tratar de uma simulação com múltiplos cenários, na qual os dados de entrada estão compilados num formato estático e a não necessidade de realizar o acionamento físico das fontes auxiliares, optou-se por realizar uma implementação agregando as operações em uma única função, mas de forma a manter as mesmas premissas utilizadas nos conjuntos de blocos previamente apresentados.

A função inicia-se com a definição dos dados do sistema: reservatório (volume e temperatura desejada) e fontes auxiliares (potência, rendimento e consumo). Do *node* são extraídas as informações: data, temperatura inicial, temperatura final e tarifas. A partir destes é realizado o cálculo para definir a necessidade de complementar o aquecimento e, caso necessário, são calculados a quantidade de energia térmica e custos para os seguintes cenários:

- Complemento volume total do reservatório;
- Complemento para 5 banhos;
- Complemento para 4 banhos;
- Complemento para 3 banhos;
- Complemento para 2 banhos;
- Complemento para 1 banho.

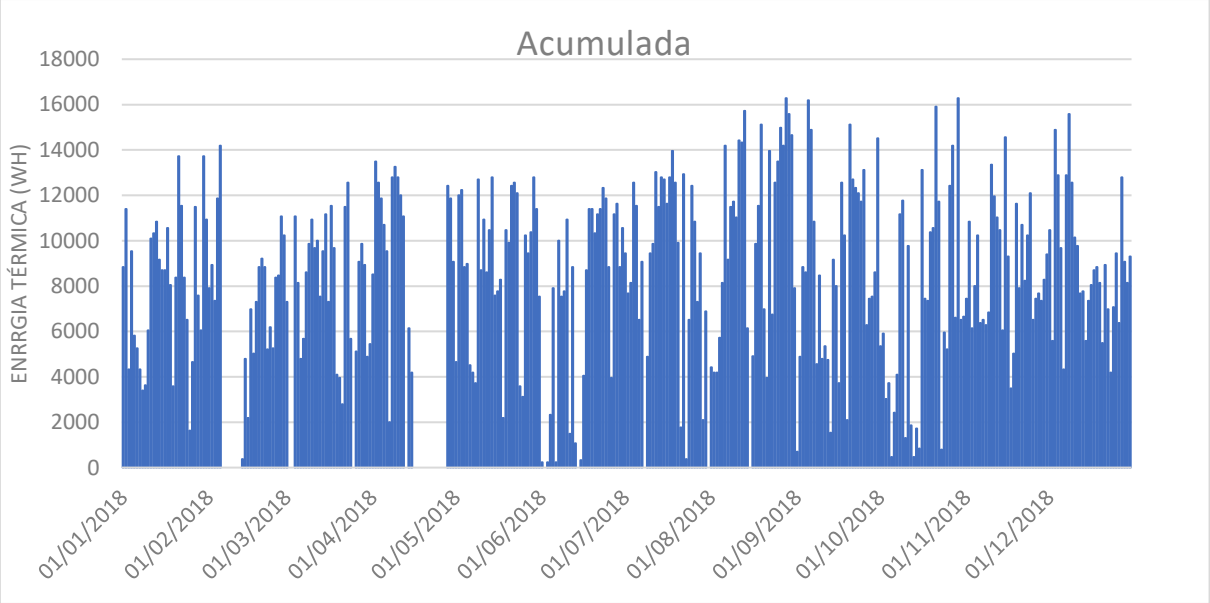
Os custos foram calculados para os três tipos de fonte auxiliar (Elétrica, GLP e GN). Para o cálculo dos complementos por quantidade de banhos, considerou-se o consumo de 60 litros de água a 40 ºC por banho. As demais considerações foram:

- Um reservatório térmico com 400 litros;
- Aquecedor de água a gás com potência de 13,5 kW;
- Rendimento: 85% (GN) / 84% (GLP);
- Consumo de gás: 1,22 m³/h (GN) / 0,98 kg/h (GLP).

6. RESULTADOS

A base histórica utilizada para simulação contemplava os valores iniciais e finais de temperatura do reservatório térmico para um período registrado de 341 dias do ano de 2018. Os resultados obtidos a partir do cálculo da energia térmica acumulada (em Wh) estão representados no Gráfico 4.

Gráfico 4: Energia térmica acumulada no período de 2018.

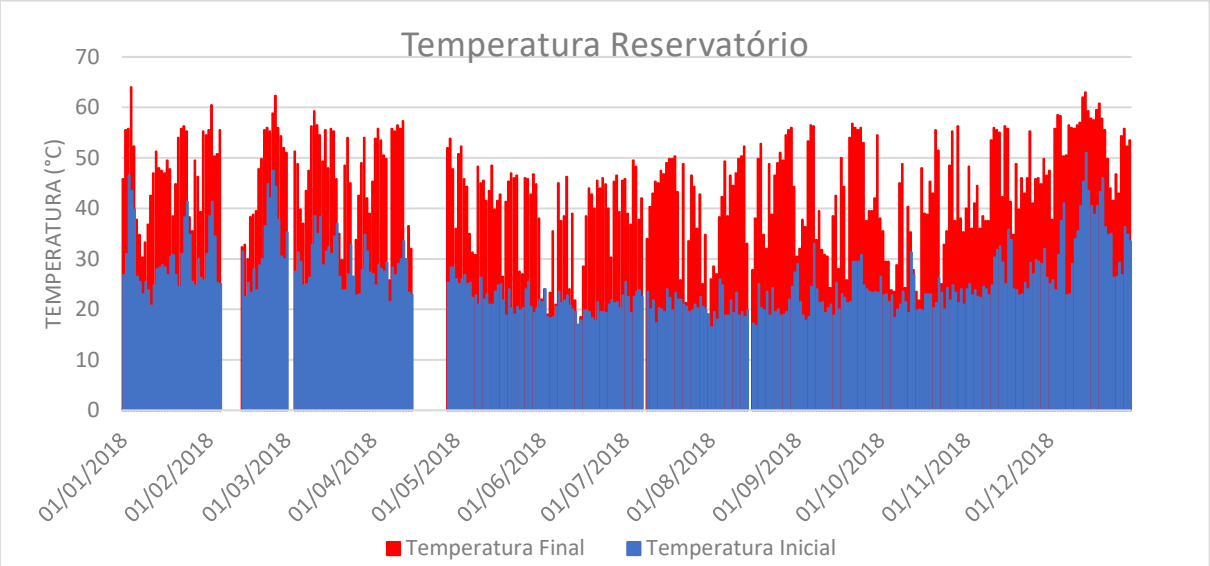


Fonte: Autoria própria.

A partir do gráfico da energia térmica acumulada, desconsiderando os períodos em que não houve registro de fevereiro (apenas 21 dias registrados) e abril (apenas 18 dias registrados), pode-se observar dias nos quais o acumulado diário de energia térmica foi baixo (principalmente no período que compreende o mês de junho).

Nota-se também que nos meses de janeiro e dezembro (pleno verão), nos quais há uma expectativa de maior geração térmica, o acumulado é menor se comparado a dias de outros meses. Isso se deve ao fato do indicador ser baseado no delta de temperatura, e para o período em questão a condição de temperatura inicial da água é mais alta resultando em uma menor diferença, conforme pode-se observar no Gráfico 5.

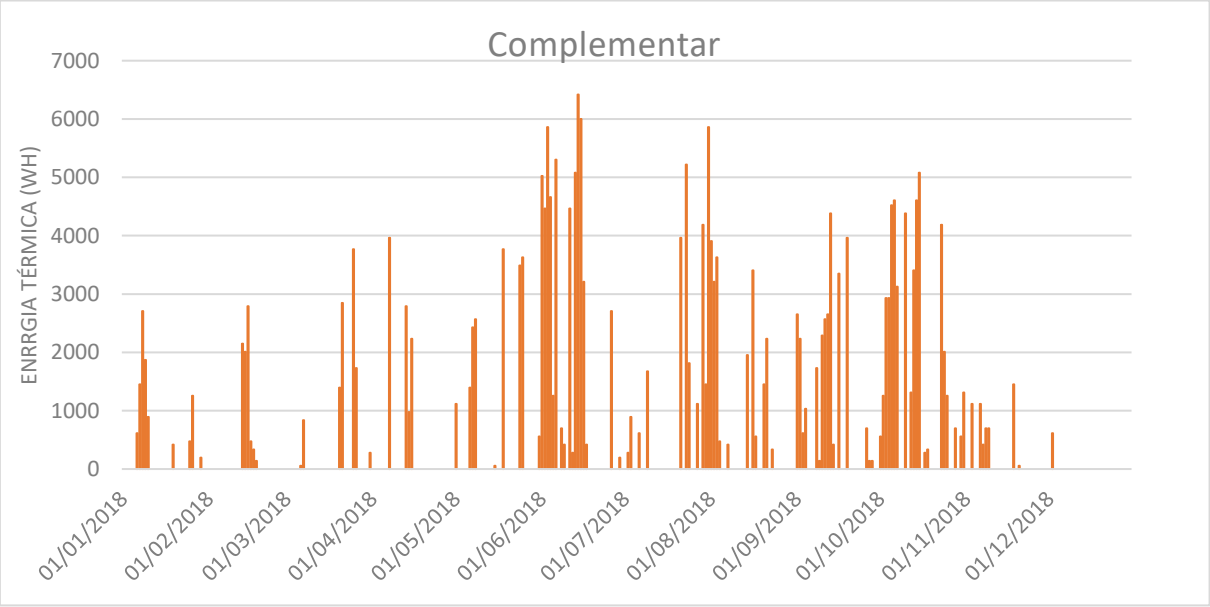
Gráfico 5: Temperatura reservatório no período de 2018.



Fonte: Autoria própria.

No Gráfico 6 é apresentado o resultado da energia térmica necessária para complementar o aquecimento em dias nos quais a insolação foi insuficiente.

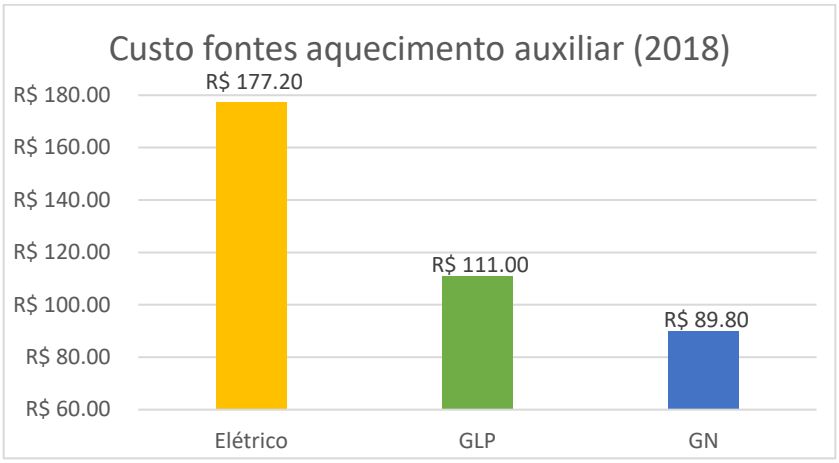
Gráfico 6: Energia térmica complementar.



Fonte: Autoria própria.

A partir da energia térmica necessária, simulou-se os custos necessários para realizar o complemento com cada um dos três tipos de fontes auxiliar. Os resultados para o período em questão, estão representados no Gráfico 7.

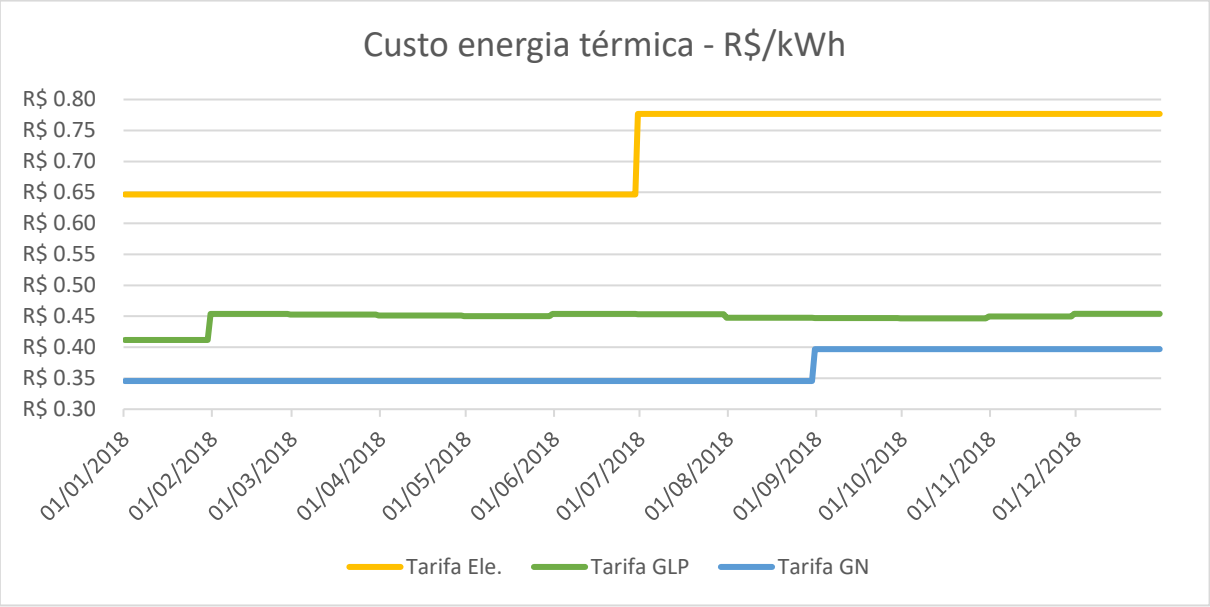
Gráfico 7: Custo fontes auxiliares período 2018.



Fonte: Autoria própria.

Para o período o gás natural foi a fonte que apresentou o menor custo, seguida do GLP e energia elétrica. No comparativo mês a mês, conforme Gráfico 8, o resultado se repete, apesar da ocorrência de reajustes nas tarifas, estes não foram significativos a ponto de alterar a fonte auxiliar com menor custo. Também deve-se considerar que a tarifa para fonte elétrica utilizada foi apenas a base (desconsiderando as bandeiras tarifárias), estas tornariam a fonte ainda mais cara.

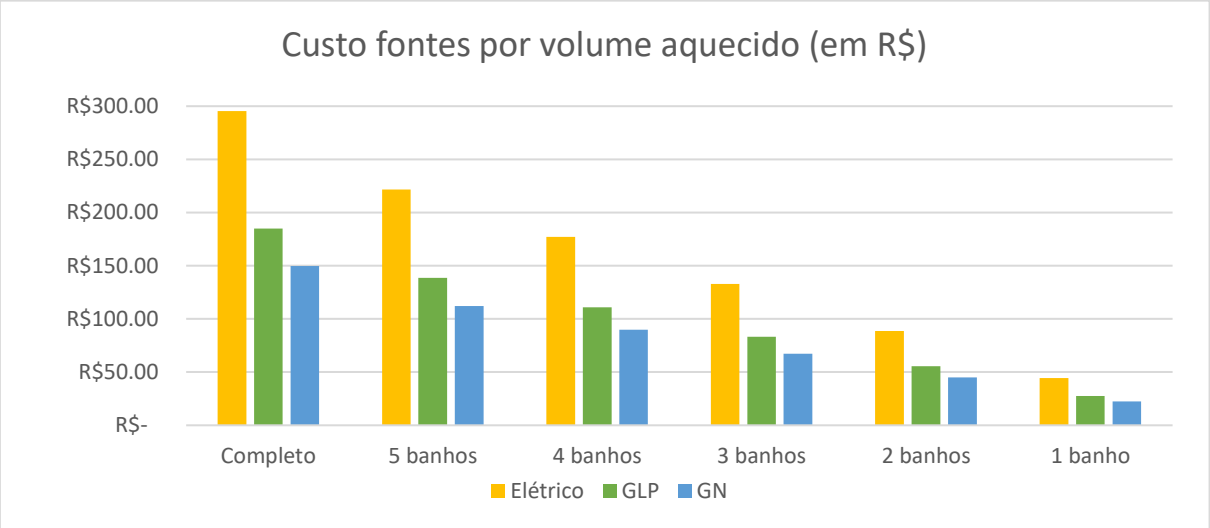
Gráfico 8: Custo energia térmica período 2018.



Fonte: Autoria própria.

Simulou-se também as condições de controle do volume de água aquecido: total, 5 banhos, 4 banhos, 3 banhos, 2 banhos e 1 banho. Os resultados estão representados no Gráfico 9.

Gráfico 9: Custo por volume aquecido.



Fonte: Autoria própria.

Para os cenários simulados pode-se também verificar a diferença percentual com relação ao aquecimento do volume completo do reservatório. Os resultados estão descritos na Tabela 7.

Tabela 7: Custos por volume de água aquecida.

Volume	Elétrico	GLP	GN	Dif. %
Completo	R$ 295,40	R$ 185,00	R$ 149,60	0%
5 banhos	R$ 221,60	R$ 138,70	R$ 112,20	-25%
4 banhos	R$ 177,20	R$ 111,00	R$ 89,80	-40%
3 banhos	R$ 132,90	R$ 83,20	R$ 67,30	-55%
2 banhos	R$ 88,60	R$ 55,50	R$ 44,90	-70%
1 banho	R$ 44,30	R$ 27,70	R$ 22,40	-85%

Fonte: Autoria própria.

7. CONCLUSÃO

Com base nos resultados encontrados a partir da simulação, com dados de um sistema real, o controle energético inteligente para sistemas de aquecimento solar se mostra benéfico para otimizar o consumo energético e monetário no quesito de volume de água aquecido. Tratando-se do seletor de fonte auxiliar com menor custo, os resultados de sua aplicação não se mostraram expressivos diante do atual cenário das tarifas, pois com a tarifa elétrica a um custo bem mais elevado se comparada a do gás, tem-se um cenário praticamente estático, minimizando a necessidade de automatizar o processo de seleção.

A aplicação do Node-RED, como ferramenta de programação baseada em fluxos, demonstrou-se bastante versátil durante a implementação do sistema. A integração nativa com o *broker* MQTT e a possibilidade de expansão da paleta adicionando novos *nodes* através da biblioteca suprimiram todas as necessidades para implementação do projeto.

A maior dificuldade encontrada no desenvolvimento desse projeto foi a impossibilidade da experimentação prática para obtenção de resultados mais precisos da implementação do sistema. Tal impossibilidade se deve principalmente às dimensões do sistema e alto custo. Idealmente para execução da experimentação, seriam necessários dois sistemas de aquecimento solar, de mesmas características, operando sob as mesmas condições climáticas e perfil de consumo, desta forma possibilitando o comparativo direto.

Para melhoria do sistema, sugerem-se trabalhos futuros que venham a otimizar e integrar novas possibilidades como, por exemplo:

- Explorar cenários nos quais a energia elétrica possa ser utilizada a custos menores: a partir da tarifa branca ou aproveitamento de excedentes gerados por um sistema fotovoltaico;
- Realizar a integração de todo o sistema de controle implementado em um único *hardware*;

- Aplicação de métodos de aprendizado de máquina para, a partir da base de dados, realizar a definição do volume de água a ser aquecido;
- Realização de um experimento prático para atestar a eficácia do sistema.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Sistema de aquecimento solar de água em circuito direto - Projeto e instalação ABNT NBR 15569.** Rio de Janeiro: [s.n.].

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Sistemas solares térmicos e seus componentes - Coletores solares Parte 1: Requisitos gerais. ABNT NBR 15747-1**. Rio de Janeiro: [s.n.].

AYOMPE, L. M. et al. Validated TRNSYS model for forced circulation solar water heating systems with flat plate and heat pipe evacuated tube collectors. **Applied Thermal Engineering**, v. 31, n. 8–9, p. 1536–1542, 1 jun. 2011.

BAUCHSPIESS, A. **Aplicações em Engenharia de Redes Neurais Artificiais, Lógica Fuzzy e Sistemas Neuro-Fuzzy**. [s.l.] Universidade de Brasília, nov. 2008. Disponível em: <http://www.ene.unb.br/adolfo/Lectures/ICIN/isi.pdf>. Acesso em: 16 maio. 2018.

BERTOLDO, L. **Princípios Da Teoria Térmica**. [s.l.] Clube de Autores, 2009.

DISTEFANO, J. J.; STUBBERUD, A. R.; WILLIAMS, I. J. **Feedback and control systems**. 2. ed ed. New York: McGraw-Hill, 2012.

EPE. **Demanda de Energia 2050**: NOTA TÉCNICA DEA 13/15. Rio de Janeiro, RJ: Empresa de Pesquisa Energética, jan. 2016. Disponível em: <http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-227/topico-202/DEA%2013-15%20Demanda%20de%20Energia%202050.pdf>. Acesso em: 16 maio. 2018.

EPE. **Relatório Síntese do Balanço Energético Nacional 2017 – Ano base 2016**. Rio de Janeiro, RJ: Empresa de Pesquisa Energética, jun. 2017. Disponível em: <https://ben.epe.gov.br/BENRelatorioSintese.aspx?anoColeta=2017&anoFimColeta=2016>. Acesso em: 16 maio. 2018.

ESPRESSIF. **ESP8266ex Resources**. Disponível em: <https://www.espressif.com/en/products/hardware/esp8266ex/resources>. Acesso em: 10 nov. 2018.

INFLUXDATA. **InfluxDB Documentation**. Disponível em: <https://docs.influxdata.com/influxdb/v1.7/>. Acesso em: 14 mai. 2018.

INMET. **Normais Climatológicas do Brasil 1981-2010**. Disponível em: <http://www.inmet.gov.br/portal/index.php?r=clima/normaisClimatologicas>. Acesso em: 3 nov. 2018.

IOTBYTES. **NodeMCU Pinout**. Disponível em: <https://iotbytes.wordpress.com/nodemcu-pinout/>. Acesso em: 10 nov. 2018.

LIMA, J. B. A. **Otimização de sistema de aquecimento solar de água em edificações residenciais unifamiliares utilizando o programa TRNSYS.** text—[s.l.] Universidade de São Paulo, 17 dez. 2003.

MAXIM INTEGRATED. **DS18B20 Programmable Resolution1-Wire Digital Thermometer.** Disponível em: <https://datasheets.maximintegrated.com/en/ds/DS 18B20.pdf >. Acesso em: 10 out. 2018.

MORRISON, J. P. **Flow-based programming: a new approach to application development**. 2. ed ed. Unionville, Ont.: J. P. Morrison, 2011.

MQTT ORG. **MQTT FAQ**. Disponível em: <http://mqtt.org/faq>. Acesso em: 10 nov. 2018.

NEVES, J. C. **Avaliação Técnico-Econômica De Um Aquecedor Solar De Água Com Coletor De Tubos a Vácuo Na Cidade De Cascavel–PR**. [s.l.] Universidade Estadual do Oeste do Paraná, 2013.

NODERED ORG. **Documentation: User Guide**. Disponível em: <https://nodered.org/docs/>. Acesso em: 14 nov. 2018.

NODERED ORG. **Flows Mytimeout**. Disponível em: < https://flows.nodered.org/node/node-red-contrib-timeout> Acesso em: 5 mai. 2019.

NORWEGIAN CREATIONS. **MQTT – What Is It? And How Can You Use It?**. Disponível em: <https://www.norwegiancreations.com/2017/07/mqtt-what-is-it-and-how-can-you-use-it/>. Acesso em: 14 nov. 2018.

RINNAI. **Sinergia entre aquecedores a gás e sistemas solares**. Disponível em: <http://www.rinnai.com.br/almanaque/sinergia-solares-2017>. Acesso em: 1 nov. 2018.

SIQUEIRA, D. A. **Estudo do desempenho de um aquecedor solar de baixo custo. Uberlândia**. Dissertação de mestrado Universidade Federal de Uberlândia. 2009.

APÊNDICE A

Este apêndice descreve o código exemplo utilizado no microcontrolador ESP8266 para coleta dos dados de temperatura e acionamento das fontes.

Arquivo: main.cpp

Função: *Setup* inicial e loop.

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <PubSubClient.h>
#include <time.h>
#include <ArduinoOTA.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <RingBuf.h>
#include <TelnetSpy.h>

TelnetSpy LOG;

//#define DEBUG Serial
#define DEBUG LOG

#define HOSTNAME "ESP8266-Solar"

const char ssid[] = "NOME_WIFI";
const char pass[] = "SENHA_WIFI";
const char clientId[] = HOSTNAME;
const char username[] = "username_mqtt";
const char password[] = "senha_mqtt";

static const char fp[] PROGMEM = "A0:81:7C:41:7A:4C:7D:33:CB:4A:54:02:C6:A6:C2:67:30:A0:E2:8A";
const char *   host = "ihle.com.br";
const uint16_t port = 8883;
const char *   path = "/";

BearSSL::WiFiClientSecure net;
//MQTTClient client;
PubSubClient client(net);

/***************************** Pinos ***************************************/
int releele=16; // D0
int relegas=5; // D2
int relecirc=4; // D1

/************************* Variaveis ***********************************/
long lastReconnectAttempt = 0;
unsigned long cincosec=5000,lastcincosec,dezsec=10000,lastdezsec,trintasec=30000,lasttrintasec,lastminuto,minuto=60000;//tempo loop
float temp_boiler,temp_retornoc,temp_retornog,temp_saidagc,temp_probe;
char temp_boiler_env[10]="00.00",temp_retornoc_env[10]="00.00",temp_retornog_env[10]="00.00",temp_saidagc_env[10]="00.00",temp_probe_env[10]="00.00";

unsigned long lastMillis = 0,now =0;
time_t tnow;

```

```
48. #include <sensores.h>
49. #include <mqtt_func.h>
50. #include <functions.h>
51. 
52. // Set time via NTP
53. void setClock() {
54.  configTime(-3 * 3600, 1, "pool.ntp.org", "time.nist.gov");
55. 
56.  DEBUG.print("Waiting for NTP time sync: ");
57.  tnow = time(nullptr);
58.  while (tnow < 8 * 3600 * 2) {
59.    delay(500);
60.    DEBUG.print(".");
61.    ArduinoOTA.handle();
62.    tnow = time(nullptr);
63.  }
64.  DEBUG.println("");
65.  struct tm timeinfo;
66.  gmtime_r(&tnow, &timeinfo);
67.  DEBUG.print("Current time: ");
68.  DEBUG.print(asctime(&timeinfo));
69. }
70. 
71. 
72. void setup() {
73.  LOG.setWelcomeMsg(HOSTNAME ": Welcome to the Telnet\n\n");
74.  DEBUG.begin(115200);
75.  DEBUG.println();
76.  DEBUG.println();
77. //Setup Pinos
78.   pinMode(relegas, OUTPUT);
79.   pinMode(relecirc, OUTPUT);
80.   pinMode(releele, OUTPUT);
81.   digitalWrite(relegas, LOW);
82.   digitalWrite(relecirc, LOW);
83.   digitalWrite(releele, LOW);
84. 
85.  setup_sensores();
86. // We start by connecting to a WiFi network
87.  DEBUG.print("Connecting to ");
88.  DEBUG.println(ssid);
89.  WiFi.mode(WIFI_STA);
90.  WiFi.hostname(HOSTNAME);
91.  ArduinoOTA.setHostname(HOSTNAME );
92. 
93.     WiFi.begin(ssid, pass);
94.     while (WiFi.status() != WL_CONNECTED)
95.     {
96.         DEBUG.print(".");
97.         delay(500);
98.     }
99.     DEBUG.println("connected!");
100. 
101.  DEBUG.println("");
102. 
103.  DEBUG.println("WiFi connected");
104.  DEBUG.println("IP address: ");
105.  DEBUG.println(WiFi.localIP());
106. 
107. 
108. 
109. //Arduino OTA
110.   ArduinoOTA.onStart([]() {
111.     String type;
112.     if (ArduinoOTA.getCommand() == U_FLASH)
```

```
113.       type = "sketch";
114.     else // U_SPIFFS
115.       type = "filesystem";
116.
117.     // NOTE: if updating SPIFFS this would be the place to unmount SPIFFS using SPIF
    FS.end()
118.     DEBUG.println("Start updating " + type);
119.   });
120.   ArduinoOTA.onEnd([]() {
121.     DEBUG.println("\nEnd");
122.   });
123.   ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
124.     DEBUG.printf("Progress: %u%%\r", (progress / (total / 100)));
125.   });
126.   ArduinoOTA.onError([](ota_error_t error) {
127.     DEBUG.printf("Error[%u]: ", error);
128.     if (error == OTA_AUTH_ERROR) DEBUG.println("Auth Failed");
129.     else if (error == OTA_BEGIN_ERROR) DEBUG.println("Begin Failed");
130.     else if (error == OTA_CONNECT_ERROR) DEBUG.println("Connect Failed");
131.     else if (error == OTA_RECEIVE_ERROR) DEBUG.println("Receive Failed");
132.     else if (error == OTA_END_ERROR) DEBUG.println("End Failed");
133.   });
134.   ArduinoOTA.begin();
135.
136.  setClock();
137.   net.setInsecure();
138.  //fetchFingerprint();
139.   //fetchKnownKey();
140.  //fetchCertAuthority();
141.
142.  // Note: Local domain names (e.g. "Computer.local" on OSX) are not supported by Ard
    uino.
143.  // You need to set the IP address directly.
144.  //
145.  // MQTT brokers usually use port 8883 for secure connections.
146.  //client.setWill(HOSTNAME "/stats/conectado", ctime(&tnow));
147.  //client.begin(host, port, net);
148.  //client.onMessage(messageReceived);
149.  client.setServer(host, port);
150.  client.setCallback(callback);
151.  lastReconnectAttempt = 0;
152.  mqtt_connect();
153. }
154.
155. void loop() {
156.  tnow = time(nullptr);
157.  now = millis();
158.  ArduinoOTA.handle();
159.   LOG.handle();
160.
161.  if (!client.connected()) {
162.     if (now - lastReconnectAttempt > 5000) {
163.       lastReconnectAttempt = now;
164.       // Attempt to reconnect
165.       if (mqtt_connect()) {
166.         lastReconnectAttempt = 0;
167.       }
168.     }
169.   } else {
170.     // Client connected
171.
172.     client.loop();
173. }
174.
```

```
175./****************************** EXECUTA A CADA 5SEG *******************************
    *******/
176.if (now - lastcincosec > cincosec) {
177.  lastcincosec = now;
178.  //DEBUG.println("5 SEG");
179.
180.  //COMANDOS PARA EXECUCAO A CADA 5s
181.  if(deice_flag == true){
182.    deice();
183.  }
184.
185.  if(circf_flag == true){
186.    circf();
187.  }
188.
189.}
190./****************************** EXECUTA A CADA 10SEG ******************************
    ********/
191.if (now - lastdezsec > dezsec) {
192.  lastdezsec = now;
193.  //DEBUG.println("10 SEG");
194.
195.  if(bomba_flag == true){
196.    leitura_sensores();
197.    //publica_temp();
198.    circulacao();
199.  }
200.
201.  //COMANDOS PARA EXECUCAO A CADA 10s
202.
203.
204.}
205./****************************** EXECUTA A CADA 30SEG ******************************
    ********/
206.if (now - lasttrintasec > trintasec) {
207.  lasttrintasec = now;
208.  //DEBUG.println("30 SEG");
209.
210.  //COMANDOS PARA EXECUCAO A CADA 30s
211.
212.  leitura_sensores();
213.  publica_temp();
214.
215.}
216./****************************** EXECUTA A CADA 1MIN *******************************
    *******/
217.if (now - lastminuto > minuto) {
218.  lastminuto = now;
219.  //DEBUG.println("1 min");
220.
221.  //COMANDOS PARA EXECUCAO A CADA 1min
222.  deice();
223.  circulacao();
224.  client.publish(HOSTNAME "/stats/uptime", ctime(&tnow), false);
225.
226.}
227.
228.
229.
230.//delay(10);  // <- fixes some issues with WiFi stability
231.yield();
232.}
```

Arquivo: Sensores.h

Função: *Setup* e leituras dos sensores de temperatura.

```
// Conectar o pino central dos sensores ao pino 12
#define ONE_WIRE_BUS 12
#define TEMPERATURE_PRECISION 10

// Setup a oneWire instance to communicate with any OneWire devices
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
DeviceAddress boiler = { 0x28, 0xFF, 0xF7, 0x28, 0x81, 0x16, 0x03, 0xBE };
DeviceAddress retornoc = { 0x28, 0xFF, 0xF4, 0x1E, 0x81, 0x16, 0x03, 0xB1 };
DeviceAddress retornog = { 0x28, 0xFF, 0x99, 0x53, 0x74, 0x16, 0x04, 0x0A };
DeviceAddress saidagc = { 0x28, 0xFF, 0xC5, 0x19, 0x81, 0x16, 0x05, 0xA3 };
DeviceAddress probe = { 0x28, 0xFF, 0xF4, 0x41, 0x80, 0x16, 0x04, 0xDB  };


void setup_sensores(){
  sensors.begin();
  // set the resolution to 10 bit (good enough?)
  sensors.setResolution(boiler, TEMPERATURE_PRECISION);
  sensors.setResolution(retornoc, TEMPERATURE_PRECISION);
  sensors.setResolution(retornog, TEMPERATURE_PRECISION);
  sensors.setResolution(saidagc, TEMPERATURE_PRECISION);
  sensors.setResolution(probe, TEMPERATURE_PRECISION);

  DEBUG.println();
  DEBUG.println("- - - - - - - - - -");
  DEBUG.println();
  DEBUG.print("Boiler Resolution: ");
  DEBUG.print(sensors.getResolution(boiler), DEC);
  DEBUG.println();
  DEBUG.print("RetornoC Resolution: ");
  DEBUG.print(sensors.getResolution(retornoc), DEC);
  DEBUG.println();
  DEBUG.print("RetornoG Resolution: ");
  DEBUG.print(sensors.getResolution(retornog), DEC);
  DEBUG.println();
  DEBUG.print("SaidaGC Resolution: ");
  DEBUG.print(sensors.getResolution(saidagc), DEC);
  DEBUG.println();
  DEBUG.print("Teste: ");
  DEBUG.print(sensors.getResolution(probe), DEC);
  DEBUG.println();
  DEBUG.println("- - - - - - - - - -");
  DEBUG.println();
}

float printTemperature(DeviceAddress deviceAddress)
{
  float tempC = sensors.getTempC(deviceAddress);
  if (tempC != DEVICE_DISCONNECTED_C)
  {
    //DEBUG.print("Current Temp C: ");
    DEBUG.print(tempC);
  }
  else DEBUG.print("DEVICE DISCONNECTED");
  return tempC;
}

```

```
60.
61. //Leitura Sensores
62.
63. void leitura_sensores(){
64.   DEBUG.println(ctime(&tnow));
65.   DEBUG.print("Lendo temperaturas...\n\r");
66.   sensors.requestTemperatures();
67.
68.   DEBUG.print("Temperatura Boiler: ");
69.   temp_boiler = printTemperature(boiler);
70.   DEBUG.print("\n\r");
71.
72.   DEBUG.print("Temperatura Retorno C: ");
73.   temp_retornoc = printTemperature(retornoc);
74.   DEBUG.print("\n\r");
75.
76.   DEBUG.print("Temperatura Retorno G: ");
77.   temp_retornog = printTemperature(retornog);
78.   DEBUG.print("\n\r");
79.
80.   DEBUG.print("Temperatura Saida GC: ");
81.   temp_saidagc = printTemperature(saidagc);
82.   DEBUG.print("\n\r");
83.
84.   DEBUG.print("Temperatura Probe: ");
85.   //temp_probe = printTemperature(probe);
86.   DEBUG.print("\n\r");
87.
88.   DEBUG.println("- - - - - - - - - -");
89.
90. }
91.
92. //BUSCA SENSORES
93. //Chamar usando a função:
94. //findDevices(ONE_WIRE_BUS);
95.
96. uint8_t findDevices(int pin)
97. {
98.   OneWire ow(pin);
99.
100.   uint8_t address[8];
101.   uint8_t count = 0;
102.
103.
104.   if (ow.search(address))
105.   {
106.     DEBUG.print("\nuint8_t pin");
107.     DEBUG.print(pin, DEC);
108.     DEBUG.println("[][8] = {");
109.     do {
110.       count++;
111.       DEBUG.println("  {");
112.       for (uint8_t i = 0; i < 8; i++)
113.       {
114.         DEBUG.print("0x");
115.         if (address[i] < 0x10) DEBUG.print("0");
116.         DEBUG.print(address[i], HEX);
117.         if (i < 7) DEBUG.print(", ");
118.       }
119.       DEBUG.println("  },");
120.     } while (ow.search(address));
121.
122.     DEBUG.println("};");
123.     DEBUG.print("// nr devices found: ");
124.     DEBUG.println(count);
```

```
125.   }
126.
127.   return count;
128. }
```

Arquivo: mqtt_func.h

Função: Publicar e receber mensagens nos tópicos MQTT.

```
uint8_t retries = 10;

boolean mqtt_connect() {
  if (client.connect(clientId, username, password, HOSTNAME "/stats/conectado", 0, 1
, ctime(&tnow), 1)) {
    // ... and resubscribe

    //Comandos
    client.subscribe(HOSTNAME "/cmd/circf");
    client.subscribe(HOSTNAME "/cmd/sensor_scan");
    client.subscribe(HOSTNAME "/cmd/reboot");
    //Config
    client.subscribe(HOSTNAME "/cfg/circulacao");
    client.subscribe(HOSTNAME "/cfg/deice");
    client.subscribe(HOSTNAME "/cfg/aquecimento");
    client.subscribe(HOSTNAME "/cfg/intervalo_deice");
    client.subscribe(HOSTNAME "/cfg/intervalo_circd");
    client.subscribe(HOSTNAME "/cfg/intervalo_bomba");
    client.subscribe(HOSTNAME "/cfg/intervalo_min_bomba");
    client.subscribe(HOSTNAME "/cfg/timeout_bomba");
    client.subscribe(HOSTNAME "/cfg/fator_bomba");
    client.subscribe(HOSTNAME "/cfg/temperatura_corte");
    DEBUG.println("\nConnected to MQTT!");
    retries = 10;
  }
  else {
    DEBUG.print("MQTT failed, rc=");
    DEBUG.println(client.state());
     retries--;
       if (retries == 0) {
         // basically die and wait for WDT to reset me
         DEBUG.print("MQTT: reset...");
         ESP.restart();
         //while (1);
        }
  }
  return client.connected();
}

void callback(char* topic, byte* payload, unsigned int length) {
  // handle message arrived
  //DEBUG.println("incoming: " + topic + " - " + payload);

  if(strcmp(topic, HOSTNAME "/cmd/reboot") == 0) {
    // the strings are equal
    DEBUG.println("Reiniciando...");
    client.publish(HOSTNAME "/stats/reboot", ctime(&tnow), false);
    delay(2000);
    ESP.restart();
  }
  //Verifica status acionamento gas
  if(strcmp(topic, "comando/glp") == 0) {
    // the strings are equal
    if(strcmp(payload, "on") == 0){
        //caso payload = on, liga gas
        liga_gas();
    }else{
        //caso payload != on, desliga gas
        desliga_gas();
```

```
    }
  }
  //Verifica status acionamento eletrico
    if(strcmp(topic, "comando/eletrico") == 0) {
    // the strings are equal
    if(strcmp(payload, "on") == 0){
        //caso payload = on, liga eletrico
        liga_ele();
    }else{
        //caso payload != on, desliga eletrico
        desliga_ele();
    }
  }


}

void publica_temp(){
  //CONVERTE
  dtostrf(temp_boiler,7,2,temp_boiler_env);
  dtostrf(temp_retornoc,7,2,temp_retornoc_env);
  dtostrf(temp_retornog,7,2,temp_retornog_env);
  dtostrf(temp_saidagc,7,2,temp_saidagc_env);

  //ENVIA
  client.publish(HOSTNAME "/temp/boiler", temp_boiler_env, false);
  client.publish(HOSTNAME "/temp/rcoletor", temp_retornoc_env, false);
  client.publish(HOSTNAME "/temp/rgas", temp_retornog_env, false);
  client.publish(HOSTNAME "/temp/sgas", temp_saidagc_env, false);
}
```

Arquivo: functions.h

Função: Executar os acionamentos das saídas a rele: circulação coletores reservatório, aquecimento gás, aquecimento elétrico e anti-congelamento.

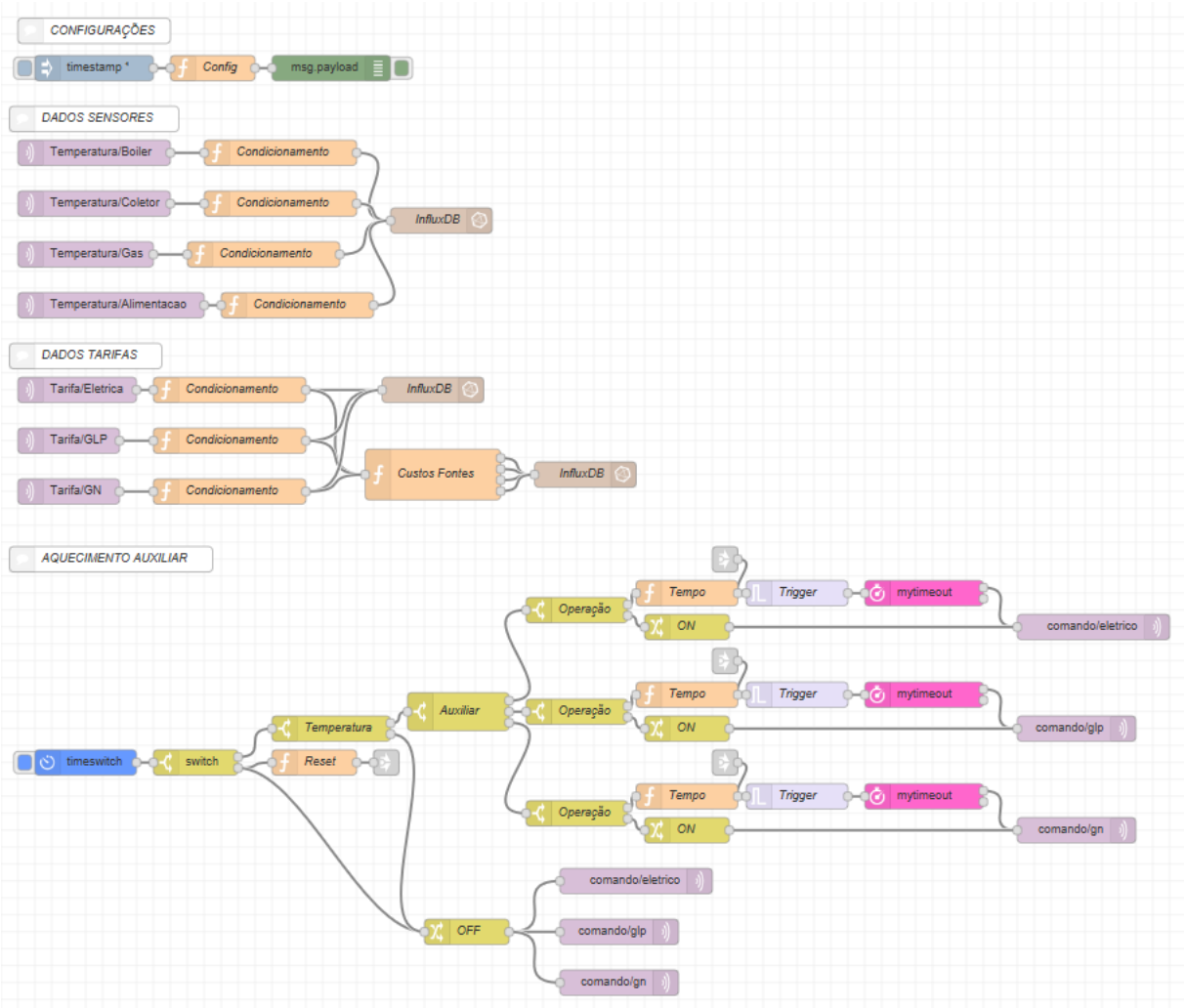
```
1.  boolean wifi_ativa=false,circ=false,circf_flag=false,deice_flag=false,circulacao_fla
    g=false,bomba_flag=false;
2.  boolean bomba_circ=true,bomba_gas=false;
3.  float temp_corte=60,fator_bomba=1.3; //temperatura para desligar bomba
4.  unsigned long intervalo_deice=30000,intervalo_circf=20000,mintime_bomba=15000,lastin
    tervalo_deice=0,lastintervalo_circf=0,timeout_bomba=60000,last_bomba=0;
5.  unsigned long intervalo_bomba=10*60000; //intervalo entre acionamentos bomba
6.
7.  void liga_bomba() {
8.    DEBUG.println("Liga Bomba");
9.    digitalWrite(relecirc, HIGH);
10.   bomba_flag = true;
11.
12. }
13. void desliga_bomba() {
14.   DEBUG.println("Desliga Bomba");
15.   digitalWrite(relecirc, LOW);
16.   bomba_flag = false;
17.
18. }
19.
20. void liga_gas() {
21.   DEBUG.println("Liga Aquecimento Gas");
22.   digitalWrite(relegas, HIGH);
23.   bomba_flag = true;
24.
25. }
26. void desliga_gas() {
27.   DEBUG.println("Desliga Aquecimento Gas");
28.   digitalWrite(relegas, LOW);
29.   bomba_flag = false;
30.
31. }
32.
33. void liga_ele() {
34.   DEBUG.println("Liga Aquecimento Eletrico");
35.   digitalWrite(releele, HIGH);
36.   bomba_flag = true;
37.
38. }
39. void desliga_ele() {
40.   DEBUG.println("Desliga Aquecimento Eletrico");
41.   digitalWrite(releele, LOW);
42.   bomba_flag = false;
43.
44. }
45.
46. void deice() {
47.   if ((deice_flag == true) && ((now - lastintervalo_deice) > intervalo_deice)) {
48.     DEBUG.println(F("Desliga Bomba ICE"));
49.     desliga_bomba();
50.     deice_flag = false;
51.   }
52.   else if ((temp_retornoc < 8) && (DEVICE_DISCONNECTED_C != temp_retornoc) && (deice
    _flag == false)) {
53.     DEBUG.println(F("Liga Bomba ICE"));
54.     lastintervalo_deice = now;
```

```
55.     deice_flag = true;
56.     liga_bomba();
57.   }
58. }
59.
60. void circf() {
61.   if ((circf_flag == true) && ((now - lastintervalo_circf) > intervalo_circf)) {
62.     DEBUG.println(F("Desliga Circ Forcada"));
63.     desliga_bomba();
64.     circf_flag = false;
65.     client.publish(HOSTNAME "/stats/circulacao", "Circulacao Forcada OFF", false);
66.   }
67.   else if (circf_flag == false) {
68.     DEBUG.println(F("Liga Circ Forcada"));
69.     lastintervalo_circf = now;
70.     circf_flag = true;
71.     liga_bomba();
72.     client.publish(HOSTNAME "/stats/circulacao", "Circulacao Forcada ON", false);
73.   }
74. }
75.
76.
77.
78. void circulacao() {
79.   //leitura temperatura consumo, retorno gas
80.   if (bomba_circ == false) {
81.     client.publish(HOSTNAME "/stats/circulacao", "Bomba Desativada", false);
82.     return;
83.   }
84.   else if (deice_flag == true) {
85.     client.publish(HOSTNAME "/stats/circulacao", "DeIce ON", false);
86.     return;
87.   }
88.   else if (circf_flag == true) {
89.     client.publish(HOSTNAME "/stats/circulacao", "Circulacao Forcada ON", false);
90.     return;
91.   }
92.   else if((temp_boiler == DEVICE_DISCONNECTED_C)||(temp_retornoc == DEVICE_DISCONNEC
    TED_C)||(temp_saidagc == DEVICE_DISCONNECTED_C)){
93.     client.publish(HOSTNAME "/stats/circulacao", "Erro Sensor", false);
94.     return;
95.   }
96.   else if ((temp_boiler > temp_corte) && (bomba_flag = true)) { //desliga por atingi
    r temperatura corte
97.     desliga_bomba();
98.     client.publish(HOSTNAME "/stats/circulacao", "Bomba Off: Temperatura Corte", fal
    se);
99.   }
100.   else if((bomba_flag==true) && ((now - last_bomba)>timeout_bomba)){ //desliga atrav
    es do timeout
101.     desliga_bomba();
102.     client.publish(HOSTNAME "/stats/circulacao", "Bomba Off: Timeout", false);
103.   }
104.   else if ((temp_boiler > temp_corte) && (bomba_flag==false)) { //temperatura corte

105.     client.publish(HOSTNAME "/stats/circulacao", "Temperatura Corte Boiler", false);

106.     return;
107.   }
108.   else if ((28 > temp_retornoc) && (bomba_flag = false)) { //temperatura coletor ins
    uficiente
109.     client.publish(HOSTNAME "/stats/circulacao", "Temperatura Coletor Baixa", false)
    ;
110.     return;
111.   }
```

```
112.  else if ((28 > temp_retornoc) && (bomba_flag = true)) { //desliga por temperatura
    insuficiente coletor
113.    desliga_bomba();
114.    client.publish(HOSTNAME "/stats/circulacao", "Bomba Off: Temperatura Coletor Bai
    xa", false);
115.  }
116.  else if ((bomba_flag == false) && ((temp_boiler*fator_bomba) < temp_retornoc) && (
    (temp_saidagc + 4) < temp_retornoc)) { //liga atraves do diferenciais de temperatura

117.    last_bomba = now;
118.    liga_bomba();
119.    client.publish(HOSTNAME "/stats/circulacao", "Bomba On: Circulacao", false);
120.  }
121.  else if ((bomba_flag == true) && ((now - last_bomba) > mintime_bomba) && (((temp_b
    oiler + 4) > temp_retornoc) || ((temp_saidagc + 2) > temp_retornoc))) { //desliga at
    raves do diferenciais de temperatura
122.    desliga_bomba();
123.    client.publish(HOSTNAME "/stats/circulacao", "Bomba Off: Circulacao", false);
124.  }
125.}
```

APÊNDICE B

Este apêndice descreve o código completo do sistema de controle proposto utilizado na ferramenta Node-RED.



O código pode ser importado no Node-RED através do menu: “Import → Clipboard”

```
1. [{"id":"efe4729c.04a598","type":"tab","label":"TCC","disabled":false,"info":""},{"id
   ":"f9d66a9e.e9b63","type":"comment","z":"efe4729c.04a598","name":"CONFIGURAÇÕES","in
   fo":"Teste","x":110,"y":36,"wires":[]},{"id":"965e6736.23ab28","type":"inject","z":"
   efe4729c.04a598","name":"","topic":"","payload":"","payloadType":"date","repeat":"",
   "crontab":"","once":true,"onceDelay":0.1,"x":110,"y":80,"wires":[["3f8e9cf4.47425c"]
   ]},{"id":"3f8e9cf4.47425c","type":"function","z":"efe4729c.04a598","name":"Config","
   func":"//DADOS DO SISTEMA\nglobal.set(\"aux_ele\",true);\nglobal.set(\"aux_gn\",true
   );\nglobal.set(\"aux_glp\",true);\n\n//Ajustes\nglobal.set(\"temperatura_banho\",42)
   ;\nglobal.set(\"qtd_banhos\",4);\nglobal.set(\"m_h2o\",60); // Massa de água por ban
   ho (em kg)\nglobal.set(\"operacao\",\"quantidade\"); //ajusta operação do sistema pa
   ra quantidade de agua aquecida (quantidade banhos)\n//global.set(\"operacao\",\"temp
   eratura\"); //ajusta operação do sistema para operação por temperatura (aquecimento
   todo reservatório)\n\n//Apoio Eletrico\nglobal.set(\"rend_ele\",0.99); // Rendimento
   \nglobal.set(\"pot_ele\",3000); // Potencia em Watts\n\n//Apoio GLP\nglobal.set(\"re
```

```
nd_glp\",0.84); // Rendimento\nglobal.set(\"cons_glp\",0.75); // Consumo em kg/h\ngl
obal.set(\"pot_glp\",10000); // Potencia em Watts\n\n//Apoio GN\nglobal.set(\"rend_g
n\",0.85); // Rendimento\nglobal.set(\"cons_gn\",0.9); // Consumo em m³/h\nglobal.se
t(\"pot_gn\",10000); // Potencia em Watts\n\n// CONSTANTES\nglobal.set(\"ce_h2o\",1.
163); // Calor especifico da agua em (Wh / kg.ºC)\nglobal.set(\"vol_boiler\",400); /
/Volume reservatorio (l)\n//global.set(\"vol_util\",0.75); //Volume util reservatori
o (%)\n\nreturn msg;","outputs":1,"noerr":0,"x":250,"y":80,"wires":[["5b150221.c07d5
4"]]},{"id":"5b150221.c07d54","type":"debug","z":"efe4729c.04a598","name":"","active
":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":390,
"y":80,"wires":[]},{"id":"38255708.65da9","type":"comment","z":"efe4729c.04a598","na
me":"DADOS SENSORES","info":"","x":110,"y":140,"wires":[]},{"id":"ef88ce59.e11528","
type":"function","z":"efe4729c.04a598","name":"Condicionamento","func":"var original
val = msg.payload; //Carrega dado do payload MQTT\nvar temp = \"0.00\"; //Declara va
riavel para temperatura\n\n// converte valor originalval para float com duas casas d
ecimais\ntemp = parseFloat(originalval).toFixed(2)-
0.00;\n\n// carrega dado para variavel global Node-
RED\nglobal.set(\"temperatura_boiler\",temp);\n\n// prepara saída para gravação banc
o de dados InfluxDB \nmsg.payload = {\n    temperatura: temp,\n    sensor: \"boiler\
",\n}\n\nreturn msg;","outputs":1,"noerr":0,"x":330,"y":180,"wires":[["23594b07.de87
a4"]]},{"id":"bae39846.2519","type":"mqtt in","z":"efe4729c.04a598","name":"","topic
":"Temperatura/Boiler","qos":"0","datatype":"auto","broker":"16b1d16d.b1f81f","x":11
0,"y":180,"wires":[["ef88ce59.e11528"]]},{"id":"c1015dad.cf03a8","type":"function","
z":"efe4729c.04a598","name":"Condicionamento","func":"var originalval = msg.payload;
 //Carrega dado do payload MQTT\nvar temp = \"0.00\"; //Declara variavel para temper
atura\n\n// converte valor originalval para float com duas casas decimais\ntemp = pa
rseFloat(originalval).toFixed(2)-0.00;\n\n// carrega dado para variavel global Node-
RED\nglobal.set(\"temperatura_coletor\",temp);\n\n// prepara saída para gravação ban
co de dados InfluxDB \nmsg.payload = {\n    temperatura: temp,\n    sensor: \"coleto
r\",\n}\n\nreturn msg;","outputs":1,"noerr":0,"x":330,"y":240,"wires":[["23594b07.de
87a4"]]},{"id":"50c400f5.38bf88","type":"mqtt in","z":"efe4729c.04a598","name":"","t
opic":"Temperatura/Coletor","qos":"2","datatype":"auto","broker":"16b1d16d.b1f81f","
x":110,"y":240,"wires":[["c1015dad.cf03a8"]]},{"id":"23594b07.de87a4","type":"influx
db out","z":"efe4729c.04a598","influxdb":"6f62f03f.fecbd8","name":"InfluxDB","measur
ement":"Temperaturas","precision":"","retentionPolicy":"","x":520,"y":260,"wires":[]
},{"id":"750e48e7.b4b3e8","type":"function","z":"efe4729c.04a598","name":"Condiciona
mento","func":"var originalval = msg.payload; //Carrega dado do payload MQTT\nvar te
mp = \"0.00\"; //Declara variavel para temperatura\n\n// converte valor originalval
para float com duas casas decimais\ntemp = parseFloat(originalval).toFixed(2)-
0.00;\n\n// carrega dado para variavel global Node-
RED\nglobal.set(\"temperatura_gas\",temp);\n\n// prepara saída para gravação banco d
e dados InfluxDB \nmsg.payload = {\n    temperatura: temp,\n    sensor: \"gas\",\n}\
n\nreturn msg;","outputs":1,"noerr":0,"x":310,"y":300,"wires":[["23594b07.de87a4"]]}
,{"id":"d25bf076.b7d298","type":"mqtt in","z":"efe4729c.04a598","name":"","topic":"T
emperatura/Gas","qos":"2","datatype":"auto","broker":"16b1d16d.b1f81f","x":100,"y":3
00,"wires":[["750e48e7.b4b3e8"]]},{"id":"c8fe73e3.6141","type":"function","z":"efe47
29c.04a598","name":"Condicionamento","func":"var originalval = msg.payload; //Carreg
a dado do payload MQTT\nvar temp = \"0.00\"; //Declara variavel para temperatura\n\n
// converte valor originalval para float com duas casas decimais\ntemp = parseFloat(
originalval).toFixed(2)-0.00;\n\n// carrega dado para variavel global Node-
RED\nglobal.set(\"temperatura_alim\",temp);\n\n// prepara saída para gravação banco
de dados InfluxDB \nmsg.payload = {\n    temperatura: temp,\n    sensor: \"alim\",\n
}\n\nreturn msg;","outputs":1,"noerr":0,"x":350,"y":360,"wires":[["23594b07.de87a4"]
]},{"id":"31bd1d18.5e5bfa","type":"mqtt in","z":"efe4729c.04a598","name":"","topic":
"Temperatura/Alimentacao","qos":"2","datatype":"auto","broker":"16b1d16d.b1f81f","x"
:130,"y":360,"wires":[["c8fe73e3.6141"]]},{"id":"6a9fdfcf.6a12e8","type":"mqtt in","
z":"efe4729c.04a598","name":"","topic":"Tarifa/Eletrica","qos":"0","datatype":"auto"
,"broker":"16b1d16d.b1f81f","x":90,"y":460,"wires":[["a6ac7300.be93a"]]},{"id":"a6ac
7300.be93a","type":"function","z":"efe4729c.04a598","name":"Condicionamento","func":
"var originalval = msg.payload; //Carrega dado do payload MQTT\nvar tar = \"0.0000\"
; //Declara variavel para temperatura\n\n// converte valor originalval para float co
m quatro casas decimais\ntar = parseFloat(originalval).toFixed(4)-
0.0000;\n\n// carrega dado para variavel global Node-
RED\nglobal.set('tar_ele',tar);\n\n// prepara saída para gravação banco de dados Inf
luxDB \nmsg.payload = {\n    tarifa: tar,\n    topic: \"eletrica\",\n}\n\nreturn msg
;","outputs":1,"noerr":0,"x":270,"y":460,"wires":[["fa67e09e.2bfe28","368ee472.13bb4
```

```
c"]]},{"id":"184c36b0.566761","type":"mqtt in","z":"efe4729c.04a598","name":"","topi
c":"Tarifa/GLP","qos":"0","datatype":"auto","broker":"16b1d16d.b1f81f","x":80,"y":52
0,"wires":[["8d74c810.0866a8"]]},{"id":"fa67e09e.2bfe28","type":"influxdb out","z":"
efe4729c.04a598","influxdb":"6f62f03f.fecbd8","name":"InfluxDB","measurement":"Tarif
as","precision":"","retentionPolicy":"","x":510,"y":460,"wires":[]},{"id":"8d74c810.
0866a8","type":"function","z":"efe4729c.04a598","name":"Condicionamento","func":"var
 originalval = msg.payload; //Carrega dado do payload MQTT\nvar tar = \"0.0000\"; //
Declara variavel para temperatura\n\n// converte valor originalval para float com qu
atro casas decimais\ntar = parseFloat(originalval).toFixed(4)-
0.0000;\n\n// carrega dado para variavel global Node-
RED\nglobal.set('tar_glp',tar);\n\n// prepara saída para gravação banco de dados Inf
luxDB \nmsg.payload = {\n    tarifa: tar,\n    topic: \"glp\",\n}\n\nreturn msg;","o
utputs":1,"noerr":0,"x":270,"y":520,"wires":[["fa67e09e.2bfe28","368ee472.13bb4c"]]}
,{"id":"208708b0.26ff9","type":"mqtt in","z":"efe4729c.04a598","name":"","topic":"Ta
rifa/GN","qos":"0","datatype":"auto","broker":"16b1d16d.b1f81f","x":80,"y":580,"wire
s":[["c744be7d.e46b28"]]},{"id":"c744be7d.e46b28","type":"function","z":"efe4729c.04
a598","name":"Condicionamento","func":"var originalval = msg.payload; //Carrega dado
 do payload MQTT\nvar tar = \"0.0000\"; //Declara variavel para temperatura\n\n// co
nverte valor originalval para float com quatro casas decimais\ntar = parseFloat(orig
inalval).toFixed(4)-0.0000;\n\n// carrega dado para variavel global Node-
RED\nglobal.set('tar_gn',tar);\n\n// prepara saída para gravação banco de dados Infl
uxDB \nmsg.payload = {\n    tarifa: tar,\n    topic: \"gn\",\n}\n\nreturn msg;","out
puts":1,"noerr":0,"x":270,"y":580,"wires":[["fa67e09e.2bfe28","368ee472.13bb4c"]]},{
"id":"d1727e27.140e48","type":"comment","z":"efe4729c.04a598","name":"DADOS TARIFAS"
,"info":"","x":100,"y":420,"wires":[]},{"id":"368ee472.13bb4c","type":"function","z"
:"efe4729c.04a598","name":"Custos Fontes","func":"//Calcula o custo por fonte\n\nvar
 tar_ele = global.get('tar_ele');\nvar tar_glp = global.get('tar_glp');\nvar tar_gn
= global.get('tar_gn');\n\nvar rend_ele = global.get(\"rend_ele\")\n\nvar pot_glp =
global.get(\"pot_glp\");\nvar cons_glp = global.get(\"cons_glp\");\nvar rend_glp = g
lobal.get(\"rend_glp\");\n\nvar pot_gn = global.get(\"pot_gn\");\nvar cons_gn = glob
al.get(\"cons_gn\");\nvar rend_gn = global.get(\"rend_gn\");\n\nvar custo_aux_ele =
((1000/rend_ele)/1000)*tar_ele;\nvar custo_aux_glp = (((1000/rend_glp)/(pot_glp))*co
ns_glp)*(tar_glp/13);\nvar custo_aux_gn = (((1000/rend_gn)/(pot_gn))*cons_gn)*tar_gn
;\n\n//Define Arrays\nvar custos = [];\nvar nome_fontes = [];\n\nif(global.get(\"aux
_ele\")===true){\n    custos.push(custo_aux_ele);\n    nome_fontes.push(\"Elétrico\"
);\n}\n\nif(global.get(\"aux_gn\")===true){\n    custos.push(custo_aux_gn);\n    nom
e_fontes.push(\"Gás Natural\");\n}\n\nif(global.get(\"aux_glp\")===true){\n    custo
s.push(custo_aux_glp);\n    nome_fontes.push(\"GLP\");\n}\n\nvar melhor_fonte = nome
_fontes[custos.indexOf(Math.min.apply(Math,custos))];\n\nif(global.get(\"aux_glp\")=
==false&&global.get(\"aux_gn\")===false&&global.get(\"aux_ele\")===false){\n    melh
or_fonte=\"Desativado\";\n}\n\nglobal.set(\"melhor_fonte\",melhor_fonte)\n\nvar msg1
 = {payload: {custo:custo_aux_ele,topic:\"eletrica\"}};\nvar msg2 = {payload: {custo
:custo_aux_glp,topic:\"glp\"}};\nvar msg3 = {payload: {custo:custo_aux_gn,topic:\"gn
\"}};\nvar msg4 = {payload: {fonte:melhor_fonte,topic:\"melhor\"}};\n\nreturn [msg1,
msg2,msg3,msg4];","outputs":4,"noerr":0,"x":510,"y":560,"wires":[["7e765a59.a6e84c"]
,["7e765a59.a6e84c"],["7e765a59.a6e84c"],["7e765a59.a6e84c"]]},{"id":"7e765a59.a6e84
c","type":"influxdb out","z":"efe4729c.04a598","influxdb":"6f62f03f.fecbd8","name":"
InfluxDB","measurement":"Custo","precision":"","retentionPolicy":"","x":690,"y":560,
"wires":[]},{"id":"18cc7966.2cceef","type":"timeswitch","z":"efe4729c.04a598","name"
:"","mytopic":"","lat":"-25.43965","lon":"-
49.26777","starttime":"6000","endtime":"0","duskoff":"0","dawnoff":"0","sun":true,"m
on":true,"tue":true,"wed":true,"thu":true,"fri":true,"sat":true,"jan":true,"feb":tru
e,"mar":true,"apr":true,"may":true,"jun":true,"jul":true,"aug":true,"sep":true,"oct"
:true,"nov":true,"dec":true,"x":100,"y":900,"wires":[["fe6ebfa4.d659e8"]]},{"id":"fe
6ebfa4.d659e8","type":"switch","z":"efe4729c.04a598","name":"","property":"payload",
"propertyType":"msg","rules":[{"t":"eq","v":"1","vt":"num"},{"t":"eq","v":"0","vt":"
num"}],"checkall":"true","repair":false,"outputs":2,"x":230,"y":900,"wires":[["4e507
cb8.42b394"],["a8a3d607.17492","2d566f30.95d1d8"]]},{"id":"4e507cb8.42b394","type":"
switch","z":"efe4729c.04a598","name":"Temperatura","property":"temperatura_boiler","
propertyType":"global","rules":[{"t":"lt","v":"temperatura_banho","vt":"global"},{"t
":"gte","v":"temperatura_banho","vt":"global"}],"checkall":"true","repair":false,"ou
tputs":2,"x":390,"y":860,"wires":[["72034dd8.83b3ac"],["2d566f30.95d1d8"]]},{"id":"a
8eae99f.7fb658","type":"mqtt out","z":"efe4729c.04a598","name":"","topic":"comando/e
letrico","qos":"","retain":"","broker":"16b1d16d.b1f81f","x":750,"y":1040,"wires":[]
},{"id":"f3c29931.aa8c88","type":"mqtt out","z":"efe4729c.04a598","name":"","topic":
```

```
"comando/glp","qos":"","retain":"","broker":"16b1d16d.b1f81f","x":730,"y":1100,"wire
s":[]},{"id":"1e39cc60.691ffc","type":"mqtt out","z":"efe4729c.04a598","name":"","to
pic":"comando/gn","qos":"","retain":"","broker":"16b1d16d.b1f81f","x":730,"y":1160,"
wires":[]},{"id":"72034dd8.83b3ac","type":"switch","z":"efe4729c.04a598","name":"Aux
iliar","property":"melhor_fonte","propertyType":"global","rules":[{"t":"eq","v":"Elé
trico","vt":"str"},{"t":"eq","v":"Gás Natural","vt":"str"},{"t":"eq","v":"GLP","vt":
"str"}],"checkall":"false","repair":false,"outputs":3,"x":540,"y":840,"wires":[["952
b65ae.97e528"],["85b5f215.5d2b6"],["576b306e.724b6"]]},{"id":"dbc207d5.8e184","type"
:"mytimeout","z":"efe4729c.04a598","name":"","outtopic":"","outsafe":"on","outwarnin
g":"","outunsafe":"off","warning":"10","timer":"5","debug":false,"ndebug":false,"ign
oreCase":false,"repeat":false,"again":false,"x":1090,"y":700,"wires":[["8540c051.623
b08"],[]]},{"id":"eb5b1724.3f7f2","type":"mytimeout","z":"efe4729c.04a598","name":""
,"outtopic":"","outsafe":"on","outwarning":"","outunsafe":"off","warning":"10","time
r":"5","debug":false,"ndebug":false,"ignoreCase":false,"repeat":false,"again":false,
"x":1090,"y":820,"wires":[["ebd76a88.a6ced8"],[]]},{"id":"27e07522.f5267a","type":"m
ytimeout","z":"efe4729c.04a598","name":"","outtopic":"","outsafe":"on","outwarning":
"","outunsafe":"off","warning":"10","timer":"5","debug":false,"ndebug":false,"ignore
Case":true,"repeat":false,"again":false,"x":1090,"y":940,"wires":[["2ee2a4d9.8347e4"
],[]]},{"id":"9600a43.9c0edd8","type":"function","z":"efe4729c.04a598","name":"Tempo
","func":"//Carrega variaveis\nce_h2o = global.get('ce_h2o');\nm_h2o = global.get('m
_h2o');\nqtd_banhos = global.get('qtd_banhos');\ntemperatura_banho = global.get('tem
peratura_banho');\ntemp_boiler_final = global.get('temperatura_boiler');\nrendimento
 = global.get('rend_ele');\npotencia = global.get('pot_ele');\n\n//Calcula quantidad
e de energia térmica necessária\nvar qtd_ene_nes = (ce_h2o * m_h2o * qtd_banhos * (t
emperatura_banho - temp_boiler_final))/rendimento;\n\n//Calcula tempo em segundos pa
ra \nvar timer = Math.round((qtd_ene_nes/potencia)*3600);\n\nmsg.payload = {\n    \"
payload\": \"on\",\n    \"timeout\": timer,\n    \"warning\": 30\n}\n\nreturn msg;",
"outputs":1,"noerr":0,"x":810,"y":700,"wires":[["d114bd2d.ece9f"]]},{"id":"7c3736e6.
c996a8","type":"function","z":"efe4729c.04a598","name":"Tempo","func":"//Carrega var
iaveis\nce_h2o = global.get('ce_h2o');\nm_h2o = global.get('m_h2o');\nqtd_banhos = g
lobal.get('qtd_banhos');\ntemperatura_banho = global.get('temperatura_banho');\ntemp
_boiler_final = global.get('temperatura_boiler');\nrendimento = global.get('rend_glp
');\npotencia = global.get('pot_glp');\n\n//Calcula quantidade de energia térmica ne
cessária\nvar qtd_ene_nes = (ce_h2o * m_h2o * qtd_banhos * (temperatura_banho - temp
_boiler_final))/rendimento;\n\n//Calcula tempo em segundos para \nvar timer = Math.r
ound((qtd_ene_nes/potencia)*3600);\n\nmsg.payload = {\n    \"payload\": \"on\",\n
 \"timeout\": timer,\n    \"warning\": 30\n}\n\nreturn msg;","outputs":1,"noerr":0,"
x":810,"y":820,"wires":[["43b082c.c7486fc"]]},{"id":"8a6aa625.fc4bc8","type":"functi
on","z":"efe4729c.04a598","name":"Tempo","func":"//Carrega variaveis\nce_h2o = globa
l.get('ce_h2o');\nm_h2o = global.get('m_h2o');\nqtd_banhos = global.get('qtd_banhos'
);\ntemperatura_banho = global.get('temperatura_banho');\ntemp_boiler_final = global
.get('temperatura_boiler');\nrendimento = global.get('rend_gn');\npotencia = global.
get('pot_gn');\n\n//Calcula quantidade de energia térmica necessária\nvar qtd_ene_ne
s = (ce_h2o * m_h2o * qtd_banhos * (temperatura_banho - temp_boiler_final))/rendimen
to;\n\n//Calcula tempo em segundos para \nvar timer = Math.round((qtd_ene_nes/potenc
ia)*3600);\n\nmsg.payload = {\n    \"payload\": \"on\",\n    \"timeout\": timer,\n
  \"warning\": 30\n}\n\nreturn msg;","outputs":1,"noerr":0,"x":810,"y":940,"wires":[
["2e38b734.80bdc"]]},{"id":"8540c051.623b08","type":"mqtt out","z":"efe4729c.04a598"
,"name":"","topic":"comando/eletrico","qos":"","retain":"","broker":"16b1d16d.b1f81f
","x":1290,"y":740,"wires":[]},{"id":"ebd76a88.a6ced8","type":"mqtt out","z":"efe472
9c.04a598","name":"","topic":"comando/glp","qos":"","retain":"","broker":"16b1d16d.b
1f81f","x":1270,"y":860,"wires":[]},{"id":"2ee2a4d9.8347e4","type":"mqtt out","z":"e
fe4729c.04a598","name":"","topic":"comando/gn","qos":"","retain":"","broker":"16b1d1
6d.b1f81f","x":1270,"y":980,"wires":[]},{"id":"952b65ae.97e528","type":"switch","z":
"efe4729c.04a598","name":"Operação","property":"operacao","propertyType":"global","r
ules":[{"t":"eq","v":"quantidade","vt":"str"},{"t":"eq","v":"temperatura","vt":"str"
}],"checkall":"true","repair":false,"outputs":2,"x":680,"y":720,"wires":[["9600a43.9
c0edd8"],["83e1d23e.490358"]]},{"id":"85b5f215.5d2b6","type":"switch","z":"efe4729c.
04a598","name":"Operação","property":"operacao","propertyType":"global","rules":[{"t
":"eq","v":"quantidade","vt":"str"},{"t":"eq","v":"temperatura","vt":"str"}],"checka
ll":"true","repair":false,"outputs":2,"x":680,"y":840,"wires":[["7c3736e6.c996a8"],[
"78858e79.2917a8"]]},{"id":"576b306e.724b6","type":"switch","z":"efe4729c.04a598","n
ame":"Operação","property":"operacao","propertyType":"global","rules":[{"t":"eq","v"
:"quantidade","vt":"str"},{"t":"eq","v":"temperatura","vt":"str"}],"checkall":"true"
,"repair":false,"outputs":2,"x":680,"y":960,"wires":[["8a6aa625.fc4bc8"],["c674b675.
```

```
ad8948"]]},{"id":"92471bb3.2cfbf","type":"link out","z":"efe4729c.04a598","name":"RE
SET","links":["d16124fc.a22bb","ec92b540.73f328","31d961a5.a22a16"],"x":455,"y":900,
"wires":[]},{"id":"2e38b734.80bdc","type":"trigger","z":"efe4729c.04a598","op1":"","
op2":"0","op1type":"pay","op2type":"str","duration":"0","extend":false,"units":"ms",
"reset":"","bytopic":"all","name":"Trigger","x":940,"y":940,"wires":[["27e07522.f526
7a"]]},{"id":"43b082c.c7486fc","type":"trigger","z":"efe4729c.04a598","op1":"","op2"
:"0","op1type":"pay","op2type":"str","duration":"0","extend":false,"units":"ms","res
et":"","bytopic":"all","name":"Trigger","x":940,"y":820,"wires":[["eb5b1724.3f7f2"]]
},{"id":"d114bd2d.ece9f","type":"trigger","z":"efe4729c.04a598","op1":"","op2":"0","
op1type":"pay","op2type":"str","duration":"0","extend":false,"units":"ms","reset":""
,"bytopic":"all","name":"Trigger","x":940,"y":700,"wires":[["dbc207d5.8e184"]]},{"id
":"d16124fc.a22bb","type":"link in","z":"efe4729c.04a598","name":"","links":["92471b
b3.2cfbf"],"x":855,"y":660,"wires":[["d114bd2d.ece9f"]]},{"id":"a8a3d607.17492","typ
e":"function","z":"efe4729c.04a598","name":"Reset","func":"//RESET PARA OPERAÇÃO MOD
O QUANTIDADE DE AGUA\nmsg.reset = true;//define reset trigger\n\nreturn msg;","outpu
ts":1,"noerr":0,"x":370,"y":900,"wires":[["92471bb3.2cfbf"]]},{"id":"ec92b540.73f328
","type":"link in","z":"efe4729c.04a598","name":"","links":["92471bb3.2cfbf"],"x":85
5,"y":780,"wires":[["43b082c.c7486fc"]]},{"id":"31d961a5.a22a16","type":"link in","z
":"efe4729c.04a598","name":"","links":["92471bb3.2cfbf"],"x":855,"y":900,"wires":[["
2e38b734.80bdc"]]},{"id":"8cc7fbb9.2d36b8","type":"comment","z":"efe4729c.04a598","n
ame":"AQUECIMENTO AUXILIAR","info":"","x":130,"y":660,"wires":[]},{"id":"2d566f30.95
d1d8","type":"change","z":"efe4729c.04a598","name":"OFF","rules":[{"t":"set","p":"pa
yload","pt":"msg","to":"off","tot":"str"}],"action":"","property":"","from":"","to":
"","reg":false,"x":550,"y":1100,"wires":[["a8eae99f.7fb658","f3c29931.aa8c88","1e39c
c60.691ffc"]]},{"id":"78858e79.2917a8","type":"change","z":"efe4729c.04a598","name":
"ON","rules":[{"t":"set","p":"payload","pt":"msg","to":"on","tot":"str"}],"action":"
","property":"","from":"","to":"","reg":false,"x":810,"y":860,"wires":[["ebd76a88.a6
ced8"]]},{"id":"83e1d23e.490358","type":"change","z":"efe4729c.04a598","name":"ON","
rules":[{"t":"set","p":"payload","pt":"msg","to":"ON","tot":"str"}],"action":"","pro
perty":"","from":"","to":"","reg":false,"x":810,"y":740,"wires":[["8540c051.623b08"]
]},{"id":"c674b675.ad8948","type":"change","z":"efe4729c.04a598","name":"ON","rules"
:[{"t":"set","p":"payload","pt":"msg","to":"on","tot":"str"}],"action":"","property"
:"","from":"","to":"","reg":false,"x":810,"y":980,"wires":[["2ee2a4d9.8347e4"]]},{"i
d":"16b1d16d.b1f81f","type":"mqtt-
broker","z":"","name":"IHLE.COM.BR","broker":"localhost","port":"1883","clientid":""
,"usetls":false,"compatmode":false,"keepalive":"60","cleansession":true,"birthTopic"
:"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"
","willTopic":"","willQos":"0","willPayload":""},{"id":"6f62f03f.fecbd8","type":"inf
luxdb","z":"","hostname":"127.0.0.1","port":"8086","protocol":"http","database":"Equ
ip_Status","name":"","usetls":false,"tls":""}]
```