



**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA**

ALESSANDRO SOARES DA SILVA

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE E
MONITORAMENTO REMOTO DE UM FORNO INDUSTRIAL DE
UMA EMPRESA DO POLO DE DUAS RODAS DO DISTRITO
INDUSTRIAL**

Manaus
2021

ALESSANDRO SOARES DA SILVA

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE E
MONITORAMENTO REMOTO DE UM FORNO INDUSTRIAL DE
UMA EMPRESA DO POLO DE DUAS RODAS DO DISTRITO
INDUSTRIAL**

Pesquisa desenvolvida durante a disciplina de Trabalho de conclusão de curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para a obtenção do título de Engenheiro em Eletrônica.

Orientador: André Luiz Printes, Me.

Manaus
2021

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia – EST

Reitor:

Cleinaldo de Almeida Costa, Dr.

Vice-Reitor:

Cleto Cavalcante de Souza Leal, Me.

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo, Me.

Coordenador do Curso de Engenharia:

Daniel Gusman del Rio, Dr.

Banca Avaliadora composta por:

Data da defesa: 22/07/2021

Prof. André Luiz Printes, Me (Orientador)

Prof. Jozias Parente de Oliveira, Dr (Avaliador 1)

Prof. Fábio de Souza Cardoso, Dr (Avaliador 2)

CIP – Catalogação na Publicação

Silva, Alessandro Soares

Desenvolvimento de sistema para controle e monitoramento remoto industrial de uma empresa do polo de duas rodas do Distrito Industrial / Alessandro Soares da Silva; [orientado por] Prof. André Luiz Printes – Manaus: 2021.

70 f. p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica). Universidade do Estado do Amazonas, 2021.

1. Controle e Monitoramento, 2. Medição de Temperatura
3. *Raspberry PI*, 4. *NodeRed*, 5. *Arduino*.
I. Luiz Printes, André.

ALESSANDRO SOARES DA SILVA

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE E
MONITORAMENTO REMOTO DE UM FORNO INDUSTRIAL DE
UMA EMPRESA DO POLO DE DUAS RODAS DO DISTRITO
INDUSTRIAL**

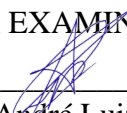
Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade Estadual do Amazonas, como pré-requisito para a obtenção do título de Engenheiro em Eletrônica.

Nota obtida: _____ (_____)

Aprovada em ____/____/____.

Área de concentração: Sistemas embarcados, Internet das Coisa/(IoT).

BANCA EXAMINADORA


Orientador: André Luiz Printes, Me.


Avaliador: Jozias Parente de Oliveira, Dr.


Avaliador: Fábio de Souza Cardoso, Dr.

Manaus
2021

DEDICATÓRIA

Dedico este trabalho ao meu pai, minha esposa e a todos os meus familiares e amigos que direta ou indiretamente me apoiaram nesta caminhada. Meu muito obrigado.

RESUMO

O presente documento tem a finalidade de apresentar o desenvolvimento de um sistema de controle e monitoramento remoto de um forno industrial, que foi aplicado em um equipamento de uma empresa do Polo de Duas Rodas do Distrito Industrial, a fim de auxiliar a evitar perdas de recursos, produtividade e eventuais acidentes de trabalho. Este sistema tem a capacidade de enviar mensagens por e-mail e sms, sempre que valores de temperatura excederem limites pré-determinados. O sistema é apresentado por meio de um modelo real e a transmissão de dados é executada por meio da rede industrial da referida empresa através do protocolo Ethernet. Primeiramente, mostra-se a fundamentação teórica das tecnologias utilizadas neste projeto: Medição de temperatura e seu processamento digital, tecnologia embarcada *Raspberry PI*, parâmetros de configuração do software *Node-RED*, Placa de desenvolvimento Arduino Mega, Arduino IDE, automação industrial e CLP. Em seguida, em Métodos e Materiais, são apresentadas as etapas de desenvolvimento do projeto e os materiais utilizados. Posteriormente, em Implementação do projeto, há a exposição do que foi executado durante as etapas do trabalho, onde serão apresentados os fluxogramas dos programas desenvolvidos no Software *Node-RED* para a validação do sistema. Foram realizados exaustivos testes assim como o envio de pacotes de mensagens por meio da rede industrial. É mostrado também a implementação e integração do sistema ao painel elétrico do forno. Posteriormente é apresentado o desenvolvimento da interface gráfica para visualização dos gráficos dos valores de temperatura do forno em tempo real. Na sequência, em Resultados Obtidos, são mostrados e analisados os dados do projeto em seu pleno funcionamento. Finalmente, conclui-se por meio dos dados obtidos que a tecnologia embarcada *Raspberry Pi* é uma alternativa viável para projetos de baixo custo de monitoramento remoto e sua utilização em IoT (internet das coisas) no auxílio da aplicação dos conceitos da Indústria 4.0.

Palavras-chaves: Controle e Monitoramento, Medição de Temperatura, *Raspberry PI*, *Node-RED*, Arduino.

ABSTRACT

The purpose of this document is to present the development of a remote control and monitoring system for an industrial furnace, which was applied to a company's equipment in the two-wheeled hub of the Industrial District, in order to help prevent loss of resources, productivity and possible accidents at work. This system has the ability to send messages by e-mails and sms whenever temperature values exceed predetermined limits. The system is presented through a real model and the data transmission is performed through the industrial network of the referred company through the Ethernet protocol. First, it shows the theoretical foundation of the technologies used in this project: Temperature measurement and its digital processing, Raspberry PI embedded technology, Node-RED software configuration parameters, Arduino Mega development board, Arduino IDE, industrial automation and PLC. Then, in Methods and Materials, the stages of project development and the materials used are presented. Later, in Project Implementation, there is an exposition of what was executed during the work stages, where the flowcharts of the programs developed in the Node-RED Software for the system validation will be presented. Exhaustive tests were carried out, as well as the sending of message packages through the industrial network. The implementation and integration of the system with the oven's electrical panel is shown. Subsequently, the development of the graphical interface for visualization of the graphs of the furnace temperature values in real time is presented. Then, in Results Obtained, the data of the project in its full functioning are shown and analyzed. Finally, it is concluded from the data obtained that the Raspberry PI embedded technology is a viable alternative for low-cost remote monitoring projects and its use in IoT (internet of things) to help apply the concepts of Industry 4.0.

Keywords: Control and Monitoring, Temperatura measurement, Raspberry PI, Node-RED, Arduino.

LISTA DE FIGURAS

Figura 1 - Esquema simplificado de um termopar	16
Figura 2 - Modelo típico de um termopar.....	17
Figura 3 - Amostragem e Quantização de um sinal.....	18
Figura 4 - Conversor A/D com rede R/2R.....	20
Figura 5 - Conversor A/D de rampa digital.....	21
Figura 6 - Aspectos gerais do CLP.....	23
Figura 7- Representação das funções básicas no diagrama <i>ladder</i>	23
Figura 8 - <i>Raspberry Pi</i> 3 Modelo B+	24
Figura 9 - Placa Arduino Mega.	26
Figura 10 - IDE Arduino	28
Figura 11 - Interface gráfica do <i>Raspbian</i>	30
Figura 12 - Ambiente de desenvolvimento <i>Node-RED</i>	31
Figura 13 - <i>Raspberry Pi Imager</i>	36
Figura 14 - Posição de encaixe do cartão micro SD.....	37
Figura 15 - Plataforma <i>Node-RED</i> em execução.....	38
Figura 16 - <i>Firmata Standard</i> no IDE Arduino.....	40
Figura 17 - Módulo Borne com Arduino Mega.....	41
Figura 18 - Rede resistiva inserida no Arduino.....	42
Figura 19 - Configuração das entradas e saídas no <i>Node-RED</i>	43
Figura 20 - Linhas de código de entrada do sistema	44
Figura 21 - Linhas de código do bloco de alarme	45
Figura 22 - Criação da dashboard no <i>Node-RED</i>	45
Figura 23 - Configuração da tela interativa	46
Figura 24 - Relação de conversão do cartão analógico do CLP	48
Figura 25 - Programação ladder no CLP <i>Mitsubishi</i>	49
Figura 26 - Disposição interna dos módulos no painel	50
Figura 27 - Painel do sistema integrado ao forno industrial.....	51
Figura 28 - Configuração de acesso à rede.....	52
Figura 29 - Gráfico de objetivo de meta da empresa.....	55
Figura 30 - Fluxograma operacional	55
Figura 31 - Visualização da primeira tela do sistema.....	56

Figura 32 – Visualização da segunda tela do sistema	57
Figura 33 – Modelo do corpo do e-mail da mensagem de alerta	58

LISTA DE TABELAS

Tabela 1 – Especificações do <i>Raspberry Pi</i>	25
Tabela 2 – Especificação do módulo Arduino Mega	27
Tabela 3 – Relação temperatura x tensão do forno	47
Tabela 4 - Características técnicas do cartão analógico do PLC.....	48

SUMÁRIO

INTRODUÇÃO	13
1 REFERENCIAL TEÓRICO	16
1.1 MEDIÇÃO DE TEMPERATURA.....	16
1.2 CONVERSÃO A/D E D/A	18
1.2.1 Quantização e Amostragem do Sinal.....	18
1.2.2 Conversão Digital-Analógica (D/A).....	19
1.2.3 Conversão Analógica-Digital (A/D).....	20
1.3 AUTOMAÇÃO INDUSTRIAL E CLP	22
1.4 RASPBERRY PI	24
1.5 ARDUINO.....	26
1.6 SOFTWARES	28
1.6.1 IDE Arduino	28
1.6.2 Sistema Operacional Raspbian	29
1.6.3 Node-RED	30
2 MÉTODOS E MATERIAIS	32
2.1 MATERIAIS UTILIZADOS NO PROJETO.....	34
3 IMPLEMENTAÇÃO DO PROJETO.....	35
3.1 INSTALAÇÃO DA IMAGEM DO RASPBIAN	35
3.2 CONFIGURAÇÃO DO NODE-RED NO RASPBERRY PI	37
3.3 INSTALAÇÃO DA BIBLIOTECA FIRMATA NA PLACA ARDUINO.....	39
3.3.1 Placa borne Arduino GPIO.....	40
3.3.2 Arduino com <i>Pulldown</i> Externo.....	41
3.4 DESENVOLVIMENTO DO PROGRAMA DE SUPERVISÃO NO <i>NODE-RED</i>	42
3.4.1 Entradas e Saídas do Sistema	43
3.4.2 Bloco de função da temperatura e padrão de alarme.....	44
3.4.3 Desenvolvimento da dashboard.....	45
3.4.4 Trabalhando com o sinal analógico de temperatura	47
3.5 MONTAGEM DOS MÓDULOS NO PAINEL ELÉTRICO	50
3.6 PARAMETRIZAÇÃO DO SISTEMA PARA ACESSO À REDE INDUSTRIAL	52
4 RESULTADOS OBTIDOS	54
4.1 RESULTADO DO SISTEMA DE MONITORAMENTO REMOTO DO FORNO	54
CONCLUSÃO.....	59
REFERÊNCIAS BIBLIOGRAFICAS	60
APÊNDICE A – CÓDIGO COMPLETO DA PRIMEIRA TELA	62
APÊNDICE B - CÓDIGO COMPLETO DA SEGUNDA TELA	63
APÊNDICE C – ESQUEMA ELÉTRICO DO FORNO COM A INTEGRAÇÃO DO SISTEMA DE MONITORAMENTO.....	64

APÊNDICE D – ESQUEMA ELÉTRICO DO PROJETO ARDUINO MEGA 2560.....	68
APÊNDICE E – DIAGRAMA DE PINAGEM DO ARDUINO MEGA 2560	69
APÊNDICE F – DIAGRAMA EM BLOCOS DO SISTEMA	70

INTRODUÇÃO

Este trabalho tem como tema o desenvolvimento de sistema para controle e monitoramento remoto de um forno industrial de uma empresa do polo de duas rodas do distrito industrial. Sabe-se que os processos industriais são uma sequência linear de operações que são realizadas em tempos variáveis com o intuito de fabricar e fornecer produtos para o mercado consumidor. As corporações empresariais almejam a competitividade como objetivo de sustentação e sobrevivência a longo prazo, logo a competitividade significa atender simultaneamente os requisitos de mercado/cliente e objetivos estratégicos. Para que esses processos sejam eficientes é importante aproveitar ao máximo os recursos nele presente (RITZMAN; KRAJEWSKI, 2004).

Um recurso muito importante nos processos industriais é o equipamento, que precisa estar sempre disponível para atender ao seu propósito. E nas últimas décadas, com o aumento da competitividade industrial, tornou-se imprescindível a incorporação de tecnologias de coleta de dados em tempo real aos processos produtivos, evidenciando o conceito de Indústria 4.0, que introduz a integração dos sistemas computacionais e a internet aos processos de automação convencionais (MAZZAFERRO, 2018). O problema que a pesquisa se propôs a resolver está diretamente ligado a implementação das ideias da indústria 4.0, para assim evitar problemas relacionados com a baixa temperatura do alumínio líquido contido no interior do forno em períodos de ausência de suporte técnico na fábrica, evitando a perda precoce do equipamento.

As Indústrias Inteligentes ou *Smart Factories*, como são conhecidas as empresas que aplicam a alta tecnologia ao seu processo produtivo, indicam vários benefícios e pontos positivos, porém, sua implementação vem agregada a vários desafios, que vão desde a qualificação de mão de obra especializada, a altos investimentos financeiros na área de tecnologia da informação. Os recentes desenvolvimentos em áreas como microeletrônica, computação em nuvem, novos processos de fabricação de circuitos integrados, permitiu a evolução e aplicabilidade da *internet-of-things* (IoT) ou internet das coisas aos processos produtivos de forma mais eficiente e economicamente viável (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2016).

Seguindo essa premissa e aplicando os conceitos mencionados, o presente trabalho testa a hipótese de que seja possível o desenvolvimento de um sistema de controle e

monitoramento remoto de um forno industrial em uma empresa do Distrito Industrial de Manaus. Este sistema é construído com base no computador embarcado conhecido como *Raspberry PI* que é desenvolvido e distribuído pela organização sem fins lucrativos Britânica *Raspberry Pi Foundation*. Esse dispositivo estará conectado à rede industrial dessa Companhia Multinacional e terá a função de monitorar as variações de temperatura no interior do referido forno. Uma vez que, a temperatura interna oscile fora do limite de valores pré-estabelecidos, o sistema será capaz de enviar e-mail e sms de alerta a pessoas previamente determinadas, que terão a possibilidade de visualizar o processo em tempo real e atuar remotamente ao comando elétrico do forno, via aplicativo desenvolvido no sistema, de modo a reestabelecer as condições normais de trabalho, evitando o deterioramento da estrutura do forno e prejuízos econômicos à empresa.

O sistema de controle e monitoramento remoto, agrega o conhecimento multidisciplinar vistos nas disciplinas de linguagem de programação, projetos de sistemas microprocessados, eletrônica analógica, eletrônica de potência, sensores e instrumentação, comandos elétricos, processamento digital de sinais, redes de comunicação e automação industrial.

A implementação do referido projeto de pesquisa se justifica primeiramente por possibilitar o aumento da vida útil do equipamento, uma vez que, pode evitar o decaimento da temperatura de forma prematura evitando assim a perda parcial ou total da estrutura em decorrência da solidificação do alumínio. Em segundo lugar temos a redução no custo orçamentário vinculada as atividades de hora extra em função do destacamento de colaboradores técnicos para o atendimento do equipamento.

Realizou-se uma revisão teórica que contempla os assuntos de medição de temperatura através de sensores específicos (termopar), a digitalização desses sinais medidos, o sistema operacional utilizado no *Raspberry* e tópicos sobre alguns parâmetros importantes de comunicação. Também foram abordados os softwares que foram empregados para programação do sistema microcontrolado e microprocessado utilizado nesta pesquisa.

Este trabalho está dividido nos seguintes capítulos: Referencial teórico, Métodos e Materiais, Implementação do projeto e Resultados obtidos.

No primeiro capítulo é abordada uma revisão dos assuntos referentes aos dispositivos e softwares utilizados e conceitos do protocolo de comunicação. No segundo capítulo são mostrados os métodos utilizados para se obter o objetivo final. No terceiro capítulo,

implementação do projeto, que aborda a execução dos passos citados no capítulo de métodos. Mostra-se todas as etapas de construção do sistema, assim como a criação da interface de acesso ao usuário. Os programas feitos em *JavaScript* durante a implementação do projeto, são mostrados nos Apêndices.

No quarto capítulo, Resultados obtidos, mostram-se os resultados dos experimentos realizados na implementação, e faz-se análises baseadas no referencial teórico e no conhecimento adquirido durante todo o processo de pesquisa e implementação do projeto.

E por fim é apresentada a conclusão que aborda a análise do comportamento dos módulos utilizados e a discussão acerca da utilização destes dispositivos para a aplicação em monitoramento e controle de processos.

1 REFERENCIAL TEÓRICO

1.1 MEDIÇÃO DE TEMPERATURA

A temperatura é um dos parâmetros físicos mais comuns e talvez um dos mais medidos na indústria, pois afetam diretamente boa parte dos processos manufaturados e neste contexto é classificado como processo térmico. O conceito de temperatura diz respeito ao nível de agitação molecular de um corpo ou energia cinética média das moléculas que a compõe (BENTLEY, 1993).

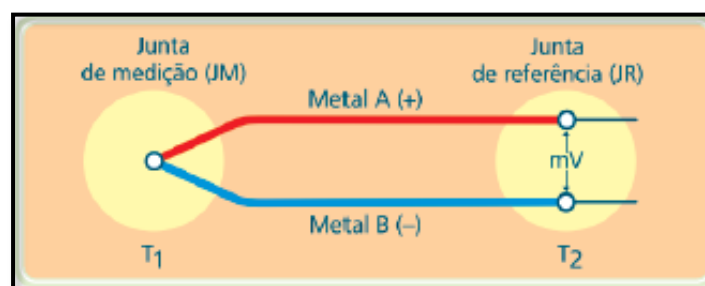
Todo sistema possui variáveis a serem medidas em um processo produtivo, e não é diferente para os processos térmicos, onde, existem dispositivos que auxiliam a mensurar tal grandeza, e podemos dividi-las em categorias tais como:

- Indicadores
- Sensor/transdutor

O tipo de cada dispositivo de medição se baseia segundo a forma como os diferentes materiais se comportam quando expostos ao calor, caracterizando o modelo de medição desejado como indicação, pontos de medida, saída analógica.

Desta forma, quando um corpo tende a se expandir, como resposta a exposição ao calor, podemos ter uma medição de indicação visual, pressão ou mudança dimensional. No entanto, existem processos ao qual a resposta está na alteração da resistência elétrica do componente de medição (termistor), gerando assim tensões que variam de acordo com o material utilizado para medir (termopar), ou ainda diferentes níveis de irradiação de energia (BENTLEY, 1993).

Figura 1 – Esquema simplificado de um termopar



Fonte: (ROGGIA; FUENTES, 2016, p.31).

O termopar faz uso de um fenômeno físico chamado de efeito Seebeck, onde dois metais distintos são postos em contato através de uma junção, e à medida que a diferença de temperatura entre o ponto de junção e a outra extremidade se altera é produzido um potencial elétrico que é proporcional a temperatura da junção. Para um determinado processo, o modelo de um termopar deve levar em consideração as temperaturas que se deseja mensurar de forma que, é importante se ter orientação do fabricante quanto ao melhor modelo para determinada faixa de temperatura (ARENLY, 2001).

Sem dúvida os termopares são os sensores mais utilizados em processos térmicos de chão de fábrica, muito em função de sua robustez e simplicidade de aplicação. No entanto, é importante salientar que o sinal de saída de um termopar é analógico, e dessa forma, para aplicações em sistemas digitais, se faz necessário realizar a conversão analógico-digital (ARENLY, 2001).

A figura 1, mostra o esquema simplificado de um termopar, onde a tensão de saída que representa a diferença de temperatura entre as junções é dada pela equação 1. Enquanto a figura 2 ilustra um modelo típico de termopar industrial.

$$V_{out} = V_1 - V_2 \quad (\text{Equação 1})$$

Figura 2 – Modelo típico de um termopar



Fonte: ALUTAL (sd, p.1)

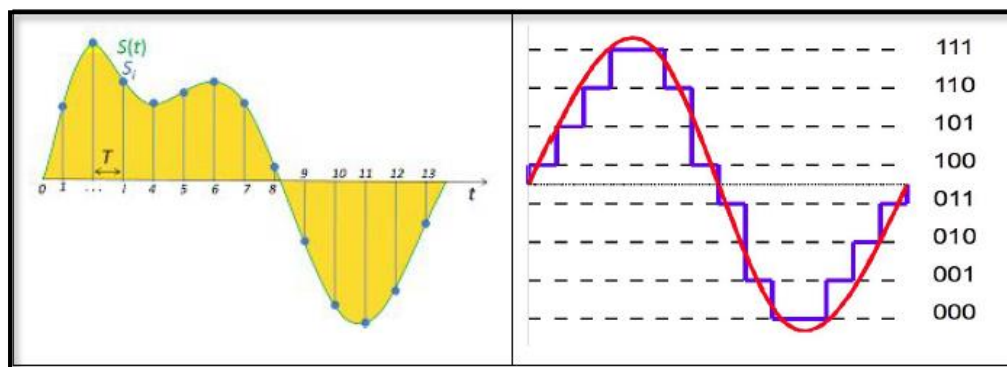
1.2 CONVERSÃO A/D E D/A

Como visto na seção 1.1, os termopares convertem os sinais físicos, nesse caso a temperatura, em sinais elétricos. Vale ressaltar que a maior parte das grandezas físicas são analógicas em sua essência, no entanto, os sistemas microprocessados trabalham com sinais digitais. Desta forma, para realizar o controle de sistemas através do processamento digital dos sinais, é necessário a utilização de circuitos que realizem a conversão entre sinais analógicos (mundo físico) e os sinais digitais (mundo computacional). O circuito de conversão do sinal digital para o sinal analógico é conhecido como conversor digital-analógico (D/A), enquanto a conversão inversa é realizada pelo conversor analógico-digital (A/D) (TOCCI; WIDMER; MOSS, 2011).

1.2.1 Quantização e Amostragem do Sinal

Um sinal classificado como analógico tem a característica de ser contínuo no tempo, ou seja, pode possuir infinitos valores ao longo de uma escala de tempo. Essa característica impede sua utilização em processos digitais, pois seria impossível registrar infinitos valores, desta forma, se faz necessário amostrar e quantizar o sinal em uma sequência finita de números que representam uma aproximação do mesmo, sendo assim possível representar digitalmente esse valor. Portanto, um sinal digital nada mais é do que um sinal analógico amostrado e representado por um valor discreto (quantizado), é importante salientar que nesse processo ocorrem perdas de informação e um circuito bem aplicado e elaborado é fundamental para um bom projeto (TOCCI; WIDMER; MOSS, 2011).

Figura 3 – Amostragem e Quantização de um sinal



Fonte: (OLIVEIRA; SHIN-TING, 2019, p.3)

A figura 3, ilustra os conceitos mencionados. É necessário deixar claro que, existem técnicas baseadas em conceitos científicos que validam a construção de um esquema de amostragem e quantização, conhecido como circuitos *sample and hold*, um exemplo é o teorema de Nyquist-Shannon, que afirma ser possível reconstruir um sinal analógico, limitado a banda de frequências, a partir de uma sequência de amostragem de valores originais e obtidas em intervalos menores que $1/2f_m$, onde f_m é a maior frequência em Hertz (Hz) do sinal original (SEDRA; SMITH, 2007).

Quando a frequência de amostragem do sinal é menor que $2f_m$, o sistema passa a apresentar um problema conhecido como falseamento ou *aliasing*, e uma forma de contorná-lo é aplicando uma filtragem no sinal de modo a “suavizá-lo”, removendo assim as componentes de altas frequências antes de passar pelo circuito de amostragem. É possível ainda melhorar a resolução do sinal quantizado, ou seja, melhorar a capacidade de diferenciação entre o sinal analógico e o sinal digitalizado, para isso é necessário aumentar a quantidade de *bits* para representar um valor, alterando assim seu fundo de escala. Em outras palavras, quanto maior for o número de *bits* de representação, maior a resolução do conversor (TOCCI; WIDMER; MOSS, 2011).

A resolução pode ser entendida como a menor variação na saída de um conversor em função de uma mudança em sua entrada, ou seja, a resolução é o tamanho do degrau e tem sua expressão dada pela equação 2, onde A_{fs} é o valor de saída do fundo de escala e N é a quantidade de *bits*.

$$R = \frac{A_{fs}}{2^N - 1} \quad (\text{Equação 2})$$

1.2.2 Conversão Digital-Analógica (D/A)

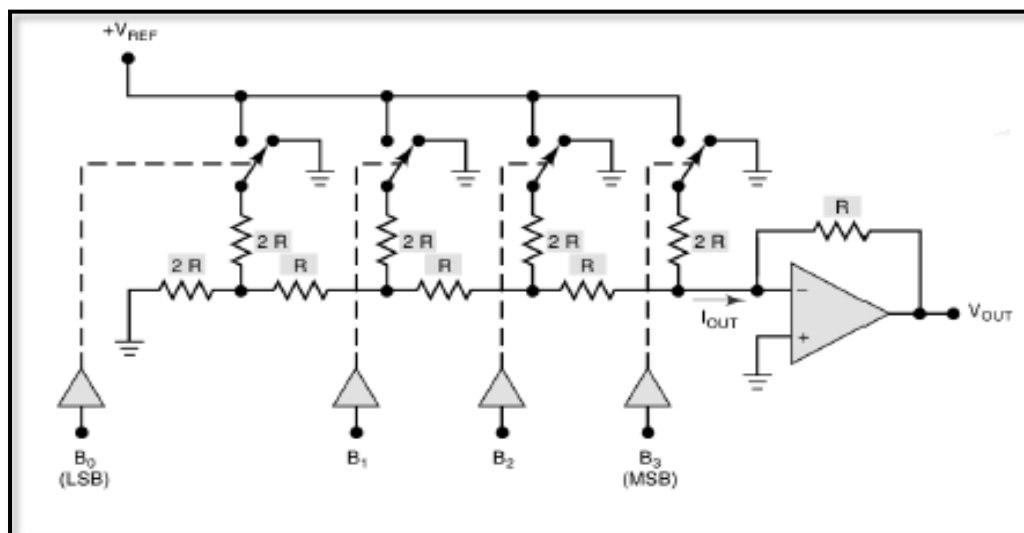
Um conversor digital-analógico é um circuito que realiza a conversão de códigos binários (digital), em sinais analógicos, normalmente na forma de corrente ou tensão. Obviamente não se trata de um valor analógico real, pois como mencionado na seção anterior, por mais que se tenha um conversor com resolução altíssima, ainda assim seria impossível se trabalhar com infinitos números (SEDRA; SMITH, 2007).

A figura 4 ilustra o circuito de um conversor D/A, existem várias topologias de circuitos que realizam essa função, no entanto algumas são mais eficientes que outras. Neste exemplo temos um circuito com rede resistiva, conhecido como rede R/2R.

Basicamente, o circuito tem o intuito de gerar para cada bit do código binário original, uma quantidade de sinal analógico e posteriormente proceder com a soma deles, respeitando uma ponderação de acordo com sua posição no código. Mas especificamente nesse circuito de rede resistiva, o mesmo possui a vantagem de se empregar apenas dois valores múltiplos de resistência, logicamente que a precisão nesse tipo de circuito é diretamente proporcional a precisão dos resistores empregados, o valor de V_{out} é dada pela equação 3 (FLOYD, 2007).

$$V_{OUT} = -\frac{V_{REF}}{16} * B \quad (\text{Equação 3})$$

Figura 4 – Conversor A/D com rede R/2R



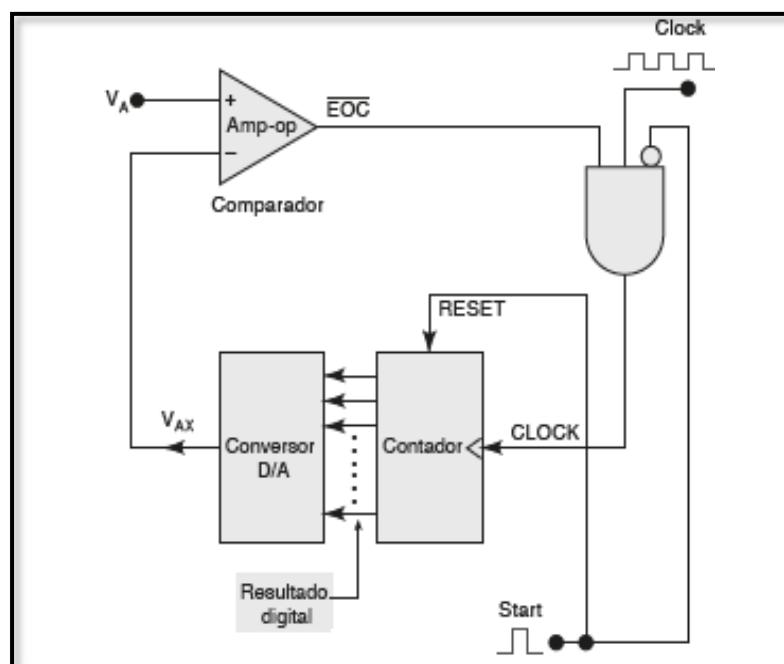
Fonte: (TOCCI; WIDMER; MOSS, 2011, p.639)

1.2.3 Conversão Analógica-Digital (A/D)

O conversor analógico-digital tem a função de ler uma entrada analógico, normalmente valores expressos em tensão ou corrente, e transformá-los em sinais digitais, ou seja, em valores binários. Este circuito é muito mais complexo e inclusive utiliza um conversor D/A em sua arquitetura construtiva. O processo envolve os conceitos de amostragem e quantização do sinal vistos na seção 1.2.1, uma versão mais simples do conversor A/D faz alusão ao circuito de rampa digital, onde existe um circuito contador que sofre incremento a cada ciclo e esse valor é amostrado em um circuito comparador através de realimentação (TOCCI; WIDMER; MOSS, 2011).

O circuito é ilustrado na figura 5, este modelo possui um problema associado ao seu tempo de conversão, que dobra para cada bit (resolução) adicionado à saída do circuito. É notório que, quando a informação de entrada está próximo do valor de fundo de escala, o tempo tende ao máximo do circuito que depende é claro da frequência de clock do contador. Uma solução eficaz é a conversão por aproximações sucessivas, que permite fixar o tempo de conversão independentemente do valor a ser convertido, esse método pode ser aplicado com uma simples substituição do circuito contador por um registrador conhecido como *successive approximation register – SAR*. (TOCCI; WIDMER; MOSS, 2011).

Figura 5 – Conversor A/D de rampa digital



Fonte: (TOCCI; WEDMER; MOSS, 2011, p.646)

Em sistemas embarcados ou em qualquer sistema digital, não é primordial se ter o conhecimento aprofundado de todos os detalhes do circuito, pois com a evolução da microeletrônica, este tipo de circuito é oferecido em grande quantidade no mercado, encapsulados em circuitos integrados. O mais importante para o projetista é conhecer os parâmetros associados aos circuitos e que são informados pelos fabricantes dos componentes, desta forma pode se fazer a melhor escolha para a aplicação em questão. Logo parâmetros como resolução, precisão, erro de offset, tempo de estabilização, monotonicidade são informações muito importantes e que devem ser levados em consideração no planejamento de um projeto de conversão (FLOYD, 2007).

1.3 AUTOMAÇÃO INDUSTRIAL E CLP

A indústria moderna como a conhecemos hoje em dia, dificilmente conseguiria se manter competitiva no mercado sem a aplicação da automação industrial em seus processos produtivos. Mas em última análise, considerando que um bom conceito de automação industrial é a substituição do trabalho humano por máquinas, então é correto afirmar que o processo de automação se deu no início do século XVIII, com a criação da máquina a vapor na primeira Revolução Industrial (RIBEIRO, 1999).

A ideia de produção em larga escala visando a alta produtividade e maximização do lucro, alavancaram a modernização dos processos industriais. Com a rápida evolução tecnológica, sobretudo nas áreas de eletrônica e da computação nos anos 50 e 60 do século XX, rapidamente sistemas elétricos ultrapassados controlados por relés eletromecânicos foram sendo substituídos por controles modulares digitais conhecidos como MODICON, um dos primeiros modelos de controlador lógico programável ou CLP (SILVEIRA; LIMA, 2003).

O fato de a indústria de circuitos integrados ter evoluído ao ponto de permitir a criação dos primeiros microprocessadores nos anos de 1970, permitiu que um número cada vez maior de cálculos complexos fosse realizado em períodos cada vez menores, o que culminou no desenvolvimento do computador pessoal (PC). Agregado ao processamento digital de sinal e ao moderno processo de controle o CLP surge como instrumento fundamental na indústria moderna (RIBEIRO, 1999).

O Controlador Lógico Programável (CLP) é classificado hoje como um dispositivo eletrônico digital, dotado de elementos de *hardware* e *software* com extensa aplicação na indústria. Este dispositivo pode ser programado através de linguagem de programação específica e emula diagramas elétricos chamados de *ladder* de relés para operar em um ambiente de controle em tempo real. Possui uma CPU que tem a função de “varrer” os módulos de entrada e saída (I/O), tomar decisões de controle sequencial previamente definida em sua programação, e repetir esse processo de modo ininterrupto (ROGGIA; FUENTES, 2016).

A figura 6 mostra os aspectos gerais que o CLP possui, vale ressaltar que existe uma gama infundável de marcas, modelos e tamanhos desses dispositivos. Obviamente seus valores variam de acordo com sua potência computacional e aplicações.

Figura 6 – Aspectos gerais do CLP



Fonte: (ROGGIA; FUENTES, 2016, p.82)

Por se tratar de um computador dedicado a ambientes industriais, sua arquitetura construtiva é robusta uma vez que, o dispositivo trabalha em ambientes agressivos com muita humidade, sujeira, altas temperaturas, poeira e em alguns casos com muita vibração. O sistema deve possuir imunidade a ruído elétrico, com especificações em muitos casos comparáveis aos padrões militares. Uma diferença do CLP para computadores convencionais é que as operações em um CLP são executadas de forma sequencial permitindo uma fácil programação dos registradores, contadores, temporizadores em aplicações de controle de tempo real, possuem relógio interno (*clocks*), bateria interna para manter a alimentação do circuito do relógio e dos parâmetros e programas em uma eventual falta de energia. (ROGGIA; FUENTES, 2016).

Figura 7 – Representação das funções básicas no diagrama *ladder*.

Função	Equação	Símbolo	Diagrama de contatos
NOT NÃO	$Q1 = \bar{I1}$		
AND E	$Q1 = I1 \cdot I2$		
OR OU	$Q1 = I1 + I2$		

Fonte: (ROGGIA; FUENTES, 2016, p.89)

Na figura 7 temos a representação das funções lógicas básicas no diagrama ladder. Outra característica intrínseca ao CLP é a incorporação por parte dos fabricantes de autodiagnóstico para uma fácil pesquisa de defeitos e reparos, além de possuírem componentes modulares de

simples troca (*plug in*) em caso de defeito, em alguns modelos é possível realizar o reparo com o equipamento em pleno funcionamento sem que haja interrupção de controle ao restante do sistema. Um CLP realiza a função de integrar equipamentos externos acionados por alta tensão, na ordem de 100 V ou mais, a dispositivos discretos de entrada com lógica TTL (lógica transistor-transistor) ou seja, com tensões de trabalho da ordem de 5 V. (RIBEIRO, 1999).

1.4 RASPBERRY PI

O sistema de monitoramento industrial é aplicado com o intuito de monitorar e controlar processos de plantas industriais e eventualmente analisar o seu desempenho através da coleta de informações que são armazenados em banco de dados. Essa aplicação faz parte de um grupo de 10 tecnologias essenciais para implementação da indústria 4.0, a utilização da internet das coisas, da realidade aumentada e simulação virtual podem consumir tempo e principalmente recursos financeiros. No entanto, é possível utilizar *hardware* de suporte na forma de um *Raspberry Pi* como dispositivo para processar os dados, enquanto sua visualização, ou forma gráfica, com a utilização de *softwares* como o *Node-RED*, reduzindo de forma substancial os custos do projeto (WIGUNA; SUBARI, 2017).

O *Raspberry Pi* é um computador de bolso com tamanho aproximado de um cartão de crédito, que foi desenvolvido no Reino Unido pela fundação *RASPBERRY PI* com o objetivo de promover o ensino de ciência básica e programação nas escolas secundárias de baixa renda (MAN, 2016).

Figura 8 - Raspberry Pi 3 Modelo B+



Fonte: O próprio Autor

Apesar de pequeno o *Raspberry Pi* tem uma capacidade de processamento e como todo computador, necessita de um sistema operacional. É compatível com sistemas baseados em *GNU/Linux*, mas qualquer linguagem que possa ser compilada na arquitetura ARM a partir da versão 6 pode ser usada para o desenvolvimento de software, inclusive a Microsoft possui uma versão para o microcomputador chamado de *Windows IoT* (KAWAMURA, 2017).

Tabela 1 - Especificações do *Raspberry Pi* modelo 3B+

GPU	Videocore IV @400 MHz
Processador	QUAD Core @1.2 GHz
Armazenamento	Micro SD
Porta Ethernet	Gigabit Ethernet over USB 2.0 (300 Mbps)
WiFi	Dual-band 2.4 GHz e 5 GHz IEEE 802.11.b/g/n/ac wireless LAN
Processador Chipset	Broadcom BCM2837B0, Cortex-A53 64 Bits SoC 1.4 GHz
Bluetooth Le	Bluetooth 4.2/BLE
GPIO	40 pinos
USB	4x 2.0 portas USB
RAM	1 GB LPDDR2 SD RAM
Max. Corrente/Tensão	2.5 A / 5V
Vídeo	HDMI

Fonte: (WIGUNA; SUBARI, 2017, p.2)

A figura 8 mostra o aspecto da placa do *Raspberry Pi* que possui um processador, um processador de vídeo, slot para cartão de memória, interface USB, saída de vídeo padrão HDMI, porta ethernet e 40 pinos GPIO que funcionam como interface com vários dispositivos eletrônicos, a tabela 1 apresenta as especificações do modelo 3B+. Essa conexão com o ambiente externo é um dos motivos que popularizaram o *Raspberry Pi* no meio acadêmico e

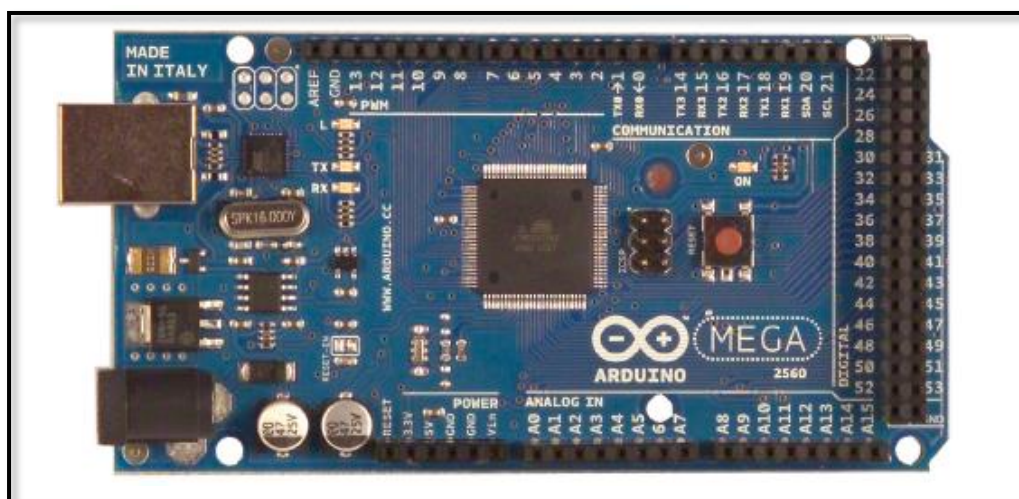
científico, pois abriu possibilidades de aplicações muito extensas aliadas como o baixo custo do dispositivo (WIGUNA; SUBARI, 2017).

1.5 ARDUINO

O Arduino é um *kit* de desenvolvimento projetado em uma placa eletrônica equipada com software de código aberto, que usa a família de microcontroladores *ATMega* da *Atmel* e funciona como um microcontrolador de placa única projetada para facilitar o uso eletrônico em vários campos. A placa é composta por um processador *Atmel AVR* e o *software* possui sua linguagem de programação própria (ISKANDAR; MUHAJIRIN; LISAH, 2017).

Criado e desenvolvido no *Ivrea Interaction Design Institute* na Itália, o *kit* de desenvolvimento Arduino é voltado para um público com pouca ou nenhuma experiência em eletrônica e programação, e que se popularizou rapidamente devido ao seu baixo custo em comparação com outras plataformas de microcontroladores. O modelo mais comum é o Arduino Uno, basicamente um *kit* para iniciantes, entretanto há outros modelos oferecidos pela empresa italiana. É possível encontrar no mercado muitos modelos derivados da placa, isso por que a arquitetura do projeto Arduino é aberto (*open-source*), ou seja, qualquer entusiasta pode acessar o site do fabricante e realizar o *download* do projeto de *hardware* com todas as suas especificações sem qualquer custo, obedecendo somente o critério de não usar o nome da marca Arduino, que é patentada (ARDUINO, 2021).

Figura 9 – Placa Arduino Mega



Fonte: ARDUINO (sd, p.2)

O fato de a plataforma Arduino tornar-se popular no mundo inteiro, fez com que muitos módulos conhecidos como *Shields*, fossem produzidos em larga escala por diversos fabricantes, facilitando a integração e o desenvolvimento de sistemas embarcados. Entre os principais Shields podemos destacar o módulo relé, que existem no mercado com diversas configurações e canais. Eles são perfeitos para controlar diversos dispositivos, inclusive motores e aparelhos eletrônicos (OLIVEIRA, 2017).

Na figura 9 é ilustrado a placa Arduino do modelo Mega, usada nesse projeto, e na tabela 2 pode-se verificar as principais características elétricas dessa placa que possui bastante recurso e normalmente é aplicada em projetos mais elaborados.

Tabela 2 - Especificações do módulo Arduino Mega

Microcontrolador	ATMega 2560
Tensão de Operação	5V
Tensão de Entrada (recomendada)	7-12V
Tensão de Entrada (limites)	6-20V
Pino I/O digitais	54 (sendo 14 usados como saída PWM)
Pino Analógico	16
Memória Flash	256 KB (8KB reservada para bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidade de Clock	16 MHz (Cristal externo)
Corrente máxima por pino I/O	20mA

1.6 SOFTWARES

1.6.1 IDE Arduino

O ambiente de programação do Arduino utiliza a linguagem C++ que é um compilador GCC, e as ferramentas, bem como seus parâmetros são totalmente configuráveis por meio de arquivo de configuração de *hardware* do Arduino. Essa abordagem garante a modularidade do IDE Arduino no suporte à sua ampla gama de placas oficiais, além de fornecer meios para adicionar fornecedores de *hardware* de outros fabricantes ao ambiente de programação. O IDE Arduino não requer *firmware* e é compilado em bloco monolítico e gravado diretamente na memória de programa, o que garante maior poder de processamento, visto que o processador fica exclusivamente dedicado ao programa, no entanto não há nenhum tipo de gerenciamento de *hardware* e *software* (OLIVEIRA, 2017).

A plataforma de programação é compatível com quase todos os sistemas operacionais existentes no mercado, sendo necessário realizar o seu *download* no *site* oficial da Arduino (<http://www.arduino.cc>). O ambiente de programação conta sempre com duas funções importantes: *setup()*, que é executada sempre ao iniciar o microcontrolador; e a função *loop()*, onde o programa roda continuamente em um laço sem fim. No menu *Sketch* em incluir bibliotecas, pode-se baixar uma biblioteca nova ou apontar para uma pasta do computador onde já encontra os códigos que serão inseridos (BLUM, 2016).

Figura 10 – IDE Arduino



```

StandardFirmata | Arduino 1.8.3
Arquivo Editar Sketch Ferramentas Ajuda

StandardFirmata
49
50 #ifdef FIRMATA_SERIAL_FEATURE
51 SerialFirmata serialFeature;
52 #endif
53
54 /* analog inputs */
55 int analogInputsToReport = 0; // bitwise array to store pin reporting
56
57 /* digital input ports */
58 byte reportPINS[TOTAL_PORTS]; // 1 = report this port, 0 = silence
59 byte previousPINS[TOTAL_PORTS]; // previous 8 bits sent
60
61 /* pins configuration */
62 byte portConfigInputs[TOTAL_PORTS]; // each bit: 1 = pin in INPUT, 0 = anything else
63
64 /* timer variables */
65 unsigned long currentMillis; // store the current value from millis()
66 unsigned long previousMillis; // for comparison with currentMillis
67 unsigned int samplingInterval = 19; // how often to run the main loop (in ms)
68
69 /* i2c data */
70 struct i2c_device_info {
71   byte addr;
72   int reg;
73   byte bytes;
74   byte stopIX;
75 };
76
77

```

Fonte: O próprio autor

Para este projeto a placa Arduino irá funcionar como uma interface de dados entre o ambiente externo e o software *Node-RED* instalado no módulo *Raspberry Pi*. Para que haja uma comunicação entre o *Node-RED* e o Arduino será necessário instalar uma biblioteca específica chamada *StandardFirmata*. Basicamente esse código implementa o protocolo *Firmata*, que é um protocolo de comunicação com microcontroladores a partir de um *software* em um computador, esse protocolo pode ser implementado em firmware em qualquer arquitetura de microcontrolador. Na figura 10 pode-se visualizar o aspecto do ambiente gráfico da IDE do Arduino.

1.6.2 Sistema Operacional Raspbian

Sistemas Operacionais podem ser definidos como uma coletânea de programas ou códigos que tem a função de realizar a *interface* entre o *software* e o *hardware*, onde seu principal propósito é proporcionar um ambiente onde o usuário de um subsistema possa executar programas no *hardware* de forma eficiente. Ao sistema operacional é atribuído a função de gerenciar recursos e dispositivos de entrada e saída, e oferecer uma simples *interface* para aplicativos e usuários (TANENBAUM; BOS, 2016).

Como visto no item 1.4, o *Raspberry Pi* é um computador baseado em arquitetura *ARM*, e como tal, também necessita de um sistema operacional para operar. Existem alguns sistemas operacionais desenvolvidos para o *Raspberry Pi*, no entanto o mais conhecido e utilizado é o *Raspbian*, uma vez que é atualmente o sistema operacional oficial para todos os modelos do microcomputador, no endereço (<http://downloads.raspberrypi.org/raspbian/images/>) é possível acessar todas as versões de imagens do *Raspbian* (JUCA; SANDRO, 2018).

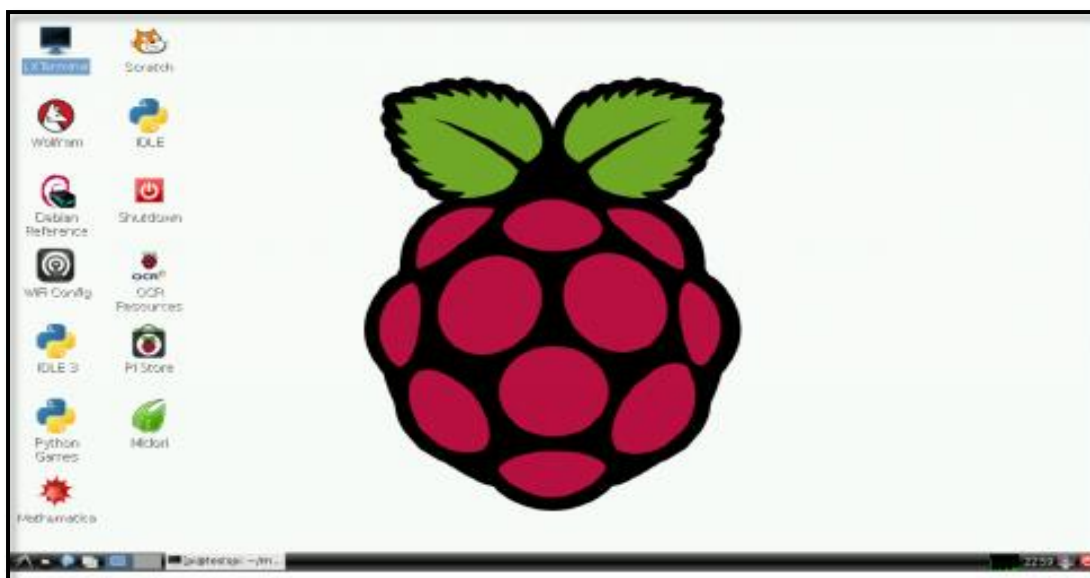
O *Raspbian* é um sistema baseado na distribuição LINUX e é um sistema quase completo, como padrão vem instalado diversas aplicações inclusive o *software* utilizado nesse projeto, o *Node-RED*, *drivers* mais usuais e algumas ferramentas que facilitam as configurações básicas. Existem alguns critérios mínimos que devem ser respeitados para um bom funcionamento do dispositivo que são:

- Fonte de alimentação de 5V e uma corrente de saída mínima de 2.5A;
- Cartão Micro SD com capacidade mínima de 4GB e de preferência classe 10.

A imagem do *Raspbian* existente no repositório da *Raspberry Pi Foundation*, deve ser gravada no *SD card*, que deve ser encaixada no *slot* do computador. Após a energização, o

módulo *Raspberry Pi* iniciará a etapa de *boot* e ao fim do processo o sistema está pronto para ser utilizado, a figura 11 exibe a tela gerada pelo *Raspberry Pi* para a *interface* do usuário.

Figura 11 – *Interface* gráfica do *Raspbian*



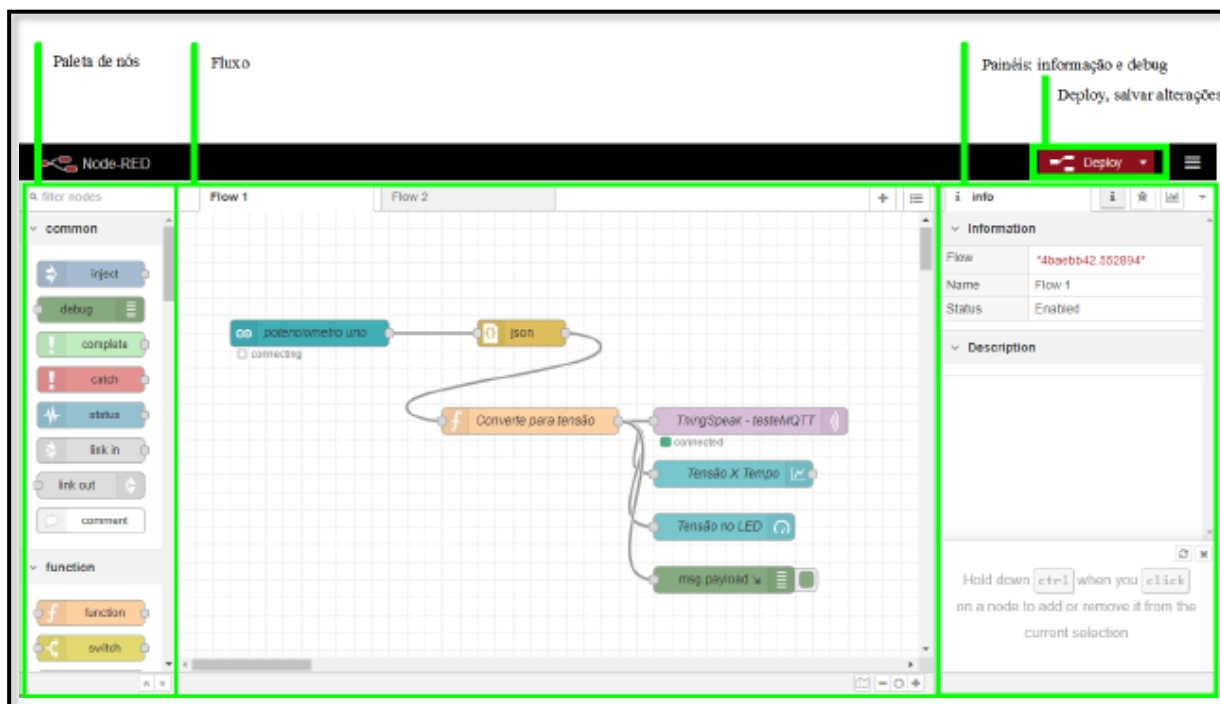
Fonte: O próprio autor

1.6.3 Node-RED

O *Node-RED* é um ambiente de desenvolvimento baseado em *JavaScript* de código aberto baseado em *Node.js*, desenvolvido por um grupo emergente de engenheiros da IBM em 2013 é um *software* excelente para o desenvolvimento de sistemas para aplicação em internet das coisas (*IoT*). Mais especificamente é um ambiente de programação virtual baseado em de criação de blocos e fluxo de dados, que interliga o meio físico para o ambiente computacional conectando o *hardware* e o *software*, é ideal para escrever processos de dados facilitando seu processamento (NODE-RED, 2021).

O *software* pode ser usado para compilar facilmente a lógica de processamento de dados e transferir os dados processados para o sistema de nível superior como um servidor SQL, um sistema de gerenciamento corporativo, serviço baseado em nuvem ou mesmo exibi-lo em *dashboard*. Em vez de programar uma página *web* exibindo dados de sensores diferentes e gerando código-fonte enormes, o *Node-RED* fornece uma *interface* de painel que permite criar uma *dashboard* de excelente qualidade sem a necessidade de conhecimento especial em HTML e CSS (FERENCZ; DOMOKOS, 2019).

Figura 12 – Ambiente de desenvolvimento *Node-RED*



Fonte: O próprio autor

O *Node-RED* contém nós que fornecem várias funções, muitas delas disponíveis no repositório do *GitHub*. Os processos criados são armazenados usando objetos *JSON*, o *Node-RED* consiste em 4 elementos básicos: painel de nó, painel de fluxo, painel de informações e painel de informação. A figura 12 ilustra seu ambiente de desenvolvimento, é ideal para a criação aplicativos de ambiente em tempo real e prototipagem, sua plataforma pode fornecer à indústria a capacidade de construir e testar novos protótipos integrados ou combinados com tecnologia existente antes de investir recursos significativos na inovação.

Uma biblioteca integrada permite que o usuário salve funções úteis, modelos ou fluxos para a reutilização. O tempo de execução do programa é leve e é montado na plataforma *Node.js*, aproveitando ao máximo seu modelo orientado a eventos. Isso o torna ideal para ser executado em hardware de baixo custo, como o *Raspberry Pi*, bem como na nuvem. Como mencionado anteriormente, o *Node-RED* tem código aberto e por esse motivo possui mais de 225.000 módulos no repositório de pacotes desenvolvidos por programadores do mundo inteiro, facilitando assim a extensão a gama de nós da biblioteca para adicionar novos recursos. (NODE-RED, 2021)

2 MÉTODOS E MATERIAIS

Neste capítulo é mostrado o desenvolvimento do projeto desde a pesquisa teórica em conjunto com as primeiras medições em campo, passando pela montagem do protótipo para teste de comunicação entre os módulos, até a criação do sistema e sua integração com o painel elétrico do forno industrial, estabelecendo a *interface* do sistema de monitoramento.

O sistema consiste em monitorar a temperatura do forno industrial por meio da leitura do sinal do termopar já existente no forno. Em seguida, o valor é injetado na entrada analógica da placa Arduino onde será ser convertido para um valor digital e direcionado para o computador *Raspberry Pi* modelo 3B+, através de comunicação serial via protocolo *Firmata*.

O valor discreto da temperatura será processado pelo *software Node-RED* versão 1.0.6, que irá gerar uma *dashboard* exibindo o dado através de duas formas distintas, a primeira, através de um mostrador por ponteiro tipo *gauge*, que mostra o valor instantâneo da temperatura, e a segunda através de um gráfico exibindo a evolução da temperatura ao longo de 12 horas, fornecendo assim um histórico da medição.

Baseado em sua lógica de programação em blocos o *software Node-RED (v1.0.6)* decidirá se enviará um alerta por e-mail e sms, parametrizado pelos limites de temperatura estabelecido. A *dashboard* mencionada terá uma segunda tela de navegação, contendo informações sobre o *status* do forno, assim como botões de ação que permitem ligar ou desligar partes específicas do forno. O sistema foi desenvolvido nas seguintes etapas:

Na primeira etapa foram realizadas pesquisas e estudos práticos na área de programação, dando ênfase aos principais comandos utilizados no ambiente do terminal *Linux*, e na sequência um estudo aplicado à programação em linguagem *JavaScript* e *C++*. Desta forma melhorando o entendimento e uso das ferramentas aplicadas no projeto. Foram acessados os *sites* dos desenvolvedores dos *softwares* e sistema operacional utilizados no sistema. Ainda nessa etapa foi adquirido o cartão micro SD e instalado o sistema operacional *Raspbian* versão 4.10 realizando as devidas configurações.

Na segunda etapa, já como os componentes adquiridos, foram iniciadas as instalações de *softwares* nos módulos. Primeiramente inserido o cartão micro SD na placa *Raspberry Pi (v1.0.6)* realizando assim o primeiro processo de *boot*, o próximo passo foi carregar o código *StandardFirmata* versão 2.1 na placa Arduino modelo Mega 2560, para depois dar início a

montagem do painel elétrico e realizar os testes de transmissão de dados no intuito de se observar o comportamento do sistema em caráter de teste.

Um ponto importante dessa etapa é que todos os testes foram realizados em modo local, ou seja, como um monitor conectado a saída HDMI e *mouse* e teclado conectados as portas usb da placa *Raspberry Pi (v1.0.6)* uma vez que, ainda não tínhamos acesso à rede industrial, pois o ponto de rede ainda estava em processo de instalação pelos responsáveis do setor de Tecnologia da Informação da empresa.

Na terceira etapa fez-se alterações no programa original do CLP modelo *Mitsubishi fx3* do forno, o intuito dessa alteração será melhor detalhado na seção 3.4.4. Mas em resumo, fora acrescentado 3 novas linhas de código com o objetivo de retransmitir o valor de temperatura fornecido pelo termopar, para uma saída analógica do CLP. Este sinal proveniente do CLP foi redirecionado para a entrada analógica da placa Arduino, desta forma fornecendo a informação de temperatura do forno ao sistema de monitoramento. As alterações efetuadas na programação *ladder* do CLP *Mitsubishi fx3* só foram possíveis, após um estudo criterioso do *datasheet* do fabricante quanto aos endereços de memórias e registradores reservados para a manipulação de valores junto a IHM do forno.

Nessa etapa também foi finalizado a infraestrutura de acesso à rede, e dessa forma foi possível conectar o sistema de monitoramento a rede industrial da empresa, o que possibilitou o acesso remoto ao sistema e facilitou o processo de ajuste fino junto a programação no *Node-RED (v1.0.6)*.

Na quarta etapa foram feitos os testes de comunicação do dispositivo *Raspberry Pi 3B+* com a rede industrial, simulado eventos de alarme para a verificação do envio de *e-mails* aos devidos responsáveis, assim como a configuração no ambiente de programação do *Node-RED (v1.0.6)* para estabelecer conexão com o servidor de uma empresa terceira, que é responsável pela conversão dos dados de alertas enviados, em mensagens de texto sms para *smartphones* cadastrados.

Vale ressaltar que, nessa fase foi realizado uma reunião de validação do projeto junto a diretoria da empresa, que levantou alguns pontos de melhorias no sentido de estabelecer segurança de acesso, entretanto de modo geral os diretores aprovaram e validaram o sistema de monitoramento.

2.1 MATERIAIS UTILIZADOS NO PROJETO

Para manter todos os módulos protegidos do calor e da poeira do ambiente externo, utilizou-se os seguintes itens.

- 1 painel elétrico CEMAR (600x300x200 mm).
- 1 peça de trilho para painel – modelo: din TS35.
- 1 peça de canaleta recortada aberta com tampa – modelo: HD1P – *Hellermann*.
- 2 ventiladores de painel 220V – Cod.: PAX-12 (Prabox).

Durante a montagem dos módulos do sistema de monitoramento, foi utilizado os materiais listados abaixo.

- 1 módulo de sistema embarcado Arduino Mega 2560.
- 1 placa borne para Arduino Mega 2560 – *Shield (v2)*.
- 1 módulo *Raspberry Pi 3B+*.
- 1 cartão micro SD 64GB classe 10 (Sandisk).
- 1 cabo de comunicação USB.
- 1 cabo de comunicação padrão *Ethernet* com conector RJ45.
- 1 fonte de alimentação entrada 220V – saída 5/12V.
- 1 módulo relé 8 canais - bobina 5V – modelo: SRD-05VDC-SL-C.
- 8 relés Siemens/Scrack – bobina 24V – modelo: RT424524.
- 40 Bornes Comuns – modelo: WEG 6mm Conexão BTWP.
- 10m de cabo fio flexível – modelo: AWG-13020PT 20 vias – 0.5mm.
- 3m de cabo flexível – modelo: Pp 3x1,5mm T-500V.
- Interruptor Seccionador 10 A com manopla – modelo: VCF3 – *Schneider Electric*
- Ferro de Solda – modelo: SC-60 220V - *Hikari*.
- Solda (Estanho).
- 50 terminais agulha.
- 50 terminais garfo.
- 12 espaçadores com rosca de latão.
- 12 parafusos Philips 2.5x10mm.
- 8 resistores de 1K Ω .
- 1 multímetro com alicate amperímetro – Flir – CM 46

3 IMPLEMENTAÇÃO DO PROJETO

Este capítulo apresenta o processo de desenvolvimento e implementação do presente projeto. Foram aplicadas diversas tecnologias e utilizadas diferentes plataformas de programação, além de realizado a montagem de um painel elétrico de padrão industrial, uma vez que foi desenvolvido um produto de aplicação prática.

São apresentados neste capítulo os seguintes tópicos:

- a) Instalação da Imagem do *Raspbian*;
- b) Configuração do *Node-RED* no *Raspberry Pi*;
- c) Instalação da biblioteca *Firmata* na placa Arduino;
- d) Desenvolvimento do programa de supervisão no *Node-RED*;
- e) Montagem dos módulos no painel elétrico;
- f) Parametrização do sistema para acesso à rede industrial.

3.1 INSTALAÇÃO DA IMAGEM DO RASPBIAN

O *Raspberry Foundation* declara o *Raspbian* como sendo o seu sistema operacional oficial, desta forma recomendando a utilização em seus módulos computacionais e oferecendo total suporte técnico desde sua instalação até sua operação, sendo assim seguimos todas as instruções do site oficial no processo de cópia da imagem do sistema operacional no cartão micro SD.

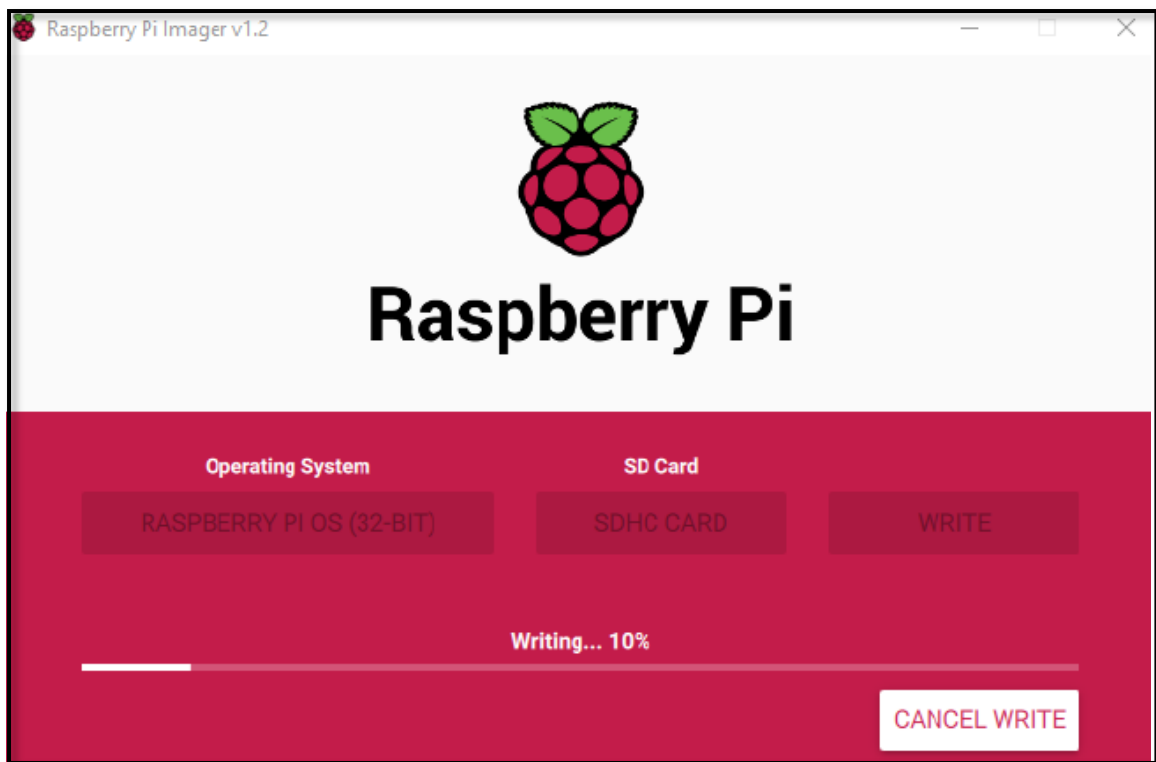
Existem algumas recomendações que a fundação julga importante, que são: adquirir uma fonte de alimentação estável e que forneça 5V de tensão com corrente mínima de 2.5A em sua saída, e um cartão micro SD com capacidade mínima de 4GB, para que o sistema operacional funcione sem apresentar problemas de travamento ou lentidão no processamento.

Uma vez cumpridas as exigências acima, foi acessado o repositório <https://www.raspberrypi.org/software/operating-systems/#raspberrypi-os-32-bit>, onde foi baixado o ***Raspberry Pi OS com desktop e software recomendado*** em sua versão 4.10, no entanto é comum que sistemas baseados em *Linux* sofram alterações constantes e de fato a versão mais recente disponível é a 5.10, mas isso não influencia no resultado final. Temos um ponto de atenção nesta etapa do processo, pois no repositório existem outras duas opções do sistema operacional, que são as versões somente com *desktop*, e a *Lite*. A opção *desktop* com

software recomendado já vem com o *Node-RED* e por consequência o *NodeJS* instalados, assim não precisamos nos preocupar com este processo.

Em seguida foi instalado o *software Raspberry Pi Imager* versão 1.2, disponível no repositório do site <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>, que é uma ferramenta para gravação do cartão SD, possui uma *interface* gráfica amigável e é de fácil utilização. O *software* faz o download da imagem e a instalação automaticamente no cartão SD, como mostrado na figura 13.

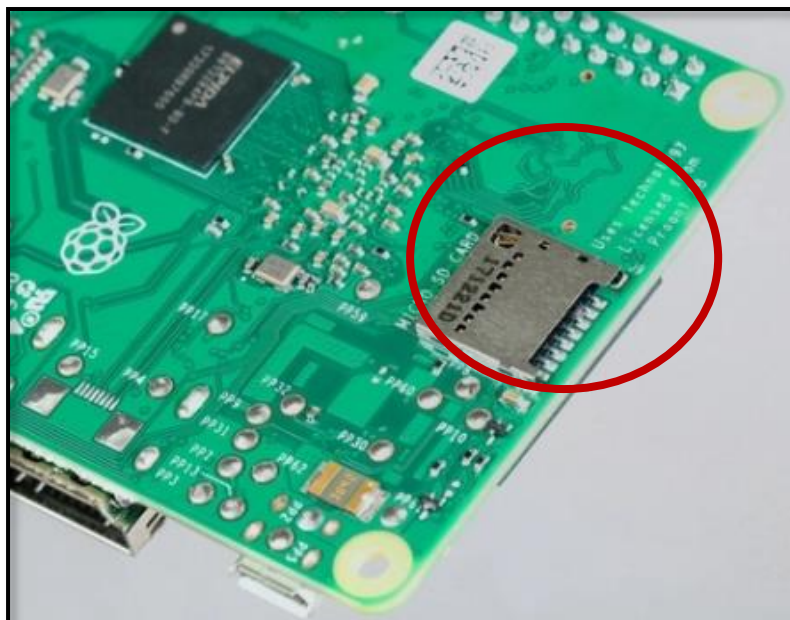
Figura 13 – Raspberry Pi Imager



Fonte: O próprio autor

Uma vez que o cartão micro SD esteja carregado com o sistema operacional, o passo seguinte se resume em encaixar o cartão no *slot* destinado para esse propósito na parte inferior da placa *Raspberry Pi* como mostrado na figura 14 e energizar o circuito com a fonte de 5V. A partir daí, inicia-se o processo de *boot* de forma automática e ao final deste ciclo temos o sistema operacional funcionando e pronto para operar, a figura 11 da seção 1.6.2 ilustra a aparência da *interface* gráfica do sistema em funcionamento.

Figura 14 – Posição de encaixe do cartão micro SD.



Fonte: O próprio autor

3.2 CONFIGURAÇÃO DO NODE-RED NO RASPBERRY PI

O *Node-RED* é um *software* que funciona dentro da plataforma do *NodeJS*, mas como informado na seção 3.1, uma vez que temos o sistema operacional recomendado em funcionamento no computador *Raspberry Pi* não há necessidade de realizar o processo de instalação de ambos os *softwares*. Desta forma nesta etapa foi necessário apenas checar e configurar pequenos detalhes de acesso ao *Node-RED*.

Uma vez no ambiente de área de trabalho do *Raspbian*, é necessário acessar terminal e digitar o comando:

```
➤ Node-red
```

O *software* será carregado e neste processo, que dura alguns segundos, será mostrado na tela várias informações referentes as versões dos subsistemas agregados ao *Node-RED*. A informação mais importante, no entanto, refere-se ao endereço de IP (*internet protocol* – protocolo de rede) que deve ser acessado através do navegador, a figura 15 apresenta o *Node-RED* em execução com o destaque para o endereço de acesso. Este número de IP refere-se ao endereço *host* interno da própria máquina, no entanto se o computador estiver vinculado a uma rede via acesso *Wi-Fi* ou mesmo cabeado, a rede em questão identificará o dispositivo com um

endereço válido, logo o *Node-RED* poderá a ser acessado através deste novo endereço, mantendo logicamente a porta de acesso **:1883**.

Figura 15 – Plataforma Node-RED em execução

```
16 Jun 17:11:33 - [info]
Welcome to Node-RED
=====
16 Jun 17:11:33 - [info] Node-RED version: v1.0.6
16 Jun 17:11:33 - [info] Node.js version: v12.18.1
16 Jun 17:11:33 - [info] Windows_NT 10.0.19041 x64 LE
16 Jun 17:11:36 - [info] Loading palette nodes
16 Jun 17:11:40 - [info] Dashboard version 2.16.2 started at /ui
16 Jun 17:11:41 - [info] Settings file : \Users\rechfy\.node-red\settings.js
16 Jun 17:11:41 - [info] Context store : 'default' [module=memory]
16 Jun 17:11:41 - [info] User directory : \Users\rechfy\.node-red
16 Jun 17:11:41 - [warn] Projects disabled : editorTheme.projects.enabled=false
16 Jun 17:11:41 - [info] Flows file : \Users\rechfy\.node-red\flows_Alessandro.json
16 Jun 17:11:41 - [info] Creating new flow file
16 Jun 17:11:41 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

16 Jun 17:11:41 - [info] Starting flows
16 Jun 17:11:41 - [info] Started flows
16 Jun 17:11:41 - [info] Server now running at http://127.0.0.1:1883/
```

Fonte: O próprio autor

Quando o computador é desligado, obviamente o programa *Node-RED* para de ser executado e não teremos mais o acesso ao mesmo. Deste modo, todas as vezes que religarmos o computador se faz necessário realizar o processo inicialização do programa via terminal, no entanto este problema foi contornado com a configuração do recurso de inicialização automática do *software* juntamente com o *boot* do sistema operacional, e para tal faz-se necessário digitarmos no terminal de comando o código mostrado abaixo:

```
➤ sudo systemctl enable nodered.service
```

Desta forma o sistema sempre estará ativo enquanto o sistema operacional estiver funcionando, sem a necessidade de intervenção de um usuário externo para habilitar o serviço.

3.3 INSTALAÇÃO DA BIBLIOTECA FIRMATA NA PLACA ARDUINO

A escolha do módulo Arduino se justifica neste trabalho por ser uma placa de aplicação e desenvolvimento barata e por oferecer o conceito de modularidade exigido no escopo do projeto, ou seja, em caso de defeitos a troca seria rápida e sem a necessidade de implementação e processos complexos. Portanto, para o funcionamento eficiente do sistema é necessário que seja importado da biblioteca e transferido para o Arduino um *firmware* que permita sua comunicação com o computador servidor *Raspberry Pi*, ou seja, a máquina que possui o servidor *Node.js* instalado e ficará ligado ao Arduino através de um cabo USB.

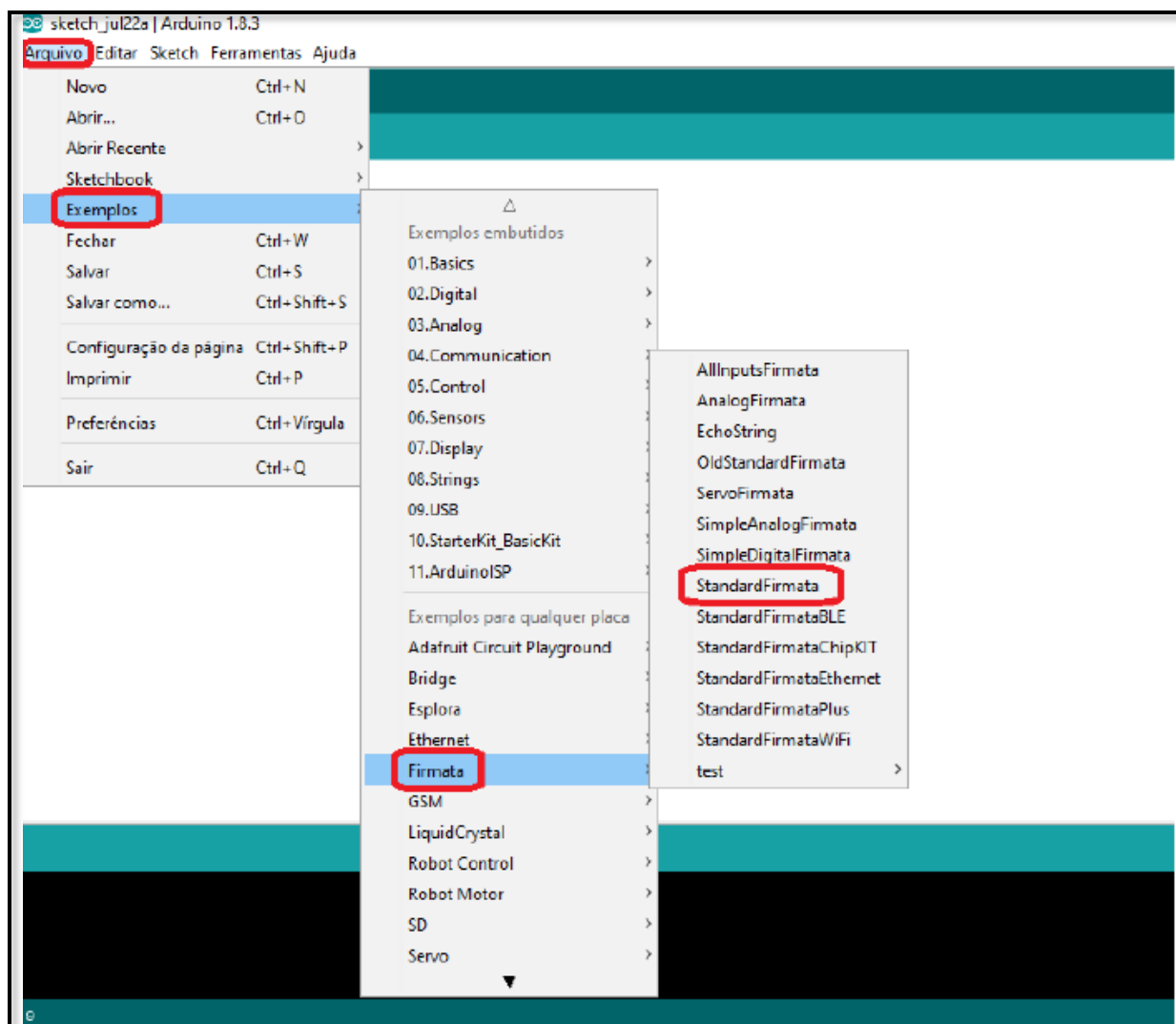
Os desenvolvedores da *nodered.org* sugerem a utilização do protocolo *Firmata* que é uma biblioteca de comunicação do *hardware* com o computador servidor. Ele permitirá que o *hardware* seja controlado pelo *software*. Então para esta implementação se fez necessário a instalação da biblioteca *Firmata* no Arduino como um *firmware* e assim ele estabelecerá a comunicação entre *hardware* e *software* no computador servidor para que este controle o Arduino.

Firmata é um protocolo genérico para comunicação com microcontroladores a partir de software, este protocolo pode ser implementado em qualquer microcontrolador e em qualquer arquitetura microcontrolada, e tem o objetivo de permitir o controle total do *hardware* do Arduino. O protocolo é baseado no formato de mensagem *midi* no qual os comandos são enviados em 8 bits e os dados em 7 bits, as mensagens podem ter qualquer tamanho.

A figura 16 ilustra o caminho executado dentro do ambiente *IDE* Arduino para selecionar o programa *StandardFirmata* que será carregado na área de edição de código. Antes do código ser enviado para o Arduino é necessário executar a verificação do código, esta etapa servirá para evitar a transferência de código com erros, o que pode danificar o microcontrolador. Após a verificação é necessário realizar o *upload* para a placa Arduino, ao termino do processo uma mensagem de confirmação será mostrada no rodapé inferior informando a confirmação da transferência do arquivo.

Ao termino da transferência, a comunicação entre o computador e o Arduino poderá ser feita permitindo que uma página web através do servidor *Node.js* controle todo o *hardware* do Arduino. Controlar o microcontrolador por meio do protocolo *firmata* tem suas vantagens e as utilizamos para contribuir no desenvolvimento dessa aplicação, podemos destacar o fato de que o programa implementado não estará limitado pela memória RAM ou *flash* do microcontrolador.

Figura 16 – Firmata Standard no IDE Arduino



Fonte: O próprio autor

3.3.1 Placa borne Arduino GPIO

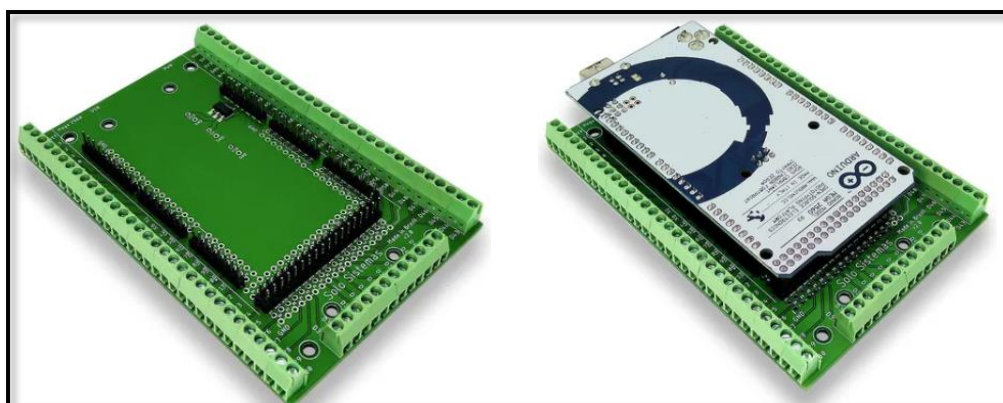
O padrão de pinos, que dão acesso aos contatos de entrada e saída nas placas Arduino em todos os modelos, são conectores fêmeas de modo que são encaixados terminais de modelo tipo agulha. Não é exagero afirmar que esse padrão apesar de prático, dificulta muito sua aplicação em projetos mais robustos, onde há a necessidade de conexões mais firmes e confiáveis.

Portanto, optou-se em incorporar ao projeto a *Shield* de expansão de bornes para Arduino Mega que é uma placa adaptadora que foi projetada para permitir ao projetista utilizar

conectores bornes em seus terminais. Por ser mais robusto e por consequência, menos susceptível a mau contato em comparação com as barras padrões de pinos.

A figura 17a mostra a aparência visual da placa, enquanto a figura 17b demonstra o conjunto Arduino e placa de expansão montados. A escolha dessa *Shield* no método de trabalho agilizou o processo de montagem do painel elétrico, que será visto mais à frente.

Figura 17 – Módulo Borne com Arduino Mega



(a) Shield Borne

(b) Conjunto Arduino/Shield

Fonte: O próprio autor

3.3.2 Arduino com *Pulldown* Externo

Como já mencionado, o protocolo *Firmata* permite que o computador servidor, neste caso o *Raspberry Pi*, controle o *hardware* do módulo Arduino. Desta forma o Arduino passa a se tornar uma extensão do *software Node-RED*, no entanto à medida que o projeto foi evoluindo, notou-se um problema vinculado a esse procedimento.

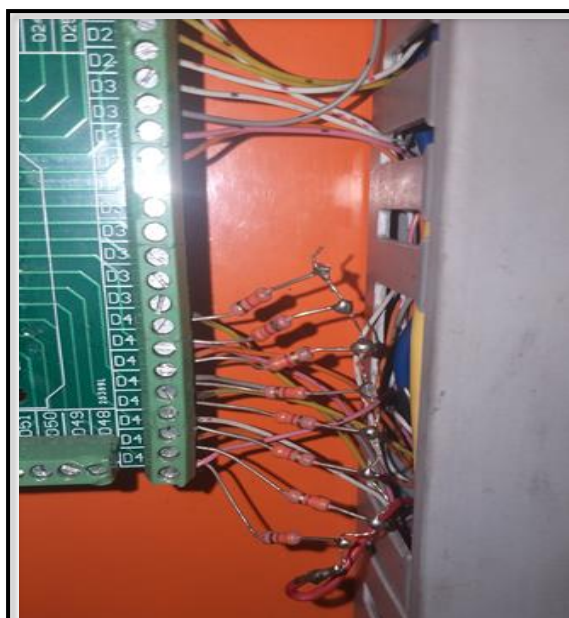
O projeto de arquitetura construtiva do *hardware* do Arduino permite que em todas as aplicações onde necessita-se de polarização resistiva nas entradas digitais, conhecido como modo *pullup* ou *pulldown*, sejam implementadas via parametrização inseridos nas linhas de código dos programas, de forma que não há a necessidade de utilização dos resistores externos.

No entanto o *software Node-RED* não dispõe dessa aplicabilidade, o que obrigou a inserção de uma rede resistiva ligado a todas as entradas digitais do sistema. Isso foi necessário para manter os pinos em nível lógico baixo quando nenhuma entrada estiver ativa. A figura 18 mostra a montagem do conjunto de resistores conectadas as entradas utilizadas no projeto.

Nesta etapa o único ponto de atenção foi com relação a escolha do valor do resistor a ser utilizado, por questões de disponibilidade de estoque foi decidido aplicar resistores de 660Ω , que em última análise, baseado na equação 4, atende as especificações de corrente máxima de 20mA por pino como visto na tabela 2.

$$R = \frac{5V}{660\Omega} = 7,5 * 10^{-3} A \quad (\text{Equação 4})$$

Figura 18 – Rede resistiva inserida no Arduino



Fonte: O próprio autor

3.4 DESENVOLVIMENTO DO PROGRAMA DE SUPERVISÃO NO *NODE-RED*

Uma linguagem de programação baseada em blocos funciona como um quebra-cabeça, onde cada peça é um comando e quando montamos uma sequência de peças conseguimos escrever um programa, desta forma o *software Node-RED* é baseado, uma forma simples e intuitiva de programação. Obviamente existem blocos especiais que necessitam de programação de funções mais específicas, que devem ser escritas em linguagem *JavaScript*. Nessa seção será descrita a programação de todo o sistema de controle e supervisão do sistema de monitoramento.

A programação de todo o sistema foi montada de forma semelhante a uma estrutura de linguagem ladder de um CLP, desta forma facilitando o entendimento e análise das linhas de comando.

3.4.1 Entradas e Saídas do Sistema

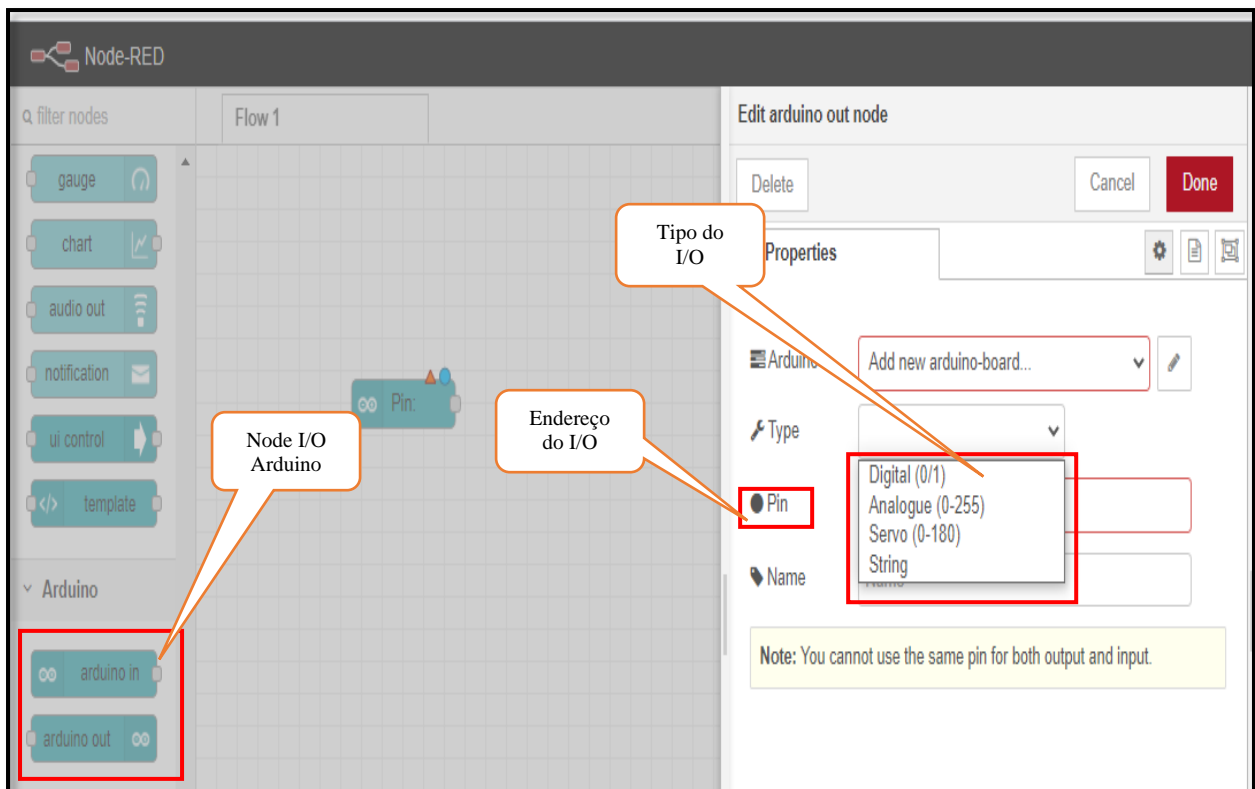
Primeiramente, usando o terminal de comando do *Raspbian* foi adicionado os ícones de acesso as entradas e saídas do Arduino no *Node-RED* com o seguinte comando:

```
➤ npm i -g node-red-node-arduino
```

Posteriormente foram definidos os endereços dos pinos destinados as 8 entradas e 8 saídas, pois foi desta forma que o sistema foi projetado. Então os pinos de 26 a 33 foram definidos como entrada, os pinos de 40 a 47 como saídas e o pino A0 como entrada analógica. A figura 19 ilustra o *layout* e a maneira como se configura os endereços de pinos no *software Node-RED*.

Cada endereço foi inserido um bloco dentro do Fluxo 1 e 2 (*Flow 1 e 2*) e feito o mesmo procedimento de configuração para cada um deles, seguindo o padrão de entradas do lado esquerdo e saídas do lado direito da tela.

Figura 19 – Configuração das entradas e saída no *Node-RED*



Fonte: Próprio autor

3.4.2 Bloco de função da temperatura e padrão de alarme

O bloco de função dentro do *Node-RED* tem a finalidade de executar todas as linhas de programas definidas pelo programador, portanto é um bloco especial que funciona como uma caixa preta. Foi necessário usar o bloco de funções com o objetivo de filtrar, retirar uma média dos valores de temperatura e ajustar os erros provenientes da conversão analógico-digital, o ajuste de erro se resume em multiplicar o valor por uma constante definida de forma empírica. Ao final da operação o valor resultante do código em questão é enviado para a próxima etapa que é responsável por exibir os valores em um gráfico, figura 20 mostra as linhas de programa desenvolvidas em linguagem *JavaScript*.

Figura 20 – Linhas de código de entrada do sistema

```
1 var adc = Number(msg.payload);
2 var temp = 0;
3 var i;
4 var msg1;
5 var msg2;
6 var temp_atual = (adc*0.960);
7-   for (i=0;i<5000;i++){
8       temp = temp_atual + temp;
9-   }
10 temp = temp/5000;
11 msg1 = {payload: parseInt(temp)};
12-   if(parseInt(temp) < 600){
13       msg2 = {payload: "TEMPERATURA BAIXA"};
14       return [msg1,msg2]
15-   }else if(parseInt(temp) >= 800){
16       msg2 = {payload: "TEMPERATURA ALTA"};
17       return [msg1,msg2]
18-   }else{
19       msg2 = {payload: "TEMPERATURA NORMAL"};
20       return[msg1,msg2]
21-   }
22
23
```

Fonte: Próprio autor

O segundo bloco de função utilizado dentro do sistema tem a finalidade de receber a média de temperatura, analisar sua oscilação e decidir, baseado nessa análise, se encaminha o sinal de alerta para o próximo bloco, que é responsável por enviar o e-mail e sms de alarme. O sinal de alarme é na verdade uma composição de um texto contendo as informações de data, hora e minuto do evento, o equipamento em questão e o *link* de acesso para o endereço da *dashboard*, a figura 21 mostra as linhas de códigos utilizada nesse referido bloco.

Figura 21 – Linhas de código do bloco de Alarme

```

Edit function node > JavaScript editor

1 | var temp = msg.payload;
2 | if (temp > 800 || temp < 540){
3 |     msg.to = "alessandro.soares@yamaha-motor.com.br";
4 |     msg.from = "alessandro.soares@yamaha-motor.com.br";
5 |     var d = new Date();
6 |     var mensagem = "Temperatura do Forno Fusor 3 LP em ";
7 |     msg.payload = "ALARME: " + d + mensagem + temp + " graus." + " LINK DE ACESSO: http://172.31.204.187:1880/";
8 |     return msg;
9 | }
10

```

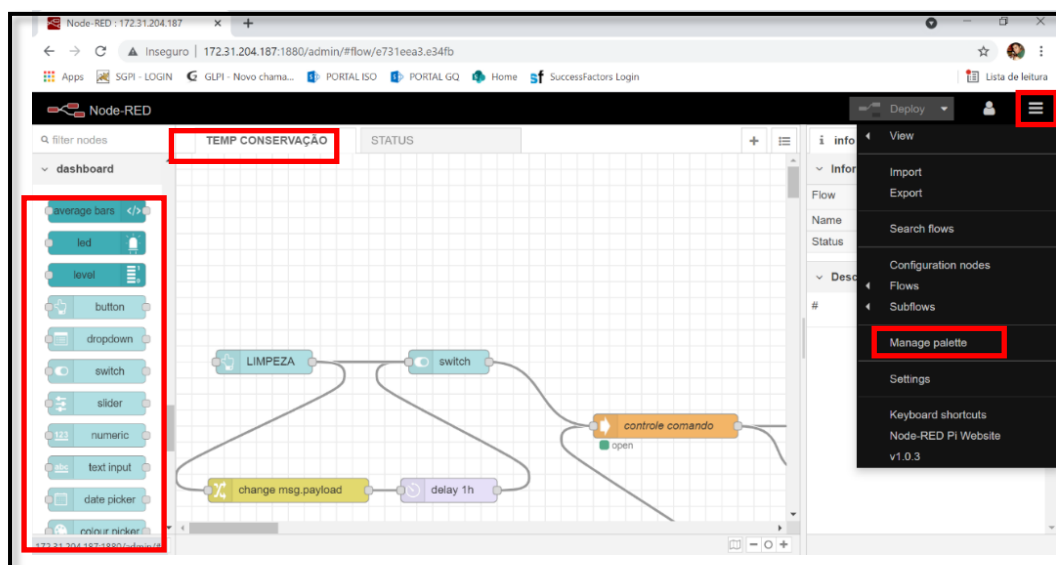
Fonte: O Próprio autor

3.4.3 Desenvolvimento da dashboard

O *Node-RED* possui a opção de acrescentar blocos funcionais diretamente de repositórios como *GITHUB*. Como mencionado anteriormente, por se tratar de um *software* de código aberto existem milhares de colaboradores no mundo que desenvolvem vários tipos de blocos, com variadas funções.

Dessa forma foi baixado do repositório um conjunto de blocos que são responsáveis por criar a tela de acesso gráfico do sistema, a figura 22 exhibe o caminho para baixar novas funções dentro do ambiente de desenvolvimento do *Node-RED*. Obviamente para baixar um bloco novo o *Raspberry Pi* precisa estar conectada à internet.

Figura 22 – Criação da dashboard no Node-RED

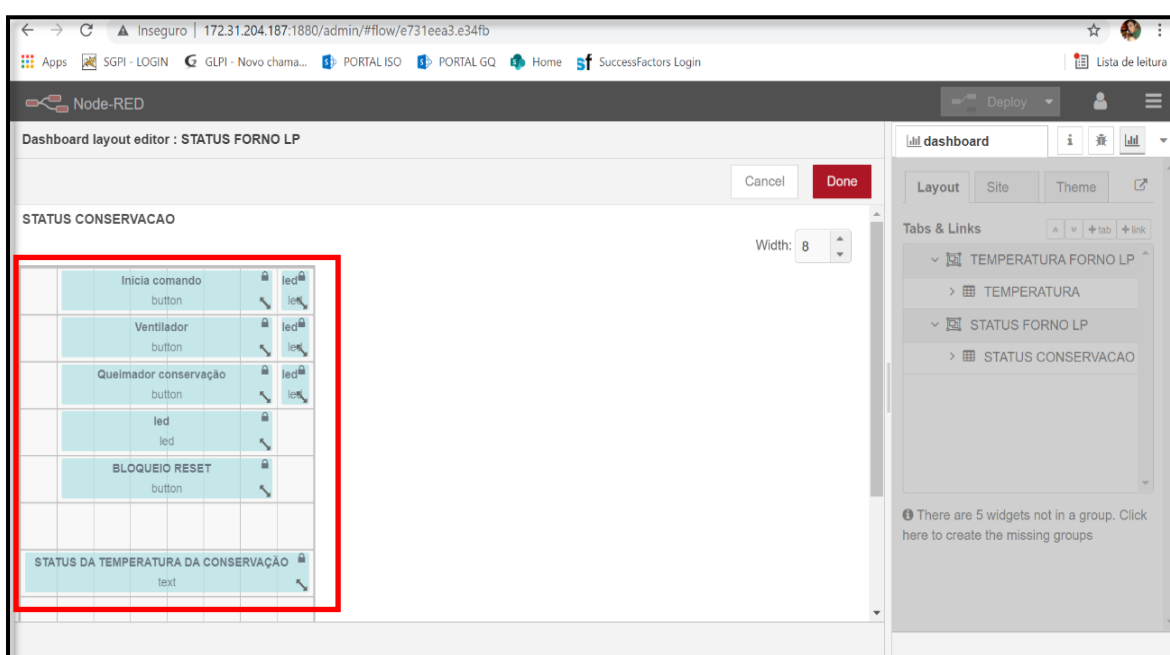


Fonte: O próprio autor

O conjunto de blocos com a descrição “*dashboard*” na aba esquerda da figura 22, uma vez inserido dentro do fluxo de trabalho o *software* cria automaticamente uma área onde pode ser efetuado o ajuste de layout da tela que será exibida, cada fluxo criado deve possuir uma identificação, isso facilita o gerenciamento de todos os blocos dentro do referido fluxo. A figura 23 mostra a tela de formatação do painel iterativo.

Neste projeto foi utilizado os blocos de gráfico (*chart*), medidor tipo ponteiro (*gauge*), botões (*button*), chaves (*switch*), texto (*text*) e lâmpadas (*led*) cada um deles foi configurado de acordo com a necessidade das especificações do projeto.

Figura 23 – Configuração da Tela Interativa



Fonte: O próprio autor

Pode-se observar na figura 22 que o sistema de monitoramento possui duas telas criadas e configuradas, o *layout* de exibição da tela chamada “*status conservação*” foi organizado de acordo com a figura 23, lado esquerdo, ou seja será a forma como o usuário irá visualizar a disposição dos elementos existentes na tela no celular ou na tela de um computador. Nota-se a facilidade de criação de sistemas com a *interface* gráfica com a utilização do *Node-RED*.

Especificamente nesse projeto ficou definido que a primeira tela irá exibir os valores de temperatura de duas formar diferentes, valor instantâneo representado pelo ponteiro, e o valor acumulado representado pelo gráfico. A segunda tela irá exibir o status do forno e todo o

controle pertinente ao comando elétrico do forno industrial, nessa segunda tela temos vários botões e lâmpadas que irão indicar o que está ligado ou desligado no forno em tempo real.

Observa-se a importância da segurança relacionado ao acesso deste sistema, pois o usuário pode realizar operações efetivas na funcionalidade do equipamento e isso pode se tornar um fator de risco, por esse motivo foi limitado o acesso aos encarregados e chefes dos setores responsáveis, essa limitação foi definida em reunião com a diretoria da empresa. Essa decisão ocorreu em virtude da impossibilidade de encontrar um bloco de função que realizasse a tarefa de cadastro de usuário para acesso das telas de trabalho.

3.4.4 Trabalhando com o sinal analógico de temperatura

O ponto central desse projeto é justamente a variável que irá ser medido e controlado, neste caso a temperatura. A técnica necessária para extrair uma medida elétrica analógica de um termopar e transformá-la em valor digital, de modo a ser interpretado pelo *software*, foi o maior desafio encontrado ao longo de todo o desenvolvimento do projeto. Houve algumas tentativas frustradas em realizar tal tarefa, pois de acordo com as primeiras medições feitas em campo e que são mostradas na tabela 3, os valores de tensão de trabalho giram em torno de algumas dezenas de milivolts e claramente temos uma relação linear.

Tabela 3 – Relação temperatura x tensão do forno

TEMPERATURA (°C)	TENSÃO (mV)
440	16.1
619	23.2
636	23.9
741	29.5
792	31.4
804	31.8

Fonte: O próprio autor

Como a faixa de trabalho do conversor A/D do Arduino varia de 0 a 5V, o primeiro passo seria amplificar o sinal em 100 vezes e depois amostrar esse valor amplificado no pino

A0. No entanto os circuitos testados como solução apresentaram falhas no momento da implementação. A solução encontrada para esse problema foi retransmitir o sinal digitalizado pelo CLP *Mitsubishi* do forno através de uma saída do cartão analógico, no entanto havia mais um ponto a ser considerado nesta solução, os níveis de tensão dos sinais analógicos tratados pelo CLP têm sua variação entre 0 e 10V.

Para contornar esse obstáculo tivemos a ideia de realizar uma operação matemática dentro do ambiente de programação do CLP para em seguida lançar esse valor na saída analógica do cartão, para implementar essa solução se fez necessário estudar em detalhes o *datasheet* do fabricante do CLP *Mitsubishi* modelo FX3, principalmente no tópico de endereçamento dos registros especiais de memória, e relação de entre os sinais digitais e analógicos do CLP.

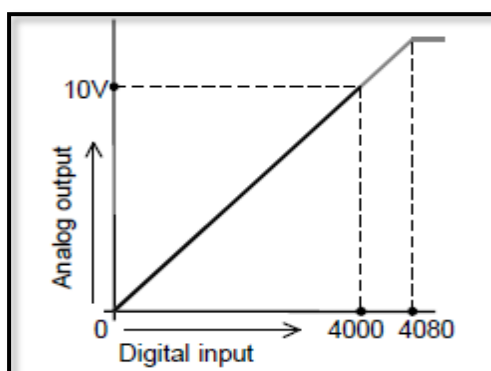
Na tabela 4 é verificado as características técnicas do cartão analógico de saída do CLP *Mitsubishi* modelo FX3, e na figura 24 temos o gráfico que especifica a relação de conversão dos valores digitais para valores analógicos.

Tabela 4 – Características técnicas do cartão analógico do CLP

Variação analógica de saída	0V a 10V DC
Quantidade de bits	12 bits
Resolução	2,5mV (10V/4000)
Acuracidade total	0,5% a 25°C e 1,0% até 55°C
Tempo de conversão D/A	200µs

Fonte: (MITSUBISHI ELETRIC, 2019)

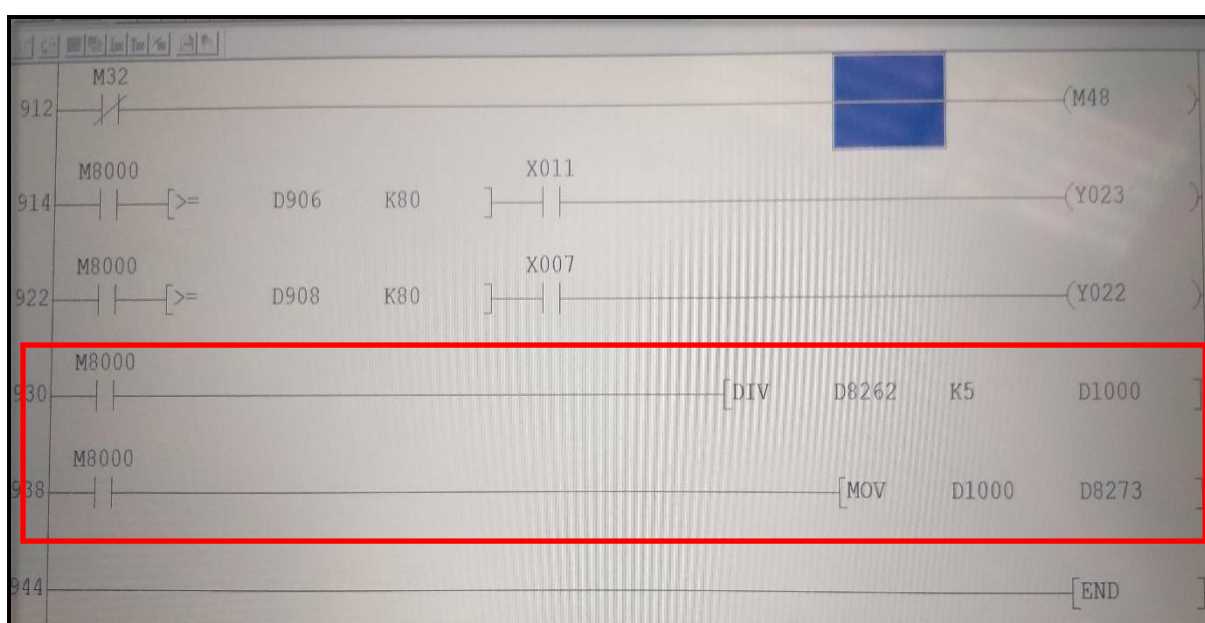
Figura 24 – Relação de conversão cartão analógico do CLP



Fonte: (MITSUBISHI ELETRIC, 2019, p.F7)

Na prática foi necessário encontrar o endereço de memória D8262 que guarda o valor digitalizado da temperatura proveniente do termopar, realizar uma divisão por uma constante igual a “5” guardando o resultado no endereço D1000, depois mover o valor do endereço D1000 para o registrador D8273, que é exatamente o endereço reservado para a saída número 4 do cartão analógico. Todos esses passos foram escritos em duas linhas de código na linguagem *ladder* como mostrado na figura 25.

Figura 25 – Programação *ladder* no CLP *Mitsubishi*



Fonte: O próprio autor

Ao final da implementação da solução tivemos como resultado um sinal proporcional a real temperatura do forno e com uma variação de tensão entre 0 e 5 volts, exatamente o que desejávamos como resposta. Obviamente, o fato de o sinal originado do termopar ter passado por dois processos de conversão, era de se esperar que houvesse perda de informação. No entanto como mencionado no item 3.4.2 e mostrado na figura 20, essa perda foi tratada realizando o ajuste com o auxílio de uma constante alcançada de forma empírica.

Foram feitos testes durante duas semanas para avaliar as variações dos valores indicado no sistema de monitoramento, e tivemos um desvio de 2°C a 3°C acima do valor real medido no termopar. Levando em consideração que estamos tratando de um forno que trabalha na faixa de 750°C em média, essa variação é mais do que aceitável para um sistema com custo de implementação abaixo dos R\$ 2.000,00.

3.5 MONTAGEM DOS MÓDULOS NO PAINEL ELÉTRICO

Esse item descreve o processo de instalação dos componentes no painel elétrico do sistema de monitoramento e sua integração com o comando elétrico do forno industrial, esta etapa caracteriza-se por ser um serviço extremamente braçal, mas isso não significa a ausência de técnicas para a sua realização. A disposição da montagem dos componentes dentro do painel seguiu uma sequência de modo a facilitar a manutenção de cada módulo.

Figura 26 – Disposição interna dos módulos no painel

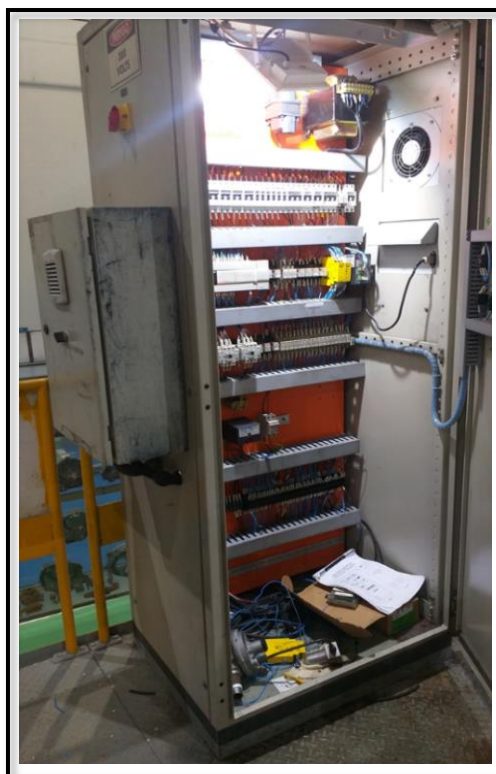


Fonte: O próprio autor

A figura 26 mostra a estética do painel de monitoramento, na imagem podemos verificar a organização e segurança do painel montado, itens esses exigidos pelas normas NR-10, NBR 5410 e a NBR IEC 60439-3, importante mencionar que houve alguns problemas relacionados a essa etapa do projeto, principalmente com relação ao acondicionamento dos componentes internos.

Primeiramente é importante entender que o painel do sistema de monitoramento foi agregado a estrutura do painel principal do forno, como mostra a figura 27, que fica a poucos metros da zona de trabalho onde o alumínio é derretido e mantido em sua forma líquida, sendo assim um ambiente de temperatura elevada e com a alta concentração de partículas em suspensão.

Figura 27 – Painel do sistema integrado ao forno industrial



Fonte: O próprio autor

Logo foi necessário a escolha de um painel com uma vedação apropriada, esse item foi tratado como ponto importante dentro do planejamento do projeto, mas isso causou um efeito colateral. Internamente a temperatura do painel de sistema de monitoramento estava alta e isso se agravou com a condução do calor gerado pelo painel principal, com isso o cooler do *Raspberry Pi* não era suficiente para resfriar o processador do computador.

Durante 2 semanas foi feito o monitoramento da temperatura do processador e verificado um aumento expressivo no mesmo, a ponto de comprometer a comunicação entre os módulos *Raspberry Pi* e Arduino. Para solucionar esse problema foi acrescentado dois ventiladores específicos para aplicações em painéis elétricos, sendo um deles montado em cima da placa *Raspberry Pi* com o intuito de manter a temperatura de trabalho sempre abaixo de 50°C.

Desta forma o objetivo foi alcançado e eliminamos mais um problema que surgiu durante a execução do projeto. Essa etapa foi encerrada com a ligação da fiação elétrica entre os dois comandos, realizando a integração entre os dois sistemas, obviamente foi realizado os testes de funcionalidades dos controles e conexão entre os módulos.

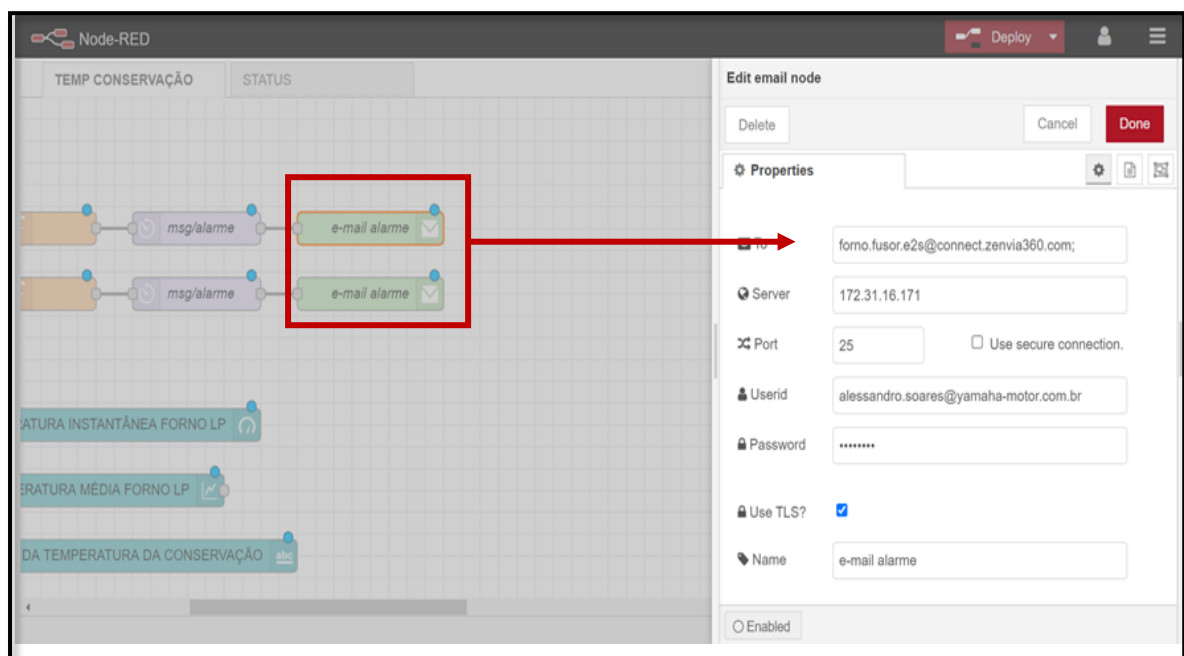
3.6 PARAMETRIZAÇÃO DO SISTEMA PARA ACESSO À REDE INDUSTRIAL

Esta foi a última etapa do processo de desenvolvimento do projeto, nesse período o sistema já estava funcionando de forma satisfatório, no entanto restava definir um IP fixo na rede, e parametrizar dentro do ambiente de programação do *Node-RED* o bloco responsável pelo acesso ao servidor de e-mail e o servidor do serviço de sms, este último se trata de um serviço pago.

Todos esses itens descritos necessitavam do auxílio do setor de tecnologia da informação da empresa, pois o referido setor é quem iria definir e configurar os parâmetros para o acesso à rede industrial, definir o IP (*internet protocol*) reservado para esse tipo de serviço, fornecer os dados para o acesso ao servidor e por fim tratar com a empresa que presta o serviço de sms.

Depois do processo de identificação na rede industrial do *Raspberry Pi* com o IP fixo, o próximo passo foi configurar o bloco de acesso ao servidor de e-mail e SMS, a figura 28 mostra o bloco que realiza tal conexão. Após a finalização das referidas configurações foram realizados os testes de envio de alarmes para a caixa de entrada de todas as pessoas cadastradas no serviço, essas pessoas estavam definidas de acordo com as saídas da reunião que envolveram os gestores responsáveis pelas respectivas áreas.

Figura 28 – Configuração de acesso à rede



Fonte: O próprio autor

Um detalhe importante e que só foi identificado após o sistema ter entrado em funcionamento, é referente a inserção de blocos de atrasos antes de todas as funções principais e o bloco de envio de alertas, isso porque a frequência com que as mensagens estavam sendo enviadas era proporcional a frequência do tráfego de dados, ou seja, a quantidade de mensagens era muito grande o que ocasionava excesso de mensagens nas caixas de entrada do correio eletrônico de todos os participantes.

4 RESULTADOS OBTIDOS

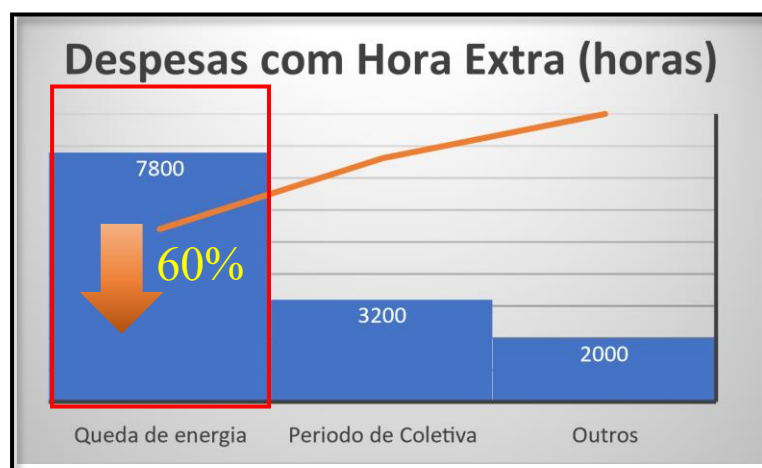
Este capítulo mostra os resultados obtidos após a implementação de todas as etapas do projeto, e levando em consideração que o objetivo final do trabalho era consiliar o desenvolvimento de um projeto de pesquisa satisfatório com relevância acadêmica, e ao mesmo tempo ter um alinhamento com os interesses da empresa, que afinal foi quem patrocinou e acreditou nas ideias apresentadas como sendo atingíveis, entendemos que tal objetivo foi alcançado dentro das especificações programada.

4.1 RESULTADO DO SISTEMA DE MONITORAMENTO REMOTO DO FORNO

A última fase, que consistiu na integração do sistema de monitoramento com a rede industrial da empresa foi a mais delicada, pois dependia da atuação de outro departamento, nesse caso o setor de Tecnologia da Informação, que conta com um número reduzido de pessoas comparado com as atividades exercidas dentro da empresa, e que se agravou devido o fator pandemia, obrigando a empresa a adotar o sistema de *homeoffice*.

Pórem, após a conclusão dessa etapa foi possível realizar os testes de desempenho do sistema, que por sinal foi considerado satisfatório pelo setores envolvidos. O objetivo esperado pela empresa era ter em mãos um sistema que viabilizasse o gerenciamento operacional aplicando os conceitos de indústria 4.0, reduzindo o índice de custo gerado pelo fator hora extra em intervenções no equipamento e investindo o mínimo de recursos financeiro.

Figura 29 – Gráfico de objetivo de meta da empresa

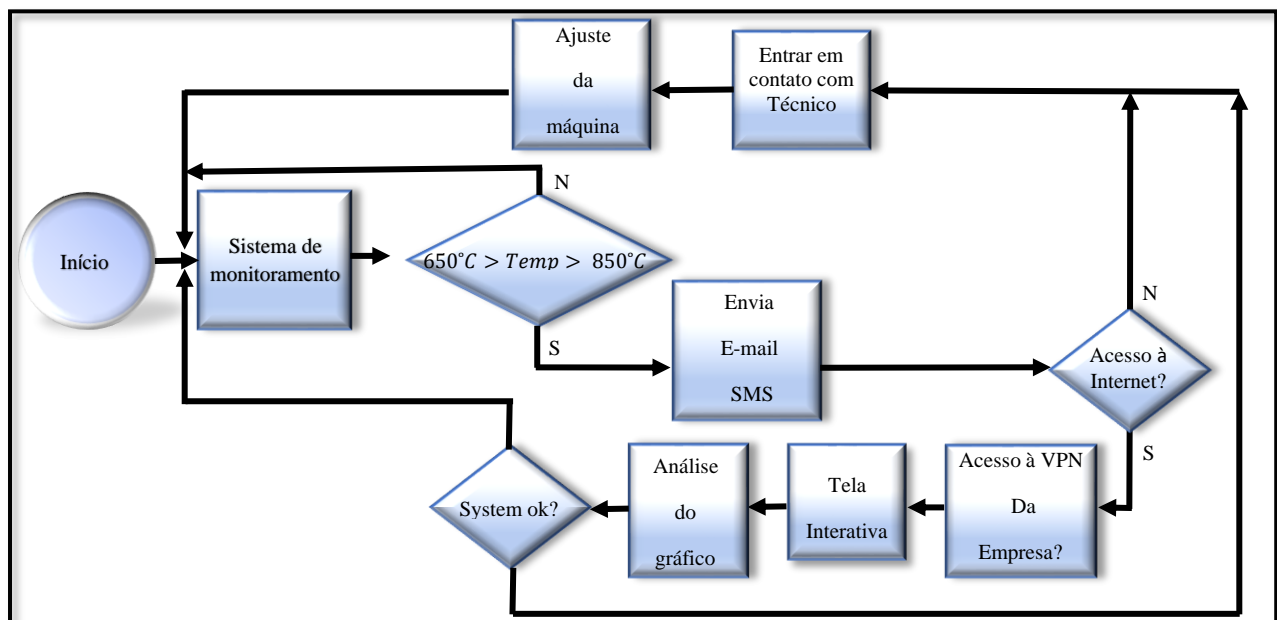


Fonte: O próprio autor

O gráfico mostrado na figura 29 que estratifica o valor de hora extra nos anos de 2018 e 2019 referente ao setor de manutenção da empresa em questão, a proposta de solução que o projeto oferece é a redução de 60% do valor total de horas. Ainda não se tem dados suficientes para que seja feito o levantamento dos resultados após a implementação do trabalho, isso em virtude da crise que a pandemia causou, o que impactou diretamente na coleta desses dados. Entretanto, o presente projeto alterou a sistemática de trabalho utilizada pelo setor onde o sistema foi implantado. Uma vez que quem fazia o alerta, em relação ao alarme de baixa temperatura no forno, era o grupo terceirizado responsável pela segurança patrimonial.

Hoje o fluxo seguido é demonstrado na figura 30, onde podemos analisar que o sistema de monitoramento remoto permite ao usuário interagir com o sistema de forma que permita a antecipação de ações no sentido de correção de falhas. Claro que o sistema tem suas deficiências, e isso também é evidenciado no fluxograma, pois se o usuário não puder acessar o sistema, por motivo de falta de acesso à internet ou mesmo a própria rede interna da empresa, o fluxo antigo volta a prevalecer.

Figura 30 – Fluxograma operacional



Fonte: O próprio autor

Este projeto tinha como intuito realizar o acesso ao sistema através da internet aberta, sem restrições e com o auxílio da computação em nuvem, ou seja, a ideia original tinha como premissa o desenvolvimento do sistema dentro do ambiente de serviço como a *AWS* ou *AZURE*, entretanto isso não foi possível devido a restrições impostas pela setor de Segurança da

Informação e Diretoria da empresa, com a aplicação da política interna da LGPD (Lei Geral de Proteção de Dados). Logo, em última análise não foi um problema técnico e sim uma determinação imposta pela empresa no sentido de salvaguardar possíveis acessos indevidos.

Mas como resultado final temos um sistema capaz de monitorar a temperatura do forno industrial e avaliar suas variações ao longo do tempo, e caso necessário emitir mensagens de alertas acerca de desvio do valor padrão do processo previamente especificado, as figuras 31 e 32 ilustram o aspecto visual que qualquer usuário terá ao acessar remotamente o sistema.

Figura 31 – Visualização da primeira tela do sistema



Fonte: O próprio autor

Na primeira tela teremos dois pontos de análise, o primeiro é um gráfico que mostra o registro da temperatura ao longo de 12 horas, valor limite imposto pelo software, e o segundo é um ponteiro que fornece o valor instantâneo da temperatura em tempo real, existe ainda um

botão que tem a função de bloquear o envio das mensagens de alertas, este último foi uma requisição solicitada pelo usuário responsável do setor onde se implementou o projeto.

É importante lembrar que no gráfico da temperatura os valores processados após as 12 horas mencionadas anteriormente serão perdidos, isso por que não temos um sistema de banco dados. O *software Node-RED* permite a inclusão e configuração de um banco de dados, no entanto não tínhamos recursos suficientes para tal implementação, logo consideramos como um ponto de melhoria para trabalhos futuros.

Figura 32 – Visualização da segunda tela do sistema



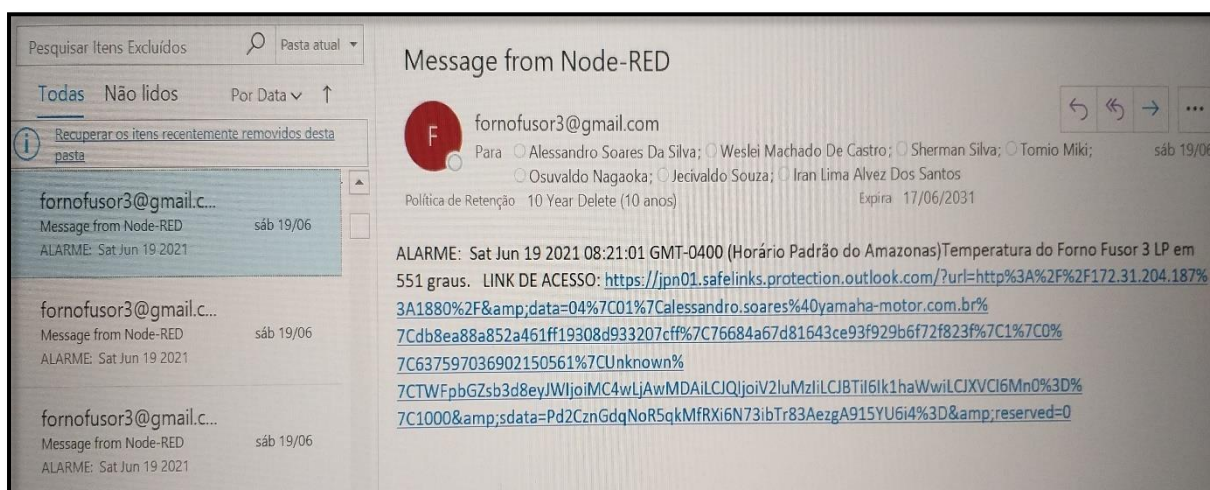
Fonte: O próprio autor

A segunda tela é responsável pelo gerenciamento do sistema elétrico do forno e seu funcionamento, a tela possui lâmpadas e textos de status, além de botões que permitem a integração dos comandos pertinentes ao forno. Em outras palavras é possível ligar e desligar

o equipamento através da *interface* gráfica mostrada na figura 32, e foi justamente essa funcionalidade que preocupou a diretoria no sentido de restringir sua funcionalidade exclusivamente dentro da rede interna da empresa, o que nos limitou no sentido da abrangência de acesso, com isso temos o segundo ponto de melhoria para trabalhos futuros, que é o desenvolvimento de um sistema que tenha o controle de acesso mais robusto e seguro.

A figura 33 mostra a estrutura do corpo da mensagem de alerta enviada para o e-mail das pessoas cadastradas, o modelo do texto em questão é o mesmo nas mensagens sms, bastando o usuário clicar no *link* em azul e automaticamente o *browser* irá direcioná-lo para a tela do sistema, porém o usuário deve estar acessando a rede industrial da empresa, ou seja, fora dela precisa realizar o acesso VPN, e neste ponto temos a limitação mencionada anteriormente.

Figura 33 – Modelo de e-mail da mensagem de alerta



Fonte: O próprio autor

CONCLUSÃO

No desenvolvimento do presente projeto de pesquisa foram realizadas revisões acerca de assuntos como: medição de temperatura por meio de sensores industriais, técnicas de conversão de sinais, as tecnologias mais recentes aplicadas a indústria 4.0 e seus parâmetros de configuração. Foram abordados em seguida a respeito das características técnicas da placa *Raspberry Pi* e Arduino, pois estes foram de suma importância para a implementação do *hardware* do sistema. Também consta uma apresentação sobre os softwares Arduino IDE, *Node-RED* e o sistema operacional *Raspbian*, utilizados no desenvolvimento de aplicação de medição de temperatura e no sistema de controle e monitoramento remoto.

No desenvolvimento do projeto observou-se a importância do conhecimento multidisciplinar obtido na academia, pois para se construir um sistema de controle dessa magnitude é necessário aplicar inúmeros conhecimentos teóricos e práticos para se obter um resultado satisfatório. É comum não possuir todas as técnicas e conhecimentos necessários em projetos industriais, e nesse sentido pode-se observar a versatilidade que um engenheiro de campo deve possuir, nem sempre os métodos aplicados funcionam e cabe ao profissional encontrar diferentes soluções para os problemas.

No sistema de monitoramento se obteve a plotagem dos valores de temperatura no gráfico como o esperado. Fez-se uma conexão do sistema com a rede industrial da empresa em questão. Para trabalhos futuros pode-se implementar como melhorias, um sistema de monitoramento com mais segurança de acesso, capaz de monitorar e identificar todas as pessoas previamente cadastradas e gerar um relatório completo e se necessário bloquear o seu acesso. Além de desenvolver um banco de dados para armazenar as informações recebidas em arquivos para que se possa realizar um efetivo gerenciamento de processo.

Com base nos resultados obtidos, verificar-se que, por meio da utilização de tecnologias baseadas em código aberto, como é o caso do *software Node-RED*, é possível solucionar o problema apresentada neste trabalho, que foi, monitorar a temperatura de um forno industrial e controlar o sistema do mesmo de forma remota e com baixo custo de investimento. Desta forma, a hipótese apresentada na introdução teve sua veracidade comprovada.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALUTAL. **Termopar, termorresistência e sensor de temperatura para fundição de alumínio**. [s.d]. Disponível em <<https://www.alutal.com.br/br/produto/termopar-termorresistencia-sensor-de-temperatura-para-fundicao-de-aluminio>>. Acesso em 01 jun. 2021.
- ARDUINO. **Arduino Mega 2560**. [s.d]. Disponível em <<https://store.arduino.cc/usa/mega-2560-r3>>. Acesso em 20 mai. 2021.
- ARENY, R. P. **Sensores y Acondicionadores de Señal**. 4. ed. Barcelona: Marcombo S.A, 2001.
- BENTLEY, J. P. **Sistemas de medición, principios y aplicaciones**. 1. ed. México: Continental S.A, 1993.
- BLUM, J. **Explorando o Arduino: Técnicas e ferramentas para mágicas de engenharia**. 1. ed. Rio de Janeiro: Alta Books, 2016.
- CONFEDERAÇÃO NACIONAL DA INDÚSTRIA. **Desafios para indústria 4.0 no Brasil**. 2016. Disponível em <https://static.portaldaindustria.com.br/media/filer_public/d6/cb/d6cbfba-4d7e-43a0-9784-86365061a366/desafios_para_industria_40_no_brasil.pdf> Acesso em 15 mar. 2021.
- FERENCZ, K.; DOMOKOS, J. **Using Node-RED platform in an industrial environment**. 2019. Disponível: em < <http://opiware.com/wp-content/uploads/2020/12/Node-RED-in-industrial-env-.pdf> >. Acesso em 10 jan. 2021.
- FLOYD, T. L. **Sistemas digitais: fundamentos e aplicações**. 9. ed. Porto Alegre: Bookman, 2007.
- ISKANDAR, A; MUHAJIRIN, M; LISAH, L. **Sistem Keamanan Pintu Berbasis Arduino Mega**. 2017. Disponível em < <http://journal.upgris.ac.id/index.php/JIU/article/view/1803>>. Acesso em 20 abr. 2021.
- JUCA, S; PEREIRA, R. **Aplicações Práticas de sistemas embarcados Linux utilizando Raspberry Pi**. 1. ed. Rio de Janeiro: PoD, 2018.
- KAWAMURA, D. K. C. **Estudo exploratório em Nodejs e Node-RED para internet of things**. 2017. 56p. Dissertação (Graduação), Instituto Municipal de Ensino Superior de Assis – IMESA – Assis, 2017.
- MAN, J. **Raspberry Pi 3 Model B Technical Specifications**. 2016. Disponível em <<https://www.element14.com/community/docs/DOC-80899/1/raspberry-pi-3-model-b-technical-specifications>>. Acesso em 30 abr. 2021.
- MAZZAFERRO, J. A. E. **Indústria 4.0 e a qualidade da informação**. 2018. Disponível em < <https://www.scielo.br/j/si/a/z9VfBmny3hhvcWvTsxW6YzN/?lang=pt&format=pdf> >. Acesso em 15 jan. 2021.
- MITSUBISHI ELECTRIC. **Programmable Controller: User's Manual**. 2019. Disponível em <https://dl.mitsubishielectric.com/dl/fa/document/manual/plc_fx/jy997d16701/jy997d16701r.pdf>. Acesso em 15 fev. 2021.
- NODE-RED. **Site Oficial do Node-RED**. 2021. Disponível em <<https://nodered.org/>>. Acesso em 01 fev. 2021.

OLIVEIRA, S. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. 1. ed. São Paulo: Novactec, 2017.

OLIVEIRA, J. R.; SHIN-TING, W. **Conversores entre Sinais Analógicos e Digitais**. 2019. Disponível em < <https://www.dca.fee.unicamp.br/courses/EA075/2s2019/notas/topico8.pdf> >. Acesso em 20 fev. 2021.

RIBEIRO, M. A. **Automação Industrial**. 4. ed. Salvador: Tek Treinamento & Consultoria Ltda, 1999.

RITZMAN, L. P.; KRAJEWSKI, L. J. **Administração da produção e operações**. 1. ed. São Paulo: Pearson Prentice Hall, 2004.

ROGGIA, L.; FUENTES, R. C. **Automação Industrial**. 1. ed. Santa Maria: Rede e-Tec Brasil, 2016.

SEDRA, A. S.; SMITH, K. **Microeletrônica**. 5. ed. São Paulo: Pearson Prentice Hall, 2007.

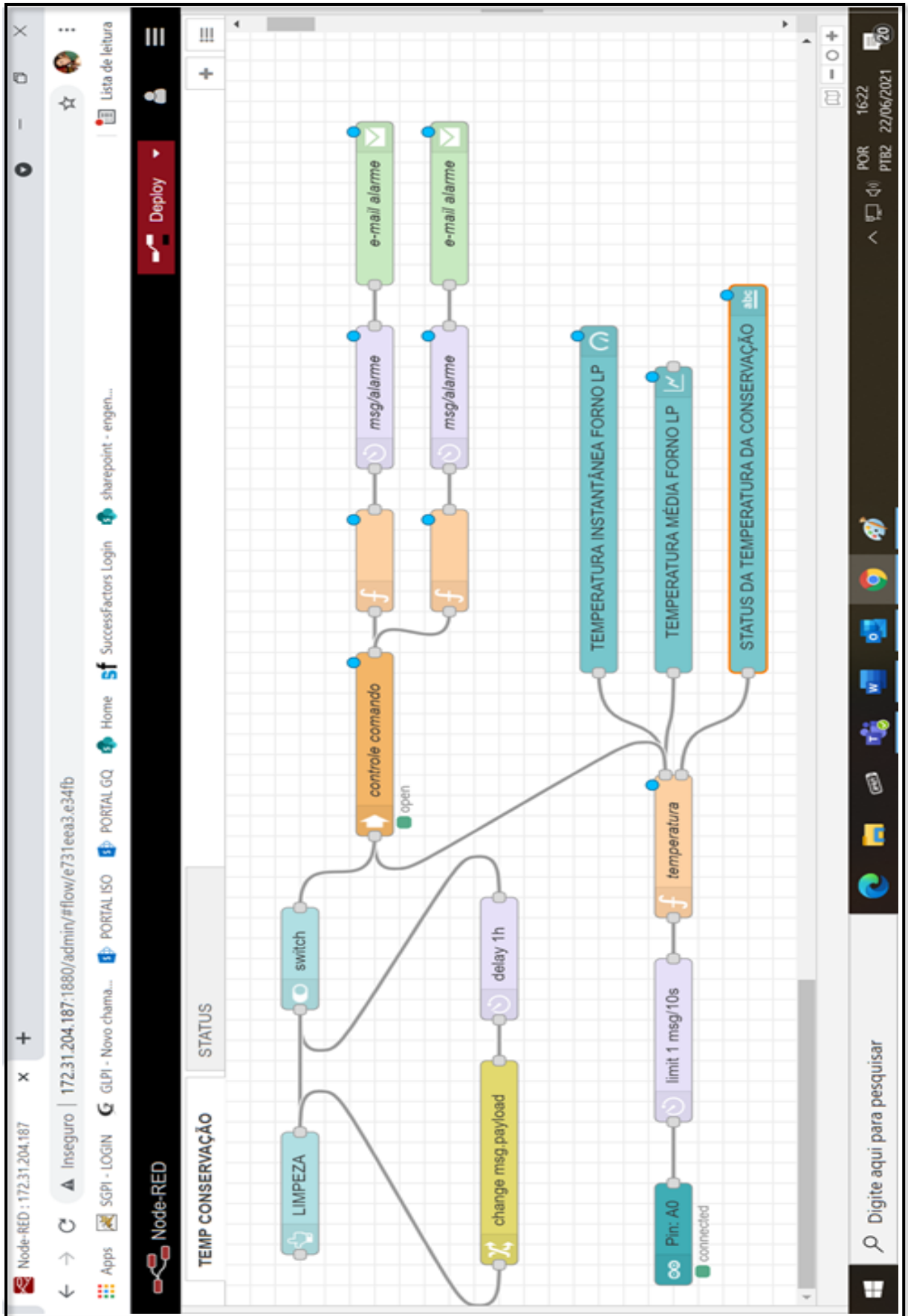
SILVEIRA, L.; LIMA, W. Q. **Um breve histórico conceitual da Automação Industrial e Redes para Automação Industrial**. 2003. Disponível em < https://www.dca.ufrn.br/~affonso/FTP/DCA447/trabalho1/trabalho1_13.pdf >. Acesso em 30 abr. 2021.

TANENBAUM, A. S.; BOS, H. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2016.

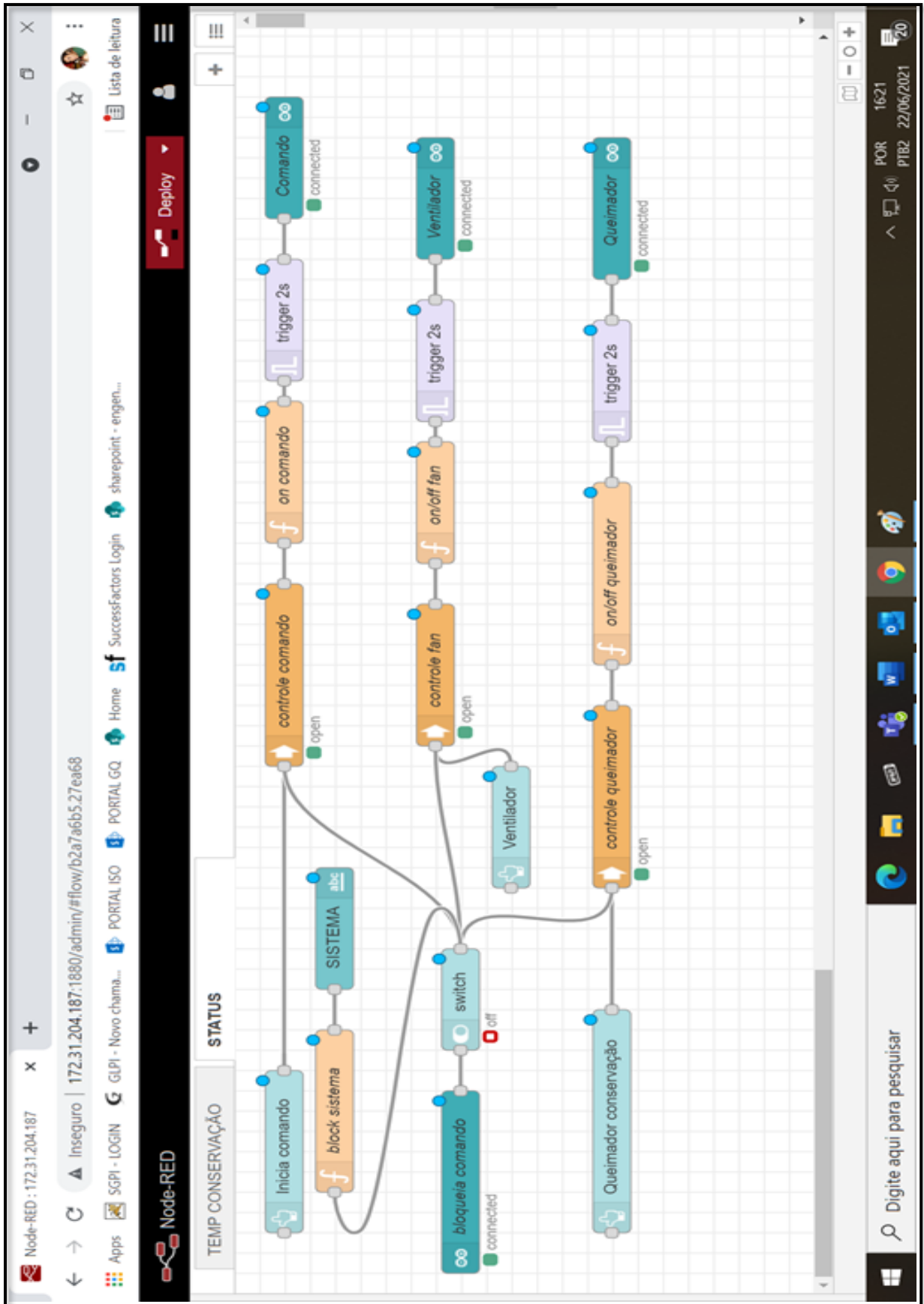
TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas digitais: princípios e aplicações**. 11. ed. São Paulo: Pearson Prentice, 2011.

WIGUNA, E. H.; SUBARI, A. **Rancang Bangun Sistem Monitoring Ketinggian Air Dan Kelembaban Tanah Pada Penyiram Tanaman Otomatis Dengan HMI (Human Machine Interface) Berbasis Raspberry Pi Menggunakan Software Node-RED**. 2017. Disponível em < <https://doi.org/10.14710/gt.v19i3.21878> >. Acesso em 20 mar. 2021.

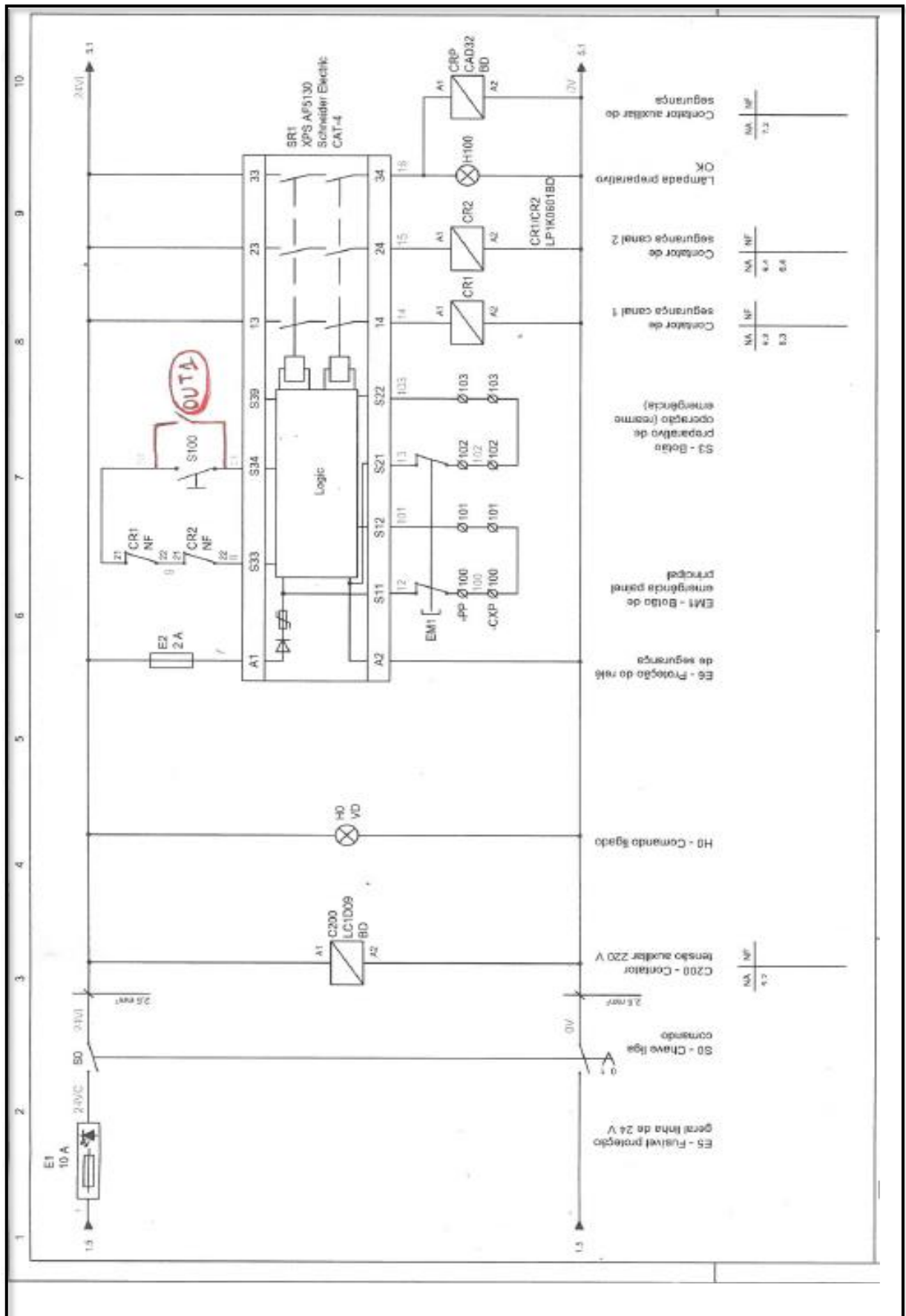
APÊNDICE A – CÓDIGO COMPLETO DA PRIMEIRA TELA

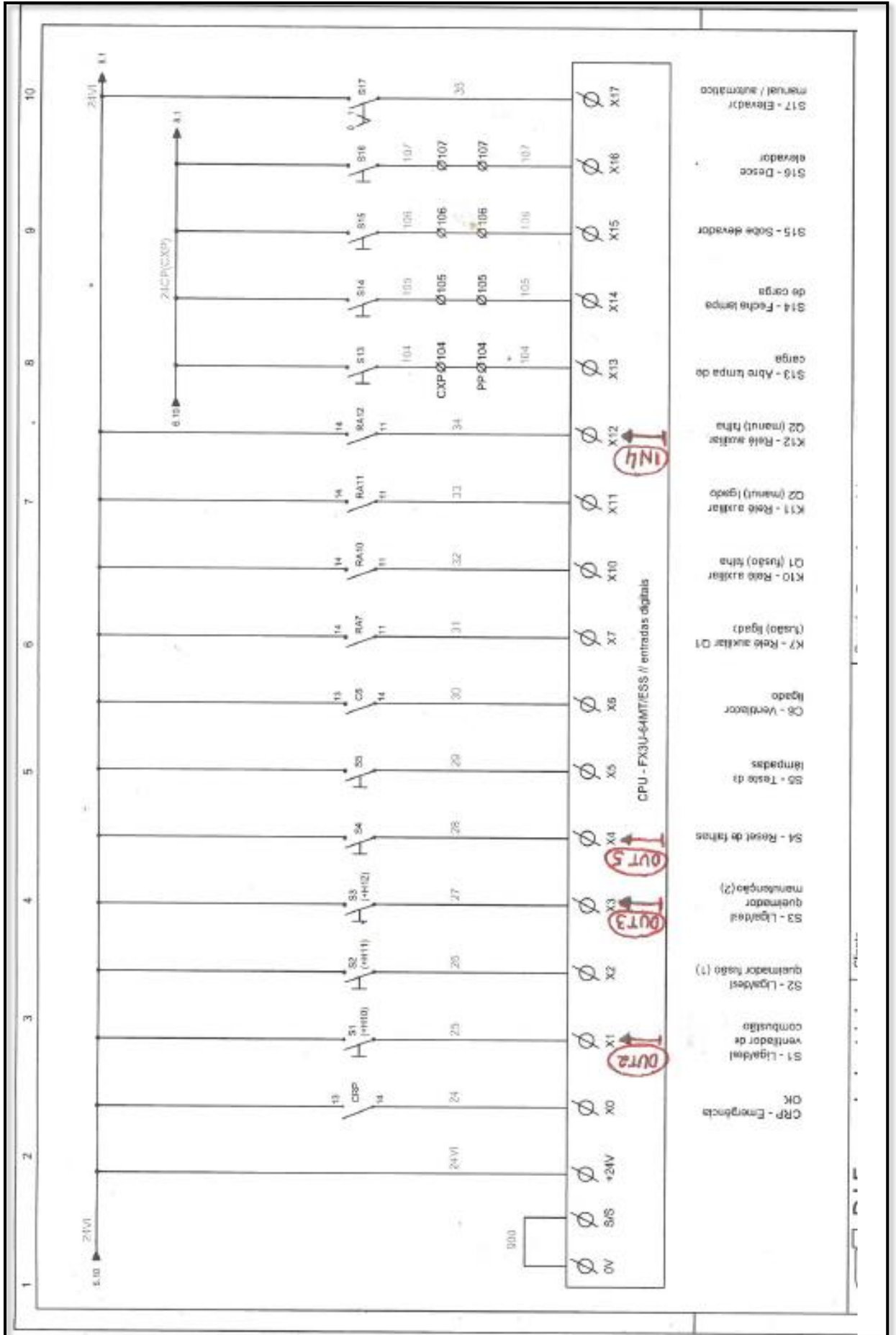


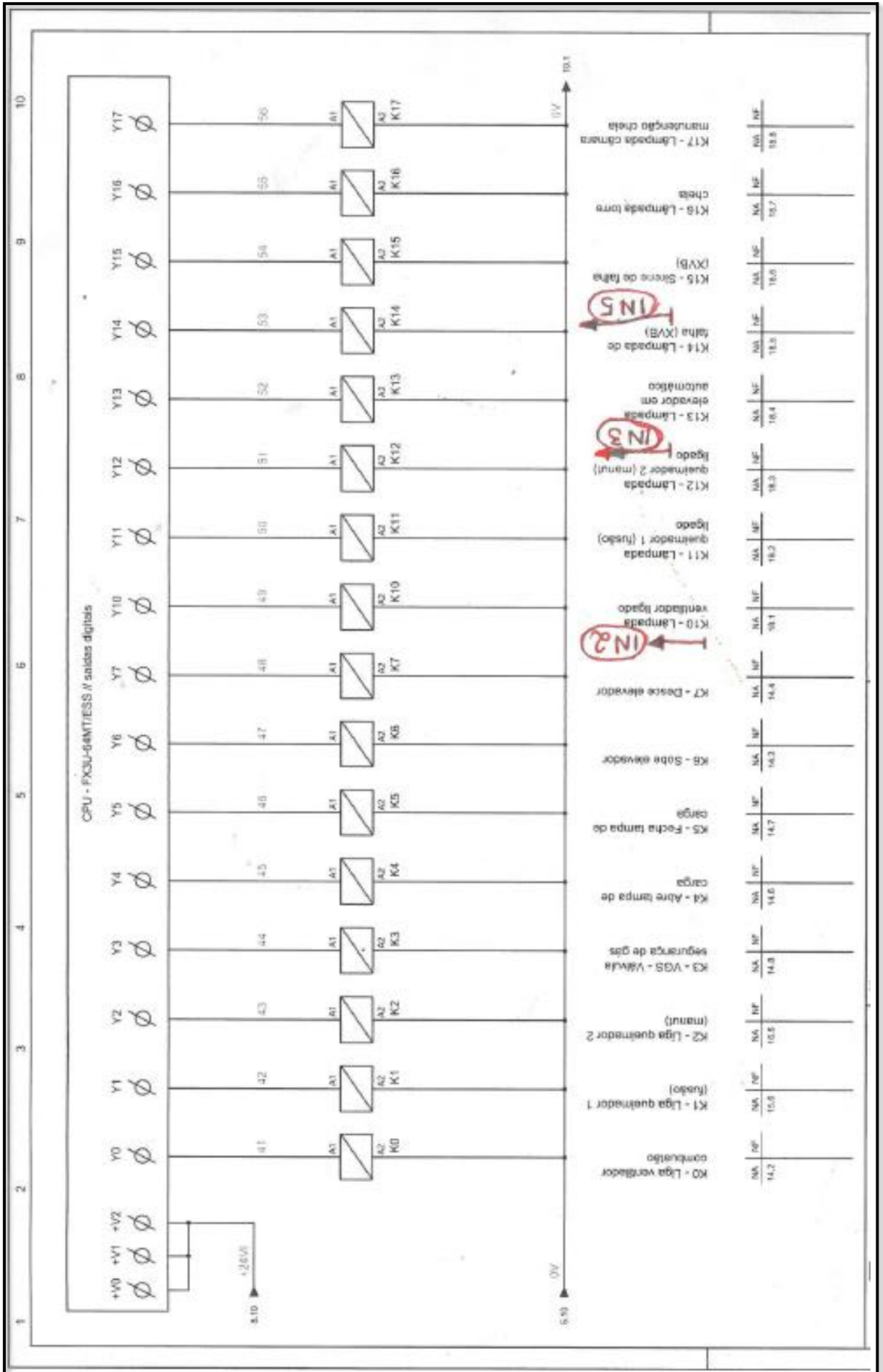
APÊNDICE B - CÓDIGO COMPLETO DA SEGUNDA TELA

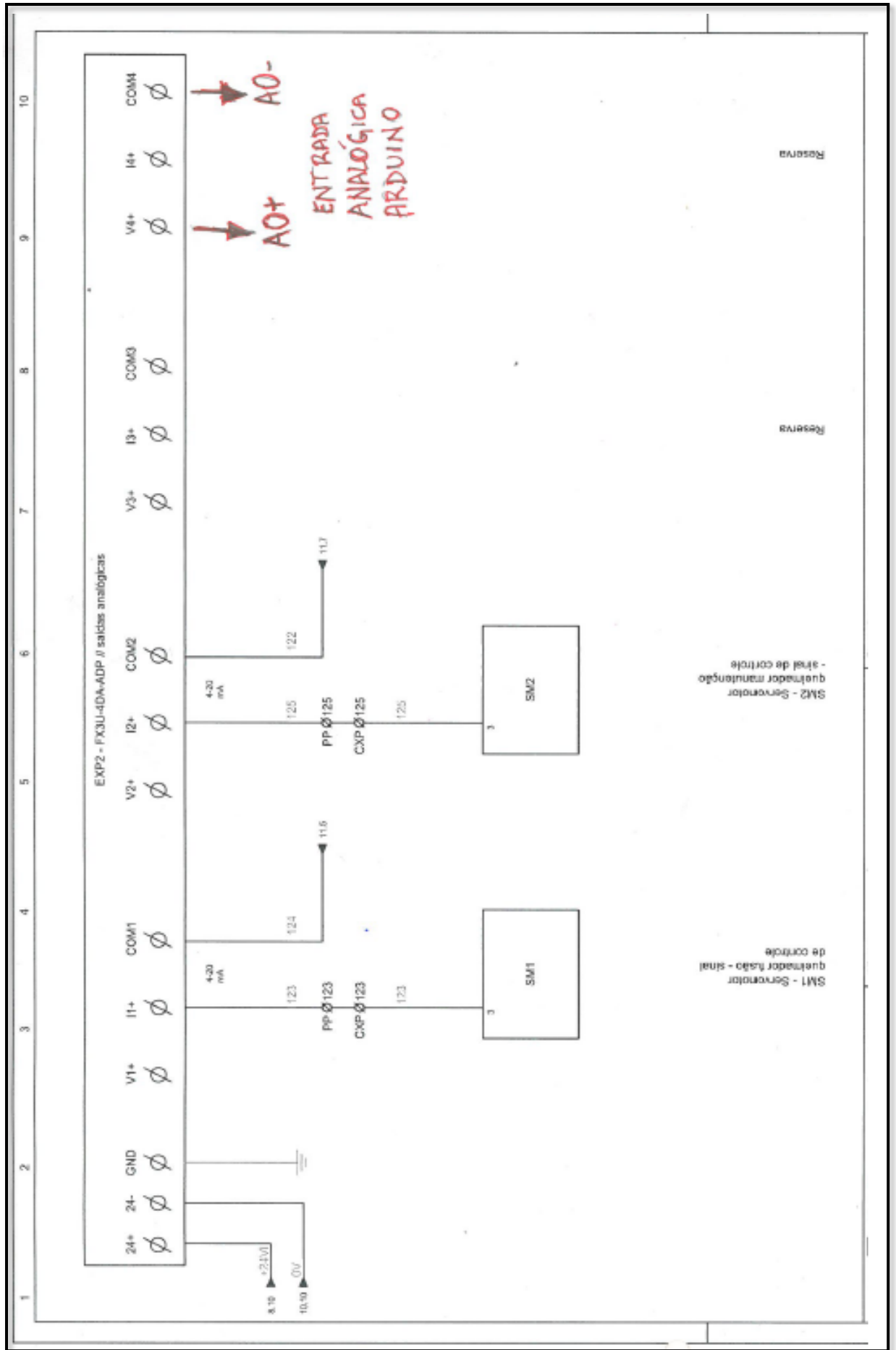


APÊNDICE C – ESQUEMA ELÉTRICO DO FORNO COM A INTEGRAÇÃO DO SISTEMA DE MONITORAMENTO

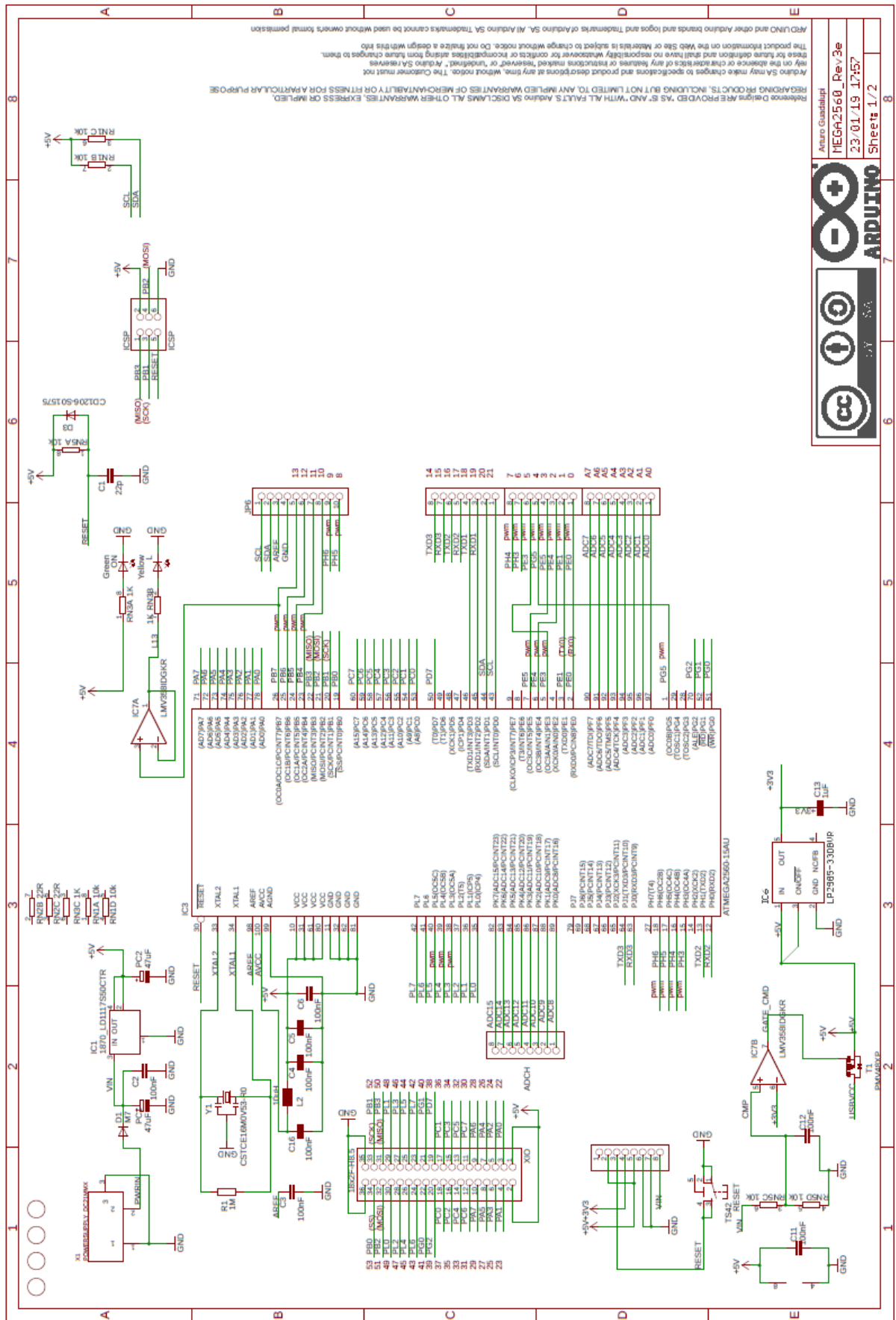








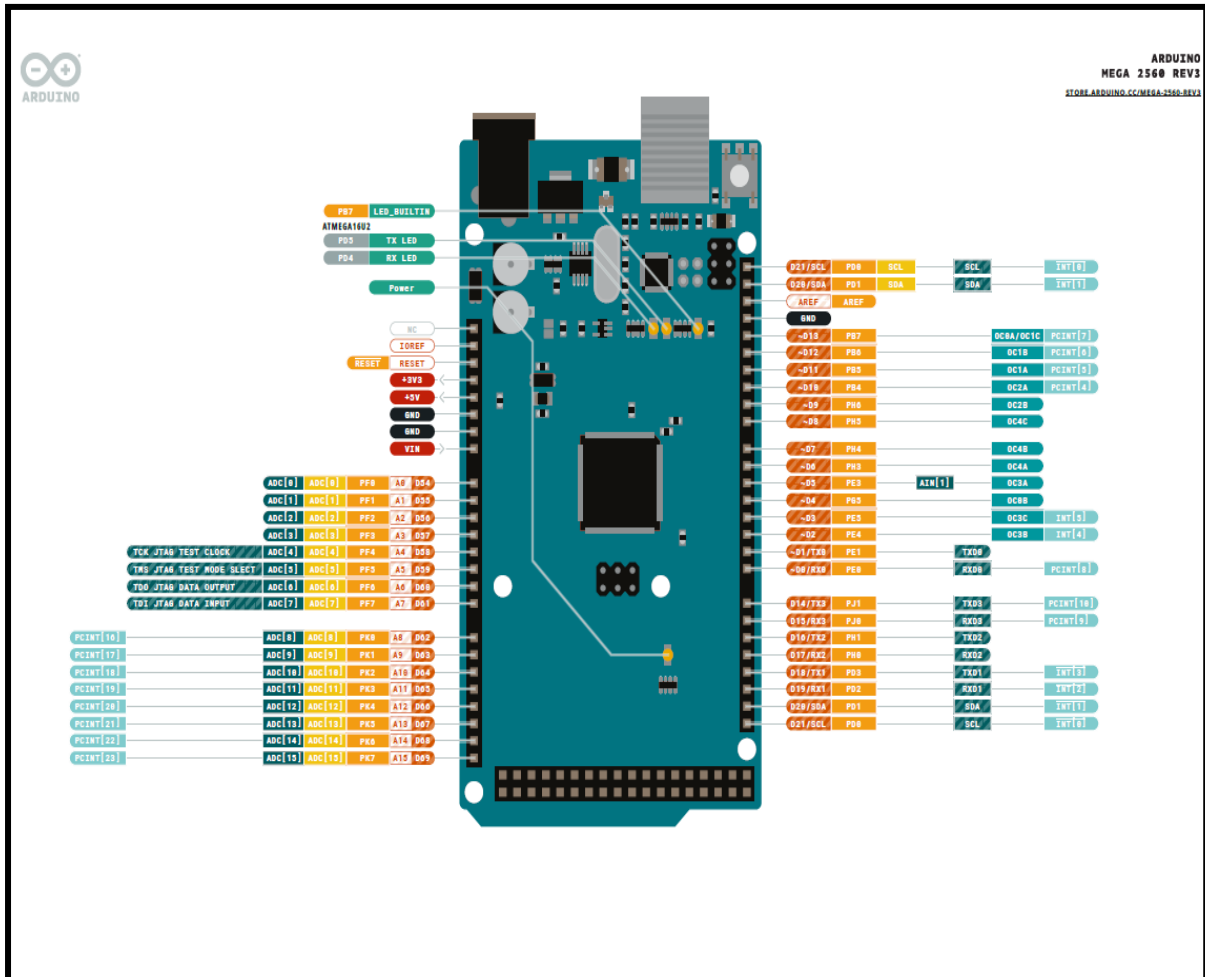
APÊNDICE D – ESQUEMA ELÉTRICO DO PROJETO ARDUINO MEGA 2560



Reference Designers are provided as is and with all faults. Arduino SA disclaims all other warranties, express or implied, regarding its products, including but not limited to, any implied warranties of merchantability or fitness for a particular purpose. The product information on the Web Site or Materials is subject to change without notice. Do not make a design without this file. Arduino SA may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or character of any features or instructions marked "reserved" or "undefined". Arduino SA reserves the right to change the definition and shall have no responsibility whatsoever for conflicts or inconsistencies arising from future changes to them. ATmega and other Arduino brands and logos and Trademarks of Arduino SA. All Arduino SA Trademarks cannot be used without owners formal permission.

Arturo Guadalupe
 MEGA2560_Rev3e
 23/01/19 17:57
 Sheet 1/2

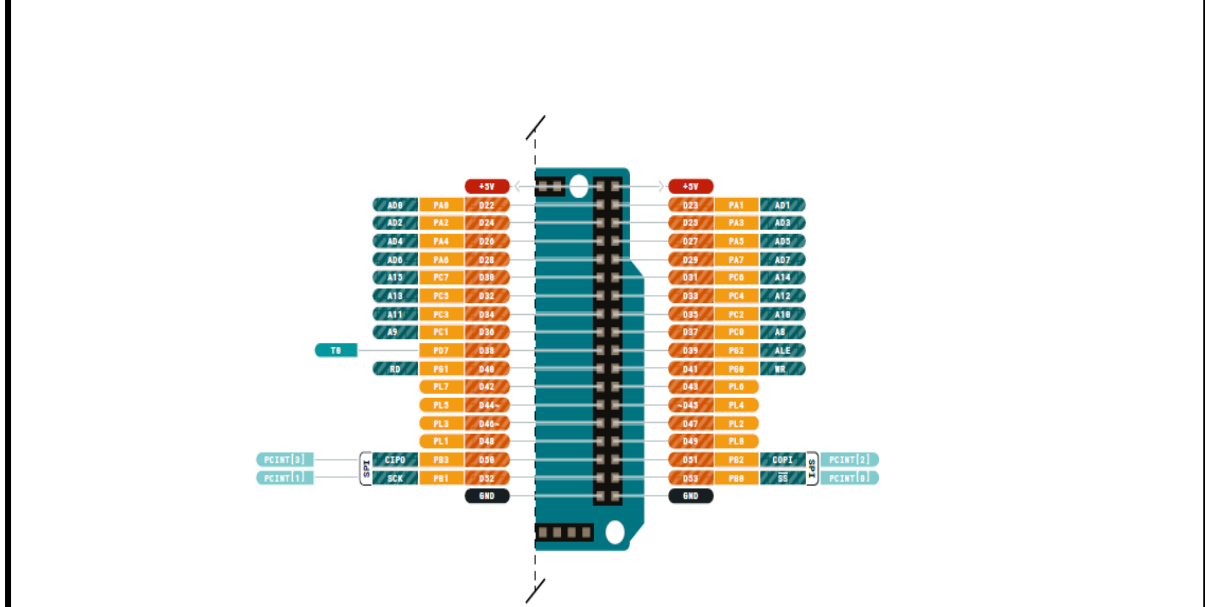
APÊNDICE E – DIAGRAMA DE PINAGEM DO ARDUINO MEGA 2560



ARDUINO MEGA 2560 REV3
ST088.ARDUINO.CC/MEGA-2560-REV3

ARDUINO

Digital pins D22-D53



APÊNDICE F – DIAGRAMA EM BLOCOS DO SISTEMA

