

THIAGO FERNANDES OLIVEIRA DE LIMA

**Desenvolvimento de sistema de baixo custo para monitoramento e controle de processos
de usinagem usando Raspberry Pi®**

Thiago Fernandes Oliveira de Lima

Desenvolvimento de sistema de baixo custo para monitoramento e controle de processos de usinagem usando Raspberry Pi®

Dissertação apresentada à Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, para a obtenção do título de Mestre em Engenharia Mecânica na área de Processamento e Caracterização de Materiais Avançados.

Orientador: Prof. Dr. Manoel Cleber Sampaio Alves

Coorientador: Prof. Dr. Marcos Tadeu Tibúrcio Gonçalves

Coorientador: Prof. Dr. César Augusto Galvão de Moraes

L732d Lima, Thiago Fernandes Oliveira de
Desenvolvimento de sistema de baixo custo para monitoramento e controle de processos de usinagem usando Raspberry Pi® / Thiago Fernandes Oliveira de Lima – Guaratinguetá, 2021.
133 f : il.
Bibliografia: f. 120-124

Dissertação (Mestrado) – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2021.
Orientador: Prof. Dr. Manoel Cleber Sampaio Alves
Coorientadores: Prof. Dr. Marcos Tadeu Tibúrcio Gonçalves ; Prof. Dr. César Augusto Galvão de Moraes

1. Usinagem. 2. Raspberry Pi (Computador). 3. Emissão acústica.
I. Título.

CDU 621.9(043)

THIAGO FERNANDES OLIVEIRA DE LIMA

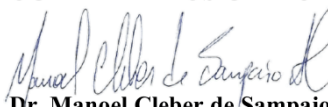
ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE

“MESTRE EM ENGENHARIA MECÂNICA”

PROGRAMA: ENGENHARIA MECÂNICA

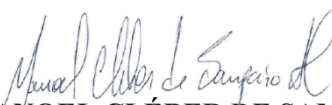
CURSO: MESTRADO

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO



Prof. Dr. Manoel Cleber de Sampaio Alves
Coordenador

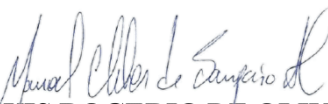
BANCA EXAMINADORA:



Prof.^a. Dr.^a. MANOEL CLÉBER DE SAMPAIO ALVES

Orientador - UNESP

participou por videoconferência



Prof. Dr. LUIS ROGERIO DE OLIVEIRA HEIN

UNESP

participou por videoconferência



Prof. Dr. CLEVERSON PINHEIRO

Instituto Federal de São Paulo

participou por videoconferência

DADOS CURRICULARES

THIAGO FERNANDES OLIVEIRA DE LIMA

NASCIMENTO	05.04.1985 – Sorocaba/SP
FILIAÇÃO	João Francisco Santos de Lima Vera de Oliveira Santos de Lima
2004/2010	Curso de Graduação Engenharia Industrial Madeireira – Universidade Estadual Paulista “Júlio de Mesquita Filho”
2015/2017	Curso de Graduação Análise e Desenvolvimento de Sistemas – Centro Universitário Internacional
2018/2019	Licenciatura R2 Física – Faculdade Facece
2018	Curso de Pós-Graduação Especialização em Banco de Dados – Claretiano Centro Universitário

A Vera, João e Rose.

AGRADECIMENTOS

Em primeiro lugar agradeço à Deus por ter trilhado esse caminho colocando pessoas que tanto me ajudaram a desenvolver este trabalho, dando força e coragem para sempre seguir em frente,

aos meus pais *Vera* e *João* e ao meu irmão *Douglas*, por sempre terem me apoiado e incentivado a desenvolver o trabalho,

à minha noiva *Roseli* por sempre ter me apoiado e suportado nos momentos mais difíceis, me motivando a sempre estar focado no trabalho,

ao meu orientador *Prof. Dr. Manoel Cleber Sampaio Alves* por ter acreditado que eu era capaz de desenvolver o trabalho por ele proposto, sempre mostrando os caminhos e incentivando a sempre estar aprimorando o projeto,

aos meus coorientadores *Prof. Dr. Marcos Tadeu Tibúrcio Gonçalves* e *Prof. Dr. César Augusto Galvão de Moraes* por sempre estarem me auxiliando no trabalho,

ao *Prof. Dr. Fabio Minoru Yamaji* que me incentivou a iniciar os estudos na pós-graduação,

ao *Alexandre Paiva Gaspar*, grande amigo que sempre me incentiva a aperfeiçoar academicamente e profissionalmente,

à *Etec Dr. Demétrio Azevedo Júnior* por disponibilizar o Laboratório de Metalurgia para realizar os ensaios, mesmo em tempos de pandemia,

ao *Prof. Cristian Rossi Ribeiro* por ter me auxiliado nos ensaios, sempre prestativo, dedicado e assertivo,

à *Câmara Municipal de Itapeva* por permitir que eu cursasse as disciplinas e realizasse os ensaios do projeto flexibilizando o meu horário de trabalho na instituição,

à todos os meus amigos que estiveram ao meu lado durante o desenvolvimento do trabalho, em especial ao *Luís Ricardo* e o *Alfredo Adolpho* que me encorajaram a iniciar e continuar o projeto,

Obrigado!

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

“O ótimo é inimigo do bom”

François-Marie Arouet

RESUMO

O desenvolvimento da automação industrial teve como principal motivação a redução do esforço físico para a realização de tarefas, entretanto, os custos envolvidos para a aquisição e manutenção de tais equipamentos são altos. A área acadêmica também é afetada, pois para uma instituição arcar com os custos de sensores, atuadores e programas de computadores proprietários para o ensino da automação industrial está cada vez maior. Buscou-se com este trabalho o desenvolvimento de uma plataforma de aquisição de dados de baixo custo para usinagem de materiais, com a utilização do Raspberry Pi para processamento e monitoramento dos dados adquiridos e sensores com baixo valor de aquisição, sendo esses dados de vibração e de captura da emissão acústica, utilizando para validação do protótipo ensaios de usinagem de uma peça de alumínio. Construiu-se esta plataforma de aquisição com a utilização do Raspberry Pi 3 Model B, de um conversor analógico digital, sensores de vibração e para captação de emissão acústica, jumpers e matriz de contatos para acomodação dos materiais. Além disso, desenvolveu-se no Raspberry Pi três aplicações na linguagem Python, sendo um para aquisição dos dados que são utilizados na aplicação para exibição local de gráficos e uma terceira aplicação para aquisição de dados utilizado no monitoramento remoto dos dados, ou seja, os dados sendo adquiridos no Raspberry Pi e mostrados remotamente em outro microcomputador dentro da mesma rede. Realizou-se 27 (vinte e sete ensaios), torneando uma peça de alumínio, variando-se entre os ensaios a profundidade de corte, o avanço e a posição dos equipamentos para a aquisição. Os dados adquiridos foram exibidos na tela de acordo com o fenômeno físico que estava sendo observado. Pelas restrições aplicadas dado o estado de calamidade pública devido ao agravamento da doença do coronavírus 2019, não foi possível calibrar os sensores e validar os dados adquiridos. O sistema proposto pode receber mais sensores, pois a produção do protótipo contempla essa possibilidade, sendo necessário apenas pequenas alterações nas aplicações. Diante dos resultados apresentados, a plataforma de baixo custo desenvolvida pode ser uma solução a ser considerada para a aquisição de dados de usinagem de materiais, pelo menos em nível acadêmico, dada a sua flexibilidade, adaptabilidade e baixo custo de aquisição de novos sensores, em comparação com os sistemas comercialmente estabelecidos. O custo total dos materiais utilizados para a plataforma de aquisição de dados ficou em torno de \$85 (oitenta e cinco dólares americanos).

PALAVRAS-CHAVE: Usinagem de materiais. Aquisição de dados. Monitoramento. Emissão Acústica. Raspberry Pi.

ABSTRACT

The development of industrial automation had as its main motivation the reduction of physical effort to perform tasks, however, the costs involved for the acquisition and maintenance of such equipment are high. The academic area is also affected, because for an institution to bear the costs of sensors, actuators and proprietary computer programs for teaching industrial automation is increasing. The aim of this work was to develop a low-cost data acquisition platform for machining materials, using the Raspberry Pi for processing and monitoring the acquired data and sensors with low acquisition value, these vibration data and to capture the acoustic emission, using machining tests of an aluminum part to validate the prototype. This acquisition platform was built using the Raspberry Pi 3 Model B, an analog to digital converter, vibration sensors and sensors for capturing acoustic emission, jumpers and breadboard to accommodate the materials. In addition, three applications were developed in the Raspberry Pi in the Python language, one for data acquisition that are used in the application for local display of graphs and a third application for data acquisition used in the remote monitoring of the data, that is, the data being acquired on the Raspberry Pi and displayed remotely on another microcomputer within the same network. 27 (twenty-seven tests) were carried out, turning an aluminum part, varying the depth of cut, the feed and the position of the equipment for the acquisition between the tests. The acquired data was displayed on the screen according to the physical phenomenon being observed. Due to the restrictions applied given the state of public calamity due to the worsening of the coronavirus disease 2019, it was not possible to calibrate the sensors and validate the acquired data. The proposed system can receive more sensors, as the production of the prototype contemplates this possibility, requiring only small changes in the applications. In view of the results presented, the low-cost platform developed can be a solution to be considered for the acquisition of material machining data, at least at an academic level, given its flexibility, adaptability and low cost of acquiring new sensors, in compared to commercially established systems. The total cost of materials used for the data acquisition platform was around \$85 (eighty-five US dollars).

KEYWORDS: Machining Materials. Data acquisition. Monitoring. Raspberry Pi. Acoustic Emission.

LISTA DE FIGURAS

Figura 1 - Três paradigmas relativamente distintos da usinagem.	22
Figura 2 - Direção dos movimentos de corte, de avanço e efetivo no torneamento (a) e no fresamento discordante (b). Ângulo da direção de avanço ϕ , da direção efetiva η e do plano de trabalho.	23
Figura 3 - Percursos de corte l_c , de avanço l_f e efetivo l_e no fresamento discordante.	24
Figura 4 - Seção transversal vertical no torneamento e detalhe da geometria da pastilha de corte no canto inferior direito. A estrutura para dinamômetro e o termopar são utilizadas apenas em laboratório, para medição das forças de usinagem e temperatura em geral.	25
Figura 5 - Principais componentes e movimentos de um torno.	26
Figura 6 - Cunha de corte da ferramenta de torneamento.	28
Figura 7 - Dureza e tenacidade de materiais para ferramentas de corte.	29
Figura 8 - Variação do acabamento superficial com avanço de 135m/min.	31
Figura 9 - Fontes de emissão acústica em diferentes estágios de remoção de material.	33
Figura 10 - Fontes de emissão acústica no torneamento.	34
Figura 11 – Estrutura comum de um microcontrolador.	37
Figura 12 - Simplificação da organização de processadores ARM.	38
Figura 13 - Estrutura física e lógica de um computador.	40
Figura 14 - Estrutura interna de um Controlador Lógico Programável.	44
Figura 15 - Termopar, módulo de entrada analógico que possui um conversor de sinais analógicos para digitais para todas as entradas, barramento para transferência de dados e um controlador lógico programável.	46
Figura 16 - Elementos internos de um sensor indutivo.	47
Figura 17 - Elementos internos de um sensor capacitivo.	48
Figura 18 - Alguns dos atuadores utilizados em automação industrial: (a) servo motor, (b) motor de corrente contínua, (c) motor de passo, (d) motor linear, (e) cilindro pneumático e (f) solenóide.	50
Figura 19 - Diagrama funcional de um sistema de aquisição de dados baseado em microcomputador.	51
Figura 20 - Os dois tipos básicos de amplificadores são os com ou sem inversão de saída. ...	53
Figura 21 - Placa Arduino Uno.	55
Figura 22 - Vista superior do Raspberry Pi 3 Model B.	58
Figura 23 - Representação gráfica das portas do GPIO do Raspberry Pi, para os modelos 3B+,	

3B, 2B, Zero, A+ e B+).	59
Figura 24 - Imagem ilustrativa da protoboard e de fios jumper.	60
Figura 25 - Setup experimental montado para aquisição da temperatura de corte.	62
Figura 26 - Modelo esquemático do monitoramento das forças lidas pelas células de carga.	63
Figura 27 - Sistema de aquisição de dados de baixo custo, com Raspberry Pi, uma placa personalizada logo acima e acoplada a ela os sensores e o display LCD. Logo abaixo, uma régua para demonstrar o desenho compacto da solução.	64
Figura 28 – Sensor para captação da emissão acústica – MAX9814.	66
Figura 29 – Sensor GY-521 – Acelerômetro e Giroscópio.	67
Figura 30 – Conversor de sinais analógicos para digitais de 8 canais com resolução de 10 bits - MCP3008.	67
Figura 31 – Esquema de ligação do conversor MCP 3008 com o Raspberry Pi 3 Model B....	68
Figura 32 – Vista superior do Raspberry Pi 3 Model B na sua configuração, com cabo de energia, monitor, mouse, teclado e cabo de rede conectados à placa.	69
Figura 33 – Prototipação da plataforma de aquisição de dados de baixo custo.	71
Figura 34 - Esquema de conexões do conversor de sinais analógicos para digitais e dos sensores com o Raspberry Pi.	72
Figura 35 - Tela da aplicação para monitoramento local da aquisição de dados.	74
Figura 36 – Esquema do monitoramento remoto dos dados lidos pelo Raspberry Pi.	75
Figura 37 - Tela da aplicação para monitoramento remoto da aquisição de dados.	75
Figura 38 - Bancada criada no Estudo A para acomodar o sistema de aquisição de dados da operação de lixamento.	77
Figura 39 - Torno convencional fabricado pela Romi, modelo Imor Oficina 420.	78
Figura 40 - Bancada criada no Estudo B para acomodar o sistema de aquisição de dados da operação de torneamento, com os sensores próximos à interface ferramenta-peça e a plataforma no carro.	79
Figura 41 - Esquema básico de um acelerômetro.	80
Figura 42 - Configuração experimental apresentando a ferramenta e o acelerômetro.	81
Figura 43 - Efeito da velocidade, avanço e espessura de corte na vibração e temperatura.	82
Figura 44 - Posicionamento do acelerômetro/giroscópio (a) afastado da ferramenta. (b) próximo à ferramenta. Fotos tiradas antes dos ensaios.	83
Figura 45 - Fragmento de código da aplicação para monitoramento local da aquisição, responsável por calcular a rotação em Y do sensor GY-521.	83
Figura 46 - (a) Geometria da ferramenta nova. (b) Geometria da ferramenta desgastada. (c)	

Locais de fixação. (d) Excitações (F1: Geração S0; F2: Força de corte; F3: Atrito com a peça; F4: Atrito com o cavaco).....	84
Figura 47 – Configuração montada para aquisição de dados de emissão acústica.....	86
Figura 48 - Posicionamento do sensor para captação de emissão acústica (a) afastado da ferramenta. (b) próximo à ferramenta.....	87
Figura 49 – Função responsável pela aquisição em volts dos dados do sensor MAX9814.	87
Figura 50 – Em destaque na cor amarela as conexões entre os jumpers.	88
Figura 51 – Capacitor eletrolítico utilizado para recuperar temporariamente as portas USB do Raspberry Pi.	89
Figura 52 - (a) Ligação dos resistores <i>pull-up</i> . (b) Leitura dos canais do conversor de sinais analógicos para digitais através da aplicação desenvolvida para teste de leitura dos valores de entrada desses canais.	91
Figura 53 - Gráficos gerados na aplicação, com o lixamento da madeira (a) e do alumínio (b).	92
Figura 54 - Consumo de recursos computacionais no Raspberry Pi ao executar a aplicação designada ao monitoramento local da aquisição de dados da plataforma.	93
Figura 55 - (a) Consumo de memória RAM (KB), processamento (%) e rede (B/s) do microcomputador cliente para exibição dos gráficos. (b) Consumo de processamento (%) e memória RAM (%) do Raspberry Pi para a aquisição dos dados.	94
Figura 56 - (a) Raspberry Pi e protoboard soltas. (b) Todos os componentes fixados no carro.	96
Figura 57 - Microcomputador utilizado para o monitoramento em tempo real da aquisição de dados e para operação remota do Raspberry Pi via VNC.....	97
Figura 58 - Gráficos de linha gerados com os pontos adquiridos do sensor de captação de emissão acústica. (a) Dados gerados a partir dos gráficos da página HTML, no formato CSV. (b) Dados gerados a partir dos arquivos brutos, no formato TXT.....	99
Figura 59 – Gráficos de linha gerados com os pontos adquiridos do sensor de vibração (a) Dados gerados a partir dos gráficos da página HTML, no formato CSV. (b) Dados gerados a partir dos arquivos brutos, no formato TXT.	100
Figura 60 - Indicação de áreas com maior variação da vibração na usinagem.	101
Figura 61 – Gráfico da vibração em função da emissão sonora da usinagem.....	102
Figura 62 - (a) Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 01 e 02. (b) Diagramas comparando os valores obtidos da vibração (°) nas Análises 01 e 02.	103

Figura 63 - (a) Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 03 e 04. (b) Diagramas comparando os valores obtidos da vibração (°) nas Análises 03 e 04.	104
Figura 64 - Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 05, 06, 07 e 08.	105
Figura 65 - Diagramas comparando os valores obtidos da vibração (°) nas Análises 05, 06, 07 e 08.	106
Figura 66 – Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 09 e 10.	107
Figura 67 – Diagramas comparando os valores obtidos da vibração (°) nas Análises 09 e 10.	108
Figura 68 - Teste de Levene da emissão acústica entre as análises.....	109
Figura 69 - Teste de Kolmogorov-Smirnov nas amostras das análises da emissão acústica.	109
Figura 70 - Comparação de médias das análises de emissão acustica utilizando o teste de Tukey.	111
Figura 71 – Diagramas comparando os valores obtidos da emissão acústica (dB) das Análises 01 até 08.	112
Figura 72 - Teste de Levene da vibração entre as análises.....	113
Figura 73 - Teste de Kolmogorov-Smirnov nas amostras das análises da vibração.	113
Figura 74 - Comparação de médias das análises da vibração utilizando o teste de Tukey. ...	114
Figura 75 - Diagramas comparando os valores obtidos da vibração (°) das Análises 01 até 08.	115
Figura 76 - (a) Representação gráfica do teste de Tukey aplicado nas análises da emissão acústica. (b) Representação gráfica do teste de Tukey aplicado nas análises da vibração.	116

LISTA DE QUADROS E TABELAS

Quadro 1 - Diferenças entre os computadores de placa única e os computadores comuns.	57
Quadro 2 - Posição dos sensores, avanço e profundidade de corte em cada ensaio do Estudo B.	78
Quadro 3 - Roteiro para análise estatística dos dados adquiridos no Estudo B.....	103
Tabela 1 - Componentes utilizados no protótipo e seus respectivos valores em dólares americanos.	65
Tabela 2 - Levantamento estatístico comparando as Análises 04 e 08.	112

SUMÁRIO

1	INTRODUÇÃO	18
2	REVISÃO BIBLIOGRÁFICA.....	22
2.1	USINAGEM.....	22
2.2	TORNEAMENTO	24
2.3	FERRAMENTAS DE CORTE	27
2.3.1	Materiais para ferramentas de corte	28
2.4	VIBRAÇÃO NO PROCESSO DE USINAGEM	29
2.5	EMIÇÃO ACÚSTICA NA USINAGEM	31
2.6	INFORMÁTICA	35
2.6.1	Estrutura básica de um computador	35
2.6.2	Estrutura básica de um microcontrolador.....	37
2.6.3	Arquitetura ARM.....	38
2.6.4	Sistema Operacional e programas de computadores.....	39
2.7	AUTOMAÇÃO.....	40
2.7.1	Analógico e digital	41
2.7.2	Controladores	42
2.7.2.1	Computadores, Sistema de controle distribuído (DCS) e sistemas embarcados....	42
2.7.2.2	Controladores Lógicos Programáveis (CLP)	43
2.7.2.2.1	<i>Módulos de entrada e saída digitais do CLP</i>	44
2.7.2.2.2	<i>Módulos de entrada e saída analógicos do CLP</i>	44
2.7.3	Sensores	46
2.7.3.1	Sensores Analógicos	46
2.7.3.2	Sensores Digitais	48
2.7.4	Atuadores	49
2.8	AQUISIÇÃO DE DADOS.....	50
2.8.1	Condicionamento de sinais	52
2.9	PLATAFORMAS DE COMPUTAÇÃO EMBARCADAS	54
2.9.1	Arduino	54
2.9.2	Computador de placa única (SBC)	56
2.9.3	Raspberry Pi	57
2.9.4	Aquisição de dados com plataformas de computação embarcadas.....	60
3	MATERIAL E MÉTODOS.....	65

3.1	PLATAFORMA DE AQUISIÇÃO DE DADOS	65
3.1.1	Sensores de baixo custo	65
3.1.1.1	Sensor para captação da emissão acústica.....	66
3.1.1.2	Sensor de vibração	66
3.1.2	Conversor Analógico-Digital (ADC) de 8 canais	67
3.1.3	Computador de placa única	68
3.1.3.1	Instalação e configuração inicial do Raspberry Pi	69
3.1.4	Conexão física do conversor de sinais analógicos para digitais e dos sensores no Raspberry Pi	71
3.1.5	Programação dos módulos no Raspberry Pi	72
3.2	ENSAIOS ENVOLENDOS USINAGEM	76
3.3	ESTUDO A	76
3.4	ESTUDO B	77
3.4.1	Métodos para medição da vibração na usinagem	80
3.4.1.1	Acelerômetros	80
3.4.1.2	Modelo empírico para medição da vibração	81
3.4.2	Medição da emissão acústica e sonora na usinagem	83
4	RESULTADOS E DISCUSSÃO	88
4.1	PLATAFORMA DE AQUISIÇÃO DE DADOS DE BAIXO CUSTO	88
4.1.1	Leitura dos dados dos sensores	90
4.2	ESTUDO A	91
4.2.1	Monitoramento local da aquisição	92
4.2.2	Monitoramento remoto da aquisição	93
4.3	ESTUDO B	95
4.3.1	Aquisição dos dados	96
4.3.2	Análise estatística	102
4.3.2.1	Comparação entre a Análise 01 e a Análise 02	103
4.3.2.2	Comparação entre a Análise 03 e a Análise 04	104
4.3.2.3	Comparação entre as Análises 05, 06, 07 e 08	105
4.3.2.4	Comparação entre a Análise 09 e Análise 10	107
4.3.2.5	Comparação entre as Análises 01, 02, 03, 04, 05, 06, 07 e 08	109
4.3.2.5.1	Emissão acústica	109
4.3.2.5.2	Vibração	113
5	CONCLUSÃO	118

REFERÊNCIAS	120
APÊNDICE A – CÓDIGO-FONTE DOS PROGRAMAS DESENVOLVIDOS PARA AQUISIÇÃO E MONITORAMENTO NA LINGUAGEM PYTHON.....	125
APÊNDICE B – CÓDIGO-FONTE DO PROGRAMA PARA MONITORAMENTO DESENVOLVIDO NA LINGUAGEM HTML	132

1 INTRODUÇÃO

A evolução da automação industrial tem como principal objetivo a redução do esforço físico para a realização de tarefas. O desenvolvimento da automação é parte importante da evolução industrial, pois com a automação as máquinas são mais rápidas e precisas, elevando-se a produtividade com menor índice de erros e desperdício de material (ROGGIA, FUENTES, 2016). Com a evolução da automação conseguiu-se aumentar a produtividade para suprir a demanda por produtos, principalmente da área de tecnologia, haja vista a quantidade de pessoas que possuem dispositivos eletrônicos como: smartphones, microcomputadores, assistentes pessoais (Amazon[®] Echo, Google Dot) e aparelhos televisores (KEMP, 2019).

Entretanto, o avanço da tecnologia da informação foi de suma importância na evolução da automação, pois os sensores, atuadores, interfaces humano-máquina e outros componentes se utilizam de elementos da tecnologia da informação, como microcontroladores, interfaces de entrada e saída (I/O), microprocessadores e módulos de memória. Basicamente, segundo Stallings (2016), o computador é um sistema complexo que contém milhares de componentes eletrônicos e no tocante à estrutura, tem-se o seguinte nos computadores de um processador com um núcleo, conhecida como Estrutura Top-Level: CPU (Unidade Central de Processamento), Memória principal, Entrada e Saída (E/S) e Sistema de interconexão (barramento).

Com o aumento do poder de processamento e a consequente redução de tamanho dos componentes, iniciou-se a expansão dos microcontroladores. Os controladores são dispositivos que realizam cálculos, processamento e gerenciamento de E/S, sendo parte de um sistema de automação. São utilizados principalmente nos Controladores Lógicos Programáveis (CLP), que controlam a automação na linha de produção na indústria. Os controladores lógicos programáveis são utilizados para receber múltiplas entrada e saída, através de sensores e atuadores. De acordo com a entrada recebida por um ou mais sensores, isso depende da programação do controlador lógico programável, ele pode acionar um ou mais atuadores em tempo real, controlando diversos aspectos da produção e auxiliando na segurança.

No início do século XXI, com o avanço da tecnologia e barateamento de componentes eletrônicos ocorreu o desenvolvimento de algumas ferramentas open source (código aberto) baseadas em microcontroladores para utilização a prototipação da automação e, em alguns casos, é utilizado em produção. Uma dessas ferramentas foi o Arduino[®], que na sua versão mais popular, a Uno, permite a 13 (treze) entradas digitais e 6 (seis) entradas analógicas. A linguagem de programação utilizada é a C, datada da década de 1970, extremamente poderosa e de fácil

aprendizado. O Arduino vem sendo utilizado desde o final da década de 2000 como instrumento de ensino para a eletrônica e programação de computadores nas escolas (BROCK *et al.*, 2009). Porém, dada a difícil situação econômica de muitos países e, visando a inclusão digital do maior número de pessoas possíveis, um protótipo de \$35 (trinta e cinco dólares americanos) foi apresentado em maio de 2011. Este equipamento, com nome de Raspberry Pi, possui dimensões pouco maiores que um cartão bancário e conta com uma arquitetura semelhante à de um microcomputador comum, tornando possível sua utilização como um computador pessoal, pois conta com entradas USB (universal serial bus), saída HDMI para exibição de imagem, processador arquitetura ARM, memória RAM e memória para armazenamento de dados utilizando um cartão *microSD*. Porém, o que surgiu como alternativa de baixo custo para promover o maior alcance da tecnologia em locais economicamente subdesenvolvidos, logo se tornou um fenômeno na área de tecnologia, dada seu poder de processamento e flexibilidade para o desenvolvimento de projetos na área eletrônica e em inteligência artificial (NORRIS, 2017).

Com todo este contexto, busca-se o desenvolvimento de plataformas para aquisição de dados e controle de baixo custo, pois os custos envolvidos em soluções proprietárias são consideravelmente elevados. Com a utilização de sensores que custam em torno de \$5 (cinco dólares americanos) e com razoável exatidão tem-se utilizado cada vez mais soluções open source (código aberto) voltadas à automação desenvolvidas tendo como base as placas de prototipação (Arduino, Raspberry Pi). Acrescenta-se a facilidade de conexão com a internet destes equipamentos, favorecendo o controle e o monitoramento remoto de sensores e atuadores. Ao final da década de 2010 existem diversas ferramentas open source para a automação, que possuem comunidades on-line, as quais contribuem para o desenvolvimento destas ferramentas, pois funcionam como fórum de suporte e treinamento para a programação destes equipamentos. Tem-se atualmente diversas opções de baixo custo e de confiabilidade atestada no mercado, algumas delas clones de produtos oficiais, que podem atender diversos projetos, de acordo com sua complexidade como, por exemplo, Arduino, Raspberry Pi® Model B, HummingBoard-i2 e BeagleBone Black, sendo que o Raspberry Pi® possui uma grande comunidade on-line que oferece amplo suporte para resolução de problemas, aumentando-se assim sua relevância perante as plataformas de menor custo do mercado (HABER *et al.*, 2015). Além das placas, os sensores e atuadores possuem bibliotecas gratuitas para desenvolvimento em diversas plataformas, o que favorece a expansão desses equipamentos, tornando-os ainda mais acessíveis para utilização em diversas soluções, como por exemplo, em processos de usinagem (LEWIS; CAMPBELL; STAVROULAKIS, 2016).

O processo de usinagem de materiais se caracteriza pelos movimentos entre a ferramenta e peça durante a usinagem. Ferraresi (1970) define que no torneamento obtém-se superfícies de revolução com o auxílio de ferramentas monocortantes, onde a peça gira em torno do eixo principal de rotação da máquina e a ferramenta se desloca simultaneamente segundo uma trajetória no mesmo plano com o eixo. Durante este processo de usinagem ocorrem diversos fenômenos, resultando em algumas grandezas físicas. Algumas dessas variáveis de saída no processo de torneamento são: temperatura, vibração, emissão acústica, força e potência. O estudo destas variáveis é importante para compreender o comportamento da ferramenta durante a usinagem, pois a interação entre a ferramenta e a peça possui uma influência direta nestas variáveis, tais como o consumo de energia e rugosidade (HECKER *et al.* 2007).

Ao final da década de 2010 enfrenta-se uma grave crise financeira e institucional no Brasil, onde está havendo redução no investimento nas áreas de Ciências e Tecnologia, o que vem prejudicando o desenvolvimento de pesquisas em diversos setores, dada a escassez de recursos destinados às instituições públicas de ensino (GADELHA, 2017). Dado esse cenário, vem-se buscando alternativas de baixo custo, mas que mantenham a qualidade dos experimentos e dos dados coletados. Pode-se citar a grande demanda de equipamentos eletrônicos para o ensino, principalmente em cursos técnicos de nível médio da área de eletrônica, eletrotécnica e informática. Conforme supracitado, as soluções proprietárias na área de automação demandam alto investimento para a sua aquisição e manutenção. Com isso, as soluções de baixo custo utilizando as ferramentas open source já citadas surgem como uma alternativa promissora, pois estas estão cada vez mais maduras e estáveis para utilização em versões de produção. Com os produtos atualmente disponíveis no mercado, é concebível o desenvolvimento de soluções de baixo custo para aquisição de dados na área de usinagem. Existem sensores de baixo custo capazes de captar algumas das variáveis de saída do torneamento: vibração, emissão acústica, força e potência. Tais soluções podem ser empregadas principalmente na área educacional, após a calibração e validação, tornando-se uma importante alternativa técnica para o monitoramento e controle de processos diante do atual cenário financeiro.

Como o objetivo principal pretende-se:

- Desenvolver um sistema de aquisição de dados (parte física e lógica) para monitoramento de processos de usinagem, utilizando plataforma de baixo custo Raspberry Pi®.

Como objetivos secundários tem-se:

- Desenvolver sistema de aquisição de dados para processos de usinagem para captação

de sinais digitais e analógicos para monitoramento de sinais de vibração, emissão acústica e sonora;

- Desenvolver a interface gráfica para o monitoramento do processo, para análise acurada e em tempo real dos dados adquiridos.

- Desenvolver a interface gráfica para o controle do processo, como anteparo as ações que o sistema deverá realizar quando identificar condições críticas na usinagem.

- Aplicar o sistema a ser desenvolvido no estudo da usinabilidade do alumínio com ferramenta com pastilha de metal duro, analisando vibração e emissão acústica e sonora.

- Analisar o custo do sistema desenvolvido comparado com sistemas comerciais existentes.

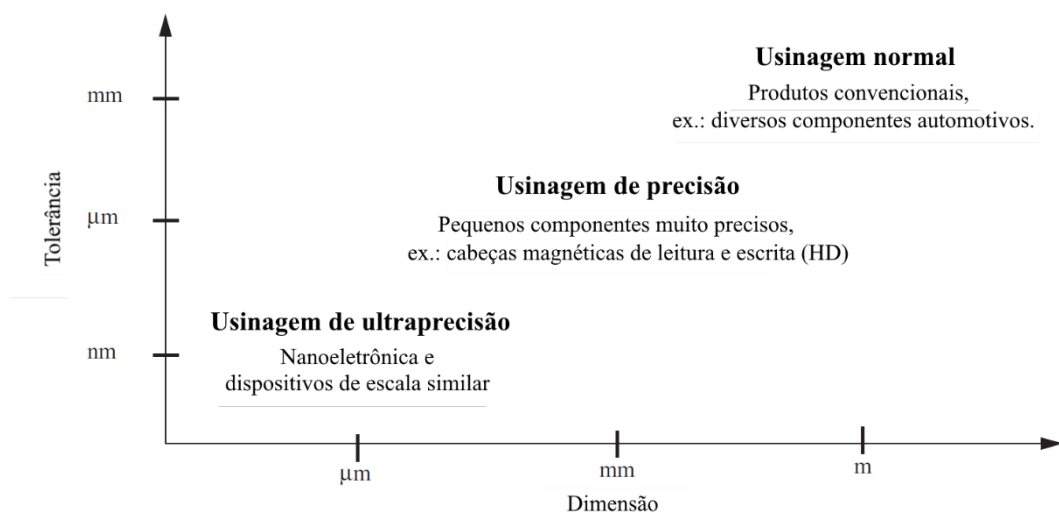
2 REVISÃO BIBLIOGRÁFICA

A automação na usinagem engloba diversos conceitos, desde o processo da usinagem até seu controle e monitoramento através das IHM (Interface Humano-Máquina). Para alcançar um bom desenvolvimento de um sistema de baixo custo para aquisição de dados da usinagem de materiais, é necessário melhor compreensão sobre os seguintes assuntos: Usinagem, Aquisição de Dados, Automação, Informática e Plataformas de prototipagem eletrônica.

2.1 USINAGEM

Trent e Wright (2000) entendem que não há uma definição precisa para a usinagem de metais, sendo basicamente o processo de remoção de cavaco através da cunha de corte de uma ferramenta. Na indústria, o termo usinagem é utilizado para definir operações onde há formação de cavacos. Entretanto, Ferraresi (1970) define que as operações de usinagem ao conferir à peça a forma, as dimensões, o acabamento, ou uma combinação destes três, produz uma porção de material irregular retirada da peça pela ferramenta, denominado cavaco. Por fim, Diniz, Marcondes e Coppini (2013) cita que os movimentos entre a peça e a ferramenta que permitem a ocorrência do processo de usinagem. Pode-se notar que basicamente o processo de usinagem é a interação entre peça e ferramenta que produz cavaco, a fim de se conferir forma final à uma peça inicialmente bruta.

Figura 1 - Três paradigmas relativamente distintos da usinagem.



Fonte: Adaptado de Wirtz (1991, apud Trent; Wright, 2000, p. 2).

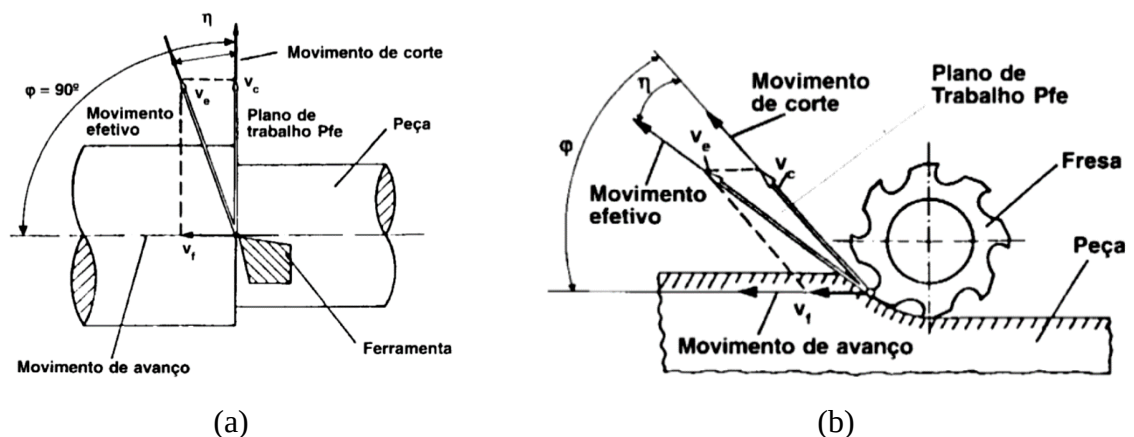
A usinagem, embora normalmente associada a grandes indústrias (automotiva, aeroespacial etc.), é parte crucial em diversas atividades de fabricação que, em alguns casos, requerem alta precisão e qualidade no produto. Wirtz (1991, apud TRENT; WRIGHT, 2000, p. 2) propõe três níveis de precisão na usinagem (Figura 1).

Existem dois tipos de movimentos na usinagem, classificados como ativos e passivos por Diniz, Marcondes e Coppini (2013). Ferraresi (1970) complementa que os movimentos ativos causam diretamente a saída do cavaco e os passivos não têm participação direta na formação do cavaco. São esses movimentos que permitem a ocorrência do processo de usinagem e são considerados no projeto e na fabricação das máquinas-ferramentas que irão realizar o processo.

Ferraresi (1970) descreve três movimentos ativos (Figura 2): movimento de corte, movimento de avanço e movimento efetivo de corte:

- Movimento de corte: deslocamento entre a peça e a ferramenta que sem o movimento de avanço proporciona apenas uma remoção do cavaco.
- Movimento de avanço: movimento entre a peça e a ferramenta que, concomitante ao movimento de corte origina um levantamento contínuo de cavaco, durante vários ciclos. O avanço ainda pode ser uma resultante dos movimentos de avanço principal e lateral.
- Movimento efetivo de corte: este movimento é a resultante dos movimentos de corte e de avanço, realizados ao mesmo tempo e continuamente. Diniz, Marcondes e Coppini (2013) complementa que, quando o movimento de avanço é descontínuo, o movimento efetivo é o próprio movimento de corte.

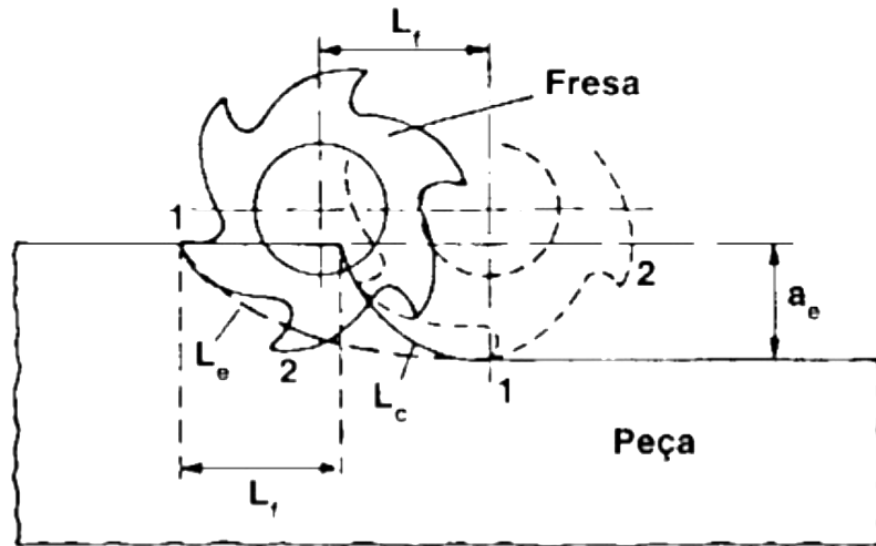
Figura 2 - Direção dos movimentos de corte, de avanço e efetivo no torneamento (a) e no fresamento discordante (b). Ângulo da direção de avanço ϕ , da direção efetiva η e do plano de trabalho P_{fe} .



Fonte: Adaptado de Diniz, Marcondes e Coppini (2013, p. 15-16).

A todos os movimentos supracitados, estão associados direções, sentidos, velocidades e percursos que são importantes no processo de usinagem. Ferraresi (1970) distingue as direções (Figura 2) e percursos (Figura 3) em: de corte, de avanço e efetiva de corte.

Figura 3 - Percursos de corte l_c , de avanço l_f e efetivo l_e no fresamento discordante.



Fonte: Adaptado de Diniz, Marcondes e Coppini (2013).

Finalmente, Ferraresi (1970) distingue a velocidade de corte, a velocidade de avanço e a velocidade efetiva de avanço.

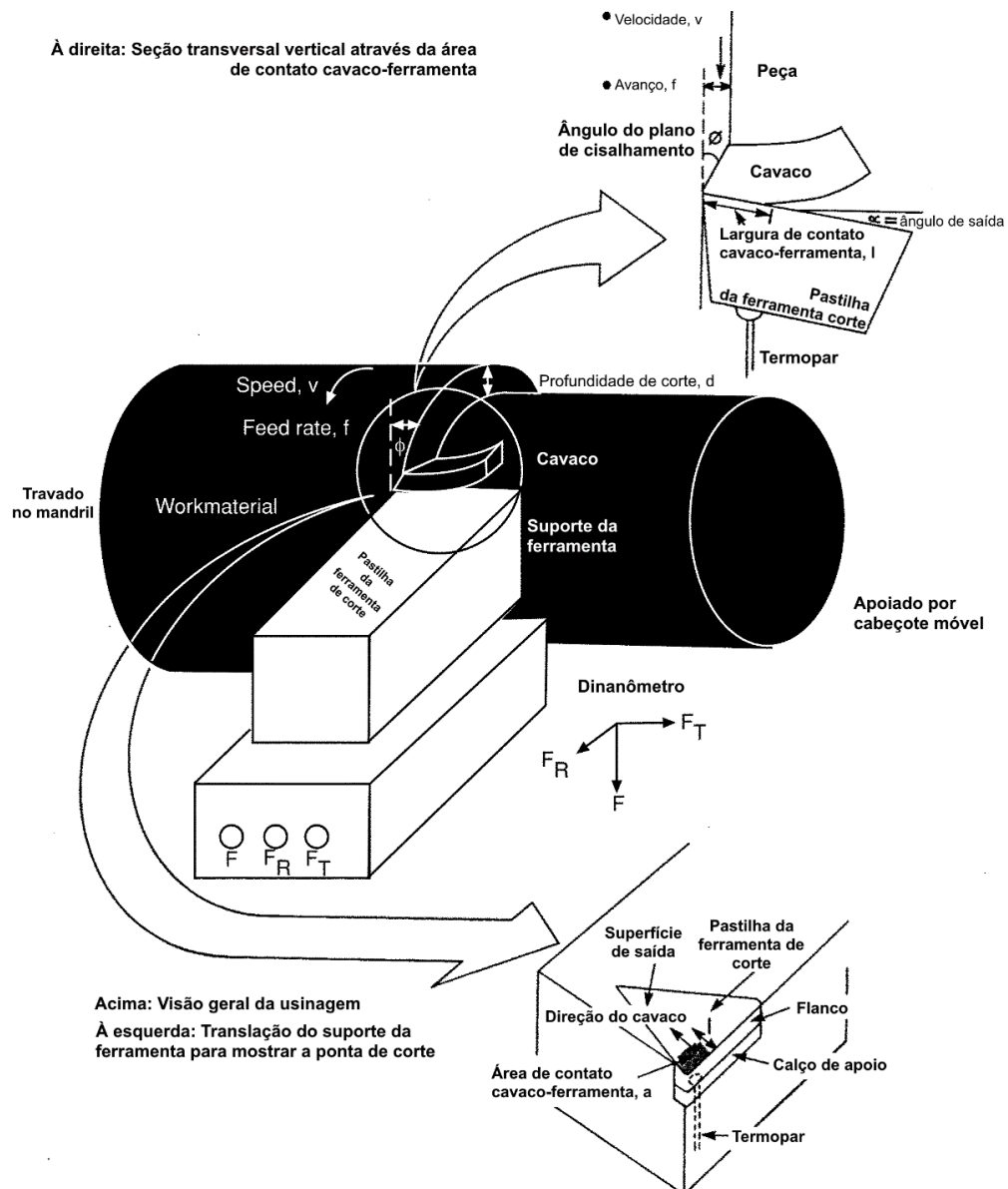
- Velocidade de corte: velocidade instantânea do ponto de referência da *aresta cortante*, com relação a direção e o sentido do corte.
- Velocidade de avanço: velocidade instantânea da *ferramenta* com relação a direção e o sentido de avanço.
- Velocidade efetiva de corte: velocidade instantânea do ponto de referência da *aresta cortante*, com relação a direção efetiva de corte.

2.2 TORNEAMENTO

Segundo Ferraresi (1970), a operação de torneamento (Figura 4) é o processo destinado a obter superfície de revolução com auxílio de um ou mais ferramentas *monocortantes*. Para isso, a peça gira em torno do eixo principal de rotação da máquina e a ferramenta se desloca ao mesmo tempo seguindo uma trajetória no mesmo plano com o eixo supracitado. Trent e Wright (2000) complementa que o torneamento, também chamado de corte semi ortogonal em

laboratórios, é a operação mais utilizada na usinagem de metais. A velocidade de corte (V), taxa que a superfície bruta passa pela aresta de corte, é expressa em m/min e o avanço (f) é a distância percorrida na direção axial pela ferramenta a cada ciclo de trabalho, conforme apresentado na Figura 4. Considerando a usinagem de metais, a velocidade de corte varia entre 3 e 200 m/min e o avanço entre 0,0125 e 2,5 mm por ciclo. Tem-se ainda a profundidade/espessura de corte (a_p), que é a espessura de material removido da peça, medido na direção radial. O produto escalar da velocidade corte, avanço e profundidade é uma medida de eficiência na operação de corte.

Figura 4 - Seção transversal vertical no torneamento e detalhe da geometria da pastilha de corte no canto inferior direito. A estrutura para dinamômetro e o termopar são utilizadas apenas em laboratório, para medição das forças de usinagem e temperatura em geral.



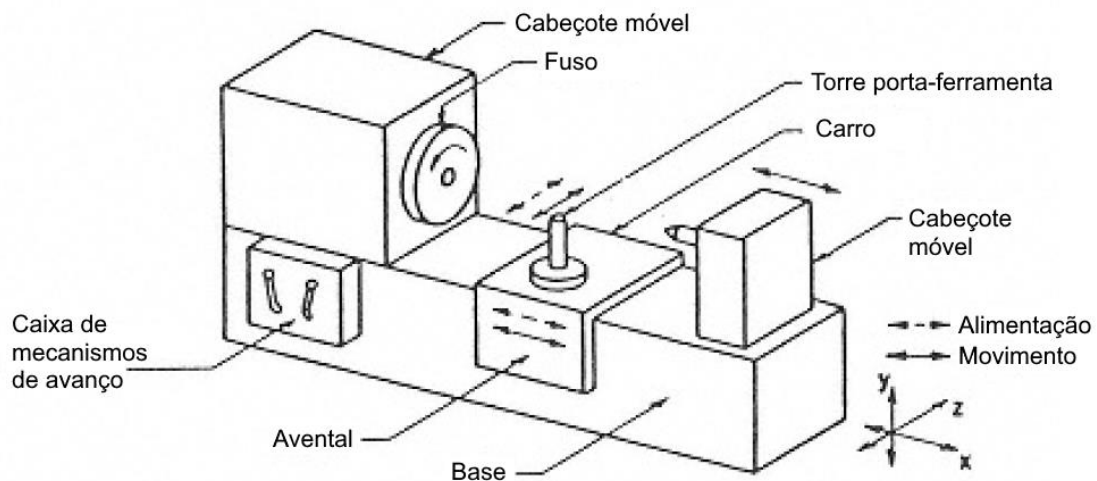
Fonte: Adaptado de Trent e Wright (2000, p. 10).

Com relação à classificação do torneamento, Ferraresi (1970) determina que ele pode ser retilíneo ou curvilíneo. No torneamento retilíneo, a ferramenta se desloca em segundo uma trajetória retilínea e pode ser dividido em cilíndrico, cônico, radial ou perfilhamento. Com relação ao torneamento cilíndrico, é o processo no qual a ferramenta se desloca numa trajetória curvilínea. As operações de torneamento também podem ser classificadas de acordo com sua finalidade:

- torneamento de desbaste: operação anterior à de acabamento, visando deixar a peça nas dimensões próximas às finais;
- torneamento de acabamento: operação destinada a deixar a peça com um acabamento superficial e em suas dimensões finais.

As máquinas utilizadas para o torneamento, segundo Stephenson e Agapiou (1996), são os mais antigos e versáteis equipamentos para usinagem, sendo conhecido como torno. Os tornos podem ter eixo horizontal ou vertical, sendo que para usinar peças com grandes dimensões utiliza-se os tornos com eixo vertical.

Figura 5 - Principais componentes e movimentos de um torno.



Fonte: Adaptado de Stephenson e Agapiou (1996).

São vários tipos de tornos, cada um adaptado a certas condições de uso, requerendo um bom projeto para evitar subdimensionamento, que pode levar à um gargalo no processo produtivo. Stephenson e Agapiou (1996) descrevem alguns tornos, sendo eles:

- tornos mecânicos: incluem os tornos verticais e os tornos horizontais, estes projetados para produzir superfícies planas normais ao eixo de rotação, em peças com grande diâmetro.

- torno revolver: tem como característica a fabricação em série de peças, pois possui várias ferramentas montadas na torre como se fosse o tambor de um revólver, com vários encaixes.
- torno multifuso (tornos automáticos): várias peças são transferidas entre os eixos rotativos e usinadas usando trocadores de ferramentas ou tornos copiadores, onde a ferramenta é acionada por cames ou mecanismo similar.
- torno copiador: geralmente utilizados para produção em massa de peças com múltiplas diâmetros ou contornos complexos.
- torno CNC (Comando Numérico Computadorizado): são utilizados em operações complexas de torneamento, onde se exige alta precisão. Possui controle numérico computadorizado para usinar peças com precisão. Existem programas de computadores para desenhos que convertem os arquivos gerados para a Linguagem G, que é padronizada para estes equipamentos.

2.3 FERRAMENTAS DE CORTE

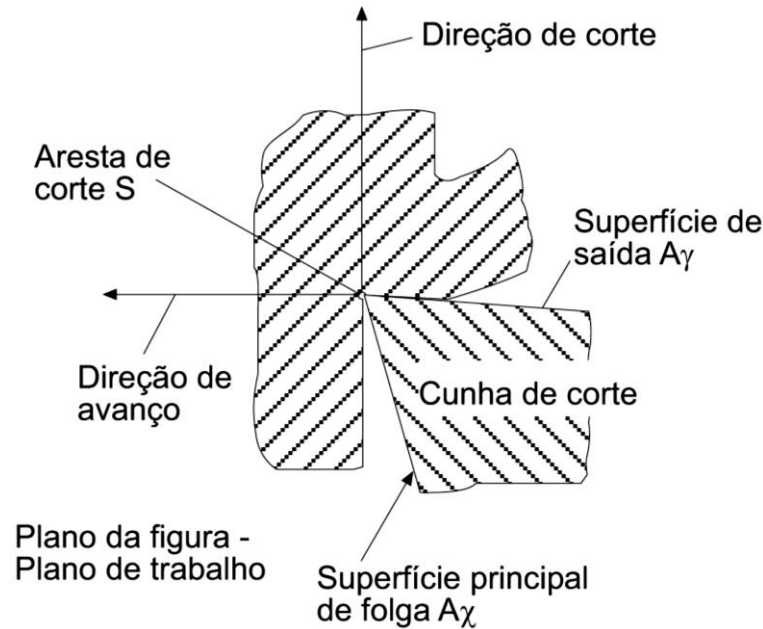
Trent e Wright (2000) lembram que as ferramentas de corte no início da história da usinagem eram construídas em sua grande parte por artistas plásticos. Durante a evolução da indústria, dos produtos e dos materiais das peças ocorreu um maior enfoque no desenvolvimento das ferramentas de corte, sendo estudados diversos materiais para a constituição destas ferramentas, sempre obedecendo aos seguintes critérios: vida da ferramenta, taxa de material removido, superfície da peça após a passagem da ferramenta, flexibilidade em diversas aplicações e o custo.

Geralmente as ferramentas de corte possuem as seguintes partes construtivas (Figura 6), descritas assim por Diniz, Marcondes e Coppini (2013):

- Parte de corte: parte ativa da ferramenta que é constituída pelas suas cunhas de corte, sendo esta construída sobre um suporte através do qual é possível fixar a ferramenta para construção, afiação, reparo, controle e trabalho.
- Cunha de corte: cunha de corte formada pela intersecção das superfícies de saída e de folga.
- Superfície de saída (A_γ): área sobre a qual o cavaco é formado e escoado durante sua saída da região de corte.
- Superfície principal de folga (A_α): superfície da ferramenta que mantém sua

aresta principal de corte e que encontra com a superfície de corte. Ferraresi (1970) acrescenta que esta superfície também é denominada de superfície de incidência.

Figura 6 - Cunha de corte da ferramenta de torneamento.



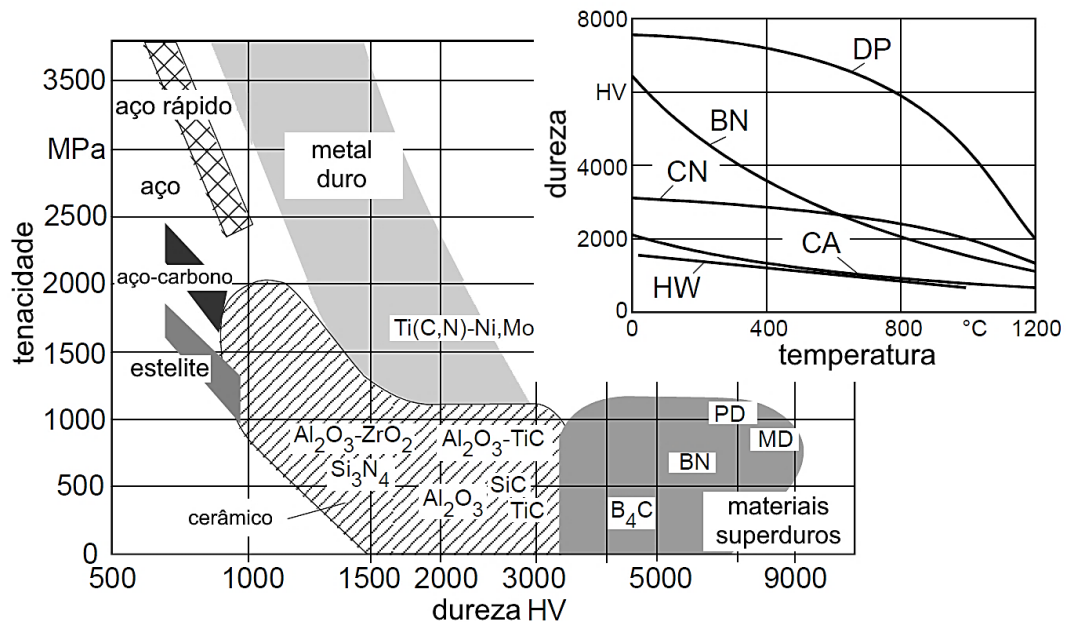
Fonte: Diniz, Marcondes e Coppini (2013, p. 25).

2.3.1 Materiais para ferramentas de corte

As ferramentas demandam novos processos de construção, pois são sujeitas cada vez mais ao alto nível stress pelas máquinas, pois são utilizadas tecnologias como usinagem a seco, alta velocidade de avanço ou usinagem de alta performance. Segundo Byrne, Dornfeld e Denkena (2003), uma ferramenta ideal combina alta dureza com boa tenacidade e estabilizante químico. Entretanto, alta dureza com boa tenacidade representam propriedades opostas no material, pois ou o material tem alta dureza ou possui boa tenacidade, ainda não existindo um único material que contemplem as três situações ideais. Mas, para reduzir o desgaste combina-se o material da ferramenta com um revestimento resistente ao desgaste de 3 a 10 μm .

Para a seleção do material para ser utilizado na ferramenta, Ferraresi (1970) entende que os seguintes critérios devam ser atendidos: material a ser usinado, natureza da operação da usinagem, condição da máquina, forma e dimensões da ferramenta e custo do material para a ferramenta. Diante disso, Byrne, Dornfeld e Denkena (2003), complementa que não devem ser consideradas somente as propriedades mecânicas do material, a resistência ao choque em função da temperatura é uma das características mais importantes do material (Figura 7).

Figura 7 - Dureza e tenacidade de materiais para ferramentas de corte.



Fonte: Adaptado de Byrne, Dornfeld e Denkena (2003, p. 132).

O metal duro é um material muito utilizado para ferramentas. A sua estrutura é formada basicamente por carbonetos (que fornecem resistência ao desgaste e dureza a quente – WC, TiC, TaC) e um ligante metálico (atua na ligação dos carbonetos frágeis – Co ou Ni). Originalmente o material era composto por carboneto de tungstênio (WC+Co), sendo muito eficiente na usinagem de ferros fundidos cinzentos, mas sofrendo com a baixa resistência na superfície de saída quando usinava aço. Para mitigar este problema, adicionou-se TiC, TaC e/ou NbC aos WC+Co, conseguindo-se reduzir significativamente a craterização quando se usinava aço. As velocidades de corte, quando comparado com o aço rápido, aumentaram em praticamente dez vezes: de 35m/min para 250-300m/min. Os metais duros são classificados por classes, da maior para a menor velocidade de corte: P, M, K, N, S e H (MACHADO et al, 2009).

2.4 VIBRAÇÃO NO PROCESSO DE USINAGEM

Segundo Stephenson e Agapiou (1996), durante o corte uma quantidade de energia é dissipada através da deformação plástica e do atrito, onde uma parte dela é transmitida aos elementos do sistema, induzindo vibrações – são movimentos relativos entre a ferramenta e a peça. Estas vibrações podem causar o desgaste prematuro do equipamento e a quebra da ferramenta de corte, prejudicando a produtividade e a danificando a peça que está sendo usinada. Logo, vibrações é um fator que limita a velocidade de corte em muitas operações,

reduzindo as taxas de produção. Por isso, Choudhury, Goudimenko e Kudinov (1997) complementam sobre a importância do controle da vibração no processo de usinagem, pois essa variável afeta negativamente o acabamento da superfície, desgaste da ferramenta, força de corte etc.

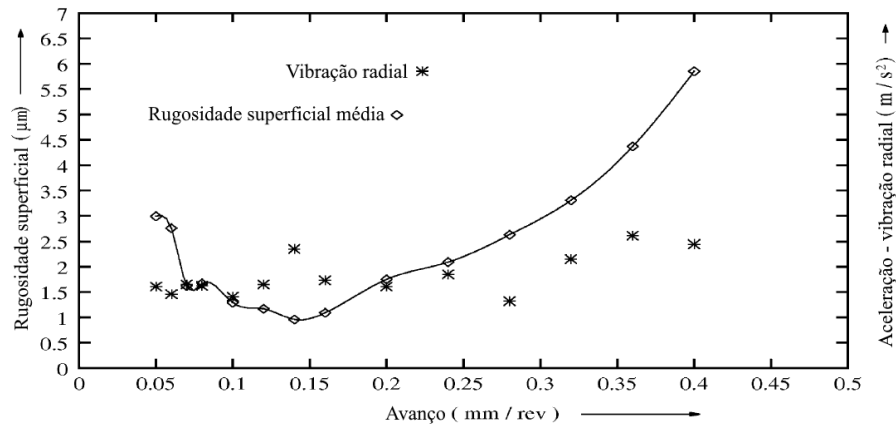
Alguns métodos foram desenvolvidos para estudar o comportamento dinâmico na usinagem, com dois objetivos: identificar maneiras para o projeto de máquinas estáveis e identificar condições de corte dinamicamente estáveis. Tradicionalmente, são estudadas formas de reduzir as deflexões, que produzem erros dimensões nas peças usinadas. Outrossim, a rigidez e amortecimento do sistema são os principais itens de projetos voltados ao controle a amplitude e estabilidade das vibrações. A vibração no sistema também pode surgir a partir da vibração da peça ou do acessório, mas esta é menos agressiva para o processo de usinagem do que a vibração da estrutura da máquina (STEPHENSON; AGAPIOU, 1996).

Com relação à determinação quantitativa da vibração, de acordo com Lamb (2013) um extensômetro (Figura 41) pode ser ajustado para medir vibração.

Risbood, Dixit e Sahasrabudhe (2003) propuseram modelos para previsão do acabamento superficial e o desvio dimensional de peças no torneamento, onde avaliou-se a força de corte e a vibração envolvidas no processo. Utilizou-se como ferramenta de corte o aço rápido, metal duro e metal duro revestido por TiN, com ou sem utilização de fluidos de corte. Os modelos foram baseados em redes neurais, utilizando-se experimentos empíricos (Figura 8) para validação e comparação com os modelos simulados.

Para medição empírica das forças de corte utilizou-se um dinamômetro e para medição das vibrações, um acelerômetro. O modelo neural proposto na maioria das situações apresentou valores com erros baixos, porém em alguns casos os valores previstos diferiram consideravelmente das medições. Constatou-se que para prever apenas a rugosidade superficial é necessário apenas um acelerômetro, medindo a vibração radial.

Figura 8 - Variação do acabamento superficial com avanço de 135m/min.



Fonte: Adaptado de Risbood, Dixit e Sahasrabudhe (2003, p. 210).

Por outro lado, Elangovan *et al.* (2015) utilizaram aprendizado de máquina para prever a rugosidade superficial no torneamento com aquisição apenas dos dados de vibração. Fixou-se quatro parâmetros de corte (velocidade, avanço, espessura de corte e desgaste de flanco) em diferentes ensaios (Figura 42) para validação e calibração do modelo, totalizando 81 experimentos em um torno CNC, com acelerômetro piezoelétrico para aquisição dos dados de vibração, utilizando-se ferramentas com insertos de metal duro *TNMG160408*. Construiu-se um modelo de previsão utilizando análise de componentes principais (ACP) para reduzir a dimensionalidade dos dados, regressão múltipla, que permite prever uma variável a partir de outras variáveis preditoras com ajuste do R^2 , que indica o quão perto estão os dados da linha de regressão ajustada. A aquisição de dados baseada no acelerômetro mostrou-se muito econômica. Obteve-se uma razoável previsibilidade utilizando o ACP, com menor utilização de recursos computacionais, mostrando-se útil no desenvolvimento de previsão de rugosidade superficial utilizando reconhecimento de padrões.

2.5 EMISSÃO ACÚSTICA NA USINAGEM

A emissão acústica é uma variável importante na usinagem de materiais, pois pode indicar ou prever várias situações que ainda não se apresentam a nível macroscópico, como por exemplo, uma falha catastrófica ou o fim da vida de uma ferramenta de corte. Barry e Byrne (2001) entendem emissão acústica (EA) como sendo uma onda transitória da tensão elástica gerada pela rápida liberação de energia da deformação elástica de certos fenômenos microestruturais de um material. Esses fenômenos incluem: movimento de deslocamento, macla, fratura dúctil e decoesão lamelar, início e propagação de fissuras, transformação de fase

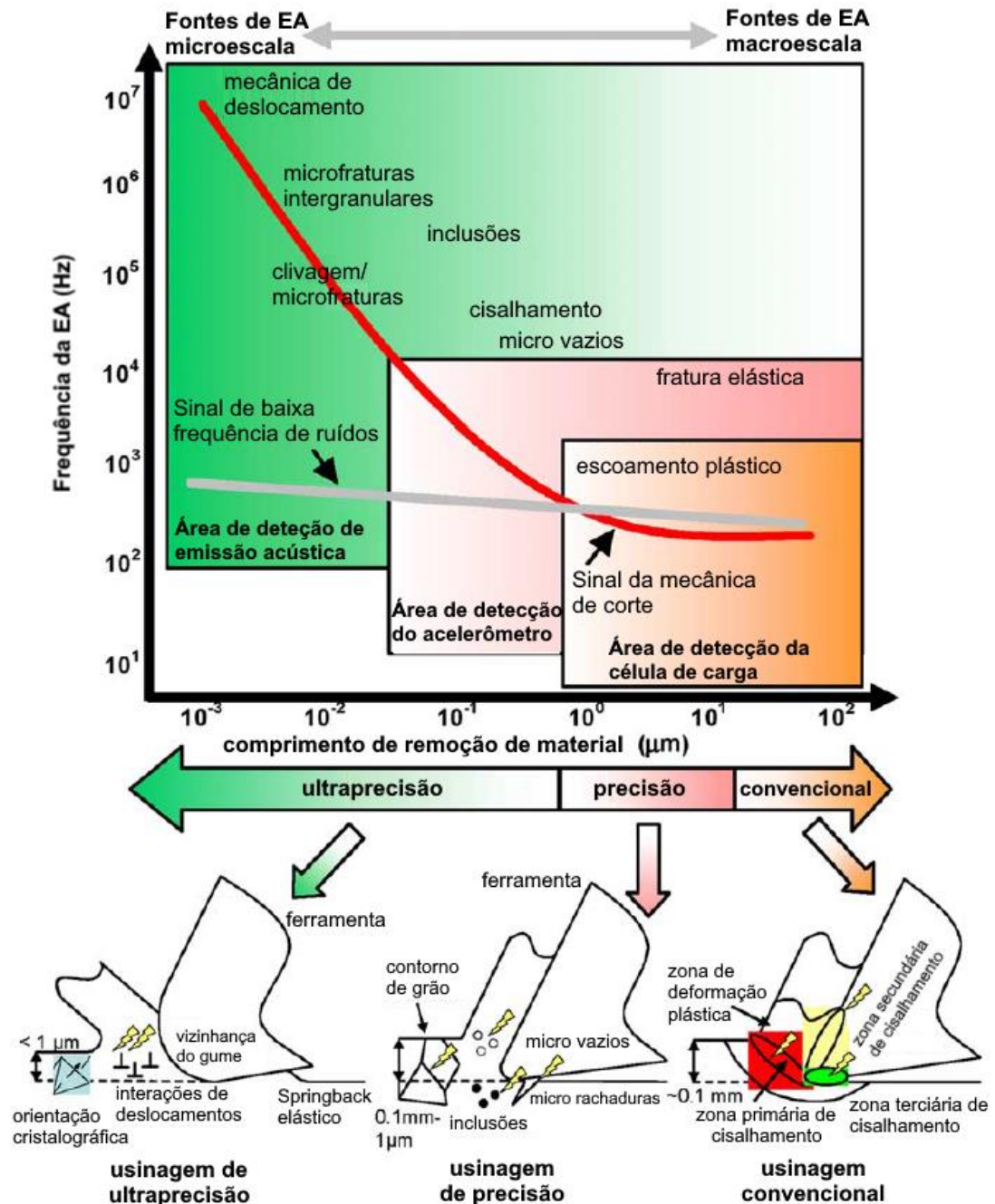
martensítica e atrito (HEIPLE & CARPENTER, 1987 apud BARRY & BYRNE, 2001).

De acordo com Beggan *et al.* (1999), a emissão acústica é uma das muitas variáveis que devem ser monitoradas e informadas aos operadores na usinagem. Essa emissão acústica é originada nas interfaces entre ferramenta/peça, ferramenta/cavaco e peça/cavaco (Figura 9); sendo diretamente afetada por alterações nas condições de usinagem, como por exemplo, o fim da vida da ferramenta. O monitoramento da emissão acústica é facilitado na usinagem pois a faixa de frequência sonora é maior do que as vibrações das máquinas e do ruído do ambiente. Logo, com utilização de filtro high-pass, os dados de emissão acústica da usinagem podem ser obtidos. Antes de uma falha catastrófica da ferramenta, há uma grande liberação de energia de emissão acústica na fratura e logo depois os níveis caem e permanecem os ruídos de fundo (Pflueg & Nedess, 1996 apud BEGGAN *et al.*, 1999). Por outro lado, Holford (2000, apud KISHAWY *et al.*, 2018) perceberam que as fontes de emissão acústica na usinagem (Figura 9) podem ser divididas em duas partes: fontes primárias e secundárias. As fontes primárias envolvem as causas mecânicas da fratura e o crescimento de trincas por fadigas, por exemplo: fraturas dúcteis, clivagem e microfraturas. As fontes secundárias de emissão acústica envolvem o processo de fechamento de trincas (quando as faces opostas de uma trinca permanecem em contato) e o desenvolvimento de trincas, por exemplo: detecção de trincas, vazamentos, identificação de peças soltas e desgastes). Os sinais de emissão acústica podem ser expressos através de diversos parâmetros, sendo eles: frequência média, amplitude do pico de sinal, *rise time* (tempo gasto por um sinal para mudar de um sinal baixo para um alto), contagem de *ring-down* (número de vezes que o sinal excede os limites).

Na usinagem de precisão, as fontes de emissão acústica podem ser um pouco diferentes da usinagem convencional. De acordo com Lee *et al.* (2006), a origem pode estar na interação da microestrutura da ponta da ferramenta com a peça, como vazios, inclusões (partículas), contorno de grãos e deslocamento de massa nas zonas de cisalhamento (Figura 9).

Com relação ao processo de micro torneamento (usinagem em uma área entre 1 e 10^5 mm²), Ribeiro Filho *et al.* (2019) descrevem que a emissão acústica é sensível à operação de corte e não sofre interferência do equipamento, mas tem relação com a aspereza da superfície da peça, ou seja, os sinais de emissão acústica aumentam com progressivamente de acordo com a rugosidade superficial e a micro dureza. O resfriamento do processo por ventilação resultou em maior variação na emissão acústica, porém as forças de corte podem ser maiores.

Figura 9 - Fontes de emissão acústica em diferentes estágios de remoção de material.



Fonte: Adaptado de Lee *et al.* (2006, p. 179).

No torneamento, segundo Beggan *et al.* (1999), existem quatro mecanismos principais de geração de sinais de emissão acústica: deformação do material, atrito e quebra de cavacos (Figura 10), colisões entre cavaco e a ferramenta ou a peça. Na zona primária de cisalhamento, a deformação do material da peça e os mecanismos de formação de trincas são fontes reconhecidas de emissão acústica (Kannatey-Asibu & Dornfeld, 1981 apud BEGGAN *et al.*, 1999). Na zona secundária de cisalhamento, a origem da emissão acústica reside no atrito entre

2.6 INFORMÁTICA

A história é dividida por várias fases, algumas delas tão importantes que podem ser chamadas de ondas: Revolução Agropecuária, Revolução Industrial e Informática. Dada a velocidade e exatidão de um computador, desde que seja bem programado, pode executar algumas tarefas que antes era exclusiva de um ser humano, liberando-se assim as pessoas para executar demais tarefas. Logo, o homem não vive mais em função da máquina, mas pensa, desenvolve, especifica e a máquina produz (NASCIMENTO; HELLER, 1996).

2.6.1 Estrutura básica de um computador

O *hardware* de um computador, de acordo com Nascimento e Heller (1996), é toda a parte física do computador e o *software* é um programa usado para obter os resultados esperados de acordo com a programação atribuída. No processamento da informação, há a entrada do dado, o processamento pelas unidades lógica e de controle e a saída do dado processado. O conjunto das unidades de armazenamento, de lógica e aritmética e controle é denominado Unidade Central de Processamento (CPU, em inglês) e as demais unidades são chamadas de periférico.

A estrutura de um computador que possui apenas um processador (single-processor), também conhecida como estrutura Top-Level, é descrita por Stallings (2016) da seguinte forma:

- CPU (Unidade Central de Processamento): controla as operações do computador e as funções de processamento de dados utilizando a Unidade de Controle, Unidade de lógica e aritmética e registradores, com auxílio do barramento interno para intercomunicação entre esses componentes. A velocidade de processamento é medida em *MIPs* (milhões de instruções por segundos e marcada pelo relógio de frequência em Mhz – mega-hertz), ou seja, um processador com *clock* de 1 GHz oscila na frequência de 1.000 x 1.000.000 de instruções por segundo, o que resulta em 1 bilhão de instruções por segundo (NASCIMENTO; HELLER, 1996).
 - Unidade de controle: controla as operações da CPU, ou seja, controla o computador.
 - Unidade de Lógica e Aritmética (ULA): realiza as operações de processamento de dados de um computador.
 - Registradores: é a memória de armazenamento dentro da CPU, que pode armazenar uma certa quantidade de bits. Um registrador importante é o CPSR (Current

Processor Status Register – Registrador de Estado Atual do Programa), que contém informações sobre o estado atual do processador, inclusive do modo de execução.

- Barramento interno: Promove a comunicação entre os componentes internos da CPU.

- Memória principal: Armazena dados. Geralmente a memória principal é a Memória RAM, que serve para armazenar temporariamente o conteúdo de um programa de computador para acesso direto da CPU, pois o barramento entre a memória RAM e a CPU é muito mais veloz em comparação ao caminho do disco rígido até a CPU.

- Entrada e Saída (E/S): Comunica dados entre o computador e os dispositivos conectados externamente ao computador.

- Sistema de interconexão (barramento): Promove a comunicação entre a CPU, memória principal e dispositivos de E/S.

O computador trabalha com sistema numérico binário, apenas entende 0 e 1, que representam estados de tensão elétrica (1-ligado ou 0-desligado). Mas o bit não representa o caractere, representa apenas o menor pedaço do caractere quando este é armazenado na memória. Logo, segundo Nascimento e Heller (1996) para representar o caractere tem-se o byte, que é um conjunto com 8 bits, ou seja, para armazenar o número 6_{10} (110_2), ele ficará assim na memória: 0000 0110. Atualmente os computadores trabalham com mais de 8 bits, chegando até 64 bits nos microcomputadores atuais. Simplificando, o computador processa a informação bit a bit (computador de 8 bits), de quatro em quatro caracteres (computadores de 32 bits) ou de oito em oito caracteres (computadores de 64 bits). Logo, tem-se o termo *palavra*, conjunto de caracteres (1 byte = 8 bits) que são processados de cada vez num computador.

De acordo com Stallings (2016), temos três tipos de instruções para processadores: Computador com um conjunto complexo de instruções (Complex Instruction Set Computer – CISC), Computador com um conjunto reduzido de instruções (Reduced Instruction Set Computer – RISC) e Superscalar. Ao comparar os processadores com arquitetura CISC's em relação aos com arquitetura RISC's, temos a principal diferença baseada no tipo de instrução que cada um pode processar.

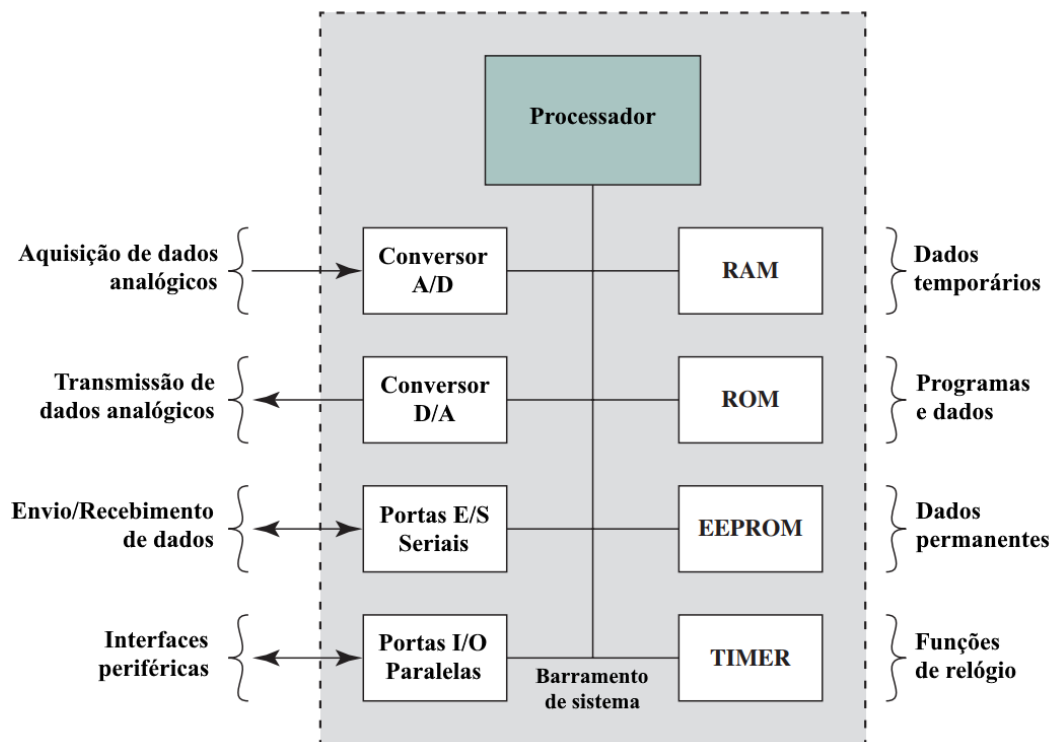
Segundo Abd-El-Barr e El-Rewini (2005), a arquitetura RISC precisa executar mais instruções para processar uma mesma função e cada instrução é executada mais rápida em relação à arquitetura CISC. Para otimizar esse processamento, os processadores com arquitetura RISC utilizam a área que seria dedicada aos decodificadores de instruções complexas para entregar muitos registros do CPU, unidades de execução adicionais e caches para instruções, possibilitando uma redução no tráfego entre o processador e a memória. A arquitetura CISC

requer um complexo esquema de decodificação, estando assim sujeita a atrasos na parte lógica. Sobre qual design utilizar, devem ser levados em conta a estratégia e o uso dos processadores.

2.6.2 Estrutura básica de um microcontrolador

Pode-se considerar o microcontrolador (Figura 11) como um computador em um chip, focado em economia de energia e de recursos. Segundo Stallings (2016) ele é um chip que possui uma estrutura básica, com processador baseado na arquitetura RISC, memória ROM (não volátil para o programa), memória RAM (volátil) para entrada e saída, um sistema de clock e uma unidade de controle para entrada e saída. Os microcontroladores são utilizados em diversas áreas, ultimamente mais em processos de automação. A arquitetura de seus processadores varia entre 4 e 32 bits, sendo que eles operam em clock na faixa dos Mhz, muito mais lentos em relação aos processadores de microcomputadores que operam em GHz. Geralmente os microcontroladores não proporcionam interação humana, sendo programados para executarem tarefas simples apenas quando requisitados.

Figura 11 – Estrutura comum de um microcontrolador.



Fonte: Adaptado de Stallings (2016, p. 33).

Com relação à organização deste tipo de processador (Figura 12), os dados são trocados entre processador e memória externa através do barramento. As instruções passam pela unidade de decodificação antes de ser executada pela unidade de controle. Os itens são colocados num conjunto de registradores de 32 bits. As instruções de processamento geralmente possuem dois registradores de origem, **Rn** e **Rm**; e um registrador de resultados/destino, **Rd**. Os valores de origem alimentam a Unidade de Lógica e Aritmética ou uma unidade multiplicadora. Além disso, os processadores ARM podem incluir uma unidade de *hardware* que pode alterar ou rotacionar o valor de **Rm** antes deste entrar na ULA. Essa alteração ou rotação que ocorre dentro do ciclo de instrução potencializa e flexibiliza muitas operações de processamento de dados. Os resultados da operação retornam para o Rd. As instruções de carga e armazenamento de múltiplos registradores podem utilizar a saída das ULA para gerar os seus endereços de memória (STALLINGS, 2016).

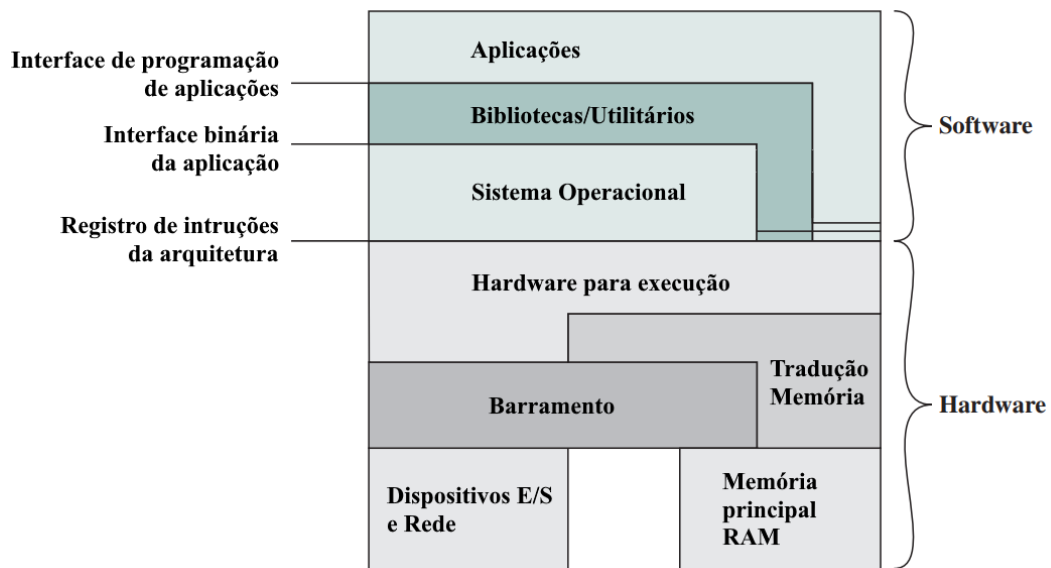
2.6.4 Sistema Operacional e programas de computadores

O sistema operacional (S.O.) nada mais é do que um programa de computador, porém, tem como diferencial proporcionar o controle de forma amigável e eficiente para o usuário de um computador e todos os seus periféricos.

Segundo Stallings (2016), pode-se visualizar o programa de computador e os componentes físicos usados para realizar as tarefas ao usuário em camadas (Figura 13), sendo que o usuário está interessado apenas no resultado de sua requisição.

O programa de computador ou aplicativo é desenvolvido através de uma linguagem de programação que é compilada em *linguagem de máquina* (gera-se um executável binário) ou interpretada (não há geração de executável), onde todo o controle da parte física do computador é facilitado com a utilização de bibliotecas ou programas, que são fornecidas pelos responsáveis da linguagem de programação.

Figura 13 - Estrutura física e lógica de um computador.



Fonte: Adaptado de Stallings (2016, p. 277).

2.7 AUTOMAÇÃO

Com o desenvolvimento da tecnologia, principalmente os computadores, nos séculos XX e XXI, ocorreu a redução de custo de itens de informática e eletrônica, por isso, houve grande inserção destes componentes em todos os setores da economia. Especificamente na indústria, tal desenvolvimento colaborou para a maior adoção de uma ferramenta cada vez mais importante, a qual proporciona o aumento da escala, da eficiência e da qualidade produtiva: a automação.

Convém diferenciar mecanização da automação, pois, de acordo com Roggia e Fuentes (2016) a mecanização consiste em reduzir o esforço físico e auxiliar em atividades, enquanto a automação permite a realização de tarefas por meio de máquinas que são controladas automaticamente, auxiliada por sensores e atuadores. A automação começou a ganhar destaque no início da Revolução Industrial, no século XVIII, permitindo a produção de mercadorias em maior escala, concepção de máquinas mais precisas e produtivas; e novas fontes de energia, como o vapor, em substituição à energia hidráulica ou muscular. A energia elétrica passou a ser utilizada no século XIX, onde ocorreram diversos avanços nas áreas industriais, comunicações e de transportes. Conforme já exposto, os computadores são a base de toda a tecnologia na área de automação contemporânea, estando presente em praticamente todas as áreas de conhecimento.

Haja vista toda a vantagem que a automação garante ao processo produtivo, existem

algumas desvantagens que devem ser analisadas, as quais Lamb (2013) descreve da seguinte forma: não existe tecnologia para automatizar todos os processos, o custo é maior do que se utilizado o trabalho manual em processos que não sejam repetitivos e de alto volume, os custos iniciais são relativamente altos e deve possuir mão de obra especializada na empresa para realizar a manutenção de toda a parte de automação. Ainda assim, em muitos casos as vantagens se sobressaem. Em países com alto índices de automação, a população desfruta de um padrão de vida mais alto em relação aos países com baixos índices. Além disso, a mão de obra passou a ser mais especializada e complexa com a implementação da automação, em relação à mão de obra de tarefas manuais.

Quanto ao tipo de processo, Roggia e Fuentes (2016) dividem a automação industrial em duas modalidades:

- processos de manufatura: são aqueles onde há grande movimentação mecânica de partes, como por exemplo, na indústria automobilística, onde há diversos recursos de automação, como robôs e esteiras. As grandezas mais comuns são força, velocidade e deslocamento.
- processos contínuos: são aqueles processos em que há pouca movimentação mecânica de partes, como uma estação de tratamento de efluentes. As grandezas mais comuns são temperatura, pressão e vazão.

2.7.1 Analógico e digital

Tem-se vários conceitos e terminologias que são importantes na automação, pois a automação requer a utilização de diversas ferramentas para o monitoramento e controle de processos produtivos.

O mais básico elemento é o estado **digital**, onde um interruptor ou sinal pode ser 0 (desligado) ou 1 (ligado). De acordo com Lamb (2013), diversos elementos na automação podem ser representados basicamente por 0 ou 1: sensores, estado de um motor, válvula ou luz piloto, além da própria máquina em si. Porém, esse conceito não é amplo para alguns equipamentos, pois eles podem trazer mais informações do que simplesmente estar ligado ou desligado, como informação de velocidade, rotação e outros parâmetros que só podem ser descritos numericamente, através dos sinais de alterações na tensão ou corrente. Nesses casos, é utilizada uma representação **analógica** do valor, que pode ser um número inteiro ou decimal. Os sinais analógicos são conectados a um circuito e depois convertido em número digital. Além das entradas analógicas, as saídas também são na forma de alterações na tensão ou correntes,

podendo ser convertidas a partir de valores digitais, podendo assim controlar, por exemplo, a velocidade de um motor ou a posição de uma válvula. De toda forma, para a manipulação dos valores pelos computadores e sistemas de controles, estes devem ser convertidos na entrada utilizando circuitos de conversão analógico-digital (ADC) e na saída por circuitos de conversão digital-analógico (DAC), sendo esses conversores projetados para trabalhar numa faixa fixa de sinais baseada na aplicação. O número de etapas digitais que um conversor é capaz de conhecer é descrito pelo número de bits do sinal digital. Por exemplo, um DAC de 16 bits em relação a um de 14 bits exibe um maior número de valores subdivididos dentro de um intervalo e possui maior resolução. Outra especificação relacionada com sinais analógicos é a linearidade, que está relacionada ao processo de conversão ou a aspectos do sinal medido, ou seja, entende-se como o desvio do sinal convertido em relação ao original.

Para a conversão das tensões lidas em unidade de medida, como massa, realiza-se muitas vezes a interpolação entre dois ou mais valores. Por exemplo, tem-se uma leitura analógica da massa de um corpo através de um sensor, onde 0V representam 0 kg; 2,5 V representam 10 kg e 5,0 V representam 20 kg. Temos uma formula do tipo $y = m.x + b$, que no caso é equivalente a equação $y = 4.x$, onde y é a massa e x o valor lido no sensor em volts.

2.7.2 Controladores

Os controladores são uma parte importante de uma automação industrial, pois de acordo com Lamb (2013), são eles que promovem parte dos cálculos, computação e gerenciamento de dispositivos de entrada e saída, sendo que os controladores podem funcionar em rede em toda a planta ou em um núcleo.

Podem ser reconhecidos como controladores os seguintes componentes: computadores, sistema de controle distribuído, controladores lógicos programáveis e sistemas embarcados.

2.7.2.1 Computadores, Sistema de controle distribuído (DCS) e sistemas embarcados

Os computadores são utilizados para a programação de alguns componentes da linha, como os controladores lógicos programáveis e ainda, servem como interfaces humano-máquina (IHM), onde a saída se dá por monitores. Possuem como vantagem o baixo custo, dada a sua versatilidade. Os Sistemas de controle distribuídos (DCS) são utilizados em processos contínuos ou em lotes, onde os sensores e atuadores são conectados ao DCS trabalhando de forma integrada, utilizando um ponto de ajuste para controlar o fluxo de material pela planta.

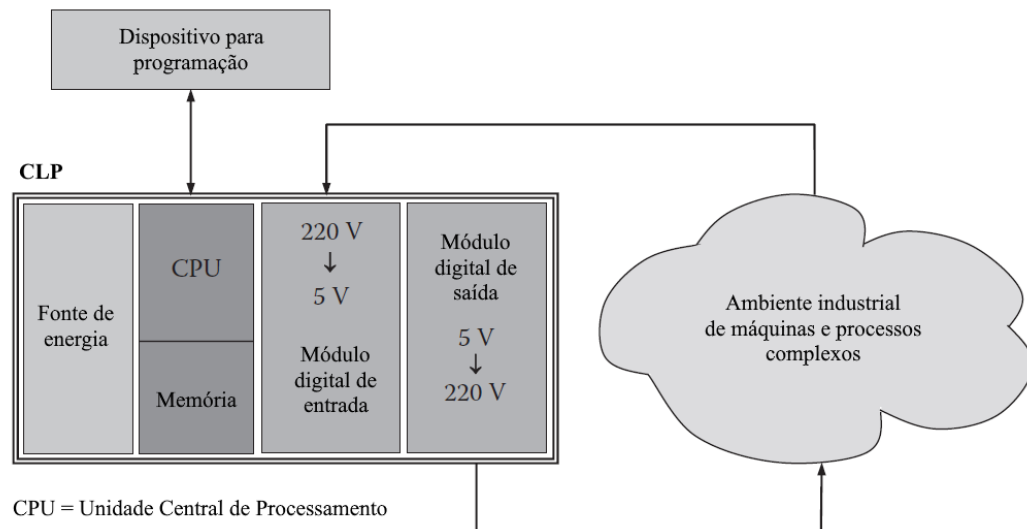
São comuns em indústrias químicas. Pela forma integrada de trabalhar, esse sistema promove o controle, manutenção e operação de forma mais fácil, permitindo ainda agilidade na tomada de decisões. Um DCS pode utilizar várias estações de trabalho em rede ou apenas uma, off-line. Os Sistemas Embarcados são basicamente sistemas computacionais dedicados para realização de algumas funções, não possuindo grande poder computacional, sendo geralmente partes de um dispositivo mais complexo (LAMB, 2013).

2.7.2.2 Controladores Lógicos Programáveis (CLP)

Os controladores lógicos programáveis (Figura 14) são amplamente utilizados em plantas automatizadas, sendo basicamente computadores digitais utilizados como controladores eletromecânicos de processos. Eles são programáveis com auxílio de microcomputadores ou através de interface do próprio controlador lógico programável.

Os controladores lógicos programáveis podem aceitar diversas linguagens de programação e scripts, mas a linguagem de programação mais comum é a denominada Ladder (escada), que é uma linguagem de fácil aprendizado e de ampla aplicação, pois pode-se controlar todos os dispositivos de saída que estão conectados ao controlador lógico programável, de acordo com as informações lidas através dos dispositivos de entrada nele conectados. Esses equipamentos são construídos de forma a suportar a maior parte das intempéries encontradas no ambiente de produção, garantindo grande robustez ao controlador lógico programável. Os controladores lógicos programáveis podem funcionar em redes ou off-line; a estrutura pode ser compacta ou serem modulares. Como são utilizados microcontroladores para gerenciar os controladores lógicos programáveis, eles não são conhecidos por possuírem grande espaço de memória ou alto poder computacional, entretanto, conseguem desempenhar de forma satisfatória às tarefas as quais são programados.

Figura 14 - Estrutura interna de um Controlador Lógico Programável.



Fonte: Adaptado de Manesis e Nikolakopoulos (2018, p. 194).

2.7.2.2.1 Módulos de entrada e saída digitais do CLP

Basicamente, conforme Manesis e Nikolakopoulos (2018), os módulos de entrada digitais recebem sinais discretos (valores digitais 0 ou 1) através dos dispositivos de entrada – que podem ser contínuos (CC) ou alternados (CA), convertendo-os para processamento posterior e sendo isolados eletricamente e modulados à uma tensão de 5V, enquanto são transferidos para a memória do CLP (Controlador Lógico Programável). Conforme supracitado, os sinais digitais possuem apenas 2 estados: ON (ligado) ou OFF (desligado). Com relação a esses módulos, o estado OFF significa sem tensão e o estado ON corresponde à tensão nominal do módulo, que podem ser de 24, 48, 110 ou 230 VDC (voltagem em corrente contínua) ou VAC (voltagem em corrente alternada), sendo mais comum o de 24 VDC devido ao menor risco e intensidade de acidentes elétricos (por exemplo, choques). Ademais, são comuns módulos digitais com 4, 6, 8, 10, 16 ou 32 entradas. Os módulos de saída digital funcionam de maneira similar aos módulos de entrada, diferenciando-se apenas em sua aplicação.

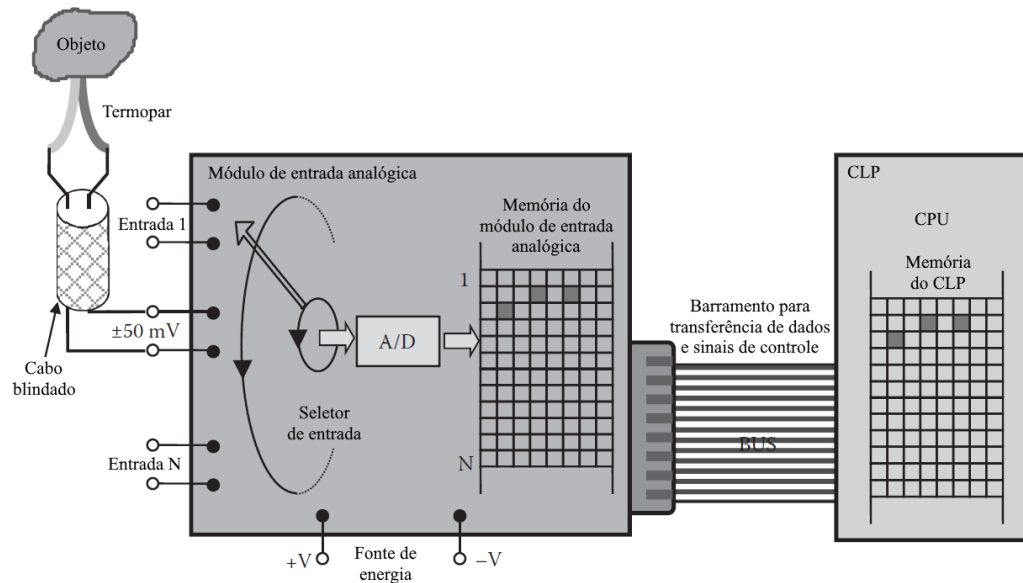
2.7.2.2.2 Módulos de entrada e saída analógicos do CLP

Conforme Manesis e Nikolakopoulos (2018), um módulo de entrada analógica possui todos os circuitos necessários para a entrada de sinais analógicos (geralmente na forma de tensão ou corrente elétrica) e sua conversão para digital, que é realizado com um conversor analógico-digital, muitas vezes integrado. Após a conversão, os valores digitais

correspondentes ao sinal analógico são armazenados em uma memória acessível pela Unidade Central de Processamento (CPU), a qual realiza tarefas programadas no controlador lógico programável. Os sinais analógicos são padrões e compatíveis com os sinais de saída destes dispositivos (sensores ou atuadores). Com relação aos módulos digitais, os módulos analógicos possuem como principal diferencial uma unidade para armazenamento dos valores digitais dos sinais analógicos. Tomando-se como exemplo um termopar do tipo R (platina-platina) apresentado na Figura 15, após a detecção da temperatura de um objeto, aplica-se uma *tensão analógica* correspondente num cabo blindado (utilizado para reduzir ruídos em sinais elétricos fracos) que está conectado à uma entrada analógica do módulo. Após a conversão desses sinais em valores digitais e o armazenamento desses dados na memória da entrada analógica, os dados podem ser transferidos ao controlador lógico programável através de um barramento, onde após a transferência desses dados para a memória do controlador lógico programável este executará as rotinas de acordo com os valores ali contidos. Ademais, todo módulo de entrada analógica possui apenas um conversor de sinais analógicos para digitais que é comum a todas as entradas do módulo.

Sobre os módulos de saída analógicos, eles realizam o processo inverso em relação aos módulos de entrada, ou seja, os módulos de saída recebem os valores digitais através do controlador lógico programável, realizam sua transformação para tensão ou corrente elétrica para realizar a operação através de um atuador ou dispositivo analógico. De acordo com Manesis e Nikolakopoulos (2018), os valores digitais dos sinais digitais são geralmente de 16 bits – sendo 11 bits para representação digital do sinal analógico, e são convertidos em sinais analógicos geralmente em correntes elétricas de 4 a 20 mA ou em tensões entre -10 VDC a +10 VDC.

Figura 15 - Termopar, módulo de entrada analógico que possui um conversor de sinais analógicos para digitais para todas as entradas, barramento para transferência de dados e um controlador lógico programável.



Fonte: Adaptado de Manesis e Nikolakopoulos (2018, p. 217).

2.7.3 Sensores

Segundo Roggia e Fuentes (2016), o sensor é um elemento sensível a uma forma de energia do ambiente, relacionando informações sobre algo que precisa ser mensurado, como por exemplo: temperatura, pressão, emissão acústica, corrente ou vazão.

De acordo com Manesis e Nikolakopoulos (2018), os sensores transformam a variação quantitativa de um fenômeno e o converte em um sinal de saída elétrico, o qual pode ser analógico ou digital. Em sensores analógicos há a produção de um sinal contínuo **proporcional** à variação medida, por exemplo, em sensores de pressão a corrente varia entre 4 e 20 mA DC e a tensão de 0 a 10 V DC, que será proporcional à leitura da pressão entre 0 e 725 psi. Já os sensores digitais produzem um sinal de saída **discreto** na forma de 0 (OFF) ou 1 (ON). Logo, tem-se sensores analógicos ou transdutores (saída proporcional) e sensores digitais ou discretos (saída discreta).

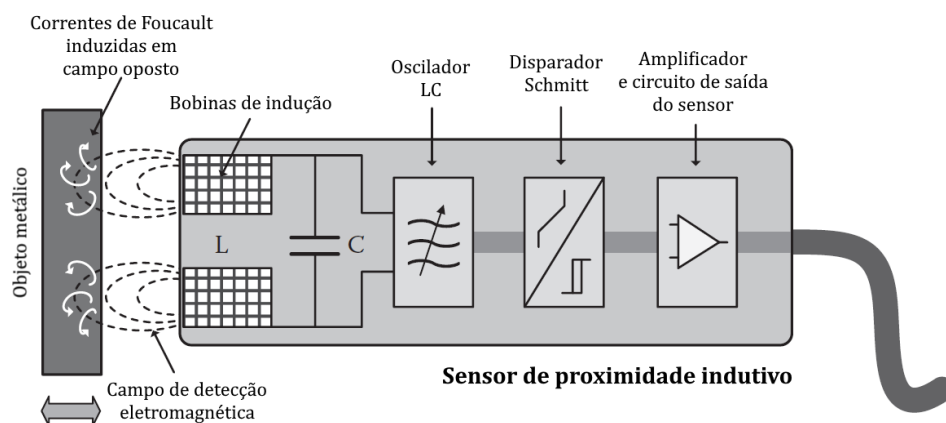
2.7.3.1 Sensores Analógicos

Segundo Kurfess (2005), os sensores analógicos além de medir distância, força, torque, aceleração ou tensão, podem ainda medir posição angular e de translação em relação a um ponto

de referência. Os sensores e tecnologias mais comuns para medir essas variações são:

- LVDT (transdutor de deslocamento variável linear): geralmente utilizado para medição de movimento retilíneo (movimento mecânico ou vibrações). É constituído de um núcleo móvel que se estende até o centro de um tubo oco, sendo que uma bobina primária e duas secundárias são enroladas em torno deste tubo. O núcleo é fixado ao objeto que se deseja mensurar o movimento (KURFESS, 2005).
- Temperatura: Há diversos tipos de sensores para medir temperatura, que possuem diversas características e dependem de aplicação e o princípio de operação. Os tipos mais comuns são: *termostatos*, *termopares*, *detector resistivo de temperatura (RTD)*, *termistores* e *infravermelhos*. Os sensores de temperatura infravermelho medem a energia infravermelha emitida por objeto como resultado de seu estado térmico, obtendo-se assim a sua temperatura (MANESIS & NIKOLAKOPOULOS, 2018).
- Indutivos (sem contato): são sensores de proximidade que identificam quando há a passagem de um objeto metálico, seja ferroso ou não, não há contato. Geralmente é constituído por duas bobinas, que geram um campo eletromagnético de alta frequência a frente do sensor (Figura 16), comumente atingindo uma faixa de 30% do diâmetro da bobina ativa. É esse campo magnético que identifica objetos metálicos. Um oscilador LC também faz parte do sensor. Esse tipo de sensor pode detectar objetos metálicos mesmo que estejam protegidos por objetos não metálicos (KURFESS, 2005).

Figura 16 - Elementos internos de um sensor indutivo.

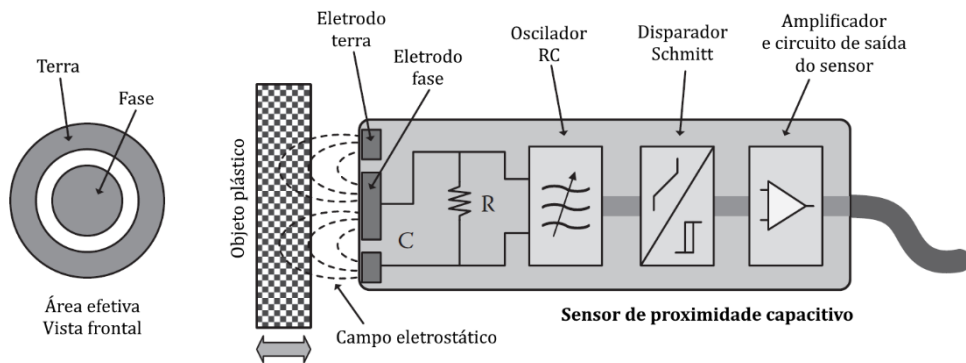


Fonte: Adaptado de Manesis e Nikolakopoulos (2018, p. 42).

- Capacitivos (sem contato): também são sensores de proximidade, mas identificam objetivos metálico e não metálicos (Figura 17). Em geral, ele tem um campo de atuação na ordem de aproximadamente 25% do diâmetro do sensor. Um atenuante do

desempenho deste tipo de sensor é a contaminação do ambiente que pode reduzir a constante dielétrica entre o sensor e o objeto alvo (KURFESS, 2005). De acordo com Manesis e Nikolakopoulos (2018), esse tipo de sensor gera um campo eletrostático, podendo identificar materiais como papel, vidro, madeira ou líquidos, sendo que o princípio de utilização é o mesmo do sensor indutivo. No lugar do oscilador LC que compõe o sensor indutivo, no sensor capacitivo entra no lugar um oscilador RC.

Figura 17 - Elementos internos de um sensor capacitivo.



Fonte: Adaptado de Manesis e Nikolakopoulos (2018, p. 43).

- Ópticos: esses sensores também são de proximidade, sem contato, baseando-se na emissão de um feixe de luz – por exemplo, laser ou LED, que é recebido por elemento fotossensível (KURFESS, 2005).
- Ultrassônicos: esses sensores emitem um pulso de alta frequência (na ordem de 20 kHz) – gerados por transdutores piezelétrico ou eletrostático, até o objeto ao qual se deseja saber a distância que se encontra. O tempo que leva para esse pulso chegar ao objeto e retornar é proporcional à distância entre o sensor e o objeto alvo (LAMB, 2013).
- Strain gauge/Extensômetro: é o tipo de sensor mais utilizado na robótica. Pode-se mensurar com esse tipo de sensor a força, pressão, potência e torque. A sua resistência elétrica é alterada de acordo com a tensão mecânica de entrada, podendo ser positiva (tração) ou negativa (compressão). É utilizada uma *ponte de Wheatstone* para medir a alteração de resistência do sensor, estando disponíveis com valores de resistência entre 30 e 3000Ω.
- Tem-se ainda outros tipos de sensores, como acelerômetros e de efeito hall.

2.7.3.2 Sensores Digitais

Conforme Kurfess (2005), os sensores digitais são utilizados são do tipo estático, pois seu

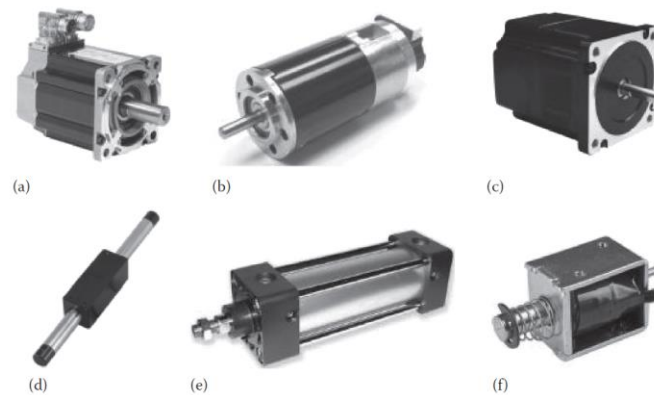
valor é baseado no estado digital da saída. Sempre sua saída será o estado ON/OFF (LIGADO/DESLIGADO). Geralmente são utilizados para controle de proximidade e de limites; e cortinas de luz para segurança de operadores. De acordo com Kurfess (2005), existem alguns tipos de sensores digitais, como por exemplo:

- Interruptores: os interruptores podem vir configurados com contatos normalmente abertos (NA) ou normalmente fechados (NC). Os tipos mais comuns são SPDT (polo simples, duas direções) e DPST (duplo polo, única direção).
- De proximidade: servem para determinar a posição de um objeto. São conhecidos o de proximidade, limites e cortina de luz. Adicionalmente, são semelhantes aos análogos analógicos e possuem os mesmos tipos, diferenciando-se apenas na saída discreta nos sensores digitais e na opção de desativar o disparador Schmitt.
- Sensores PNP e NPN: Esses tipos de sensores utilizam transistores no driver de saída, sendo classificados como **PNP** (o sinal de saída do sensor é **p**ositivo, mais comum) ou **NPN** (o sinal de saída do sensor é **n**egativo). A maioria dos sensores de proximidade na indústria são de estado sólido, termo que descreve o tipo de componente existente no sensor, como por exemplo, os transistores, que alteram a saída do sensor após a detecção de um objeto.

2.7.4 Atuadores

Segundo Manesis e Nikolakopoulos (2018), os atuadores (Figura 18) são dispositivos que utilizam algum tipo de energia – por exemplo: pneumática ou hidráulica, para produzir uma força necessária para fornecer movimento à um objeto ou acionar algo. Lamb (2013) entende que os atuadores podem ser de natureza linear, rotativa ou uma combinação de ambos, onde os atuadores lineares geram movimento rotativo empurrando um eixo rotativo fixado num eixo e os atuadores rotativos, como por exemplo, os motores, são utilizados para gerar movimento linear através de correia ou parafuso.

Figura 18 - Alguns dos atuadores utilizados em automação industrial: (a) servo motor, (b) motor de corrente contínua, (c) motor de passo, (d) motor linear, (e) cilindro pneumático e (f) solenóide.



Fonte: Adaptado de Manesis e Nikolakopoulos (2018, p. 20).

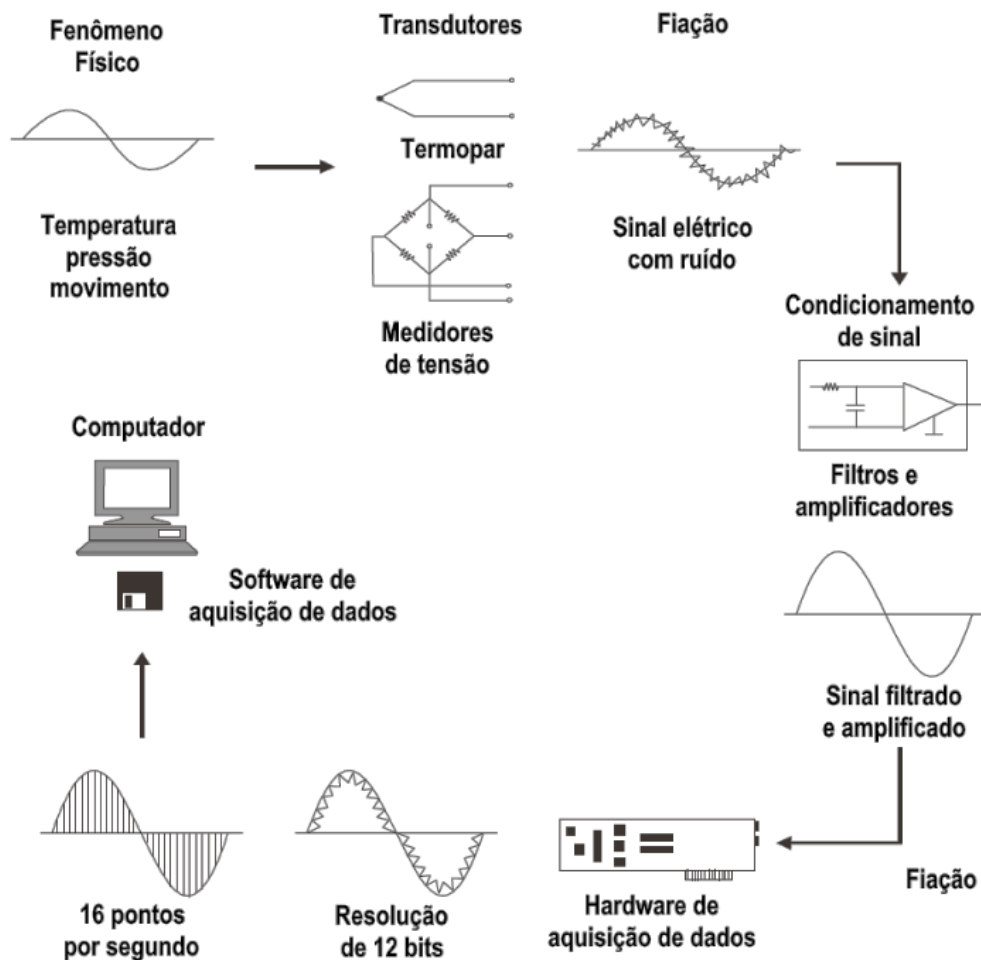
2.8 AQUISIÇÃO DE DADOS

De acordo com Park e Mackay (2003), a aquisição de dados é o processo em que os fenômenos físicos do mundo real são transformados em sinais elétricos, os quais posteriormente são convertidos num formato digital para processamento, análise e armazenamento por um computador. Wöstenkühler (2005) complementa que os sistemas digitais são amplamente utilizados, pois os circuitos são de baixo custo, precisos e relativamente simples de implementar. Ademais, os microcomputadores estão cada vez mais poderosos e, com isso, vem sendo utilizados para cálculos complexos de controle e medição digital. Conforme já mencionado, os dispositivos possuem conversores de sinais de analógico para digital e de digital para analógico, sendo conhecidos como conversores de dados. Segundo Park e Mackay (2003), além desses conversores, os sistemas de aquisição de dados empregam o uso dos seguintes elementos, de acordo com o diagrama apresentado na Figura 19: sensores e transdutores; fiação; condicionadores de sinais; parte física de aquisição de dados; microcomputadores (sistemas operacionais) e aplicações de aquisição de dados.

Conforme Park e Mackay (2003), os transdutores e sensores promovem a interface entre o mundo real e o sistema de aquisição de dados pela conversão dos fenômenos captados em sinais elétricos que são processados pelo sistema de aquisição de dados. A fiação promove a comunicação entre os elementos existentes no sistema de aquisição de dados, variando o tipo e as dimensões de acordo com a sua aplicação. O condicionamento de sinal é realiza a filtragem, amplificação, linearização, isolamento e excitação dos sinais elétricos para que possam ser processados com a maior qualidade possível pelo sistema de aquisição de dados. Os

componentes eletrônicos para a aquisição e controle de dados são responsáveis pela entrada, processamento e conversão dos dados de analógico para digital, utilizando conversores de sinais analógicos para digitais, quando for o caso; o processamento desses dados e a saída destes, convertendo-os de digitais para analógicos utilizando conversor de sinais digitais para digitais; saída de sinais analógicos para os processos de controle. As aplicações de aquisição de dados funcionam como aplicações nos microcomputadores que já possuem sistemas operacionais multitarefas instalados, proporcionando uma visão completa do sistema de aquisição de dados, com telas interativas, programa de controle dedicado para entrada e saída, relatórios e gerenciador de comunicações. Por fim, a configuração e arquitetura dos microcomputadores é determinante para o desempenho dos sistemas de aquisição de dados baseados em PCs, sendo desejável processador com arquitetura de no mínimo de 32 bits para se obter um mínimo de desempenho para aplicações de aquisição de dados em tempo real.

Figura 19 - Diagrama funcional de um sistema de aquisição de dados baseado em microcomputador.



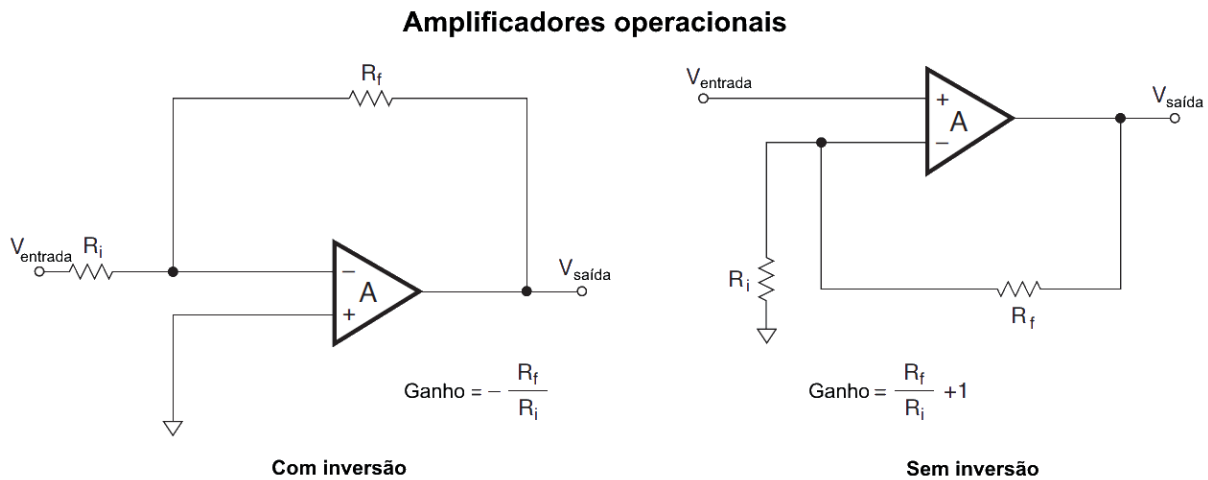
Fonte: Adaptado de Park e Mackay (2003, p. 2).

2.8.1 Condicionamento de sinais

De acordo com Park e Mackay (2003), o condicionamento de sinais é o pré-processamento dos dados necessários para converter os sinais elétricos recebidos pelos transdutores em sinais sem ruídos, amplificados, isolados e linearizados para que as placas ou outros componentes eletrônicos do sistema de aquisição de dados possam aceitar. Em alguns casos os transdutores requerem a excitação de corrente ou tensão. A maioria dos sistemas de aquisição de dados baseados em PC possuem pelo menos um tipo de equipamento para condicionamento do sinal. O condicionamento de sinal pode ser aplicado em diferentes tipos de produtos para condicionamento, abrangendo várias faixas de preço, desempenho, modularidade e facilidade de uso. A seguir, serão apresentadas algumas formas de condicionamento de sinais:

- **Amplificação:** Segundo Measurement Computing Corporation (2012), a maioria dos sistemas de aquisição de dados amplificam o sinal antes de seu processamento, utilizando amplificadores básico, que são facilmente configuráveis para amplificar ou atenuar o sinal. Esses amplificadores possuem muitos componentes em seus circuitos, mas são retratados como um simples bloco lógico e, apenas alguns resistores e capacitores determinam seu funcionamento. Geralmente, os amplificadores possuem dois terminais lógicos de entrada, denominados por terminal inversor (-) ou terminal não inversor (+), dois terminais para alimentação (+Vcc e -Vcc) e um terminal de saída (+). O ganho é equivalente à relação entre os valores das resistências de feedback (R_f) e a de entrada (R_i) (Figura 20). Há de se observar um limite para que a amplificação seja realizada sem dados, que é de 2V menor que a tensão de alimentação, por exemplo, para uma alimentação de 15V pode-se amplificar um sinal de entrada de até 13V. Dada sua versatilidade, faz com que este seja um componente analógico universal para condicionamento de sinais.

Figura 20 - Os dois tipos básicos de amplificadores são os com ou sem inversão de saída.



Fonte: Adaptado de Measurement Computing Corporation (2012, p. 41).

- **Isolamento:** De acordo com Park e Mackay (2003), o isolamento exerce várias funções importantes num sistema de aquisição de dados, como por exemplo, proteção de segurança para os equipamentos caros que compõe um sistema de aquisição de dados: microcomputador e placas DA, além disso, protege operadores contra descargas eletrostáticas, descargas atmosféricas e falhas em equipamentos. Os métodos mais comuns de isolamento são: óticos, magnético ou capacitivo. O isolamento ótico é utilizado para sinais digitais, o isolamento magnético e capacitivo para sinais analógicos.

- **Filtragem:** Segundo Park e Mackay (2003), a filtragem remove ruídos indesejáveis nos sinais adquiridos antes de serem amplificados e entregues aos conversores AD. De acordo com Measurement Computing Corporation (2012), os tipos de projetos mais comuns utilizados para filtragem de sinais são: Filtro Butterworth, Filtro Chebyshev e Filtro Bessel. Esses filtros possuem características que se adequam de acordo com o projeto que será utilizado, sejam aplicações *high-pass*, *low-pass*, *band-pass* ou *band-reject*. Podem ainda ser utilizados como filtros de rede ativos ou passivos.

- **Linearização:** Geralmente as funções de transferências em equipamentos eletrônicos possuem um fator não linear e, segundo Measurement Computing Corporation (2012), esse fator na maioria das vezes é desprezível e pode ser ignorado ou compensado via componentes eletrônicos ou programa de computador. Porém, alguns equipamentos possuem uma relação não linear da leitura do fenômeno com a tensão de saída grande o bastante para exigir uma compensação, como em termopares.

2.9 PLATAFORMAS DE COMPUTAÇÃO EMBARCADAS

Com o desenvolvimento da microinformática e dos meios de produção, os componentes eletrônicos passaram a ter seu custo reduzido, o que contribuiu para o avanço de vários setores. Um dos setores que se beneficiou com esse desenvolvimento foi o da automação, com o surgimento de plataformas para prototipagem eletrônica, sendo algumas de código aberto, o que aumenta a adoção e a contribuição da comunidade no desenvolvimento de novos recursos para esses equipamentos.

No ano de 2005, *Massimo Banzi* e *David Cuartielles* desenvolveram comercialmente o projeto o Arduino, que possui grandes vantagens sobre outras plataformas de desenvolvimento baseadas em microcontroladores, dada a facilidade de utilização e a rapidez com que os projetos são desenvolvidos sem grandes conhecimentos em eletrônica. A linguagem de programação mais utilizada para a programação em Arduino é a linguagem C, datada da década de 1960 e que possui grande número de tutoriais na Internet (MCROBERTS, 2011).

Com o intuito de universalizar o uso de microcomputadores, a Fundação Raspberry Pi realizou o desenvolvimento de um equipamento que pudesse ser barato e ao mesmo tempo atender à exigência de aplicações essenciais, como editor de texto e navegador de Internet. Em 2012, lançou-se a primeira versão do Raspberry Pi, uma placa contendo a estrutura básica de um microcomputador – com processador ARM e uma GPIO de 40 canais, cujo valor não passava de 35 dólares americanos, atingindo-se assim o objetivo de entregar um equipamento rápido e barato o suficiente para ser inserido em mercados subdesenvolvidos, focado no ensino de computação nas escolas (HALFACREE, 2018).

Devido ao grande sucesso dessas placas, ocorreu a popularização da chamada internet das coisas, onde equipamentos agora podem ser controlados e monitorados remotamente. Em comum, essas placas possuem o baixo custo e um vasto material de auxílio disponível na internet como principais chamarizes. Infelizmente, também existem no mercado muitas placas que são clones dos oficiais, em alguns casos de qualidade duvidosa.

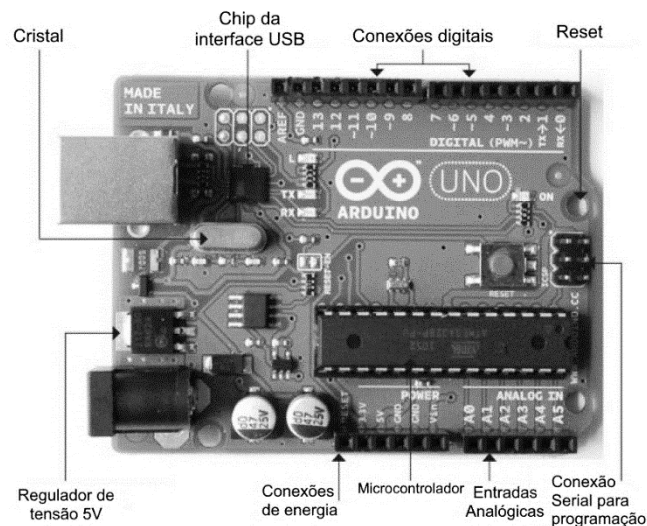
2.9.1 Arduino

McRoberts (2011) define o Arduino como um pequeno computador que pode receber uma programação para o processamento de entradas e saídas do dispositivo com os componentes externos a eles conectados, sendo denominado também de plataforma de computação física ou embarcada – sistema que pode interagir com o seu ambiente através de

seus componentes eletrônicos e programas de computador.

Mesmo com todos os fabricantes de microcontroladores fornecendo as ferramentas e placas para desenvolvimento de soluções, incluindo o Atmel que fornece o ATmega328 para o Arduino, esses kits servem apenas para engenheiros eletrônicos profissionais, não para entusiastas, requerendo deste público um maior investimento para a plena compreensão de seu funcionamento em relação ao Arduino, reduzindo-se sua adoção pela maior parte do público. Existem diversas versões oficiais das placas do Arduino, variando as dimensões, número de portas e processamento: Uno, Duemilanove, Diecimila, Mega, Nano, Bluetooth e Lilypad (MONK, 2012).

Figura 21 - Placa Arduino Uno.



Fonte: Adaptado de Monk (2012).

Pode-se entender que o Arduino nada mais é do que uma interface para microcontrolador, com entradas e saídas digitais e analógicas. Segundo Monk (2012), a placa do Arduino possui os seguintes componentes:

- Alimentação de energia da placa: logo abaixo do conector USB encontra-se o regulador de tensão de 5V, que regula a tensão entre 7 e 12V. observa-se que o chip regulador de tensão de 5V ocupa muito espaço, mas isso se deve à necessidade de dissipar o calor quando é regulada a tensão em correntes razoavelmente altas.
- Conexões de energia: estão localizadas na parte inferior da placa, a primeira denominada RESET serve para levar a ROM do Arduino para sua configuração de fábrica. As demais, 3,5V; 5V e 9V (Vin) são utilizadas para levar aos componentes que estão conectados ao Arduino essas tensões. Por último, a conexão GND é a terra, possuindo a tensão de 0V.

- Entradas analógicas: são seis conexões na parte inferior da placa (Figura 21), denominadas de A0, A1, A2, A3, A4 e A5. São medidas as tensões dos componentes conectadas a elas, geralmente variando de 0 a 5V, obtendo-se assim a leitura dos sensores conectados ao Arduino, como por exemplo os termopares. A corrente é mínima, dada a grande resistência que há (ou deve haver) até o pino. Essas conexões podem ser utilizadas para entrada e saída de sinais digitais.
- Conexões digitais: na parte superior da placa (Figura 21) observa-se catorze portas, numeradas de 0 a 13. Essas são as portas digitais do Arduino e podem ser de entrada ou saída. Elas trabalham com 0V ou 5V e 40mA, a não ser as portas PWM que possuem um til (~) do lado número da conexão, essas simulam o comportamento das portas analógicas, variando entre 0 e 5V. A tensão de 5V e a corrente de 40mA são suficientes para acender um *led* ou acionar um *buzzer*, mas não são capazes de acionar diretamente um motor elétrico. Convém observar que as portas digitais 0 e 1 são usadas para a transferência de dados com o computador, devendo ser desconectadas quando for realizar o upload de um programa para o Arduino.
- Microcontrolador: está localizado logo acima das portas analógicas um chip no formato retangular preto, com 28 pinos. O microcontrolador utilizado no Arduino é o modelo ATmega328 da Atmel. Ele possui processador de 8 bits com clock de 20 Mhz, 32 Kb de memória flash, 2 Kb de SRAM e EEPROM com capacidade de 1 Kb. Este último é uma memória não-volátil responsável por armazenar dados considerados importantes pelo usuário. As instruções dos programas são salvas na memória flash.
- Cristal oscilador: é um componente de quartzo, no formato retangular e na cor prateada, responsável por determinar com precisão a sincronia de tempo das operações (adição, subtração ou outra operação matemática) do microcontrolador, trabalhando na velocidade de 16 Mhz (16 milhões de vezes por segundo), limitando o Arduino a esse valor de clock, ao invés dos 20 Mhz do microcontrolador.

2.9.2 Computador de placa única (SBC)

Segundo Pajankar (2017), os computadores de placa única é um computador que possui toda a funcionalidade de um microcomputador, mas todos os seus componentes estão numa simples placa impressa. Todos os seus componentes, como CPU, memória RAM e GPU (unidade de processamento gráfico) estão soldados na placa. No entanto, esse tipo de computador não possui slots de expansão para periféricos e não permite o upgrade de seus

componentes, pois eles estão soldados na placa. O Quadro 1 apresenta as principais diferenças entre um computador de placa única e um microcomputador comum.

Quadro 1 - Diferenças entre os computadores de placa única e os computadores comuns.

Computador de placa única (SBC)	Computador comum
Não é modular	Modular
Componentes não podem ser trocados e nem melhorados	Componentes podem ser trocados ou melhorados
É um sistema em um chip (SoC)	Não é um SoC
É compacto	Possui grandes dimensões
É portátil	A maioria não é ou é semi portátil
Consome pouca energia	Consome mais energia
É mais barato que um computador comum	Custa mais do que um SBC

Fonte: Adaptado de Pajankar (2017, p. 2).

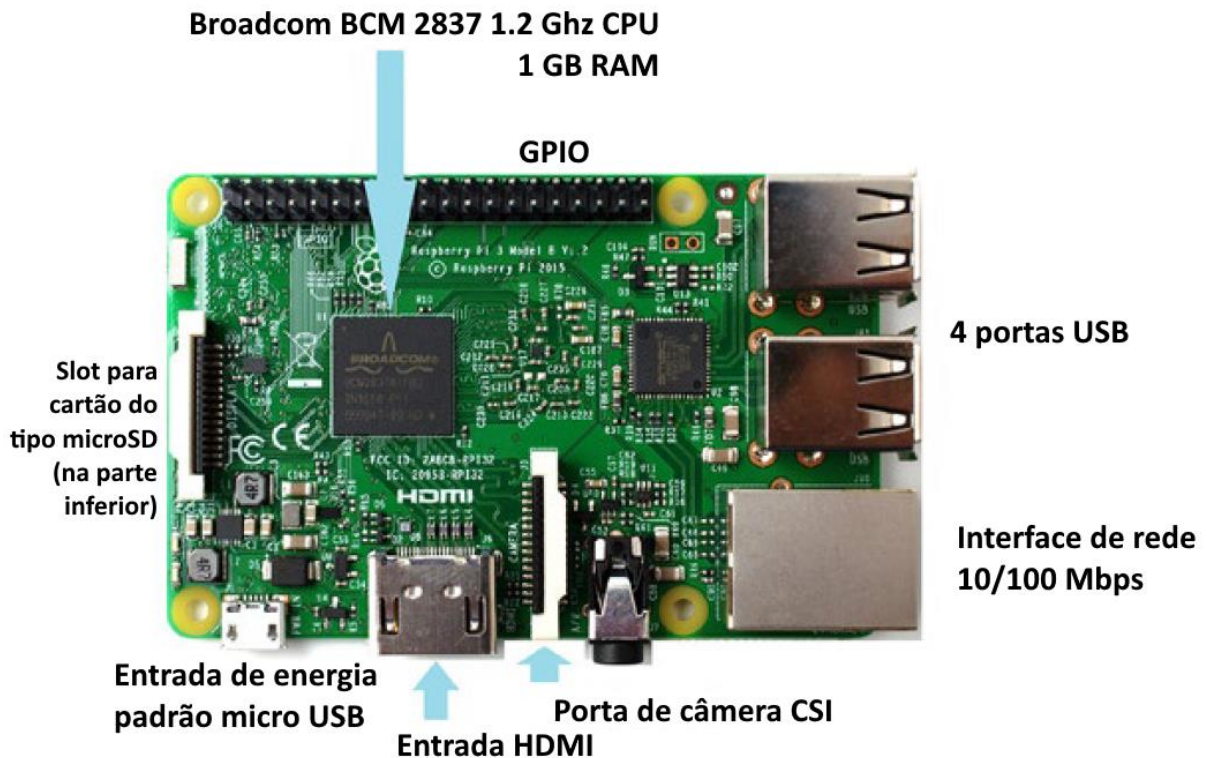
2.9.3 Raspberry Pi

Halfacree (2018) define o Raspberry Pi como um computador pequeno, notável e de baixo custo, pois ele pode ser utilizado para diversas aplicações as quais antes requeriam um computador normal. Pajankar (2017) complementa com a informação de que o Raspberry Pi nada mais é do que um computador de placa única, do tamanho de um cartão de crédito comum, sendo desenvolvido para promover o ensino de computação básica nas escolas de países em desenvolvimento, através de uma plataforma computacional de baixo custo.

Existem várias versões do Raspberry Pi desde o Modelo A, sendo sempre atualizados com relação às tecnologias mais eficientes do momento, atualmente o Raspberry Pi encontra-se na versão 4 Model B, podendo possuir 2 GB, 4 GB ou 8 GB de memória RAM, esta última configuração custando 75 dólares americanos (UPTON, 2020). Em cada versão, geralmente são oferecidos os Modelos A, A+, B e B+, sendo que o modelo com mais recursos e poder computacional é o Modelo B+ e o modelo com menos recursos e menor poder computacional da versão é o Modelo A (HALFACREE, 2018).

Em fevereiro de 2016 foi lançado o Raspberry Pi 3 Model B, também conhecido como 3B, uma atualização de modelos anteriores, sendo representada graficamente pela Figura 22.

Figura 22 - Vista superior do Raspberry Pi 3 Model B.



Fonte: Adaptado de Pajankar (2017, p. 5).

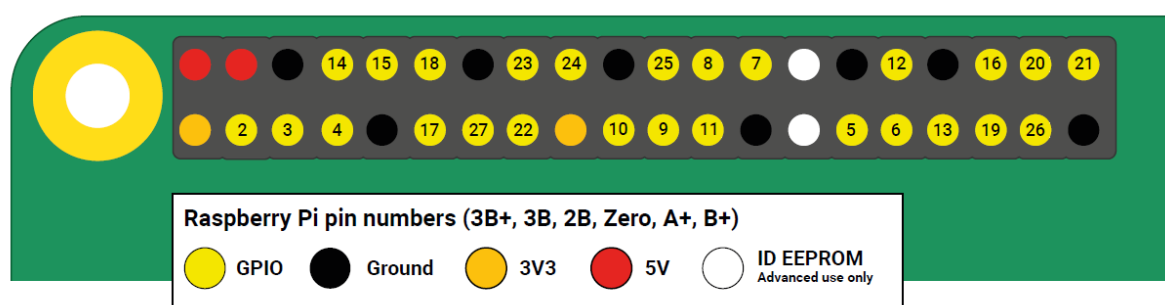
Como observa-se na Figura 22, o Raspberry Pi possui basicamente todos os componentes de um computador comum, só que soldados numa placa e em dimensões bem inferiores se comparadas a um microcomputador. Entretanto, os componentes possuem características particulares para este tipo de computador, conforme descrito por Halfacree (2018):

- CPU e GPU: como ele é baseado na estrutura SoC, um chip de silício possui boa parte do sistema do Raspberry Pi: CPU (responsável pelo processamento) e GPU (responsável pelo processamento gráfico). O processador é ARM, que tem como principais compromissos aliar um ótimo desempenho a um baixo consumo energético, esse tipo de processador vem sendo cada vez mais adotado pelos fabricantes de computadores comuns, não somente para os computadores de placa única.
- Memória RAM: funciona da mesma forma do que nos demais dispositivos, servindo como armazenamento temporário dos dados que precisam ser processados pelo processador. Os dados são copiados do cartão microSD (que é a unidade de armazenamento não volátil do Raspberry Pi) para a memória RAM, cujo barramento para comunicação com o processador possui alta velocidade e os dados são processados e devolvidos mais rapidamente.
- Rádio: esse componente confere ao Raspberry Pi a capacidade de se comunicar

com outros dispositivos via conexão sem fio, seja redes de computadores Wi-Fi ou *bluetooth*, para se comunicar com periféricos como mouses, teclados, sensores ou smartphones.

- Conexão USB e de rede Ethernet: há um chip localizado logo atrás das portas USB que é responsável pelo controle das conexões USB e de rede Ethernet. Com isso, o Raspberry Pi é capaz de trabalhar com os dispositivos que são conectados através das quatro portas USB e de se comunicar numa rede de computadores padrão Ethernet, que oferece uma conexão muito mais estável do que a Wi-Fi.
- PMIC: para gerenciar a energia recebida pelo equipamento via porta de energia padrão micro USB, há um chip PMIC (Power Management Integrated Circuit) que realiza todo esse gerenciamento energético.
- GPIO: por fim, localizado na parte superior da Figura 22 está o GPIO (general-purpose input/output), que possui 40 pinos divididos em duas fileiras de 20 pinos. Este é um recurso adicional que permite conectar diversos dispositivos ao Raspberry Pi, como *led*, botões e sensores digitais de todo tipo. Para conectar sensores analógicos no Raspberry Pi, é necessário que seja adicionado um conversor AD através da GPIO, onde o número de portas analógicas varia de acordo com o modelo do conversor. Entretanto, nem todas as portas do GPIO (Figura 23) servem para conectar dispositivos externos, algumas portas são dedicadas ao fornecimento de energia, aterramento ou ainda reservadas para o Raspberry Pi.

Figura 23 - Representação gráfica das portas do GPIO do Raspberry Pi, para os modelos 3B+, 3B, 2B, Zero, A+ e B+).



Fonte: Adaptado de Halfacree (2018, p. 121).

As portas **GPIO** servem para conexão lógica de dispositivos e são numeradas não sequencialmente de 2 até 27. Os pinos **3V3** fornecem alimentação de energia com tensão de 3,3 V; que é a mesma tensão que o Raspberry Pi trabalha internamente. Os pinos **5V** fornecem uma alimentação de energia elétrica a uma tensão de 5,0V; que é a mesma recebida na entrada de energia micro USB do Raspberry Pi. Os pinos GND (**Ground**, terra) são utilizados para fechar

um circuito, fornecendo 0V de tensão. Por fim, os pinos **ID EEPROM** são reservados para a conexão de componentes eletrônicos especiais (HAT) e outros acessórios.

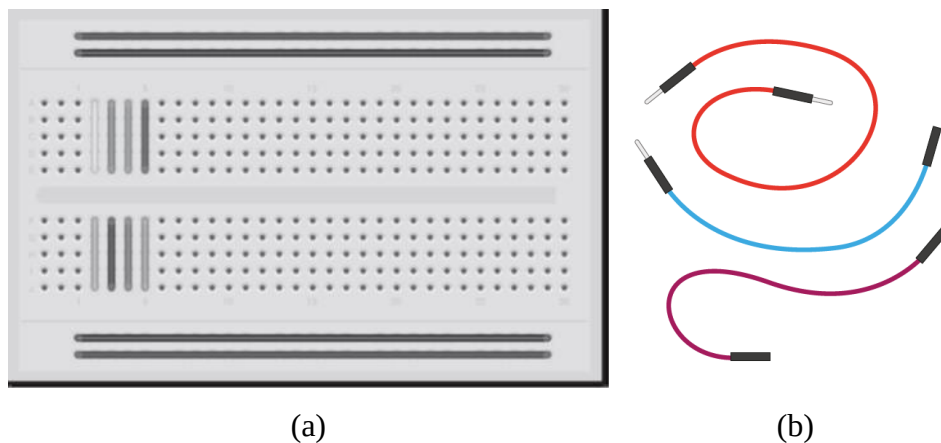
Da mesma forma que um computador comum possui um sistema operacional, o Raspberry Pi também possui sistemas operacionais especialmente desenvolvidos para ele, baseados em Linux em sua maioria, mas também há sistemas operacionais Windows para estes equipamentos. O sistema operacional desenvolvido pela Fundação Raspberry Pi é o **Raspberry Pi OS**, outrora conhecido como *Raspbian*.

2.9.4 Aquisição de dados com plataformas de computação embarcadas

Dado o aumento do poderio computacional, a versatilidade oferecida e o avanço dos sensores de baixo custo, as plataformas de computação embarcadas começaram a ser utilizadas em diversos projetos os quais antes apenas soluções proprietárias eram utilizadas. Essas soluções geralmente carregam alto custo de aquisição de manutenção, além de não oferecer a versatilidade para incrementos de funcionalidades durante sua aplicação.

Entretanto, durante o desenvolvimento de projetos utilizando as plataformas, são utilizados alguns componentes que tem como principal vantagem a flexibilidade, pois, por exemplo, não envolvem solda na etapa de prototipação permitindo diversas mudanças até que se atinja o objetivo do projeto e depois dessa etapa sim, ocorre a impressão de um circuito numa placa e a solda dos componentes. A placa utilizada para prototipação é denominada *protoboard* (Figura 24a) e a ligação entre os componentes é realizada via fios com conexões em suas extremidades denominados *fio jumper* (Figura 24b).

Figura 24 - Imagem ilustrativa da protoboard e de fios jumper.



Fonte: (a) McRoberts (2011, p. 46). (b) Halfacree (2018, p. 123).

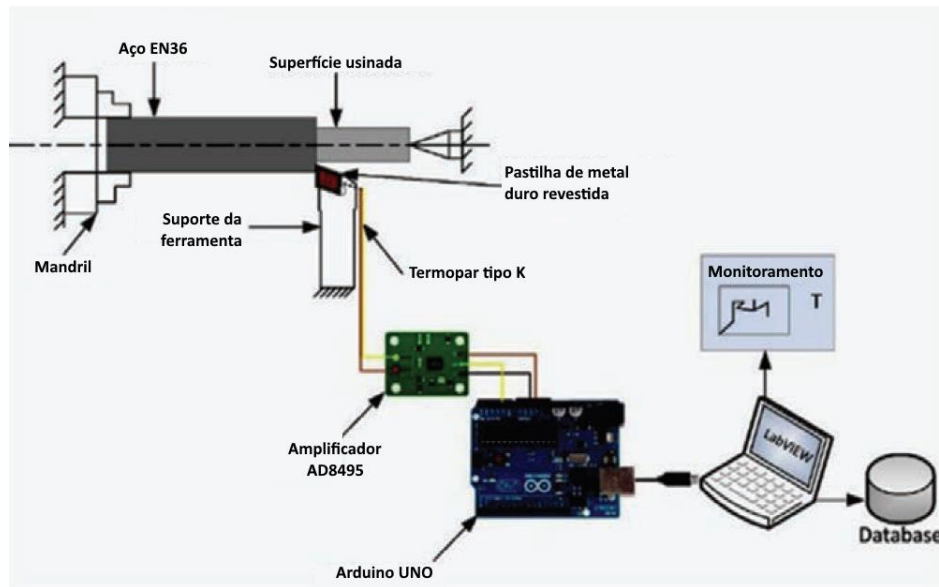
Os principais componentes utilizados na fase de prototipação são:

- Protoboard: também conhecida como matriz de contatos é utilizada para *prototipar* um circuito eletrônico sem a necessidade de solda. A placa consiste numa série de furos dispostos em grade, os quais são conectados por uma tira de metal condutivo. O barramento nas extremidades superior e inferior disposto horizontalmente é projetado para carregar alimentação (geralmente de 5V) e aterramento. Em protoboards maiores existe uma divisão neste barramento, favorecendo a alimentação da placa com diferentes tensões. No interior da protoboard, os contatos estão ligados verticalmente. (MCROBERTS, 2011).
- Fio jumper: esses fios conectam os componentes entre si, no barramento de alimentação e aterramento e com as plataformas de computação. Eles estão disponíveis em duas versões: as duas conexões do tipo “*macho*” (M2M), as duas conexões do tipo “*fêmea*” (F2F) e uma conexão do tipo “*macho*” e outra do tipo “*fêmea*” (M2F). As portas de entrada e saída (GPIO) do Raspberry Pi são conectadas através de jumpers com extremidades do tipo *fêmea*, já as portas do Arduino são conectadas através de jumpers com extremidades do tipo *macho*. As protoboards recebem apenas fio jumpers com extremidades do tipo “*macho*” (HALFACREE, 2018).

Gosai e Bhavsar (2016) desenvolveram um sistema de aquisição de dados para medir a temperatura de corte no torneamento de aço EN36 com ferramenta de metal duro revestida utilizando para a medição de temperatura um termopar de baixo custo do tipo K, com faixa de medição entre -200°C e +1200°C e um amplificador de sinal conectado à uma porta analógica do Arduino Uno, o qual estava ligado à um microcomputador (Figura 25). Porém, o processamento dos dados é realizado através do programa de computador proprietário LabView – o qual demanda alto custo de aquisição e os dados adquiridos são armazenados numa base de dados.

Um dos objetivos do estudo era o ajuste dos parâmetros de corte para redução da temperatura para se obter maior tempo de vida da ferramenta e, após tratamento dos dados e otimização da aquisição e de modelos matemáticos, observou-se que o erro na medição das temperaturas foi menor do que 10% e os valores otimizados para os parâmetros de corte alcançaram a temperatura mínima com confiabilidade de 98,7%, o que, segundos os autores, é altamente aceitável.

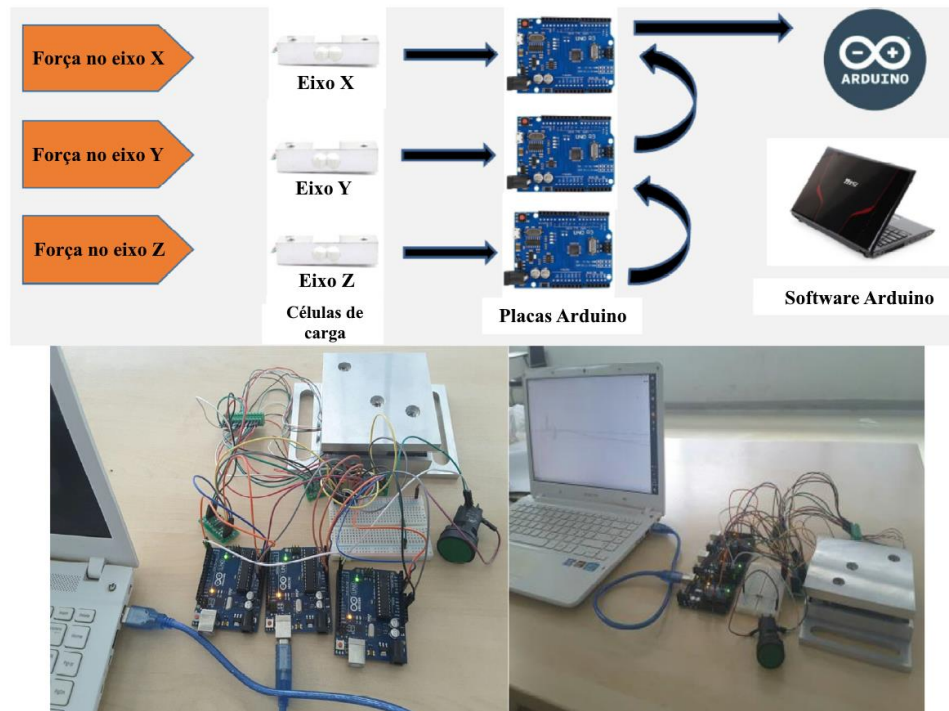
Figura 25 - Setup experimental montado para aquisição da temperatura de corte.



Fonte: Adaptado de Gosai e Bhavsar (2016, p. 315).

Por outro lado, Lapsomthop *et al.* (2019) construíram um sistema de aquisição de dados para adquirir a força aplicada em cada eixo (x, y, z) de um mini torno CNC. Foram construídas três células de carga, uma para cada eixo, sendo estas posicionadas em cada eixo para adquirir a força correspondente. Utilizou-se três placas Arduino Uno e um amplificador de célula de carga (HX711) para coletar o sinal das três células. Um Arduino Uno era conectado no computador, as outras duas placas estavam ligadas no Arduino conectado ao computador (Figura 26). Neste caso utilizou-se a própria aplicação do Arduino para salvar os dados para serem processadas. Após a verificação dos dados, observou-se que a faixa precisa de medição da força ficou entre 120 e 190N, faixa a qual obteve-se um erro de 0,048 %. Porém, como não havia sistemas de medição adequados, a análise para entender as respostas dos sensores não pode ser realizada totalmente, prejudicando a validação do sistema. Caso esses equipamentos estivessem disponíveis, o sistema poderia ser ajustado para uma maior precisão, entretanto o sistema apresentou alto desempenho, além de um desenho compacto, leve e de fácil utilização.

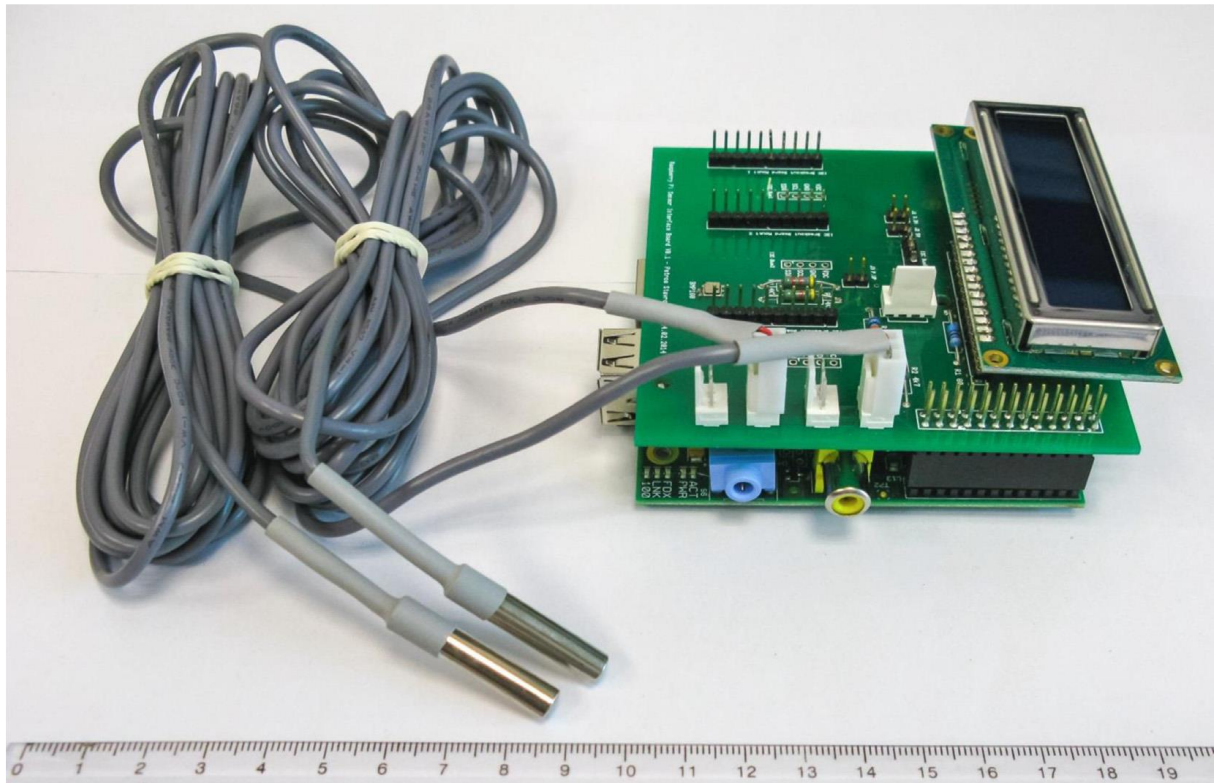
Figura 26 - Modelo esquemático do monitoramento das forças lidas pelas células de carga.



Fonte: Adaptado de Lapsomthop *et al.* (2019, p. 1934).

Por fim, Lewis *et al.* (2016) desenvolveram um sistema para aquisição de dados do ambiente de um laboratório (Figura 27): temperatura do ar, pressão e umidade relativa do ar. O sistema foi calibrado em laboratório obedecendo padronização internacional. Para a aquisição e processamento dos dados utilizou-se o Raspberry Pi Model B, uma placa PCB com visor LCD acoplado para leitura imediata das medições, para medir a temperatura do ar utilizou-se o sensor de temperatura do tipo sonda DS18B20 – cuja faixa de medição varia entre -55°C e $+125^{\circ}\text{C}$, para mensurar a pressão utilizou-se o sensor de pressão da marca Bosch, modelo BMP080 – cuja faixa de medição varia entre 300 hPa e 1000 hPa e para adquirir dados de umidade do ar utilizou-se o sensor de umidade SHT75 – possui resolução de 0,05 % UR. A fim de se garantir a estabilidade e confiabilidade do sistema, ele operou continuamente por aproximadamente seis meses, onde os desvios da leitura de temperatura não foram maiores do que $0,1^{\circ}\text{C}$. Logo, os equipamentos de baixo custo, os quais passem por um processo de calibração e validação, podem ser utilizados como uma alternativa relativamente confiável se comparadas a outros sistemas comercialmente estabelecidos, mas que demandam alto investimento de aquisição, manutenção e treinamento.

Figura 27 - Sistema de aquisição de dados de baixo custo, com Raspberry Pi, uma placa personalizada logo acima e acoplada a ela os sensores e o display LCD. Logo abaixo, uma régua para demonstrar o desenho compacto da solução.



Fonte: Lewis *et al.* (2016, p. 232).

3 MATERIAL E MÉTODOS

O trabalho foi dividido em dois segmentos: desenvolvimento da plataforma computacional para a aquisição de dados e os ensaios de usinagem para calibração e validação do sistema de aquisição de dados proposto.

Os procedimentos para o desenvolvimento da plataforma e os ensaios estão descritos nas seções posteriores.

3.1 PLATAFORMA DE AQUISIÇÃO DE DADOS

Conforme a proposta do trabalho, para o desenvolvimento da plataforma de aquisição de dados (PAD) utilizou-se materiais e ferramentas que acarretassem o menor custo possível, sem prejudicar a qualidade da plataforma, conforme apresentado na Tabela 1.

Tabela 1 - Componentes utilizados no protótipo e seus respectivos valores em dólares americanos.

Componente	Valor (\$)
Raspberry Pi 3 Model B	60.00
Fonte 12V chaveada	6.00
Sensor analógico MAX9814	4.50
Sensor GY-521	4.50
Conversor AD MCP3008	6.00
Matriz de contato com 830 pontos	3.00
Matriz de contato com 170 pontos	0.75
Jumpers e resistores	0.50
Total	85.25

Fonte: Elaborado pelo autor.

3.1.1 Sensores de baixo custo

Diante do escopo oferecido pelo presente trabalho, optou-se pela utilização de sensores cujo valor de aquisição não superasse a casa dos 10 (dez) dólares americanos, encontrando-se alternativas no mercado compatíveis com esse orçamento, as quais serão apresentadas nas

seções a seguir.

3.1.1.1 Sensor para captação da emissão acústica

Utilizou-se o sensor analógico MAX9814 (Figura 28) para a captação da emissão acústica, cujo valor de aquisição foi de \$4.50 (quatro dólares americanos e cinquenta centavos). Este sensor possui controle automático de ganho automático (AGC), possui resposta entre as frequências de 20Hz a 20 KHz e trabalha em temperaturas entre -40 e +85°C. Este sensor é ligado em um canal do conversor de sinais analógicos para digitais.

Figura 28 – Sensor para captação da emissão acústica – MAX9814.

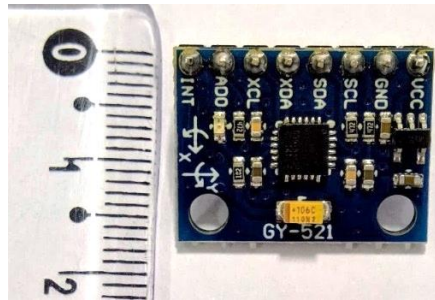


Fonte: Elaborado pelo autor.

3.1.1.2 Sensor de vibração

Para analisar a vibração optou-se por um sensor analógico do tipo giroscópio e acelerômetro, cujo valor de aquisição foi de \$4.50 (quatro dólares americanos e cinquenta centavos). Através do giroscópio obtém-se os dados de orientação, direção, movimento angular e rotação; possibilitando assim a aquisição dos dados de deslocamento de um plano nos três eixos (x, y e z), conseguindo-se maior detalhamento da movimentação de um objeto num plano. O sensor é denominado GY-521 (Figura 29), mas o chip responsável pelo processamento das informações é o MPU6050, que utiliza o protocolo de comunicação de baixa velocidade I2C. Logo, os dados de vibração nos ensaios de usinagem serão viabilizados através da medição do deslocamento em um plano no eixo x, y ou z. Este sensor é ligado diretamente numa porta de entrada digital do Raspberry Pi.

Figura 29 – Sensor GY-521 – Acelerômetro e Giroscópio.



Fonte: Elaborado pelo autor.

3.1.2 Conversor Analógico-Digital (ADC) de 8 canais

Alguns dos sensores apresentados nas seções anteriores são do tipo analógico, ou seja, eles fornecem uma saída de valores numa faixa de valores que varia de acordo com a sua resolução em bits, por exemplo, para sensores com resolução de 10 bits a resolução é de 2^{10} que é igual a 1024 pontos.

Figura 30 – Conversor de sinais analógicos para digitais de 8 canais com resolução de 10 bits - MCP3008

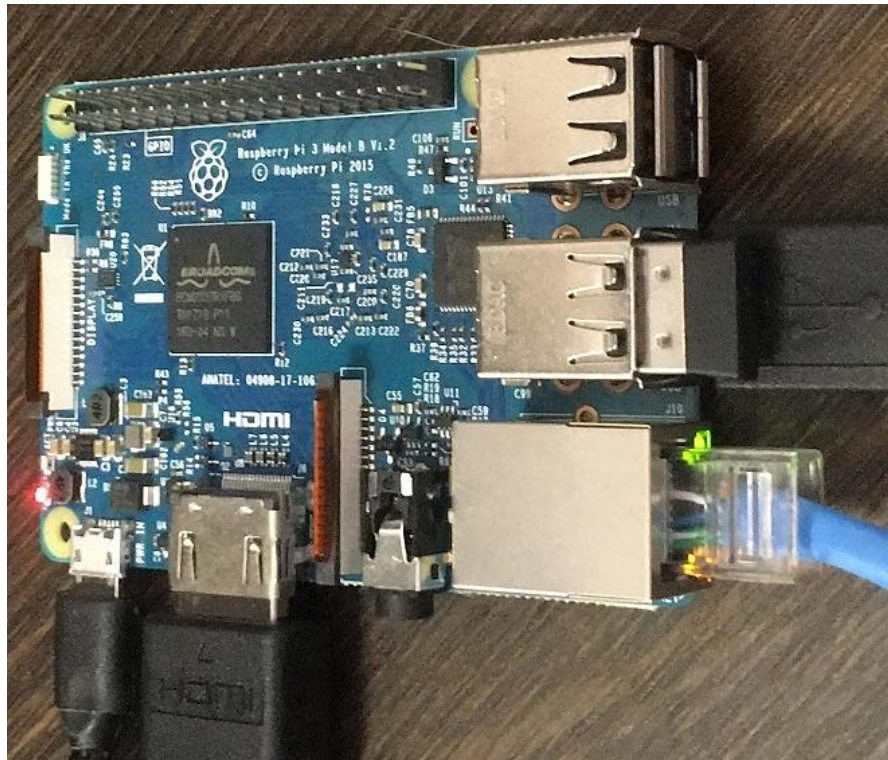


Fonte: Elaborado pelo autor.

Logo, a utilização de um conversor analógico-digital neste trabalho deve-se à característica do Raspberry Pi – utilizado como central de processamento dos dados da aquisição, que possui apenas entradas e saídas digitais através do GPIO, mas são necessárias entradas e saídas analógicas para leitura dos sensores.

O conversor analógico-digital selecionado é o MCP3008, que possui oito canais e 10 bits de resolução, comunicando-se através do protocolo SPI, que é suportado pelo Raspberry Pi após a instalação e configuração de bibliotecas específicas para este fim. O valor de aquisição deste componente ficou em torno de \$6 (seis dólares americanos). Este conversor analógico-digital é ligado com o Raspberry Pi com jumpers de acordo com o diagrama apresentado na Figura 31.

Figura 32 – Vista superior do Raspberry Pi 3 Model B na sua configuração, com cabo de energia, monitor, mouse, teclado e cabo de rede conectados à placa.



Fonte: Elaborado pelo autor.

3.1.3.1 Instalação e configuração inicial do Raspberry Pi

O Raspberry Pi 3 Model B não possui memória de armazenamento incorporada, sendo assim para o armazenamento de dados não voláteis utilizou-se um cartão de memória do tipo micro SD com capacidade de 32GB, onde instalou-se neste cartão o Sistema Operacional (SO) Raspbian, na época identificado desta maneira, sendo que atualmente o sistema é denominado Raspbian OS. Este sistema é derivado do Sistema Operacional Debian, distribuição Linux composta inteiramente de programa de computador livre. Cabe salientar que há diferentes Sistemas Operacionais a serem utilizados com o Raspberry Pi, entretanto optou-se pelo Raspbian OS.

A instalação deste sistema operacional foi possível com a utilização de um microcomputador para o download do arquivo ISO de instalação deste sistema, onde com o auxílio do programa de computador *balenaEtcher* configurou-se os arquivos de forma a serem identificados pelo Raspberry Pi quando iniciado e proceder no processo de instalação do sistema, para que este seja configurado adequadamente e funcione de forma efetiva junto ao Raspberry Pi.

Este computador pequeno possui 04 (quatro) portas USB que permite a conexão de dispositivos de entrada, como teclado e mouse. Possui ainda uma saída de vídeo no padrão HDMI, que permite a visualização e gerenciamento do sistema instalado no pequeno computador com a utilização de monitores. Por fim, possui 03 (quatro) formas de comunicação com dispositivos exteriores, sendo eles:

- comunicação sem fio de baixa potência (Wi-Fi), utilizado para acessar dispositivos dentro de uma rede através de ondas de rádio;
- comunicação sem fio utilizando a tecnologia Bluetooth, que possui alcance máximo de 15 (quinze) metros e é utilizado para acessar dispositivos após o pareamento entre eles e o pequeno computador;
- comunicação através de cabos de rede, utilizando o padrão Fast Ethernet.

É importante salientar essa capacidade de comunicação com dispositivos exteriores nativa do Raspberry Pi, pois esta permite a constante comunicação do computador com a Internet, importante para se manter atualizado o sistema com os últimos pacotes de segurança e de funcionalidades de diversos serviços que são utilizados pelo Raspbian OS. Através dessa comunicação com a rede, é possível gerenciar o Raspberry Pi remotamente através de aplicações que suportam o protocolo VNC (Virtual Networking Computing), onde é possível gerenciar o Raspberry Pi com as mesmas funcionalidades disponíveis localmente. Para o presente utilizou-se a aplicação RealVNC[®], configurando-se antes o Raspberry Pi para que ele ficasse disponível na mesma rede de um outro microcomputador para o seu gerenciamento remoto.

Tão logo concluída a instalação do Raspbian OS, realizou-se a atualização do sistema operacional e procedeu-se para a instalação do Ambiente de Desenvolvimento Integrado do Python, que oferece um editor de código e interpretador responsável pela interpretação do código fornecido, visto que o Python funciona na estrutura de interpretador, onde cada vez que um código é executado ele é interpretado pelo interpretador da linguagem em que foi desenvolvido e realiza as ações programadas, sendo este processo de funcionamento diferente dos programas de computadores desenvolvidos, por exemplo, com a Linguagem C, onde o código-fonte é compilado através de um compilador, que traduz o código da linguagem de programação em um equivalente na linguagem de máquina, fornecendo um arquivo executável do programa para funcionar num determinado sistema operacional. Utilizou-se o comando `apt-get install build-essential python-dev python-smbus python-pip git -y` para instalação do kit de desenvolvimento Python.

Após a instalação do IDE Python no Raspberry Pi procedeu-se a instalação e configuração

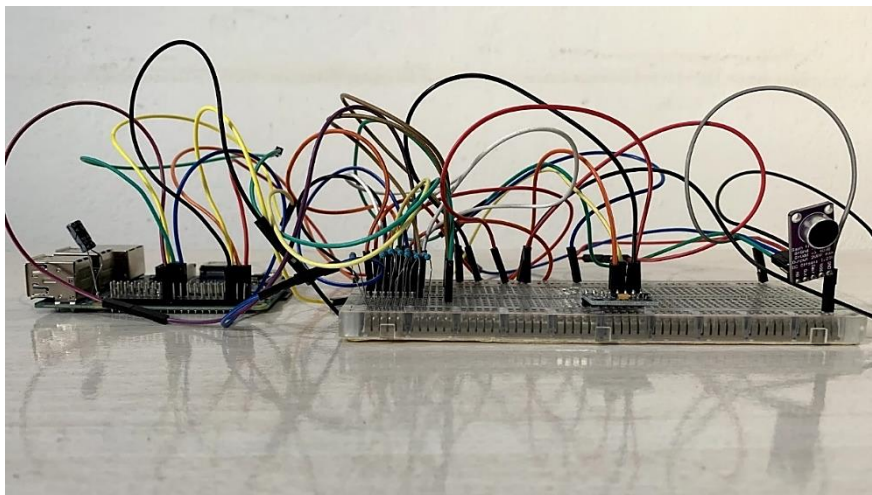
de bibliotecas que são necessárias para a comunicação de alguns módulos que virão a ser instalados no *GPIO* do Raspberry Pi com a Linguagem Python.

3.1.4 Conexão física do conversor de sinais analógicos para digitais e dos sensores no Raspberry Pi

Para que os sensores possam enviar dados ao Raspberry Pi, optou-se pela conexão física dos componentes ao Raspberry Pi, através de jumpers e protoboards. Como o projeto encontra-se em fase de prototipação, os jumpers e protoboards permitem uma flexibilidade bem maior, pois a conexão dos componentes é realizada de forma muito mais simples e sem a utilização de solda.

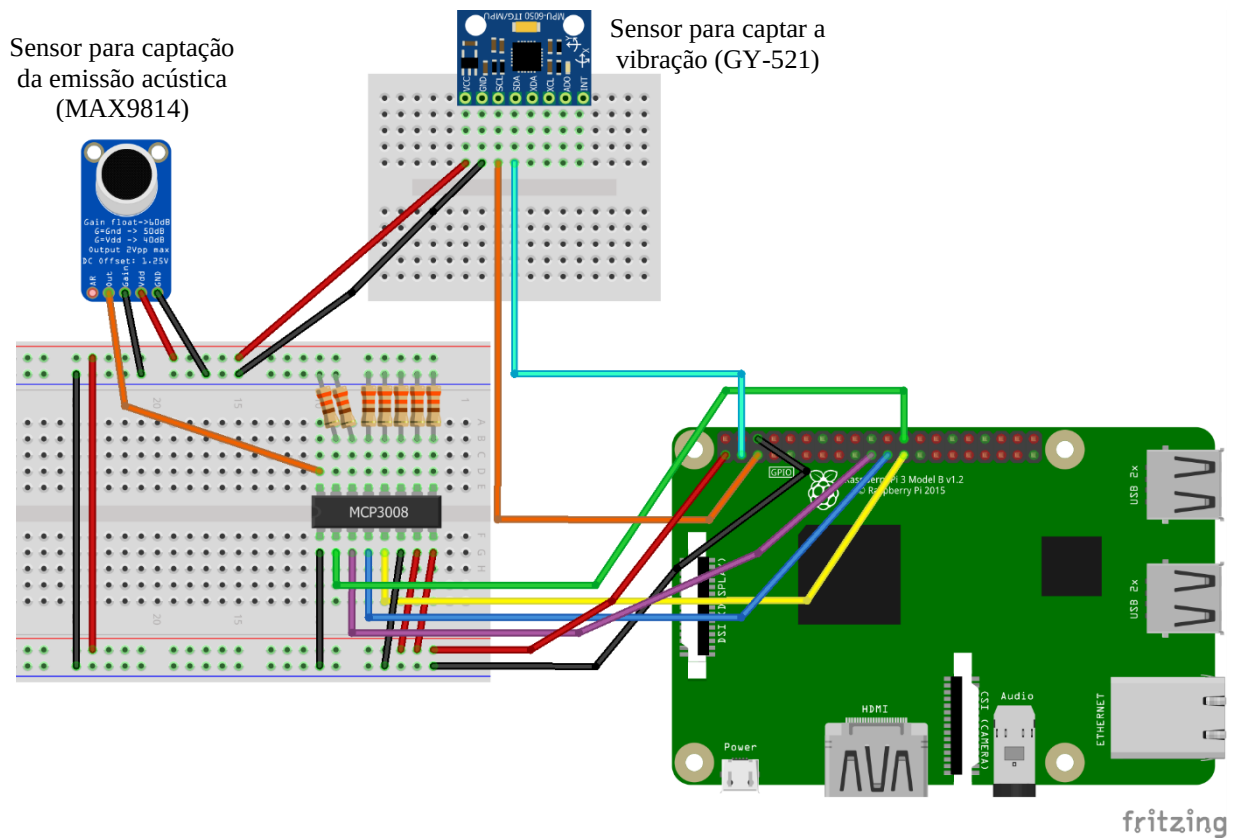
Logo, qualquer modificação que se faça necessária é rapidamente executada com a desconexão e conexão de componentes com os jumpers diretamente no Raspberry Pi ou na protoboard. Obtém-se assim, um alto nível de produtividade no desenvolvimento do sistema de aquisição de dados, haja vista que as alterações físicas são realizadas de maneira rápida e sem prejuízos aos componentes. A ligação dos sensores, resistores e do conversor de sinais analógicos para digitais na protoboard e a conexão deste com o Raspberry Pi são apresentadas na Figura 33 e Figura 34.

Figura 33 – Prototipação da plataforma de aquisição de dados de baixo custo.



Fonte: Elaborado pelo autor.

Figura 34 - Esquema de conexões do conversor de sinais analógicos para digitais e dos sensores com o Raspberry Pi.



Fonte: Elaborado pelo autor.

3.1.5 Programação dos módulos no Raspberry Pi

Após a conexão física dos entes responsáveis pela aquisição dos dados com o Raspberry Pi, faz-se necessária a programação lógica para a leitura e o processamento dos dados lidos e enviados pelos sensores ao Raspberry Pi.

O desenvolvimento ocorreu no IDE Python e no terminal Linux (também conhecido como shell), onde os comandos são informados no formato texto e após pressionar a tecla ENTER estes são executados pelo Sistema Operacional, no caso o Raspbian que é baseado numa distribuição Linux. Dependendo do comando, o SO pode ser afetado ao nível do kernel (núcleo do SO que realiza a ligação entre ele e os componentes eletrônicos do Raspberry Pi e gerenciam os recursos do sistema), por isso os comandos devem ser conferidos antes de sua execução, a fim de se evitar uma falha catastrófica no Raspberry Pi.

De início configurou-se o conversor de sinais analógicos para digitais para que este pudesse ser compreendido pelo Raspberry Pi, ativando-se o SPI executando o seguinte comando no terminal: `sudo raspi-config`, opção Advanced Options, A5 SPI e Enable SPI

Interface. Reiniciou-se o Raspberry Pi com o comando `shutdown -r now`. Após o reinício, executou-se a seguinte sequência de comandos para a instalação das bibliotecas para comunicação do conversor de sinais analógicos para digitais com o Python:

- `cd ~`
- `git clone https://github.com/adafruit/Adafruit_Python_MCP3008.git`
- `cd Adafruit_Python_MCP3008`
- `sudo python setup.py install`
- `sudo pip install adafruit-mcp3008`

Posteriormente realizou a ativação do I2C para utilização do sensor de vibração executando-se o mesmo comando anterior no terminal: `sudo raspi-config`, opção Advanced Options, A6 I2C e Enable I2C. Reiniciou-se novamente o Raspberry Pi.

Para verificar o funcionamento dos canais do conversor de sinais analógicos para digitais, desenvolveu-se um módulo de testes (*mcp_test.py*), o qual a cada ciclo de tempo apresentava em tela a leitura de cada canal. Após esta verificação, acrescentou-se neste módulo de testes o código para a leitura do sensor MAX9814, observando-se assim se a leitura ocorre ou não de forma adequada.

Por fim, acrescentou-se neste módulo o código relacionado à leitura dos dados recebidos pelo sensor GY-521 para observar se os dados estavam sendo recebidos de acordo com o fenômeno físico observado.

Para o monitoramento em tempo real dos dados adquiridos pelos sensores, decidiu-se pelo desenvolvimento de duas soluções:

- **monitoramento local** baseado em Python e executado no Raspberry Pi, ou seja, a visualização gráfica dos dados adquiridos é no próprio Raspberry Pi, bem como toda a carga de processamento para a renderização dos gráficos a cada intervalo de tempo. Para tal, desenvolveu-se dois módulos sob a Linguagem Python, um responsável pela aquisição de dados dos sensores e armazenamento dos dados adquiridos em um arquivo formato texto (1), denominado *aquisicao.py*. O outro módulo, também desenvolvido em Python, é o monitoramento em tempo real da aquisição (2), com exibição de gráficos com dados recuperados a cada determinado intervalo de tempo dos arquivos textos gerados pelo módulo (1), sendo que era possível pausar o monitoramento e definir o valor considerado como 0 para a geração dos gráficos, denominado *monitoramento.py*. Os dois módulos demandaram mais de 500 linhas de códigos para o seu pleno funcionamento;

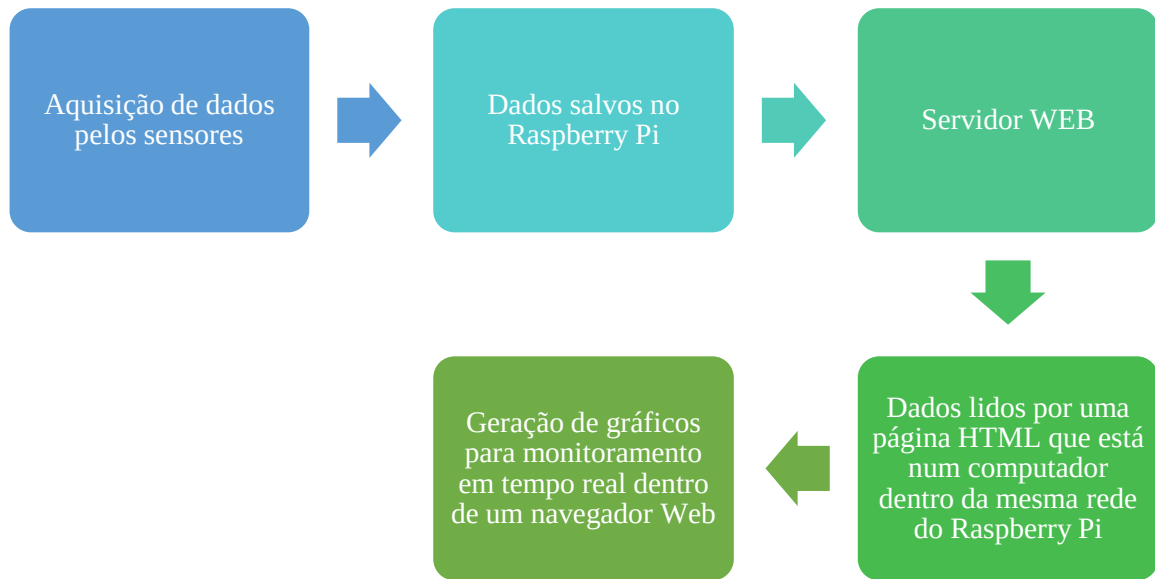
Figura 35 - Tela da aplicação para monitoramento local da aquisição de dados.



Fonte: Elaborado pelo autor.

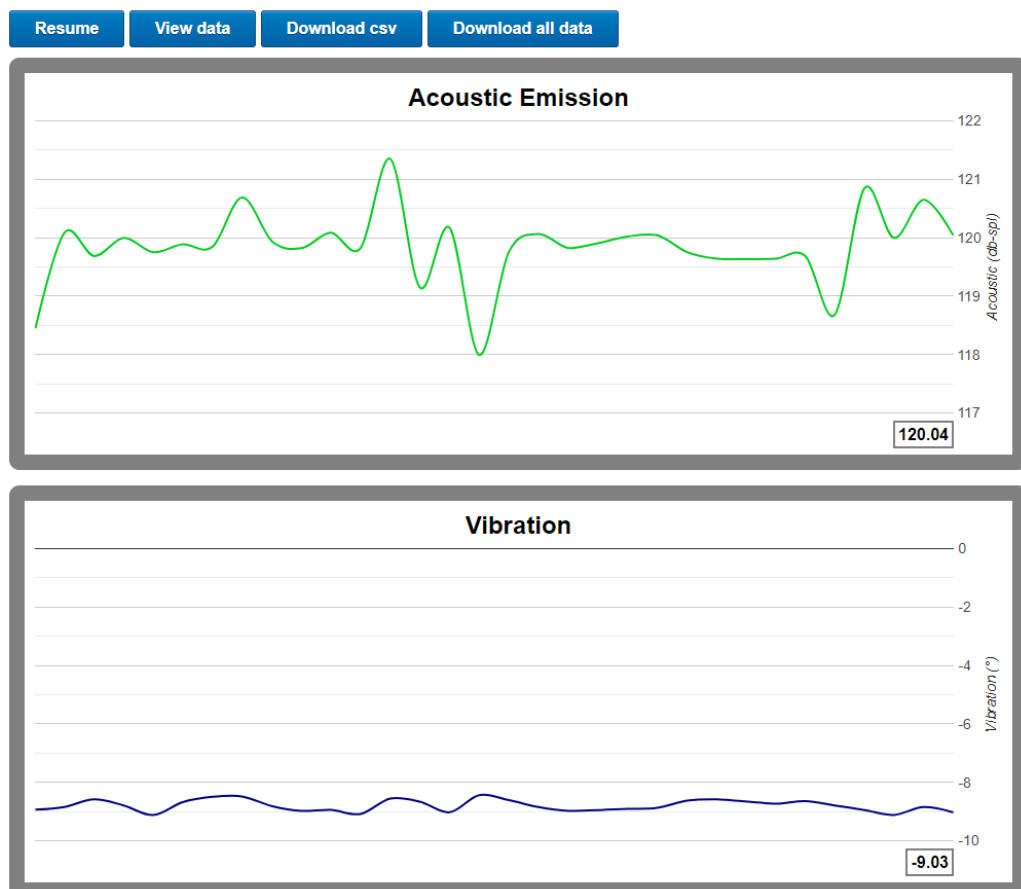
- **monitoramento remoto** (Figura 36) com o desenvolvimento de um módulo em Python para salvar os dados no Raspberry Pi em formato texto, configuração de um servidor Web no Raspberry Pi (servidor) para que uma página HTML (página de internet) (Figura 37) programada com código em Java Script possa realizar leitura dos dados e a montagem em tempo real de gráficos de acordo com esses dados – atualização de 500 milissegundos, sendo que esta página é executada num computador (cliente) dentro da mesma rede do Raspberry Pi. Esta página pode ser aberta por qualquer navegador Web que esteja instalado no computador cliente, em qualquer sistema operacional que ofereça suporte à exibição de janelas. Paralelamente, é gerado um arquivo de apoio do tipo TXT (texto puro) onde são gravados todos os pontos adquiridos, sendo este salvo dentro do Raspberry Pi e acessível pela página HTML. Os botões possuem, respectivamente, as seguintes funcionalidades: interromper e retomar o monitoramento on-line, visualizar os dados lidos para a construção dos gráficos, download dos dados em arquivo do tipo CSV (Comma-separated values) e a visualização de todos os pontos lidos pelos sensores, em arquivo do tipo TXT (texto puro) que pode ser convertido para o padrão CSV.

Figura 36 – Esquema do monitoramento remoto dos dados lidos pelo Raspberry Pi.



Fonte: Elaborado pelo autor.

Figura 37 - Tela da aplicação para monitoramento remoto da aquisição de dados.



Fonte: Elaborado pelo autor.

3.2 ENSAIOS ENVOLVENDO USINAGEM

Durante o desenvolvimento da plataforma de baixo custo, realizaram-se diversos testes que não envolveram usinagem, apenas simulação de vibração da superfície e emissão acústica e sonora para verificação da aquisição dos dados a partir dos sensores.

Entretanto, dois estudos com usinagem de materiais foram realizados em diferentes fases do desenvolvimento da plataforma de aquisição, principalmente no que se refere à maturidade da aplicação para aquisição e monitoramento.

Logo, os experimentos da plataforma de aquisição de baixo custo envolvendo usinagem foram nomeados da seguinte forma: Estudo A e Estudo B.

3.3 ESTUDO A

Para verificação inicial do funcionamento da plataforma de aquisição de dados de baixo custo utilizou-se uma lixadeira. Entretanto, os experimentos focaram na análise do funcionamento dos sensores, se eles estavam apresentando graficamente o fenômeno físico observado.

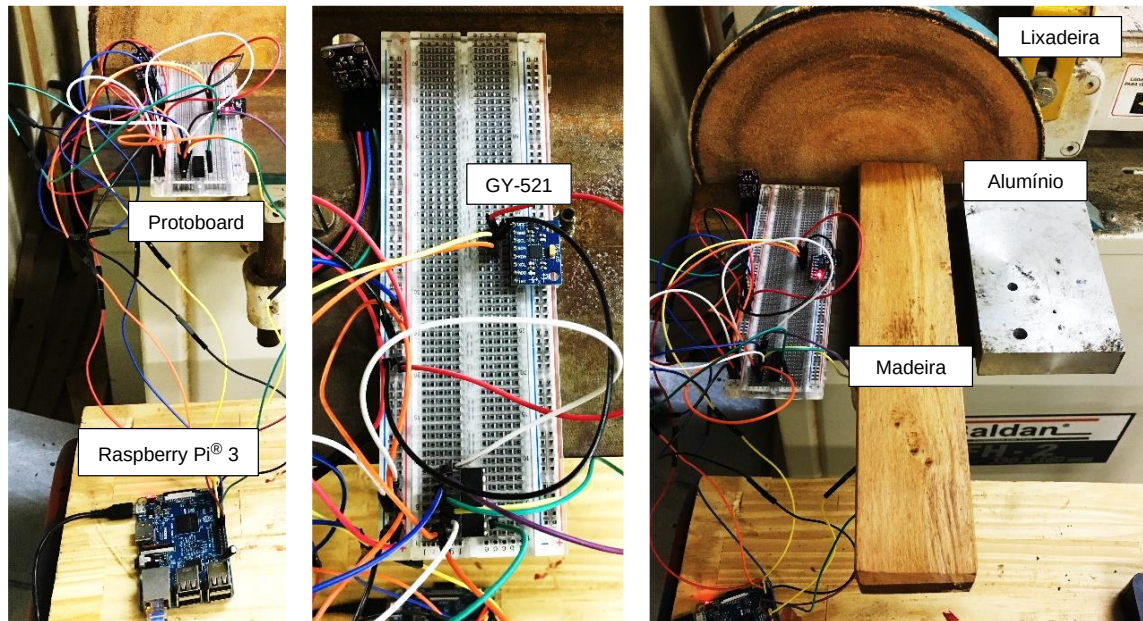
Os ensaios foram realizados no Laboratório de Metalurgia da Etec Dr. Demétrio Azevedo Júnior, em Itapeva/SP. Utilizou-se uma lixadeira da marca Baldan, modelo LFH-2, uma lixa 120, uma amostra de madeira (*Hymenolobium sp*) e uma amostra de alumínio na forma de prisma quadrangular. Cada peça foi lixada 3 (três) vezes – totalizando 6 (seis) ensaios, a fim de apenas se analisar o comportamento do sistema de aquisição, sem compromisso direto com os valores adquiridos.

O ambiente para realização dos ensaios (Figura 38) foi arranjado de forma a comportar os sensores na mesa, além de se dedicar uma mesa para acomodação do Raspberry Pi e do roteador Wi-Fi necessário para conexão ao Raspberry Pi via VNC em um microcomputador na mesma rede, para controlar a aquisição.

O microcomputador utilizado é um notebook da marca Dell com a seguinte configuração: Processador Intel® Core™ i7-8750H CPU @ 2.20GHz, 8 GB DDR4 2.667 MHz de memória RAM. O roteador utilizado é da marca Cisco, modelo RV180W.

O monitoramento da aquisição de dados deu-se na modalidade local, observando-se os gráficos gerados diretamente pela aplicação desenvolvida em Python e que estava sendo executado diretamente no Raspberry Pi, conforme apresentado pela Figura 35.

Figura 38 - Bancada criada no Estudo A para acomodar o sistema de aquisição de dados da operação de lixamento.



Fonte: Elaborado pelo autor.

3.4 ESTUDO B

Para verificar o pleno funcionamento da plataforma de aquisição de dados de baixo custo, mesmo em condições ambientais severas, optou-se pela utilização da plataforma próxima à área de usinagem sem qualquer tipo de proteção, ou seja, todos os sensores, módulos, fiação, protoboards e o Raspberry Pi estavam expostos à poeira, ao cavaco e a outros tipos de resíduos decorrentes da usinagem e do ambiente, conforme demonstrado.

Realizou-se o torneamento de um corpo de prova cilíndrico de alumínio bruto com diâmetro inicial de 75 mm, não havendo controle sobre o acabamento da peça após a usinagem.

Os ensaios foram realizados no Laboratório de Metalurgia da Etec Dr. Demétrio Azevedo Júnior, em Itapeva/SP.

Utilizou-se um torno convencional, fabricado pela Romi, modelo Imor Oficina 420, que possui distância entre pontas de 1.500 mm, diâmetro sobre o barramento de 420/440 mm e velocidade de 31,5 a 1.000 RPM (Figura 39). Na fixação da peça no torno utilizou-se contraponto. Para o corte, utilizou-se ferramenta com pastilha de metal duro.

Os equipamentos utilizados como suporte para o funcionamento da plataforma ficaram afastados do torno, o roteador para rede sem e com fio ficou acomodado numa cadeira à 2 metros do torno e o microcomputador utilizado para monitorar e salvar os dados adquiridos ficou numa mesa, também a 2 metros de distância do torno.

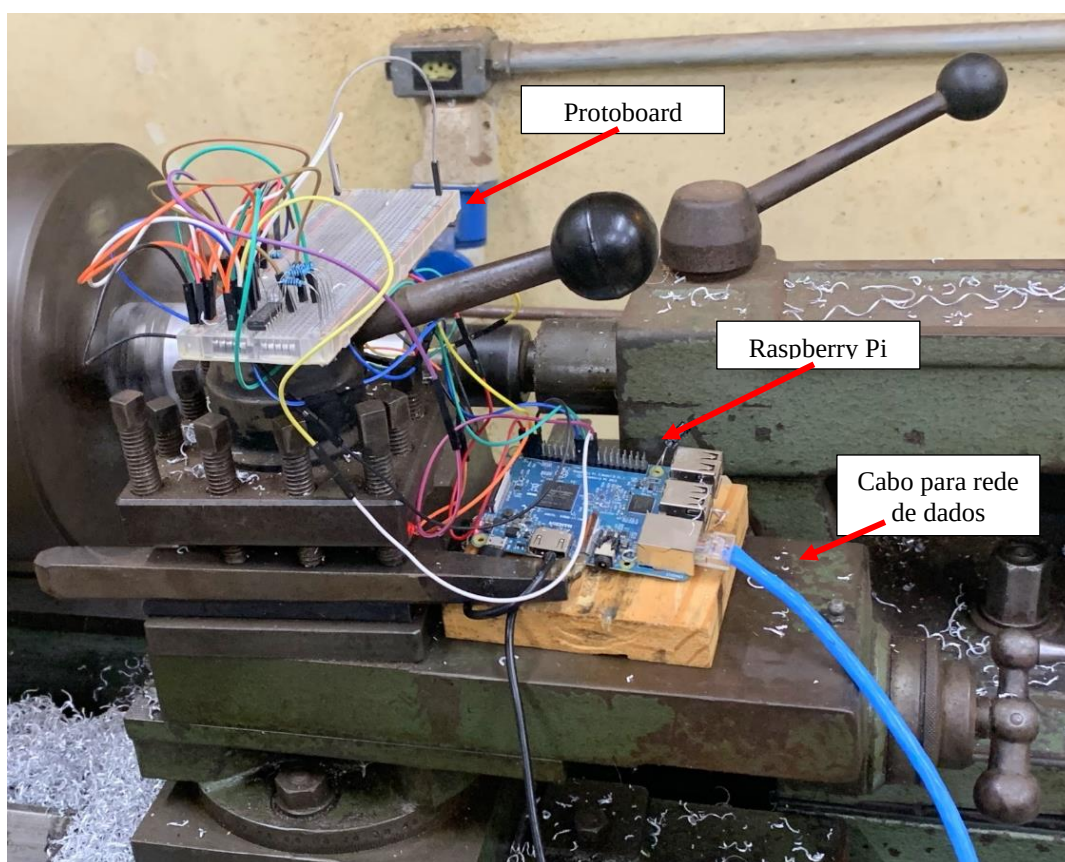
Quadro 2 - Posição dos sensores, avanço e profundidade de corte em cada ensaio do Estudo B.
(conclusão)

	10	11	12	13	14	15	16	17	18
P.S.	p	p	p	p	p	p	p	p	p
f (mm/rot)	0,271	0,271	0,271	0,197	0,197	0,197	0,145	0,145	0,145
a _p (mm)	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
	19	20	21	22	23	24	25	26	27
P.S.	p	p	p	p	p	p	p	p	p
f (mm/rot)	0,135	0,135	0,135	0,099	0,099	0,099	0,072	0,072	0,072
a _p (mm)	0,5	0,5	0,5	1,0	1,0	1,0	1,0	1,0	1,0

Legenda: P.S. – Posição do Sensor; p – próximo à ferramenta; a – afastado da ferramenta.

Fonte: Elaborado pelo autor.

Figura 40 - Bancada criada no Estudo B para acomodar o sistema de aquisição de dados da operação de torneamento, com os sensores próximos à interface ferramenta-peça e a plataforma no carro.



Fonte: Elaborado pelo autor.

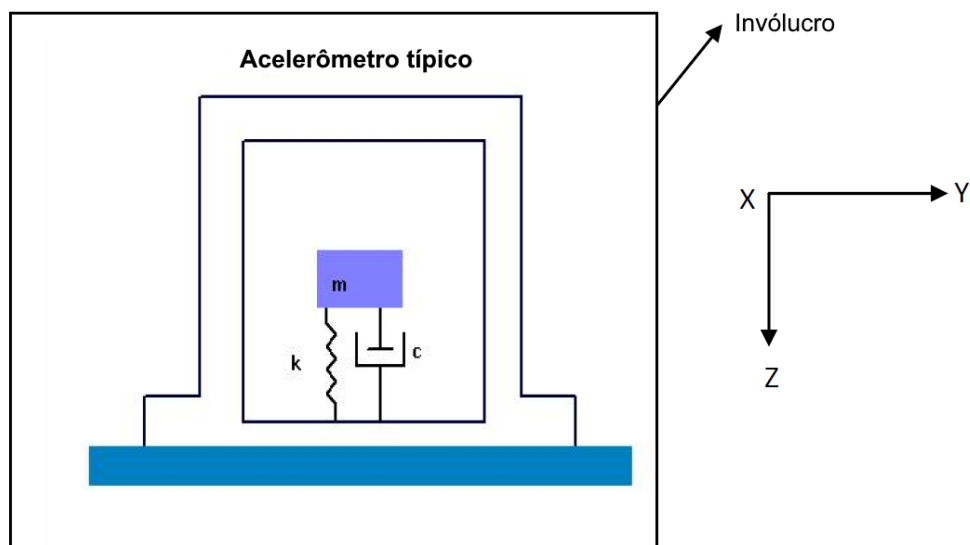
3.4.1 Métodos para medição da vibração na usinagem

De acordo com Stephenson e Agapiou (1996), vários modelos matemáticos podem ser utilizados para estudar a dinâmica na usinagem, como simulações cinemáticas e métodos por elementos finitos (MEF). Para a medição experimental, utiliza-se transdutores de vibração mecânica, sendo o mais utilizado o acelerômetro.

3.4.1.1 Acelerômetros

De acordo com Prasad (2010), o acelerômetro mede a aceleração física de um objeto ocasionadas por forças inerciais ou excitação mecânica. Conceitualmente, o acelerômetro (Figura 41) se comporta como uma massa amortecida numa mola, ou seja, quando há uma aceleração a massa é deslocada e esse deslocamento é utilizado para medir a aceleração. Para conversão do movimento em sinal elétrico, utiliza-se técnicas piezoelétricas, piezo resistivas e capacitivas. O funcionamento consiste em uma massa (m) presa a uma mola rígida (k) que por sua vez está fixada num recipiente. Um amortecedor com coeficiente de amortecimento (c) geralmente é acoplado à massa em paralelo com a mola, com o intuito de estabilizar rapidamente o sistema quando aplicada uma aceleração. Quando essa massa sofre aceleração linear, uma força (equivalente à massa multiplicada pela aceleração) atua sobre a massa de prova causando um flexionamento, que é detectado e convertido em sinal elétrico.

Figura 41 - Esquema básico de um acelerômetro.

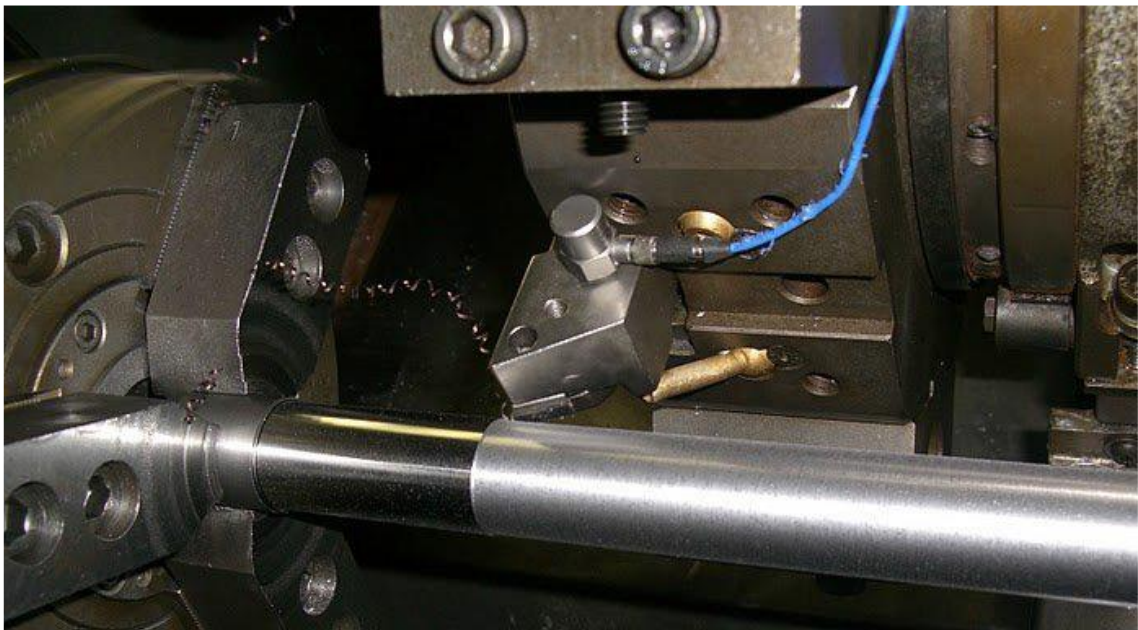


Fonte: Adaptado de Prasad (2010).

3.4.1.2 Modelo empírico para medição da vibração

Ghosh *et al.* (2018) otimizaram um modelo empírico de medição de temperatura e da vibração no torneamento duro em CNC, realizando o estudo deste torneamento com ferramentas de metal duro revestidas usinando peças de aço ANSI 4140 em três diferentes setups: variando a velocidade, avanço e profundidade de corte. Totalizou-se sessenta ensaios, sendo vinte ensaios com cada configuração. O planejamento foi baseado na metodologia de superfície de resposta (MSR), que é utilizada para planejar uma resposta quando há influência de vários fatores. Utilizou-se um acelerômetro para medir a vibração do sistema, onde processou-se os dados com a conversão de sinais analógicos em digitais utilizando a relação da série de Fourier, obtendo-se valores da frequência em hertz (Hz). Para a medição da temperatura utilizou-se um termômetro infravermelho, com valores em graus Celsius.

Figura 42 - Configuração experimental apresentando a ferramenta e o acelerômetro.

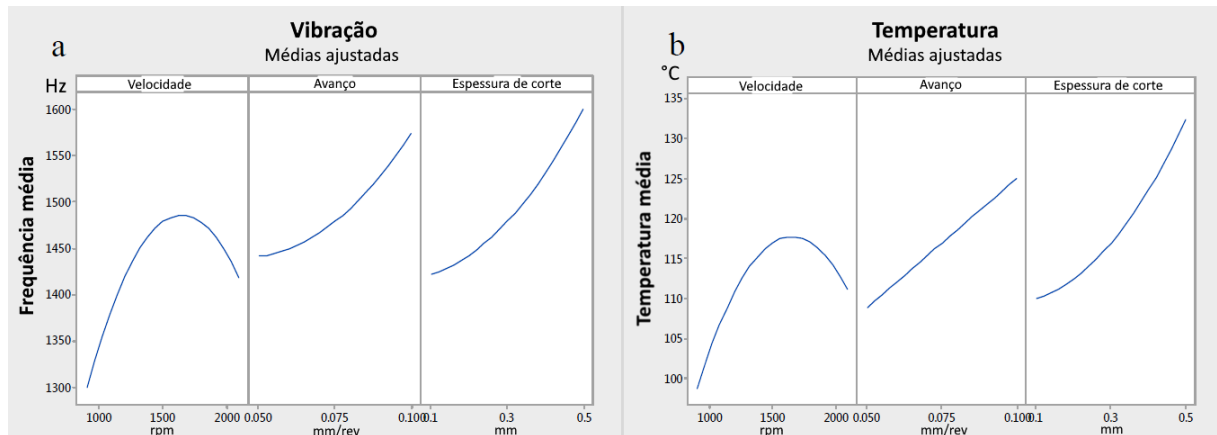


Fonte: Elangovan *et al.* (2015, p. 284).

A regressão dos dados de vibração e temperatura foram realizados com o auxílio do programa de computador MINITAB, onde conseguiu-se equações em função da velocidade, avanço e espessura de corte, plotando-se gráficos (Figura 43) das médias dos valores de vibração e temperatura em função dos parâmetros supracitados. Por fim, realizou-se um teste empírico para validar as equações, conseguindo-se valores bem próximos aos previstos pelas equações: Frequência (previsto 1360 Hz, medido 1388 Hz) e Temperatura (previsto 102 °C,

medido 103,6 °C). Por fim, a análise estatística utilizando a análise de variância (ANOVA) validou o modelo matemático para a obtenção da temperatura e vibração. A análise estatística demonstrou que a velocidade de corte possui baixa influência sobre a vibração, enquanto a espessura de corte é estatisticamente significativa na vibração e na temperatura de corte.

Figura 43 - Efeito da velocidade, avanço e espessura de corte na vibração e temperatura.

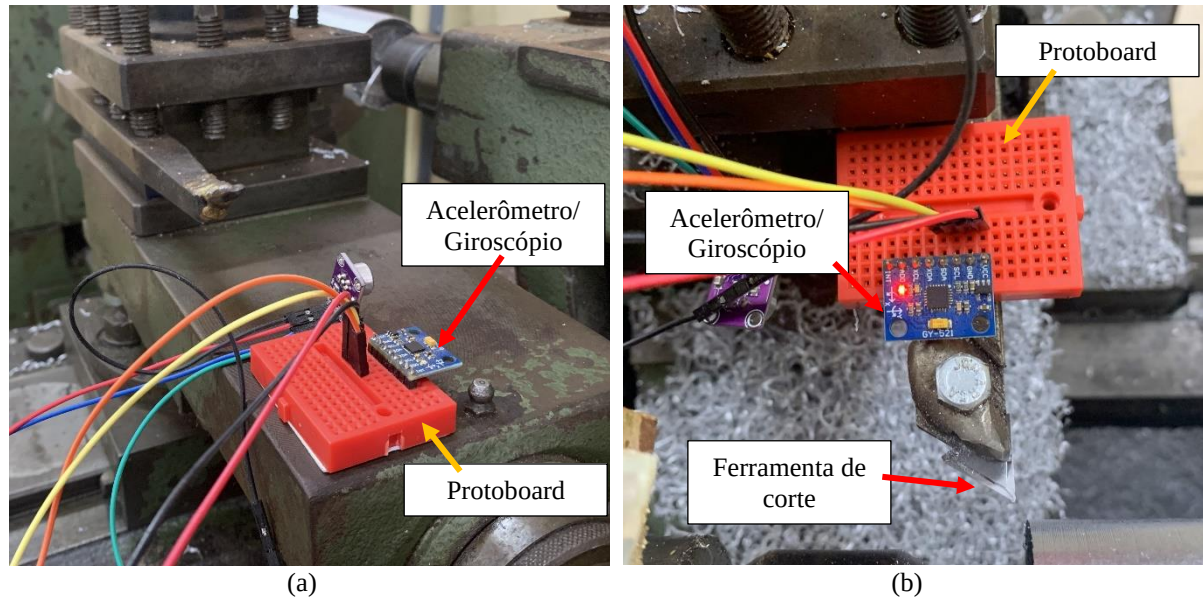


Fonte: Adaptado de Ghosh *et al.* (2018, p. 12398).

No Estudo A, para a aquisição dos dados da vibração ocasionada pelo processo de usinagem na bancada experimental, optou-se pela utilização de um acelerômetro (Figura 29) colocado próximo ao local da usinagem, conforme apresentado pela Figura 38.

No Estudo B, optou-se por fixar com adesivo do tipo dupla face a protoboard onde estava conectada o sensor de vibração a uma distância em torno de 30 cm da ferramenta de corte, no carro no mesmo lado da ferramenta (Figura 44a). Após nove ensaios, de forma semelhante ao praticado por Ghosh *et al.* (2018), resolveu-se fixar a protoboard com o sensor de vibração a aproximadamente 5 cm da ferramenta de corte (Figura 44b).

Figura 44 - Posicionamento do acelerômetro/giroscópio (a) afastado da ferramenta. (b) próximo à ferramenta. Fotos tiradas antes dos ensaios.



Fonte: Elaborado pelo autor.

Na aplicação para a aquisição dos dados desenvolvido no Python, a rotação em graus no eixo Y do sensor GY-521, que caracteriza o deslocamento do sensor causado pela vibração do sistema, é calculado conforme a Figura 45.

Figura 45 - Fragmento de código da aplicação para monitoramento local da aquisição, responsável por calcular a rotação em Y do sensor GY-521.

```

radians = math.atan2(x, dist(y,z))
return -math.degrees(radians)

```

Fonte: Elaborado pelo autor.

3.4.2 Medição da emissão acústica e sonora na usinagem

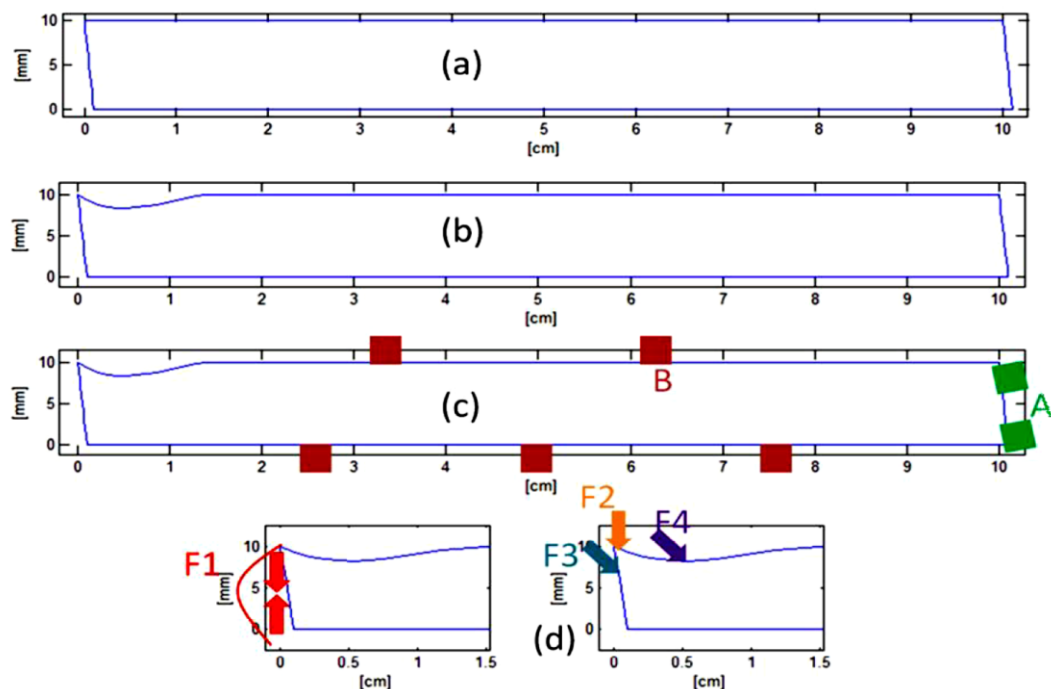
Diversos métodos para o estudo da emissão acústica na usinagem são aplicados, principalmente com auxílio da computação e eletrônica, inserindo-se sensores e sistemas de aquisição de dados para capturar, filtrar e tratar esses sinais. A abordagem da origem e propagação das ondas é diversa, podendo-se focar no comportamento da emissão acústica em ferramenta de corte, peça, cavaco, máquina e na interação entre essas interfaces.

Papacharalampopoulos *et al.* (2013) propuseram o estudo da propagação das ondas acústica em ferramentas de corte, a fim de se entender como a emissão acústica é afetada pela geometria da ferramenta. Para tal, utilizou-se simulação bidimensional do Método dos

Elementos de Contorno (BEM, em inglês), devido à sua alta precisão e exigência apenas da malha de limite. A geometria para uma ferramenta nova e desgastada é apresentada na Figura 46. Após a realização das simulações, observou-se que o comprimento de onda deve ser comparável ao tamanho do desgaste para que seja visível alguma alteração na onda propagada, sendo aparente na frequência de 800 Hz para o aço, onde os comprimentos de onda foram de 4 e 7 mm, comparáveis ao tamanho do desgaste: 2 mm de profundidade e 10 mm de largura. Quanto mais rígido o material, menor deslocamento é observado. Por fim, nomeou-se as origens observadas que afetam a propagação das ondas de emissão acústica de acordo com o material e a geometria da ferramenta:

- excitação – altera a fase de espectro do sinal da emissão acústica;
- fixação – faz com que as maiores frequências sejam dominantes na emissão acústica;
- microestrutura – reduz a faixa do espectro da emissão acústica;
- desgaste – altera a distribuição de energia nas frequências dominantes para frequências relativamente baixas.

Figura 46 - (a) Geometria da ferramenta nova. (b) Geometria da ferramenta desgastada. (c) Locais de fixação. (d) Excitações (F1: Geração S0; F2: Força de corte; F3: Atrito com a peça; F4: Atrito com o cavaco).



Fonte: Papacharalampopoulos *et al.* (2013, p. 427).

Kishawy *et al.* (2018) analisaram a aplicação da emissão acústica para monitorar e controlar algumas variáveis associadas ao processo de usinagem: desgaste da ferramenta de corte, quebra e fratura de cavacos, efeito chatter e formação de aresta postiça de corte. Montou-se uma configuração experimental para a usinagem de Inconel 718. Identificou-se que a emissão acústica oferece um monitoramento efetivo de caracterização da formação e morfologia do cavaco, qualidade superficial, progressão do desgaste da ferramenta. A localização (Figura 47a) e a configuração do sensor de emissão acústica são fatores que influenciam no desempenho da técnica de monitoramento por emissão acústica.

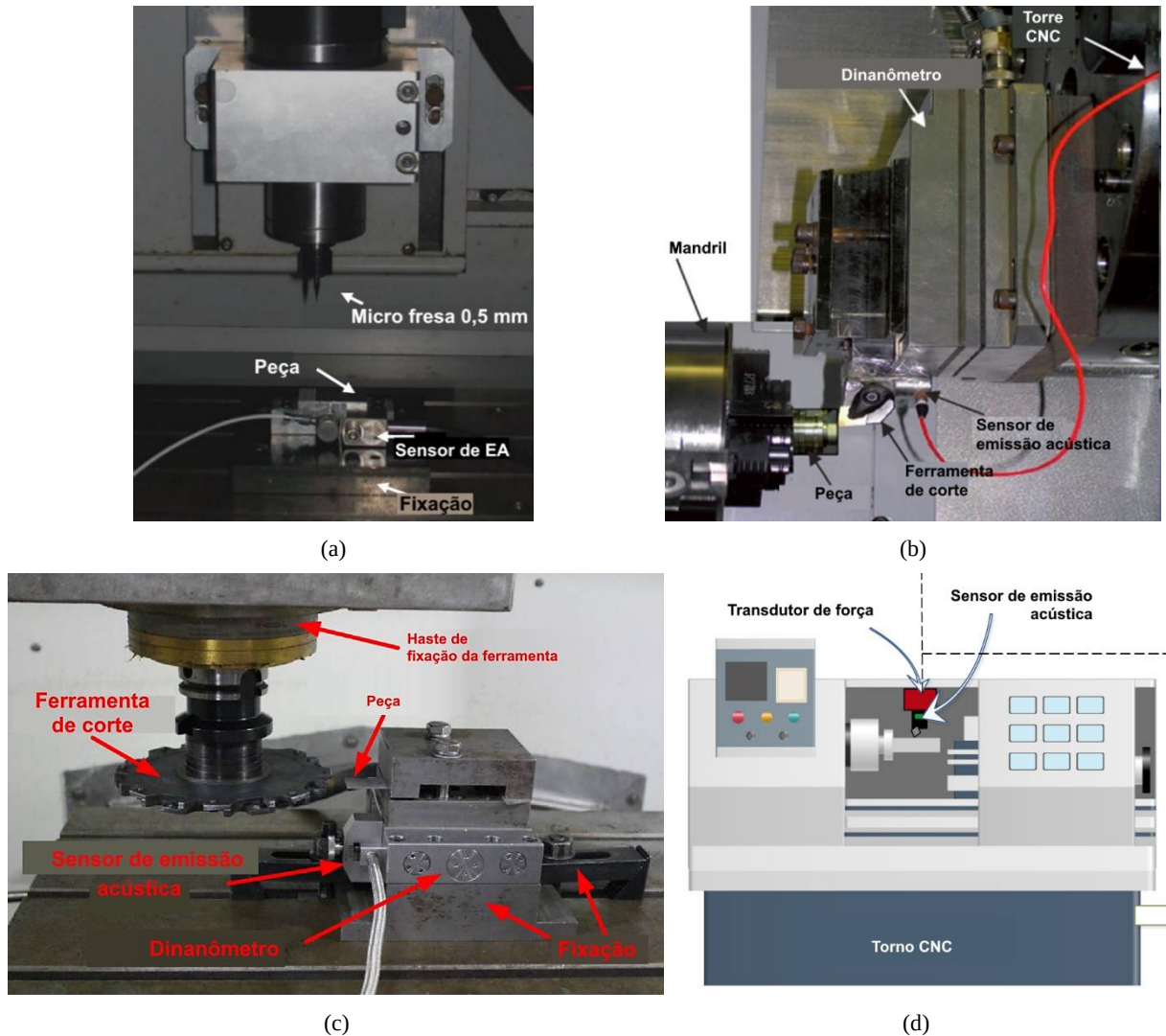
Para o estudo dos efeitos do desgaste de ferramenta na formação e morfologia do cavaco no processo de furção, utilizando emissão acústica, Prakash *et al.* (2014) utilizaram uma configuração experimental (Figura 47b) colocando-se o sensor de emissão acústica próxima à interface ferramenta/peça/cavaco, fixando-se o sensor através de rosca, como se tivesse parafusado.

Para o estudo da emissão acústica durante a formação de cavaco no processo de usinagem de alumínio e Inconel 718, Wang e Liu (2017) realizaram o estudo empírico com o posterior processamento dos sinais e análise dos dados. Para isto, construiu-se uma configuração experimental utilizando para captar a emissão acústica um sensor do tipo braçadeira magnética, que é fixada como se fosse um parafuso numa área muito próxima da interface ferramenta/peça/cavaco (Figura 47c).

Ahmed *et al.* (2019) utilizaram o mesmo sensor do tipo braçadeira magnética para captar a emissão acústica, no caso essa emissão fora utilizada para monitorar a formação de aresta postiça de corte no torneamento de aço, situando o sensor (Figura 47d) logo abaixo do transdutor de força e acima da interface ferramenta/peça/cavaco. Observou-se que o sistema de monitoramento desenvolvido pode ser utilizado para classificar a APC durante a usinagem.

Geralmente, os valores da emissão acústica são recebidos na forma de tensão, em Volts. Posteriormente, os dados são tratados e convertidos em frequência (Hz), RMS ou decibéis. A emissão acústica vem sendo utilizada com relativo sucesso na observação de diversos fenômenos relacionados ao processo de usinagem, conseguindo-se perceber, por exemplo, quando a ferramenta está danificada ou no final de sua vida. Conforme a Figura 47, observa-se a utilização de sensores sempre fixados próximos às fontes de emissão acústica, na interface ferramenta/peça/cavaco.

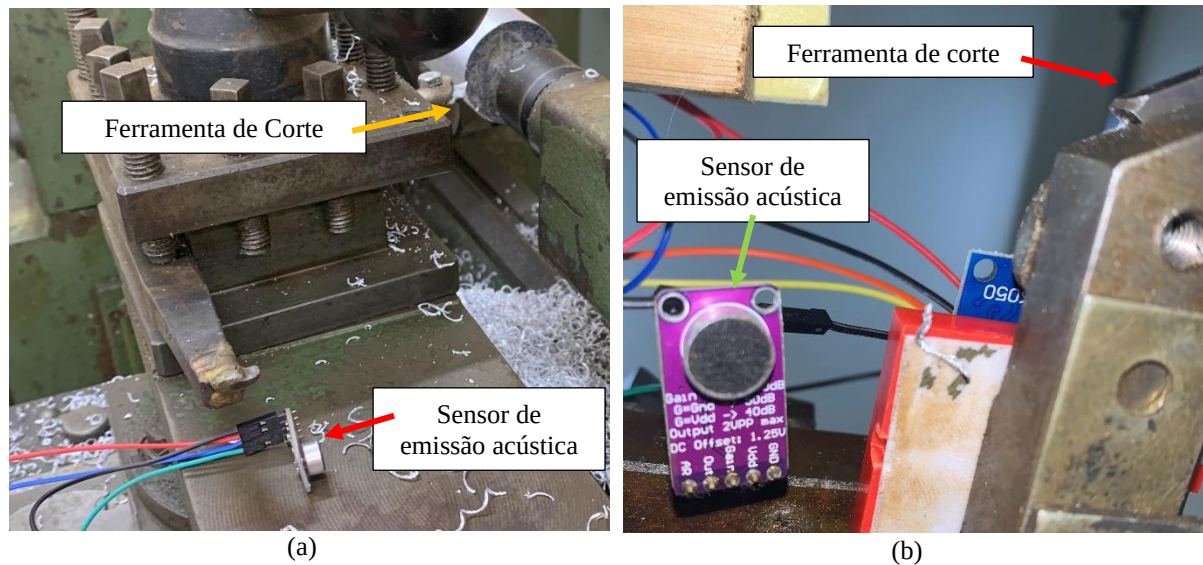
Figura 47 – Configuração montada para aquisição de dados de emissão acústica.



Fonte: (a) Kishawy *et al.* (2018, p. 1399). (b) Prakash *et al.* (2014, p. 1501). (c) Adaptado de Wang e Liu (2017, p. 115). (d) Adaptado de Ahmed *et al.* (2019, p. 2258).

No Estudo B, para a captura da emissão acústica ocasionada pelo processo de usinagem na bancada experimental, optou-se primeiramente pela captura da emissão acústica e sonora pelo método semelhante ao de Wang e Liu (2017), afastado da ferramenta de corte (Figura 48a), para depois utilizar o método semelhante ao utilizado por Kishawy *et al.* (2018), com o sensor (Figura 28) bem próximo ao local da interface entre ferramenta e peça, buscando-se assim uma maior sensibilidade da emissão acústica da usinagem (Figura 48b).

Figura 48 - Posicionamento do sensor para captação de emissão acústica (a) afastado da ferramenta. (b) próximo à ferramenta.



Fonte: Elaborado pelo autor.

Os dados foram adquiridos em volts pelo sensor de captação de emissão acústica através da função apresentada na Figura 49 e posteriormente convertida para decibéis (dB).

Figura 49 – Função responsável pela aquisição em volts dos dados do sensor MAX9814.

```

1.  def emissao_sonora(mcp, port):
2.      sampleWindow = 50
3.      sample = 0
4.
5.      startMillis = int(round(time.time() * 1000))
6.      peakToPeak = 0
7.      signalMax = 0
8.      signalMin = 1024
9.
10.     while((int(round(time.time() * 1000)) - startMillis) < sampleWindow):
11.         try:
12.             sample = mcp.read_adc(port)
13.             if(sample < 1024):
14.                 if(sample > signalMax):
15.                     signalMax = sample
16.                 elif(sample < signalMin):
17.                     signalMin = sample
18.         except Exception as e:
19.             print("Erro ao abrir sensor analógico: ", str(e))
20.
21.     peakToPeak = signalMax - signalMin
22.     volts = (peakToPeak * 3.3) / 1024
23.
24.     return volts

```

Fonte: Elaborado pelo autor.

4 RESULTADOS E DISCUSSÃO

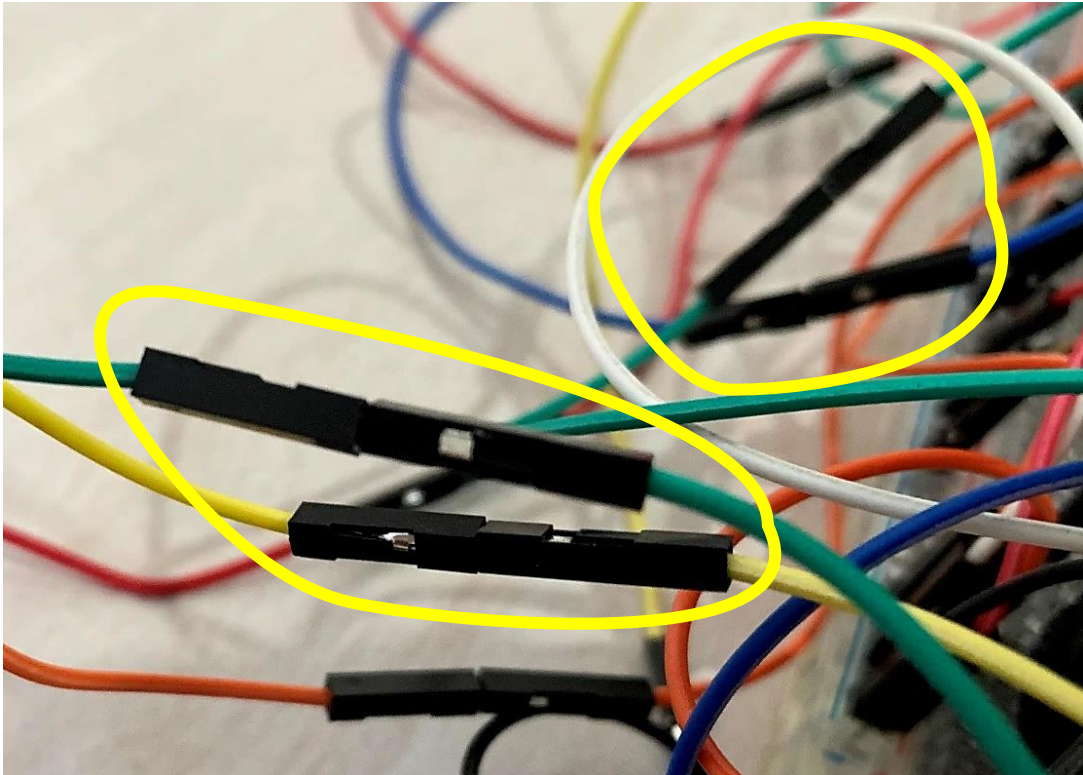
De acordo com o desenvolvimento modular das etapas do presente trabalho, opta-se por apresentar os resultados em cada etapa da construção dos módulos da plataforma de aquisição de dados e dos ensaios de usinagem.

4.1 PLATAFORMA DE AQUISIÇÃO DE DADOS DE BAIXO CUSTO

Conforme supracitado, para a ligação entre os componentes físicos da plataforma utilizou-se jumpers. Estes jumpers quando ligados de forma incorreta causam problemas de conexão, prejudicando a leitura de sensores e o desempenho da plataforma em si.

Verificou-se diversos problemas relacionados à má conexão de componentes via jumpers, principalmente quando houve a necessidade de estender o comprimento da conexão entre o sensor e a protoboard, com a ligação entre jumpers com conexão macho e fêmea, conforme ilustrado na Figura 50. Tais ocorrências tão logo observadas foram prontamente sanadas.

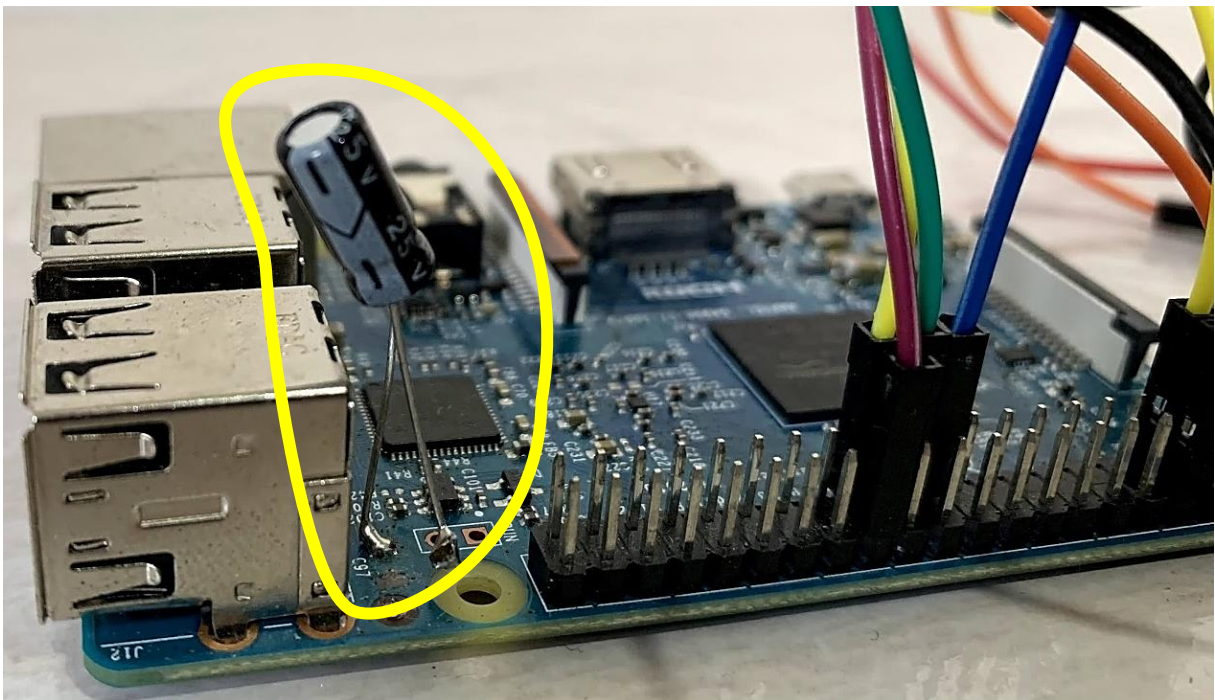
Figura 50 – Em destaque na cor amarela as conexões entre os jumpers.



Fonte: Elaborado pelo autor.

Inicialmente, projetou-se a instalação de um monitor LCD com diagonal de 3,5 polegadas incorporado ao Raspberry Pi através da porta digital de entrada e saída, sendo inclusive realizado a aquisição deste equipamento e a realização de alguns testes. Entretanto, os testes para utilização da tela de cristal líquido (LCD) acarretaram na perda de um capacitor SMD (sigla para Surface Mounting Device) relacionado à controladora das 04 portas USB, impossibilitando a utilização do Raspberry Pi até a aquisição de um capacitor eletrolítico com uma capacitância semelhante ao capacitor SMD danificado, o qual foi soldado à placa e ocorreu o retorno do funcionamento das portas USB por tempo suficiente para a configuração do acesso remoto ao Raspberry Pi, sendo que a porta Ethernet está configurada com IP dinâmico, ou seja, um dispositivo (denominado de servidor DHCP – Dynamic Host Configuration Protocol) na rede em que ele for conectado será o responsável por designar um endereço IP ao Raspberry Pi, sendo assim possível o seu acesso remoto via aplicação que suporte o protocolo VNC por outro microcomputador conectado na mesma rede.

Figura 51 – Capacitor eletrolítico utilizado para recuperar temporariamente as portas USB do Raspberry Pi.



Fonte: Elaborado pelo autor.

Porém, notou-se que os outros componentes do Raspberry Pi não foram danificados, sendo assim, o mesmo Raspberry Pi foi utilizado no desenvolvimento da plataforma para aquisição de dados.

4.1.1 Leitura dos dados dos sensores

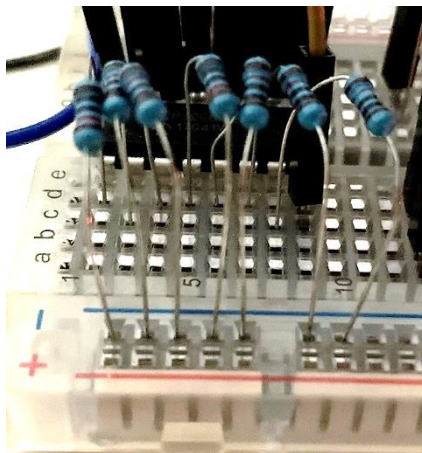
Após a certificação do pleno funcionamento da parte física da plataforma, partiu-se para a verificação dos módulos desenvolvidos para aquisição dos dados lidos pelos sensores.

Num primeiro momento, verificou-se os valores de entrada nos canais do conversor de sinais analógicos para digitais. Na primeira etapa de testes observou-se que o local onde estava instalado o conversor de sinais analógicos para digitais na protoboard apresentava uma forma diferente da comum, apresentando leve dilatação. Após a retirada do conversor de sinais analógicos para digitais constatou-se que houve um pequeno derretimento, que provavelmente foi fruto de uma ligação incorreta do conversor de sinais analógicos para digitais levando-o ao superaquecimento e provocando a dilatação térmica no local onde estava alojado na protoboard. Pautando-se pela cautela e a segurança das informações adquiridas, foi realizada a aquisição de um novo conversor de sinais analógicos para digitais, nas mesmas especificações do anterior, ou seja, o MCP3008.

Após esse contratempo, as verificações seguiram, onde constatou-se que todos os canais apresentavam leitura, mas apenas o canal 7 do conversor de sinais analógicos para digitais possuía dispositivo conectado. Partiu-se para duas hipóteses: problema no conversor de sinais analógicos para digitais ou problema com o código da aplicação que realizava o monitoramento.

Observou-se que o problema desta leitura fantasma era a ausência de resistores que fossem do tipo *pull-up* nessas entradas do conversor de sinais analógicos para digitais que não estavam ocupadas. Conectou-se um resistor com resistência nominal de 300Ω por canal que estivesse desocupado na ligação neutra de energia da protoboard (Figura 52a), mitigando o problema e fazendo com que os canais sem sensor conectado apresentassem leituras iguais 0 (zero), conforme observado na Figura 52b.

Figura 52 - (a) Ligação dos resistores *pull-up*. (b) Leitura dos canais do conversor de sinais analógicos para digitais através da aplicação desenvolvida para teste de leitura dos valores de entrada desses canais.



(a)

```

root@raspberrypi:/usr/lib/cgi-bin# python mcp_test.py
Reading MCP3008 values, press Ctrl-C to quit...

```

0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	398
0	0	0	0	0	0	0	396
0	0	0	0	0	0	0	399
0	0	0	0	0	0	0	398
0	0	0	0	0	0	0	396
0	0	0	0	0	0	0	397
0	0	0	0	0	0	0	408
0	0	0	0	0	0	0	393
0	0	0	0	0	0	0	399
0	0	0	0	0	0	0	405
0	0	0	0	0	0	0	396
0	0	0	0	0	0	0	391
0	0	0	0	0	0	0	395
0	0	0	0	0	0	0	410
0	0	0	0	0	0	0	406
0	0	0	0	0	0	0	391
0	0	0	0	0	0	0	385
0	0	0	0	0	0	0	405
0	0	0	0	0	0	0	395

(b)

Fonte: Elaborado pelo autor.

Havia um sensor conectado ao canal 7 do conversor de sinais analógicos para digitais nos testes, por isso a leitura diferente do valor 0.

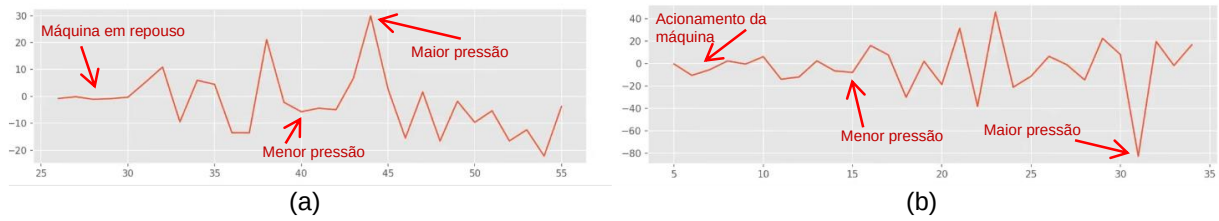
Com o agravamento da doença do coronavírus 2019 (COVID-19) e a sua declaração como pandemia mundial (WHO, 2020), surgiram as políticas de isolamento social a fim de se conter a propagação da doença (SÃO PAULO, 2020). Devido à decretação do estado de calamidade pública, o acesso às instituições e empresas foi restringido e não foi possível realizar a calibração dos sensores e a validação dos dados adquiridos.

4.2 ESTUDO A

Durante a realização dos testes para verificação do funcionamento do monitoramento local da aquisição de dados, observou-se que os dados adquiridos eram equivalentes aos fenômenos físicos aos quais os sensores eram submetidos, haja vista que o de vibração captava corretamente o deslocamento do plano quando esse era alterado através de perturbações.

Nos ensaios de usinagem, observou-se que a representação visual da vibração que estava ocorrendo na mesa durante a usinagem dos materiais era compatível com o fenômeno, tanto na usinagem da amostra composta de alumínio, quanto na usinagem da amostra de madeira conforme ilustrado na Figura 53.

Figura 53 - Gráficos gerados na aplicação, com o lixamento da madeira (a) e do alumínio (b).



Fonte: Elaborado pelo autor.

O sensor de emissão acústica apresentou funcionamento com valores discrepantes, não sendo observada a mesma variação do fenômeno acústico, sendo assim não considerado para os resultados do Estudo A. Após a investigação, percebeu-se que o seu mal funcionamento era devido ao conversor de sinais analógicos para digitais ter sofrido superaquecimento, sendo realizada a troca do conversor de sinais analógicos para digitais e após a instalação do novo conversor, o sensor de emissão acústica começou apresentar valores coerentes com o fenômeno acústico observado.

É importante pontuar que a representação gráfica ocorria simultaneamente à ocorrência do fenômeno, sem apresentar atrasos no processamento dos gráficos.

O monitoramento é iniciado após se pressionar o botão “Iniciar monitoramento” na aplicação, iniciando-se a contagem de tempo a partir deste momento. O gráfico (Figura 53) tem seu início no acionamento da lixadeira, passando pela operação de lixamento até a interrupção do lixamento, mas ainda com a máquina em funcionamento, interrompendo-se o gráfico ao pressionar o botão “Parar monitoramento”.

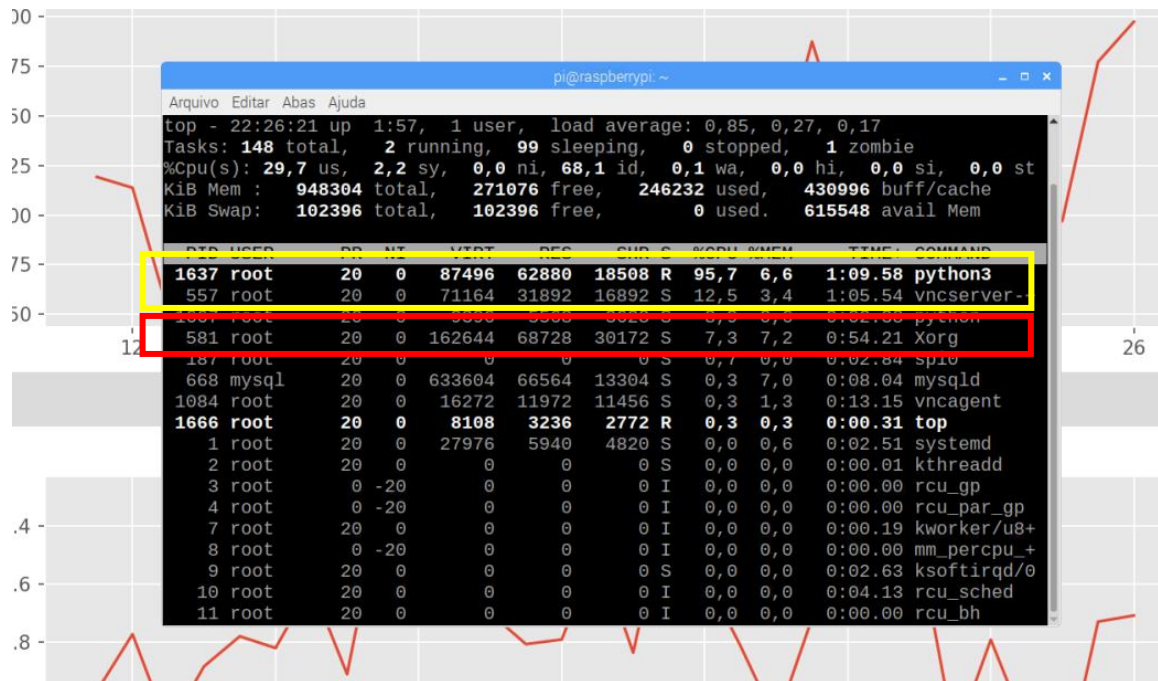
A plataforma de aquisição de dados funcionou sem proteção para os componentes eletrônicos, sendo sujeito à resíduos da usinagem, que ocorrem no lixamento sob forma sólida e em pequenas dimensões. Entretanto, o sistema não interrompeu o funcionamento por conta da entrada destes resíduos nos componentes.

4.2.1 Monitoramento local da aquisição

Com o desenvolvimento do módulo em Python para monitoramento da aquisição de dados e com a sua consequente utilização para verificação de funcionamento tanto do módulo, quanto dos sensores, observou-se um fato importante: havia um esforço grande do Raspberry Pi em processar a aquisição e a exibição dos gráficos no monitoramento, sendo possível notar lentidão no Raspberry Pi. Para a verificação do consumo de recursos computacionais utiliza-se

o comando `top` no terminal do Raspberry Pi.

Figura 54 - Consumo de recursos computacionais no Raspberry Pi ao executar a aplicação designada ao monitoramento local da aquisição de dados da plataforma.



Fonte: Elaborado pelo autor.

O consumo de recursos computacionais para apresentação dos gráficos é destacado pela cor amarela na Figura 54, onde verifica-se um consumo de processamento na ordem de 95% e a utilização de memória RAM observado foi na casa de 6,6 % (em torno de 66MB, dos 1024MB disponíveis).

Por outro lado, o módulo auxiliar para leitura dos dados adquiridos e o para armazenamento no formato texto apresentou consumo de processamento em torno de 9% e a utilização de memória RAM foi de 0,6% (em torno de 6MB, dos 1024MB disponíveis). Essas informações são destacadas na cor vermelha pela ilustração apresentada na Figura 54.

Esses valores permaneceram estáveis durante a execução dos módulos, sendo notada uma pequena variação, sempre em torno de no máximo 3%.

4.2.2 Monitoramento remoto da aquisição

Após o desenvolvimento da solução do monitoramento remoto, realizou-se testes para verificação do funcionamento da aplicação.

Durante a realização dos ensaios, confrontou-se os valores dos sensores adquiridos no

Raspberry Pi com o exibido pelo monitoramento on-line, a fim de se verificar a fidelidade dos valores lidos na placa com os valores exibidos pelos gráficos no microcomputador.

O monitoramento é iniciado ao abrir o arquivo HTML no navegador no microcomputador do cliente, não necessitando de botão para o início desta leitura. Conforme observado, o consumo de recursos computacionais pelo monitoramento é de responsabilidade do microcomputador cliente (Figura 55a) e o Raspberry Pi (Figura 55b) fica responsável pela aquisição dos dados, havendo assim um balanceamento na utilização dos recursos computacionais pelos dois dispositivos. Destaca-se que o monitoramento on-line pode funcionar em vários microcomputadores ao mesmo tempo.

Figura 55 - (a) Consumo de memória RAM (KB), processamento (%) e rede (B/s) do microcomputador cliente para exibição dos gráficos. (b) Consumo de processamento (%) e memória RAM (%) do Raspberry Pi para a aquisição dos dados.

Guia...	92.936K	9.4	1.394 B/s	%CPU %MEM	TIME+ COMMAND
				9,2 0,6	0:06.10 python

(a)

(b)

Fonte: Elaborado pelo autor.

Com relação ao monitoramento on-line, observa-se pela que a escala do gráfico se ajusta aos valores máximo e mínimo lidos, com o intuito de facilitar a visualização dos gráficos. Com o objetivo de haver melhor leitura dos gráficos, optou-se pela exibição apenas dos últimos trinta pontos, o que equivale aos últimos quinze segundos de monitoramento. Convém reforçar que todos os dados lidos pelo monitoramento podem ser acessados pelo botão “View data”.

Com relação à aquisição de dados, o sistema foi capaz de ler 100 (cem) pontos por segundo de cada sensor. O sistema de aquisição funcionou sem proteção para os componentes eletrônicos, sendo sujeito à possíveis interferências do ambiente. Entretanto, o sistema em nenhum momento cessou o funcionamento.

Esta estratégia para o monitoramento da aquisição apresentou funcionamento adequado com os valores lidos pelos sensores e exibidos no Raspberry Pi, pois como os dispositivos estão na mesma rede, o tempo de comunicação entre eles é de cerca 2 (dois) milissegundos, não havendo assim perda de dados por atraso na comunicação.

Por fim, a utilização dos recursos computacionais no Raspberry Pi foi reduzida na ordem de mais de 1.000% em relação à solução que demanda a utilização da aplicação no Raspberry Pi para o monitoramento on-line (Figura 54), ou seja, reduziu-se em mais de 10 (dez) vezes a utilização do processador, aumentando-se assim o número de pontos lidos por segundo e a

eficiência da aquisição dos dados. Com relação aos pontos lidos por segundo, cabe ressaltar a necessidade de otimizar o programa desenvolvido para a aquisição, a fim de aumentar o número de pontos lidos.

4.3 ESTUDO B

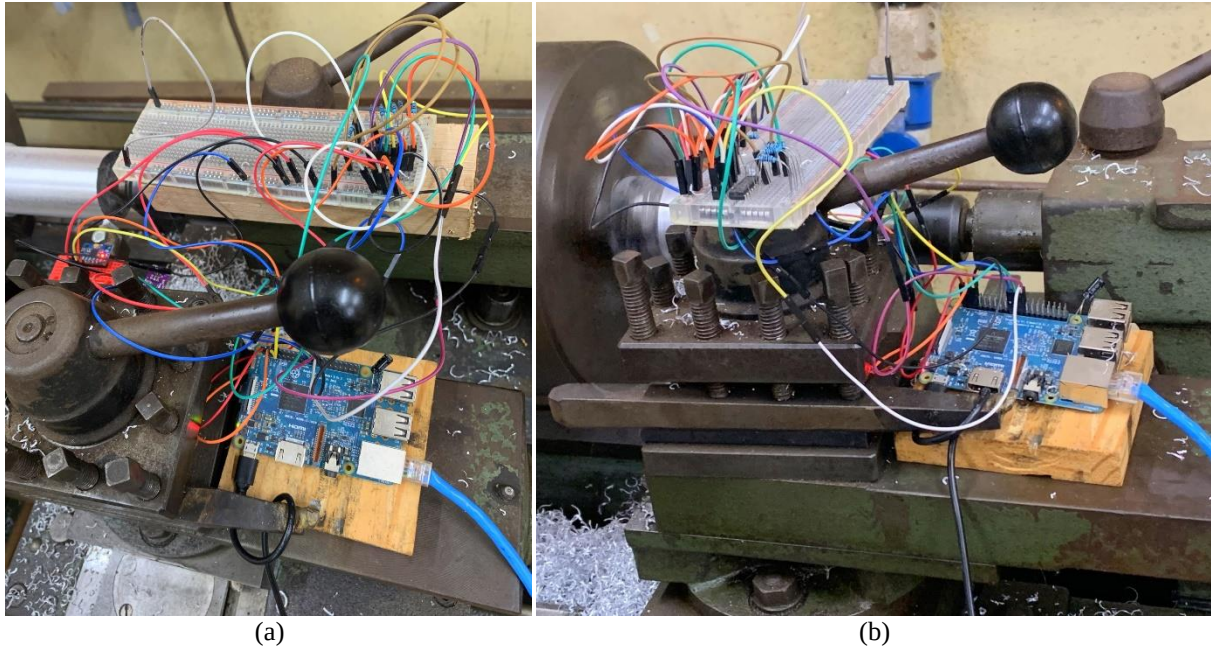
Diante do menor consumo de recursos computacionais no monitoramento remoto observado no Estudo A, optou-se por esta modalidade durante os ensaios do Estudo B.

Entretanto, observou-se que era necessária uma conexão à Internet para utilizar a biblioteca que facilita a geração dos gráficos no navegador, sendo assim o microcomputador ficou conectado via cabo com o roteador junto à plataforma de aquisição de dados e a conexão sem fio do microcomputador foi utilizada para receber a internet compartilhada por um aparelho smartphone.

Da mesma forma que no Estudo A, todos os componentes da plataforma de aquisição ficaram expostos, inclusive diretamente na interface ferramenta-peça, recebendo todo tipo de resíduos decorrentes do torneamento. Além disso, não houve tratamento com relação aos ruídos ambientais, entretanto no momento do estudo não havia alunos no prédio. Outrossim, evitou-se a produção de ruídos durante os ensaios, como conversas ou movimentação de outros equipamentos.

Quando os sensores estavam distantes da área de usinagem, o Raspberry Pi e a protoboard ficaram numa posição fixa, não se movendo junto com o carro (Figura 56a), o que obrigou a carregar manualmente esses componentes, com o risco de desligá-los dado os movimentos bruscos desta operação. Somente a partir do Ensaio 10 que houve a fixação de todos os componentes no carro (Figura 56b), não sendo mais necessário carregar os equipamentos.

Figura 56 - (a) Raspberry Pi e protoboard soltas. (b) Todos os componentes fixados no carro.



Fonte: Elaborado pelo autor.

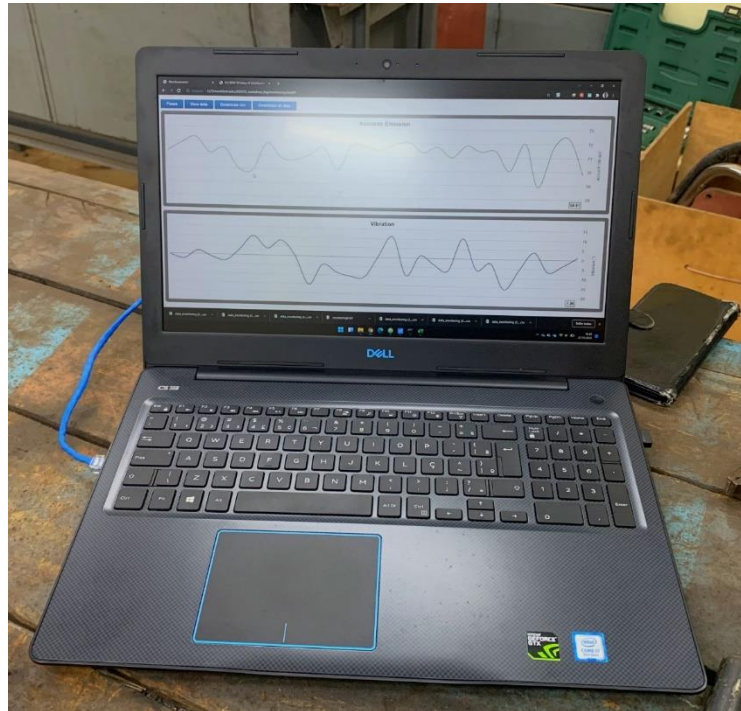
4.3.1 Aquisição dos dados

Dada a exposição dos componentes, em alguns ensaios ocorreu a paralização da aquisição, pois cavacos faziam com o que o circuito entrasse em curto e interrompessem o script em Python que realizava a leitura dos dados dos sensores. Entretanto, a paralização era temporária, bastando para retomar a aquisição reiniciar o script. Nenhum equipamento foi danificado, mesmo com a severidade das condições de trabalho dos componentes.

A comunicação do microcomputador estendeu-se para a operação no Raspberry Pi, sendo possível através de uma aplicação VNC (Virtual Network Computing – permite a visualização de interfaces remotas através de uma conexão segura), operando-se o Raspberry Pi como se ele estivesse conectado à um monitor. Em nenhum momento ocorreu a queda de conexão dos dados, tanto da internet quanto da rede utilizada para a comunicação entre o microcomputador e a plataforma de aquisição de dados.

Os dados foram monitorados em tempo real no microcomputador, que permitiu todo o controle dos dados dos ensaios e a exportação dos valores utilizados para a plotagem dos gráficos na página HTML, conforme Figura 57.

Figura 57 - Microcomputador utilizado para o monitoramento em tempo real da aquisição de dados e para operação remota do Raspberry Pi via VNC.



Fonte: Elaborado pelo autor.

Na página de internet que faz a função de interface do monitoramento remoto, foram adquiridos 6.580 pontos nos 27 ensaios, média de 244 pontos por ensaio, sendo que 3.290 pontos são do sensor de captação de emissão acústica e 3.290 pontos são do sensor de vibração, sendo estes salvos em arquivos com extensão CSV. O arquivo de texto bruto gerado com os pontos adquiridos totalizou 75.028 pontos nos 27 ensaios, sendo 37.514 pontos do sensor de captação de emissão acústica e 37.514 pontos do sensor de vibração. Em média, foram adquiridos 2.000 pontos a cada 52 segundos, ou seja, um ponto a cada 26 milissegundos. Esses dados estão na forma bruta, sem qualquer tipo de tratamento.

Os dados salvos no Raspberry Pi eram gerados a cada 1.000 iterações da aquisição de dados, ou seja, eram escritos 2.000 pontos no arquivo TXT de uma vez, pois isso reduzia a requisição ao sistema de arquivos do Raspberry Pi, aumentando-se o desempenho. Lembrando que, diferente do monitoramento remoto, não havia definição de intervalo mínimo para aquisição dos pontos.

Juntando-se todos os pontos de todos os ensaios, obtidos a partir dos arquivos CSV gerados pela página HTML e a partir dos pontos brutos gerados diretamente no arquivo TXT, sem qualquer tratamento dos dados, tem-se os seguintes gráficos de linhas em ordem cronológica de aquisição, gerados com auxílio do programa de computador Microsoft Excel.

Percebe-se pelos gráficos a mesma tendência, ainda que um pouco diferente nos gráficos gerados pelos pontos obtidos na página HTML. Os dados eram salvos da página HTML, porém continuavam a serem gravados no arquivo TXT no Raspberry Pi.

Analisando do início dos gráficos da Figura 58 até a linha vertical verde, nota-se que os valores estavam menores no início, com aumento abrupto após o quarto inicial, mantendo-se com média mais alta, mesmo com algumas variações por vezes abaixo da média inicial.

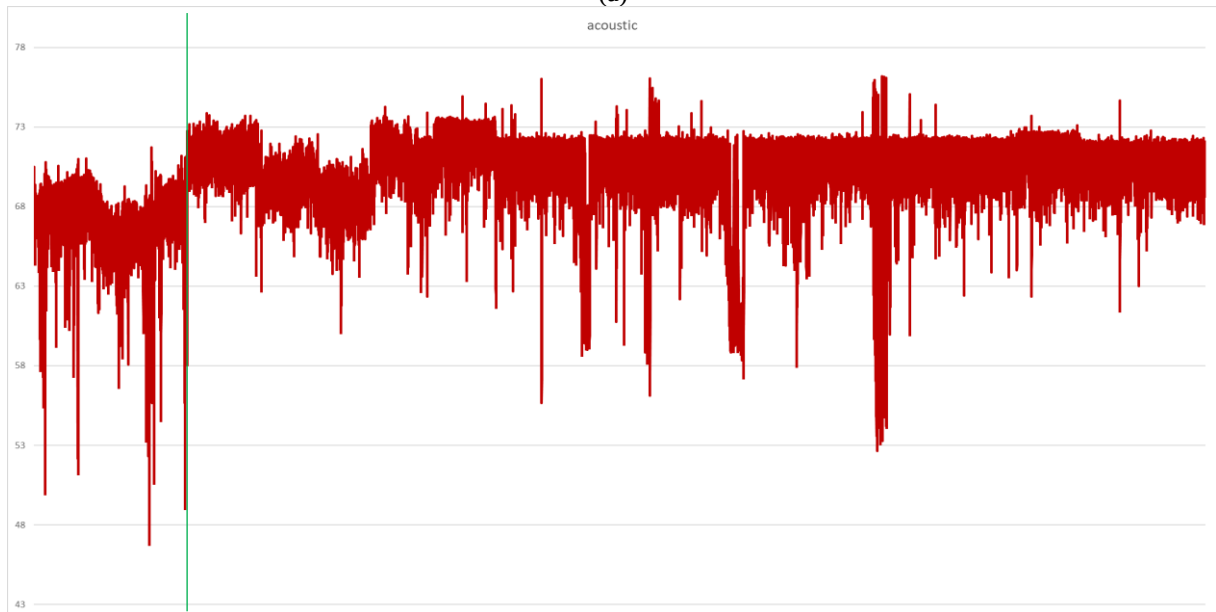
Por outro lado, observando a partir da linha vertical vermelha dos gráficos da Figura 59, nota-se um aumento da amplitude das linhas, onde os pontos menores e maiores estão mais distantes do que no início dos gráficos. Ainda assim, em alguns setores do gráfico a amplitude permaneceu pequena, em relação ao restante.

Ainda, nota-se nos gráficos gerados a partir dos dados dos arquivos TXT, alguns momentos breves em que a emissão acústica era menor e a amplitude da vibração pequena e de repente um aumento abrupto desses indicadores, o que pode ser atribuído ao torno estar em repouso e logo depois ser ligado.

Figura 58 - Gráficos de linha gerados com os pontos adquiridos do sensor de captação de emissão acústica. (a) Dados gerados a partir dos gráficos da página HTML, no formato CSV. (b) Dados gerados a partir dos arquivos brutos, no formato TXT.



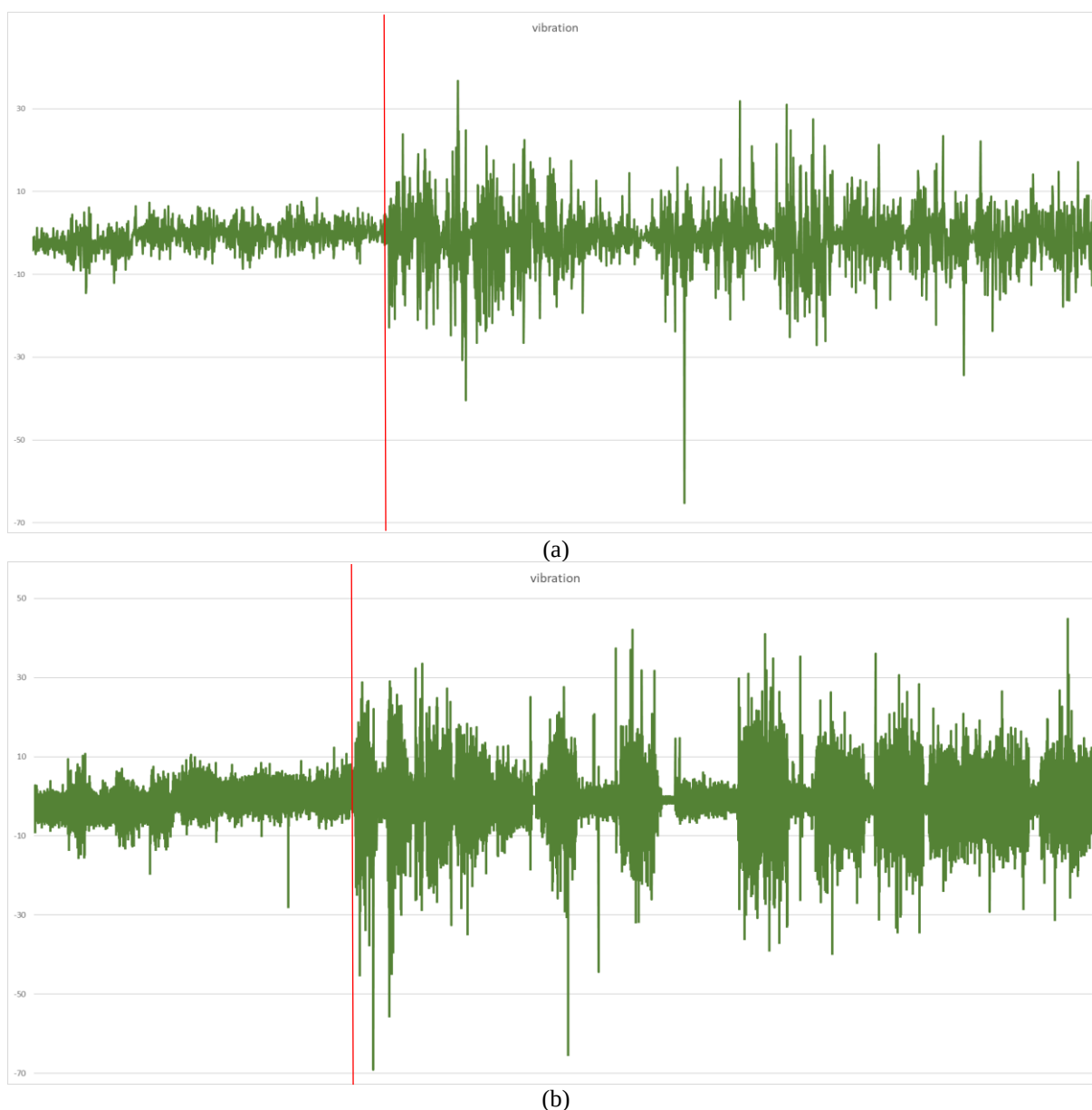
(a)



(b)

Fonte: Elaborado pelo autor.

Figura 59 – Gráficos de linha gerados com os pontos adquiridos do sensor de vibração (a) Dados gerados a partir dos gráficos da página HTML, no formato CSV. (b) Dados gerados a partir dos arquivos brutos, no formato TXT.



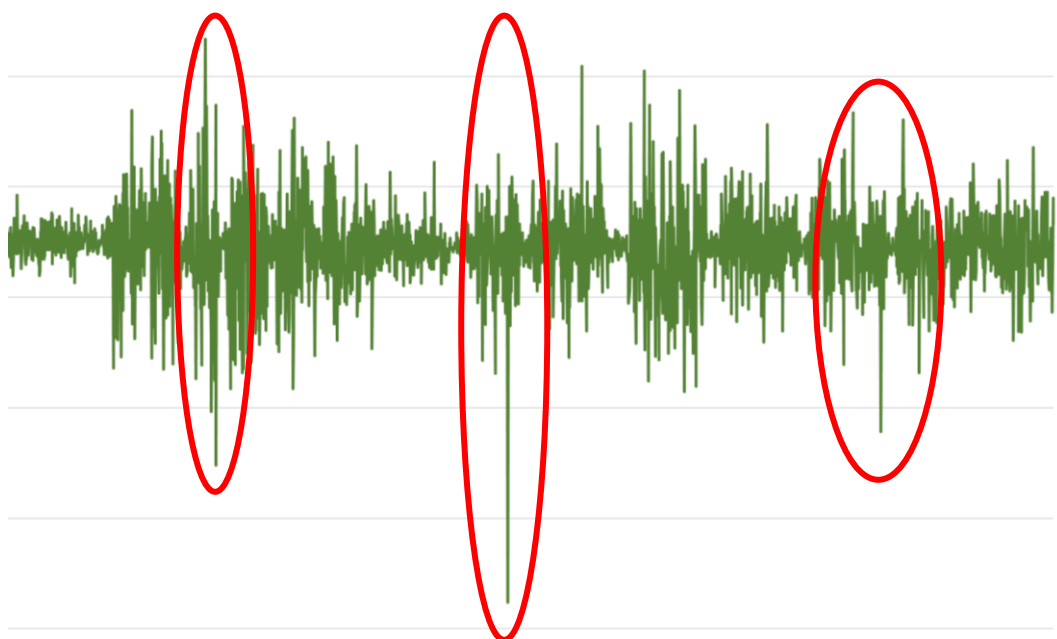
Fonte: Elaborado pelo autor.

Quanto mais próximo da usinagem, maior a quantidade de cavacos que a plataforma pode receber. No caso da usinagem do alumínio, um metal, os cavacos podem invadir áreas sensíveis dos sensores e da protoboard, e ocasionar o curto-circuito. Quando isso ocorreu, os sensores tinham a aquisição interrompida, gerando leituras constantes na página HTML, pois o script Python que opera dentro do Raspberry Pi era interrompido sem a gravação de novos pontos adquiridos, sendo que o monitoramento remoto sempre recupera o último ponto gravado no arquivo para geração do gráfico.

Conforme corroborado na Figura 60, as áreas indicadas em vermelho referem-se a

provavelmente a passagem da ferramenta de corte impurezas que havia no corpo de prova de alumínio, que causaram uma maior amplitude de vibração, dado o esforço da ferramenta em vencer a provável impureza e quando logo após, conseguiu. Esse fenômeno foi capturado de forma ágil pela aquisição, sendo observado no monitoramento remoto da aquisição no microcomputador.

Figura 60 - Indicação de áreas com maior variação da vibração na usinagem.



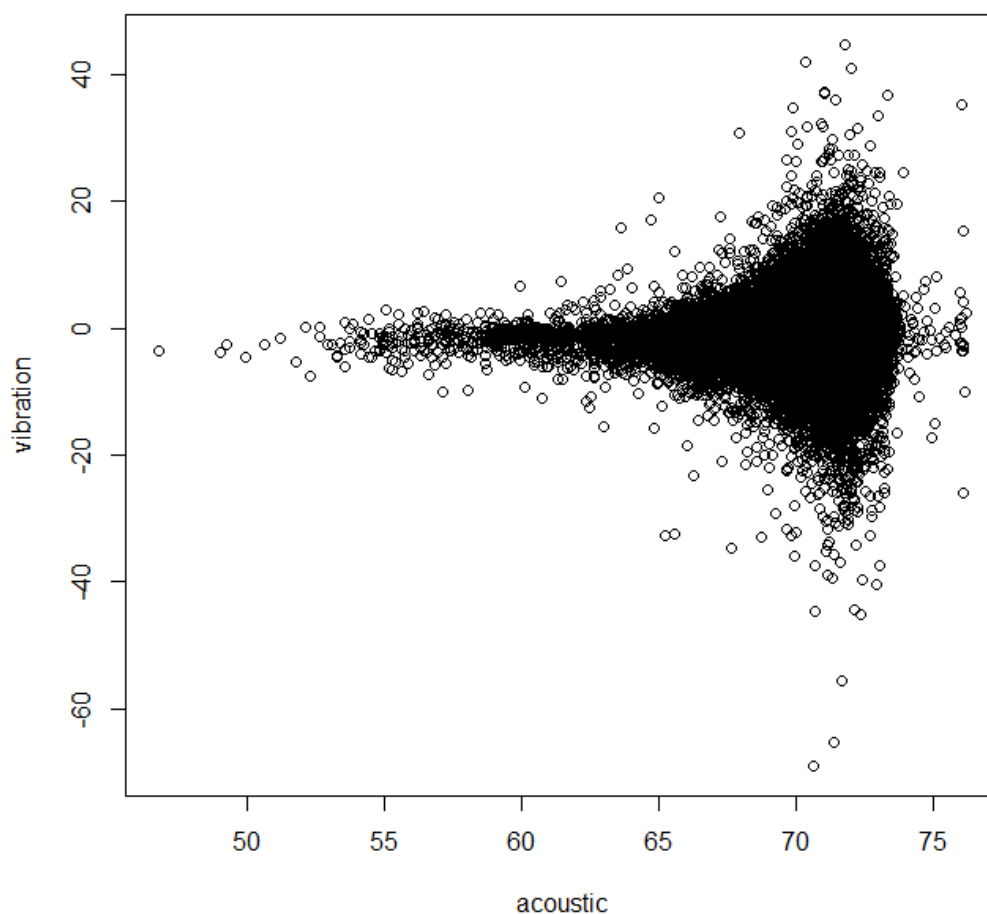
Fonte: Elaborado pelo autor.

Verifica-se pela Figura 61 que quanto maior a emissão acústica, maior a vibração. Essa situação corrobora com a observada in loco, pois quando se aumentava a vibração na usinagem, se notava um aumento no ruído.

Entretanto, provavelmente a maior fonte de emissão acústica é o próprio torno, visto que o seu ruído não foi isolado, sendo capturado no momento da usinagem. Todavia, a vibração capturada pelo sensor era diretamente do suporte da ferramenta, logo provavelmente a vibração adquirida vinha de fato da interface peça-ferramenta. Buscou-se ajustar o sensor de vibração para que ele, quando o torno estava desligado, adquirisse valores próximos ou iguais a zero, o que pode ser notado pela Figura 61, onde o lado esquerdo do gráfico converge próximo a zero.

Observou-se no momento da aquisição, que quando o torno não estava usinando a peça, os valores de vibração permaneciam próximos a zero e emissão acústica estável em torno de 68 dB.

Figura 61 – Gráfico da vibração em função da emissão sonora da usinagem.



Fonte: Elaborado pelo autor.

4.3.2 Análise estatística

Para realização da análise estatística dos dados utilizou-se o programa de computador R, agrupando os ensaios com variáveis semelhantes e analisando todos os dados num contexto geral, com a inclusão de todos os dados adquiridos tanto pela página HTML que realizou o monitoramento remoto quanto pelos dados gravados no arquivo TXT no Raspberry Pi, conforme roteirizado no Quadro 3.

Quadro 3 - Roteiro para análise estatística dos dados adquiridos no Estudo B.

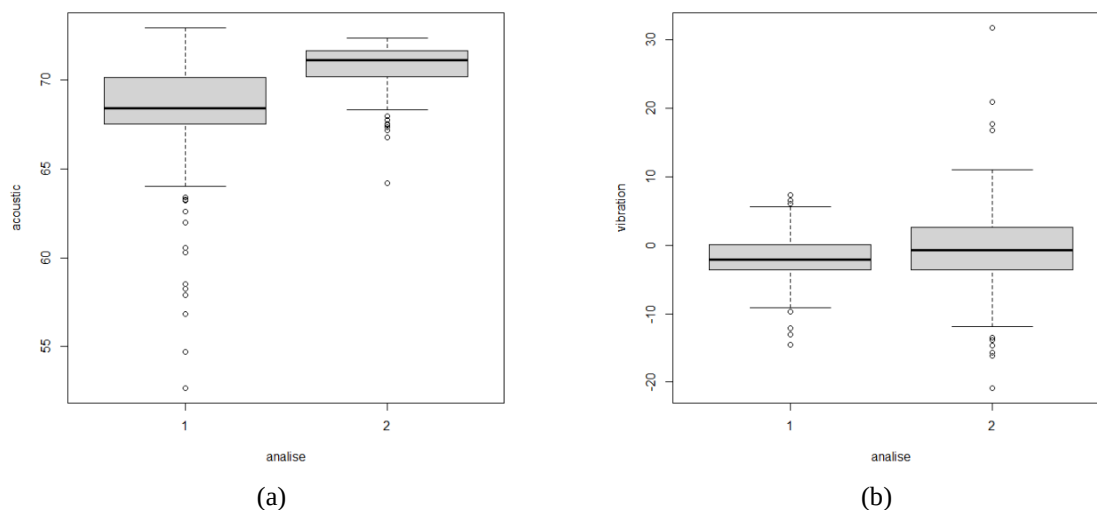
Análise 01 – Ensaios 01, 02, 03. (sensores afastados)
Análise 02 – Ensaios 19, 20 e 21. (sensores próximos)
Análise 03 – Ensaios 04, 05 e 06. (d = 0,5 mm)
Análise 04 – Ensaios 22, 23 e 24. (d = 1,0 mm)
Análise 05 – Ensaios 07, 08, 09, 16, 17 e 18.
Análise 06 – Ensaios 10, 11 e 12.
Análise 07 – Ensaios 13, 14 e 15.
Análise 08 – Ensaios 25, 26 e 27.
Análise 09 – Todos os dados exportados pela página HTML.
Análise 10 – Todos os dados gravados diretamente no Raspberry Pi.

Fonte: Elaborado pelo autor.

4.3.2.1 Comparação entre a Análise 01 e a Análise 02

A fim de se comparar a influência da distância dos sensores na aquisição de dados, optou-se por comparar os resultados da Análise 01 e 02 utilizando *boxplot* (diagrama de caixa), pois na 01 os sensores estavam afastados da interface peça-ferramenta e na 02 os sensores estavam bem próximos, conforme demonstrado pela Figura 56.

Figura 62 - (a) Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 01 e 02. (b) Diagramas comparando os valores obtidos da vibração (°) nas Análises 01 e 02.



Fonte: Elaborado pelo autor.

Percebe-se na Figura 62a que a mediana da Análise 02 está maior em relação à Análise 01, com os valores de mínimo e máximo bem mais próximos na Análise 02, ou seja, nesta análise os valores não variaram muito, se comparados aos dados da Análise 01, onde nota-se maior quantidade de valores atípicos (outlier), que são os círculos vazados além dos valores de mínimo e máximo do *boxplot*.

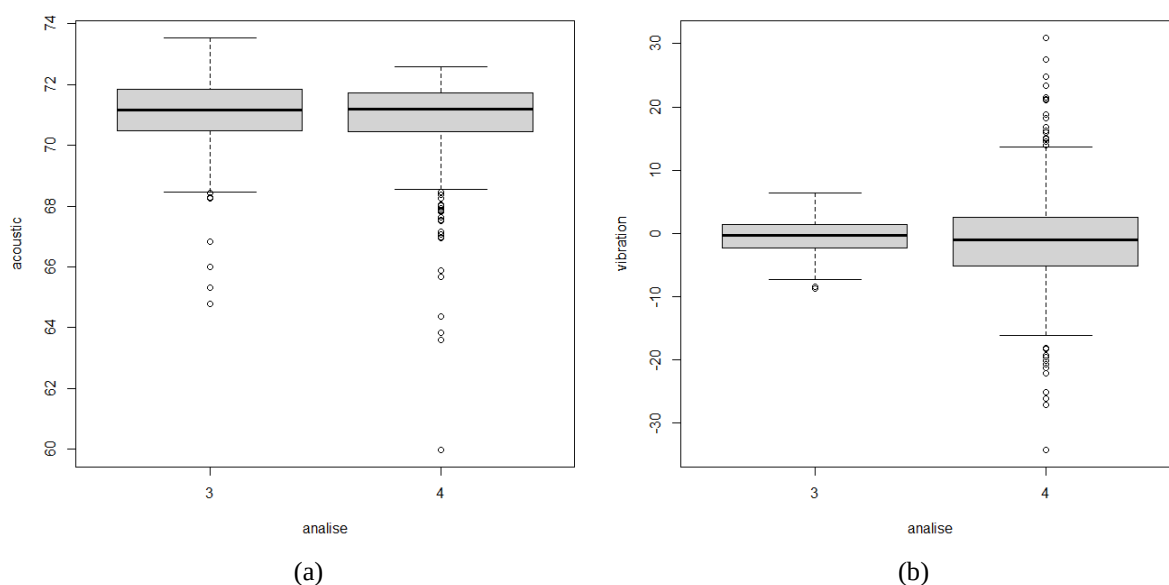
Com relação a Figura 62b, a mediana da Análise 01 e da Análise 02 estão mais próximas, porém a distância entre o primeiro (Q1) e terceiro quartil (Q3) é maior na Análise 02, bem como os valores de mínimo e máximo. Observa-se também a maior incidência de outlier na Análise 02.

Portanto, observa-se maior emissão acústica na Análise 02 em relação a Análise 01 e maior vibração na Análise 02, quando se comparado a Análise 01.

4.3.2.2 Comparação entre a Análise 03 e a Análise 04

Com o intuito de comparar as profundidades de corte diante de uma mesma condição dos outros parâmetros de usinagem, resolveu-se comparar os resultados da Análise 03 (profundidade de 0,5 mm) e da Análise 04 (profundidade de 1,0 mm), novamente utilizando *boxplot*.

Figura 63 - (a) Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 03 e 04. (b) Diagramas comparando os valores obtidos da vibração (°) nas Análises 03 e 04.



Fonte: Elaborado pelo autor.

Conforme observado na Figura 63a, não ocorreu variação significativa da emissão acústica entre as Análises 03 e 04, com a mediana, o primeiro e o terceiro quartil bem próximos entre as análises. A quantidade de valores discrepantes é semelhante entre a Análise 03 e 04.

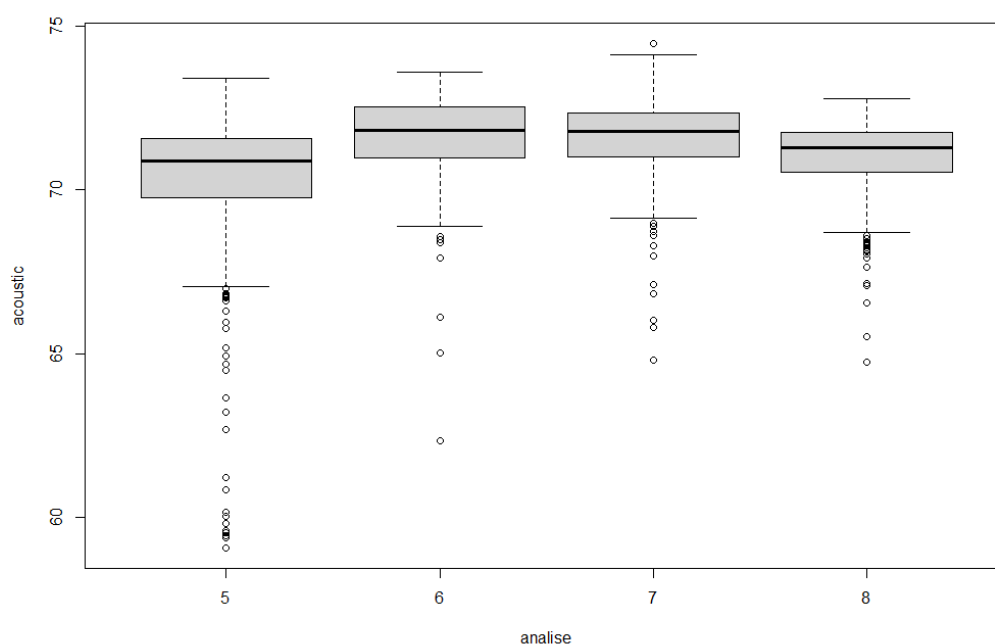
Por outro lado, ocorreu variação significativa na vibração, de acordo com a Figura 63b. Nota-se que a distância entre o primeiro e terceiro quartil na Análise 04 é maior do que na Análise 03, o que indica maior vibração na Análise 04 em relação a Análise 03. A incidência de valores atípicos foi maior na Análise 04.

Logo, observa-se emissão acústica semelhante entre as Análises 03 e 04 e maior vibração na Análise 04, quando comparada à Análise 03.

4.3.2.3 Comparação entre as Análise 05, 06, 07 e 08

Resolveu-se comparar os dados das Análises 05, 06, 07 e 08 por elas terem diferentes avanços e profundidade de corte, pois assim poderemos obter uma melhor investigação da influência do avanço e da profundidade de corte na emissão acústica e na vibração, utilizando-se boxplot. A análise que possui profundidade de corte diferente é a 08, sendo ela maior em relação ao parametrizado nas Análises 05, 06, 07 e 08.

Figura 64 - Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 05, 06, 07 e 08.

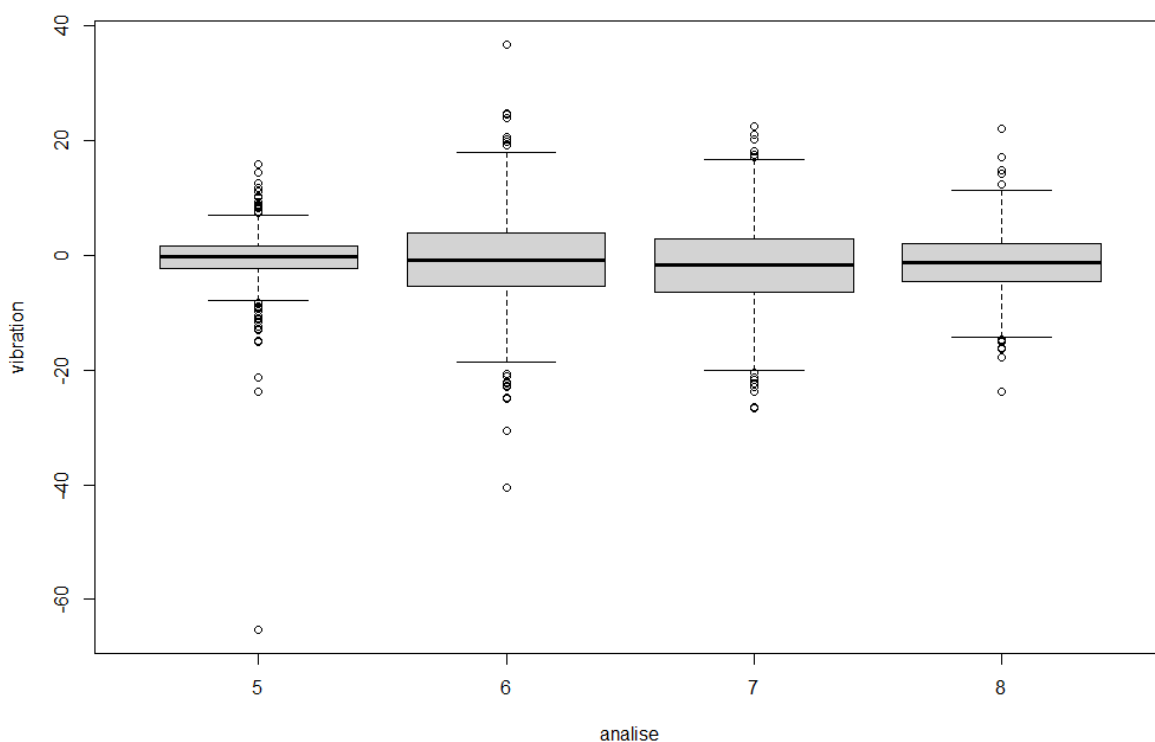


Fonte: Elaborado pelo autor.

Conforme apresentado pela Figura 64, a Análise 06 apresentou maior mediana e maior valor do terceiro quartil, com a Análise 07 estando bem próxima na mediana e no valor do primeiro quartil. A Análise 05 apresentou menor mediana da emissão sonora, com a Análise 08 em seguida. A Análise 05 possuía o menor avanço em relação as Análises 06 e 07, que possuem profundidade de corte de 0,5 mm.

A Análise 08 possui o menor avanço entre as Análises, porém possui a maior profundidade de corte: 1,0 mm. Logo, não é surpreendente que a Análise 05 tenha obtido menor emissão acústica, seguida de perto pela Análise 08. A Análise 06 possui o maior avanço, sendo que ela também apresentou maiores valores de emissão acústica.

Figura 65 - Diagramas comparando os valores obtidos da vibração (°) nas Análises 05, 06, 07 e 08.



Fonte: Elaborado pelo autor.

Com relação à vibração no processo de usinagem, de acordo com a Figura 65 há uma evidente semelhança entre as Análise 06 e 07, sendo que a mediana da Análise 06 é um pouco maior do que a Análise 07. De forma bem clara, a vibração na Análise 05 é a menor de todas, seguida pela Análise 08. Observa-se que a Análise 06 (maior avanço) possui a maior incidência de valores discrepantes, e a Análise 08 (menor avanço) possui a menor quantidade,

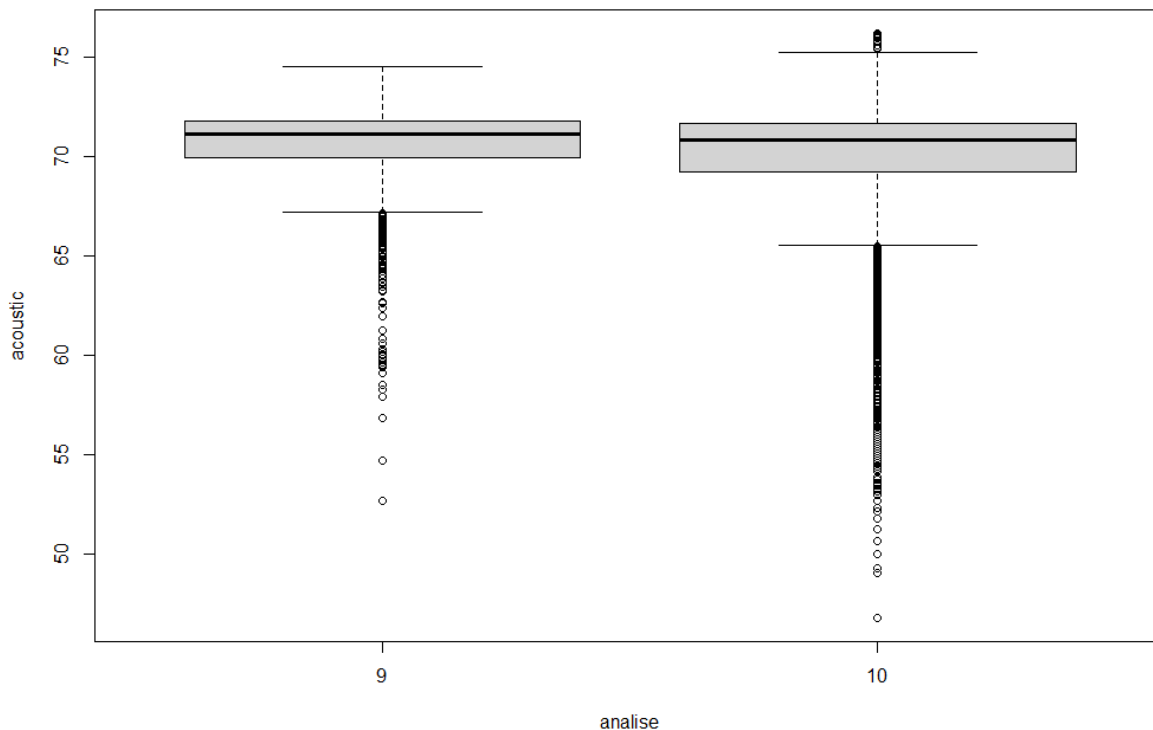
demonstrando maior homogeneidade nos valores adquiridos. Além disso, a menor distância entre o primeiro e o terceiro quartis é verificado na Análise 05.

Portanto, observa-se que a emissão acústica e a vibração foram maiores na Análise 06 com a Análise 07 bem próxima. A menor emissão acústica e o menor índice de vibração são verificados na Análise 05, sempre seguida de perto pela Análise 08.

4.3.2.4 Comparação entre a Análise 09 e Análise 10

Por fim, serão verificados os dados adquiridos pelos arquivos CSV (salvos diretamente da página HTML – Análise 09) e arquivos TXT (salvos no Raspberry Pi – Análise 10). É importante reforçar que a quantidade de pontos na Análise 10 é mais de onze vezes maior do que o observado na Análise 09.

Figura 66 – Diagramas comparando os valores obtidos da emissão sonora (dB) nas Análises 09 e 10.

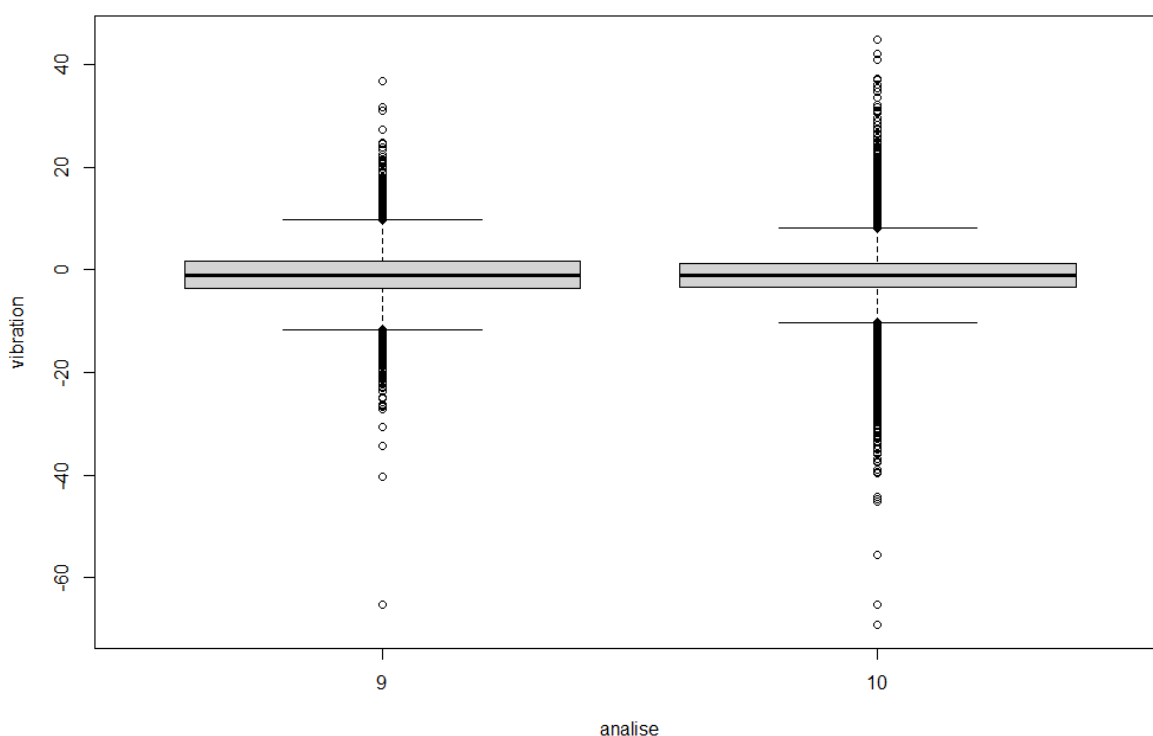


Fonte: Elaborado pelo autor.

De acordo com a Figura 66, emissão acústica das duas análises apresenta valores de mediana (71,09 dB na Análise 09 e 70,81 dB na Análise 10) e terceiro quartil bem semelhantes

(71,77 dB na Análise 09 e 71,66 dB na Análise 10), diferindo apenas no primeiro quartil (69,92 dB na Análise 09 e 69,19 dB na Análise 10). Conforme esperado, dada a maior quantidade de pontos, observa-se maior quantidade de valores atípicos na Amostra 10 em relação à Amostra 09.

Figura 67 – Diagramas comparando os valores obtidos da vibração (°) nas Análises 09 e 10.



Fonte: Elaborado pelo autor.

Com relação aos valores da vibração na usinagem observados na Figura 67, verifica-se que os diagramas estão bem semelhantes entre a Análise 09 e 10. A mediana da Análise 09 é de $-0,9753^\circ$ e da Análise 10 é de $-1,092^\circ$; o valor do primeiro quartil da Análise 09 é $-3,6541^\circ$ e da Análise 10 é $-3,398^\circ$; o valor do terceiro quartil da Análise 09 é $1,7065^\circ$ e da Análise 10 é $1,247^\circ$. Dada a maior quantidade de pontos, a Análise 10 apresenta o maior número de valores discrepantes.

Portanto, não houve diferença significativa entre os dados de emissão acústica e vibração adquiridos na página HTML (2 pontos a cada 500 milissegundos) com relação a todos os dados adquiridos (1 ponto a cada 26 milissegundos).

4.3.2.5 Comparação entre as Análises 01, 02, 03, 04, 05, 06, 07 e 08

Por fim, realizou-se a comparação entre todas as Análises – com exceção da 09 e 10, para verificar globalmente o comportamento dos ensaios realizados, com a variação de alguns parâmetros de entrada e de localização dos sensores.

4.3.2.5.1 Emissão acústica

Utilizando o programa de computador R, realizou-se a Análise de Variância (ANOVA) entre as análises e testou-se as premissas da ANOVA: homogeneidade das amostras, com o teste de Levene; normalidade dos resíduos, com o teste de Kolmogorov-Smirnov. Após essa verificação, realizou-se a comparação de médias par-a-par, com o teste de Tukey. Não se utilizou o teste de Shapiro-Wilk, pois o número de observações é em torno de 3.200 (três mil e duzentos) amostras.

Figura 68 - Teste de Levene da emissão acústica entre as análises.

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)
group   7  22.334 < 2.2e-16 ***
      3282
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fonte: Elaborado pelo autor.

Com base na Figura 68, fundamentado pela estatística F e seu valor-p abaixo de 0,05; pode-se afirmar com grande confiança que as médias das análises diferem de acordo com a alteração dos parâmetros da usinagem.

Figura 69 - Teste de Kolmogorov-Smirnov nas amostras das análises da emissão acústica.

```
One-sample Kolmogorov-Smirnov test
```

```
data:  rnorm(analise_acustica.anova)
D = 0.21812, p-value = 0.4985
alternative hypothesis: two-sided
```

Fonte: Elaborado pelo autor.

De acordo com a Figura 69, após a realização do teste de Kolmogorov-Smirnov que as análises possuem uma distribuição muito próxima da normal. Embora o pressuposto da

normalidade não tenha sido plenamente satisfeito, optou-se pela manutenção da aplicação da análise. Com isso, tem-se que nem todas as médias são iguais, logo, realizar-se-á o teste de Tukey para verificação dos grupos que possuem médias diferentes.

Após a realização do teste de Tukey, observar-se-á o valor de p_{adj} . Quando este valor é menor do que 0,05; fornece uma alta confiança estatística de que as médias das análises diferem ao nível de significância de 5%.

Conforme observado em destaque pela Figura 70, é possível afirmar que as médias não se diferem nas seguintes situações: Análises 02 e 03, Análises 02 e 04, Análises 02 e 05, Análises 02 e 08, Análises 03 e 04, Análises 03 e 08, Análises 04 e 08, Análises 06 e 07. A Análise 02 possui o segundo menor avanço e a menor profundidade de corte dentre as análises. Por outro lado, a Análise 08 possui o menor avanço, porém a maior profundidade de corte. Com isso, provavelmente devido a esses parâmetros que as médias Análises 02 e 08 são as que menos se diferem em relação as demais análises.

Por outro lado, nas demais comparações é possível afirmar que as médias entre as análises se diferem ao nível de significância de 5%.

Conforme esperado as maiores diferenças apresentadas foram da Análise 01 em comparação as demais análises, pois os sensores na Análise 01 estavam afastados da região de corte, evidenciando a influência da localização dos sensores na aquisição dos dados.

A Análise 01 obteve o menor índice de emissão acústica, mesmo possuindo os mesmos valores dos parâmetros de usinagem da Análise 02. Entretanto, a localização dos sensores, é diferente entre as análises, estando a aproximadamente 30 cm da ferramenta na Análise 01 e a 5 cm na Análise 02.

Figura 70 - Comparação de médias das análises de emissão acustica utilizando o teste de Tukey.

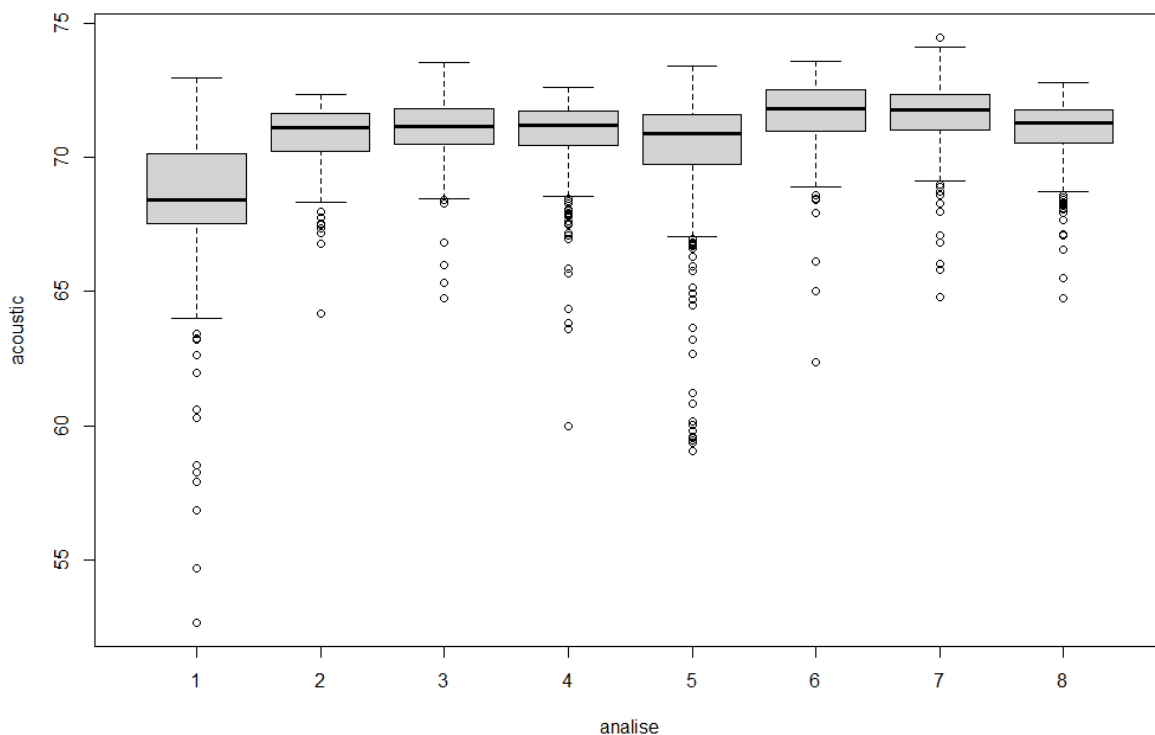
```
Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = acoustic ~ analise, data = analise_acustica)

$analise
      diff      lwr      upr      p adj
a02-a01 2.27523005 1.8261895 2.72427063 0.00000000
a03-a01 2.53177777 2.1403837 2.92317188 0.00000000
a04-a01 2.35154404 2.0457826 2.65730544 0.00000000
a05-a01 1.87087117 1.5753812 2.16636110 0.00000000
a06-a01 3.09768633 2.7130109 3.48236171 0.00000000
a07-a01 3.06000645 2.7081550 3.41185792 0.00000000
a08-a01 2.44675703 2.1009677 2.79254640 0.00000000
a03-a02 0.25654772 -0.2415261 0.75462157 0.7729552
a04-a02 0.07631399 -0.3577116 0.51033961 0.9994849
a05-a02 -0.40435888 -0.8312107 0.02249297 0.0784392
a06-a02 0.82245628 0.3296446 1.31526796 0.0000121
a07-a02 0.78477640 0.3171359 1.25241693 0.0000105
a08-a02 0.17152698 -0.2915697 0.63462361 0.9518654
a04-a03 -0.18023373 -0.5543061 0.19383866 0.8279812
a05-a03 -0.66090660 -1.0266311 -0.29518209 0.0000013
a06-a03 0.56590856 0.1249831 1.00683403 0.0025624
a07-a03 0.52822868 0.1156277 0.94082967 0.0026631
a08-a03 -0.08502074 -0.4924645 0.32242301 0.9984281
a05-a04 -0.48067287 -0.7528035 -0.20854221 0.0000025
a06-a04 0.74614229 0.3791056 1.11317900 0.0000000
a07-a04 0.70846241 0.3759865 1.04093836 0.0000000
a08-a04 0.09521299 -0.2308408 0.42126680 0.9873812
a06-a05 1.22681516 0.8682901 1.58534021 0.0000000
a07-a05 1.18913528 0.8660803 1.51219027 0.0000000
a08-a05 0.57588586 0.2594441 0.89232757 0.0000010
a07-a06 -0.03767987 -0.4439130 0.36855328 0.9999932
a08-a06 -0.65092930 -1.0519233 -0.24993528 0.0000245
a08-a07 -0.61324943 -0.9828715 -0.24362731 0.0000141
```

Fonte: Elaborado pelo autor.

Figura 71 – Diagramas comparando os valores obtidos da emissão acústica (dB) das Análises 01 até 08.



Fonte: Elaborado pelo autor.

Cabe destacar a Análise 06, que possui o maior avanço dentre todas as Análises e, consequentemente, corroborado com a Figura 71, possui os maiores valores de emissão acústica dos ensaios. A Análise 07 que está bem próxima à Análise 06, possui o segundo maior valor para o avanço.

Tabela 2 - Levantamento estatístico comparando as Análises 04 e 08.

	Análise 04	Análise 08	%
f_{mm}	0,099	0,072	37,5
Q1	70,44 dB	70,54 dB	0,14
Mediana	71,19 dB	71,29 dB	0,14
Q3	71,72 dB	71,77 dB	0,07

Fonte: Elaborado pelo autor.

Comparando-se as Análises 04 e 08 em relação as demais, sendo que foi utilizado maior profundidade de corte e valores de avanço muito próximos nessas análises, elas possuem índices

muito semelhantes entre si, mas maiores do que as demais amostras que possuem avanço maior, mas menor profundidade de corte (Tabela 2). Somente as Análises 06 e 07 possuem maior índice de emissão acústica em relação às Análises 04 e 08. O avanço na Análise 06 foi 3,8 vezes maior do que a Análise 08, enquanto a profundidade de corte na Análise 08 foi duas vezes maior se comparada à Análise 06.

4.3.2.5.2 Vibração

A análise estatística foi semelhante à utilizada para as análises da emissão acústica, ou seja, realizou-se a Análise de Variância (ANOVA) entre as análises e testou-se as premissas da ANOVA: homogeneidade das amostras, com o teste de Levene; normalidade dos resíduos, com o teste de Kolmogorov-Smirnov. Após essa verificação, realizou-se a comparação de médias par-a-par, com o teste de Tukey.

Figura 72 - Teste de Levene da vibração entre as análises.

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)
group   7  22.334 < 2.2e-16 ***
      3282
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fonte: Elaborado pelo autor.

De acordo com o Teste de Levene apresentado pela Figura 72, fundamentado pela estatística F e seu valor-p abaixo de 0,05; pode-se afirmar com grande confiança que as médias das análises diferem de acordo com a alteração dos parâmetros da usinagem.

Figura 73 - Teste de Kolmogorov-Smirnov nas amostras das análises da vibração.

```
One-sample Kolmogorov-Smirnov test

data:  rnorm(analise_vibracao.anova)
D = 0.14921, p-value = 0.8937
alternative hypothesis: two-sided
```

Fonte: Elaborado pelo autor.

Após a realização do teste de Kolmogorov-Smirnov (Figura 73), verifica-se que as análises possuem uma distribuição normal, pois o valor-p é maior do que 0,05. Com isso, tem-se que nem todas as médias são iguais, logo, realizar-se-á o teste de Tukey para verificação dos

grupos que possuem médias diferentes.

Diferente da emissão acústica, nos ensaios para aquisição de dados de vibração as médias se diferenciaram apenas em duas situações: entre as Análises 01 e 05 e entre as Análises 05 e 07, com p adj menor do que 0,05. Provavelmente, a vibração da máquina, que não foi filtrada na aquisição, não permitiu que fossem observados os dados de vibração apenas originários da usinagem. A saber, na Análise 05 foram agrupados dados de seis ensaios com os mesmos parâmetros de usinagem, diferente dos demais que agruparam apenas três ensaios cada.

Figura 74 - Comparação de médias das análises da vibração utilizando o teste de Tukey.

```
Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = vibration ~ analise, data = analise_vibracao)

$analise
      diff      lwr      upr      p adj
a02-a01 1.41993287 -0.22993172 3.0697975 0.1524512
a03-a01 1.38855532 -0.04950467 2.8266153 0.0673610
a04-a01 0.73426540 -0.38916293 1.8576937 0.4940884
a05-a01 1.34308007 0.25739117 2.4287690 0.0044233
a06-a01 0.78148935 -0.63188468 2.1948634 0.7023550
a07-a01 0.02668478 -1.26608766 1.3194572 1.0000000
a08-a01 0.54372890 -0.72677018 1.8142280 0.8998921
a03-a02 -0.03137755 -1.86140015 1.7986451 1.0000000
a04-a02 -0.68566747 -2.28036411 0.9090292 0.8976236
a05-a02 -0.07685280 -1.64519160 1.4914860 0.9999999
a06-a02 -0.63844352 -2.44913188 1.1722448 0.9630709
a07-a02 -1.39324809 -3.11145261 0.3249564 0.2135330
a08-a02 -0.87620397 -2.57771333 0.8253054 0.7731715
a04-a03 -0.65428992 -2.02870644 0.7201266 0.8367382
a05-a03 -0.04547525 -1.38922000 1.2982695 1.0000000
a06-a03 -0.60706597 -2.22711403 1.0129821 0.9488130
a07-a03 -1.36187054 -2.87784882 0.1541077 0.1155403
a08-a03 -0.84482642 -2.34185597 0.6522031 0.6797661
a05-a04 0.60881467 -0.39104763 1.6086770 0.5879525
a06-a04 0.04722395 -1.30134209 1.3957900 1.0000000
a07-a04 -0.70758062 -1.92916353 0.5140023 0.6495958
a08-a04 -0.19053650 -1.38852320 1.0074502 0.9997351
a06-a05 -0.56159072 -1.87888324 0.7557018 0.9017295
a07-a05 -1.31639529 -2.50336370 -0.1294269 0.0177262
a08-a05 -0.79935117 -1.96202109 0.3633187 0.4247685
a07-a06 -0.75480457 -2.24738615 0.7377770 0.7892042
a08-a06 -0.23776045 -1.71109241 1.2355715 0.9997082
a08-a07 0.51704412 -0.84102123 1.8751095 0.9443634
```

Fonte: Elaborado pelo autor.

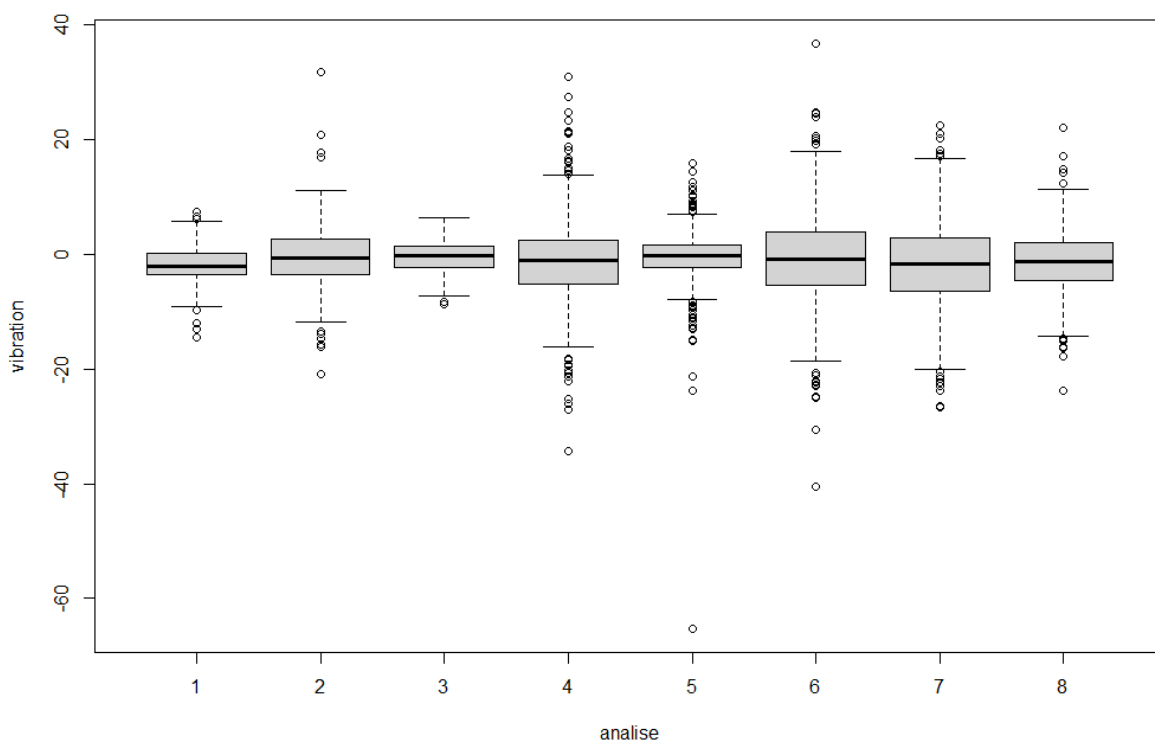
No que se refere à vibração observada pelos diagramas da Figura 75, a Análise 06 apresentou a maior variação entre os quartis, e, como na emissão acústica, a Análise 07 possui

a segunda maior variação entre os quartis.

A Análise 01 possui baixa variação entre os quartis e menores valores adquiridos da vibração, muito embora essa análise estava com o sensor de vibração afastado da interface peça-ferramenta em relação à todas as demais análises, logo, da mesma forma que na emissão acústica, era esperado a menor vibração capturada pelo sensor, ainda que nas mesmas condições de usinagem da Análise 02.

Com relação às Análises 04 e 08, que possuem maior profundidade de corte e menor avanço dentre todas as análises, apresentaram valores de vibração semelhantes.

Figura 75 - Diagramas comparando os valores obtidos da vibração (°) das Análises 01 até 08.



Fonte: Elaborado pelo autor.

Ao se comparar os gráficos gerados com relação ao posicionamento dos sensores, Figura 44, quando os sensores estavam afastados da interface peça-ferramenta, eles apresentaram na aquisição de dados valores menores em comparação aos dados adquiridos quando os sensores estavam próximos da interface peça-ferramenta, de acordo com os gráficos constantes na Figura 58 e na Figura 59.

A partir dos dados adquiridos, pode-se verificar a influência da mudança dos parâmetros

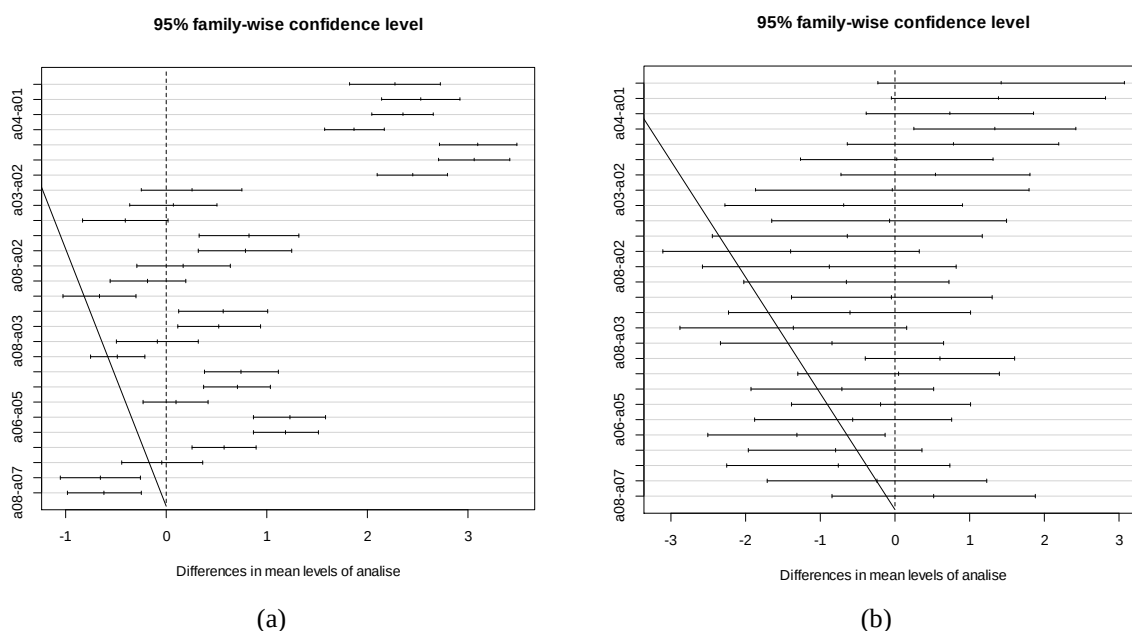
de usinagem na emissão acústica e na vibração, onde quanto maior o avanço, maior a emissão acústica e a vibração. Todavia, pelos resultados apresentados no Estudo B, o aumento da profundidade de corte mostrou-se mais impactante do que o aumento do avanço, principalmente quando analisada a vibração, que apresentou maior amplitude em relação à profundidade de corte menor.

Logo, conforme esperado a maior emissão acústica e maior vibração ocorreu quando se estipulou um alto valor para o avanço; 2,73 vezes maior do que o menor valor de avanço utilizado quando a profundidade de corte foi de 0,5 mm. Porém, mesmo com um avanço dessa ordem, os valores não ficaram distantes do identificado para um avanço 2,73 menor, mas com profundidade de corte de 1,0 mm.

Portanto, pelas informações fornecidas pelos diagramas, entende-se que a interferência da profundidade de corte na emissão acústica e na vibração, nas condições de usinagem do Estudo B, é maior do que o aumento do avanço.

Porém, ao se observar o teste de Tukey aplicados nos dados adquiridos da emissão acústica e da vibração, percebe-se que eles não possuem o mesmo comportamento quando os parâmetros da usinagem são modificados, conforme apresentado pela Figura 76. A variação das diferenças das médias entre as análises é menor na emissão acústica em relação a vibração, fato que pode ter contribuído no menor número entre análises com médias diferentes na vibração (2), em relação à emissão acústica (20).

Figura 76 - (a) Representação gráfica do teste de Tukey aplicado nas análises da emissão acústica. (b) Representação gráfica do teste de Tukey aplicado nas análises da vibração.



Fonte: Elaborado pelo autor.

Sobre o protótipo, uma melhoria proposta para o script que não foi desenvolvida, é a identificação do erro de leitura da porta do sensor a fim de se evitar a interrupção do script e, em último caso, reiniciar o script. Entretanto, como os gráficos estavam sendo monitorados no microcomputador, prontamente os scripts eram reiniciados, sem maior prejuízo da aquisição.

5 CONCLUSÃO

No Estudo A, o sistema desenvolvido apresentou funcionamento coerente com o apresentado na usinagem, pois o comportamento tanto do acionamento da máquina, quanto da operação de usinagem em si são bem representados pelos gráficos (Figura 53), onde quando aumentou-se a pressão da peça sobre a lixa, o gráfico representou adequadamente o aumento da vibração (picos superiores e inferiores), com o maior deslocamento do sensor GY-521.

No caso do Estudo B, a plataforma de aquisição de dados de baixo custo, no que se refere à representação do fenômeno, funcionou corretamente. Tanto para a aquisição da vibração quanto para a captação da emissão sonora pelos sensores, representando pelos gráficos construídos em tempo real no monitoramento o fenômeno físico observado na usinagem, inclusive em situações críticas, como, por exemplo, a provável passagem da ferramenta por impurezas da peça usinada.

Inclusive, de acordo com as situações estudadas na usinagem da peça de alumínio, a análise estatística indicou resultados coerentes com o esperado como, por exemplo, quando a profundidade de corte ou o avanço aumentavam, os valores adquiridos da emissão acústica e da vibração também aumentavam, validando-se a aquisição dos dados pela plataforma.

A localização dos sensores é um fator essencial para a correta aquisição dos dados de usinagem, pois quanto mais próximo ao local do evento, maior a precisão.

Sugere-se o isolamento de ruídos ambientais e a maior mitigação possível da emissão acústica que tenha origem o equipamento de usinagem, visando capturar a emissão sonora decorrente apenas do fenômeno da usinagem.

Devido ao estado de calamidade pública (SÃO PAULO, 2020) pelo agravamento da doença do coronavírus 2019 (WHO, 2020), não foi possível realizar a calibração dos sensores e a validação dos dados adquiridos, pois o acesso às instituições e empresas ficaram restringidos.

Diante do verificado nos ensaios, o sistema proposto pode receber mais sensores, pois em testes no desenvolvimento da aplicação para a aquisição, com o aumento de canais o desempenho do sistema não foi prejudicado, apenas o número de pontos por segundo de cada sensor foi reduzido proporcionalmente ao número de sensores.

Portanto, torna-se válido a implementação deste sistema para monitoramento de outros fenômenos da usinagem de materiais sem que haja grande dispêndio financeiro, apenas aquisição de sensores correspondentes, utilização de mais jumpers para ligação entre os componentes e pequena adaptação na aplicação para acomodar o aumento do número de

sensores.

Com relação ao custo da plataforma de aquisição de dados, totalizando-se apenas os materiais do sistema de aquisição, descontando os materiais para os ensaios, a lixadeira, o microcomputador para desenvolvimento da aplicação e manipulação do Raspberry Pi[®], o tempo dedicado para a pesquisa, produção e testes da aplicação, ele ficou em torno de \$85 (oitenta e cinco dólares americanos).

Sendo assim, com os resultados obtidos e o custo de investimento, o sistema proposto, após o devido amadurecimento, é uma alternativa viável para a aquisição de dados com baixo custo de investimento e de implementação.

REFERÊNCIAS

ABD-EL-BARR, M.; EL-REWINI, H. **Fundamentals of computer organization and architecture**. Hoboken, New Jersey: John Wiley & Sons, 2005. 288 p.

AHMED, Y. S. *et al.* Use of acoustic emission and cutting force signals to monitor built-up edge formation in stainless steel turning. **The International Journal Of Advanced Manufacturing Technology**, London, v. 103, n. 5-8, p. 2257-2276, 2019. Disponível em: <https://doi.org/10.1007/s00170-019-03607-3>. Acesso em: 10 mar. 2020.

BARRY, J; BYRNE, G. Study on acoustic emission in machining hardened steels Part 1: acoustic emission during saw-tooth chip formation: Acoustic emission during saw-tooth chip formation. **Proceedings Of The Institution Of Mechanical Engineers, Part B: Journal of Engineering Manufacture**, London, v. 215, n. 11, p. 1549-1559, 2001. Disponível em: <https://dx.doi.org/10.1243/0954405011519457>. Acesso em: 05 mar. 2020.

BEGGAN, C. *et al.* Using acoustic emission to predict surface quality. **The International Journal Of Advanced Manufacturing Technology**, London, v. 15, n. 10, p. 737-742, 1999. Disponível em: <https://doi.org/10.1007/s001700050126>. Acesso em: 10 mar. 2020.

BROCK, J. D.; BRUCE, R. F.; REISER, S. L. Using arduino for introductory programming courses. **Journal of Computing Sciences in Colleges**, Chicago, v. 25, i. 2, p. 129-130, 2009. Disponível em: <https://dl.acm.org/doi/10.5555/1629036.1629057>. Acesso em: 10 mar. 2020.

BYRNE, G.; DORNFELD, D.; DENKENA, B. Advancing cutting technology. **Cirp Annals**, Amsterdam, v. 52, n. 2, p.483-507, 2003. Trabalho apresentado na 53rd General Assembly of CIRP, 2003, Montreal. Disponível em: [http://dx.doi.org/10.1016/s0007-8506\(07\)60200-5](http://dx.doi.org/10.1016/s0007-8506(07)60200-5). Acesso em: 01 mar. 2020.

CHOUDHURY, S. K.; GOUDIMENKO, N. N.; KUDINOV, V. A. On-line control of machine tool vibration in turning. **International Journal Of Machine Tools And Manufacture**, New York, v. 37, n. 6, p. 801-811, 1997. Disponível em: [http://dx.doi.org/10.1016/s0890-6955\(96\)00031-4](http://dx.doi.org/10.1016/s0890-6955(96)00031-4). Acesso em: 04 mar. 2020.

DINIZ, A. E.; MARCONDES, F. C.; COPPINI, N. L. **Tecnologia da usinagem dos materiais**. 8. ed. São Paulo: Artliber, 2013. 272 p.

ELANGO VAN, M. *et al.* Machine learning approach to the prediction of surface roughness using statistical features of vibration signal acquired in turning. **Procedia Computer Science**, Amsterdam, v. 50, p. 282-288, 2015. Trabalho apresentado na Big Data and Cloud Computing Challenges (ISBCC'15) , 2015, Chennai. Disponível em: <https://dx.doi.org/10.1016/j.procs.2015.04.047>. Acesso em: 04 mar. 2020.

FERRARESI, D. **Fundamentos da usinagem dos metais**. São Paulo: Blucher, 1970, 751p.

FURBER, S. **ARM system-on-chip architecture**. 2. ed. [S.l.]: Addison-Wesley Professional, 2000. 432 p.

GADELHA, R. M. Educação no Brasil: desafios e crise institucional. **Pesquisa & Debate**, v. 28, n. 1, 2017. Disponível em: <https://ken.pucsp.br/rpe/article/download/33530/23123>. Acesso em: 01 mar. 2020.

GHOSH, P. S. *et al.* Empirical modelling and optimization of temperature and machine vibration in CNC hard turning. **Materials Today: Proceedings**, Amsterdam, v. 5, n. 5, p. 12394-12402, 2018. Disponível em: <https://doi.org/10.1016/j.matpr.2018.02.218>. Acesso em: 05 mar. 2020.

GOSAI, M.; BHAVSAR, S. N. Experimental study on temperature measurement in turning operation of hardened steel (EN36). **Procedia Technology**, Gujarat, v. 23, p. 311-318, 2016. Trabalho apresentado no 3rd International Conference on Innovations in Automation and Mechatronics Engineering, 2016, Gujarat. Disponível em: <https://doi.org/10.1016/j.protcy.2016.03.032>. Acesso em: 27 maio 2020.

HABER, R. E. *et al.* Artificial cognitive control with self-x capabilities: a case study of a micro-manufacturing process. **Computers In Industry**, Amsterdam, v. 74, p.135-150, 2015. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0166361515300014>. Acesso em: 03 jul. 2019.

HALFACREE, G. **The official Raspberry Pi beginner's guide**: How to use your new computer. Cambridge: Raspberry Pi Trading, 2018. Disponível em: https://www.raspberrypi.org/magpi-issues/Beginners_Guide_v1.pdf. Acesso em: 21 abr. 2020.

HECKER, R.L. *et al.* Grinding force and power modeling based on chip thickness analysis. **The International Journal of Advanced Manufacturing Technology**, London, v. 33, p. 449–459, 2007. Disponível em: <https://www.proquest.com/openview/f9d45896f745d6a99fa42040a0ef6ef3/1?pq-origsite=gscholar&cbl=2044010>. Acesso em: 01 mar. 2020.

KEMP, S. **Global digital report 2019**. [S.l.]: We Are Social, 2019. Disponível em: <https://wearesocial.com/global-digital-report-2019>. Acesso em: 10 fev. 2020.

KISHAWY, H. A. *et al.* Application of acoustic emissions in machining processes: analysis and critical review.: analysis and critical review. **The International Journal Of Advanced Manufacturing Technology**, London, v. 98, n. 5-8, p. 1391-1407, 2018. Disponível em: <http://dx.doi.org/10.1007/s00170-018-2341-y>. Acesso em: 10 mar. 2020.

KURFESS, T. R. **Robotics and automation handbook**. Boca Raton, Florida: CRC Press, 2005. 602 p.

LAMB, F. **Industrial automation**: Hands On. New York: Mcgraw-hill Education, 2013. 368 p.

LAPSOMTHOP, W. *et al.* Design and experimental investigation on 3- component force sensor in mini CNC milling machine. **Materials Today: Proceedings**, Amsterdam, v. 17, p. 1931–1938, 2019. Trabalho apresentado no The First Materials Research Society of Thailand International Conference, 2017, Chiang Mai. Disponível em: <http://dx.doi.org/10.1016/j.matpr.2019.06.232>. Acesso em: 27 maio 2020.

LEE, D. *et al.* Precision manufacturing process monitoring with acoustic emission. **International Journal Of Machine Tools And Manufacture**, New York, v. 46, n. 2, p. 176-188, 2006. Disponível em: <https://dx.doi.org/10.1016/j.ijmachtools.2005.04.001>. Acesso em: 09 mar. 2020.

LEWIS, A. J.; CAMPBELL, M.; STAVROULAKIS, P. Performance evaluation of a cheap, open source, digital environmental monitor based on the Raspberry Pi. **Measurement**, Amsterdam, v. 87, p.228-235, 2016. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0263224116001871>. Acesso em: 03 jul. 2019.

MACHADO, A. R. *et al.* **Teoria da usinagem dos materiais**. São Paulo: Edgard Blucher, 2009. 384 p.

MANESIS, S.; NIKOLAKOPOULOS, G. **Introduction to industrial automation**. Boca Raton, Fl: Crc Press, 2018. 457 p.

MCROBERTS, M. **Arduino básico**. São Paulo, SP: Novatec Editora, 2011. 453 p.

MEASUREMENT COMPUTING CORPORATION. **Data acquisition handbook: a reference for DAQ and analog & digital signal conditioning**. 3. ed. Massachusetts: Measurement Computing Corporation, 2012. 145 p. Disponível em: <https://www.mccdaq.com/support/Data-Acquisition-Handbook>. Acesso em: 16 abr. 2020.

MONK, S. **Programming Arduino: Getting Started with Sketches**. [S.l.]: McGraw-Hill, 2012. 176 p.

NARANG, S. **Raspberry Pi 3 tutorial 11 – GPIO analog sensor + MCP3008 / MCP3004**. [S.l.], Module143, 2016. Disponível em: http://invent.module143.com/daskal_tutorial/raspberry-pi-3-gpio-analog-sensor-mcp3008-mcp3004/. Acesso em: 14 out. 2020.

NASCIMENTO, A. J.; HELLER, J. L. **Introdução à informática**. 3. ed. rev. São Paulo: Makron Books, 1996. 151 p.

NORRIS, D. J. **Beginning artificial intelligence with the Raspberry Pi**. [S.l.]: Apress, 2017. 396 p.

PAJANKAR, A. **Raspberry Pi image processing programming: Develop Real-Life Examples with Python, Pillow, and SciPy**. Maharashtra, India: Apress, 2017. 126 p.

PAPACHARALAMPOPOULOS, A. *et al.* Acoustic emission signal through turning tools: a computational study.: A Computational Study. **Procedia Cirp**, Amsterdam, v. 8, p. 426-431, 2013. Trabalho apresentado no 14th CIRP Conference on Modeling of Machining Operations, 2013, Turin. Disponível em: <https://dx.doi.org/10.1016/j.procir.2013.06.128>. Acesso em: 10 mar. 2020.

PARK, J.; MACKAY, S. **Practical data acquisition for instrumentation and control systems**. New Jersey: Newnes, 2003. 432 p.

PRAKASH, M. *et al.* Investigations on the effects of tool wear on chip formation mechanism and chip morphology using acoustic emission signal in the microendmilling of aluminum alloy. **The International Journal Of Advanced Manufacturing Technology**, London, v. 77, n. 5-8, p. 1499-1511, 2014. Disponível em: <https://dx.doi.org/10.1007/s00170-014-6562-4>. Acesso em: 11 mar. 2020.

PRASAD, M. S. **Design simulation and fabrication of micromachined acceleration sensor**. 2010. 125 f. Tese (Doutorado em Mechanical Engineering) – Jawaharlal Nehru Technological University, Hyderabad, 2010. Disponível em: <http://hdl.handle.net/10603/2272>. Acesso em: 05 mar. 2020.

RASPBERRY PI FOUNDATION. **Raspberry Pi 3 model B**. [S.l.]: Raspberry Pi, [2016]. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/?resellerType=home>. Acesso em: 14 out. 2020.

RIBEIRO FILHO, S. L. M. *et al.* Monitoring of microturning process using acoustic emission signals. **Journal Of The Brazilian Society Of Mechanical Sciences And Engineering**, Heidelberg, v. 41, n. 6, p. 1-11, 2019. Disponível em: <https://dx.doi.org/10.1007/s40430-019-1774-z>. Acesso em: 10 mar. 2020.

RISBOOD, K. A; DIXIT, U. S; SAHASRABUDHE, A. D. Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process. **Journal Of Materials Processing Technology**, Amsterdam, v. 132, n. 1-3, p. 203-214, 2003. Elsevier BV. Disponível em: [https://dx.doi.org/10.1016/s0924-0136\(02\)00920-2](https://dx.doi.org/10.1016/s0924-0136(02)00920-2). Acesso em: 04 mar. 2020.

ROGGIA, L.; FUENTES, R. C. **Automação industrial**. Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2016. Disponível em: https://www.ufsm.br/unidades-universitarias/ctism/cte/wp-content/uploads/sites/413/2018/12/06_automacao_industrial.pdf. Acesso em: 16 fev. 2020.

SÃO PAULO (Estado). Decreto nº 64.879, de 20 de março de 2020. Reconhece o estado de calamidade pública, decorrente da pandemia do COVID-19, que atinge o Estado de São Paulo, e dá providências correlatas. **Diário oficial do estado de São Paulo**, São Paulo, 2020. Disponível em: <https://www.saopaulo.sp.gov.br/wp-content/uploads/2020/03/decretos-64879-e-64880.pdf>. Acesso em: 14 out. 2020.

STEPHENSON, D. A.; AGAPIOU, J. S. **Metal cutting theory and practice**. Michigan, USA: CRC Press, 1996. 914 p.

STALLINGS, W. **Computer organization and architecture**. 10. ed. [S.l.]: Pearson Education, 2016. 833 p.

TRENT, E. M.; WRIGHT, P. K. **Metal cutting**. 4. ed. Woburn: Butterworth-heinemann, 2000. 464 p.

UPTON, E. **8GB Raspberry Pi 4 on sale now at \$75**. Raspberry Pi Blog: Cambridge, 2020. Disponível em: <https://www.raspberrypi.org/blog/8gb-raspberry-pi-4-on-sale-now-at-75/>. Acesso em: 27 maio 2020.

WANG, B.; LIU, Z. Acoustic emission signal analysis during chip formation process in high speed machining of 7050-T7451 aluminum alloy and Inconel 718 superalloy. **Journal Of Manufacturing Processes**, London, v. 27, p. 114-125, 2017. Disponível em: <https://dx.doi.org/10.1016/j.jmapro.2017.04.003>. Acesso em: 11 mar. 2020.

World Health Organization (WHO). **Coronavirus disease 2019 (COVID-19)**. Situation Report. 51 Genebra: WHO; 2020. Disponível em: <https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200311-sitrep-51-covid-19.pdf>. Acesso em: 14 out. 2020.

WÖSTENKÜHLER, G. Data acquisition systems (DAS) in general. In: SYDENHAM, P. H; THORN, R. **Handbook of measuring system design**. New York: John Wiley and Sons Ltd, 2005. v. 3. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/0471497398.mm452>. Acesso em: 12 abr. 2020.

APÊNDICE A – CÓDIGO-FONTE DOS PROGRAMAS DESENVOLVIDOS PARA AQUISIÇÃO E MONITORAMENTO NA LINGUAGEM PYTHON

aquisicao.py

```

1.  #!/usr/bin/python
2.  # -*- coding: utf-8 -*-
3.  # Import SPI library (for hardware SPI) and MCP3008 library.
4.  import Adafruit_GPIO.SPI as SPI
5.  import Adafruit_MCP3008
6.
7.  import time
8.  import math
9.  import smbus
10. import json
11.
12. def read_byte(reg):
13.     return bus.read_byte_data(address, reg)
14.
15. def read_word(reg):
16.     h = bus.read_byte_data(address, reg)
17.     l = bus.read_byte_data(address, reg+1)
18.     value = (h << 8) + l
19.     return value
20.
21. def read_word_2c(reg):
22.     val = read_word(reg)
23.     if (val >= 0x8000):
24.         return -((65535 - val) + 1)
25.     else:
26.         return val
27.
28. def dist(a,b):
29.     return math.sqrt((a*a)+(b*b))
30.
31. def get_y_rotation(x,y,z):
32.     radians = math.atan2(x, dist(y,z))
33.     return -math.degrees(radians)
34.
35. def get_x_rotation(x,y,z):
36.     radians = math.atan2(y, dist(x,z))
37.     return math.degrees(radians)
38.
39. def emissao_sonora(mcp, port):
40.     sampleWindow = 50
41.     sample = 0
42.
43.     startMillis = int(round(time.time() * 1000))
44.     peakToPeak = 0
45.     signalMax = 0
46.     signalMin = 1024
47.
48.     while((int(round(time.time() * 1000)) - startMillis) < sampleWindow):
49.         try:
50.             sample = mcp.read_adc(port)
51.             if(sample < 1024):
52.                 if(sample > signalMax):
53.                     signalMax = sample
54.                 elif(sample < signalMin):
55.                     signalMin = sample
56.         except Exception as e:
57.             print("Erro ao abrir sensor analógico: ", str(e))

```

```

58.
59.     peakToPeak = signalMax - signalMin
60.     volts = (peakToPeak * 3.3) / 1024
61.
62.     return volts
63.
64. def line_write(filename, line):
65.     line = "<?php header('Access-Control-Allow-Origin: *');header('Access-Control-
        Allow-Headers: Content-Type');echo '" + line + "';?>"
66.     with open(filename, 'w') as f:
67.         f.write(line)
68.
69. def line_prepend(filename, line):
70.     with open(filename, 'a') as f:
71.         f.write(line)
72.
73. # Nomes dos arquivo
74. sr_file = '/var/www/html/monitoring.php'
75. daq_file = '/var/www/html/monitoring.txt'
76.
77. # Limpa conteúdo dos arquivo
78. open(daq_file, 'w').close()
79.
80. # Registros giroscópio
81. power_mgmt_1 = 0x6b
82. power_mgmt_2 = 0x6c
83.
84. # Hardware SPI configuration:
85. SPI_PORT = 0
86. SPI_DEVICE = 0
87.
88. # Abre o sensor analógico
89. try:
90.     mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE))
91. except Exception as e:
92.     print("Erro ao abrir sensor analógico:", str(e))
93.
94. # Sensibilidade em dBV/Pa
95. mic_sensitivity = -44
96.
97. # Sensibilidade em V.RMS/Pa
98. v0 = 0.001259
99.
100. # Tensão de alimentação em V
101. mic_voltage = 3.3
102.
103. # Ganho do microfone
104. mic_gain = 40
105.
106. # Nível Bias
107. bias_level = (1023 / mic_voltage) * 0
108.
109. # Giroscópio
110. bus = smbus.SMBus(1) # bus = smbus.SMBus(0) fuer Revision 1
111. address = 0x68 # via i2cdetect
112.
113. # Ativação para endereçar módulo
114. bus.write_byte_data(address, power_mgmt_1, 0)
115.
116. # Valor armazenado
117. ch_ac = ""
118. i = 1
119. inicio = time.time()
120.
121. with open(daq_file, 'a') as f:
122.     while True:

```

```

123.         #try:
124.         # Cálculo do dB (SPL)
125.         v1 = emissao_sonora(mcp, 7)
126.         #v1 = mcp.read_adc(7)
127.         db_spl = 20 * math.log10(v1 / v0)
128.         db_spl = mic_sensitivity + db_spl + 94
129.         db_spl = db_spl - mic_gain
130.         ch0_valor = db_spl
131.
132.         # Leitura do giroscópio
133.         acelerometro_xout = read_word_2c(0x3b)
134.         acelerometro_yout = read_word_2c(0x3d)
135.         acelerometro_zout = read_word_2c(0x3f)
136.
137.         acelerometro_xout_escalar = acelerometro_xout / 16384.0
138.         acelerometro_yout_escalar = acelerometro_yout / 16384.0
139.         acelerometro_zout_escalar = acelerometro_zout / 16384.0
140.
141.         ch1_valor = get_y_rotation(acelerometro_xout_escalar,
acelerometro_yout_escalar, acelerometro_zout_escalar)
142.
143.         # Gera vetor com os dados adquiridos
144.         ch_r = {
145.             "acoustic": str(ch0_valor),
146.             "vibration": str(ch1_valor)
147.         }
148.
149.         # Converte os dados para o formato json
150.         j = json.dumps(ch_r)
151.         line_write(sr_file, j)
152.
153.         ch_ac = ch_ac + str(ch0_valor) + ";" + str(ch1_valor) + "\n"
154.         i = i + 1
155.
156.         # Escreve a linha no arquivo designado
157.         if i == 1000:
158.             f.write(ch_ac)
159.             i = 1
160.
161.             final = time.time()
162.             tempo = final - inicio
163.             print str(tempo)
164.
165.             inicio = time.time()
166.             ch_ac = ""

```

monitoramento.py

```

1.  #!/usr/bin/python
2.  # -*- coding: utf-8 -*-
3.
4.  import matplotlib
5.  matplotlib.use("TkAgg")
6.  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
    NavigationToolbar2TkAgg
7.  from matplotlib.figure import Figure
8.  import matplotlib.animation as animation
9.  from matplotlib import style
10.
11. import tkinter as tk
12. from tkinter import ttk, messagebox, END
13.
14. import subprocess
15. from subprocess import PIPE, STDOUT
16.
17. LARGE_FONT= ("Verdana", 12)
18. style.use("ggplot")
19.
20. f_ch0 = Figure(figsize=(4,3), dpi=150)
21. f_ch1 = Figure(figsize=(4,3), dpi=150)
22. a = f_ch0.add_subplot(111)
23. b = f_ch1.add_subplot(111)
24.
25. ch0_calibracao = 0
26. ch1_calibracao = 0
27.
28. monitoramento_aquisicao = None
29.
30. def animate_ch0(i):
31.     pullData = open("/usr/lib/cgi-bin/ch0.txt", "r").read()
32.     dataList = pullData.split('\n')
33.     xList = []
34.     yList = []
35.     j = 0
36.
37.     for eachLine in dataList:
38.         if j > 29:
39.             break
40.
41.         if len(eachLine) > 1:
42.             x, y = eachLine.split(',')
43.             xList.append(float(x))
44.             yList.append(float(y) - ch0_calibracao)
45.             j = j + 1
46.
47.     a.clear()
48.     a.plot(xList, yList)
49.
50. def animate_ch1(i):
51.     pullData = open("/usr/lib/cgi-bin/ch1.txt", "r").read()
52.     dataList = pullData.split('\n')
53.     xList = []
54.     yList = []
55.     j = 0
56.
57.     for eachLine in dataList:
58.         if j > 29:
59.             break
60.         if len(eachLine) > 1:
61.             x, y = eachLine.split(',')

```

```

62.         xList.append(float(x))
63.         yList.append(float(y) - ch1_calibracao)
64.         j = j + 1
65.
66.     b.clear()
67.     b.plot(xList, yList)
68.
69.     def ch0_leitura():
70.         pullData = open("/usr/lib/cgi-bin/ch0.txt", "r").read()
71.         dataList = pullData.split('\n')
72.         return dataList[0].split(',')
73.
74.     def ch1_leitura():
75.         pullData = open("/usr/lib/cgi-bin/ch1.txt", "r").read()
76.         dataList = pullData.split('\n')
77.         return dataList[0].split(',')
78.
79.     def monitoramento_start():
80.         global monitoramento_aquisicao
81.         monitoramento_aquisicao = subprocess.Popen(['python', '/usr/lib/cgi-
bin/aquisicao.py'])
82.
83.     def monitoramento_stop():
84.         if monitoramento_aquisicao is not None:
85.             monitoramento_aquisicao.terminate()
86.
87.     def configuracao_salvar(cfg_ch0_calibracao, cfg_ch1_calibracao):
88.         global ch0_calibracao, ch1_calibracao
89.         ch0_calibracao = float(cfg_ch0_calibracao)
90.         ch1_calibracao = float(cfg_ch1_calibracao)
91.         messagebox.showinfo("Config", "Valores salvos com sucesso")
92.
93.     def on_closing():
94.         if messagebox.askokcancel("Sair", "Realmente deseja sair?"):
95.             if monitoramento_aquisicao is not None:
96.                 monitoramento_stop()
97.             app.destroy()
98.
99.     class FrmMain(tk.Tk):
100.         def __init__(self, *args, **kwargs):
101.
102.             tk.Tk.__init__(self, *args, **kwargs)
103.
104.             tk.Tk.iconbitmap(self)
105.             tk.Tk.wm_title(self, "Geração de gráficos")
106.
107.
108.             container = tk.Frame(self)
109.             container.pack(side="top", fill="both", expand = True)
110.             container.grid_rowconfigure(0, weight=1)
111.             container.grid_columnconfigure(0, weight=1)
112.
113.             self.frames = {}
114.
115.             for F in (FrmHome, FrmConfiguracao, FrmMonitoramentoGrafico):
116.
117.                 frame = F(container, self)
118.
119.                 self.frames[F] = frame
120.
121.                 frame.grid(row=0, column=0, sticky="nsew")
122.
123.             self.show_frame(FrmHome)
124.
125.         def show_frame(self, cont):
126.

```

```

127.         frame = self.frames[cont]
128.         frame.tkraise()
129.
130.
131.     class FrmHome(tk.Frame):
132.
133.         def __init__(self, parent, controller):
134.             tk.Frame.__init__(self, parent)
135.             label = tk.Label(self, text="Página Inicial", font=LARGE_FONT)
136.             label.pack(pady=10, padx=10)
137.
138.             button = ttk.Button(self, text="Configurações",
139.                                 command=lambda: controller.show_frame(FrmConfiguracao))
140.             button.pack()
141.
142.             button2 = ttk.Button(self, text="Monitoramento",
143.                                  command=lambda:
144.                  controller.show_frame(FrmMonitoramentoGrafico))
145.             button2.pack()
146.
147.     class FrmConfiguracao(tk.Frame):
148.
149.         def __init__(self, parent, controller):
150.             tk.Frame.__init__(self, parent)
151.             label = tk.Label(self, text="Configurações", font=LARGE_FONT)
152.             label.pack(pady=10, padx=10)
153.
154.             button1 = ttk.Button(self, text="Retornar ao inicio",
155.                                  command=lambda: controller.show_frame(FrmHome))
156.             button1.pack()
157.
158.             button2 = ttk.Button(self, text="Monitoramento",
159.                                  command=lambda:
160.                  controller.show_frame(FrmMonitoramentoGrafico))
161.             button2.pack()
162.
163.             lb1 = ttk.Label(self, text="Calibrar emissão acústica")
164.             lb1.pack()
165.             lb1['text'] = ("Calibrar emissão acústica - atual:
166. {0:s}").format(ch0_leitura()[1])
167.
168.             txt1 = tk.Text(self, height=1, width=30, wrap='none')
169.             txt1.pack()
170.
171.             lb2 = ttk.Label(self, text="Calibrar giroscópio")
172.             lb2.pack()
173.             lb2['text'] = ("Calibrar giroscópio - atual: {0:s}").format(ch1_leitura()[1])
174.
175.             txt2 = tk.Text(self, height=1, width=30, wrap='none')
176.             txt2.pack()
177.
178.             txt1.insert(END, str(ch0_calibracao))
179.             txt2.insert(END, str(ch1_calibracao))
180.
181.             button3 = ttk.Button(self, text="Salvar", command=lambda:
182.                  configuracao_salvar(txt1.get("1.0", END), txt2.get("1.0", END)))
183.             button3.pack()
184.
185.     class FrmMonitoramentoGrafico(tk.Frame):
186.
187.         def __init__(self, parent, controller):
188.             tk.Frame.__init__(self, parent)
189.             label = tk.Label(self, text="Monitoramento da Usinagem", font=LARGE_FONT)
190.             label.pack(pady=10, padx=10)

```

```

189.
190.     button1 = ttk.Button(self, text="Retornar ao inicio",
191.                           command=lambda: controller.show_frame(FrmHome))
192.     button1.pack()
193.
194.     button3 = ttk.Button(self, text="Iniciar monitoramento",
195.                           command=monitoramento_start)
196.     button3.pack()
197.
198.     button2 = ttk.Button(self, text="Parar monitoramento",
199.                           command=monitoramento_stop)
200.     button2.pack()
201.
202.     label_ch0 = tk.Label(self, text="Emissão acústica (dB SPL)", font=LARGE_FONT)
203.     label_ch0.pack(pady=10, padx=10)
204.
205.     canvas_ch0 = FigureCanvasTkAgg(f_ch0, self)
206.     canvas_ch0.draw()
207.     canvas_ch0.get_tk_widget().pack(fill=tk.X, expand=True)
208.
209.     label_ch1 = tk.Label(self, text="Vibração (graus em Y)", font=LARGE_FONT)
210.     label_ch1.pack(pady=10, padx=10)
211.
212.     canvas_ch1 = FigureCanvasTkAgg(f_ch1, self)
213.     canvas_ch1.draw()
214.     canvas_ch1.get_tk_widget().pack(fill=tk.X, expand=True)
215.
216.
217. app = FrmMain()
218. ani_ch0 = animation.FuncAnimation(f_ch0, animate_ch0, interval=500)
219. ani_ch1 = animation.FuncAnimation(f_ch1, animate_ch1, interval=500)
220. app.protocol("WM_DELETE_WINDOW", on_closing)
221. app.mainloop()

```

APÊNDICE B – CÓDIGO-FONTE DO PROGRAMA PARA MONITORAMENTO DESENVOLVIDO NA LINGUAGEM HTML

monitoring.html

```

1.  <!DOCTYPE html>
2.  <html>
3.  <head>
4.      <meta charset="utf-8" />
5.      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6.
7.      <title>Data Acquisition with Raspberry Pi</title>
8.      <style type="text/css">
9.  body{font-family:Arial,Helvetica,sans-serif}h2{text-
    align:center;margin:0}div.value{margin-top:5px;text-align:right;margin-
    right:5%}div.value span{padding:3px;border:2px solid gray;font-
    weight:700}div.monitor{display:block;padding:10px;margin-bottom:15px;border:15px
    solid gray;border-radius:10px}.td{text-align:center}div.dbtn{padding:10px
    0;width:100%}textarea{display:none;padding:10px;font-family:Arial,Helvetica,sans-
    serif;font-size:16px;height:300px;width:500px;max-width:95%;margin-
    bottom:15px;border:1px solid #000;border-radius:5px;outline:0;resize:none;overflow-
    y:auto}a.btn{display:inline-block;margin:0 auto;box-shadow:inset 0 1px 0 0
    #54a3f7;background:linear-gradient(to bottom,#007dc1 5%,#0061a7 100%);background-
    color:#007dc1;border-radius:3px;border:1px solid
    #124d77;cursor:pointer;color:#fff;font-family:Arial;font-size:16px;font-
    weight:700;padding:8px 24px;text-decoration:none;text-shadow:0 1px 0
    #154682}a.btn:hover{background:linear-gradient(to bottom,#0061a7 5%,#007dc1
    100%);background-color:#0061a7}a.btn:active{position:relative;top:2px}.no-
    show{display:none}
10.     </style>
11.     <script
12.     src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
13.     <script src="https://www.gstatic.com/charts/loader.js"></script>
14.     </head>
15.     <body>
16.     <div class="tc dbtn">
17.         <a href="#" class="btn" id="btn-pause-resume">Pause</a>
18.         <a href="#" class="btn" id="btn-value-export">View data</a>
19.         <a href="#" class="btn" id="btn-value-export-csv">Download csv</a>
20.         <a href="http://ENDereco_DO_RASPBERRY_PI/monitoring.txt" class="btn"
21.         target="_blank">Download all data</a>
22.     </div>
23.     <div><textarea id="value-export">timestamp;acoustic;vibration</textarea></div>
24.     <div class="no-show">
25.         <h2>Temperature</h2>
26.         <div id="monitor-temperature"></div>
27.         <div class="value" id="monitor-temperature-value"></div>
28.     </div>
29.     <div class="monitor">
30.         <h2>Acoustic Emission</h2>
31.         <div id="monitor-acoustic"></div>
32.         <div class="value" id="monitor-acoustic-value"></div>
33.     </div>
34.
35.     <div class="monitor">
36.         <h2>Vibration</h2>
37.         <div id="monitor-vibration"></div>
38.         <div class="value" id="monitor-vibration-value"></div>
39.     </div>
40.     <script>

```

```

41. google.charts.load("current",{callback:function(){var e=new
google.visualization.LineChart(document.getElementById("monitor-acoustic")),t=new
google.visualization.LineChart(document.getElementById("monitor-
vibration")),a={curveType:"function",animation:{duration:0,easing:"out",startup:!0},h
Axis:{title:"Time"},vAxis:{title:"Acoustic (db-
spl)"},series:{0:{targetAxisIndex:1,color:"#02CD20"}},height:300,legend:{position:"no
ne"},chartArea:{left:0,width:"95%",height:"95%"}},n={curveType:"function",animation:{
duration:0,easing:"out",startup:!0},hAxis:{title:"Time"},vAxis:{title:"Vibration
(°)"},series:{0:{targetAxisIndex:1,color:"navy"}},height:300,legend:{position:"none"}
,chartArea:{left:0,width:"95%",height:"95%"}},o=new
google.visualization.DataTable;o.addColumn("datetime","Time"),o.addColumn("number","T
emperature");var i=new
google.visualization.DataTable;i.addColumn("datetime","Time"),i.addColumn("number","A
coustic");var r=new
google.visualization.DataTable;r.addColumn("datetime","Time"),r.addColumn("number","V
ibration");var l=new google.visualization.DateFormat({pattern:"hh:mm:ss"}),u=new
google.visualization.NumberFormat({pattern:"#,##0.0"});d();var
s=setInterval(d,500);function c(e){var
t=""+(e.getMonth()+1),a=""+e.getDate(),n=e.getFullYear(),o=""+e.getHours(),i=""+e.get
Minutes(),r=""+e.getSeconds();return
t=t.length<2?"0"+t:t,a=a.length<2?"0"+a:a,o=o.length<2?"0"+o:o,i=i.length<2?"0"+i:i,r
=r.length<2?"0"+r:r,[a,t,n].join("/")+" "+[o,i,r].join(":")}function d(){var
o=null,l=null,u=!1,s=new
Date;$ajax({url:"http://ENDereco_DO_RASPBERRY_PI/monitoring.php",dataType:"json",asy
nc:!1,success:function(e){u=!0,o=parseFloat(e.acoustic),l=parseFloat(e.vibration),$("
textarea#value-export").val($("#textarea#value-
export").val()+"\n"+[c(s,o,l).join(";")]}},l=u&&($("#div#monitor-acoustic-
value").html("<span>"+Math.round(100*(o+Number.EPSILON))/100+"</span>"),$("#div#monito
r-vibration-
value").html("<span>"+Math.round(100*(l+Number.EPSILON))/100+"</span>"),m(e,i,a,s,o),
m(t,r,n,s,l))}function
m(e,t,a,n,o){t.addRow([n,o]),t.getNumberOfRows()>30&&t.removeRow(0),l.format(t,0),u.f
ormat(t,1),e.draw(t,a)}$("#a#btn-pause-
resume").click(function(){if("Pause"==$(this).html().trim()){$(this).html("Resume"),clea
rInterval(s),s=0:}($("#this).html("Pause"),s=setInterval(d,500))}),$("#a#btn-value-
export").click(function(){if("block"==$("#textarea#value-
export").css("display")){$("#textarea#value-
export").css("display","none");$("#a#btn-value-export-csv").click(function(){var
e=document.createElement("a");e.setAttribute("href","data:text/csv;charset=utf-
8,"+encodeURIComponent($("#textarea#value-
export").val().replace(/\./g,"")),e.setAttribute("download","data_monitoring.csv"),
e.style.display="none",document.body.appendChild(e),e.click(),document.body.removeChi
ld(e)},$("#textarea#value-
export").click(function(){$(this).select()}}},packages:["corechart"]});
42. </script>
43. </body>
44. </html>

```