



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
DEPARTAMENTO DE ENGENHARIAS DA MOBILIDADE

Anderson Vinícius de Oliveira Rosa

**DESENVOLVIMENTO DE DISPOSITIVO REMOTO DE BAIXO CUSTO PARA
SENSORIAMENTO DE CARGAS DE TORQUE**

Joinville, Santa Catarina – Brasil
2022

Anderson Vinícius de Oliveira Rosa

**DESENVOLVIMENTO DE DISPOSITIVO REMOTO DE BAIXO CUSTO PARA
SENSORIAMENTO DE CARGAS DE TORQUE**

Trabalho de Conclusão de Curso submetido ao Departamento de Engenharias da Mobilidade da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Engenharia Automotiva.

Orientador: Andrea Piga Carboni, Dr.

Coorientador: Marcos Alves Rabelo, Dr.

Joinville, Santa Catarina – Brasil
2022

Catalogação na fonte pela Biblioteca Universitária da Universidade Federal de Santa Catarina.

Anderson Vinícius de Oliveira Rosa

Desenvolvimento de dispositivo remoto de baixo custo para sensoriamento de cargas de torque / Anderson Vinícius de Oliveira Rosa; Orientador, Andrea Piga Carboni, Dr.; Coorientador, Marcos Alves Rabelo, Dr. - Joinville, Santa Catarina - Brasil, 17 de março de 2022.

86 p.

Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, CTJ – Centro Tecnológico de Joinville, Departamento de Engenharias da Mobilidade.

Inclui referências

1. Palavra-Chave 1, 2. Palavra-Chave 2, 3. Palavra-Chave 3, I. Andrea Piga Carboni, Dr. II. Marcos Alves Rabelo, Dr. III. Departamento de Engenharias da Mobilidade IV. Desenvolvimento de dispositivo remoto de baixo custo para sensoriamento de cargas de torque

Anderson Vinícius de Oliveira Rosa

**DESENVOLVIMENTO DE DISPOSITIVO REMOTO DE BAIXO CUSTO PARA
SENSORIAMENTO DE CARGAS DE TORQUE**

Este(a) Trabalho de Conclusão de Curso foi julgado adequado(a) para obtenção do Título de Bacharel em Engenharia Automotiva, e foi aprovado em sua forma final pelo Departamento de Engenharias da Mobilidade do CTJ – Centro Tecnológico de Joinville da Universidade Federal de Santa Catarina.

Joinville, Santa Catarina – Brasil, 17 de março de 2022.

Leonardo Moreto Elias, Dr.
Coordenador de TCC da Engenharia
Automotiva

Banca Examinadora:

Andrea Piga Carboni, Dr.
Orientador
Universidade Federal de Santa
Catarina – UFSC

Antônio Otaviano Dourado, Dr.
Universidade Federal de Santa
Catarina – UFSC

Sérgio Junichi Idehara, Dr.
Universidade Federal de Santa
Catarina – UFSC

AGRADECIMENTOS

Os agradecimentos principais são direcionados à Gerald Weber, Miguel Frasson, Leslie H. Watter, Bruno Parente Lima, Flávio de Vasconcellos Corrêa, Otavio Real Salvador, Renato Machnievscz¹ e todos aqueles que contribuíram para que a produção de trabalhos acadêmicos conforme as normas ABNT com L^AT_EX fosse possível.

Agradecimentos especiais são direcionados ao Centro de Pesquisa em Arquitetura da Informação² da Universidade de Brasília (CPAI), ao grupo de usuários *latex-br*³ e aos novos voluntários do grupo *abnT_EX2*⁴ que contribuíram e que ainda contribuirão para a evolução do abnT_EX2.

¹ Os nomes dos integrantes do primeiro projeto abnT_EX foram extraídos de <http://codigolivre.org.br/projects/abntex/>

² <http://www.cpai.unb.br/>

³ <http://groups.google.com/group/latex-br>

⁴ <http://groups.google.com/group/abntex2> e <http://abntex2.googlecode.com/>

RESUMO

TBD

Palavras-chaves: Palavra-Chave 1. Palavra-Chave 2. Palavra-Chave 3.

ABSTRACT

TBD

Keywords: Keyword 1. Keyword 2. Keyword 3.

LISTA DE FIGURAS

Figura 1	- Forças internas atuando em um corpo em equilíbrio	17
Figura 2	- Exemplo de diagrama tensão-deformação	18
Figura 3	- Deflexão de uma barra sob carga de tração	19
Figura 4	- Representação do efeito da deflexão em uma viga sob flexão . .	20
Figura 5	- Deflexão e distribuição de tensão em uma sob flexão	21
Figura 6	- Extensômetro fixado á um corpo de prova	22
Figura 7	- Ponte de Wheatstone	23
Figura 8	- Princípio de funcionamento de um amplificador de sinal	25
Figura 9	- Ilustração de ruidos presentes em sinais analógicos e digitais .	26
Figura 10	- Representação gráfica de um sinal analógico em forma digital .	26
Figura 11	- Diagrama de blocos dos estágios de um sistema de medição . . .	27
Figura 12	- Controlador ESP8266	28
Figura 13	- Conversor analógico digital HX711	29
Figura 14	- Regressão linear	33
Figura 15	- Etapas da metodologia PRODIP	34
Figura 16	- Etapas do projeto informacional	35
Figura 17	- Etapas do projeto conceitual	36
Figura 18	- Datalogger DMI TCR 44es	38
Figura 19	- Dispositivo Bridge101A	39
Figura 20	- Projeto inicial do dispositivo	41
Figura 21	- Matriz de listagem de atributos	44
Figura 22	- Matriz QFD	45
Figura 23	- Conceito de funcionamento inicial	46
Figura 24	- Matriz morfológica	48
Figura 25	- Matriz de avaliação	49
Figura 26	- Soluções escolhidas	50
Figura 27	- Modelagem da montagem do dispositivo	51
Figura 28	- Desenho em vistas padrões (primeiro diedro)	52
Figura 29	- Projeto elétrico do dispositivo	53
Figura 30	- Esquema da programação do dispositivo	54
Figura 31	- Objeto NetworkDevice	55
Figura 32	- Interface gráfica	59
Figura 33	- Protótipo preparado para os experimentos	60
Figura 34	- Dispositivo de ensaio de flexão	61
Figura 35	- Sistema de medição utilizado por (MINELA, 2017)	62
Figura 36	- Sistema de de medição utilizando o LINX ADS2002	63
Figura 37	- Protótipo do dispositivo desenvolvido conectado com o extenso- metro no dispositivo de flexão	64

Figura 38 – Aplicaçao do 'peso 3' no dispositivo de flexão	66
Figura 39 – Sinal obtido sem a aplicação de cargas no dispositivo	68
Figura 40 – Sinal obtido pela aplicação da cargas de calibração baixa	68
Figura 41 – Sinal obtido pela aplicação da cargas de calibração alta	69
Figura 42 – Amostras e valores médios obtidos	69
Figura 43 – Função de ajuste de dados de deformação em relação aos valores obtidos no amplificador	70
Figura 44 – Comparação dos valores convertidos usando diferentes funções de ajuste	71
Figura 45 – Erro entre valores encontrados com a função de ajuste e os valores esperados pela análise analítica	71

LISTA DE TABELAS

Tabela 1	– Funções do pacote NumPy utilizadas	31
Tabela 2	– Comparação entre dispositivos encontrados	38
Tabela 3	– Resultado pesquisa de mercado	43
Tabela 4	– Prioridade dos requisitos definidos pela matriz QFD	45
Tabela 5	– Levantamento de componentes	47
Tabela 7	– Valores de massas utilizadas para aplicação das cargas nos dispositivos de flexão e torção	65

LISTA DE CÓDIGOS

Código 1	– Controller Program	54
Código 2	– NetworkDevice	55
Código 3	– Network Device calibrate offset	56
Código 4	– Network Device calibrate offset	58

LISTA DE SÍMBOLOS

Γ Letra grega Gama

Λ Lambda

ζ Letra grega minúscula zeta

\in Pertence

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
2	REFERENCIAL TEÓRICO	16
2.1	SOLICITAÇÕES E RESISTÊNCIA DOS MATERIAIS	16
2.1.1	Deformação e limites do material	17
2.1.2	Deformação de uma viga em flexão	19
2.2	EXTENSOMETRIA	21
2.2.1	Ponte de Wheatstone	23
2.3	OBTENÇÃO DE SINAIS	25
2.4	SISTEMAS DE MEDAÇÃO	27
2.5	ANÁLISE DOS SINAIS OBTIDOS	30
2.5.1	Ambiente de desenvolvimento computacional Python	31
2.5.2	Análise dos valores nominais	32
2.6	METODOLOGIA DE DESENVOLVIMENTO DE PROJETO DE PRODUTO	32
2.6.1	Fase de planejamento	33
2.6.2	Projeto informacional	34
2.6.3	Projeto conceitual	34
2.6.4	Projeto preliminar	35
2.6.5	Projeto detalhado	35
3	DESENVOLVIMENTO	37
3.1	PLANEJAMENTO DO PROJETO	37
3.1.1	Mapeamento tecnológico	37
3.1.2	Pesquisa científica	39
3.2	PROJETO INFORMACIONAL	41
3.3	PROJETO CONCEITUAL	46
3.4	PROJETO DETALHADO	50
3.5	VALIDAÇÃO DO PROTÓTIPO DO DISPOSITIVO	59
3.5.1	Dispositivo para ensaio de flexão	61
3.5.2	Sistemas de medição	62
3.5.3	Cargas aplicadas	64
4	RESULTADOS	67
4.1	SINAIS OBTIDOS	67

4.1.1	Sinais de calibração	67
4.1.2	Ruídos presentes	67
4.1.3	Valores nominais	67
4.2	FUNÇÕES DE AJUSTE	68
5	CONSIDERAÇÕES FINAIS	72
	REFERÊNCIAS	73
	APÊNDICE A – CODIGODISPOSITIVO	76
	APÊNDICE B – CODIGOSOFTWARE	78
B.1	OBJETO NETWORKDEVICE	78
B.2	INTERFACE GRÁFICA	80

1 INTRODUÇÃO

Atualmente, é notado o constante aumento da importância da otimização dos projetos de componentes em projetos de produtos na indústria automotiva, produtos altamente otimizados resultam em um menor custo de material e de fabricação dos componentes, (HIBBELE, 2010) afirma que “a carga para a qual um elemento é projetado pode ser diferente das cargas realmente aplicadas. As dimensões estipuladas no projeto de uma estrutura ou máquina podem não ser exatas, na realidade, por causa de erros de fabricação ou cometidos na montagem de seus componentes”.

Segundo (HIBBELE, 2010) “Para se garantir a segurança, é preciso escolher uma tensão admissível que restrinja a carga aplicada a um valor menor do que a carga que o elemento pode suportar totalmente.” Então como resposta às incertezas envolvidas no projeto analítico de um componente os projetistas devem projetar componentes que suportam forças superiores às presentes na utilização do componente. O que resulta em altos valores de fator de segurança em um projeto, o que causa impacto monetário e aumento de massa do componente. Uma das maneiras que permite a diminuição de valores de fator de segurança é a alimentação do projeto do componente com dados de cargas que representam o mais próximo o possível aos presentes na situação real.

Dados reais de utilização podem ser obtidos por sensores em componentes reais ou de teste submetidos a situações reais, porém atualmente certos parâmetros não podem ser facilmente medidos de maneira direta em um veículo, dentre eles forças normais e forças torcionais (NURPRASETIO *et al.*, 2018). Hibbeler afirma que “as medições de deformação são experimentais e, uma vez obtidas, podem ser relacionadas com as cargas aplicadas, ou tensões, que agem no interior do corpo.” Logo conclui-se que uma maneira direta de medir as forças internas atuantes em um componentes é obtendo os dados de deformação local.

Foi observado que dispositivos utilizados para obter dados de deformação em tempo real com precisão são usualmente utilizados em testes de impacto e de controle de qualidade em componentes pela indústria automotiva, esses dispositivos apresentam altos níveis de precisão e confiabilidade e, consequentemente altos custos, o que inviabiliza sua utilização fora do produto final. Os valores de deformação local em um componente podem ser obtidos utilizando sensores de deformação chamados de extensômetros, esses sensores apresentam uma boa disponibilidade no mercado e são amplamente utilizados em células de carga. Os sinais gerados por esse tipo de sensor devem ser instrumentados, ampliados e convertidos para possibilitar sua obtenção por uma interface controladora.

O presente trabalho propõe o desenvolvimento de um protótipo de um dispositivo de baixo custo para obtenção de dados de deformação em componentes. O desen-

volvimento do dispositivo seguirá a metodologia de projeto de produto PRODIP com o objetivo de garantir replicabilidade, permitir futuras otimizações e expansões e facilitar sua implementação em um caso real. Por fim, o funcionamento, efetividade e precisão do protótipo desenvolvido será avaliado comparando dados obtidos pelo protótipo e por um dispositivo industrial homologado, seguindo a metodologia de ([MINELA, 2017](#)).

1.1 OBJETIVOS

Os objetivos do trabalho são apresentados nas seções a seguir.

1.1.1 Objetivo Geral

Desenvolver um dispositivo de baixo custo para obtenção de dados em tempo real de deflexão em componentes mecânicos.

1.1.2 Objetivos Específicos

Obter dados de deflexão em vigas

Obter módulo de elasticidade de uma liga desconhecida

Desenvolver utilizando tecnologias de código aberto

Obter valores de precisão do protótipo desenvolvido

2 REFERENCIAL TEÓRICO

O presente capítulo apresenta os conceitos teóricos necessários para o desenvolvimento do princípio de funcionamento do dispositivo. São apresentados tópicos referentes a solicitações mecânicas e resistência dos materiais, princípios de sensoriamento de deformação e instrumentação de extensômetros, obtenção de sinais e transmissão de dados. Também será apresentado as principais tecnologias necessárias para o processo de desenvolvimento do dispositivo.

2.1 SOLICITAÇÕES E RESISTÊNCIA DOS MATERIAIS

Um entendimento introdutório sobre resistência dos materiais é necessário de modo a entender sobre os comportamentos físicos de um componente mecânico que sofre a ação de cargas externas. O ponto de partida do estudo da resistência dos materiais é o da análise do comportamento mecânico de um componente em equilíbrio.

Utilizando as equações de estática, deve-se determinar as forças e os momentos resultantes que agem no interior de um corpo, com a finalidade de verificar e garantir a integridade do mesmo durante o uso (HIBBELER, 2010). Um corpo em equilíbrio, deve satisfazer a [Equação \(1\)](#) e [Equação \(2\)](#), que descrevem o balanço estático conforme a segunda lei de newton.

$$\sum F_x = \sum F_y = \sum F_z = 0 \quad (1)$$

$$\sum M_x = \sum M_y = \sum M_z = 0 \quad (2)$$

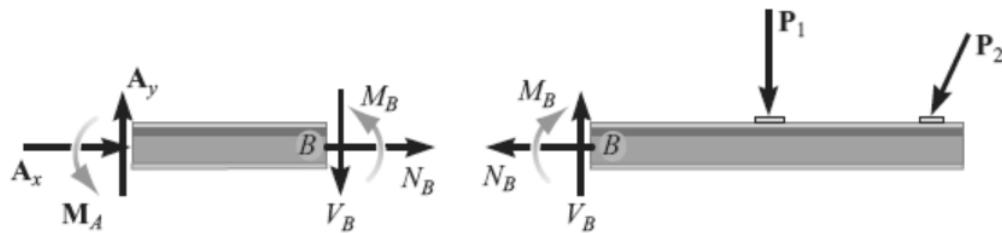
onde

F_i : Forças axiais aplicadas no corpo no eixo "i"

M_i : Momentos aplicados no corpo no eixo "i"

Para ser mantida a condição de integridade do corpo do material sobre forças externas devem estar presentes forças e momentos internos ao seu corpo, ([HIBBELER, 2010](#)) ressalta que “Uma das mais importantes aplicações da estática na análise de problemas de resistência dos materiais é poder determinar a força e o momento resultantes que agem no interior de um corpo e que são necessários para manter a integridade do corpo quando submetido a cargas externas” e que “a força e o momento que agem em um ponto específico da área seccionada de um corpo representam os efeitos resultantes da distribuição de forças que agem sobre a área seccionada”. A [Figura 1](#) apresenta uma representação gráfica da atuação de forças internas em um material:

Figura 1 – Forças internas atuando em um corpo em equilíbrio



Fonte: ([HIBBEKER, 2010](#))

Uma vez que se tem a informação das forças internas atuantes em um ponto no corpo e na seção do material, então, pode-se partir para a análise das tensões e deformações do local de análise.

2.1.1 Deformação e limites do material

Quando um segmento de um corpo sob balanço estático se encontra sob a ação de forças internas, este segmento apresentará uma variação de seu comprimento relativo à força aplicada. Deformação é definido como a mudança de comprimento por unidade de comprimento, logo, é um valor adimensional, e é calculada pela [Equação \(3\)](#).([NORTON, 2011](#))

$$\varepsilon = \frac{l-l_0}{l_0} \quad (3)$$

onde

ε : Deformação

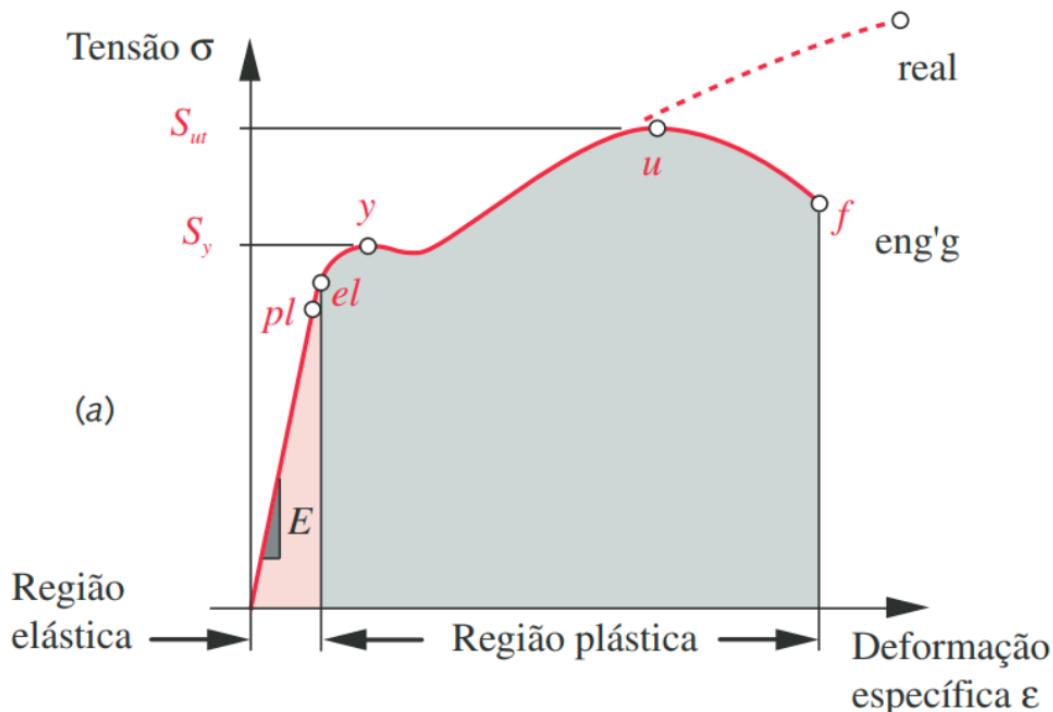
l : Comprimento da barra após deformação

l_0 : Comprimento da barra sem deformação

Com o objetivo de descobrir os limites no qual um material pode-se deformar antes de sua ruptura devem ser analisados seus diagramas tensão-deformação. Hibbeler ressalta a importância na análise desse tipo de diagrama, uma vez que eles proporcionam meios para a obtenção de dados sobre resistência à tração ou compressão de um material sem independentemente de suas características físicas e geométricas ([HIBBEKER, 2010](#)). Um exemplo de diagrama tensão-deformação é mostrado na [Figura 2](#).

Analizando o diagrama anterior pode-se notar uma zona de relacionamento linear entre a força aplicada no corpo de prova utilizado para construir o diagrama e sua deformação, nesta região é observado o comportamento de deformação elástica do

Figura 2 – Exemplo de diagrama tensão-deformação



Fonte: (NORTON, 2011)

material e sobre seu limite (NORTON, 2011) afirma que os pontos pl e el “marca o limiar entre as regiões de comportamento elástico e comportamento plástico do material. Os pontos el e pl normalmente são tão próximos que eles quase sempre são considerados o mesmo.”

Na maior parte dos materiais de engenharia é verificada uma relação linear entre deformação e tensão dentro da região elástica, logo, um aumento proporcional na força aplicada em um material resulta em um aumento proporcional das deformações locais caso a condição de tensão esteja dentro do limite elástico, esse fato foi descoberto por Robert Hooke, em 1676, em molas e é conhecido como Lei de Hooke (HIBBELER, 2010). A lei de Hooke é apresentada na Equação (4).

$$\sigma = E\varepsilon \quad (4)$$

onde

σ : Tensão interna do material

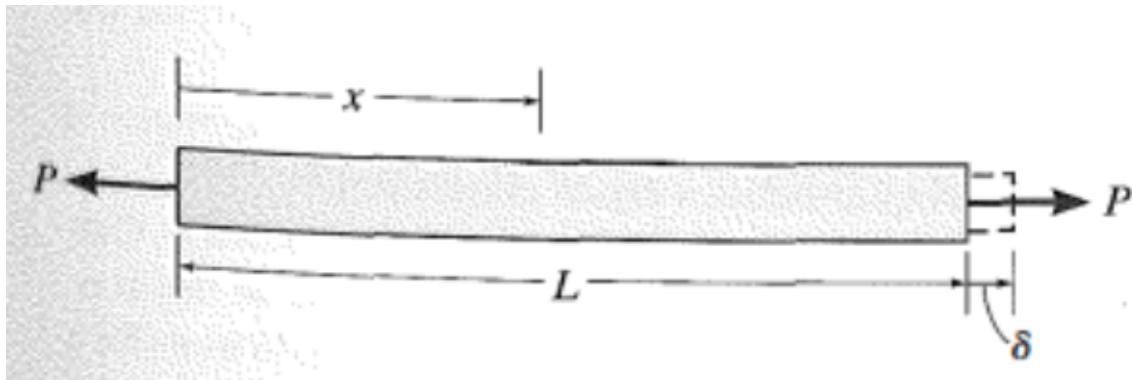
E : Módulo de elasticidade do material

ε : Deformação presente no material

A variável E da equação da Lei de Hooke é representa a inclinação da curva tensão-deformação e é chamada de Módulo de Young, ou módulo de elasticidade do material (NORTON, 2011). Norton também afirma que o Módulo de Young “é uma medida da rigidez do material em sua região elástica e tem as mesmas unidades da tensão. A maioria dos metais exibe esse comportamento linear e também tem módulos de elasticidade que variam muito pouco com tratamentos térmicos ou com a adição de elementos de liga.”

Para uma barra constituída de um material homogêneo e isotrópico e submetida a forças axiais que tem seu centro de atuação no centro da seção da barra essas cargas irão gerar uma tensão normal uniforme ao longo do seu comprimento sobre a seção transversal (HIBBEKER, 2010).

Figura 3 – Deflexão de uma barra sob carga de tração



Fonte: (HIBBEKER, 2010)

O alongamento ou contração de um segmento de reta por unidade de comprimento é denominado deformação normal e segue a Equação (5).

$$\delta = \frac{PL}{AE} \quad (5)$$

onde

δ : Deformação

P : Carga de tração aplicada na barra

A : Área da seção transversal

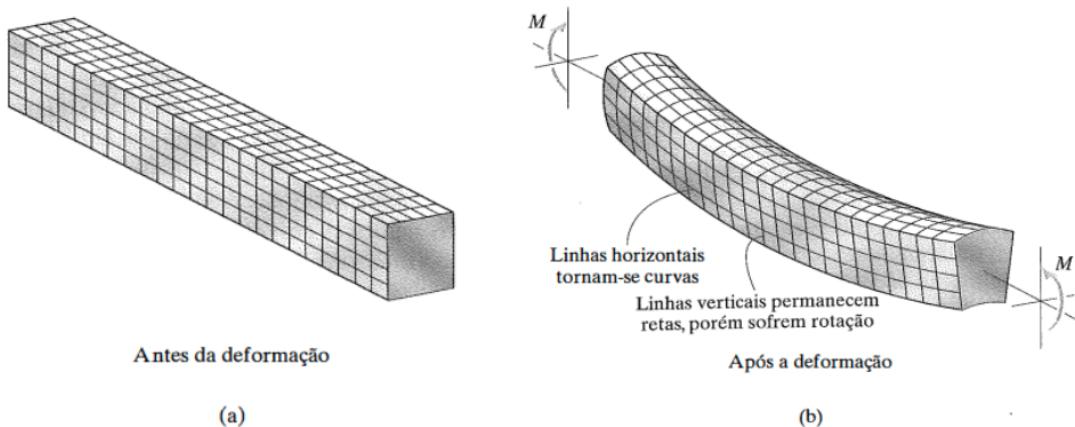
L : Comprimento da barra

2.1.2 Deformação de uma viga em flexão

A flexão é presente em um corpo sempre que as forças não são aplicadas na direção normal da sua seção transversal. Segundo (HIBBEKER, 2010) “O momento

fletor é causado pelas cargas externas que tendem a fletir o corpo em torno de um eixo que se encontra no plano da área.” e que nesse momento “tende a produzir uma variação linear da deformação normal no interior de uma viga”. A [Figura 4](#) mostra uma representação ilustrativa do efeito do momento fletor em uma viga.

Figura 4 – Representação do efeito da deflexão em uma viga sob flexão



Fonte: ([HIBBELER, 2010](#))

Em todo caso em que o material seja homogêneo e isotrópico e que a Lei de Hooke seja aplicável, pode-se relacionar o momento fletor presente com a distribuição de tensão na seção ([HIBBELER, 2010](#)). Deve-se notar que assim como a barra em torção, a tensão, e eventualmente a deflexão presente será função da distância entre o ponto de interesse e o centro da área da seção transversal do material. A [Equação \(6\)](#) caracteriza a distribuição de tensão ao longo da seção do componente.

$$\sigma = -\frac{My}{I} \quad (6)$$

onde

M : Momento fletor

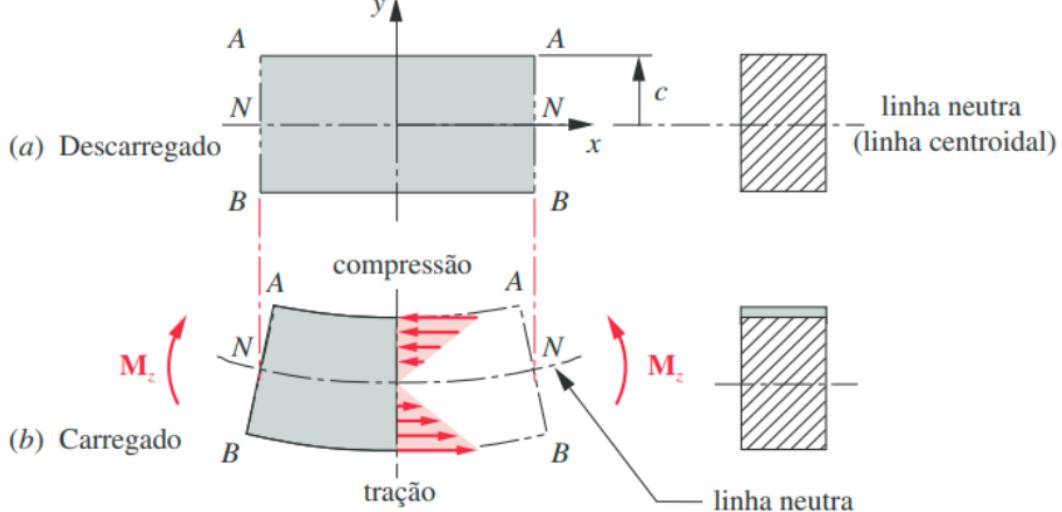
y : Distância entre limite da área e seu centróide

I : Momento de inércia da seção transversal

O valor de I é igual ao momento de inércia da seção do material sobre carga de flexão e a variável y representa a distância entre o centróide da área e o ponto de análise de tensão, deve-se notar que as tensões máximas para qualquer corpo em flexão sempre acontecerão na superfície do material, e que enquanto um ponto qualquer está sob forças de tração, o ponto inverso a este estará sob forças de compressão. Uma vez conhecido o módulo de elasticidade do material e a distribuição de tensão de na

Figura 5 – Deflexão e distribuição de tensão em uma sob flexão

10



Fonte: (NORTON, 2011)

seção de um corpo sob flexão, pode-se obter, utilizando a lei de hooke, os valores de deflexão causados pelas cargas de flexão.

As deformações em um componente podem ser altamente visíveis ou praticamente imperceptíveis se não forem utilizados equipamentos que façam medições precisas (HIBBELER, 2010). Considerando essa afirmação deve-se também ser estudado o método experimental de obtenção de dados de deflexão nos componentes.

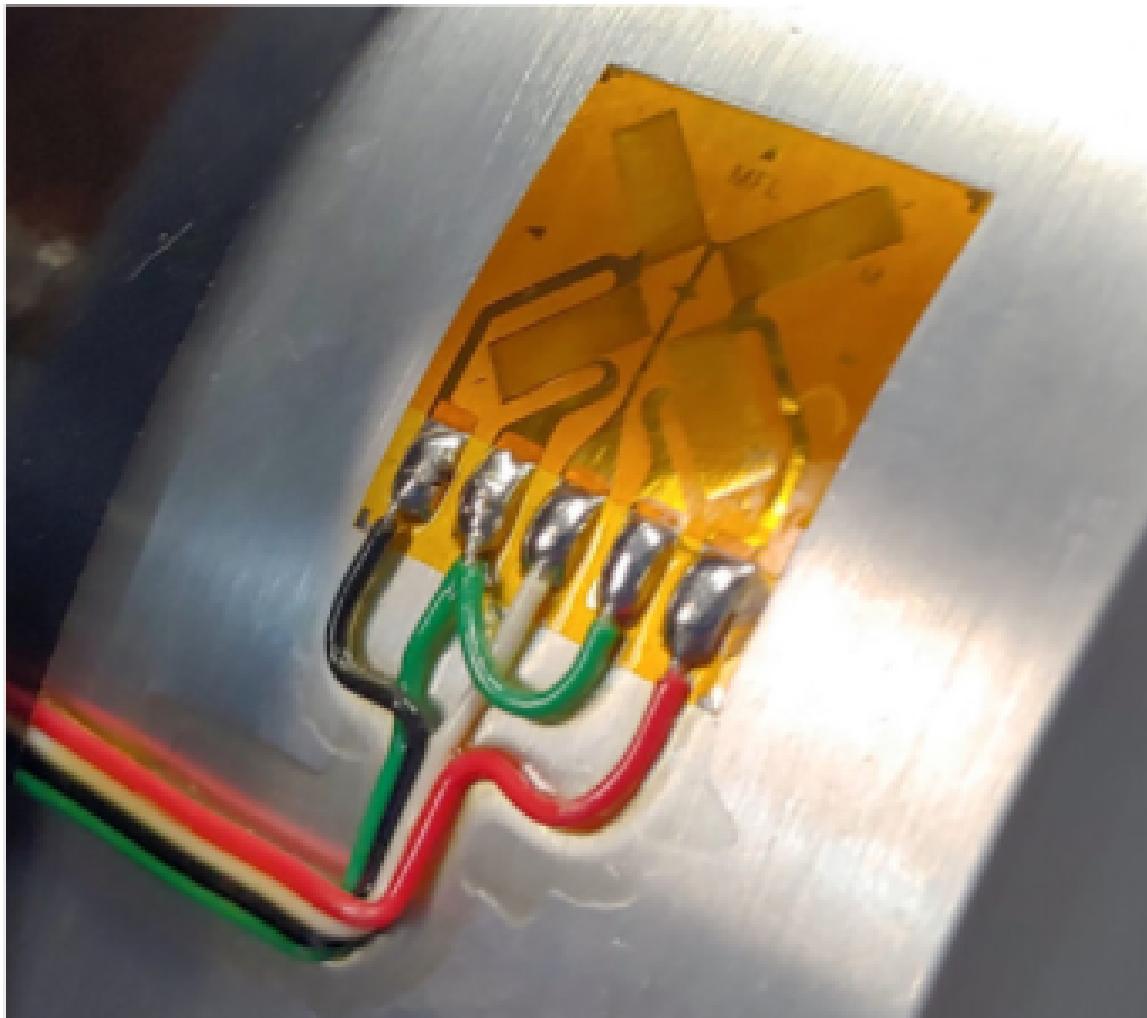
2.2 EXTENSOMETRIA

O extensômetro de resistência elétrica é o dispositivo mais utilizado para medir a deflexão em uma superfície, o princípio de funcionamento desse tipo de sensor é baseado no efeito de variação de resistência elétrica de um condutor quando ocorre uma variação de área da sua seção transversal (HOLLMAN, 2011).

Caso um extensômetro esteja fortemente fixado a um corpo de um material em uma direção específica, qualquer carga que deforma a superfície desse corpo de prova irá deformar igualmente o extensômetro, logo pode-se considerar o extensômetro como uma parte integrante do corpo de prova e qualquer deformação que aconteça no corpo de prova acontecerá igual no extensômetro (HIBBELER, 2010).

O fator de extensão ou gage factor, parâmetro que especifica a relação entre a variação da resistência nominal em um extensômetro para um valor unitário de deflexão, é um valor especificado pelo fabricante, então, e a resistência nominal do extensômetro são valores especificados pelo fabricante do sensor, então medindo

Figura 6 – Extensômetro fixado á um corpo de prova



Fonte: <https://binsfeld.com/category/strain-gage/> acesso em fev 2022

um valor e variação de resistência elétrica no extensômetro pode-se obter um valor de deformação local (HOLLMAN, 2011). A Equação (7) mostra uma relação entre a variação de resistência elétrica no extensômetro e os parâmetros repassados pelo fabricante.

$$K\varepsilon = \frac{\Delta R_s}{R_s} \quad (7)$$

sendo

K : Gage Factor

ε : Deformação no extensômetro

ΔR_s : Variação de resistência causada pela deformação

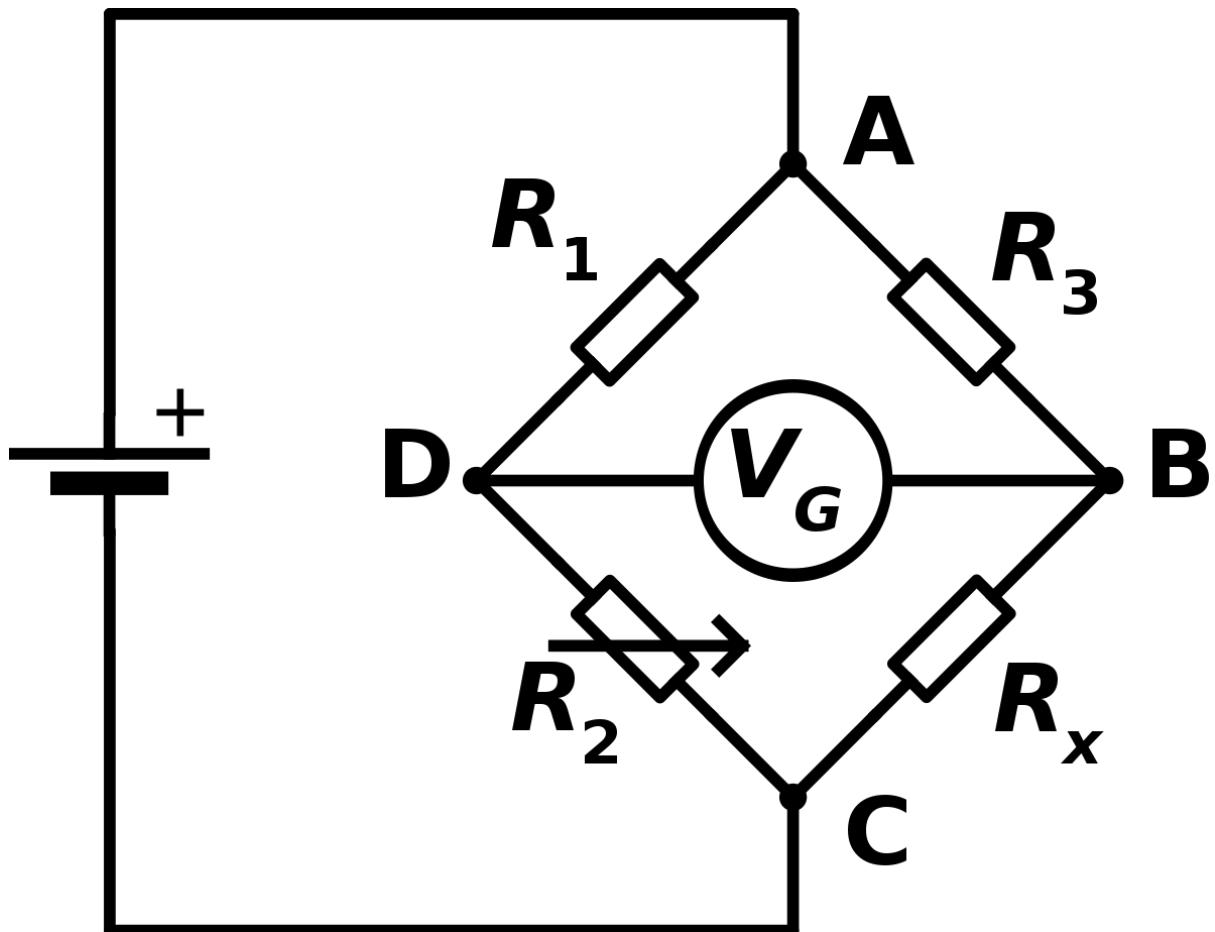
R_s : Resistência nominal do extensômetro

Porém, deve-se notar que os valores de deflexão esperados para um metal dentro de sua zona de deformação elástica são muito pequenos, o que acarreta em pequenas variações de resistência no extensômetro. Com o objetivo de facilitar a medição da deflexão, devem ser utilizados artifícios de instrumentação como um circuito de ponte com a finalidade de detectar com maior variação as mudanças de resistência do sensor.

2.2.1 Ponte de Wheatstone

Circuitos de ponte são utilizados para prover melhores medições e precisões em uma variedade de aplicações de medição de resistência elétrica, indutância e capacitação sob condições tanto estáticas quanto transientes. Dentre diversos tipos de circuitos de ponte a ponte de Wheatstone, demonstrada na Figura 7 se demonstra como um dos tipos de circuito elétrico mais utilizado e facilita a leitura da variação de resistência de sensores que apresentam baixas variações de resistência elétrica na sua operação. (HOLLMAN, 2011)

Figura 7 – Ponte de Wheatstone



A ponte de Wheatstone é normalmente utilizada em comparações e medições de resistência elétrica que variam de 1Ω até $1 M\Omega$ (HOLLMAN, 2011).

Utilizando as leis de kirchhoff para analisar a saída de tensão entre os pontos B e D se obtém a Equação (8).

$$V_{out} = V_{in} \frac{R_1 R_3 - R_2 R_4}{(R_1 + R_2)(R_3 + R_4)} \quad (8)$$

onde

E : Tensão de saída da ponte

V : Tensão de excitação da ponte

R_i : Resistência nominal dos resistores da ponte

Em uma aplicação onde o sensor de deformação representa uma resistência variável dentro do circuito e os outros resistores apresentam resistências iguais ao do valor nominal do sensor utilizado, pode-se combinar a equação prévia com a equação do fator de extensão para obter uma relação entre tensão obtida e valor de extensão apresentado no sensor, logo a equação de transferência do circuito é representada na Equação (9).

$$\frac{V_{out}}{V_{in}} = \frac{k}{4}(\varepsilon_1 - \varepsilon_2 + \varepsilon_3 - \varepsilon_4) \quad (9)$$

onde

V_{out} : Tensão de saída da ponte

V_{in} : Tensão de excitação da ponte

k : Gage factor

ε_i : Valor de deformação no extensômetro "i"

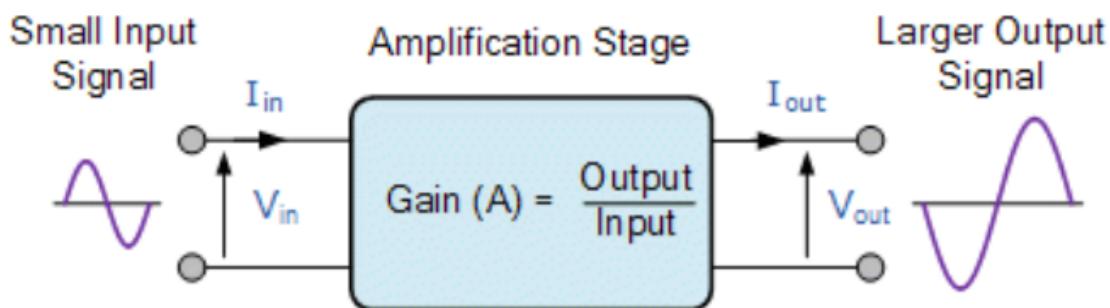
Circuitos de ponte se mostram de grande utilidade em experimentos práticos e são amplamente utilizados na medição da resistência de transdutores como extensômetros e outros tipos de sensores que convertem uma grandeza física em uma variação de resistência. Para medições estáticas, a tensão de saída do circuito de ponte é normalmente medido utilizando um voltímetro ou um dispositivo de coleta de dados de tensão (HOLLMAN, 2011).

Uma vez conhecido o fato de que não aconteceram grandes variações de tensão em um extensômetro na sua operação, devido ao fato do material apresentar pequenos valores de deformação dentro de sua zona elástica, pode-se concluir que o sinal de saída da ponte de Wheatstone não será de grande ordem de grandeza, logo, deve-se estudar métodos de amplificação dessa tensão com o objetivo de facilitar a obtenção das leituras por um voltímetro digital ou placa de controle.

2.3 OBTENÇÃO DE SINAIS

Medidas experimentais podem ocorrer de diversas formas e em vários casos os sinais são considerados fracos, logo eles devem ser amplificados com o objetivo de facilitar sua utilização por um dispositivo de saída. A maior parte dos amplificadores de sinal atuais utilizam circuitos integrados ou dispositivos de estado sólido para amplificar um sinal fraco analógico (HOLLMAN, 2011).

Figura 8 – Princípio de funcionamento de um amplificador de sinal



Fonte: https://www.electronics-tutorials.ws/amplifier/amp_1.html
acesso em fev 2022

O grau de amplificação de um amplificador pode ser dado pela Equação (10), que relaciona o sinal de entrada recebido pelo amp com o sinal de saída, que é lido pelo controlador.

$$\text{Gain}(A) = \frac{\text{input}}{\text{output}} \quad (10)$$

onde

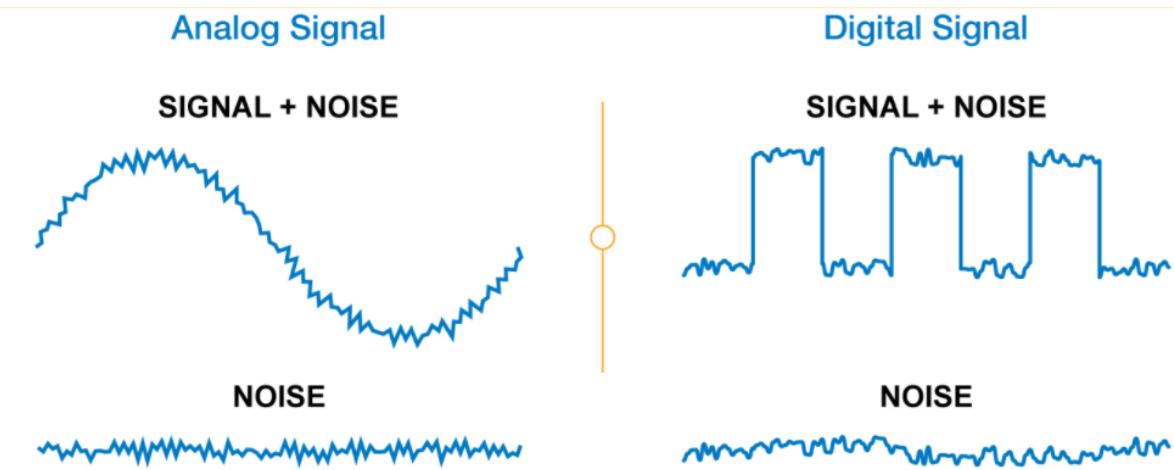
$\text{Gain}(A)$: Grau de amplificação do amplificador de sinal

input: Sinal de entrada

output: Sinal amplificado

Devido a efeitos aleatórios ou conhecidos ruídos característicos sempre estarão presentes em situações de tomada de medidas, esses ruídos podem ser filtrados utilizando circuitos que apenas permitem que uma certa parte das frequências que compõem o sinal obtido passem adiante no circuito a fim de modificar o sinal de saída do amplificador. Essa filtragem não resolve todos os problemas que podem ser encontrados, porém melhora significativamente o resultado de um experimento (HOLLMAN, 2011).

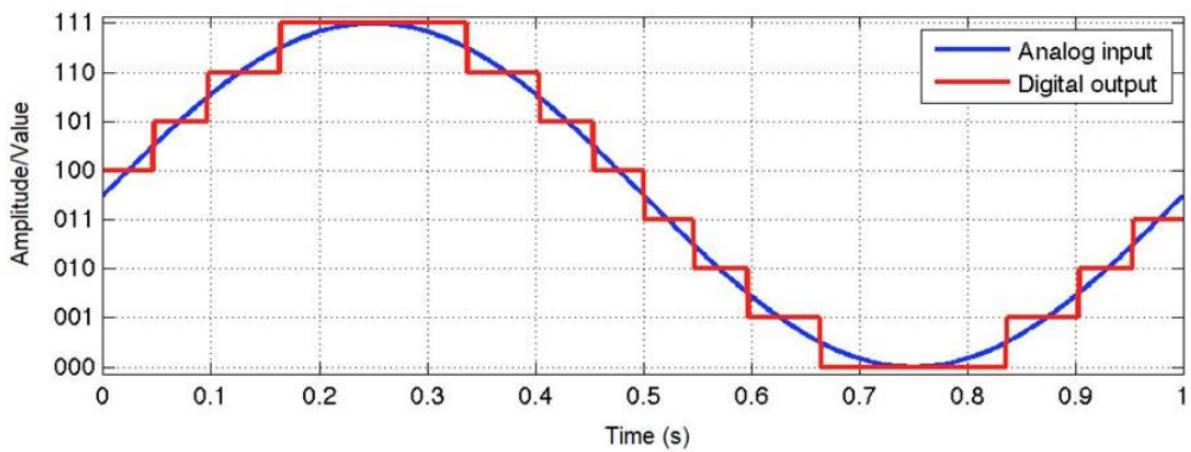
Figura 9 – Ilustração de ruidos presentes em sinais analógicos e digitais



Fonte: www.quora.com/How-can-digital-signals-possess-noise-immunity acesso em fev 2022

Uma vez que os sinais encontrados até aqui no sistema são de característica analógica e espera-se que a utilização e tratamento deles ocorra em um computador ou placa controladora como Arduino, que opera de maneira digital, deve-se então converter essas informações de tensão de um meio analógico para um meio digital, para isso é utilizado um conversor digital-analógico.

Figura 10 – Representação gráfica de um sinal analógico em forma digital



Fonte: www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters acesso em fev 2022

Em um meio analógico, as variáveis físicas são processadas como valores num meio contínuo, enquanto em um meio digital, valores são caracterizados por uma representação discreta. Uma faz razões para o tratamento de sinais de maneira digital é imunizar o sinal obtido do efeito de ruídos durante a transmissão, uma vez que definir

se um sinal obtido é de valor 1 ou 0 é muito mais fácil do que definir se é um valor entre os infinitos possíveis em um meio contínuo ([HOLLMAN, 2011](#)).

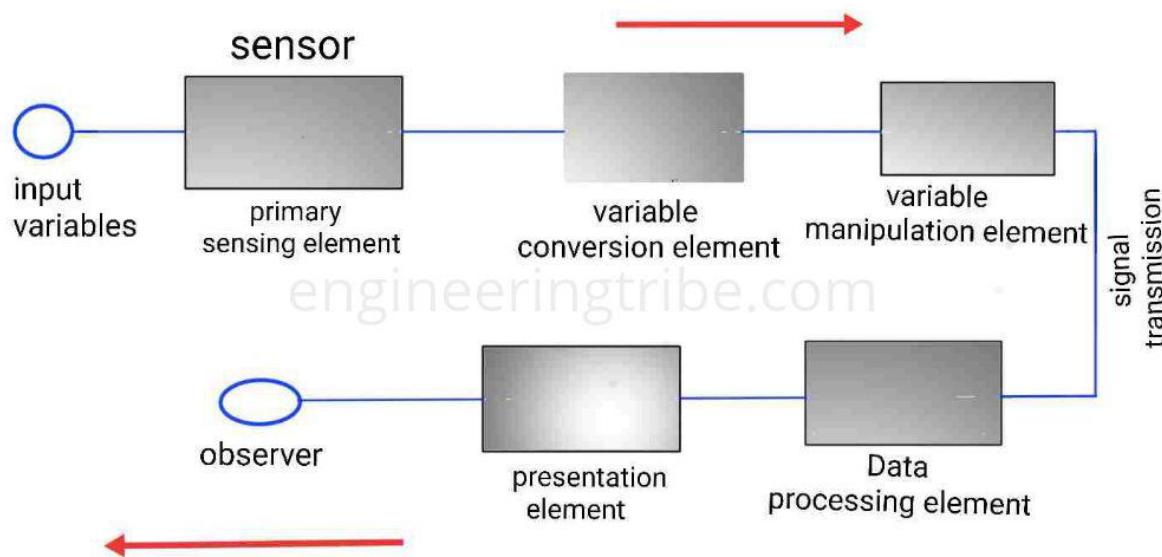
Com a finalidade de não ser perdidas informações no momento de conversão de um sinal do meio analógico para a forma digital, deve ser seguido o teorema sampling que estipula que a taxa de leitura de um sinal de maneira digital necessita ser pelo menos duas vezes o valor da frequência desse sinal no meio analógico (**HOLLMAN, 2011**).

A aquisição e processamento subsequente dos sinais obtidos pode ser feito de diversas formas, desde simples cálculos e obtenção manuais de dados até utilizando rotinas computacionais complexas. O objetivo do sistema de aquisição de dados é o de coletar, processar e/ou armazenar os dados obtidos em um experimento ou medição (HOLLMAN, 2011).

2.4 SISTEMAS DE MEDAÇÃO

A maior parte dos sistemas de medição podem ser divididos em três partes, um estágio de detecção da medida física, um estágio intermediário de amplificação, filtragem e conversão de sinal e um estágio final, que engloba a obtenção e processamento do sinal por um dispositivo de controle ([HOLLMAN, 2011](#)).

Figura 11 – Diagrama de blocos dos estágios de um sistema de medição



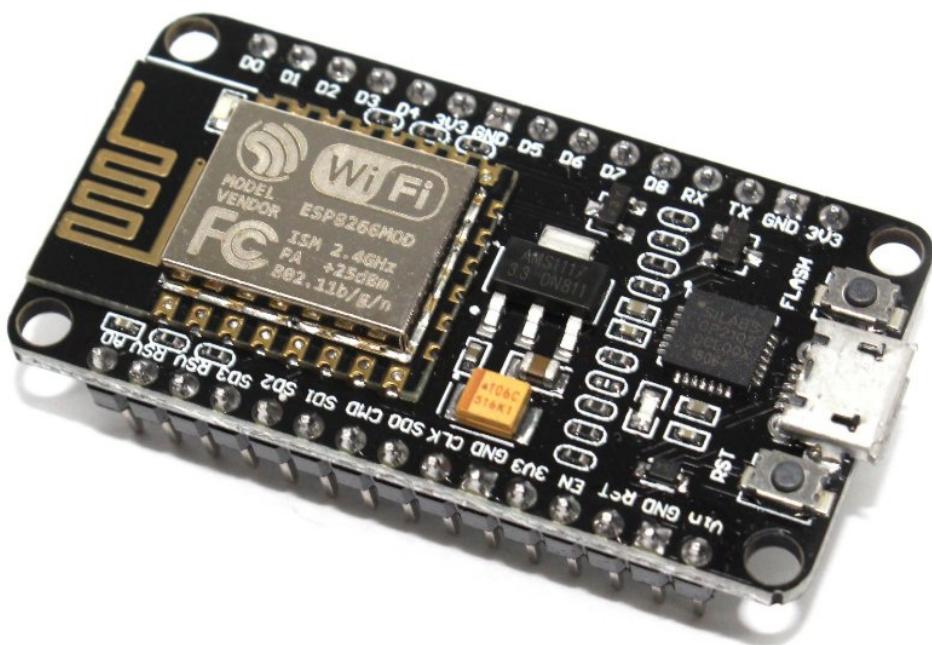
Fonte: <https://www.engineeringtribe.com/2020/09/generalized-measurement-system.html>
acesso em fev 2022

Uma medição é considerada estática quando a grandeza física analisada não apresenta mudanças no tempo como uma viga sobre uma carga constante de flexão.

caso essa viga é sujeita a um tipo de vibração, ou a um carregamento cíclico, então não pode mais se considerar o sinal estático ([HOLLMAN, 2011](#)).

Para desenvolver o sistema de controle e obtenção do sinal foi utilizado a plataforma de desenvolvimento ESP8266. Suas principais vantagens sobre a plataforma Arduino, de menor preço e mais amplamente utilizada, é devido ao fato de que o ESP8266 apresenta em sua construção um módulo de comunicação sem fio wifi integrado, o que reduz o tamanho do dispositivo e o torna mais barato em aplicações que necessitam de comunicação wireless ([ESPRESSIF, 2022](#)).

Figura 12 – Controlador ESP8266



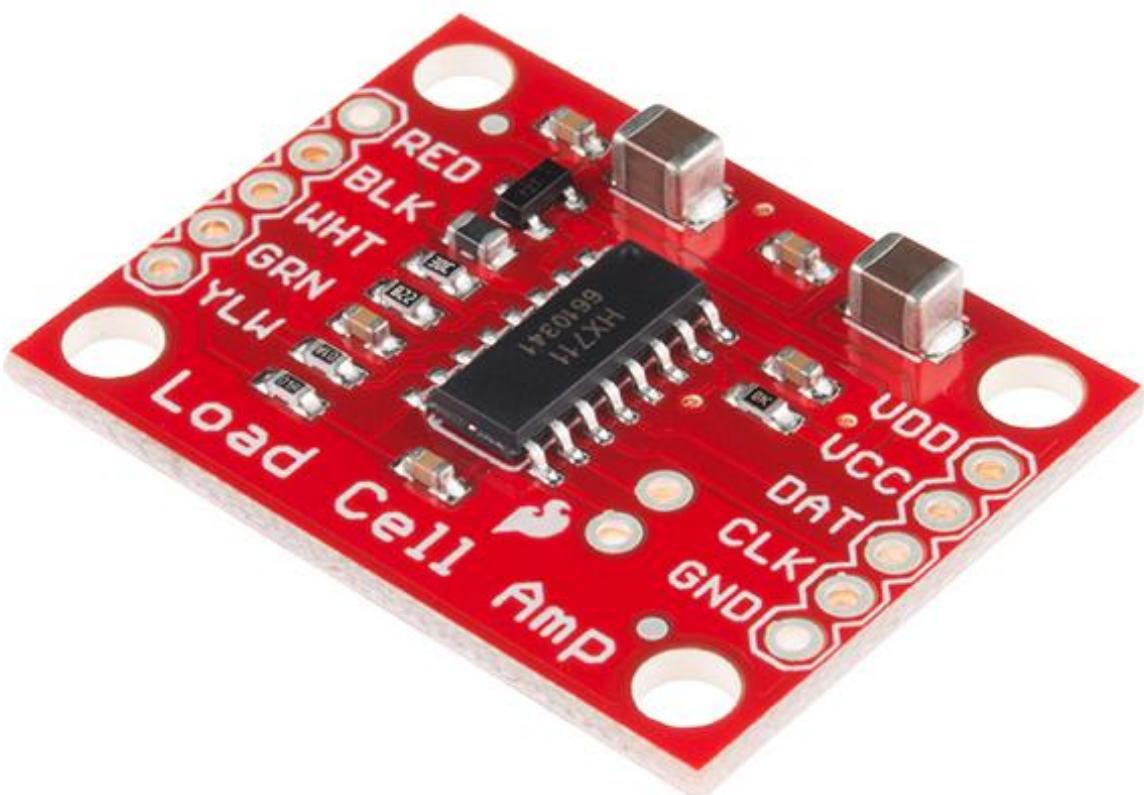
Fonte: [amazon.com](https://www.amazon.com)

A programação do controlador é feita utilizando uma linguagem de programação baseada na linguagem C + + adaptada para a utilização em placas de controle utilizando o ambiente de desenvolvimento Arduino IDE, que permite a utilização de extensões para programação e utilização de módulos externos, como o amplificador de sinal.

O HX711 é conversor analógico digital de precisão com resolução de 24 bits, desenvolvido para utilização para balanças e aplicações de controle industriais para se conectar diretamente com circuitos de ponte. O módulo é capaz de obter sinais na frequência de até 80 amostras por segundo e fator de ganho de 128. O amplificador de sinal que pode ser utilizado para obtenção de sinais com tensão tão baixas quanto na faixa de $\pm 20m$, o que permite pequenos sinais serem medidos com alta resolução

(AVIA, 2022).

Figura 13 – Conversor analógico digital HX711



Fonte: [amazon.com](https://www.amazon.com)

Os dados obtidos pelo sistema de medição são todos em formato digital em forma de vetores unidimensionais compostos pelos valores das amostras obtidas durante o tempo do experimento, esses valores são transmitidos em tempo real para um computador que executa um programa de obtenção de dados para realizar transformações mais complexas e análises dos sinais obtidos em tempo real.

2.5 ANÁLISE DOS SINAIS OBTIDOS

Algum tipo de análise deve sempre ser feita em todo tipo de conjuntos de dados experimentais. Várias considerações entram na determinação final da validade dos resultados experimentais, erros podem acarretar na invalidade dos dados mesmo quando estes foram obtidos com cautela. Alguns erros são de natureza aleatória, e outros podem ser por natureza física ou por descuido do experimentador, como flutuações eletrônicas, fricção ou desgaste dos componentes, esses tipos de erros devem ser descartados imediatamente (HOLLMAN, 2011).

Leituras individuais em um instrumento podem variar devido a erros de natureza aleatória, que seguem uma distribuição estatística normal, e o experimentador pode estar desejando obter o valor médio de diversas leituras realizadas (HOLLMAN, 2011). A Equação (11) obtém o valor médio para uma medição experimental consistente de diversas leituras.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^x a_i \quad (11)$$

Para cada leitura é esperado um valor de desvio lido, deve se notar que quanto melhor for o sistema de medição menor serão os valores de desvio obtidos no conjunto de leituras, o desvio padrão, representado pela Equação (12), se mostra como um bom indicador da situação dos desvios, e consequentemente da exatidão de um sistema de medição.

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{n}} \quad (12)$$

Essa equação de desvio padrão deve ser utilizada para grandes populações de amostras ou para quando os dados obtidos podem ser comparados com grandezas conhecidas (HOLLMAN, 2011). Para se obter a informação de se os valores experimentais estão de acordo com os desejados pode-se utilizar o teste do chi quadrado, representado na Equação (13).

$$\chi^2 = \sum \frac{(observed - expected)^2}{expected} \quad (13)$$

Esse teste é uma importante ferramenta de teste de qualquer resultado de distribuição experimental esperada. Se o valor de chi quadrado é igual a zero então a distribuição assumida é exatamente a distribuição real, quanto maior o valor de chi quadrado, menor é a correlação entre os dados medidos e os reais. (HOLLMAN, 2011)

2.5.1 Ambiente de desenvolvimento computacional Python

A linguagem computacional Python é utilizada tanto para desenvolvimento do software que realiza a comunicação entre o sistema de medição e o computador durante a utilização, quanto para a criação de rotinas para obtenção dos dados estatísticos para cada amostra obtida.

Python se caracteriza como uma linguagem de programação humanamente amigável, básica e de fácil leitura e entendimento. Que permite ao usuário a utilização de extensões e pacotes com funções de conveniência para resolver a maior parte dos problemas computacionais encontrados ([TIM HALL, 2010](#)).

A extensão numpy é um pacote fundamental para computação científica utilizando a linguagem de programação Python. O numpy é uma ferramenta utilizada para o processamento de dados em forma vetorial, uni ou multidimensional, seu funcionamento é baseado na conversão dos dados numéricos do formato de lista para um formato específico, altamente otimizado chamado ndarray. O pacote numpy também apresenta diversas funções matemáticas, lógicas, estatísticas, algébricas feitas para serem utilizadas com objetos ndarray, isso acarreta em uma minimização do tempo de processamento de um programa se comparado com utilizando funções nativas de python ([DEVELOPERS, P., 2022](#)). As principais funções utilizadas são demonstradas na [Tabela 1](#).

Tabela 1 – Funções do pacote NumPy utilizadas

Função	Descrição	Parâmetros de entrada	Parâmetros de saída
array	Cria um objeto do numpy	Lista python	Vetor ndarray
average	Obtem valor médio	Vetor ndarray	Valor médio
std	Obtém desvio padrão	Vetor ndarray	Valor desvio padrão
amax	Obtém valor máximo	Vetor ndarray	Valor máximo
amin	Obtém valor mínimo	Vetor ndarray	Valor mínimo

Fonte: O autor (2022)

Uma grande gama de outros pacotes em python usam como base a estrutura de dados e funções presentes, como o pandas, que é utilizado para facilitar a manipulação e armazenamento de dados em formato de tabular, como planilhas e bancos de dados ([DEVELOPERS, P., 2022](#)). Dados em formatos tabulares do pandas podem facilmente ser processados, analisados e armazenados utilizando funções do numpy e funções nativas do pandas.

Com o auxílio do processamento de dados tabulares e utilizando as funções estatísticas do pacote numpy pode-se facilmente obter os valores nominais e de erro

de cada medida tomada com o dispositivo de medição. Então deve-se partir para a fase da análise dos dados nominais obtidos.

2.5.2 Análise dos valores nominais

Uma vez obtidos todos os valores nominais para cada situação experimental analisada, devem ser criadas representações gráficas da distribuição dos resultados obtidos para isso é utilizado o pacote `scipy`, que é uma coleção de algoritmos matemáticos e funções de conveniência, desenvolvidos em cima do pacote `numpy`. Ele proporciona ao usuário funções e classes para manipulação e visualização de dados científicos. O pacote `scipy` se mostra como um forte competidor aos ambientes de desenvolvimento mais comumente utilizados, como o Matlab, Scilab e o Octave ([DEVELOPERS, S., 2022](#)).

A principal visualização a ser obtida no experimento é o gráfico de distribuição de cargas aplicados pelos valores de tensão da ponte de Wheatstone obtidos, então, utilizando o método de minimização dos quadrados, pode-se obter os parâmetros básicos que descrevem uma função matemática em que os erros entre valores observados do experimento e os estimados sejam o mínimo o possível.

O gráfico de regressão linear dos dados experimentais é obtido utilizando a função `scipy.linalg.regress` do pacote `scipy`.

2.6 METODOLOGIA DE DESENVOLVIMENTO DE PROJETO DE PRODUTO

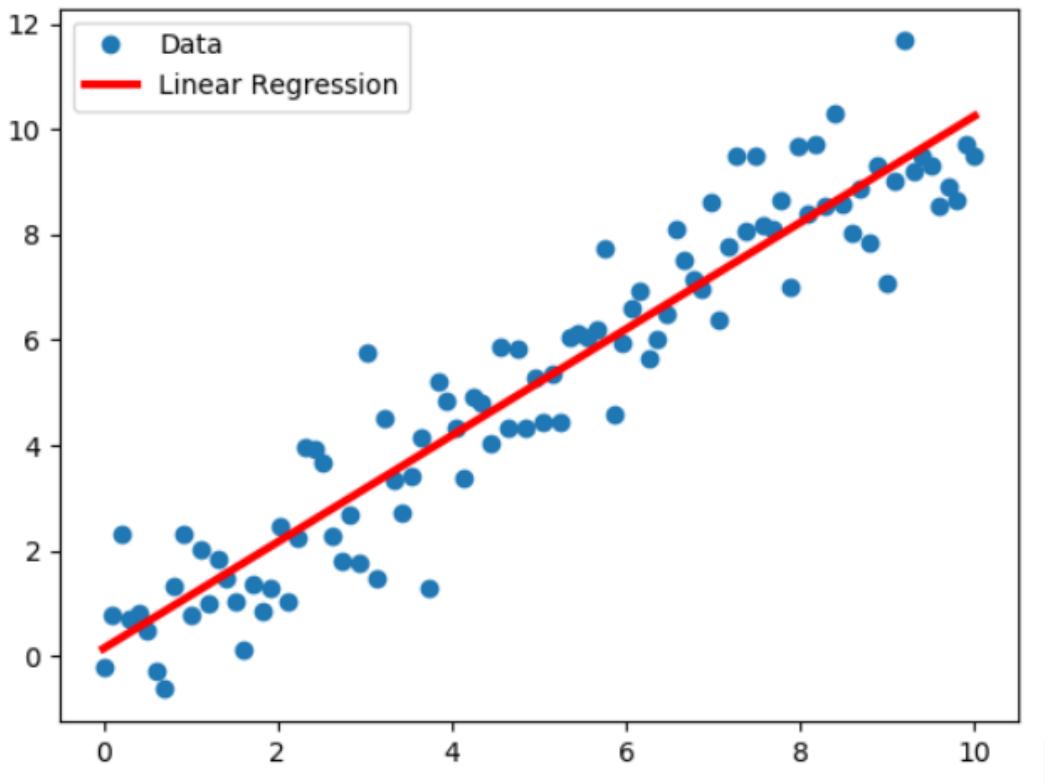
Desenvolvimento de produto entende-se como o processo de transformação de informações e conceitos até a produção e uso de um produto. Para se desenvolver um novo produto é necessário saber o que fazer, para quem fazer, quando fazer, com que fazer e como fazer. Esta organização é denominada metodologia de projeto ou metodologia de desenvolvimento de produtos. ([BACK et al., 2008](#))

O projeto de um produto engloba todas as etapas de definição das funções e características operacionais necessárias em um produto a ser desenvolvido, o modelo PRODIP divide o projeto em macro etapas, cada uma contemplando uma fase do desenvolvimento de um produto, uma visão geral das etapas dessa metodologia é mostrada na [Figura 5](#). ([NEDIP-UFSC, 2021](#)).

Novos produtos não precisam ser necessariamente produtos totalmente originais. Um produto novo pode ser obtido pela atualização, melhorias e/ou modificações de um produto existente, desta forma um produto existente pode ser reintroduzido a um novo nicho de mercado, e ele será considerado um novo produto. Para problemas de pequeno porte, pode ocorrer de que não exista a necessidade de se seguir um longo e rigoroso caminho para o desenvolvimento do projeto do produto. ([BACK et al., 2008](#))

Embásado no argumento indicado por Back em sua obra, a metodologia seguida segue as etapas apresentadas na ([NEDIP-UFSC, 2021](#)), porém, nem todas as ferra-

Figura 14 – Regressão linear



Fonte: www.researchgate.net/figure/Linear-Regression-model-sample-illustration_fig3_340271573 acesso em fev 2022

mentas e sub etapas apresentadas serão rigorosamente seguidas. O autor ignorou ou simplificou algumas etapas, principalmente nas fases informacional e conceitual do projeto, uma vez que o objetivo deste trabalho não é o de gerar um produto pronto para produção em massa.

2.6.1 Fase de planejamento

A fase de planejamento do projeto visa definir as etapas de desenvolvimento das ideias selecionadas utilizando definições de escopo, cronograma, orçamento riscos etc. Nessa etapa são definidas as ideias do problema e produto, então é feito um mapeamento tecnológico, em que, segundo a metodologia "são organizadas as informações do mercado, produto e tecnologia ao longo do tempo. Essas informações são correlacionadas e servem de base para estabelecer o plano de produtos."(NEDIP-UFSC, 2021).

O resultado da fase de planejamento do projeto é um documento que contém informações relacionadas ao escopo do projeto, como o problema a ser resolvido e as ideias base de resolução do problema. O escopo elaborado é o principal guia que

Figura 15 – Etapas da metodologia PRODIP



Fonte: ([NEDIP-UFSC, 2021](#))

direciona o desenvolvimento do produto e de suas funcionalidades. As próximas etapas apresentadas do projeto servem para solucionar metodologicamente o problema base definido no escopo.

2.6.2 Projeto informacional

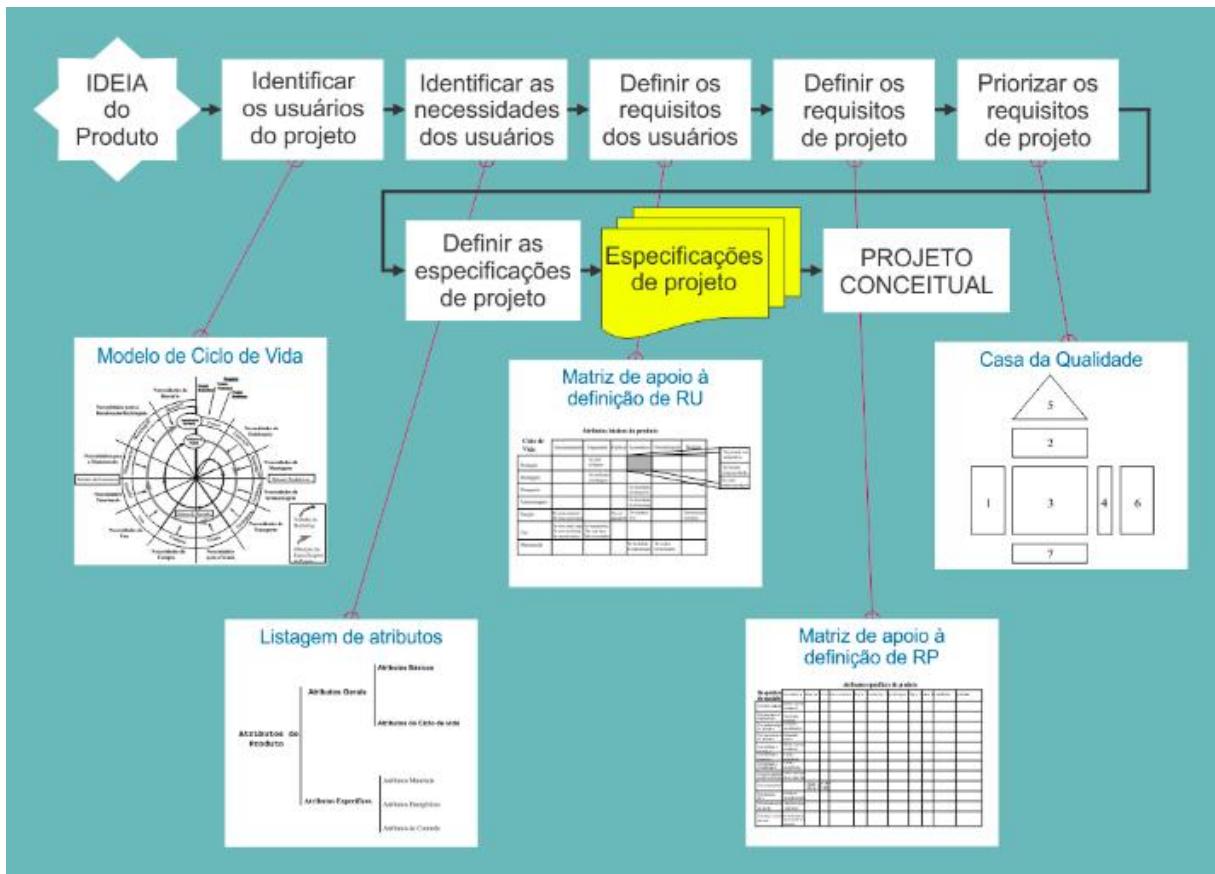
Nessa fase o objetivo é o estabelecimento das especificações de projeto, as quais irão orientar o desenvolvimento técnico do produto. O principal método empregado é a matriz da casa da qualidade da metodologia QFD (Quality Function Deployment). O projeto informacional utiliza ferramentas para definição de especificações de projeto que irão orientar o desenvolvimento do produto, o principal é a matriz QFD, utilizada para definir a importância dos requisitos do produto. ([NEDIP-UFSC, 2021](#))

No final do projeto informacional, é obtido de maneira clara e organizada quais são os requisitos do cliente em relação ao produto que está em desenvolvimento e quais são as funções o produto necessários que o produto necessita ter a fim de realizar os requisitos do cliente. Também se obtém, de forma quantizada, a ordem de prioridade nas quais o projeto necessita realizar os requisitos.

2.6.3 Projeto conceitual

Nesta etapa do projeto inicia-se o projeto e desenvolvimento das soluções conceituais para se atingir os requisitos obtidos na etapa anterior, o projeto conceitual é caracterizado pela fase criativa onde são geradas e avaliadas técnica e economicamente as alternativas para resolução do problema. Dentre as ferramentas utilizadas nessa fase se destacam matriz síntese de funções, matriz morfológica e matrizes multicritério de seleção. ([NEDIP-UFSC, 2021](#))

Figura 16 – Etapas do projeto informacional



Fonte: ([NEDIP-UFSC, 2021](#))

2.6.4 Projeto preliminar

No projeto preliminar é definido a forma final do produto, nessa fase são definidas características geométricas, de montagem, materiais para fabricação, características ergonômicas e de segurança e processos de manufatura do produto. Também pode ser realizados testes com protótipos para prova de conceito e otimização das características do produto desenvolvido até essa fase. ([BACK et al., 2008](#))

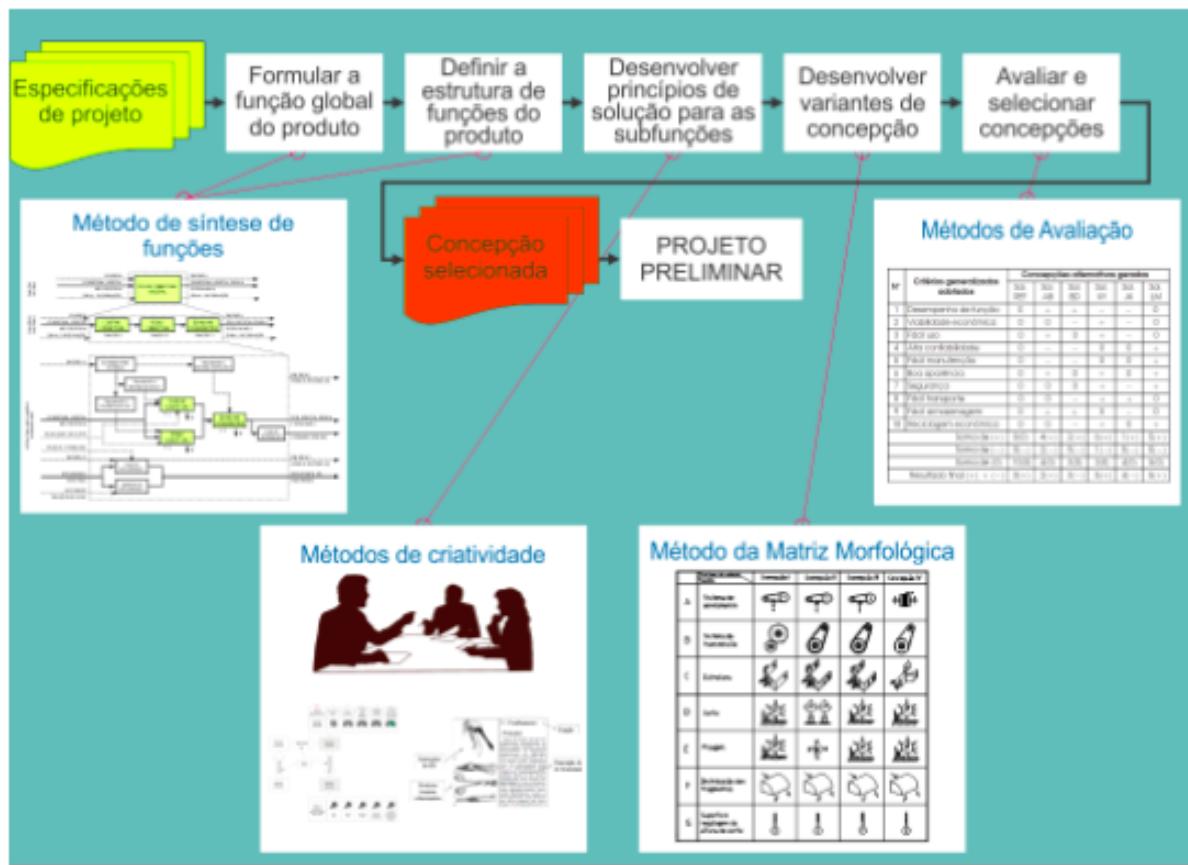
Os resultados dessa fase são as documentações de viabilidade econômica e requisitos de manufatura, e um protótipo funcional do produto.

2.6.5 Projeto detalhado

A elaboração do projeto detalhado se destina a vários propósitos, como a aprovação do protótipo, a finalização das especificações dos componentes e o detalhamento do plano de manufatura. ([BACK et al., 2008](#))

Nessa fase são gerados as documentações de especificação dos componentes, como desenhos técnicos, esquemas elétricos, planos de manufatura e softwares utili-

Figura 17 – Etapas do projeto conceitual



Fonte: (NEDIP-UFSC, 2021)

zados. Nesta fase são feitos testes em campo e laboratório e acontece a otimização do protótipo com objetivo de preparação para produção.

3 DESENVOLVIMENTO

O presente capítulo é dividido em duas fases, a primeira apresenta o desenvolvimento e os resultados das etapas da metodologia de desenvolvimento do protótipo do dispositivo. E a segunda seção apresenta a metodologia do experimento utilizado como prova de conceito do dispositivo desenvolvido.

3.1 PLANEJAMENTO DO PROJETO

O problema a ser solucionado é o de difícil acesso e altos custos para a obtenção de dados de deformação em tempo real. A primeira etapa para realizar o planejamento do projeto foi uma pesquisa de custo e disponibilidade de produtos iguais ou semelhantes a este propósito no mercado digital, os resultados da pesquisa são apresentados na próxima subseção.

3.1.1 Mapeamento tecnológico

Foram encontrados inúmeros dispositivos para telemetria e obtenção de sinais de sensores para aplicações industriais, de alta performance, ou para aplicações muito específicas, estes foram ignorados devido a altos custos ou baixa flexibilidade em relação ao dispositivo proposto. Dos produtos encontrados, três se mostraram mais semelhantes ao que será desenvolvido.

A empresa Isso disponibiliza em sua loja virtual Dois dispositivos para controle e obtenção de dados, denominados dmi tcr 44es, mostrado na [Figura 18](#), e dmi tcr 88es. Segundo o fabricante esses dispositivos são utilizados para aplicações de acionamentos e telemetria remota, e são indicados para uso em automações residenciais e industriais, a única diferença observada entre os dois modelos são o número de entradas e saídas de sinais que o dispositivo possui. ([ISSO, 2022a](#))

Outro dispositivo, denominado Bridge101A, mostrado na [Figura 19](#), fabricado pela empresa Madgetech foi encontrado, segundo a empresa é um data logger compacto que mede e armazena valores de tensões elétricas, e é normalmente utilizado com extensômetros, células de carga e outras sensores de baixa tensão, e que o dispositivo é utilizado para calcular com precisão parâmetros de tensão, torque, deformação e pressão ao longo do tempo. ([MADGETECH, 2022](#))

A [Tabela 2](#) mostra uma comparação entre os dados de utilização obtidos pela documentação dos dois dispositivos previamente apresentados.

Dentre os valores apresentados na tabela fica claro os altos preços envolvidos em qualquer aplicação que necessite a utilização desse tipo de dispositivo. Muitos deles necessitam ainda de programas para programação e utilização proprietários e que ainda tem custos adicionais para utilização. O dispositivo desenvolvido neste

Figura 18 – Datalogger DMI TCR 44es



Fonte: (ISSO, 2022a)

Tabela 2 – Comparação entre dispositivos encontrados

	Isso DMI TCR 44es	Isso DMI TCR 88es	Madgetech Bridge101A
Dimensões	124x117x55mm	190x117x55mm	36x64x16mm
Comunicação com PC	Conexão Ethernet	Conexão Ethernet	Conexão USB
Taxa de leitura	Não disponibilizado	Não disponibilizado	4 hz
Faixa de tensão de leitura	Não disponibilizado	Não disponibilizado	±30 mV
Faixa de preço	R\$1100,00	R\$1300,00	R\$2800,00

Fonte: O autor 2022

trabalho tem como objetivo um preço consideravelmente menor que os analisados e programável e utilizável utilizando tecnologias gratuitas e de código aberto.

Uma possível aplicação mais específica de um produto semelhante aos pesquisados seria um dispositivo que possibilite a telemetria direta de cargas de torque presentes em eixos de tração durante seu funcionamento, para isso seria necessário que o dispositivo fosse desenvolvido voltado para esse tipo de aplicação, a falta de um dispositivo desse tipo disponível no mercado levou ao autor elaborar uma pesquisa científica focado em encontrar dispositivos desenvolvidos semelhantes a ideia.

Figura 19 – Dispositivo Bridge101A



Fonte: ([MADGETECH, 2022](#))

3.1.2 Pesquisa científica

Com o intuito de facilitar o desenvolvimento do dispositivo, foi realizada uma revisão sistemática de trabalhos científicos e acadêmicos disponíveis nas bases de dados Web of Science, Springer, Sciedirect e Google Scholar, utilizando como palavras-chave “Dynamic, Torque, Shaft, Sensor, Strain, Gauge”, os principais obtidos são apresentados nessa subseção.

Um artigo desenvolvido por ([NIEDWOROK, 2014](#)) relata o desenvolvimento e aplicação de um sistema de sensoriamento de torque em tempo real em um eixo cardan de um carro de mina utilizando a medição da deformação utilizando extensômetro com

transferência dos dados via radiofrequência, o trabalho também indica que o posicionamento do sensor necessita estar em contato com a superfície de maior deformação do componente, o autor realiza uma análise por elementos finitos para encontrar esse local. O artigo também aponta que o sinal vindo do sensor deve ser ampliado utilizando uma ponte de Wheatstone para conseguir ter a instrumentação correta da grandeza. O artigo mostrou resultados satisfatórios e não discutiu sobre ruídos e imprecisões presentes nos dados obtidos.

(NURPRASETIO *et al.*, 2018) desenvolve, em seu trabalho, um sistema de medição para veículos terrestres, aplicado em uma bancada de testes que simula o estado de veículos terrestres em operação, o sistema utiliza um microprocessador Arduino nano de fácil acesso e baixo custo. Os dados são transmitidos via comunicação bluetooth. O artigo também ilustra o processo de calibração do dispositivo feito antes do teste dinâmico, assim como no trabalho anterior, também é enfatizada a necessidade das metodologias de instrumentação do sinal vindo do extensômetro. Seus resultados também se mostraram promissores, porém o autor indica que é necessário a remoção dos ruídos de medição, o que segundo ele será endereçado em um trabalho futuro.

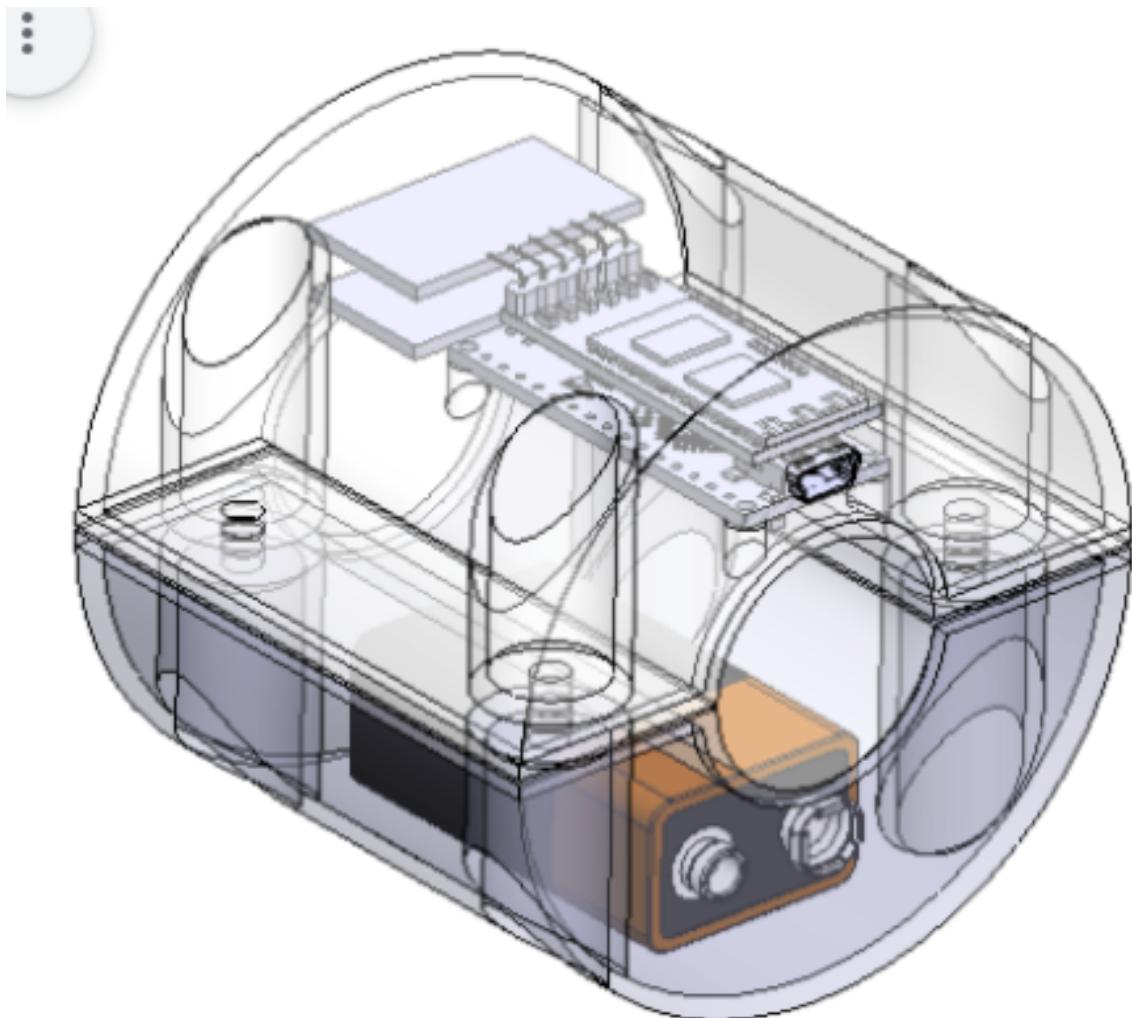
(GHARGHAN; NORDIN; ISMAIL, 2017) compara um sistema de medição similar ao dos dois trabalhos prévios com um sistema de medição de torque em tempo real de alto custo utilizado por ciclistas profissionais no pedivela. O artigo introduz a tecnologia de transmissão de dados Zigbee, que consegue transmitir dados a um baixo consumo energético. Após a obtenção dos dados, o autor utiliza as ferramentas de análise estatística de Bland-Altman e porcentagem de erro médio absoluto para a validação do sistema.

(SILVA; VARANIS; MERELES, 2017) compara os dados de um sistema semelhante aos anteriores com resultados de análises de modelo matemático analítico e análise por elementos finitos aplicados em bancadas de viga engastada com carga na ponta e de torque aplicado em um eixo com um dos lados travados, diferente dos trabalhos anteriores, este possui uma seção com o desenvolvimento das equações dos modelos utilizados, e assim como os artigos anteriores foram encontrados resultados satisfatórios.

A ideia inicial da concepção do produto seria o do desenvolvimento de um dispositivo que obtivesse os valores de torque em um eixo em tempo real. Para que fosse possível solucionar tal problema, o dispositivo teria que obter os dados de tensão dos pólos de uma ponte de wheatstone, com um extensômetro montado a um eixo sob torque e transmitir os dados obtidos via conexão sem fio em tempo real a um computador. O projeto inicial do dispositivo é mostrado na [Figura 20](#).

O conceito é composto por uma placa de desenvolvimento Arduino, que controla um amplificador de sinal que por si obtém os dados de tensão de uma ponte de wheatstone. Os dados obtidos pela ponte de wheatstone então são transmitidos utilizando um módulo bluetooth HC-05. O custo total desses dois componentes é baixo, e um protóti-

Figura 20 – Projeto inicial do dispositivo



Fonte: O autor 2022

po de dispositivo pode ser montado por aproximadamente R\$150,00, desconsiderando os componentes de encapsulamento. O projeto informacional é iniciado considerando este conceito como a ideia inicial do produto.

3.2 PROJETO INFORMACIONAL

Uma vez que a ideia inicial do produto é caracterizada pelo grande potencial de ser de baixo custo e de ser consideravelmente compacto em relação aos outros produtos encontrados no mercado, pode-se traçar uma estratégia de definição de público alvo, o que facilita as tomadas de decisão durante o desenvolvimento do produto. Dentro os possíveis públicos que são beneficiados pelas características do produto que será desenvolvido, as equipes de competição Baja SAE foram escolhidas como público alvo para guiar o desenvolvimento do dispositivo, uma vez que o baixo custo, baixas

dimensões e flexibilidade de aplicação são características desejadas para utilização nesses grupos.

Foi elaborado um formulário eletrônico que apresenta a ideia do conceito de funcionamento de um dispositivo para sensoriar dados de torque em tempo real. No formulário são apresentadas duas perguntas iniciais, que avaliam a importância da obtenção desse tipo de dado e a importância de serem obtidas em tempo real, além das duas perguntas iniciais, são apresentados os seguintes tópicos sobre a necessidade do produto em desenvolvimento de:

- a) Qual sua opinião sobre a importância da obtenção de dados de torque, potência e rotação do sistema de propulsão durante a utilização?;
- b) Qual sua opinião sobre o recebimento a distância e ilustração desses dados em tempo real?;
- c) Capacidade de gravação/armazenamento de dados;
- d) Ser de baixo preço;
- e) Ser compacto;
- f) Ser leve;
- g) Suportar grande variação da faixa de torque;
- h) Ser a prova de água, fluidos, poeira, lama, etc;
- i) Ser resistente a impactos;
- j) Ser de fácil utilização e montagem;
- k) Ser de fácil manutenção;
- l) Possuir bateria de longa duração (duração da prova mais longa);
- m) Possuir disjuntor/botão de liga/desliga;

Todas as perguntas e tópicos foram respondidos pela seleção de uma nota de 1 a 5 para cada questão, onde o valor 1 significa que o requisito listado é de baixa importância e o valor 5 é de extrema importância. As respostas do formulário são avaliadas pela análise do valor médio obtido pelos valores respondidos. Foram obtidas as respostas de 7 membros de diferentes equipes que participam da competição. Os resultados são mostrados na [Tabela 3](#).

Com as respostas do formulário de pesquisa do público alvo, pode ser elaborada uma matriz de listagem de atributos, que serve para mapear quais funcionalidades produto deve ter para atender cada um dos requisitos do produto, e quais sistemas necessários para garantir as funcionalidades listadas, a matriz é mostrada na [Figura 21](#).

A partir da análise da matriz de listagem de atributos e da pesquisa com o público alvo é feito um levantamento das principais características do produto final:

- a) Preço final

- b) Capacidade de processamento
- c) Taxa de obtenção de dados
- d) Sistema de armazenamento
- e) Software com interface gráfica
- f) Comunicação sem fio
- g) Capacidade da bateria
- h) Diâmetro da carcaça
- i) Comprimento da carcaça
- j) Sistema de vedação
- k) Sistema de resistência a impactos
- l) Massa total

Com os dados de avaliação quantitativa dos requisitos do produto obtidos pela pesquisa com o público alvo e os dados das características do produto é elaborada uma matriz de avaliação de qualidade, ou QFD. Essa matriz serve para correlacionar os requisitos com as características desejadas do produto, com a finalidade de apontar quais as qualidades devem ser priorizadas na etapa de desenvolvimento do projeto.

Tabela 3 – Resultado pesquisa de mercado

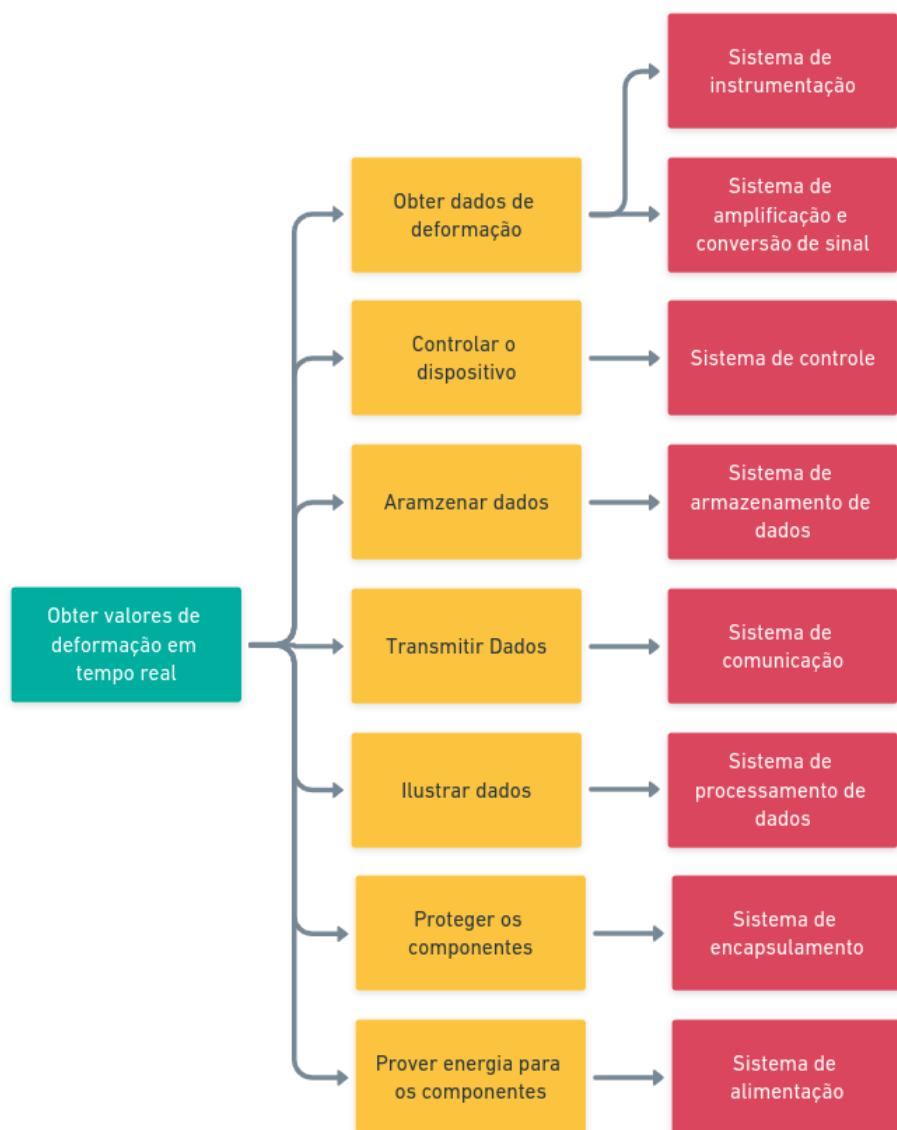
Pergunta/Requisito	Importância
Importância de obtenção dos dados	4.875
Importancia de obtenção em tempo real	3.875
Ser de baixo preço	3.625
Ser compacto	3.875
Ser leve	3.75
Suportar grande variação da faixa de torque	4.125
Ser a prova de água, fluidos, poeira, lama, etc	4.75
Ser resistente a impactos	4
Ser de fácil utilização e montagem	3.375
Ser de fácil manutenção	3.75
Possuir bateria de longa duração	4.25
Possuir botão de liga/desliga	3.375
Capacidade de gravação de dados	4.125

Fonte: O autor 2022

Os símbolos preenchidos na matriz qfd representam os fatores de correlação, que numéricamente são iguais a 9 para correlações fortes, 3 para correlações médias e 1 para correlações fracas.

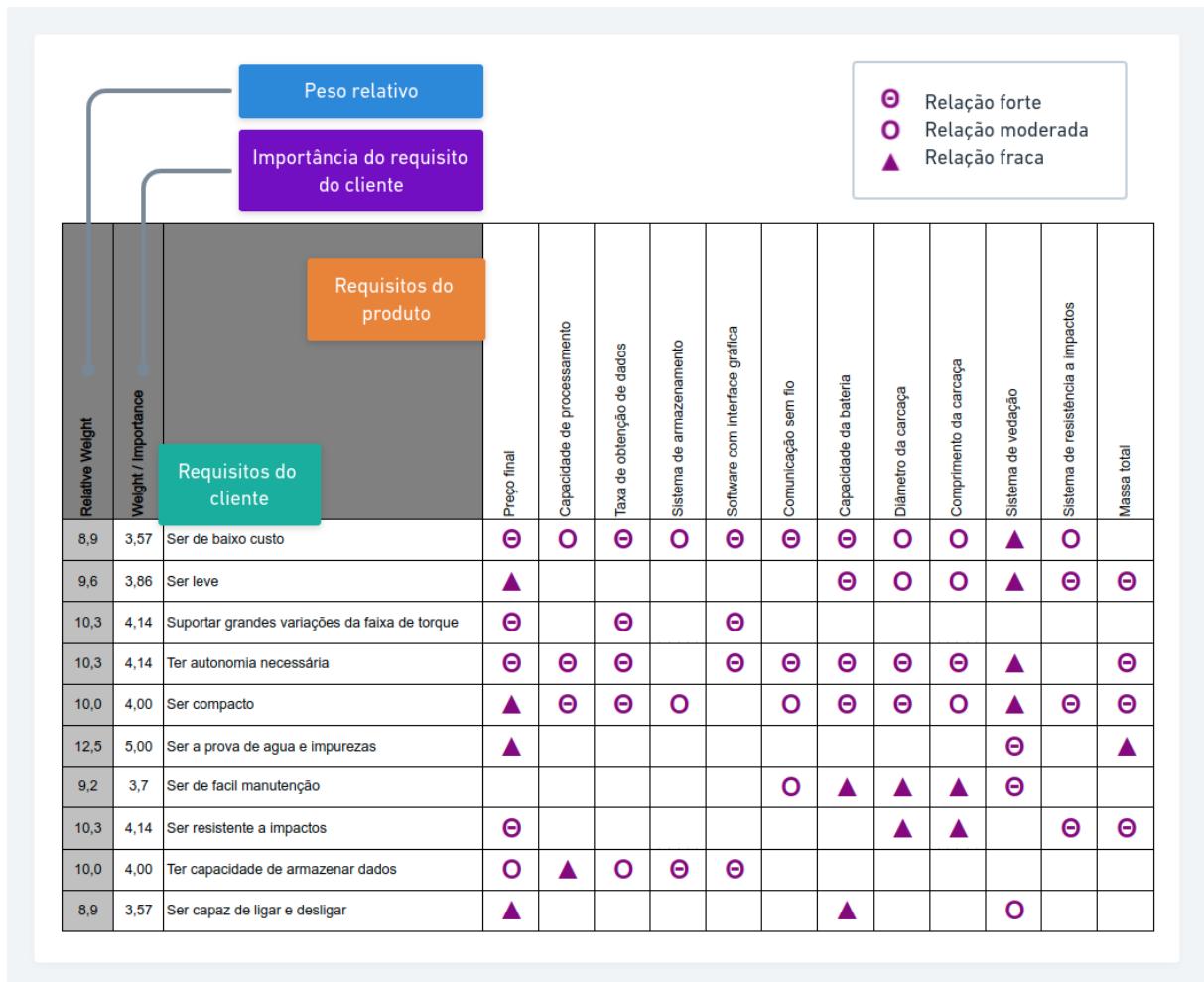
Os resultados de peso de importância para cada requisito de qualidade é calculado pelo somatório das multiplicações entre fator de correlação e o peso relativo relativos dos requisitos do cliente. Os resultados de priorização obtidos pela matriz QFD são apresentados na [Tabela 4](#).

Figura 21 – Matriz de listagem de atributos



Fonte: O autor 2022

Figura 22 – Matriz QFD



Fonte: O autor 2022

Tabela 4 – Prioridade dos requisitos definidos pela matriz QFD

Característica	Peso de importância	Prioridade
Preço Final	429,5	alta
Capacidade de processamento	219,2	baixa
Taxa de obtenção de dados	385,4	média
Sistema de armazenamento	146,3	baixa
Software com interface gráfica	355,5	média
Comunicação sem fio	230,6	baixa
Capacidade da bateria	367,3	média
Diâmetro da carcaça	257,7	baixa
Comprimento da carcaça	197,9	baixa
Sistema de vedação	260,8	baixa
Sistema de resistência a impactos	295,8	baixa
Massa total	374,4	média

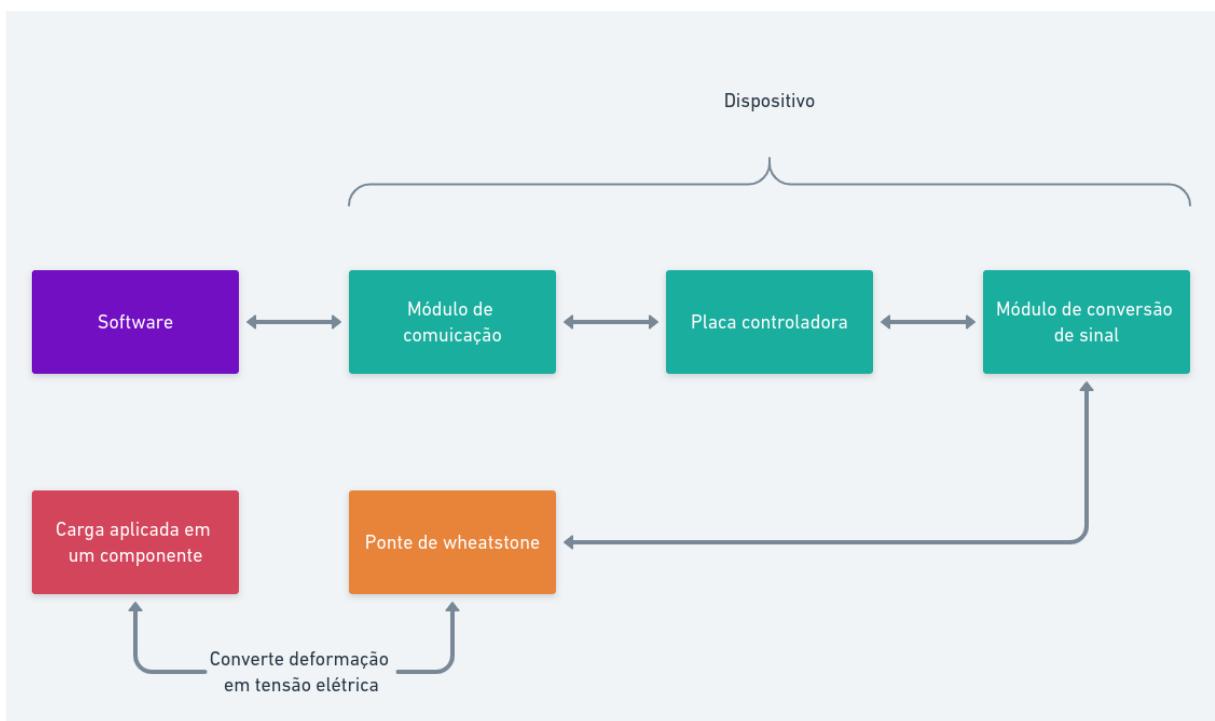
Fonte: O autor 2022

A priorização dos requisitos é utilizado para guiar a tomada de decisões de projeto e escolha de componentes nos estágios do projeto conceitual.

3.3 PROJETO CONCEITUAL

Na fase de projeto conceitual são avaliadas tecnicamente e economicamente as alternativas levantadas para se desenvolver um dispositivo que atenda os requisitos de cliente e de projeto. O conceito de funcionamento básico do dispositivo é ilustrado na [Figura 23](#).

Figura 23 – Conceito de funcionamento inicial



Fonte: O autor 2022

As opções de placa controladora avaliadas foram o Arduino Nano, o ESP32 ou o ESP8266, todos podem ser programados utilizando a linguagem de programação Arduino, e podem ser encontrados por preços menores que 60 reais, as principais diferenças entre os controladores são de que os módulos ESP possuem opções de comunicação sem fio integradas.

Caso seja necessária a utilização de comunicação sem fio em um módulo que não possua integrado as principais opções disponíveis são o módulo bluetooth HC-05, que tem o custo aproximado de 50 reais e o módulo wifi ESP8266, que tem o custo aproximado de 25 reais. A comunicação bluetooth funciona com a transferência de dados diretamente entre o módulo e o receptor. E a comunicação wifi pode ser feita

tanto pela conexão do dispositivo receptor diretamente ao módulo bluetooth quanto pela comunicação entre servidor e cliente.

As opções de módulo de amplificação de sinal avaliadas foram o HX711, que se apresenta como um módulo altamente utilizado em aplicações de células de carga porém apresenta uma taxa de leitura baixa, de até 80 amostras por segundo, e o ADS1115, que possui diversas opções de fator de amplificação que podem ser programadas e uma taxa de amostragem de 860 amostras por segundo. Ambos os módulos de amplificação já possuem conversores ADC integrados.

Apenas uma solução foi considerada como adequada para armazenamento de dados locais pelo dispositivo, que é a utilização de um módulo de cartão Micro SD, devido às baixas dimensões. Também foram avaliadas opções de alimentação energética para o dispositivo, materiais para construção do encapsulamento externo. Uma breve apresentação do levantamento de componentes é mostrado na [Tabela 5](#).

Tabela 5 – Levantamento de componentes

Tipo de componente	Características principais	Preço aproximado
Controladores		
Arduino nano	compacto e fácil de programar	R\$60.00
ESP 32	comunicação sem fio integrada	R\$40.00
Amplificadores de sinal		
HX711	Desenvolvido para aplicações de célula de carga	R\$20.00
ADS1115	Alta taxa de leitura	R\$70.00
LM358	Alta taxa de ganho	R\$10.00
Comunicação sem fio		
Bluetooth HC-05	Difícil programação, emite e recebe dados	R\$50.00
Módulo wifi	Facil acesso a qualquer dispositivo na mesma rede	R\$25.00
Módulo radiofrequênciia	Comunicação apenas do transmissor para o receptor	R\$15.00
Armazenamento		
Módulo cartão micro SD + cartão SD	Permite armazenamento de dados	R\$30.00
Alimentação		
4x pilha AAA	mais compacto, alta disponibilidade	R\$30.00
1x Bateria 9V	alta tensão e duração	R\$40.00
2x Bateria li-ion	grandes dimensões, recarregável	R\$35.00
Encapsulamento		
Carcaça ABS	Pode fabricado por impressão 3D	–
Carcaça multi material	Maior custo com fabricação, maior resistência	–

Fonte: O autor 2022

Após feito o levantamento de componentes pode-se montar a matriz morfológica, que serve para relacionar cada função elementar principal originada da matriz de funções do produto com os componentes levantados anteriormente. A matriz morfológica é mostrada na [Figura 24](#).

Figura 24 – Matriz morfológica

Função Elementar	Princípios de solução			Critérios de seleção
Possuir sistema de amplificação de sinal	Módulo amplificador de sinal LM358	Módulo amplificador de sinal HX711	Modulo ADS1115	custo, dimensões, praticidade, desempenho
Possuir sistema de processamento de dados	Microprocessador Arduino nano	Microprocessador ESP32	Microprocessador ESP8266	custo, dimensões, praticidade, desempenho, gasto energético
Possuir sistema de armazenamento de dados	Modulo de armazenamento em cartão SD			custo, dimensões, praticidade
Possuir sistema de comunicação	Transferência de dados via bluetooth (IEEE 802.15.1)	Transferencia via Wi-fi (IEEE 802.11)	Tranferencia via radiofrequência	custo, desempenho, praticidade, gasto energético
Possuir sistema de plotagem de dados	Software base python	Rotina Matlab	Software base c++	custo, praticidade
Possuir sistema de encapsulamento	Carcaça de ABS	Carcaça multi material		custo, desempenho, facilidade de fabricação
Possuir sistema de alimentação	Bateria 9V	4 pilhas AAA 1.5V	Bateria de ion de lítio	capacidade, peso, dimensões, praticidade

Fonte: O autor 2022

Na matriz morfológica também foi adicionado uma coluna para listar os principais requisitos de avaliação de cada produto, esses requisitos mostrados estão diretamente relacionados com os requisitos de cliente e produto obtidos da análise da matriz da casa da qualidade. Para a avaliação e comparação das soluções em relação aos requisitos levantados é utilizado a matriz de avaliação, mostrada na [Figura 25](#) a matriz de avaliação lista as possíveis tecnologias utilizadas para a solução e os requisitos, e para cada relação uma nota é definida entre -3, que significa que a tecnologia é a mais adequada ao requisito relacionado, até 3, que significa que a tecnologia é a menos adequada ao requisito relacionado.

Figura 25 – Matriz de avaliação

	Custo	Peso	Dimensões	Praticidade	Desempenho	Gasto energético	Facilidade de fabricação	Nota final
Módulo amplificador de sinal LM358	-3	0	3	3	3	0	0	6
Módulo amplificador de sinal HX711	0	0	0	-3	-3	0	0	-6
Modulo ADS1115	0	0	0	-3	-1	0	0	-4
Microprocessador Arduino nano	1	0	-1	0	1	-1	1	1
Microprocessador ESP32	3	0	0	-3	-3	3	-1	-1
Microprocessador ESP8266	1	0	0	-3	-1	1	-1	-3
Modulo de armazenamento em cartão SD	0	0	0	0	0	0	0	0
Transferência de dados via bluetooth (IEEE 802.15.1)	3	0	3	1	0	1	0	8
Transferencia via Wi-fi (IEEE 802.11)	1	0	3	-3	0	1	0	2
Tranferencia via radiofrequênciia	0	0	1	3	-3	3	0	4
Software base python	-3	0	0	-3	0	0	0	-6
Rotina Matlab	3	0	0	3	0	0	0	6
Software base c++	-3	0	0	-1	0	0	0	-4
Carcaça de ABS	-3	-3	0	0	0	0	-1	-7
Carcaça multi material	3	1	0	0	0	0	3	7
Bateria 9V	1	1	3	0	0	0	3	8
4 pilhas AAA 1.5V	0	0	1	-1	0	0	1	1
Bateria de ion de lítio	3	3	3	1	0	0	3	13

Fonte: O autor 2022

A matriz de avaliação resulta em um valor quantitativo que avalia o quanto adequado cada opção tecnológica é ao projeto do dispositivo. Após a análise dos resultados da matriz de avaliação as tecnologias escolhidas são mostradas na [Figura 26](#).

Figura 26 – Soluções escolhidas

SISTEMA	SOLUÇÃO		
AMP DE SINAL	LM358	HX711	ADS1115
PROCESSADOR	Arduino nano	ESP32	ESP8266
ARMAZENAMENTO	cartão SD		
COMUNICAÇÃO	Bluetooth	Wifi	Radiofrequência
SOFTWARE	Python	Matlab	C++
ENCAPSULAMENTO	Carcaça de ABS	Carcaça multi material	
ALIMENTAÇÃO	1 x 9V	4 x AAA 1.5V	li-ion

Fonte: O autor 2022

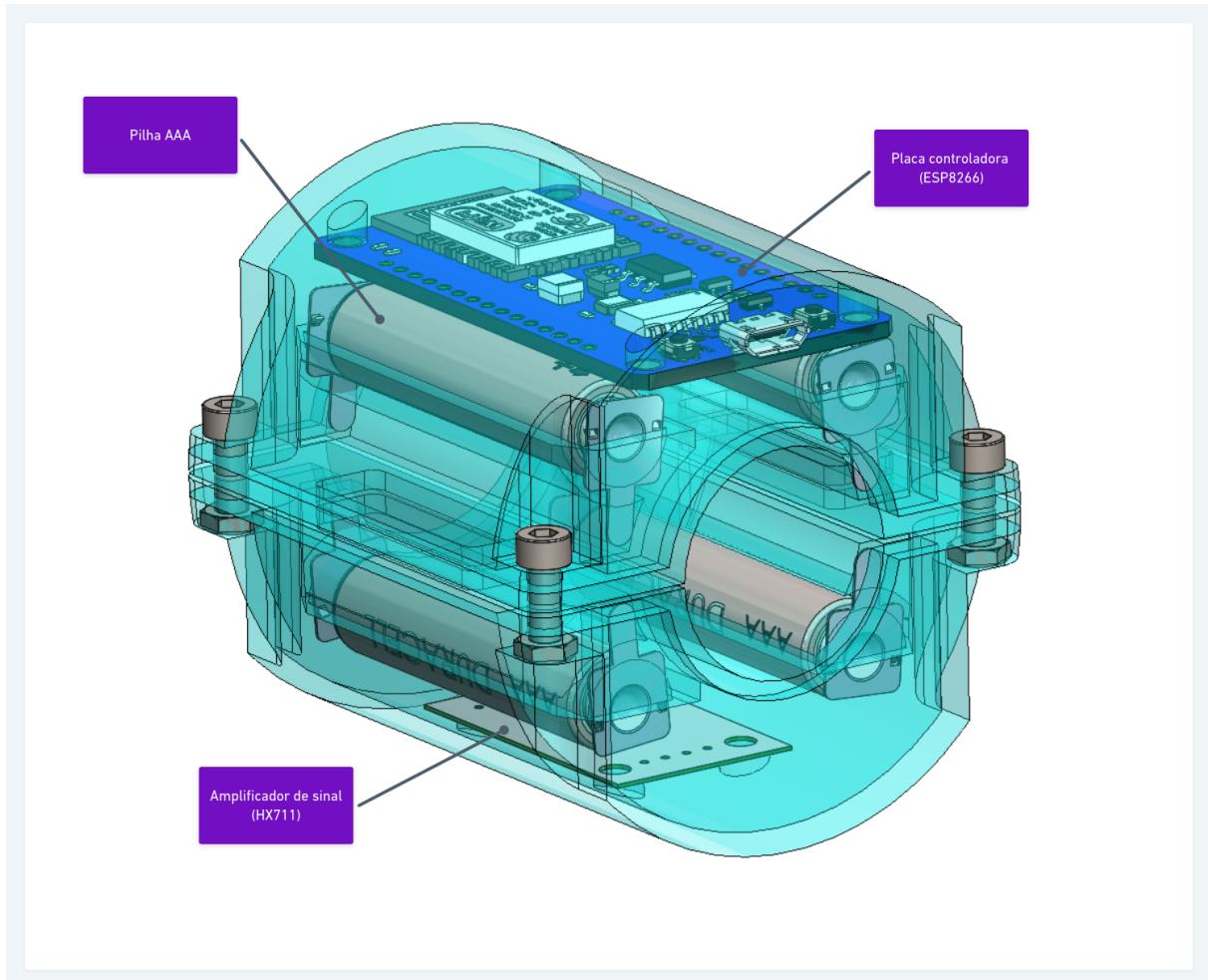
Após definido as tecnologias que serão utilizadas no desenvolvimento do produto, deve-se seguir para a etapa de projeto detalhado.

3.4 PROJETO DETALHADO

A documentação do projeto detalhado do produto é composta pelos projetos mecânico, elétrico/eletônico e pelo desenvolvimento do software de comunicação.

O projeto mecânico do dispositivo foi desenvolvido utilizando o software Solidworks 2017, os modelos da placa de controle ESP8266 e do amplificador de sinal HX711 foram obtidos na plataforma de compartilhamento de modelos GrabCAD. Após ser definido um arranjo inicial da posição dos componentes no dispositivo foi modelado um componente de encapsulamento para o dispositivo. O modelo da montagem do dispositivo é ilustrado na [Figura 27](#).

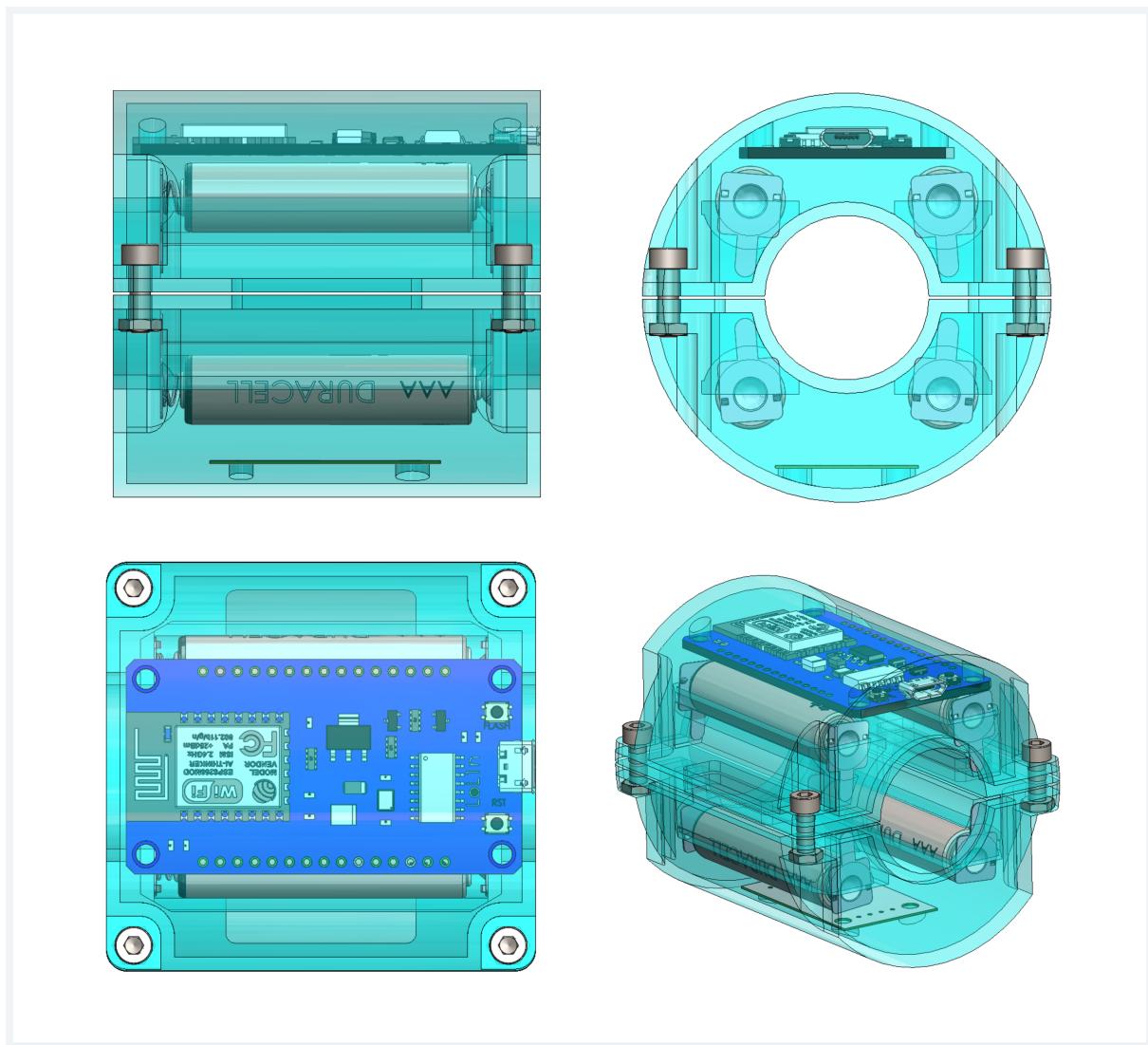
Figura 27 – Modelagem da montagem do dispositivo



Fonte: O autor 2022

A ilustração do projeto do dispositivo nas três vistas padrão e vista isométrica é mostrado na [Figura 28](#).

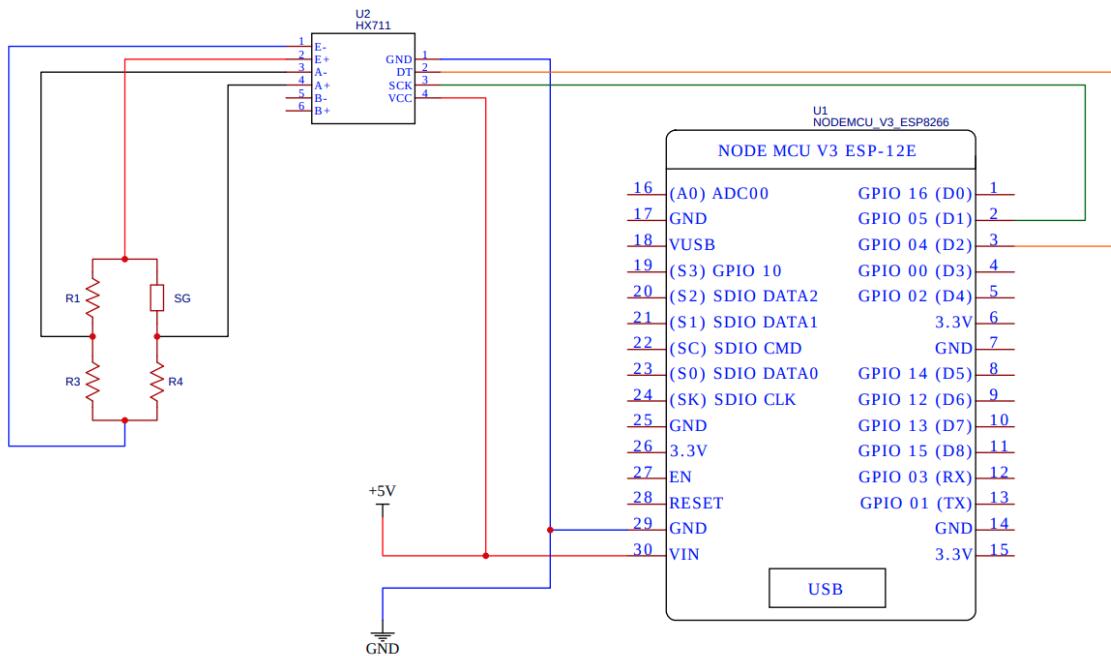
Figura 28 – Desenho em vistas padrões (primeiro diedro)



Fonte: O autor 2022

O projeto elétrico do dispositivo foi desenvolvido utilizando o software para desenvolvimento de placas eletrônicas EasyEDA e é mostrado na [Figura 29](#).

Figura 29 – Projeto elétrico do dispositivo



Fonte: O autor 2022

A fase de desenvolvimento do software engloba duas etapas principais, a primeira é o desenvolvimento da programação da placa de controle, que é feita utilizando o programa Arduino IDE para conectar e programar o controlador da plataforma. Uma introdução a estrutura do programa de controle e das suas funções é mostrada na Figura 30

Figura 30 – Esquema da programação do dispositivo



Fonte: O autor 2022

A programação da placa de controle é feita com uma linguagem de programação baseada em C++, com funções específicas para a programação de controladores de plataformas abertas, como o Arduino e o ESP8266.

A função handleRead utiliza o método hx711.read(), conforme o [Código 1](#), para obter os valores de tensão na saída da ponte de wheatstone e os enviar ao computador via rede local. O programa completo do controlador completo é apresentado no [Apêndice A](#).

Código 1 – Função de obtenção de sinal pelo amplificador HX711

```

1
2 void handleRead() {
3     if (server.method() != HTTP_POST) {
4         server.send(405, "text/plain", "Method now allowed");
5     } else {

```

```

6     server.send(200, "text/plain", String(hx711.read()) +
7         server.arg("plain"));
8 }
```

Uma vez configurado o controlador do dispositivo, foi desenvolvido utilizando a linguagem python um objeto computacional para facilitar a conexão e utilização dos dados originados pelo dispositivo. O objeto e suas funções é ilustrado pela [Figura 31](#).

Figura 31 – Objeto NetworkDevice



Fonte: O autor 2022

As funções de base do objeto são mostradas na [Código 2](#). O método `__init__` - recebe e salva como parâmetros do objeto os dados de endereço de ip do dispositivo e intervalo de leitura inseridos, e o método `read_line` utiliza a função `post` do pacote `requests` para enviar uma requisição ao endereço de ip do objeto e retorna o valor obtido em forma de texto de maneira calibrada, caso o argumento `calibrated` seja igual a verdadeiro, ou não calibrado, caso contrário.

Código 2 – Métodos base do objeto NetworkDevice

```
1
2 """
3 Creates a local network connection device object
4 """
5
6 def __init__(self, ip, delay_time):
7     self.url = f'http://[{ip}]/read'
8     self.delay_time = delay_time
9     self.offset = 0
10    self.a = 1
11    self.b = 0
12
13 def read_line(self, calibrated: bool = False):
14     """returns the last line of data sent by the device"""
15     read = int(post(self.url).text) - self.offset
16     if calibrated:
17         return self.a * read + self.b
18     else:
19         return read
```

Caso seja necessária a obtenção de dados em sequência ao longo do tempo foram desenvolvidas as funções `read_samples` e `plot`, mostradas na [Código 3](#), o número de amostras é passado para a obtenção repetida dos dados, utilizando o parâmetro de intervalo de leitura para determinar o intervalo de execução da chamada do método `read_line`, e é retornado um objeto de lista do numpy com os dados obtidos.

Código 3 – Métodos para executar experimentos do objeto `NetworkDevice`

```
1
2 def read_samples(self, n_of_samples: int, plot: bool = False,
3                  calibrated: bool = False):
4     """returns a number of data lines given a sample size"""
5
6     if not plot:
7         value_list = array([])
8         while len(value_list) < n_of_samples:
9             value_list = append(
10                 value_list,
11                 int(self.read_line(calibrated=calibrated)))
12             sleep(self.delay_time)
13             # print(f"sample: {len(value_list)}, value:
14             # {int(value_list[-1])}")
```

```
14
15     else:
16         value_list = self.plot_data(n_of_samples=n_of_samples,
17                                     calibrated=calibrated)
18
19
20 def plot_data(self, n_of_samples: int, calibrated: bool = False):
21     """plot the numerical data received"""
22     value_list = array([])
23
24     plt.ion()
25     fig, axs = plt.subplots(1)
26     fig.suptitle('Readings')
27
28     while len(value_list) < n_of_samples:
29         try:
30             value_list = append(value_list,
31                                 int(self.read_line(calibrated=calibrated)))
32
33             axs.cla()
34             axs.plot(value_list[-25:])
35
36             plt.pause(self.delay_time)
37
38         except:
39             print("error!")
40
41
42     plt.close()
43
44     return value_list
```

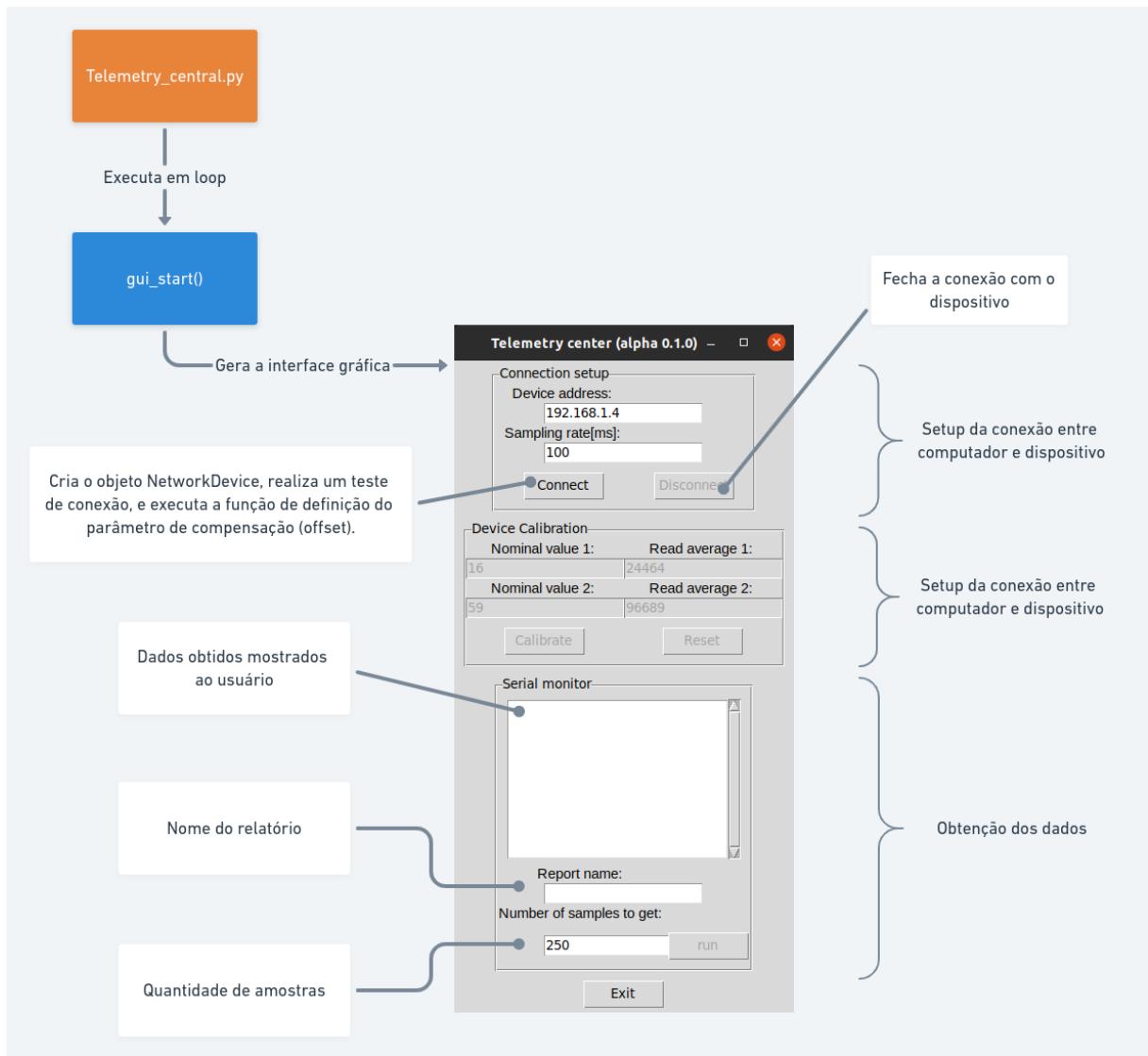
As funções de calibração do objeto de conexão com o dispositivo são mostradas no [Código 4](#). O método `calibrate_offset` serve para definir o ponto de leitura quando nenhuma carga está presente, e funciona obtendo 40 amostras utilizando o método `read_samples`, arredondando o valor médio dessas amostras a um número inteiro e definindo o parâmetro `offset` do objeto ao valor obtido. A função `calibrate_readings` utiliza duas cargas nominais conhecidas e seus sinais obtidos em forma de dicionário e utiliza estes valores para definir os parâmetros `a` e `b` da função linear de calibração, com a finalidade de ser utilizada no método `read_line` para converter os valores em forma de bits para um formato de valor nominal da grandeza física calibrada.

Código 4 – Métodos de calibração do objeto NetworkDevice

```
1
2 def calibrate_offset(self):
3     """sets the offset value for the device data gathering"""
4     self.offset = 0
5     self.offset = round(self.read_samples(n_of_samples=40).mean())
6
7 def calibrate_readings(self, calibration_dict_1: dict,
8                         calibration_dict_2: dict):
9     """calibrate the device readings"""
10    w1 = calibration_dict_1['nominal_value']
11    w2 = calibration_dict_2['nominal_value']
12    r1 = round(calibration_dict_1['signal'].mean())
13    r2 = round(calibration_dict_1['signal'].mean())
14
15    self.b = w2 / ((r2 / r1) * (w1 - 1) + 1)
16    self.a = (w1 - self.b) / r1
```

Com o intuito de facilitar a utilização do método pelo usuário é desenvolvido uma interface gráfica simples, uma breve apresentação da interface desenvolvida utilizando o pacote tkinter é mostrada na [Figura 32](#).

Figura 32 – Interface gráfica



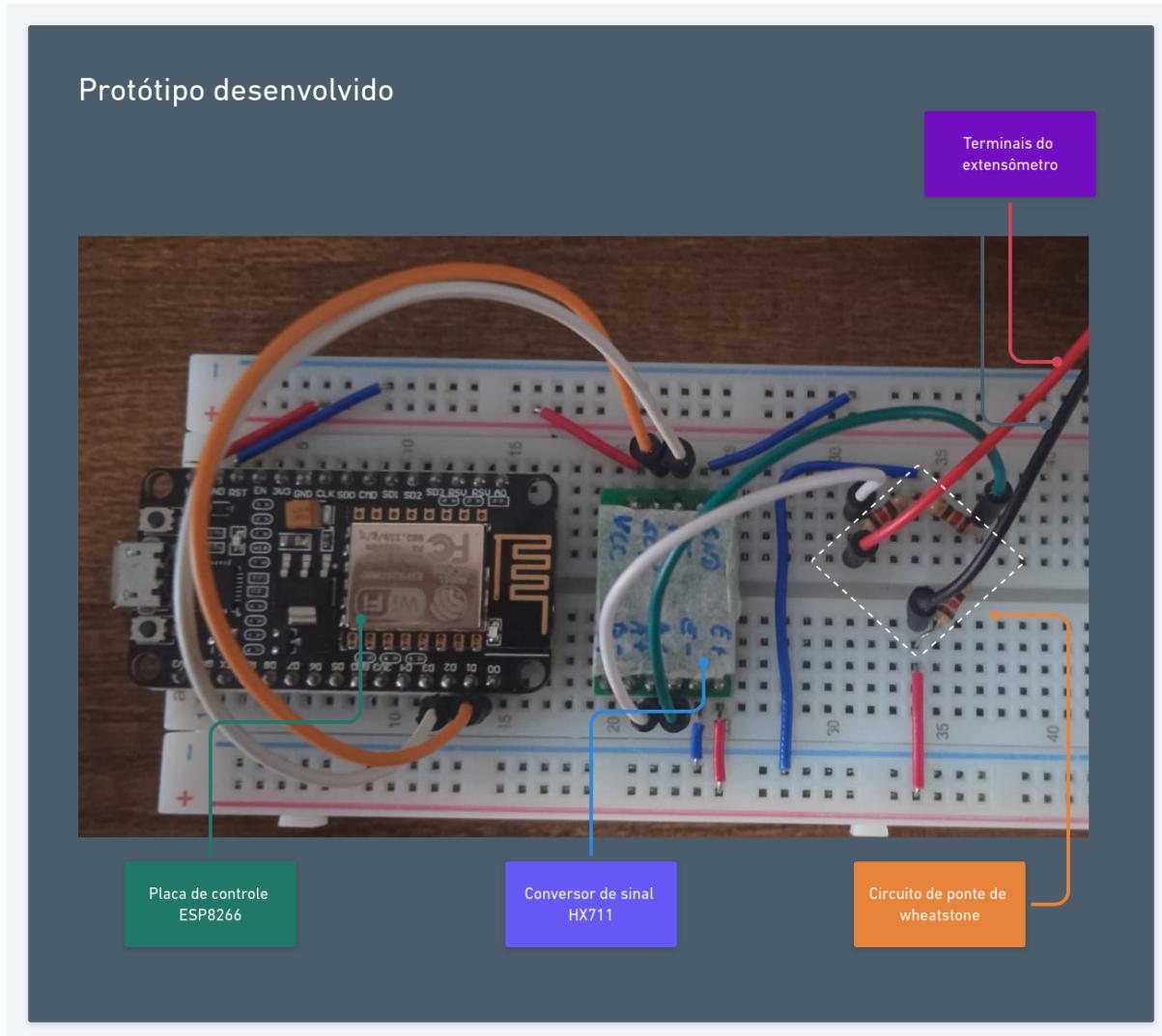
Fonte: O autor 2022

Os códigos completos para o objeto NetworkDevice e para a geração da interface gráfica são apresentados no [Apêndice B](#).

3.5 VALIDAÇÃO DO PROTÓTIPO DO DISPOSITIVO

Após a conclusão da fase de projeto detalhado foi montado o protótipo do dispositivo, com a finalidade ser executar experimentos de validação. O dispositivo montado é mostrado na [Figura 33](#)

Figura 33 – Protótipo preparado para os experimentos



Fonte: O autor 2022

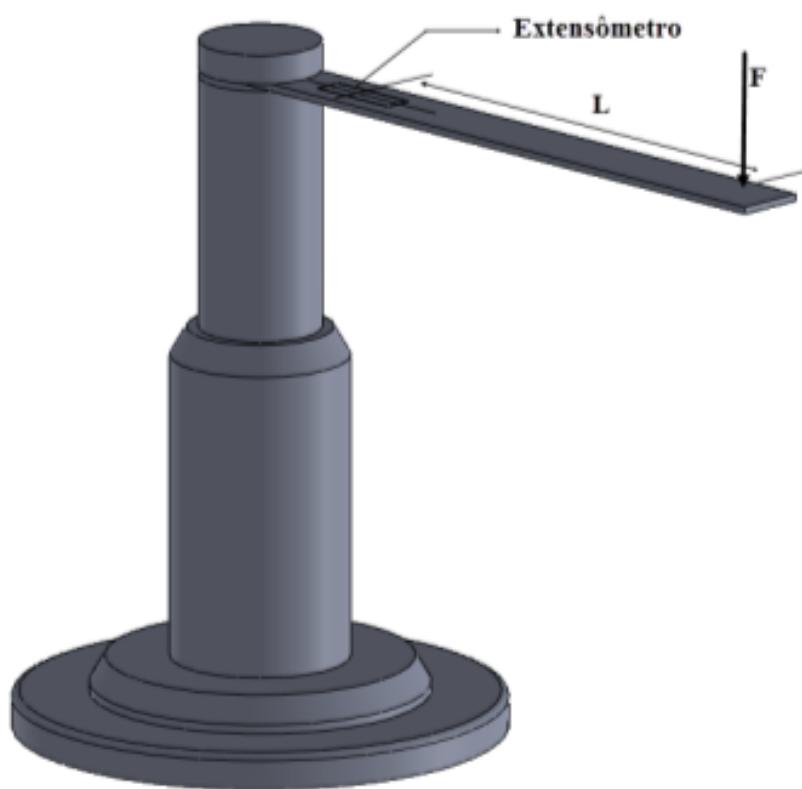
A metodologia de execução do experimento para validar o funcionamento do protótipo desenvolvido segue a mesma metodologia do experimento realizado no trabalho de conclusão de curso de ([MINELA, 2017](#)), uma introdução aos principais tópicos dessa metodologia e eventuais diferenças entre os dispositivos utilizados nos dois trabalhos para cada tópico é feita nessa seção.

Os ensaios realizados envolvem a leitura de extensômetros colados em corpos de prova sobre ação de cargas específicas. Um dispositivo de análise de cargas de flexão e um para cargas em torção foram desenvolvidos e produzidos por ([MINELA, 2017](#)), os mesmos dispositivos foram utilizados para este trabalho.

3.5.1 Dispositivo para ensaio de flexão

O dispositivo para realizar o ensaio de flexão é composto por uma viga de seção retangular de 20mm de largura por 2mm de espessura, que mede 200mm de comprimento e é de uma liga desconhecida de alumínio, e uma base projetada e fabricada em aço 1020 para fixar a viga em uma situação de engaste na viga. A imagem mostra o projeto do dispositivo desenvolvido por ([MINELA, 2017](#)).

Figura 34 – Dispositivo de ensaio de flexão



Fonte: ([MINELA, 2017](#))

Na viga é colado um extensômetro unidimensional para obter os dados de deformação, as propriedades do extensômetro utilizado são mostradas na [Tabela 6](#).

As cargas são aplicadas na extremidade livre da viga a distância de 150mm entre o centro do extensômetro e o ponto de aplicação da carga. Com a finalidade de se obter os valores de deformação causados pela aplicação de cada carga, cada trabalho utiliza-se de um sistema de medição diferente, apresentado na seção seguinte.

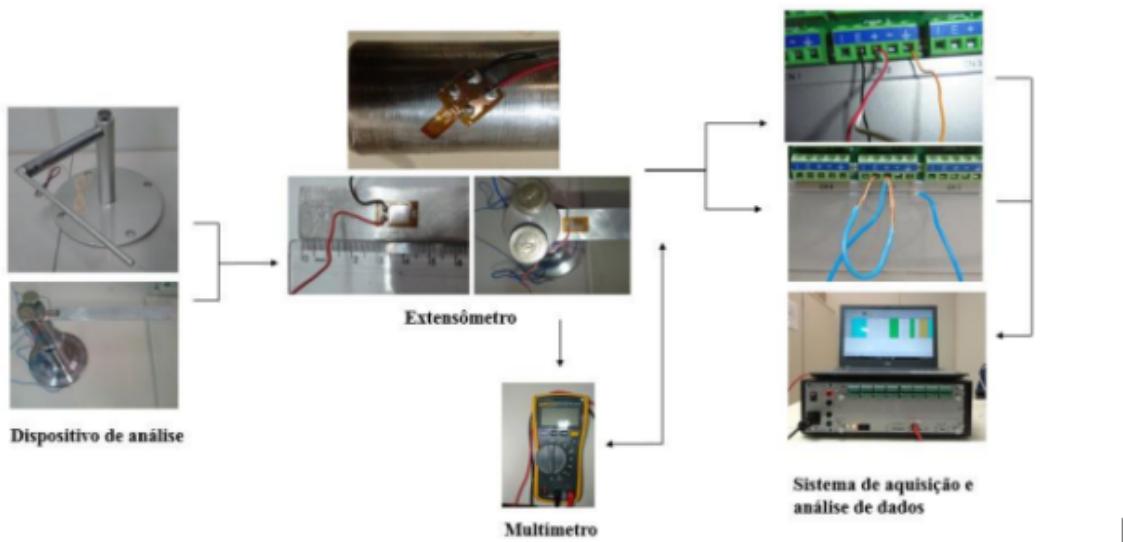
Tabela 6 – Propriedades do extensômetro colado ao dispositivo de flexão

Marca	Micro Measurements ®
Tipo de extensômetro	EA-06-250AF-120
Resistência Elétrica	$120 \pm 0.15\% \Omega$
Factor de gage até $75^{\circ}F$	$2.025 \pm 0.5\%$
Comprimento	6.35
Limite de temperatura	-75°C á 175°C para medições estáticas

Fonte: Adapdato de ([MINELA, 2017](#))

3.5.2 Sistemas de medição

Tanto o sistema de medição utilizado em ([MINELA, 2017](#)) quanto o utilizado para o desenvolvimento deste trabalho seguem o mesmo princípio de funcionamento.

Figura 35 – Sistema de medição utilizado por ([MINELA, 2017](#))Fonte: ([MINELA, 2017](#))

O sistema de medição utilizado por ([MINELA, 2017](#)) é composto por um dispositivo de obtenção de dados ADS2002, desenvolvido e fabricado pela LYNX Tecnologia, que é conectado a um computador. O ADS2002 obtém os dados gerados pelo sensor e os envia ao computador via conexão ethernet. O computador utiliza o software AQ-Dados para conexão com o ADS2002, calibração e aferição dos extensômetros, e o software AqAnalysis para fazer o processamento dos sinais e gerar relatórios de análise, ambos os programas são desenvolvidos pela fabricante do dispositivo de obtenção de dados.

A conexão entre o dispositivo de obtenção de dados e o extensômetro ocorre

Figura 36 – Sistema de medição utilizando o LINX ADS2002



Fonte: (MINELA, 2017)

utilizando os terminais elétricos do canal específico de análise, o ADS2002 já possui, integrado em si, os componentes elétricos do circuito da ponte de wheatstone para a instrumentação do sinal. A calibração do extensômetro afim dos resultados serem mostrados como valores deformação, ao invés de tensões, é feita utilizando a [Equação \(14\)](#) que define um valor de fator de engenharia em função do fator gage FG da resistência média do extensômetro RM , resistência de calibração RC , que é disponibilizada pelo fabricante.

$$VE = \left(\frac{1}{FG} \left(\frac{RM}{RM+RC} \right) \right) \quad (14)$$

O sistema de medição utilizando o protótipo desenvolvido pelo autor é feito pela conexão entre o controlador ESP 32 e o computador via o software desenvolvido em python para realizar a transferência dos dados obtidos entre o controlador e o computador em tempo real.

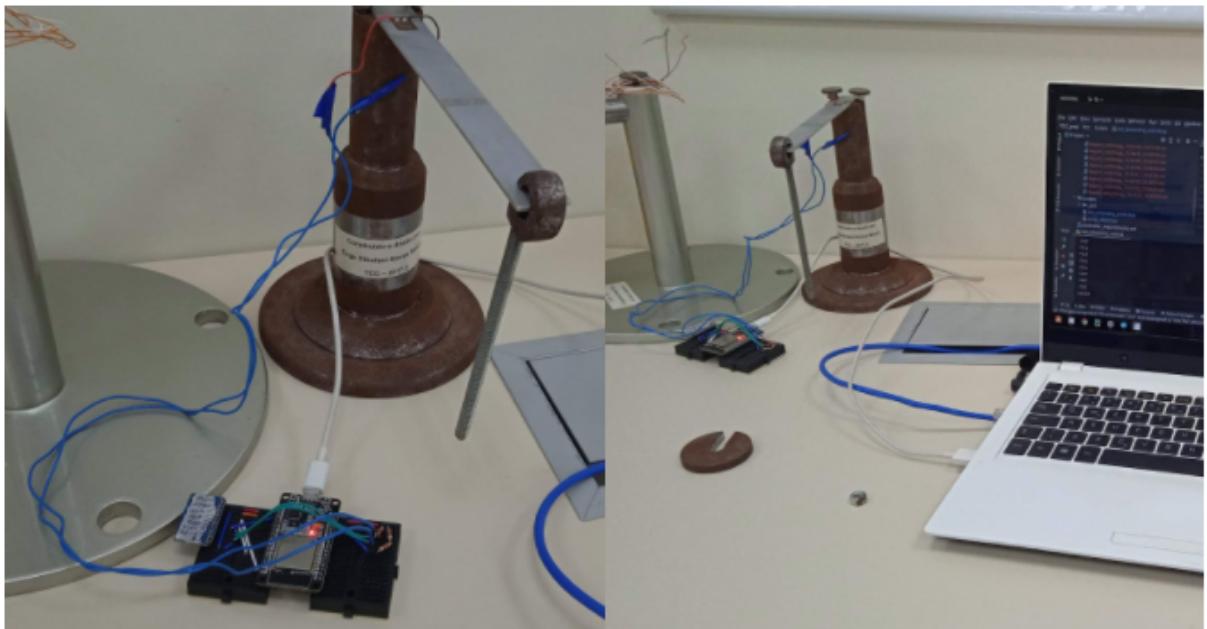
A calibração do dispositivo desenvolvido pelo autor é feita utilizando os sinais resultantes da aplicação de duas massas distintas conhecidas. Os valores nominais obtidos nos sinais alimentam a [Equação \(15\)](#) que é utilizada para converter os valores discretos obtidos em bits pelo amplificador de sinal para um valor de gradeza física desejado.

$$F(W) = \frac{output_{high}-output_{low}}{W_{high}-W_{low}}(W - W_{low}) + output_{low} \quad (15)$$

onde

$F(W)$: Função de calibração

Figura 37 – Protótipo do dispositivo desenvolvido conectado com o extensômetro no dispositivo de flexão



Fonte: O autor 2021

W : Valor nominal do sinal

W_{high} : Carga de calibração de massa alta

W_{low} : Carga de calibração de massa baixa

$output_{high}$: Valor nominal do sinal obtido pela aplicação da carga alta

$output_{low}$: Valor nominal do sinal obtido pela aplicação da carga baixa

A calibração ocorre por um método automatizado implementado no software de comunicação entre o controlador e o computador. Após calibrado, os dados são mostrados como os valores de grandeza física calibrada.

3.5.3 Cargas aplicadas

(MINELA, 2017) utiliza pesos com massas pré definidas apoiadas utilizando um fuso de fixação para aplicação das cargas no ponto “F” no dispositivo, mostrado na figura. Dentre as massas utilizadas por (MINELA, 2017) duas não foram localizadas pelo autor deste trabalho, com a finalidade de poder se obter resultados que se possam fazer comparações diretas entre os trabalhos foram substituídas as massas não encontradas por massas semelhantes.

Todos os valores de massa dos pesos utilizados foram obtidos novamente pelo autor utilizando o valor médio de três leituras obtidas por uma balança de precisão disponibilizada pelo laboratório de metrologia da UFSC Joinville. Os valores obtidos

são apresentados na [Tabela 7](#).

Tabela 7 – Valores de massas utilizadas para aplicação das cargas nos dispositivos de flexão e torção

Item	Massa obtida	Massa (MINELA, 2017)	Variação
Porca	3,06 g	14,98 g	78,84 %
Fuso	48,63 g	48,78 g	0,31 %
Peso 1	86,73 g	99,68 g	12,99 %
Peso 2	198,38 g	198,36 g	0,01 %
Peso 3	997,13 g	997,30 g	0,02 %
Peso 4	497,66 g	497,69 g	0,02 %
Peso 5	495,25 g	496,22 g	0,20 %

Fonte: O autor 2022

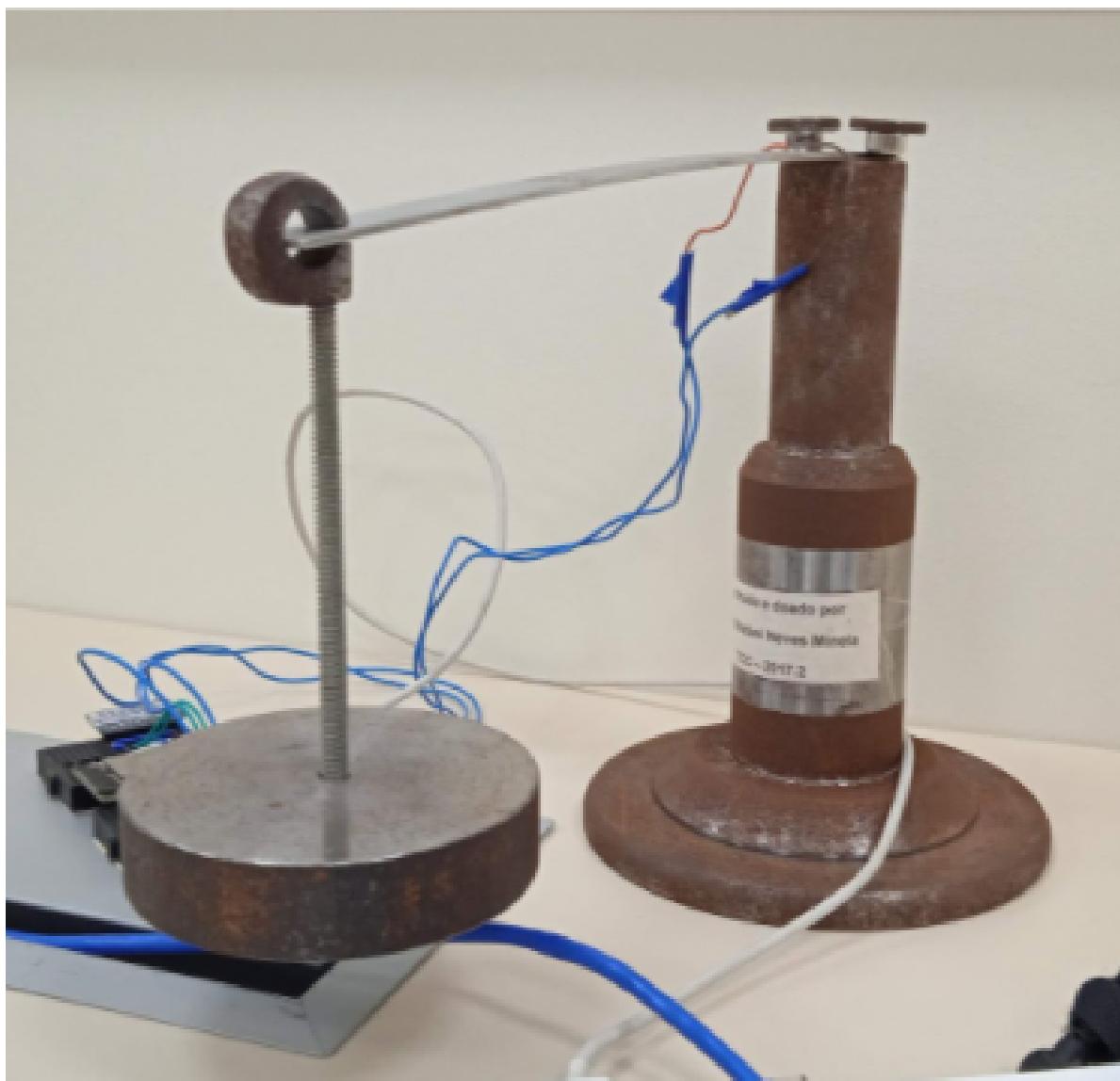
As massas da porca e do “peso 1” utilizados em ([MINELA, 2017](#)) variam de forma considerável em relação aos pesos utilizados neste trabalho, então para todas as comparações diretas de resultados experimentais obtidos entre os dois trabalhos deve ser aplicados fatores de correção de 11.42g para a porca e 12.95g para o “peso 1”.

O método de aplicação de cargas é caracterizado pela aplicação na extremidade livre da viga das massas utilizando o fuso como suporte, a Figura mostra a viga de alumínio defletida pela aplicação da carga do “peso 3”.

Na extremidade da viga encontra-se uma demarcação para auxiliar o posicionamento do apoio do fuso e as cargas são aplicadas de maneira cuidadosa de modo que não sejam geradas forças de impulso na viga. O sistema de medição obtém valores em bits proporcionais a carga aplicada no experimento.

Após a obtenção e armazenamento dos sinais obtidos nos experimentos prosegue-se para a etapa de análise dos resultados.

Figura 38 – Aplicação do 'peso 3' no dispositivo de flexão



Fonte: O autor 2021

4 RESULTADOS

Este capítulo apresenta os sinais e valores obtidos no experimento realizado no dispositivo de flexão. Os tópicos aqui analisados apresentam a comparação dos resultados obtidos pelo experimento realizado pelo autor com os valores obtidos pelo estudo de caso feito por ([MINELA, 2017](#)).

O objetivo primário da comparação dos resultados é o de se obter dados descritivos de performance do dispositivo desenvolvido em relação a um sistema de medição industrial homologado, no final do capítulo são indicados as situações de melhor performance do protótipo.

4.1 SINAIS OBTIDOS

Os sinais captados pelo sistema de medição desenvolvido seguem um formato trapezoidal, onde as zonas iniciais e finais representam os momentos em que a viga não se encontrava sob a aplicação da carga, e a zona intermediária representa a total aplicação da carga no dispositivo.

4.1.1 Sinais de calibração

Nota-se que os valores obtidos sem cargas aplicadas não se igualam a valores nulos, isso acontece pois existe um diferencial de tensão entre os pólos da ponte de Wheatstone, esse diferencial pode ser causado por imprecisão nos valores de resistência dos resistores utilizados, fatores térmicos na execução do experimento que alteram a resistência do extensômetro, ou imprecisões ou ruídos que podem estar presentes nos componentes eletrônicos do amplificador de sinal utilizado.

Uma vez que o sinal sem cargas aplicadas apresenta ruídos e flutuações indesejadas, foi decidido a utilização de duas cargas para calibração do dispositivo, os sinais obtidos pela aplicação das cargas são mostradas nas [Figura 40](#) e [Figura 41](#)

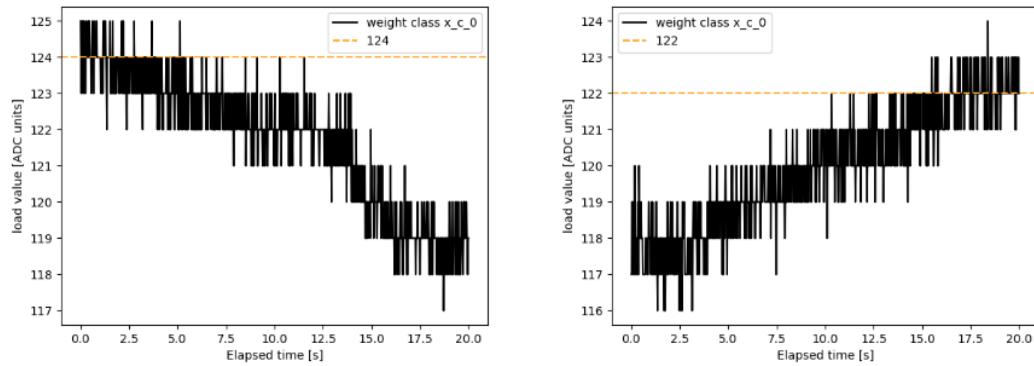
4.1.2 Ruídos presentes

Os ruídos presentes no sinal se mostram de maior criticidade nas situações em que menores cargas são aplicadas.

4.1.3 Valores nominais

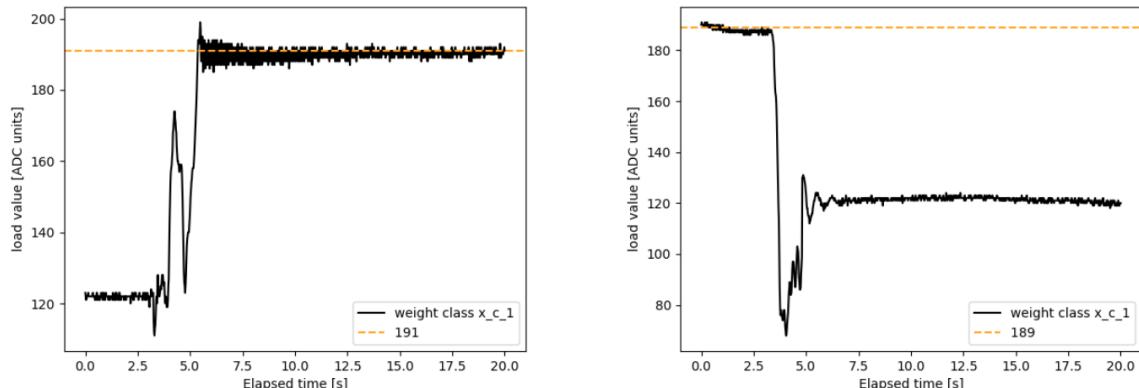
A [Figura 42](#) mostra os valores nominais de deformação para cada leitura da zona intermediária de cada sinal para cada combinação de cargas realizadas na execução do experimento, assim como o valor médio das amostras para cada combinação e o desvio padrão dos dados na zona intermediária de cada sinal como avaliação dos ruídos presentes.

Figura 39 – Sinal obtido sem a aplicação de cargas no dispositivo



Fonte: O autor 2022

Figura 40 – Sinal obtido pela aplicação da cargas de calibração baixa

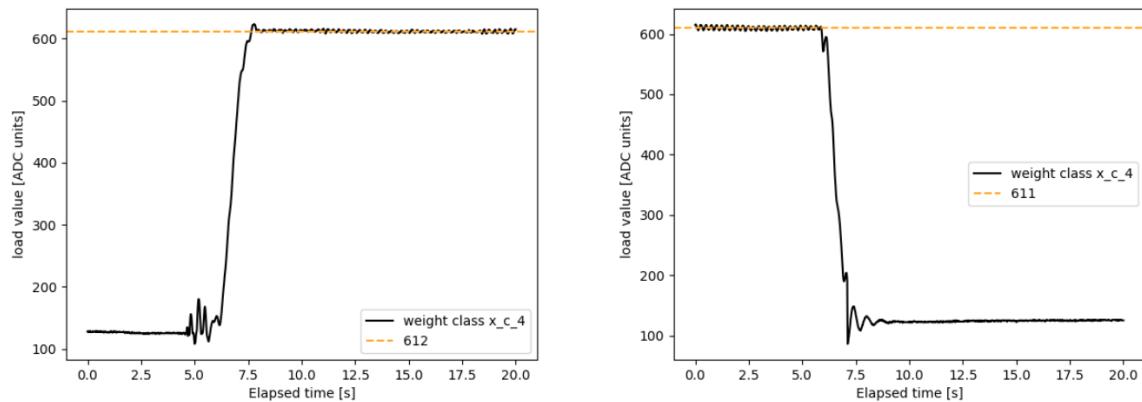


Fonte: O autor 2022

4.2 FUNÇÕES DE AJUSTE

A resolução das funções de minimização de mínimos quadrados é feita utilizando a função `stats.linregress` do pacote SciPy, foram analisados a influência da utilização de quantidade e diferentes pontos nos indicadores obtidos na criação da função de ajuste, conforme a [Figura 43](#).

Figura 41 – Sinal obtido pela aplicação da cargas de calibração alta



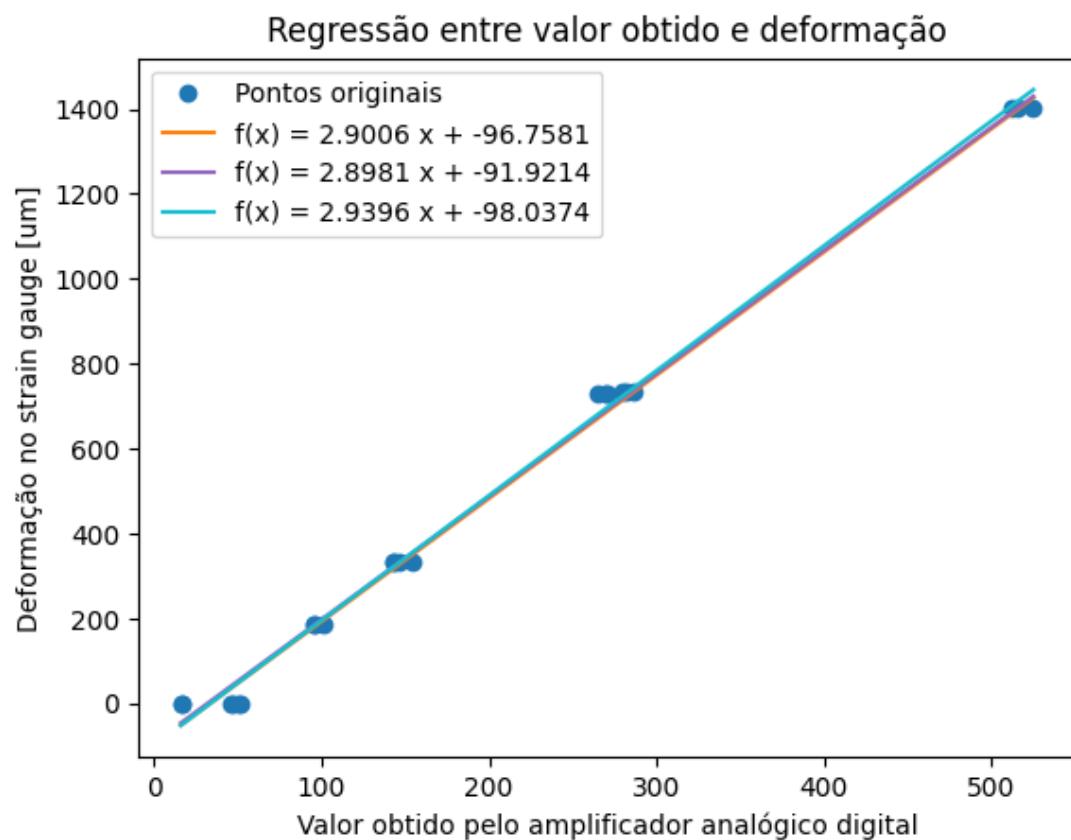
Fonte: O autor 2022

Figura 42 – Amostras e valores médios obtidos

Carga aplicada	Valor das amostras	Valor médio das amostras
weight_value	Valor obtido	Valor médio
0.14 kg	179	185.57
0.14 kg	182	
0.14 kg	196	
0.25 kg	318	320.93
0.25 kg	318	
0.25 kg	327	
0.5469 kg	672	680.60
0.5469 kg	684	
0.5469 kg	686	
0.5494 kg	715	723.15
0.5494 kg	721	
0.5494 kg	733	
1.05 kg	1388	1403.82
1.05 kg	1397	
1.05 kg	1426	

Fonte: O autor 2022

Figura 43 – Função de ajuste de dados de deformação em relação aos valores obtidos no amplificador



Fonte: O autor 2022

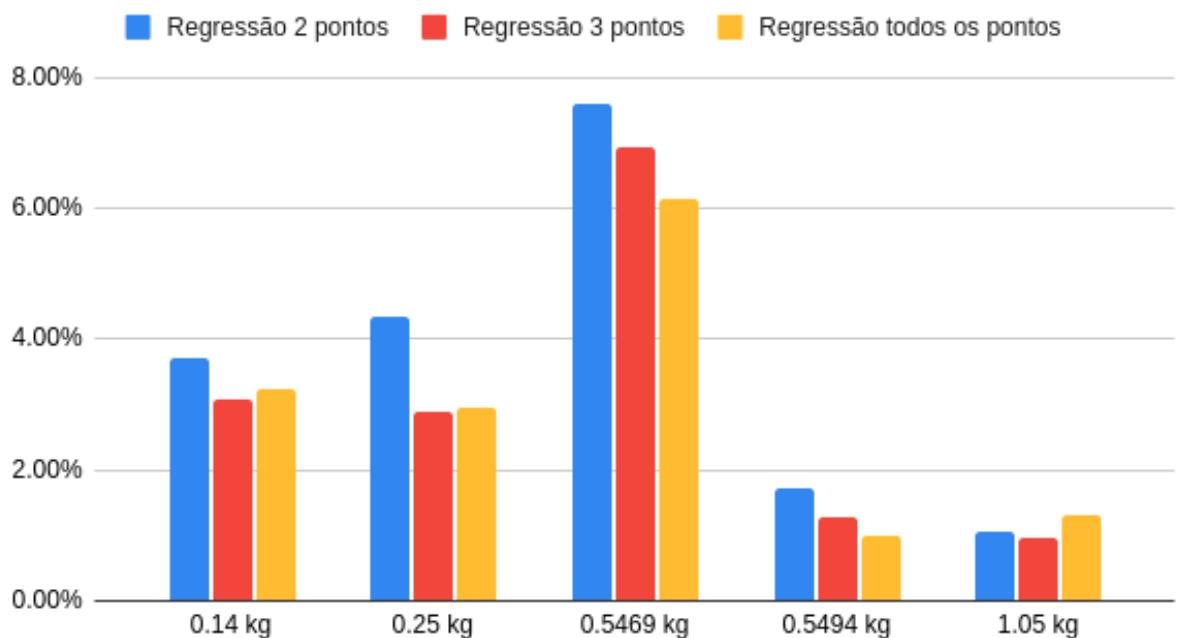
Figura 44 – Comparação dos valores convertidos usando diferentes funções de ajuste

Carga aplicada	Valor das amostras			Valor médio das amostras		
	Reg 2 pts	Reg 3 pts	Reg all pts	Reg 2 pts	Reg 3 pts	Reg all pts
0.14 kg	179	183	181	185.57	190.16	188.09
0.14 kg	182	186	184			
0.14 kg	196	201	199			
0.25 kg	318	323	322	320.93	325.41	325.27
0.25 kg	318	323	322			
0.25 kg	327	331	331			
0.5469 kg	672	676	681	680.60	684.77	689.79
0.5469 kg	684	688	693			
0.5469 kg	686	691	696			
0.5494 kg	715	720	725	723.15	727.28	732.90
0.5494 kg	721	725	731			
0.5494 kg	733	737	743			
1.05 kg	1388	1392	1407	1403.82	1407.37	1422.74
1.05 kg	1397	1401	1416			
1.05 kg	1426	1430	1445			

Fonte: O autor 2022

Figura 45 – Erro entre valores encontrados com a função de ajuste e os valores esperados pela análise analítica

Erro para qnt de pontos da regressão



Fonte: O autor 2022

5 CONSIDERAÇÕES FINAIS

Lipsum me [31-33]

REFERÊNCIAS

- AVIA. **HX711 Datasheet**. English. Jan. 2022. Disponível em:
[<https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf>](https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf).
Citado na p. [29](#).
- BACK, Nelson *et al.* **Projeto integrado de produtos: planejamento, concepção e modelagem**. [S.I.]: Barueri: Manole, 2008. Citado nas pp. [32](#), [35](#).
- DEVELOPERS, NumPy. **Documentação NumPy**. English. Jan. 2022. Disponível em:
 [<https://numpy.org/doc/stable/>](https://numpy.org/doc/stable/).
- DEVELOPERS, Pandas. **Documentação Pandas**. English. Jan. 2022. Disponível em:
 [<https://pandas.pydata.org/docs/>](https://pandas.pydata.org/docs/). Citado na p. [31](#).
- DEVELOPERS, SciPy. **Documentação SciPy**. English. Jan. 2022. Disponível em:
 [<https://docs.scipy.org/doc/>](https://docs.scipy.org/doc/). Citado na p. [32](#).
- DOEBELIN, Ernest O. **Measurement system: Application and design**. 3. ed. Ohio: McGraw- Hill, 1983.
- ESPRESSIF. **Documentação ESP8266**. English. Jan. 2022. Disponível em:
[<https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf>](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). Citado na p. [28](#).
- GHARGHAN, Sadik Kamel; NORDIN, Rosdiadee; ISMAIL, Mahamod. DEVELOPMENT AND VALIDATION OF A TRACK BICYCLE INSTRUMENT FOR TORQUE MEASUREMENT USING THE ZIGBEE WIRELESS SENSOR NETWORK. **International Journal on Smart Sensing and Intelligent Systems**, Walter de Gruyter GmbH, v. 10, n. 1, p. 124–145, 2017. DOI: [10.21307/ijssis-2017-206](https://doi.org/10.21307/ijssis-2017-206). Citado na p. [40](#).
- HIBBEKER, R. C. **Resistência dos materiais**. 7. ed. São Paulo (SP): Pearson Education do Brasil, 2010. ISBN 9788576053736. Citado nas pp. [14](#), [16–21](#).
- HOLLMAN, Jack Philip. **Experimental methods for engineers**. 8. ed. [S.I.]: McGraw - Hill, 2011. Citado nas pp. [21–25](#), [27](#), [28](#), [30](#).
- ISSO. **Documentação DMI TCR 44es**. English. Jan. 2022. Disponível em:
[<https://isso.digital/produtos/48/dmi_tcr_44es>](https://isso.digital/produtos/48/dmi_tcr_44es). Citado nas pp. [37](#), [38](#).

ISSO. **Documentação DMI TCR 88es.** English. Jan. 2022. Disponível em: <https://isso.digital/produtos/48/dmi_tcr_88es>.

MADGETECH. **Documentação Bridge101A.** English. Jan. 2022. Disponível em: <<https://www.madgetech.com/products/bridge101a/>>. Citado nas pp. 37, 39.

MINELA, Sthefani Neves. **EXTENSOMETRIA: ESTUDO E APLICAÇÃO.** 2017. Diss. (Mestrado) – UFSC, Joinville. Citado nas pp. 15, 60–65, 67.

NEDIP-UFSC. **Metodologia PRODIP.** Dez. 2021. Disponível em: <<https://nedip.ufsc.br/sessao/modelo-prodip-1>>. Citado nas pp. 32–36.

NIEDWOROK, Andrzej. Dynamic torque measurement of drive transmission in vehicles being in motion using strain gauges measurement technique and wireless data transmission. *In:* 2014 16th European Conference on Power Electronics and Applications. [S.I.]: IEEE, ago. 2014. DOI: 10.1109/epe.2014.6910707. Citado na p. 39.

NORTON, Robert. **Machine design : an integrated approach.** Edição: 4th. 4. ed. Boston: Prentice Hall, 2011. ISBN 9780136123705. Citado nas pp. 17–19, 21.

NURPRASETIO, Ignatius Pulung *et al.* Development of Static and Dynamic Online Measurement System for Ground Vehicles. *In:* 2018 5th International Conference on Electric Vehicular Technology (ICEVT). [S.I.]: IEEE, out. 2018. DOI: 10.1109/icevt.2018.8628346. Citado nas pp. 14, 40.

SILVA, Anderson; VARANIS, Marcus; MERELES, Arthur Guilherme. A consistent experimental approach for strain measurement: Two case studies. *In:* PROCEEDINGS of the XXXVIII Iberian Latin American Congress on Computational Methods in Engineering. [S.I.]: ABMEC Brazilian Association of Computational Methods in Engineering, 2017. DOI: 10.20906/cps/cilamce2017-0718. Citado na p. 40.

SILVA, Anderson Langone; VARANIS, Marcus; MERELES, Arthur Guilherme *et al.* A study of strain and deformation measurement using the Arduino microcontroller and strain gauges devices. **Revista Brasileira de Ensino de Física**, FapUNIFESP (SciELO), v. 41, n. 3, dez. 2018. DOI: 10.1590/1806-9126-rbef-2018-0206.

TIM HALL, J.-P. Stacey. **Python 3 for Absolute Beginners.** [S.I.]: Springer-Verlag GmbH, mar. 2010. ISBN 9781430216339. Disponível em: <https://www.ebook.de/de/product/12469709/tim_hall_j_p_stacey_python_3_for_absolute_beginners.html>. Citado na p. 31.

Apêndices

APÊNDICE A – ALGORITMO DE PROGRAMAÇÃO DO CONTROLADOR DO DISPOSITIVO

```
1
2 #include <ESP8266WiFi.h>
3 #include <WiFiClient.h>
4 #include <ESP8266WebServer.h>
5 #include <ESP8266mDNS.h>
6 #include <Q2HX711.h>
7
8 #ifndef STASSID
9 #define STASSID "SSID"
10 #define STAPSK "WIFIPASS"
11 #endif
12
13 const char* ssid      = STASSID;
14 const char* password = STAPSK;
15
16 const byte hx711_data_pin = D2;
17 const byte hx711_clock_pin = D1;
18
19 Q2HX711 hx711(hx711_data_pin, hx711_clock_pin);
20
21 ESP8266WebServer server(80);
22
23 void handleRead() {
24     if (server.method() != HTTP_POST) {
25         server.send(405, "text/plain", "Method now allowed");
26     } else {
27         server.send(200, "text/plain", String(hx711.read()) +
28                     server.arg("plain"));
29     }
30 }
31 void setup(void) {
32     Serial.begin(9600);
33     WiFi.begin(ssid, password);
34     Serial.println("");
35
36     // Wait for connection
37     while (WiFi.status() != WL_CONNECTED) {
```

```
38     delay(500);
39     Serial.print(".");
40 }
41 Serial.println("");
42 Serial.print("Connected to ");
43 Serial.println(ssid);
44 Serial.print("IP address: ");
45 Serial.println(WiFi.localIP());
46
47 if (MDNS.begin("esp8266")) {
48     Serial.println("MDNS responder started");
49 }
50
51 server.on("/read", handleRead);
52
53 server.begin();
54 Serial.println("HTTP server started");
55 }
56
57 void loop(void) {
58     server.handleClient();
59 }
```

APÊNDICE B – ALGORITMOS DESENVOLVIDOS PARA O SOFTWARE DE CONEXÃO

B.1 OBJETO NETWORKDEVICE

```
1
2 from time import sleep
3
4 from matplotlib import pyplot as plt
5 from numpy import array, append
6 from requests import post
7
8
9 class NetworkDevice:
10     """
11     Creates a local network connection device object
12     """
13
14     def __init__(self, ip, delay_time):
15         self.url = f'http://{ip}/read'
16         self.delay_time = delay_time
17         self.offset = 0
18         self.a = 1
19         self.b = 0
20
21     def read_line(self, calibrated: bool = False):
22         """returns the last line of data sent by the device"""
23         read = int(post(self.url).text) - self.offset
24         if calibrated:
25             return self.a * read + self.b
26         else:
27             return read
28
29     def calibrate_offset(self):
30         """sets the offset value for the device data gathering"""
31         self.offset = 0
32         self.offset =
33             round(self.read_samples(n_of_samples=40).mean())
34
35     def calibrate_readings(self, calibration_dict_1: dict,
36                           calibration_dict_2: dict):
37         """calibrate the device readings"""
```

```
36     w1 = calibration_dict_1['nominal_value']
37     w2 = calibration_dict_2['nominal_value']
38     r1 = round(calibration_dict_1['signal'].mean())
39     r2 = round(calibration_dict_1['signal'].mean())
40
41     self.b = w2 / ((r2 / r1) * (w1 - 1) + 1)
42     self.a = (w1 - self.b) / r1
43
44     def read_samples(self, n_of_samples: int, plot: bool = False,
45                      calibrated: bool = False):
46         """returns a number of data lines given a sample size"""
47
48         if not plot:
49             value_list = array([])
50             while len(value_list) < n_of_samples:
51                 value_list = append(
52                     value_list,
53                     int(self.read_line(calibrated=calibrated)))
54                 sleep(self.delay_time)
55             # print(f"sample: {len(value_list)}, value:
56             #       {int(value_list[-1])}")
57
58         else:
59             value_list = self.plot_data(n_of_samples=n_of_samples,
60                                         calibrated=calibrated)
61
62     def plot_data(self, n_of_samples: int, calibrated: bool = False):
63         """plot the numerical data received"""
64         value_list = array([])
65
66         plt.ioff()
67         fig, axs = plt.subplots(1)
68         fig.suptitle('Readings')
69
70         while len(value_list) < n_of_samples:
71             try:
72                 value_list = append(value_list,
73                                     int(self.read_line(calibrated=calibrated)))
74
75
76
77
78
79
79
```

```
74         axs.cla()
75         axs.plot(value_list[-25:])
76
77         plt.pause(self.delay_time)
78
79     except:
80         print("error!")
81
82     plt.close()
83
84     return value_list
```

B.2 INTERFACE GRÁFICA

```
1 """
2 """
3 script to start the user interface
4 """
5
6 from datetime import datetime
7 from time import sleep
8
9 from tkinter import * # sudo apt-get install python3-tk
10 from tkinter import scrolledtext
11 from tkinter.ttk import *
12
13 from pandas import DataFrame
14
15 from device.network_device import NetworkDevice
16
17
18 def gui_start():
19     """starts the GUI"""
20
21     def button_connect():
22         """connects device in the selected ip address"""
23         txt_address.config(state='disabled')
24         txt_read_delay.config(state='disabled')
25         btn_connect.config(state='disabled')
26         btn_disconnect.config(state='normal')
27
28         nonlocal wireless_device
```

```
29     wireless_device = NetworkDevice(
30         ip=txt_address.get(),
31         delay_time=int(txt_read_delay.get())*1E-3
32     )
33     check_device_compatibility(wireless_device)
34
35     def button_stop():
36         """close connection with device"""
37         txt_address.config(state='normal')
38         txt_read_delay.config(state='normal')
39         btn_connect.config(state='normal')
40         btn_disconnect.config(state='disabled')
41         btn_run.config(state='disabled')
42
43     def button_calibration_lock():
44         """calibrates the device"""
45         from numpy import array
46         btn_calibrate_lock.config(state='disabled')
47         txt_r2.config(state='disabled')
48         txt_r1.config(state='disabled')
49         txt_v1.config(state='disabled')
50         txt_v2.config(state='disabled')
51         wireless_device.calibrate_readings(
52             {'nominal_value': int(txt_v1.get()), 'signal':
53                 array([int(txt_r1.get())])},
54             {'nominal_value': int(txt_v2.get()), 'signal':
55                 array([int(txt_r2.get())])}
56         )
57         serial_monitor.config(state='normal')
58         serial_monitor.insert(END, f'device calibrated!! \n')
59         serial_monitor.insert(END, f'factor a: {wireless_device.a}\n')
60         serial_monitor.insert(END, f'factor b: {wireless_device.b}\n')
61         serial_monitor.config(state='disabled')
62         txt_v1.config(state='disabled')
63         txt_v2.config(state='disabled')
64         txt_r1.config(state='disabled')
65         txt_r2.config(state='disabled')
66         btn_calibrate_unlock.config(state='normal')

67     def button_calibration_unlock():
```

```
67     """resets device calibration"""
68     wireless_device.a = 0
69     wireless_device.b = 0
70     btn_calibrate_lock.config(state='normal')
71     txt_v1.config(state='normal')
72     txt_v2.config(state='normal')
73     txt_r1.config(state='normal')
74     txt_r2.config(state='normal')
75     btn_calibrate_unlock.config(state='disabled')
76
77     def check_device_compatibility(Device):
78         """gets a sample of data from the ip address"""
79
80         serial_monitor.config(state='normal')
81         serial_monitor.insert(END, f'connecting to device in
82                               address:\n{device_ip}...')
83
84         try:
85             result = Device.read_line()
86             serial_monitor.insert(END, f'success!! \n')
87             serial_monitor.insert(END, f'value: {result} \n')
88             serial_monitor.insert(END, f'calibrating offset... ')
89             wireless_device.calibrate_offset()
90             serial_monitor.insert(END, f'done!\n')
91             serial_monitor.see('end')
92             serial_monitor.config(state='disabled')
93             btn_run.config(state='normal')
94             btn_calibrate_lock.config(state='normal')
95             txt_v1.config(state='normal')
96             txt_v2.config(state='normal')
97             txt_r1.config(state='normal')
98             txt_r2.config(state='normal')
99
100            except:
101                serial_monitor.config(state='normal')
102                serial_monitor.insert(END, 'Incompatible device! \n\n')
103                serial_monitor.config(state='disabled')
104                serial_monitor.see('end')
105                button_stop()
106
107            def run_experiment():
108                """plots stuff"""
```

```
108     DataFrame(  
109         wireless_device.read_samples(  
110             n_of_samples=int(txt_n_samples.get()),  
111             plot=True,  
112             calibrated=True  
113         )  
114     ).to_csv(f"results/{datetime.now().__str__()}_{txt_report_name.get()}"")  
115  
116     wireless_device = None  
117  
118     # initial device port and read rate data  
119     device_ip = '192.168.1.4'  
120     read_delay = 100  
121  
122     # initial device calibration  
123     v1 = 16          # nominal value of the first data point  
124     v2 = 59          # nominal value of the second data point  
125     r1 = 24464        # read value of the first data point  
126     r2 = 96689        # read value of the first data point  
127  
128     # starting the gui object  
129     root = Tk()  
130     root.title("Telemetry center (alpha 0.1.0)")  
131  
132     # connection setup frame  
133     frame_connection = LabelFrame(root, text="Connection setup")  
134     frame_connection.pack(padx=10, pady=5)  
135  
136     lbl_address = Label(frame_connection, text="Device address: ",  
137                           font=('Arial', 11))  
138     lbl_address.grid(row=10, column=0)  
139     txt_address = Entry(frame_connection)  
140     txt_address.insert(END, device_ip)  
141     txt_address.grid(row=11, column=0, columnspan=2)  
142  
143     lbl_read_delay = Label(frame_connection, text="Sampling  
144         rate[ms]: ", font=('Arial', 11))  
145     lbl_read_delay.grid(row=30, column=0, padx=10)  
146     txt_read_delay = Entry(frame_connection)  
147     txt_read_delay.insert(END, read_delay)  
148     txt_read_delay.grid(row=31, column=0, columnspan=2)
```

```
148     btn_connect = Button(frame_connection, text='Connect',
149                           command=button_connect)
150     btn_connect.grid(row=41, column=0, pady=10)
151     btn_disconnect = Button(frame_connection, text='Disconnect',
152                               command=button_stop, state='disabled')
153     btn_disconnect.grid(row=41, column=1, padx=20)
154
155 # calibration setup frame
156 frame_calibration = LabelFrame(root, text="Device Calibration")
157 frame_calibration.pack(padx=10, pady=5)
158
159
160     lbl_v1 = Label(frame_calibration, text="Nominal value 1: ",
161                     font=('Arial', 11))
162     lbl_v1.grid(row=50, column=0)
163     txt_v1 = Entry(frame_calibration)
164     txt_v1.insert(END, v1)
165     txt_v1.config(state='disabled')
166     txt_v1.grid(row=51, column=0, columnspan=1)
167
168     lbl_v2 = Label(frame_calibration, text="Nominal value 2: ",
169                     font=('Arial', 11))
170     lbl_v2.grid(row=60, column=0, padx=10)
171     txt_v2 = Entry(frame_calibration)
172     txt_v2.insert(END, v2)
173     txt_v2.config(state='disabled')
174     txt_v2.grid(row=61, column=0, columnspan=1)
175
176     lbl_r1 = Label(frame_calibration, text="Read average 1: ",
177                     font=('Arial', 11))
178     lbl_r1.grid(row=50, column=1)
179     txt_r1 = Entry(frame_calibration)
180     txt_r1.insert(END, r1)
181     txt_r1.config(state='disabled')
182     txt_r1.grid(row=51, column=1, columnspan=1)
183
184     lbl_r2 = Label(frame_calibration, text="Read average 2: ",
185                     font=('Arial', 11))
186     lbl_r2.grid(row=60, column=1)
187     txt_r2 = Entry(frame_calibration)
188     txt_r2.insert(END, r2)
189     txt_r2.config(state='disabled')
```

```
184     txt_r2.grid(row=61, column=1, columnspan=1)
185
186     btn_calibrate_lock = Button(frame_calibration, text='Calibrate',
187         command=button_calibration_lock, state='disabled')
188     btn_calibrate_lock.grid(row=71, column=0, pady=10)
189
190     btn_calibrate_unlock = Button(frame_calibration, text='Reset',
191         command=button_calibration_unlock, state='disabled')
192     btn_calibrate_unlock.grid(row=71, column=1, pady=10)
193
194     # experiment frame
195     frame_monitor = LabelFrame(root, text="Serial monitor")
196     frame_monitor.pack(padx=10, pady=5)
197
198     serial_monitor = scrolledtext.ScrolledText(frame_monitor,
199         font=('Arial', 10), width=32, height=10, state='disabled')
200     serial_monitor.grid(row=10, column=0, columnspan=2, padx=10,
201         pady=5)
202
203     lbl_report_name = Label(frame_monitor, text="Report name: ",
204         font=('Arial', 11))
205     lbl_report_name.grid(row=20, column=0, padx=0)
206     txt_report_name = Entry(frame_monitor)
207     txt_report_name.insert(END, '')
208     txt_report_name.grid(row=21, column=0, columnspan=2)
209
210     lbl_n_samples = Label(frame_monitor, text="Number of samples to
211         get: ", font=('Arial', 11))
212     lbl_n_samples.grid(row=30, column=0, padx=0)
213     txt_n_samples = Entry(frame_monitor)
214     txt_n_samples.insert(END, 250)
215     txt_n_samples.grid(row=31, column=0, columnspan=2)
216
217     btn_run = Button(frame_monitor, text='run',
218         command=run_experiment, state='disabled')
219     btn_run.grid(row=31, column=1, pady=10)
220
221     # kill process button
222     frame_root = Frame(root)
223     frame_root.pack()
224     btn_kill = Button(frame_root, text='Exit', command=root.destroy)
225     btn_kill.grid(row=100, pady=5)
```

```
219
220     # keep gui running in loop
221     root.mainloop()
222
223
224 if __name__ == '__main__':
225     gui_start()
```