

UNIVERSIDADE DO OESTE DE SANTA CATARINA

EDUARDO KAGINSKI BEVILAQUA

MEDIDOR ELETRÔNICO DA PRODUÇÃO LEITEIRA POR ANIMAL PARA SISTEMA
DE ORDENHA BALDE AO PÉ

CHAPECÓ, SC
2021

EDUARDO KAGINSKI BEVILAQUA

MEDIDOR ELETRÔNICO DA PRODUÇÃO LEITEIRA POR ANIMAL PARA SISTEMA
DE ORDENHA BALDE AO PÉ

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação, Área das Ciências Exatas e da Terra, da Universidade do Oeste de Santa Catarina – UNOESC Campus de Chapecó como requisito parcial à obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Jacson Luiz Matte

Chapecó, SC
2021

*Aos meus familiares pelo apoio e confiança
irrestrito em todos os momentos de minha vida.*

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, pela força cedida nos momentos difíceis e pela oportunidade de concluir essa etapa importante da minha vida.

Agradecer também aos meus pais, Ezequiel Bevilaqua e Solange Bevilaqua, meu irmão Vitor Bevilaqua e minha vó Marines Riva, que sempre acreditaram em mim e por nem um momento deixaram de me apoiar, sempre prestando todo o apoio necessário para realizar meus objetivos.

Minha namora, Júlia Rossari que esteve comigo desde o início da minha graduação, sempre me ajudando, apoiando e entendendo meus momentos de ausência em decorrência desse trabalho.

Meu Tio, José Antônio Riva, pelo espaço cedido para a realização desse trabalho.

A todos os professores que me auxiliaram no desenvolvimento desse trabalho. Em especial o professor Jacson Luiz Matte, que aceitou me orientar nesse trabalho e nunca deixou de prestar o suporte necessário.

A todos os meus colegas do Curso de Engenharia da Computação, pelo convívio e companhia durante esses anos.

Por último, agradecer à Universidade do Oeste de Santa Catarina e todo o seu corpo docente.

*Nossa maior fraqueza está em desistir.
O caminho mais certo de vencer é tentar mais uma vez.*
(Thomas Edison)

RESUMO

A produção de leite é uma importante atividade agrícola e muito comum no estado de Santa Catarina, novas tecnologias para esse setor estão mais comuns. Entretanto, a muitas pequenas propriedades que possuem poucas tecnologias aplicadas e pouco controle sobre suas produções. Acompanhar a produção de uma vaca-leiteira é uma tarefa importante para garantir a qualidade e reduzir os gastos na propriedade, dado que ao identificar grandes variações na produção há possibilidade ser um indicativo de doença bovina. Para fornecer ao agricultor um maior controle sobre sua produção e com um baixo custo de implementação, foi desenvolvido um sistema embrancado capaz de realizar a medição do leite produzido por animal, além de uma Dashboard para visualização dos dados coletados. A categoria de ordenhadeira mais comum nessas pequenas propriedades é o sistema Balde ao Pé, com base nisso ficou definido que a melhor forma de realizar a medição seria no tarro de leite calculando a quantidade de leite no interior do recipiente. Inicialmente alguns sensores foram selecionados, são eles: sensor de pressão, sensor laser e células de cargas, esses sensores foram testados de modo a identificar qual a melhor opção para realizar devida medição. Todas as informações coletadas são processadas por um microcontrolador ESP32, que processará os dados e enviará para um servidor WEB, onde os dados estarão disponíveis para visualização. Após os testes concluídos foi definido que o sensor de pressão não pode ser utilizado pois apresenta interferência com relação a temperatura, diminuindo a confiança do sistema. Já o sensor laser não foi capaz de medir líquidos, sendo também inviável. Dessa forma foi definido que a melhor opção é a célula de carga isso, pois, o modelo obteve a melhor precisão na captura dos dados e é de fácil instalação no sistema de ordenha. A validação do protótipo ocorreu de duas maneiras, a primeira delas quantitativa, foi realizada em bancada e foram realizadas análises para definir a precisão do sistema. Nesses testes o sistema possuiu uma margem de erro máxima de 4%. A segunda etapa de validação foi realizada junto a um produtor de leite, essa qualitativa, teve como objetivo avaliar a usabilidade do sistema, para essa avaliação foi aplicado um questionário e avaliado a partir da escala likert, obtendo o resultado de 4,2. Com isso o sistema atingiu o objetivo de realizar a medição do leite produzido por um animal, gerando as informações necessárias para que o produtor possa acompanhar o desempenho de produção e ter um melhor gerenciamento sobre rebanho no que tange os aspectos da produção leiteira.

Palavras-chave: Medidor eletrônico de leiteiro, sistemas embarcados, produção leiteira, esp32, célula de carga.

ABSTRACT

Milk production is an important agricultural activity and very common in the state of Santa Catarina, new technologies for this sector are more common. However, to many small properties that have few applied technologies and little control over their production. Monitoring the production of a dairy cow is an important task to guarantee the quality and reduce expenses on the property, since the identification of large variations in production is likely to be an indicator of bovine disease. To provide the farmer with greater control over his production and with a low cost of implementation, a white system capable of measuring the milk produced per animal was developed, as well as a Dashboard to visualize the data collected. The most common category of milking machine in these small properties is the Balde ao Pé system, based on this it was defined that the best way to perform the measurement would be in the milk container, calculating the amount of milk inside the container. Initially, some sensors were selected, they are: pressure sensor, laser sensor and load cells, these sensors were tested in order to identify the best option to perform the proper measurement. All information collected is processed by an ESP32 microcontroller, which will process the data and send it to a WEB server, where the data will be available for viewing. After the tests were completed, it was defined that the pressure sensor cannot be used because it interferes with temperature, reducing the reliability of the system. The laser sensor was not able to measure liquids, being also unfeasible. Thus, it was defined that the best option is the load cell, because the model obtained the best accuracy in data capture and is easy to install in the milking system. The prototype validation took place in two ways, the first being quantitative, it was performed on a bench and analyzes were carried out to define the accuracy of the system. In these tests the system had a maximum margin of error of 4%. The second stage of validation was carried out with a milk producer, this qualitative one, aimed to evaluate the usability of the system, for this evaluation a questionnaire was applied and evaluated using the Likert scale, obtaining a result of 4.2. With this, the system achieved the objective of measuring the milk produced by an animal, generating the necessary information so that the producer can monitor production performance and have a better management of the herd regarding aspects of milk production.

Keywords: Electronic dairy meter, embedded systems, dairy production, esp32, load cell.

LISTA DE ILUSTRAÇÕES

Ilustração 1 - Top 20 países produtores de leite e produtividade	14
Ilustração 2 - Gráfico derivados do leite em valor de venda de 2017	15
Ilustração 3 - Glândula mamária (Ubre).....	16
Ilustração 4 - Ciclo liberação do leite.....	17
Ilustração 5 - Ordenhadeira balde ao pé móvel	19
Ilustração 6 - Ordenhadeira balde ao pé fixa	20
Ilustração 7 - Exemplo medição a laser.....	21
Ilustração 8 - Estrutura sensor piezoresistivo	22
Ilustração 9 - Exemplo medição por pressão diferencial (Tanque aberto)	23
Ilustração 10 - Exemplo medição por pressão diferencial (Tanque Fechado)	24
Ilustração 11 -Modelo instalação célula de carga.....	25
Ilustração 12 - Microcontroladores encapsulados	26
Ilustração 13 - Componentes de um microcontrolador	26
Ilustração 14 - Pinout ESP32.....	27
Ilustração 15 - Conversor HX711.....	28
Ilustração 16 - ADS1115	29
Ilustração 17 - Exemplo de um sensor	30
Ilustração 18 – Sensor VL53L0X.....	31
Ilustração 19 - Sensor MPU-6050	31
Ilustração 20 - Sensor MPX5100DP	32
Ilustração 21 - Célula de carga	32
Ilustração 22 - Modelo ponte Wheatstone	33
Ilustração 23 - Arquitetura básica de um site	34
Ilustração 24 - Fluxograma esboço do projeto na prática.....	40
Ilustração 25 - Modelo cascata	44
Ilustração 26 - Diagrama de caso de uso	47
Ilustração 27 - Diagrama ER	48
Ilustração 28 - Wireframe tela login.....	49
Ilustração 29 - Wireframe tela inicial	49
Ilustração 30 - Wireframe tela animais.....	50
Ilustração 31 - Diagramas SysML	51

Ilustração 32 - Digrama de blocos	52
Ilustração 33 - Digrama de bloco interno	53
Ilustração 34 - Diagrama máquina de estados	54
Ilustração 35 - Diagrama elétrico do circuito	55
Ilustração 36 - Diagrama temporal	55
Ilustração 37 - Estrutura Framework	61
Ilustração 38 - Estrutura pasta Models	63
Ilustração 39 - Estrutura Controllers	64
Ilustração 40 - Estrutura pasta Views	68
Ilustração 41 - Resultado dashboard finalizado	70
Ilustração 42 - Resultado tela animais	71
Ilustração 43 - Resultado tela cadastrar animais	71
Ilustração 44 - Sistema montado na protoboard	72
Ilustração 45 - Sistema de energia antes da calibração	73
Ilustração 46 - Calibração sistema de energia	74
Ilustração 47 - Sistema de energia calibrado	74
Ilustração 48 - Menu de serviço sistema embarcado	77
Ilustração 49 - Tela calibração sensor laser	78
Ilustração 50 - Gráfico saída sensor de pressão	79
Ilustração 51 - Tela calibração sensor de pressão	79
Ilustração 52 - Tela de calibração célula de carga	80
Ilustração 53 - Recipiente protótipo de medição	81
Ilustração 54 - Protótipo balança	81
Ilustração 55 - Parte inferior protótipo balança	82
Ilustração 56 - Ambiente de teste dos sensores	83
Ilustração 57 - Copo de medição	83
Ilustração 58 - Teste dos sensores com 1 litro	84
Ilustração 59 - Gráfico ajuste sensor de pressão	85
Ilustração 60 - Gráfico ajuste sensor laser	87
Ilustração 61 - Diagrama de blocos atualizado	90
Ilustração 62 - Diagrama de bloco interno atualizado	91
Ilustração 63 - Diagrama elétrico atualizado	92
Ilustração 64 - Layout PCB	93
Ilustração 65 - Modelo 3D placa	93

Ilustração 66 - Placa de Circuito Impresso	94
Ilustração 67 - Placa montada na case	95
Ilustração 68 - Modelo adesivo case.....	95
Ilustração 69 - Protótipo Pronto	96
Ilustração 70 - Menu seleciona animal	98
Ilustração 71 – Diagrama de passos para o processo de ordenha	100
Ilustração 72 - o diagrama do processo de calibração	101
Ilustração 73 - Animais cadastrados no sistema	102
Ilustração 74 - Tela dashboard com os resultados obtidos no teste de laboratório	103
Ilustração 75 - Lista animais com os resultados obtidos no teste de laboratório.....	104
Ilustração 76 - Medição da corrente do circuito	104
Ilustração 77 - Gráfico eficiência MT3608	105
Ilustração 78 - Teste de campo	105
Ilustração 79 - Lista animais com os dados obtidos no teste de campo	106

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais	45
Quadro 2 - Requisitos não funcionais	45
Quadro 3 - Lista de materiais	56
Quadro 4 - Tabela Likert	59
Quadro 5 - Código SQL banco de dados.....	60
Quadro 6 - Código configuração banco de dados.....	62
Quadro 7 - Código AnimaisModel	63
Quadro 8 – Código php da função listarAnimais().....	64
Quadro 9 - Código php para inserir medida	65
Quadro 10 - Código php função obterQuantidadeMensal()	66
Quadro 11 - Código php função obterQuantidadeDiaria	66
Quadro 12 - Código php função obterQuantidadeAnimal	67
Quadro 13 - Código Ajax	68
Quadro 14 - Código que gera os gráficos	69
Quadro 15 - Bibliotecas Firmware	75
Quadro 16 - Setup Firmware	75
Quadro 17 - Código Firmware inicia sensores	76
Quadro 18 - Código firmware manipulação EEPROM.....	76
Quadro 19 - Código firmware captura dados sensore laser.....	77
Quadro 20 - Código firmware captura dados sensores de pressão	78
Quadro 21 - Código calibração célula de carga.....	80
Quadro 22 - Bibliotecas firmware atualizadas	96
Quadro 23 - Código firmware conectar wifi	97
Quadro 24 - Código firmware captura dados balança	97
Quadro 25 - Código firmware receber lista animais.....	98
Quadro 26 - Código firmware ordenha.....	99
Quadro 27 - Código firmware envia medidas para o servidor	99

LISTA DE TABELAS

Tabela 1 - Quantitativo de obras usadas por assunto.....	42
Tabela 2 - Quantitativo Total de obras usadas	43
Tabela 3 - Tabela padrão de eficiência	57
Tabela 4 - Tabela calibração sensor de pressão sem ajuste	84
Tabela 5 - Tabela calibração sensor de pressão com ajuste	85
Tabela 6 - Tabela calibração sensor laser sem ajuste	86
Tabela 7 - Tabela calibração sensor laser com ajuste.....	88
Tabela 8 - Tabela calibração célula de carga.....	89
Tabela 9 - Tabela resultado obtidos na validação de laboratório	103
Tabela 10 - Resultado questionário de avaliação	107

LISTA DE ABREVIAÇÕES E SIGLAS

A/D	- Conversor analógico-digital
ADC	- Conversor analógico-digital
Cm	- Centímetros
DER	- Diagrama entidade relacionamento
DC	- Corrente continua
ER	- Entidade relacionamento
EEPROM	- Memória Somente de Leitura Programável Apagável Eletricamente
G/ML	- Gramas por Mililitros
GPIO	- General Purpose Input/Output
I2C	- Inter-Integrated Circuit
I/O	- Input/output
KPA	- Kilopascal
L	- Litros
M	- Metros
M/S	- Metros por segundo
Ma	-Miliampères
ML	- Mililitros
Mm	- Milímetro
PCI	- Placa de circuito impresso
PSI	- Libra-força por polegada quadrada
RAM	- Random Access Memory
ROM	- Read Only Memory
SCL	- Serial Clock
SDA	- Serial Data
SGBD	- Sistema de Gerenciamento de Banco de Dados
SQL	- Structured Query Language
TCC	- Trabalho de conclusão de curso
TOF	- Time of Flight
V	- Volt
VCC	- Alimentação positiva
VIN	- Tensão de entrada

SUMÁRIO

1 INTRODUÇÃO	10
1.1 OBJETIVOS	12
1.1.1 Objetivo Geral	12
1.1.2 Objetivos Específicos.....	12
1.2 ORGANIZAÇÃO DO TRABALHO	12
2 REVISÃO DA LITERATURA	14
2.1 PRODUÇÃO LEITEIRA E CONSUMO LEITEIRO.....	14
2.2 LEITE	15
2.3 ORDENHA.....	17
2.3.1 Ordenhadeira balde ao pé.....	18
2.4 SISTEMAS DE MEDAÇÃO DE NÍVEL E VOLUME	20
2.4.1 Medição a laser	21
2.4.2 Medição por pressão diferencial	21
2.4.3 Medição por célula de carga	24
2.5 SISITEMAS EMBARCADOS	25
2.5.1 Microcontroladores	26
2.5.2 Conversores ADC	28
2.5.3 Sensores	29
2.5.4 Energia.....	33
2.6 SISTEMAS WEB	34
2.6.1 Linguagem de programação	34
2.6.2 Banco de dados	36
3 PROCEDIMENTOS METODOLÓGICOS E TÉCNICOS	38
3.1 CAMPO OU ÁREA DE ESTUDO	38

3.2 CARACTERIZAÇÃO DA METODOLOGIA DE PESQUISA	38
3.2.1 Fluxo de realização da pesquisa	39
3.3 QUESTÕES DE PESQUISA	41
3.4 APLICAÇÃO DA METODOLOGIA DE PESQUISA	41
3.4.1 Construção do Referencial Teórico	41
3.4.2 Desenvolvimento do Sistema Computacional	43
3.4.3 Participantes do Estudo	56
3.4.4 Coleta e Análise dos Dados	57
4 APRESENTAÇÃO DA SOLUÇÃO E ANÁLISE DOS RESULTADOS.	60
4.1 SOFTWARE.....	60
4.1.1 Banco de dados	60
4.1.2 Web	61
4.2 HARDWARE	71
4.2.1 Montagem protoboard	71
4.2.2 Firmware de calibração dos sensores	74
4.2.3 Montagem do protótipo de medição	80
4.2.4 Análise dos sensores	82
4.2.5 Modificações de Hardware	90
4.2.6 Hardware final e Case.....	94
4.2.7 Firmware final	96
4.3 CATARCTERISTICAS E FUNÇOES.....	100
4.4 ANÁLISE DOS RESULTADOS	101
4.4.1 Teste de Eficiência	101
4.4.2 Teste no Campo	105
5 CONSIDERAÇÕES FINAIS	108
REFERÊNCIAS	110
APÊNDICES E ANEXOS	120
APÊNDICE A – PESQUISA-ENTREVISTA COM AGRICULTOR.....	120

APÊNDICE B – QUESTIONÁRIO DE AVALIZAÇÃO DO PROJETO	122
APÊNDICE C – DIAGRAMA ELETRICO CIRCUITO COMPLETO	123
APÊNDICE D – DIAGRAMA ELETRICO CIRCUITO FINAL	124
ANEXO A – PARTES DATASHEET MPX5100DP	125
ANEXO B – PARTES DATASHEET VL53L0X	129
ANEXO C – PARTES DATASHEET MT3608	132
ANEXO D – PARTES DATASHEET ADS1115	136
ANEXO E – PARTES DATASHEET HX711	141

1 INTRODUÇÃO

A região oeste de Santa Catarina é destaque na produção de leite no estado e está em constante melhoramento buscado o aumento da produção e qualidade do leite (JOCHIMS *et al.*, 2016). No estado 78% das propriedades rurais são do modelo familiar e essas são responsáveis por 50,7% do valor total bruto da produção rural (SANTA CATARINA, 2019). Já especificamente na produção de leite o cenário mudou, em 2006 santa catariana possuía 66.226 produtores que vendiam leite e em 2017 esse número caiu para 38.613 (DEBONA, 2019), porém vale destacar que apesar disso a produção de leite cresceu e principalmente aumentou a média de produção por animal (SEBRAE, 2021).

Isso é fruto de novas tecnologias que aumentam a eficiência das propriedades agrícolas, porém muitas dessas propriedades possuem poucas tecnologias o que prejudica diretamente e indireta na produção e qualidade do produto (RODRIGUES, 2015). Muitas vezes a falta de tecnologia ocorre pelo alto custo de implementação ou até mesmo em alguns casos pela falta de produtos específicos, como é o caso do medidor eletrônico de leite para ordenhadeiras do tipo balde ao pé. Os medidores de leite existentes no mercado são instalados na tubulação do leite e como as ordenhadeiras balde ao pé não possuem uma tubulação a instalação desses sistemas não é viável.

Esse trabalho visa desenvolver um sistema embarcado capaz de aferir a quantidade de leite produzido por uma vaca em lactação, a medição ocorrerá através de um conjunto de sensores instalados no tarro do leite e funcionará somente em ordenhas do tipo balde ao pé.

Inicialmente foram realizados testes com três diferentes formas de medição, a primeira delas consiste em um sensor de distância a laser instalado na parte superior do tarro, após analisar os dados dos sensores, foram executados cálculos matemáticos para determinar a quantidade de leite que possui no tarro. Na segunda forma a medição ocorrerá com auxílio de um sensor de pressão instalado na parte inferior do tarro, com base na pressão que o líquido causa é possível detectar a quantidade de líquido presente no recipiente. A terceira forma tratasse da medição por células de carga, nesse sistema o tarro de leite foi colocado sobre uma plataforma e o sistema funcionaria como uma balança.

Todos os sinais dos sensores foram controlados por um microcontrolador ESP 32 que é o responsável por coletar os sinais gerados pelos sensores, processar os dados, exibir a quantidade de leite produzido por cada animal em um display e posteriormente enviar esses dados para um servidor web.

Possui também uma aplicação web, onde ficam disponíveis para visualização todos os dados colhidos pelo sistema embarcado, além disso, através do sistema web é possível cadastrar os animais da propriedade. Dessa forma possibilitando o controle de produção de cada animal.

Um dos pontos mais críticos em uma propriedade leiteira é a falta de controle e gestão. Geralmente em pequenas propriedades nem um tipo de dado é coletado, sejam eles sobre rebanhos, quantidade de produção, insumos, entre outros diversos dados que auxiliam o produtor na gestão de sua propriedade (RODRIGUES, 2015).

Uma pesquisa realizada pelo Sebrae-SP (2016) aponta que 65% dos produtores entrevistados não realizam nem um tipo de controle em sua propriedade, dos 35% que realizam o controle é importante destacar que 43% fazem o controle em agendas ou cadernos e 34% fazem o controle de cabeça.

Em entrevista realizada com um agricultor da região, disponível para visualização no “Apêndice A”, o agricultor destaca que utiliza apenas recursos visuais para medir a produção de um animal e não anota os dados em local nenhum, não possuindo um histórico de produção. Assim é possível afirmar que o quesito tecnologia e gerenciamento são falhos em uma propriedade e necessitam atenção.

Em medidores de leite mecânicos um problema muito grave pode acontecer, esse problema é causado, pois, as partes mecânicas funcionam como um obstáculo para o leite e isso faz com que ocorram flutuações no nível de Vácuo da tubulação. Entretanto, o maior problema desse efeito é que ele pode ser prejudicial à saúde do ubre do animal (ALMIRANTE *et al.* 2005 apud ZANINELLI; TANGORRA 2007).

Controlar a produção de leite de uma vaca é importante, pois a quantidade de leite produzido por um animal possui forte relação com a saúde deles. Caso uma vaca apresente uma queda brusca na produção, pode apontar que ela está com algum problema de saúde ou está passado por um estresse calórico. Além disso, o controle da produção também pode auxiliar na nutrição dos animais, identificando possíveis necessidades de mudança na alimentação (QUINTÃO *et al.*, 2011).

Com base nessas informações e com a finalidade de aumentar a precisão na aquisição dos dados, foram instalados sensores nos tarros para calcular a quantidade de leite produzido por cada animal. Após a conclusão do protótipo foram realizados testes de campo, onde o agricultor irá avaliar se a precisão do sistema de medição é aceitável e capaz de aumentar o controle da produção.

1.1 OBJETIVOS

Nesse tópico serão apontados os objetivos gerais e objetivos específicos do presente projeto.

1.1.1 Objetivo Geral

Desenvolver um sistema embarcado com a capacidade de calcular a quantidade de leite produzido por um animal em uma ordenha balde ao pé, dessa forma auxiliando o produtor na gestão da produção de leite.

1.1.2 Objetivos Específicos

- Realizar estudos com o intuito de aprofundar os conhecimentos sobre o processo de ordenha.
- Levantar requisitos para o desenvolvimento do modelo proposto.
- Desenvolver um protótipo do sistema embarcado.
- Desenvolver web service
- Desenvolver painel web de visualização
- Realizar testes em bancada.
- Realizar testes e validar o protótipo em uma ordenha.

1.2 ORGANIZAÇÃO DO TRABALHO

Esse trabalho está dividido em 5 capítulos, cada um descrevendo uma etapa do desenvolvimento do trabalho. No primeiro capitulo está a introdução ao tema, capitulo utilizado para introduzir o leitor ao tema, apresentando os objetivos e motivos para a realização desse trabalho.

O capítulo 2 é a revisão da literatura, um capitulo que contém todo o referencial teórico do trabalho. Para a construção do referencial teórico foram utilizados livros, artigos, documentos e sites oficiais. Em cada um dos subtópicos foram coletados diversos materiais de autores diferentes, com o intuito de aprofundar o conhecimento sobre o assunto.

O próximo capítulo é o “PROCEDIMENTOS METODOLÓGICOS E TÉCNICOS”, neste contém a parte de modelagem e metodologia do trabalho, capítulo importante para

demonstrar como foi o desenvolvimento tanto no que se refere a parte escrita, como a prática. Com os diagramas e tabelas.

O capítulo 4 é exclusivamente para descrever como foi o processo de desenvolvimento prático, uma documentação de quais passos foram necessários para se chegar à solução completa, também foram descritos os testes e resultados obtidos.

Por último o capitulo de considerações finais, o objetivo desse é comparar os resultados obtidos os resultados que eram esperados no início do trabalho. Também serão realizadas sugestões para trabalhos futuros, uma forma para que outros estudantes possam dar sequência a esse trabalho.

2 REVISÃO DA LITERATURA

Nesse capítulo serão abordados alguns assuntos importantes para o desenvolvimento deste trabalho.

2.1 PRODUÇÃO LEITEIRA E CONSUMO LEITEIRO

No ano de 2017 a produção leiteira mundial foi de 827,9 bilhões de litros, sendo deste total 675,6 bilhões referentes a leite de vaca, 120,4 bilhões de leite de Búfala e 31,9 bilhões de leite de cabra e ovelha. Quando comparando com o ano de 2016 a produção mundial obteve um crescimento de 3,3%, empurrado pelo crescimento de 6,1% na Oceania e 4,9% na Ásia. (RESENDE *et al.*, 2019)

Ilustração 1 - Top 20 países produtores de leite e produtividade

	PRODUÇÃO DE LEITE DE VACA (MILHÕES DE T)		VARIAÇÃO 2016 PARA 2017	PRODUTIVIDADE/VACA (KG/LACTAÇÃO)		VARIAÇÃO 2016 PARA 2017
	2016	2017		2016	2017	
1.EUA	96,4	97,7	1,4	10.350	10.457	1,0
2.Índia	77,4	83,6	8,0	1.576	1.643	4,3
3.Brasil	33,7	33,5	- 0,5	1.710	1.963	14,8
4.Alemanha	32,7	32,7	0,0	7.746	7.780	0,4
5.Federação Russa	30,5	30,9	1,4	4.236	4.389	3,6
6.China	37,2	30,8	- 17,2	2.921	2.561	- 12,3
7.França	24,5	24,4	- 0,2	6.744	6.722	- 0,3
8.Nova Zelândia	21,7	21,4	- 1,4	4.166	4.237	1,7
9.Turquia	16,8	18,8	11,8	3.090	3.143	1,7
10.Paquistão	13,1	16,1	22,6	1.041	1.230	18,2
11.Reino Unido	14,9	15,3	2,1	7.862	8.042	2,3
12.Holanda	14,3	14,3	- 0,2	7.985	8.587	7,5
13.Polônia	13,2	13,7	3,4	6.172	6.357	3,0
14.México	11,6	11,8	1,4	4.674	4.695	0,4
15.Itália	10,8	11,4	5,6	5.914	6.354	7,4
16.Ucrânia	10,1	10,3	1,4	4.736	4.920	3,9
17.Argentina	9,9	10,1	2,0	2.828	3.001	6,1
18.Uzbequistão	7,9	10,0	27,9	1.669	2.003	20,0
19.Austrália	7,7	8,8	14,0	4.943	5.788	17,1
20.Canadá	7,5	8,1	7,8	8.082	8.757	8,3

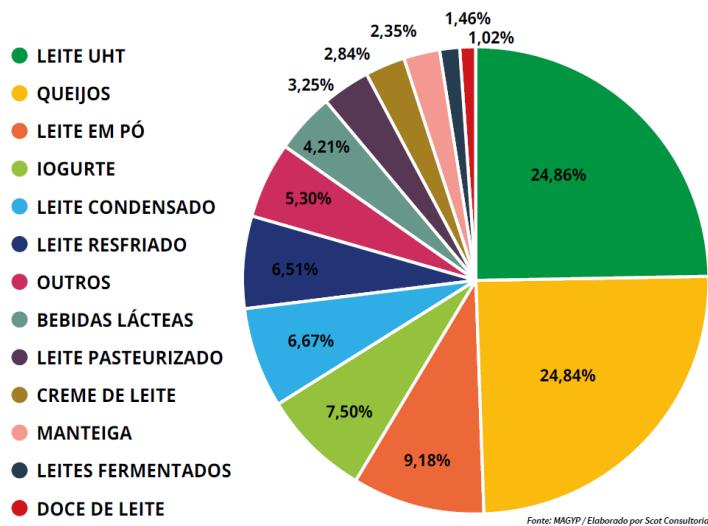
Fonte: Resende *et al.* (2019).

A Ilustração 1 apresenta os vinte maiores países produtores de leite vaca juntamente com a quantidade produzida por cada vaca, como é possível observar o brasil se encontra na 3 posição atrás da Índia e do Estados Unidos. No ano de 2017 a produção brasileira foi de 33,5 bilhões de litros de leite de vaca, total de 0,5% menor que a produção no ano anterior, porém é importante observar que produtividade por vaca aumentou em torno 14,8% um fato que é muito positivo pois demonstra que o processo está sendo modernizado.

A região brasileira que mais produz leite é a região sul do país, sendo formada por Paraná, Santa Catarina e Rio Grande do Sul. Essa região é responsável por produzir 35,7% do leite produzido no brasil, totalizando 12 bilhões de litros de leite por ano (ZOCCAL, 2019). Em Santa Catarina o destaque na produção de leite é para a região oeste do estado, nessa região a bovinocultura de leite está presente em 63% das propriedades rurais e é responsável por produzir 75% do total de leite produzido no estado (JOCHIMS *et al.*, 2016).

No mundo cada habitante consome anualmente uma média 116,5 litros de leite e essa média cresce em torno de 1,2% ao ano, desde 1999. O brasil possui um consumo acima da média global, porém é importante destacar que essa média está caindo desde 2014, em 2017 o consumo de leite por habitante chegou à marca de 166 litros mesma marca atingida no ano de 2012. Mas apesar disso o consumo possui um crescimento médio de 2,7% ao ano levando em consideração os últimos 10 anos (SIQUEIRA, 2019).

Ilustração 2 - Gráfico derivados do leite em valor de venda de 2017



Fonte: Siqueira (2019).

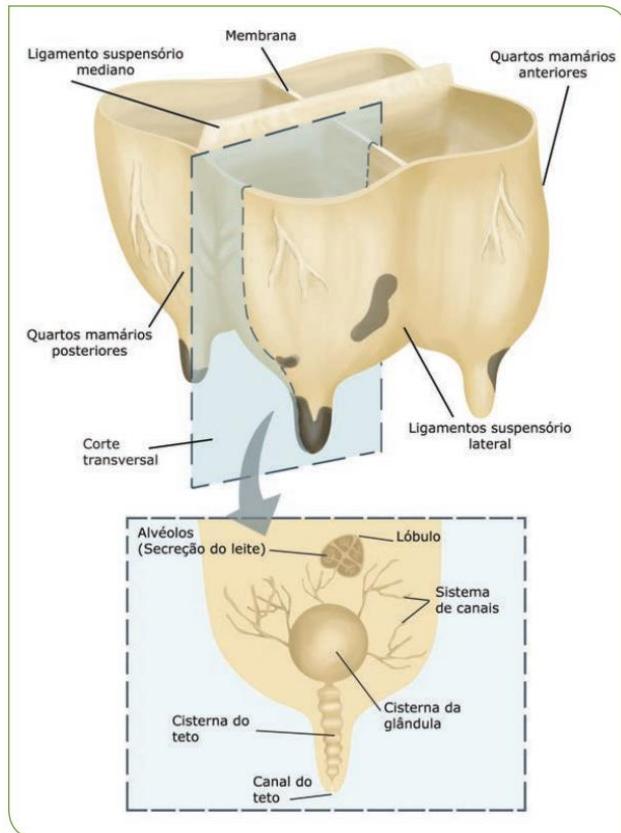
A Ilustração 2 mostra a participação no valor de vendas de cada produto derivado do leite no ano de 2016, como é possível observar dois produtos possuem as vendas disparadas dos demais são eles: O leite UHT líder de vendas e os queijos que possuem vendas muito próximas ao leite UHT.

2.2 LEITE

O leite é produzido pela glândula mamária também conhecida como ubre, essa possui a capacidade de gerar e armazenar o leite. A glândula mamária é formada por tecido e

ligamentos que fornecem sustentabilidade e resistência ao ubre, como é possível observar na Ilustração 3 a glândula é dividida em quatro quartos mamários, cada quarto funciona de forma independente e cada um possui um teto responsável por eliminar o leite. (SENAR, 2016)

Ilustração 3 - Glândula mamária (Ubre)



Fonte: Senar (2016).

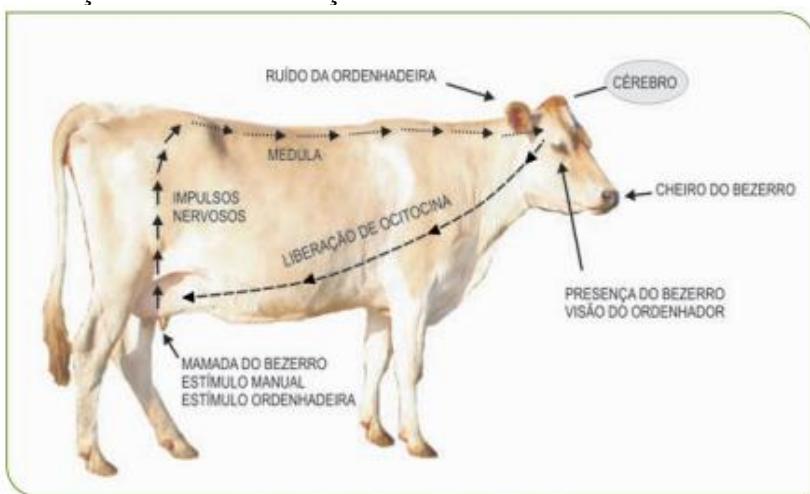
Segundo Soares (2013) o leite é gerado pelas células secretoras. Porém para que isso ocorra o sistema gastrointestinal deve absorver os nutrientes consumido pelo animal e transportar esses nutrientes para a corrente sanguínea. O sangue então passa pela glândula mamaria e serve como base para a síntese do leite, dessa forma para que uma vaca produza 1 litro de leite é preciso que passe pela glândula mamaria de 300 a 500 litros de sangue (RIBEIRO; CARVALHO, 2020).

O leite nada mais é que uma mistura de água e nutrientes, essa mistura é formada por cerca de 87% de água e o restante de elementos sólidos. Os elementos sólidos são os carboidratos que correspondem de 4,7 a 5,2%, as proteínas que correspondem de 3 a 4%, as gorduras que corresponde de 3,5 a 5,3%, as vitaminas e os sais minerais (BRITO *et al.*, 2020a).

2.3 ORDENHA

O processador realizado para extrair o leite de um animal é denominado ordenha, para que esse processo aconteça é necessário realizar estímulos externos no animal, esses estímulos podem ser a mamada de um bezerro ou até a simples presença do bezerro. Ao receber esse estímulo a vaca libera na sua corrente sanguínea um hormônio chamado ocitocina e assim o leite começa a ser expelido (SENAR, 2016).

Ilustração 4 - Ciclo liberação do leite



Fonte: Senar (2016)

A Ilustração 4 mostra o ciclo que ocorre com a vaca para iniciar o processo de ordenha, ela recebe estímulos que podem ser o cheiro do bezerro ou a visão do bezerro juntamente com um estímulo no ubre, ocasionando a liberação da ocitocina e assim iniciando o procedimento de ordenha.

É essencial que o processo de ordenha seja executado com o máximo de higiene possível, a fim de melhorar a qualidade do leite e evitar perdas na produção (BEHMER, 1999, p. 31 apud ALBA, 2018). Dessa forma é necessário realizar toda uma preparação antes de iniciar a ordenha, segundo Zanelo *et al.* (2011) os passos recomendados para realizar a ordenha são:

1. Realizar a condução das vacas de forma calma e silenciosa, para não deixar as vacas estressadas.
2. Caso o ubre esteja muito sujo, é recomendado lavá-lo.
3. Realiza a limpeza dos tetos com alguma solução desinfetante. As soluções mais utilizadas são a base de iodo.

4. Realizar a secagem dos tetos com papel toalha descartável.
5. Realizar o teste da mastite.
6. Realizar a ordenha, mantendo o ambiente tranquilo e sem barulho excessivo.
7. Após terminar a ordenha e retirar as teteiras, mergulhar os tetos em uma solução desinfetante.
8. Libera o animal para comer.
9. Realizar a limpeza dos equipamentos utilizados.

Realizando os passos mencionado acima de forma correta, as chances de contaminação no leite são muito menores, garantindo a qualidade do leite e a saúde do animal (AGROLINE, 2020).

Atualmente existem duas formas de realizar esse processo que são a ordenha manual e a ordenha mecanizada. Como o próprio nome já fala a ordenha manual é o processo de retirada do leite utilizando apenas as mãos do ordenhador, já na ordenha mecânica o processo de extração do leite é realizado com o auxílio de equipamentos, que são acoplados no ubre do animal (BETTERO, 2020).

A ordenhas mecânicas são divididas em duas categorias, são elas: ordenha balde ao pé e ordenha canalizada. Na ordenha canalizada o leite é extraído do animal e traves de tubulações e encaminhado diretamente para o tanque de refrigeração, esse processo é mais seguro pois o leite sofre menos manipulação, porém possui um custo de implementação maior e dessa forma é recomendado para produtores de grande porte (BETTERO, 2020).

2.3.1 Ordenhadeira balde ao pé

As ordenhadeiras do modelo balde ao pé são consideradas o modelo mais simples de ordenha mecânica, é recomendada para pequenos produtores e possui um baixo custo de implementação. Nesse modelo as vacas são ordenhadas individualmente e de forma automática, por meio de um sistema de vácuo (PICCOLI, 2019). Esse sistema é caracterizado por possuir uma fácil instalação, um funcionamento simples e ainda um baixo custo de manutenção (MILKPARTS, 2020).

O leite ordenhado por uma ordenha desse modelo vai para um reservatório chamado taro, que é colocado do lado no animal. Após concluir a ordenha ou o taro encher é necessário levar o mais rápido possível o leite até o refrigerador, a fim de evitar contaminação e exposição a temperatura acima do recomendado (SILVA, 2014).

O princípio de funcionamento de uma ordenha é o mesmo, independente do modelo. Basicamente uma ordenha é composta por quatro partes principais, que segundo Wattiaux (2014) são:

- Sistema de vácuo: O sistema de vácuo é composto por uma bomba geradora de vácuo, um regulador de vácuo, um reservatório e tubulações que transportam o vácuo.
- Pulsadores: São dispositivos que provocam a alteração no nível do vácuo, fazendo com que ocorra a pulsação responsável por fazer o movimento da ordenha.
- Um conjunto de teteiras: O conjunto é composto por quatro teteiras e um copo coletor, as teteiras são acopladas nos tetos da vaca e realizam o movimento de sucção. O copo coletor une as quatro teteiras e nele existe uma válvula que abre ou fecha o sistema de vácuo, essa válvula é utilizada para tirar ou acoplar as teteiras na vaca.
- Reservatório: Ao sair do conjunto de teteiras o leite passa por uma tubulação e vai até um reservatório, no caso da ordenha balde ao pé o reservatório é o taro que fica ao lado do animal. Essa é a principal diferença entre um sistema balde ao pé e o sistema canalizado, pois no sistema canalizado o leite já vai direto para o refrigerador.

Atualmente possui dois modelos de ordenha balde ao pé, que são: o sistema móvel e o sistema fixo (CARVALHO, 2020). O sistema móvel apresentado na Ilustração 5 como o próprio nome já menciona é totalmente móvel e pode ser levado em para qualquer lugar, desde que possua energia elétrica. Nesse sistema todos as partes da ordenha são instaladas sobre um carrinho com rodas, dessa forma toda a estrutura é levada até o animal para realiza a ordenha.

Ilustração 5 - Ordenhadeira balde ao pé móvel



Fonte: Implemis (2020)

Na ilustração 6 é possível observar o modelo fixo de ordenha balde ao pé, nesse modelo a bomba de vacuo fica afastada e o vácuo é transportado por tubulações. A ilustração 6 ainda apresenta de forma detalhada todas as partes da ordenhadeira.

Ilustração 6 - Ordenhadeira balde ao pé fixa



Fonte: Milkparts (2020)

2.4 SISTEMAS DE MEDIÇÃO DE NÍVEL E VOLUME

A medição de nível é utilizada na indústria para calcular a quantidade de produto que contém no interior de um reservatório, dependendo do sensor utilizado é possível obter dados contínuos do nível, aonde a medição é constante e linear, esses também podem ser aplicados em diversos tipos de reservatórios, como: tanques abertos, tanques fechados ou pressurizados, silos, entre outros (THOMAZINI; ALBUQUERQUE, 2011, p. 130).

Como o objetivo desse trabalho é obter a quantidade de leite em litros que contêm dentro de um tarro de leite cilíndrico, serão utilizados cálculos matemáticos para que seja possível obter a quantidade a partir do nível em que o leite está. O cálculo matemático para volume cilíndrico é $V = \pi * r^2 * h$, aonde V representa o volume, r representa o raio do cilindro e h é a altura do cilindro (GOUVEIA, 2020). O valor de “r” será uma constante obtida através da equação $r = \frac{\text{diametro do tarro}}{2}$, e o valor de h será uma variável referente ao nível do tarro e será obtida pelos sensores instalados no tarro.

Dentre os diversos tipos e formas diferentes de se realizar a medição de nível destacam-se as formas apresentadas nos tópicos abaixo:

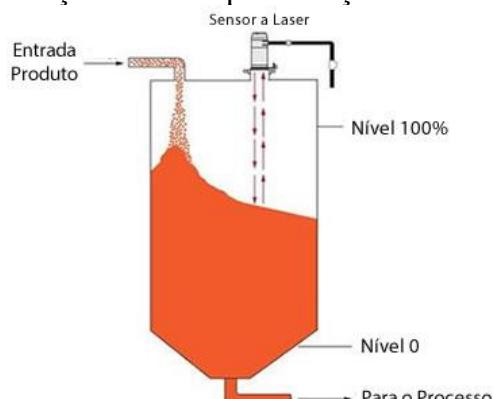
2.4.1 Medição a laser

Medir o nível a laser é uma forma de medição continua, capaz de medir sem contato direto com o produto e com precisão milimétrica. Produtos ou substâncias que são reflexivas, como por exemplo, a água ou o leite podem ser medidos dessa maneira, ainda segundo Silveira (2020) esse tipo de medição também é recomendado para substâncias abrasivas, sólidos e líquidos opacos. Outra vantagem é que esse sistema não sofre interferência por condições do ambiente aonde será realizada a medição como: temperatura ou pressão (THOMAZINI; ALBUQUERQUE, 2011, p. 139). Com isso esse se torna um bom sistema para ser utilizado na medição ne nível do tarro de leite

Um sensor a laser possui duas partes principais que são o emissor e o receptor. O emissor emite um pulso de laser na direção do objeto alvo da medição, quando o pulso toca no objeto alvo ele volta para o sensor e é recebido pelo receptor. Para obter a distância entre o sensor e o objeto, baste calcular o tempo que o pulso leva do momento em que foi disparado até o momento que ele retorna para o sensor (SILVEIRA, 2020). Esse efeito é chamado de TOF (Time of Flight) que em português significa “Tempo de voo” (MOTA, 2018).

O vidro não atrapalha na medição do laser, dessa forma é possível proteger o sensor de agentes poluentes ou até mesmo de altas pressões com um vidro aumentando a durabilidade e eficiências dele (SILVEIRA, 2020).

Ilustração 7 - Exemplo medição a laser



Fonte: Silveira (2020)

2.4.2 Medição por pressão diferencial

Para entender como utilizar o sensor de pressão diferencial para medir o nível em um reservatório, é necessário primeiramente entender o teorema de Stevin. Simon Stevin foi um

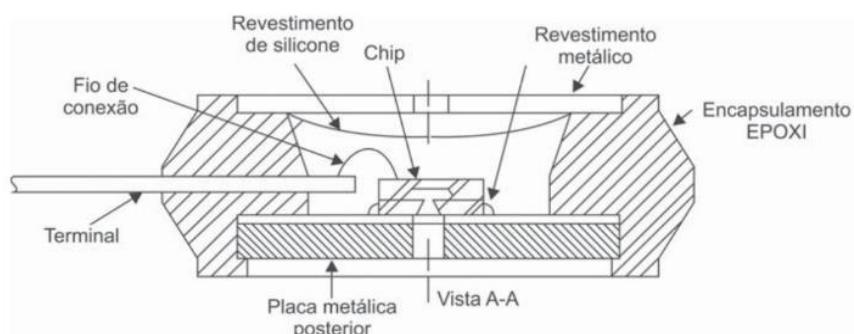
físico, engenheiro e matemático que nasceu na Bélgica no ano de 1548, eu um de seus principais feitos desvendou o mistério da pressão hidrostática (SILVA, 2020). Stevin apontou que “a pressão desenvolvida por um fluido depende exclusivamente da sua altura.” (THOMAZINI; ALBUQUERQUE, 2011, p. 145). Em outra definição Gomes, Amaral e Prado (2019) diz que “a pressão que um líquido exerce sobre uma superfície depende apenas da altura da coluna do líquido e da área da superfície, não importando o tamanho ou a forma do recipiente.”

A partir da teoria, Stevin desenvolveu a equação para o cálculo da pressão hidrostática, que é: P (pressão) = d (densidade) * g (gravidade) * h (altura). A equação indica que a pressão hidrostática em um ponto é igual ao produto da densidade do fluido, pela aceleração da gravidade e pela altura da coluna. (SILVA JÚNIOR, 2016). Aplicando conceitos de matemática é possível concluir que para determinar a altura, tendo o conhecimento sobre a pressão basta aplicar a seguinte equação: $h = \frac{P}{d \times g}$.

Dessa forma é possível utilizar um sensor de pressão diferencial para medir a pressão do fluido e assim calcular a altura da coluna. Como já dito anteriormente, tendo conhecimento sobre a altura da coluna é possível com facilidade chegar até o volume do fluido (SILVA, 2020).

Sensores de pressão integrado ou também conhecido como sensores piezoresistivo, são um tipo de sensor fabricados utilizando a tecnologia microeletrônica do silício. Um sensor piezoresistivo utiliza as características de um resistor, esse efeito resistivo é formado sobre uma lâmina de silício chamada de diafragma, quando uma pressão é aplicada sobre essa lâmina ela sobre uma modificação na sua estrutura, alterando o valor da resistência de forma linear. Dessa forma esse tipo de sensor é capaz de gerar uma saída elétrica analógica proporcional a pressão aplicada sobre o diafragma (THOMAZINI; ALBUQUERQUE, 2011, p. 126). A ilustração 8 apresenta a estrutura interna de um sensor de pressão integrado.

Ilustração 8 - Estrutura sensor piezoresistivo



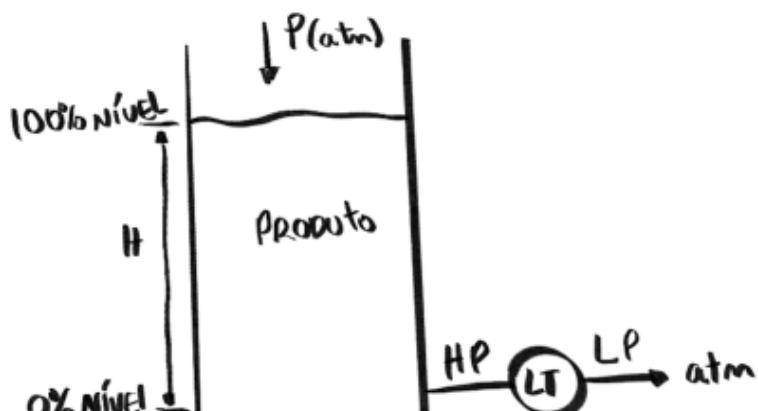
Fonte: Thomazini e Albuquerque (2011, p.126)

Os sensores de pressão diferencial possuem duas entradas de pressão, são elas, a câmera de alta pressão que sempre deve ser acoplada na parte inferior do reservatório e a câmera de baixa pressão que é acoplada na parte superior ou externa do reservatório. Para identificar o local correto da acoplagão das câmeras, é necessário conhecer o tanque ou reservatório em que o sensor será instalado, basicamente existem dois tipos de tanques, são eles, os tanques abertos e os tanques fechados (ANDRADE, 2020).

Os tanques abertos possuem uma abertura para a atmosfera, dessa forma não são pressurizados. A instalação do sensor de pressão nesse tipo de tanque é simples, basta conectar a câmara de alta pressão na parte inferior do reservatório e câmera de baixa pressão fica livre na atmosfera. Já os tanques fechados são selados, ou seja, existe uma pressurização em seu interior e essa pressão pode ser negativa (vácuo) ou positiva, nesse tipo de tanque a câmara de alta pressão fica acoplada na parte inferior do tanque e a câmara de baixa pressão fica acoplada na parte superior do tanque (ROURE, 2018).

Ao calcular então a diferença entre a câmara de baixa pressão e a câmara de alta pressão é possível obter o valor da pressão que o fluido exerce sobre o ponto aonde a câmara de alta pressão está acoplada (PUCPR, 2020). E assim utilizando a equação de Stevin descobrir o nível do líquido.

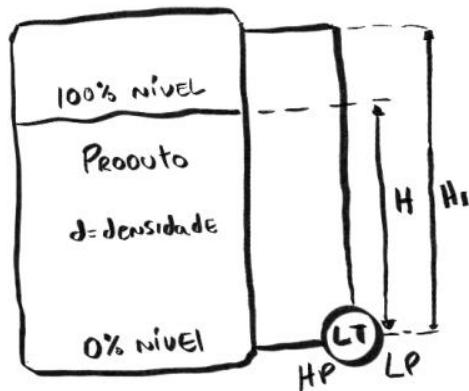
Ilustração 9 - Exemplo medição por pressão diferencial (Tanque aberto)



Fonte: Andrade (2020)

A ilustração 9 demonstra como dever a instalação do medidor de pressão em um sistema de tanque aberto, observe que a nomenclatura HP representa a câmara alta pressão e está conectada a parte inferior do reservatório. Enquanto a nomenclatura LP representa a câmara de baixa pressão e está livre na atmosfera.

Ilustração 10 - Exemplo medição por pressão diferencial (Tanque Fechado)



Fonte: Andrade (2020)

A Ilustração 10 demonstra como dever a instalação do medidor de pressão em um sistema de tanque fechado, as nomenclaturas são as mesmas da ilustração e a principal diferença é que agora a câmara de baixa pressão está conectada na parte superior do reservatório.

2.4.3 Medição por célula de carga

O sistema de medição por célula de carga é um sistema simples, que consiste basicamente por instalar transdutor na parte inferior do recipiente a ser medido. É um sistema de medição indireta, ou seja, não possui contato com o conteúdo a ser mensurado (ROURE, 2018). Quando o líquido começar a entrar no recipiente a carga aplicada sobre os transdutores também aumenta e dessa forma é possível identificar o nível (SILVEIRA, 2020).

Para entender o sistema é necessário entender esses transdutores, que são denominados células de carga. Essas células são estruturas mecânicas que quando aplicado uma carga sobre geram uma pequena deformação, essa deformação causa uma variação na resistência, essa variação é uma saída linear correspondente a carga aplicada. (THOMAZINI; ALBUQUERQUE, 2011, p. 113). Em outra definição Werneck, (1996 apud PAIXÃO; SAUTHIER; REIMBOLD, 2017). aponta células de cargas como estruturas eletromecânicas que possui a capacidade de transformar um esforço elástico uma saída elétrica.

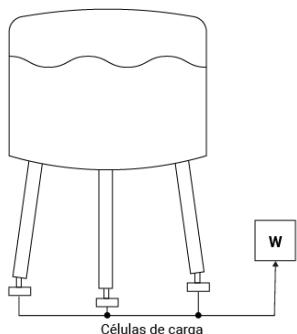
Entre as diversas aplicação para células de carga seu principal uso é em balanças comerciais e podem ser utilizadas individualmente ou em conjunto de duas, três ou quatro células, dependendo da carga que será medida. (THOMAZINI; ALBUQUERQUE, 2011, p. 117)

Mas além de serem capaz de medir o nível do recipiente uma vantagem desse tipo de medição é a capacidade de realizar a medição do volume do recipiente de forma direta. Para isso basta conhecer a fórmula da densidade, segundo César, Paoli e Andrade (2018) a densidade

é a relação entre a massa e o volume. Logo tendo conhecimento sobre a massa e a densidade é possível descobrir qual o volume do líquido ou sólido.

Na Ilustração 11 é possível observar um sistema com células de cargas instaladas em um tanque.

Ilustração 11 -Modelo instalação célula de carga



Roure (2018)

2.5 SISTEMAS EMBARCADOS

É possível definir um sistema embarcado como um software e um hardware agrupados em um único dispositivo, esse dispositivo possui seus recursos dimensionados e são construídos para executar somente uma ou várias tarefas já definidas no início do projeto. Dessa forma em um sistema embarcado o objetivo final é menor e bem definido, diferente do que acontece em um computador convencional, onde um computador é construído com o intuito de ser utilizado para múltiplos objetivos (DELAI, 2013).

Segundo Oliveira e Andrade (2010, p.20 e 72) um sistema embarcado pode ser definido como "... sistemas que possuem uma capacidade de processamento de informações vinda de um software que está sendo processado internamente nessa unidade." O autor ainda aponta que "De nada adianta ter um hardware perfeitamente montado, sem ter um software adequado para controlá-lo." esse software interno é chamado de firmware e é responsável por realizar o controle do hardware.

Os sistemas em embarcados estão inseridos na vida das pessoas, muitas vezes sem que elas percebam ou saibam o que é um sistema embarcado. Basicamente todos os dispositivos eletrônicos vendidos atualmente, são eles: impressora, forno micro-ondas, Televisão, entre outros eletrodomésticos comuns possuem um sistema embarcado, responsável por controlar o dispositivo (EMBARCADOS, 2020).

2.5.1 Microcontroladores

É fundamental para o funcionamento de um sistema embarcado que ele possua um microcontrolador. Microcontroladores são circuitos integrados utilizados para realizar o tratamento de sinais digitais e são encontrados comercialmente em diversos tipos de encapsulamento. Esses chips possuem um custo baixo e são bem pequenos, sendo ideias para aplicações desde automação residencial ou industrial até em eletrônicos e eletrodomésticos que utilizamos no dia a dia (OLIVEIRA; ANDRADE, 2010, p.34). A ilustração 12 mostra como são os microcontroladores e seus diferentes tipos de encapsulamento.

Ilustração 12 - Microcontroladores encapsulados

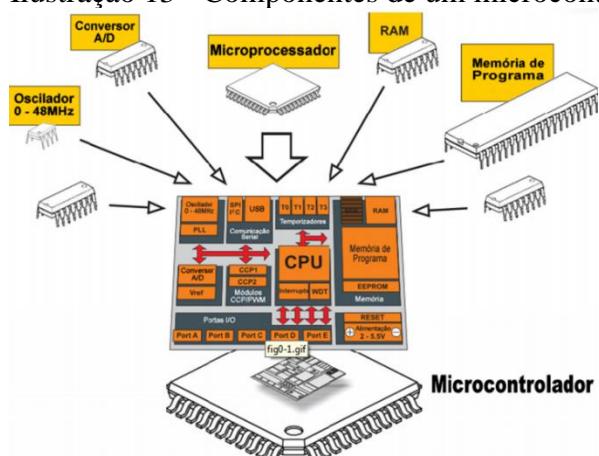


Fábio dos Reis

Fonte: Reis (2015)

No interior de um microcontrolador contém todos os periféricos necessário para executar uma tarefa, como: memória RAM, memoria ROM, unidade de processamento, portas I/O, conversores A/D, barramentos, entre outros periféricos (REIS, 2015). A ilustração 13 demonstra todos os componentes que o microcontrolador possui.

Ilustração 13 - Componentes de um microcontrolador



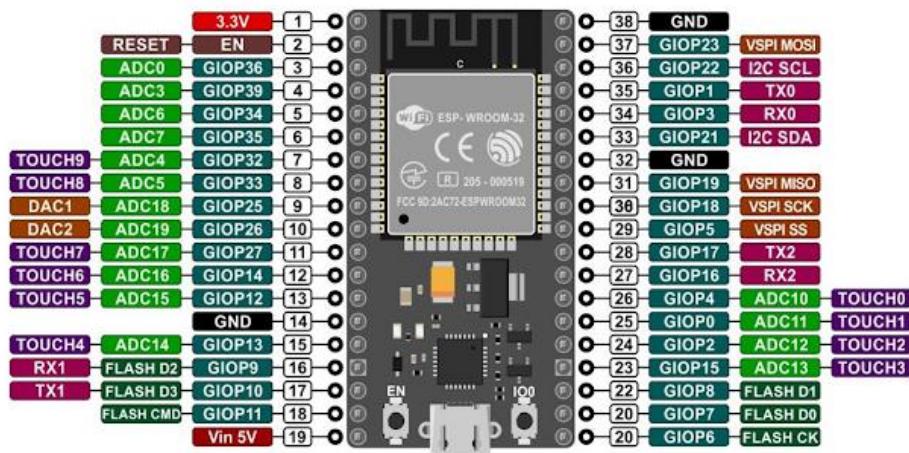
Fonte: Bastos (2020)

2.5.1.1 Esp32

No ano de 2016 a Espressif lançou o ESP32, o microcontrolador é o sucessor do ESP8266 que foi um chip revolucionário por já conter WIFI embutido, ser versátil e de baixo custo. Para manter o sucesso do seu antecessor alguns recursos foram adicionados entre eles o Bluetooth, mais memória RAM e ROM e uma capacidade de processamento maior, tudo isso mantendo o preço acessível (MURTA, 2018).

O ESP32 possui um total de 38 pinos, contendo pinos de alimentação e GPIOs (General Purpose Input/Output). Esses pinos de GPIO podem ser configurados para funcionar como entrada ou saída de dados, além de que cada porta possui a capacidade de aceitar um tipo de comunicação, entre esses formatos de comunicação se destacam sinal digitais, sinal analógico e comunicação I2C (FERNANDO K, 2018). A ilustração 14 mostra o ESP32 e a descrição de cada terminal.

Ilustração 14 - Pinout ESP32



Fonte: Fernando k (2018)

A comunicação I2C (Inter Integrated Circuit) é um protocolo de comunicação síncrona desenvolvido pela Philips ainda na década de 1980, esse protocolo possibilita a comunicação serial e bidirecional entre 2 dispositivos. A transmissão ocorre utilizando 2 canais de comunicação o SDA e o SCL, o canal SDA é responsável pela transmissão dos dados e o canal SCL é responsável pelo sinal de clock (VERONESI, 2005).

O chip pode ser alimentado através da porta USB com 5V ou através da entrada VIN utilizado também 5V, além disso o chip possui um regulador de tensão interno que converte o 5V em 3,3V. A saída desse regulador é o pino 3V3 que além de funcionar como uma saída de

3,3V funciona também como uma entrada de 3,3V, dessa forma é possível alimentar o microcontrolador com 5V ou 3,3V (MURTA, 2018).

2.5.2 Conversores ADC

Um conversor ADC também denominado conversor analógico/Digital ou conversor A/D, são dispositivos que possuem a capacidade de converter uma amostra de sinal analógico em um sinal digital. Esses conversores são utilizados para diversas aplicações e estão mais comuns na eletrônica atual, isso em razão do aumento do uso que componentes digitais (VASCONCELLOS, 2011).

É possível também definir um conversor ADC em um dispositivo que lê um sinal analógico, seja ele, uma tensão ou corrente e converte esse sinal em uma representação binária equivalente ao valor lido. Uma das propriedades de um ADC é a resolução, essa medida em bits e pode ser representada por 2^N , logo com conversor com resolução de 10bits possui resolução de 1024, ou seja, de 0 a 1023. Imaginando que esse conversor realize medições de 0 a 5V, calculado: $5/1024$ é possível chegar ao resultado de 0,00488mV e isso pode ser definido como resolução do ADC, pois 1 bit de saída do ADC representa 0,00488mV (FALLEIRO, 2015).

Dois conversores serão utilizados para a aplicação desse trabalho, sendo o HX711 e o ADS1115. O primeiro deles o HX711 Ilustração 15, é um conversor de 24 bits que será utilizado na célula de carga, esse conversor irá medir a saída da ponte de Wheatstone, amplificando o sinal de entrada e retornando uma saída binária correspondente. Esse sensor é recomendado para a aplicação de células de carga, pois possui capacidade de medir pequenos valores e já possui filtro integrado, fazendo com que o nível de ruído seja amenizado (JESUS JUNIOR; SEGUNDO, 2019).

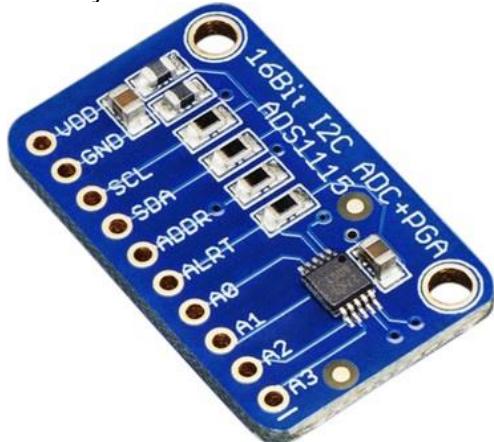
Ilustração 15 - Conversor HX711



Locatelli (2019)

O outro conversor utilizado é o ADS1115 exposto na Ilustração 16, esse um conversor analógico/Digital de 16 bits, sendo 15 bits de resolução e 1 bit de sinal positivo ou negativo. Internamente o conversor possui um amplificador, o que permite que ele seja capaz de realizar leituras de até 6,144V. Sendo $2^{15} = 32.768$, a saída digital dele é de 0 a 32.767, logo $\frac{32.767}{6.144V} = 0,1875\text{ mV}$, assim o sinal digital 1 representa 0,1875Mv (MICROCONTROLLERS LAB, 2020).

Ilustração 16 - ADS1115



Microcontrollers Lab (2020)

Além disso, esse componente possui um multiplexador na entrada, o que permite a leitura independe das quatro entradas que ele possui. A comunicação do módulo com o controlador pode ser realizada através do protocolo I2C ou SCL, sendo I2C o protocolo escolhido para esse trabalho (WAHYUDI *et al.*, 2018).

2.5.3 Sensores

Sensores são dispositivos sensíveis a formas de energias disponíveis no ambiente sejam elas térmica ou cinética por exemplo (THOMAZINI; ALBUQUERQUE, 2011, p. 15). Lamb (2015, p. 79) define sensores como sendo dispositivos fornecedores de dados para sistemas eletrônicos e em uma definição mais simples Patsko (2006) diz “Literalmente, podemos definir a palavra sensor como “aquivo que sente”.”.

Thomazini e Albuquerque (2011, p.17) utilizam o termo transdutor e o definem como “Os transdutores transformam uma grandeza física (temperatura, pressão etc.) em um sinal de tensão ou corrente que pode ser facilmente interpretado por um sistema de controle.”, sendo

assim transdutor pode ser definido como um dispositivo completo que possuem um sensor internamente. Patsko (2006) aponta que os sensores são divididos em dois tipos sendo eles: os sensores digitais e os analógicos.

Sensores digitais ou também chamados de sensores discretos são sensores que fornecem apenas duas informações (1 ou 0). Chaves e botões por exemplo são um tipo de sensor discreto, pois fornecem uma informação ao sistema, outro tipo de sensor discreto são os fotoelétricos, esse modelo possui um receptor e um emissor, o emissor gera uma luz e o estado de saída irá depender se o receptor recebeu essa luz ou não (LAMB, 2015, p.79 e 81).

Um sensor analógico, tem o objetivo de transformar uma grandeza física em uma grandeza elétrica correspondente e de forma linear e dentro de um range de atuação. Essa grandeza elétrica de saída pode ser apresentada no formato de tensão, corrente ou resistência (NATALE, 2008, p. 158).

Ilustração 17 - Exemplo de um sensor



Fonte: Autor (2020) baseado em Thomazini e Albuquerque (2011, p.17)

A Ilustração 17 representa o funcionamento de um sensor, no exemplo o sensor do tipo analógico recebe em sua entrada uma grandeza natural a temperatura, após passar pelo sensor a temperatura é transformada em um sinal elétrico correspondente, esse valor elétrico por sua vez pode ser utilizado em sistemas eletrônicos e computacionais. Alguns sensores serão utilizados no desenvolvimento deste trabalho, esses sensores serão abordados na sequência da leitura.

O sensor VL53L0X é um sensor de medição a laser de modelo TOF, com a capacidade de medir um objeto independentemente do nível de refletância a até 2 M de distância e com excelente precisão (SANTANA, 2018). O VL53L0X é um sensor de classe 1 e comprimento de onda infravermelho de 940nm “Anexo B”. O pulso laser emitido pelo sensor é invisível para o olho humano e viaja a uma velocidade de 299.792.458 m/s, dessa forma o sensor é capaz de realizar a medição da distância em apenas 33 milissegundos (DUTRA, 2018). O sensor possui um microcontrolador STM32 integrado em seu circuito, dessa forma possibilitando a comunicação I2C. Para acessar os dados capturados pelo sensor a Adafruit disponibiliza uma

biblioteca com funções implementadas, dessa forma facilitando a utilização do sensor (TEIXEIRA, 2019). A ilustração 18 apresenta o sensor de distância a laser que será utilizado no desenvolvimento deste trabalho.

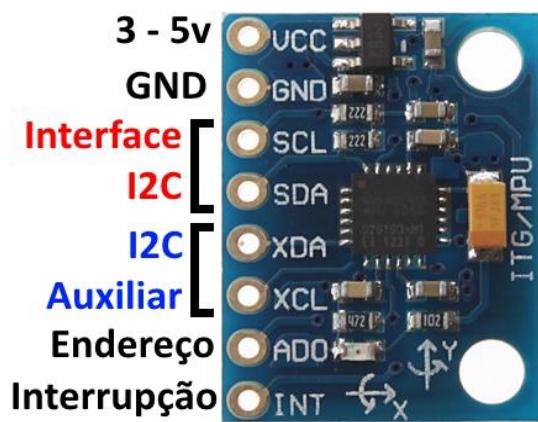
Ilustração 18 – Sensor VL53L0X



Fonte: Teixeira (2019)

O sensor MPU-6050 é um sensor que possui as funções de giroscópio e acelerômetro, ele possui um total de 6 eixos sendo eles 3 eixos (x, y e z) para o acelerômetro e 3 eixos (x, y e z) para o giroscópio. O sensor é alimentado com 5V pela porta Vcc e os dados obtidos por ele são enviados para o controlador usando uma comunicação no formato I2C, utilizado as portas SDA e SCL (SILVA *et al.*, 2014). A Ilustração 19 mostra o sensor MPU-6050 e demonstra a descrição de cada pino que ele possui, esse sensor será utilizado para medir a inclinação do tarro de leite.

Ilustração 19 - Sensor MPU-6050

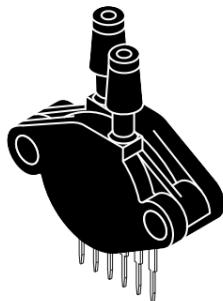


Fonte: Thomsen (2014)

Entre os sensores de pressão existentes se destacam os sensores da série MPXxxxx, esses sensores podem ser aplicados em na medição de ar, água, ou qualquer outro fluido e podem ser aplicados por exemplo na medição de nível em caixas de águas, tanques de líquidos, medição de ar comprimido, entre outras aplicações que envolvam medir a pressão (FORNELL, 2020).

O sensor MPX-5100 é um sensor de pressão diferencial projetado para uma alta gama de aplicação e com tecnologias de ponta. Possui a capacidade de medir pressões de 0 a 100KPa, equivalente a 14,5 psi. O sensor é alimentado com 5V e sua saída é um sinal analógico de 0 a 5V correspondente a pressão aplicada (FREESCALE SEMICONDUCTOR, 2009). A ilustração 20 mostra o sensor MPX-5100DP, esse é o sensor que será utilizado para medir a pressão do líquido no tarro de leite.

Ilustração 20 - Sensor MPX5100DP



Fonte: Freescale Semiconductor (2009)

Atualmente existem diversos modelos de células de cargas, com capacidades que variam de gramas para toneladas. Entre esses modelos a Ilustração 21 mostra o modelo genérico da sparkfun, uma célula relativamente barata com capacidade de pesar até 50 kg por células (SPARKFUN, 2021).

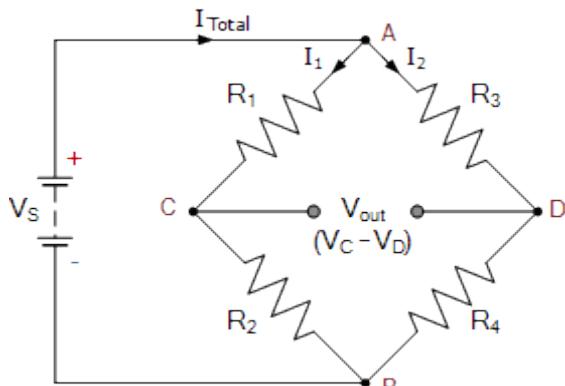
Ilustração 21 - Célula de carga



Sparkfun (2021)

Essas células possuem 3 fio de saída, sendo preto, vermelho e branco, além da possibilidade de ser ligadas individualmente ou em conjunto de duas, três ou quatro células. Para a ligação de 4 células é necessário montar uma ponte de Wheatstone entre as células conforme a Ilustração 22. Aonde os fios brancos e pretos realizam as ligações entre as células e os fios vermelhos são conectados ao ADC (LOCATELLI, 2019).

Ilustração 22 - Modelo ponte Wheatstone



Locatelli (2019)

2.5.4 Energia

Existem atualmente diversos tipos de baterias de lítio, entre elas as células cilíndricas 18650, essas celulares fornecem 3,7V e são muito utilizadas nas indústrias, em equipamentos eletrônicos portáteis e por esse motivo é fabricado em grande escala, tornando assim um produto com valor reduzido (MENAIA, 2013). O nome é uma referência as dimensões da célula 18mm de diâmetro e 65mm de comprimento, dessa forma o nome foi padronizado com 18650. Em sua composição química a célula é constituída a partir de uma folha de alumínio revestida por um material ativo formando assim o eletrodo positivo, e uma folha de alumínio revestida por carbono micro poroso formando o eletrodo negativo (FORISCHI; SOUZA, 2015).

O módulo TP4056 é um módulo de carregamento linear de corrente e tensão constante utilizado para carregar baterias de lítio (VARSHNEY, 2018). Por ser um módulo pequeno sua utilização é recomendada em aplicações portáteis, o módulo é alimentado através da entrada USB ou pinos de alimentação com 5V e a saída para carregamento é de 4,2V e a corrente pode ser ajustada. O módulo possui ainda proteção de carregamento reverso, sensor de temperatura interno e encerramento automático do ciclo de carregamento (LI *et al*, 2015).

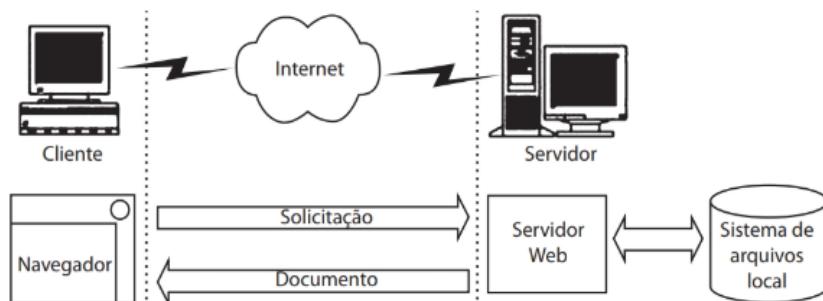
O MT3608 é um intensificador de carga DC, sendo capaz de elevar uma tensão baixa de 2V para até 28V (TRIPATHI *et al.*, 2018). O autor Azmi *et al* (2019) define como sendo um intensificador de DC para DC, que pode receber uma entrada de 2v a 24v e elevar essa tensão de 5V a 28V.

2.6 SISTEMAS WEB

O desenvolvimento de um site normalmente é dividido entre o front-end e o back-end. O back-end é responsável por desenvolver aquilo que o usuário final não consegue enxergar, como a lógica do sistema, a estrutura de dados e as funções. Já o front-end é responsável por desenvolver aquilo que o usuário final vê, que é o layout do sistema (LEDUR *et al.*, 2019, p.75).

Diferente de um site estático aonde todas as informações contidas nele são fixas e atribuídas através de códigos HTML, em uma aplicação web as informações são obtidas através de fonte de dados como um banco de dados, dessa forma as informações podem variar deixando o site dinâmico (MILETTO; BERTAGNOLLI, 2014, p.28).

Ilustração 23 - Arquitetura básica de um site



Fonte: miletto e bertagnolli (2014, p.28)

É possível visualizar na ilustração 23 que o usuário realiza uma requisição através do navegador, essa requisição é enviada para o servidor que processara e retornara os arquivos correspondentes, depois disso as informações solicitadas ficam à disposição para visualização através da interface web (MILETTO; BERTAGNOLLI, 2014, p.28).

2.6.1 Linguagem de programação

Uma das linguagens de programação mais básicas e importantes da internet é o HTML (HyperText Markup Language), que em português significa linguagem de marcação de

hipertexto. O princípio de HTML surgiu por volta de 1980, porém apenas em 1990 Tim Berners-Lee desenvolveu o primeiro navegador que servia apenas como editor de texto. A partir desse feito a linguagem foi evoluindo passando a aceitar imagens, integrações com outras linguagens, e muitos outros atributos (FLATSCHART, 2011).

O código HTML é composto por marcadores chamados de tags, cada tag representa um tipo de informação ou característica, como por exemplo a tag <p> significa parágrafo e a tag <h1> significa título (XAVIER, 2019). Por ser uma linguagem de marcação o HTML dispensa o uso de compiladores e o próprio navegador é responsável por interpretar o código, dessa forma quando acessamos um site escrito em HTML o navegador busca o código em um servidor e interpreta as tags, apresentando no navegador o conteúdo do site (VENETIANER, 1996 apud MORAIS, 2016).

Para melhorar a aparência de uma página HTML foi atribuído a seus recursos o CSS (Cascading Style Sheets) ou como conhecido em português por folha de estilos, esse recurso possibilita realizar diversas alterações visuais em um documento HTML como: alterar a cor, fonte ou tamanho de uma letra, alterar cor de fundo, desenhar bordas, entre muitos outros recursos, tornando a página mais formosa e apresentável (ALVES, 2014b, p.77).

Em 1995 Brendam Fich desenvolveu a linguagem JavaScript, por ser uma linguagem de script ela só pode ser executada em um ambiente específico que no caso é o navegador. A sua função é criar rotinas que podem ser inseridas no código HTML. (ALVES, 2015, p.157). Essas rotinas pode ser uma simples verificação de que se uma tecla do mouse ou teclado estão pressionadas, a partir das rotinas são possíveis criar eventos, esses eventos têm como objetivo alterar o comportamento de uma página HTML (FLANAGAN, 2011, p.1 e 10).

O PHP é uma linguagem de script open source utilizada no desenvolvimento de aplicações WEB, foi desenvolvida por Ramus Ledorf no ano de 1995 e tem como destaque a possibilidade de integração com a linguagem HTML. Dessa forma é possível atribuir um código PHP dentro de uma tag HTML, apenas atribuindo as tags do PHP que são <?php ?>, assim tornando as páginas HTML dinâmicas (MILETTO; BERTAGNOLLI, 2014, p.172).

A partir do PHP é possível criar aplicações de alta complexidade, garantindo confiabilidade e desempenho. Através do PHP é possível criar funções, declarar variáveis ou constantes, enfim é uma linguagem completa que além de todos os recursos encontrados em outras linguagem contém ainda funções próprias que auxiliam no desenvolvimento de suas aplicações. Para compilar e executar os códigos PHP é necessário um interpretador PHP, esse interpretador é adquirido através de um servidor web aonde os arquivos PHP serão armazenados (SARAIVA; BARRETO, 2018, p.57).

Para tornar o desenvolvimento mais ágil surgiram os frameworks. Segundo D'avillar (2019) framework pode ser definido como um “conjunto de técnicas, ferramentas ou conceitos pré-definidos usados para resolver um problema”, em outra definição Silva e Moreira (2004) aponta que “um framework pode ser definido como o esqueleto de uma aplicação...”. Sendo assim framework nada mais é do que um conjunto de códigos reutilizáveis, contendo funções, metodologias e técnicas com o intuito de facilitar o desenvolvimento de uma aplicação, servindo como base no início de um projeto, reduzindo custos e acelerando o processo de desenvolvimento (SOUZA, 2019).

2.6.2 Banco de dados

Baseado na preocupação do homem em guardar informações e posteriormente acessá-las surgiu um conceito de armazenar todas as informações em um único repositório, esse repositório foi denominado como banco de dados (CARDOSO; CARDOSO, 2012, p.15 e 16). O autor Machado (2014) define banco de dados como “... um conjunto de dados devidamente relacionados”, onde dados podem ser definidos como um elemento ou uma informação que conterá algum significado obscuro e será armazenado.

Após muitas melhorias nos conceitos de banco de dados, na década de 1980 surgiu o modelo relacional, o qual é utilizado até os dias atuais (CARDOSO; CARDOSO, 2012, p.16). No modelo relacionais os dados são armazenados em tabelas, essas tabelas são constituídas de colunas e linhas, as colunas representam os atributos, exemplo: (nome, idade) e as linhas representam os registros, exemplo: (Pedro,23). Isso possibilitou o relacionamento entre tabelas, aumentando o desempenho e segurança do banco de dados (ALVES, 2014a, p.19).

Em 1970 surgiu o SEQUEL (Structured English Query Language) que após sofrer diversas melhorias, por volta de 1977 teve seu nome alterado por questões jurídicas e passou a se chamar SQL (Structured Query Language) em português Linguagem Estruturada de Pesquisa. Essa linguagem se mostrou muito eficiente na manipulação de dados e por esse motivo se tornou popular nos SGBDs (MACHADO,2014, p.306).

Para utilizar e gerenciar um banco de dados foram desenvolvidos sistemas gerenciadores de banco de dados também conhecidos como SGBD (Data Base Management System). Entre os mais utilizados está o Mysql, ele utiliza a linguagem SQL para manipular o banco e é muito utilizado em sistemas web (PISA, 2012).

Para facilitar o gerenciamento de um banco de dados, algumas aplicações foram desenvolvidas como é o caso do phpMyAdmin. O phpMyAdmin um aplicativo desenvolvido

em PHP utilizado para criar ou realizar manutenção em banco de dados MySQL. Através dele é possível realizar o gerenciamento do banco de dados, efetuando todas as operações possíveis de um banco de dados através de uma interface intuitiva, além da possibilidade de realizar a execução de códigos SQL (SARAIVA; BARRETO, 2018, p.125).

3 PROCEDIMENTOS METODOLÓGICOS E TÉCNICOS

Esse capítulo é dividido em 6 subcapítulos, onde serão apresentados os assuntos referentes a delimitação do estudo, caracterização da metodologia de pesquisa, participantes do estudo, questões de pesquisa, aplicação da metodologia e limitações do estudo.

3.1 CAMPO OU ÁREA DE ESTUDO

O projeto visa a aplicação em pequenas propriedades rurais que possuem poucas vacas e utilizam o sistema de ordenha balde ao pe. Essas propriedades utilizam mão de obra familiar, sendo essa prática predominante no Oeste de Santa Catarina (RODRIGUES, 2015).

A área de estudo do presente trabalho foi uma propriedade de agricultura familiar pertencente ao senhor José Antônio Riva, localizada na Linha Marcon interior da cidade de Chapecó-SC. Em sua propriedade José possui 14 vacas que produz uma média de 130 litros de leite por dia utilizando uma ordenhadeira do tipo balde ao pé.

3.2 CARACTERIZAÇÃO DA METODOLOGIA DE PESQUISA

A pesquisa se caracteriza com uma pesquisa de natureza aplicada, pois ao final do trabalho foi gerado um produto aplicado resolvendo um problema em uma área específica. De acordo com Silva e Menezes (2005) a pesquisa de natureza aplicada tem como objetivo gerar conhecimento com a finalidade de desenvolver uma aplicação prática e solucionando problemas específicos.

O trabalho possui uma abordagem híbrida, ou seja, possui uma abordagem quantitativa e qualitativa. Possui uma abordagem quantitativa, pois foram analisadas as informações numéricas obtidas pelos sensores referentes a quantidade de leite produzido afim de validar o protótipo. Uma abordagem quantitativa pode ser definida como um tipo de pesquisa aonde os resultados podem ser quantificados, a resposta para um problema é obtida através da análise dos dados brutos ou manipulados através de cálculos matemáticos (FONSECA, 2002). Por outro lado, possui a abordagem qualitativa pois foi avaliada a utilização do sistema pelo produtor. A abordagem qualitativa apresenta informações descritivas do ambiente natural, dessa forma os dados são analisados individualmente e por esse motivo não é utilizada nem um tipo de cálculo matemático para definir os resultados

Segundo Gil (2008) uma pesquisa exploratória possui a característica de ser uma pesquisa aonde o objetivo é disponibilizar uma visão geral sobre um assunto específico, buscando acrescentar conhecimento sobre o assunto, esse tipo de pesquisa tem como característica realizar pesquisas bibliográficas e entrevistas com pessoas que possuem conhecimento sobre o assunto. Como base nas informações acima é possível apontar que esse trabalho possui a forma exploratória de atingir o objetivo, pois, foi necessário realizar pesquisas em materiais bibliográficos e entrevistas com pessoas da área para realizar em levantamento de informações sobre o assunto abordado (SILVA; MENEZES, 2005).

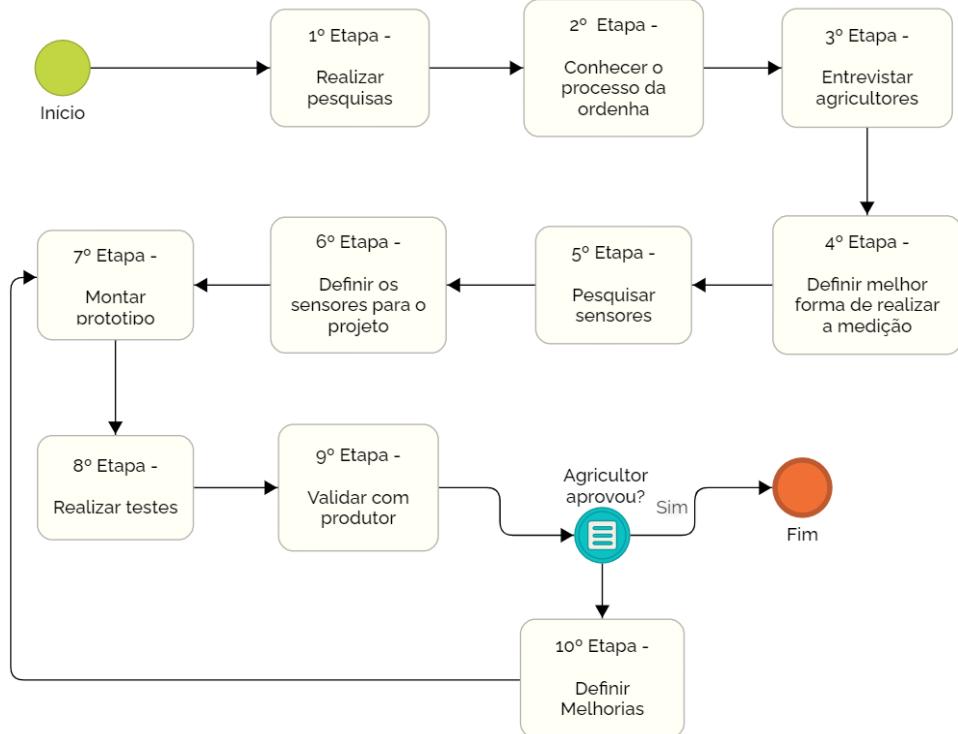
Para desenvolvimento do trabalho foram utilizados os seguintes procedimentos técnicos:

- Pesquisa bibliográfica pode ser definida como uma pesquisa que utiliza como referencias materiais já publicados em livros, artigos científicos e materiais publicados em mídias digitais (FONSECA, 2002).
- Pesquisa documental é um tipo de pesquisa semelhante a pesquisa bibliográfica, porém na pesquisa documental é utilizado materiais de diversas fontes e não apenas materiais publicados oficialmente. Esse tipo de material podem ser: artigos de revistas, fotos, reportagens, relatórios de empresas e diversos outros tipos de materiais (GIL, 2008).
- Pesquisa experimental consiste em um tipo de pesquisa onde são realizados experimentos modificando variáveis e parâmetros, com o intuito de analisar os resultados obtidos e observando a influência que cada variável causa (GIL, 2008).
- Pesquisa de campo é caracterizada por ser um tipo de pesquisa aonde além de realizar levantamento de informações através de matérias bibliográficas, se utiliza como fonte de dados entrevistas ou questionários realizados com pessoas ligadas diretamente com a área de estudo (FONSECA, 2002).
- Estudo de caso é um tipo de estudo aonde o principal objetivo é obter um conhecimento amplo sobre um determinado assunto, buscando entender os comportamentos de um determinado sistema ou área de estudo (GIL, 2008).

3.2.1 Fluxo de realização da pesquisa

Nesse capítulo será demonstrado na forma de um fluxograma que está disponível na Ilustração 24, os passos que foram realizados para a elaboração do trabalho, desde o início do próprio até o objetivo final.

Ilustração 24 - Fluxograma esboço do projeto na prática



Fonte: Autor (2020).

Na primeira etapa do projeto foram realizadas pesquisas em livros, artigos, publicações e sites com a finalidade de adquirir conhecimento sobre o assunto abordado no trabalho e até mesmo para aprimorar conhecimentos já adquiridos. Na sequência dessa etapa pesquisando em sites, revistas e vídeos foi realizada uma pesquisa com o intuito de conhecer na prática o processo realizado por um produtor na ordenha de uma vaca.

Na terceira etapa foi realizada uma entrevista a um agricultor produtor de leite, onde de forma pessoal um questionário foi respondido com o objetivo de aprofundar os conhecimentos sobre o processo prático e analisar uma forma adequada de realizar a medição. Em seguida na etapa 4 após analisar os dados obtidos na entrevista definir qual a melhor forma para se realizar a medição da quantidade de leite produzido por cada bovino leiteiro.

Com base na forma de medição definida na etapa 4, a etapa 5 constituirá em uma pesquisa mais aprofundada e específica para identificar quais sensores podem ser utilizados para realizar a medição. Na etapa seguinte com base na pesquisa realizada foi determinado quais os sensores ideias para a aplicação e como eles serão instalados no sistema de ordenha, juntamente nessa etapa foram realizados testes práticos em laboratório para medir a eficiência desses sensores. Com base nesses testes foi decidido qual o melhor sensor para realizar devida medição.

Na etapa 7 foi realizada a montagem do protótipo do sistema com base nas definições e requisitos determinados nas etapas anteriores, essa parte engloba todo o processo de desenvolvimento desde a modelagem até o produto pronto para os testes. Após concluir a montagem do protótipo na etapa 8 foram realizados testes na propriedade produtora de leite utilizando o sistema de ordenha para verificar se a medição do sistema proposto está apropriada e se o operador da ordenha aprova o produto.

A nona etapa é uma validação que ocorrerá com o produtor de leite, essa validação tem como objetivo identificar se o protótipo do sistema funciona na prática. Caso a validação falhe o projeto avança para a etapa 11 aonde são definidas melhorias e adequações a serem realizadas para melhorar o funcionamento do sistema, após isso volta para sétima etapa montando um novo ou adequando o protótipo já existente. Caso a validação da etapa 10 seja concluída com êxito o projeto é determinado como concluído.

3.3 QUESTÕES DE PESQUISA

- O protótipo do sistema eletrônico de medição de leite proposto neste projeto é apto para medir a quantidade de leite em um sistema de ordenha balde ao pé com precisão aceitável?

3.4 APLICAÇÃO DA METODOLOGIA DE PESQUISA

Nesse capítulo serão abordados os assuntos referentes a metodologia de pesquisa, apontado a construção do referencial teórico, as metodologias de desenvolvimento, modelagem das aplicações e como serão realizados os testes.

3.4.1 Construção do Referencial Teórico

No desenvolvimento do referencial teórico deste trabalho diversos materiais foram utilizados como alicerce. Como base de informações foram utilizadas três fontes principais de pesquisa, são elas: Google Scholar, Scielo e biblioteca online da Unoesc e os materiais buscados são: artigos, livros, documentos (acadêmicos e de fornecedores) e sites oficiais.

O primeiro passo para a realização da pesquisa foi levantar algumas palavras chaves que ajudaram a definir e limitar a pesquisa, para esse trabalho as palavras escolhidas foram: medidor

de leite, ordenha, produção leiteira, leite, medidor de volume, sistemas embarcados e sensores, software web e metodologias de pesquisa e modelagem. Além dessas palavras chaves outras pesquisas foram realizadas com o intuito de aprofundar os conhecimentos do autor. A Tabela 1 contém a quantidade de obras utilizadas para cada assunto, é possível observar que para certos assuntos os materiais são escassos, e a quantidade do tipo de material também muda dependendo do assunto.

Tabela 1 - Quantitativo de obras usadas por assunto

Assunto	Tipo	Qtd
Medidor de leite	Livros	0
	Artigos	2
	Sites	1
	Documentos	0
Ordenha	Livros	0
	Artigos	7
	Sites	3
	Documentos	1
Produção leiteira	Livros	0
	Artigos	4
	Sites	1
	Documentos	0
Leite	Livros	0
	Artigos	1
	Sites	2
	Documentos	2
Medidor de volume	Livros	1
	Artigos	3
	Sites	3
	Documentos	2
Sistemas embarcados e sensores	Livros	5
	Artigos	6
	Sites	5
	Documentos	6
Software web	Livros	7
	Artigos	0
	Sites	4
	Documentos	1
Metodologias de pesquisa e modelagem	Livros	4
	Artigos	2
	Sites	7
	Documentos	1

Fonte: O autor

A Tabela 2 apresenta a quantidade de obras por tipo e total que foram utilizadas no decorrer deste trabalho.

Tabela 2 - Quantitativo Total de obras usadas

Tipo	Qtd
Livros	17
Artigos	25
Sites	27
Documentos	13
Total	82

Fonte: O autor

Algumas obras podem ser definidas como fundamentais para o desenvolvimento deste trabalho, são elas:

O livro Sensores Industriais: fundamentos e aplicações de Daniel Tomazini e Pedro Urbano Braga Albuquerque, foi fundamental para conhecer e entender os métodos de medição de nível, utilizados atualmente. Essa obra serviu de base na escolha dos sensores e formatos de como a medição poderia ser realizada.

O material publicado pela organização Instrumentação e controle e escrito por Marcel Roure, denominado “Medição de Nível: tudo para você virar um especialista no assunto. Tudo para você virar um especialista no assunto.” Completou os assuntos abordados no livro de Tomazini e Albuquerque, e trouxe mais clareza na compreensão dos assuntos de medição de nível.

O artigo de Michel A. Wattiuax, chamado de “A MÁQUINA DE ORDENHA. Essenciais em Gado de Leite: Lactação e Ordenha.”. Foi de suma importância para conhecer o processo e funcionamento do sistema de ordenha.

O livro Engenharia de Software de Lan Sommerville, auxiliou no componente de modelagem do software, apresentando os conceitos básicos da modelagem e modelos de desenvolvimento.

Por último uma obra que não foi citada diretamente, mas possuiu grande importância, foi o anuário do leite 2019. O anuário do leite é uma revista publicada anualmente pela Embrapa, dessa revista foram retirados diversos artigos que são citados neste trabalho e esses artigos serviram para atualizar o cenário atual da produção leiteira.

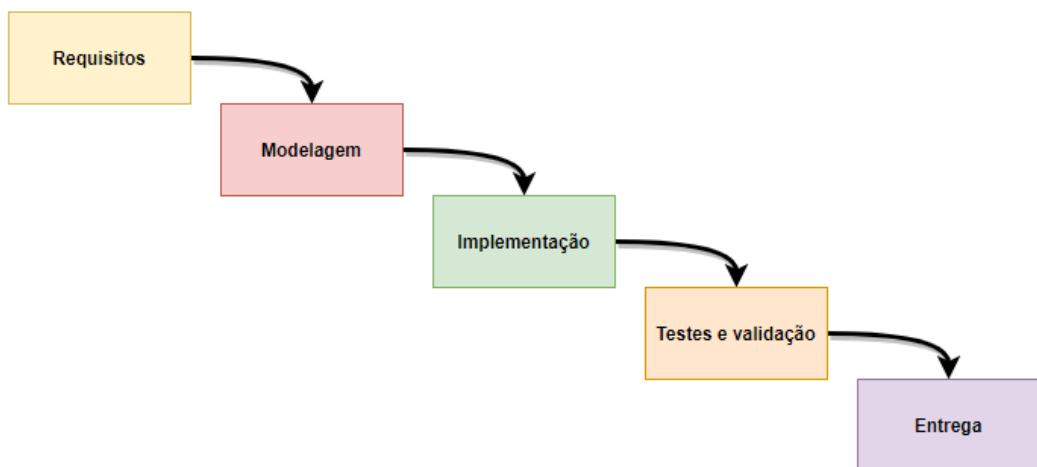
3.4.2 Desenvolvimento do Sistema Computacional

O primeiro passo para desenvolver a metodologia é escolher qual o modelo de software mais indicado para o projeto. Para o desenvolvimento desse trabalho foi escolhido o modelo cascata, esse modelo é o mais antigo da engenharia de software, foi formalizado por Royce no

ano de 1970. É um modelo simples e de fácil entendimento para ambos os envolvidos no projeto, funcionando muito bem em projetos cujos requisitos estejam bem definidos (DEVMEDIA, 2020).

Também conhecido como modelo clássico, o modelo cascata é um modelo de desenvolvimento aonde cada etapa do projeto ocorre de forma sequencial. Dessa forma só após concluir uma etapa é possível avançar para a próxima, fazendo com que o produto seja concluído sem entregas parciais e impossibilitando alterações em etapas já concluídas durante o desenvolvimento (PRESSMAN; MAXIM, 2016, p.42).

Ilustração 25 - Modelo cascata



Fonte: Autor (2020) baseado em Pressman e Maxim (2016)

Para o desenvolvimento deste trabalho foi utilizado o modelo cascata. Na Ilustração 25 é possível observar as etapas que foram realizadas no desenvolvimento o trabalho. Na primeira etapa são especificados os requisitos do projeto, na segunda etapa é realizado modelagem aonde foram desenvolvidos os diagramas e detalhes do projeto, na etapa seguinte foi realizada a implementação da solução e logo após foram realizados testes e a validação. Essa etapa com intuito verificar o funcionamento do produto, para pôr último realizar a entrega do produto completo e funcional.

3.4.2.1 Levantamento de Requisitos

Como mencionado anteriormente, o primeiro passo realizado para iniciar o desenvolvimento do projeto é a listagem dos requisitos, que podem ser divididos entre

requisitos funcionais e requisitos não funcionais. Os requisitos funcionais podem ser definidos como as ações que o sistema deve realizar e que caso essas ações não estejam atuando adequadamente, o funcionamento do sistema estará comprometido (SOMMERVILLE, 2011). O Quadro 1 apresenta os requisitos funcionais desse trabalho.

Quadro 1 - Requisitos funcionais

Requisitos Funcionais	
ID	Descrição
RF1	Medir indiretamente a quantidade de leite
RF2	Mostrar a quantidade de leite medida
RF3	Realizar a medição sem a necessidade de estar conectado a uma rede elétrica
RF4	Possuir uma rede wireless

Fonte: Autor (2020)

Ao relatar a necessidade de o sistema medir de forma indiretamente, faz referência a uma medição sem necessidade de processos manual, uma medição automática. O sistema também deverá informar a quantidade de leite ordenhado com o intuito de acompanhar o processo e por último, por se tratar de uma ordenha móvel, não faz sentido o sistema de medição estar conectado à rede de internet cabeadas e nem a uma rede elétrica. Por esse motivo ele deverá realizar a medição de forma sem fio.

Por outro lado, os requisitos não funcionais são as restrições que as ações do sistema possuem, dessa forma caso alguma dessas restrições não esteja funcionando o sistema não será gravemente comprometido e continuará funcionando (SOMMERVILLE, 2011). Os requisitos não funcionais desse trabalho estão apresentados no Quadro 2.

Quadro 2 - Requisitos não funcionais

Requisitos Não Funcionais	
ID	Descrição
RNF1	Exibir no display o status da medição
RNF2	Realizar a medição de maneira simples e intuitiva
RNF3	Possuir uma interface visualização simples e de fácil entendimento
RNF4	Realizar a medição em menos de 10 segundos

Fonte: Autor (2020)

Para os requisitos não funcionais é de importância o sistema realizar a medição mostrando em tempo real a quantidade que está sendo ordenhado. Na maioria dos casos os agricultores não têm muita familiaridade com tecnologia, por esse motivo o sistema deve ser de fácil entendimento e simples, além de não demorar muito para realizar a medição, dessa forma causando o mínimo impacto no processo da ordenha.

3.4.2.2 Modelagem

Após levantar os requisitos, foi realizada a modelagem do protótipo desenvolvido nesse trabalho. Para isso foram utilizados alguns diagramas pertencentes a UML como o diagrama de classe e o diagrama de caso de uso, além de outros pertencentes a SysML como o diagrama de blocos, de blocos internos, de caso de uso e de atividades. Esses diagramas são utilizados com o objetivo de modelar, organizar e documentar o desenvolvimento do sistema e serão apresentados na sequência da leitura.

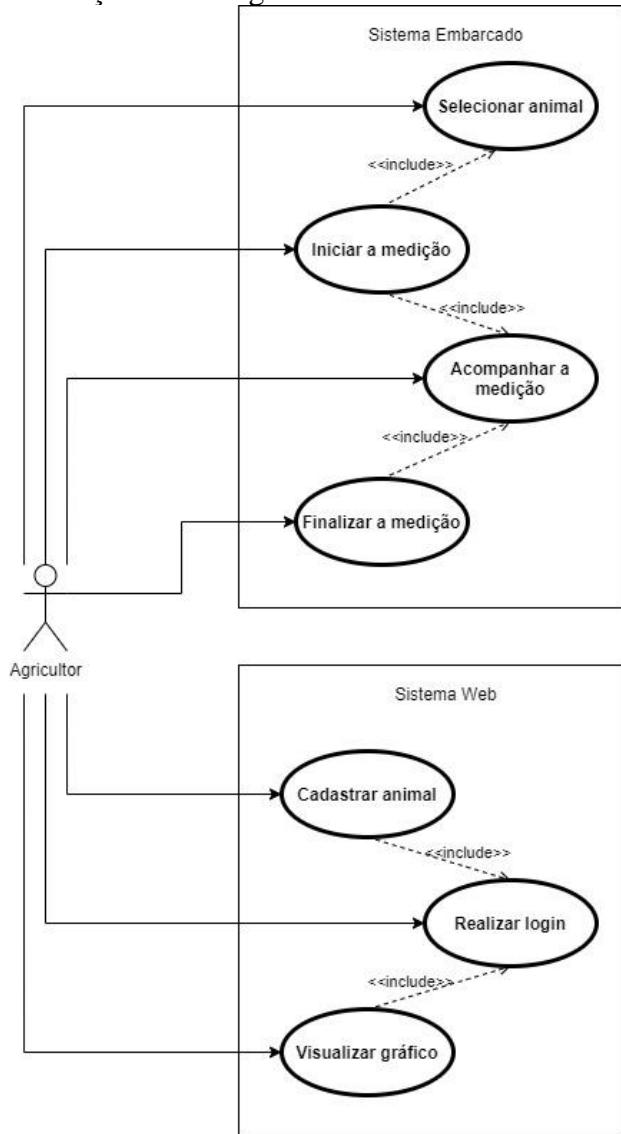
A modelagem foi dividida em duas etapas, na primeira delas modelado o software e a segunda constituirá na modelagem do hardware.

3.4.2.2.1 Software

A Linguagem de Modelagem Unificada mais conhecida como UML teve sua primeira versão desenvolvida por Grady Booch, Jim Rumbaugh e Ivair Jacobson no ano de 1997. A UML surgiu da junção e aperfeiçoamento de métodos de modelagem de software utilizados na época e seu objetivo é servir como uma linguagem para modelar e/ou documentar um projeto de software (PRESSMAN; MAXIM, 2016, p.869).

O diagrama de caso de uso tem a finalidade de mostrar como aconteceram as transações do sistema e auxiliar no levantamento de requisitos. Nesse diagrama são representados os atores que podem ser divididos entre atores internos e externos, um ator pode representar um software, um hardware ou uma pessoa que irá interagir com o sistema. O principal objetivo é descrever as funcionalidades do sistema mostrando quem são os atores participantes do processo (SYSML.ORG, 2020b).

Ilustração 26 - Diagrama de caso de uso

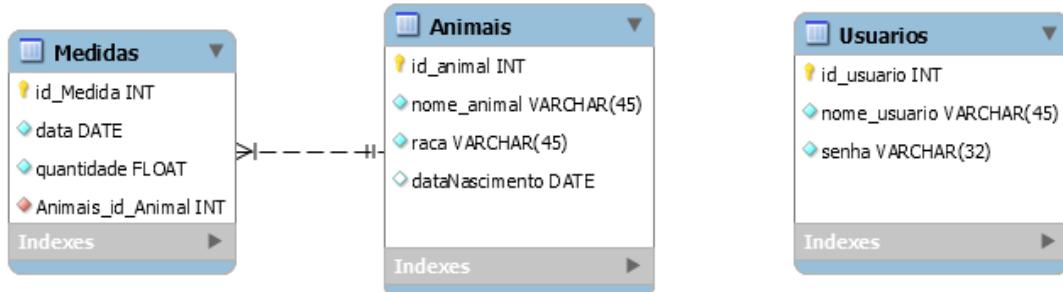


Fonte: Autor (2020)

Diagrama de caso de uso apresentado na Ilustração 26 aponta todas as funções que o sistema possui e as interações que o usuário irá realizar. Os <<include>> que estão no diagrama representam que para realizar tal operação é necessário que a operação a que o include aponta também seja executada, por exemplo, para iniciar a medição o agricultor necessita selecionar o animal e verificar o nivelamento.

Diagrama de entidade relacionamento (DER) ou também conhecido como diagrama ER, é um modelo de diagrama utilizado em sistema que não possuem orientação a Objeto e foi desenvolvido na década de 1970 por Peter Chen. É um diagrama utilizado para projetar banco de dados e demonstras os relacionamentos de cada tabela do banco de dados, cada entidade é representada por um retângulo e seus relacionamentos são representados por linhas que ligam as entidades (LUCIDCHART, 2020).

Ilustração 27 - Diagrama ER



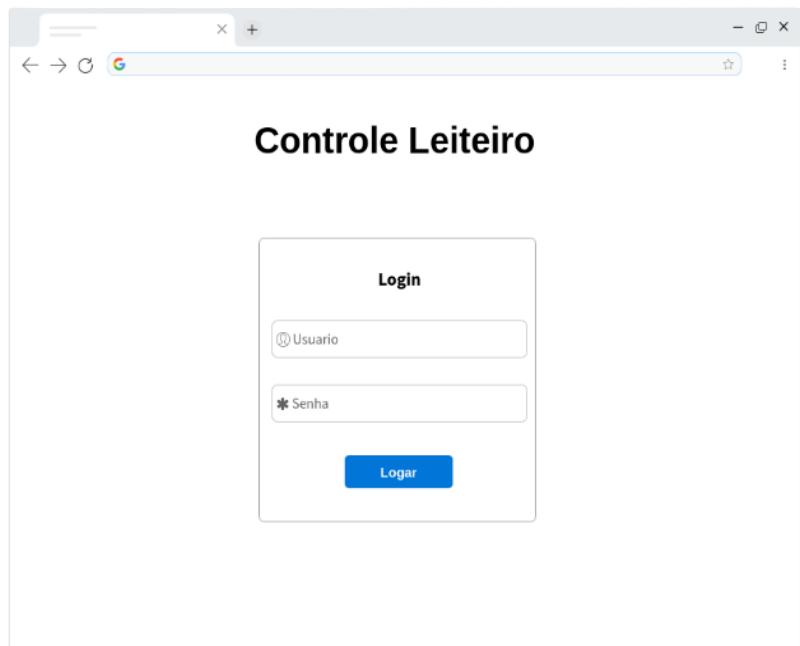
Fonte: Autor (2020)

O diagrama ER apresentado na Ilustração 27 aponta que o banco de dados do sistema possui 3 tabelas, sendo elas: uma tabela isolada de Usuários que é utilizada apenas para armazenar os dados de login e outras duas tabelas de Animais e Medidas com relações entre si. A tabela Medidas armazena as medidas realizadas pelo sistema embarcado e a tabela animais armazena as informações sobre cada animal, dessa forma às duas possuem uma relação em que um animal pode possuir zero ou diversas medidas e uma medida pode pertencer a apenas um animal.

Uma etapa importante na construção do software é a criação dos wireframes, segundo Antunes (2019) “Wireframe é o esboço do esqueleto de uma solução digital”, sendo assim é possível definir wireframe como sendo uma demonstração de como serão as telas e as funções de um determinado sistema. Essa demonstração pode ser elaborada no papel ou utilizando ferramentas digitais e é um recurso que auxilia no desenvolvimento do sistema, melhorando o planejamento e o entendimento do projeto.

O sistema desenvolvido é composto por um total de três telas, aonde cada tela apresenta informações e funcionalidades diferentes, essas telas serão apresentadas na sequência da leitura.

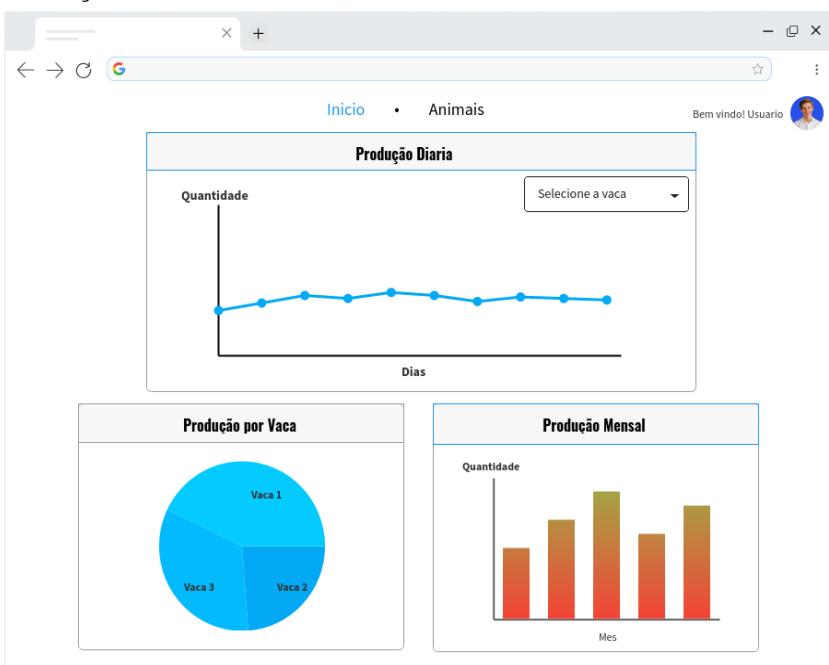
Ilustração 28 - Wireframe tela login



Fonte: Autor (2020)

A primeira tela do sistema representada na Ilustração 28 é a tela de login, nessa tela o agricultor realizará o login que lhe concedera acesso ao sistema.

Ilustração 29 - Wireframe tela inicial



Fonte: Autor (2020)

A segunda tela representada pela Ilustração 29 é a tela de Início do sistema, nessa tela ficaram disponíveis para visualização 3 gráficos, sendo eles: um gráfico da produção diária, um

gráfico da produção por animal e um gráfico da produção mensal. No gráfico da produção diária será demonstrado a quantidade de leite produzido um animal por dia durante um período específico de dias, o animal que será exibido no gráfico pode ser alterado através um campo disponível no canto superior direito. Através de um gráfico no formato pizza será demonstrado a espécie de um ranking da produção por animal durante o mês e por último um gráfico da produção mensal que exibirá a quantidade total de leite produzido por mês durante um período.

Na parte superior está disponível uma barra de navegação, que é utilizada para navegar entre as telas do sistema. Junto a essa barra está sendo exibido as informações do usuário que está conectado, sendo possível através dela realizar o logoff do sistema ou alterar as informações do login.

Ilustração 30 - Wireframe tela animais

Cod	Nome	Raça	Idade	Media produção diaria	Media produção mensal
1	Vaca-1	Jersey	6 anos	12 litros	360 litros
2	Vaca-2	Jersey	8 anos	15 litros	450 litros
3	Vaca-3	Jersey	9 anos	10 litros	300 litros

Fonte: Autor (2020)

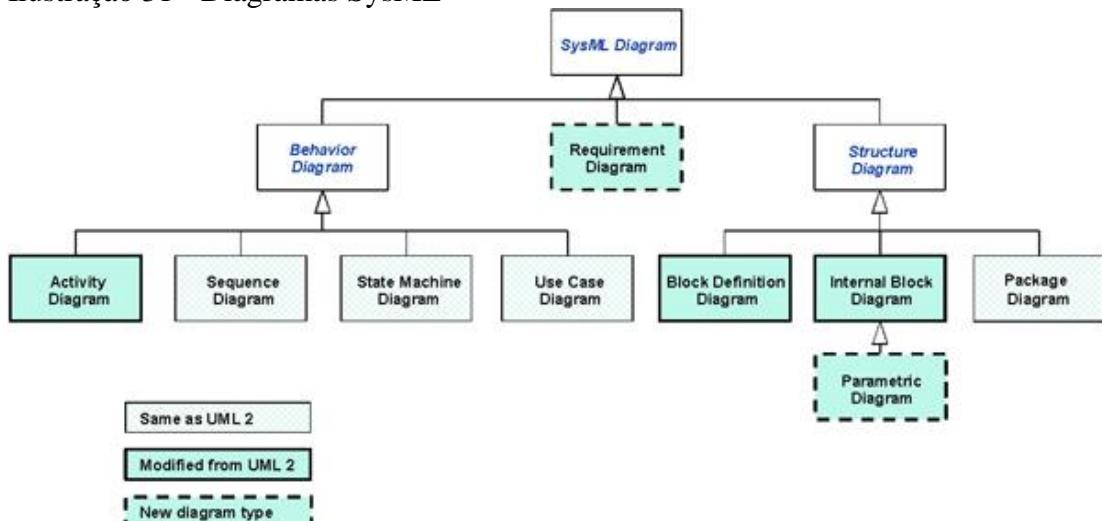
A terceira tela do sistema chamada Animais que está sendo representada na Ilustração 30 é a tela utilizada para cadastrar os animais e verificar informações mais detalhadas sobre cada animal. Essa tela possui um campo para realizar o cadastro de um animal e abaixo uma tabela que disponibiliza para visualização todos os animais cadastrados com suas informações, é possível ainda fazer uma busca de um animal através do campo "Pesquisar Animal".

É importante destacar que todas essas wireframes representam apenas um esboço de como as telas ficarão e os dados utilizados são fictícios. É possível que após o desenvolvimento alguns elementos gráficos serão expostos de maneira diferente ao apresentado nos Wireframes.

3.4.2.2.2 Hardware

SysML é uma linguagem de modelagem gráfica que pode ser considerada uma extensão da linguagem UML, pode ser utilizada para especificar, analisar e projetar sistemas complexos sejam eles de hardware ou software. A OMG (Object Management Group) disponibilizou em setembro de 2007 a primeira versão da OMG SysML, contendo diversos diagramas alguns deles iguais aos da UML, outros modificados e outros novos como é possível observar na Ilustração 31 (OMGSYSML, 2020).

Ilustração 31 - Diagramas SysML

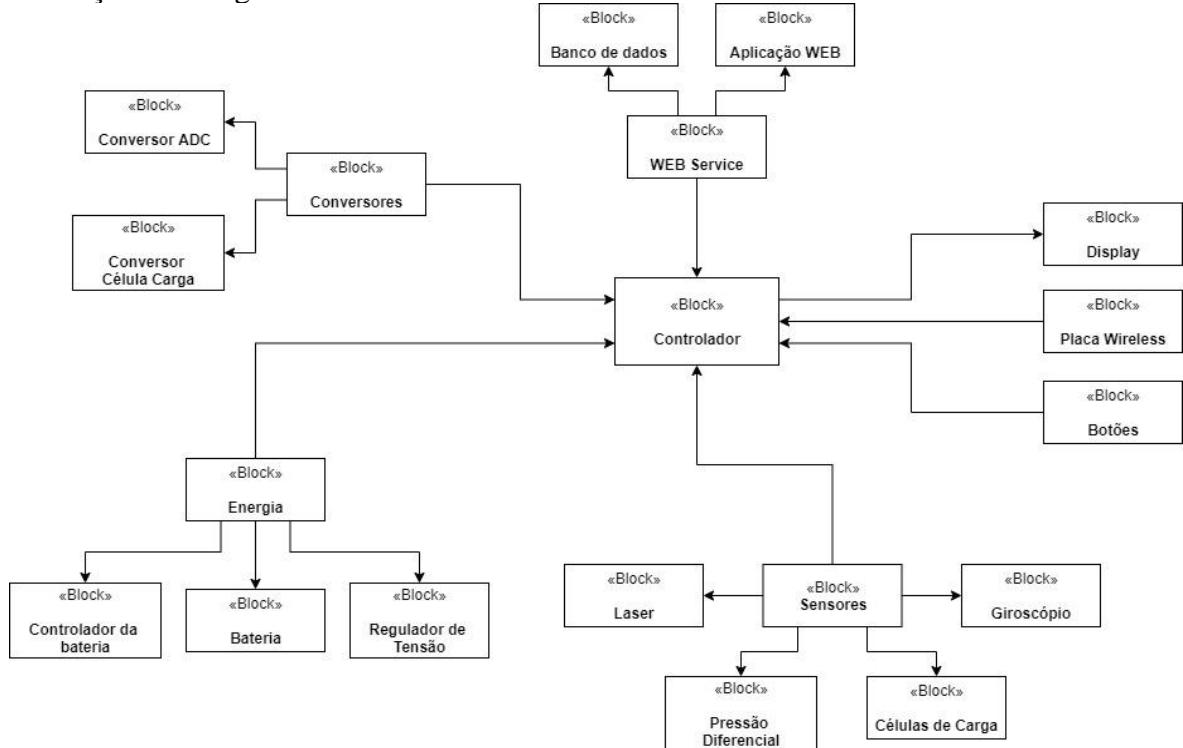


Fonte: Omgsysml (2020)

Apenas dois diagramas são novos, são eles o diagrama paramétrico e o diagrama de requisitos, outros 3 diagramas sofreram alterações são eles: o diagrama de atividades, o diagrama de definição de blocos e o diagrama de bloco interno. Os demais diagramas permanecem iguais aos pertencentes a UML.

Segundo a Sysml.org (2020a) o diagrama de blocos é um diagrama estrutural pertencente à família SysML e tem como objetivo mostrar os componentes de um sistema. Cada bloco pode representar um componente do sistema como: um software, um hardware e até mesmo pessoas envolvidas no processo, além de que cada bloco pode conter informações sobre o componente ao qual ele se refere.

Ilustração 32 - Diagrama de blocos

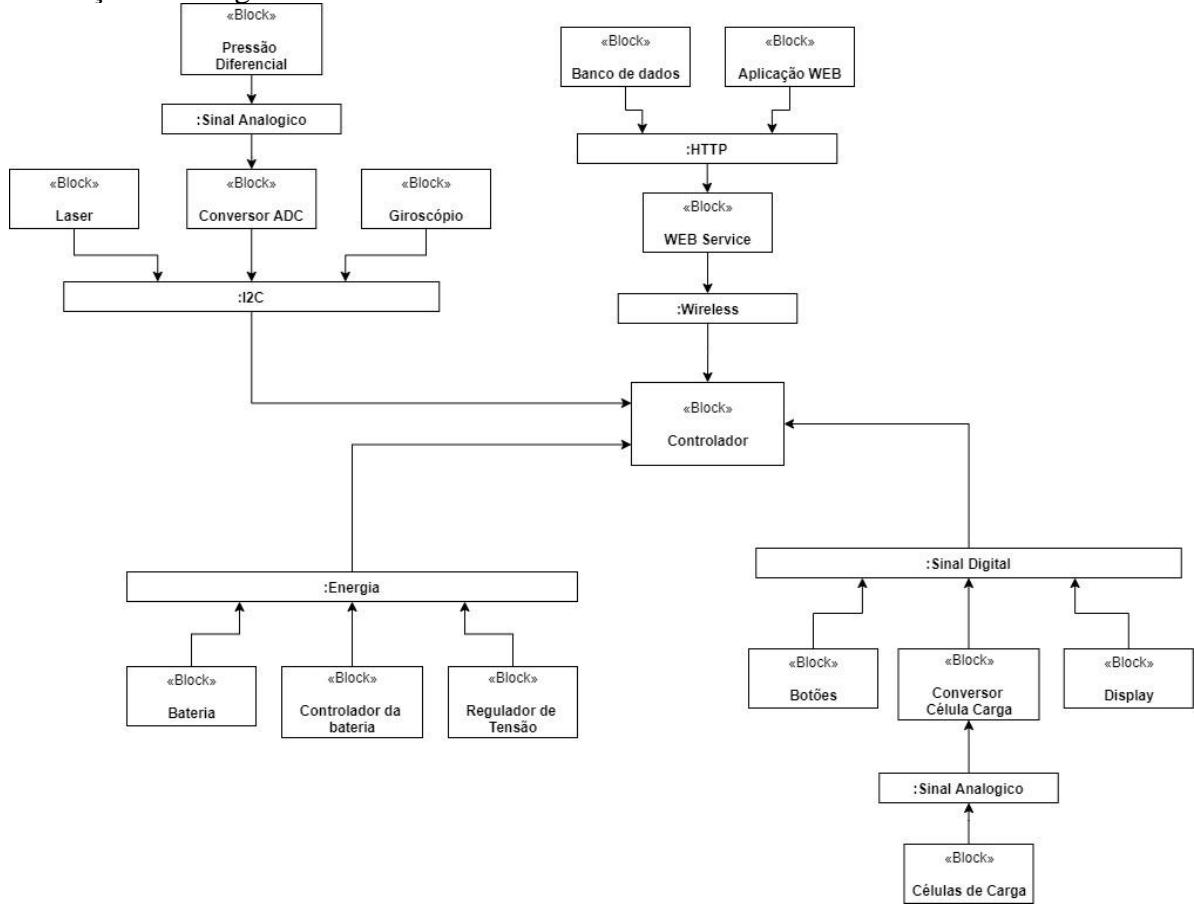


Fonte: Autor (2020)

A Ilustração 32 demonstra o diagrama de blocos deste trabalho. O sistema é composto por diversos blocos, onde, cada bloco representa um conjunto de componentes ou um componente específico do sistema. No centro do diagrama apresenta-se o bloco referente ao controlador, esse bloco realiza comunicação com os outros blocos do sistema, alguns blocos estão conectados diretamente, como, por exemplo: O bloco de Botões, LCD e Placa Wireless. Já outros blocos possuem sub-blocos como é o caso do bloco sensores que está conectado diretamente ao bloco controlador, mas possui sub-blocos representando os sensores específicos, o mesmo acontece para os blocos de energia, web service e conversores. Essa separação foi realizada com o intuito de melhorar a apresentação do diagrama.

O diagrama de bloco interno é uma subdivisão do diagrama de blocos, que apresenta o sistema em um nível mais baixo, ou seja, possuindo um grau maior de detalhes. Esse diagrama demonstra a estrutura interna de um bloco e exibe como se dará o fluxo das informações entre os blocos ou subsistemas (FLAUSINO, 2013).

Ilustração 33 - Diagrama de bloco interno

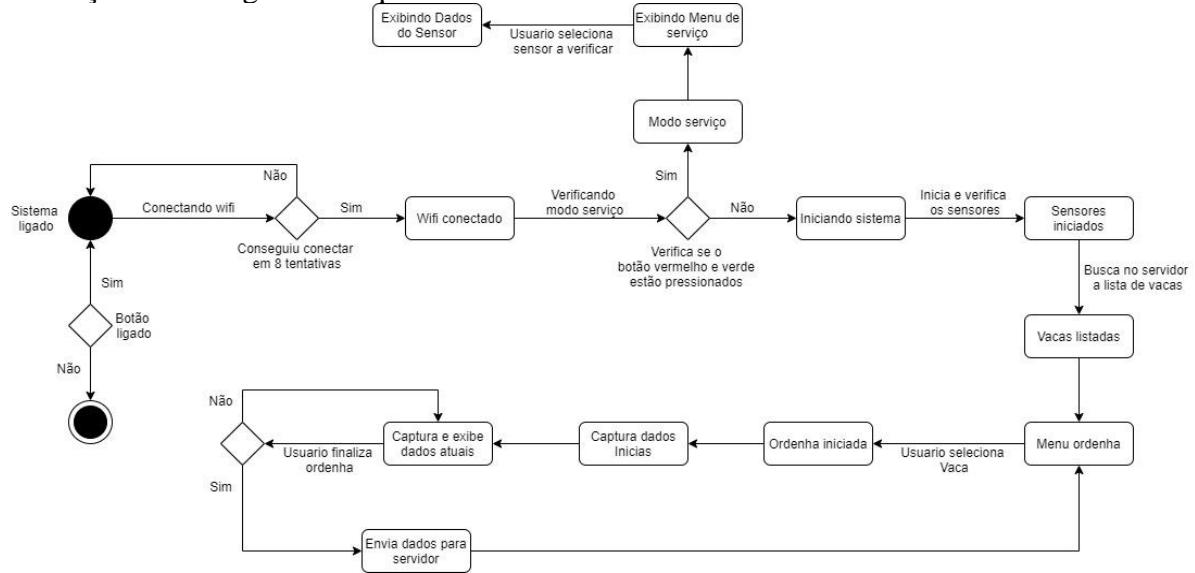


Fonte: Autor (2020)

A Ilustração 33 representa o diagrama de bloco interno, esse diagrama demonstra todas as partes do sistema e apresenta como é realizada a comunicação entre cada uma das partes. Esse sistema é composto por sensores que se comunicam por protocolo Wireless, I2C, sinal Digital, sinal analógico e HTTP.

Com o intuito de explanar o funcionamento do sistema de forma geral, expondo suas funções e estados o diagrama da Ilustração 34 demonstra os passos que o sistema realiza e quais as interações que o usuário deverá fazer para utilizar o sistema.

Ilustração 34 - Diagrama máquina de estados

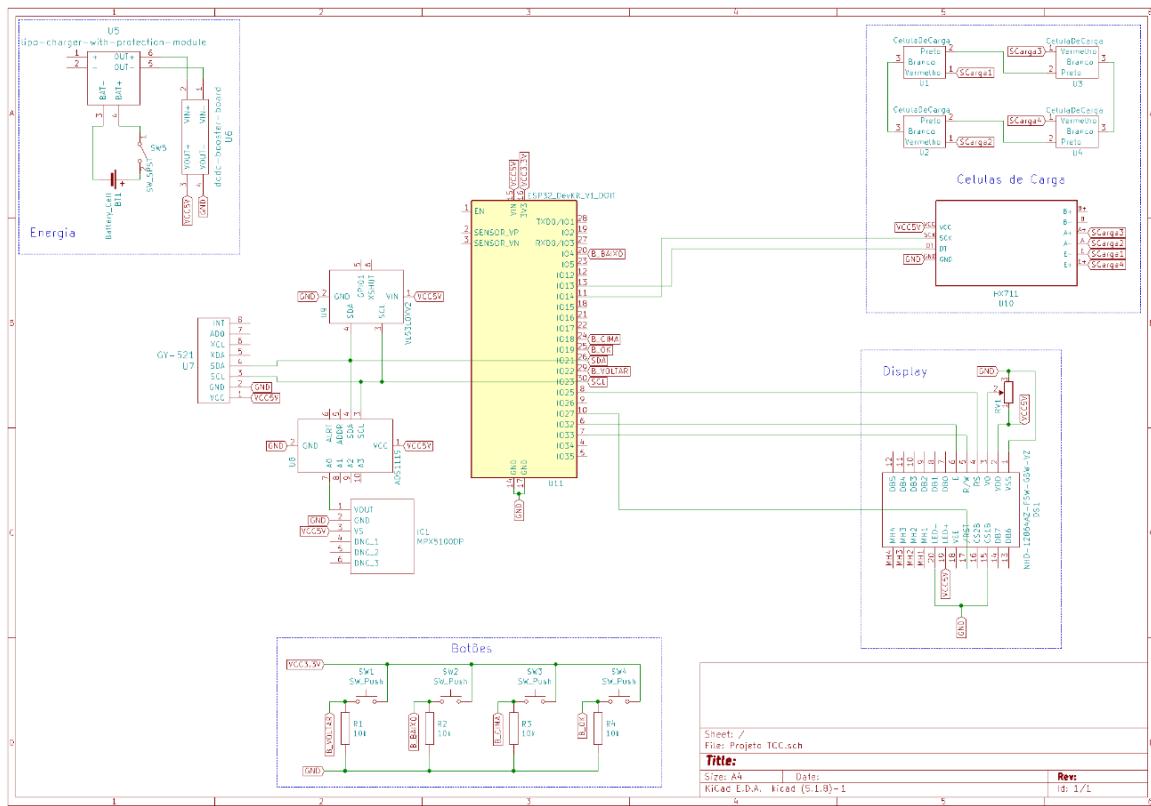


Fonte: Autor (2020)

Diagrama de circuito ou também conhecido como diagrama esquemático é uma representação de todos os componentes presentes no circuito, assim como suas respectivas ligações. Cada componente utilizado possui um símbolo específico que o identifica e todos os componentes podem ser nomeados e/ou numerados, auxiliando na identificação. Todas as ligações são representadas por linhas e as interseções são representadas por pontos (SOUZA, 2016).

O Diagrama elétrico foi desenvolvido na ferramenta Kicad versão 5.1.8. O kicad é uma ferramenta profissional e open source para desenvolvimento de projetos eletrônicos, com essa ferramenta é possível desenvolver diagramas elétricos, projetos de placas de circuitos impressos, lista de matérias e outras funções.

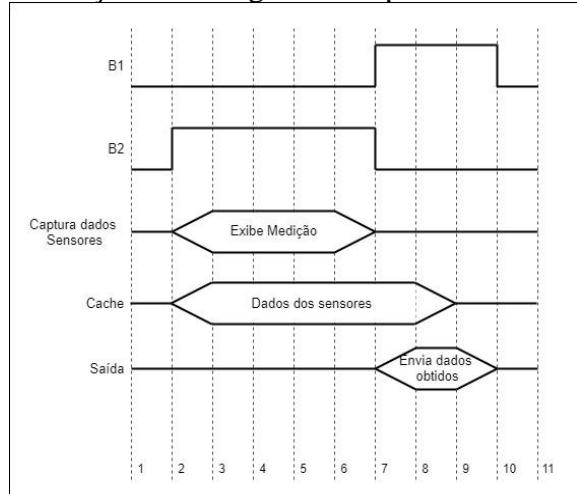
Ilustração 35 - Diagrama elétrico do circuito



Fonte: Autor (2020)

Na Ilustração 35 está sendo representado o diagrama elétrico do circuito, também disponível no “Apêndice C” em alta resolução, exibindo todos os dispositivos utilizados seguindo de suas ligações, etiquetas de identificação e seus respectivos valores. O Diagrama foi dividido em algumas partes, são elas: energia, botões, células de carga e display, na parte central do sistema está o microcontrolador com os demais sensores.

Ilustração 36 - Diagrama temporal



Fonte: Autor (2020)

O digrama apresentado na Ilustração 36 é o diagrama temporal, aonde, B2 representa o estado de início, dessa forma quando B2 está com sinal logico 1 os sensores Giroscópio, laser, sensor de pressão e célula de cargas são ativados para começar a leitura dos dados, junto com os sensores inicia também o cache que armazena os valores dos sensores. O B1 representa o final da ordenha, sendo assim, quando ele é ativo o sinal B2 o todos os sensores são desativados, em contrapartida, é habilitado a função de saída, que recebe os dados disponibilizados no cache e os envia para o servidor. Ao finaliza a função de envio os dados armazenados no cache são deletados e B1 retorna para o estado logico 0 e o sistema fica aguardando uma nova inicialização.

Os materiais necessários para montar a placa estão disponíveis no Quadro 3 juntamente com a quantidade que de cada item que foi utilizado.

Quadro 3 - Lista de materiais

Item	Quantidade
Esp32 38 pinos	1
Sensor de pressão diferencial MPX5100DP	1
Sensor de Distância Laser – VL53L0XV2	1
Giroscópio 3 Eixos MPU-6050 GY-521	1
Display LCD 128x64 ST7920	1
Chave Gangorra ON-OFF	1
Botão Push Botton	4
Potenciômetro 10K	1
Resistor 10K	4
Bateria 18650 Li-Ion Recarregável 3.7V	1
Carregador de Bateria de Lítio TP4056	1
Regulador de tensão MT3608	1
MX711	1
ADS 1115	1
Célula de carga 3 fio 50kg	4

Fonte: Autor (2020)

3.4.3 Participantes do Estudo

No total três pessoas fazem parte do desenvolvimento do referido trabalho são elas:

- Eduardo Kaginski Bevilaqua um universitário do 9º período de Engenharia da computação pela UNOESC-Chapecó, responsável pelo desenvolvimento do trabalho.
- Sr. Jose Antônio Riva um produtor de leite, dono da propriedade onde foram executados os testes práticos.
- Jacson Luiz Matte que foi o professor orientador do projeto, é graduado em ciências da computação pela Universidade Federal da Fronteira Sul, especialista em Tecnologias para aplicações WEB pela Universidade Norte do Paraná e com experiência em sistemas embarcados.

3.4.4 Coleta e Análise dos Dados

Como o projeto é dividido em duas partes, a primeira parte irá consistir em avaliar a eficiência de cada um dos sensores e a segunda do protótipo pronto.

3.4.4.1 Eficiência Individual

Nessa etapa será realizada a medição com valores conhecidos e realizado uma comparação com os valores medidos pelos sensores. A tabela que foi utilizada nessa comparação é a Tabela 3, aonde no campo litros é o valor que foi inserido no reservatório, no campo litros Medida é o valor que o sensor mediu, o campo diferença corresponde o erro entre a quantidade inserida e o valor medido e por último o campo Erro% que contém o erro na medição em porcentagem.

Tabela 3 - Tabela padrão de eficiência

Sensor XX			
litros Medida	litros	Diferença	Erro %
	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		

Fonte: Autor (2021)

A pós concluir todos os testes o sensor com melhor desempenho foi utilizado no protótipo, que seguiu para a segunda etapa de testes.

3.4.4.2 Eficiência do protótipo

Essa etapa também foi dividida em duas partes, a primeira delas com sistema em laboratório foi ser executado um teste parecido com o anterior, porém agora com o sistema montado em uma PCB e com apenas o sensor selecionado. Também foi realizada uma verificação se a integração entre o hardware e o software estava funcionando de maneira correta.

Para o teste de medição foi analisado a margem de erro do sistema em porcentagem, para isso basta comparar o valor medido com a quantidade adicionada executando a seguinte equação: $Erro\ em\ \% = \left(\frac{Medição\ do\ sistema - Litros}{Litros} \right) * 100$.

O ICAR (*International Committee for Animal Recording*) em português comitê internacional para registro de animais é uma organização mundial responsável por validar e certificar os medidores de leite. Ela aponta que um medidor que possui uma taxa de variação menor que 2,5% é considerado um medidor com alta precisão, e são considerados medidores com precisão normal e aceitável aqueles que apresentam taxa de variação entre 5 e 7,5% (WILMINK et al., 1998, apud QUINTÃO, 2011). Isso significa que para uma vaca que produz 10L de leite a margem de erro pode ser de no máximo 750ML, dessa forma a medição pode ser de 9,25L até 10,75L.

Quando o protótipo do sistema embarcado e a aplicação web estiverem prontos e testados em laboratório, serão realizados os testes e validação do sistema junto ao produtor de leite. Para isso o sistema foi coloca em funcionamento na propriedade e utilizado por dois dias seguidos em duas vacas. Essa validação de caráter qualitativo foi composta por perguntas no formato escala Likert, tendo como objetivo avaliar o protótipo perante o seu usuário.

A escala Likert foi criada pelo norte-americano Rensis Likert no ano de 1932, um questionário simples com as possibilidades de resposta “sim” ou “não”, é possível saber apenas se a pessoa entrevistada concorda ou não concorda com algo. Já a principal característica da escala Likert é avaliar de uma forma mais específica, identificando o quanto ele concorda ou discorda de algo (SCHERMANN, 2019).

Nesse modelo ao invés de realizar perguntas, são apresentadas afirmações e as respostas esperadas são concordo ou não concordo com a afirmação. Cada opção de resposta possui uma pontuação sendo a mais comum de 1 a 5, isso significa que possuem 5 opções de resposta

(FRANKENTHAL, 2020). A escala Likert utilizada no questionário deste trabalho está apresentada no Quadro 4, nela é possível observar a pontuação e a descrição de cada alternativa.

Quadro 4 - Tabela Likert

Discordo Totalmente	Discordo	Neutro	Concordo	Concordo Totalmente
1	2	3	4	5

Fonte: Autor (2020)

Tento a tabela com as alternativas o questionário que foi aplicado para avaliação deste projeto está disponível para visualização no “Apêndice B”.

Após o questionário ser totalmente respondido pelo agricultor, foi realizada as análises das respostas obtidas. É possível avaliar as alternativas individualmente, ou seja, ver quantas foram “Concordo totalmente”, quantas foram “Discordo” e assim por diante. Como pode ser possível avaliar quantos concordam e quantos não concordam para isso é necessário somar “Concordo totalmente” com “Concordo” e assim obter o total de aceitação, e fazer o mesmo para o Discordo para obter o total de discordância.

4 APRESENTAÇÃO DA SOLUÇÃO E ANÁLISE DOS RESULTADOS

A construção do protótipo foi dividida em 4 grandes partes, sendo, a primeira parte, software, a segunda hardware, a terceira funcionalidades e a quarta testes e análise dos resultados.

4.1 SOFTWARE

Nesse tópico serão apresentadas as etapas realizadas para construir o software, que consiste em uma aplicação Web desenvolvida com o framework Codeigniter 4. Inicialmente foi necessário realizar a instalação e configurações de algumas fermentas fundamentais para o desenvolvimento, são elas: XAMP um pacote de fermentas que contém Apache, Mysql e phpMyAdmin, Visual Studio Code utilizado como editor de código-fonte, POSTMAN que é uma ferramenta utilizada para testar APIs e mysql workbench uma ferramenta de criação de banco de dados.

4.1.1 Banco de dados

Com a ferramenta mysql workbench versão 8.0, foi possível transformar o diagrama ER apresentado no tópico 3.4.2.1.1 em um código SQL. Com o código gerado foi criado o banco no phpmyadmin e executado o código SQL, com isso todas as tabelas foram criadas e o banco ficou pronto para ser utilizado. O Quadro 5 exibe o código SLQ gerado pelo software.

Quadro 5 - Código SQL banco de dados

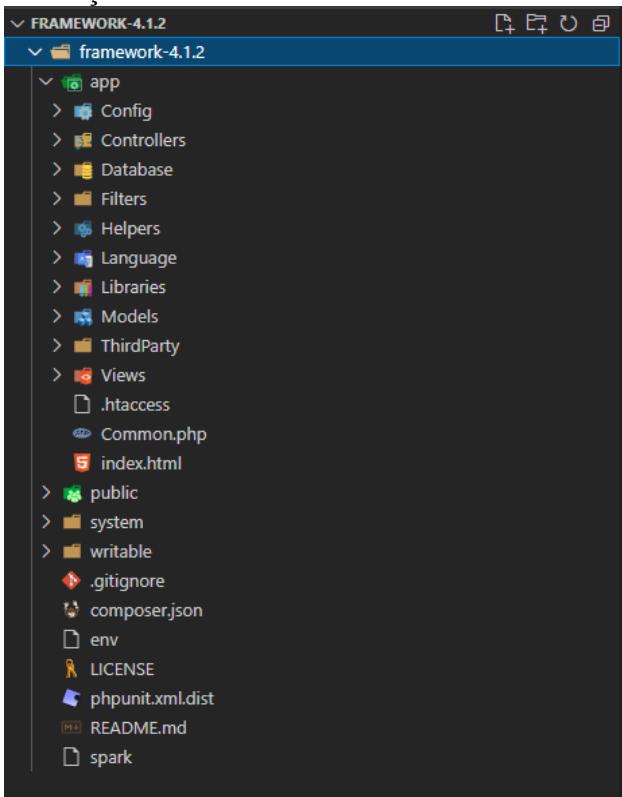
-- Table `bancoTCC`.`Usuarios` ----- CREATE TABLE IF NOT EXISTS `bancoTCC`.`Usuarios` (`id_usuario` INT NOT NULL AUTO_INCREMENT, `nome_usuario` VARCHAR(45) NOT NULL, `senha` VARCHAR(32) NOT NULL, PRIMARY KEY (`id_usuario`), UNIQUE INDEX `id_usuario_UNIQUE` (`id_usuario` ASC), UNIQUE INDEX `nome_usuario_UNIQUE` (`nome_usuario` ASC)) ENGINE = InnoDB; ----- -- Table `bancoTCC`.`Animais` ----- CREATE TABLE IF NOT EXISTS `bancoTCC`.`Animais` (`id_animal` INT NOT NULL AUTO_INCREMENT, `nome_animal` VARCHAR(45) NOT NULL, `raca` VARCHAR(45) NOT NULL, `dataNascimento` DATE NULL, PRIMARY KEY (`id_animal`)) ENGINE = InnoDB;	-- Table `bancoTCC`.`Medidas` ----- CREATE TABLE IF NOT EXISTS `bancoTCC`.`Medidas` (`id_Medida` INT NOT NULL AUTO_INCREMENT, `data` DATE NOT NULL, `quantidade` FLOAT NOT NULL, `Animais_id_Animal` INT NOT NULL, PRIMARY KEY (`id_Medida`), INDEX `fk_Medidas_Animais_idx` (`Animais_id_Animal` ASC), CONSTRAINT `fk_Medidas_Animais` FOREIGN KEY (`Animais_id_Animal`) REFERENCES `bancoTCC`.`Animais` (`id_animal`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;
--	--

Fonte: Autor (2020)

4.1.2 Web

A primeira etapa ao trabalhar com codeigniter consiste em realizar o download do modelo padrão, esse modelo está disponível no site oficial e pode ser baixado gratuitamente. Ao baixar o arquivo ele veio compactado e ao descompactar o arquivo foi obtida a estrutura de pastas da ilustração 37.

Ilustração 37 - Estrutura Framework



Fonte: Autor (2020)

As pastas system e writable são pastas do sistema e nada precisa ser feito com elas, a pasta public contém os arquivos públicos ela pode ser utilizada para adicionar códigos CSS ou arquivos de outros frameworks, já na pasta app contém todos os arquivos do projeto e todos os códigos estarão nessa pasta. Nos tópicos abaixo serão apresentados todos os códigos e o funcionamento do framework.

4.1.2.1 Configurações iniciais

Inicialmente foi necessário descompactar os arquivos para a pasta htdocs do Xamp que contêm os projetos. Na raiz dos arquivos do framework contém um arquivo chamado env esse

arquivo foi renomeado para .env e adicionado em seu conteúdo o texto “CI_ENVIRONMENT = development”. Isso é o modo de operação do framework e ao informar development ele ficou modo desenvolvimento, com isso ele retorna os erros de código e outras informações.

Após isso no arquivo APP/Config/App.php deve se alterar a variável \$baseURL, o seu valor é o caminho da pasta que contém seus arquivos do projeto. Exemplo \$baseURL = 'http://localhost/AplicacaoTCC/'.

Com o banco de dados já criado e pronto para uso, foi necessitando realizar a configuração no Codeigniter. Para isso basta acessar o arquivo APP/Config/Database.php e editar as variáveis: hostname, username, password e database conforme as respectivas informações do seu banco de dados. O Quadro 6 evidencia a estrutura configurada.

Quadro 6 - Código configuração banco de dados

```
public $default = [
    'DSN'        => '',
    'hostname'   => 'localhost', //Endereço servidor banco de dados
    'username'   => 'root', // Usuario do banco de dados
    'password'   => '', // Senha do banco de dados
    'database'   => 'bancotcc', // Nome do banco de dados
    'DBDriver'   => 'MySQLi',
    'DBPrefix'   => '',
    'pConnect'   => false,
    'DBDebug'   => (ENVIRONMENT !== 'production'),
    'cacheOn'   => false,
    'cacheDir'   => '',
    'charset'   => 'utf8',
    'DBCollat'   => 'utf8_general_ci',
    'swapPre'   => '',
    'encrypt'   => false,
    'compress'   => false,
    'strictOn'   => false,
    'failover'   => [],
    'port'       => 3306,
];
```

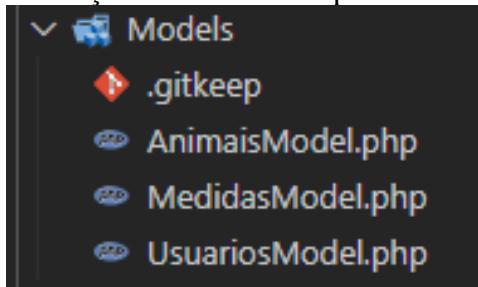
Fonte: Autor (2020)

Após todos os passos anteriores serem completados o banco de dados e o codeigniter já estão configurados e pronto para o uso.

4.1.2.2 Model

Na pasta Model contém os arquivos de acesso ao banco de dados, para cada tabela do banco foi criado um arquivo nometabelaModel.php, como apresenta a Ilustração 38.

Ilustração 38 - Estrutura pasta Models



Fonte: Autor (2020)

Cada arquivo contém a mesma estrutura exibida no Quadro 7 e o que irá mudar de um arquivo para o outro é os dados da tabela. A classe é composta por 5 atributos, e cada atributo refere-se a uma informação da tabela, o Quadro 7 demonstra o código para a tabela animais e como foram realizadas as configurações.

Quadro 7 - Código AnimaisModel

```
<?php namespace App\Models;
use CodeIgniter\Model;
class AnimaisModel extends Model {
    protected $table = 'Animais'; // Nome da tabela
    protected $primaryKey = 'id_animal'; // Chave primária da tabela
    protected $allowedFields = ['nome_animal', 'raca', 'dataNascimento']; // demais campos da tabela
    protected $returnType = 'array'; // Formato de retorno de dados, pode ser Array ou object
    protected $afterFind = ['protecaoXSS']; // chama a função protecaoXSS após realizar uma busca
    protected function protecaoXSS($data){
        $data = esc($data);
        return $data;
    }
?>
```

Fonte: Autor (2020)

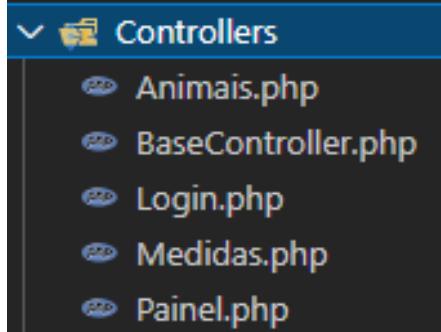
Além dos atributos também contém uma função, denominada protecaoXSS que como o nome já sugere é responsável por realizar proteção contra-ataques XSS. Essa função é chamada pelo atributo \$afterFind, que é responsável por passar os dados pelo filtro sempre após uma busca no banco de dados.

4.1.2.3 Controllers

Na pasta Controllers ficam os arquivos de controle e lógica do sistema, ou seja, a parte back-end da aplicação. Para cada Model foi criado um Controller com as funções de CRUD e funções adicionais listadas no tópico 3.4.2. Além de dois Controllers adicionais, o Login responsável por efetuar a autenticação do Login e cadastro dos usuários e o Painel que processa

os dados do dashboard. Mais um arquivo padrão chamado BaseController que é criado pelo framework. A Ilustração 39 apresenta o diretório Controllers.

Ilustração 39 - Estrutura Controllers



Fonte: Autor (2020)

O Controller Animais.php como já citado anteriormente é responsável por realizar o CRUD dos animais, além de uma função demostrada no Quadro 8 que é responsável por enviar a lista de animais para o hardware.

Quadro 8 – Código php da função listarAnimais()

```
public function listaAnimais(){
    if($this->request->getMethod() === 'post'){ // Verifica se ocorreu uma requisição POST
        $dados = $this->request->getPost(); // Recebe os dados POST
        if($dados['chave'] == "ABC123"){ // Verifica se a chave esta correta
            $arrayDados = array(); // Declara o array com os dados
            $animaisModel = new \App\Models\AnimaisModel(); // Variável de acesso ao banco
            $Animais = $animaisModel->find(); // Realiza a Busca no banco de dados
            $i=1;
            foreach($Animais as $animais){ // Realiza a varredura dos dados retornados pela pesquisa
                $arrayDados['vaca' . $i] = $animais['nome_animal']; // Monta o array dos dados
                $i++;
            }
            if(isset($arrayDados)){
                $myJSON = json_encode($arrayDados); // Converte o array em uma arquivo Json
                echo $myJSON; // Exibe o Arquivo Json
            }
        } else{
            echo "Chave Incorreta";
        }
    }
}
```

Fonte: Autor (2020)

Resumidamente essa função busca no banco de dados os animais cadastrados, monta um array com um identificador e nome do animal, converte para o formato Json e por último retorna o resultado para quem o solicitou. Na aplicação em questão o resultado é retornado para sistema embarcado que irá receber, manipular e utilizar esses dados.

Quadro 9 - Código php para inserir medida

```
public function inserir(){
    if($this->request->getMethod() === 'post'){
        $dados = $this->request->getPost();
        if($dados['chave'] == "ABC123"){
            $animaisModel = new \App\Models\AnimaisModel();
            $animal = $animaisModel->where('nome_animal',$dados['nome_animal'])->find();
            if(count($animal) != 0){
                foreach($animal as $animal){
                    $animal = $animal;
                }
                $medidasModel = new \App\Models\MedidasModel();
                $medidasModel->set('data',date('Y-m-d H:i:s'));
                $medidasModel->set('quantidade',$dados['quantidade']);
                $medidasModel->set('Animais_id_Animal',$animal['id_animal']);
                if($medidasModel->insert()){
                    echo "Medida registrada com sucesso";
                }
                else{
                    echo "Erro ao registrar medida";
                }
            }
            else{
                echo "Animal não cadastrado";
            }
        }
        else{
            echo "Chave incorreta";
        }
    }
}
```

Fonte: Autor (2020)

A função inserir() que está disponível no Quadro 9 é uma função do Controller Medidas.php. Essa é a função do sistema que tem o objetivo de receber as medidas capturada na ordenha, após receber a função valida se a chave está correta, verifica se o animal existe no banco de dados, adiciona a data da ordenha e insere os dados na tabela “Medidas” do banco de dados.

Outras três funções que estão no Controller Medidas.php são as funções obterQuantidadeMensal(),obterQuantidadeDiaria() e obterQuantidadeAnimal(). Essas tem a responsabilidade de buscar os dados no banco de dados, tratar e retornar os dados para o dashboard.

No Quadro 10 a função obterQuantidadeMensal(). Para capturar a leitura mensal a função possui um for de 1 a 12, que fazem referências aos meses do ano. Para cada mês é realizado uma consulta no banco passando como parâmetro de busca as datas limites do mês. Exemplo: para o mês janeiro é realizado no banco uma consulta dos dias 01/01 a 31/01. Após buscar todas as leituras do mês o resultado é armazenado em um array e a busca parte para o próximo mês.

Quadro 10 - Código php função obterQuantidadeMensal()

```
public function obterQuantidadeMensal(){
    if(isset($_SESSION['Logado'])){
        $medidasModel = new \App\Models\MedidasModel();
        $arrayDados = array();
        for($i = 1; $i <= 12; $i++){
            $cont = 0;
            $medidas = $medidasModel->where('data >=', "2021-$i-01")
                ->where('data <=', "2021-$i-31")->find();
            foreach($medidas as $valor){
                $cont = ($valor['quantidade']) + $cont;
            }
            $array = (object) array("medida" => $cont);
            array_push($arrayDados, $array);
        }
        $myJSON = json_encode($arrayDados);
        echo $myJSON;
    }
}
```

Fonte: Autor (2020)

A função obterQuantidadeDiaria(), disponível no Quadro 11 busca os dados da produção diária de um animal, para isso ela utiliza as funções strtotime() de manipulação de datas. A estrutura de pesquisa é como da função anterior, porém agora a busca é diária logo o for vai de 0 a 30 e com isso são pesquisados dados dos últimos 30 dias.

Outra diferença é que nessa função captura os dados de todos os animais ou apenas do animal selecionado no painel. Para isso quando o animal é selecionado no gráfico o valor da SESSION[‘AnimalPainel’] fica igual ao valor da ID do animal selecionado e na busca possuiu um filtro que captura os valores apenas para o animal que a SESSION aponta.

Quadro 11 - Código php função obterQuantidadeDiaria

```
public function obterQuantidadeDiaria(){
    if(isset($_SESSION['Logado'])){
        $medidasModel = new \App\Models\MedidasModel();
        $arrayDados = array();
        for($i = 0; $i <= 30; $i++){
            $cont = 0;
            $data = strtotime(date('Y-m-d'));
            $temp = strtotime("-$i day", $data);
            $dataLimite = date('Y-m-d', $temp);
            if($_SESSION['AnimalPainel'] == "Todas"){
                $medidas = $medidasModel->where('data', $dataLimite)->find();
            } else{
                $medidas = $medidasModel->where('data', $dataLimite)
                    ->where('Animais_id_Animal', $_SESSION['AnimalPainel'])->find();
            }
            foreach($medidas as $valor){
                $cont = ($valor['quantidade']) + $cont;
            }
            $array = (object) array("data" => (date('d/m', $temp)), "medida" => $cont);
            array_push($arrayDados, $array);
        }
        $myJSON = json_encode($arrayDados);
        echo $myJSON;
    }
}
```

Fonte: Autor (2020)

Por último a função obterQuantidadeAnimal(), Quadro 12, essa função captura a quantidade total que um animal produziu, essa função percorre todos os animais do banco obtendo o total das medições registradas. A soma de todas as medições, é igual à produção total do animal, o resultado de cada animal é armazenado no array com o nome do animal, para ser posteriormente utilizado no dashboard.

Quadro 12 - Código php função obterQuantidadeAnimal

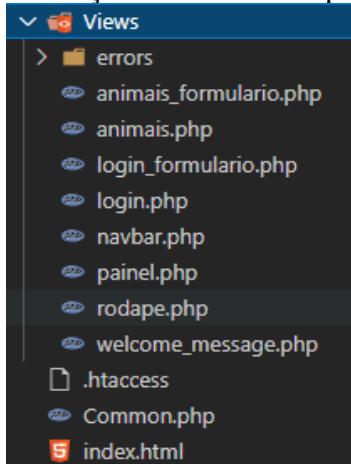
```
public function obterQuantidadeAnimal(){
    if(isset($_SESSION['Logado'])){
        $medidasModel = new \App\Models\MedidasModel();
        $animaisModel = new \App\Models\AnimaisModel();
        $arrayDados = array();
        $dados = $animaisModel->find();
        foreach($dados as $valor){
            $cont = 0;
            $medidas = $medidasModel->where('Animais_id_Animal','=',$valor['id_animal'])
                ->find();
            foreach($medidas as $medidas){
                $cont = ($medidas['quantidade']) + $cont;
            }
            $array = (object) array("nome_animal" =>
                $valor['nome_animal'], "quantidadeTotal" => $cont);
            array_push($arrayDados, $array);
        }
        $myJSON = json_encode($arrayDados);
        echo $myJSON;
    }
}
```

Fonte: Autor (2020)

4.1.2.4 Viwes

As Viwes são os arquivos de Visualização, a parte front-end da aplicação, ou seja, nesses arquivos contém os códigos HTML e Javascript responsáveis pelas telas do sistema. Como é possível observar na Ilustração 40 o sistema em questão possui 8 arquivos de visualização, sendo um arquivo welcome_mansage.php que se trata de um arquivo padrão do framwork, os arquivos rodape.php e navbar.php que correspondem a partes das telas respectivamente ao rodapé da aplicação e a barra de navegação. E os demais arquivos que corresponde as telas do sistema, responsáveis pela tela de login, cadastro dos usuários, animais, cadastro dos animais e painel.

Ilustração 40 - Estrutura pasta Views



Fonte: Autor (2020)

Dentre as telas uma das mais importantes é ao dashboard(painel), essa tela possui as funções Ajax e da biblioteca Charts.io. Que são responsáveis respectivamente por buscar os dados dos controllers e criar os gráficos. No total são 3 funções Ajax e três funções de geração de gráfico.

Quadro 13 - Código Ajax

```
$.ajax({
    type: "POST", // Tipo de Requisição
    url: 'http://localhost/AplicacaoTCC/public/Medidas/obterQuantidadeDiaria',//Endereço da Função
    timeout: 3000,
    datatype: 'JSON',
    contentType: "application/json; charset=utf-8",
    cache: false,
    success: function(myJSON) {
        var nomeArray = [];
        var qtdArray = [];
        data = JSON.parse(myJSON); // Recebe os dados
        for(var i=0; data.length>i;i++){ // Realiza a varredura dos dados recebidos
            nomeArray.push(data[i].data); //Monta o array
            qtdArray.push(data[i].medida); //Monta o array
        }
        nomeArray.reverse(); // Inverte a ordem do array
        qtdArray.reverse(); // Inverte a ordem do array
        graficoDiario(nomeArray, qtdArray); // Chama a função que gera os graficos
    },
});
```

Fonte: Autor (2020)

O Quadro 13 exibe a uma função Ajax, essa função visa obter os dados da produção diária. Para isso ela chama a função do back-end mencionada anteriormente que busca esses dados, cria um array com a data e a medida, aplica o reverse() que interver os valores do array e por final chama a função graficoDiario que gera o gráfico para visualização. As demais funções Ajax seguem o mesmo escopo do modelo demostrado acima, alterando a URL do endereço da função e o nome dos atributos retornados na variável data.

Como já citado anteriormente os gráficos também foram gerados em Javascript e com o auxílio da biblioteca Chart.io. Para cada gráfico existe uma função que recebe os dados do Ajax e gera os gráficos, o Quadro 14 mostra os códigos e as funções. No total são 3 funções sendo uma para o gráfico diário, uma para o gráfico mensal e outra para o gráfico total por animal. Todas as funções possuem o mesmo escopo e o que irá mudar de uma para a outra são os valores das labels e o type que representa o tipo de gráfico. Para a produção diária foi utilizado um gráfico do tipo line, para produção mensal do tipo bar e para produção por animal do tipo doughnut.

Quadro 14 - Código que gera os gráficos

```

function graficoDiario(nome, qtd){
    var ctx = document.getElementById('producaoDiaria').getContext('2d');
    var chart = new Chart(ctx, {
        type: 'line', // Tipo do Grafico
        data: {
            labels: nome, // Nome de cada atributo do Grafico
            datasets: [
                {
                    label: 'Quantidade em Litros', // Nome do valor
                    backgroundColor: 'rgba(0, 0, 0, 0.0)', // Cor do grafico
                    borderColor: 'rgb(0, 100, 200)', // Cor da borda
                    data: qtd // Valores
                }
            ],
            options: {}
        });
}
function graficoMensal(qtd){
    var ctx = document.getElementById('producaoMensal').getContext('2d');
    var chart = new Chart(ctx, {
        type: 'bar',
        data: {
            labels: ['Janeiro', 'Fevereiro', 'Março', 'Abril', 'maio', 'Junho', 'Julho',
                     'Agosto','Setembro','Outubro', 'Novembro', 'Dezembro'],
            datasets: [
                {
                    label: 'Quantidade em Litros',
                    backgroundColor: 'rgb(0, 150, 255)',
                    borderColor: 'rgb(0, 150, 255)',
                    data: qtd
                }
            ],
            options: {}
        });
}
function graficoAnimal(nome, qtd){
    var ctx = document.getElementById('producaoPorVaca').getContext('2d');
    var chart = new Chart(ctx, {
        type: 'doughnut',
        data: {
            labels: nome,
            datasets: [
                {
                    label: 'Quantidade em Litros',
                    backgroundColor: ['rgb(0, 150, 255)', 'rgb(255, 50, 0)', 'rgb(150, 255, 0)'],
                    borderColor: 'rgb(255, 255, 255)',
                    data: qtd
                }
            ],
            options: {}
        });
}

```

Fonte: Autor (2020)

Para mostrar o gráfico no HTML basta chamar a função javascript na tag canvas, ex: <canvas id="producaoPorVaca"></canvas>, e o gráfico já sera exibido. Na aplicação em questão cada gráfico foi colocado em um card e no card da produção diária existe a função para mudar de vaca. Dessa foram é possível personalizar o conteúdo do primeiro gráfico podendo selecionar entre as vacas ou deixando em todas.

A Ilustração 41 demostra o resultado da tela do dashbord, com os 3 gráficos, um navbar para navegar entre as telas e a opção para mudar de animal no gráfico de produção diária.

Os dados utilizados nessa imagem são apenas para demonstração.

Ilustração 41 - Resultado dashboard finalizado



Fonte: Autor (2020)

Outra tela importante do sistema é a tela de animais, essa tela proporciona a lista dos animais cadastrados e suas informações básicas, além de informações adicionais como quantidade média por ordenha e quantidade total ordenhada no animal. Para cada um possui a possibilidade de deletá-lo ou editar as informações que são cadastráveis, além da possibilidade de adicionar outro animal pressionando no ícone +. A Ilustração 42 apresenta a tela comentada.

Ilustração 42 - Resultado tela animais

Cod	Nome	Raça	Data Nascimento	Qtd Media por Ordenha	Quantidade Total	Ações
4	Demostração		08/05/2019	9.66666666666667 Litros	29 Litros	

Fonte: Autor (2020)

Apenas uma alteração foi realizada quando comparado os resultados da aplicação com os wireframes apresentados no capítulo anterior. Essa alteração foi no cadastro do animal, que por motivos de lógica de programação passou a ser uma nova tela. A foto dessa tela está disponível na Ilustração 43.

Ilustração 43 - Resultado tela cadastrar animais

Cadastrar Animal

Fonte: Autor (2020)

4.2 HARDWARE

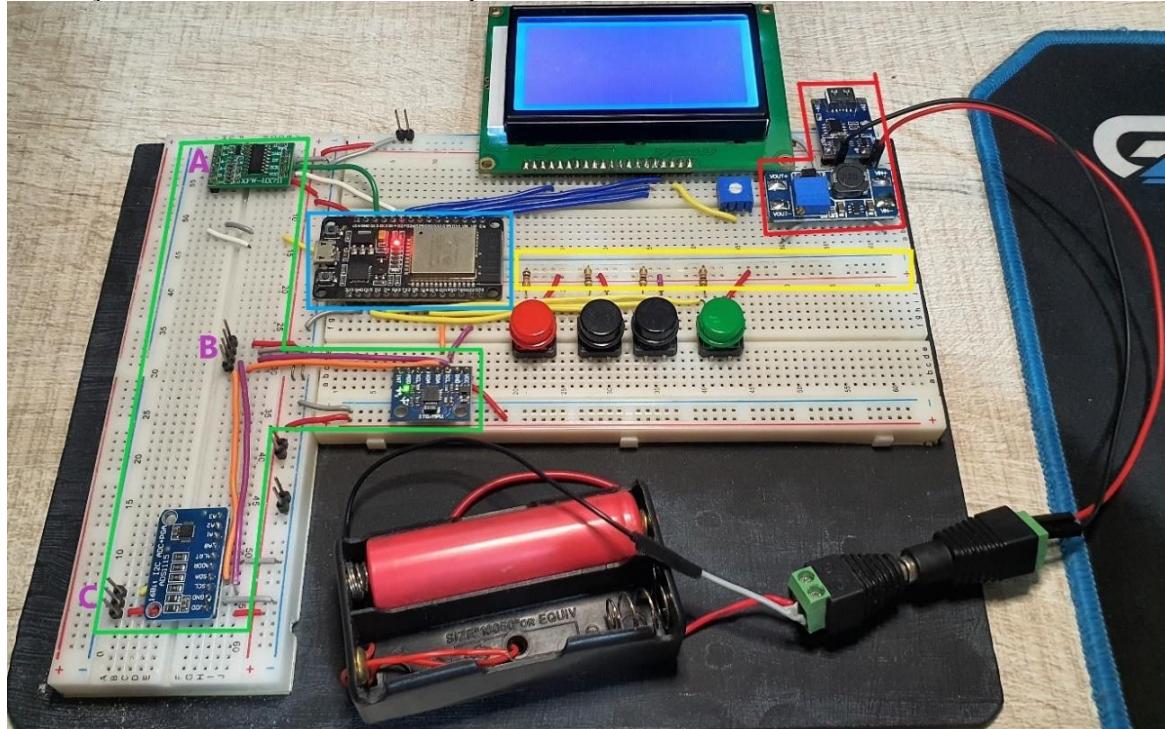
Para a montagem do hardware foi inicialmente realizada uma implementação com todos os sensores, esse hardware é o que foi previsto no capítulo 3. Esse processo tem o intuito de testar e comparar os sensores escolhidos, de modo a apontar qual a melhor opção para realizar devida a medição. Dessa forma foram desenvolvidos dois hardwares, o primeiro com todos os sensores e o segundo apenas com o sensor que obteve o melhor desempenho na medição.

O primeiro hardware foi montado em protoboard e realizado teste em bancada, já para o segundo foi confeccionada uma PCB e além do teste em bancada foram realizados testes em campo. Nos itens abaixo serão apresentados todos esses processos.

4.2.1 Montagem protoboard

O passo inicial foi a montagem do circuito proposto no diagrama da Ilustração 34 em uma protoboard, realizado a ligação de todos os dispositivos e componentes utilizados. A Ilustração 44 exibe a montagem demarcando as áreas do circuito.

Ilustração 44 - Sistema montado na protoboard



Fonte: Autor (2020)

A área demarcada pelas linhas amarelas são as trilhas de energia alimentadas com 3.3V, responsável por alimentar os Botões. As demais trilhas são alimentadas com 5V e são responsáveis por alimentar os sensores e dispositivos.

A área em vermelho representa a alimentação do sistema, essa parte possui o TP4056 que atua como controlador de carga da bateria, uma bateria 18650 e o buck-boost MT3608 que é responsável por elevar atenção da bateria que é 4,2V para 5V, suprindo as necessidades do circuito.

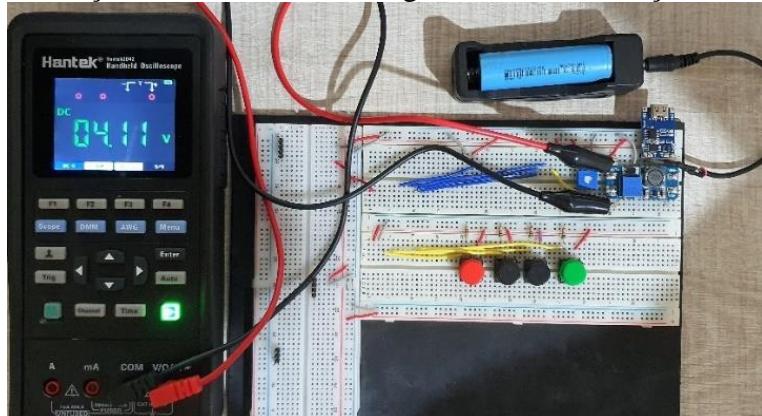
A parte central concebe a IHM, os dispositivos de interação com o usuário. Essa parte é composta por um display de 3,3 Polegadas e os 4 botões, sendo eles: um botão OK, um botão Voltar/Cancelar, um botão cima e outro baixo utilizados para navegar entre as opções e menus do sistema. A área Azul contempla o microcontrolador ESP32 responsável por interligar e controlar todos os módulos e dispositivos.

Por fim a área verde que é composta pelos sensores e conversores. Como os sensores foram instalados no protótipo de medição para cada sensor foi colocado um conector, representados pelas siglas A (Célula Carga), B (Laser), C (Pressão). Além disso, essa área é composta por dois conversores ADC, o Hx711 e o ADS1115, responsáveis por controlar as células de carga e o sensor de pressão respectivamente. O sensor Laser e o giroscópio são conectados diretamente ao microcontrolador por I2C.

Antes de ligar todo o circuito foi necessário realizar o ajuste do buck-boost, para realizar esse ajuste foram retirados da protoboard todos os componentes que não fazem parte do sistema de energia, deixando apenas a bateria, o controlador de carga, o buck-boost e os botões (pois não há risco de danificação por sob ou sub tensão).

Ao ligar o sistema conforme demonstra a Ilustração 45 a saída do circuito estava com 4.11V o objetivo é regular essa saída para 5V.

Ilustração 45 - Sistema de energia antes da calibração



Fonte: Autor (2020)

Com uma chave o potenciômetro do buck-boost foi ajustando até o momento em que a saída do circuito foi igual à saída desejada. A Ilustração 46 demonstra o processo de regulagem.

Ilustração 46 - Calibração sistema de energia



Fonte: Autor (2020)

Após a regulagem, a Ilustração 47 mostra o circuito ajustado para os exatos 5V. Depois da ligação de todos os componentes a tensão reduziu para a casa dos 4.8V, para compensar essa redução foi realizado outro ajuste agora com todos os componentes conectados.

Ilustração 47 - Sistema de energia calibrado



Fonte: Autor (2020)

4.2.2 Firmware de calibração dos sensores

O firmware que foi apresentado nesse tópico é o firmware que foi utilizado nos testes dos sensores e foi desenvolvido no Arduino IDE. Vale ressaltar que nesse tópico estará disponível apenas as partes principais do código, o código completo está disponível no Github¹. O primeiro passo para o desenvolvimento desse firmware foi encontrar e instalar as bibliotecas que são utilizadas, no total foram utilizadas 8 bibliotecas que estão disponíveis no Quadro 15.

¹ Repositório github: <https://github.com/EduardoBevilaqua/TCC2>

Quadro 15 - Bibliotecas Firmware

```
#include <HX711.h> //Biblioteca Celula de carga https://github.com/bogde/HX711
#include <U8g2lib.h> //Biblioteca Display
#include <Adafruit_VL53L0X.h> //Biblioteca Sensor de Distancia a Laser
#include <SPI.h> //SPI
#include <Wire.h> //Biblioteca I2C
#include <MPU6050_tockn.h> //Biblioteca Giroscopio
#include <EEPROM.h> // Biblioteca para Manipular EEPROM
#include <Adafruit_ADS1015.h> //Biblioteca conversor ADC
```

Fonte: Autor (2020)

Cada biblioteca possui um papel importante no código, elas possuem um conjunto de instruções e funções que funcionam como um intermediário entre o hardware e o firmware.

No decorrer do código tem o setup, Quadro 16, essa parte tem o objetivo de inicializar os recursos. Após iniciar os recursos básicos de porta Serial, display e I2C a mensagem “Iniciando Dispositivo!” sera exibida no display. O código também faz uma verificação se os botoes “ok” e “voltar” estão pressionados, dessa forma se inicializar o dispositivo com esses dois botoes pressionado o sistema entrará em modo serviço, modo desenvolvido para realizar o ajuste e calibração dos sensores. Após a verificação do modo serviço o sistema chama as demais funções que inicializam os outros sensores, essas funções serão abordadas no decorrer desse trabalho. Por final o código verifica se esta em modo serviço para entrar no menu de serviço.

Quadro 16 - Setup Firmware

```
void setup(void) {
    Serial.begin(115200);
    display("Iniciando Dispositivo!", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    u8g2.begin();
    Wire.begin();
    //Verifica se os botoes estão precionados para entrar em modo serviço
    if(digitalRead(botaoVoltar) == HIGH && digitalRead(botaoOk) == HIGH){
        statusMenuServico = 1;
        display("Modo Servico!", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
        delay(700);
    }
    // inicia Sensores
    ads.begin();
    inicializaLaser();
    inicializaGiroscopio();
    inicializaBalaca();
    if(statusMenuServico == 1){
        menuServico();
    }
}
```

Fonte: Autor (2020)

O Quadro 17, apresenta as funções de inicialização dos sensores. Para os sensores laser e giroscópio apenas é chamada a função begin() que inicializa o mesmo, e exibe no display que o sensor está sendo iniciado. Já para as células de cargas denominadas no código como balança, alguns outros passos foram realizados. O primeiro desses passos é buscar na EEPROM o valor

armazenado do fator de calibração, após isso, foi iniciado o sensor chamando a função begin(), passando os pinos no qual o HX711 foi conectado ao microcontrolador. Após iniciar outras duas funções são chamadas, que são a tare() que zerar a balança e por último a função set_scale() sendo responsável por setar o valor de calibração.

Quadro 17 - Código Firmware inicia sensores

```
void inicializaLaser(){
    display("Iniciando Sensor Laser!", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    lox.begin();
    delay(1000);
}
void inicializaGiroscopio(){
    display("Iniciando Giroscopio!", 5, 35,"",62,45,"",50,50,"",0,0,"",0,0);
    mpu6050.begin();
    mpu6050.calcGyroOffsets(true);
}
void inicializaBalanca(){
    display("Iniciando Balanca!", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    calibration_factor = lerEeprom(10);
    balanca.begin(OUT_Balanca, CLK_Balanca);
    balanca.set_scale();
    balanca.tare();
    balanca.set_scale(calibration_factor);
}
```

Fonte: Autor (2020)

No Quadro 18 a parte do código que realiza a gravação e leitura na EEPROM. Para a gravação dois valores são passados para a função, que são a posição da memória e o valor a ser gravado. Os passos necessários para realizar essa operação são: iniciar a EEPROM passando o tamanho total dela, zerar o valor atual da posição que será realizada a gravação, dividir os bytes do valor que irá ser gravado pela metade (isso para que o valor possa ser armazenado em dois espaços), passar o valor a ser gravado para a função write e por último executar um commit() que salvara os dados. Para a leitura basta passar a posição que deseja ser lida e executar a função read().

Quadro 18 - Código firmware manipulação EEPROM

```
void gravarEeprom(int value, int pos) {
    display("Gravando Valor = ", 5, 5,String(value),5,60,"",0,0,"",0,0,"",0,0);
    delay(100);
    EEPROM.begin(1024);
    EEPROM.write(pos, 0);
    EEPROM.write(pos + 1, 0);
    byte hiByte = highByte(value);
    byte loByte = lowByte(value);
    EEPROM.write(pos, hiByte);
    EEPROM.commit();
    EEPROM.write(pos + 1, loByte);
    EEPROM.commit();
    EEPROM.end();
}
int lerEeprom(int pos) {
    EEPROM.begin(1024);
    byte hiByte = EEPROM.read(pos);
    byte loByte = EEPROM.read(pos + 1);
```

```

EEPROM.end();
int saida = word(hiByte, loByte);
return saida; }

```

Fonte: Autor (2020)

A Ilustração 48 mostra o menu desenvolvido para escolher o sensor que será realizada a calibração. No decorrer do texto será apresentado a calibração de cada um dos sensores.

Ilustração 48 - Menu de serviço sistema embarcado



Fonte: Autor (2020)

A função do Quadro 19 captura os dados do sensor laser e trata o valor. Para capturar a distância a função `measure.RangeMilliMeter` da biblioteca “Adafruit_VL53L0X.h” é utilizada e retorna a distância em MM. Com o intuito de diminuir o erro na medição são capturadas 3000 amostras, uma, a cada 5ms, o acumulado de medições são armazenados na variável `distancia` que após concluir as 3000 medidas retorna a média do valor total. Dividindo o valor encontrado em MM por 10 obtemos a medida em cm e o nível se refere à altura da coluna de líquido no recipiente, para isso é subtraída a altura total do reservatório pela altura medida.

Quadro 19 - Código firmware captura dados sensore laser

```

void exibeDadosLaser(){
    VL53L0X_RangingMeasurementData_t measure;
    lox.rangingTest(&measure, false);
    float distanciaMM=0;
    for(int i=0;i<3000;i++){
        distanciaMM = distanciaMM + measure.RangeMilliMeter;
        delay(0.5);
    }
    distanciaMM = distanciaMM/3000;
    distanciaMM = (0.9065 * distanciaMM) - 0.9082; //Função de calibração
    float distanciaCM = distanciaMM / 10;
    float nivel = 36 - distanciaCM;
    display("Distancia MM: "+String(distanciaMM),5,10,"Distancia CM: "+String(distanciaCM),5,25,"Nivel:
    "+String(nivel),5,40,"",0,0,"",0,0);
}

```

Fonte: Autor (2020)

Por último os dados obtidos foram exibidos no display, em uma tela igual a da Ilustração 49. Observação: os dados apresentados na Ilustração são fictícios.

Ilustração 49 - Tela calibração sensor laser



Fonte: Autor (2020)

Para o sensor de pressão alguns cálculos a mais são utilizados, o trecho do código que captura e exibe os dados do sensor estão disponíveis no Quadro 20. Para realizar a medição com esse sensor foi utilizado o conversor analógico digital ADS1115 de 16bits, com o intuito de aumentar a precisão na leitura da tensão. Uma vez a resolução do sensor de pressão é 45mV/KPa, logo 1KPa equivale a 45mV e 1KPa no teorema de Stevin equivale a uma coluna de água de 10,19 cm. Ao calcular o volume de água ou leite contido em 10,19 cm, considerando um tarro convencional que contém diâmetro interno de 32 cm, obtemos um total de 8,04 Litros. Um valor alto e que torna o sistema ineficaz.

Quadro 20 - Código firmware captura dados sensores de pressão

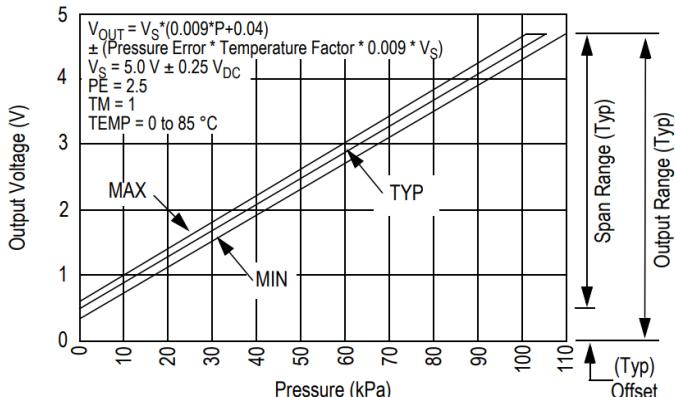
```
void exibeDadosPressao(){
    float Volt = 0;
    adc0 = ads.readADC_SingleEnded(0);
    Volt = (adc0 * 0.1875)/1000;
    double pressao = (( Volt - 0.04 * 5 ) / (0.009 * 5)) + (2.5 * 1 * 0.009 * 5);
    double nivel = ((pressao * 1000) / (997 * 9.8)) * 100;
    float litros = (3.14 * (14.75 * 14.75) * nivel)/1000;
    litros = (0.8999 * litros) + 0.3502; //Função de calibração
    display("Pressao: " + String(pressao), 5, 10,"Nivel: " + String(nivel),5,25,"Qtd: " +
    String(litros,3),5,40,"Tensao: " + String(Volt, 6),5,55,"",0,0);
}
```

Fonte: Autor (2020)

O primeiro passo é capturar a leitura do ads, para isso a variável adc0 recebe o valor da porta 0 do conversor, seguindo os exemplos da biblioteca Adafruit_ADS1015.h². Para transformar o valor obtido em Volts é necessário aplicar a equação $Volt = \frac{adc0*0.1875}{1000}$, sendo adc0 a saída do ADS1115. Após isso o valor já está em volts e assim é possível capturar a pressão.

² Link da biblioteca <https://www.arduinolibraries.info/libraries/adafruit-ads1-x15>

Ilustração 50 - Gráfico saída sensor de pressão



Fonte: DataSheet “Anexo A”

A Ilustração 50 contém a fórmula para capturar a pressão, ajustando a fórmula para a aplicação a equação que foi utilizada é: $\text{Pressão} = \frac{\text{Volt} - 0.04 * 5}{0.009 * 5} + (2.5 * 1 * 0.009 * 5)$, sendo “Volt” a tensão calculada no passo anterior. Após calcular a pressão, utilizando a fórmula do teorema de Steven é possível descobrir a altura da coluna do líquido, e com a altura utilizando a fórmula de volume para um cilindro facilmente é possível obter a quantidade de líquido no interior do recipiente.

Ilustração 51 - Tela calibração sensor de pressão



Fonte: Autor (2020)

A ilustração 51 contém uma foto tela de calibração do sensor de pressão, com os dados fictícios.

Para capturar os dados da balança, Quadro 21, o primeiro passo é setar o valor de calibração da balança, após isso 5 amostras são obtidas da balança e a média dessas amostras é o peso. Como para a calibração foi utilizado água, a quantidade em litros é igual à quantidade em kg, pois a densidade da água é 1 kg/L.

Quadro 21 - Código calibração célula de carga

```
balanca.set_scale(calibration_factor);
float litros = 0, peso = 0;
for(int i=0;i<5;i++){
    peso = peso + balanca.get_units();
    delay(0.5);
}
peso = (peso / 5);
litros = peso;
display("Peso: "+String(peso,3)+" KG" ,5,10,"Litros: "+String(litros,3)+" Litros",5,25,"Fator:
" + String(calibration_factor),5,40,"",5,40);
delay(100) ;
```

Fonte: Autor (2020)

A tela com os dados obtidos e que foi utilizado para calibrar a célula de carga está disponível na Ilustração 52.

Ilustração 52 - Tela de calibração célula de carga

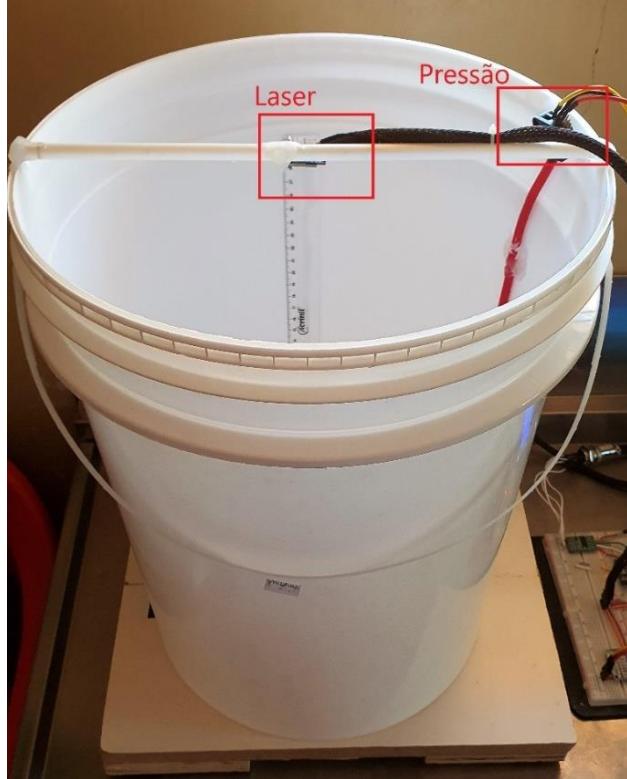


Fonte: Autor (2020)

4.2.3 Montagem do protótipo de medição

Para realizar os testes de medição um balde de 20 Litros, disponível na Ilustração 53 foi preparado para ser utilizado como suplemento do tarro de leite. Nesse recipiente foram instalados os sensores e uma régua na parede interna para acompanhar o nível. O sensor de pressão foi instalado na parede lateral e o sensor de laser em uma haste bem no centro do recipiente.

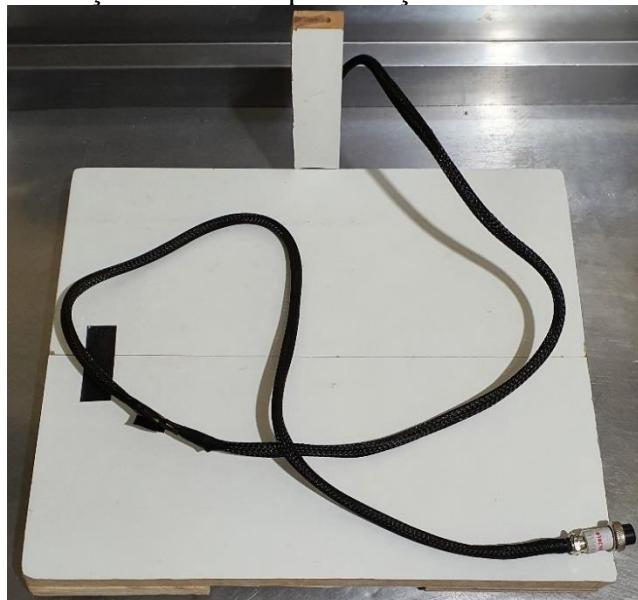
Ilustração 53 - Recipiente protótipo de medição



Fonte: Autor (2020)

As células de cargas foram montadas sob uma estrutura de madeira conforme demonstra a Ilustração 54. A estrutura funciona como uma balança e servira de suporte para o tarro de leite ou protótipo de medição. As dimensões dessa estrutura são: largura: 34 cm e comprimento: 29 cm, sendo que o conjunto de células possui capacidade para pesar até 200KG.

Ilustração 54 - Protótipo balança



Fonte: Autor (2020)

Foram alocadas uma célula em cada canto da estrutura seguindo as ligações da Ilustração 35, dessa forma a estrutura funciona como uma balança. Na saída das células um foi adicionado conector circular de 4 Vias o qual se conectara a central de controle. A Ilustração 55 mostra a parte inferior da balança, com as células de carga instaladas uma em cada canto da estrutura.

Ilustração 55 - Parte inferior protótipo balança



Fonte: Autor (2020)

4.2.4 Análise dos sensores

Após realizar os testes e calibrar os sensores todos os resultados obtidos foram anotados em uma tabela, aonde a partir desses dados, foi realizada uma análise individual com o intuito de demonstrar quais sensores são eficazes para a aplicação. Nos subtópicos abaixo será abordado essa análise, apontando os resultados, prós e contras de cada sensor.

No código do sensor de pressão e no sensor laser revelados anteriormente, existe uma equação cuja linha está comentada com “//Função de calibração”. Essa equação é a equação linear utilizada para ajustar o valor de saída. Os passos realizados para chegar a essa equação serão expostos nesse tópico.

Todos os testes foram com o recipiente de teste posto sobre a balança e os demais sensores conectados a protoboard. A ilustração 56 demostra o ambiente de teste que seguiu o seguinte padrão:

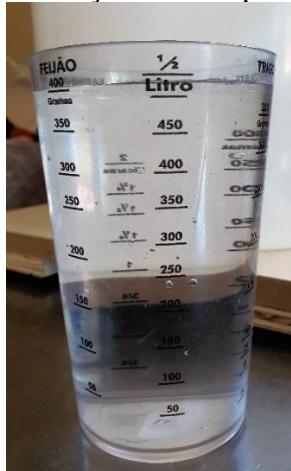
Ilustração 56 - Ambiente de teste dos sensores



Fonte: Autor (2020)

Foi adicionado ao recipiente 1 litro de água por vez, após inserir um litro a medição era anotada e outro litro era adicionado. Para contabilizar um litro foi utilizado O item da Ilustração 57, um copo de medição com capacidade de 500ML, logo eram adicionados dois copos por vez.

Ilustração 57 – Copo de medição



Fonte: Autor (2020)

Cada sensor foi calibrado de forma individual e foi adicionado um total de 8 litros de água para cada sensor. A Ilustração 58 apresenta a calibração da célula de carga com 1 litro adicionado.

Ilustração 58 - Teste dos sensores com 1 litro



Fonte: Autor (2020)

4.2.4.1 Sensor de pressão

Seguindo o padrão de teste descrito anteriormente o primeiro sensor a ser testado foi o sensor de pressão MPX5100DP. Conforme pode ser verificado na Tabela 4 a medição apresentou o maior erro quando o recipiente estava vazio e o melhor resultado com 3 litros onde não obteve erro na medição. A média de erros foi de 13,44%.

Tabela 4 - Tabela calibração sensor de pressão sem ajuste

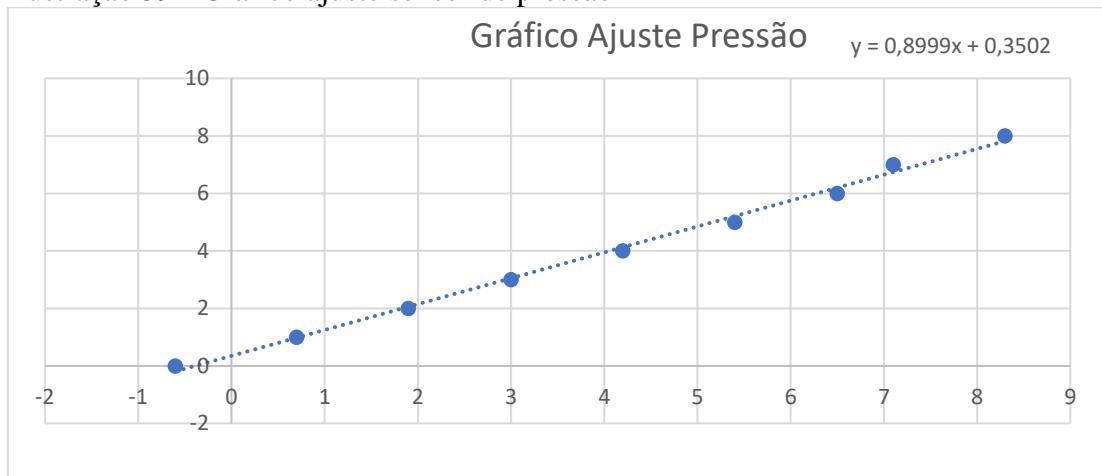
Sensor Pressão Sem Ajuste			
litros Medida	litros	Diferença	Erro %
-0,6	0	-0,6	60%
0,7	1	-0,3	30%
1,9	2	-0,1	5%
3	3	0	0%
4,2	4	0,2	5%
5,4	5	0,4	8%
6,5	6	0,5	8%
7,1	7	0,1	1%
8,3	8	0,3	4%

Fonte: Autor (2020)

Com intuito de melhorar a medição e trazer resultados mais fiéis a realidade, com o auxílio do Exel, foi criado um gráfico linear com os resultados obtidos. A partir desse gráfico

foi gerado uma equação linear que possui a função de ajustar os resultados. A Ilustração 59 exibe esse gráfico com a equação.

Ilustração 59 - Gráfico ajuste sensor de pressão



Fonte: Autor (2020)

Após o uso da Equação linear o resultado obtido foi o apresentado na Tabela 5, conforme pode ser observado a medição apresentou um resultado mais preciso do que sem o uso da equação. A média de erros agora caiu para 4% sendo um valor menor que o valor obtido antes do uso da equação.

Tabela 5 - Tabela calibração sensor de pressão com ajuste

Sensor Pressão Com Ajuste			
litros Medida	litros	Diferença	Erro %
0,1	0	0,1	10%
1,1	1	0,1	10%
2	2	0	0%
3,1	3	0,1	3%
4,1	4	0,1	2%
4,9	5	-0,1	-2%
6,1	6	0,1	2%
7,2	7	0,2	3%
8,3	8	0,3	4%

Fonte: Autor (2020)

Porém, apesar do sistema com o sensor de pressão MPX5100DP apresentar uma medição com boa precisão. Sua utilização não é recomendada pelos seguintes motivos:

- Como citado nos tópicos anteriores e segundo o datasheet “Anexo A” o sensor não pode ser utilizado diretamente em contato com o líquido e por esse motivo ele ficou instalado

na parte superior do tarro, medindo a pressão do ar. Porém, o ar possuir propriedades diferentes do líquido, fato que diminui a confiabilidade da medição.

- A pressão do ar tem relação com a temperatura, em testes de bancada utilizando um secador de cabelo. Foi realizado o aquecimento do ambiente interno do recipiente e a consequência foi uma variação de até 1,5 litros alterando a temperatura em poucos graus célsius. Com isso em um cenário aonde ao produtor realiza a ordenha em uma manhã fria com temperaturas próximo de 0 graus e em tardes quentes onde essa temperatura pode estar próxima aos 30 graus, a medição será diferente em cada dia e o sistema necessitaria de uma calibração para cada temperatura.
- Outro problema verificado é necessidade de aguardar o líquido estabilizar para realizar a medição. Com o líquido em movimento a medição varia em torno de 500ML e após cerca de 1 minuto o líquido estabiliza e a variação cai para cerca de 100ML. Contudo, esse tempo para estabilização pode ser maior dependente do nível do recipiente e a agitação do líquido.

4.2.4.2 Sensor Laser

Para o sensor a laser o teste inicial e a calibração foram executados inicialmente com um sólido. Para realiza esse teste o sensor foi instalado no suporte da balança e uma caixa de papelão foi utilizada como objeto de referência. A Ilustração XX demostra esse ambiente de teste, a caixa de papelão foi afastada de 1 em 1 cm, das distâncias de 5 cm a 29 cm.

Conforme o datasheet “Anexo B” do sensor ele é capaz de medir de 5 cm a 120 cm de distância, fato comprovado em testes, pois em distância abaixo de 5 cm o sensor apresentava valores imprecisos. Segundo a Tabela 6 que demostra os resultados obtidos na medição, o erro máximo foi de 30% e a média de erro foi de 18%.

Tabela 6 - Tabela calibração sensor laser sem ajuste

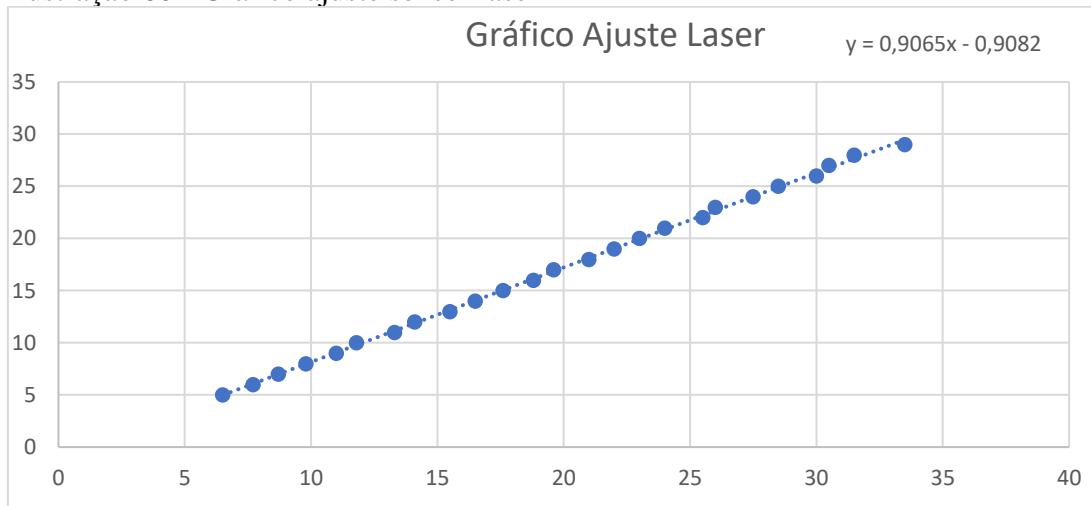
Sensor Laser Sem Ajuste			
Distância Medida	Distância	Diferença	Erro %
6,5	5	1,5	30%
7,7	6	1,7	28%
8,7	7	1,7	24%
9,8	8	1,8	23%
11	9	2	22%
11,8	10	1,8	18%
13,3	11	2,3	21%

14,1	12	2,1	18%
15,5	13	2,5	19%
16,5	14	2,5	18%
17,6	15	2,6	17%
18,8	16	2,8	18%
19,6	17	2,6	15%
21	18	3	17%
22	19	3	16%
23	20	3	15%
24	21	3	14%
25,5	22	3,5	16%
26	23	3	13%
27,5	24	3,5	15%
28,5	25	3,5	14%
30	26	4	15%
30,5	27	3,5	13%
31,5	28	3,5	13%
33,5	29	4,5	16%

Fonte: Autor (2020)

Com a finalidade de diminuir a margem de erro foi gerado o gráfico e aplicada a equação linear, conforme demonstra a Ilustração 60.

Ilustração 60 - Gráfico ajuste sensor laser



Fonte: Autor (2020)

Após refazer os testes, porém dessa vez aplicando a equação linear obtivemos resultados diferentes e com uma melhoria em comparação com o anterior. Dessa vez o sensor foi capaz de medir a distância com uma média de erro de 5% e apresentando o pior resultado com na distância de 5 cm com um erro de 8%, conforme demonstra a Tabela 7.

Tabela 7 - Tabela calibração sensor laser com ajuste

Sensor Laser Com Ajuste			
Distância Medida	Distância	Diferença	Erro %
5,4	5	0,4	8%
6,3	6	0,3	5%
7,4	7	0,4	6%
8,1	8	0,1	1%
9,4	9	0,4	4%
10,6	10	0,6	6%
11,5	11	0,5	5%
11,9	12	-0,1	1%
12,8	13	-0,2	2%
13,7	14	-0,3	2%
15,6	15	0,6	4%
16,8	16	0,8	5%
18	17	1	6%
19,2	18	1,2	7%
20,1	19	1,1	6%
20,9	20	0,9	4%
22,2	21	1,2	6%
23,4	22	1,4	6%
24,1	23	1,1	5%
24,8	24	0,8	3%
26,2	25	1,2	5%
27,3	26	1,3	5%
28,5	27	1,5	6%
29,9	28	1,9	7%
31	29	2	7%

Fonte: Autor (2020)

É importante ressaltar que todos os testes foram realizados em um ambiente fechado com uma luz de led ligada, durante as medições foram realizadas tentativas de apagar a luz de modo a verificar a interferência da mesma, porém pouca diferença foi identificada na medição.

Porem além do sensor medir com precisão um caixa de papelão ao realizar a tentativa de medir um líquido, como a água a medição não foi precisa e confiável, isso, pois a mesma variava de forma aleatória. Ao aprofundar os conhecimentos sobre o acontecimento foi identificado que o problema possivelmente é gerado por refracção. Em outro teste foi realizada uma tentativa de medir leite UHT integral e ocorreu o mesmo problema presenciado com a água.

4.2.4.3 Celula Carga

Os testes e calibração da célula de carga foi de uma maneira um pouco diferente dos demais. Para esse sistema o único ajuste que necessita ser feito é na variável fatorCalibacao, aonde ajustando o valor dessa variável é possível regular a precisão do sistema. Para realizar esse ajuste como já citando foi anteriormente utilizado o próprio dispositivo, o qual possui uma função para calibrar a balança.

Para realizar a calibração foram utilizados dois pesos conhecidos, um com 5 kg e outro com 10 kg. Inicialmente esses pesos foram aferidos com o uso de uma balança comercial, para posteriormente ser realizado o processo de calibração, que seguiu o seguinte roteiro:

1. Retirado todos os pesos que estavam sobre a balança.
2. Realizado a tara do seu peso.
3. Adicionado 5 kg sobre a balança.
4. Ajustado o valor da variável FatorCalibração até a balança estar medido aproximadamente 5 kg.
5. Adicionado mais 10kg sobre a balança.
6. Ajustado o valor da variável FatorCalibração até a balança estar medido aproximadamente 15 kg.

Com a balança calibrada foram realizados os testes de medição com o recipiente, os resultados estão disponíveis na tabela 8.

Tabela 8 - Tabela calibração célula de carga

Ajuste Carga			
litros Medida	litros	Diferença	Erro %
0	0	0	0,0%
0,997	1	-0,003	0,3%
2	2	0	0,0%
2,994	3	-0,006	0,2%
3,998	4	-0,002	0,0%
5	5	0	0,0%
5,998	6	-0,002	0,0%
7	7	0	0,0%
8,01	8	0,01	0,1%

Fonte: Autor (2020)

A célula de carga não possuiu pontos falhos na medição se mantendo um sistema preciso e confiável. Em comparação com os demais sensores ficou claro que essa é a melhor opção para o projeto. Possuindo como principal vantagem a precisão na medição e a possibilidade de instalação sem modificações no taro de leite.

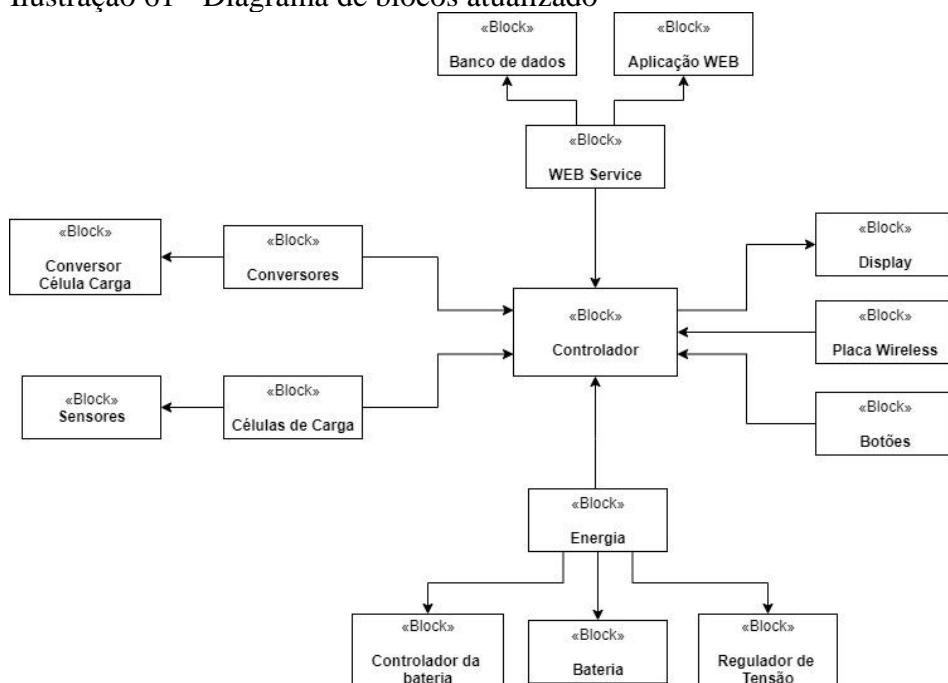
4.2.5 Modificações de Hardware

Após a análise de todos os sensores proposto e levando em consideração os dados obtidos notasse que o melhor e mais preciso sensor para realizar a devida medição é a célula de carga. Dessa forma foi montado um hardware mais simplificado, retirando do circuito os sensores e componentes que não são necessários para o funcionamento do sistema ou célula de carga.

Alguns diagramas apresentados anteriormente foram adequados ao novo circuito, são eles: o diagrama de blocos, diagrama de blocos interno e esquemático elétrico. Além disso, foi adicionado o layout da PCB, esquema projetado para a produção da placa de circuito impresso.

A Ilustração 61 demonstra o diagrama de blocos. Como é possível observar e citando anteriormente ele possui os mesmos componentes do diagrama apresentado no capítulo 3 a única diferença é a remoção dos demais sensores, são eles: sensor laser, sensor de pressão, conversor ADC e giroscópio.

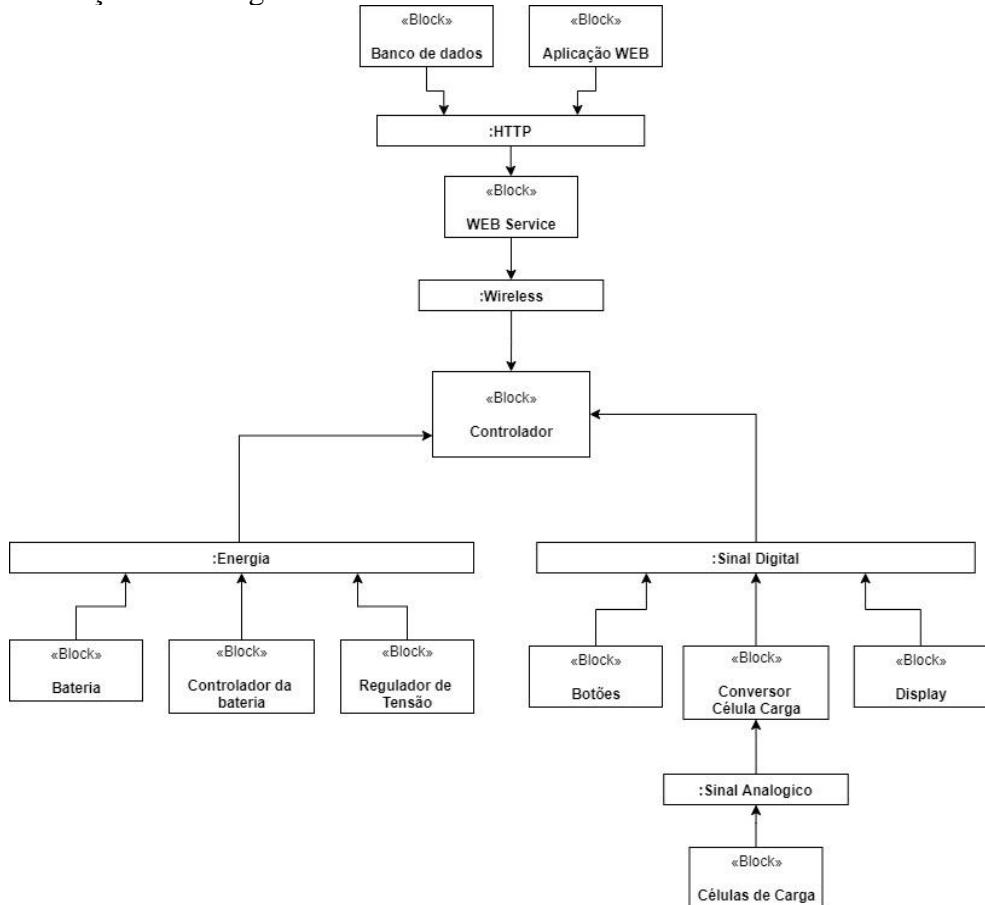
Ilustração 61 - Diagrama de blocos atualizado



Fonte: Autor (2020)

O diagrama da Ilustração 62 é o diagrama de bloco interno, esse diagrama possui os mesmos componentes do diagrama apresentado acima contendo como única diferença a comunicação entre os componentes.

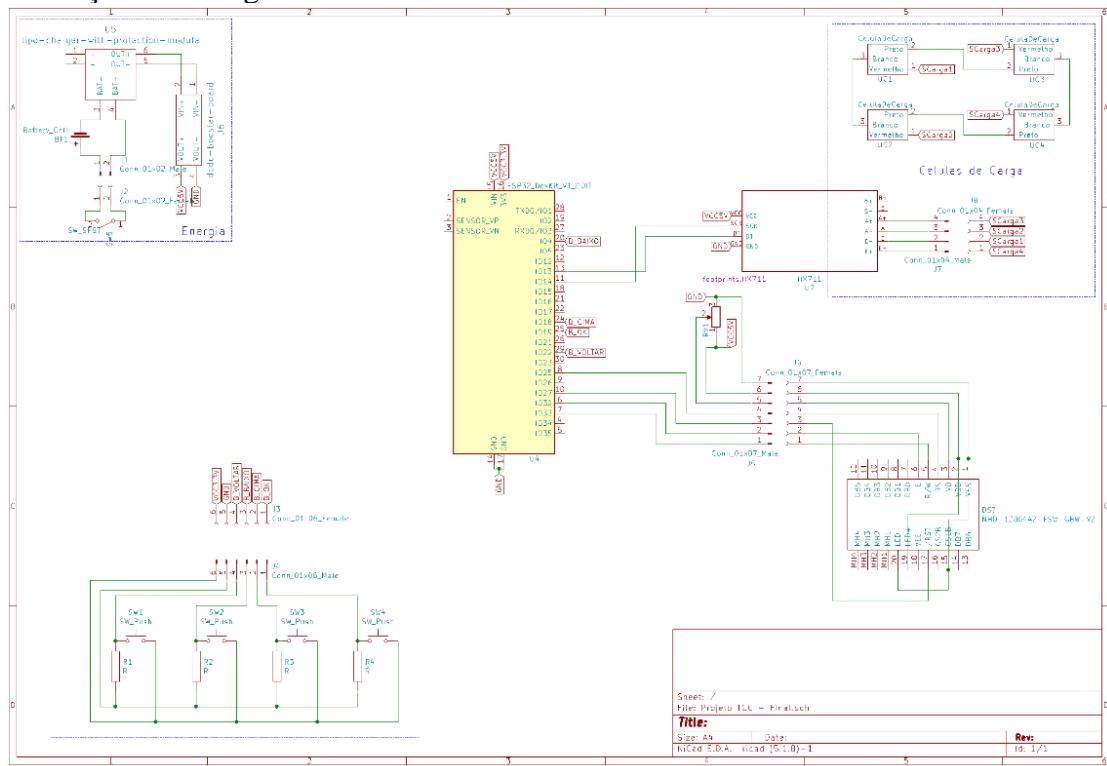
Ilustração 62 - Diagrama de bloco interno atualizado



Fonte: Autor (2020)

Outro diagrama atualizado foi o esquemático elétrico disponível na Ilustração 63 e no “Apêndice D” em alta resolução. Esse diagrama seguindo o mesmo procedimento realizado nos anteriores foi removido os componentes que não serão utilizados. Além de serem adicionados conectores que foram utilizados no desenvolvimento da placa de circuito impresso para realizar a comunicação entre componentes que foram instalados fora da placa principal.

Ilustração 63 - Diagrama elétrico atualizado

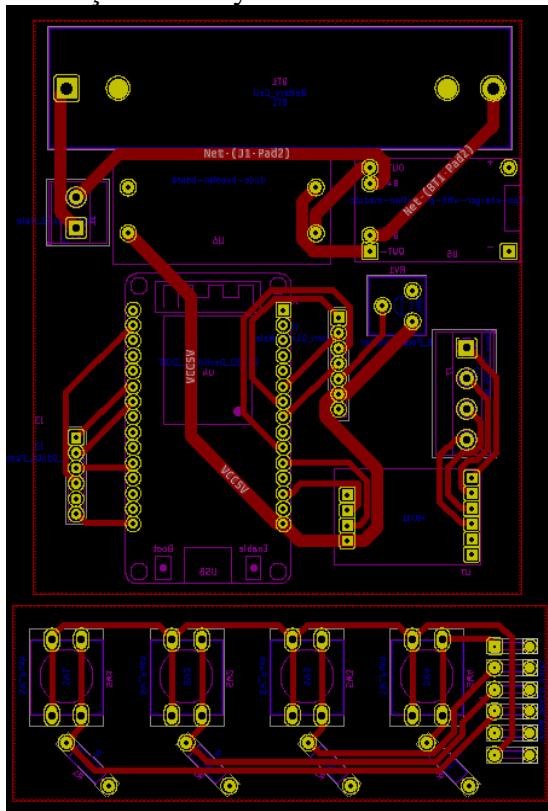


Fonte: Autor (2020)

Os componentes U4, U7, U6, U5, BT1, RV1, J1, J3, J5 e J7 são componentes que foram adicionados diretamente sobre a placa, os demais componentes foram conectados a placa através dos conectores.

Por último um novo diagrama foi adicionado, esse é o layout da PCB, um arquivo que sera utilizado no desenvolvimento da placa de circuito impresso. Foram desenvolvidas duas placas, uma placa para o controlador que é de 8 cm X 9 cm e outra para os botões com dimensões de 9 cm X 3 cm. Na Ilustração 64 contém o projeto das placas, cada uma das linhas vermelhas representam as trilhas que realizam a comunicação entre os componentes.

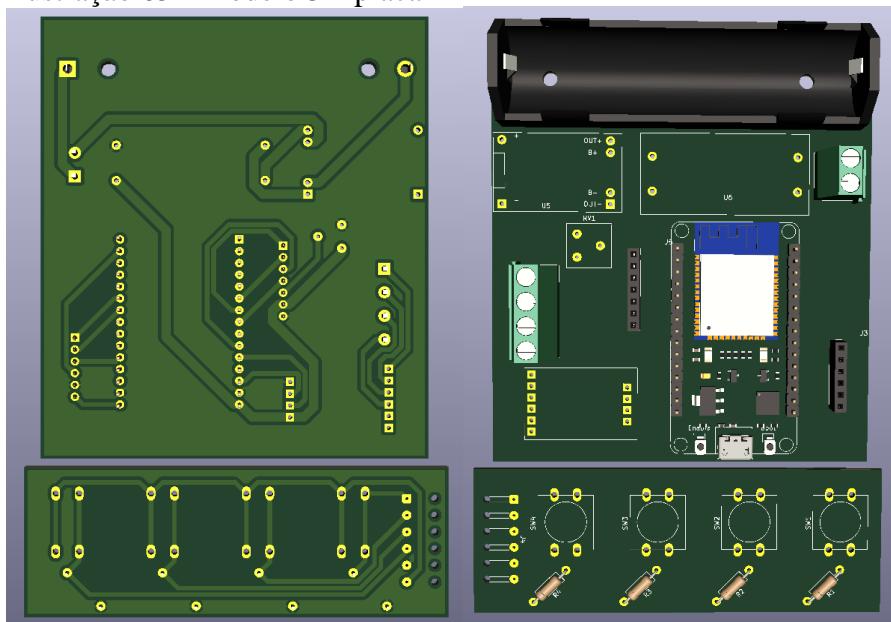
Ilustração 64 - Layout PCB



Fonte: Autor (2020)

O kicad software utilizado para projetar as placas possui a possibilidade de gerar uma visualização 3D do modelo. Essa visualização tanto da parte superior como na parte inferior está disponível na Ilustração 65.

Ilustração 65 - Modelo 3D placa

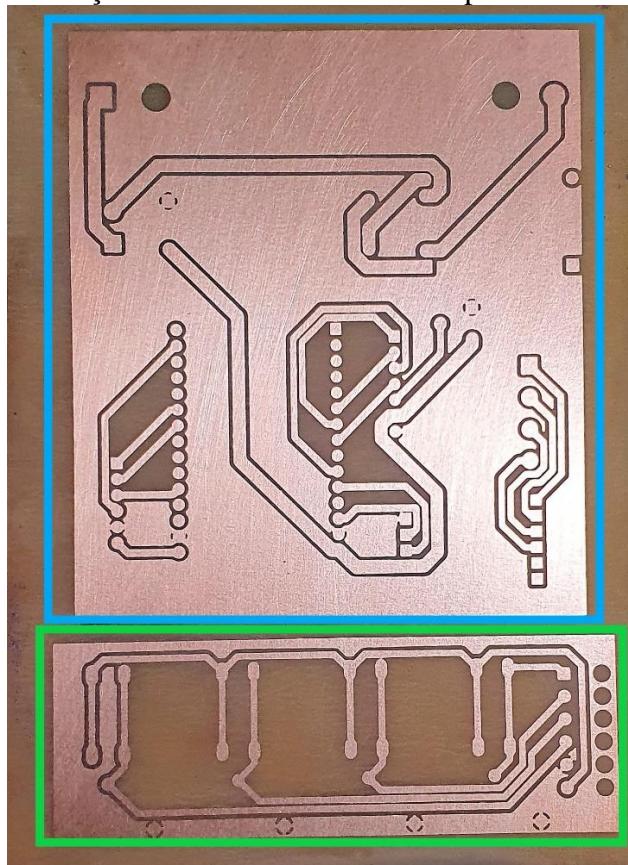


Fonte: Autor (2020)

4.2.6 Hardware final e Case

O primeiro passo para o desenvolvimento do protótipo final foi a manufatura da placa de circuito impresso, A PCB foi desenvolvida com o método manual com o método fotossensível e o resultado obtido está disponível na Ilustração 66. Sendo a parte azul do controlador e a verde a placa dos botões.

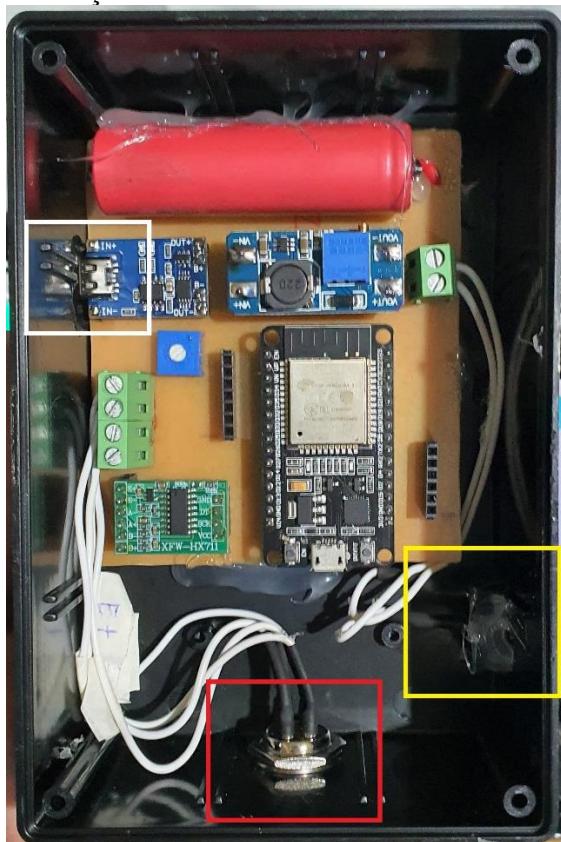
Ilustração 66 - Placa de Circuito Impresso



Fonte: Autor (2020)

Com a PCB desenvolvida e os componentes soldados o próximo passo foi preparar a case que foi utilizada no protótipo. Tratasse de uma caixa de montagem da marca MOLD nas dimensões de comprimento: 142 mm, largura: 92 mm e Altura: 51 mm. Na Ilustração 67 está a caixa de montagem com a placa já posicionada sobre ela.

Ilustração 67 - Placa montada na case

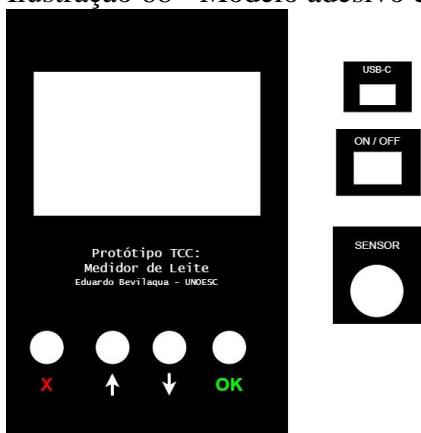


Fonte: Autor (2020)

Na área circulada em vermelho está posicionado a saída para a balança, na área amarela está o botão Liga e desliga do sistema e na área branca está a saída da USB Type-C para carregamento.

Com o dispositivo montado sobre a case foram projetados os adesivos para identificar a case, assim como os botões e conectores. Esse adesivo foi elaborado com o auxílio da ferramenta draw.io e a Ilustração 68 é o modelo do adesivo utilizado.

Ilustração 68 - Modelo adesivo case



Fonte: Autor (2020)

Com os adesivos colados e todos os componentes fixados o protótipo foi concluído e o resultado está na Ilustração 69.

Ilustração 69 - Protótipo Pronto



Fonte: Autor (2020)

4.2.7 Firmware final

Algumas alterações também foram realizadas no firmware. Todas as partes do código que faziam referência aos sensores e que não seriam mais utilizados, foram removidas e algumas partes novas foram adicionadas. Essas funções que foram adicionadas fazem referência a parte de comunicação e a parte da ordenha.

Algumas bibliotecas novas também foram utilizadas, são elas: a “WiFi.h” que realiza a comunicação com a rede Wifi, a “HTTPClient.h” utilizada para comunicação HTTP e a “ArduinoJson.h” utilizada para manipular dados Json. O Quadro 22 mostra a lista das bibliotecas atualizadas.

Quadro 22 - Bibliotecas firmware atualizadas

```
#include <HX711.h> //Biblioteca Celula de carga https://github.com/bogde/HX711
#include <Arduino.h>
#include <U8g2lib.h> //Biblioteca Display
#include <Wifi.h> //Biblioteca WiFi
#include <EEPROM.h> //Biblioteca para manipular EEPROM
#include <HTTPClient.h> //Biblioteca Para requisição HTTP
#include <ArduinoJson.h> //Biblioteca para manipular Json
```

Fonte: Autor (2020)

A primeira função configurada foi a conectarWifi, disponível no Quadro 23. Essa função tem o objetivo de realizar a conexão com a rede wifi. No wifi begin basta passar o nome da rede wifi e a senha e aguardar a conexão, foi implementado uma função para resetar o esp32 caso não consiga se conectar ao wifi após 8 tentativas. Essa função foi implementada após avaliar o funcionamento e detectar que normalmente ele se conecta à rede até a oitava tentativa e se a caso não conectar após 8 tentativas, demora um tempo muito grande para realizar a conexão. Dessa forma realizado o reset o mesmo se conecta de forma mais rápida.

Quadro 23 - Código firmware conectar wifi

```
void conectarWifi(){
    display("Conectando Wifi!", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    int cont = 0;
    WiFi.begin("Eduardoo", "123eduardo2017");
    while (WiFi.status() != WL_CONNECTED) {
        cont++;
        delay(400);
        if(cont>=8){
            esp_restart();
        }
    }
}
```

Fonte: Autor (2020)

A função do Quadro 24 foi adicionada ao firmware e tem o objetivo de capturar os dados da balança. A função contém o mesmo escopo da função de calibração da célula de carga, porém agora ao invés de água, o sistema será utilizado para medir leite e por esse motivo foi necessário levar em consideração a densidade do leite. Segundo Brito (2020b) “A densidade do leite é, em média, 1,032 g/mL, podendo variar entre 1,023 e 1,040 g/mL.”. Assim utilizando a densidade média, o primeiro passo é converter 1,032 g/mL para 1,032 kg/L e aplicar a seguinte equação para obter o valor em litros: $Litros = \frac{Peso}{1,032\ Kg/L}$.

Quadro 24 - Código firmware captura dados balança

```
float dadosBalaca(){
    float litros = 0;
    float peso = 0;
    for(int i=0;i<5;i++){
        peso = peso + balanca.get_units();
        delay(0.5);
    }
    peso = peso / 5;
    litros = peso / 1,032;
    return (litros);
}
```

Fonte: Autor (2020)

A função "receberAnimais" do Quadro 25, é a função que busca no servidor PHP a lista de animais cadastrados no sistema. Após receber o retorno do servidor é realizado um

tratamento nos dados, esse tratamento tem o objetivo de eliminar todas as informações do cabeçalho HTTP, deixando a variável resposta apenas com o Json enviando. Com apenas o Json na variável a função “deserializeJson” decodifica os dados, sendo passados para as variáveis Vaca1, Vaca2 e Vaca3.

Quadro 25 - Código firmware receber lista animais

```
void receberAnimais(){
    StaticJsonDocument<300> doc;
    char json[500] = {0};
    const char* serverName = "http://192.168.0.199/AplicacaoTCC/public/Animais/listaAnimais";
    HttpClient http;
    http.begin(serverName);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    int httpResponseCode = http.POST("chave=ABC123");
    if(httpResponseCode>0){
        String resposta = http.getString();
        int lim1 = resposta.indexOf("<");
        resposta = resposta.substring(0,lim1 - 2);
        resposta.replace(" ", "");
        resposta.replace("\n", "");
        resposta.trim();
        resposta.toCharArray(json, 500);
        deserializeJson(doc, json);
        String vaca11 = doc["vaca1"];
        String vaca22 = doc["vaca2"];
        String vaca33 = doc["vaca3"];
        vaca1 = vaca11;
        vaca2 = vaca22;
        vaca3 = vaca33;
    }
    http.end();
}
```

Fonte: Autor (2020)

Após receber a lista de animais o sistema está preparado para iniciar a ordenha, um menu igual ao da Ilustração 70 é exibido e a partir desse menu o usuário do sistema seleciona qual animal será ordenhado.

Ilustração 70 - Menu seleciona animal



Fonte: Autor (2020)

Quando o animal for selecionado a ordenha inicia e a função do Quadro 26 é chamada. O primeiro passo do sistema é chamar a função tare() que zera a balança, desconsiderando o

leite que já estava no tarro ou até mesmo o peso do taro caso seja a primeira ordenha. Após isso no display é exibido a mensagem “Ordenha iniciada” e a partir desse ponto o sistema já começa a calcular a quantidade de leite, até que a tecla verde seja pressionada, indicando o final da ordenha.

Quadro 26 - Código firmware ordenha

```
void ordenha(String Vaca){
    float quantidade;
    display("Iniciando Ordenha!", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    delay(100);
    display("Capturando Dados...", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    delay(100);
    balanca.tare();
    while(statusOrdenha == 1){
        quantidade = dadosBalaca();
        display("Ordenha Iniciada! ", 5, 20,"Litros: "+String(quantidade,3)+" L",5,35,"",45,35,"",0,0,"",0,0);
        if(digitalRead(botaoOk) == HIGH){
            display("Finalizando Ordenha! ", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
            delay(100);
            enviarMedidas(Vaca, quantidade);
            statusOrdenha = 0;
        }
        if(digitalRead(botaoVoltar) == HIGH){
            display("Ordenha Cancelada! ", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
            delay(500);
            statusOrdenha = 0;
        }
        delay(50);
    }
    display("Finalizando Ordenha! ", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    delay(500);
}
```

Fonte: Autor (2020)

Por último a função enviarMedidas, Quadro 27, essa função tem o objetivo de enviar os resultados da ordenha para o servidor. O envio é feito pelo método POST e são enviados a quantidade de leite, nome do animal e chave, utilizada para segurança.

Quadro 27 - Código firmware envia medidas para o servidor

```
void enviarMedidas(String nome, float medida){
    String readString;
    String data = "chave=";
    data += "ABC123";
    data += "&quantidade=";
    data += medida;
    data += "&nome_animal=";
    data += nome;
    display("Enviando Dados ", 5, 35,"",0,0,"",0,0,"",0,0,"",0,0);
    String link = "POST /AplicacaoTCC/public/Medidas/inserir HTTP/1.1";
    if(client.connect("192.168.0.199", 80)){
        client.println(link);
        client.println("Host: 192.168.0.199");
        client.println("Content-Type: application/x-www-form-urlencoded");
        client.print("Content-Length: ");
        client.println(data.length());
        client.println();
        client.println(data);
        client.println();
    }
}
```

```

if(client){
    while(client.connected()){
        while(client.available()){
            char c = client.read();
            readString += c;
        }
    }
    readString.trim();
    Serial.println(readString);
    client.stop();
    readString = "";
}

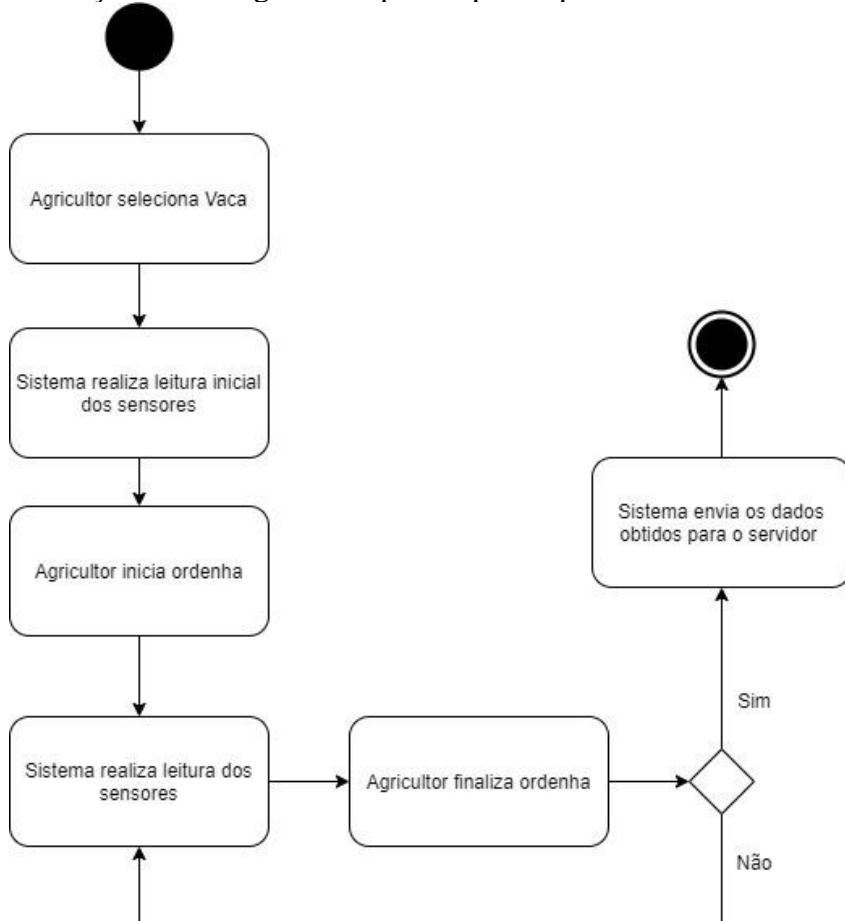
```

Fonte: Autor (2020)

4.3 CATARCTERISTICAS E FUNÇOES

Para realizar a medição da ordenha o usuário deverá realizar os passos informados no diagrama da Ilustração 71.

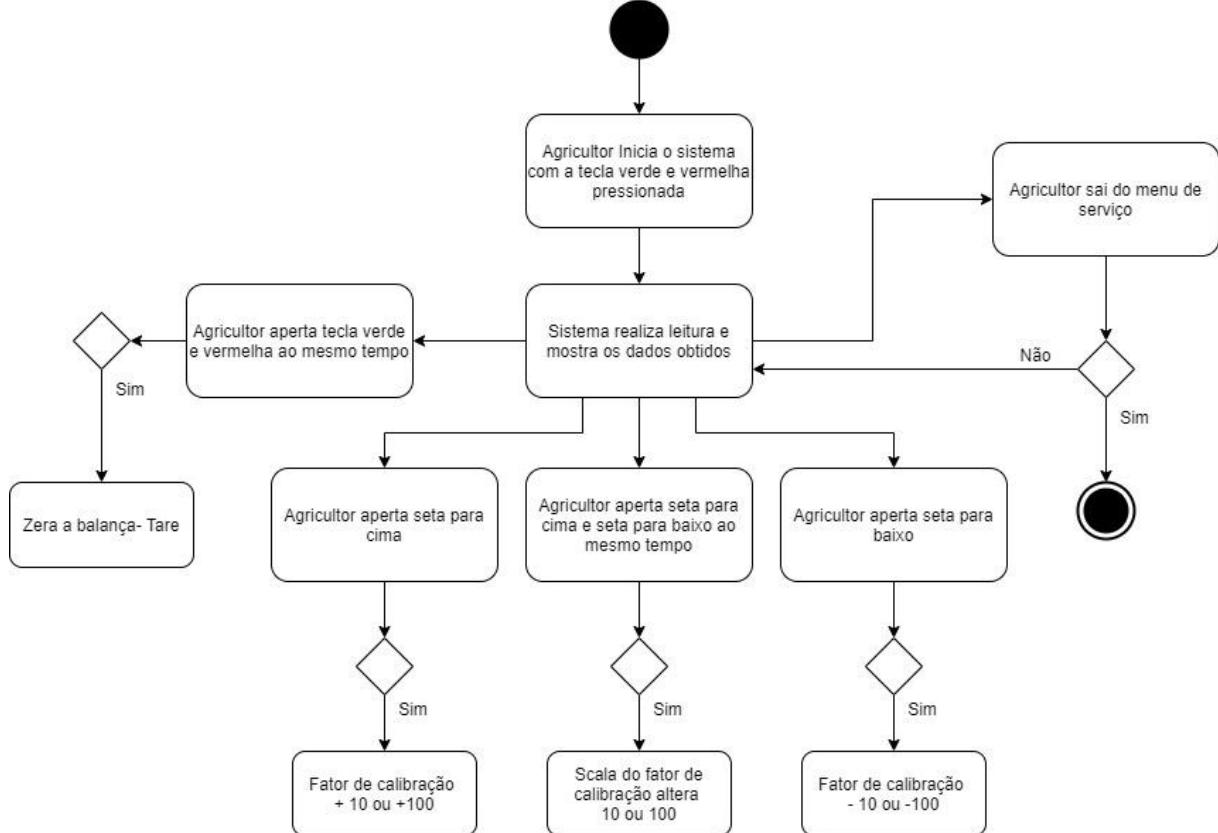
Ilustração 71 – Diagrama de passos para o processo de ordenha



Fonte: Autor (2020)

No processo de calibração do sistema alguns passos são necessários, a ilustração 72 a seguir mostra o diagrama do processo de calibração, explanando os passos e atalhos de teclas necessários para o procedimento.

Ilustração 72 - o diagrama do processo de calibração



Fonte: Autor (2020)

4.4 ANÁLISE DOS RESULTADOS

O teste do protótipo foi ser dividido em duas partes, a primeira medirá a eficiência do sistema e foi realizada em bancada, já o segundo que foi realizado no campo e avaliou a usabilidade do sistema.

4.4.1 Teste de Eficiência

Esse teste foi executado da mesma forma que decorreu o teste no tópico 4.2.4, foi adicionado uma quantidade conhecida de água no recipiente e essa quantidade foi comparada com o valor medido. Para esse teste foi necessário realizar um ajuste na função do firmware

que realiza a captura dos dados da balança, como foi utilizado água a variável da densidade passará de 1,032 para 1,0 ficando condizente com a densidade da água.

Ilustração 73 - Animais cadastrados no sistema

ANIMAIS						
Cod	Nome	Raça	Data Nascimento	Qtd Media por Ordenha	Quantidade Total	Ações
1	Princesa	Jersey	06/05/2016	0 Litros	0 Litros	 
2	Morena	Jersey	08/02/2017	0 Litros	0 Litros	 
3	Mimosa	Jersey	08/08/2017	0 Litros	0 Litros	 

Fonte: Autor (2020)

O primeiro passo foi cadastrar às três vacas no sistema, que foram os mesmos animais ordenhados no teste de campo. Para esse teste a aplicação Web foi utilizada de forma local e ficou hospedado em um computador pessoal. Após cadastrar os animais conforme a Ilustração 73 os seguintes passos foram realizados:

1. Iniciado a ordenha para o animal selecionado;
2. Adicionado a água no recipiente conforme Tabela 9;
3. Finalizado a ordenha;

Esses passos citados acima foram realizados duas vezes para cada animal, simulando as duas ordenhas que são realizadas diariamente. A Tabela 9 demonstra os resultados obtidos nesse processo, sendo que no campo “Quantidade Adicionada” contém a quantidade que foi colocada no recipiente e o campo “Quantidade Medida” refere-se a quantidade que foi medida pelo sistema. Já o campo “Erro %” é o erro em porcentagem que a mediação obteve, esse valor foi obtido utilizando a fórmula $Erro\ em\ \% = \left(\frac{Medição\ do\ sistema - Litros}{Litros} \right) * 100$, citada no capítulo 3.4.4.2.

Tabela 9 - Tabela resultado obtidos na validação de laboratório

Ordenha 1				
Nome	Quantidade Adicionada	Quantidade Medida	Diferença	Erro %
Princesa	4,00	4,10	0,10	2%
Morena	5,00	5,12	0,12	2%
Mimosa	3,50	3,42	-0,08	2%
Ordenha 2				
Nome	Quantidade Adicionada	Quantidade Medida	Diferença	Erro %
Princesa	3,50	3,65	0,15	4%
Morena	4,00	4,12	0,12	3%
Mimosa	3,00	3,06	0,06	2%

Fonte: Autor (2020)

Analizando os dados da tabela 9, é possível observar que o erro máximo do sistema foi de 4%. Sendo esse considerado pelo agricultor um valor baixo, logo que essa porcentagem é equivalente a 0,150L ou 150mL. A Ilustração 74 apresenta a tela do Dashboard após esse processo com os dados já computados, e os resultados expostos nos gráficos.

Ilustração 74 - Tela dashboard com os resultados obtidos no teste de laboratório



Fonte: Autor (2020)

E a ilustração 75 que mostra a lista dos animais após a ordenha, mostrando o valor total ordenhado no animal e a quantidade média por ordenha.

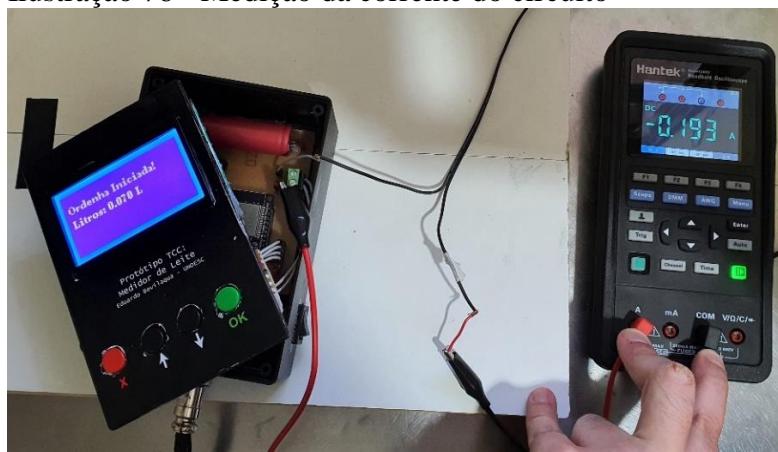
Ilustração 75 - Lista animais com os resultados obtidos no teste de laboratório

ANIMAIS						
Cod	Nome	Raça	Data Nascimento	Qtd Media por Ordenha	Quantidade Total	Ações
1	Princesa	Jersey	06/05/2016	3.8 Litros	7.6 Litros	 
2	Morena	Jersey	08/02/2017	4.585 Litros	9.17 Litros	 
3	Mimosa	Jersey	08/08/2017	3.285 Litros	6.57 Litros	 

Fonte: Autor (2020)

Também foram realizados testes da eficiência do sistema de energia, mais precisamente do MT3608, para esse teste foi realizada a medição da corrente que o sistema necessita para funcionar. Para realizar essa medição foi removido o MT3608 do circuito e no lugar dos pinos Vout+ e Vout- foi adicionado uma fonte de 5V/1A, com isso o sistema de energia com a bateria estava desconectado do circuito e alimentação era suprida pela fonte. Com esse esquema montado foi realizado com o auxílio de um multímetro a medição da corrente que o circuito estava consumindo. Na Ilustração 76 está representado o processo de medição da corrente.

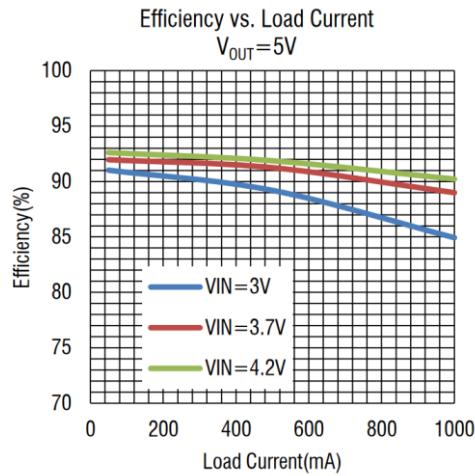
Ilustração 76 - Medição da corrente do circuito



Fonte: Autor (2021)

Durante o processo de medição ou de envio das informações a corrente consumida pelo circuito ficou em torno de 190 mA, com picos máximos de 300 mA. Com o auxílio da Ilustração 77 que apresenta o gráfico de eficiência vs carga do componente MT3608, foi possível obter a eficiência do MT3608, que segundo a tabela é em torno de 92%. Isso considerando a corrente de pico registrada e uma tensão de entrada de 4,2V, representado a bateria totalmente carregada.

Ilustração 77 - Gráfico eficiência MT3608



Fonte: Datasheet “Anexo C”

4.4.2 Teste no Campo

Após todo o sistema validado em bancada foi efetivado um teste prático na propriedade do Senhor José, nesse teste foi realizado a ordenha de três animais no período vespertino. A Ilustração 78 é uma foto tirada na propriedade durante o processo de ordenha.

Ilustração 78 - Teste de campo



Fonte: Autor (2020)

A Aplicação Web ficou hospedada no notebook do autor e uma rede foi criada entre o dispositivo e o notebook, isso, pois a conexão com a internet estava comprometida no dia dos testes. O primeiro passo no local foi apresentar o sistema para o agricultor, mostrando os passos e como deve ser a utilização do sistema e após isso as ordenhas foram iniciadas.

No início da ordenha os tetos eram pendurados no suporte e nesse momento a ordenha era iniciada no sistema. No momento em que a vaca parava de dar leite os tetos eram retirados e pendurados novamente no suporte, nesse momento a ordenha era finalizada no sistema. Esse processo foi realizado com o intuito de amenizar o erro causado pelo peso das teteiras.

No final da ordenha das três vacas o leite foi retirado do tarro e despejado em um balde no qual iria para o resfriador, esse era um balde de 18 litros no qual contia $\frac{3}{4}$ de leite em seu interior. Com o auxílio de uma balança comercial esse balde foi pesado e o resultado foi de 14,5 kg de leite, descontando o peso do balde. Utilizando a equação $Litros = \frac{Peso}{1,032\ Kg/L}$, obtemos que dentro do balde contia um total de 14,05 Litros, foi então realizada uma análise com os dados obtidos no sistema. A Ilustração 79 demonstra a imagem do sistema após as ordenhas.

Ilustração 79 - Lista animais com os dados obtidos no teste de campo

ANIMAIS						
Cod	Nome	Raça	Data Nascimento	Qtd Media por Ordenha	Quantidade Total	Ações
1	Princesa	Jersey	06/05/2016	4.8 Litros	4.8 Litros	 
2	Morena	Jersey	08/02/2017	5.7 Litros	5.7 Litros	 
3	Mimosa	Jersey	08/08/2017	4.2 Litros	4.2 Litros	 

Fonte: Autor (2020)

Somando o total das 3 ordenhas obtemos o total de 14.7 litros, valor próximo da quantidade que estava dentro do recipiente.

Também foi aplicada no produtor a pesquisa disponível no “Apêndice B”, essa com o intuito de avaliar a usabilidade do sistema.

Tabela 10 - Resultado questionário de avaliação

Pergunta	Questionário de Avaliação				
	1	2	3	4	5
O sistema acoplado no tarro possui uma utilização intuitiva e de fácil utilização.				X	
O sistema de medição em um todo (embarcado + Aplicação Web) traz benefícios no controle da produção.					X
As alterações realizadas para utilizar o sistema não afetarão consideravelmente o processo de ordenha.				X	
O sistema de medição em questão é mais atrativo que os sistemas convencionais.				X	
Você usaria o sistema em sua ordenha.				X	

Fonte: Autor (2020)

A Tabela 10 contém as respostas obtidas, que foram quatro respostas em concordo e uma em concordo totalmente. Somando a média das respostas na escala likert foi obtido 4,2. Além da escala likert foi perguntado ao agricultor críticas e sugestões para o projeto e a resposta dele foi:

- O sistema é atrativo, porém ele acredita que se o valor for muito alto não compensa o investimento para um agricultor do porte dele.
- Também comentou que o sistema é atrativo por já disponibilizar os dados e salvar para visualização posterior.
- Por último ele comentou que seria interessante se não houvesse a necessidade do processo de retirar as teteiras e pendurá-las a cada ordenha. Seria mais prático se pudesse tirar de uma vaca e já colocar em outra, iniciando o sistema nesse meio tempo.

5 CONSIDERAÇÕES FINAIS

A proposta desse trabalho era desenvolver um sistema embarcado com a capacidade de medir a quantidade de leite ordenhado por animal em uma ordenha do tipo balde ao pé. Para atingir o objetivo foram realizadas pesquisas bibliográficas e visitas técnicas com o intuito de conhecer o processo de ordenha, para assim poder definir qual a melhor forma para realizar a medição.

Nessas pesquisas foram encontrados alguns modelos existentes de soluções para medições, porém esse contém um custo elevado e são feitos para instalação em sistema de ordenha canalizada. Foi definido então que a melhor forma de medição seria medindo a quantidade no tarro de leite e então, foram selecionados alguns sensores que pudessem ser instalados no recipiente.

Um dos sensores é o sensor laser, esse seria utilizado para medir a altura da coluna de leite que estaria no interior do tarro e a partir dessa altura calcular o volume de leite no interior do tarro. Inicialmente já existiam preocupações sobre o sensor, fato provocado pela espuma gerada pelo leite. Contudo, em testes laboratoriais o sensor se mostrou ineficiente na medição de líquido, sendo inutilizável no sistema em questão.

Outro sensor testado foi o sensor de pressão diferencial, esse usaria o mesmo método do sensor laser para chegar no volume do leite, calculando a altura da coluna. Em testes laboratoriais o sensor mostrou certa eficiência alcançando margem de erro na medição máxima de 10%. Porém, o sensor MPX5100DP que foi o selecionado, não pode ser utilizado diretamente em contato com o líquido e com isso seu uso foi adaptado para a solução, sendo instalado na parte superior do recipiente e realizando a medição através de uma tubulação. Dessa forma o sensor estava medindo a pressão do ar, que possui relação com a temperatura, logo a medição era fortemente afetada com as variações de temperatura do ambiente. O que tornou o sistema inviável.

O último sistema de medição testado foi o com a célula de carga. Esse tem o funcionamento semelhante a uma balança, e para realizar a medição basta ser colocado sob o tarro de leite. A medição ocorre de forma simples e calcula a quantidade de litros de leite considerando a sua densidade e a massa que é o atributo medido com sensor. O sistema se mostrou eficiente nos testes chegando a margens de erro máxima de 4% em testes de laboratório.

Além disso, o sistema dispensa qualquer modificação no tarro de leite, o que possibilita a instalação em diversos modelos de tarro, diminui os custos e não possui o risco de causar

flutuações no vácuo da ordenha. Sendo assim a melhor escolha entre os modelos de medição escolhidos no início do trabalho.

Com o protótipo concluído, todos os requisitos funcionais RF1, RF2, RF3 e RF4 foram atendidos, pois, o sistema pode medir a quantidade de leite no tarro de forma automática, necessitando interação com o usuário apenas para indicar o início e o fim da ordenha. Além de mostrar a quantidade que foi medida e funcionar sem a necessidade de estar conectado a uma rede elétrica ou uma rede de internet cabeada.

Os requisitos não funcionais RNF1, RNF2, RNF3 e RNF4 foram atendidos e comprovados pelo agricultor no momento da validação do sistema em campo. Para RNF1 o sistema mostra em tempo real a quantidade que está sendo medida, exibindo um status da medição. A aprovação dos requisitos RNF2 e RNF3 foi respondida no questionário aplicado ao agricultor, no qual ele respondeu com concordo a pergunta que se referia a utilização do sistema. Com testes o requisito RNF4 foi aprovado, pois, em condições com internet e conexões lentas o sistema não demorou em nem um momento mais de 10 segundos para realizar a leitura dos sensores e envio das informações.

Com relação à questão de pesquisa que é: “O protótipo do sistema eletrônico de medição de leite proposto neste projeto é apto para medir a quantidade de leite em um sistema de ordenha balde ao pé com precisão aceitável?”. Com uma margem de erro de 4% o produtor aprovou o sistema, argumentando que o dispositivo será útil para estimar a produção e identificar variações bruscas nos níveis de leite ordenhados.

Contudo, apesar de o projeto ter alcançado os objetivos algumas melhorias e ajustes podem ser aplicados no sistema. Nos itens abaixo ficam sugestões de melhorias e outras aplicações que podem ser aplicadas em trabalhos futuros:

- Acompanhando as medições com o sistema no teste de campo, foi possível observar um padrão de ordenha. No início da ordenha o animal libera uma grande quantidade de leite, com o decorrer da ordenha essa quantidade reduz e o sistema tende a se estabilizar. Com base no comentário realizado pelo agricultor, que explanou que o sistema seria mais atraente se tivesse uma forma de reduzir os passos necessários na troca de animal, fica como sugestão analisar esse padrão. Com o intuito de tornar o sistema automático, identificando o início e o final da ordenha sem a necessidade de interação nessas etapas.
- Uma melhoria no sistema de energia, adicionando mais uma bateria aumentaria a autonomia do sistema. Além de que com duas baterias ligadas em série, não a necessidade de utilizar o buck-boost, aumentando assim a eficiência energética do dispositivo.

REFERÊNCIAS

AGROLINE (org.). **Guia prático para ordenhar vacas.** Disponível em:
<https://blog.agroline.com.br/guia-pratico-para-ordenhar-vacas/>. Acesso em: 25 maio 2020.

ALBA, Cristiane Brenda Dall'. **RELATÓRIO DE ESTÁGIO DE PRÁTICA PROFISSIONAL SUPERVISIONADO EM MEDICINA VETERINÁRIA NA ÁREA DE BOVINOCULTURA DE LEITE.** 2018. 87 f. TCC (Graduação) - Curso de Medicina Veterinária, Universidade do Oeste de Santa Catarina, Xanxerê, 2018.

ALVES, William Pereira. **Banco de Dados.** São Paulo: érica, 2014a. 161 p

_____. **Desenvolvimento e design de sites.** São Paulo: Érica, 2014b. 161 p.

_____. **Projetos de sistemas web:** conceitos, estruturas, criação de banco de dados e ferramentas de desenvolvimento. São Paulo: Érica, 2015. 168 p.

AMORIM JUNIOR, J P *et al.* MEDIÇÃO REMOTA DE PARÂMETROS ELÉTRICOS USANDO IOT BASEADA NO MICROCONTROLADOR ESP32. **X Encontro Científico de Física Aplicada.** Vitória. Mai. 2019. Disponível em: <http://pdf.blucher.com.br.s3-sa-east-1.amazonaws.com/physicsproceedings/ecfa2019/24.pdf>. Acesso em: 01 jun. 2020.

ANDRADE, Fabrício. **Transmissor de pressão diferencial para medição de nível.** Disponível em: <https://automacaoecartoons.com/2019/02/04/transmissor-de-pressao-diferencial-para-medicao-de-nivel/>. Acesso em: 10 abr. 2020.

ANTUNES, Ana. **O QUE É WIREFRAME E POR QUE UTILIZAR PARA CRIAR UMA IDEIA?** 2019. Disponível em: <https://gobacklog.com/blog/o-que-e-wireframe/>. Acesso em: 14 maio 2020.

AZMI, M. Taujin *et al.* Rancang Bangun Sumber Daya Untuk Charger Baterai Menggunakan Energi Panas Matahari Berbasis Termoelektrik Generator. **SCIENCE ELECTRO**, v. 11, n. 2, 2019. Disponível em: <http://riset.unisma.ac.id/index.php/jte/article/view/3554/3184>. Acesso em: 25 maio 2020.

BASTOS, Alex Vidigal. **Microcontroladores:** curso engenharia de controle e automação. Ouro Preto: Ufop, 2020. 51 slides, color. Disponível em:
http://www.decom.ufop.br/alex/arquivos/sof_bas_ECA/Microcontroladores.pdf. Acesso em: 18 maio 2020.

BETTERRO, Vitor Pereira. **TIPOS DE ORDENHA:** conheça e entenda a diferença entre eles. CONHEÇA E ENTENDA A DIFERENÇA ENTRE ELES. Nutrição e saúde animal. Disponível em: <https://nutricaoesaudeanimal.com.br/tipos-de-ordenha/>. Acesso em: 26 maio 2020.

BRITO, Maria Aparecida *et al.* **Composição.** Embrapa. Disponível em:
https://www.agencia.cnptia.embrapa.br/Agencia8/AG01/arvore/AG01_128_21720039243.html#:~:text=Ag%C3%Aancia%20de%20Informa%C3%A7%C3%A3o%20Embrapa%20Com

posi%C3%A7%C3%A3o&text=O%20leite%20%C3%A9%20uma%20combina%C3%A7%C3%A3o,prote%C3%ADnas%2C%20sais%20minerais%20e%20vitaminas. Acesso em: 25 maio 2020.

BRITO, Maria Aparecida et al. Densidade Relativa. Embrapa. Disponível em: https://www.agencia.cnptia.embrapa.br/Agencia8/AG01/arvore/AG01_196_21720039246.html#:~:text=A%20densidade%20do%20leite%20%C3%A9,leite%20desnatado%2C%20cerca%20de%201%2C035.. Acesso em: 29 maio 2020.

CARDOSO, Virgínia; CARDOSO, Giselle. Sistema de banco de dados: uma abordagem introdutoria e aplicada. São Paulo: Saraiva, 2012. 144 p.

CARVALHO, Armando da Costa. Princípios básicos para ter sucesso com Ordenhadeira Mecânica. Juiz de Fora: Embrapa Gado de Leite, 2020. 101 slides, color. Disponível em: http://ead.senar.org.br/wp-content/uploads/capacitacoes_conteudos/bovino_cultura_de_leite/CURSO_4_SQL/AULA_6_PRINCIPIOS_BASICOS_SUCESSO_ORDENHADEIRA_MECANICA.pdf. Acesso em: 28 maio 2020.

CÉSAR, Janaína; PAOLI, Marco-Aurélio de; ANDRADE, João Carlos de. A determinação da densidade de sólidos e líquidos. **Revista Chemkeys**, [S.L.], n. 7, p. 1-8, 17 set. 2018. Universidade Estadual de Campinas. <http://dx.doi.org/10.20396/chemkeys.v0i7.9618>. Disponível em: <https://econtents.bc.unicamp.br/inpec/index.php/chemkeys/article/view/9618/5038>. Acesso em: 14 maio 2021.

D'AVILLAR, Priscila. Você sabe o que são Frameworks? E para o que servem? 2019. Dinamica. Disponível em: <https://dinamicatreinamentos.com/blog/carreira/o-que-sao-frameworks/>. Acesso em: 28 maio 2020.

DEBONA, Darci. Quantidade de produtores que vendem leite tem queda de 41% em 12 anos em Santa Catarina. 2019. Disponível em: <https://www.nsctotal.com.br/noticias/quantidade-de-produtores-que-vendem-leite-tem-queda-de-41-em-12-anos-em-santa-catarina>. Acesso em: 09 jun. 2021.

DELAI, Andre Luiz. Sistemas Embarcados: a computação invisível. a computação invisível. 2013. Disponível em: <https://www.hardware.com.br/artigos/sistemas-embarcados-computacao-invisivel/conceito.html>. Acesso em: 21 abr. 2020.

DEVMEDIA. Ciclos de Vida do Software. Disponível em: <https://www.devmedia.com.br/ciclos-de-vida-do-software/21099>. Acesso em: 20 abr. 2020.

DUTRA, Jesse Willian Andretta. DESENVOLVIMENTO DE UM PROTÓTIPO DE DISPOSITIVO PARA CONTAGEM DE VEÍCULOS EM VIAS URBANAS. 2018. 48 f. TCC (Graduação) - Curso de Engenharia de Transportes e Logística, Universidade Federal de Santa Catarina, Joinville, 2018. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/188103/TCC_JESSE_WILLIAN_ANDRETTA_DUTRA.pdf?sequence=1&isAllowed=y. Acesso em: 25 maio 2020.

EMBARCADOS (org.). **Sistema Embocado:** o que é? qual a sua importância? O que é? Qual a sua importância? Disponível em: <https://www.embarcados.com.br/sistema-embocado/>. Acesso em: 14 maio 2020.

FALLEIRO, Fellipe Diogo. **CONVERSOR ANALÓGICO-DIGITAL COM CAPACITORES MÍNIMOS INTEGRADO NA TECNOLOGIA CMOS.** 2015. 89 f. TCC (Graduação) - Curso de Engenharia Eletrônica e de Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

FERNANDO K (org.). **ESP32: Detalhes internos e pinagem.** 2018. Disponível em: <https://www.fernandok.com/2018/03/esp32-detalhes-internos-e-pinagem.html>. Acesso em: 05 jun. 2020.

FLANAGAN, David. **JavaScript:** o guia definitivo. 6. ed. Porto Alegre: Bookman, 2011. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=zWNyDgAAQBAJ&oi=fnd&pg=PR1&dq=javascript&ots=IzDdw0H5dJ&sig=u82HK7pZQDinnDGsvBcTWqkwDV4#v=onepage&q=javascript&f=false>. Acesso em: 24 maio 2020.

FLATSCHART, Fábio. **HTML5:** embarque imediato. Rio de Janeiro: Brasport, 2011. Disponível em: https://books.google.com.br/books?id=_cgsCgAAQBAJ&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. Acesso em: 04 jun. 2020.

FLAUSINO, Maiara Guimarães. **SysML para engenharia simultânea de sistemas espaciais:** Relatório final de projeto de iniciação científica. São José dos Campos: Ufscar, 2013. 183 p. Disponível em: <<http://mtc-m16d.sid.inpe.br/col/sid.inpe.br/mtc-m19/2013/09.04.11.33.58/doc/Maiara%20Flausino.pdf?metadatarepository=sid.inpe.br/mtc-m19/2013/09.04.11.33.59&mirror=sid.inpe.br/mtc-m19@80/2009/08.21.17.02.53>>. Acesso em: 08 maio. 2020.

FLORES, Maria da Glória Cataldi Couto. **Teste embarcado de conversores analógico-digitais.** 2003. 135 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003. Disponível em: <https://www.lume.ufrgs.br/handle/10183/3508>. Acesso em: 05 jun. 2020.

FONSECA, João José Saraiva da. **Metodologia da pesquisa científica.** Fortaleza: Uece, 2002. 127 p.

FORISCHI, Felipe de Souza; SOUZA, Renato Aparecido de. **SISTEMA ELETRÔNICO PARA ESTUDO DE CARGA E DESCARGA EM BATERIAS DE ÍONS DE LÍTIO.** 2015. 113 f. TCC (Graduação) - Curso de Engenharia Eletrica, Universidade Federal do ParanÁ, Curitiba, 2015. Disponível em: <http://www.eletrica.ufpr.br/p/arquivostccs/353.pdf>. Acesso em: 25 maio 2020.

FORNELL (org.). **SENSORES DE PRESSÃO (Linha MPX).** Disponível em: <https://blog.fornell.com.br/2020/02/14/sensores-de-pressao-linha-mpx/>. Acesso em: 27 maio 2020.

FRANKENTHAL, Rafaela. **Entenda a escala Likert e como aplicá-la em sua pesquisa.** Disponível em: <https://mindminers.com/blog/entenda-o-que-e-escala-likert/>. Acesso em: 15 jun. 2020.

FREESCALE SEMICONDUCTOR (org.). **MPX5500 Series.** 2009. DataSheet. Disponível em: <https://www.nxp.com/docs/en/data-sheet/MPX5100.pdf>. Acesso em: 15 maio 2020.

FRITZING. **About.** Disponível em: <https://fritzing.org/about/>. Acesso em: 02 maio 2020.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social.** 6. ed. São Paulo: Editora Atlas S.a., 2008. 220 p.

GOMES, Andreia Vaz; AMARAL, Elessandra Martins de Souza; PRADO, Rogério Junqueira. Determinação da densidade de líquidos imiscíveis pelo princípio de Stevin. **Revista Brasileira de Ensino de Física**, São Paulo, v. 41, n. 3, p. 1-1, jan. 2019. Disponível em: <https://doi.org/10.1590/1806-9126-rbef-2018-0313>. Acesso em: 16 maio 2020.

GOUVEIA, Rosimar. **Volume do Cilindro.** Disponível em: <https://www.todamateria.com.br/volume-do-cilindro/>. Acesso em: 15 maio 2020.

IMPLEMIS (org.). **Ordenhadeira Móvel Implemis Com 1 Conjunto E 1 Tarro.** Disponível em: <https://implemis.com.br/catalogo/ordenhadeira-movel-implemis-com-1-conjunto-e-1-tarro/>. Acesso em: 29 maio 2020.

JESUS JUNIOR, Bruno Ferreira; SEGUNDO, Alan Kardek Rêgo. Sistema de Aquisição de Dados de Extensometria Aplicado a um Tambor Descascador de Toras de Madeira.

JOCHIMS, Felipe *et al.* O leite para o Oeste Catarinense. **Agropecuária Catarinense**, Florianópolis, v. 29, n. 3, p. 18-21, set. 2016. Disponível em: <http://publicacoes.epagri.sc.gov.br/index.php/RAC/article/download/67/44>. Acesso em: 07 abr. 2020.

JUNIO, José Carlos de Souza; PAIXÃO, Renato Rodrigues. **Circuitos Eletroeletrônicos:** fundamentos e desenvolvimento de projetos logicos. São Paulo: érica, 2014. 153 p.

LAMB, Frank. **Automação industrial na prática.** Porto Alegre: Amgh, 2015. 361 p.

LEDUR, Cleverson Lopes *et al.* **PROGRAMAÇÃO BACK END II.** Porto Alegre: Sagah EducaÇÃo, 2019. 220 p

LI, Xin et al. Design and Realization of Intelligent Eyeshade Based on STC89C58RD+. In: **2nd International Conference on Civil, Materials and Environmental Sciences**. Atlantis Press, 2015. Disponível em: <https://www.atlantis-press.com/proceedings/cmes-15/18153>. Acesso em: 25 maio 2020.

LOCATELLI, Caroline. **BALANÇA COM CÉLULA DE CARGA E HX711.** 2019. Disponível em: <https://www.curtocircuito.com.br/blog/Categoria%20Arduino/balanca-com-celula-de-carga-e-hx711>. Acesso em: 5 out. 2020.

LUCIDCHART. **O que é um diagrama entidade relacionamento?** Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento>. Acesso em: 05 maio 2020.

MACHADO, Felipe Nery Rodrigues. **Banco de Dados:** projeto e implementação. 3. ed. São Paulo: érica, 2014. 401 p.

MENAIA, Celso Miguel de Carvalho. **Desenvolvimento do sistema de protecção,monitorização e gestão de baterias de iões de lítio do tipo LiFePO4 para aplicação em veículos eléctricos.** 2013. 211 f. Dissertação (Mestrado) - Curso de Engenharia Electrotécnica, Instituto Superior de Engenharia de Lisboa, Lisboa, 2013.

MICROCONTROLLERS LAB. **ADS1115 I2C external ADC with ESP32 in Arduino IDE.** Disponível em: <https://microcontrollerslab.com/ads1115-external-adc-with-esp32/>. Acesso em: 10 nov. 2020.

MILETTO, Evandro Manara; BERTAGNOLLI, Silvia de Castro. **Desenvolvimento de software II:** introdução ao desenvolvimento web com html, css, javascript e php. Porto Alegre: Bookman Companhia Editora, 2014. 276 p.

MILKPARTS (org.). **Ordenhadeira Balde ao Pé.** Disponível em: <https://www.milkparts.com.br/produtos/ordenhadeira-balde-ao-pe>. Acesso em: 28 maio 2020. MORAIS, Dieime de. **AUTOMATIZAÇÃO DO CADASTRO DE CONTROLE DE ACESSO A REDE. ESTUDO DE CASO:** ceo udesc. 2016. 83 f. TCC (Graduação) - Curso de Sistema de Informações, Unoesc, Chapecó, 2016.

MORISHITA, André Hiroki; CHIYO, Daniel Eiji Martins; COSTA, Vinícius Jerez. **Prototipagem de placas de circuito impresso por método térmico e fotossensível.** 2018. 14 f. Relatório Final - Curso de Departamento de Engenharia Elétrica e de Computação, Universidade de São Paulo, São Carlos, 2018. Disponível em: https://edisciplinas.usp.br/pluginfile.php/4843396/mod_resource/content/1/Relatorio_Lab_projetos_ver2.pdf. Acesso em: 10 maio 2020.

MOTA, Ricardo Fábio da Silva. **Sensor de distância por infravermelhos para a caracterização do espaço de trabalho.** 2018. 124 f. Dissertação (Mestrado) - Curso de Engenharia Eletrotécnica e de Computadores, Universidade Nova de Lisboa, Lisboa, 2018. Disponível em: https://run.unl.pt/bitstream/10362/71025/1/Mota_2018.pdf. Acesso em: 24 maio 2020.

MURTA, Gustavo. **Conhecendo o ESP32:** introdução. Introdução. 2018. Disponível em: <https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1>. Acesso em: 20 maio 2020.

NATALE, Ferdinando. **Automação Industrial.** 10. ed. São Paulo: Érica, 2008. 257 p.

OLIVEIRA, André Schneider de; ANDRADE, Fernando Souza de. **Sistemas Embarcados:** hardware e firmware na prática. 2. ed. São Paulo: Érica, 2010. 320 p.

OLIVEIRA, Luis Fernando Tividini; SILVA, Sandro Pereira. Mudanças institucionais e produção familiar na cadeia produtiva do leite no Oeste Catarinense. **Revista de Economia e Sociologia Rural**, [s.l.], v. 50, n. 4, p. 705-720, dez. 2012. FapUNIFESP (SciELO).

<http://dx.doi.org/10.1590/s0103-20032012000400007>. Disponível em:
https://www.scielo.br/scielo.php?pid=S0103-20032012000400007&script=sci_arttext. Acesso em: 10 abr. 2020.

OMGSYML. WHAT IS SYSML? Disponível em: <http://www.omgysml.org/what-is-sysml.htm>. Acesso em: 28 abr. 2020.

PAIXÃO, Elisiane Pelke; SAUTHIER, Luis Fernando; REIMBOLD, Manuel Martin Pérez. Ensaio de células de carga. **Salão do Conhecimento**, 2017.

PATSKO, Luís Fernando. Tutorial–aplicações, funcionamento e utilização de sensores. **Maxwell Bohr: Instrumentação Eletrônica**, 2006.

PICCOLI, Jeferson Luiz. **Sistema Balde ao pé**: começando a conhecer os tipos de ordenhadeiras! 2019. Canal do Leite. Disponível em:
<https://canaldoleite.com/columnas/jeferson-luiz-piccoli/sistema-balde-ao-pe-comecando-a-conhecer-os-tipos-de-ordenhadeiras/>. Acesso em: 26 maio 2020.

PISA, Pedro. **O que é e como usar o MySQL?** 2012. TechTudo. Disponível em:
<https://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>. Acesso em: 04 jun. 2020.

PRESSMAN, Roger S.; MAXIM, Bruce R.. **Engenharia de Software**: uma abordagem profissional. 8. ed. Porto Alegre: Amgh, 2016. 940 p.

PUCPR (org.). **MEDIÇÃO DE NÍVEL**. Disponível em:
<https://aiecp.files.wordpress.com/2012/07/3-2-pc3b3s-pucpr-3-medic3a7c3a3o-de-nc3advel.pdf>. Acesso em: 10 maio 2020.

REIS, Fábio dos. **Introdução aos Microcontroladores**. 2015. Eletrônica Geral. Disponível em: <http://www.bosontreinamentos.com.br/elettronica/elettronica-geral/introducao-aos-microcontroladores/>. Acesso em: 17 maio 2020.

RESENDE, João Cesar de *et al.* Produção e produtividade de leite no mundo. **Anuário Leite 2019**, Juiz de Fora, v. 2, p. 46-48, jan. 2019. Anual. Disponível em:
<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/198698/1/Anuario-LEITE-2019.pdf>. Acesso em: 10 abr. 2020.

RIBEIRO, Marlice Teixeira; CARVALHO, Armando da Costa. **Fisiologia da produção de leite no Úbere**. Embrapa. Disponível em: <https://ruralpecuaria.com.br/tecnologia-e-manejo/bovinocultura-de-leite/fisiologia-da-producao-de-leite-no-ubere.html>. Acesso em: 25 maio 2020.

RODRIGUES, Loraine Gomes et al. Bacia Leiteira do Extremo Oeste de Santa Catarina: Estudo do Rebanho, Manejo e Gestão visando a Qualidade do Leite em Unidades Produtoras. In: **Revista do Congresso Sul Brasileiro de Engenharia de Alimentos**. 2015.

ROURE, Marcel de. **Medição de Nível – Tudo para você virar um especialista no assunto**. 2018. Disponível em: <https://instrumentacaoecontrole.com.br/tudo-sobre-medicao-de-nivel/>. Acesso em: 08 abr. 2020.

SANTANA, Ricardo Henrique Cavalcanti de. **FURADEIRA/PARAFUSADEIRA, À BATERIA, COM DISPOSITIVO ÓPTICO DE AUTO DESLIGAMENTO, LIMITADOR DE PROFUNDIDADE DE CORTE.** 2018. 83 f. TCC (Graduação) - Curso de Engenharia Elétrica, Centro Universitário Augusto Motta, Rio de Janeiro, 2018.

Disponível em:

https://www.researchgate.net/profile/Andre_Pinheiro21/publication/338178916_Furadeira_Parafusadeira_a_Bateria_com_Dispositivo_Optico_de_Auto_Desligamento_Limitador_de_Profundidade_de_Corte/links/5e054947a6fdcc2837415cb1/Furadeira-Parafusadeira-a-Bateria-com-Dispositivo-Optico-de-Auto-Desligamento-Limitador-de-Profundidade-de-Corte.pdf. Acesso em: 21 maio 2020.

SANTA CATARINA: Secretaria de Estado da Agricultura, da Pesca e do Desenvolvimento Rural. **Agricultura familiar responde por metade do faturamento da agropecuária catarinense.** 2019. Disponível em:

<https://www.sebrae.com.br/sites/PortalSebrae/ufs/sc/noticias/os-avancos-da-cadeia-produtiva-do-leite-no-oeste-catarinense,dc5119f55cd89510VgnVCM1000004c00210aRCRD>. Acesso em: 22 abr. 2021.

SARAIVA, Maurício de Oliveira; BARRETO, Jeanine dos Santos. **DESENVOLVIMENTO DE SISTEMAS COM PHP.** Porto Alegre: Sagah, 2018.

SCHERMANN, Daniela. **Escala de Likert:** como usar a pergunta de escala no seu questionário de pesquisa. como usar a pergunta de escala no seu questionário de pesquisa. 2019. Opinionbox. Disponível em: <https://blog.opinionbox.com/pergunta-de-escala-ou-escala-de-likert/>. Acesso em: 15 jun. 2020.

SEBRAE (Santa Catarina). **Os avanços da cadeia produtiva do leite no oeste catarinense.** Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/ufs/sc/noticias/os-avancos-da-cadeia-produtiva-do-leite-no-oeste-catarinense,dc5119f55cd89510VgnVCM1000004c00210aRCRD>. Acesso em: 22 abr. 2021.

SEBRAE-SP (org.). **Pesquisa Setor/Segmento Agropecuário de Leite:** São Paulo: Sebrae-sp, 2016. 56 slides, color. Disponível em:
<https://m.sebrae.com.br/Sebrae/Portal%20Sebrae/UFs/SP/Pesquisas/Agropecua%CC%81ria%20de%20Leite.pdf>. Acesso em: 26 mar. 2020.

SENAR (org.). **LEITE:** ordenha manual de bovinos. 5. ed. Brasília: Coleção Senar, 2016. 72 p. Disponível em: <https://www.cnabrasil.org.br/assets/arquivos/134-LEITE.pdf>. Acesso em: 16 maio 2020.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação.** 4. ed. Florianópolis: Ufsc, 2005. 139 p.

SILVA, Elaine Quintino da; MOREIRA, Dilvan de Abreu. Um Framework de Componentes para o Desenvolvimento de Aplicações Web Robustas de Apoio à Educação. **Xv Simpósio Brasileiro de Informática na Educação,** São Paulo, p. 158-167, jan. 2004. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/317/303>. Acesso em: 04 maio 2020.

- SILVA, Hugo. **Medição de nível em tanques pela pressão diferencial.** WIKA do Brasil. Disponível em:
<http://www.profibus.org.br/artigos/Medicao+de+nivel+em+tanques+pela+pressao+diferencial.html>. Acesso em: 11 maio 2020.
- SILVA, Marco Antonio da *et al.* Sistema para monitoramento de vibrações utilizando Arduino. **Revista Interatividade**, v. 2, n. 2, p. 142-153, 2014. Disponível em:
<http://www.firb.br/editora/index.php/interatividade/article/view/144>. Acesso em: 02 jun. 2020.
- SILVA, Patrícia Braga. **CARACTERIZAÇÃO DA QUALIDADE DO LEITE EM PROPRIEDADES NA MICRORREGIÃO DE SÃO JOÃO DEL -REI -MG Silva.** 2014. 59 f. TCC (Graduação) - Curso de Bacharelado em Zootecnia, Universidade Federal de São João del Rei, São João del Rei, 2014. Disponível em: https://ufsjiang.edu.br/portal-repositorio/File/cozoo/TCC/2014-1/TCC_PatriciaBragaSilva.pdf. Acesso em: 29 maio 2020.
- SILVA JÚNIOR, Andouglas Gonçalves da. **Pressão nos Fluidos.** Natal: Instituto Federal do Rio Grande do Norte, 2016. 16 slides, color. Disponível em:
<https://docente.ifrn.edu.br/andouglassilva/disciplinas/mecanica-dos-fluidos/aula-2-pressao-teorema-de-stevin/view>. Acesso em: 17 maio 2020.
- SILVEIRA, Cristiano Bertulucci. **Sensor de Nível:** 10 tipos diferentes para aplicar na indústria. 10 Tipos Diferentes para Aplicar na Indústria. Disponível em:
<https://www.citisystems.com.br/sensor-de-nivel/>. Acesso em: 1 abr. 2020.
- SIQUEIRA, Kenny B. Consumo de leite e derivados no Brasil. **Anuário Leite 2019**, Juiz de Fora, v. 2, p. 24-25, jan. 2019. Anual. Disponível em:
<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/198698/1/Anuario-LEITE-2019.pdf>. Acesso em: 10 abr. 2020.
- SOARES, Frederico Aécio Carvalho. **COMPOSIÇÃO DO LEITE:** fatores que alteram a qualidade química. 2013. 7 f. Pós-Graduação- Curso de Ciências Veterinárias, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013. Disponível em:
<https://www.ufrgs.br/lacvet/site/wp-content/uploads/2013/10/leiteFred.pdf>. Acesso em: 10 maio 2020.
- SOMMERVILLE, Ian. **Engenharia de Software.** 9. ed. São Paulo: Pearson Prentice Hall, 2011. 548 p.
- SOUZA, Fábio. **Aprenda interpretar um diagrama esquemático.** 2016. Disponível em:
<https://www.embarcados.com.br/interpretar-um-diagrama-esquematico/>. Acesso em: 05 maio 2020.
- SOUZA, Ivan de. **Framework:** descubra o que é, para que serve e por que você precisa de um para o seu site. descubra o que é, para que serve e por que você precisa de um para o seu site. 2019. Disponível em: <https://rockcontent.com/blog/framework/>. Acesso em: 17 maio 2020.
- SPARKFUN. **Load Sensor - 50kg (Generic).** Disponível em:
<https://www.sparkfun.com/products/10245>. Acesso em: 28 maio 2021.

SURVEYMONKEY (org.). **Calculadora de margem de erro.** Disponível em: <https://pt.surveymonkey.com/mp/margin-of-error-calculator/>. Acesso em: 14 jun. 2020.

SYML.ORG. **What is a SysML Block Definition diagram?** Disponível em: <https://sysml.org/sysml-faq/what-is-block-definition-diagram.html>. Acesso em: 08 maio 2020.

SYML.ORG. **What is a SysML Use Case diagram?** Disponível em: <https://sysml.org/sysml-faq/what-is-use-case-diagram.html>. Acesso em: 08 maio 2020.

TEIXEIRA, Gustavo. **PROJETO COM SENSOR DE DISTÂNCIA VL53L0X E ARDUINO PARA ALTA PRECISÃO.** 2019. Disponível em: <https://www.usinainfo.com.br/blog/projeto-com-sensor-de-distancia-vl53l0x-e-arduino-para-alta-precisao/>. Acesso em: 27 maio 2020.

THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano Braga de. **Sensores Industriais:** fundamentos e aplicações. 8. ed. São Paulo: Érica, 2011. 224 p.

THOMSEN, Adilson. **Tutorial:** acelerômetro mpu6050 com arduino. Acelerômetro MPU6050 com Arduino. 2014. Disponível em: <https://www.filipeflop.com/blog/tutorial-acelerometro-mpu6050-arduino/>. Acesso em: 27 maio 2020.

TRIPATHI, Abhishek *et al.* Solar and Wind Hybrid Power Generation. **Ijsart**, Pune, v. 4, n. 5, p. 864-868, maio 2018. Disponível em: http://bvucoepune.edu.in/wp-content/uploads/2018/BVUCOEP-DATA/Research_Publications/2017_18/37.pdf. Acesso em: 25 maio 2020.

QUINTÃO, Leonardo Cotta et al. EFICIÊNCIA DE MEDIDORES VOLUMÉTRICOS DO SISTEMA DE ORDENHA MECÂNICA. **Anais Iii Simpac**, Viçosa-mg, v. 3, n. 1, p. 206-212, dez. 2011. Anual.

VARSHNEY, Vikas et al. Solar Powered Smart Bag. **Proceeding of the 12th INDIACom. New Delhi: Bharati Vidyapeeth's Institute of Computer Applications and Management**, 2018. Disponível em: https://www.researchgate.net/profile/Vikas_Varshney4/publication/324683398_Solar_Powered_Smart_Bag/links/5adbe3e3458515c60f5e07dd/Solar-Powered-Smart-Bag.pdf. Acesso em: 25 maio 2020.

VASCONCELLOS, Rodrigo Durães de. **PROJETO DE UM CONVERSOR ANALÓGICO/DIGITAL POR APROXIMAÇÕES SUCESSIVAS DE 12 BITS.** 2011. 120 f. TCC (Graduação) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, 2011. Disponível em: https://repositorio.ufmg.br/bitstream/1843/BUOS-8UJJ7V/1/disserta_o_rodrigo_dur_es.pdf. Acesso em: 21 maio 2021.

VERONESI, Ricardo Luís Martins. **RTRASSOC51:** módulo de comunicação i2c reconfigurável ri2c. 2005. 115 f. Dissertação (Mestrado) - Curso de CiÊncia da ComputaÇÃo, Centro Universitário Eurípides de Marília, Marília, 2005. Disponível em: <https://aberto.univem.edu.br/bitstream/handle/11077/297/RtrASSoc51%20%e2%80%93%20M%c3%b3dulo%20de%20Comunica%c3%a7%C3%a3o%20I2C%20Reconfigur%c3%a1vel%20%e2%80%93%20rI2C.pdf?sequence=1&isAllowed=y>. Acesso em: 07 jun. 2020.

XAVIER, Thiago. **O que é HTML e qual sua funcionalidade?** 2019. Disponível em: <https://rockcontent.com/blog/html/>. Acesso em: 04 jun. 2020.

WATTIAUX, Michel A.. A MÁQUINA DE ORDENHA. **Essenciais em Gado de Leite: Lactação e Ordenha.** A, p. 85-88. 2014. Disponível em: https://federated.kb.wisc.edu/images/group226/52752/19-25/de_22.pt.pdf. Acesso em: 27 maio 2020.

ZANELA, Maira Balbinotti *et al.* **Manejo de Ordenha.** Pelotas: Embrapa Clima Temperado, 2011. 21 p. Disponível em: <https://www.infoteca.cnptia.embrapa.br/bitstream/doc/934943/1/Documento342.pdf>. Acesso em: 04 maio 2020.

ZANINELLI, Mauro; TANGORRA, Francesco Maria. Development and testing of a “free-flow” conductimetric milk meter. **Computers And Electronics In Agriculture**, [s.l.], v. 57, n. 2, p. 166-176, jul. 2007. Elsevier BV. <http://dx.doi.org/10.1016/j.compag.2007.03.004>.

ZOCCAL, Rosangela. Leite nas grandes regiões brasileiras. **Anuário Leite 2019**, Juiz de Fora, v. 2, p. 52-59, jan. 2019. Anual. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/198698/1/Anuario-LEITE-2019.pdf>. Acesso em: 10 abr. 2020.

APÊNDICES E ANEXOS

Nesse capítulo serão apresentados os Apêndices e anexos, esses são documentos importantes para o desenvolvimento do trabalho, como entrevistas e pesquisas realizadas.

APÊNDICE A – PESQUISA-ENTREVISTA COM AGRICULTOR

Pesquisa realizada pelo Google Formulários com um produtor de leite da região, com o objetivo de conhecer sua propriedade, conhecer seus processos e identificar suas necessidades.

1. Seu nome?

Resposta: José Antonio Riva

2. Em que cidade a propriedade está localizada?

Resposta: Linha Marcon - Interior de Chapecó, SC

3. Quantas vacas a propriedade possui?

Resposta: 8 vacas

4. Quantos litros de leite são produzidos por mês?

Resposta: 3700

5. Qual tipo de ordenha a propriedade utiliza?

Resposta: Ordenha mecânica - Balde ao pé

6. É realizado algum tipo de medição na quantidade de leite produzido?

Resposta: Não

7. Atualmente é realizado algum controle na produção leiteira?

Resposta: Nem um controle é realizado na propriedade, quem fornece a quantidade de leite produzido é o laticínio e com base nisso é realizada apenas uma estimativa dos custos e receitas geradas pelo leite. Assim sabemos se deu lucro ou prejuízo.

8. Atualmente é realizado algum controle individual na produção de leite por animal?

Resposta: Não, apenas visualmente olhando no taro de leite eu avalio se uma vaca aumentou ou diminuiu a produção.

9. É interessante para uma propriedade desse porte adquirir e utilizar um sistema de medição eletrônico?

Resposta: Desde que o sistema seja de fácil utilização e possui um baixo custo, sim eu usaria.

APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO PROJETO

1. O sistema acoplado no tarro possui uma utilização intuitiva e de fácil utilização.

1-Discordo Totalmente	2-Discordo	3-Neutro	4-Concordo	5-Concordo Totalmente

2. O sistema de medição em um todo (embarcado + Aplicação Web) traz benefícios no controle da produção.

1-Discordo Totalmente	2-Discordo	3-Neutro	4-Concordo	5-Concordo Totalmente

3. As alterações realizadas para utilizar o sistema não afetarão consideravelmente o processo de ordenha.

1-Discordo Totalmente	2-Discordo	3-Neutro	4-Concordo	5-Concordo Totalmente

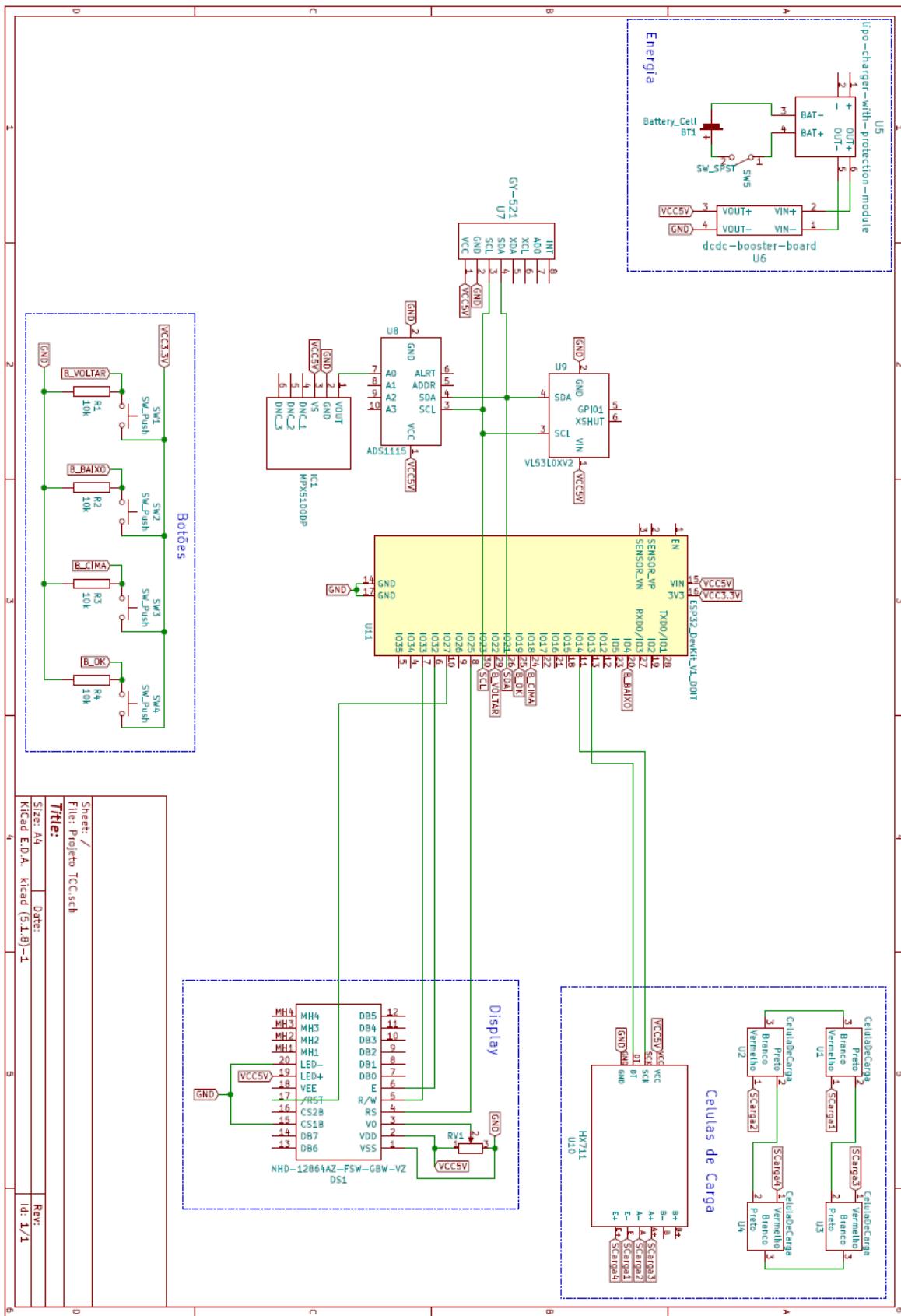
4. O sistema de medição em questão é mais atrativo que os sistemas convencionais.

1-Discordo Totalmente	2-Discordo	3-Neutro	4-Concordo	5-Concordo Totalmente

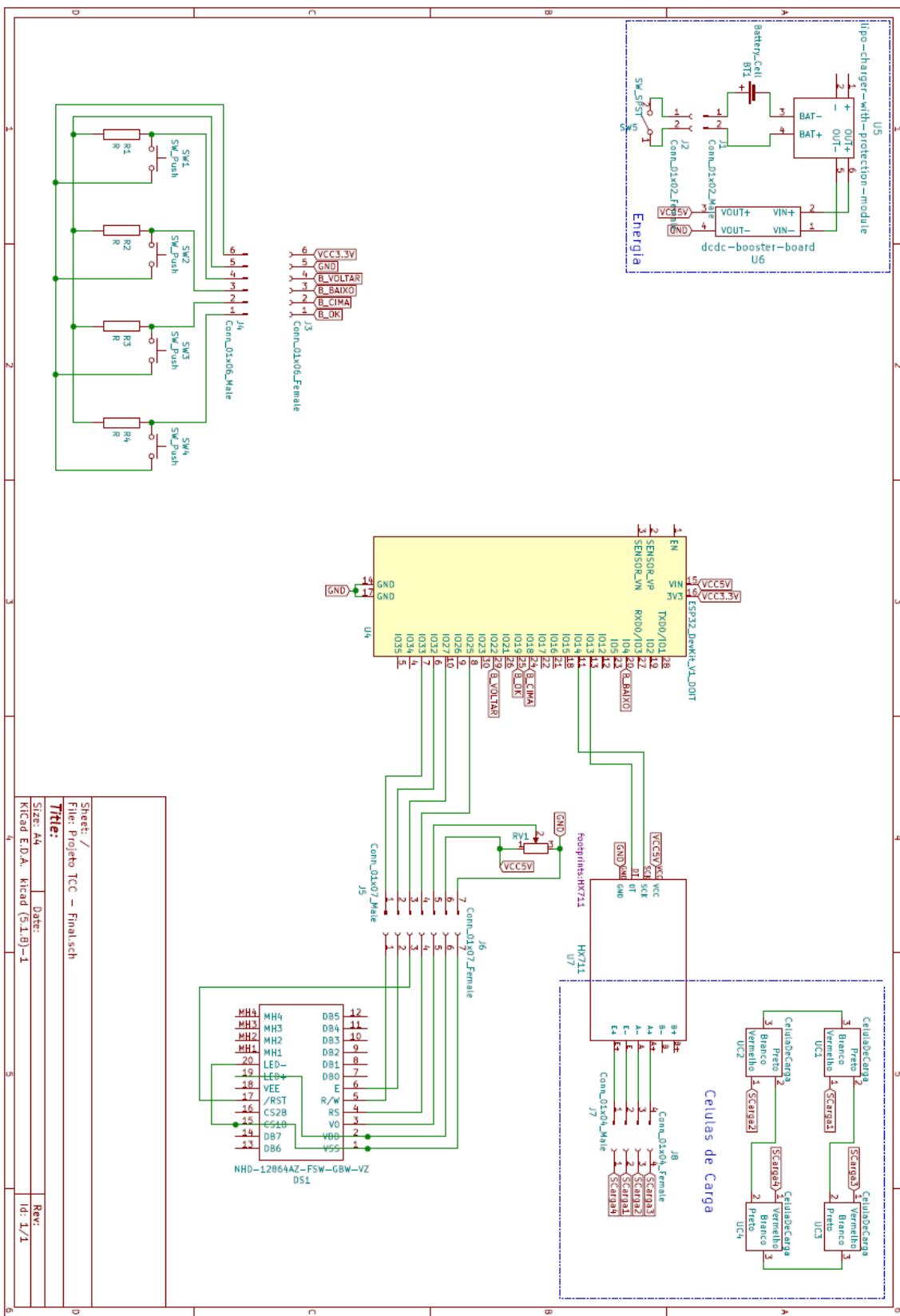
5. Você usaria o sistema em sua ordenha.

1-Discordo Totalmente	2-Discordo	3-Neutro	4-Concordo	5-Concordo Totalmente

APÊNDICE C – DIAGRAMA ELETRICO CIRCUITO COMPLETO



APÊNDICE D – DIAGRAMA ELETRICO CIRCUITO FINAL



ANEXO A – PARTES DATASHEET MPX5100DP

NXP Semiconductors
Data Sheet: Technical Data

Document Number: MPX5100
Rev. 14, 12/2018



MPX5100, 0 to 100 kPa, Differential, Gauge, and Absolute, Integrated, Pressure Sensors

The MPX5100 series piezoresistive transducer is a state-of-the-art monolithic silicon pressure sensor designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This patented, single element transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high-level, analog output signal that is proportional to the applied pressure.

Features

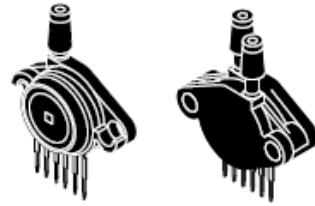
- 2.5% maximum error over 0 to 85 °C
- Ideally suited for microprocessor or microcontroller-based systems
- Patented silicon shear stress strain gauge
- Available in absolute, differential and gauge configuration
- Durable epoxy unibody element
- Easy-to-use chip carrier option

Typical applications

- Patient monitoring
- Process control
- Pump/motor control
- Pressure switching
- White goods

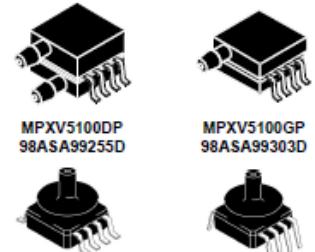
MPX5100

Unibody packages



MPX5100AP/GP
98ASB42796B MPX5100DP
98ASA42797B

Small outline packages



MPXV5100DP
98ASA99255D MPXV5100GP
98ASA99303D
MPXV5100GC6U
98ASB17757C MPXV5100GC7U
98ASB17759C

Ordering information

Device name	Shipping	Package	# of Ports			Pressure type			Device marking
			None	Single	Dual	Gauge	Differential	Absolute	
Unibody Package (MPX5100 Series)									
MPX5100AP	Tray	98ASB42796B		*				*	MPX5100AP
MPX5100DP	Tray	98ASB42797B			*			*	MPX5100DP
MPX5100GP	Tray	98ASB42796B		*		*			MPX5100GP
Small Outline Package (MPXV5100 Series)									
MPXV5100DP	Tray	98ASA99255D			*			*	MPXV5100DP
MPXV5100GC6U	Rail	98ASB17757C	*			*			MPXV5100G
MPXV5100GC7U	Rail	98ASB17759C	*			*			MPXV5100G
MPXV5100GP	Tray	98ASA99303D		*		*			MPXV5100GP

NXP reserves the right to change the detail specifications as may be required to permit improvements in the design of its products.

© 2018 NXP Semiconductors, B.V.



1 General Description

1.1 MPX5100AP/DP/GP Block diagram

Figure 1 shows a block diagram of the internal circuitry integrated on a pressure sensor chip in a unibody package.

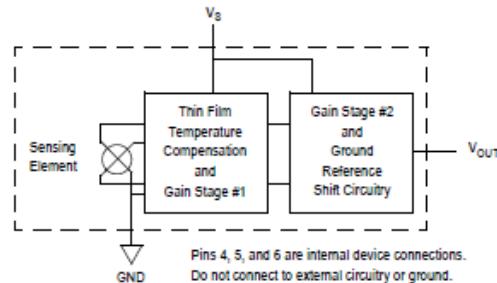


Figure 1. Integrated pressure sensor block diagram

1.2 MPX5100AP/DP/GP Pinout (Unibody)

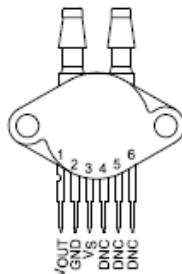


Figure 2. Device pinout (top view)

Table 1. Pin functions

Pin	Name	Function
1	V _{OUT}	Output voltage
2	GND	Ground
3	V _S	Voltage supply
4	DNC	Do not connect to external circuitry or ground.
5	DNC	Do not connect to external circuitry or ground.
6	DNC	Do not connect to external circuitry or ground.

MPX5100

2 Mechanical and Electrical Specifications

2.1 Maximum ratings

Table 3. Maximum ratings⁽¹⁾

Rating	Symbol	Value	Unit
Maximum pressure	P _{max}	400	kPa
Storage temperature	T _{stg}	-40 to +125	°C
Operating temperature	T _A	-40 to +125	°C

1. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

2.2 Operating characteristics

Table 4. Operating characteristics (V_G = 5 Vdc, T_A = 25 °C.)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure range ⁽¹⁾ Gauge, differential: MPX5100G/MPXV5100G Absolute: MPX5100AP	P _{OP}	0 15	— —	100 115	kPa
Supply voltage ⁽²⁾	V _G	4.75	5.0	5.25	V _{DC}
Supply current	I _O	—	7.0	10	mAdc
Minimum pressure offset ⁽³⁾ , (0 to 85 °C) @ V _G = 5.0 V	V _{OFF}	0.088	0.20	0.313	V _{DC}
Full-scale output ⁽⁴⁾ , differential and absolute (0 to 85 °C) @ V _G = 5.0 V	V _{FEO}	4.587	4.700	4.813	V _{DC}
Full-scale Span ⁽⁵⁾ , differential and absolute (0 to 85 °C) @ V _G = 5.0 V	V _{FSS}	—	4.500	—	V _{DC}
Accuracy ⁽⁶⁾	—	—	—	±2.5	%V _{FSS}
Sensitivity	V/P	—	45	—	mV/kPa
Response time ⁽⁷⁾	t _R	—	1.0	—	ms
Output source current at full-scale output	I _{O+}	—	0.1	—	mAdc
Warm-up time ⁽⁸⁾	—	—	20	—	ms
Offset stability ⁽⁹⁾	—	—	±0.5	—	%V _{FSS}

1. 1.0 kPa (kiloPascal) equals 0.145 psi.

2. Device is ratiometric within this specified excitation range.

3. Offset (V_{OFF}) is defined as the output voltage at the minimum rated pressure.

4. Full-scale output (V_{FEO}) is defined as the output voltage at the maximum or full-rated pressure.

5. Full-scale span (V_{FSS}) is defined as the algebraic difference between the output voltage at full-rated pressure and the output voltage at the minimum rated pressure.

6. Accuracy (error budget) consists of the following:

Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.

Temperature hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.

Pressure hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from minimum or maximum rated pressure at 25 °C.

ToSpan: Output deviation over the temperature range of 0 to 85 °C, relative to 25 °C.

ToOffset: Output deviation with minimum pressure applied over the temperature range of 0 to 85 °C, relative to 25 °C.

Variation from nominal: The variation from nominal values, for offset or full-scale span, as a percent of V_{FSS} at 25 °C.

7. Response time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.

8. Warm-up time is defined as the time required for the product to meet the specified output voltage after the pressure has been stabilized.

9. Offset stability is the product's output deviation when subjected to 1000 hours of pulsed pressure, temperature cycling with bias test.

MPX5100

3 On-chip Temperature Compensation and Calibration

Figure 5 shows the sensor output signal relative to pressure input. Typical, minimum, and maximum output curves are shown for operation over a temperature range of 0 to 85 °C using the decoupling circuit shown in Figure 7. The output will saturate outside of the specified pressure range.

Figure 6 illustrates both the Differential/Gauge and the Absolute Sensing Chip in the basic chip carrier. A fluorosilicone gel isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the sensor diaphragm.

The MPX5100 series pressure sensor operating characteristics, and internal reliability and qualification tests are based on use of dry air as the pressure media. Media, other than dry air, may have adverse effects on sensor performance and long-term reliability. Contact the factory for information regarding media compatibility in your application.

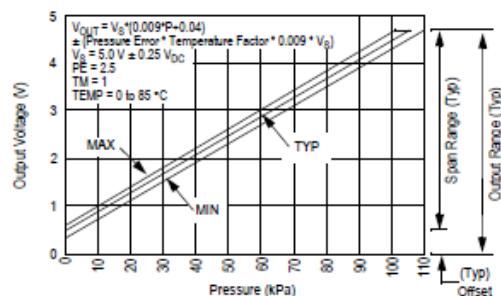


Figure 5. Output versus pressure differential

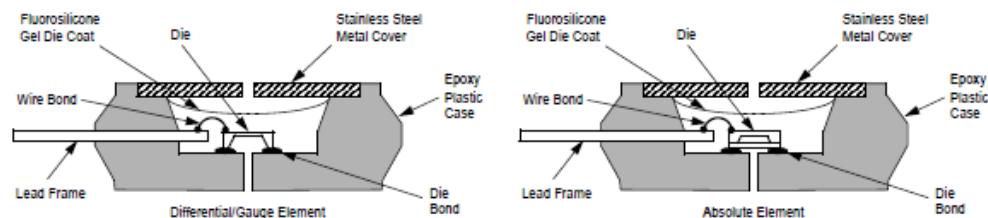


Figure 6. Cross-sectional diagrams (not-to-scale)

Figure 7 shows the recommended decoupling circuit for interfacing the output of the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.

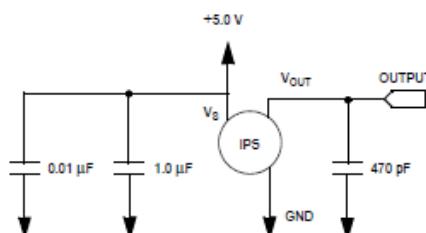


Figure 7. Recommended power supply decoupling and output filtering
(For additional output filtering, please refer to application note AN1846.)

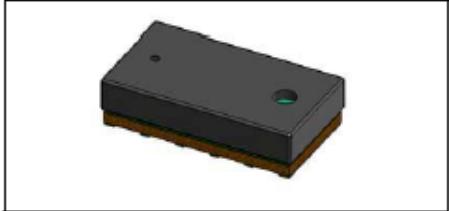
ANEXO B – PARTES DATASHEET VL53L0X



VL53L0X

World's smallest Time-of-Flight ranging and gesture detection sensor

Datasheet - production data



Features

- Fully integrated miniature module
 - 940 nm laser VCSEL
 - VCSEL driver
 - Ranging sensor with advanced embedded micro controller
 - 4.4 x 2.4 x 1.0 mm
- Fast, accurate distance ranging
 - Measures absolute range up to 2 m
 - Reported range is independent of the target reflectance
 - Advanced embedded optical cross-talk compensation to simplify cover glass selection
- Eye safe
 - Class 1 laser device compliant with latest standard IEC 60825-1:2014 - 3rd edition
- Easy integration
 - Single reflowable component
 - No additional optics
 - Single power supply
 - I2C interface for device control and data transfer
 - Xshutdown (reset) and interrupt GPIO
 - Programmable I2C address

Applications

- User detection for personal computers/laptops/tablets and IoT (energy saving)
- Robotics (obstacle detection)
- White goods (hand detection in automatic faucets, soap dispensers etc.)
- 1D gesture recognition.
- Laser assisted autofocus. Enhances and speeds up camera autofocus system performance, especially in difficult scenes (low light levels, low contrast) or fast moving video mode.

Description

The VL53L0X is a new generation Time-of-Flight (ToF) laser-ranging module housed in the smallest package on the market today, providing accurate distance measurement whatever the target reflectances unlike conventional technologies. It can measure absolute distances up to 2m, setting a new benchmark in ranging performance levels, opening the door to various new applications.

The VL53L0X integrates a leading-edge SPAD array (Single Photon Avalanche Diodes) and embeds ST's second generation FlightSense™ patented technology.

The VL53L0X's 940 nm VCSEL emitter (Vertical Cavity Surface-Emitting Laser), is totally invisible to the human eye, coupled with internal physical infrared filters, it enables longer ranging distances, higher immunity to ambient light, and better robustness to cover glass optical crosstalk.

April 2018

DocID029104 Rev 2

1/10

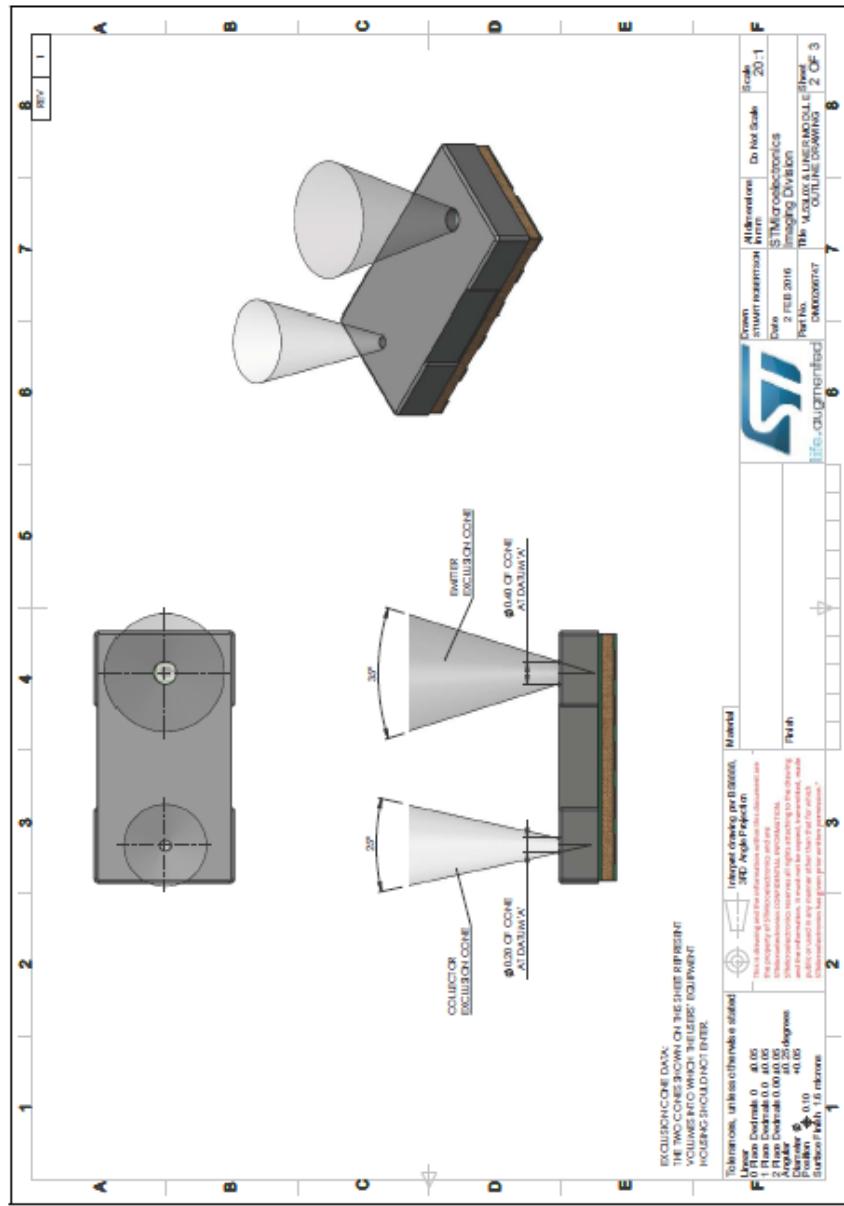
This is information on a product in full production.

www.st.com

Outline drawing

VL53L0X

Figure 23. Outline drawing (page 2/3)



7 Laser safety considerations

The VL53L0X contains a laser emitter and corresponding drive circuitry. The laser output is designed to remain within Class 1 laser safety limits under all reasonably foreseeable conditions including single faults in compliance with IEC 60825-1:2014 (third edition).

The laser output will remain within Class 1 limits as long as the STMicroelectronics recommended device settings (API settings) are used and the operating conditions specified are respected.

The laser output power must not be increased by any means and no optics should be used with the intention of focusing the laser beam.

Caution: Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

Figure 25. Class 1 laser product label



ANEXO C – PARTES DATASHEET MT3608



AEROSEMI

MT3608

2A, High Efficiency
1.2MHz Current Mode Step-Up Converter

FEATURES

- Integrated 80mΩ Power MOSFET
- 2V to 24V Input Voltage
- 1.2MHz Fixed Switching Frequency
- Internal 4A Switch Current Limit
- Adjustable Output Voltage
- Internal Compensation
- Up to 28V Output Voltage
- Automatic Pulse Frequency Modulation Mode at Light Loads
- up to 93% Efficiency
- Available in a 6-Pin SOT23-6 Package

APPLICATIONS

- Battery-Powered Equipment
- Set-Top Boxed
- LCD Bias Supply
- DSL and Cable Modems and Routers
- Networking cards powered from PCI or PCI express slots

GENERAL DESCRIPTION

The MT3608 is a constant frequency, 6-pin SOT23 current mode step-up converter intended for small, low power applications. The MT3608 switches at 1.2MHz and allows the use of tiny, low cost capacitors and inductors 2mm or less in height. Internal soft-start results in small inrush current and extends battery life.

The MT3608 features automatic shifting to pulse frequency modulation mode at light loads. The MT3608 includes under-voltage lockout, current limiting, and thermal overload protection to prevent damage in the event of an output overload. The MT3608 is available in a small 6-pin SOT-23 package.

TYPICAL APPLICATION

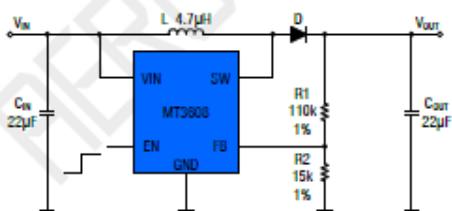
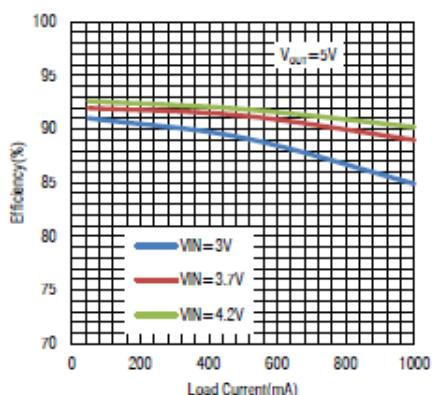


Figure 1. Basic Application Circuit



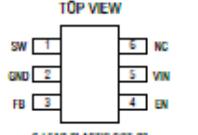
AEROSEMI

MT3608 2A,High Efficiency,1.2MHz Step-Up Converter

ABSOLUTE MAXIMUM RATINGS (Note 1)

VIN,EN voltages.....	-0.3V to 26V	Junction Temperature(Note2).....	160°C
FB Voltages.....	-0.3V to 6V	Operating Temperature Range.....	-40°C to 85°C
SW Voltage.....	-0.3V to 30V	Lead Temperature(Soldering,10s).....	300°C
Power Dissipation.....	0.6W	Storage Temperature Range.....	-65°C to 150°C
Thermal Resistance θ_{JC}	130°C/W	ESD HBM(Human Body Mode).....	2kV
Thermal Resistance θ_{JA}	250°C/W	ESD MM(Machine Mode).....	200V

PACKAGE/ORDER INFORMATION

TOP VIEW	Order Part Number	Package	Top Marking
 6-LEAD PLASTIC SOT-23 $T_{JMAX} = 160^\circ\text{C}$, $\theta_{JA} = 250^\circ\text{C}/\text{W}$, $\theta_{JC} = 130^\circ\text{C}/\text{W}$	MT3608	SOT23-6	B628DC

PIN DESCRIPTION

Pin Name	Pin Number	Description
SW	1	Power Switch Output. SW is the drain of the internal MOSFET switch. Connect the power inductor and output rectifier to SW. SW can swing between GND and 28V.
GND	2	Ground Pin
FB	3	Feedback Input. The FB voltage is 0.6V. Connect a resistor divider to FB.
EN	4	Regulator On/Off Control Input. A high input at EN turns on the converter, and a low input turns it off. When not used, connect EN to the input supply for automatic startup.
VIN	5	Input Supply Pin. Must be locally bypassed.
NC	6	No Connect

AEROSEMI

MT3608 2A, High Efficiency, 1.2MHz Step-Up Converter

ELECTRICAL CHARACTERISTICS (Note 3)

($V_{IN} = V_{EN} = 5V, T_A = 25^\circ C$, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
Operating Input Voltage		2		24	V
Under Voltage Lockout				1.98	V
Under Voltage Lockout Hysteresis			100		mV
Current (Shutdown)	$V_{EN} = 0V$		0.1	1	μA
Quiescent Current (PFM)	$V_{FB} = 0.7V$, No switch		100	200	μA
Quiescent Current (PWM)	$V_{FB} = 0.5V$, switch		1.6	2.2	mA
Switching Frequency			1.2		MHz
Maximum Duty Cycle	$V_{FB} = 0V$	90			%
EN Input High Voltage		1.5			V
EN Input Low Voltage				0.4	V
FB Voltage		0.588	0.6	0.612	V
FB Input Bias Current	$V_{FB} = 0.6V$	-50	-10		nA
SW On Resistance			80	150	$m\Omega$
SW Current Limit	$V_{IN} = 5V$, Duty cycle=50%		4		A
SW Leakage	$V_{SW} = 20V$			1	μA
Thermal Shutdown			155		$^\circ C$

Note 1: Absolute Maximum Ratings are those values beyond which the life of a device may be impaired.

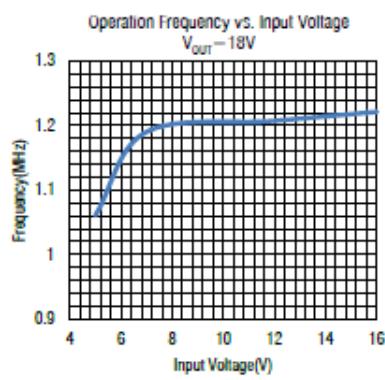
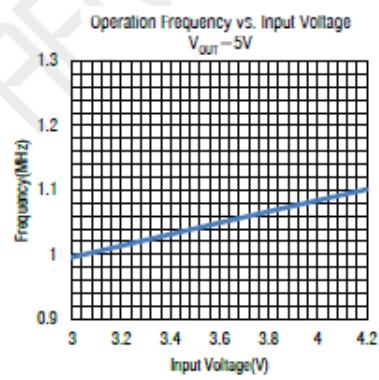
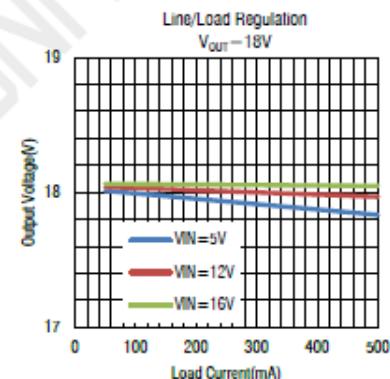
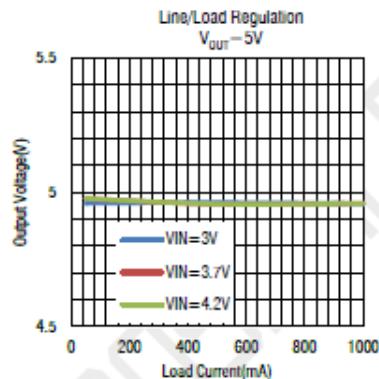
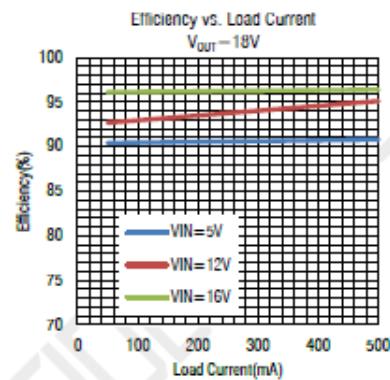
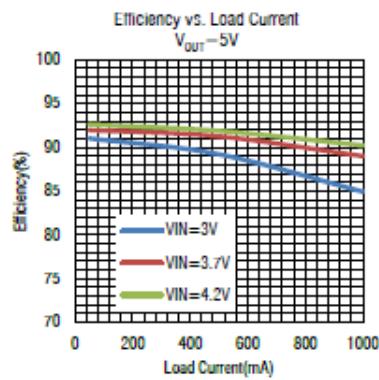
Note 2: T_J is calculated from the ambient temperature T_A and power dissipation P_D according to the following formula: $T_J = T_A + (P_D) \times (250^\circ C/W)$.

Note 3: 100% production test at $25^\circ C$. Specifications over the temperature range are guaranteed by design and characterization.

AEROSEMI

MT3608 2A,High Efficiency, 1.2MHz Step-Up Converter

TYPICAL PERFORMANCE CHARACTERISTICS



ANEXO D – PARTES DATASHEET ADS1115



www.ti.com



ADS1113
ADS1114
ADS1115

SBAS444B – MAY 2009 – REVISED OCTOBER 2009

Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference

Check for Samples: [ADS1113](#) [ADS1114](#) [ADS1115](#)

FEATURES

- ULTRA-SMALL QFN PACKAGE:
2mm × 1.5mm × 0.4mm
- WIDE SUPPLY RANGE: 2.0V to 5.5V
- LOW CURRENT CONSUMPTION:
Continuous Mode: Only 150µA
Single-Shot Mode: Auto Shut-Down
- PROGRAMMABLE DATA RATE:
8SPS to 860SPS
- INTERNAL LOW-DRIFT
VOLTAGE REFERENCE
- INTERNAL OSCILLATOR
- INTERNAL PGA
- I²C™ INTERFACE: Pin-Selectable Addresses
- FOUR SINGLE-ENDED OR TWO
DIFFERENTIAL INPUTS (ADS1115)
- PROGRAMMABLE COMPARATOR
(ADS1114 and ADS1115)

APPLICATIONS

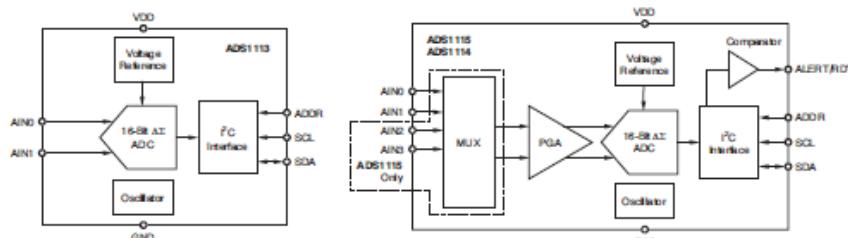
- PORTABLE INSTRUMENTATION
- CONSUMER GOODS
- BATTERY MONITORING
- TEMPERATURE MEASUREMENT
- FACTORY AUTOMATION AND PROCESS
CONTROLS

DESCRIPTION

The ADS1113, ADS1114, and ADS1115 are precision analog-to-digital converters (ADCs) with 16 bits of resolution offered in an ultra-small, leadless QFN-10 package or an MSOP-10 package. The ADS1113/4/5 are designed with precision, power, and ease of implementation in mind. The ADS1113/4/5 feature an onboard reference and oscillator. Data are transferred via an I²C-compatible serial interface; four I²C slave addresses can be selected. The ADS1113/4/5 operate from a single power supply ranging from 2.0V to 5.5V.

The ADS1113/4/5 can perform conversions at rates up to 860 samples per second (SPS). An onboard PGA is available on the ADS1114 and ADS1115 that offers input ranges from the supply to as low as ±256mV, allowing both large and small signals to be measured with high resolution. The ADS1115 also features an input multiplexer (MUX) that provides two differential or four single-ended inputs.

The ADS1113/4/5 operate either in continuous conversion mode or a single-shot mode that automatically powers down after a conversion and greatly reduces current consumption during idle periods. The ADS1113/4/5 are specified from -40°C to +125°C.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.
I²C is a trademark of NXP Semiconductors.
All other trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date.
Products conform to specifications per the terms of the Texas
Instruments standard warranty. Production processing does not
necessarily include testing of all parameters.

Copyright © 2009, Texas Instruments Incorporated

ADS1113
ADS1114
ADS1115

SBAS444B – MAY 2009 – REVISED OCTOBER 2009

www.ti.com

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ORDERING INFORMATION

For the most current package and ordering information, see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.

ABSOLUTE MAXIMUM RATINGS⁽¹⁾

	ADS1113, ADS1114, ADS1115	UNIT
VDD to GND	-0.3 to +5.5	V
Analog input current	100, momentary	mA
Analog input current	10, continuous	mA
Analog input voltage to GND	-0.3 to VDD + 0.3	V
SDA, SCL, ADDR, ALERT/RDY voltage to GND	-0.5 to +5.5	V
Maximum junction temperature	+150	°C
Storage temperature range	-60 to +150	°C

(1) Stresses above those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. Exposure to absolute maximum conditions for extended periods may affect device reliability.

PRODUCT FAMILY

DEVICE	PACKAGE DESIGNATOR MSOP/QFN	RESOLUTION (Bits)	MAXIMUM SAMPLE RATE (SPS)	COMPARATOR	PGA	INPUT CHANNELS (Differential/ Single-Ended)
ADS1113	BROI/N6J	16	860	No	No	1/1
ADS1114	BRNI/N5J	16	860	Yes	Yes	1/1
ADS1115	BOGI/N4J	16	860	Yes	Yes	2/4
ADS1013	BRMI/N9J	12	3300	No	No	1/1
ADS1014	BRQI/N8J	12	3300	Yes	Yes	1/1
ADS1015	BRPI/N7J	12	3300	Yes	Yes	2/4


www.ti.com

 ADS1113
 ADS1114
 ADS1115

SBAS444B –MAY 2009–REVISED OCTOBER 2009

ELECTRICAL CHARACTERISTICS

All specifications at -40°C to $+125^{\circ}\text{C}$, VDD = 3.3V, and Full-Scale (FS) = $\pm 2.048\text{V}$, unless otherwise noted.
 Typical values are at $+25^{\circ}\text{C}$.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115			UNIT
		MIN	TYP	MAX	
ANALOG INPUT					
Full-scale input voltage ⁽¹⁾	$V_{IN} = (AIN_P) - (AIN_N)$		$\pm 4.096/\text{PGA}$		V
Analog input voltage	AIN_P or AIN_N to GND	GND		VDD	V
Differential input impedance			See Table 2		
Common-mode input impedance	FS = $\pm 6.144\text{V}^{(1)}$		10		$M\Omega$
	FS = $\pm 4.096\text{V}^{(1)}, \pm 2.048\text{V}$		6		$M\Omega$
	FS = $\pm 1.024\text{V}$		3		$M\Omega$
	FS = $\pm 0.512\text{V}, \pm 0.256\text{V}$		100		$M\Omega$
SYSTEM PERFORMANCE					
Resolution	No missing codes	16			Bits
Data rate (DR)		8, 16, 32, 64, 128, 250, 475, 860			SPS
Data rate variation	All data rates	-10		10	%
Output noise		See Typical Characteristics			
Integral nonlinearity	DR = 8SPS, FS = $\pm 2.048\text{V}$, best fit ⁽²⁾			1	LSB
Offset error	FS = $\pm 2.048\text{V}$, differential inputs		± 1	± 3	LSB
	FS = $\pm 2.048\text{V}$, single-ended inputs		± 3		LSB
Offset drift	FS = $\pm 2.048\text{V}$		0.005		LSB/ $^{\circ}\text{C}$
Offset power-supply rejection	FS = $\pm 2.048\text{V}$		1		LSB/V
Gain error ⁽³⁾	FS = $\pm 2.048\text{V}$ at 25°C		0.01	0.15	%
	FS = $\pm 0.256\text{V}$		7		ppm/ $^{\circ}\text{C}$
	FS = $\pm 0.048\text{V}$		5	40	ppm/ $^{\circ}\text{C}$
Gain drift ⁽³⁾	FS = $\pm 6.144\text{V}^{(1)}$		5		ppm/ $^{\circ}\text{C}$
	FS = $\pm 4.096\text{V}^{(1)}$		80		ppm/V
	FS = $\pm 2.048\text{V}$		0.02	0.1	%
PGA gain match ⁽³⁾	Match between any two PGA gains		0.05	0.1	%
Gain match	Match between any two inputs		3		LSB
Common-mode rejection	At dc and FS = $\pm 0.256\text{V}$		105		dB
	At dc and FS = $\pm 2.048\text{V}$		100		dB
	At dc and FS = $\pm 6.144\text{V}^{(1)}$		90		dB
	f _{CM} = 60Hz, DR = 8SPS		105		dB
	f _{CM} = 50Hz, DR = 8SPS		105		dB
DIGITAL INPUT/OUTPUT					
Logic level					
V _{IH}		0.7VDD		5.5	V
V _{IL}		GND – 0.5		0.3VDD	V
V _{OL}	I _{OL} = 3mA	GND	0.15	0.4	V
Input leakage					
I _H	V _{IH} = 5.5V			10	μA
I _L	V _{IL} = GND	10			μA

(1) This parameter expresses the full-scale range of the ADC scaling. In no event should more than VDD + 0.3V be applied to this device.

(2) 99% of full-scale.

(3) Includes all errors from onboard PGA and reference.

ADS1113
ADS1114
ADS1115

SBAS444B – MAY 2009 – REVISED OCTOBER 2009



www.ti.com

ELECTRICAL CHARACTERISTICS (continued)

All specifications at -40°C to $+125^{\circ}\text{C}$, VDD = 3.3V, and Full-Scale (FS) = $\pm 2.048\text{V}$, unless otherwise noted.
 Typical values are at $+25^{\circ}\text{C}$.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115			UNIT
		MIN	TYP	MAX	
POWER-SUPPLY REQUIREMENTS					
Power-supply voltage		2		5.5	V
	Power-down current at 25°C		0.5	2	μA
Supply current	Power-down current up to 125°C			5	μA
	Operating current at 25°C		150	200	μA
	Operating current up to 125°C			300	μA
Power dissipation	VDD = 5.0V		0.9		mW
	VDD = 3.3V		0.5		mW
	VDD = 2.0V		0.3		mW
TEMPERATURE					
Storage temperature		-60		+150	$^{\circ}\text{C}$
Specified temperature		-40		+125	$^{\circ}\text{C}$

PIN CONFIGURATIONS**PIN DESCRIPTIONS**

PIN #	DEVICE			ANALOG/DIGITAL INPUT/OUTPUT	DESCRIPTION
	ADS1113	ADS1114	ADS1115		
1	ADDR	ADDR	ADDR	Digital Input	I ^C slave address select
2	NC ⁽¹⁾	ALERT/RDY	ALERT/RDY	Digital Output	Digital comparator output or conversion ready (NC for ADS1113)
3	GND	GND	GND	Analog	Ground
4	AIN0	AIN0	AIN0	Analog Input	Differential channel 1: Positive Input or single-ended channel 1 Input
5	AIN1	AIN1	AIN1	Analog Input	Differential channel 1: Negative Input or single-ended channel 2 Input
6	NC	NC	AIN2	Analog Input	Differential channel 2: Positive Input or single-ended channel 3 Input (NC for ADS1113/4)
7	NC	NC	AIN3	Analog Input	Differential channel 2: Negative Input or single-ended channel 4 Input (NC for ADS1113/4)
8	VDD	VDD	VDD	Analog	Power supply: 2.0V to 5.5V
9	SDA	SDA	SDA	Digital I/O	Serial data: Transmits and receives data
10	SCL	SCL	SCL	Digital Input	Serial clock input: Clocks data on SDA

(1) NC pins may be left floating or tied to ground.

ADS1113
ADS1114
ADS1115

SBAS444B – MAY 2009 – REVISED OCTOBER 2009



www.ti.com

TYPICAL CHARACTERISTICS

At $T_A = +25^\circ\text{C}$ and $\text{VDD} = 3.3\text{V}$, unless otherwise noted.

OPERATING CURRENT vs TEMPERATURE

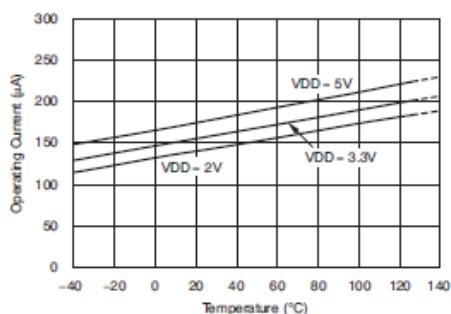


Figure 2.

SHUTDOWN CURRENT vs TEMPERATURE

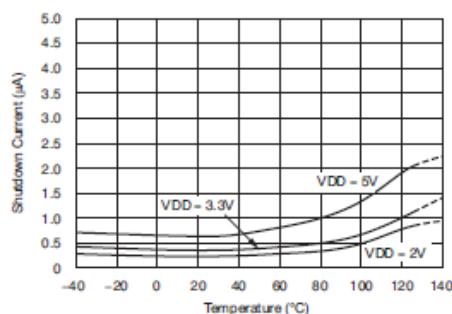


Figure 3.

SINGLE-ENDED OFFSET ERROR vs TEMPERATURE⁽¹⁾

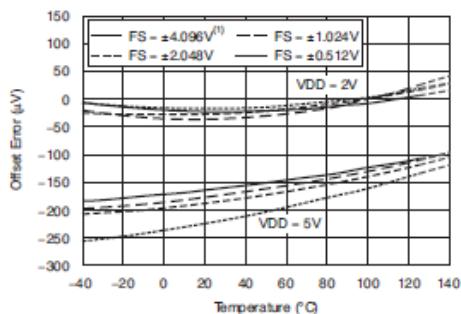


Figure 4.

DIFFERENTIAL OFFSET vs TEMPERATURE

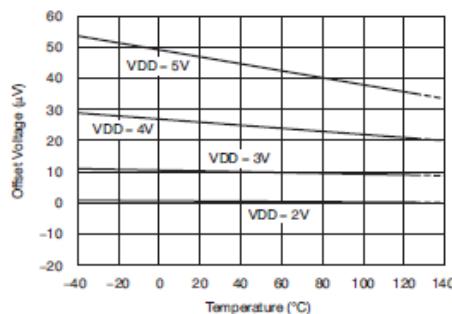


Figure 5.

GAIN ERROR vs TEMPERATURE

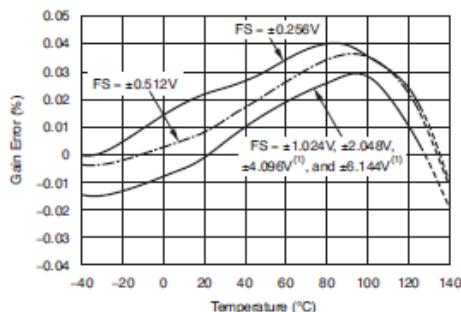


Figure 6.

GAIN ERROR vs SUPPLY

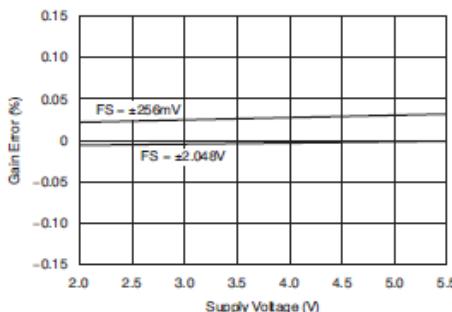


Figure 7.

(1) This parameter expresses the full-scale range of the ADC scaling. In no event should more than $\text{VDD} + 0.3\text{V}$ be applied to this device.

ANEXO E – PARTES DATASHEET HX711



HX711

24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of $\pm 20\text{mV}$ or $\pm 40\text{mV}$ respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

FEATURES

- Two selectable differential input channels
- On-chip active low noise PGA with selectable gain of 32, 64 and 128
- On-chip power supply regulator for load-cell and ADC analog power supply
- On-chip oscillator requiring no external component with optional external crystal
- On-chip power-on-reset
- Simple digital control and serial interface: pin-driven controls, no programming needed
- Selectable 10SPS or 80SPS output data rate
- Simultaneous 50 and 60Hz supply rejection
- Current consumption including on-chip analog power supply regulator:
normal operation < 1.5mA, power down < 1uA
- Operation supply voltage range: 2.6 ~ 5.5V
- Operation temperature range: -40 ~ +85°C
- 16 pin SOP-16 package

APPLICATIONS

- Weigh Scales
- Industrial Process Control

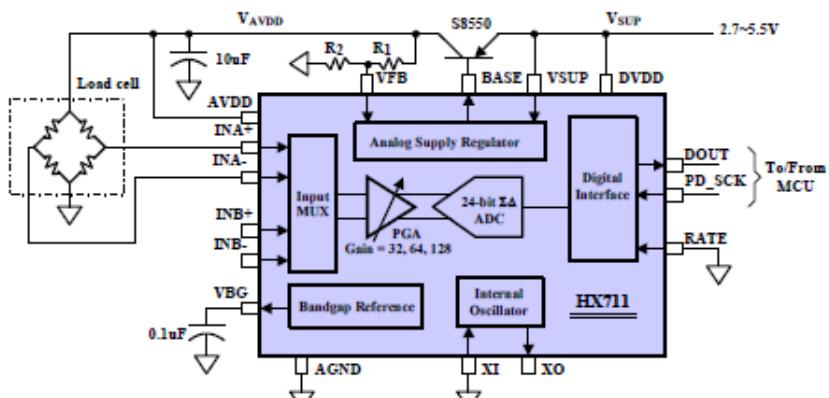


Fig. 1 Typical weigh scale application block diagram



HX711

KEY ELECTRICAL CHARACTERISTICS

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	V(imp)-V(inn)		$\pm 0.5(\text{AVDD}/\text{GAIN})$		V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0		10		Hz
	Internal Oscillator, RATE = DVDD		80		
	Crystal or external clock, RATE = 0		$f_{\text{ch}}/1,105,920$		
	Crystal or external clock, RATE = DVDD		$f_{\text{ch}}/138,240$		
Output data coding	2's complement	800000	7FFFFFF		HEX
Output settling time ⁽¹⁾	RATE = 0		400		ms
	RATE = DVDD		50		
Input offset drift	Gain = 128		0.2		mV
	Gain = 64		0.4		
Input noise	Gain = 128, RATE = 0		50		nV(rms)
	Gain = 128, RATE = DVDD		90		
Temperature drift	Input offset (Gain = 128)		± 6		nV/°C
	Gain (Gain = 128)		± 5		
Input common mode rejection	Gain = 128, RATE = 0		100		dB
Power supply rejection	Gain = 128, RATE = 0		100		dB
Reference bypass (V_{BG})			1.25		V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6	5.5		V
	AVDD, VSUP	2.6	5.5		
Analog supply current (including regulator)	Normal		1400		μA
	Power down		0.3		
Digital supply current	Normal		100		μA
	Power down		0.2		

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

Table 2 Key Electrical Characteristics