

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ALAIN DUBUS PINHEIRO CARVALHO
HENRI STADZISZ
SERGIO AUGUSTO MASSAMI NAKANISHI
VITOR MOTA FYLYK

**PROTÓTIPO DE SISTEMA COMPUTACIONAL PARA PESAGEM
AUTÔNOMA DE FRANGOS**

TRABALHO DA DISCIPLINA DE OFICINA DE INTEGRAÇÃO 3

CURITIBA
2014

ALAIN DUBUS PINHEIRO CARVALHO
HENRI STADZISZ
SERGIO AUGUSTO MASSAMI NAKANISHI
VITOR MOTA FYLYK

PROTÓTIPO DE SISTEMA COMPUTACIONAL PARA PESAGEM AUTÔNOMA DE FRANGOS

Trabalho de Disciplina de Graduação – Oficina de Integração 3, apresentado ao Curso de Engenharia de Computação, do Departamento Acadêmico de Eletrônica e do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção da aprovação na disciplina.

Professor(es): Prof. MSc. Guilherme Alceu Schneider

Prof. Dr. Gustavo Benvenuto Borba

CURITIBA
2014

RESUMO

CARVALHO, Alain D. P.; STADZISZ, Henri; NAKANISHI, Sergio A. M.; FYLYK, Vitor M. **Protótipo de sistema computacional para pesagem autônoma de frangos.** 2014. 76 f. Monografia (Curso de Engenharia de Computação) – Departamento Acadêmico de Eletrônica e Departamento Acadêmico de Informática, Universidade Tecnológica do Paraná. Curitiba, 2014.

O projeto apresentado neste documento descreve o desenvolvimento de um protótipo de sistema de pesagem de aves. Este sistema é controlado via um aplicativo presente em um celular Windows Phone remoto conectado em uma rede sem fio (Wi-Fi) que está ligada ao módulo de pesagem embarcado. O objetivo é a avaliação contínua do peso para controle do crescimento dos frangos de uma granja sem que haja a necessidade de contato direto com os animais. Com precisão de 13 gramas nas medições, o sistema determina em intervalos de tempo quantos frangos estão presentes na balança e qual o peso médio delas. Isso é feito por meio de um algoritmo que toma como base o desvio padrão e a curva esperada de crescimento, que é pré-definida na placa embarcada.

Palavras-chave: Pesagem de Aves. Célula de Carga. Instrumentação. Filtragem Digital. Windows Phone.

ABSTRACT

CARVALHO, Alain D. P.; STADZISZ, Henri; NAKANISHI, Sergio A. M.; FYLYK, Vitor M. **Protótipo de sistema computacional para pesagem autônoma de frangos.** 2014. 76 f. Monografia (Curso de Engenharia de Computação) – Departamento Acadêmico de Eletrônica e Departamento Acadêmico de Informática, Universidade Tecnológica do Paraná. Curitiba, 2014.

This document describes the development of a bird weighing system prototype. This system is controlled by an application (App) installed in a mobile Windows Phone which is remotely connected to a wireless network (Wi-Fi) that is linked to the embedded weighing module. The goal is the continuous evaluation of the weight to control the chicken growth in a farm without the need of direct contact with the animals. With a measuring precision of 13 grams (approximately 0.459 ounces), the system determines periodically how many chickens are present on top of the scale and which is the average weight of them. That is accomplished with an algorithm based on the standard deviation and the expected growth curve, which is pre-defined on the embedded board.

Keywords: Bird weighting. Load cell. Instrumentation. Digital filtering. Windows Phone.

LISTA DE ILUSTRAÇÕES

Figura 1 - Tipo de pesagem manual de aves em granjas (cone).....	4
Figura 2 - Tipo de pesagem manual de aves em granjas (pelos pés)	4
Figura 3 - Sistema de pesagem de aves TOLEDO	5
Figura 4 - Sistema de pesagem de aves GSI Brasil.....	5
Figura 5 - Desenvolvimento fisiológico de matrizes	11
Figura 6 - Ponte resistiva	13
Figura 7 - Extensômetro do tipo folha	15
Figura 8 - Célula de carga típica	15
Figura 9 - Placa LPC 1769.....	17
Figura 10 - Camadas do modelo OSI e seus respectivos protocolos	19
Figura 11 - Fluxograma do CSMA/CD	20
Figura 12 - Fluxograma do CSMA/CA.....	22
Figura 13 - Formação do quadro do padrão IEEE 802.11	22
Figura 14 - Ilustração da IDE LPCXpresso	24
Figura 15 - Interface do sistema Windows Phone 8.1	25
Figura 16 - Interface do Microsoft Visual Studio 2013.....	26
Figura 17 - Interface do Microsoft Expression Blend 4.....	26
Figura 18 - Elementos que compõem o sistema desenvolvido.....	28
Figura 19 - Esquemático de ligação da placa embarcada	30
Figura 20 - Diagrama do funcionamento da função principal	32
Figura 21 – Faixas de interpretação do peso adquirido.....	33
Figura 22 - Sinal de exemplo capturado com a placa de desenvolvimento	36
Figura 23 - Destaque para erros de captura	37
Figura 24 - Esquemático do condicionamento analógico	38
Figura 25 - Operações para tratamento do sinal capturado	39
Figura 26 - Sinal da Figura 22 submetido à filtragem da mediana.....	40
Figura 27 - Resposta em frequência dos filtros digitais IIR	42

Figura 28 - Sinal da Figura 26 após passagem pelo filtro IIR	42
Figura 29 - Sinal da Figura 28 submetido a uma redução de cadência.....	43
Figura 30 - Sinal da Figura 29 submetido à determinação por histerese.....	43
Figura 31 - Sinal da Figura 29 submetido à determinação por histerese com mais ou menos uma unidade do ADC.....	44
Figura 32 - Relação entre peso despejado no recipiente e o peso medido com o sistema de aquisição	45
Figura 33 - Diagrama das principais classes do aplicativo	47
Figura 34 - Engines desenvolvidas para o aplicativo.....	48
Figura 35 - Tabela de Histórico.....	49
Figura 36 - Tabela de Ciclo.....	50
Figura 37 - Tabela de DiaPeriodo	50
Figura 38 - Tabela de Amostra	51
Figura 39 - Tabela de BarraHistograma.....	51
Figura 40 - Modelo de interface: resumo.....	52
Figura 41 - Modelo de interface: relação de ciclos	52
Figura 42 - Modelo de interface: novo ciclo.....	53
Figura 43 - Modelo de interface: histograma.....	53
Figura 44 - Modelo de interface: amostras de dados	54
Figura 45 - Modelo de interface: relação entre curvas	54
Figura 46 - Modelo de interface: informações	55
Figura 47 - Modelo de interface: configurações	55
Figura 48 - Cronograma de desenvolvimento de projeto.....	62

LISTA DE QUADROS

Quadro 1 - Dados de crescimento de aves Ross 308	12
Quadro 2 - Formação do quadro do padrão IEEE 802.3	21
Quadro 3 - Gastos com itens do projeto.....	29
Quadro 4 - Estrutura de Histograma	34
Quadro 5 - Estrutura de Barra.....	34
Quadro 6 - Estrutura de ResumoHora	35
Quadro 7 - Estrutura de Amostra	35
Quadro 8 - Coeficientes dos filtros digitais IIR	41
Quadro 9 - Protocolos de comunicação	46

SUMÁRIO

1	INTRODUÇÃO.....	1
1.1	TEMA.....	1
1.2	DELIMITAÇÃO DO ESTUDO.....	3
1.3	PROBLEMA.....	3
1.4	OBJETIVOS.....	6
1.5	JUSTIFICATIVA.....	6
1.6	PROCEDIMENTOS METODOLÓGICOS.....	7
1.7	EMBASAMENTO TEÓRICO.....	8
1.8	ESTRUTURA DO TRABALHO.....	8
2	FUNDAMENTAÇÃO.....	10
2.1	CONTROLE DO CRESCIMENTO DE AVES.....	10
2.2	CÉLULA DE CARGA.....	13
2.3	MICROCONTROLADOR.....	16
2.4	PADRÃO DE CONEXÃO.....	18
2.5	PLATAFORMA DE DESENVOLVIMENTO DA PLACA EMBARCADA.....	23
2.6	SISTEMA WINDOWS PHONE.....	24
2.7	PLATAFORMA DE DESENVOLVIMENTO DO APLICATIVO.....	25
3	DESENVOLVIMENTO.....	28
3.1	VISÃO GERAL DO SISTEMA.....	28
3.2	ORÇAMENTO DO PROTÓTIPO.....	29
3.3	PROJETO EMBARCADO.....	29
3.4	INTERPRETAÇÃO DO SINAL DA BALANÇA.....	35
3.5	PROTOCOLOS DE COMUNICAÇÃO.....	45
3.6	MODELO ESTRUTURAL DO APLICATIVO.....	47
3.7	BANCO DE DADOS.....	48
3.8	INTERFACE DO APLICATIVO.....	51
4	CONCLUSÕES E TRABALHOS FUTUROS.....	56
4.1	CONCLUSÕES.....	56
4.2	TRABALHOS FUTUROS.....	57
	REFERÊNCIAS.....	58
	APÊNDICES.....	62
	APÊNDICE A – Cronograma.....	62
	APÊNDICE B – Riscos.....	65

1 INTRODUÇÃO

Este trabalho de pesquisa aplicada se situa na área de emprego da engenharia de computação ao setor agropecuário. Este é um dos principais setores econômicos brasileiros e tem inúmeras demandas por automação e sistemas de informação.

Entre as dificuldades encontradas no setor, observou-se uma necessidade particular na avicultura no que diz respeito ao controle da criação de frangos. Este tipo de criação exige um acompanhamento rigoroso das diferentes etapas de produção com vistas a garantir altos índices de produtividade e consequente rentabilidade para os negócios.

Este capítulo apresenta o tema da pesquisa ressaltando a importância do setor para a economia brasileira. Apresenta-se, também, o problema que motivou o desenvolvimento deste trabalho, os objetivos estabelecidos, os procedimentos metodológicos adotados e o embasamento teórico. Por fim, descreve-se a estrutura do trabalho.

1.1 TEMA

O Brasil é, desde 2005, o maior exportador de frangos do mundo, alcançando 37% do mercado internacional. O segundo colocado são os Estados Unidos com 33%. Só o Paraná é responsável por quase 30% dessas exportações. A avicultura brasileira representa 1,5% do PIB, gerando – somente em impostos – mais de 6 bilhões de reais anuais. (RIZZOTTO, 2014)

A produção de frango é realizada em todo o país, sendo que 90% dessa produção é feita no sistema de integração vertical, no qual a empresa controla o processo de produção em sua totalidade, ou seja, a companhia controla tudo, desde a produção até a distribuição dos produtos.

A avicultura brasileira é mundialmente reconhecida como uma das mais desenvolvidas, tendo uma produtividade excepcional. Esse reconhecimento foi obtido devido à implementação de programas de qualidade como melhoramentos genéticos, nutrição, manejo e boas práticas de produção. Isso faz com que seja fornecido um produto de alta qualidade tanto para o mercado nacional, quanto para o internacional.

Existe hoje, no Brasil, um protocolo de boas práticas de produção de frangos que é adotado pelas granjas, mais como uma normativa do que como uma sugestão. Esse protocolo foi criado pela União Brasileira de Avicultura (UBA) em conjunto com

a Associação Brasileira de Exportadores de Frango (ABEF) em 2005. “As Boas Práticas de Produção para a cadeia produtiva de frangos de corte são uma das formas de garantir o padrão de qualidade exigido pelo mercado interno e externo. Somente com a junção de vários procedimentos implementados e definidos por programas de qualidade obtém-se competitividade global dos produtos” (UNIÃO BRASILEIRA DE AVICULTURA, 2008).

Vários fatores influenciam a eficiência do processo de produção avícola. Dentre eles estão as boas condições de ambiente e sanidade e, principalmente, o uso de novas tecnologias. É nessa área de tecnologias que o Brasil sofre desvantagens perante outros países produtores de carne de frango. Países como Estados Unidos e integrantes da União Europeia, utilizam grandes quantidades de tecnologia para monitorar e controlar suas granjas, tentando garantir assim o menor nível de estresse por contato humano possível com os animais e a maior qualidade do ambiente. Enquanto isso, no Brasil, boa parte da produção é feita ainda com intervenção manual, ou seja, os animais são criados com intenso contato humano.

Como o produto final deve ter a aceitação do mercado, este pode requerer auditorias e avaliações para que o produto seja comercializado, principalmente quando se trata de exportações. Para isso é necessário um controle rigoroso e registros que permitam a rastreabilidade da produção, permitindo que, se houver algum problema, o mesmo possa ser identificado e corrigido facilmente. O contato humano pode aumentar a quantidade de problemas na produção, assim como o número de mortes, e desencorajar a aceitação em mercados de importação. Desta forma, a utilização de tecnologias facilita a aceitação em mercados estrangeiros ao produto nacional, pois os riscos de contaminação e outros problemas durante a produção são reduzidos (POLONIO, 2009).

Como em outras áreas, ainda há alguma resistência de produtores de menor porte quanto à utilização de novas tecnologias na produção, principalmente na área escolhida pela equipe para este projeto que é a pesagem de aves, uma vez que eles consideram a pesagem de responsabilidade do abatedouro. Contudo, as grandes produtoras que, como mencionado anteriormente, detêm toda a cadeia de sua produção até a distribuição, têm grande interesse nessa área para poderem reduzir perdas e terem uma capacidade mais elevada de resposta a situações atípicas.

Nesse contexto, esse projeto pretende desenvolver um protótipo que tem a intenção de ajudar e promover futuros estudos ligados à produção de aves para abate

por meio da pesagem autônoma e precisa, tornando assim as empresas mais competitivas frente ao mercado internacional. Desta forma, este documento apresenta a proposta de um sistema computacional para pesagem autônoma aplicável à avicultura, em particular à criação de frangos.

1.2 DELIMITAÇÃO DO ESTUDO

O acompanhamento do crescimento de frangos é uma atividade essencial para a avicultura desta espécie pois tem papel central para alcançar produtividade e rentabilidade nos negócios. O acompanhamento do crescimento de frangos envolve o controle de diversos fatores incluindo o balanceamento alimentar, microclima das granjas (i.e., controle da temperatura, humidade e luminosidade) e controle da engorda das aves.

Esse projeto trata da questão particular do controle da evolução de peso das aves que é um parâmetro fundamental para garantir a qualidade do processo de criação. Quando o peso das aves está fora dos padrões, todo seu crescimento torna-se comprometido aumentando o ciclo de crescimento e exigindo intervenção no processo de criação. Desta forma, este estudo sobre o crescimento de frangos foi delimitado para tratar exclusivamente a questão da pesagem contínua das aves durante seu ciclo de crescimento.

O projeto visa conceber e desenvolver o protótipo de um sistema computacional para pesagem contínua dos frangos em um ambiente de criação de aves (i.e., em uma granja). Serão estudadas as exigências para realizar tal função e serão selecionadas e desenvolvidas tecnologias que permitam pôr em prática um sistema de pesagem. Não está incluído neste estudo a integração do sistema de pesagem com outros sistemas de informação empregados no processo de criação de frangos.

Outra delimitação relevante de estudo é o cenário de testes do protótipo, o sistema será testado em um ambiente adaptado utilizando objetos para simular os pesos dos frangos. Isso ocorre devido a não disponibilidade, em primeiro momento, de uma granja para efetuar os testes.

1.3 PROBLEMA

A pesagem de aves feita hoje no Brasil é, com raras exceções, de forma manual. Esse tipo de pesagem é extremamente lento, pois é necessário pesar uma

ave por vez e isso tem que ser feito por um funcionário. Esta pesagem manual é, também, ineficaz pois a quantidade de amostras que se faz é muito pequena e isso reduz a precisão do estudo estatístico, além de ser estressante para as aves pois elas precisam ser capturadas para tal procedimento, causando um distúrbio emocional muito grande, assim como mostrado nas Figuras 1 e 2.



Figura 1 - Tipo de pesagem manual de aves em granjas (cone)
Fonte: Unesp (2014)

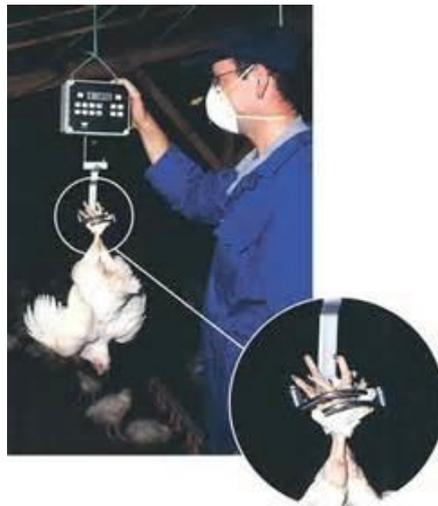


Figura 2 - Tipo de pesagem manual de aves em granjas (pelos pés)
Fonte: Weltech (2014)

Para resolver esse problema, algumas empresas desenvolveram soluções de automação de pesagem, contudo essas tecnologias têm algumas restrições quanto às funcionalidades. Duas dessas soluções podem ser visualizadas nas Figuras 3 e 4, a seguir.



Figura 3 - Sistema de pesagem de aves TOLEDO

Fonte: <http://www.toledobrasil.com.br/balanca/balancas-para-pesar-animais/2096a>



Figura 4 - Sistema de pesagem de aves GSI Brasil

Fonte: http://www.gsibrasil.ind.br/produtos/index.php?areas_id=24&produtos_categorias_id=5&id=36

Apesar de resolverem boa parte do problema, talvez esses sistemas pudessem ser melhor desenvolvidos para tornar o processo de produção mais rápido, eficaz e mais seguro tanto para as empresas e funcionários, quanto para as aves. Os principais problemas existentes nestes sistemas são a dificuldade de acesso aos dados de pesagem e a necessidade de instalação de fiações dentro dos criadouros.

A intenção deste projeto é aproveitar as melhores ideias dessas soluções já implementadas e testadas, e ainda acrescentar alguns recursos para limitar mais o contato entre humanos e aves.

1.4 OBJETIVOS

Nesta seção são apresentados os objetivos geral e específicos do trabalho, relativos ao problema anteriormente apresentado.

1.4.1 Objetivo Geral

O objetivo desse trabalho é desenvolver um protótipo de sistema computacional para pesagem autônoma de frangos para o setor aviário, sem necessidade de interação entre funcionários e aves. O sistema a ser construído poderá ser acessado e controlado remotamente.

1.4.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Estudar o processo de crescimento e engorda de frangos
- Projetar e montar um equipamento eletrônico de medidas de peso
- Desenvolver uma lógica de controle das medidas de peso de frangos
- Desenvolver um modelo de comunicação entre a placa e o *smartphone*
- Desenvolver um aplicativo para *smartphone* visando o acesso remoto ao sistema de pesagem

1.5 JUSTIFICATIVA

O procedimento comum adotado no Brasil para a pesagem de frangos durante seu ciclo de crescimento nas granjas é o uso de balanças convencionais e a manipulação direta das aves por amostragem. Este processo manual de pesagem afeta a criação podendo causar estresse nas galinhas, podendo levar até a morte delas, gerando prejuízo para seus criadores. Além disso, faz-se necessário a contratação de um funcionário que deve se deslocar entre os criadouros diariamente para realizar a pesagem. As perdas financeiras ocasionadas por esse tipo de procedimento justificam a implementação de uma alternativa melhor para a pesagem das aves. Desta forma, o controle da pesagem de frangos tem grande impacto nos resultados desta área econômica.

O desenvolvimento de um sistema de pesagem automática de frangos envolve vários conhecimentos e representa um desafio em termos de engenharia. Dentro do curso de Engenharia de Computação, o conhecimento obtido torna-se extremamente útil para a concepção, construção e implementação de um sistema para a pesagem de frangos de maneira automática. Para isso, serão utilizados conhecimentos em software e em hardware. É de extrema importância o conhecimento de software e hardware, pois será utilizado durante todo o projeto, já que para a pesagem automática, faz-se necessária a comunicação entre o hardware que será construído e o software, que processará as informações. Um sistema automático de pesagem seria vantajoso para as granjas, uma vez que reduziria drasticamente o contato direto com as aves e, assim, diminuiria o número de mortes por estresse devido a esse contato. A utilização de tal sistema garantiria ainda uma amostragem maior dos dados de pesagem, melhorando a curva de crescimento obtida. Uma vez que essa curva determina todo o funcionamento da granja e, até mesmo, o seu planejamento de lucro, é muito importante que ela seja o mais precisa possível.

1.6 PROCEDIMENTOS METODOLÓGICOS

Neste trabalho, adotou-se como métrica de medição a curva de crescimento de frangos, especialmente do tipo AgRoss, que possui curvas amplamente divulgadas (AVIAGEM, 2012). O peso será medido por uma balança e o sistema usará a curva de referência para calcular o número de frangos na amostra e o peso individual médio de cada ave.

A natureza deste trabalho é de pesquisa aplicada, pois irá gerar um protótipo de aplicação prática, dirigido a um problema específico. A abordagem é qualitativa, envolvendo testes do projeto em ambiente simulado. Os procedimentos técnicos envolvem pesquisa bibliográfica, desenvolvimento e experimentação.

O desenvolvimento será composto pelo levantamento da teoria de sensores de peso e do crescimento de aves e o projeto, desenvolvimento e testes de um sistema de medição da curva de crescimento.

Serão apresentados detalhes sobre células de carga e circuitos de condicionamento de seu sinal, de modo a bem fundamentar o estudo. O controle de crescimento de aves contará também com referências explicativas e, sendo o país um dos maiores produtores mundiais, serão consultados manuais e guias para

agropecuáristas que explicam como se dá esta produção e todos os cuidados necessários.

Com estas informações, o projeto do sistema de medição terá uma base prática, isto é, de acordo com as práticas empregadas nas granjas nacionais. Com a consulta à documentação do hardware utilizado, poderá ter início o desenvolvimento de um protótipo que realize o objetivo proposto, compondo o resultado deste trabalho.

1.7 EMBASAMENTO TEÓRICO

Para fundamentar a temática do trabalho e sua relevância, foram utilizados os referenciais de Pitombeira (2011), Polonio (2009) e União Brasileira de Avicultura (2008). A teoria da curva de crescimento na avicultura é fundamentada a partir de França et al. (2008), Aviagem (2012) e Cobb-Vantress (2009). As referências de Sodalec Distribution (2008), Toledo do Brasil (2014) e Toledo (2014) trazem as características de produtos atuais semelhantes ao proposto.

Na parte do projeto, Brusamarello (2014), Portal célula de carga (2004) e Campilho (2000) fornecem a teoria do uso de células de carga. Além destas, são consultadas as documentações do hardware e software utilizado.

1.8 ESTRUTURA DO TRABALHO

Este trabalho de pesquisa está organizado da seguinte forma:

Capítulo 1, Introdução: apresenta o tema, as delimitações do estudo, o problema e os objetivos do projeto, a justificativa, os procedimentos metodológicos, o embasamento teórico e esta própria estrutura do trabalho.

Capítulo 2, Fundamentação: apresenta exemplos, teorias e definições de conhecimentos relacionados com o tema e objetivo da pesquisa.

Capítulo 3, Desenvolvimento: apresenta a especificação do sistema de pesagem, sua arquitetura, o modelo do software, as técnicas de cálculo e uma descrição do protótipo construído, assim como das ferramentas e tecnologias adotadas. Os resultados obtidos em cada fase de desenvolvimento serão apresentados.

Capítulo 4, Conclusões e trabalhos futuros: apresenta as conclusões obtidas com a realização da pesquisa e são sugeridos trabalhos futuros que poderiam ser desenvolvidos a partir do estudo realizado.

2 FUNDAMENTAÇÃO

Este capítulo apresenta os principais conhecimentos teóricos necessários à compreensão do restante do trabalho. Abordam-se questões como o crescimento de aves, células de carga, microcontrolador, plataforma de desenvolvimento e comunicação.

2.1 CONTROLE DO CRESCIMENTO DE AVES

Pelas referências relacionadas no capítulo anterior, nota-se a clara importância econômica que a produção avícola possui no âmbito nacional e estadual. Para que haja um melhor aproveitamento dos recursos de criação e, conseqüentemente, melhores rendimentos, várias pesquisas foram desenvolvidas visando melhorar a produtividade (França, 2008) e várias empresas têm feito o aperfeiçoamento genético para os mais diferentes tipos de produção (Cobb-Vantress, 2009), gerando raças específicas de animais.

Algumas das características que afetam a produção avícola são a densidade dos alojamentos das aves, as variações de temperatura ambiental e a temperatura do piso, os sistemas de alimentação e hidratação, a qualidade do ar, o asseio do galpão antes da chegada de novos lotes de aves, a qualidade da cama (material distribuído sobre o chão para diluir excretas e isolar os animais da baixa temperatura do piso), a iluminação e campânulas de calor, o tamanho dos galpões, a qualidade da água, o manejo nutricional, a biossegurança e sanitização da granja, a vacinação dos pintos, entre outros (Cobb-Vantress, 2009).

Para que haja um controle do consumo de recursos da granja de modo a também avaliar a lucratividade, Cobb-Vantress (2009) recomenda em seu manual de manejo a manutenção de dados de criação, que inclusive são exigidos por autoridades de alguns países antes do abate. Estes dados incluem a mortalidade e a refugagem, consumo diário de ração e água e a proporção de um em relação ao outro, tratamento da água, temperatura e umidade mínimas e máximas diárias, alterações de manejo e número de aves encaminhadas para o abate.

Um dos fatores essenciais, principalmente para o abate, é o ganho de peso dos animais, que permite saber a conversão de ração e água em peso e determinar se os animais estão saudáveis e se a produtividade está no nível ideal. Assim, as empresas de melhoramento genético publicam manuais com objetivos de desempenho para os

seus produtos específicos, de modo que, com um manejo adequado, a produtividade seja a máxima possível para aquele tipo de animal (Aviagem, 2012).

O cálculo dos valores para comparação com os objetivos parte da coleta de dados de peso das aves, que é feita por amostras no galpão a não menos do que 5% dos animais, com uma frequência semanal, em geral (Aviagem, 2008). Estas medições fornecem um peso médio relativo à granja toda e permitem acompanhar se o desempenho em termos de engorda é o máximo esperado para aquela raça de animal.

A Figura 5 apresenta um modelo de curva de crescimento para matrizes, mas que é muito próxima ao dos frangos de corte (Aviagem, 2008), sendo que para os frangos, o horizonte de tempo considerado é, em geral, de no máximo 70 dias, após os quais são destinados ao abate. Como as aves para abate não serão matrizes, não há necessidade de esperar-se até a maturidade sexual, o que permite o abate a um prazo curto com um bom crescimento corporal.

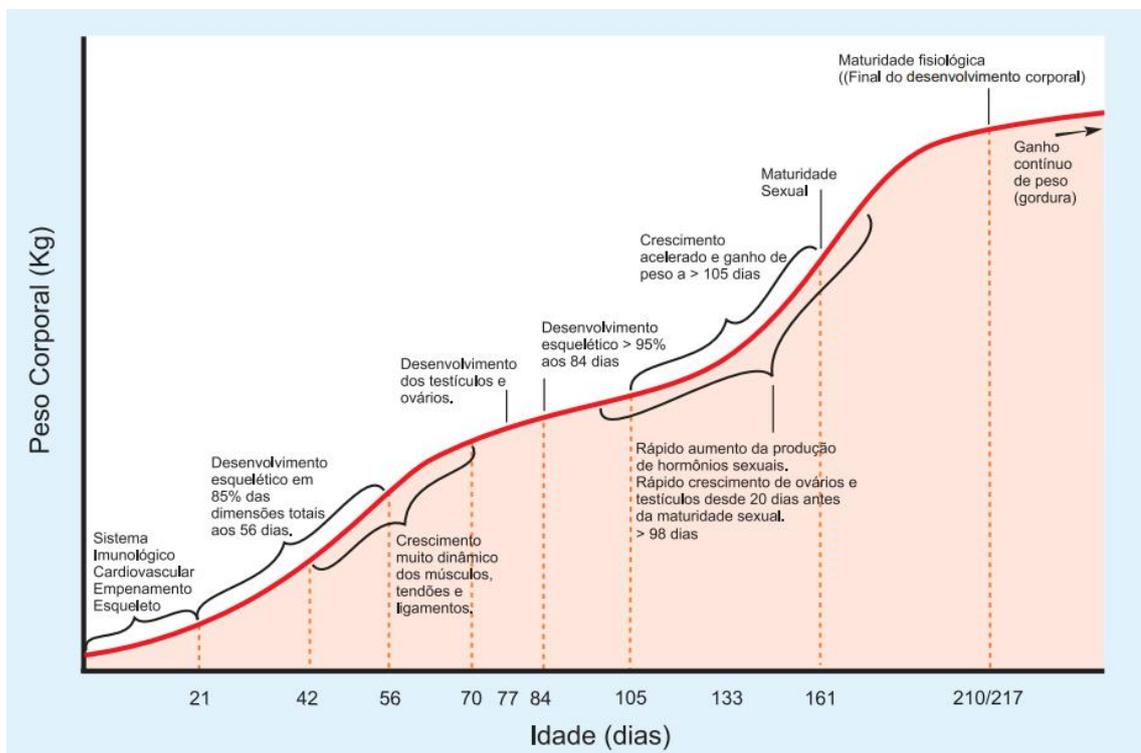


Figura 5 - Desenvolvimento fisiológico de matrizes
Fonte: Aviagem (2008)

Para este projeto, será considerada a curva de crescimento dos animais do tipo Ross 308 (Aviagem, 2008), cujos dados encontram-se no Quadro 1. Este quadro apresenta o objetivo de desempenho para uma criação mista (machos e fêmeas).

Quadro 1 - Dados de crescimento de aves Ross 308

Dias	Peso(g)	Dias	Peso(g)	Dias	Peso(g)
0	42	25	1226	50	3495
1	56	26	1309	51	3582
2	72	27	1393	52	3669
3	89	28	1479	53	3754
4	109	29	1567	54	3838
5	132	30	1656	55	3920
6	157	31	1746	56	4002
7	185	32	1836	57	4082
8	217	33	1928	58	4160
9	251	34	2020	59	4238
10	289	35	2113	60	4313
11	330	36	2207	61	4388
12	375	37	2300	62	4460
13	422	38	2394	63	4531
14	473	39	2488	64	4600
15	527	40	2581	65	4668
16	585	41	2675	66	4733
17	645	42	2768	67	4797
18	709	43	2861	68	4856
19	775	44	2954	69	4919
20	844	45	3046	70	4978
21	916	46	3137		
22	990	47	3228		
23	1066	48	3318		
24	1145	49	3407		

Fonte: Aviagem (2008)

O acompanhamento da curva de crescimento torna possível avaliar o desenvolvimento da granja e é uma ferramenta útil no sentido de ajudar a prever a lucratividade de uma criação ao final do ciclo de um lote, quando as aves concluíram seu desenvolvimento. Deste modo, conhecendo-se o peso médio das aves dia a dia e administrando o suprimento de ração e água de forma que atinjam a curva de crescimento esperada, obter-se-á o máximo desempenho determinado para aquele tipo de ave, conforme especificado pela empresa de melhoramento responsável. Além

disso, o acompanhamento do peso das aves, permite a tomada de ações corretivas quando constatado desvios na curva de evolução do peso.

2.2 CÉLULA DE CARGA

Neste projeto serão utilizadas células de carga para a medição do peso das aves criadas em granjas. Descreve-se, a seguir, como estas células são compostas. A configuração em “ponte de *Wheatstone*” é um dos métodos utilizados para a medição de desequilíbrios em resistências, na qual, em geral, uma, duas ou quatro das resistências da ponte são transdutores resistivos, nos quais uma grandeza física é transformada em resistência. A Figura 6 mostra um circuito com configuração em ponte resistiva.

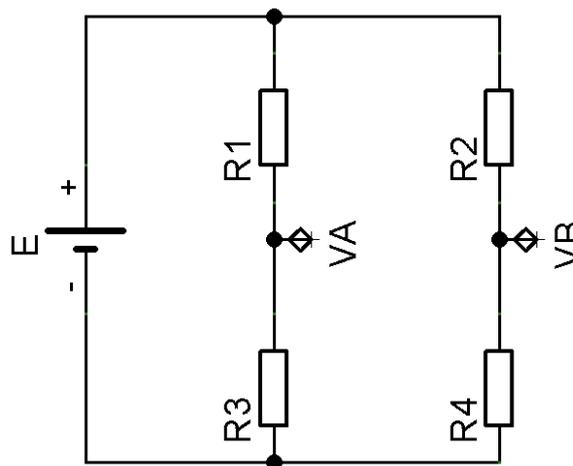


Figura 6 - Ponte resistiva
Fonte: autoria própria

O resultado da medição é a diferença da tensão nos pontos $V_A - V_B$, resultando em $V_O = V_A - V_B$. A ponte estará em equilíbrio quando atende à Equação 1, pelo que $V_O = 0$ (Campilho, 2000).

$$R_2 \cdot R_3 = R_1 \cdot R_4 \quad (1)$$

Pela análise dos divisores de tensão, resulta que:

$$V_O = V_A - V_B = E \left(\frac{R_3}{R_1 + R_3} - \frac{R_4}{R_2 + R_4} \right) \quad (2)$$

Colocando-se apenas um braço ativo na ponte, i.e. fazendo-se $R_3 = R + \Delta R$, e deixando-a em equilíbrio ao pôr-se $R_1 = R_2 = R_4 = R$, a tensão de saída será

aproximadamente igual ao descrito na Equação 3, levando em conta que $\Delta R \ll R$, pois este é o caso dos extensômetros, conforme discutido a seguir.

$$V_O = E \left(\frac{R + \Delta R}{2R + \Delta R} - \frac{R}{2R} \right) = E \frac{\Delta R}{4R + 2\Delta R} \approx E \frac{\Delta R}{4R} \quad (3)$$

A sensibilidade $S_v = \frac{\partial V_O}{\partial \Delta R}$ é igual a $E/4R$, mostrando que a sensibilidade da ponte pode ser aumentada com a tensão de alimentação, mas será reduzida por um fator de 4 vezes a resistência da ponte. Ao colocar-se dois sensores de mesma natureza e de características iguais na ponte, com $R_3 = R_2 = R + \Delta R$ e $R_1 = R_4 = R$, a aproximação para a tensão de saída será dada pela equação (4).

$$V_O = E \left(\frac{R + \Delta R}{2R + \Delta R} - \frac{R}{2R + \Delta R} \right) = E \left(\frac{\Delta R}{2R + \Delta R} \right) \approx E \frac{\Delta R}{2R} \quad (4)$$

Nota-se então que, ao utilizar o segundo sensor, a sensibilidade dobra ($S_v = \frac{\partial V_O}{\partial \Delta R} = \frac{E}{2R}$) (Campilho, 2000). Este é um resultado importante quando os sensores em questão apresentam variações de ordem muito pequena em relação ao ponto de operação, como é o caso para extensômetros. Este resultado pode ainda ser melhorado, ao utilizar também sensores de natureza e características iguais, mas com reação inversa, e.g. se antes foram colocados dois sensores de tração, agora serão dois novos de compressão. Neste caso, tem-se $R_3 = R_2 = R + \Delta R$ e $R_1 = R_4 = R - \Delta R$. A tensão de saída será, sem aproximações, igual a:

$$V_O = E \left(\frac{R + \Delta R}{2R} - \frac{R - \Delta R}{2R} \right) = E \frac{\Delta R}{R} \quad (5)$$

A sensibilidade da ponte novamente dobra ($S_v = E/R$) e, além disto, como não houve aproximação para V_O , a operação será linear em relação a ΔR . Há, portanto, no uso de 4 sensores, duas características importantes, que é a elevação da sensibilidade e também a linearidade em relação às variações medidas, tornando esta configuração ideal para a disposição de sensores resistivos. Além disto, há no mercado a venda de conjuntos de 4 sensores advindos das mesmas condições de fabricação, permitindo que sejam muito semelhantes e que produzam excelentes resultados nesta configuração (Campilho, 2000).

Extensômetros resistivos são transdutores passivos que convertem uma deformação em variação de resistência. São, essencialmente, formados por um condutor ou semiconductor posto sobre uma base elástica, que é aplicada sobre a

superfície em que se quer medir os esforços mecânicos. São largamente utilizados em células de carga de balanças, para a medição de peso, empregando a configuração de ponte de quatro transdutores (Campilho, 2000).

Os extensômetros de tipo folha são os mais populares na medição de força, pois são pequenos, altamente lineares e de baixa impedância (Figura 7). São compostos de ligas metálicas ou semicondutores, que geram diferentes características, pertinentes às várias aplicações. Deve-se ter cuidado especial na sua colagem nas superfícies em que se quer medir a deformação, pois a histerese e resistência elétrica, entre outros, podem ser afetados (Brusamarello, 2014).

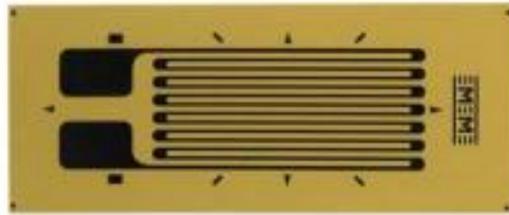


Figura 7 - Extensômetro do tipo folha
Fonte: Brusamarello (2014)

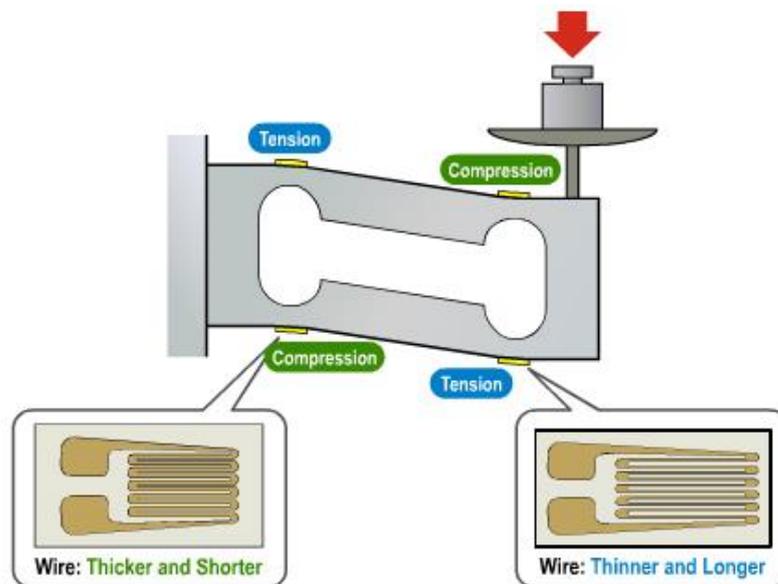


Figura 8 - Célula de carga típica
Fonte: Brusamarello (2014)

A célula de carga utilizada em balanças em geral é a do tipo de duas vigas bi-engastadas, como ilustrado na Figura 8. Ao colocar a carga em uma extremidade livre da célula, dois transdutores sofrem compressão e dois sofrem tração, como esperado de seu uso na ponte de medição (Brusamarello, 2014). Deste modo, o peso

da carga será proporcional à deformação medida, que gera variação de resistência nos transdutores e que, por sua vez, é traduzido em diferença de tensão na ponte de medição.

As vigas da célula de carga são construídas de modo a responderem a diferentes faixas de pesos sem que se quebrem ou sofram deformação permanente e, por isso, as células de carga comerciais possuem uma série de características que as diferenciam, além do formato. Duas delas são: a capacidade nominal e sensibilidade. A capacidade nominal diz respeito ao máximo peso que pode medir, com fatores de segurança de 50% de sobrecarga sem danos no funcionamento (peso acima do nominal) e de 300% para ruptura (peso para que haja defeito irreversível). A sensibilidade representa o quanto a tensão da ponte variará em relação à tensão de alimentação para a capacidade nominal em mV/V, que, em geral, é da ordem de 2 a 3 mV/V (Portal célula de carga, 2004).

Assim, uma célula em ponte alimentada com 10 V, com capacidade nominal de 20 Kg e sensibilidade de 2 mV/V deverá sofrer uma diferença de tensão nos terminais de saída de 20 mV ao ter sobre si toda a carga nominal. Considerando-se estas características, pode-se calcular a amplificação necessária para que se faça a leitura do sinal em uma faixa de tensão mais alta, principalmente para a leitura por um conversor analógico-digital.

2.3 MICROCONTROLADOR

Um microcontrolador é um dispositivo eletrônico programável capaz de realizar diferentes tipos de processamentos de dados, sinais de entrada e saída e cálculos de acordo com a lógica de sua programação. Existe atualmente um grande número de arquiteturas de microcontroladores, cada qual disponibilizando recursos variados. A escolha de uma arquitetura depende da aplicação pretendida e de suas exigências em termos de capacidade de processamento e recursos como memória e interfaces.

Neste trabalho será empregado um microcontrolador de arquitetura ARM, especificamente um ARM Cortex-M3, acompanhado de uma placa LPCexpress, ambos produzidos pela empresa NXP. A arquitetura ARM foi escolhida, principalmente, pelo seu baixo custo e sua grande popularidade atualmente, sendo o microcontrolador mais vendido mundialmente.

O modelo da placa utilizado neste trabalho será a LPC1769, que é utilizada para aplicações embarcadas, tendo como características um alto nível de integração e um baixo consumo de energia. Ela opera em frequências de CPU de até 120 MHz. A CPU ARM Cortex-M3 incorpora um pipeline de 3 estágios e utiliza a arquitetura Harvard com barramentos de dados, de instruções e de periféricos separados.

O complemento para periféricos da placa LPC1769 (Figura 9) inclui até 512 KB de memória flash, até 64 KB de memória de dados, *Ethernet* MAC, interface USB / Host / OTG, 8 canais do controlador DMA para propósitos gerais, 4 UARTs, dois canais CAN, 2 controladores SSP, interface SPI, 3 interfaces I2C-bus, 2 entradas e 2 saídas de interface I2S-bus, 8 canais ADC de 12 bits, DAC de 10 bits, controle de motor PWM, interface *QuadratureEncoder*, quatro temporizadores de uso geral, 6 saídas PWM de propósito geral, Real-Time Clock (RTC) de baixíssimo consumo com alimentação por bateria separada, e até 70 pinos de I/O de propósito geral. Os pinos do LPC1769/68/67/66/65/64/63 são compatíveis com os pinos da série de microcontroladores com ARM7 (LPC236x).

O ARM Cortex-M3 é um microprocessador de 32 bits para uso geral. Ele oferece um alto desempenho e um baixo consumo de energia. Além disso, ele disponibiliza novos recursos, como uma baixa latência de interrupção e múltiplos barramentos centrais capazes de fazer acessos simultâneos. Técnicas de *pipeline* estão implementadas, assim, todas as partes de processamento e sistemas de memória podem operar continuamente. Tipicamente, enquanto uma instrução é executada, a próxima está sendo decodificada e a terceira está sendo coletada da memória.

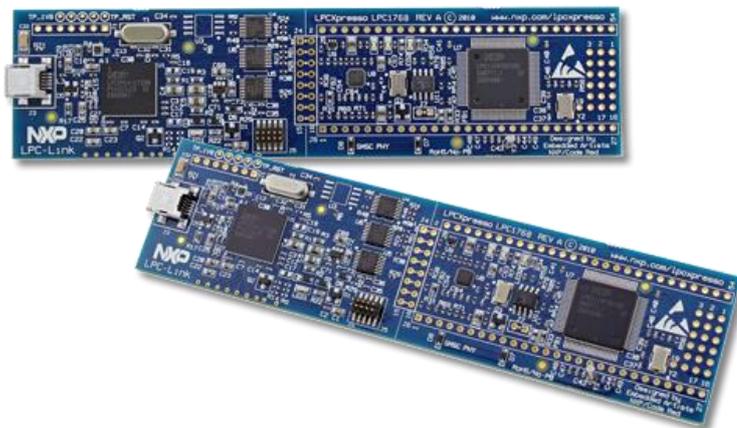


Figura 9 - Placa LPC 1769
Fonte: Embeddedartists (2014)

O RTC (*Real Time Clock*) é um conjunto de contadores que faz a medida do tempo enquanto o sistema está ligado e, opcionalmente, quando está desligado. O RTC do LPC1769 foi projetado para consumir pouca energia. O RTC funciona apenas com o fornecimento de energia do chip central e, quando é alimentado pela bateria, ele irá operar com menos de 2,1 V. O erro da informação fornecida pelo RTC é muito baixo, podendo ser de menos de um segundo por dia quando em funcionamento com uma tensão e temperatura constantes. O RTC possui um pequeno conjunto de registradores de backup (20 bytes) para armazenar os dados enquanto a parte principal do LPC1769 estiver desligada.

O bloco *Ethernet* contém um circuito de *Ethernet* 10 Mbit/s ou 100 Mbit/s (dependendo da versão de distribuição da placa) completo, projetado para proporcionar desempenho otimizado através da utilização da aceleração do hardware por DMA. As características incluem um conjunto generoso de registradores de controle, operação *half* ou *full duplex*, controle de fluxo, quadros de controle, aceleração de hardware para repetição de transmissão e possui transmissão e recepção automática de quadros. O bloco de *Ethernet* e a CPU dividem o D-code do ARM Cortex-M3 e os sistemas de barramento através da matriz multicamadas AHB para acessar os diversos blocos de SRAM para dados de *Ethernet*, controle e informação de *status*.

2.4 PADRÃO DE CONEXÃO

Para entender o funcionamento do protocolo TCP/IP, deve-se primeiramente conhecer suas funções separadamente, conforme descrito a seguir.

O endereço IP é um número de 32 bits, usualmente dividido em 4 octetos, que se anexa ao pacote de dados que será transmitido na camada de rede. Este conjunto de octetos, geralmente escritos em números decimais, são utilizados para identificar os dispositivos e as redes que estão conectados entre si. Vale lembrar que dispositivos de redes diferentes, por exemplo com endereços 200.1.1.0 e 172.16.0.0, não são capazes de se conectar a não ser que exista um roteador ou um *switch* gerenciável para fazer a rota entre essas redes. É importante saber, também, que existem dois endereços que nunca podem ser utilizados, o endereço de rede e o de broadcast.

O endereço de rede é dado pelo valor mais baixo do último octeto da rede ou sub-rede e o de broadcast, pelo mais alto. Ou seja, em uma rede 200.1.1.0/24, o 24 significa a máscara de rede 255.255.255.0, o endereço de rede será 200.1.1.0 e o de broadcast 200.1.1.255. Contudo, em uma rede 200.1.1.4/32, 32 significa agora a máscara de rede 255.255.255.252, o endereço de rede será 200.1.1.4 e o de broadcast é 200.1.1.6. (LAGES, 2014)

Existem dois tipos de protocolos IP, o IPv4 e o IPv6. Como neste projeto não haverá necessidade de uma quantidade extremamente grande de conexões, uma vez que elas são limitadas também pelo *Access Point*, é mais simples e suficiente que seja utilizado o IPv4, pois este contém apenas os mencionados 32 bits que já possibilitam alcançar 254 hosts em uma mesma rede, contra os 128 bits do IPv6.

O TCP (*Transmission Control Protocol*) é um protocolo pertencente à camada de transporte com o intuito de fazer o controle dos quadros que são enviados e recebidos por um determinado dispositivo. Nesta mesma camada, existem outros protocolos de envio e recebimento de quadros, sendo que o mais utilizado depois do TCP é o UDP (*User Datagram Protocol*), que não faz controle de chegada de dados.

Neste projeto será utilizado o TCP pelo fato de que ele é capaz de verificar a integridade dos dados entregues a um destinatário, ou seja, ele faz uma verificação para saber se não existem quadros faltantes nem corrompidos na comunicação entre os dispositivos e se houver, o remetente o reenvia. Na Figura 10 é possível visualizar as camadas do modelo OSI e como os protocolos em questão se distribuem dentro delas.

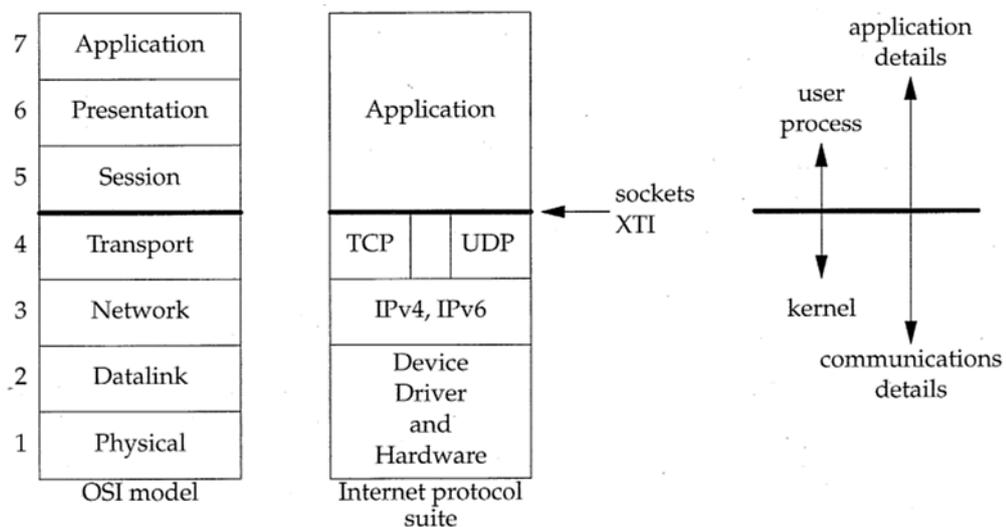


Figura 10 - Camadas do modelo OSI e seus respectivos protocolos
 Fonte: Lages (2014)

No TCP/IP a formação do socket que vai conter os dados a serem enviados é dada pela associação do número da porta pela qual essa comunicação vai ocorrer com o endereço IP. Essa associação de portas pode ser feita de forma dinâmica, desde que os serviços que serão utilizados não sejam aqueles que tem portas pré-definidas, como TELNET, HTTP e FTP.

A identificação da conexão entre os dois dispositivos é dada pelo par de *sockets*, um em cada extremidade. A interface se baseia em funções como *open*, *close*, *send* e *recieve*. (LAGES, 2014)

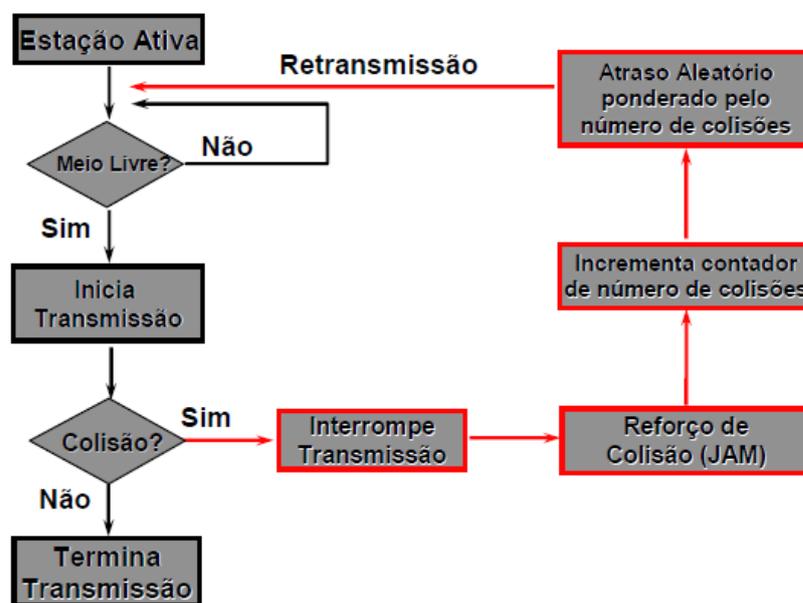


Figura 11 - Fluxograma do CSMA/CD
Fonte: PUC-RIO (2014)

O padrão IEEE 802.3u, também conhecido como *FastEthernet*, utiliza o controle de acesso ao meio conhecido como CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*). O funcionamento do CSMA/CD está ilustrado na Figura 11.

Esse controle de acesso ao meio é baseado em verificação de disponibilidade de conexão, ou seja, caso o meio esteja ocupado, o sistema espera por um tempo para realizar uma nova pesquisa. Caso o meio esteja livre o pacote é enviado e se houver colisão, o sistema vai tratar essa colisão e, então, retransmitir o pacote quando o meio estiver novamente livre.

Os padrões 802.3 e 802.11 da IEEE trabalham apenas nas camadas física e de enlace do modelo OSI, sendo que a camada de enlace é subdividida em LLC e MAC. O LLC é responsável pelo tipo de serviço que será feito na camada de enlace como, por exemplo, multiplexação e controle de fluxo de erros.

O endereço MAC é o responsável pelo acesso ao meio. Ele é o endereço físico imutável de 48 bits presente na placa de rede do aparelho que está fazendo a conexão.

O quadro com os dados a serem enviados no padrão 802.3u é formado por 7 bytes de sincronização, 1 byte para identificar o início do quadro em si (SFD – *Start Frame Delimiter*), 12 bytes com os endereços MAC do destino e da origem respectivamente, 2 bytes para determinar o tamanho dos dados, um bloco com tamanho variável de 46 à 1500 bytes para os dados e 4 bytes para fazer a checagem da integridade dos dados (*Check Sum*). Esse quadro pode ser melhor visualizado no Quadro 2.

Quadro 2 - Formação do quadro do padrão IEEE 802.3

Preâmbulo (7 bytes)	SFD (1 byte)	MAC destino (6 bytes)	MAC origem (6 bytes)	Tamanho dos dados (2 bytes)	Dados (46 até 1500 bytes)	<i>Check Sum</i> (4 byte)
------------------------	--------------------	-----------------------------	----------------------------	--------------------------------------	------------------------------------	----------------------------------

Fonte: PUC-RIO (2014)

O padrão 802.11 possui duas diferenças principais com relação ao 802.3, que são a formação do quadro e o tipo de controle de acesso ao meio. Este padrão é mais conhecido hoje como Wi-Fi e é licenciado pela Wi-Fi Alliance. Contudo, o Wi-Fi é apenas um tipo de conexão baseada no 802.11. O padrão estabelece normas de criação e uso de redes sem fio. (INFOWESTER, 2014)

Enquanto o padrão 802.3 utiliza o CSMA/CD como controle de acesso ao meio, o 802.11 faz uso do CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Na Figura 12 pode-se visualizar o fluxograma do CSMA/CA.

No caso do CSMA/CA, o controle de acesso se dá através da verificação do estado do meio e então, caso ocupado, do sorteio de um valor de *backoff*. Após esse sorteio, o meio é verificado repetidamente enquanto não estiver livre, e quando estiver, se o valor de *backoff* for maior que 0, o pacote vai esperar para ser enviado. No entanto se o valor for 0 o quadro é transmitido e esperasse o *acknowledge* para confirmar a chegada e a integridade dos dados enviados e, se isso não ocorre, o sistema aumenta o valor máximo para o *backoff* e retorna ao sorteio.

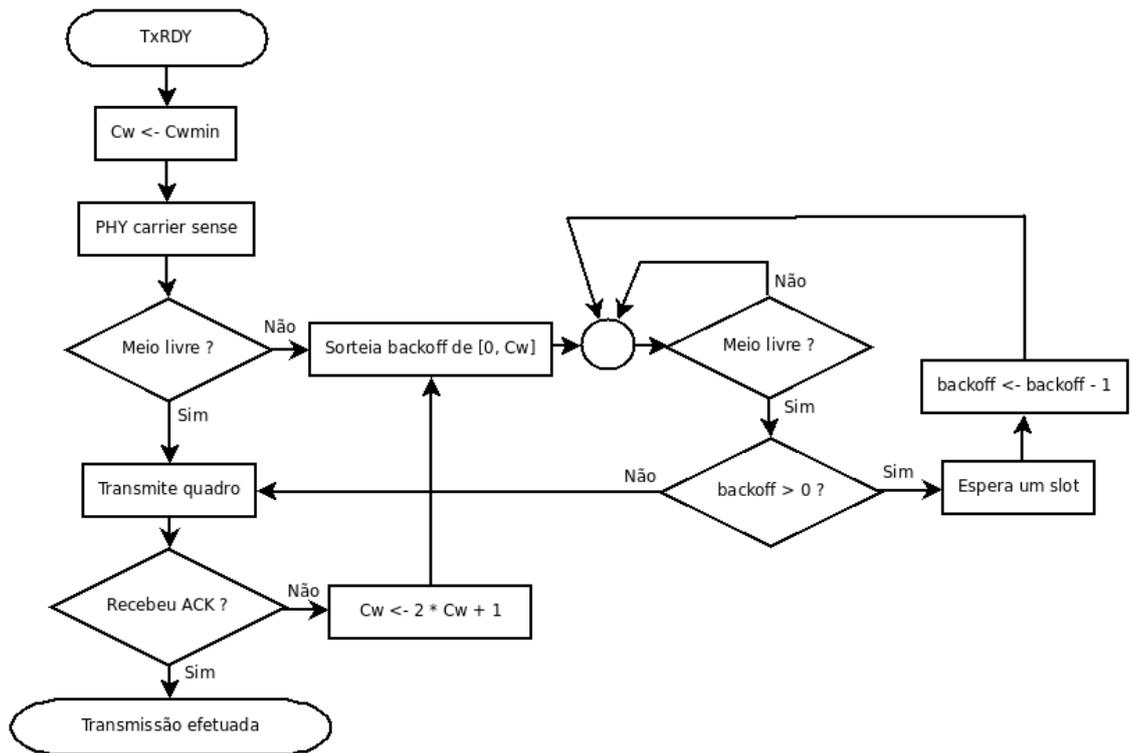


Figura 12 - Fluxograma do CSMA/CA
 Fonte: Sobral (2011)

Com relação ao quadro do 802.11, ele é formado da seguinte maneira: 2 bytes para controle do quadro, indicando versão do protocolo, tipo de quadro, se o quadro está fragmentado, informações privadas e 2 bits para *Distribution System* (DS) que determinam o significado dos 4 campos de endereços; e 2 bytes para duração da identidade do quadro, ou seja, o tempo pelo qual o meio vai ficar ocupado com esse quadro; campos de endereço de 1 a 3 com 6 bytes cada, contendo os endereços MAC; 2 bytes para controle de sequência, responsável por evitar ACK's duplicados; mais 6 bytes para outro endereço MAC (vale lembrar que os 4 endereços não precisam ser utilizados ao mesmo tempo); de 0 a 2312 bytes para dados; e 4 bytes para *Check Sum* assim como no padrão 802.3. A Figura 13 mostra mais em detalhes a formação do quadro do padrão 802.11.

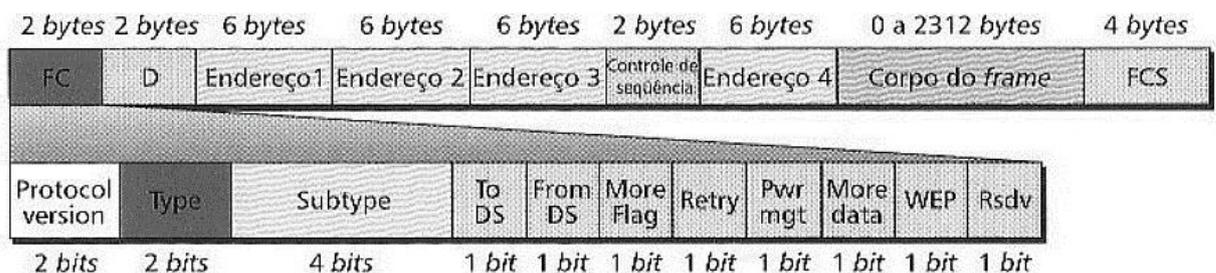


Figura 13 - Formação do quadro do padrão IEEE 802.11
 Fonte: Sobral (2014)

O entendimento aprofundado de cada um desses quadros não é necessário, pois a conversão do padrão 802.3u para o 802.11 é feita de forma transparente para o usuário pelo *Access Point*.

O padrão 802.11, assim como o 802.3, tem variações dentro de suas próprias definições. Isso quer dizer que dentro do 802.11 há os tipos 802.11a/b/g/n/ac, sendo o “ac” o modelo mais recente. Apesar da existência desses tipos diferentes dentro da definição mais ampla do 802.11, as principais diferenças se dão com relação à frequência de transmissão, número de usuários por *Access Point* e velocidade máxima de transmissão em bps (bits por segundo). (INFOWESTER, 2014)

O padrão 802.3 possui subtipos, mas eles não interferem no número de usuários por aparelho, uma vez que neste formato a conexão se dá via cabo conectado diretamente ao dispositivo. Contudo, a taxa de transmissão e o tipo de codificação do meio físico variam de um para outro.

No caso deste projeto, não há interesse de buscar outros padrões do 802.3 além do 802.3u, pois a única conexão de rede permitida pela placa embarcada é *FastEthernet*.

2.5 PLATAFORMA DE DESENVOLVIMENTO DA PLACA EMBARCADA

A IDE (*Integrated Development Enviroment*) que será utilizada para desenvolver o software do sistema é a fornecida pela NXP, fabricante do microcontrolador, e pode ser encontrada no site da empresa. Esta IDE foi desenvolvida por outra empresa, chamada CodeRed, e utiliza a linguagem C para programação. A Figura 14 ilustra a IDE utilizada.

Dentro do ambiente de desenvolvimento, que recebe o nome de LPCXpresso v4.2.3_292, encontra-se uma área para a edição do código (à direita), uma de gerenciamento do projeto (à esquerda) e um console de *debug* (na parte inferior da tela). Uma vantagem dessa plataforma é que no site da fabricante do microcontrolador existem alguns códigos de exemplo para as respectivas placas. No caso deste projeto, foi escolhida a placa LPCXpresso 1769. Assim, esses códigos poderão servir como base para a implementação do software que será utilizado neste projeto.

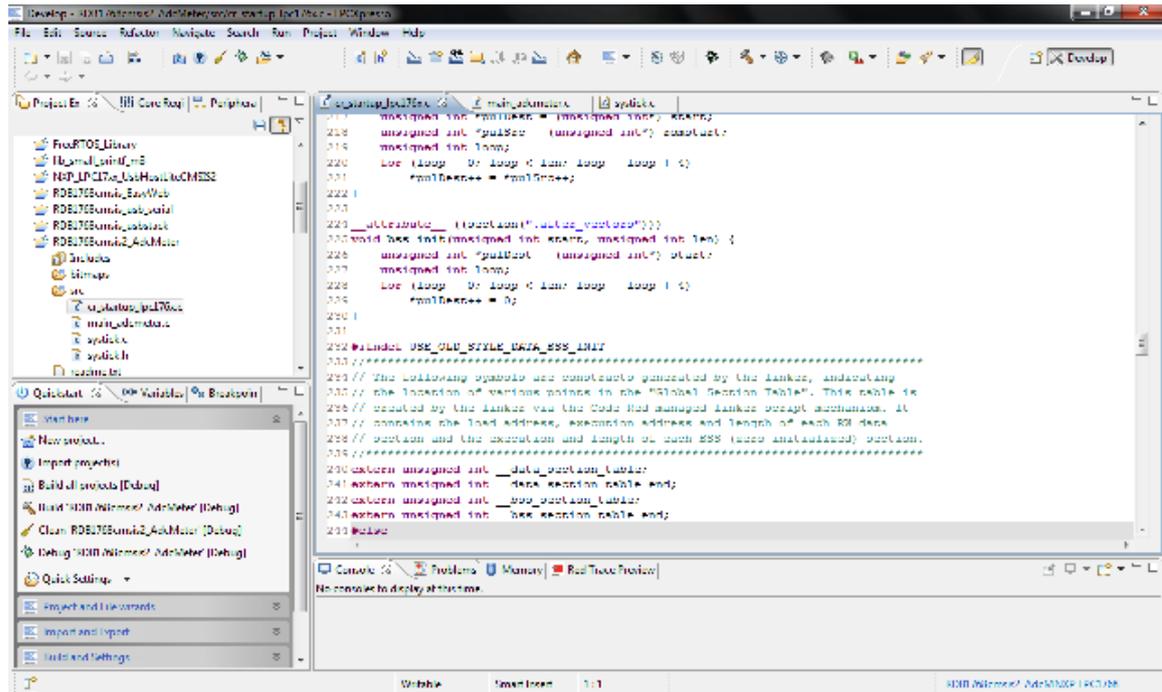


Figura 14 - Ilustração da IDE LPCXpresso
Fonte: autoria própria

2.6 SISTEMA WINDOWS PHONE

O sistema operacional escolhido para a criação do aplicativo que faz parte do protótipo desse trabalho foi o *Windows Phone*, desenvolvido pela Microsoft. A história desse sistema operacional móvel iniciou-se em 1995, originando-se dentro do *Windows CE* e ganhando a denominação própria de *Windows Mobile* em 2000. Atualmente está no lançamento 8.1, versão ao qual o aplicativo desse trabalho foi desenvolvido (AMPRIMOZ, 2011). Graças à compatibilidade com versões anteriores, o aplicativo não funciona exclusivamente na última versão disponibilizada do sistema operacional, mas com todas as versões da família do Windows Phone 8.

O grande diferencial do Windows Phone é sua interface. Diferente de qualquer outra empresa, a Microsoft desenvolveu uma linguagem de design que unifica a experiência dos usuários dentro do seu sistema. Isso significa uma padronização nos princípios de estilo do telefone nas interfaces tanto de uso das funções do sistema como também dos aplicativos criados para a plataforma. O princípio mais importante se resume em: limpo, leve, amplo e fluido (do inglês: “*Clean, Light, Open and Fast*”). (FALLE, 2012). A Figura 15 apresenta um pouco da interface do Windows Phone 8.1.



Figura 15 - Interface do sistema Windows Phone 8.1
Fonte: Sullivan (2014)

2.7 PLATAFORMA DE DESENVOLVIMENTO DO APLICATIVO

A interface de desenvolvimento de aplicativos, serviços e jogos para o Windows Phone é o próprio editor Microsoft Visual Studio. O Visual Studio possui diferentes versões e módulos, permitindo o desenvolvimento de aplicativos para toda família Windows, *sites* em diversas linguagens, jogos para computadores e consoles, *scripts* em Python e Ruby, entre outros projetos.

O Microsoft Visual Studio é disponibilizado gratuitamente em uma versão com menos recursos de análise e teste do projeto. A versão gratuita também não permite a comercialização das aplicações criadas através dele. Para estudantes, a Microsoft oferece as versões completas da sua IDE de desenvolvimento. Isso permite aos estudantes criar trabalhos e protótipos como o desta monografia, sem se preocupar em investir na compra de suítes completas de desenvolvimento. (Visual Studio, 2014)

Seguindo os avanços da Engenharia de Software, a Microsoft investiu na separação entre o desenvolvimento da interface e do processamento das informações do programa. Essa independência originou uma família de produtos que se integram com o Visual Studio para edição da interface de projetos, chamada de Microsoft Expression. A Figura 16 apresenta o Microsoft Visual Studio 2013 e a Figura 17 o Microsoft Expression Blend 4.

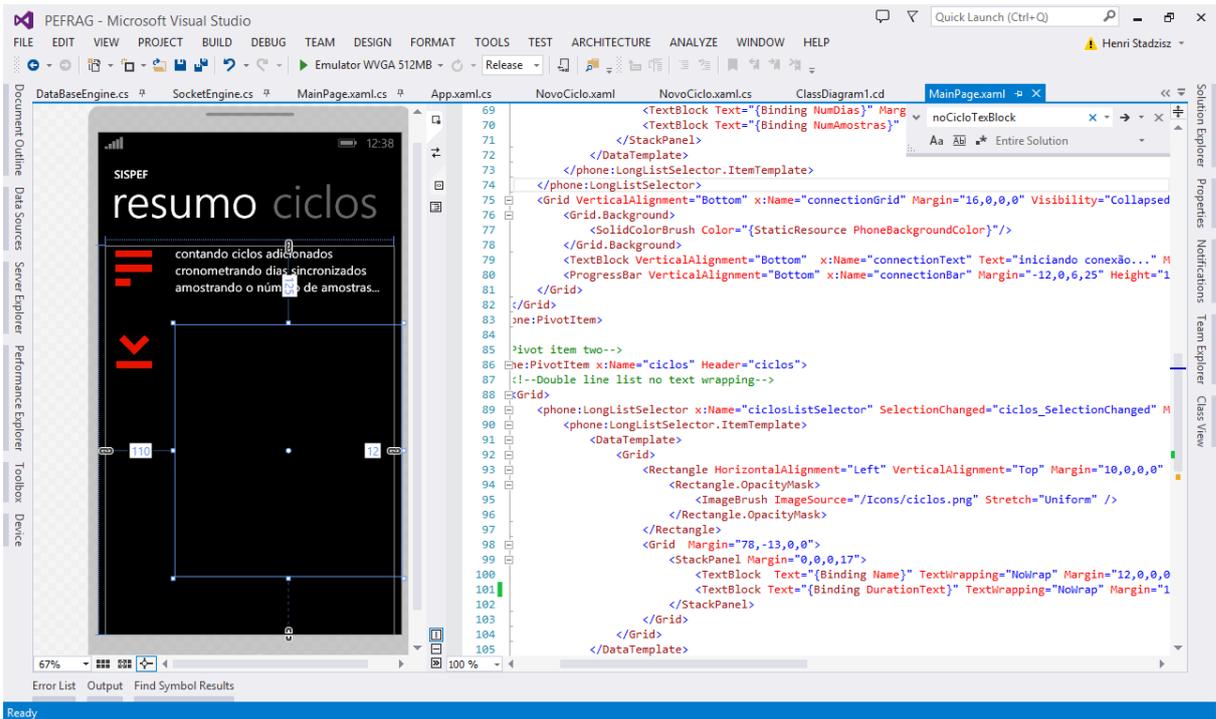


Figura 16 - Interface do Microsoft Visual Studio 2013
Fonte: autoria própria

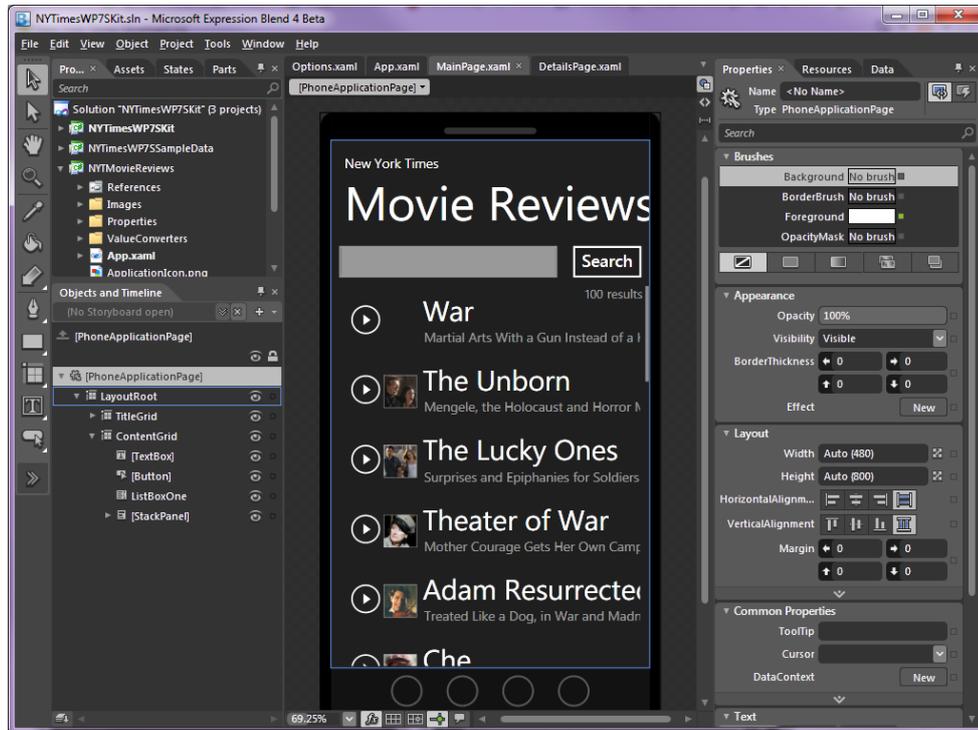


Figura 17 - Interface do Microsoft Expression Blend 4
Fonte: Synergist (2010)

3 DESENVOLVIMENTO

Este capítulo trata sobre o desenvolvimento que se seguiu de modo a alcançar os objetivos propostos para o trabalho.

3.1 VISÃO GERAL DO SISTEMA

O sistema desenvolvido é constituído por um conjunto de elementos, fundamentais para o seu funcionamento. O primeiro deles é o aplicativo para o telefone celular. Por meio dele são obtidos e armazenados os dados das pesagens feitas durante os dias de um ciclo. Nele, as informações são processadas e, a partir delas, o aplicativo é capaz de apresentar esse conjunto de informações para o acompanhamento das aves, como por exemplo em forma de histogramas diários.

A comunicação sem fio entre o telefone e a placa embarcada é feita com o auxílio de um *Access Point*. Ele fica conectado fisicamente ao microcontrolador e via WiFi com o telefone celular. Com os dois dispositivos conectados à mesma rede (intermediada pelo *Access Point*), é possível estabelecer uma conexão e enviar as informações entre eles. O microcontrolador é fundamental para o sistema, tendo em vista que suas funções são relacionadas à coleta e armazenamento de dados que são obtidos diretamente da balança. O último elemento que compõe o sistema é a balança, que faz as pesagens de forma precisa dos frangos e transmite esse valor para o microcontrolador.



Figura 18 - Elementos que compõem o sistema desenvolvido
Fonte: autoria própria

A Figura 18 mostra as partes do que compõem o sistema. A imagem mostra um celular com Windows Phone, sistema escolhido para a construção do aplicativo.

3.2 ORÇAMENTO DO PROTÓTIPO

A escolha dos componentes do sistema levou em conta, além dos aspectos técnicos, os custos de aquisição. A Quadro 3 apresenta os gastos com a compra dos componentes necessários para a construção do protótipo. O custo total do projeto foi R\$ 447,00 , do qual se destaca o preço da placa eletrônica, que foi de R\$ 54,00 a unidade. Com exceção do *Access Point*, que foi importado diretamente do país de origem, todos os componentes foram adquiridos no Brasil, mais especificamente, de fornecedores em Curitiba.

Quadro 3 - Gastos com itens do projeto

Atividade/ item	Custo aproximado
Access Point Tenda A5	R\$ 100,00
Célula de carga de 20 kg	R\$ 85,00
3x Placa NPX LPC1769	R\$ 162,00
Construção de placa impressa	R\$ 45,00
Componentes eletrônicos do filtro	R\$ 20,00
Acrílico para a estrutura do protótipo	R\$ 20,00
Estrutura da balança	R\$ 15,00
Total	R\$ 447,00

Fonte: autoria própria

Apesar do custo total ter sido alto, alguns componentes contidos no quadro já pertenciam a alguns integrantes do grupo, fazendo com que o valor desembolsado por integrante para a confecção do protótipo não fosse tão elevado.

3.3 PROJETO EMBARCADO

Essa seção apresenta como foram desenvolvidas as principais partes do projeto embarcado. Ela é bastante importante, pois nos permite ter uma visão mais clara de como implementar o sistema na prática.

3.3.1 Esquemático

Existem três conexões no microcontrolador utilizado no desenvolvimento do projeto. A primeira é uma alimentação de bateria para manter o *Real Time Clock* (RTC) funcionando independentemente de haver ou não uma outra fonte de alimentação na placa. Desta forma, a placa para fazer o controle das operações continua funcionando enquanto houver energia na bateria.

A segunda, mais crucial para o funcionamento do projeto, é a entrada do sinal analógico que vem dos filtros que estão entre a balança e o circuito embarcado. Esse sinal entra pelo pino número 15 do microcontrolador, utilizado para a conversão analógico-digital, assim como definido pelo manual do usuário da placa embarcada.

Tem-se também, com intuito de possibilitar a conexão entre a placa e o dispositivo móvel, quatro pinos que são utilizados para interfacear a rede *Ethernet*: estes são os pinos números 32 até 35 ligados nos pinos 2, 1, 6 e 3 do conector tipo RJ45 fêmea, respectivamente. A Figura 19 apresenta esse modelo.

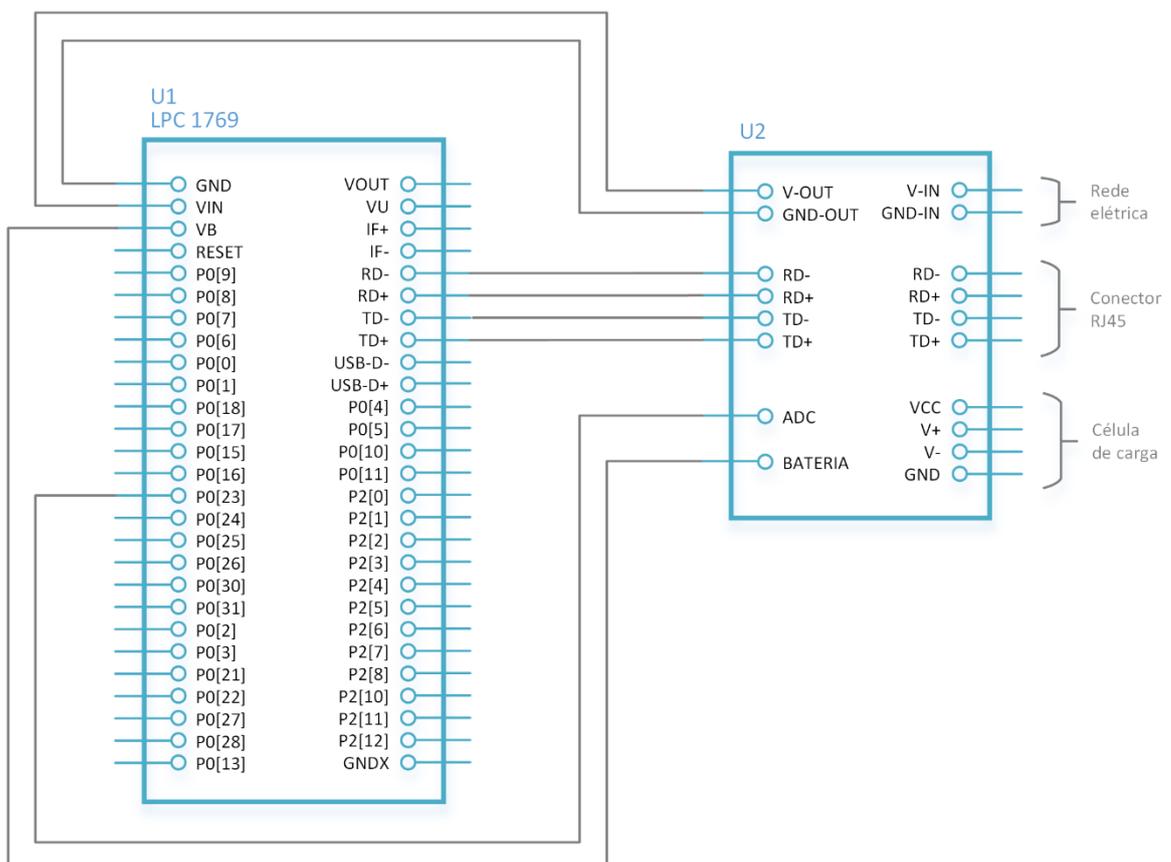


Figura 19 - Esquemático de ligação da placa embarcada
Fonte: autoria própria

3.3.2 Desenvolvimento do software

O desenvolvimento do software da placa embarcada envolveu três fases principais. Primeiramente foi desenvolvido o software de controle do conversor analógico-digital (ADC). Este software é controlado por um RTC e por um *Repetitive Interrupt Timer* (RIT). Desta forma pode-se controlar quantas amostras são feitas em cada período de tempo. No caso deste projeto, a especificação determinava que seria necessário utilizar o ADC a cada 30 segundos para adquirir uma amostra de peso aplicado na balança. A criação do algoritmo de tratamento desses dados também se deu paralelamente à essa fase do desenvolvimento, porém foi efetivamente implementado apenas no final, pois não interferia na captação dos dados e dependia de outras definições que seriam feitas em fases posteriores.

A segunda parte do software embarcado consistiu em desenvolver as estruturas de dados e, só então, desenvolver o código de gerenciamento de memória. Como a placa embarcada utilizada tem uma memória não volátil (Flash) limitada a 64 kilobytes e esse tipo de memória tem um número limitado de reescritas, o projeto teve que levar em consideração que o espaço ocupado por cada tipo de variável deveria respeitar as capacidades de cada bloco de memória e, dentro desses blocos, a capacidade de cada setor. Por causa dessas limitações de memória, o projeto das variáveis, por vezes, faz com que elas sejam maiores do que o necessário em relação a valores utilizados. Contudo, esses tamanhos são justificados pela necessidade de completar esses blocos de memória sempre com a mesma quantidade de dados para não desperdiçar reescritas.

A terceira e última fase de desenvolvimento na placa embarcada envolveu determinar padrões de comunicação que seriam utilizados para a interação da mesma com o celular. Esse protocolo está apresentado no Quadro 4, presente na seção Protocolos de Comunicação.

3.3.3 Classe Principal

O diagrama presente na Figura 20 mostra a sequência de ações realizadas pela função principal do código embarcado, responsável por controlar tudo que a placa precisa fazer.

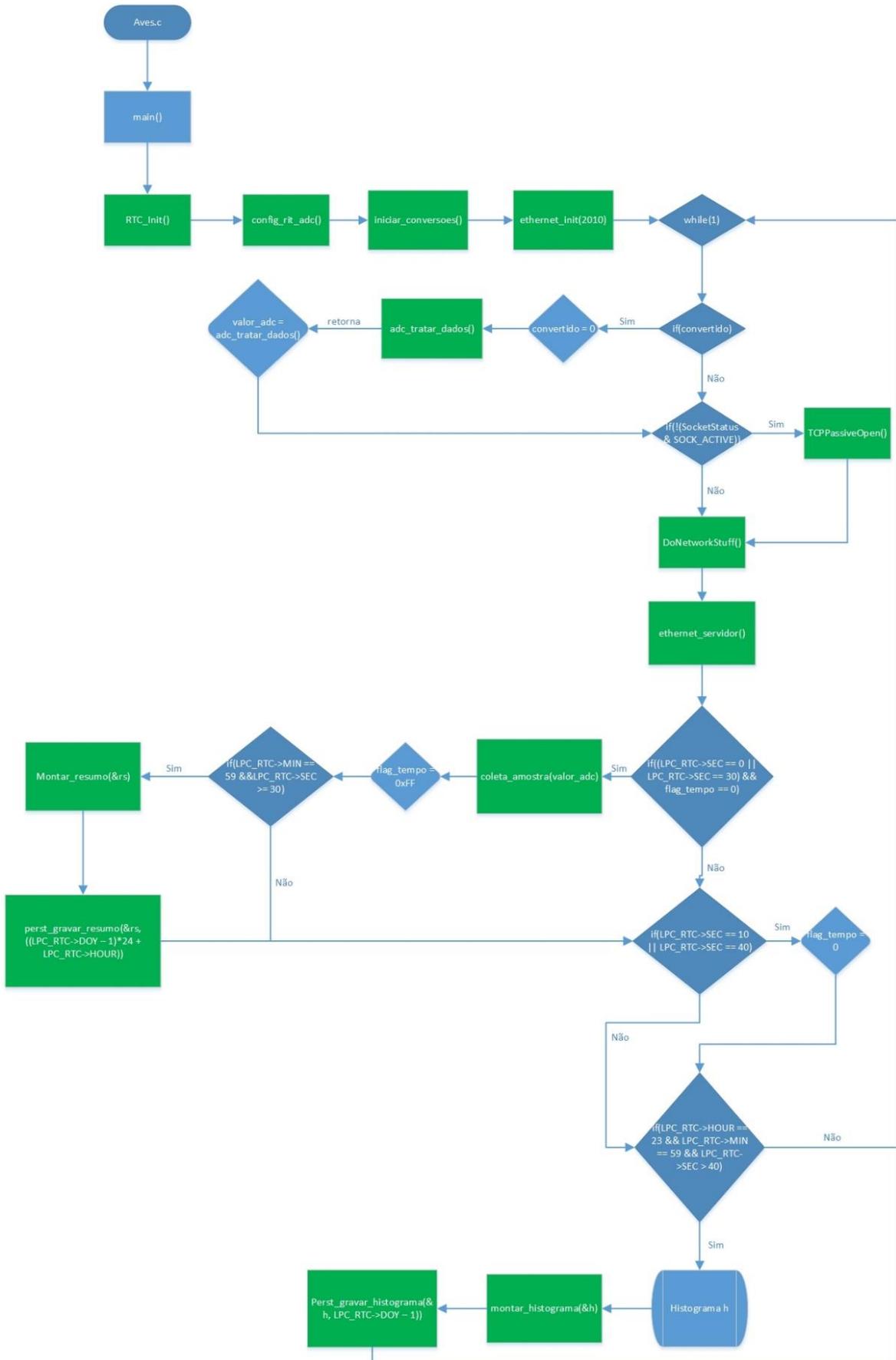


Figura 20 - Diagrama do funcionamento da função principal
Fonte: autoria própria

Ao iniciar o sistema embarcado o software faz algumas inicializações através de chamadas de funções externas, indicadas em verde, e começa um loop infinito de aquisição e tratamento dos dados. Esse tratamento é seguido por uma abertura do *socket* para envio e recebimento dos dados caso seja solicitado.

Após o encerramento da comunicação o sistema verifica se precisa realizar uma nova medida, feita a cada 30 segundos. O resumo dos dados coletados durante uma hora é montado após o fechamento da hora e o histograma do dia é calculado a partir das amostras coletadas durante todo o dia.

3.3.4 Classificação de amostras

Para determinar o número de galinhas ou quando a amostra deve ser descartada, é preciso levar em consideração o peso máximo de cada galinha naquele dia do ciclo e quantas galinhas no máximo estarão em cima da balança ao mesmo tempo. O diagrama presente na Figura 21 mostra como o algoritmo define quantas galinhas estão presentes na amostra adquirida, assumindo um desvio aceitável no peso das galinhas de 10% do peso que elas deveriam ter naquele dia. Caso a medida não esteja dentro desse desvio, a amostra é descartada. Caso essa amostra seja menor do que 60% do peso definido pela curva padrão naquele dia, essa medida é considerada como sujeira. Na Figura 21, o eixo y representa uma razão do peso esperado para uma galinha no dia em questão.

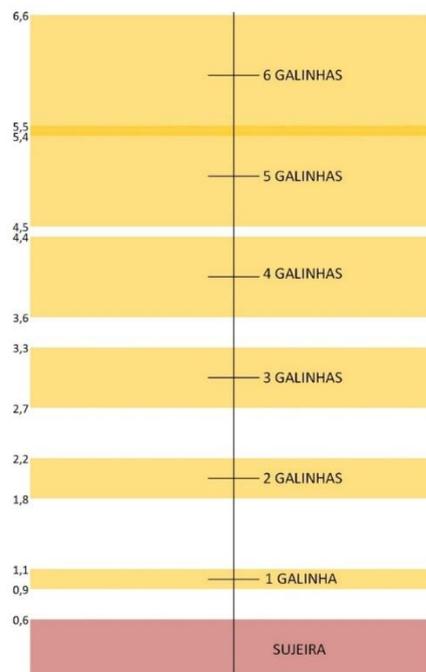


Figura 21 – Faixas de interpretação do peso adquirido
Fonte: autoria própria

Percebe-se que a partir de seis galinhas o algoritmo começa a sobrepor valores, interferindo na capacidade de determinação de quantas galinhas estão na medida. Porém, como a balança está limitada a vinte quilos e o peso máximo de uma galinha é de aproximadamente cinco quilos, definiu-se que o número máximo de galinhas em uma amostra é de cinco, eliminando-se a área de erro.

3.3.5 Estrutura dos dados

Essas estruturas de dados são preenchidas com os valores obtidos da balança e calculados pelo algoritmo, e são salvos na memória flash (NVRAM) do microcontrolador. Os dados são gravados em conjuntos de 256 bytes de forma a aproveitar da melhor maneira possível o número máximo de reescritas de cada bloco de memória, aumentando o número de ciclos que podem ser medidos antes que seja necessária a substituição da memória flash.

Quadro 4 - Estrutura de Histograma

ID	flag	barra[9]	amostras_descartadas	dia[2]	mes[2]	ano[4]	padding[16]	checksum
TIPO	uint8	Barra	uint16	uint8	uint8	uint8	uint8	uint8

Fonte: autoria própria

A estrutura dos histogramas está representada no Quadro 4. Ela é formada por uma **flag** que marca posições vazias de memória, uma variável **amostras_descartadas** que determina o número de amostras que foram consideradas erradas, o dia, o mês e o ano do calendário do controlador; um **padding** para preencher os espaços de memória que ficariam vazios dentro do setor da memória, um **checksum** para verificar que nenhum dado foi corrompido e nove barras de 32 bits do tipo **Barra**, explicado a seguir.

Quadro 5 - Estrutura de Barra

ID	Quantidade	peso
TIPO	uint16	uint16

Fonte: autoria própria

Uma **Barra** é utilizada apenas pela estrutura **Histograma** e elas são formadas por uma quantidade de galinhas que se encaixam nessa faixa de peso, definida pela variável **quantidade**. A variável **peso** determina o peso que essa **Barra** vai

representar. O tipo dessas variáveis que compõem a barra se encontram no Quadro 5.

Quadro 6 - Estrutura de ResumoHora

ID	flag	numero	peso_medio	padding[10]	Checksum
TIPO	uint8	uint16	uint16	uint8	uint8

Fonte: autoria própria

O resumo da hora que será feito no final de cada hora e então salvo na memória flash, é formado por uma **flag** para determinar espaços vazios na memória, um **numero** que marca quantas galinhas foram medidas nessa hora, um **peso_medio** utilizado para guardar o valor médio do peso dessas galinhas medidas, um **padding** para preenchimento e um **checksum** para determinar se não houve perda ou comprometimento dos dados. O Quadro 6 apresenta detalhadamente os tipos de variáveis do resumo da hora que serão armazenadas.

Quadro 7 - Estrutura de Amostra

ID	flag	numero	peso	padding	Checksum
TIPO	uint8	uint16	uint16	uint8	uint8

Fonte: autoria própria

O Quadro 7 apresenta os tipos de variáveis que compõem a estrutura **Amostra** (que também será armazenada na memória), que possui também uma **flag** para determinar espaços vazios que fica, assim como nos outros, em 0xFF caso a leitura seja inválida; um **numero** que é determinado pelo algoritmo baseando-se no peso e no desvio máximo aceito, e apresenta quantas galinhas foram medidas naquela amostra, um **peso** que é o peso total medido, um **padding** para preenchimento de memória e um **checksum** para garantir a integridade dos dados.

3.4 INTERPRETAÇÃO DO SINAL DA BALANÇA

Essa seção apresenta o desenvolvimento da construção das lógicas de aquisição e interpretação do sinal proveniente da célula de carga e que chega ao microcontrolador.

3.4.1 Sinal bruto

A medição de peso deste projeto é feita com a estrutura de uma balança usada da marca Urano, cuja célula de carga está acoplada. Sua capacidade nominal é de 20 kg, mas ela não possui identificação suficiente em seu corpo para que se possa levantar seu conjunto de dados técnicos.

Para tanto, foi experimentalmente verificado que a sua resistência nos terminais de alimentação é 404 Ω , devendo igualmente ser a resistência de cada braço da ponte. É um valor próximo aos 350 Ω mencionados em Interface Inc. (2013). Do mesmo modo, a sensibilidade foi medida obtendo-se aproximadamente 2,05 mV/V (como esperado pela fundamentação teórica), resultando em 18,4 mV de diferença de tensão na ponte - para 20 kg - se a alimentação fornecida for de 9 V.

A característica do sinal de saída da ponte de medição de uma célula de carga é naturalmente ruidosa e sujeita a derivas ao longo do tempo, como se pode ver em Slattery e Nie (2005), o que traz a necessidade de filtragem. De modo a escolher a sequência necessária de filtragem, foram levantados sinais de exemplo usando a célula disponível e um circuito simples, composto apenas de um amplificador de instrumentação e um divisor de tensão (para adequar o sinal à faixa do conversor analógico-digital de 12 bits, com tensão de referência de 3,3 V).

Um dos sinais obtidos é o da Figura 22, capturado com a placa de desenvolvimento a ser utilizada no protótipo final do projeto. Neste contexto, a balança foi mantida estática, sem nenhum peso exceto o de seu prato.

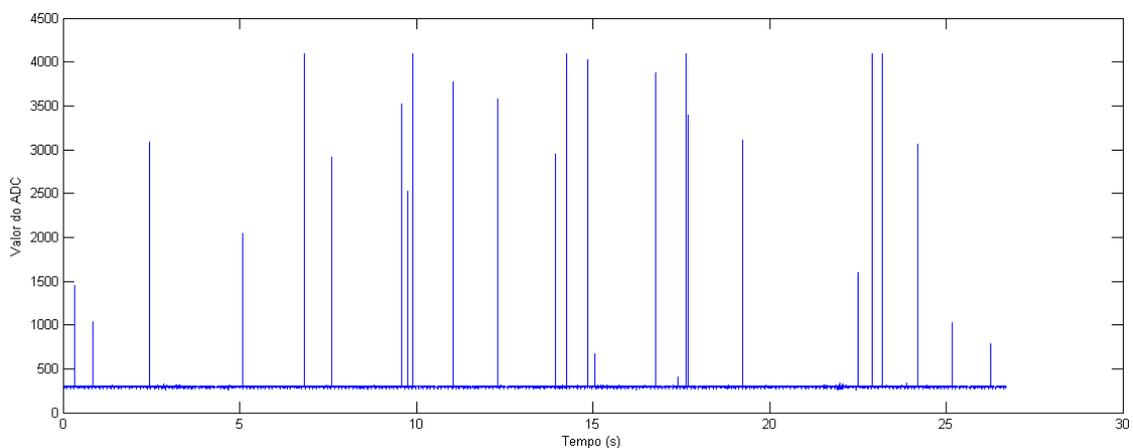


Figura 22 - Sinal de exemplo capturado com a placa de desenvolvimento
Fonte: autoria própria

Numa duração de 27 segundos, é visível a presença de amostras com valores inválidos. Estas acontecem comumente para este modelo de placa de desenvolvimento (MBED, 2009, MBED, 2011). A maioria das sugestões feitas em NXP Semiconductors (2010) foram testadas, mas não produziram efeito (RIGAUD, 2011); as que não puderam ser testadas não o foram por motivos de incompatibilidade técnica com os objetivos do projeto.

Também nota-se o ruído de baixa amplitude que causa variação ao sinal contínuo esperado, natural em sistemas de aquisição com 12 bits (NXP SEMICONDUCTORS, 2010).

Na Figura 23, destaca-se uma das regiões em que há falha na captura pelo conversor. Pode-se notar que não costumam acontecer seguidamente e que são valores de destaque (valores elevados).

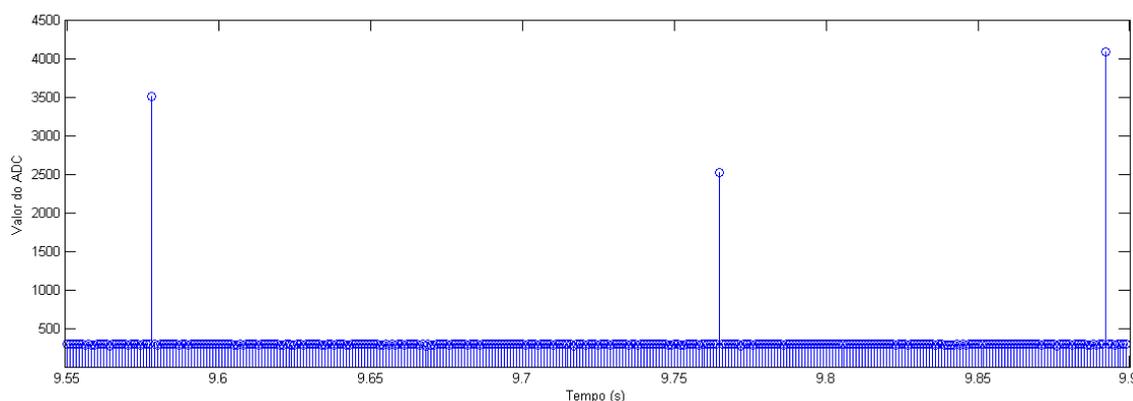


Figura 23 - Destaque para erros de captura
Fonte: autoria própria

3.4.2 Circuito montado

De modo a condicionar o sinal para a sua captura pelo conversor analógico-digital do microcontrolador, o circuito externo foi projetado de modo a regular a tensão de alimentação da ponte de medição e amplificadores, amplificar o sinal e realizar uma filtragem analógica básica, sendo que uma filtragem mais aprimorada será feita digitalmente.

O circuito desenvolvido está apresentado na Figura 24. É composto de 6 seções: regulador de tensão (A), ponte de medição (B), amplificador de instrumentação (C), divisor resistivo (D), filtro passa-baixas de primeira ordem (E) e resistência de acoplamento (F).

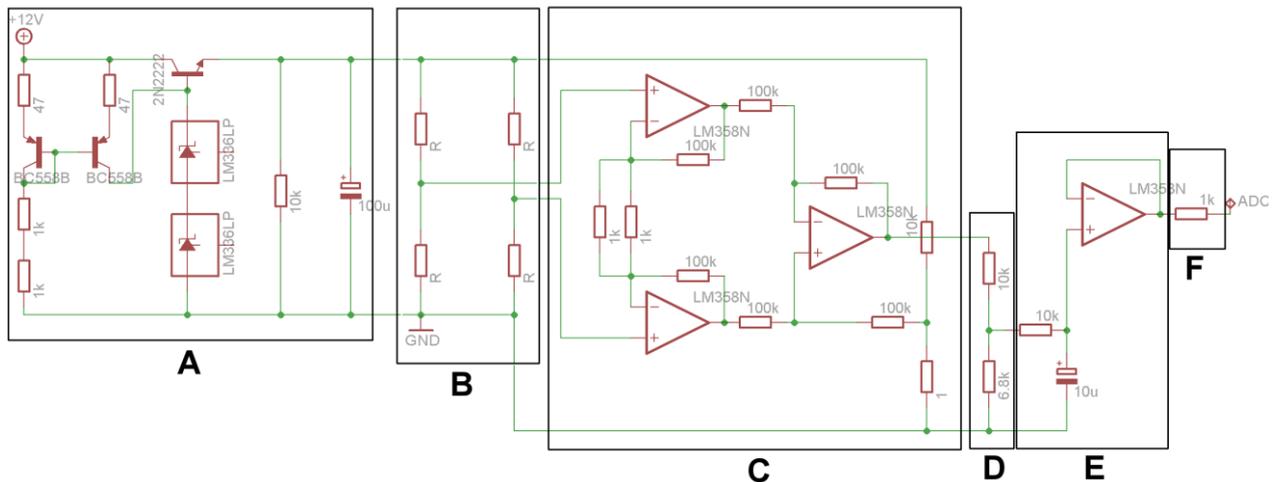


Figura 24 - Esquemático do condicionamento analógico
 Fonte: autoria própria

Na seção A encontra-se um regulador de tensão simplificado, que utiliza um espelho de corrente para alimentar duas referências de tensão de 5.0 V ligadas em série. O espelho de corrente reduz a sensibilidade da corrente que passa nas referências em relação à tensão da fonte de alimentação, e também fornece a corrente de base para o transistor em coletor comum, que deverá alimentar o restante do circuito a aproximadamente 0,7 V a menos que a tensão das referências, portanto 9,3 V. A tensão resultante é apropriadamente baixa para que não cause aumento de temperatura e desregulação da ponte de medição, como se pode ver nas considerações em Interface Inc. (2013). Esta seção foi baseada em Appaloosa (2012). O propósito desta seção é manter a tensão no circuito o mais estável possível, para que mudanças na alimentação não influam de forma significativa na calibração do sistema, para além de evitar ruídos vindos da rede elétrica.

A seção B consiste na ponte de medição, cuja teoria já foi revista. A diferença de tensão esperada na saída é 19,07 mV (considerando a tensão de alimentação de 9,3 V e a sensibilidade de 2,05 mV/V).

A seção C é um amplificador de instrumentação implementado a partir de componentes LM358, que possuem boa resposta para sinais quando a sua alimentação não é simétrica. Este circuito foi baseado na folha de dados do componente (TEXAS INSTRUMENTS, 2013), com uma modificação que permite um ajuste de desvio de tensão no estágio de saída do amplificador, de modo a compensar qualquer tensão negativa que possa haver quando o peso sobre a balança for o mínimo. O ganho de tensão é 401, produzindo na saída um sinal máximo de 7,65 V (saída de 19,07 mV multiplicado pelo ganho 401).

A seção D reduz a tensão de saída do amplificador para que seja aceitável ao conversor analógico-digital. Optou-se por utilizar um divisor de tensão ao invés de reduzir o ganho do amplificador para que as tensões de deslocamento dos amplificadores operacionais tivessem também uma redução proporcional ao divisor, fazendo com que a razão entre sinal e deslocamento fosse melhorada. Para uma tensão de 8 V, os valores de resistência indicados produzem 3,24 V, um pouco menos que o valor máximo admissível pelo conversor analógico-digital.

A seção E forma, junto com a seção D, um filtro passa-baixas de primeira ordem com corte em 1,13 Hz, condicionando o sinal para amostragem. A frequência de amostragem projetada é de 1 kHz, portanto à taxa de Nyquist, 500 Hz, a atenuação deverá ser de aproximadamente 53 dB. O propósito desta seção é reduzir o ruído de alta frequência do sinal antes que seja passado ao conversor analógico-digital.

A seção F consiste de uma resistência para acoplar a saída do filtro à entrada do conversor. Apesar de não haver instruções quanto à sua necessidade, notou-se experimentalmente que há uma significativa redução nos valores de conversão errados quando se utiliza esta resistência e, portanto, este é o seu propósito. A documentação do microcontrolador define uma resistência máxima de 7,5 k Ω para este valor (NXP SEMICONDUCTORS, 2014).

3.4.3 Conversão e filtragem

Após a passagem pelo circuito de aquisição completo, o sinal adquirido pelo conversor analógico-digital é praticamente igual ao já apresentado na Figura 22. Para que se possa realizar uma leitura de peso com alguma estabilidade, foram feitas filtragens de modo a remover os valores incorretos, reduzir a presença de ruído de alta frequência inerente ao alto número de bits e ainda estabilizar o valor final quanto a pequenas variações em torno de um valor médio.

Para tanto, a sequência de operações do diagrama na Figura 25 foi desenvolvida com base nas características observadas do sinal na Figura 22.

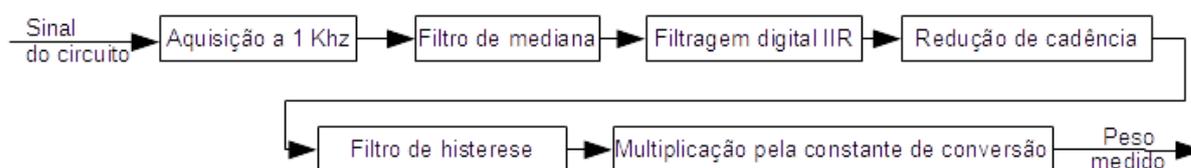


Figura 25 - Operações para tratamento do sinal capturado
Fonte: autoria própria

O primeiro passo consiste na aquisição do sinal tratado pelo circuito analógico. Utiliza-se o conversor analógico-digital de 12 bits incorporado ao microcontrolador da placa de desenvolvimento. A frequência de amostragem foi definida como 1 kHz, uma baixa frequência visto que a movimentação dos animais e por consequência a velocidade de variação de peso sobre a balança é lenta. A Figura 22 é um exemplo de sinal adquirido.

Após a aquisição, busca-se eliminar as medições incorretas que ocorrem esporadicamente no conversor incorporado no microcontrolador. Optou-se por uma filtragem da mediana com 8 valores: guardam-se as últimas 8 amostras capturadas, que são ordenadas de acordo com o seu valor, e escolhe-se como saída do bloco o quarto valor mais baixo. O número de valores foi determinado pela observação de outros cenários de aquisição em que se pudessem eliminar todas as amostragens incorretas, mas que também simplificasse a necessidade de processamento. Tendeu-se também a escolher um valor abaixo do central (o quarto em vez da média entre o quarto e o quinto) por que as amostragens incorretas ocorrem em geral para valores muito maiores que a tendência real. A Figura 26 exibe o sinal da Figura 22 após a passagem pelo filtro da mediana. Nota-se a remoção completa dos valores incorretos e destaca-se agora o ruído de altas frequências.

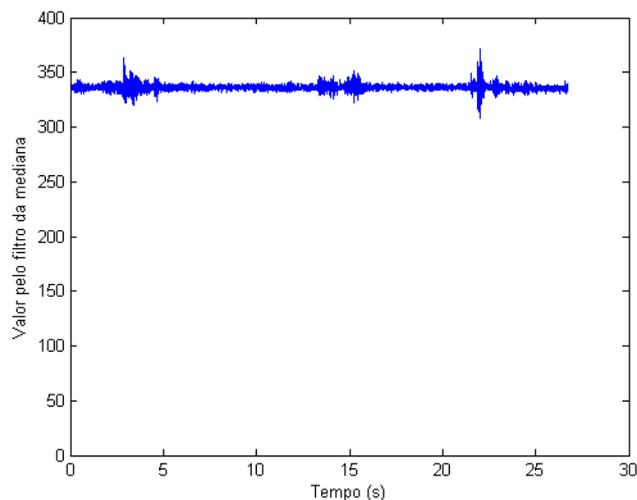


Figura 26 - Sinal da Figura 22 submetido à filtragem da mediana
Fonte: autoria própria

Procede-se em seguida à filtragem digital do sinal pelo modelo de resposta ao impulso infinita (IIR), que consiste em 4 filtros do tipo *Butterworth* de primeira ordem, passa-baixas. Os coeficientes dos filtros foram gerados com o software MATLAB versão 2012b, com frequências de corte em 20 Hz, 10 Hz, 5 Hz e 2,5 Hz. A escolha

destas frequências busca atenuar fortemente as altas frequências e permite que as baixas sejam preservadas sem que a resposta ao impulso tenha um tempo de estabelecimento muito longo. Neste caso, este valor fica em torno de 0,5 segundos, admissível pela baixa velocidade de movimentação dos animais sobre a balança. Não há oposição ao uso de filtragem de resposta infinita, visto que o sinal contínuo é o mais importante, não havendo quaisquer problemas em relação à resposta em fase.

O software gera os coeficientes utilizando a equação clássica de *Butterworth* para filtragem analógica (OPPENHEIM et. al., 1999), e utiliza a transformação bilinear para que passem para o plano Z, resultando em uma função de transferência como da equação 6 (OPPENHEIM et. al., 1999).

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_0 z^{-1}} \quad (6)$$

Para as frequências especificadas, os coeficientes utilizados encontram-se no Quadro 8. É facilmente visível que a localização dos zeros é sempre -1, pois b_0 e b_1 são iguais. Do mesmo modo, nota-se a posição dos polos sobre o eixo real, distanciando-se cada vez mais do valor 1 quanto mais aumenta a frequência.

Quadro 8 - Coeficientes dos filtros digitais IIR

	2,5 Hz	5 Hz	10 Hz	20 Hz
b0 e b1	0,00779293629195155	0,0154662914031034	0,0304687470912538	0,0591907038184055
a0	-0,984414127416097	-0,969067417193793	-0,939062505817492	-0,881618592363189

Fonte: autoria própria

A implementação dos filtros em software é realizada utilizando a estrutura em forma direta II em blocos de segunda ordem, agrupando os filtros em pares e gerando os novos coeficientes pela multiplicação das funções de transferência (OPPENHEIM et. al., 1999).

A resposta em frequência em escala linear deste filtro é apresentada na Figura 27. Nota-se a forte atenuação das baixas frequências. No detalhe à direita, vê-se a faixa de 0 a 10 Hz.

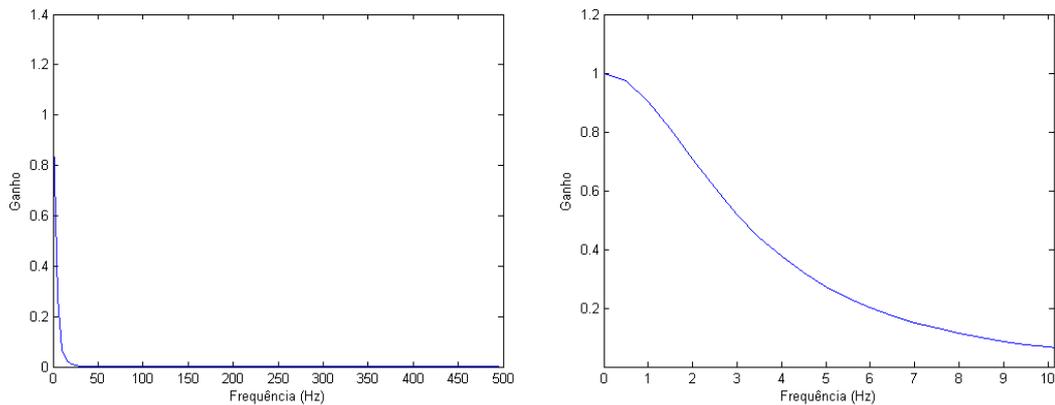


Figura 27 - Resposta em frequência dos filtros digitais IIR
Fonte: autoria própria

A Figura 28 mostra o sinal da Figura 26 após a submissão a este filtro. Pode-se ver que o ruído de alta frequência antes presente foi quase completamente removido; porém, no detalhe à direita, ainda há a necessidade de estabilizar o resultado de modo a que o valor inteiro resultante espelhe o peso sem mudanças bruscas. No mesmo detalhe, pode-se notar o tempo de estabilização.

De modo a se deixar o sinal mais estável, o sinal que sai do bloco de filtragem IIR é submetido primeiramente a uma redução de cadência, de 1000 Hz para 50 Hz, portanto 20 vezes menor. Para isto, o sinal é acumulado em grupos de 20 amostras, as quais geram um valor médio a ser usado pelo próximo bloco. Esta redução faz com que variações nos valores das amostras sejam menores, mas com uma boa resposta a movimentações na balança. A Figura 29 mostra o resultado do sinal da Figura 28 após a redução de cadência.

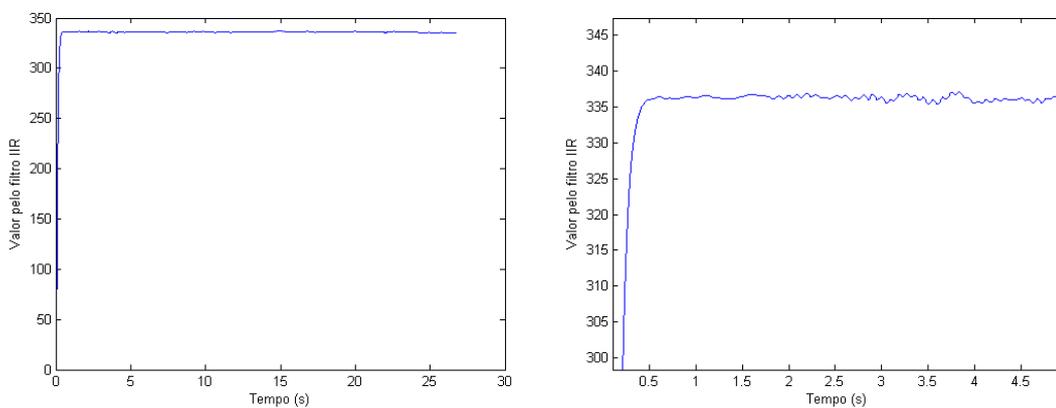


Figura 28 - Sinal da Figura 26 após passagem pelo filtro IIR
Fonte: autoria própria

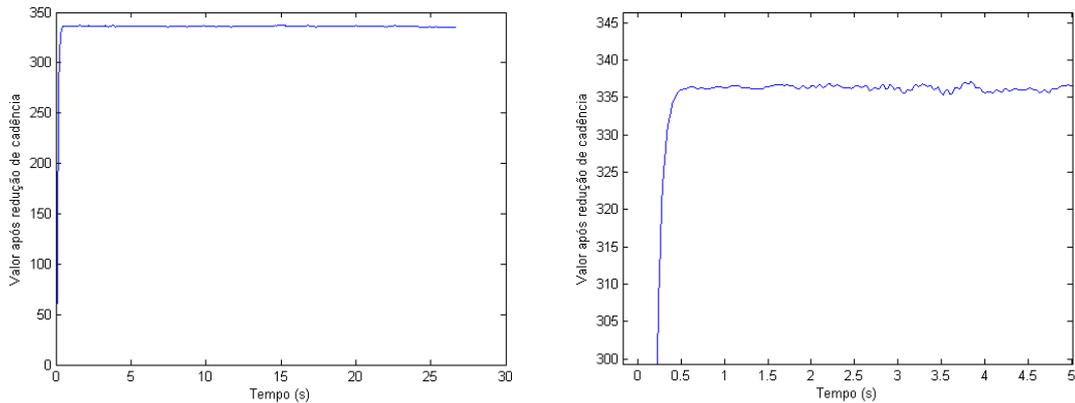


Figura 29 - Sinal da Figura 28 submetido a uma redução de cadência
Fonte: autoria própria

Finalmente, para que se possa chegar a um valor inteiro que represente o peso medido, é utilizado um redutor de variações baseado em histerese: define-se uma faixa de valores limite em que o sinal pode variar, experimentalmente definido como 4 unidades do ADC, e o valor de saída é o valor central da faixa; se o sinal ultrapassar o limite superior ou inferior desta faixa, as faixas são realocadas de modo a acompanhar o sinal para que sempre permaneça dentro dos limites. As faixas movem-se em valores reais, enquanto que o valor de saída do bloco é o ponto central entre as faixas truncado para um valor inteiro. Assim, a medição de saída é grandemente estabilizada, com uma significativa redução da resposta a variações de baixa amplitude.

A Figura 30 mostra uma sobreposição do sinal da Figura 29 com o resultado desta filtragem, além de incorporar os limites de histerese definidos a cada amostra liberada pelo bloco anterior. É visível pela linha central a redução da variação do sinal. Observe-se que o sinal a ser filtrado corresponde ao peso da balança em descanso, durante todo o tempo.

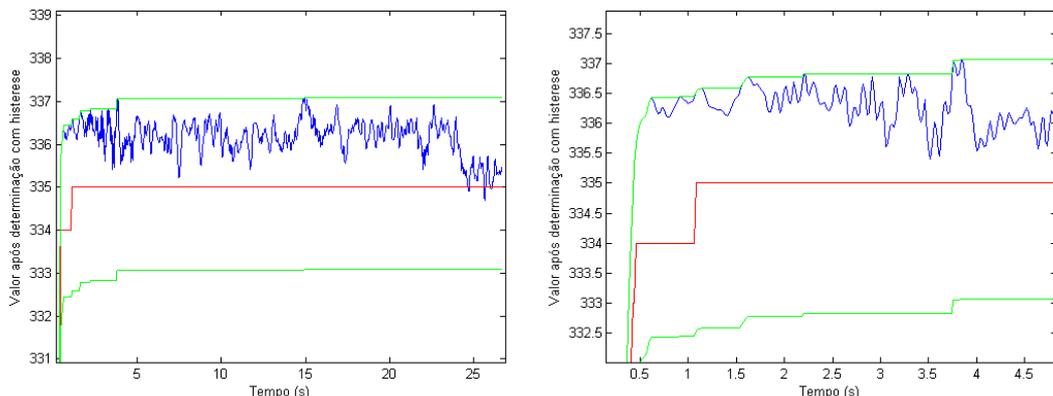


Figura 30 - Sinal da Figura 29 submetido à determinação por histerese
Fonte: autoria própria

Com uma redução da faixa de valores da histerese é possível reduzir o erro em relação à tendência média do sinal, mas as variações também aumentarão. A Figura 31 mostra o mesmo sinal, mas para uma faixa de valores de mais ou menos 1 unidade do ADC.

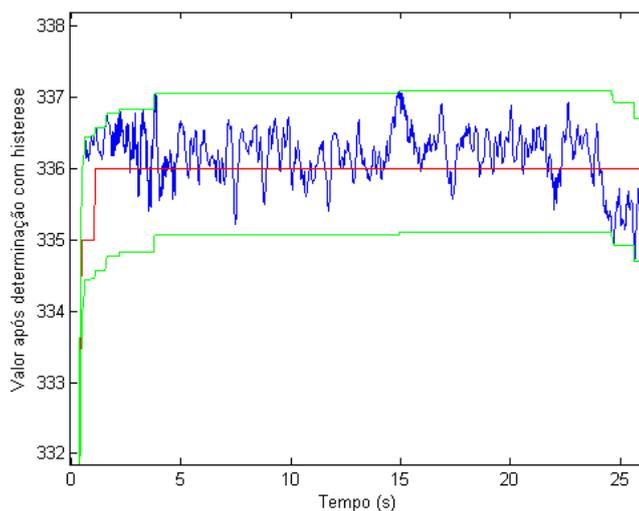


Figura 31 - Sinal da Figura 29 submetido à determinação por histerese com mais ou menos uma unidade do ADC
Fonte: autoria própria

Para converter o resultado final das filtragens para um valor em gramas, foi primeiramente medido o valor para a balança com um recipiente vazio e depois se encheu o recipiente com 15 litros de água, o que é aproximadamente 15 kg, donde resultou que cada unidade inteira da saída equivale a 5,2178 gramas. A medida foi repetida 4 vezes, sendo este resultado uma média. A medição repetitiva de um mesmo peso resultou em um desvio padrão de 13,71 gramas. Quando em utilização corrente, a tara da balança é variável e muda com o tempo e a temperatura (SLATTERY e NIE, 2005), cabendo ao algoritmo de classificação dos pesos das aves realizar o desconto.

De modo a verificar a linearidade da relação entre o peso e as medições resultantes desta sequência de operações, levantou-se os valores para uma série de incrementos de pesos utilizando novamente água e um recipiente sobre a balança, o que resultou na curva da Figura 32. A linha de tendência é uma regressão a partir dos valores levantados. A correlação entre os valores de peso incrementais e a medição é de 0,999998, mostrando a boa linearidade entre os dois.

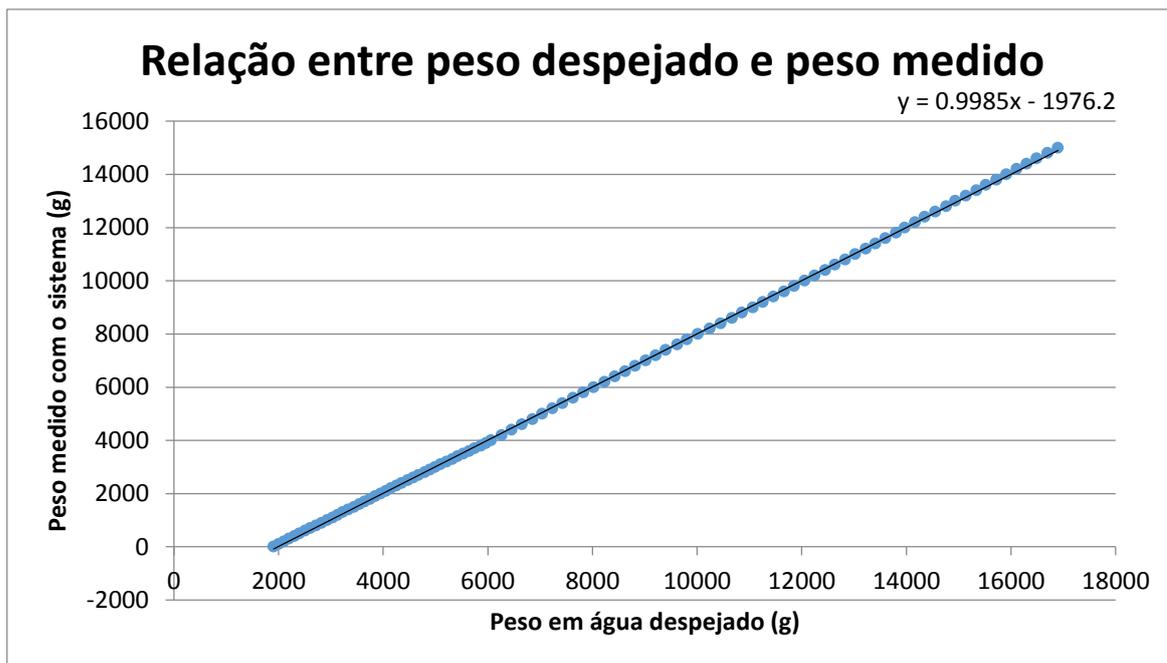


Figura 32 - Relação entre peso despejado no recipiente e o peso medido com o sistema de aquisição
Fonte: autoria própria

3.5 PROTOCOLOS DE COMUNICAÇÃO

A tabela que representa o protocolo de comunicação entre o aplicativo para o Windows Phone e o microcontrolador está apresentado no Quadro 9. Esse protocolo foi criado para organizar o padrão de envio e recebimento de dados.

O primeiro caractere presente no pacote enviado corresponde ao comando específico que está sendo submetido. O último caractere presente no pacote sempre será “#”, indicando o término a mensagem transmitida pelo outro dispositivo. O interior da mensagem é composto por dados, que variam em cada requisição.

A primeira linha, por exemplo, representa a mensagem com função de enviar os dados de um dia específico. Quando os dados são enviados do celular para o microcontrolador no formato definido para essa mensagem, ele irá verificar o primeiro caractere. Ao detectar o caractere “E”, específico dessa função, seguido de um número inteiro e depois de um caractere “#”, ele irá imediatamente responder com um “O” (caractere utilizado pelo microcontrolador para informar que o comando pedido pelo celular foi executado), depois irá colocar na resposta as informações do histograma do dia pedido pelo celular e colocará finalmente o caractere “#”.

Quadro 9 - Protocolos de comunicação

Função	Parâmetros	Resposta
Enviar os dados do dia pedido.	'E' + uint8_t dia + '#'	'O' + Histograma (inclui a data de calendário), 24 resumos de hora, + '#'
Criar novo ciclo; apagar tudo.	'N' + uint32_t ID_do_ciclo, uint8_t duração_em_dias + '#'	'O#'
Atualizar a data atual do ciclo	'A' + uint8_t dia_do_ciclo + '#'	'O#'
Verificação de ID do ciclo	'I#'	'O' + uint32_t ID_do_ciclo + '#'
Verificação do dia do ciclo da placa	'D#'	'O' + uint8_t dia_do_ciclo + '#'
Enviar o calendário da placa	'H#'	'O' + uint8_t dia, uint8_t mes, uint16_t ano, uint8_t hora, uint8_t minuto, uint8_t segundo + '#'
Configurar o calendário a partir do telefone	'U' + uint8_t dia, uint8_t mes, uint16_t ano, uint8_t hora, uint8_t minuto, uint8_t segundo + '#'	'O#'
Obter o último valor filtrado do ADC	'Q#'	'O' + uint16_t valor + '#'
Erro na interpretação do comando	-	'X#'
Forçar o valor atual como tara	'T#'	'O#'

Fonte: autoria própria

3.6 MODELO ESTRUTURAL DO APLICATIVO

O aplicativo criado é estruturado em cinco grandes classes, que estão diretamente ligadas às páginas que exibem conteúdo para o usuário. Por esse motivo, cada uma delas possui um código visual *Extensible Application Markup Language* (XAML) atrelado a ela. A Figura 33 apresenta essas classes com seus respectivos nomes. A seção de Interface da Aplicação exibe e detalhada a função de cada página ligada à essas classes.



Figura 33 - Diagrama das principais classes do aplicativo
Fonte: autoria própria

Além das classes descritas, foram criadas duas *engines*, parte independente do programa que reúne um conjunto de ferramentas para controlar determinada funcionalidade. Nesse caso, as funcionalidades que necessitam ser controladas são a comunicação através de *sockets* e o acesso ao banco de dados, explicado a seguir na seção Banco de Dados. A Figura 34 apresenta os principais métodos das *engines* criadas.

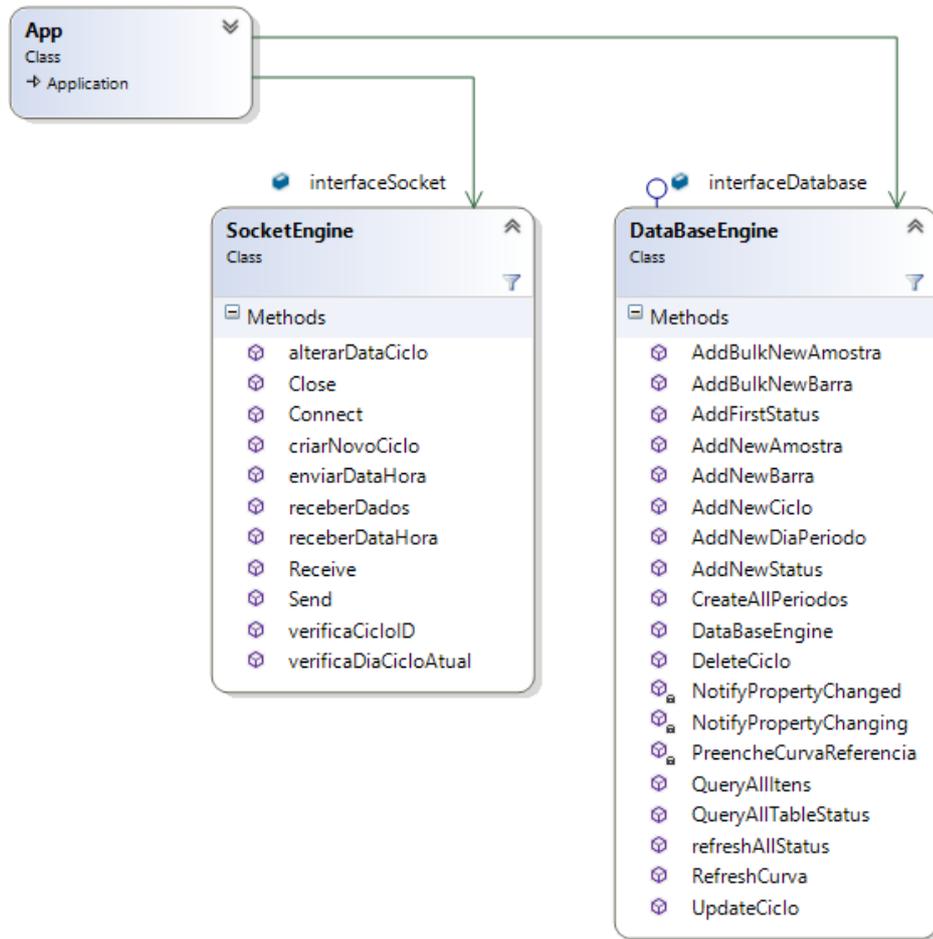


Figura 34 - Engines desenvolvidas para o aplicativo
 Fonte: autoria própria

3.7 BANCO DE DADOS

Esse projeto envolve grande manipulação de dados que são gerados a cada fração de minuto pelo dispositivo embarcado. Parte desses dados são armazenados momentaneamente. Porém, a função principal desse dispositivo embarcado é a coleta de informações, não o armazenamento contínuo deles. Para isso, é de vital importância que o aplicativo que se conecta com o módulo embarcado armazene o resultado de várias coletas de maneira não-volátil e segura, além de poder gerenciar com bom desempenho o acesso a um grande volume dessas informações para os usuários. Um conjunto de banco de dados atende a essa necessidade.

A plataforma Windows Phone disponibiliza nativamente uma opção de interface de banco de dados, chamada de *Database Language Integrated Query* (DLINQ) ou apenas *LINQ to SQL* (*Structured Query Language*). A interface DLINQ permite à aplicação não somente modelar estruturas de dados relacionais como também utilizar

operadores de consulta e expressões lambda para filtrar rapidamente informações dentro dos bancos de dados existentes. Finalmente, a grande vantagem dessa interface é unir a declaração da estrutura do banco de dados com a implementação dos dados do *Model-View-ViewModel* (MVVM), padrão arquitetural usado em engenharia de software que permite isolamento entre o processamento da lógica do programa e da interface.

O estudo do problema de armazenamento resultou na modelagem de cinco tabelas de dados. A primeira tabela está representada na Figura 35. Ela armazena o histórico das ações efetuadas entre o aplicativo e a placa embarcada, fornecendo diariamente um resumo da manipulação das informações e uma contagem total do que está armazenado no celular em questão.

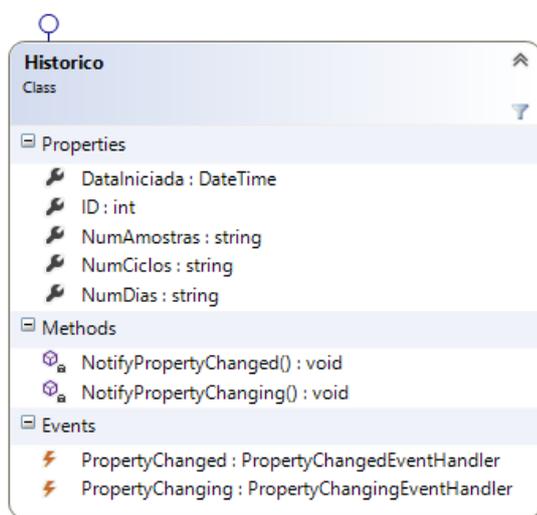


Figura 35 - Tabela de Histórico
Fonte: autoria própria

A segunda tabela criada está apresentada na Figura 36. Uma entidade dessa tabela representa o conjunto completo de informações referente a um ciclo de crescimento das aves. Diferente da anterior, a tabela de **Ciclo** possui associações com outra tabela. Um registro de **Ciclo** se associa a um conjunto de entidades **DiaPeriodos**, pertencentes a tabela de **DiaPeriodos**. Existe mais uma diferença na estrutura dessa tabela: a propriedade **versionDB**. Nesse caso é importante o uso de versionamento para evitar que em tantos acessos possa acontecer uma alteração de forma paralela, pois quatro outras entidades e suas propriedades se relacionam abaixo de cada entidade **Ciclo**. O LINQ se utiliza automaticamente dessa variável para gerir a atualização das informações no banco de dados.

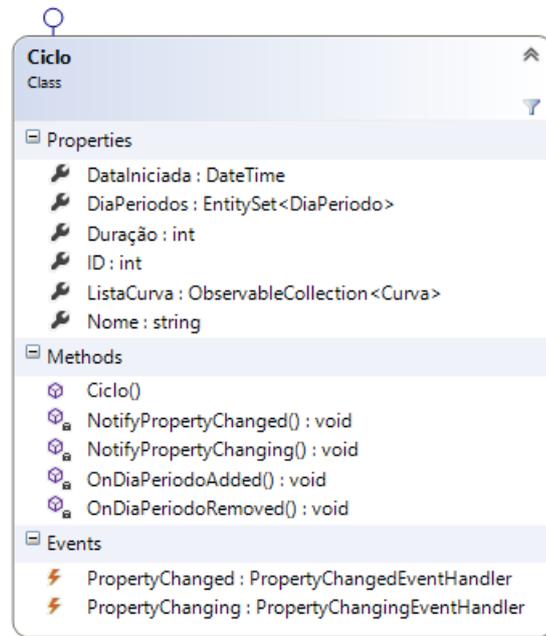


Figura 36 - Tabela de Ciclo
 Fonte: autoria própria

A tabela com as propriedades de **DiaPeriodos** está apresentada na Figura 37. Cada registro dessa entidade representa um dos dias que compõe o período de um ciclo de produção de frangos. Essa tabela possui associação com a entidade **Amostra** e **BarraHistograma**.

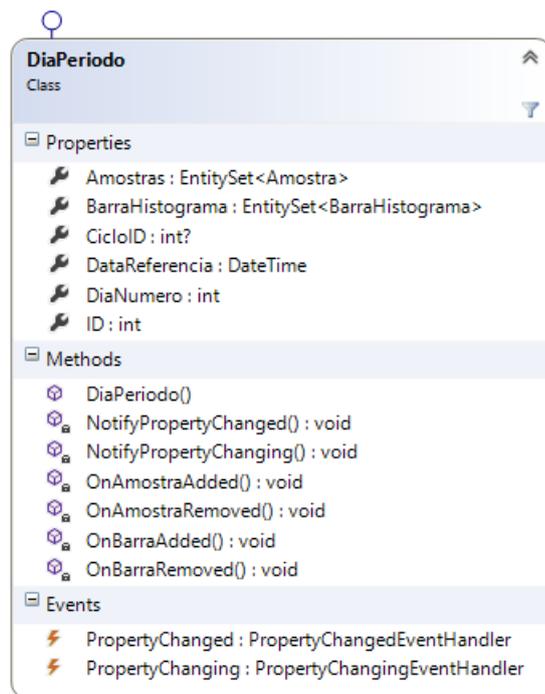


Figura 37 - Tabela de DiaPeriodo
 Fonte: autoria própria

A tabela de **Amostra** está apresentada na Figura 38. Uma entidade de **Amostra** representa o resumo dos dados coletados durante uma hora pelo módulo embarcado.

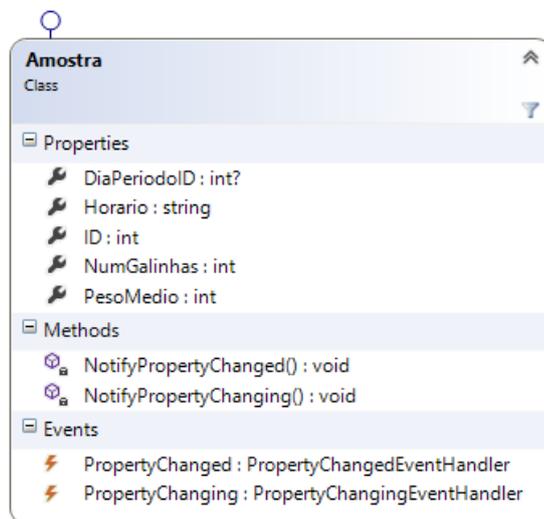


Figura 38 - Tabela de Amostra
Fonte: autoria própria

A última das tabelas modeladas é a tabela de **BarraHistograma**, apresentada na Figura 39. Ela armazena entidades que representam uma das barras do gráfico de histograma diário exibido pela aplicação.

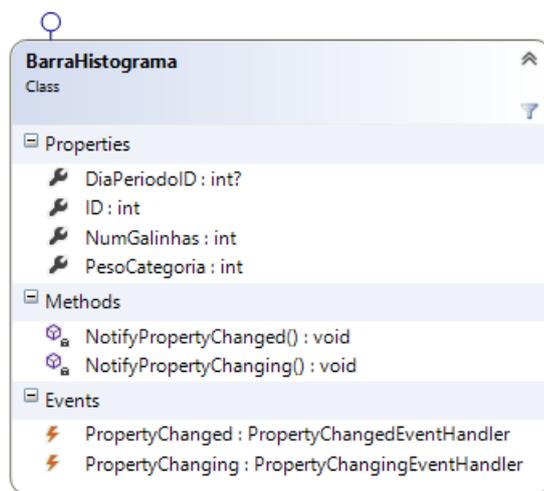


Figura 39 - Tabela de BarraHistograma
Fonte: autoria própria

3.8 INTERFACE DO APLICATIVO

O aplicativo criado reúne um conjunto de diferentes páginas que exibem as informações dos ciclos, apresentam resumos dos seus acontecimentos e permitem ao usuário configurar parâmetros do módulo embarcado. As Figuras 40 a 47 descrevem as páginas mais importantes do aplicativo.



Resumo (interface principal)

Esta é a página inicial do aplicativo. Ela apresenta um resumo indicando a quantidade de ciclos adicionados, o número de dias do ciclo que estão sincronizados com o aplicativo e as amostras obtidas. Além disso, ainda existe o botão de sincronizar, que serve para a sincronização dos dados da placa com o aplicativo e o botão de atualizar que reatualiza todas as contagens e dados.

Esta interface foi implementada no *pivot* principal, ou seja, ela pode ser deslocada para a esquerda, dando lugar à interface de ciclos.

Figura 40 - Modelo de interface: resumo
Fonte: autoria própria

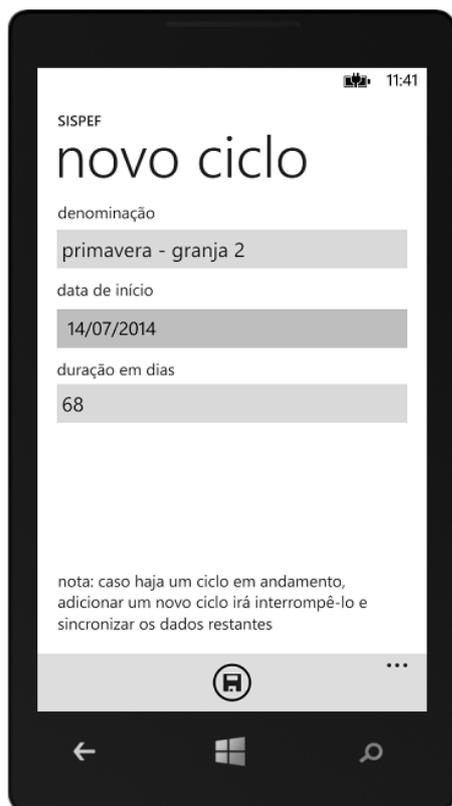


Relação de Ciclos (interface principal)

Esta página apresenta a relação dos ciclos adicionados no banco de dados do aplicativo. É possível ainda adicionar um ciclo a qualquer momento clicando no botão que se encontra na parte inferior da tela. A primeira linha de cada item indica seu nome e a segunda mostra a informação de duração do ciclo.

Ao clicar em um dos ciclos listados, o usuário é levado à interface com informações sobre o ciclo selecionado.

Figura 41 - Modelo de interface: relação de ciclos
Fonte: autoria própria



Novo Ciclo

Esta interface é utilizada para o cadastramento de um novo ciclo. Nela, encontram-se as opções para a criação de um nome para o ciclo, sua data de início e duração dele em dias. A duração máxima aceitável de um ciclo é de 71 dias.

No canto inferior da tela apresentada encontra-se um botão para salvar o ciclo criado. Ciclos muito longos podem levar alguns segundos para serem adicionados, desencadeando uma tela de espera.

Figura 42 - Modelo de interface: novo ciclo
Fonte: autoria própria



Histograma (interface de ciclo)

Esta página mostra o histograma do dia selecionado pelo usuário, ou seja, sua função é apresentar de forma gráfica a distribuição dos pesos das aves amostradas em um determinado dia. Essa página está sempre relacionada a um ciclo cadastrado no aplicativo. Ela indica ainda a data em que as amostras foram coletadas.

Esta interface foi implementada no *pivot* de ciclo. Assim, ela pode ser deslocada para a esquerda, dando lugar à página de dados ou pode ser deslocada para a direita, dando lugar à página de informações.

Figura 43 - Modelo de interface: histograma
Fonte: autoria própria

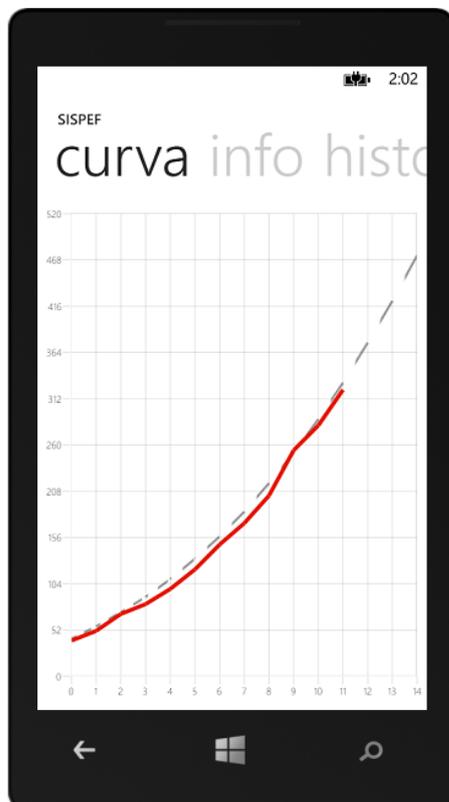


Amostra de Dados (interface de ciclo)

A página mostra os dados colhidos das pesagens de forma mais detalhada. Ela mostra ao usuário o resumo da quantidade de aves que foram pesadas em um determinado horário, o peso total amostrado e o peso médio de cada ave. Da mesma forma que na página da Figura 43, o usuário pode alterar o dia que deseja observar as informações.

Esta interface foi implementada no *pivot* de ciclo. Assim, ela pode ser deslocada para a esquerda, dando lugar à página de curvas ou pode ser deslocada para a direita, dando lugar à página de histograma.

Figura 44 - Modelo de interface: amostras de dados
Fonte: autoria própria

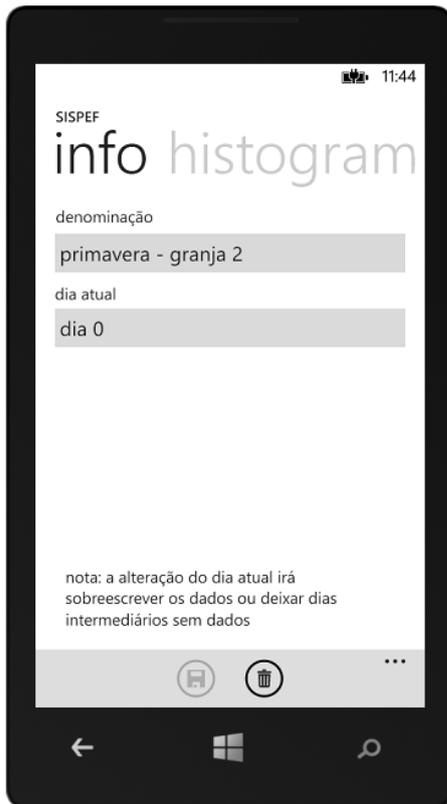


Relação entre Curvas (interface de ciclo)

Esta página sobrepõe duas curvas. O eixo horizontal representa dias do ciclo e o eixo vertical o médio das aves. A curva pontilhada representa a curva de referência, o peso ideal das aves ao longo dos dias do ciclo. A curva realçada representa a curva real, o quanto as aves realmente estão pesando de acordo com as medidas coletadas ao longo do ciclo.

Esta interface foi implementada no *pivot* de ciclo. Assim, ela pode ser deslocada para a esquerda, dando lugar à página de informações ou pode ser deslocada para a direita, dando lugar à página de dados.

Figura 45 - Modelo de interface: relação entre curvas
Fonte: autoria própria

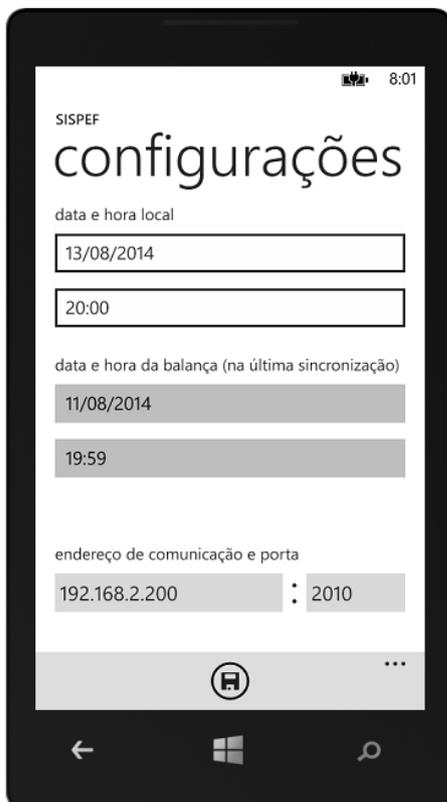


Informações (interface de ciclo)

Esta página apresenta e permite ao usuário alterar o nome do ciclo selecionado e o dia em que o ciclo se encontra. Além disso, ainda há a opção de exclusão do ciclo pelo botão localizado na parte inferior da tela.

Esta interface foi implementada no *pivot* de ciclo. Assim, ela pode ser deslocada para a esquerda, dando lugar à página de histograma ou pode ser deslocada para a direita, dando lugar à página de curvas.

Figura 46 - Modelo de interface: informações
Fonte: autoria própria



Configurações (interface auxiliar)

Esta página permite ao usuário visualizar e alterar a data e o horário do módulo embarcado, ao mesmo tempo que os contrasta com a data e horário do aplicativo. É possível, também, alterar o endereço de comunicação e porta que será utilizada. Na parte inferior da tela há a opção de salvar tais modificações.

Figura 47 - Modelo de interface: configurações
Fonte: autoria própria

4 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo estão apresentadas conclusões e algumas indicações de possíveis trabalhos futuros que a equipe considera relevante.

4.1 CONCLUSÕES

Avaliando o projeto como um todo, foi possível atingir todos os objetivos definidos no começo dos trabalhos. Contudo a precisão de 10 gramas pretendida pela equipe não pode ser alcançada para que o filtro tivesse uma variação menor do valor devido a ruídos. Isso se deve à histerese que foi definida mais alta para manter o sinal de entrada mais estável; caso esse valor seja reduzido, a precisão aumenta, porém isso causa uma variação maior no sinal, tornando a medida menos estável.

Mesmo com esse valor definido mais alto, mantendo a precisão em 13 gramas, percebe-se ainda uma variação de um degrau no ADC e isso levou a equipe a investigar o motivo. Assim, descobriu-se que o ADC da placa embarcada, que é de aproximação sucessiva, não é o mais indicado para utilização em balanças, uma vez que a célula de carga e a ponte utilizada no circuito analógico geram muitos ruídos. Descobriu-se então que o conversor mais utilizado em balanças comerciais é o sigma-delta e este poderia tornar o sistema mais estável que o ADC por aproximação sucessiva.

Com relação à comunicação dos dispositivos, os problemas iniciais que tornavam a conexão instável foram resolvidos com a troca do *Access Point* que causava a intermitência por tentar se conectar a redes externas e então desconectava os usuários da rede. O novo dispositivo de acesso utilizado não realiza essa operação de conexão com redes de internet externas e, portanto não causa a queda da conexão.

Fica claro que o protótipo desenvolvido não teria aplicação comercial direta como produto, mas com algumas funcionalidades a mais, ele poderia ser evoluído até um produto robusto, pronto para ser comercializado. Algumas melhorias, vistas pela equipe como as principais, para uma próxima fase de desenvolvimento estão apresentadas na seção de trabalhos futuros.

4.2 TRABALHOS FUTUROS

Com intuito de aproximar o projeto de um produto comercializável que realmente cumpra com a proposta de reduzir a necessidade de contato humano com os animais presentes na granja, a equipe considera que um passo importante seria montar um sistema com bateria para eliminar a necessidade de fiação elétrica dentro do criadouro.

Durante o projeto ficou claro que o conversor analógico-digital por aproximação sucessiva não era estável o suficiente para ser utilizado em um produto comercial, portanto um estudo recomendado seria para a substituição do conversor por um mais adequado para a montagem de balanças de forma a reduzir a instabilidade dos dados adquiridos e assim aumentar a precisão.

Ainda pensando no aprimoramento do projeto, seria interessante ampliar os softwares de forma que o usuário seja capaz de escolher a curva de crescimento dentre uma variedade de curvas diferentes, capacitando o projeto para a utilização em granjas com diversos tipos de frangos.

Outra ampliação de software, não tão trivial, porém de grande valor para o mercado é a capacidade de escolher o desvio padrão da curva de crescimento aceito pela granja. Dessa forma, propõe-se ampliar os softwares para incluir essa funcionalidade.

Por fim, o algoritmo utilizado para a determinação do número de aves poderia ser repensado de forma a tornar possível que mais de 5 aves estejam em cima da balança ao mesmo tempo, permitindo assim que o sistema fosse ampliado tanto física como comercialmente

REFERÊNCIAS

- AGROPECUÁRIA. **Diretórios de Projetos Tecnológicos da UNESP**. São Paulo, 2014. Disponível em:
<<http://www.unesp.br/prope/projtech/Agropecuaria/Agropec13a.htm>>. Acesso em: 20 ago. 2014.
- AMPRIMOZ, J.F. **The History of Windows Mobile**. 2011. Disponível em:
<<http://www.brighthub.com/computing/windows-platform/articles/1295.aspx>>. Acesso em: 07 ago. 2014.
- APPALOOSA. **Voltage stabiliser transistor**. 2012. 1 diagrama esquemático, 415 x 270 pixels. Disponível em
<http://commons.wikimedia.org/wiki/File:Voltage_stabiliser_transistor,_IEC_symbols.svg>. Acesso em: 10 ago. 2014.
- AVIAGEM. **Frango Ross 308: objetivos de desempenho**. Estados Unidos, Huntsville, 2012. Disponível em:
<http://pt.aviagen.com/assets/Tech_Center/BB_Foreign_Language_Docs/Portugues e/Ross308FrangosObjetivosDesempenho12PT.pdf>. Acesso em: 02 mai. 2014.
- AVIAGEM. **Manual de manejo de matrizes**. Estados Unidos, Huntsville, 2008. Disponível em:
<http://pt.aviagen.com/assets/Tech_Center/BB_Foreign_Language_Docs/Portugues e/Ross_Manual_Manejo_Matrizes.pdf>. Acesso em: 24 mai. 2014.
- BRUSAMARELLO, Valner. **Células de Carga**. Disponível em:
<<http://www.chasqueweb.ufrgs.br/~valner.brusamarello/eleinst/ufrgs7.pdf>>. Acesso em: 02 mai. 2014.
- CAMPILHO, Aurélio. **Instrumentação Electrónica. Métodos e técnicas de medição**. Porto: FEUP Edições, 2000.
- COBB-VANTRESS. **Manual de manejo de frangos de corte**. Estados Unidos, Siloam Springs, 1 abr. 2009. Disponível em:
<<http://wp.ufpel.edu.br/avicultura/files/2012/04/Cobb-Manual-Frango-Corte-BR.pdf>>. Acesso em: 02 mai. 2014.
- EMBEDDEDARTISTS. **LPC1769 LPCXPRESSO Board**. Estados Unidos, 2014. Disponível em:
<http://www.embeddedartists.com/products/lpcxpresso/lpc1769_xpr.php>. Acesso em: 20 ago. 2014.
- FALLE, Rainer. **The Metro Design Language**. 2012 Disponível em:
<<http://bsix12.com/metro-design-language/>>. Acesso em: 28 jul. 2014.

FRANÇA, Luciano C.; COELHO, Antonio A. D.; BEGIATO, Alexandre M.; SAVINO, Vicente J. M. **Determinação da Curva de Crescimento de Frangos de Corte da Linhagem AgRoss**. In: SIMPOSIO INTERNACIONAL DE INICIAÇÃO CIENTÍFICA, 16, 2008. Disponível em:

<<https://uspdigital.usp.br/siicusp/cdOnlineTrabalhoVisualizarResumo?numeroInscricaoTrabalho=3127&numeroEdicao=16>>. Acesso em: 02 mai. 2014.

INFOWESTER. **O que é Wi-Fi (IEEE 802.11)**. 2014. Disponível em <<http://www.infowester.com/wifi.php>>. Acesso em: 20 ago. 2014.

INTERFACE INC. **General Procedures for the use of Load Cells**. Scottsdale, 2013. Disponível em: <<http://www.interfaceforce.com/?General-Procedures-for-the-use-of-Load-Cells&mod=library&show=17>>. Acesso em: 10 ago. 2014.

KERNER Sean M. **What's The Top IDE?**. Jun. 2006. Disponível em <<http://www.internetnews.com/stats/article.php/3616626/Whats+The+Top+IDE.htm>>. Acesso em: 20 ago. 2014.

LAGES, Walter F. **Pilha TCP/IP**. Rio Grande do Sul, 2014. Disponível em: <<http://www.ece.ufrgs.br/~fetter/ele00002/tcpip.pdf>>. Acesso em: 20 ago. 2014.

MBED. **AnalogIn problem**. 2009. Disponível em: <<http://mbed.org/forum/mbed/topic/131/?page=1>>. Acesso em: 08 set. 2014.

MBED. **Spikes Again**. 2011. Disponível em: <<http://mbed.org/forum/mbed/topic/2919/>>. Acesso em: 08 set. 2014.

NXP SEMICONDUCTORS. **LPC1769/68/67/66/65/64/63 Product data sheet**. 24 jun. 2014. Rev. 9.5. Disponível em: <http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf>. Acesso em: 10 ago. 2014.

NXP SEMICONDUCTORS. **AN10974: LPC176x/175x 12-bit ADC design guidelines**. 1 set. 2010. Rev. 1. Disponível em: <http://www.nxp.com/documents/application_note/AN10974.pdf>. Acesso em: 10 ago. 2014.

OPPENHEIM, Alan V.; SCHAFFER, Ronald W.; BUCK, John R. **Discrete-time signal processing**. 2. ed. Upper Saddle River: Prentice Hall, 1999.

PITOMBEIRA, Kamila. Balança para aves acaba com pesagens manuais. **Portal Dia de Campo**, Rio de Janeiro, 20 abr. 2011. Disponível em: <<http://www.diadecampo.com.br/zpublisher/materias/Materia.asp?id=24124&secao=Pacotes+Tecnol%F3gicos>>. Acesso em: 02 mai. 2014.

POLONIO, Carlos A. Pesagem das Aves. **Avicultura Industrial**, Itu, 22 jul. 2009. Disponível em: <http://www.aviculturaindustrial.com.br/noticia/pesagem-das-aves/20090721161056_J_024>. Acesso em: 02 mai. 2014.

PORTAL CÉLULA DE CARGA. **Células de Carga**. 2004.
<<http://www.celuladecarga.com.br/celulas/escolhacel.htm>>. Acesso em: 02 mai. 2014.

PUC-RIO. **CSMA/CD**. Rio de Janeiro, 2014. Disponível em:
<http://www.inf.pucRio.br/~inf2056/inf2056_files/menu/material/transparencias/colcher/Ethernet.pdf>. Acesso em: 20 ago. 2014.

RIGAUD, François. **Test du convertisseur ADC de la plateforme MBED**. Meudon, 2011. Disponível em:
<http://mbed.org/media/uploads/Rigaud/rapport_test_adc_mbed_f_r_01-03-2011.pdf>. Acesso em 08 set. 2014.

RIZZOTTO, Patrycia M. **Exportação de carne de frango estará em alta em 2014**. Jan. 2014. Disponível em < <http://economia.ig.com.br/empresas/agronegocio/2014-01-21/exportacao-de-carne-de-frango-estara-em-alta-em-2014.html>>. Acesso em: 20 ago. 2014.

SLATTERY, Colm; NIE, Mariah. A reference design for high-performance, low-cost weigh scales. **Analog Dialog**, n. 39-12, dez. 2005. Disponível em:
<http://www.analog.com/library/analogdialogue/archives/39-12/weigh_scale.html>. Acesso em: 10 ago. 2014.

SOBRAL, Marcelo M. **MAC CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)**. Santa Catarina, 2011. Disponível em
<http://wiki.sj.ifsc.edu.br/wiki/index.php/RCO2-2011-2#MAC_CSMA.2FCA_.28Carrier_Sense_Multiple_Access.2FCollision_Avoidance.29>. Acesso em: 20 ago. 2014.

SOBRAL, Marcelo M. **WLAN: IEEE 802.11**. Santa Catarina, 2014. Disponível em
<<http://www.sj.ifsc.edu.br/~msobral/RCO2/slides/aula15.pdf>>. Acesso em: 20 ago. 2014.

SODALEC DISTRIBUTION. **Les fiches Sodis: Pesage**. França, Rennes, ago. 2008. Disponível em: <[http://www.sodis-france.fr/produits/40/fr/pesage-volaille\(2\).pdf](http://www.sodis-france.fr/produits/40/fr/pesage-volaille(2).pdf)>. Acesso em: 02 mai. 2014.

SULLIVAN, Craig O. **Microsoft announce Windows Phone 8.1**. 2014. Disponível em: <<http://geektech.ie/index.php/microsoft-announce-windows-phone-8-1-brings-cortana-voice-assistant-and-more/>>. Acesso em: 02 ago. 2014.

SYNERGIST. **New York Times Silverlight Kit Updated for Windows Phone 7 Series**. Mar. 2010. Disponível em
<<http://blogs.msdn.com/b/synergist/archive/2010/03/29/new-york-times-silverlight-kit-updated-for-windows-phone-7-series.aspx>>. Acesso em: 20 ago. 2014.

TEXAS INSTRUMENTS. **LM158/LM258/LM358/LM2904 Low Power Dual Operational Amplifiers**. 2013, 33p.

TOLEDO do Brasil. **Manual do usuário: 2096-A**. São Bernardo do Campo, 2014. Disponível em: <http://www.irmor.com.br/site2010/manuais_usuarios/2096%20A.pdf>. Acesso em: 02 mai. 2014.

TOLEDO DO BRASIL. **2096-A: Sistema de Pesagem Automática de Aves no Aviário**. São Bernardo do Campo, 2014. Disponível em: <<http://www.toledobrasil.com.br/balanca/balancas-para-pesar-animais/2096a>>. Acesso em: 02 mai. 2014.

UNIÃO BRASILEIRA DE AVICULTURA. **Protocolo de Boas Práticas de Produção de Frangos**. São Paulo: 2008. Disponível em: <http://www.avisite.com.br/legislacao/anexos/protocolo_de_boas_praticas_de_producao_de_frangos.pdf>. Acesso em: 02 mai. 2014.

VISUAL STUDIO. **Visual Studio – Microsoft Developer Tools**. Estados Unidos, 2014. Disponível em: <<http://www.internetnews.com/stats/article.php/3616626/Whats+The+Top+IDE.htm>>. Acesso em: 20 ago. 2014.

WELTECH. **Poultry weighing systems**. Inglaterra, Huntingdon, 2014. Disponível em: <<http://weltech.co.uk/poultry-weighing-systems.asp>>. Acesso em: 20 ago. 2014.

APÊNDICES

APÊNDICE A – Cronograma

A Figura 49 representa a distribuição das atividades durante o período de desenvolvimento do projeto do protótipo de um sistema computacional de pesagem autônoma de frangos. Abaixo estão explicadas cada uma das atividades.

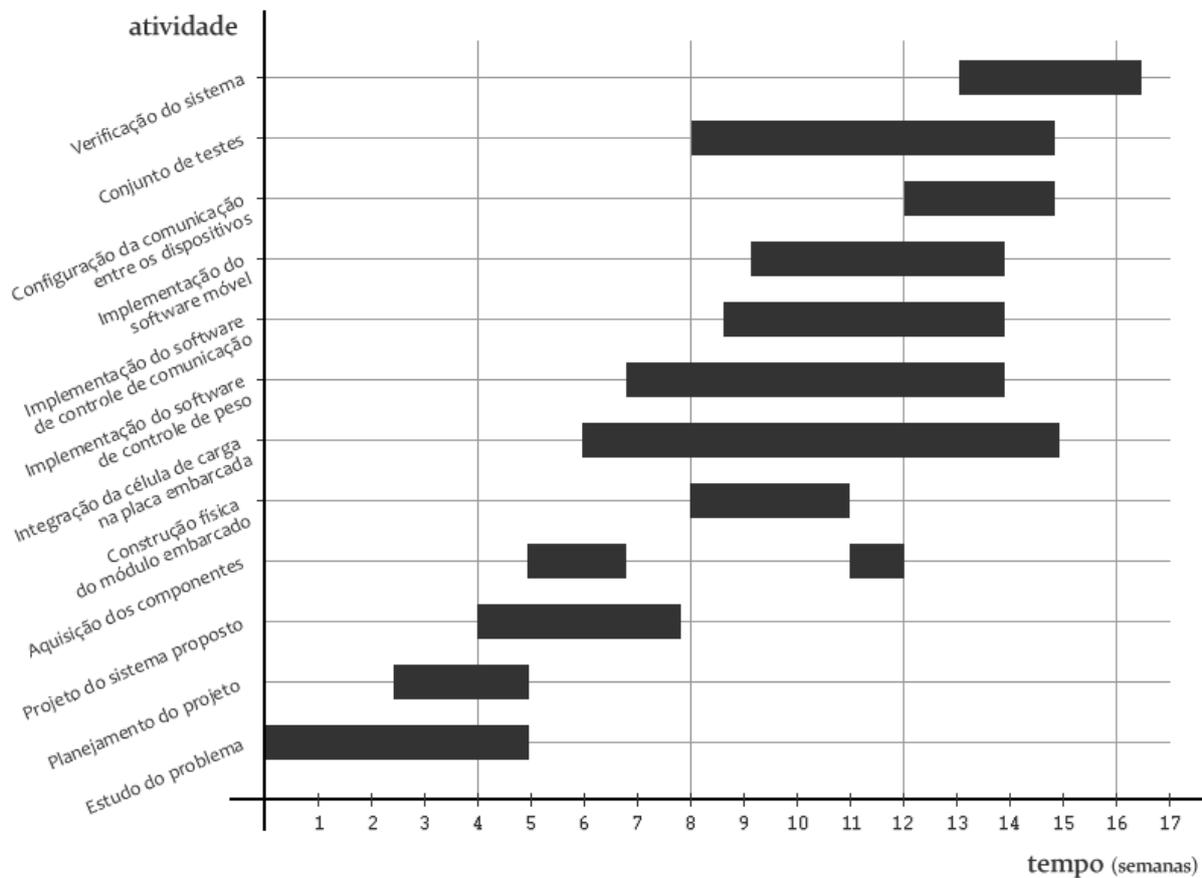


Figura 48 - Cronograma de desenvolvimento de projeto
 Fonte: autoria própria.

O estudo do problema consiste em fazer pesquisas sobre o problema para conseguir entendê-lo mais a fundo. Assim, teremos base para projetarmos o dispositivo necessário para sanar o problema da melhor forma possível. O estudo abrange tanto a temática dele (como é feita a pesagem da galinha atualmente) quanto algumas soluções já existentes para solucioná-lo.

O planejamento do projeto consiste em planejar o projeto como um todo: estimar quais serão as possíveis dificuldades encontradas durante o desenvolvimento dele, quanto de recurso financeiro utilizaremos para concluí-lo, definir um cronograma

para que o projeto esteja pronto na data desejável e fazer a escolha dos possíveis componentes que serão utilizados para a montagem dele.

A atividade de projeto do sistema proposto consiste em projetar o sistema proposto, ou seja, definir como será feito o sistema. O estudo do problema e o planejamento do projeto são importantes nesta atividade, pois com base neles, as ideias e planos serão colocados no papel de forma mais específica, a fim de definir como o sistema será desenvolvido e quais componentes serão utilizados.

A aquisição dos componentes consiste na busca dos componentes desejados para a montagem do sistema proposto. Essa atividade abrange desde a pesquisa pelos melhores preços e menores tempos para aquisição até termos os componentes em mãos para a implementação.

A construção física do módulo embarcado consiste na construção do módulo embarcado, que é composto por célula de carga, tampa metálica e placa embarcada. O foco dessa atividade é posicionar todos os componentes da maneira mais eficiente, fazendo com que todos eles funcionem perfeitamente interligados.

A atividade de integração da célula de carga na placa embarcada consiste na conexão da célula de carga à placa controladora, por meio de um circuito elétrico (que também tem a função de ajustar o valor obtido pela célula para melhor interpretação feita pela placa), fazendo com que os dados obtidos pela balança que contém a célula de carga sejam transmitidos para a placa embarcada.

A implementação do software de controle de peso consiste na concepção e implementação do software responsável por todo o controle de peso. A função principal do software será fazer a pesagem da maneira mais precisa possível das galinhas que se encontram em cima da balança, ou seja, ele precisará identificar o número de galinhas que estão em cima da balança, além de identificar a existência ou não de pesos consideráveis que não são os das galinhas.

A implementação do software de controle de comunicação consiste na concepção e implementação do software responsável pelo controle dos dados a serem comunicados. A função principal do software será controlar a comunicação entre a placa embarcada e a estação base, que é um *smartphone*.

A atividade de implementação do software móvel consiste na concepção e implementação do software que será utilizado no *smartphone* para a coleta de dados da balança. A atividade terá como foco a programação da interface gráfica, o

armazenamento local dos dados coletados da balança e a habilitação da comunicação do *smartphone* com a placa por meio de uma conexão sem fio.

Depois que o software móvel e de controle de comunicação estiverem implementados, a execução da atividade de configuração da comunicação entre os dispositivos é necessária. Ela consiste em realizar as configurações de rede para que os dispositivos consigam se comunicar de maneira eficiente.

A realização do conjunto de testes consiste em avaliar o funcionamento do software durante a fase final do desenvolvimento do projeto. Ou seja, realizar testes com cada uma das partes do sistema computacional para detectar falhas, que possivelmente serão corrigidas.

Finalmente, a verificação do sistema consiste em simular o funcionamento do sistema. Ou seja, verificar se o resultado obtido pelo sistema é o esperado, se ele está funcionando de forma eficiente e como ele se comporta. Essa atividade será feita por meio de ensaios com diferentes cenários.

APÊNDICE B – Riscos

1º Etapa: Identificação do Risco	
Denominação do Risco: atraso na aquisição de componentes	No da Identificação: 01
Descrição do Risco: caso os componentes críticos (como placa embarcada, célula de carga e <i>access point</i>) para o desenvolvimento do projeto não estejam disponíveis para compra ou demorem para serem entregues.	

2º Etapa: Avaliação do Risco					
Impacto:	Alto: (x)	Médio/Alto: ()	Médio: ()	Médio/Baixo: ()	Baixo: ()
Explique: o desenvolvimento e o resultado do projeto são dependentes da aquisição dos seus componentes.					
Probabilidade:	Alto: ()	Médio/Alto: ()	Médio: (x)	Médio/Baixo: ()	Baixo: ()
Explique: os componentes possuem um grande número de fornecedores que estão próximos ou já foram adquiridos pelos estudantes antes do início do projeto.					

3º Etapa: Resposta ao Risco	
Estratégias e Ações: relação alternativa de componentes similares aos escolhidos.	
Impacto Reavaliado: Alto	Probabilidade Reavaliada: Média/Baixa
Elaborado por: equipe completa	Data: 01/05/2014

1º Etapa: Identificação do Risco	
Denominação do Risco: calibração da célula de carga	No da Identificação: 02
Descrição do Risco: dificuldades que comprometam a montagem ou a obtenção com precisão da resposta do componente de carga da balança do sistema.	

2º Etapa: Avaliação do Risco					
Impacto:	Alto: ()	Médio/Alto: (x)	Médio: ()	Médio/Baixo: ()	Baixo: ()
Explique: a montagem realizada de uma maneira errada ou sua má calibração pode gerar resultados imprecisos e sem valor estatístico.					
Probabilidade:	Alto: ()	Médio/Alto: ()	Médio: (x)	Médio/Baixo: ()	Baixo: ()
Explique: a equipe domina os fundamentos básicos para a integração da célula em circuitos e para captura da resposta desses circuitos pelo microprocessador.					

3º Etapa: Resposta ao Risco	
Estratégias e Ações: análise mais profunda na literatura de apoio e consulta a profissionais da área, como prevista na atividade de integração da célula de carga na placa embarcada.	
Impacto Reavaliado: Médio/Alto	Probabilidade Reavaliada: Baixa
Elaborado por: equipe completa	Data: 01/05/2014

1º Etapa: Identificação do Risco	
Denominação do Risco: alcance limitado de comunicação	No da Identificação: 03
Descrição do Risco: caso a comunicação entre os dispositivos propostos no projeto seja limitada a uma distância pouco viável ou tenha problemas de interferência que impossibilitem o estabelecimento de um canal de comunicação com qualidade.	

2º Etapa: Avaliação do Risco					
Impacto:	Alto: ()	Médio/Alto: (x)	Médio: ()	Médio/Baixo: ()	Baixo: ()
Explique: complicações na comunicação podem dificultar o uso e apresentação do funcionamento da solução.					
Probabilidade:	Alto: ()	Médio/Alto: ()	Médio: ()	Médio/Baixo: (x)	Baixo: ()
Explique: os padrões a serem utilizados são robustos o suficiente para evitarem interferências comuns e garantir alcance razoável.					

3º Etapa: Resposta ao Risco	
Estratégias e Ações: componentes adicionais de rede podem ser adicionados ao sistema na proporção do problema de comunicação entre as partes do projeto.	
Impacto Reavaliado: Médio/Alto	Probabilidade Reavaliada: Baixa
Elaborado por: equipe completa	Data: 01/05/2014

1º Etapa: Identificação do Risco	
Denominação do Risco: queima de componente	No da Identificação: 04
Descrição do Risco: caso um dos componentes crítico (como placa embarcada, célula de carga e <i>access point</i>) no projeto pare de funcionar, seja por mal utilização ou eventuais defeitos presentes nele.	

2º Etapa: Avaliação do Risco					
Impacto:	Alto: ()	Médio/Alto: (x)	Médio: ()	Médio/Baixo: ()	Baixo: ()
Explique: a perda de um dos componentes causa interrupção em atividades previstas no cronograma do projeto, prolongando o tempo para a elaboração do sistema inteiro.					
Probabilidade:	Alto: ()	Médio/Alto: ()	Médio: ()	Médio/Baixo: ()	Baixo: (x)
Explique: os componentes utilizados possuem robustez contra erros básicos e garantia de qualidade na fabricação. Além disso, os integrantes da equipe foram instruídos a tomarem devidos cuidados no manuseio deles.					

3º Etapa: Resposta ao Risco	
Estratégias e Ações: na etapa de planejamento, componentes sobressalentes foram definidos caso haja necessidade de substituir os principais. Alguns componentes já foram adquiridos com mais de uma unidade para rápida resposta a esse risco.	
Impacto Reavaliado: Médio	Probabilidade Reavaliada: Baixa
Elaborado por: equipe completa	Data: 16/05/2014

1º Etapa: Identificação do Risco	
Denominação do Risco: ruídos e interferências nos sinais	No da Identificação: 05
Descrição do Risco: caso os índices de interferência externa comprometam os sinais de entrada na placa controladora, não permitindo a interpretação correta das informações recebidas.	

2º Etapa: Avaliação do Risco					
Impacto:	Alto: ()	Médio/Alto: ()	Médio: (x)	Médio/Baixo: ()	Baixo: ()
Explique: esse risco não afeta diretamente o andamento e a conclusão do projeto, porém causa imprecisões e atrasos na transmissão dos dados.					
Probabilidade:	Alto: ()	Médio/Alto: ()	Médio: ()	Médio/Baixo: (x)	Baixo: ()
Explique: o cabeamento escolhido possui proteção física e redundância. As soldas serão realizadas com a máxima precisão possível.					

3º Etapa: Resposta ao Risco	
Estratégias e Ações: a utilização de amplificadores operacionais ou de instrumentação é suficiente para eliminar consideravelmente os ruídos e interferências externas.	
Impacto Reavaliado: Médio	Probabilidade Reavaliada: Baixa
Elaborado por: equipe completa	Data: 16/05/2014