



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

**DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

DISSERTAÇÃO DE MESTRADO

DAVI MIARA KIAPUCHINSKI

**Uma plataforma para monitoramento de
espectros sonoros e pré-processamento de
canto de pássaros**

DISSERTAÇÃO DE MESTRADO

CURITIBA

2012

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

DISSERTAÇÃO

apresentada a UTFPR
para obtenção do grau de

MESTRE EM COMPUTAÇÃO APLICADA

Davi Miara Kiapuchinski

**Uma plataforma para monitoramento de
espectros sonoros e pré-processamento de
canto de pássaros**

Orientadores:

Prof. Dr. Carlos Raimundo Erig Lima

Doutor em Engenharia Mecânica pela Universidade Estadual de Campinas – Brasil (2003)

Prof. Ph.D. Celso A. AlvesKaestner

Pós-Doutor em Engenharia Elétrica pela École de Technologie Supérieure (1999)

Banca examinadora:

Prof. Dr. Carlos Raimundo Erig Lima - UTFPR

Prof. Ph.D. Alessandro L. Koerich – PUCPR

Prof. Dr. Jean M. Simão - UTFPR

Coordenador do Mestrado:

Prof. Dr. João Alberto Fabro

Curitiba, dezembro de 2012

DAVI MIARA KIAPUCHINSKI

**UMA PLATAFORMA PARA MONITORAMENTO DE ESPECTROS SONOROS E
PRÉ-PROCESSAMENTO DE CANTO DE PÁSSAROS**

DISSERTAÇÃO APRESENTADA AO
PROGRAMA DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA (PPGCA) DA
UNIVERSIDADE TECNOLÓGICA FEDERAL
DO PARANÁ (UTFPR), COMO PARTE DOS
REQUISITOS PARA A OBTENÇÃO DO
TÍTULO DE MESTRE EM COMPUTAÇÃO
APLICADA. ÁREA DE CONCENTRAÇÃO:
SISTEMAS EMBARCADOS

Orientador: Prof. Dr. Carlos R. Erig Lima

Co-Orientador: Prof. Ph.D. Celso A. A. Kaestner

CURITIBA
2012

Dados Internacionais de Catalogação na Publicação

K46 Kiapuchinski, Davi Miara

Uma plataforma para monitoramento de espectros sonoros e pré-processamento de canto de pássaros / Davi Miara Kiapuchinski. — 2012.

107f. : il. ; 30 cm

Orientador: Carlos Raimundo Erig Lima, Celso A. Alves Kaestner.

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Computação Aplicada. Área de concentração: Sistemas embarcados, Curitiba, 2012.

Bibliografia: f. 92-97.

1. Ornitologia. 2. Eletrônica. 3. Sons - Processamento. 4. Computação – Dissertações. I. Lima, Carlos Raimundo Erig, orient. II. Kaestner, Celso A. A., co-orient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Comunicação Aplicada. III. Título.

CDD (22. ed.) 004

Biblioteca Central da UTFPR, Campus Curitiba

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, por fornecer os elementos necessários que permitiram minha chegada até aqui. Agradecer também às “minhas famílias” pelo total apoio durante o período do mestrado, aos meus amigos e aos companheiros de mestrado.

Grato ao meu orientador, Prof. Carlos Raimundo Erig Lima, pelos créditos e confiança em mim depositados desde o início da nossa convivência. Ao meu co-orientador, Professor Celso A. Alves Kaestner, pelas inúmeras dicas, por acreditar na temática e por toda ajuda essencial ao trabalho.

Agradeço também toda a experiência passada pelo Professor Jean M. Simão, principalmente por sua paciência. Ao Professor Robson Ribeiro Linhares, que com seu conhecimento e experiência, deixou para mim e para o trabalho uma grande contribuição técnica. Grato ao Professor Alessandro L. Koerich, participante do projeto BrBird, pela contribuição científica na área e pela participação na banca. Aos demais Professores do programa de mestrado que se esforçam em fazer um bom trabalho, em especial ao Professor João Alberto Fabro que se mostrou um ótimo coordenador do programa.

Meus reconhecimentos ao pessoal da Marinha do Brasil que contribuíram para este trabalho fornecendo-me meios e períodos diários de trabalho. Em especial ao Capitão-de-Corveta (T) Rogers do Centro de Tecnologia da Informação da Marinha pelo total apoio e compreensão. Agradecimentos também ao Capitão-de-Fragata (EN) Auro, ao Engenheiro Oswaldo e o Engenheiro Marcelo do Centro de Manutenção de Sistemas que disponibilizaram um laboratório de eletrônica para ajudar nos testes.

Agradeço por fim, a Fundação Araucária e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) pelo apoio financeiro.

Sinceramente,

Davi Miara Kiapuchinski

RESUMO

Os atuais métodos de classificação automática dos pássaros pelo canto apresentam lacunas que justificam um aprofundamento do estudo do tema. Entre estas lacunas, por exemplo, observa-se a necessidade da evolução da pesquisa para uma maior abrangência de espécies. Outra lacuna, que será o foco deste trabalho, é observada nas etapas iniciais dos processos de classificação automática em ambientes reais e práticos (e.g. *habitat* natural).

Considerando-se que as etapas de coleta de amostras, pré-processamento, processamento e análise dos resultados estão presentes em um processo de classificação sonora de pássaros, observa-se uma carência de contribuições na etapa de coleta de amostras e, particularmente, na etapa de pré-processamento.

Neste âmbito, as rotinas projetadas nos trabalhos científicos de classificação sonora são muitas vezes dependentes de uma amostra ideal, que por sua vez não são encontradas em um ambiente real. Ou seja, para obter respostas corretas e acertos na classificação, tais rotinas necessitam de amostras filtradas e também de um escopo do ambiente previamente restrito.

Com o objetivo de prover soluções para este conjunto de deficiências, foi proposta e materializada uma plataforma de coleta e processamento dos sons coletados, cobrindo as duas primeiras etapas de classificação automática.

Em suma, este trabalho apresenta uma plataforma de aquisição e pré-processamento de cantos de pássaros. No trabalho também são discutidas vantagens e desvantagens das metodologias utilizadas, bem como são dissertados contextos pertinentes para aplicação da plataforma. As validações se dão por meio de testes práticos e comparações das amostras processadas pela plataforma, com amostras resultantes das ferramentas de processamento e análise sonora previamente existentes.

Palavras-chave: Pré-processamento sonoro, eletrônica embarcada, ornitologia.

ABSTRACT

The current methods for birdsong automatic classification present gaps which justify a deeper study of the theme. Among these shortcomings, for example, there is the need for the development of research for a wider range of species. Another gap that will be the focus of this work, is observed in the initial stages of the process of automatic classification in real environments and practical (e. g. nature).

Considering that the stages of sample collection, preprocessing, processing and analysis of results are present in a birdsong classification process, there is a deficiency of contributions in the stage of sample collection and, particularly, in the preprocessing step.

In this context, the routines designed on the scientific papers of sound classification are often dependent on an ideal sample, which are not found in a real environment. That is, to achieve correct answers and successes in classification such routines require filtered samples and also a scope of environment previously restricted.

With the objective of providing solutions to this set of deficiencies, it was proposed and implemented a platform for collecting and processing of sounds collected, covering the first two steps of automatic classification.

In summary, this paper presents a platform for acquisition and preprocessing of bird songs. On paper are also discussed about the advantages and disadvantages of the methodologies used, as well as the contexts relevant to the application of the platform. The validation occurs through practical tests and comparisons between the samples processed by the platform, with samples arising from the previously existing tools of sound processing and analysis.

Key Words: Sound pre-processing, embedded electronics, ornithology.

LISTA DE ILUSTRAÇÕES

Figura 1: Gravadores de sons (GERI e Cornell Lab).....	13
Figura 2: Estação para assustar pássaros(Sterela, França)	16
Figura 3: Estação para captura de sons (<i>CornellLabofOrnitology</i>).....	23
Figura 4: Estação para estudo dos pássaros (RememBird.com)	23
Figura 5: Função degrau unitário (Souza, 2010)	26
Figura 6: Gráficos de complexidade (Autoria própria)	27
Figura 7: Simetria de uma DTF (Brigham e Morrow, 1967).....	28
Figura 8: Algoritmo de Cooley-Tukey base dois (Cooley e Tukey, 1965).....	28
Figura 9: Funções de janelamento (Adaptado de Champagne e Labeau, 2004)	29
Figura 10: Sinusoide sintetizada com FFT (Somervuo, Härmä e Fagerlund, 2006)	30
Figura 11: Filtros FIR (Adaptado de Ess, 2011)	31
Figura 12: Filtros IIR (Adaptado de Ess, 2011)	31
Figura 13: Distribuição normal (Hotta, 2009).....	36
Figura 14: Gráfico de precisão (Vilches et. al., 2006).....	36
Figura 15: Processo de sistemas(Adaptado de Bahill e Gissing, 1998)	40
Figura 16: Formato inicial do sistema (Autoria própria)	42
Figura 17: Diagrama de requisitos (Autoria própria).....	46
Figura 18: Diagrama de casos de uso (Autoria própria).....	48
Figura 19: Diagrama de blocos (Autoria própria).....	49
Figura 20: Diagrama de Classes (Autoria Própria).....	50
Figura 21: Diagrama paramétrico (Autoria própria)	51
Figura 22: Valores das propriedades paramétricas (Autoria própria)	52
Figura 23: Efeito dos filtros no som (Autoria própria)	57
Figura 24: Plataforma de testes (friendlyarm.net)	61
Figura 25: Esquemático e layout do circuito de alimentação da UAA (Autoria própria)62	
Figura 26: Comparações gráficas entre espectros (Agranat, 2009)	63
Figura 27: Comparação entre funções de janelamento (Autoria própria)	65
Figura 28: Remoção ruídos mostrada no domínio do frequência (Autoria própria)	70
Figura 29: Remoção ruídos mostrada no domínio do tempo (Autoria própria)	71
Figura 30: Canto de ave antes e depois da aplicação do filtro (Autoria própria)	71
Figura 31: Espectros antes e depois da aplicação do filtro (Autoria própria).....	71

Figura 32: UAA (Autoria própria)	76
Figura 33: Canto pássaro filtrado (Autoria própria)	77
Figura 34: Resultado do espectro sonoro filtrado (Autoria própria)	77
Figura 35: Comparação entre espectrogramas (Autoria própria).....	78
Figura 36: Tempos de processamento (Kiapuchinski, Lima e Kaestner, 2012)	82
Figura 37: Diferença entre duas remoções de ruídos (Autoria própria)	82
Figura 38: Comparação dos espectros sonoros (Autoria própria)	83

LISTA DE TABELAS

Tabela 1: Rotinas de pré-processamento.....	34
Tabela 2: Estado da arte e da técnica da pesquisa.....	38
Tabela 3: Enquadramento da pesquisa.....	38
Tabela 4: Requisitos de sistema	43
Tabela 5: Requisitos de <i>hardware</i>	44
Tabela 6: Requisitos de <i>software</i>	45
Tabela 7: Casos de uso identificados.....	47
Tabela 8: Características do microcontrolador	54
Tabela 9: Funções importantes <i>software</i>	55
Tabela 10: Plano de testes.....	62
Tabela 11: Teste 1 – Coleta e gravação.....	64
Tabela 12: Teste 2 – Montagem arquivo WAV	64
Tabela 13: Teste 3 - Função de janelamento	66
Tabela 14: Teste 4 - Espectrograma	66
Tabela 15: Médias de tempo de execução na UAA em segundos	67
Tabela 16: Teste 5 – Extração características	68
Tabela 17: Teste 6 – Remoção de ruídos	69
Tabela 18: Médias de tempo de execução dos filtros em segundos	72
Tabela 19: Exemplo de características gerais do som coletado.....	72
Tabela 20: Exemplo de características no domínio do tempo	73
Tabela 21: Exemplo de características no domínio da frequência	73
Tabela 22: F-Measure nos conjuntos de chamadas e pulsos dos pássaros	79
Tabela 23: Medidas de classificação com áudio completo.....	80
Tabela 24: Comparação das classificações utilizando sons filtrados pela UAA	80
Tabela 25: Acertos áudio completo X Acertos pulsos correspondentes.....	81
Tabela 26: Custos do projeto	89
Tabela 27: Especificação do caso de uso ‘Gerir Variáveis’	97
Tabela 28: Especificação do caso de uso ‘Processar Som’	98
Tabela 29: Especificação do caso de uso ‘Preparar Som’	99
Tabela 30: Especificação do caso de uso ‘Formatar Saída’	99
Tabela 31: Especificação do caso de uso ‘Controlar Comunicação’	100

Tabela 32: Especificação do caso de uso 'Filtrar Som'	100
Tabela 33: Especificação do caso de uso 'Extrair Características'	101
Tabela 34: Descrição dos tipos dos atores	102
Tabela 35: Soma desajustada dos atores.....	103
Tabela 36: Tabela complexidade casos de uso	103
Tabela 37: Soma desajustada dos casos de uso	104
Tabela 38: Fatores técnicos.....	105
Tabela 39: Fatores de ambiente	106

LISTA DE ABREVIATURAS E SIGLAS

SIGLA	SIGNIFICADO	TRADUÇÃO PARA PORTUGUÊS
ARM	<i>Acorn RISC Machine</i>	<i>Máquina RISC Acorn</i>
ASA	Área de Segurança Aeroportuária	-
BPS	<i>Bytes per Sample</i>	<i>Bytes por Amostra</i>
CENIPA	Centro de Investigação e Prevenção de Acidentes Aeronáuticos	-
CEO	Centro de Estudo Ornitológicos	-
CODEC	Codificador/Decodificador	
dB	Decibel	-
DFT	<i>Discrete Fourier Transform</i>	Transformada Discreta de Fourier
DMA	<i>DirectMemory Access</i>	Acesso Direto a Memória
DSP	<i>Digital Signal Processor</i>	Processador de Sinais Digitais
DOA	<i>DirectionofArrival</i>	Direção de Chegada
E/S	Entrada/Saída	-
FFT	<i>Fast Fourier Transform</i>	Transformada Rápida de Fourier
FIR	<i>Finite Impulse Response</i>	Resposta Finita ao Impulso
FPGA	<i>Field-ProgrammableGate Array</i>	Arranjo de Portas Programável em Campo
FTP	<i>File TransferProtocol</i>	Protocolo de Transferência de Arquivo
GPS	<i>Global Positioning System</i>	Sistema de Posicionamento Global
HMM	<i>Hidden Markov Model</i>	Modelo de <i>Markov</i> Oculto
I2S	<i>Integrated InterchipSoun</i>	Som <i>Interchip</i> Integrado
IIR	<i>Infinite Impulse Response</i>	Resposta ao Impulso Infinita
IP	<i>Internet Protocol</i>	Protocolo de Internet
JTAG	<i>Joint Test ActionGroup</i>	Grupo de Ação Conjunta de Teste
KNN	<i>K NearestNeighbor</i>	K Vizinhos Mais Próximos
MFCC	<i>Mel-FrequencyCepstralcoefficients</i>	Coeficientes Cepstrais da Frequência de Mel
MISO	<i>MasterInput, SlaveOutput</i>	Entrada Mestre, Saída Escrava

MMU	<i>Memory Management Unit</i>	Unidade de Gerenciamento de Memória
MPU	<i>Memory Protection Unit</i>	Unidade de Proteção à Memória
MOSI	<i>Master Output, Slave Input</i>	Saída Mestre, Entrada Escrava
MSB	<i>Most Significant Bit</i>	Bit Mais Significativo
NOR	<i>Not Or</i>	Não Ou
NAND	<i>Not And</i>	Não E
ONG	Organizações Não Governamentais	-
PC	<i>Personal Computer</i>	Computador Pessoal
PDS	Processamento Digital de Sinais	-
PWM	<i>Pulse Width Modulation</i>	Modulação por Largura de Pulso
RAM	<i>Random Access Memory</i>	Memória de Acesso Aleatória
RISC	<i>Reduced instruction set computer</i>	Computador com um Conjunto Reduzido de Instruções
ROM	<i>Read Only Memory</i>	Memória Somente de Leitura
RTC	<i>Real Time Clock</i>	Relógio de Tempo Real
RTOS	<i>Real Time Operational System</i>	Sistema Operacional de Tempo Real
SOB	Sociedade Brasileira de Ornitologia	-
SO	Sistema Operacional	-
SOC	<i>System on Chip</i>	Sistema em Chip
SPI	<i>Serial Peripheral Interface</i>	Interface de Periféricos Serial
SPS	<i>Samples Per Second</i>	Amostras Por Segundo
SVM	<i>Support Vector Machines</i>	Máquinas de Vetor Suporte
SysML	<i>Systems Modeling Language</i>	Linguagem de Modelo de Sistemas
UAA	Unidade de Aquisição Autônoma	-
UCP	<i>Use Case Points</i>	Pontos por Caso de Uso
UML	<i>Unified Modeling Language</i>	Linguagem Modelagem Unificada
USB	<i>Universal Serial Bus</i>	Barramento Serial Universal
V	<i>Volt</i>	-

SUMÁRIO

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	IV
LISTA DE ILUSTRAÇÕES	V
LISTA DE TABELAS	VII
LISTA DE ABREVIATURAS E SIGLAS	IX
SUMÁRIO	XI
CAPÍTULO 1	13
1.1 PROBLEMA	14
1.2 CONTEXTUALIZAÇÃO	15
1.3 OBJETIVOS	17
1.3.1 Objetivo geral.....	17
1.3.2 Objetivos específicos	17
1.4 JUSTIFICATIVA	18
1.5 ESTRUTURA DO TRABALHO.....	19
CAPÍTULO 2	21
2.1 ORNITOLOGIA	21
2.2 PROCESSADOR DIGITAL DE SINAIS.....	24
2.3 PROCESSAMENTO DIGITAL DE SONS	24
2.3.1 Transformada Discreta de <i>Fourier</i>	25
2.3.2 Transformada Rápida de <i>Fourier</i>	27
2.3.3 Funções de janelamento	29
2.3.4 Filtros Digitais	30
2.4 PRÉ-PROCESSAMENTO DE AMOSTRAS SONORAS	32
2.5 MODELOS ESTATÍSTICOS E PROBABILÍSTICOS	35
2.6 CONCLUSÃO E CONSIDERAÇÕES.....	37
CAPÍTULO 3	40
3.1 INTRODUÇÃO A ENGENHARIA DE SISTEMAS	41
3.2 ENGENHARIA DE REQUISITOS.....	42
3.2.1 Requisitos de <i>hardware</i>	43
3.2.2 Requisitos de <i>software</i>	44
3.2.3 Diagrama de requisitos	45
3.2.4 Diagrama de Casos de Uso	46
3.3 PROJETO DO SISTEMA	49
3.4 INTEGRAÇÃO.....	52
3.4.1 Controlador	52
3.5 DESENVOLVIMENTO	54
3.5.1 Rotinas Importantes	55
3.5.2 Filtros	56
3.5.3 Templates de Áudio	57
3.6 CONCLUSÃO E CONSIDERAÇÕES.....	57

CAPÍTULO 4	59
4.1 CONFIGURAÇÃO DO SISTEMA	59
4.2 UNIDADE DE AQUISIÇÃO AUTÔNOMA	60
4.2.1 <i>Hardware</i>	60
4.2.2 <i>Software</i>	62
4.3 CONCLUSÃO E CONSIDERAÇÕES	74
CAPÍTULO 5	75
5.1 CONSIDERAÇÕES SOBRE OS ESTUDOS PRÁTICOS	75
5.1.1 Considerações sobre os filtros da UAA – Eficiência.....	77
5.1.2 Considerações sobre os filtros da UAA – Classificação de pássaros....	78
5.1.3 Considerações sobre os filtros da UAA – Tempos de processamento..	81
5.1.4 Considerações sobre os filtros da UAA – Comparações gráficas	83
5.2 ESTUDOS COMPARATIVOS.....	84
5.3 REFLEXÕES	85
5.4 CONCLUSÃO E CONSIDERAÇÕES	86
CAPÍTULO 6	87
6.1 TRABALHOS FUTUROS.....	88
6.1.1 Estudo da aplicabilidade para comercialização	89
REFERÊNCIAS BIBLIOGRÁFICAS	91
APÊNDICE A – DESCRIÇÃO CASOS DE USO	97
APÊNDICE B – PONTOS POR CASOS DE USO	102

INTRODUÇÃO

Não é recente a preocupação de biólogos e ambientalistas no que diz respeito à vida e o bem estar dos animais. Prova disso é o grande trabalho feito por comunidades do mundo todo que se colocam como defensoras da fauna em projetos, campanhas e ações de proteção e preservação animal.

Como exemplos de organizações direcionadas ao assunto destacam-se as Organizações Não Governamentais (ONG`s) como a *World WildlifeFund* (WWF - Fundo de Vida Animal Mundial) que promove a conservação da biodiversidade (Souza, Martins, 2009) e o Centro de Estudos Ornitológicos (CEO) que difunde e mobiliza estudos sobre os pássaros (Visconti e Figueiredo, 2007).

Além da mobilização de pessoas e recursos, realizada pelas organizações citadas objetivando a preservação da vida animal, a comunidade científica mobiliza-se juntamente com as áreas da engenharia e da computação, para contribuições no desenvolvimento de técnicas que auxiliem em estratégias para os mesmos fins. Alguns exemplos são mostrados na Figura 1 que ilustra um gravador de sons do fundo do mar da empresa *GeospaceEngineeringResourcesInternational*(GERI) e um coletor de sons da fauna marinha do *CornellLabofOrnitology*, nesta ordem.

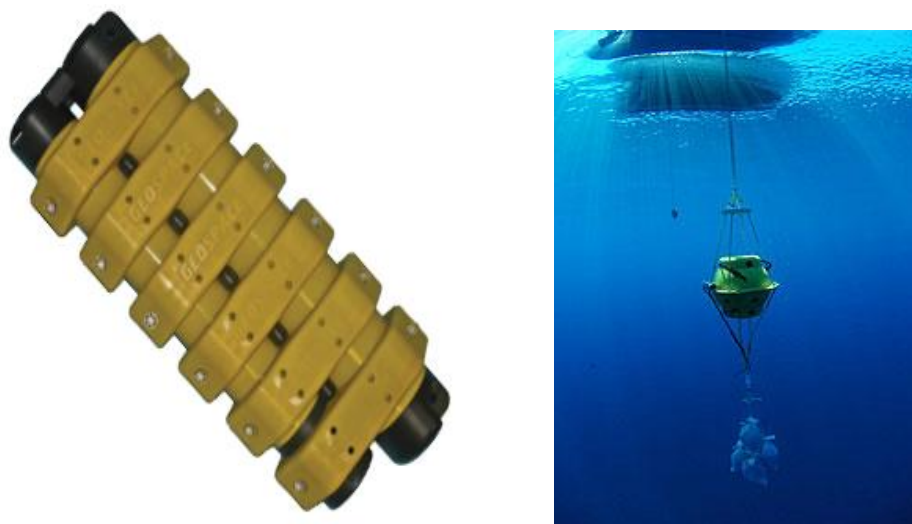


Figura 1: Gravadores de sons (GERI e Cornell Lab)

Tratando-se especificamente da monitoração de pássaros, ponto de interesse deste trabalho de pesquisa, torna-se importante citar contribuições científicas e de engenharia disponibilizadas à comunidade interessada. Desta forma também será

pertinente a identificação de questões que ainda aparecem como empecilhos para a evolução da área, tanto no que se refere aos aspectos científicos quanto aos tecnológicos.

Dentre os avanços científicos pertinentes da área, Vilches et al (2006) propôs um classificador de pássaros através do canto. O trabalho descreve uma aplicação de conceitos e técnicas sobre bases de dados que visa à extração de informações relevantes em cantos de pássaros. A conclusão do estudo é uma síntese de vários tipos de classificadores automáticos os quais posteriormente são aplicados aos dados dos animais. O método do trabalho resulta satisfatoriamente em índices superiores a 90% (noventa por cento) de acerto para algumas espécies.

Posteriormente a Vilches, vários outros pesquisadores preocuparam-se em desenvolver métodos computacionais que classificavam automaticamente os pássaros através do seu canto. Briggs (2009), por exemplo, aplicou vários extratores de características e sintetizou algumas técnicas adequando-as ao assunto. Outro exemplo é Fagerlund (2007) que se utilizou da técnica de *Support Vector Machines* (SVM) para classificar automaticamente os cantos dos pássaros. Ambos, da mesma forma que Vilches, em alguns dos testes obtiveram acertos de mais de 90% (noventa por cento).

1.1 PROBLEMA

Os trabalhos citados, os quais são mais bem relatados na seção 'Revisão Bibliográfica', obtiveram resultados satisfatórios no desempenho dos seus algoritmos classificadores. O fato é que, mesmo com estas contribuições, ainda existem limitações no sentido de viabilizar a pesquisa para que seja possível sua aplicação em ambientes reais.

Bardelliet. al. (2009) notou este fato ao testar seus algoritmos na prática. O autor realizou cinco dias de teste em ambientes abertos, e obteve mais de 50% dos resultados com problemas. Falsos positivos e poucas detecções reais foram feitas durante a coleta e a classificação. Segundo o autor, que em laboratório obteve bons resultados, os fatores mais agravantes foram ventos e ruídos intensos presentes no ambiente em que foi realizada a coleta, introduzindo perturbações e dificultando a classificação.

Agranat (2009) percebeu o mesmo problema e ressaltou pontos negativos como ruídos (provenientes de microfones convencionais) que "mancham" o sinal

sonoro. O autor ainda destaca fatores como baixas frequências aleatórias, ventos, chuva, interferências e a grande quantidade de variáveis oriundas do ambiente externo que acabam por anular o que é relevante na análise final do espectro. Conseqüentemente os resultados ostentam uma baixa taxa de acerto na classificação automática de espécies de pássaros.

Muitas vezes o canto de teste é limpo, sem ruídos e restrito a uma determinada faixa de frequência, tornando teórica toda a evolução em termos de algoritmos classificadores. Fato este que inviabiliza a utilização dos classificadores em ambientes reais e deixa sem solução problemas como o controle automatizado da biodiversidade

Outrossim, um problema que os classificadores poderiam prevenir, caso a pesquisa fosse satisfatória para situações práticas, é relacionado à segurança orgânica (bem estar dos seres humanos), em outras palavras, acidentes com aves em zonas aeroportuárias. A CENIPA (Centro de Investigação e Prevenção de Acidentes Aeronáuticos) divulgou um dado interessante em um de seus artigos. No período de dois anos (2006 a 2008) foram registrados 1321 acidentes e colisões de aves com aeronaves no espaço aéreo do Brasil. Tal fato resultou em um prejuízo de mais de três milhões de dólares às companhias aéreas (CENIPA, 2009).

Matadouros, estações de beneficiamento animal e plantações ao redor das Áreas de Segurança Aeroportuária (ASA's) são os principais fatores que levam a acidentes do gênero. Analisando a gravidade do problema, basta exemplificar que a força de impacto de um urubu em uma aeronave pode chegar a sete toneladas (Mendonça, 2009).

As ASAs são áreas que por lei podem alcançar um raio de até 20 quilômetros ao redor do aeroporto ou local de pouso e decolagem das aeronaves (Constituição do Brasil, 1998). Portanto, são bastante amplas e, mesmo respeitando esta exigência da legislação brasileira para prevenir acidentes, o CENIPA anualmente demonstra estatísticas que comprovam o risco de colisões com pássaros nessas regiões.

1.2 CONTEXTUALIZAÇÃO

Reconhecendo as necessidades mercadológicas, científicas e ambientais relacionadas à fauna, empresas diversas como a francesa Sterela (Figura 2) e institutos estadunidenses como *Acorn Naturalist* e o *Cornell Lab of Ornithology*

desenvolvem produtos (*software* e *hardware*), pesquisas e serviços relacionados ao assunto. Trabalho de institutos e laboratórios como os citados oferecem fortes bases para o desenvolvimento de pesquisas na área de ornitologia. Como resultado, a pesquisa acadêmica, por exemplo, se beneficia e evolui utilizando-se de amostras (bases de dados) confiáveis para os testes dos algoritmos classificadores.



Figura 2: Estação para assustar pássaros (Sterela, França)

Nesse contexto, engenheiros e pesquisadores diversos, no mundo todo, trabalham em paralelo para aprimorar o estudo da computação aplicado à ornitologia. Kwan et al (2005), por exemplo, apresentaram um método de detecção de direção de pássaros em campo aberto. O projeto consiste de microfones estrategicamente instalados em um campo aberto de forma a detectar a direção do canto (DOA - *Direction of Arrival*) do pássaro. Os autores utilizaram técnicas de processamento de sinais e inteligência artificial para captar não somente a direção de vôo do animal, mas também realizar a sua classificação.

Com a evolução dos microprocessadores no que diz respeito à incorporação de técnicas de processamento digital de sinais (PDS), trabalhos anteriormente inviáveis para um controlador embarcado, agora são considerados viáveis e comercialmente interessantes (Douglass, 1997). O trabalho de Wu, Siegel e Pradeep (1998) é um exemplo. No trabalho são caracterizados alguns defeitos do carro pelo som que as suas peças mecânicas produzem.

Com o passar dos anos realizaram-se pesquisas, trabalhos, patentes e propriedades intelectuais que abstraem dentro dos sistemas digitais (particularmente nos sistemas microprocessados) as técnicas de processamento de sinais analógicos. Circuitos integrados cada vez mais funcionais estão disponíveis para comercialização. Filtros digitais, rotinas matemáticas embutidas, operações

complexas com pontos flutuantes e os aceleradores matemáticos construídos em *hardware* são algumas das facilidades encontradas nestes componentes(Douglass, 1997).

Portanto, visando preencher algumas lacunas tecno-científicas, esta pesquisa se utilizará dos avanços científicos e tecnológicos citados. Conceitos relacionados ao PDS, as descobertas na área da ornitologia e os classificadores computacionais serão as bases para propor e implementar uma plataforma de coleta e processamento de cantos de pássaros, focando a etapa de coleta de amostras e a etapa de pré-processamento.

Compreende-se que, ao fornecer um sinal pré-processado, com características as quais permitam a utilização dos algoritmos de classificação amplamente abordados na literatura correlata, mas aplicados inicialmente a sinais idealizados em ambiente de laboratório, é gerada uma contribuição relevante para aplicação prática destes classificadores.

1.3 OBJETIVOS

1.3.1 Objetivo geral

Estudo, projeto e validação de uma plataforma de aquisição e pré-processamento de sinais sonoros para classificadores automáticos de sons de pássaros, visando melhorias nas coletas em ambientes reais no que diz respeito à qualidade das amostras e tempo de coleta.

1.3.2 Objetivos específicos

- Oferecer solução para o problema de classificação de cantos de pássaros, no que diz respeito ao pré-processamento das informações coletadas.
- Integrar em uma plataforma de *hardware* funcionalidades para análise e filtragem de sinais sonoros.
- Levantar eficiência de métodos computacionais de redução de ruídos sonoros e das técnicas de processamento de sinais em sistemas embarcados.
- Validar a plataforma final através de comparações gráficas e estatísticas dos sons filtrados pelo modelo, com os resultados oriundos das ferramentas já existentes de manipulação sonora.

1.4 JUSTIFICATIVA

O projeto de pesquisa do Professor Dr. Celso Antônio Alves Kaestner, nomeado '*BrBird*', iniciado em agosto de 2009, foi o principal fator motivacional desta pesquisa. Com a preocupação de controlar um ecossistema e analisar suas mudanças, o projeto foca em encontrar maneiras pouco invasivas de estudo dos pássaros. Proventos desta pesquisa vieram com os resultados encontrados pelos pesquisadores Lopes, Silla, Koerich e Kaestner (2011) em que foram aplicados algoritmos classificadores em bases de dados formadas por cantos de pássaros, alcançando acertos de mais de 95%.

A pesquisa de Lopes, baseada em artigos do Professor Kaestner (Silla, Kaestner e Koerich, 2007) e (Silla, Kaestner e Koerich, 2008), mostrou boas taxas de acertos dos algoritmos classificadores sobre as bases de dados reais. Uma conclusão importante desta pesquisa foi que os resultados não são prioritariamente dependentes do algoritmo classificador. Lopes conclui que o algoritmo escolhido, isoladamente, não é garantia de bons resultados e que fatores determinantes são as etapas de pré-processamento sonoro e extração de características.

Diante do contexto abordado, o que se propõe é uma forma de tratamento que vise facilitar algumas etapas da coleta e preparo dos cantos de pássaros. Facilitar pelo fato de que ao mesmo tempo em que a coleta é efetuada, aproveita-se os recursos técnicos atuais para realizar a etapa de pré-processamento, fundamental à classificação dos pássaros.

Com um método eficiente e automatizado de coleta do som dos pássaros, o pesquisador pode, por exemplo, dedicar maior tempo de pesquisa a tarefas mais relevantes como o estudo das bases de dados, validações e comprovações do sistema de reconhecimento através do aperfeiçoamento do algoritmo de classificação.

Baseando-se em soluções e dificuldades apresentadas pelos pesquisadores da área, este trabalho buscará uma solução na forma de uma Unidade de Aquisição Autônoma (UAA) dedicada, mostrada como uma plataforma embarcada que realiza o preparo dos sons que são recolhidos em ambientes reais. Este preparo envolve ações como filtragem, cálculos pertinentes para detecção de um som em potencial e etapas de separação inicial do som que podem auxiliar na classificação. Espera-se

com tal plataforma, contribuir com uma solução útil que ajude na evolução de aplicações que sejam mais práticas e genéricas do que as feitas até agora nos laboratórios.

É importante destacar, para fins de conclusões futuras, que o desenvolvimento deste trabalho não busca um classificador ideal para todos os tipos de pássaros, ou que funcione de forma automática para classificá-los diretamente na natureza. Também é válido salientar que, contrariamente ao que ocorre na maioria dos trabalhos de classificação, nos testes de validação desta pesquisa não serão separadas determinadas espécies de pássaros para validar algum processo de classificação automática.

A parte de tal questão, a missão aqui é viabilizar e automatizar a pesquisa prática de ornitologia, no que diz respeito à coleta de canto de pássaros. Viabilizar segundo os objetivos aqui apresentados é fornecer condições para que os ótimos resultados mostrados nesta introdução, obtidos comumente em laboratório, possam ser obtidos também em condições reais.

Durante o desenvolvimento deste trabalho serão enumerados e explicados vários conceitos relacionados ao assunto, particularmente àqueles que tratam sobre processamento de sinais em sistemas embarcados e como este recurso pode ser aplicado à classificação e pré-processamento de cantos de pássaros. Ao final deverá ser obtido um dispositivo computacional embarcado, projetado utilizando técnicas de engenharia de sistemas, e que seja vantajoso à realização da tarefa em questão.

1.5 ESTRUTURA DO TRABALHO

Para apresentar o modelo UAA de pré-processamento sonoro e descrever suas vantagens sobre as demais formas de coleta do som dos pássaros, esta dissertação de mestrado será dividida em seis capítulos.

No capítulo dois são apresentadas as bases conceituais desta pesquisa. O capítulo traz as explicações sobre ornitologia (estudo dos pássaros) e enquadra o processamento digital de sinais sonoros ao estudo. São mostradas visões e exemplos sobre pré-processamento sonoro, modelos estatísticos e extratores de características.

No capítulo três são detalhadas as etapas de construção do modelo que formará a UAA. O capítulo segue uma abordagem utilizando engenharia de sistemas

com *SysML* abrange etapas do desenvolvimento do trabalho (modelagem, desenvolvimento, teste e produção).

No capítulo quatro é mostrado a materialização da modelagem construída no capítulo anterior. Ainda são mostrados os testes de coleta de sons, bem como os testes realizados nos filtros da UAA e no extrator de características.

No capítulo cinco o modelo da UAA é validado e discutido com base nos testes realizados no capítulo anterior enfatizando as funcionalidades e vantagens do método adotado. Nesta etapa a UAA é avaliada em termos de desempenho, é também comparada às ferramentas de análise sonora já existentes e são feitas as comprovações do extrator de características embarcado utilizando comparações com ferramentas previamente desenvolvidas. Ainda são comparadas amostras tratadas e pré-processadas na UAA com amostras manipuladas em ferramentas já existentes e testadas em trabalhos correlatos (Lopes e Kaestner, 2010).

No capítulo seis as conclusões sobre este trabalho são ostentadas, sendo abordadas projeções da aplicabilidade da plataforma desenvolvida em outras áreas de estudo, bem como outras propostas de trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA

Nas seções a seguir serão definidas e explanadas informações relativas ao estado da arte e da técnica (científico e tecnológico) que este trabalho está inserido. No que diz respeito a aspectos do estado da arte, serão mostradas contribuições no âmbito de classificadores computacionais e técnicas de pré-processamento de sinais sonoros. Ainda para a completa satisfação dos objetivos, serão mostrados tópicos envolvendo sistemas embarcados e abordado o estado da técnica aplicada nos sistemas comerciais autônomos relacionados ao tema, bem como as soluções práticas adotadas na área.

2.1 ORNITOLOGIA

No Brasil, o conceito de ornitologia é conhecido pelos pesquisadores da área a mais de 50 anos, prova disso é a existência de artigos sobre o assunto que datam da primeira metade do século XX (Straube, 2001).

O ornitólogo paranaense Fernando Costa Straube, é um exemplo de estudioso famoso por se preocupar com a temática e publicar trabalhos sobre a história e a evolução da ornitologia no Brasil. Straube (2001) resumiu e narrou em uma pesquisa, quase dois séculos de história da ornitologia do estado do Paraná. O autor detalha cronologicamente em seu artigo os métodos, características e peculiaridades dos estudiosos que contribuíram para este ramo no estado.

Notam-se no trabalho os altos e baixos na história da área. Por um lado, o autor positivamente cita ornitólogos que chegaram a levantar mais de 400 espécies de pássaros para estudos didáticos e científicos. Porém, por outro lado, escreve também sobre uma grande quantidade de estudiosos imigrantes e biólogos estrangeiros, os quais levaram para museus estrangeiros, grande parte do conteúdo disponível sobre a fauna dos pássaros brasileiros.

Straube, presidente da Sociedade Brasileira de Ornitologia (SBO) em 2010, defende os princípios de conscientização de ideias relacionadas à natureza e também enfatiza que estes devem ser levados a sério quando se estuda o ramo. Exemplo destas ideias é a maneira correta de observação e levantamento das aves em um ambiente. Isto porque esta simples ação, dependendo de como for realizada,

pode ser considerada nociva e impactante ao bem estar do meio ambiente e da fauna (Pinilla, 2000).

Tal fato acontece, pois campanhas intensas com objetivos benéficos e planejamento duvidoso acontecem e trazem, em alguns casos, ações nocivas ao nicho ecológico das espécies (Straube, 2005). Motivos como esse fazem com que cada vez mais sejam valorizados meios menos intrusivos para realizar o levantamento e acompanhamento dos pássaros e de toda a fauna.

Entretanto, o que acontece na maioria dos grandes levantamentos contemporâneos é ainda a utilização de métodos tradicionais de levantamento qualitativo e quantitativo ornitológico. Ou seja, ainda são utilizadas técnicas convencionais da área, como por exemplo, captura por redes-neblina e espingardas de caça com baixo calibre (StraubeKrul e Scherer, 2005). Estas ações fazem com que tais métodos sejam muito nocivos aos animais, aumentando assim as chances de impactos ambientais.

Devido a essa e outras preocupações, é cada vez mais valorizado meios de realização do estudo não intrusivos. Anéis de identificação com dispositivos de rádio-frequência que permitem identificação à distância de pássaros soltos dos cativeiros são um exemplo de recursos utilizados que contribuem para o levantamento das espécies. Apitos atrativos para visualização e análise, construção de abrigos e microfones estrategicamente instalados são outros exemplos de cuidados interessantes e importantes para o estudo.

Ainda no que diz respeito a técnicas contemporâneas para o estudo em questão, está a análise do canto dos animais ou, tecnicamente, a bioacústica. Institutos estadunidenses como o *AcornNaturalist* e o *Cornell Lab of Ornithology* desenvolvem *softwares* profissionais e produtos que incentivam e ajudam no que diz respeito ao estudo. Um exemplo destes produtos é ilustrado na Figura 3, a qual mostra um microfone portátil, criado pelos pesquisadores do *Cornell Lab*, que grava sons do ambiente em que está inserido. A Figura 4 mostra mais um exemplo de um produto que é uma estação para os pesquisadores da área realizar seus estudos.



Figura 3: Estação para captura de sons (CornellLabofOrnitology)



Figura 4: Estação para estudo dos pássaros (RememBird.com)

O canto de um pássaro pode indicar características diversas do animal (Liu e Kroodsmá, 2007). Entre algumas destas estão a agressividade, períodos de estadia em cativeiros, vontades de interação com outros pássaros e até sinais de atenção a outros da mesma espécie.

Partindo destas premissas, estudos e trabalhos de especialistas como Betts (2006) e Briggs (2009) comprovam que através do canto dos pássaros é possível extrair, além das características, dados importantes como sua espécie, seu gênero e até seu bem-estar.

Juntamente com estudos como de Briggs e Betts, nas próximas seções são mostradas soluções que permitem a classificação automática dos cantos dos pássaros de acordo com sua espécie.

2.2 PROCESSADOR DIGITAL DE SINAIS

Os Processadores Digitais de Sinais (PDS), do inglês *Digital Signal Processors* (DSP), são na maioria das vezes circuitos integrados (CI's) programáveis similares aos microcontroladores 'tradicionais'. A diferença entre um DSP e um CI convencional está no fato do primeiro possuir alto desempenho em processamento com operações matemáticas (Duarte, Maitelli, Salazar et. al., 2006).

Em muitos casos, os fabricantes ou montadores destes dispositivos, adicionam aos mesmos aspectos interessantes que auxiliam os projetos da área. Sistemas operacionais embarcados e rotinas especializadas fazem com que cada vez mais este tipo de tecnologia se assemelhe aos microcontroladores de propósito geral.

A diferença entre esse grupo de CI's e o restante dos controladores está no desempenho desses em processar sinais analógicos e digitais. Multiplicadores em *hardware*, grande número de registradores, dispositivos de acesso direto a memória e disponibilidade de memória rápida são características dos DSP's (Kuulusa, Nurmi, Takala, et. al, 2002).

Em se tratando de aplicações que envolvam processamento digital de sinais, os trabalhos de Watts et. al. (2009) e de Saha et. al. (2006) são interessantes para exemplificar as evoluções contemporâneas. O primeiro apresenta alguns controladores embarcados com a finalidade de processamento digital sonoro a fim de copiar o sistema de audição humano. O segundo consiste na apresentação de um protocolo baseado em compartilhamento de carga em microcontroladores para aplicações de processamento de sinais distribuídos em sistemas embarcados.

Estes e outros trabalhos citados nesta seção tratam de novas técnicas para aplicações embarcadas, mostrando grande interesse acadêmico no assunto. A próxima seção trata do processamento digital de sinais sonoros, este que será utilizado como recurso para o desenvolvimento desta pesquisa.

2.3 PROCESSAMENTO DIGITAL DE SONS

Devido ao constante aumento de complexidade das tecnologias embarcadas, constantemente trabalhos envolvendo processamento de sinais sonoros são direcionados a desenvolver soluções que integrem o PDS aos sistemas embarcados (Melo e Cário, 2008). Estes, pela evolução dos microcontroladores e circuitos

integrados, se tornaram capazes de realizar tarefas anteriormente feitas somente pelos processadores convencionais de alto desempenho.

Com o avanço dos microcontroladores embarcados, tornou-se possível incorporar ao *hardware* algoritmos que aperfeiçoam atividades importantes e comuns na análise de sinais sonoros. Exemplo disso é um gravador que agrega ao seu *hardware* filtros digitais e remove alguns ruídos no momento da coleta do som.

Esta seção trata de algumas destas atividades do processamento digital de sinais sonoros. Nas próximas subseções serão feitas considerações sobre operações matemáticas em sinais e sobre os filtros digitais aplicados a estes. Eliminação de ruídos, amostragem, normalização e discretização são outras etapas que estão envolvidas na fase de análise do sinal e portanto serão também discutidas nestas subseções.

2.3.1 Transformada Discreta de *Fourier*

Basicamente um sinal de voz ou de qualquer outro som audível, pode ser expresso como uma forma de onda que se desloca no decorrer do tempo e possui inúmeras variações em sua amplitude (Osborn, 2010).

Sendo assim, para uma análise deste sinal, altamente variado no tempo, é preciso de um recurso que o organize e facilite esta tarefa. Um recurso para esta organização é a mudança de domínio deste espectro, em outras palavras, analisá-lo em outro universo matemático. Um recurso comumente utilizado é a análise no domínio da frequência do sinal, que pode ser alcançada através de operações matemáticas que constituem a Transformada Discreta de *Fourier* ou *Discrete Fourier Transform (DFT)*.

O resultado da aplicação da *DFT* em um sinal analógico é um conjunto finito de coeficientes relacionados a um valor de tempo. Tal técnica é útil, por exemplo, em soluções de equações que têm como variável independente o tempo e que por diversos motivos podem ser difíceis de equacionar, a exemplo das equações diferenciais. Estas equações ao serem transpassadas ao domínio da frequência passam a ser interpretadas como equações algébricas, tornando-se fáceis de serem resolvidas (Spiegel, 1976). Por esse motivo, os sistemas elétricos, óticos e mecânicos são comumente resolvidos utilizando a *DFT* (Brighan, Morrow, 1967).

O exemplo abaixo ilustra esta situação com uma função de pulso unitário $u_0(t)$ (Eq. 1), a qual quando submetida à transformação de domínios, deduz-se após

algumas operações em uma equação algébrica (Eq. 2) muito mais simples que a primeira (Souza, 2010). No exemplo será utilizado a Transformada de *Laplace*, que assim como a DFT, permite converter equações diferenciais em algébricas e representá-las no domínio da frequência, sendo a Transformada de *Laplace* uma generalização da Transformada de Fourier (Zill, 2009)

$$u(n) = \begin{cases} 1 & \text{para } n = 0 \\ 0 & \text{para } n \neq 0 \end{cases} \quad \text{Eq. 1}$$

$$\mathcal{L}[u(n)] = 1 \quad \text{Eq. 2}$$

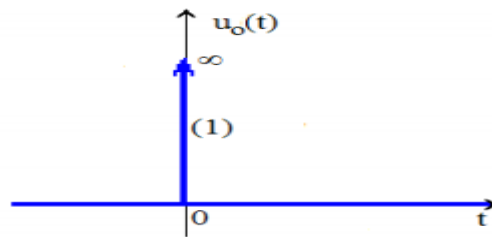


Figura 5: Função degrau unitário (Souza, 2010)

Ainda como vantagem da transformação de domínios, o resultado da transformação pode ser submetido à transformada inversa. Neste caso se corretamente tratada, a saída, pode vir a gerar um sinal (som) “mais limpo”, descartando-se, por exemplo, alguns valores no resultado da transformada que são enquadrados como ruídos do som, geralmente na forma de frequências elevadas e harmônicas não fundamentais (Galvão, Visani e Saldanha, et. al, 2001).

As equações Eq. 3 e Eq. 4 mostram as equações que representam as transformadas de *Fouriere* sua inversa respectivamente. Considerando N o número de amostras em um tempo k (0,1,2 até $N-1$), a primeira equação mapeia estas amostras discretas no domínio do tempo em uma representação discreta no domínio da frequência. A segunda equação, por sua vez, realiza o mapeamento contrário trazendo novamente o sinal no domínio temporal.

$$X(j) = \frac{1}{N} \sum_{k=0}^{N-1} x(k) e^{-i2\pi jk/N} \quad \text{Eq. 3}$$

$$x(k) = \sum_{j=0}^{N-1} X(j) e^{i2\pi jk/N} \quad \text{Eq. 4}$$

Existem alguns problemas computacionais neste tipo de transformação. Tais problemas estão relacionados a fatores como demasiada complexidade computacional nas operações matemáticas e limitações de pilhas e registradores para armazenar valores cumulativos das multiplicações da *DFT*.

A seguinte situação exemplifica estes fatores: certo tipo de controlador que detém uma limitação em sua arquitetura e por algum motivo possui uma capacidade máxima de computar um sinal em tempo real com 1000 pontos discretizados ($N = 1000$). Considerando que o mesmo controlador continuamente, em tempo real, processa toda a amostra em 1.000.000 de ciclos em instruções de multiplicação e soma (MLA nos processadores *RISC*) e que cada instrução leve 1ns, então todo o espectro seria processado em 0,001 segundo. O resultado seria uma frequência de amostragem máxima de 1kHz (Osborn, 2010), valor impraticável para processar em tempo real a maioria dos sons.

$$\begin{aligned}
 1000 \text{ amostras} &\rightarrow 1.000.000 \text{ de ciclos de } 1 \text{ ns;} \\
 1000 \text{ amostras} &\rightarrow 0,001 \text{ segundo;} \\
 \text{Período de amostragem} &\rightarrow T_s = 1 \text{ ms;} \text{ e} \\
 \text{Freq. de amostragem máxima} &\rightarrow f_s = 1 \text{ kHz}
 \end{aligned}$$

Coley juntamente com Tukey em 1965 (Cooley e Tukey, 1965), cientes das deficiências da *DFT* em aplicações em tempo real, publicaram um método eficiente capaz de processar um sinal digital no domínio da frequência com menor esforço computacional. Tal método é conhecido por Transformada Rápida de *Fourier*.

2.3.2 Transformada Rápida de *Fourier*

Assim como a *DFT*, a Transformada Rápida de *Fourier* ou *Fast Fourier Transform (FFT)*, também garante e determina maior controle sobre algum tipo de variável, equação ou um sinal sonoro. A particularidade está no fato da *FFT* levar um tempo muito menor que a *DFT* para ser computacionalmente processada.

Em termos de ordem de carga computacional, enquanto a *DFT* é enquadrada na ordem de N^2 , a *FFT* é representada por um algoritmo na ordem de $N \cdot \log_2 N$ (Brigham e Morrow, 1967). Isto porque a *FFT* utiliza-se de um método que reagrupa os coeficientes simétricos já calculados da *DFT* para serem utilizados nas operações futuras.

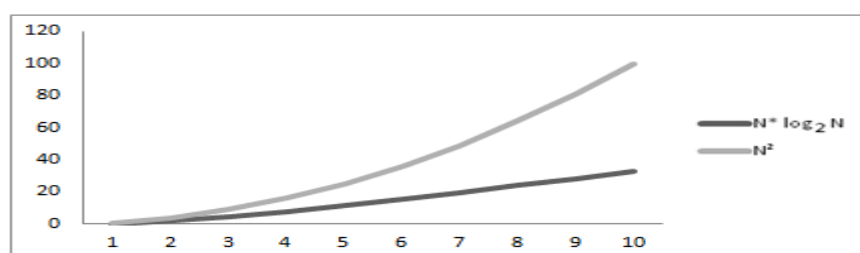


Figura 6: Gráficos de complexidade (Autoria própria)

Para resolver este problema, o algoritmo *FFT*, segundo Brigham e Morrow (1967), divide o comprimento N de uma transformada em duas partes ($N/2$). Tal divisão é possível pela redundância de informação decorrente da simetria do sinal no domínio da frequência (Figura 7).

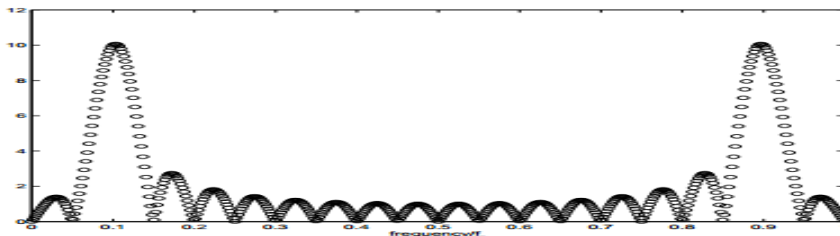


Figura 7: Simetria de uma DTF (Brigham e Morrow, 1967)

Tal simetria, portanto, permite diminuir as operações matemáticas. Em outras palavras, os valores passam a utilizar-se dos resultados das outras operações, sendo que, o que é alterado, no avanço dos cálculos, é somente o fator de giro da transformada. A Figura 8 ilustra esta situação de reutilização dos resultados das operações no algoritmo dos criadores da *FFT*.

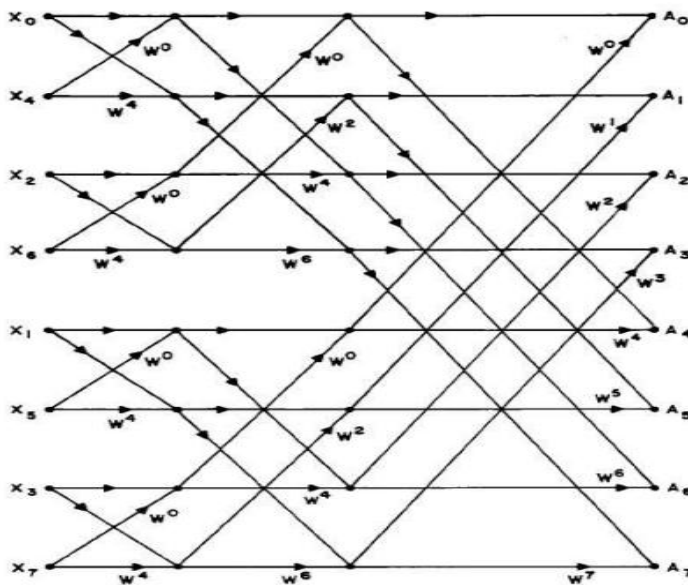


Figura 8: Algoritmo de Cooley-Tukey base dois (Cooley e Tukey, 1965)

O algoritmo de *CooleyTukey* base dois utiliza-se geralmente de amostras com comprimento que são potência de dois (16, 32, 64 etc.). Tal fato permite a adequada divisão ($N/2$) do sinal submetido à *FFT*.

Na gravação por amostragem de um sinal, limitar um sinal à N amostras que são potência de dois, pode implicar em interrupções ou truncamentos no sinal gravado. Estes truncamentos ocasionam perdas das características espectrais na

amostra e para tentar corrigir estas perdas as funções de janelamento são incorporadas as etapas da *FFT* (Champagne e Labeau, 2004).

2.3.3 Funções de janelamento

Para reduzir perdas características espectrais no sinal gravado, as funções de janelamento fazem parte da rotina de transformação do sinal sonoro para o domínio da frequência. O que é feito por estas funções consiste em um espalhamento espectral nos truncamentos existentes. A Figura 9 ilustra algumas funções de janelamento comumente utilizadas em PDS.

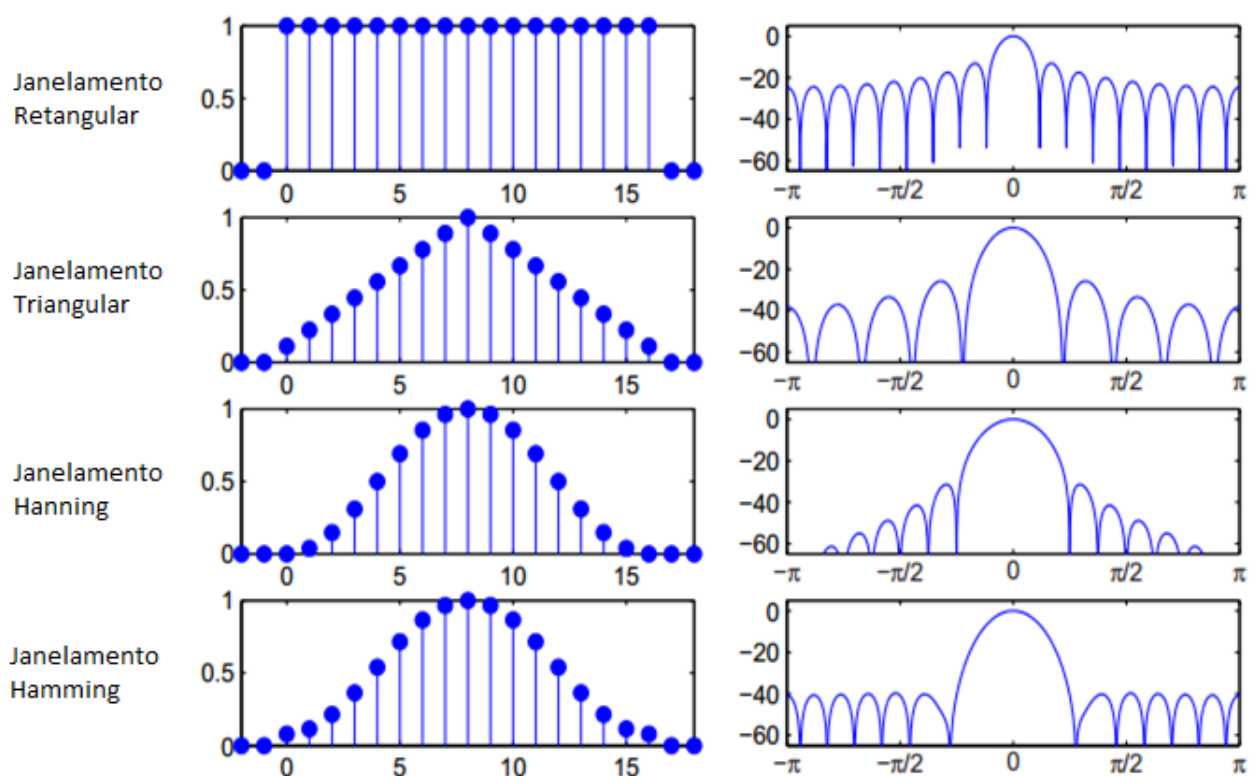


Figura 9: Funções de janelamento (Adaptado de Champagne e Labeau, 2004)

O primeiro par de gráficos na primeira linha ilustra o janelamento retangular. O gráfico da direita mostra o comportamento da janela no domínio do tempo (tempo X amplitude) e o da esquerda o comportamento no domínio da frequência (frequência X magnitude). Neste caso, para dados dentro da janela o peso um (1) é atribuído à amostra e dados que estejam fora recebem peso zero (0). Tal fato faz com que apenas certas porções dos dados sejam vistos, suprimindo-se o restante.

Nas linhas abaixo são mostrados alguns janelamentos amortecidos, que por sua vez reduzem gradualmente os pesos atribuídos aos dados analisados. A segunda linha ilustra o janelamento triangular ou de *Bartlett*, seguido do janelamento

de *Hanning* (terceira linha) e por último o de *Hamming*. O último janelamento difere do penúltimo sendo ambos parecidos com uma onda cossenoidal de meio ciclo. A sutil diferença está no fato que a última não tende a zero como é mostrado na última linha da Figura 9.

Os resultados da *FFT* com janelamento são de grande utilidade no projeto de algoritmos para reconhecimento de padrões sonoros. Somervuo, Härmä e Fagerlund (2006) com o intuito de testar algoritmos classificadores de pássaros, utilizaram a *FFT* e o janelamento de *Hanning* resultando em um espectro somente com as frequências almejadas e pertinentes da amostra (Figura 10).

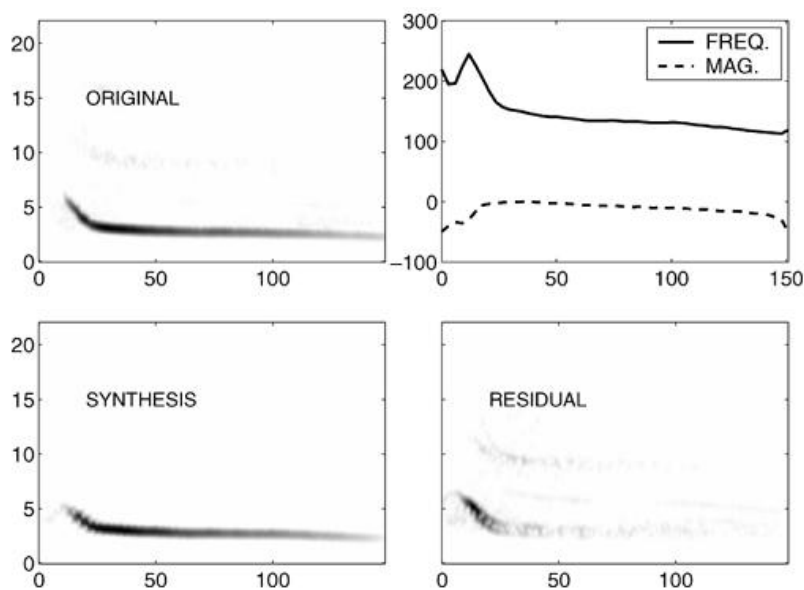


Figura 10: Sinusoide sintetizada com FFT (Somervuo, Härmä e Fagerlund, 2006)

2.3.4 Filtros Digitais

Redes de transmissão de energia elétrica, circuitos eletrônicos, circuitos ópticos e processamento digital de sinais são apenas algumas das situações ou ambientes em que comumente se faz uso de algum tipo de filtro, seja ele digital ou analógico. Um filtro analógico geralmente é representado por algum tipo de estrutura física e concreta, por exemplo, uma lente ou um circuito simples Resistor-Capacitor (RC).

Os filtros digitais, por sua vez, são filtros que são implementados geralmente em *software*, através de algoritmos específicos para cada situação. Tal característica faz dos filtros digitais uma ferramenta flexível a testes, isto pela sua dinamicidade e capacidade em manipular facilmente os parâmetros de ajuste do filtro (Ess, 2011).

Basicamente existem duas formas básicas de filtros digitais, os de Resposta ao Impulso Finito ou *Finite Impulse Response (FIR)* e os de Resposta ao Impulso Infinito ou *Infinite Impulse Response (IIR)*.

A denominação 'finita', atribuída aos filtros do tipo *FIR*, existe pelo fato da utilização de um número finito de pontos da amostra na filtragem. Este fato os torna simples de entender e programar. A Figura 11 ilustra a estrutura deste tipo de filtro. Os parâmetros de calibração são indicados pelos valores ' a_n ', os quais são multiplicados pelos valores obtidos na amostragem ou discretização do sinal.

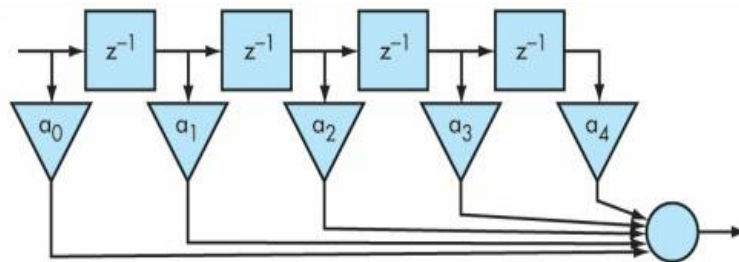


Figura 11: Filtros FIR (Adaptado de Ess, 2011)

Uma característica importante destes filtros é que em hipótese alguma estes filtros serão um oscilador (Ess, 2011). Em outras palavras, os filtros *FIR* são totalmente estáveis devido as suas etapas de saída contarem sempre com o valor da iteração em questão.

Os filtros *IIR*, por sua vez, dependem de etapas anteriores, gerando laços interdependentes uns dos outros para realizarem os cálculos. Tal característica implica que uma saída pode depender de várias entradas anteriores, podendo gerar assim repetições instáveis que podem causar oscilações indesejadas na saída do sistema (Ess, 2011). A Figura 12 ilustra o comportamento deste tipo de filtro.

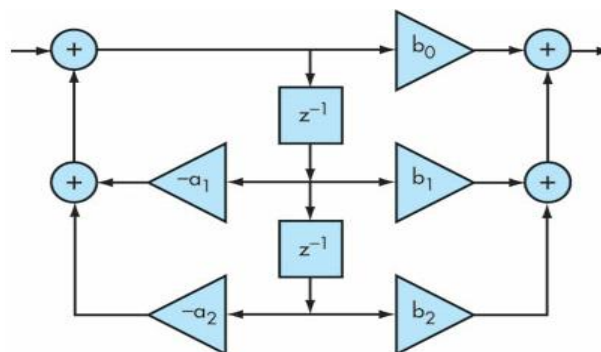


Figura 12: Filtros IIR (Adaptado de Ess, 2011)

Os filtros *FIR* e *IIR* estão intimamente relacionados com aplicações de processamento digital de sinais sonoros. Um exemplo clássico é a aplicação dos filtros na interpretação e reconhecimento da voz humana. Com uma variação de

frequência estreita (de 1 KHz até 3 KHz), alguns filtros simples, como os passa alta e os passa baixa, são capazes de realizar as filtrações necessárias, permitindo manipulação eficaz da nossa voz (Vilches et al, 2006).

2.4 PRÉ-PROCESSAMENTO DE AMOSTRAS SONORAS

A aquisição de cantos de pássaros durante muitos anos vem sendo feita de forma eficiente pelos pesquisadores, especialistas e estudiosos da área que vão até os *habitats* dos animais e realizam a coleta dos sons. O trabalho destes indivíduos, muitas vezes, envolve tarefas como separação manual dos períodos de gravações que não são pertinentes à amostra e deslocamentos no ambiente de coleta, objetivando obter melhor qualidade e resolução do som (Straube, 2001) (Straube, Krul e Carrano, 2005).

O fato é que mesmo com grande porcentagem de acerto dos algoritmos, devido aos filtros aplicados, existe uma grande dificuldade em criar uma solução mais prática para o problema de coleta e preparo dos cantos. Exemplo disso são as 2887 horas despendidas em esforço amostral da pesquisa de Straube, Krul e Carrano (Straube, Krul e Carrano, 2005), que poderiam ser amortizadas caso houvesse uma solução mais prática e dinâmica do que a coleta manual.

Agranat (2009) mostra um estudo interessante para o tópico em questão. O autor enumera algumas variáveis que dificultam o estudo e criação de uma solução prática para um classificador automático de pássaros. Exemplo destas variáveis são os ruídos encontrados nos ambientes externos, tais como sobreposição de sons e harmônicas indesejadas que aparecem no sinal. Outra dificuldade apresentada por Agranat está no fato dos pássaros, diferentes dos seres humanos, transmitirem seus sons em uma faixa de frequência ampla que vai de 100 Hz até 10 KHz, o que dificulta as ações dos filtros dado a extensa faixa em questão.

Já sabendo destes empecilhos para a coleta automatizada, Vielliard (1996), buscando qualificar o estudo, enumera algumas variáveis e aspectos que são importantes para a ornitologia no que diz respeito ao canto dos pássaros. Aspectos como posição geográfica onde o pássaro está na hora da coleta do canto, temperatura do ambiente, umidade e hora do dia são importantes e devem ser levadas em conta nas análises classificatórias e comportamentais dos animais. Ross (2006), por sua vez, complementou o estudo considerando como importante na

classificação dos pássaros, aspectos como separação dos sons em *frames* e ainda extração de mais de 20 características do som, como potência e frequência média.

Empecilhos existem ainda devido ao fato dos microfones unidirecionais gerarem espectros "borrados". Sons dos ventos, ruídos constantes, chuva, som das folhas das árvores e ainda baixas frequências oriundas de lugares mais distantes do sensor (microfone) podem ser a fonte deste sinal indesejado.

Uma primeira etapa, para a suavização destas variáveis indesejadas é, portanto, a utilização de um filtro. Um exemplo de filtro muito utilizado para esta finalidade é o filtro denominado *Wiener*, sendo este, capaz de eliminar grande parte de sons indesejados. O filtro atua sobre o sinal a partir de um som considerado ruído em uma determinada faixa do espectro, fazendo com que algum padrão parecido com o ruído indicado seja retirado do restante da amostra (Ifeachor e Jervis, 2002). Outro tipo de filtro simples que pode ser aplicado ao sinal digitalizado é o filtro corta-faixa. Altas ou baixas frequências são, muitas vezes, indesejadas no espectro sonoro. Como já descrito, sabe-se através de estudos biológicos e anatômicos a faixa de frequência aproximada que os pássaros conseguem cantar.

Outra etapa que deve ser feita a fim de preparar o som para ser classificado é a normalização em uma escala que seja comum a que o classificador analisa. Este procedimento é uma forma de organizar o sinal e também de restringir o que será interpretado. O ouvido humano realiza um procedimento parecido, escutamos apenas algumas faixas de frequências do som. Portanto, o processo consiste em tentar realçar algumas características do espectro que são importantes e devem ser consideradas no ato do processamento. Por exemplo, harmônicas redundantes se formam no espectro, oriundas de harmônicas fundamentais e nesse caso, aquelas não interessam na análise e podem ser descartadas. Um exemplo de normalização é a logarítmica. Esta trata de enfatizar harmônicas fundamentais (baixas frequências) e ignorar as harmônicas com frequências altas as quais são harmônicas redundantes à principal (Agranat, 2009).

Por peculiaridades como estas, para a descoberta de conhecimento em bases sonoras contendo cantos de pássaros, é conveniente ainda realizar um pré-processamento completo que envolva aspectos internos e externos da amostra. A Tabela 1 organiza alguns destes aspectos, discriminando cada um deles com uma breve descrição.

Tabela 1: Rotinas de pré-processamento

Fator	Importância / Solução
Hora do dia, data e temperatura	Parâmetros importantes para realizar levantamento sazonal e temporal dos espectros captados.
	A solução em sistemas embarcados é a utilização de relógios eletrônicos e sensores de temperatura.
Redução de ruídos	Eliminar borrados do espectro oriundos dos microfones unidirecionais (Agranat, 2009), sons indesejados e harmônicas não fundamentais.
	Aplicação de filtros, como por exemplo o filtro <i>Wiener</i> . (Ifeachor e Jervis, 2002).
Assimetria (<i>Skewness</i>)	Medida útil para encontrar a obliquidade da distribuição dos dados, em outras palavras, encontrar o quão simétrica é a distribuição dos dados (Downing e Clark, 2006).
	Podendo ser encontrado pela Eq. 5 onde Mt é um vetor com a distribuição dos dados, μ é a média, σ é o desvio padrão dos dados e N é o número total de elementos no vetor: $At = \frac{(\sum_{n=1}^N Mt[n] - \mu)^3}{\sigma N}$ Eq. 5
Curtose	Parâmetro estatístico essencial para medir o “achatamento” da distribuição dos dados (Downing e Clark, 2006).
	Dado pela Eq. 6 $Kt = \frac{(\sum_{n=1}^N Mt[n] - \mu)^4}{\sigma N} - 3$ Eq. 6
Centróide Espectral	Medida da forma espectral que informa um ponto de balanceamento do espectro, similar a uma média dos dados (Silla, Kaestner e Koerich, 2005).
	Representado pela Eq. 7: $Ct = \frac{\sum_{n=1}^N Mt[n] \times n}{\sum_{n=1}^N Mt[n]}$ Eq. 7
<i>Rolloff</i> Espectral	Medida da forma espectral que caracteriza os dados abaixo de determinado nível. Padrão que ajuda na frequência de corte dos ruídos (Silla, Kaestner e Koerich, 2005).
	Representado na frequência Rt apresentada pela Eq. 8 com nível de concentração igual a 85%: $\sum_{n=1}^{Rt} Mt[n] = 0.85 \sum_{n=1}^N Mt[n]$ Eq. 8
Taxa de Cruzamento zero	Forma útil de representação e comportamento do sinal sonoro. Trata-se de uma característica do sinal no domínio do tempo, diz a quantidade de vezes que o sinal variou de positivo para negativo. A função ‘ <i>sign</i> ’ retorna, respectivamente, 1 e 0 se o resultado for positivo ou negativo (Silla, Kaestner e Koerich, 2005).

Fator	Importância / Solução
	Representada por: $Z_t = \frac{1}{2} \sum_{n=1}^N \text{sign}(x[n]) - \text{sign}(x[n-1]) $ Eq. 9

Fonte: Autoria própria

Após as etapas responsáveis por “melhorar” um sinal sonoro, de forma a aumentar os resultados satisfatórios de um analisador sonoro, a próxima etapa é a submissão do som aos classificadores/analísadores sonoros. Dissertando sobre tal etapa as duas próximas seções se encarregam de tratar de tais tópicos.

2.5 MODELOS ESTATÍSTICOS E PROBABILÍSTICOS

Russell e Norvig, autores do livro ‘*Artificial Intelligence: A modern approach*’, reuniram várias técnicas e algoritmos sobre aprendizado de máquinas e sistemas de apoio a decisão. Os autores enumeram diversos métodos de representação e de tradução do conhecimento humano para um computador.

Técnicas como *Naive Bayes*, *KNearestNeighbor* (KNN) e redes neurais são exemplos de ferramentas importantes na resolução de problemas em que é necessária uma tradução de conhecimento tácito em explícito, ou seja, problemas em que é necessário formalizar o conhecimento.

Resulta-se de tais ferramentas, geralmente um nível de significância, pelo qual é possível estatisticamente dizer se alguma relação é verdadeira ou não. Sistemas dos mais variados tipos são aferidos ou não apenas com o resultado de simulações probabilísticas. Os testes de qualidade em uma linha de produção de parafusos, por exemplo, são feitos a partir de uma amostragem, sendo esta amostra suficiente para aprovar/reprovar toda uma cadeia produtiva.

No que se refere à probabilidade, existem hoje maneiras de se representar sistemas probabilísticos complexos em funções matemáticas simples. Esta representação pode se dar através dos cálculos gaussianos, os quais comumente são representados através de uma distribuição normal (Figura 13) (Laurino, 2004).

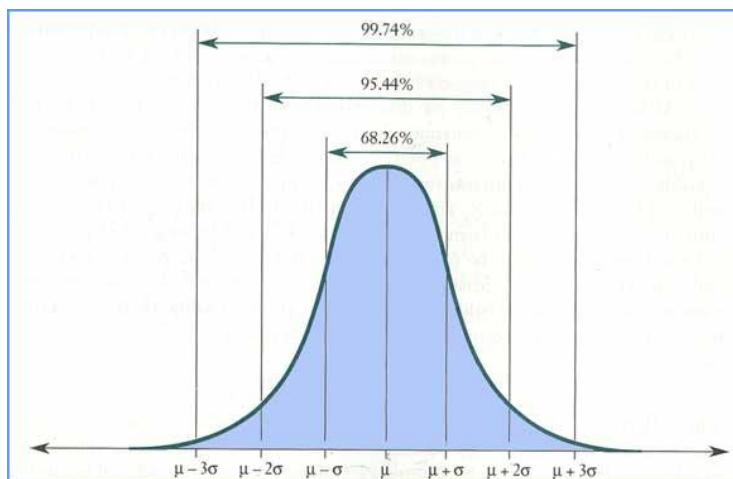


Figura 13: Distribuição normal (Hotta, 2009)

Com o auxílio da estatística e das técnicas de descoberta de conhecimento, alguns resultados nas pesquisas de classificação automática de pássaros pelo canto, demonstram uma taxa de acerto superior a 90% (Briggs, 2009).

Uma síntese parecida de classificadores para pássaros foi feita por Vilches, Escobar e Vallejo em um artigo de 2006 (Vilches, Escobar e Vallejo 2006). No estudo foram utilizadas algumas metodologias automáticas para validar os resultados. Complementado pontos fortes e fracos das metodologias os autores conseguiram apresentar resultados de alta precisão (Figura 14).

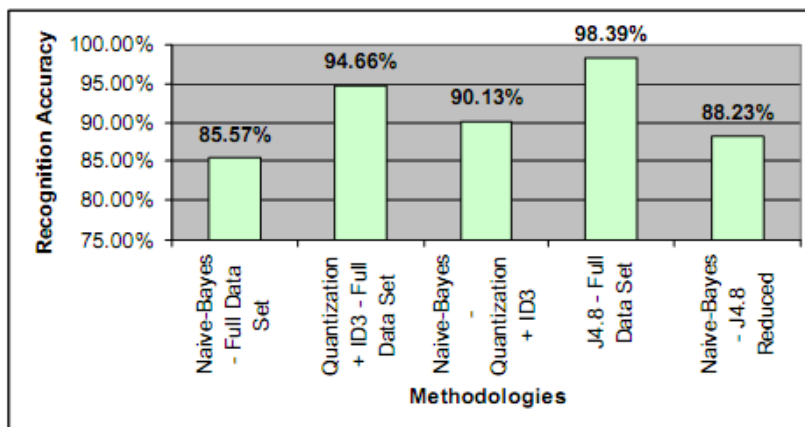


Figura 14: Gráfico de precisão (Vilches et. al., 2006)

Outros trabalhos, assemelhados aos citados, contribuíram para a formação do método de testes desta pesquisa (Capítulo 4). O trabalho de Lopes, Silla, Koerich e Kaestner (2011), já citados no capítulo anterior é um exemplo de como os testes com cantos de pássaros foram feitos. Outro trabalho é o de Fagerlund (2007), que propôs reconhecimento de sons dos pássaros usando SVM e elencou muitas das

variáveis probabilísticas (centróide espectral, taxas de cruzamento de zero, etc.) utilizadas no desenvolvimento da UAA.

2.6 CONCLUSÃO E CONSIDERAÇÕES

Mesmo com os meios eficientes de classificação automática para cantos de pássaros apresentados neste capítulo e a rápida evolução do assunto devido aos diversos interesses científicos, ambientais e profissionais da área, ainda existe uma lacuna que inviabiliza em termos de praticidade e evolução comercial todo este processo.

Fatos como ampla faixa de frequência emitida pelos mais variados pássaros e a grande quantidade de ruídos oriundos de ambientes externos são algumas das destas limitações. Estas limitações geram dificuldades na evolução das pesquisas da área o que faz com que os bons resultados apresentados sejam muitas vezes demonstrados em um ambiente ideal e restrito. Exemplo disso são as restrições a poucas espécies que são feitas nas pesquisas ou ainda as amostras de testes repetidamente preparadas e filtradas nos trabalhos científicos.

Os sons com imperfeições, segundo Agranat (2009), antes de serem testados pelos classificadores, requerem aplicação de filtros especializados, padronizações do sinal sonoro e algumas manipulações que são feitas pelos pesquisadores da área na maioria das vezes de forma manual.

Desta forma todo o potencial dos algoritmos de extração de características é atenuado quando inserido em um ambiente prático de testes, com grande quantidade de ruídos. Isto dito, o que será mostrado nas próximas seções é uma solução que pretende fornecer uma maneira de tornar utilizável todos estes bons resultados em um ambiente prático e real.

Fatores do estado da arte relacionados à ornitologia, algoritmos classificadores e processadores digitais de sinais formam a base deste trabalho e é a partir destes fatores que todo o restante do trabalho será desenvolvido. Uma organização mostrando alguns pontos fortes e fracos destes fatores é mostrada na Tabela 2.

Tabela 2: Estado da arte e da técnica da pesquisa

VANTAGENS	
Engenharia	Academia
<ul style="list-style-type: none"> • Tecnologias e produtos internacionais para coleta e análise de sons; • Produtos, bens, disponíveis no mercado que realizam processamento de sinais sonoros; • Gravadores específicos, coleta de sons da fauna para ambientes marítimos e terrestres. 	<ul style="list-style-type: none"> • Algoritmos com altos índices de acerto em classificação de pássaros através do canto; • Técnicas eficientes para pré-processamento e classificação; • Ambiente flexível para testes e otimização da solução da mesma forma que para descoberta de melhores métodos.
FALHAS	
Engenharia	Academia
<ul style="list-style-type: none"> • Alto custo; • Tecnologia Fechada; • As de fácil acesso não atendem requisitos de coleta de sons de pássaros automática. 	<ul style="list-style-type: none"> • Testes em amostras específicas e livres de variáveis do ambiente real (natureza); • Parte do projeto é dedicada a etapas de pré-processamento e preparo das amostras; • Testes restritos a poucas espécies; • Pouco se faz para tornar a pesquisa mais genérica no que diz respeito à aplicabilidade do sistema em várias espécies.

Fonte:Autoria própria

Uma vez registradas vantagens e desvantagens do que já existe sobre a classificação automática dos pássaros, é possível abstrair algumas necessidades que o assunto apresenta. A Tabela 3 organiza as atividades almeçadas para este trabalho, bem como o ponto que se deseja completar e o seu enquadramento em solução de engenharia ou da academia.

Tabela 3: Enquadramento da pesquisa

Característica deste trabalho	Característica a ser sanada/complementada
Formalização de um sistema para coleta de sons de pássaros.	Soluções caras e de difícil acesso no mercado, respectivamente, soluções importadas ou militares.
Rotinas de pré-processamento de som na coleta dos sons.	Trabalho constante e repetitivo em etapas de pré-processamento nos trabalhos da área.

Característica deste trabalho	Característica a ser sanada/complementada
Sistema embarcado para coleta de sons.	Testes em ambientes reais e maior praticidade na classificação automática dos pássaros.
Fornecer conjunto de características dos sons a classificadores automáticos.	Excessiva preocupação nas pesquisas científicas em etapas de pré-processamento.
Iniciar trabalhos em ambientes reais de classificação de pássaros.	Análise com restrição a poucas espécies e a sons filtrados e já preparados.

Fonte: Autoria própria

ENGENHARIA DE SISTEMAS

A fase inicial do estudo foi marcada pela associação de bibliografias técnicas com o estado da arte sobre ornitologia e pré-processamento de sinais sonoros. A partir do que foi justificado e estudado pela comunidade científica e tecnológica será formado para esta pesquisa um modelo desenvolvimento, testes e validação de resultados.

Analisando aspectos relacionados às deficiências acadêmicas e de engenharia da área, apresentados no capítulo anterior, será formalizado um modelo de sistema que viabilize pontos a serem melhorados no assunto. Etapas de pré-processamento sonoro, caracterização do sinal para auxiliar os classificadores automáticos e outras funcionalidades serão adicionadas a este modelo.

Tal modelo será construído através da Engenharia de Sistemas, fazendo uso da *SysML* (*SysML Specification* versão 0.9, 2005), e será de utilidade para organizar pontos do projeto, tais como custo, tempo de desenvolvimento, integração de partes e funcionalidades técnicas.

Isto dito, este capítulo foi dividido em cinco partes, estas baseadas no processo de sistemas de engenharia (Figura 15) criado por Bahill e Gissing (Bahill e Gissing, 1998) e utilizado por Weilkiens (WEILKIENS, 2006), ambos estudiosos de Engenharia de Sistemas e da *SysML*.

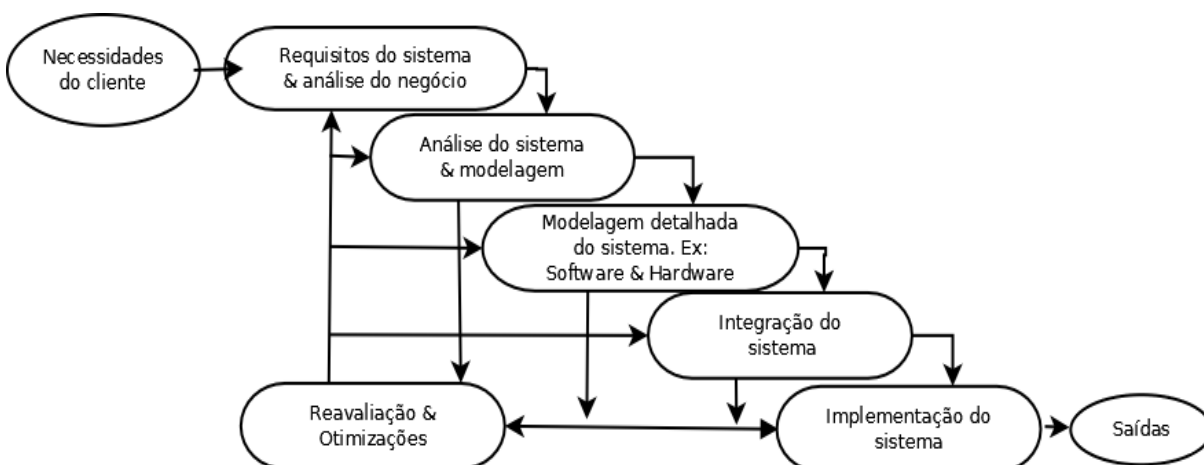


Figura 15: Processo de sistemas (Adaptado de Bahill e Gissing, 1998)

A primeira parte descreve uma ideia inicial do sistema e é mostrada na seção 'Introdução a Engenharia de Sistemas'. Esta ideia inicial será entrada para a

segunda parte, seção 3.2, intitulada como 'Engenharia de Requisitos' e que se correlaciona com as três primeiras fases do processo da Figura 15. A seção se desenvolve rotulando e definindo métricas sobre os requisitos do sistema. Dividida em quatro subseções esta seção trata:

- Dos requisitos de *hardware* do sistema fazendo relação com a parte física do sistema.
- Dos requisitos de *software*, estes que por sua vez, exploram as partes que compõem o *software* embarcado (*firmware*) do sistema.
- Da formalização e diagramação dos requisitos expostos, resultando aqui em um diagrama de requisitos, recurso presente na *SysML*.
- Por fim, da tradução dos requisitos em casos de uso e consequente parametrização em horas e custo do sistema.

Dando continuidade ao processo de Engenharia de Sistemas, a seção 3.3 trata do projeto do sistema, onde serão apresentados os modelos que representam o sistema. As duas últimas partes, seguindo o processo de Bahill e Gissing, diz respeito à integração (seção 3.4) e desenvolvimento do sistema (seção 3.5). Nestas seções são apresentadas as correlações de engenharia com os requisitos expostos na seção 3.2.

3.1 INTRODUÇÃO A ENGENHARIA DE SISTEMAS

É importante, para validar os objetivos deste trabalho, utilizar um método sistêmico que se mostre capaz de modelar por completo o que se deseja realizar. Traduzir os interesses de uma situação ou pessoa é um trabalho que demanda algum tipo de preparo anterior à aplicação para diminuir erros e gastos no projeto.

Objetivando traduzir corretamente os requisitos desta pesquisa ao modelo almejado, será utilizada desde as fases iniciais do trabalho, uma linguagem de descrição de sistemas intitulada como *SysML*. Esta, derivada da Linguagem de Modelagem Unificada ou *Unified Modeling Language* (UML), tem como função descrever estruturas complexas de sistemas, aproximando uma situação ou interesse pessoal em um modelo de engenharia (*SysML* 0.9, 2005).

Sendo assim a Figura 16 ilustra uma ideia inicial, representada por um diagrama de blocos, do sistema coletor e pré-processador de sons.

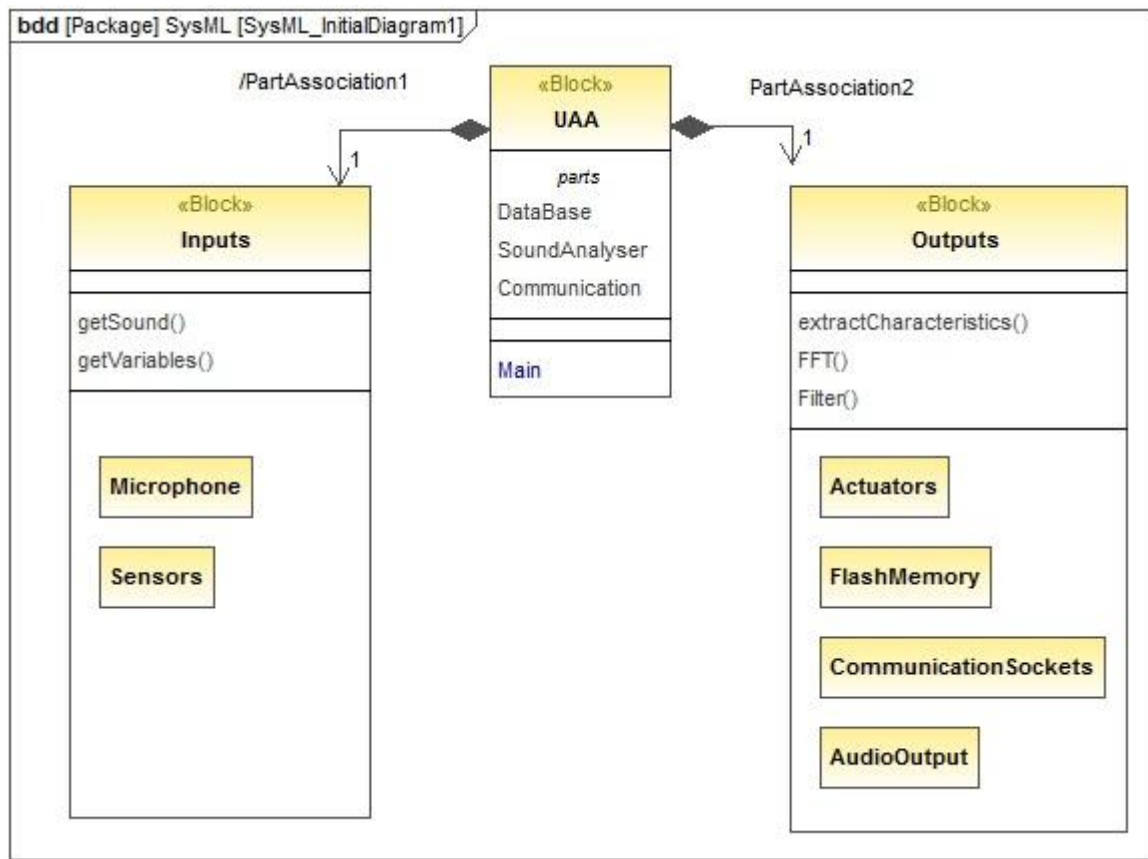


Figura 16: Formato inicial do sistema (Autoria própria)

A Unidade de Aquisição Autônoma (UAA) consiste em um equipamento dedicado a gravação e análise de sons do ambiente. Por esse motivo a Figura 16, mostra em sua unidade central um bloco que é responsável por analisar este som, armazená-lo de alguma forma e transmitir as informações a quem for interessado. Entradas e saídas compõe o sistema. Como entradas têm-se o microfone e alguns sensores, como saídas os atuadores, uma memória *flash* e *sockets* de comunicação para exportação dos dados e o áudio coletado já processado pelo equipamento.

3.2 ENGENHARIA DE REQUISITOS

A partir deste panorama geral do sistema, utiliza-se a linguagem de descrição de sistemas *SysML* para formalizar alguns pontos principais do trabalho. Dentre vários diagramas e etapas previstos pela modelagem de sistemas, está o diagrama de requisitos (ausente na UML), útil para representar de forma organizada as necessidades do sistema e desejos dos seus interessados (Weilkiens, 2006).

Para este trabalho, os requisitos serão levantados de acordo com as lacunas encontradas no estado da arte e da técnica da aquisição de sons de pássaros e

posteriormente o diagrama de requisitos será apresentado. Logo, a Tabela 4 apresenta os requisitos gerais do sistema. A partir destes será derivada toda a modelagem do trabalho, bem como outros requisitos mais específicos, que por fim formarão o diagrama de requisitos do sistema.

Tabela 4: Requisitos de sistema

ID	Descrição
REQ 001	O sistema deverá ser capaz de receber, analisar, filtrar e gravar sons oriundos do ambiente externo.
REQ 002	O sistema deverá possuir uma interface de acesso remoto a base de dados em que estarão salvos os dados referentes aos sinais sonoros coletados.
REQ 003	O sistema deve ser específico, diferenciando-se assim de um sistema ou computador de propósito geral.
REQ 004	O sistema deve ser capaz de recolher e fornecer informações relacionadas ao ambiente externo como temperatura, hora e data.
REQ 005	O sistema deve ser capaz de processar os sons de forma a prepará-los a um classificador automático de espectros sonoros.

Fonte: Autoria própria

Com intuito de separar aspectos que se referem ao *hardware* e ao *software* do sistema, foi criada uma divisão para elencar o restante dos requisitos. Esta divisão existe pelo fato que, para a completa satisfação dos requisitos do sistema, há uma necessidade de especificação dos assuntos técnicos, uma vez que a Tabela 4 é ainda uma generalização do que o sistema deve fazer. Nas próximas subseções, portanto, é mostrado o detalhamento desta derivação.

3.2.1 Requisitos de *hardware*

Nesta subseção serão justificados e explicados os requisitos de *hardware* pertinentes à construção do sistema proposto. Estes, com suas características e funcionalidades, têm a função de validar alguns dos requisitos e objetivos do projeto já apresentados na Tabela 4.

A Tabela 5 organiza estes requisitos, os quais são explicados na próxima seção. Derivando da necessidade do sistema ser específico (REQ 003), estes requisitos restringem o uso de gravadores convencionais ou computadores comuns no sistema, restando a utilização de recursos microcontrolados diferentes destes citados. Partindo deste fato será feita a escolha do controlador que será utilizado no

projeto, bem como serão apresentados circuitos e recursos importantes ao sistema almejado.

Tabela 5: Requisitos de *hardware*

ID	Descrição	Derivado do Requisito
REQ HW 201	O <i>hardware</i> deve poder atuar e sensoriado ambiente externo em que está inserido.	REQ 004
REQ HW 202	O sistema deverá conter interfaces de <i>hardware</i> para memórias <i>flash</i> (e. g.: <i>pen drive</i>).	REQ 003
REQ HW 203	O <i>hardware</i> deverá possuir suporte para interface de <i>debug</i> e formas de comunicação utilizando padrões de comunicação serial (e. g. USB e RS232).	REQ 003
REQ HW 204	O sistema deverá conter recursos de <i>hardware</i> suficientes para abstração de uma camada com um Sistema Operacional (S.O.).	REQ 003
REQ HW 205	O <i>hardware</i> deverá prever recursos de conversão de sinais analógicos em digitais para posterior tratamento.	REQ 001
REQ HW 206	O <i>hardware</i> deverá demandar baixa quantidade de potência elétrica, sendo inferior a um computador pessoal e suficiente para atuar por algumas horas alimentado por baterias, da mesma forma que deva ser protegido contra ações externas e da natureza (chuvas, ventos, umidade, etc.).	REQ 003

Fonte: Autoria própria

3.2.2 Requisitos de *software*

Como determinado pelo requisito 'REQ HW 204' o microcontrolador escolhido para desenvolvimento do protótipo deve possuir recursos que permitam a instalação de um Sistema Operacional. Tal requisito existe pelo fato de um SO no sistema facilitar o trabalho do desenvolvedor em muitas ocasiões, como construção de *drivers* para os periféricos e rotinas de controle de processos paralelos (*threads*).

Por este motivo, desenvolver todos os *drivers* e interfaces com certeza de funcionamento futuro é uma tarefa que não é prevista no cronograma do trabalho. Sendo assim, escolheu-se uma abordagem de utilizar *softwares* já prontos para implementar estas interações com o controlador principal.

O interfaceamento das aplicações e bibliotecas que serão desenvolvidas com o microcontrolador em questão será, portanto na sua grande maioria feito pelo

sistema operacional. Estas aplicações estão melhor organizadas na Tabela 6, que diz respeito as necessidades de *software* que o sistema demanda a fim de satisfazer todos os requisitos de sistema.

Tabela 6: Requisitos de *software*

ID	Descrição	Derivado do requisito
REQ SW 101	O <i>software</i> deverá filtrar do sinal sonoro ruídos, sons não desejados e faixas de frequências que não são interessantes ao trabalho.	REQ 005
REQ SW 102	O sistema deverá contar com uma forma de armazenamento de informações relevantes.	REQ 001
REQ SW 103	O sistema deverá gerar uma <i>template</i> de áudio de formato e valores interessantes para um algoritmo classificador, tais como médias estatísticas, comportamento e formas da distribuição dos dados.	REQ 005
REQ SW 104	O <i>software</i> deverá conter uma entrada/saída de dados através de <i>socket</i> de comunicação.	REQ 002
REQ SW 105	O sistema deverá conter interfaces de <i>software</i> para memórias <i>flash</i> (e. g.: <i>pen drive</i>).	REQ HW 202

Fonte:Autoria própria

3.2.3 Diagrama de requisitos

Unificando os requisitos gerais do sistema com os requisitos de *hardware* e *software*, formou-se uma hierarquia de requisitos. Esta relação é ilustrada na Figura 17 na forma do diagrama de requisitos do sistema.

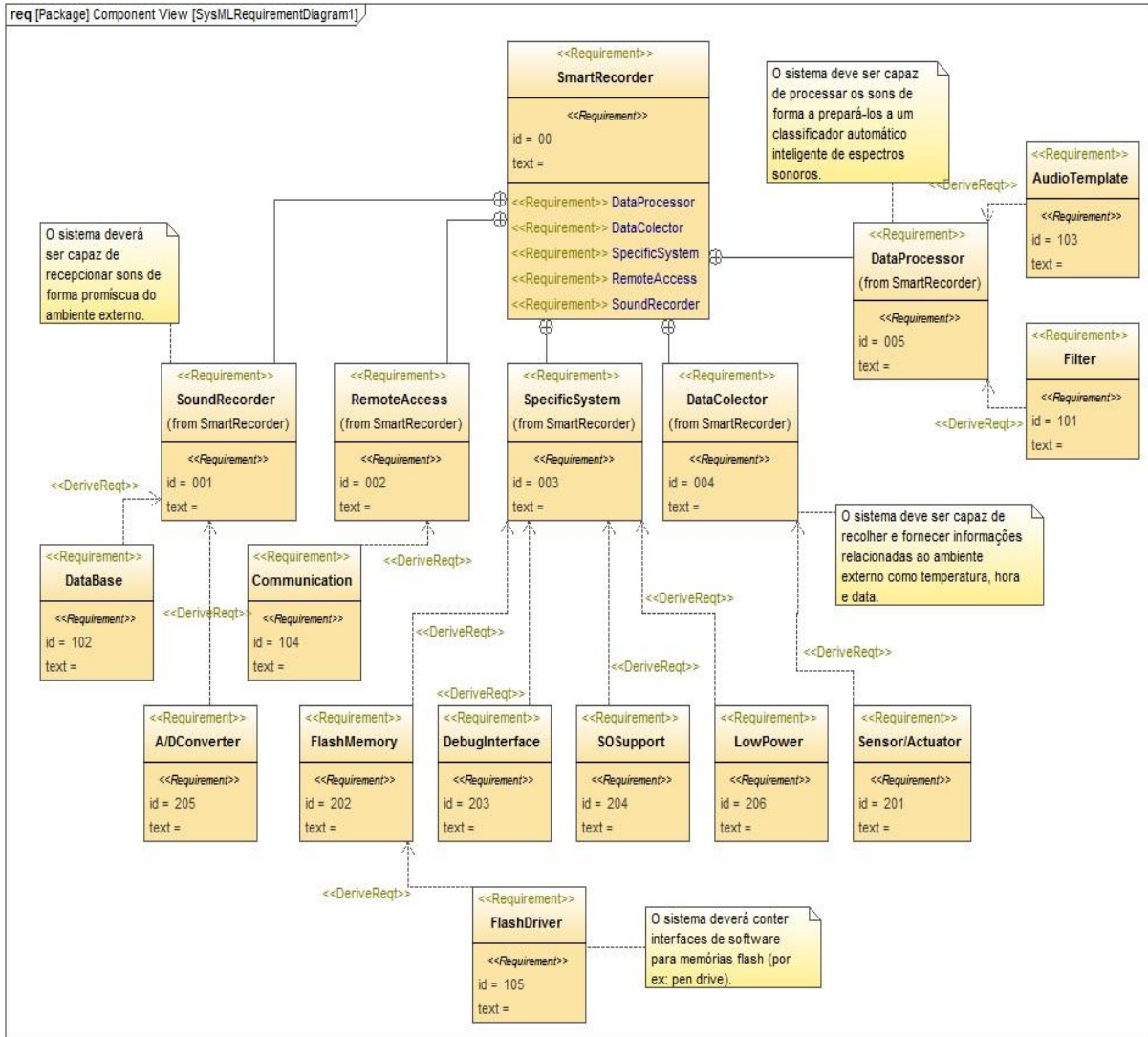


Figura 17: Diagrama de requisitos (Autoria própria)

3.2.4 Diagrama de Casos de Uso

Seguindo o processo de engenharia de requisitos, após a elicitaco de requisitos do sistema, existe a possibilidade de relacionar estes com casos de uso (Booch, Rumbauch e Jacobson, 2005). Estes so, assim como as tabelas de requisitos, teis para formalizar as necessidades do sistema e para parametrizar termos de custo e tempo ao projeto em aoes que envolvem trabalhos de engenharia.

As regras dos casos de uso, que resultar em um diagrama de casos de uso, so simples: Um caso de uso representa uma unidade discreta da interao entre um usurio (ator) seja este um humano ou uma mquina e o sistema. Um caso de uso pode "incluir" outra funcionalidade de caso de uso ou "estender" outro caso de

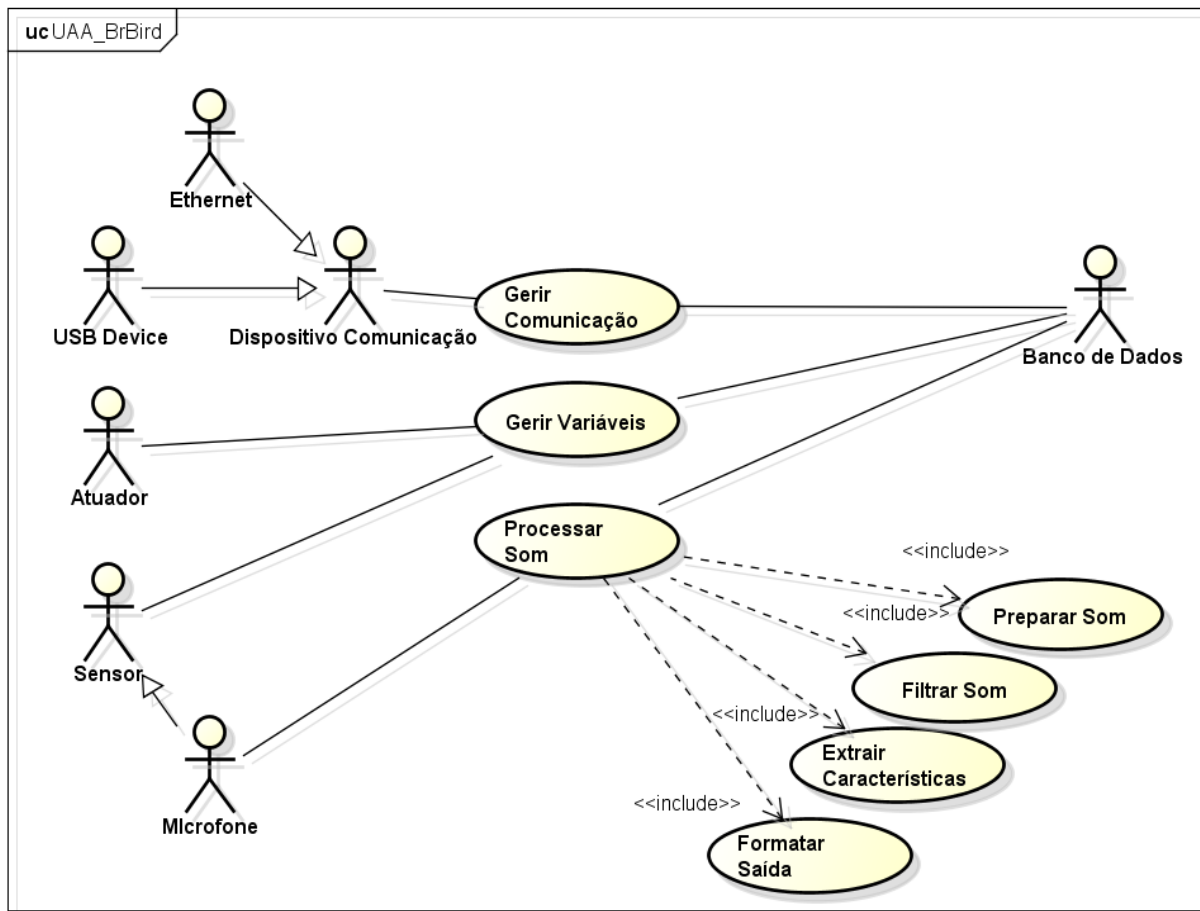
uso com seu próprio comportamento. Com base nesta definição e nos interesses do trabalho foi feita a relação da Tabela 7 a seguir.

Tabela 7: Casos de uso identificados

	Caso de Uso	Derivado do Requisito
01	Preparar Som	REQ SW 101 - O <i>software</i> deverá filtrar do sinal sonoro ruídos, sons não desejados e faixas de frequências que não são interessantes ao trabalho.
02	Gerir Variáveis	REQ 001 - O sistema deverá ser capaz de receber sons de forma promíscua e do ambiente externo; REQ HW 201 – Sensoriar variáveis do ambiente externo; REQ HW 201 - O sistema deve poder atuar e sensoriar sobre pontos e variáveis do ambiente externo; e REQ HW 201 – Atuar sobre pontos do ambiente externo
03	Formatar Saída	REQ SW 103 - O sistema deverá gerar uma <i>template</i> de áudio de formato e valores interessantes para um classificador computacional.
04	Gerir Comunicação	REQ SW 104 - O <i>software</i> deverá conter uma entrada/saída de dados através de <i>socket</i> de comunicação; e REQ SW 105 - O sistema deverá conter interfaces de <i>software</i> para memórias <i>flash</i> (por ex: <i>pen drive</i>).
05	Processar Som	REQ 005 - O sistema deve ser capaz de processar os sons de forma a prepará-los a um classificador automático de espectros sonoros.
06	Filtrar Som	Derivado de processar os sons (REQ 005).
07	Extrair Características	Derivado de processar os sons (REQ 005).

Fonte: Autoria própria

Na Figura 18 são ilustradas as interações entre os casos de uso do sistema com os atores do mesmo. Os atores foram classificados e dispostos no diagrama abaixo respeitando as interfaces e aspectos que são necessários para atender as funcionalidades do *software*.



powered by astah®

Figura 18: Diagrama de casos de uso (Autoria própria)

Como a maioria dos sistemas embarcados, o equipamento contará com sensores e atuadores, representados no diagrama como atores. Estes atores interagem com o *software* que trata as informações que vêm e vão no decorrer do funcionamento do equipamento (função do caso de uso “Gerir Variáveis”). É um critério da situação (e.g. som em potencial), armazenar ou não estas informações na memória não volátil do equipamento (banco de dados).

Toda comunicação externa com o sistema, além de originar-se dos sensores, pode chegar por um dos dispositivos de comunicação. O sistema possui um controlador de comunicação (caso de uso “Gerir Comunicação”) que é responsável por enviar ou receber informações.

Tem-se que todo som coletado passa por um processamento (caso de uso “Processar Som”) e que é submetido as seguintes etapas:

- Preparo do som: representado pelo caso de uso “Preparar Som” este preparo consiste em fases de digitalização e organização do sinal.

- Filtro sonoro: representado pelo caso de uso “Filtrar Som”, responsável por eliminar ruídos diversos do som.
- Extração de características: caso de uso “Extrair Características” que extrai características diversas do sinal, úteis na classificação automática do som.
- Formatação da saída: preparo da *templatedo* sinal, representado pelo caso de uso “Formatar Saída”.

Para melhor entendimento dos casos de uso gerados, o APÊNDICE A, detalha os passos e fornece a descrição dos Casos de Uso, bem como seus fluxos e observações.

3.3 PROJETO DO SISTEMA

Através do método apresentado e dos requisitos levantados, foi criada uma modelagem de sistema cuja finalidade é a realização de pré-processamento em sinais sonoros. Este modelo é apresentado no diagrama de blocos da Figura 19. O sistema é construído sobre uma rotina principal que tem como função controlar as *threads* do sistema. Estas, por sua vez, gerenciam eventos do sistema e ações com as demais partes, tais como o banco de dados e as comunicações externas.

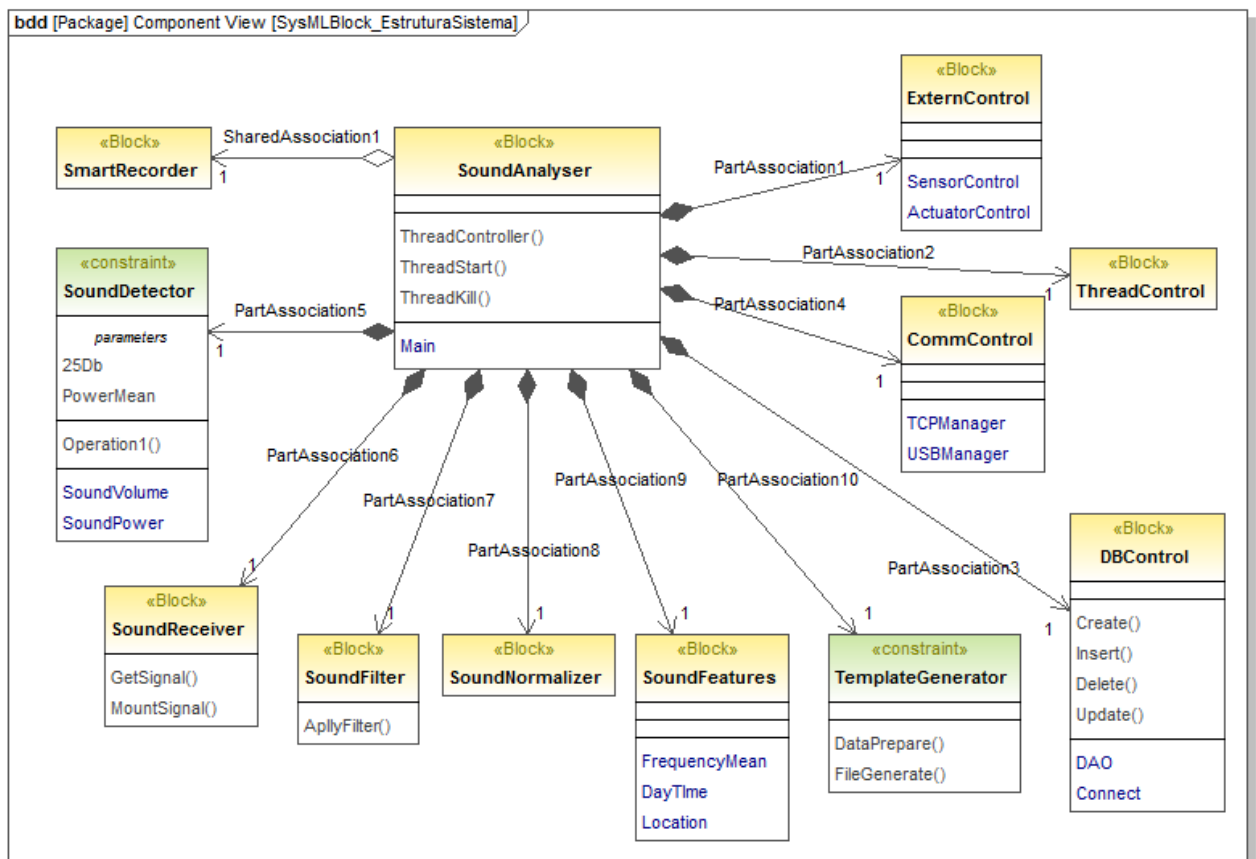


Figura 19: Diagrama de blocos (Autoria própria)

O desenvolvimento dos blocos para execução dos testes e verificação dos resultados esperados será concretizada em uma plataforma real, composta por um circuito eletrônico dotado de um microcontrolador programável. Tal desenvolvimento será feito utilizando programação Orientada a Objetos e conterá as classes ilustradas no Diagrama de Classes da Figura 20.

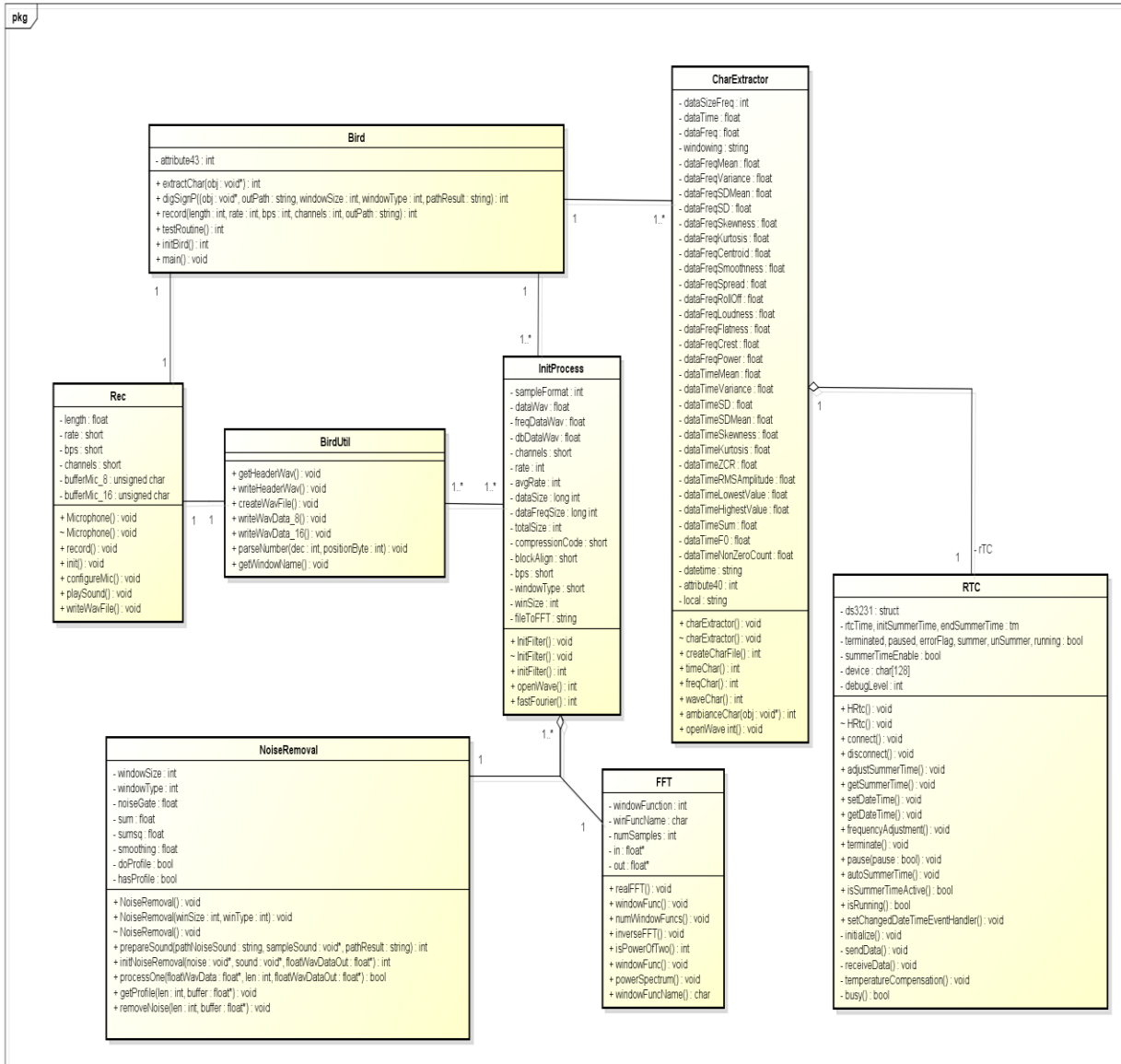


Figura 20: Diagrama de Classes (Autoria Própria)

Continuando o processo de modelagem e formalização do sistema a Figura 21 ilustra outro diagrama exclusivo da SysML. O diagrama paramétrico é um diagrama para expressar restrições e equações entre as propriedades e sua maior utilidade é criar uma integração entre a análise de engenharia e a modelagem (SysML 0.9, 2005).

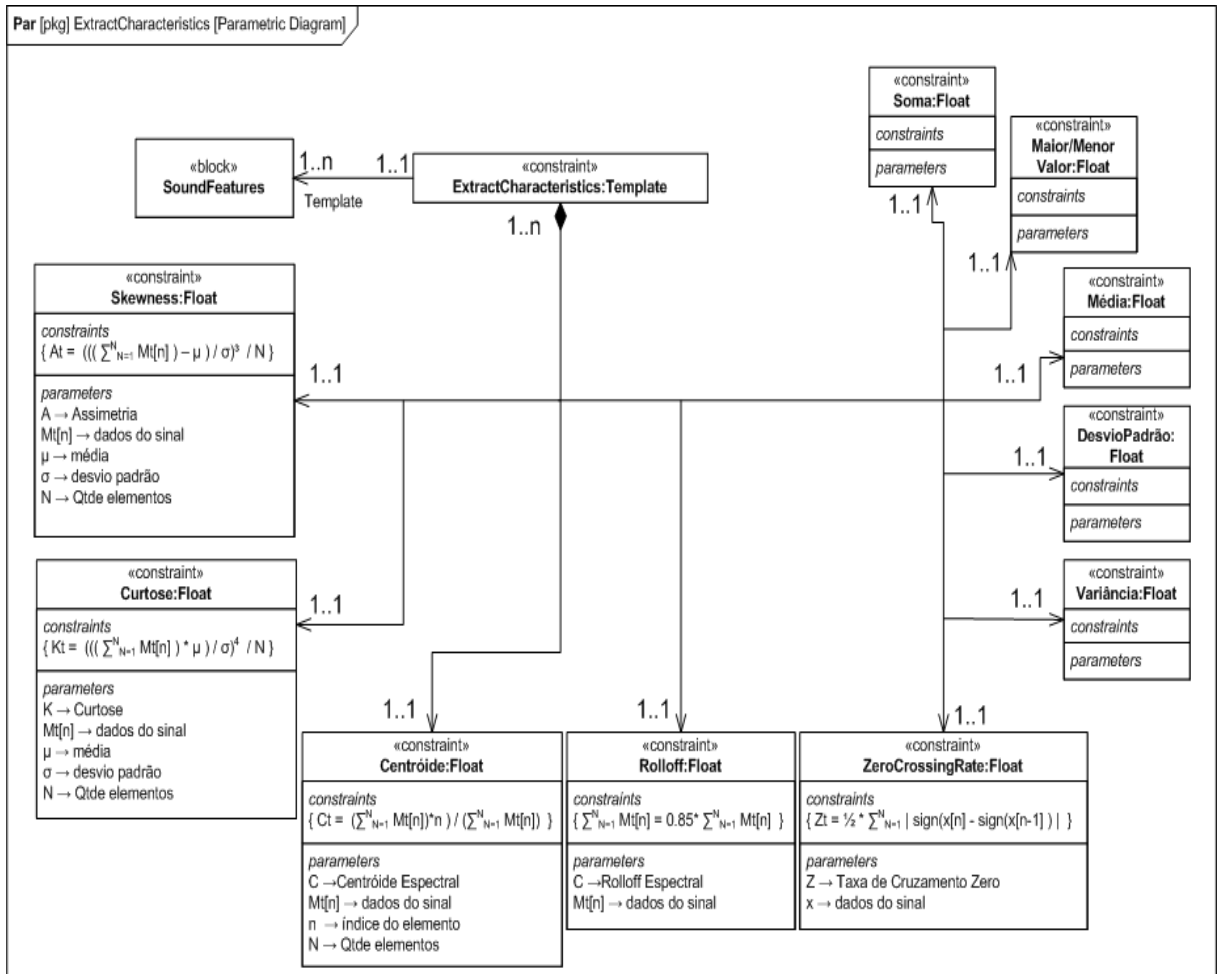


Figura 21: Diagrama paramétrico (Autoria própria)

As funções definidas no diagrama são reutilizadas em cada coleta realizada com o intuito de ditar os parâmetros de identificação do som gravado. As equações no diagrama paramétrico, na prática, têm a função de formar o gabarito (*template*) do som. As variáveis que compõe este gabarito, originadas das equações, são organizadas e relacionam-se umas com as outras como mostra a Figura 22.

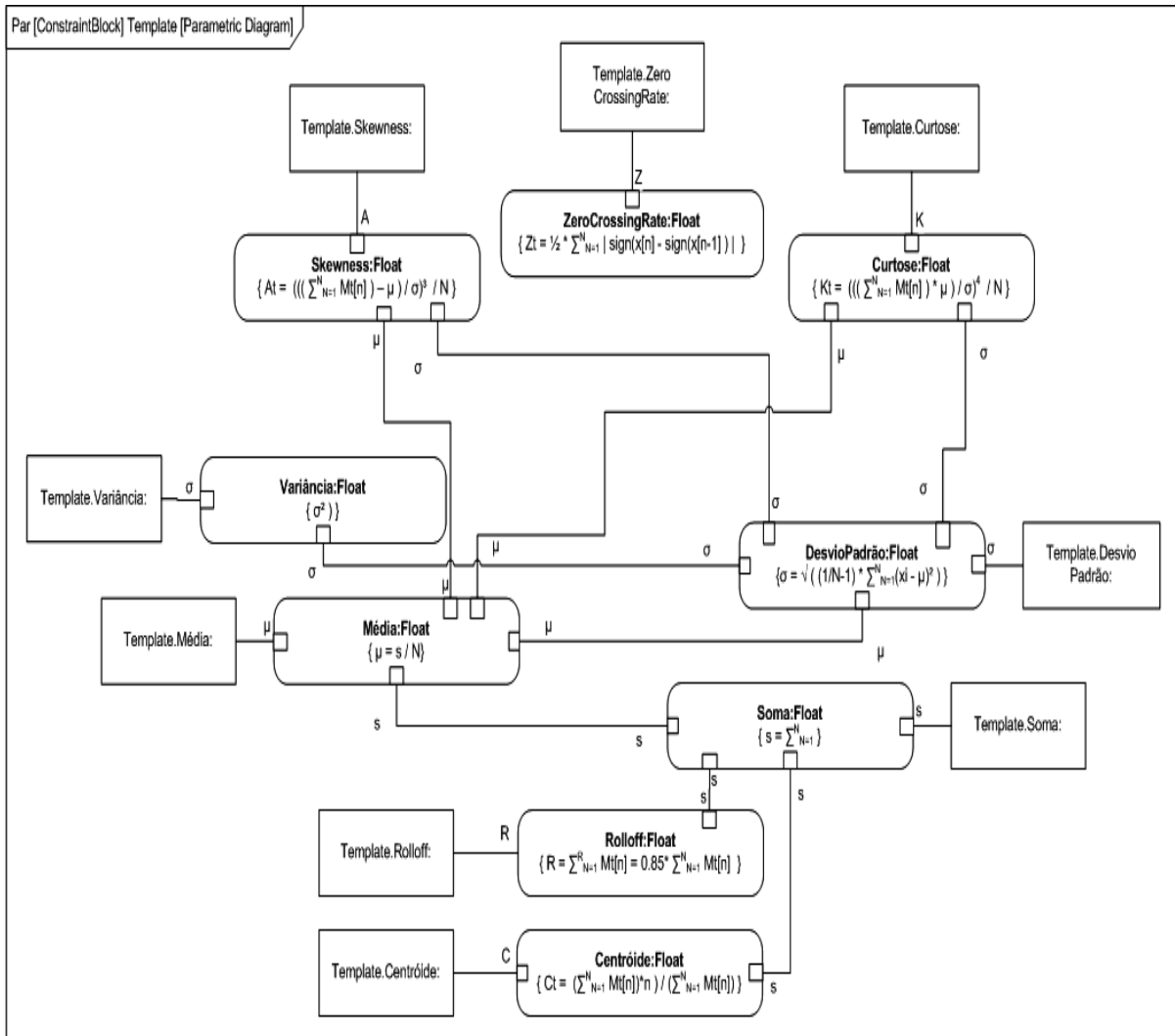


Figura 22: Valores das propriedades paramétricas (Autoria própria)

3.4 INTEGRAÇÃO

Integração em sistemas de engenharia para Weillkiens é a forma de interação de um sistema com o ambiente o qual ele está inserido Weillkiens (2006). O autor ainda exemplifica as interfaces de um sistema como sendo a integração do mesmo com determinado ambiente. Neste trabalho as interfaces do sistema com o ambiente o qual a UAA estará inserido são o *hardware* juntamente com seus periféricos.

Na divisão que se segue será mais bem explicado o controlador escolhido para o projeto, o motivo de sua escolha e como se adéqua aos requisitos levantados na seção anterior.

3.4.1 Controlador

Como arquitetura de *hardware* para desenvolvimento deste projeto, foi escolhido os controladores com núcleo *Acorn RISC Machine* (ARM). Alguns fatores

que foram analisados para realizar a escolha desta arquitetura foram enumerados na Tabela 5e, como é mostrado nos próximos parágrafos, estão diretamente ligados a características de microcontroladores de média ou alta complexidade.

Requisitos importantes como o suporte a um S.O. (REQ HW 204) e suporte a memórias *flash* (REQ HW 202) são importantes na hora da escolha da arquitetura. Respectivamente visando atender estes requisitos, foi escolhido um microcontrolador que possui memória virtual e suporte ao uso da tecnologia USB. Tais recursos facilitam o processo de desenvolvimento e de interação com o sistema depois de pronto.

Dito isso, é importante salientar que o REQ HW 204, que diz respeito à abstração de um sistema operacional, não é obrigatoriamente dependente do recurso de memória virtual, uma vez que existem SO's que não necessitam de tal recurso para operar em microcontroladores (Pereira, 2007). Além disso, o requisito também não diz respeito a um Sistema Operacional de Tempo Real (RTOS). Isso porque o sistema a ser modelado não é caracterizado por operações e tarefas que exijam agendamento e alta precisão de tempo. Escolheu-se no trabalho utilizar-se de um SO com ampla documentação disponível, gratuito e com várias facilidades já incorporadas no controlador escolhido, fazendo com que a memória virtual seja uma necessidade do sistema.

Sobre a satisfação deste e dos outros requisitos de *hardware*, analisou-se as famílias de controladores com núcleo ARM disponíveis no mercado. Uma delas se enquadra ao que é proposto e às restrições de projeto. A família ARM9 da série ARMv5T, possui núcleos já preparados para aplicações com PDS, diferente da sua antecessora, a série ARMv4T (Sloss, Symes e Wright, 2004).

O controlador escolhido para uso neste trabalho é fabricado pela *Samsung*, tem 289 pinos e seu modelo é o S3C2440A e seu núcleo é um ARM920T. Na Tabela 8 estão enumeradas características deste microcontrolador, bem como as vantagens e adequação do seu uso para aplicações quem envolvem PDS. No *datasheet* do controladorem questão (Samsung, 2004), são enumeradas várias funcionalidades que o mesmo oferece. Baseado nestas, foram separadas algumas que são pertinentes ao trabalho e também que são comprobatórias da satisfação dos requisitos.

Tabela 8: Características do microcontrolador

Característica do microcontrolador	Consideração	Requisito de Hardware Satisfeito
Interfaces de áudio <i>Integrated Interchip Sound</i> (IIS), AC97.	Coleta de sinais sonoros externos.	REQ HW 201
RTC (<i>Real Time Clock</i>) integrado.	Marcação de data e hora para coleta de informações adicionais relacionadas ao canto.	REQ HW 201
Suporte a memórias do tipo: -ROM (<i>Read Only Memory</i>) - NOR – (<i>not or</i>) para boot - NAND – (<i>not and</i>) para execução para <i>boot</i> e execução.	Requisito referente a suporte a memórias <i>flash</i> .	REQ HW 202
Memória <i>Cache</i> de dados e de programa.	Interessantes para o desempenho do SO.	REQ HW 204
Oito canais de A/D com dez <i>bits</i> de resolução e taxa de no máximo 500KSPS (Quinhentas mil amostragens por segundo).	Requisito que diz respeito à conversão de sinais analógicos em digitais para processamento adequado do som.	REQ HW 205
Dois canais SPI (<i>Serial Peripheral Interface</i>) até três seriais RS232.	Protocolo de comunicação do <i>SD Card</i> .	REQ HW 202
Instruções <i>RISC</i> de 16 (<i>thumbextension</i>) e 32 bits.	Controlador com baixo consumo de energia elétrica.	REQ HW 206
DMA (<i>DirectMemory Access</i>) quatro canais.	Ganhos em desempenho nas atividades com a memória do sistema, melhorando assim o desempenho.	REQ HW 204

Fonte:Autoria própria

Para fins de validação e desenvolvimento de um protótipo, será utilizado um *kit* de desenvolvimento para realizar os testes e validações do que se propõe. Tal escolha foi feita objetivando excluir do trabalho tarefas como projeto de placas de circuito impresso e projetos eletrônicos específicos.

3.5 DESENVOLVIMENTO

A fase de desenvolvimento diz respeito, segundo o modelo de Bahill e Gissing descrito no início do capítulo, à materialização do modelo especificado nas fases

anteriores (Bahill e Gissing, 1998). O *software*, responsável por realizar todo o trabalho de pré-processamento da UAA, é o resultado da modelagem apresentada pelo diagrama de casos de uso (Figura 18). Sendo assim as próximas divisões tratarão das principais partes deste *software*.

3.5.1 Rotinas Importantes

O *software* do microcontrolador (*firmware*), como já enfatizado durante o trabalho, não busca realizar efetivamente classificação de todos os pássaros existentes na natureza. Sua função é realizar pré-processamento das informações recolhidas em um ambiente propício a circulação das aves. Posteriormente, preparar tais informações para que possam ser utilizadas por um classificador automático que, possivelmente, estará alocado em um servidor especializado para esta prática.

A Tabela 9 organiza algumas funcionalidades, úteis para realização do pré-processamento, que foram incorporadas no *firmware* de forma a satisfazer os requisitos e objetivos desta pesquisa.

Tabela 9: Funções importantes *software*

Nome	Descrição	Resultado Esperado	Derivado do Requisito
Hora do Dia	Atribuir a amostra analisada uma hora do dia para fins de processamentos de classificação futuros.	Obter histórico detalhado das amostras.	REQ 004
Localização	Localização geográfica do aparelho monitor. Seja esta informada manualmente por um operador ou adquirida através de sinal de <i>Global Positioning System</i> (GPS).	Informação adicional da região do canto para contribuições ao método de classificação automática.	REQ 004
Data do ano	Extrair através do RTC na placa eletrônica uma data da amostra coletada.	Obter histórico detalhado das amostras e também informações importantes para regionalização e migrações dos pássaros	REQ 004
Redução de ruídos	Remover harmônicas indesejadas, ruídos oriundos do ambiente e borrões gerados pelos microfones unidirecionais.	Um espectro mais limpo e com visualização facilitada para os classificadores que utilizarão o espectro.	REQ SW 101

Nome	Descrição	Resultado Esperado	Derivado do Requisito
Escrita das informações	Gravação em memória não volátil das informações processadas	Fácil acesso aos dados analisados e permanência de informações no equipamento	REQ SW 102
Extração de características	Aplicar um extrator de características no domínio do tempo e no domínio da frequência.	Pré-classificar o som através das informações extraídas do som. Objetivando restringir espécies de pássaros possíveis para o som em questão.	REQ SW 103
Transferência via canais de comunicação	Auxiliar conexões externas com o ambiente embarcado.	Fácil interação com classificadores já existentes.	REQ SW 104
Transferência via dispositivos USB	Auxiliar conexões externas com o ambiente embarcado.	Prover formas de exportação a classificadores robustos externos à estação de gravação.	REQ SW 105

Fonte:Autoria própria

3.5.2 Filtros

Os filtros, materialização do REQ SW 101 e explicado no capítulo anterior são úteis para retirar do sinal ruídos indesejados e outros sons que não são pertinentes à situação.

Filtros redutores de ruídos, para aplicações envolvendo cantos de pássaros são úteis pelo fato do canto, muitas vezes misturar-se com o som das chuvas e do vento. Exemplificando esta situação, considerando conhecidas as formas de ruídos comuns ao lugar em que o sistema está inserido, pode-se armazenar em memória dados deste ruído de forma a aplicar o corte sobre estas faixas de valores. Desta forma o sinal analisado livra-se de alguns valores pré-determinados que não são interessantes para a análise.

Esta técnica, ilustrada na Figura 23, está presente nos *softwares* da área e utiliza-se de uma amostra ruidosa para realizar a redução dos ruídos. Para a criação dos espectros ilustrados na figura utilizou-se o *software* gratuito *Audacity 1.3*. A ação dos filtros é mostrada sobre uma amostra, retirado do site Xeno-Canto

(<http://www.xeno-canto.org/>) e corresponde a um pássaro do nordeste brasileiro chamado de *Crypturellusstrigulosus*(Inhambu-relógio).

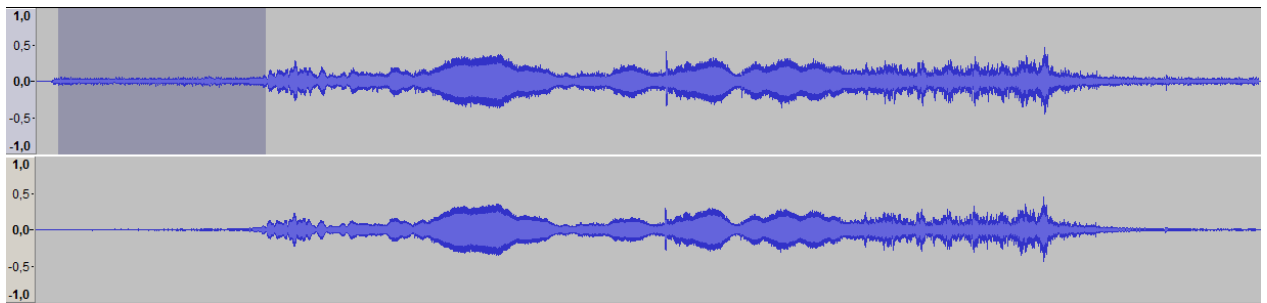


Figura 23: Efeito dos filtros no som (Autoria própria)

3.5.3 Templates de Áudio

O passo final do pré-processamento seria a identificação de um conjunto de informações oriundas do sinal que passou pelos filtros iniciais e outras formas de tratamento do sinal. Este conjunto de informações, sumarizado e organizado em uma *templatesão* valores como médias e características do sinal no domínio da frequência. O intuito desta *template* é fornecer ao classificador dados sobre o tipo e comportamento da amostra, auxiliando assim no processo de classificação.

Com um classificador pronto em uma estação remota ou servidor de processamento, este sistema pode receber apenas os dados da *template* e responder em pouco tempo se o som analisado é realmente relevante. Nesse caso a UAA pode continuar a gravar os sons oriundos do ambiente.

A técnica de *templates* é utilizada largamente em aplicações embarcadas que fazem processamento de sinal, justamente por muitas vezes existir espaço limitado em memória para se guardar toda a informação. Não seria prático, por exemplo, guardar e transferir muitas horas de som gravadas no equipamento a um servidor remoto. O desenvolvimento das rotinas de obtenção da *template* seguirá o diagrama da Figura 21

3.6 CONCLUSÃO E CONSIDERAÇÕES

A partir das de problemáticas encontradas nas técnicas de coleta dos cantos de pássaros e nos métodos científicos de processamento sonoro, esta pesquisa focou-se nas dificuldades dos pesquisadores em realizar a classificação de pássaros por seu canto em ambiente reais. Com esta abordagem, foi utilizada uma linguagem de representação de sistemas, a *SysML*, para modelar uma solução para os

problemas apresentados. O estudo iniciou-se aplicando as fases iniciais da engenharia de sistemas que dizem respeito à pré-modelagem, estudo de requisitos e parametrização do sistema.

A partir de necessidades da área, um diagrama de requisitos foi modelado (Figura 17) de forma a criar um modelo de ambiente capaz de satisfazer tais necessidades. Para isso, desmembraram-se de requisitos principais do sistema, os requisitos de *software* e *hardware*. A abordagem foi escolhida para mostrar dificuldades e caminhos críticos do desenvolvimento do protótipo de testes e para preparar os métodos de cálculo de custo e tempo da construção do projeto.

Por fim, somaram-se os motivos de escassez de aplicações adequadas e foram apresentadas as noções de viabilidade técnico-científicas necessárias para solução do problema.

MATERIALIZAÇÃO E TESTES DA UAA

O próximo passo do processo de desenvolvimento será apresentar a materialização do modelo e a criação de um plano de testes baseados nos dados da Engenharia de Sistemas (Capítulo 3). Este capítulo, portanto, tem como objetivo apresentar este desenvolvimento e os testes realizados que buscam comprovar a eficiência da UAA.

Sendo assim, na seção 4.1 será descrita a configuração da plataforma construída para realização dos testes, seguindo, na seção 4.2, da sua materialização.

As subseções que compõem a seção 4.2 (*hardware* e *software*) dizem respeito aos detalhes de implementação e dos testes levantados para esta pesquisa. São apresentados ainda os estudos detalhados das partes do sistema, os parâmetros de configuração e os procedimentos de testes.

4.1 CONFIGURAÇÃO DO SISTEMA

No trabalho foi utilizada uma versão do Linux com *kernel* 2.6.32.2 facilmente adaptável a várias aplicações finais. Com um *menu* e arquivos de configuração facilitados, a versão do Linux pode ser instalada nos controladores com núcleo ARM9 através da *cross* compilação¹ do *kernel*.

Posterior a isso foi feita a criação de uma imagem, que por fim foi inserida no microcontrolador através de uma sincronização entre o ambiente de desenvolvimento e o *bootloader* presente na memória NOR ligada ao controlador.

Utilizou-se a versão gratuita do *bootloader* *SuperVivi* devido a ampla documentação e capacidade de integração com o SO proposto. O *software* principal do sistema será o próprio Linux, sendo este o responsável por todas as interações com o *hardware* e também a camada de abstração entre a aplicação final e o microcontrolador.

A versão citada do sistema operacional é considerada muito parecida com os terminais convencionais do Linux. As diferenças principais são, dentre outras, no

¹ Compilação realizada em local diferente do lugar que o *software* irá executar (Zhang, Lin 2010).

sentido que entrada e saída de padrão de dados, que nos computadores de propósito geral são o teclado e o monitor respectivamente, nos sistemas embarcados com sistema operacional podem ser feitas através da comunicação serial RS232 ou interfaces de *debug*.

Comandos úteis como o *mount* e *umount* para manipulação de dispositivos USB, *sed*, *awk*, *cp*, *mv*, comuns ao Linux, foram *cross* compilados e inseridos no *kernel*. Estes e muitos outros comandos estão presentes nas versões embarcadas do *kernel*, bem como serviços como *telnet* e FTP (*File Transfer Protocol*) auxiliam no monitoramento gerenciamento remoto do *hardware*.

O compilador utilizado para realizar a *cross* compilação do *kernel* e dos *softwares* que atuam para satisfazer este trabalho, foi o compilador Linux ARM GCC 4.3.2.

4.2 UNIDADE DE AQUISIÇÃO AUTÔNOMA

As próximas subseções dizem respeito à materialização da modelagem do capítulo anterior, segundo o modelo de Bahill e Gissing, a implementação da UAA.

Inicia-se com a explanação do *hardware* do sistema, mostrando os circuitos de potências desenvolvidos bem como os testes a serem feitos para satisfação dos requisitos de *hardware*.

Na sequência, a subseção 4.2.2 mostra os testes realizados no *software* embarcado da UAA. Aqui são descritos os algoritmos utilizados, como se comportam na análise sonora e outras informações pertinentes.

4.2.1 *Hardware*

Uma plataforma pronta de desenvolvimento será utilizada nos testes da pesquisa. Com intuito de economia em processos de engenharia para a confecção de um *hardware* teste, será utilizado um *kit* de desenvolvimento com um microcontrolador fabricado pela sul-coreana *Samsung*, com núcleo ARM. O controlador (S3C2440A ARM920T) comunica-se com uma memória FLASH NAND de 256Mb para armazenamento de informações e outra NOR de 2Mb para o *boot* do controlador.

A plataforma de testes é mostrada na Figura 24. Um microfone já incorporado ao circuito será utilizado para a coleta de amostras do ambiente nos testes. O

microfone comunica-se com o microcontrolador através de um *chip* CODEC (Codificador/Decodificador) chamado UDA1341TS.

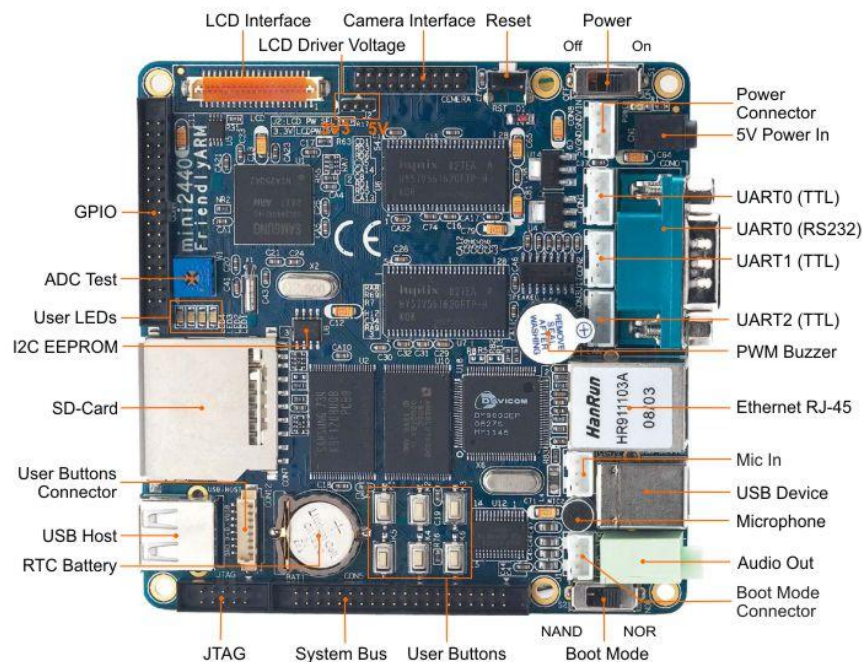


Figura 24: Plataforma de testes (friendlyarm.net)

Com isso alguns parâmetros de configuração do microfone no microcontrolador podem ser definidos:

- O tipo de microfone será apenas de um canal (mono);
- Cada amostra será armazenada em memória com tamanho 8 ou 16 bits, o que permite que o som seja representado, respectivamente, por 256 ou 65536 gradações diferentes.
- O valor de amostragem do som será de até 44100 amostras por segundo. O valor foi escolhido pelo motivo que amostragens até menores que estas são consideradas aceitáveis em gravações. Uma amostragem de 8000, por exemplo, com 16 *bits* já pode ser possuir uma “Qualidade de discurso” (VORAN, 2007). Para fins de comparação, um CD com capacidade de reprodução de um som quase perfeito, utiliza uma taxa de amostragem de 44100 amostras por segundo de gravação (AUDACITY Manual).

4.2.1.1 Circuito potência

A fim de aumentar a disponibilidade energética foi adicionado um circuito de alimentação para a UAA (Figura 25). Projetou-se de forma que duas fontes de energia paralelas e redundantes possam ser utilizadas no sistema principal. O

circuito é dotado de um CI regulador de tensão chamado LM317 que ajusta para 5V (Volt) a saída para a UAA.

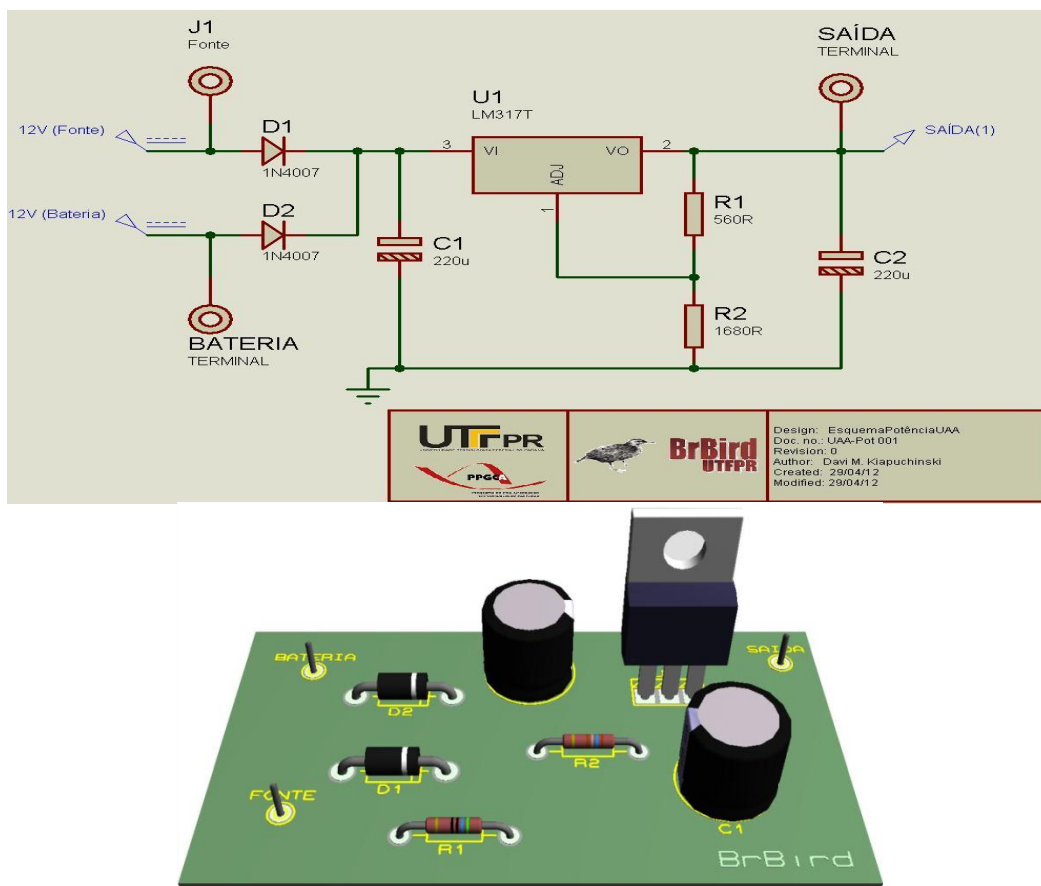


Figura 25: Esquemático e layout do circuito de alimentação da UAA (Autoria própria)

4.2.2 Software

Utilizando as configurações da UAA definidas no *hardware*, o plano de testes do *software* visa apresentar os principais testes a serem feitos sobre o sistema modelado.

Tabela 10: Plano de testes

	Teste	Análise/Propósito
1	Testes de Coleta e Gravação	Teste funcional (caixa preta) de coleta e utilização de um conjunto de bytes referentes a uma gravação sonora.
2	Teste de montagem Arquivo música '.wav'	Teste para montagem, parametrização arquivo de música convencional.
3	Testes das funções de janelamentos	Medir eficiência nas variações do tamanho e tipo de janelamento utilizado a fim de comparar os diferentes tipos de funções de janelamento implementadas.
3	Teste geração de espectrograma	Amostragem adequada, distribuição, picos, altura condizentes para um gravador de sons em ambientes externos.

Teste		Análise/Propósito
4	Teste de eficiência	Medir eficiência dos algoritmos de <i>FFT</i> e dos filtros implementados.
5	Teste de extração de Características	Espera-se que o som captado possa fornecer várias características importantes para os classificadores de sons.
6	Testes amostras com ruídos	Teste para verificar a eficiência de filtros eliminadores de ruídos e manipulações no som. A aceitação ou não se dará por comparação visual da representação gráfica do som(Figura 26) antes e depois das operações.

Fonte:Autoria própria

Um exemplo de validação dos testes será a comparação gráfica de espectros (Figura 26), assim como a feita pela pesquisa de Agranat (2009). O *software* utilizado nas comparações gráficas será o *Audacity 1.3*

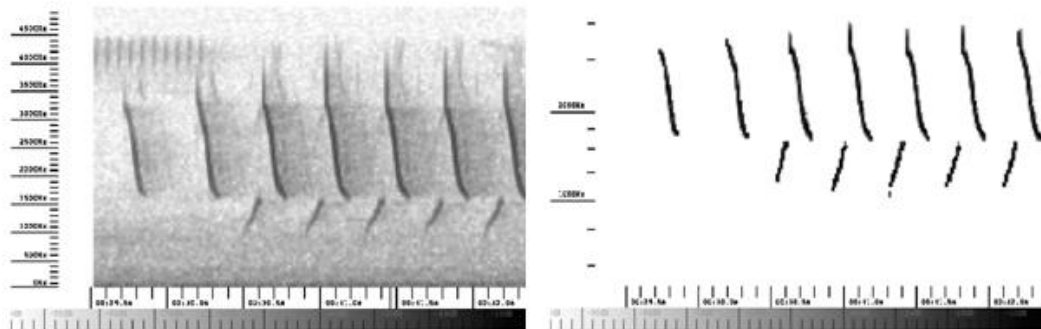


Figura 26: Comparações gráficas entre espectros (Agranat, 2009)

O *software* foi dividido, como mostrado no diagrama de blocos do sistema (Figura 19), em algumas partes principais que compõem a UAA. Na sequência, dentro destas partes serão explicados os esforços realizados para materializar estas funções. Adicionalmente serão demonstrados os testes realizados na função de forma a contemplar o plano de testes apresentado na Tabela 10 e também cumprir com os objetivos da UAA de forma a validá-la, neste trabalho, como plataforma de coleta e pré-processamento sonoro.

4.2.2.1 *SoundReceiver*

SoundReceiver (receptor de som) é a função de recepção do som, rotina esta que recebe o sinal sonoro oriundo do microfone e o disponibiliza para ser processado. Tal função disponibiliza o sinal sonoro de duas formas: através de um arquivo WAV e através de um *array* de *bytes* relativos ao som. Eis os detalhes da função materializada:

- O *array* obrigatoriamente deve ser do tipo *unsigned char* para sons com palavra (*word*) de 8 (oito) *bits* do tipo *short* para sons com palavra de 16 (dezesesseis) *bits*.
- O tamanho do arquivo WAV segue o padrão de tamanho mostrado na Eq. 10. Sendo que: o cabeçalho possui tamanho fixo para qualquer arquivo WAV (44 bytes); BPS significa *Bytes per Sample* (*Bytes por amostra*); tempo de gravação é o tempo em segundos que o som possui; e canais é o número de canais utilizado na gravação (1 canal – mono e 2 canais estéreo).

$$Tamanho = CABEÇALHO + \left(\frac{BPS \times TEMPOGRAVAÇÃO \times CANAIS}{8} \right) \quad \text{Eq. 10}$$

Os testes desta função visam abranger os testes um e dois do plano de testes (Tabela 10) e são detalhados respectivamente na Tabela 11 e na Tabela 12.

Tabela 11: Teste 1 – Coleta e gravação

Teste 1 – Coleta e gravação	
Finalidade	Comprovar funcionalidade de gravar e coletar sons externos no equipamento eletrônico (UAA).
Técnica	(a) coleta de som do ambiente pelo microfone da placa eletrônica; (b) a permutação de valores de amostragem, tempo de gravação e <i>bytes</i> por amostra; e (c) manipulações de escrita e leitura em memória.
Parâmetros	(a) Bytes por amostra; (b) Tempo de gravação; (c) Número de canais para gravação; e (d) Taxa de Amostragem: 8000 Hz, 16000 Hz, 22050 Hz ou 44100 Hz.
Resultado	Vetor dos dados representando o sinal sonoro armazenado em memória;

Fonte: Autoria própria

Tabela 12: Teste 2 – Montagem arquivo WAV

Teste 2 – Montagem arquivo WAV	
Finalidade	Garantir permanência dos dados no equipamento para posterior análise externa dos sons (análise de qualidade do som e eficiência da gravação).
Técnica	(a) Montagem do cabeçalho; (b) inserção das palavras (<i>words</i>) no formato <i>little-endian</i> ; e (c) manipulações de escrita e leitura em arquivo.
Parâmetros	(a) Nome e caminho do arquivo no sistema de arquivos e dados do som.
Resultado	Arquivo WAV em memória no equipamento possuindo características análogas ao som gravado.

Fonte: Autoria própria

As coletas e gravações dos sons serão sumarizados no próximo capítulo em uma tabela com permutações de valores de tamanho e tipo de janela, tempo de gravação e amostragem. Tal tabela será construída com intuito de analisar questões de eficiência dos algoritmos utilizados na UAA.

4.2.2.2 Funções de janelamento

Para materializar as funções de janelamento, foram escolhidas, dentre várias, três funções de janelamentocomumente utilizadas para sons em geral (Champagne e Labeau, 2004). A primeira delas,função de *Hamming* (Eq. 11), é muito parecida com a janela de *Hanning* (Eq. 12) (ver Figura 9 no CAPÍTULO 2), sendo ambas úteis em aplicações gerais de processamento digital de sinais. A terceira função escolhida é conhecida por janelamento triangular ou janela de *Bartlett* (Eq. 13).

$$w(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N-1}\right) \quad \text{Eq. 11}$$

$$w(n) = 0,5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \quad \text{Eq. 12}$$

$$w(n) = 1 - \left|\frac{2n}{N-1} - 1\right| \quad \text{Eq. 13}$$

Utilizando um canto coletado do site xeno-canto, de um pássaro sul-americano chamado Caneleiro-bordado (*Pachyramphusmarginatus*) a Figura 27 ilustra quatro funções de janelamento de tamanho 128 ($N = 2^7$). Os dois primeiros gráficos no topo da figura ilustram o janelamento de *Hamming* e o de *Hanning* respectivamente. A terceira figura (2º linha) ilustra o janelamento triangular de *Bartlett* e por último é mostrada a ausência de janelamento (retangular).

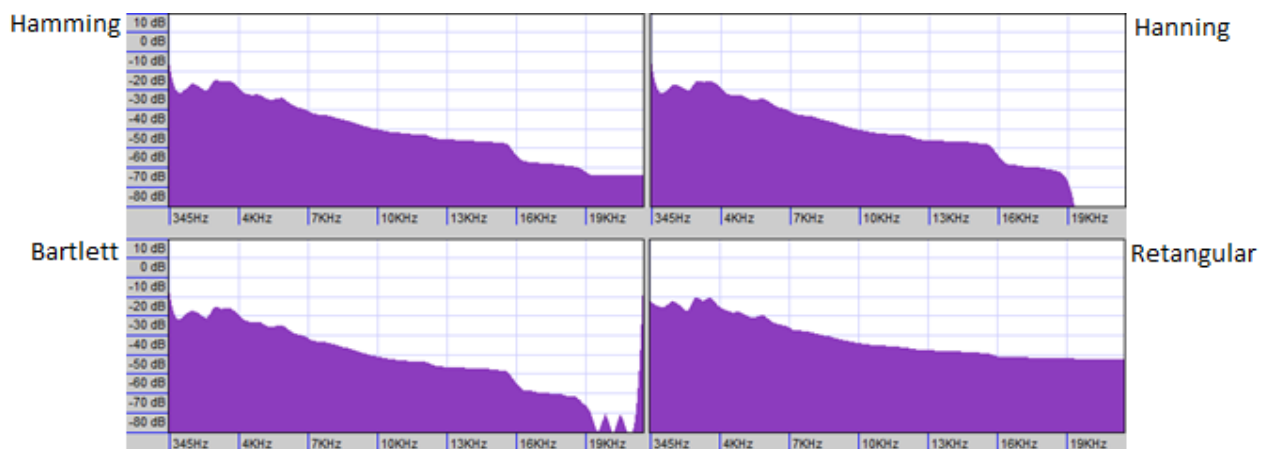


Figura 27: Comparação entre funções de janelamento (Autoria própria)

Realizou-se um teste para as funções de janelamento e este é explicado na Tabela 13.

Tabela 13: Teste 3 - Função de janelamento

Teste 3 – Função de janelamento	
Finalidade	Comprovar o funcionamento da Transformada Rápida de Fourier e encontrar relações do tipo/tamanho da janela com os tempos de execução dos filtros da UAA e da <i>FFT</i> do sinal analisado.
Técnica	Variar o tipo e tamanho da janela (sempre em potências de dois).
Parâmetros	(a) Tamanho de janela: 256, 512, 1024, 2048, 4096, 8192, 16384 ou 32768; e (b) Tipo de Janela: <i>Hamming</i> , <i>Hanning</i> ou <i>Bartlett</i> .
Resultado	Tabela com os tempos de processamento das rotinas de <i>FFT</i> e dos filtros de acordo com o janelamento utilizado (Figura 14).

Fonte: Autoria própria

4.2.2.3 Transformada Rápida de Fourier

Utilizando um algoritmo de *FFT* baseado na implementação feita por Dan Cross (CROSS, 1988), tal operação é responsável por transformar um sinal que esteja no domínio do tempo ao domínio da frequência. Cross em seu algoritmo utiliza a técnica do diagrama de *Butterfly* (análogo ao algoritmo de Cooley-Tukey base dois da Figura 8), que divide uma longa transformada de *fourier* em diversas transformadas menores.

A entrada da função materializada exige do sinal sonoro duas peculiaridades. A primeira é que o dado coletado deve estar normalizado em uma escala de -1 a +1, sendo necessária a utilização de uma equação de conversão (Eq. 14). A segunda é que o sinal deve ser processado em 'janelas' que sejam potência de 2 (dois).

$$Num_{Float} = BYTE * \frac{1}{32768} \quad \text{Eq. 14}$$

O teste realizado para a função de transformada rápida de Fourier é mostrado na Tabela 14 e tem seus resultados mostrados juntamente com a função de filtros da UAA.

Tabela 14: Teste 4 - Espectrograma

Teste 4 – Espectrograma	
Finalidade	Possuir os dados do som no domínio da frequência para a manipulação.
Técnica	Variar o tipo e tamanho da janela (sempre em potências de dois) e manipular o sinal de saída de forma a obter o resultado desejado.

Teste 4 – Espectrograma	
Parâmetros	(a) Gravações curtas com menos de cinco segundos; (b) Amostras de ruídos com um e meio segundo; (c) Frequência de Amostragem: 8000 Hz, 16000 Hz, 22050 Hz ou 44100 Hz; (d) Tamanho da janela: 256, 512, 1024, 2048, 4096, 8192, 16384 ou 32768 amostras.
Resultado	(a) Vetor de frequências da <i>FFT</i> representando o sinal sonoro; (b) Dados novamente no domínio do tempo quando submetidos à transformada inversa de <i>Fourier</i> ; e (c) Vetor em níveis de decibéis (dB).

Fonte: Autoria própria

Nos resultados, em (a) o vetor é utilizado para grande parte dos processamentos sonoros (filtros), em (b) a saída é utilizada quando se deseja montar, a partir dos dados resultantes da *FFT*, um novo arquivo WAV e em (c) os níveis são utilizados para demonstrar o espectrograma em níveis de energia sonora e com maior detalhamento gráfico.

A Tabela 15 organiza os tempos de processamento para uma amostra de 3.0 segundos. Alternando a frequência de amostragem (teste quatro do plano de testes) e o tipo de janelamento utilizado nas operações chegou-se aos tempos de processamento da transformada rápida de *Fouriere* da rotina de remoção de ruídos.

Tabela 15: Médias de tempo de execução na UAA em segundos

Freq.Amostragem (Hz)	Função de Janelamento	FFT
8000	<i>Hanning</i>	0,4400
8000	<i>Hamming</i>	0,3500
8000	<i>Bartlett</i>	0,2820
16000	<i>Hanning</i>	0,8100
16000	<i>Hamming</i>	0,6980
16000	<i>Bartlett</i>	0,5760
22050	<i>Hanning</i>	1,0100
22050	<i>Hamming</i>	0,9080
22050	<i>Bartlett</i>	0,8400
44100	<i>Hanning</i>	1,9380
44100	<i>Hamming</i>	1,7540
44100	<i>Bartlett</i>	1,3258

Fonte: Autoria própria

4.2.2.4 *SoundFeatures*

SoundFeatures (características sonora) é o bloco com a finalidade de identificar o sinal sonoro pelas suas características. Na Tabela 1, do Capítulo 2, explanou-se como tais características são importantes no pré-processamento de um sinal sonoro de forma a ajudar na sua contextualização e identificação perante um classificador automático.

Como auxílio para implementações e definições das funções utilizou-se os códigos-fonte das bibliotecas gratuitas: *Audacity* 1.3, *jAudio* 1.0 e *LibXtract* 0.4.5.

A Tabela 16 detalha o teste realizado para validar esta função. Sendo que na descrição do módulo “*TemplateGenerator*” serão mostradas as características escolhidas, bem como o resultado de um teste realizado utilizando-se os algoritmos implementados na UAA.

Tabela 16: Teste 5 – Extração características

Teste 5 – Extração características	
Finalidade	Testar extração das análises estatísticas realizadas sobre o sinal sonoro.
Técnica	Aplicar as propriedades paramétricas representadas na Figura 21.
Parâmetros	Valores do sinal no domínio do tempo e no domínio da frequência aplicados as equações das propriedades paramétricas.
Resultado	Sinal sonoro representado por diversos parâmetros podendo estes representá-lo perante a um classificador automático.

Fonte:Autoria própria

4.2.2.5 *SoundFilter*

SoundFilter(filtro sonoro) é o bloco responsável por eliminar ruídos não desejados no sinal sonoro. Aplicou-se uma função que se baseia em uma amostra ruidosa para encontrar padrões ruidosos no som. A técnica é chamada de “Portão de Ruído Espectral”ou “*SpectralNoise Gate*” a qual funciona analisando primeiramente o perfil ruidoso ou assinatura acústica do ruído previamente separado. Uma vez com este ruído, seja ele gravado pela UAA ou fornecido como padrão do ambiente, as operações de atenuação são executadas no espectro sonoro (Hodgson, 2010) (Agranat, 2009).

Tal função segue as etapas a seguir e resulta em uma amostra como a ilustrada na Figura 23.

- 1- Definição de um perfil ou assinatura acústica em todas as janelas (blocos) do ruído analisado.

- a. *FFT* na janela em análise, o resultado é um vetor de frequências com o tamanho da janela;
- b. Para cada janela calcula-se o logaritmo das posições do vetor de saída. Se o resultado for um número positivo então:
 - i. Somam-se os logaritmos de cada posição do vetor de saída;
 - ii. Adiciona-se um ao contador de perfis.
- c. Ao final de todos os janelamentos calcula-se a média de cada janelamento, de forma que todos tenham um perfil de ruído próprio. Para isso divide-se a soma dos logaritmos da janela pela sua quantidade de perfis.

2- Remoção de ruídos no som gravado do ambiente utilizando uma função de 'suavização' sonora.

- a. *FFT* e cálculo das potências, em níveis em decibéis (dB), de cada janela no sinal sonoro;
- b. Para cada janela calcula-se o logaritmo das potências do sinal;
- c. Se o resultado for menor que o perfil do ruído para a janela em questão então se atribui zero ao fator de alisamento, caso contrário, um (1.0). As operações são feitas de forma a cada janela possuir seus fatores de suavização.
- d. Tais fatores são comparados com seus valores anteriores e posteriores podendo tornar-se zero.
- e. Multiplicam-se os valores da *FFT* pelo fator encontrado e o sinal será ou não suavizado.

3- Calcula-se a transformada inversa de *Fourier* no espectro suavizado de forma a gerar um novo arquivo WAV.

Os testes realizados nesta função são detalhados na Tabela 17 e fazem referência ao teste cinco do plano de testes (Tabela 10).

Tabela 17: Teste 6 – Remoção de ruídos

Teste 6 – Remoção de ruídos	
Finalidade	Validar a rotina como um filtro redutor de ruídos.
Técnica	(a) Execução dos procedimentos descritos para a execução da função; (b) Gravação em memória dos dados sonoros; e (b) Comparações e análise visual no espectrograma sonoro.

Teste 6 – Remoção de ruídos	
Parâmetros	Dados resultantes das funções de janelamento e de transformada rápida de <i>Fourier</i> .
Resultado	Demonstração gráfica das mudanças no espectro sonoro submetido a rotina de filtros e auxílio na determinação de uma parametrização eficiente do filtro.

Fonte:Autoria própria

Os resultados dos testes realizados na função de filtro dão origem ao espectro sonoro da Figura 28.

A Figura 28 (a) ilustra o espectro de uma amostra de um segundo, enquanto que em (b) é mostrado o som a ser filtrado com três segundos de duração. Ambos os sons foram coletados pela UAA em um ambiente com ruídos de ventos e utilizando uma gravação de um assovio humano (b). Ambos os sons também possuem frequência de amostragem de 22050 Hz e 16 bits no tamanho de uma amostra (BPS). Em (c) é mostrado o resultado da aplicação do filtro sobre o sinal analisado, observando que grande parte do “borrado” encontrado no espectro original foi eliminado.

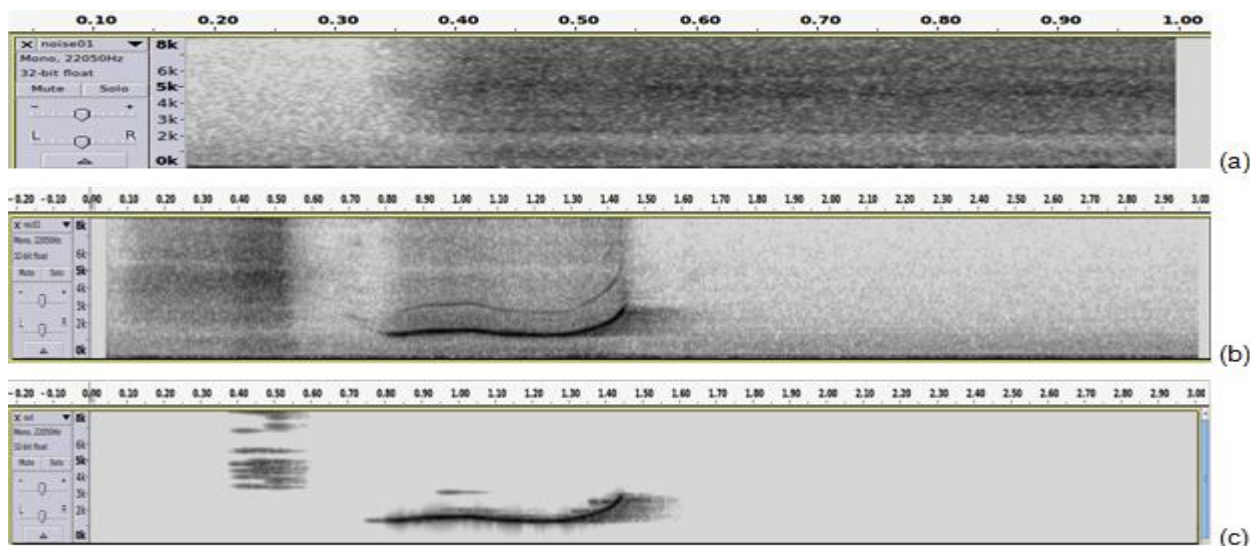


Figura 28: Remoção ruídos mostrada no domínio do frequência (Autoria própria)

A Figura 29 ilustra o mesmo som sob uma ótica temporal. O perfil do ruído (a) encontrado pelo filtro “suavizador”, aplicado na amostra em (b) resulta no sinal sonoro mostrado em (c).

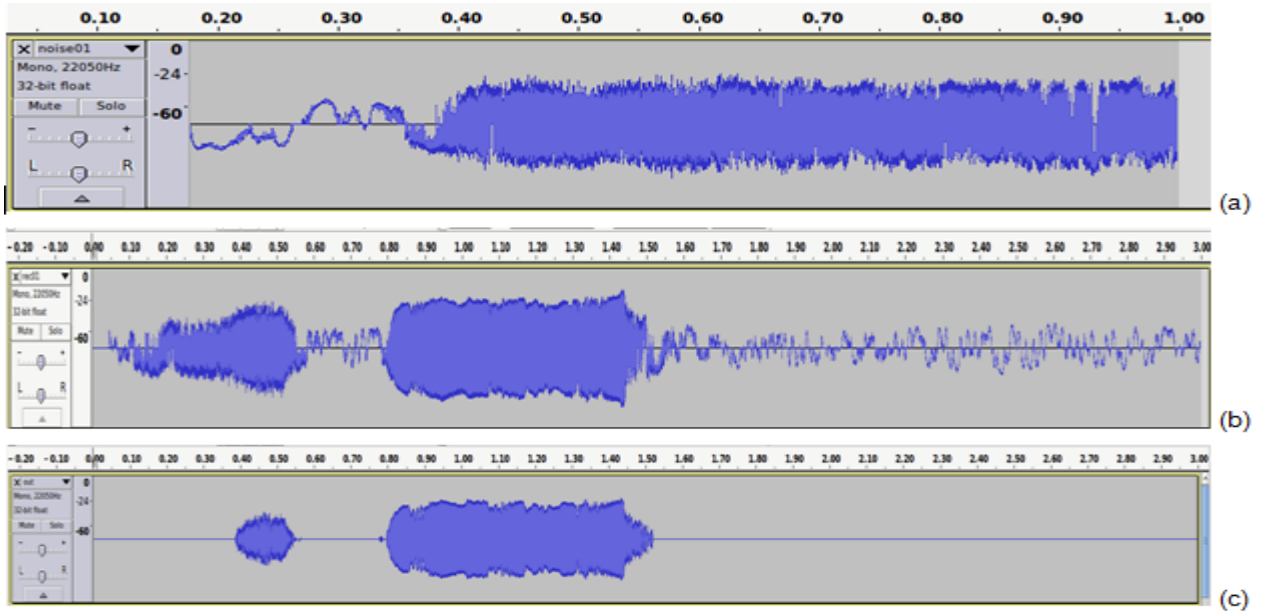


Figura 29: Remoção ruídos mostrada no domínio do tempo (Autoria própria)

Outro teste foi feito utilizando um canto de um pássaro em uma gravação, fornecida pelo site Xeno-Canto, de 11 segundos em ambiente real realizada no Paraná. O pássaro é o 'Poliptilalactea', conhecido como 'Balança Rabo Leitoso'. O teste tem o intuito de analisar a eficiência e o comportamento da rotina de remoção de ruídos em questão. A Figura 30 e a Figura 31 ilustram respectivamente, no domínio do tempo e da frequência, o sinal original seguido do sinal filtrado pelo algoritmo.

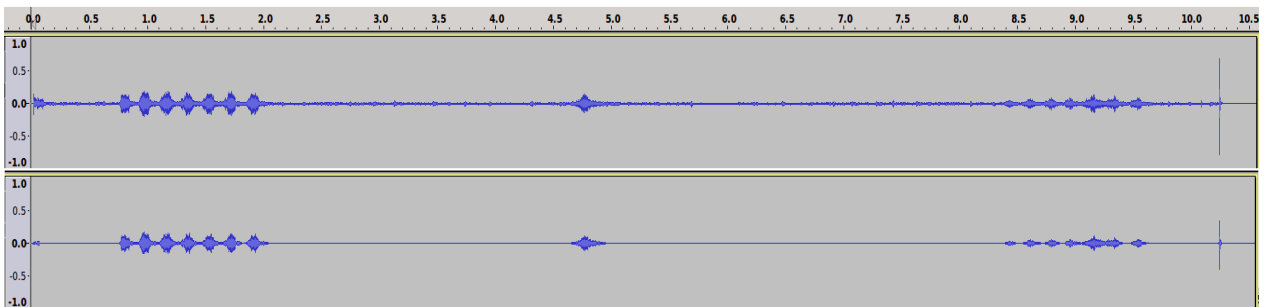


Figura 30: Canto de ave antes e depois da aplicação do filtro (Autoria própria)

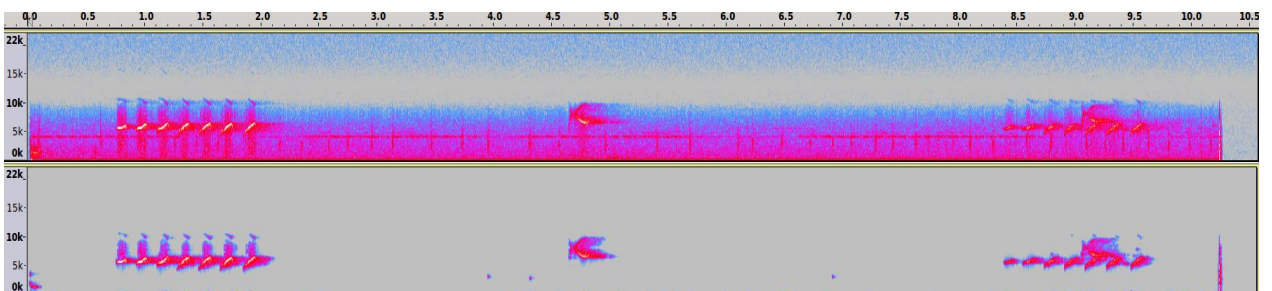


Figura 31: Espectros antes e depois da aplicação do filtro (Autoria própria)

Em complemento aos testes dos filtros, os tempos de processamento das rotinas de redução de ruídos foram medidos na UAA. Para as mesmas amostras de 3,0 segundos do teste anterior, utilizou-se amostras ruidosas como modelo de 1 e 0,5 segundo. O resultado é organizado na Tabela 18

Tabela 18: Médias de tempo de execução dos filtros em segundos

Freq. Amostragem (Hz)	Função de Janelamento	Redução Ruídos (amostras ruidosas com 1s)	Redução Ruídos (amostras ruidosas com 0.5s)
8000	<i>Hanning</i>	3,9960	3,4960
8000	<i>Hamming</i>	3,9760	3,4320
8000	<i>Bartlett</i>	2,7200	2,4860
16000	<i>Hanning</i>	8,4980	8,4980
16000	<i>Hamming</i>	7,9880	7,8420
16000	<i>Bartlett</i>	5,6740	5,4780
22050	<i>Hanning</i>	11,8960	12,0300
22050	<i>Hamming</i>	11,2500	11,3100
22050	<i>Bartlett</i>	7,9660	7,9980
44100	<i>Hanning</i>	23,7760	23,8460
44100	<i>Hamming</i>	22,9000	22,7100
44100	<i>Bartlett</i>	16,2660	16,2860

Fonte: Autoria própria

4.2.2.6 *TemplateGenerator*

TemplateGenerator (gerador da *template*) é função responsável por coletar os dados pré-processados e organizá-los em um arquivo. Tal arquivo, na classificação automática, servirá de base para o classificador conhecer características importantes do som que será classificado.

A seguir as tabelas mostram a saída do arquivo gerado para uma amostra teste coletada com taxa amostragem de 8000 Hz e submetida à *FFT* utilizando o janelamento de *Bartlett* com tamanho de 16384 (2^{14}).

Tabela 19: Exemplo de características gerais do som coletado

Exemplo de características gerais	
Nome do arquivo	/home/davi/waves/rec01.wav
Tamanho total arq. WAV	48044 bytes
Quantidade de amostras nos dados	24000 amostras
Taxa de amostragem	8000 Hz
Num. de canais	1

Exemplo de características gerais	
Código de compressão	1
Qtde de bits por amostra	16 bits
<i>Blockalign</i>	2
Data/Hora da gravação	FriMar16 10:36:07 2012

Fonte: Autoria própria

Tabela 20: Exemplo de características no domínio do tempo

Exemplo de características no domínio do tempo	
Média dos dados	-5.80317e-05
Variância dos dados	0.000441966
Desvio padrão dos dados	0.021023
Desvio médio dos dados	0.00976219
Taxa de passagem pelo zero (<i>zero crossing rate</i>)	0.0875748
Menor valor dos dados	-0.156708
Maior valor dos dados	417.58
Soma dos dados	-1.39276
Qtde de valores diferentes de zero	23191

Fonte: Autoria própria

Tabela 21: Exemplo de características no domínio da frequência

Exemplo de características no domínio da frequência	
Janelamento Utilizado	<i>BARTLETT</i>
Tamanho da Janela	16384
Centróide dos dados	0.763369
Variância dos dados	358.606
Desvio padrão dos dados	18.9369
Desvio médio dos dados	0.735093
Assimetria (<i>skewness</i>) - Classif. em localização	0.735164
Curtose dos dados - Classif. em variabilidade	121.12
<i>Rolloff</i> dos dados	1991.21

Fonte: Autoria própria

4.2.2.7 *DBControl*

DBControl(Controle do Banco de dados) é o controle e armazenamento das variáveis do sistema. Valores como taxas de amostragem e números possíveis de tamanhos de janela para a *FFT* são armazenados. Aqui também são feitos os controles de nomes e armazenamento dos arquivos WAV nas memórias do sistema.

Para realizar este controle e armazenamento não foi utilizado um banco de dados relacional, mas sim arquivos de configuração. No protótipo construído utilizou-se uma abordagem com arquivos para escrita permanente dos dados. Por esse motivo foram feitos apenas testes modificando os parâmetros de configuração deste arquivo.

4.3 CONCLUSÃO E CONSIDERAÇÕES

A materialização do sistema, mostrada neste capítulo, buscou construir uma plataforma capaz de captar sons do ambiente e prepará-lo de forma a ser utilizado em um classificador sonoro automático. Desta forma testes diversos foram realizados, abrangendo testes funcionais de um gravador comum bem como testes com as funções estatísticas de pré-processamento sonoro.

No que diz respeito aos testes de coleta e gravação dos sons do ambiente, todos se mostraram eficazes, habilitando a UAA a coletar sons em diversas faixas de frequência e armazená-los na memória do equipamento em um formato popular (.wav).

No que diz respeito ao tratamento do som, testou-se a transformada de *Fourier* deste sinal, filtragem do mesmo e extração de características do som gravado. Todos os testes foram favoráveis, tornando a UAA um equipamento capaz de realizar processamento sonoro de forma a realmente ser possível auxiliar nos estudos ornitológicos.

O próximo capítulo busca finalizar a pesquisa analisando os testes e a materialização apresentada, mensurando e adequando os limites computacionais da UAA e apresentando os estudos comparativos e as algumas reflexões sobre os resultados do trabalho.

DISCUSSÃO E ESTUDOS COMPARATIVOS

Para cumprimento dos objetivos estabelecidos nesta pesquisa, foi delimitado um escopo de trabalho que prevê o estudo e a execução de algumas etapas relativas à aquisição de sons de pássaros. Este escopo vislumbrou preparar o som coletado em ambientes reais para uso nos classificadores computacionais. O intuito deste preparo é facilitar a classificação automática em ambientes reais, retirando as etapas de preparo do som e pré-processamento da lista de tarefas destes classificadores. As demonstrações contidas no trabalho, portanto, dizem respeito às contribuições de engenharia criadas para melhorar as condições de testes em ambientes reais e também a praticidade do trabalho de ornitólogos.

Com base nestas demonstrações o intuito deste capítulo é discutir aspectos do sistema proposto. Sendo assim a primeira parte tratará de alguns resultados obtidos com a materialização e testes do modelo, como a eficiência e parâmetros de projeto. Na sequência a seção 5.2 tratará de fazer as comparações do modelo proposto com o que existe na academia e no mercado relativo à coleta e classificação de pássaros. O capítulo finaliza com a seção 5.3 trazendo as reflexões pertinentes sobre o modelo.

5.1 CONSIDERAÇÕES SOBRE OS ESTUDOS PRÁTICOS

Toda a modelagem e construção do *software* que capta e processa os sons de ambientes externos resultou em um sistema gerenciado por um processador dotado de um sistema operacional embarcado. O resultado deste sistema é ilustrado na Figura 32.



Figura 32: UAA (Autoria própria)

A diagramação do sistema, quando modelada em casos de uso, demonstrada no capítulo três resultou em um projeto de aproximadamente 1087 horas de trabalho (APÊNDICE B – PONTOS POR CASOS DE USO). Tal parâmetro foi calculado visando facilitar cálculos de homens/hora e custo de projeto para uma possível produção industrial da UAA.

5.1.1 Considerações sobre os filtros da UAA – Eficiência

Os resultados provenientes da rotina de remoção de ruídos (teste cinco do plano de testes) além de mostrarem-se úteis para eliminação de frequências indesejadas no espectro, também são úteis para eliminação de sons audíveis aos seres humanos. Prova disso é o sinal sonoro da Figura 33, que representa o canto de um pássaro chamado Inhambuguaçu (*Crypturellusobsoletus*), coletado em São Miguel Arcanjo no estado de São Paulo.

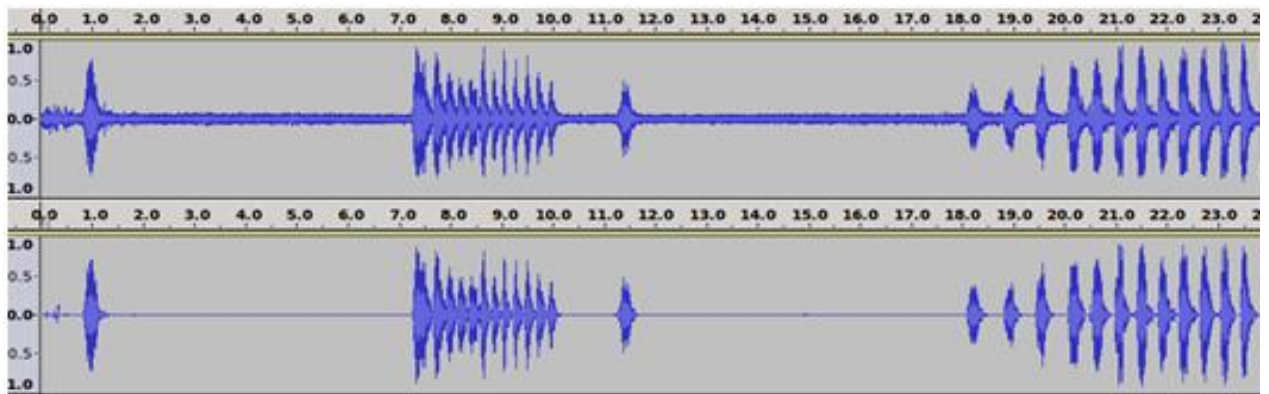


Figura 33: Canto pássaro filtrado (Autoria própria)

Durante toda a amostra ouve-se ao fundo um som de uma cachoeira, o qual serviu como amostra ruidosa no experimento ao ser separado e ajustado da amostra original em um som com duração de 1 segundo. A Figura 34 e a Figura 35 ilustram as comparações entre os espectros antes e depois (sem o som da cachoeira) da rotina de remoção de ruídos da UAA.

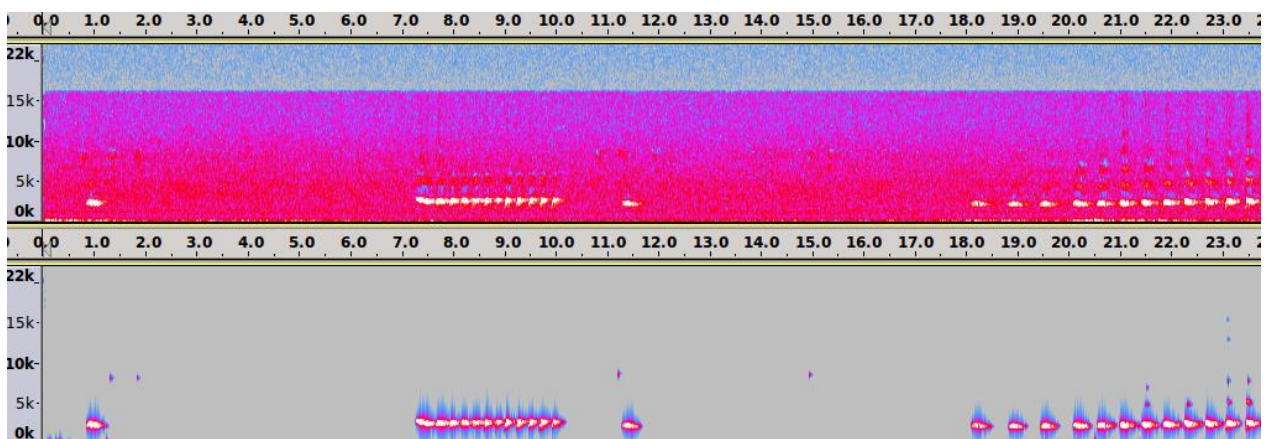


Figura 34: Resultado do espectro sonoro filtrado (Autoria própria)

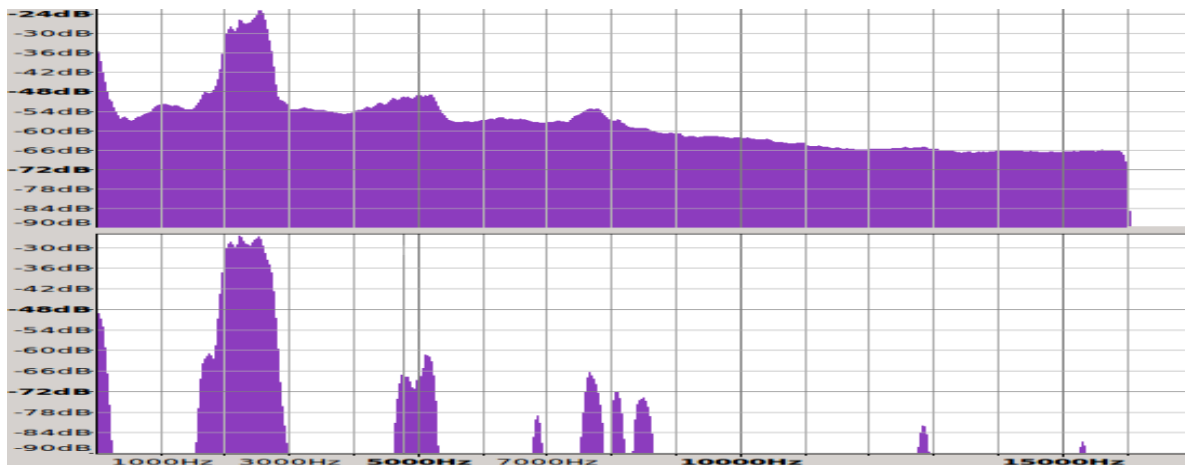


Figura 35: Comparação entre espectrogramas (Autoria própria)

5.1.2 Considerações sobre os filtros da UAA – Classificação de pássaros

O próximo passo na medição da eficiência dos filtros de remoção de ruídos da UAA foi submetê-los a testes de classificação com cantos reais de pássaros. Para tais testes dois cenários similares aos construídos por Vilches et al (2006) e Lopes, Silla, Koerich e Kaestner (2011) foram também montados neste trabalho.

O primeiro cenário baseia-se em amostras reais de três espécies de pássaros, sendo estas: 32 gravações do *Great Antshrike* ou Choró-boi (*Taraba major*), 97 cantos do pássaro *Dusky Antbird* (*Cercomacratyrannina*) e 62 gravações do *Barred Antshrike* ou Choca-barrada (*Thamnophilus doliatus*). Todas as amostras são chamadas ou pulsos dos pássaros, suprimindo pontos desinteressantes na amostra e encurtando as gravações para poucos segundos.

Todos os cantos, retirados do site Xeno-Canto, foram extraídos de ambientes reais com ruídos e sons de outros animais. As gravações estão no formato de um arquivo '.WAV' e foram submetidas ao filtro da UAA empregando uma amostra ruidosa de 1 segundo, com taxa de amostragem de 22050 Hz e janelamento de *Hanning* com tamanho de janela de 2^{10} .

O *framework* MARSYAS (Marsyas, 2010), como na pesquisa de Lopes, Silla, Koerich e Kaestner (2011), foi utilizado para gerar os vetores de características dos cantos. Fazem parte das características 64 valores diferentes entre estas médias, variâncias, centroide espectral, *rolloff*, *zero crossing rate* e doze coeficientes cepstrais da frequência de Mel (MFCC - *Mel-Frequency Cepstral Coefficients*).

Para a tarefa de classificação automática foi utilizada a ferramenta Weka (Weka, 2008) do grupo de aprendizado de máquina da Universidade de Waikato.

Foram utilizados quatro algoritmos diferentes de classificação. *NaiveBayes*, que utiliza uma abordagem probabilística, a técnica SVM, o algoritmo a árvore de decisão J4.8 e o algoritmo K vizinhos mais próximos ou *Knearestneighbors* (KNN) baseado em instâncias (instance-based).

A Tabela 22 organiza os resultados para os diferentes classificadores antes e depois da submissão do canto ao filtro da UAA. Foi utilizado como parâmetro de resultado os valores estatísticos de Medida-F ou *F-Measure* (Sasaki, 2007).

Tabela 22: F-Measure nos conjuntos de chamadas e pulsos dos pássaros

Classificador	<i>F-Measure</i> antes filtro UAA	<i>F-Measure</i> depois filtro UAA
<i>NaiveBayes</i>	89,5	88,6
SVM	94,2	97,2
J4.8	91,5	94,5
KNN (N = 3)	97,2	91,1

Fonte:Autoria própria

Nota-se com os resultados apresentados que o filtro da UAA foi suficiente para manter a eficiência do classificador. Melhorias nas classificações aconteceram utilizando o algoritmo SVM e o J.48. No caso do algoritmo KNN (N=3) houve uma piora de 6% nos resultados.

O segundo cenário montado para buscar métricas de eficiência dos cantos de pássaros submetidos aos filtros da UAA foi um cenário com gravações completas, sem separação de pulsos ou chamadas nos cantos.

A base utilizada foi semelhante à base utilizada pelos pesquisadores Lopes, Silla, Koerich e Kaestner (2011), com a diferença que os sons com longa duração, acima de 30 segundos, foram suprimidos abaixo deste limiar. Tal tarefa foi feita, devido limitações de espaço na unidade embarcada. Esta alteração não modificou qualquer parâmetro de amostragem ou precisão do som. Foram analisadas 32 gravações do pássaro *Taraba major*, 35 cantos do pássaro *Cercomacratyrannina* e 35 gravações do *Thamnophilus doliatus*.

A Tabela 23 organiza os resultados das classificações utilizando os sons completos, ou seja, sem separação por pulsos ou chamadas dos pássaros.

Tabela 23: Medidas de classificação com áudio completo

Classificador	<i>F-Measure</i> antes filtro UAA	<i>F-Measure</i> depois filtro UAA
<i>NaiveBayes</i>	84,7	91,6
SVM	91,1	89,7
J4.8	83,3	83,3
KNN (N = 3)	59,4	81,8

Fonte: Autoria própria

Neste segundo cenário, ao contrário do primeiro, o algoritmo KNN (N =3) apresentou melhorias significativas na classificação. O algoritmo SVM por sua vez, apresentou uma pequena queda nos resultados das amostras submetidas ao filtro de remoção de ruídos da UAA.

Em ambos cenários houve aproximação dos acertos do classificador antes e após a submissão do filtro. Isso mostra que não houve perda de desempenho na utilização do classificador. Para analisar a diferença entre os dois cenários a Tabela 24 sumariza os resultados dos sons filtrados pela UAA utilizando a amostra completa e somente os pulsos das respectivas.

Tabela 24: Comparação das classificações utilizando sons filtrados pela UAA

Classificador	Chamadas ou pulsos	
	Som Completo	correspondentes ao som completo
<i>NaiveBayes</i>	91,6	88,6
SVM	89,7	97,2
J4.8	83,3	94,5
KNN (N = 3)	81,8	91,1

Fonte: Autoria própria

O principal resultado observado na Tabela 24 é com relação à diferença entre as taxas de acerto nas duas situações. Diferente dos resultados de Lopes, Silla, Koerich e Kaestner (2011) (Tabela 25), os valores demonstram a diminuição drástica na diferença entre acertos nas amostras completas com relação aos acertos com amostras manipuladas e separadas em pulsos.

Tabela 25: Acertos áudio completo X Acertos pulsos correspondentes

Classifier	Full audio	Pulses
Naïve Bayes	49.6	86.9
kNN ($k = 3$)	57.3	98.4
J4.8	62.2	92.2
MLP	79.2	99.7
SVM (Polynomial)	76.0	99.4
SVM (Pearson)	71.3	99.4

Fonte: Lopes, Silla, Koerich e Kaestner (2011)

Tal fato demonstra a eficiência do filtro para classificação de amostras utilizando sons inteiros, ruidosos e não manipulados previamente. O filtro se mostrou uma forma eficaz de obter bons acertos dos classificadores não somente com amostras manipuladas previamente mais também com sons puros coletados da natureza.

Esta demonstração é importante quando se trata de um ambiente embarcado uma vez que não se dispõe de alta capacidade de processamento e as amostras coletadas necessitam ser dinamicamente manipuladas.

5.1.3 Considerações sobre os filtros da UAA – Tempos de processamento

Em termos de eficiência do código construído para realizar o pré-processamento das amostras coletadas, a Tabela 15, mostrada no capítulo anterior, organiza os tempos de execução das principais rotinas da UAA. A Figura 36 sumariza os tempos e os representa graficamente mostrando um melhor tempo de processamento para a janela de *Bartlett*.

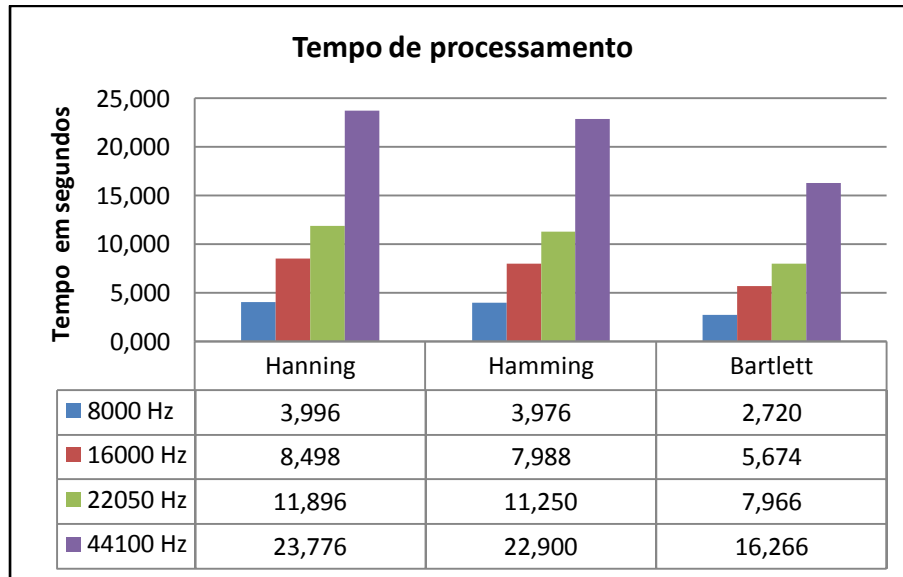


Figura 36: Tempos de processamento (Kiapuchinski, Lima e Kaestner, 2012)

Destacam-se sobre os tempos de execução algumas questões importantes. Os tempos de *FFT* para as frequências escolhidas variam de 0,30 até 1,90 segundo aproximadamente. Uma frequência de amostragem suficientemente precisa para análise de sons da natureza, por exemplo, seria 22050 Hz, o que levaria em média 1 segundo.

Sobre os tempos de execução da rotina de redução de ruídos analisou-se que o tempo da amostra ruidosa (0,5 ou 1,0 segundo) pouco interfere na duração da remoção de ruídos. A diferença esta somente na quantidade de ruído que é eliminada do sinal de saída. O espectro sonoro ilustrado a esquerda Figura 37 mostra o resultado de um sinal sonoro, coletado pela UAA e filtrado com uma amostra ruidosa de 0,5 segundo. O segundo gráfico, por sua vez, submetido ao filtro com uma amostra ruidosa de 1,0 segundo, ilustra um sinal de saída com alguns cortes adicionais em frequências ruidosas.

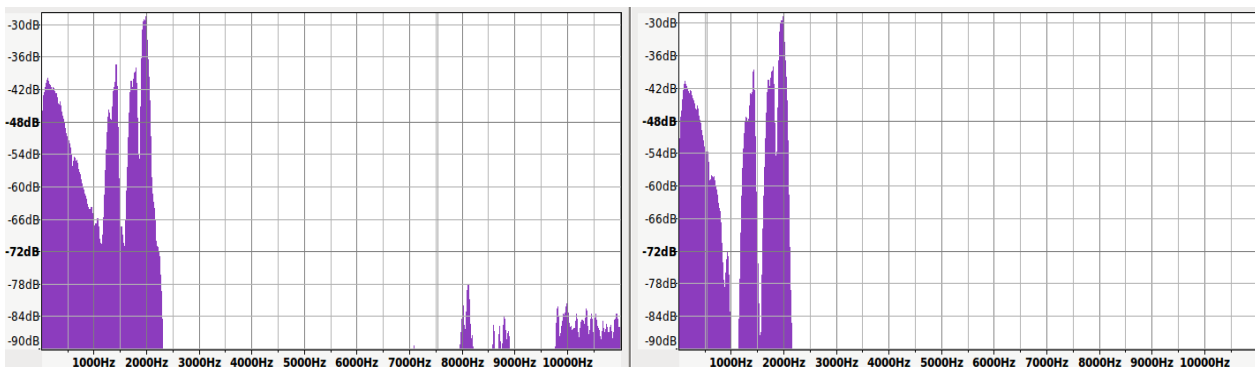


Figura 37: Diferença entre duas remoções de ruídos (Autoria própria)

Sobre o tempo total decorrido no processo de remoção de ruídos, que pode chegar até aproximadamente 24 segundos, é necessário parcimônia na parametrização do sistema dependendo da situação a fim de evitar grandes demoras no processo. Por exemplo, escolher taxas de amostragens não tão altas e análises de amostras com curta duração, melhoram consideravelmente o tempo de processamento. As possíveis melhorias qualitativas nas rotinas dos filtros são melhores descritas no próximo capítulo.

5.1.4 Considerações sobre os filtros da UAA – Comparações gráficas

Outro teste realizado, além das comparações gráficas já apresentadas, buscou analisar a eficiência do filtro comparado com uma ferramenta de análise sonora. A Figura 38 ilustra um canto de pássaro no domínio da frequência em três etapas. Na primeira etapa (a) é ilustrado o espectro original, com ruídos e frequências diversas juntas ao canto do animal. Em (b) o resultado dos filtros da UAA é apresentado utilizando um perfil ruidoso encontrado na amostra. Observe-se que o filtro reduz grande parte das frequências inúteis e o sinal de interesse passa a aparecer em destaque. Em (c) utilizou-se um filtro de ruídos existente na ferramenta de análise sonora *Audacity 1.3*. Neste caso frequências não utilizadas também foram removidas do sinal e também houve um destaque no canto do pássaro, porém não com a mesma eficiência encontrada nos filtros da UAA.

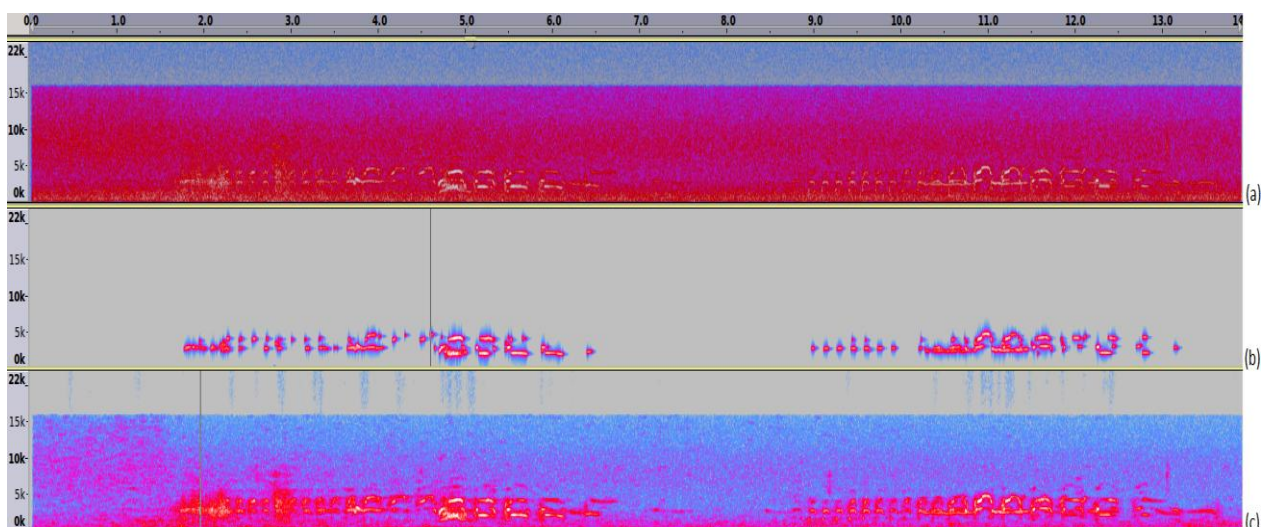


Figura 38: Comparação dos espectros sonoros (Autoria própria)

5.2 ESTUDOS COMPARATIVOS

Todos os objetivos levantados para esta pesquisa foram construídos a partir de falhas ou pontos fracos encontrados no estado da arte e da técnica apresentados em seções anteriores.

Desta forma, alterou-se o ponto de concentração dos resultados da maioria dos trabalhos na área. Tal ponto tende mostrar um foco nas evoluções metodológicas para classificação automática de pássaros, ao contrário desta pesquisa que visa encontrar um método para viabilizar todos estes resultados a situações práticas e reais.

Etapas feitas nesta pesquisa como preparar o som, melhorá-lo e submetê-lo a filtros, são etapas que alguns pesquisadores deixam algumas vezes de fazer, seja por conveniência de sons já preparados, ou pelo fato de prepará-los manualmente antes de aplicar seus algoritmos classificadores. Tal abordagem é falha se o intuito é aplicar as soluções em ambientes reais, com ruídos e sons dos mais variados tipos.

Briggs (2009), por exemplo, realizou seus testes utilizando bases de dados filtradas não focando em coletas do som, pré-processamento e extração de características. Da mesma forma Bardelli et al. (2009), que trouxe ótimos resultados em coletas já preparadas para seu método e ao testar seu algoritmo em situações reais, encontrou dificuldades na classificação devido aos diversos ruídos.

Além de prever um modelo que apresente melhorias para tais testes em condições reais, houve aqui ainda a preocupação em apresentar uma forma de tomada de decisão antecipada. Para isso um conjunto de características auxiliares sobre o sinal sonoro foi extraído (Tabela 19 a Tabela 21) visando auxiliar na contextualização do som ao classificador automático e também na escolha de um algoritmo de classificação específico para a amostra. Grande parte das características que foram utilizados nos testes de classificação na seção 5.1.2 são também extraídas pela UAA.

Fagerlund (2007), Somervuo et al (2006) e Vilches et al (2006) realizam técnicas de pré-processamento no sentido de dividir em sílabas um som (canto) já filtrado e separado como um canto em potencial, restando assim o teste de confiabilidade do método. Tal método, agindo sob uma pequena parte pertinente do espectro tende a facilitar testes em ambientes reais, sendo importante esta comparação com os resultados desta pesquisa. Os testes de eficiência na

classificação automática de pássaros na UAA utilizaram-se também de partes relevantes do sinal sonoro (chamados ou pulsos), uma vez que o intuito do teste foi demonstrar que o uso do filtro não interfere nos bons resultados já apresentados nas pesquisas da área.

Os autores supracitados, em suas publicações não ressaltaram as etapas de coleta e de preparo do som. Fazendo sempre alterações necessárias a melhores resultados na fase de testes. Prova disso são as conclusões dos autores que revelam a dificuldade em aplicação dos métodos em maior número espécies e também a necessidade existente em desenvolver classificadores mais genéricos e melhores técnicas de preparo do sinal sonoro.

Com a UAA, estas alterações (preparo, correção e filtragem) na amostra, são feitas de forma automática e na fase da coleta, automatizando ainda mais o processo como um todo.

A partir das análises do problema feitas durante o trabalho, dos resultados apresentados e das comparações entre os sistemas já existentes, uma reflexão sobre melhorias e deficiências do modelo é apresentada na seção a seguir.

5.3 REFLEXÕES

Os testes de gravação do som, de aplicação de filtros, de extração de características e de classificação (Tabela 22) mostram que a UAA tem capacidade para preparar adequadamente uma amostra real a um classificador automático. Demonstrou-se que eliminando ruídos indesejados e variáveis que não são previstas na amostra aumentam-se as chances de sucesso com amostras reais.

Outra falha apresentada no início do trabalho diz respeito à dedicação repetitiva em etapas de pré-processamento nos trabalhos da área. Filtros redutores de ruídos e filtros de frequências foram etapas aqui programadas que auxiliam na captação de um som para testes 'limpo'. Possibilitando que etapas que eram feitas anteriormente de forma manual, agora sejam feitas no momento da coleta. Além disso, com as características do som ambiente, sintetizadas pela *templatesonora*, são mostradas diversas informações importantes ao som coletado. Sendo então possível configurar previamente um classificador mais adequado à situação, minimizando desta forma, erros variados.

Sobre as melhorias, salienta-se que a usabilidade do sistema não foi testada em aplicações com grandes tempos de gravação e uso do sistema. Um maior teste

de integração e usabilidade do *software* poderia ser feito para averiguar o desempenho destes fatores.

5.4 CONCLUSÃO E CONSIDERAÇÕES

Este capítulo buscou confirmar e explicar sobre a satisfação dos objetivos elencados no início do trabalho. Tal satisfação se deve a materialização e testes realizados na UAA. Através das comparações gráficas ilustraram-se a eficiência das rotinas programadas, da mesma forma que através dos testes funcionais e práticos provou-se que o modelo projetado é suficiente para utilização em aplicações reais, adicionando praticidade as pesquisas da área.

CONCLUSÃO

Como mostrado no início do trabalho, a busca por soluções eficientes para o controle da biodiversidade dos pássaros, vem mobilizando empresas, pesquisadores e institutos a encontrar soluções inovadoras que melhorem aspectos da área. Automatizar tarefas da coleta dos cantos, auxiliar o trabalho de ornitólogos e salvaguardar a vida animal e humana são algumas das justificativas iniciais que originaram os objetivos desta pesquisa e o projeto *BrBird*.

As dificuldades discutidas no decorrer deste trabalho fazem com que alguns resultados da computação e da ornitologia sejam limitados e impraticáveis a situações reais, tais como aeroportos, usinas eólicas e ambientes abertos. Isso porque, talvez o foco de grande parte da pesquisa seja em trabalhar para melhorar os algoritmos classificados, dando menos importância a uma solução mais ampla à situação. Portanto, a solução mostrada baseou-se em analisar o problema de forma diferente, retirando o foco dos bons resultados em acertos dos classificadores automáticos e o colocando sobre a fase da coleta das informações que são à entrada destes algoritmos.

Com essa abordagem era esperado fornecer bases sólidas aos bons resultados dos classificadores, mantendo a possibilidade de aplicá-los em situações reais práticas e dinâmicas. Para isso, foi necessário o projeto de um sistema específico, funcional e portátil, intitulado como Unidade de Aquisição Autônoma (UAA). A UAA utiliza-se de rotinas de pré-processamento sobre o sinal sonoro de forma que o som de saída torne-se mais “limpo” e, portanto, apto a ser à base dos classificadores automáticos do estado da arte desta pesquisa.

Como características principais deste ambiente citam-se o fato de utilizar de um SO embarcado (Linux), ser de fácil adaptabilidade em termos de manutenções, projetos e melhorias em nível de *software*. Por um lado rotinas de preparo do som, como filtros, normalização e extração de características compõem as etapas de pré-processamento e preparo do som. Por outro lado, conexões com dispositivos USB, o uso de *templates* e *sockets* para a propagação da informação, são as características de engenharia do sistema que auxiliam a quem interessar os resultados desta pesquisa.

Para “Oferecer solução para o problema de classificação de cantos de pássaros, no que diz respeito ao pré-processamento das informações coletadas”, foi necessário um estudo utilizando a *SysML*. Através do plano de teste construído, os resultados foram métricas de eficiência do equipamento e os valores de acerto da classificação automática em amostras reais submetidas ao filtro programado.

A eficiência dos métodos computacionais de redução de ruídos sonoros foi demonstrada através da materialização de alguns requisitos (e.g. REQ SW 101) e pelo resultado dos testes das tabelas de tempos de processamento das rotinas da UAA. A integração em uma plataforma de *hardware* das funcionalidades para análise e filtragem de sinais sonoros foi satisfeita com a materialização dos requisitos de *hardware* e testes utilizando um circuito com o microcontrolador de núcleo ARM.

Ferramentas como *Audacity*, *Weka* e *Marsyas* foram utilizadas para validar a plataforma final através de comparações gráficas e estatísticas dos sons filtrados pelo modelo, com os resultados oriundos das ferramentas já existentes de manipulação sonora. O resultado desta validação foi um ambiente suficiente no sentido da classificação automática com amostras reais e não preparadas e um ambiente eficiente quando comparado aos filtros já existentes nas ferramentas de manipulação sonora.

6.1 TRABALHOS FUTUROS

O grande interesse em assuntos envolvendo biodiversidade e segurança da vida dos animais e seres humanos fornece uma quantidade considerável de futuras aplicações para um modelo como o apresentado neste trabalho. Paralelamente, vê-se que a inteligência artificial e o aprendizado de máquina muito têm evoluído nos últimos anos. Fazendo com que a união destes dois assuntos seja o grande foco para os trabalhos futuros.

Em trabalhos futuros seria interessante academicamente, além de melhorar aspectos qualitativos das rotinas, comparar a eficiência entre sistemas utilizando plataformas diferentes. Construir algoritmos de remoção de ruídos para um *hardware* reconfigurável como por exemplo o arranjo de portas programável em campo ou *Field-programmable gate array* (FPGA). Os tempos de processamento e os desempenhos poderiam ser medidos de forma a consolidar uma plataforma ótima para solucionar os problemas da área.

Outra vertente para o trabalho seguir adiante seria o desenvolvimento de um produto que possa ser produzido e comercializado em maior escala e com maiores funcionalidades. Estudos para aplicabilidade do modelo em sistemas de inteligência artificial e militares, por exemplo, poderiam ser feitos.

6.1.1 Estudo da aplicabilidade para comercialização

Um produto eletrônico para comercialização exige uma grande quantidade de testes minuciosos de integração dos blocos do sistema, bem como o refinamento das funções de forma a minimizar os custos computacionais. Como colaboração a classificação dos pássaros, funcionalidades poderiam ser adicionadas no que diz respeito a sensoriamento do ambiente. Exemplos destas seriam sensores para chuva e umidade. Desta forma, além de pré-processamento do som, bases de dados com condições climáticas poderiam ser construídas a fim de realizar comparações de rotinas da fauna que são relacionadas às condições climáticas.

Outra futura utilização para um possível produto que fosse originado deste trabalho seria a utilização do sistema atuando sobre ambientes em que animais não devem circular. Aeroportos, lixões, usinas eólicas e plantações são alguns exemplos destes ambientes. O sistema ao detectar a aproximação do animal poderia emitir alertas ou alarmes ao ambiente de forma que animal não circule por aquela área.

Agregações ao produto final como câmeras, baterias solares e microfones mais potentes podem ser também consideradas como trabalhos futuros que agregariam valor ao trabalho.

Como complementação às horas de projeto necessárias à construção da UAA a próxima divisão mostra alguns parâmetros financeiros do projeto.

6.1.1.1 Orçamento

Na Tabela 26 são apresentados, de forma aproximada, os custos da UAA..

Tabela 26: Custos do projeto

Componente	Qtde	Preço un.		Preço Total	
Kit Programação para ARM920T	1	R\$	200,00	R\$	200,00
Ferramental, Carenagem e Baterias	1	R\$	160,00	R\$	160,00
Microfone analógico	1	R\$	45,00	RS	45,00
Componentes Eletrônicos (Geral)	1	R\$	30,00	R\$	30,00
			TOTAL	R\$	435,00

Fonte: Autoria própria

Um custo importante do trabalho, como já dito, é relacionado com o resultado do Apêndice B. Aproximadamente 1090 horas é um número a ser utilizado como base para realizar um possível plano financeiro e mercadológico de um produto. Cálculos como quantidade de produtos que deveriam ser vendidos e de funcionários envolvidos no projeto podem ser feitos a partir destas horas e dos custos fixos. Sendo útil, portanto, para descobrir informações como ponto de equilíbrio e retorno de investimento do projeto.

Os *softwares* utilizados no projeto são todos enquadrados na categoria de *software* livre, ou ainda com código aberto, como é o caso do Linux e das suas bibliotecas para desenvolvimento do *software*. Sendo assim não acarretam custos diretos ao sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

AGRANAT I. Automatically identifying animal species from their vocalizations. **2009 Acoustics. Inc.**, Concord, Massachusetts, p. 22, 2009.

AUDACITY Manual. **Digital Audio.** Disponível em: <http://manual.audacityteam.org/man/Digital_Audio/pt_BR>. Acesso em 22 de agosto de 2011.

BARDELLI R., et al Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. **Pattern Recognition Letters.** p. 11, 2009.

BAHILL T. A. e GISSING B. Re-evaluating Systems Engineering Concepts Using Systems Thinking. **IEEE transactions on systems, man, and cybernetics - part C: applications and reviews.** Vol. 28, num. 4, 516-527, nov, 1998.

BETTS M. G. A., DIAMONT W. A., FORBES D. G., VILLARD M. A. e GUNN J. The importance of spatial autocorrelation, extent and resolution in predicting forest bird occurrence. **Ecological Modelling,** vol. 191, ed. 2, p.197 – 224, 2006.

BOOCH, G.; RUMBAUCH, J. e JACOBSON, I.. **UML: Guia do Usuário.** 2. ed. Elsevier Campus. p 500, 2005.

BRASIL. Lei n. 7.565, de 19 de dezembro de 1986 Dispõe sobre o Código Brasileiro de Aeronáutica, artigo 43.

BRIGGS F. e FERN Z. X. Audio Classification of Bird Species: a Statistical Manifold Approach. **Ninth IEEE International Conference on Data Mining,** 1550-4786, 2009.

BRIGHAN, E. O., MORROW R. E. The Fast Fourier Transform. A detailed explanation is offered of an algorithm that reduces computer time and allows its user to employ powerful frequency techniques once considered inefficient. **IEEE spectrum,** p. 8, 1967.

CELEPAR. **Guia de Estimativa Projeto. Metodologia CELEPAR.** Celear Informática do Paraná p.10, 2009.

CENIPA. **O Perigo da fauna na aviação brasileira.** Artigo 118, disponível em: <<http://www.cenipa.aer.mil.br/cenipa/index.php/artigos-cenipa/118-o-perigo-da-fauna-na-aviacao-civil-brasileira>> Acesso em out. 2010.

CHAMPAGNE B. e LABEAU F. Discrete Time Signal Processing, **Class Notes for the Course ECSE-412,** Department of Electrical & Computer Engineering, McGill University, p. 277, 2004

CROSS D. Fast Fourier Transforms, **C/C++ source code FFT,** disponível em <<http://groovit.disjunkt.com/analog/time-domain/FFT.html>>. Acesso em mai 2012, última atualização em mar 1988.

COOLEY W. J e TUKEY W. J. An algorithm for the machine calculation of complex Fourier series, **Mathematics of Computation**, p. 297-301, 1965.

DOUGLASS B. P. **Real-Time UML: Developing Efficient Objects for Embedded Systems**, Addison-Wesley Longman Publishing Co. ISBN:0201325799, p. 363, 1997.

DOWNING D e CLARK J. **Estatística Aplicada**, Série Essencial. Editora Saraiva, 2º ed. P.351, 2006.

DUARTE L. J. V., MAITELLI, L. A., SALAZAR, O. A. et. al. Sistemas embarcados baseados em DSP's para filtragem e pré-deteção de vazamentos em oleodutos. **Rio Oil& Expo and Conference 2006**. Rio de Janeiro, edição 6, p. 8, set. 2006.

ESS V. D. Digital Processing In The Density Domain. **Eletronic Design**. Disponível em: <<http://electronicdesign.com/article/analog-and-mixed-signal/signal-processing-in-the-density-domain-part-iv->>. Acesso em: nov 2011.

FAGERLUND S. Bird Species Recognition Using Support Vector Machines. **EURASIP Journal on Advances in Signal Processing**, vol 2007, p. 8, mar. 2007.

FRIENDLYARM. FriendlyARM Mini 2440 SBC (Single-Board Computer), Mini2440 Schematic Diagram. Disponível em <www.friendlyarm.net>. Acesso em nov 2010.

FAYYAD U., PIATETSKY G. e SMYTH P. The kdd process for extracting useful knowledge from volumes of data. **Communications of the ACM**, Nova York, vol 39, p.8, 1996.

GALVÃO H. K. R., VISANI U. C. M., SALDANHA B. C. T. et. al. Estudo comparativo sobre filtragem de sinais instrumentais usando transformadas de *Fourier* e wavelet. **Química Nova**, vol. 24, num. 6, São Paulo, p. 874, 884, nov. 2001.

HODGSON J., **Understanding Records: A Field Guide to Recording Practice**, Continuum International Publishing Group, p. 272, 2010.

HOTTA, C. The Subsidiary Patient, **Brazilian Thoughts**. Disponível em: <http://scienceblogs.com.br/brazillion/2009/07/many_times_in_my_practice/>. Acesso em fev. 2012.

IFEACHOR E. e JERVIS B. Digital signal processing, a practical approach, **Prentice Hall**, Segunda edição, 651-654, 2002.

jAudio 1.0. Disponível em: <<http://jaudio.sourceforge.net/index.html>>. Acesso em 15/03/2012.

KARNER G. Resource Estimation for Objectory Projects. **Objective Systems SF AB**. p. 9, 1993.

KIAPUCHINSKI, D. M., LIMA, C. R. E. e KAESTNER, C. A. A. Spectral Noise Gate Technique Applied to Birdsong Pre-processing on Embedded Unit, **IEEE International Symposium on Multimedia**, Irvine, California USA, 2012 – ACCEPTED PAPER IN AUGUST 2012.

KUULUSA, M., NURMI, J., TAKALA, J., et. al. A flexible DSP core for embedded systems. **Design & Test of Computers, IEEE**. Tampere, vol. 4, p. 60 – 68, ago 2002.

KWAN, C., HO, C.K, LI, Y. et al. An Automated Acoustic System to Monitor and Classify Birds. **Journal on Applied Signal Processing**, volume 2006, id 96076, p. 1-19, out. 2005.

LAURINO S. L. Arquiteturas para reconhecimento de voz em sistemas embarcados. **Programa de pós-graduação em computação, Universidade Federal do Rio Grande do Sul**, Porto Alegre, p. 11, Nov. 2004.

LibXtract 0.4.9. Disponível em: <<http://libxtract.sourceforge.net/>>. Acesso em: 18/12/2011.

LIU C. W. KROODSMA E .D.e YASUKAWA K. Dawn and daytime singing behavior of chipping sparrows (*Spizella Passerina*). **The Auk, The American Ornithologists Union**, Massachussets, vol 124 p. 44–52, 2007.

LOGAN B. Mel frequency cepstral coefficients for music modeling. **International Symposium on Music Information**, Cambridge, p. 13, 2000.

LOPES T. M., SILLA C. N., KOERICH A. L., KAESTNER A. A. C. Feature Set Comparison for Automatic Bird Species Identification. **Systems, Man, and Cybernetics (SMC) 2011 IEEE International Conference**, p. 965 – 970, 2011.

Marsyas Web Site, disponível em <<http://marsyas.info/>>. Acesso em setembro de 2012, 2010.

MENDONÇA C. A. F. Gerenciamento Do Perigo Aviário Em Aeroportos. **Conexão SIPAER**, v. 1, n. 1, Edição de Lançamento, nov 2009.

MELO S. C. M e CÁRIO F. A. S. Trajetória tecnológica do setor de telecomunicações no brasil: a tecnologia VoIP. **ANPEC-Sul**, Curitiba, p. 20, jun. 2008.

NXP Semiconductors. Product specification, data sheet. **UDA1341TS Economy audio CODEC for MiniDisc (MD) home stereo and portable applications**, p. 34, mai. de 2006.

OSBORN G. **Embedded Microcontrollers and Processor design**. Ohio, Prentice Hall, 2010. 431 p.

PEREIRA F. **Tecnologia ARM Microcontroladores de 32 bits**. São Paulo, Editora Érica. 488 p., 2007.

PINILLA J. **Manual para el anillamiento científico de aves**. SEO Bird Life, Ministério do Meio Ambiente, Espanha, p. 160, 2000.

RememBirdII. **Digital Audio Recorder for Birdwatchers**. Software Hothouse Ltd, South Chailey, United Kingdom, p. 50, 2010.

ROSS D. **Bird Call Recognition with Artificial Neural Networks, Support Vector Machines, and Kernel Density Estimation**. Master of Science, Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, p. 131, março 2006.

RUSSEL J. S. e NORVIG P. **Artificial Intelligence: A Modern Approach**, Prentice Hall, p. 1132, 2009.

SAHA S., BHATTACHARYYA, S. S. e WOLF W. A Communication Interface for Multiprocessor Signal Processing Systems. **Embedded Systems for Real Time Multimedia, Proceedings of the 2006 IEEE/ACM/IFIP Workshop on**, Seoul, p.127 – 132, 2006.

SANSUNG ELETRONICS. **S3C2440A, 32-bit CMOS microcontroller user's manual**. Revisão 1, Yongin-City, p. 596, 2004.

SILLA JR, N. C., KAESTNER A. A. C. e KOERICH L. A. Classificação automática de gêneros musicais utilizando métodos de Bagging e Boosting. **10° Simposio Brasileiro de Computação Musical**. p.10, 2005.

SILLA C. N. JR., KAESTNER C. A. A. e KOERICH A. L. Automatic Music Genre Classification using Ensemble of Classifiers. **Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'07)**, Montreal, Canada, pp. 1687-1692, 2007

SILLA C. N. JR., KAESTNER C. A. A. e KOERICH A. L. A Machine Learning Approach to Automatic Music Genre Classification, **Journal of the Brazilian Computer Society**, Vol. 14(3), pp. 7-18, 2008.

SLOSS N. A., SYMES D. e WRIGHT C. **ARM System Developer's Guide, Designing and Optimizing System Software**. Ed Elsevier, p. 703, 2004.

SOMERVUO P., HÄRMÄ A. e FAGERLUND S. Parametric Representations of Bird Sounds for Automatic Species Recognition. **IEEE Transactions on Audio Speech and Language Processing**, vol. 14, n. 6, p. 2252 – 2263, novembro, 2006.

SOUZA A., MARTINS E. **Estatuto Social da WWF-Brasil**, 2009. Disponível no sítio do WWF-Brasil: http://www.wwf.org.br/wwf_brasil.

SOUZA J. A. M. F. **Análise de Sinais, Transformadas de Laplace**, 4° edição, capítulo 5, 44 p., mar, 2010.

SPIEGEL, R. M. **Transformadas de Laplace resumo da teoria : 263 problemas resolvidos : 611 problemas propostos**. McGraw-Hill do Brasil, São Paulo, p. 344, 1976.

STERELA PRODUCTS. **EFFTEL Bird Scarer System**. Disponível em: <http://www.sterela.fr/fpdb/File/aeroporto/EFFTEL_Bird_scarer_Ind_B.pdf>. Acesso em 15/08/2011.

STRAUBE, C.F. História da Ornitologia no Paraná. **Ornitologia sem fronteiras**, Curitiba, Fundação O Boticário de Proteção à Natureza, p. 43-116, jul. 2001.

STRAUBE, C.F, **Boletim Informativo do COAVE (Clube de Observadores de Aves do Vale Europeu)**, out. 2005. Disponível em: <<http://maicon-mohr.blogspot.com.br/2010/04/entrevista-com-fernando-straube.html>>. Acesso em 10/06/2010.

STRAUBE C. F, KRUL R., CARRANO E. Coletânea da avifauna da região sul do estado do Paraná (Brasil). **Atualidades Ornitológicas N.125**, maio/junho de 2005 pág.10, p. 62.

Systems ModelingLanguage, **SysMLSpecification**. Version 0.9, draft. Jan, 2005. Disponível em: <www.SysML.org>. Acesso em 10/04/2011.

TAYLOR E. C., STABLE P. E., CODY M. et al. Sensor Arrays for Acoustic Monitoring of Bird Behavior e Diversity. **National Science Foundation**, Center for Embedded Network, Los Angeles, jan. 2005.

VIELLIARD, J. M. E. Areas of differentiation and biogeography affinities within the avifauna of Northeastern Brazil. **Anais V Congresso Brasileiro de Ornitologia**. Unicamp, Campinas, p. 184-190, 1996.

VILCHES E., ESCOBAR A. I., VALLEJO E. E et al. Data Mining Applied to Acoustic Bird Species Recognition. **18th International Conference on Pattern Recognition**, Hong Kong, vol 3, p. 400 - 403, ago. 2006.

VISCONTI M. A. e FIGUEIREDO L. F. A. **Estatuto do centro de estudos ornitológicos**, 2007. Disponível no sítio do centro de estudos ornitológicos: <http://www.ceo.org.br/>.

VORAN S. Objective Estimation of Perceived SpeechQuality—Part I: Development of theMeasuring Normalizing Block Technique. **IEEE Transactions on Speech and Audio Processing**, vol 7, num 4, p 371- 382, jul. 2007.

WATTS, L., MASSIE, D., SANSANO, A, et al. Voice processors based on the human hearing system. **Micro, IEEE**, vol 29, p. 54 – 63, abr. 2009.

WEILKIENS T. **SystemsEngineering with SysML/UML. Modeling, Analysis, Design**. Morgan Kaufmann OMG Press, 2006, p. 320.

WEKA. **WEKA Machine learning software in JAVA - Manual for Version 3-6-0**.Disponível em: <<http://www.cs.waikato.ac.nz/ml/weka/>>. Acessoem12/10/2012, dez.2008.

WU. H., SIEGEL M. e KHOSLA P. Vehicle Sound Signature Recognition by Frequency Vector Principal Component Analysis, **IEEE Instrumentation and Measurement Technology Conference**, vol. 48, p. 1005 – 1009, out. 1999.

XENO CANTO. **Community database sharing songs from around the world**. Disponívelem: <http://xeno-canto.org>. Acessoem: 10 nov. 2010.

SASAKI, Y. **The truth of the F-measure**. School of Computer Science, University of Manchester, Manchester, out. 2007.

Zill D. **Matemática Avançada para Engenharia - Equações diferenciais elementares e transformadas de Laplace**. Bookman, 3ª edição, p. 340, 2009.

ZHANG W. e LIN B. Study of intelligence embedded Linux surveillance system based on TCP/IP. **Computer Engineering and Technology (ICCET), 2010 2nd International Conference on**. Chengdu, volume 7, p. V7-75 - V7-77, jun. 2010.

APÊNDICE A – DESCRIÇÃO CASOS DE USO

Depois da explicação do caso de uso dada na seção de “Tradução dos requisitos” no capítulo 3 abaixo é mostrado uma descrição detalhada dos casos, com suas relações e seus fluxos de eventos.

Tabela 27: Especificação do caso de uso ‘Gerir Variáveis’

Nome:	Gerir Variáveis		
Descrição:	<p>Caso de Uso responsável por fazer a comunicação e a gerência das rotinas de comandos dos sensores ou atuadores bem como traduzir seus comandos específicos e interagir com o banco de dados.</p> <p>Os sensores do sistema fazem a leitura das variáveis externas e este caso de uso tem a função de processar os valores, ajustá-los e prepará-los para o processamento.</p> <p>Os atuadores do sistema ficam responsáveis por atuar externamente ao sistema sob demanda de comandos do caso de uso da função de controle das variáveis.</p>		
Ator Principal:	Sensor / Atuador	Ator de Bastidor:	Não Possui
Pré Condições	Gerir ações de alto nível relacionadas à leitura e envio de dados externos quando a situação assim exige. E também gerir ações de gravação e leitura de dados na memória do sistema.		
Pós Condições	Garantir que as leituras/atuações externas ou interações com o banco de dados sejam feitas corretamente com a classe de abstração DAO.		
Regras de Negócio	<p>Toda interação relacionada a atuação ou sensoriamento externo passa por este caso de uso para controle e correto direcionamento.</p> <p><i>Drivers</i> de comunicação bem como interfaces de banco de dados são as entradas/saídas deste caso de uso.</p>		
Fluxos Básicos	<p>1) O controle de variáveis recebe um comando (banco de dados ou dos <i>drivers</i> de interação externa).</p> <p>2) Se oriundo de um sensor, é pré-processado e enviado a função específica de interpretação do dado. Se oriundo do processamento central ou banco de dados da aplicação então é caracterizado e direcionado a saída respectiva.</p>		

Fluxos alternativos	<p>A variável é enviada ao atuador</p> <ol style="list-style-type: none"> 1) O sistema, por agendamento ou por demanda coloca como contexto de execução o tratamento da atuação externa. 2) É analisado a requisição, o contextos bem como a porta e o nível lógico de transição (0 ou 1). 3) É enviado o sinal de transição à porta ou pino especificado. O Atuador recebe o novo sinal elétrico podendo alterar de estado ou manter o estado anterior. <p>A variável é recebida pelo sensor</p> <ol style="list-style-type: none"> 1) O sistema, por agendamento ou por demanda coloca como contexto de execução o tratamento do sensoriamento externo. 2) O Monitoramento das Variáveis realiza leitura na porta, pino ou resposta de comunicação especificada. 3) O Monitoramento das Variáveis realiza processamento e preparo da informação. <p>Se pertinente à regra de negócio o Monitoramento das Variáveis envia o valor lido ou resultado da leitura ao banco de dados.</p>
----------------------------	--

Fonte: Autoria própria

Tabela 28: Especificação do caso de uso 'Processar Som'

Nome:	Processar Som		
Descrição:	Rotina principal da aplicação. Aqui serão feitas as manipulações do som, tomadas de decisão e preparação dos dados pertinentes a serem salvos em memória.		
Ator Principal:	Microfone	Ator de Bastidor:	Sensor
Pré Condições	Receber informações filtradas dos outros casos de uso e atores do sistema.		
Pós Condições	Fornecer uma resposta da interpretação do espectro sonoro positivamente o não no que diz respeito a relevância do sinal analisado.		
Regras de Negócio	Apenas os conjuntos de dados que mostrarem índices interessantes à aplicação são gravados em banco de dados.		
Fluxos Básicos	<ol style="list-style-type: none"> 1) O som preparado é enviado a este caso de uso. 2) Envio para eliminação e suavização de ruídos 3) Análise dos valores e resultado (1.a) (1.b) 4) Encaminhamento do som de saída (2.a) (2.b) 		

Fluxos Alternativos	<p>ANÁLISE PERTINENTE A CONTINUAÇÃO DA ANÁLISE DO ESPECTRO</p> <p>1.a O espectro é pertinente à normalização. 1.b O espectro é pertinente à extração de características.</p> <p>DUAS DECISÕES PODEM SER TOMADAS A PARTIR DO RESULTADO DA ANÁLISE.</p> <p>2.a Positiva o resultado é gravado em banco de dados, bem como informações de data, hora, valores e tempo de gravação e ficam disponíveis para a geração de uma <i>template</i> de características. 2.b O resultado é descartado e a análise continua.</p>
----------------------------	---

Fonte: Autoria própria

Tabela 29: Especificação do caso de uso 'Preparar Som'

Nome:	Preparar Som		
Descrição:	Caso de uso que representa a primeira etapa da análise do sinal sonoro recebido pelo microfone. Responsável por uma pré-filtragem das informações e também de repassar os dados considerados interessantes.		
Ator Principal:	Microfone	Ator de Bastidor:	Sensor
Pré Condições	Leitura dos registradores com os dados do som		
Pós Condições	Afirmção de interesse ou não sobre o que foi sensoriado.		
Regras de Negócio	Todo sinal sonoro captado pelo equipamento passa primeiramente por essa filtragem inicial e é enviado para o processamento.		
Fluxos Básicos	<ol style="list-style-type: none"> 1) <i>Polling</i> para aquisição do som 2) Transdução do sinal sonoro em valores digital 3) Montagem do sinal sonoro 4) Avaliação do sinal pré-processado 5) Encaminhamento do sinal 		

Fonte: Autoria própria

Tabela 30: Especificação do caso de uso 'Formatar Saída'

Nome:	Formatar Saída		
Descrição:	Fornecer interface facilitada para <i>softwares</i> classificadores, de forma que exista uma fácil interação entre os dois sistemas (equipamento e classificador em um servidor)		
Ator Principal:	Banco de Dados	Ator de Bastidor:	Dispositivo de Comunicação
Pré Condições	Junção e preparo dos dados brutos armazenados em memória no equipamento		
Pós Condições	Relatório em formato diferenciado das informações		

Regras de Negócio	Todo relatório a ser gerado deve ter passado por uma avaliação de potencial, no que diz respeito a importância do espectro/som para o trabalho
Fluxos Básicos	<ol style="list-style-type: none"> 1) Os dados são solicitados ao banco de dados através de ações externas as quais buscam algum tipo de informação relevante no equipamento. 2) Os dados são jogados na <i>template</i> do relatório e os dados vão ser formados 3) Calcula-se o <i>check sum</i> do arquivo gerado. 4) É feito o encaminhamento do relatório a classe gerenciadora de conexão.

Fonte:Autoria própria

Tabela 31: Especificação do caso de uso 'Controlar Comunicação'

Nome:	Gerir Comunicação		
Descrição:	Caso de uso que controla as comunicações externas que funcionam através de um protocolo de comunicação, ou seja, neste caso o dispositivo USB e as <i>sockets</i> de comunicação.		
Ator Principal:	Dispositivo de Comunicação	Ator de Bastidor:	Não Possui
Pré Condições	Solicitação interna ou externa para iniciar a comunicação		
Pós Condições	Fechamento da conexão.		
Regras de Negócio	Representar um bloco que disponibilize e envie informações para dispositivos de comunicação.		
Fluxos Básicos	<ol style="list-style-type: none"> 1. A camada de interface recebe a solicitação para tratar uma requisição de envio/recebimento de informações 2. A informação é recebida pela função de gerenciamento de comunicação 3. É feito a interpretação/parser da informação que chega 4. Inicia-se o encaminhamento 5. Término da transmissão (1.a) 		
Fluxos Alternativos	EM CASO DE RETRANSMISSÃO 1.a. Em alguns casos será feita uma verificação de erros Em caso de erros acontece a retransmissão.		

Fonte:Autoria própria

Tabela 32: Especificação do caso de uso 'Filtrar Som'

Nome:	Filtrar Som		
Descrição:	Caso de uso que tem o propósito de aplicar sobre o espectro filtros de suavização e eliminação de ruídos.		
Ator Principal:	Microfone	Ator de Bastidor:	Sensor
Pré Condições	Receber o sinal sonoro, preparado pelo caso de uso 'Preparar Som'		
Pós Condições	Som filtrado.		

Regras de Negócio	Todo som que chega à fase de processamento, na forma de um espectro sonoro, é submetido a filtros e suavizações de ruídos.
Fluxos Básicos	<ol style="list-style-type: none"> 1. O caso de uso recebe o sinal sonoro sensoriado pelo microfone e preparado para análise pelo caso de uso de preparo do som. 2. Aplicação de filtros digitais e formas de redução de ruídos. 3. Envio do som para análise de potencialidade.

Fonte: Autoria própria

Tabela 33: Especificação do caso de uso 'Extrair Características'

Nome:	Extrair Características		
Descrição:	Extração de características do espectro/sinal sonoro. Fase para início da formação de uma <i>template</i> de características do som analisado.		
Ator Principal:	Microfone	Ator de Bastidor:	Sensor
Pré Condições	Sinal sonoro normalizado.		
Pós Condições	Gravação em banco de dados		
Regras de Negócio	Obter um conjunto de características sobre o espectro/sinal sonoro.		
Fluxos Básicos	<ol style="list-style-type: none"> 1. Recebimento do sinal sonoro já normalizado. 2. Aplicação de técnicas e cálculos de extração de características. 3. Envio do som para análise final de potencialidade. 		

Fonte: Autoria própria

APÊNDICE B – PONTOS POR CASOS DE USO

Gerenciamento de Requisitos com Pontos por Casos de Uso.

Casos de Uso por função ou do inglês *Use Case Points* (UCP) é uma técnica que auxilia os projetistas a determinar o tamanho real de um projeto ou saber quantas horas seriam necessárias para concluir o projeto. Neste trabalho, para desenvolver o estudo, tomemos como modelo o processo de desenvolvimento de *software* unificado ou RUP, também conhecido como incremental e iterativo. Ainda com o RUP, utilizaremos como arcabouço o framework Pinhão da CELEPAR e o documento Guia para Estimativa do Projeto (CELEPAR, 2009). Alguns aspectos foram alterados para definir melhor um projeto de sistemas embarcados. Portanto, para fins de melhores cálculos e resultados as tabelas foram adequadas de acordo com a experiência de projetos embarcados do autor e de projetos anteriores na área.

Inicia-se, portanto, fazendo o cálculo do UUCP ou Pontos por caso de Uso Desajustados:

$$\text{UUCP} = \text{UAW} + \text{UUCW} \quad \text{Eq. 15}$$

UNADJUSTED ACTOR WEIGHT OU UAW

A classificação dos atores é feita seguindo a tabela abaixo (Tabela 34), classificando assim os pesos de cada um e ao final realizada a soma, ainda não ajustada (Tabela 35).

Tabela 34: Descrição dos tipos dos atores

Tipo de Ator	Peso	Descrição
Ator Simples	1	Uma ação externa pura (vinda pelo sensor) ou interna (efetuada pelo atuador) interage com o sistema e precisa ser reconhecida. Ou ainda processamentos simples, visualização de informações etc..

Tipo de Ator	Peso	Descrição
Ator Médio	2	Ações gerais que interagem com a aplicação principal, tais como gravação em memória, rotinas de processamento especiais, interrupções, <i>timers</i> , PWM.
Ator Complexo	3	Outro componente interagindo através de um protocolo de comunicação (ex GPS, SPI, RS232, USB, I2S) ou externamente ao equipamento.

Fonte: Autoria própria

Tabela 35: Soma desajustada dos atores

Tipo de Ator	Peso	Nº de Atores	Resultado
Ator Simples	1	3	3
Ator Médio	2	1	2
Ator Complexo	3	3	9
Total de UAW			14

Fonte: Autoria própria

Atores Simples: Atuador, Sensor, microfone.

Atores Médios: Banco de Dado.

Atores Complexos: Dispositivo Comunicação, USB *Device*, *Ethernet*

UNADJUSTED USE CASE WEIGHT OU UUCW

Da mesma maneira que os atores, os casos de uso têm suas especificações na Tabela 36 chamada de “Tabela Complexidade Casos de Uso” e atribuídos ao sistema em questão com os respectivos pesos somados na Tabela 37.

Tabela 36: Tabela complexidade casos de uso

Complexidade	Peso	Descrição
Caso de Uso Simples	5	Se há até 4 transações, incluindo também o fluxo alternativo. Deve ser possível efetuar o caso de uso acessando menos de 5 objetos (instancia de uma classe) ou até 10 acessos a memória.

Complexidade	Peso	Descrição
Caso de Uso Médio	10	Se há de 5 a 7 transações, incluindo também o fluxo alternativo. Deve ser possível efetuar o caso de uso acessando menos de 5 objetos (instancia de uma classe) ou de 11 a 20 acessos a memória.
Caso de Uso Complexo	15	Se há mais de 7 transações, incluindo também o fluxo alternativo. Deve ser possível efetuar o caso de uso acessando menos de 5 objetos (instancia de uma classe) ou mais de 20 acessos a memória .

Fonte: Autoria própria

Tabela 37: Soma desajustada dos casos de uso

Complexidade	Peso	Nº de UC	Resultado
Caso de Uso Simples	5	3	15
Caso de Uso Médio	10	3	30
Caso de Uso Complexo	15	1	15
Total de UUCW			60

Fonte: Autoria própria

Simples: Filtrar, Extrair Características e Formatar Saída.

Médios: Gerir Comunicação, Gerir Variáveis, Preparar Som.

Complexos: Processar Som.

UNADJUSTED USE CASE POINT

Soma do UAW com UUCW

$$\mathbf{UUCP = 14 + 60 = 74}$$

CÁLCULO FATORES TÉCNICOS E DE AMBIENTE

FATORES TÉCNICOS

O fator de Complexidade técnica ou TCF baseado em operações que envolvem a multiplicação do valor dado de 0 a 5 a um fator técnico multiplicado ao seu peso tabelado. Ao final é realizada a soma destes fatores e o TCF é encontrado com é demonstrado na Tabela 38 e na fórmula a seguir.

$$\text{TCF} = 0,6 + (0,01 * \text{Tfactor})$$

Eq. 16

$$\text{TCF} = 1,035$$

Tabela 38: Fatores técnicos

Fator	Fatores que Contribuem para a Complexidade	Peso	Valor	Total
F1	Sistemas Distribuídos	2	0	0
F2	Tempo de Resposta	1	3	3
F3	Eficiência para o usuário final (on-line)	1	5	5
F4	Processamento Interno Complexo	1	5	5
F5	Código Reutilizável	1	3	3
F6	Facilidade de Instalação	0,5	3	1,5
F7	Facilidade de Uso	0,5	4	2
F8	Portabilidade	2	5	10
F9	Facilidade de mudança	1	4	4
F10	Concorrência (acesso simultâneo à aplicação)	1	3	3
F11	Recursos de Segurança	1	1	1
F12	Fornece acesso direto para terceiros	1	4	4
F13	Requer treinamento especial para o usuário	1	2	2
TFactor				43,5

Fonte: Autoria própria

FATORES DE AMBIENTE

O fator de Complexidade de ambiente ou EF baseado em operações que envolvem a multiplicação do valor dado de 0 a 5 a um fator de ambiente multiplicado ao seu peso tabelado. Ao final é realizada a soma destes fatores e o EF é encontrado com é demonstrado na Tabela 39 e na fórmula a seguir.

$$EF = 1,4 + (-0,03 * Efactor) \quad \text{Eq. 17}$$

$$EF = 0,71$$

Tabela 39: Fatores de ambiente

Fator	Fatores que Contribuem para a Complexidade	Peso	Peso	Total
F1	Familiaridade da Equipe com o processo formal de desenvolvimento adotado	1,5	4	6
F2	Colaboradores de meio-período	-1	1	-1
F3	Capacidade do líder do projeto em análise de requisitos e modelagem	0,5	4	2
F4	Experiência da equipe em desenvolvimento de aplicações do gênero em questão	0,5	4	2
F5	Experiência em OO	1	4	4
F6	Motivação da equipe	1	4	4
F7	Dificuldade com a linguagem de programação	-1	2	-2
F8	Requisitos estáveis	2	4	8
EFactor				23

Fonte: Autoria própria

USE CASE POINT

Neste passo é feita a soma do valor total do sistema o *Use Case Point* ou UCP é a representado pela fórmula:

$$UCP = UUCP * TCF * EF \quad \text{Eq. 18}$$

$$UCP = 74 * 1,035 * 0,71 = 54,37$$

Agora segundo Karner (1993), utilizando como base 20 horas de trabalho por ponto de caso de Uso (UCP), tem-se um total de:

$$\text{Tempo Estimado} = \text{UCP} * 20$$

Eq. 19

$$\text{Tempo Estimado} = 1087,6 \text{ horas.}$$