UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

MUSIC EMOTIONS INTEL: IDENTIFICADOR AUTOMÁTICO DE EMOÇÕES EM MÚSICAS

THOMAS OELKE ADRIANO

BLUMENAU
2017

THOMAS OELKE ADRIANO

MUSIC EMOTIONS INTEL: IDENTIFICADOR AUTOMÁTICO DE EMOÇÕES EM MÚSICAS

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof(a). Andreza Sartori, Doutora - Orientadora

BLUMENAU
2017

MUSIC EMOTIONS INTEL: IDENTIFICADOR AUTOMÁTICO DE EMOÇÕES EM MÚSICAS

Por

THOMAS OELKE ADRIANO

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente:	Prof(a). Andreza Sartori, Doutora – Orientador, FURB
Membro:	Prof(a). Daniel Theisges, Mestre – FURB
Membro:	Prof(a). Roberto Heinzle, Doutor – FURB

Blumenau, 7 de julho de 2017

AGRADECIMENTOS

Agradeço a minha namorada, Jakeline Michele Pelin, por todo o apoio e incentivo dado ao decorrer do desenvolvimento deste trabalho.

À minha orientadora, Andreza Sartori, que teve extrema paciência e dedicação em todos os aspectos necessários para a conclusão deste trabalho.

RESUMO

Este trabalho apresenta o desenvolvimento de uma nova abordagem para a tarefa de identificação de emoções em músicas. Enquanto os métodos tradicionais de identificação de emoções em músicas se resumem em identificar uma emoção por música, este trabalho propõe, buscando uma maior assertividade, identificar um conjunto de emoções por música. Para atingir este objetivo, foi utilizada como base de treinamento o *dataset* 1000 Songs for Emotional Analisys of Music de Mohammad (2013), como conjunto de características de áudio foram utilizados Mel Frequency Cepstral Coefficients (MFCC), Spectral Centroid (SC), Zero Crossing Rate (ZCR), chromagram e tempogram, e, como algoritmo regressor o Support Vector Regression (SVR) com kernel Radial Basis Function (RBF). Como método de mapeamento emocional, foi utilizado o plano circumplexo de Russel (1980). O conjunto de características e o algoritmo regressor foram escolhidos com base no trabalho de Huq (2010). O resultado obtido com este trabalho foi mensurado tanto qualitativamente quanto quantitativamente. Com a análise qualitativa, pode-se observar que a representação de emoções utilizando um conjunto de emoções por música se adere melhor a realidade das músicas atuais do que a representação de apenas uma emoção por música. Para a análise quantitativa, fez-se uma comparação entre um valor mínimo de assertividade esperada com a assertividade obtida pela abordagem proposta por este trabalho. O valor mínimo de assertividade, isto é, o *baseline*, foi definido pelo Mean Absolute Error (MAE) entre os valores de Valência e Alerta esperados e a média deles. O resultado quantitativo mostrou espaço para melhorias, ficando abaixo do esperado.

Palavras-chave: Obtenção de informações de músicas. Reconhecimento de emoções em músicas. Regressão de vetor de suporte.

ABSTRACT

This work presents the development of a new approach to the task of identifying emotions in music. While the traditional methods of emotion recognition in music are focused in identifying a singular emotion per music, this work proposes to recognize a set of emotions by music. To this aim, as training set, it was used the dataset of Mohammad (2013), 1000 Songs for Emotional Analyzes of Music. To extract the audio characteristics was used the Mel Frequency Cepstral Coefficients (MFCC), the Spectral Centroid (SC), the Zero Crossing Rate (ZCR), the chromagram e the tempogram. A Support Vector Regression (SVR) with RBF kernel was applied to learn the algorithm. To map emotionally the emotions the circumplex model of affect of Russel (1980) was used. The set of characteristics and the regression algorithm were chosen based on the work of Huq (2010). The result obtained with this work was measured both qualitatively and quantitatively. With the qualitative analysis, it could be observed that a representation of emotions using a set of emotions by music adheres better to the reality of the current music than a representation of only one emotion per music. For a quantitative analysis, a comparison was made between the minimum value of the expected accuracy with the result obtained by the proposed approach. The minimum accuracy value, which is, the baseline, was defined by the Mean Absolute Error (MAE) between the expected Valence and Alert values and their mean. The quantitative results were not as good as expected, however showed opportunity for improvements.

Key-words: Music information retrieval. Music emotion recognition. Support vector regression.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BPM – Beats Per Minute

Hz – Hertz

kHz – Kilo Hertz

MAE – Mean Absolute Error

MER – Music Emotion Recognition

MFCC – Mel Frequency Cepstral Coefficients

MIR – Muisc Information Retrieval

MSE – Mean Squared Error

PCA – Principal Component Analisys

RBF – Radial Basis Funciton

RMS – Root Mean Squared

SC – Spectral Centroid

SVR – Support Vector Regression

VA – Valence e Arousal

ZCR – Zero Crossing Rate

SUMÁRIO

1 INTRODUÇÃO

Seres humanos interpretam música emocionalmente (HEVNER, 1936). Apesar de toda a infraestrutura técnica necessária para compor uma música, como, por exemplo, escala, tempo e tom, o produto final é a emoção.

Os trabalhos mais recentes na área abordam a tarefa de Music Emotion Recognition (MER), campo de estudos destinado a identificação de emoções de músicas, tendo como resultado uma emoção por música (KIM, 2010). Para executar a identificação de emoções, alguns trabalhos utilizam um trecho da música escolhido aleatoriamente, outros utilizam um trecho escolhido por algoritmos de Music Information Retrieval (MIR), campo de estudos destinado a extração de informações de músicas, buscando as partes mais expressivas da música. Algumas abordagens utilizam trechos de 15 segundos como entrada para a identificação de emoções, outros utilizam 30 segundos ou mais (KIM, 2010).

Todas estas abordagens possuem um mesmo problema: a perda de assertividade causada pela heterogeneidade de emoções presentes nas músicas. As abordagens de MER tradicionais utilizam um trecho da música como representante da música toda. Esta prática, dada a situação da heterogeneidade emocional, não contabiliza todas as outras possíveis emoções existentes em partes diferentes da música, distorcendo o resultado final (MEYER, 1997).

Meyer (1989), afirma que nos períodos musicais Renascentistas (1400 D.C até 1550 D.C) e Barroco (1550 D.C até 1650 D.C), as músicas continham um baixo contraste de emoções, muitas vezes possuíam uma mesma emoção. Desde o período musical clássico, que se iniciou por volta de 1700 D.C, percebe-se que as músicas possuem um maior contraste emocional. Atualmente, é comum músicas conterem múltiplas emoções: Fusion, Jazz e Progressivo são alguns exemplos de estilos musicais modernos que, no decorrer de uma mesma música possuem várias emoções que podem ser bastante contrastantes. Estas diferenças entre a quantidade de emoções contrastantes presentes em cada era musical se dão devido a gradual evolução na composição musical, onde na era renascentista, por exemplo, eram utilizados apenas uma escala em um mesmo tempo e em uma mesma entonação (MEYER, 1989).

Diante do exposto, este trabalho propõe o desenvolvimento de uma nova abordagem para categorização de músicas por emoção, buscando levar em consideração a diversidade emocional presente nas músicas. Para atingir este objetivo, foi desenvolvido um sistema para identificação de emoções de músicas capaz de identificar múltiplas emoções em uma mesma

música, evitando os problemas causados por identificar apenas uma emoção por música. Este sistema utilizou um algoritmo de regressão para predição das emoções, que foi treinado utilizando um *dataset* contendo mil músicas de diversos gêneros. Além disso, foi também utilizado uma biblioteca para MIR chamada Librosa, a qual possibilitou a implementação de rotinas de extração de características de músicas.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um algoritmo para a identificação de emoções evocadas nas músicas levando em consideração a variedade emocional presente nestas músicas.

Os objetivos específicos do trabalho são:

a) desenvolver um *script* de extração de características de áudio utilizando a biblioteca Librosa;
b) desenvolver algoritmo de identificação de emoções de músicas baseado em seu conteúdo emocional predominante, compreendendo a extensão total das músicas e levando em consideração todas as possíveis emoções presentes.

1.2 ESTRUTURA

A presente monografia está organizada em quatro capítulos: introdução, fundamentação teórica, desenvolvimento e conclusões. O capítulo 2 descreve a fundamentação teórica, na qual apresenta os conceitos básicos para este trabalho nas áreas de teoria musical, plano de emoções e as propriedades das ondas sonoras. Na sequência, o capítulo 3 descreve o desenvolvimento da solução proposta, que inclui os requisitos propostos, os diagramas de casos de usos e de classes, as ferramentas e técnicas utilizadas, assim como a operacionalidade da aplicação e a análise dos resultados obtidos. Por fim, o capítulo 4 descreve as conclusões sobre a pesquisa e apresenta as alternativas para o desenvolvimento de trabalhos futuros relacionados a este.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são relacionadas as bases teóricas utilizadas para atingir o objetivo deste trabalho. A seção 2.1 explica a correlação que existe entre as músicas e as emoções, a seção 2.2 apresenta a técnica utilizada para mapear emoções em músicas, e, por fim, a seção 2.3 expõe o conceito e as propriedades de ondas sonoras.

2.1 EMOÇÕES E MÚSICAS

As músicas são compostas por uma mistura de timbres, melodias, harmonias e ritmo. Timbres variam dependendo da combinação de instrumentos musicais envolvidos, considerando qualquer fonte sonora como instrumento. Enquanto as melodias variam em escalas e modos, harmonias variam em qualidade e ritmos que variam pela quantidade de batidas por minuto. Cada uma destas características é interpretada pelos seres humanos de uma forma, ativando determinadas emoções. (HEVNER, 1935; SCHOENBERG, 1999)

2.1.1 Emoção

O conceito de emoção representa um problema complexo. Apesar deste termo ser usado frequentemente, ao ponto de ser comum no dia a dia, a questão “o que é uma emoção?” raramente gera a mesma resposta de diferentes cientistas. Segundo Russel e Barret (1999), a psicologia provê conceitos empíricos sobre emoção, como, por exemplo, uma série de eventos complexos inter-relacionados, tendo como alvo um objeto específico, como uma pessoa ou um evento; ou uma coisa, seja passado, presente, futuro, real ou imaginado.

Segundo Izard (2013), a maioria das teorias, explicitamente ou implicitamente, admitem que emoções não são fenômenos simples. Emoções não podem ser descritas completamente por uma pessoa descrevendo suas experiências emocionais. Também não podem ser descritas completamente por medidas eletrofisiológicas de ocorrência no cérebro, sistema nervoso, sistema circulatório e respiratório, ou pelas expressões motoras. Uma definição completa de emoções precisa levar em consideração os seguintes aspectos:

a) a experiência ou sentimento consciente da emoção;
b) os processos que ocorrem no cérebro e no sistema nervoso;
c) as expressões observáveis das emoções, principalmente as que ocorrem na região facial.

Atualmente, cientistas que aplicam o reconhecimento emocional em pesquisas nas mais diversas áreas, geralmente utilizam uma das duas abordagens psicológicas mais populares para classificar as emoções: discretas e dimensionais Sartori (2015), que são descritas

respectivamente nas seções 2.1.2 e 2.1.3. Neste trabalho a emoção é medida utilizando o modelo dimensional de Russel (1980), que divide as emoções em um plano circumplexo utilizando conceitos de valência e alerta.

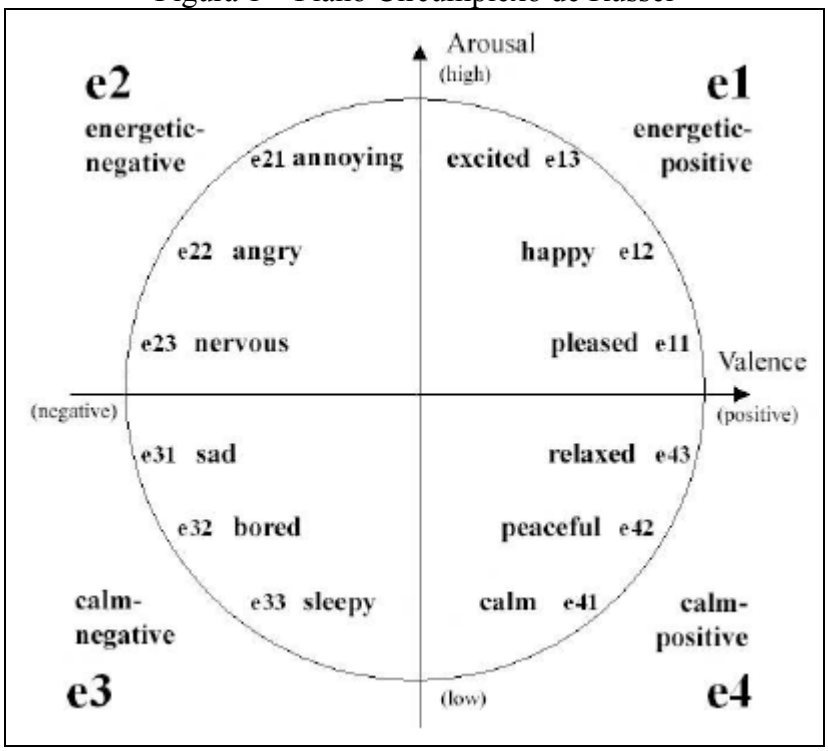
2.1.2 Representação das emoções: modelo discreto

Segundo Izard (1993), a teoria de emoções discretas afirma que existe um pequeno conjunto de emoções principais. Este conjunto de emoções principais são respostas emocionais biologicamente determinadas, cuja expressão e reconhecimento é fundamentalmente igual em todos os indivíduos, independente de etnia ou cultura. Tomking (1962) conclui que existem oito principais emoções: surpresa, interesse, alegria, raiva, medo, desgosto, vergonha e angústia. Mais recentemente, Izard (1993) delineou doze emoções discretas, rotuladas como: interesse, alegria, surpresa, tristeza, raiva, disposto, desprezo, hostilidade, medo, vergonha, timidez e culpa.

2.1.3 Representação das emoções: modelo dimensional

No modelo dimensional as emoções são expressadas com números utilizando coordenadas do plano circumplexo de Russel (1980), conhecido como plano circumplexo de Valência e Alerta (VA - Valence e Arousal). A Figura 1 representa este plano, onde o eixo Y é responsável por mapear o nível de alerta e o eixo X responsável por mapear a valência.

Figura 1 – Plano Circumplexo de Russel



Fonte: Appalachian State University (2017).

A valência pode ser entendido como o medidor de alegria ou tristeza: quanto mais próximo ao ponto extremo negativo de valência (a esquerda), mais próximo das emoções triste (*sad*) e nervoso (*nervous*). Em contrapartida, quanto mais próximo do ponto máximo de valência, mais próximo das emoções relaxado (*relaxed*) e contente (*pleased*). O Alerta, pode ser entendido como o medidor de energia: quanto mais próximo do valor mínimo, mais próximo das emoções sonolento (*sleepy*) e calmo (*calm*). Quando o Alerta está se aproximando do valor máximo, mais próximo está das emoções irritante (annoying) e excitado (excited) (RUSSEL, 1980).

2.1.4 Notas musicais

A escola de música ocidental utiliza um sistema de intervalos para mapear as frequências sonoras, que se traduzem em tons. Este sistema é composto por 12 intervalos sendo eles C (dó), C#, D (ré), D#, E (mi), F (fá), F#, G (sol), G#, A (lá), A#, B (si), onde as notas acompanhadas pelo símbolo "#" recebem o mesmo nome da respectiva nota acompanhado do termo "sustenido" (SCHMELING, 2011). Intervalos compostos por um sustenido são chamados de semitom e intervalos de 2 sustenidos são chamados de tom. O intervalo de C à C#, por exemplo, é um semitom, pois contém apenas um sustenido; o intervalo de C à D é um tom, pois contém 2 sustenidos. Para poder ser possível representar todo o espectro de frequências possíveis com tons é utilizado o conceito de oitavas (PROCOPIO, 2016).

Oitavas são intervalos formados por 8 tons, uma vez estes 8 tons atingidos, a sequência de notas é repetida em uma oitava diferente. O intervalo C-D-E-F-G-A-B-C é uma oitava pois contém 8 tons (de C à C), a nota que segue repete esta sequência em uma oitava acima, sendo possível adicionar um número seguindo a nota para definir a oitava em que esta nota se encontra: C0-D0-E0-F0-G0-A0-B0-C1-D1 (SCHOENBERG, 1999).

Tons são frequências, e a cada oitava o intervalo de frequências dobra, refletindo o fato de que a percepção auditiva humana percebe alterações de frequências baixas com mais intensidade que alterações em frequências altas. Desta forma C0 equivale a 16.45Hz e o intervalo entre C0 e D0 é de 2Hz, uma mudança de 2Hz na oitava 0 já caracteriza um tom. C1 equivale a 32.90Hz e o intervalo entre C1 e D1 é de 4Hz, nesta oitava um tom é percebido com 4Hz de diferença. C2 equivale a 65.80Hz e o intervalo entre C2 e D2 é de 8Hz, e assim por diante, tendo como limite prático a frequência máxima reconhecida pelo ouvido humano, que é, em média, 20kHz. (SCHMIDT; TURNBULL; KIM, 2010)

2.1.5 Tom

Músicas tonais são músicas construídas tendo um ou mais tons principais. Todas as vibrações sonoras possuem uma frequência de maior importância, onde o tom de uma música se traduz como a frequência de maior importância, isto é, a frequência mais notada pelo ouvinte. Uma música escrita na entonação de C (dó) não implica que apenas esta nota exista na música, mas significa que todas as notas utilizadas pertencem a escala melódica e harmônica de C (dó). Uma mesma música pode trocar de tom, este evento se chama modulação. Isto é, uma música pode iniciar em C (dó), passar para D (ré) e voltar para C (dó) ou modular para outro tom. (SCHOENBERG, 1999).

2.1.6 Melodia

A Melodia pode ser definida como uma sucessão de notas sequenciais que são tocadas separadamente sobre o ritmo. Este conjunto de notas é organizado em escalas, as quais dão um tema à música e carregam uma informação emocional distinta. Existem dezenas de tipos de escalas, cada qual reúne uma combinação de notas, existindo duas escalas principais, das quais todas as outras originam: a escala maior e a escala menor. Escalas maiores evocam emoções relacionadas a alegria, enquanto escalas menores evocam emoções relacionadas a tristeza (HEVNER, 1935; SCHOENBERG, 1999).

2.1.7 Harmonia

Harmonia é o som de duas ou mais notas tocadas ao mesmo tempo, em contraste com a melodia, que é apenas uma nota tocada por vez. A harmonia, assim como a melodia, é construída com escalas, e, adicionalmente, por qualidades harmônicas: a melodia define o tom, e as qualidades harmônicas definem quais notas são tocadas simultaneamente, sendo as principais qualidades harmônicas a maior e a menor, evocando respectivamente as emoções de alegria e tristeza (HEVNER, 1935; SCHOENBERG, 1999).

2.1.8 Ritmo

Ritmo é a fundação da música, onde as evidências mais antigas sobre composição musical apontam que as músicas eram formadas apenas por ritmo. O ritmo é responsável por dar o andamento da música, sendo medido pela unidade BPM (*Beats Per Minute*), que mede a quantidade de batidas por minuto. O estudo de Bispham (2006) sugere que ritmos acelerados evocam emoções de alerta, enquanto ritmos reduzidos evocam emoções de calma.

2.1.9 Timbre

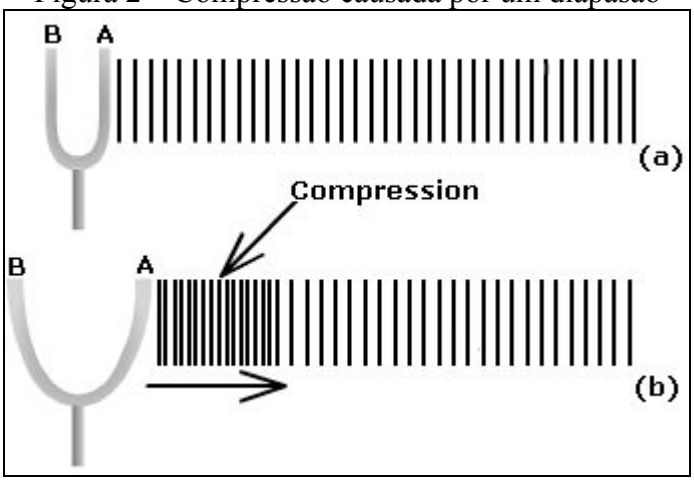
O som de um violão e um cavaquinho tocando uma mesma nota de forma idêntica, com uma perfeita sincronia em velocidade e intensidade, ainda assim produzem um som diferente. Esta característica de diferença é chamada de timbre. O timbre é definido como o conjunto de frequências gerado por uma determinada fonte sonora. No exemplo do cavaquinho e violão, ambos apresentam diferença no som, apesar de estarem sendo executados de forma perfeitamente iguais devido ao conjunto de frequências que estes respectivos instrumentos produzem (MENON, 2002). Ao contrário dos aspectos de ritmo, harmonia e melodia, o timbre não está diretamente ligado a emoções. O timbre não é encontrado isoladamente em músicas, mas em conjunto com harmonia, melodia e ritmo. Sua interpretação emocional depende da combinação destes aspectos (SCHMELING, 2011).

2.2 ONDAS SONORAS

De acordo com Tom Henderson, (2017), as ondas, sonoras e mecânicas, podem ser definidas como um distúrbio que viaja através de um meio, transportando energia de um lugar para outro. O meio é apenas o material pelo qual o distúrbio se move, podendo ser pensado como uma série de partículas interconectadas que interagem entre si. No caso das ondas sonoras, o meio de propagação mais comum é o ar, mas a propagação pode ocorrer em outros meios como metal ou água. Um diapasão (objeto com formato de um garfo, capaz de produzir ondas sonoras), vibrando é capaz de gerar uma onda sonora: quando os dentes do diapasão vão para fora, as partículas de ar adjacentes são empurradas, quando os dentes do diapasão se retraem, uma área de pressão baixa é criada, fazendo com que as partículas adjacentes de ar se movam para trás. Devido ao movimento longitudinal destas partículas, existem regiões no ar onde as partículas estarão comprimidas e outras regiões onde elas estarão separadas. Estas regiões são conhecidas respectivamente como compressão e rarefação: compressão são regiões com alta pressão de ar, enquanto rarefação são regiões com baixa pressão de ar (BERG, 1982).

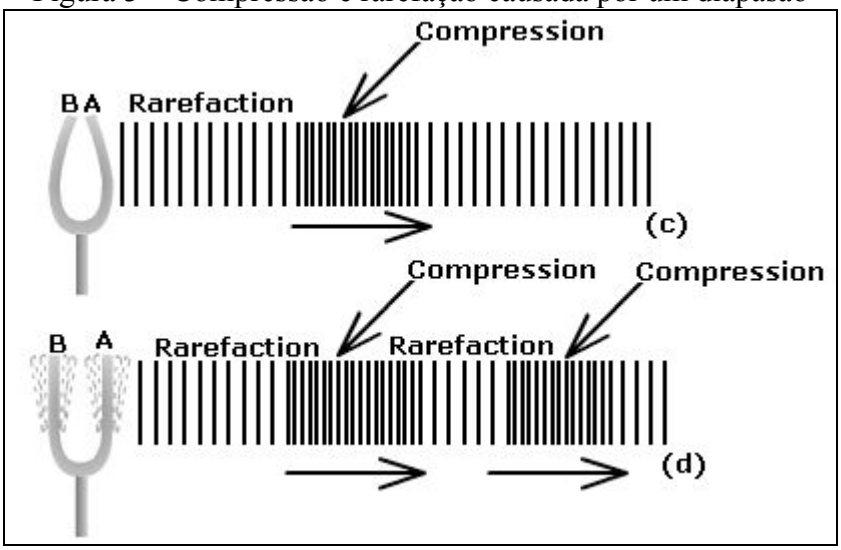
A Figura 2 e Figura 3 demonstram o exemplo do diapasão e o efeito de compressão (*compression*) e rarefação (*rarefaction*) causado pelo movimento dos seus dentes. A Figura 2(a) apresenta os dentes parados e as partículas de ar inertes e na Figura 2(b) mostra os dentes se abrindo, empurrando as partículas de ar e provocando a compressão. Na Figura 3(c) os dentes estão retornando, criando uma área de baixa pressão e provocando a rarefação, e na Figura 3(d) é exemplificado a movimentação contínua, gerando várias áreas de compressão e rarefação.

Figura 2 – Compressão causada por um diapasão



Fonte: TutorVista (2017).

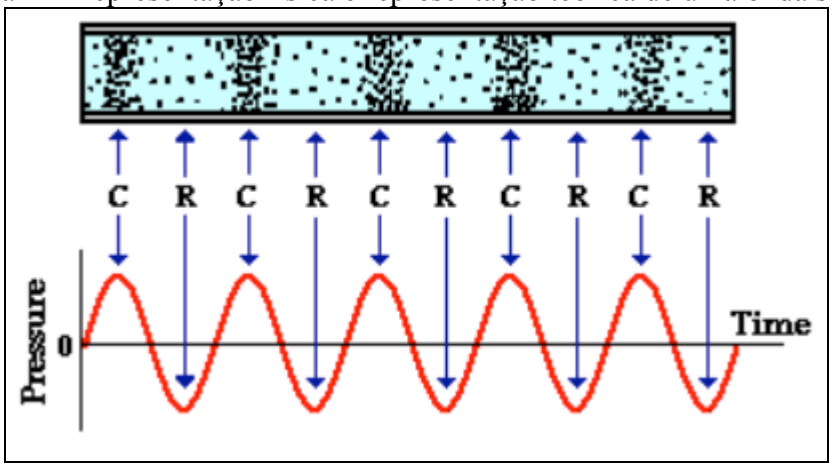
Figura 3 – Compressão e rarefação causada por um diapasão



Fonte: TutorVista (2017).

A Figura 4 demonstra um comparativo entre a representação por movimentação de partículas de ar e a representação senoidal das flutuações de pressão de ar. As letras C e R representam, respectivamente, compressão e rarefação; o termo *Pressure* significa pressão e o termo *Time* significa tempo. A onda senoidal (em vermelho) é a forma padrão utilizada para se representar ondas sonoras em dispositivos ou softwares que permitem visualizá-las.

Figura 4 – Representação física e representação teórica de uma onda sonora

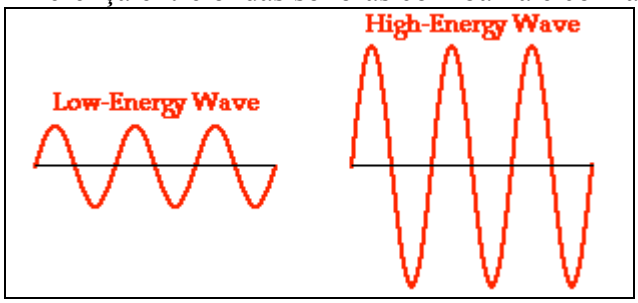


Fonte: Tom Henderson (2017).

2.2.1 Energia e Amplitude

A amplitude refere-se à intensidade com que as partículas são movimentadas de sua posição de descanso, enquanto o termo energia se refere a quantidade de energia carregada pelas partículas perturbadas, que é correlacionada com a amplitude, isto é, quanto maior a amplitude, maior a energia e vice-versa. A Figura 5 exibe ondas sonoras com diferentes amplitudes e, por sua vez, energias A onda nomeada como *low-energy wave* é detentora de menos energia e amplitude e a onda nomeada como *high-energy wave* é detentora de mais energia e amplitude. (TOM HENDERSON, 2017)

Figura 5 – Diferença entre ondas sonoras com baixa e com alta energia



Fonte: Tom Henderson (2017)

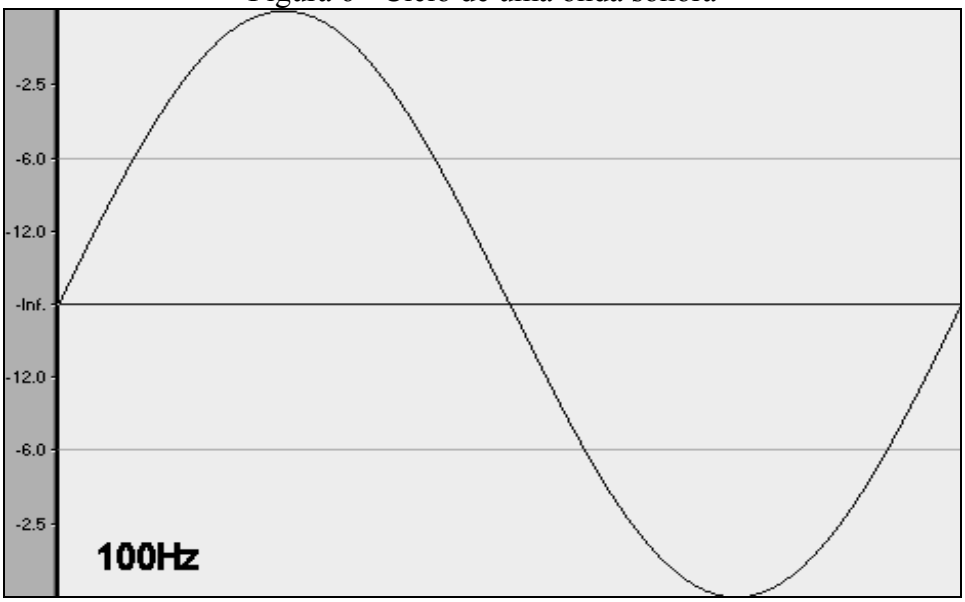
O efeito que a amplitude tem no som é interpretado pelo ouvido humano como "volume alto" ou "volume baixo". A amplitude (e consequentemente a energia) não influencia na frequência ou no timbre. (KINSLER, 1999)

2.2.2 Frequência e Tom

A frequência mede a quantidade de vezes que uma partícula vibra quando uma onda passa por elas. Pode ser mensurada pela quantidade de vezes em que uma partícula vibra para frente e para trás (compressão e rarefação) em uma medida de tempo. Esta movimentação de partículas é chamada de ciclo, sendo definido como um movimento iniciando na posição zero,

indo para compressão, passando à posição zero novamente, indo para rarefação e voltando à posição zero (TOM HENDERSON, 2017). A Figura 6 demonstra um ciclo de uma onda senoidal, onde o valor "-inf" representa o valor zero (ponto de descanso) da onda.

Figura 6 - Ciclo de uma onda sonora



Fonte: Tom Henderson (2017).

A frequência é medida em Hertz e mede a quantidade de vezes em que um ciclo ocorre em um segundo. A nota "Lá", por exemplo, é composta por uma onda sonora em uma frequência de 440Hz (440 ciclos por segundo). A frequência de uma onda não tem relação com amplitude ou timbre, sua influência é percebida pelo ouvido humano como tom. No meio musical da cultura ocidental os tons são organizados em 12 notas, sendo elas C, C#, D, D#, E, F, F#, G, G#, A, A#, B. (BERG, 1982)

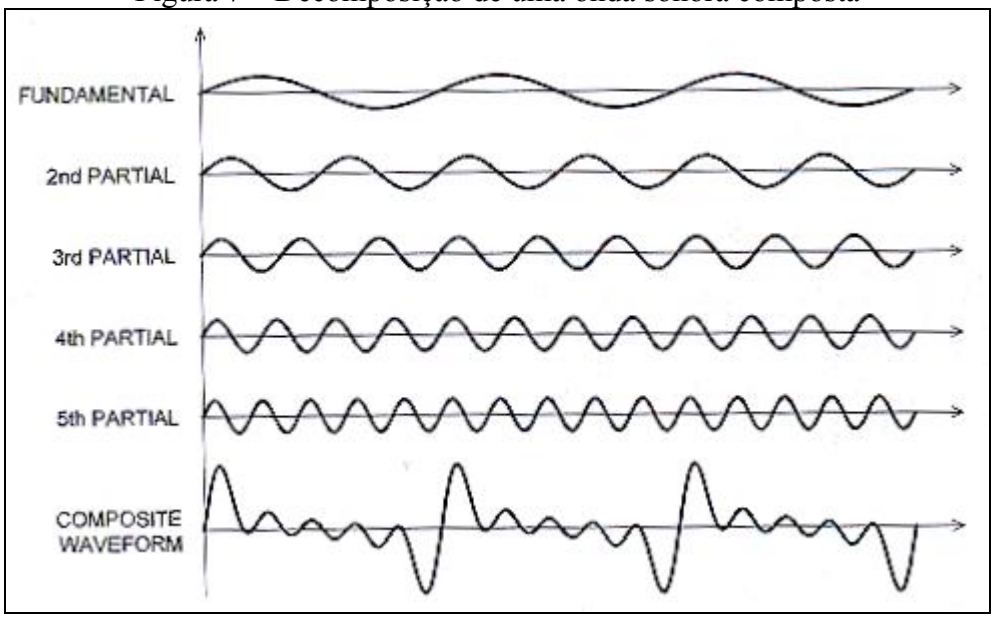
2.2.3 Frequência e Timbre

A onda senoidal exibida na Figura 4 demonstra uma onda sonora simples, composta por apenas uma frequência. Este tipo de onda não pode ser encontrado na natureza porque todos os sons produzidos são complexos, compostos por várias frequências. (KINSLER, 1999)

A Figura 7 demonstra a composição de uma onda sonora complexa, que são formadas por uma frequência fundamental e frequências secundárias, chamadas parciais. A frequência fundamental é responsável por ditar o tom em que a onda se encontra, enquanto as parciais adicionam frequências com menos amplitude à frequência fundamental, contribuindo para o timbre da onda. A ilustração com legenda *Composite Waveform* exemplifica o resultado de se somar a onda com descrição *Fundamental* às parciais 2, 3, 4 e 5. Uma onda sonora natural,

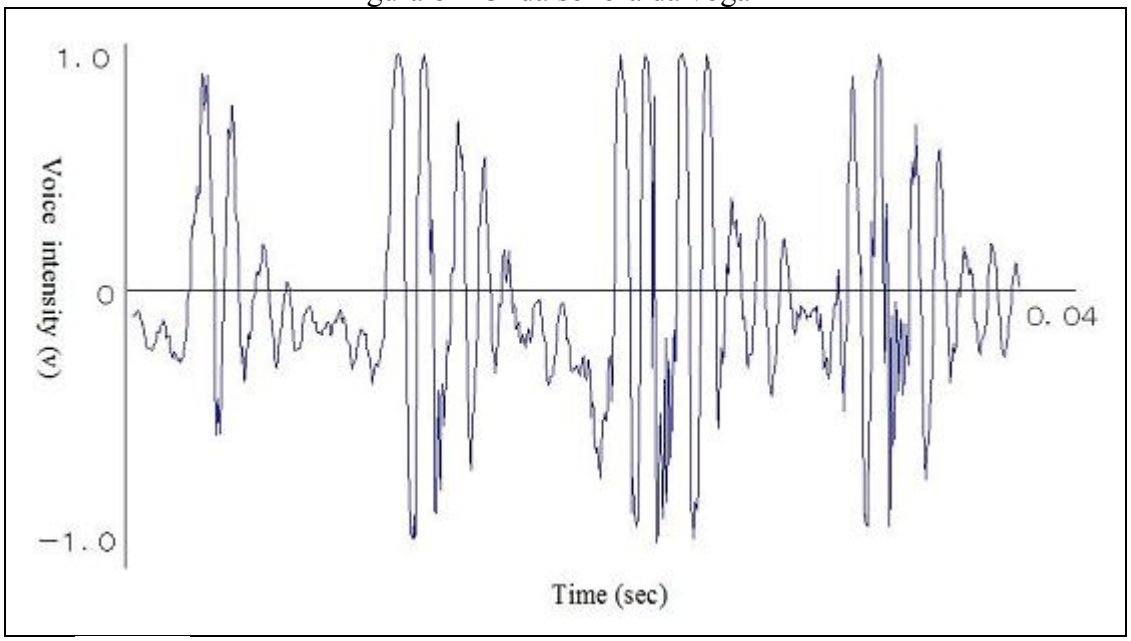
como a formada pela fala, por exemplo, envolve muitas frequências parciais adicionais e resulta em uma onda mais complexa (KINSLER, 1999). A Figura 8 exibe um fragmento da onda sonora formada pela vogal "A", a qual é uma onda composta.

Figura 7 – Decomposição de uma onda sonora composta



Fonte: Appalachian State University (2017)

Figura 8 – Onda sonora da vogal A

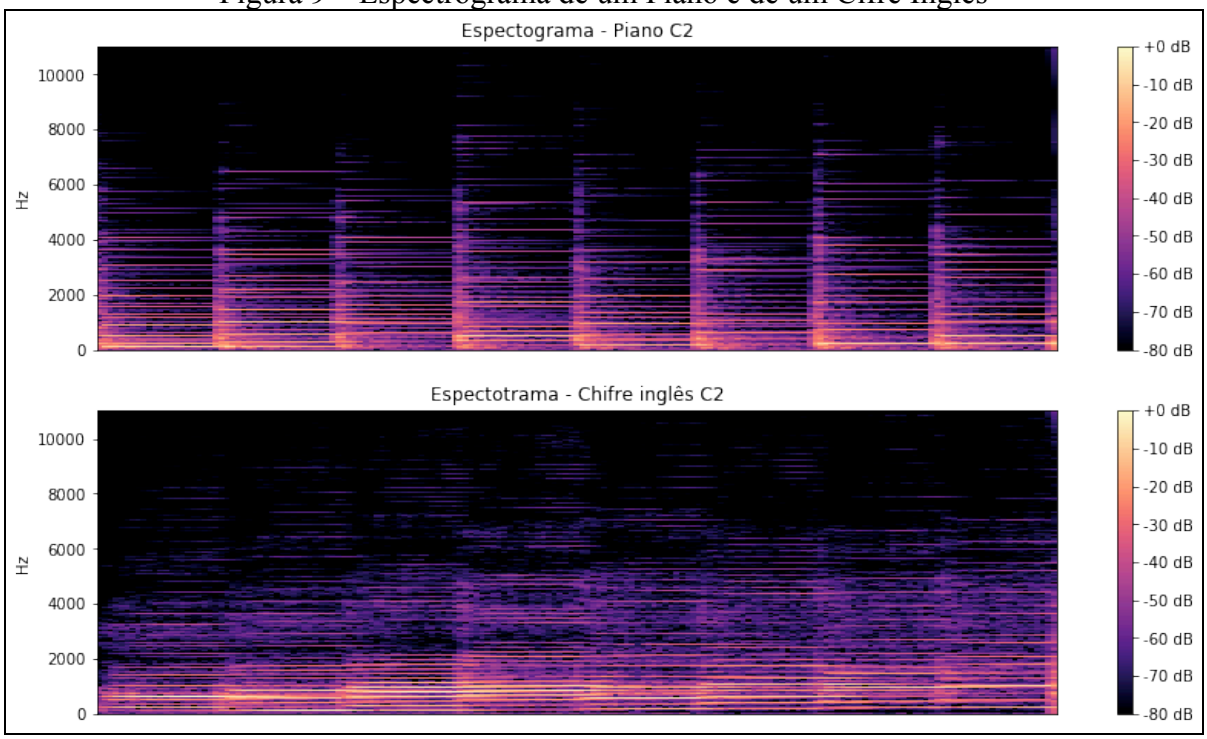


Fonte: Nakayama (2013)

Na Figura 9 é exibido o espectrograma de energia, que expõem quais frequências ocorrem ao longo do tempo, da escala de C (dó maior) iniciada em C2, tocada em um piano e em um chifre inglês. As cores representam a amplitude medida em decibéis (dB) e a posição no eixo Y representa a frequência. Nota-se que em nenhuma das notas apenas uma frequência recebe toda a energia, mas um conjunto de frequências. A frequência com mais energia é a

fundamental e também a mais notada dentre as outras pelo ouvido humano, definindo o tom do som. Além disso, apesar dos dois instrumentos estarem reproduzindo a mesma escala no mesmo tempo, as duas figuras são diferentes. Isso é devido ao conjunto de frequências que cada instrumento produz, este conjunto de frequências compõem o timbre. Em outras palavras, os gráficos retratam a diferença no timbre de cada um dos instrumentos.

Figura 9 – Espectrograma de um Piano e de um Cifre Inglês



Fonte: elaborado pelo autor.

2.3 CARACTERÍSTICAS ACÚSTICAS

Para realizar o reconhecimento de emoções em músicas utilizando apenas seu conteúdo sonoro (excluindo letra ou contexto cultural), é necessário utilizar características que consigam descrever o conteúdo acústico voltado para detecção de emoções da forma mais eficaz possível, que podem ser: timbral, harmônica e rítmica (SCHMIDT; TUNRBULL; KIM, 2010). As características extraídas são:

a) Timbre:
    a. Mel Frequency Cepstral Coefficients (MFCC);
    b. Spectral Centroid (SC);
    c. Zero Crossing Rate (ZCR);
b) Harmonia:
    a. Chromagram;
c) Tempo:

a. Tempogram.

2.3.1 Timbral

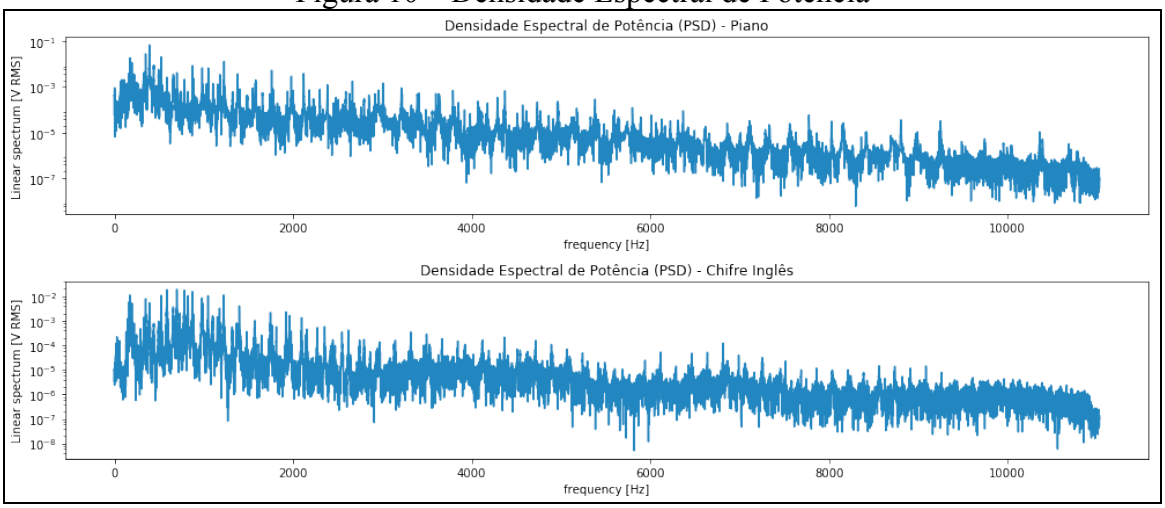
Esta seção apresenta as características de áudio extraídas para a realização deste trabalho relacionadas ao timbre.

2.3.1.1 Mel Frequency Cepstral Coefficient (MFCC)

O Mel Frequency Cepstral Coefficient (MFCC), é utilizado principalmente em sistemas de detecção de voz e instrumentos (HASAN et al, 2004), sendo definido como uma descrição concisa do formato geral de um envelope espectral. Um cepstrum (de Mel Frequency Cepstrum Coefficients) é uma representação normalizada e focada no timbre de um sinal de áudio (COOK, 2016). O termo Mel se refere a uma escala de frequências, demonstrada na Figura 12, voltada a audição humana, a qual reconhece com menos evidência as mudanças nas frequências conforme elas aumentam. Em resumo, o MFCC permite representar o timbre de um sinal de áudio da forma mais eficaz possível dentro do contexto de percepção humana (HUQ, 2010).

Para compreender o MFCC é necessário entender o envelope espectral, e para entender o envelope espectral é necessário compreender a densidade espectral de potência. A densidade espectral de potência mede a distribuição de poder nas frequências do sinal (TOM HENDERSON, 2017). Na Figura 10 é exibida a densidade espectral de potência referente a escala em C (iniciando em C2) dos instrumentos piano e chifre inglês. Ambos gráficos demonstram sinais de áudio com uma concentração maior de energia entre 0Hz e 6kHz, sendo o eixo X responsável por medir a potência da frequência em RMS (Root Mean Square, utilizada para medir a intensidade de sinais elétricos) e o eixo Y responsável por medir a frequência do áudio.

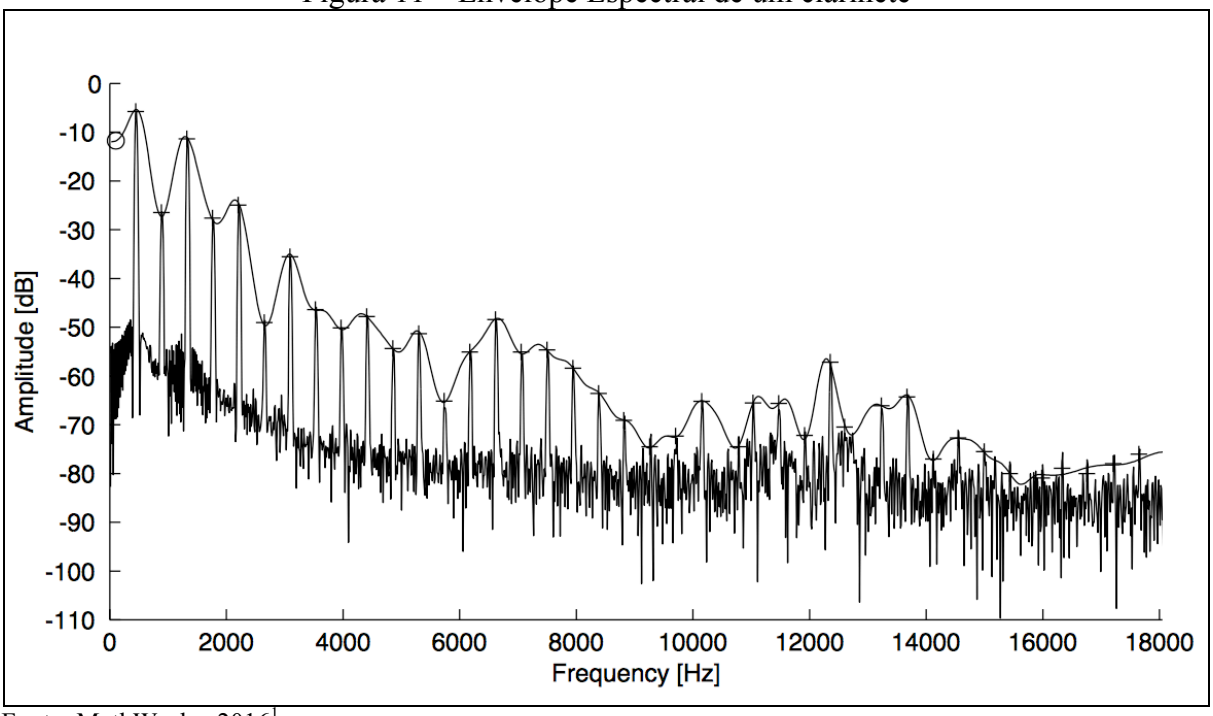
Figura 10 – Densidade Espectral de Potência



Fonte: Elaborado pelo autor.

O envelope espectral descreve a densidade espectral de potência de uma forma mais resumida, mantendo apenas uma representação do espectro por janela (KINSLE, 1999). A Figura 11 demonstra o envelope espectral de um clarinete. Nota-se que a informação apresentada pela Figura 11 é semelhante a informação apresentada pela Figura 10, isso se deve ao fato de que, em ambas as figuras, o mesmo sinal está sendo medido, isto é, a densidade espectral de potência. A diferença é que, na Figura 11 é exibido o envelope espectral, que é representado pelos símbolos de cruz juntamente com as linhas que os ligam. Cada cruz dessa representa uma janela do sinal de densidade espectral de potência. Para cada janela o ponto de maior amplitude é selecionado, gerando assim uma versão resumida do sinal de densidade espectral de potência.

Figura 11 – Envelope Espectral de um clarinete



Fonte: MathWorks, 2016[1]

Figura 12 – Relação de frequências com unidades Mel

Hz	20	160	394	670	1000	1420	1900	2450	3120	4000	5100	6600	9000	14000
mel	0	250	500	750	1000	1250	1500	1750	2000	2250	2500	2750	3000	3250

Fonte: MuEngineers, 2010[2]

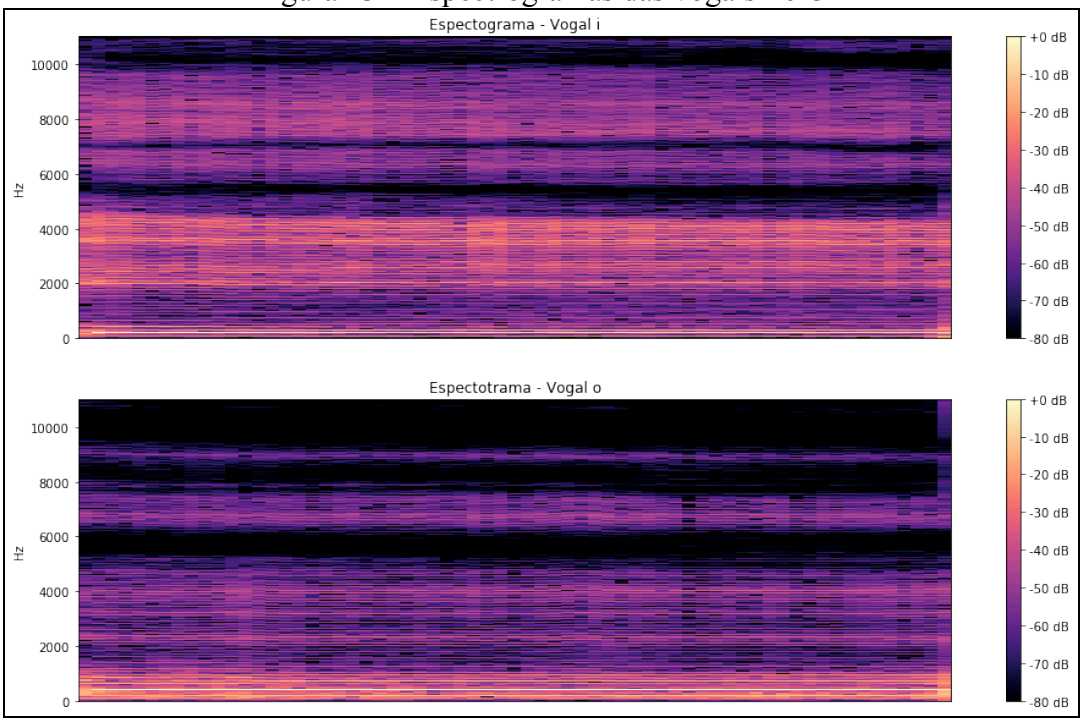
Tendo como objetivo a representação performática de timbre de um sinal de áudio (no contexto de MIR), é necessário buscar a maior representatividade timbral no menor vetor possível. O MFCC provê a versão mais resumida da informação timbral do áudio por descrever o formato geral do envelope espectral. O envelope espectral (Figura 11) permite ter a mesma informação provida pela densidade espectral de potência, porém de uma forma mais resumida, contendo apenas as frequências mais relevantes (HASAN, 2004). Devido ao fato do MFCC ser uma espécie de resumo do envelope espectral, parte da informação timbral é perdida, sendo necessário utilizar características de descrição de timbre adicionais, como o Spectral Centroid (SC) (HUQ, 2010).

[1] https://jp.mathworks.com/matlabcentral/answers/uploaded_files/23580/index.pdf
[2] http://www.muengineers.in/extc-project-list/speaker-identification

2.3.1.2 Spectral Centroid (SC)

O Spectral Centroid é definido como a medida de brilho de um som, sendo brilho uma qualidade correlacionada com a quantidade de frequências parciais altas de um sinal de áudio (NAM, 2001). Em outras palavras, quanto mais brilho, mais frequências parciais altas, e, quanto menos brilho, menos frequências parciais altas. A Figura 13 faz a comparação das vogais “i” e “o”, sendo notável a presença de parciais altas em maior quantidade na vogal "i" se comparado com a vogal "o". O som da vogal "i" é percebida com mais brilho que a vogal "o", e se as densidades espectrais de potência de ambas vogais forem analisadas, será notado que a vogal "i" possui mais frequências parciais altas que a vogal “o”.

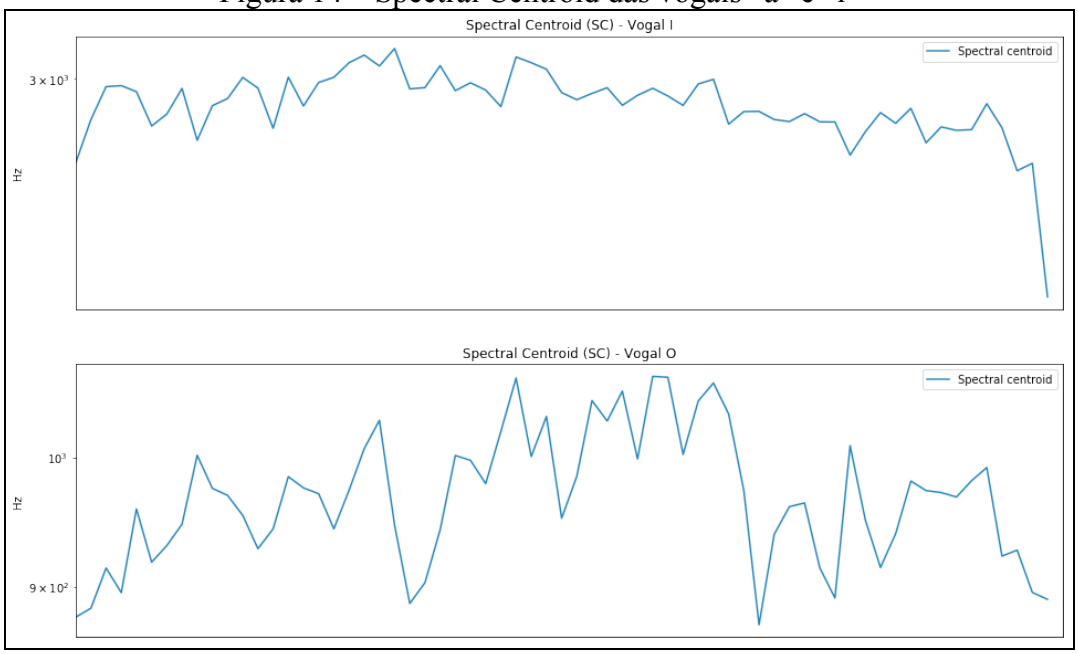
Figura 13 – Espectrogramas das vogais I e O



Fonte: Elaborado pelo Autor.

O SC pode ser adicionalmente definido como um quantificador simples da distribuição de energia na densidade espectral de potência (SCHUBERT; WOLFE, 2006). A densidade espectral de potência da vogal "i", apresentada na Figura 13, tem sua energia distribuída quase que homogeneamente pelas frequências, resultando em um SC com relativamente menos oscilação, já a densidade espectral de potência da vogal "o" tem uma distribuição de energia menos homogênea, resultando em um SC relativamente com mais oscilação. O SC das vogais "i" e "o" é demonstrado na Figura 14, sendo possível notar que a vogal "i", detentora de maior brilho, é representada com um valor mais alto (aproximadamente 3000Hz) enquanto a vogal "o", com menos brilho, é apresentada com valor mais baixo (entre 900Hz e 1kHz).

Figura 14 – Spectral Centroid das vogais “a” e “i”



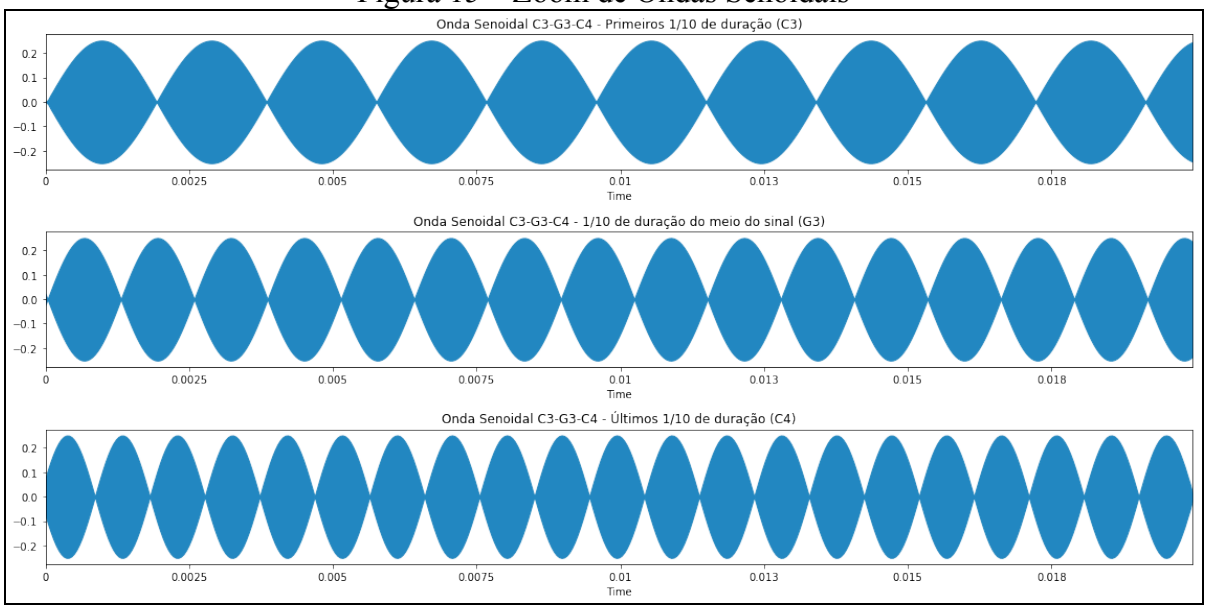
Fonte: Elaborado pelo autor.

2.3.1.3 Zero Crossing Rate (ZCR)

Segundo Bachu (2008) o Zero Crossing Rate (ZCR) serve como indicador da frequência em que a energia está concentrada em uma onda sonora. De acordo com o autor, o ZCR pode ser definido como a medida do número de vezes, em um dado intervalo ou janela, que a amplitude do sinal de áudio passa pelo valor zero. Em outras palavras, o ZCR é fortemente correlacionado com a frequência do áudio: quando a frequência é alta, o ZCR é alto, quando a frequência é baixa, o ZCR é baixo.

A Figura 15 exibe partes de uma onda senoidal composta pelas notas C3, G3 e C4, com duração total de 0.2 segundos. A segunda ilustração (G3) contém uma frequência 1.5 vezes maior que a primeira (C3) e a terceira (C4) contém uma frequência 2 vezes maior que a primeira.

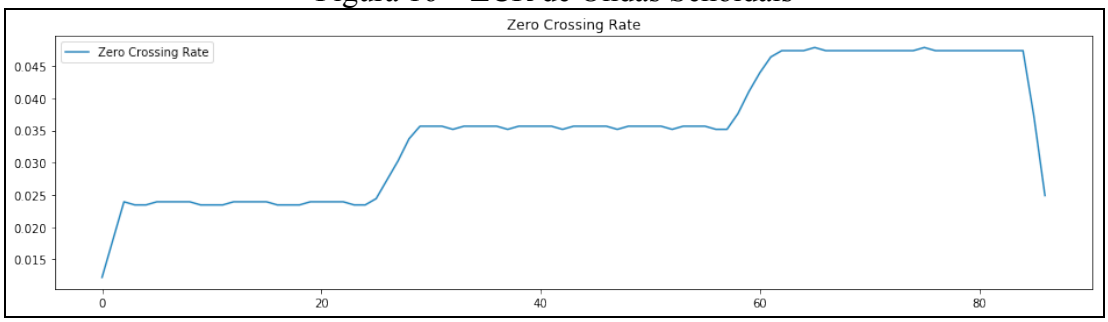
Figura 15 – Zoom de Ondas Senoidais



Fonte: Elaborado pelo autor.

A Figura 16 exibe o ZCR da onda senoidal exibida na Figura 15. Nota-se que a cada troca de nota (C3, G3 e C4) o ZCR é mais alto, sendo a última parte equivalente ao dobro da primeira e a segunda parte 1.5 vezes maior que a primeira.

Figura 16 – ZCR de Ondas Senoidais



Fonte: Elaborado pelo autor.

2.3.2 Harmônica

Esta seção irá explicar as características de áudio pertencentes a harmonia musical que serão extraídas para a realização deste trabalho.

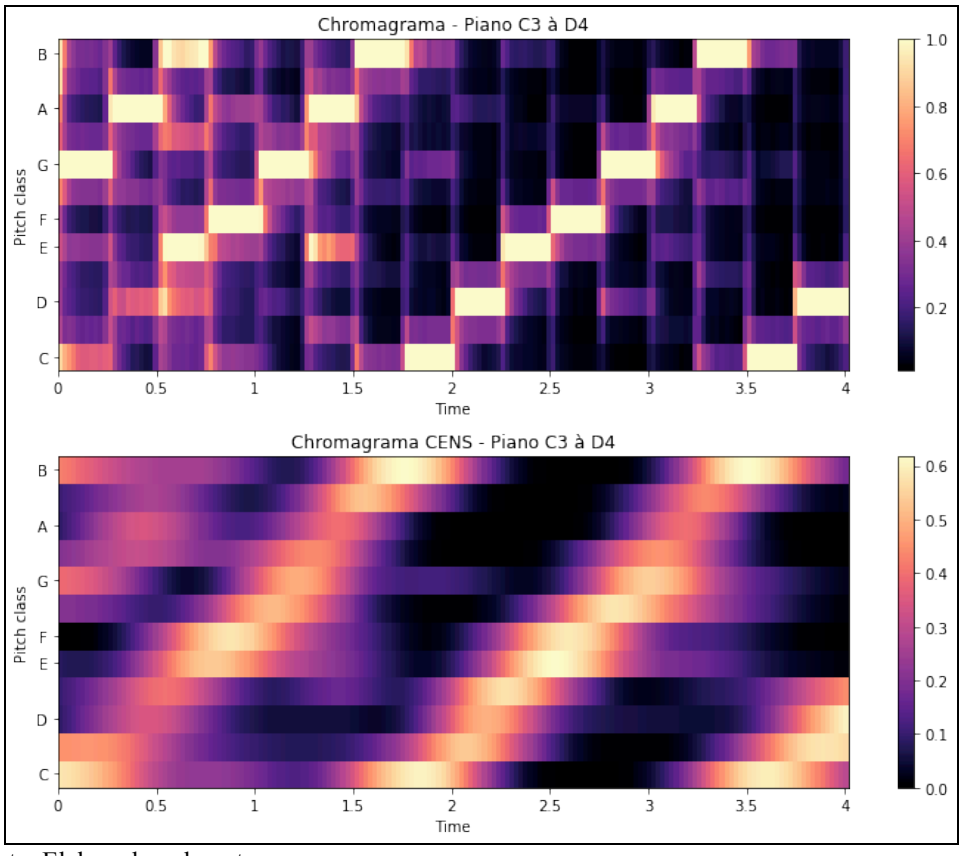
2.3.2.1 Chromagram

Segundo Iftene (2011), o chromagram é uma representação do espectro total de um sinal de áudio em uma única oitava, isto é, um intervalo de 12 semitons. O autor afirma também que o conceito de chromagram é baseado na premissa de que notas com distâncias de uma oitava são percebidas como similares, a partir disso, intervalos pertencentes a diferentes oitavas são mapeados em uma única oitava.

O chromagram é muito utilizado em sistemas de detecção de gênero musical e identificação de músicas dentre várias versões dela, como por exemplo uma mesma música performada por diferentes bandas (ENGLMEIER, 2015). Adicionalmente, o chromagram é considerado como o equivalente ao que as palavras são no ambiente textual. Mesmo músicas pertencentes a estilos diferentes seriam consideradas similares devido ao conteúdo harmônico descrito pelo chromagram, o que não seria igual se mensuradas por características de timbre como MFCC, as quais iriam classificar as músicas como pouco semelhantes (IFTENE, 2011). Estudos correlacionam o modo da escala com a percepção emocional, sendo escalas maiores correlacionadas com sentimentos alegres e escalas menores correlacionadas com sentimentos tristes (THAUT, 2005).

O chromagram utilizado neste trabalho possui um processamento posterior, conhecido como CENS (Chroma Energy Normalized Statistics). O CENS absorve mudanças de escopo dinâmico, timbre e micro desvios de tempo, proporcionando dados mais eficientes para processamentos posteriores. A Figura 17 exibe, respectivamente, o chromagram e o chromagram CENS de uma escala de C iniciando em C2 e terminando em D4, reproduzida por um piano. Nota-se que, mesmo as notas passando de uma oitava (C2 à D4 passa por três oitavas), o chromagram as mapeia para uma oitava apenas.

Figura 17 – Chromagram de C3 à D4



Fonte: Elaborado pelo autor.

2.3.3 Temporal

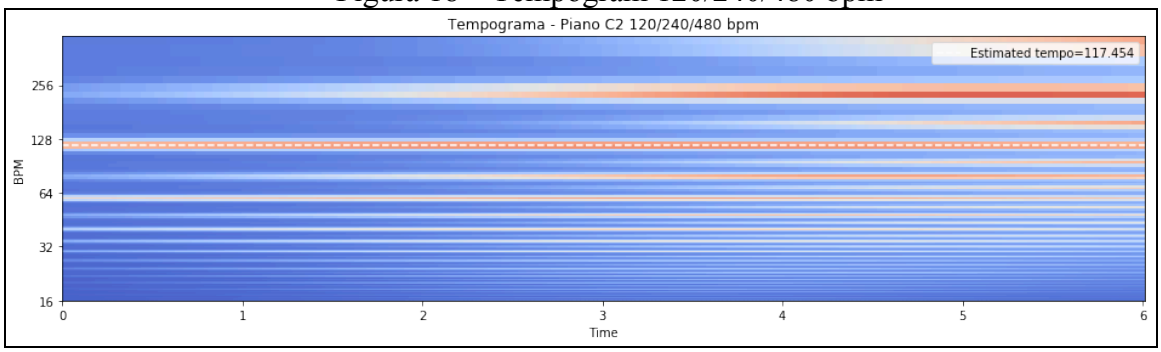
Existem várias características que mensuram o tempo de uma música, como, por exemplo, CDF (Complex-Domain Onset Detection Function), BH (Beat Histogram), OR (Onset Rate) e Tempogram. Buscando obter o maior detalhamento da informação de tempo em um número reduzido de características, apenas o Tempogram foi utilizado neste trabalho.

2.3.3.1 Tempogram

O tempo é uma característica pertencente a todas as músicas, estando presente, na maioria das vezes, de forma dinâmica ao longo do decorrer da música. Não é possível dizer que uma música possui um tempo constante sem perder certa precisão na informação. Isso se dá devido ao fato de que o tempo é utilizado pelo compositor da mesma forma que o timbre e a harmonia: com o fim de atingir algum significado, em última instância emocional. Desta forma, ora o tempo acelera, ora desacelera (CROSSLEY-HOLLAND, 2014).

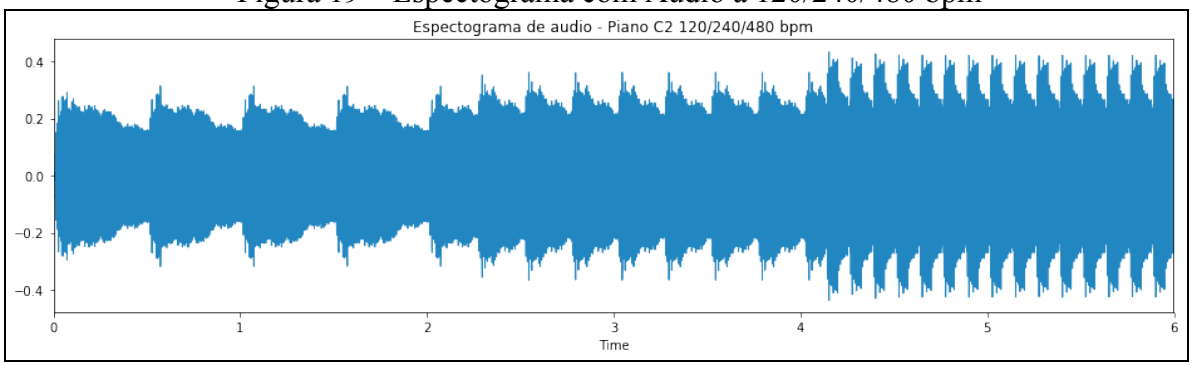
O tempograma revela características de tempo locais, não apenas um número para a música inteira, promovendo riqueza na representação de tempo em uma música mesmo para músicas com alta expressividade de tempo (tempos variados em uma mesma música) (GROSCHE; MÜLLER, 2011). Na Figura 18 é exibido o tempograma de um piano tocando uma nota nos tempos de 120, 240 e 480 bpm. A Figura 19 exibe o espectro deste áudio para melhor visualização das batidas.

Figura 18 – Tempograma 120/240/480 bpm



Fonte: Elaborado pelo autor.

Figura 19 – Espectograma com Áudio a 120/240/480 bpm



Fonte: Elaborado pelo autor.

2.4 TRABALHOS CORRELATOS

Esta seção apresenta os trabalhos que possuem características semelhantes aos principais objetivos da abordagem proposta. O primeiro descreve a utilização de lógica difusa na categorização de emoções (YANG et. al., 2006). O segundo propõem a utilização de algoritmos de múltiplas classes (multilabel) para categorização de músicas por emoção (TROHIDIS, K. et al., 2008) e o terceiro desenvolveu uma solução para extração de emoções de músicas utilizando regressão (YANG et al., 2008). Por fim, o quarto trabalho correlato testou sistematicamente, com o objetivo de encontrar a abordagem de identificação de emoções com regressão com menor erro, uma série de abordagens envolvendo: conjunto de características, algoritmos e pré-processamentos (Huq, 2010).

2.4.1 Music emotion classification: a fuzzy approach

O trabalho de Yang, Liu e Chen (2006) aborda a categorização de músicas levando em consideração a natureza subjetiva da emoção. Emoções são subjetivas de tal forma que são tratadas como opiniões: pessoas tem opiniões em relação a emoção predominante em músicas.

Com base nesta premissa, utilizou-se lógica difusa em conjunto com o plano circumplexo de Russel (1980) para obter a categoria emocional de músicas. Classificadores difusos, diferente de classificadores tradicionais, vinculam um vetor de classes a uma amostra, enquanto classificadores tradicionais vinculam apenas uma classe a uma amostra. Tendo como exemplo a categorização musical por emoções, onde uma música seria uma amostra e uma emoção uma classe, uma abordagem classificatória tradicional vincularia uma emoção a uma música, enquanto a classificação difusa vincularia várias emoções a uma música.

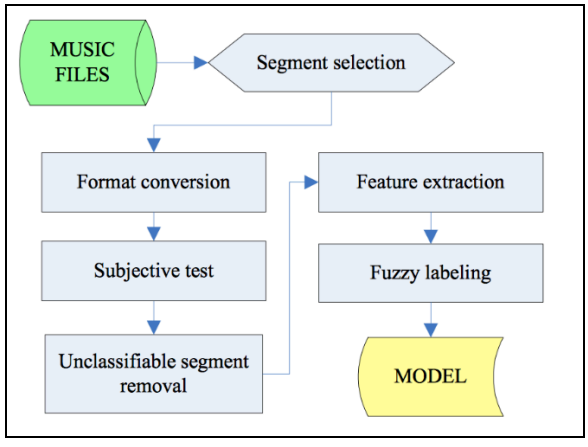
O sistema proposto é dividido em duas partes: o Gerador de Modelo (GM) e o Classificador de Emoção (CE). O GM gera um modelo baseado nas características das amostras de treinamento e nas classes apontadas por voluntários, enquanto o CE aplica este

modelo na classificação das amostras de classificação. A Figura 20 e Figura 21 descreve o fluxo de execução do GM e CE.

A Figura 20 representa a fase de GM, tendo as etapas:

a) Segment Selection: 243 amostras de músicas populares Ocidentais, Chinesas e Japonesas foram obtidas, onde um trecho de 25s de cada música foi extraído;
b) format Conversion: cada um dos segmentos é convertido a um sample rate de 22kHz, 16bit mono, no formato PCM WAV;
c) subjective Test: voluntários classificam cada um destes trechos conforme suas próprias opiniões utilizando os 4 quadrantes do plano de Russel (RUSSEL, 1980), resultando nas classes 1, 2, 3 e 4;
d) unclassifiable Segment Removal: trechos que foram classificados pelos voluntários com emoções distintas são removidas;
e) Feature Extraction: trechos de áudio processados pela ferramenta PsySound para extração de características;
f) Fuzzy Labeling: as características são enviadas aos algoritmos de lógica difusa, sendo eles: Fuzzy KNN Classifier (FKNN) (KELLER et.al., 1985) e Fuzzy Nearest Mean Classifier (FNM);
g) MODEL: modelo gerado pelo fuzzy labeling contendo, para cada trecho musical (extraído na fase segment selection) um conjunto de características e um vetor de classes. Este modelo será utilizado na classificação de novas músicas.

Figura 20 - Diagrama de bloco do GM

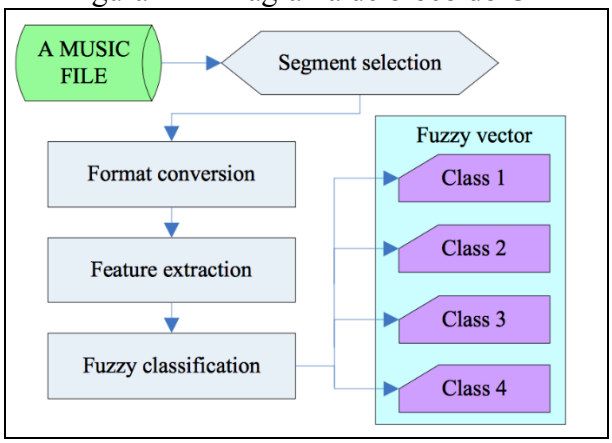


Fonte: Yang, Liu e Chen (2006).

A Figura 21 representa a fase de CE. Esta fase possui apenas uma etapa diferente da fase de GM, que é a etapa *fuzzy classification*. Nesta etapa o modelo gerado pela fase de GM é utilizado para classificar novas músicas teoricamente não vistas antes pelo algoritmo.

O resultado é considerado o maior valor presente no vetor difuso (*fuzzy vector*). Este vetor possui a intensidade (de 0 a 1) das quatro classes, por exemplo v(0.1, 0, 1, 0) é um vetor onde a classe 1 possui peso 0.1, a classe 2 possui peso 0, a classe três possui peso máximo e a classe quatro possui peso 0, tendo como resultado final a classe três pelo fato de possuir o maio peso dentre as 4.

Figura 21 - Diagrama de bloco do CE



Fonte: Yang, Liu e Chen (2006).

2.4.2 Multilabel classification of music into emotions

Wieczorkowska (2006) aborda o problema de classificação de músicas em emoções com um modelo de classificação multi classe. Diferente da maioria das outras abordagens na classificação de músicas por emoção ou simplesmente extração de emoções de músicas, que tem como saída uma classe para uma música, esta abordagem resulta em uma ou mais classes para uma música.

Utilizou-se um *dataset* contendo 100 músicas dos gêneros: Clássico, Reggae, Rock, Pop, Hip-Hop, Techno e Jazz. De cada música foi extraído um trecho de 30 segundos seguintes aos primeiros 30 segundos da música. Por fim, cada um destes trechos foi convertido a uma taxa de amostragem de 22.050Hz, 16bit amostra, mono.

Para criar as classes foi utilizado o modelo de emoções de Tellegen-Watson-Clark (Figura 22), resultando nas classes presentes no Quadro 1. Estas classes foram utilizadas nos algoritmos de classificação. O modelo de Tellegen-Watson-Clark foi concebido a partir do modelo de Russel (1980), na qual foram adicionados apenas dois eixos, um na diagonal principal e um na diagonal oposta, sendo assim os dois eixos principais (existentes no modelo de Thayer) continuam tendo o mesmo significado. Os dois eixos diagonais representam a junção dos eixos principais adjacentes. Por exemplo: a extremidade da diagonal principal no

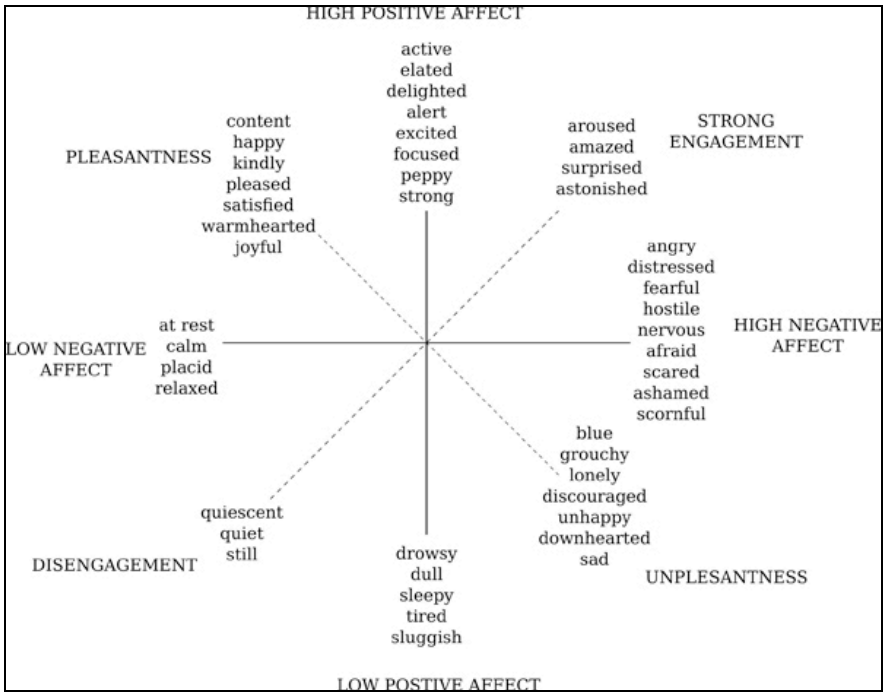
quadrante inferior direito representa a junção de Alto Efeito Negativo (*High Negative Effect*) com Baixo Efeito Negativo (*Low Positive Effect*), resultando em Desagradável (*Unplesantness*).

Quadro 1 – Relação de Classes

Classe	Descrição
L1	Espantado-surpreso
L2	Feliz-satisfeito
L3	Relaxado-calmo
L4	Quieto-imóvel
L5	Triste-solitário
L6	Nervoso-temeroso

Fonte: Elaborado pelo autor.

Figura 22 - Modelo Tellegen-Watson-Clark



Fonte: Wieczorkowska (2006).

Para a extração de características foi utilizado o framework Marsyas, sendo as características pertencentes a duas categorias: rítmo e timbre. Os algoritmos classificadores comparados foram: binary relevance (BR), label powerset (LP), random k-labelsets (RAKEL) e multilabel k-nearest neighbor (MLkNN).

A Tabela 1 exibe a acurácia de cada um dos algoritmos classificadores utilizados para cada uma das possíveis classes. O algoritmo RAKEL se mostrou mais assertivo em praticamente todas as classificações. Além disso, foi possível perceber que este modelo de classificação é mais preciso com emoções como L4 (quieto) do que L2 (feliz), conforme apresentado no Quadro 1.

Tabela 1 - Precisão dos Algoritmos

	BR	LP	RAKEL	MLkNN
L1	0.7900	0.7906	0.7982	0.7446
L2	0.7115	0.7380	0.7587	0.7195
L3	0.7720	0.7705	0.7854	0.7221
L4	0.8997	0.8992	0.9031	0.7969
L5	0.8287	0.8093	0.8236	0.7051
L6	0.8322	0.8142	0.8238	0.7422

Fonte: Wieczorkowska (2006).

2.4.3 A Regression Approach to Music Emotion Recognition

O trabalho de Yang (2013) abordou o reconhecimento de emoções de músicas como um problema de regressão linear para predizer os valores de valência (positivo-negativo) e alerta (calmo-excitado) de cada amostra musical. Uma vez associado com um valor de valência e alerta, cada amostra musical se torna um ponto no plano circumplexo de Russel (RUSSEL, 1980), permitindo assim a obtenção de músicas a partir de um ponto neste plano.

A extração de características foi feita utilizando os aplicativos PsySound (CABRERA D., 1999) e Marsyas (TZANETAKIS; COOK, 2002), resultando na extração de 114 características. Para selecionar as características mais importantes foi utilizada a técnica RreliefF[3].

Na classificação algumas opções de regressores lineares foram testadas, e a opção que mais proporcionou assertividade no processo de predição dos valores de valência e alerta foi o método de Máquina de Vetores de Suporte (*Support Vector Machine* ou SVM). Adicionalmente, para melhorar a performance do algoritmo de classificação, foi aplicada a técnica de Análise de Componente Principal (PCA) para diminuir a correlação entre valência e alerta.

2.4.4 Automated Music Emotion Recognition

O trabalho de Huq (2010) avalia várias abordagens de reconhecimento de emoções em músicas. Todas as abordagens testadas dividem uma mesma base de configurações:

[3] RreliefF é um algoritmo resultado da modificação do algoritmo RELIEFF, originalmente utilizado para problemas de categorização, para utilização com problemas de regressão.

a) utilização de regressão;

b) utilização do plano de Russel (1980)

c) utilização apenas do conteúdo acústico (sem as letras) das músicas.

O Quadro 2 exibe quais são as características acústicas utilizadas pelo trabalho.

Quadro 2 – Relação de características

Nome	Categoria
MFCC	Timbral
HFC	Timbral
SC	Timbral
ZCR	Timbral
SONES	Amplitude
RMS	Amplitude
Chroma	Harmonica
TS	Harmonica
CDTS	Harmonica
TriadSeq	Harmonica
TriadInt	Harmonica
CDF	Ritmica
BH	Ritmica
OR	Ritmica

Fonte: Huq (2010)

Várias técnicas de pré-processamento de características também foram avaliadas, relacionadas nas tabelas Tabela 2 para valência e na

Tabela 3 para alerta. A coluna Algoritmo apresenta os algoritmos utilizados para cada uma das técnicas de pré-processamento avaliadas. A coluna Nenhum exibe o MAE resultante de se executar o algoritmo sem nenhum pré-processamento. A coluna STD exibe o resultado utilizando o pré-processamento de distrubuição normal, onde todos os valores são normalizados para terem média 0 e desvio padrão 1. A coluna PCA indica o resultado proveniente da execução dos algoritmos após submeter as características para o algoritmo de redução de dimensionalidade Principal Component Analisys.

Tabela 2 – Relação de técnicas de pré-processamento utilizadas: Valência

Algoritmo	Nenhum	STD	PCA
Baseline	0.236		
LR	0.296	0.298	0.329
RT	0.217	0.217	0.235
LWR-SLR	0.222	0.222	0.230
M5P	0.225	0.224	0.231
SVR-RBF	0.236	0.198	0.198

Fonte: Huq (2010)

Tabela 3 – Relação de técnicas de pré-processamento utilizadas: Alerta

Algoritmo	Nenhum	STD	PCA

Baseline	0.305		
LR	0.201	0.203	0.221
RT	0.208	0.208	0.216
LWR-SLR	0.221	0.221	0.207
M5P	0.177	0.177	0.190
SVR-RBF	0.304	0.156	0.157

Fonte: Huq (2010)

Em conclusão, Huq (2010) afirma que o único pré-processamento com ganho notável é o de STD, e o algoritmo com menor MAE dentre os algoritmos testados, listados nas tabelas Tabela 2 e

Tabela 3, é o SVR com kernel RBF. Buscando um ponto de partida robusto, este trabalho faz uso dos resultados obtidos pelo trabalho de Huq (2010), utilizando parte de seu conjunto de características, algoritmo, kernel de algoritmo, pré-processamentos e medição de erro.

3 DESENVOLVIMENTO

Este capítulo apresenta os detalhes de desenvolvimento deste trabalho. A seção 3.1 se inicia com a descrição dos requisitos funcionais e não funcionais levantados para o desenvolvimento da aplicação, seguida da especificação do trabalho (na seção 3.2), onde apresenta os diagramas de casos de uso e de classes. A seção 3.3 aborda assuntos referentes à implementação, listando as ferramentas utilizadas no desenvolvimento e explicando sobre a operacionalidade da aplicação. Por fim, a seção 3.4 analisa os resultados obtidos a partir do presente projeto.

3.1 REQUISITOS

O projeto proposto deve atender aos requisitos funcionais, apresentados no Quadro 3, e aos requisitos não funcionais, apresentados no Quadro 4. Os requisitos funcionais estão relacionados aos casos de uso apresentados na Figura 23.

Quadro 3 – Requisitos do projeto

RF01	Identificar o conjunto de emoções presentes na música (UC1)
RF02	Permitir configurar execuções para testar abordagens diferenciadas (UC2)
RF03	Coletar informações de assertividade das identificações feitas (UC3)
RF04	Salvar o resultado das execuções em uma base de dados (UC4).

Fonte: Elaborado pelo autor.

Quadro 4 – Requisitos Não Funcionais do projeto

RNF01	Utilizar o algoritmo de regressão que obteve o melhor resultado
RNF02	Utilizar o conjunto de características que obteve o resultado mais eficaz
RNF03	Utilizar a linguagem Python
RNF04	Utilizar os frameworks: Pandas, SciKit-Learn e LIBSVM

Fonte: Elaborado pelo autor.

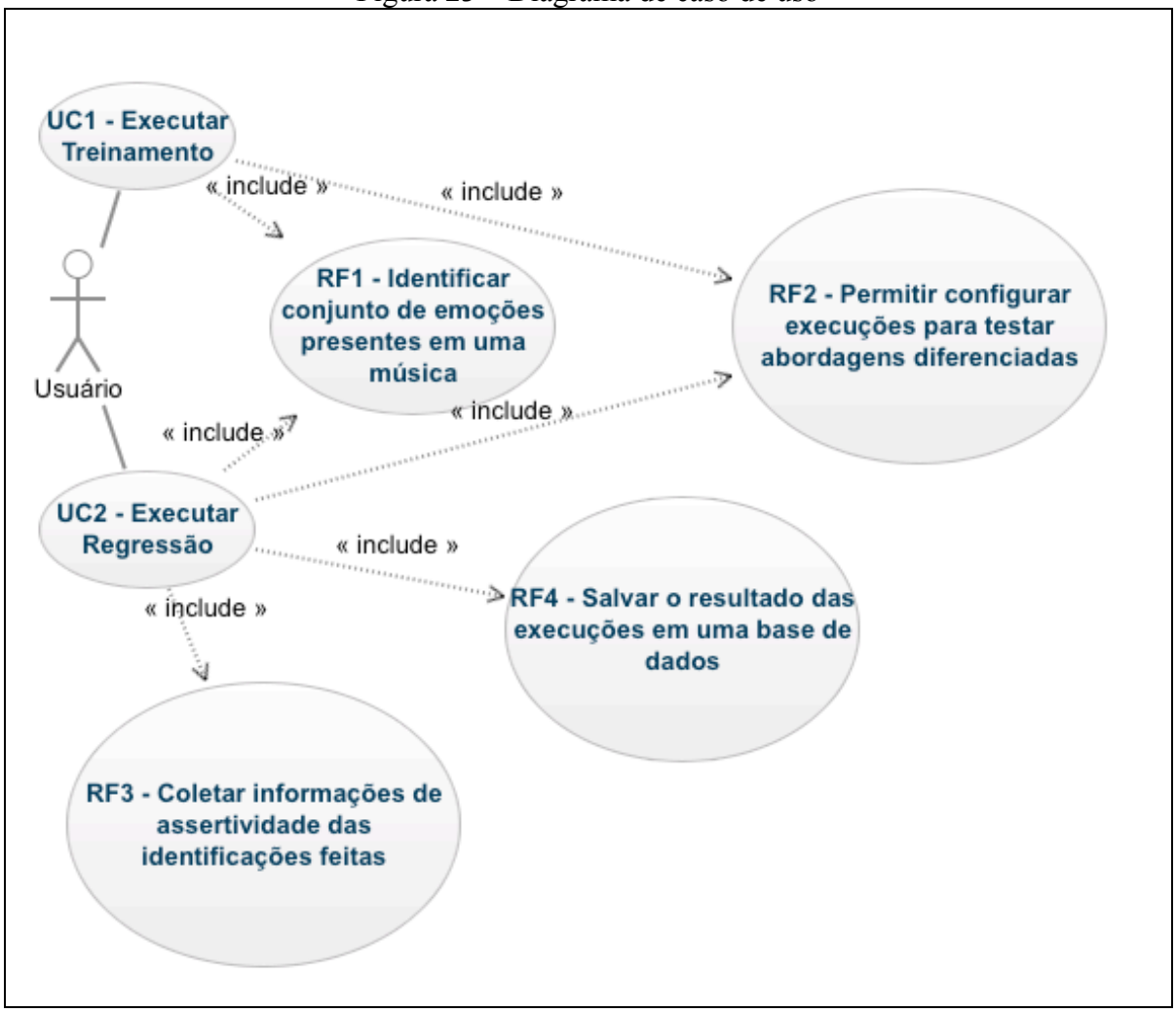
3.2 ESPECIFICAÇÃO

Nesta seção é descrita a especificação do Music Emotions Intel. A especificação é composta por um diagrama de casos de uso e um diagrama de classes, ambos feitos utilizando a ferramenta GenMyModel.

3.2.1 Diagrama de casos de uso

Essa seção demonstra as funções exercidas pela ferramenta, a qual tem apenas um ator. A Figura 23 exibe o diagrama de caso de uso.

Figura 23 – Diagrama de caso de uso



Fonte: elaborado pelo autor.

O Caso de Uso `Executar Treinamento` (UC1) engloba rotinas para treinamento do algoritmo regressor. As rotinas executadas são:

a) leitura do arquivo de áudio utilizando o bitrate configurado;
b) extração das características do áudio pertinentes para a identificação de emoções;
c) submissão das características e das anotações para o algoritmo de regressão;
d) criação do modelo regressor treinado.

O caso de uso `Executar Regressão` (UC2) engloba rotinas para a execução do algoritmo regressor previamente treinado. As rotinas executadas são:

a) leitura do arquivo de áudio utilizando o bitrate configurado;
b) extração das características do áudio pertinentes para a identificação de emoções;

c) submissão das características para o algoritmo de regressão;

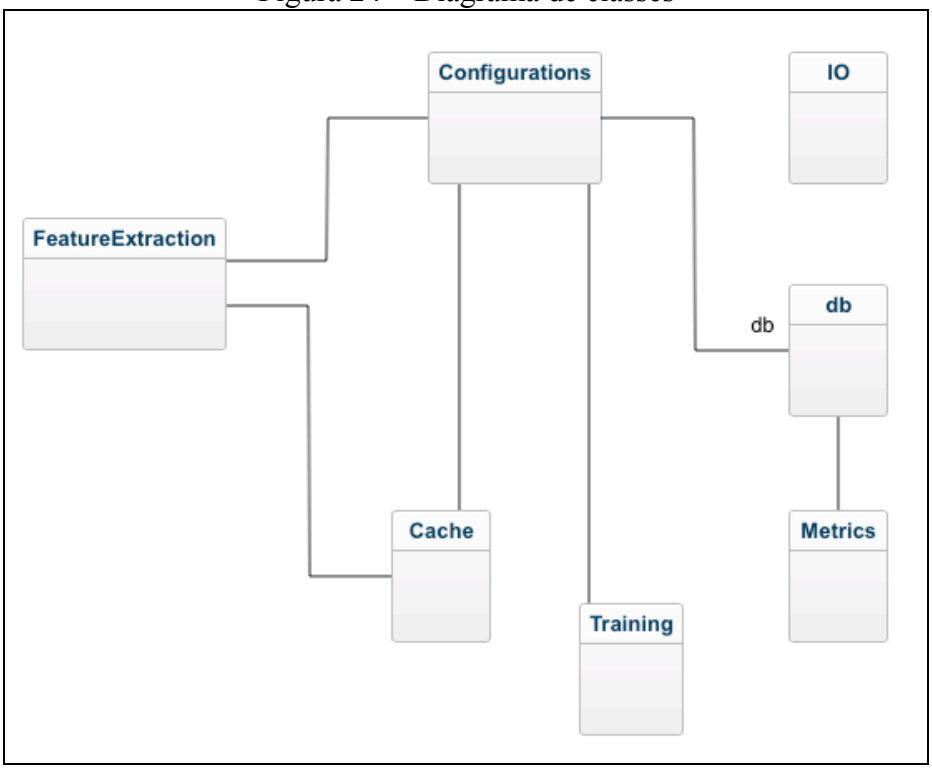
d) obtenção das métricas de assertividade;

e) persistência das métricas de assertividade.

3.2.2 Diagrama de classes

Nessa seção é abordado o diagrama de classes do sistema. A Figura 24 apresenta o diagrama de classes do sistema sem a informação de métodos e atributos nas classes.

Figura 24 – Diagrama de classes



Fonte: Elaborado pelo autor.

A classe `Configurations` concentra todas as configurações utilizadas pela ferramenta. Esta classe carrega seu conteúdo a partir de um arquivo de configurações (configs.ini - Quadro 7) e expõe suas configurações através de métodos `get`. Todas as informações carregadas a partir do arquivo de configurações são imutáveis. A classe `IO` agrega métodos de leitura e escrita em disco, como o método `load_audio`, por exemplo, que recebe como parâmetro o caminho de um arquivo de áudio e retorna uma matriz 2d representando seu conteúdo. O nome IO provém da sigla de *In Out,* termo que representa rotinas de leitura e escrita em memória.

A classe `Cache` e responsável por criar e carregar arquivos de cache. Os arquivos de cache são arquivos contendo versões pré-processadas de informações como: matriz de

características, modelos probabilísticos, entre outros; o mecanismo de cache facilita a execução utilizando perfis de configuração parecidos, reaproveitando etapas já processadas.

A classe `Metrics` agrega todas as funcionalidades relacionadas a métricas de assertividade e performance dos algoritmos preditivos. A classe `Db` expõem métodos para persistência e leitura de dados em uma base de dados relacional. Por fim, a classe `FeatureExtraction` reúne todas as funções para as etapas de extração de características e pré-processamento de características.

3.3 IMPLEMENTAÇÃO

Nesta seção são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

Para implementar o trabalho proposto utilizou-se o ambiente de desenvolvimento PyCharm versão 171.4249 juntamente com o Jupyter Notebook versão 4.3.0. O PyCharm proporciona um editor de texto inteligente e com vários recursos focados no aumento de produtividade para a linguagem Python. O Jupyter Notebook permite criar relatórios com gráficos de forma prática e rápida.

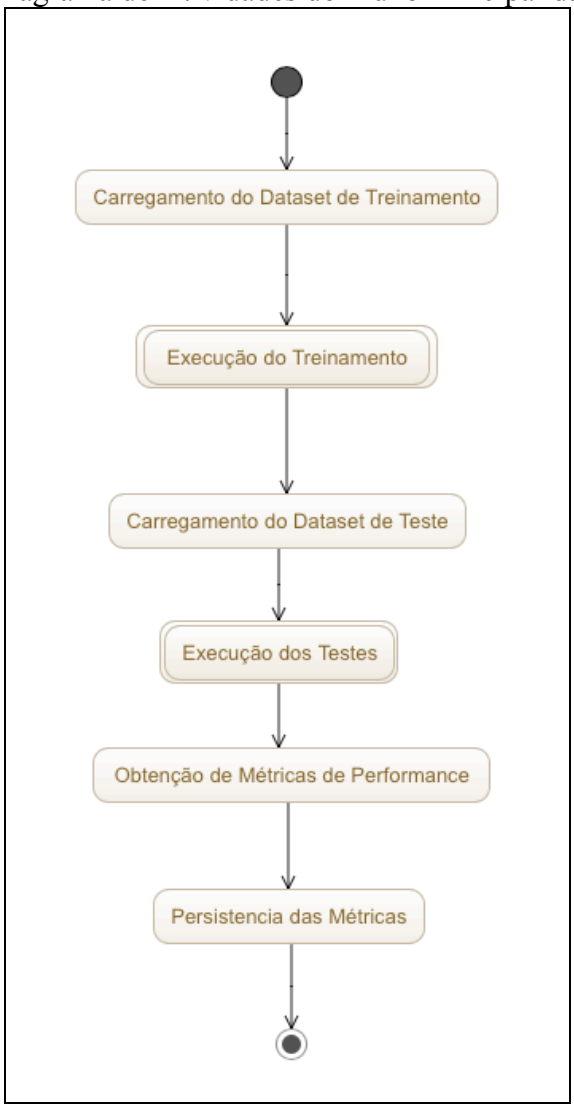
Além destas ferramentas, também foram utilizadas as bibliotecas: Scikit Learn, Pandas, Numpy, Matplot Lib e Librosa. A biblioteca Scikit Learn provê vários algoritmos preditivos, algoritmo de métricas, algoritmos de processamento de dados e escolha de características; a biblioteca Pandas possibilita a manipulação de conjunto de dados de forma prática e eficaz; a biblioteca Numpy expõem funcionalidades de manipulação de estruturas (vetores, matrizes) e números; a biblioteca Matplot Lib possibilita a geração de gráficos a partir de dados; a biblioteca Librosa provê toda a funcionalidade de processamento de sinais de áudio e extração de características de áudio. A Figura 25, Figura 26 e Figura 27 demonstra o fluxo principal de atividades da ferramenta através de um diagrama de atividades.

Na Figura 25 é apresentado o fluxo macro da execução. Primeiro o *dataset* é carregado, que inclui os arquivos de áudio de treinamento juntamente com arquivos de anotações de valência e alerta de cada áudio. Após isso, o treinamento é executado (demonstrado em mais detalhes na Figura 26), onde os arquivos de áudio são segmentados e os vetores de características são extraídos. O conjunto de características escolhido foi baseado no trabalho de Huq (2010), no qual várias combinações de características e vários algoritmos

de aprendizagem de máquina foram experimentados, tendo como resultado final de maior assertividade o conjunto de características utilizado por este trabalho. Em seguida o pré-processamento de características é executado, transformando os vetores de características para um formato mais eficaz. Após isso os valores de anotações de valência e alerta são carregados para serem vinculados a aos seus respectivos vetores de características. Por final, os vetores de características e os valores de anotações são submetidos ao algoritmo regressor SVR-RBF (Support Vector Regression com kernel Radial Basis Function).

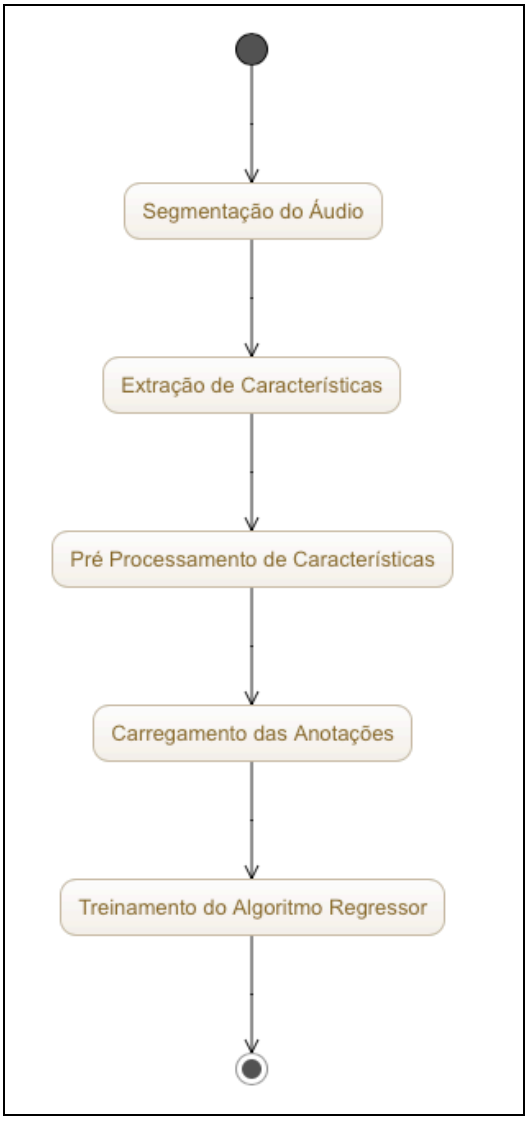
Terminando a execução do treinamento, a parcela de dados destinada a testes, proveniente do mesmo *dataset* de treinamento, é carregada para a execução dos testes e obtenções de métricas de performance (MAE). Ao final as informações de métricas são persistidas em uma base de dados relacional para futuras comparações. A rotina de execução de testes, demonstrada na Figura 27, é semelhante à rotina de execução de treinamento, diferindo apenas no último passo, que ao invés de utilizar os vetores de características para treinar, utiliza para testar.

Figura 25– Diagrama de Atividades do Fluxo Principal da Ferramenta



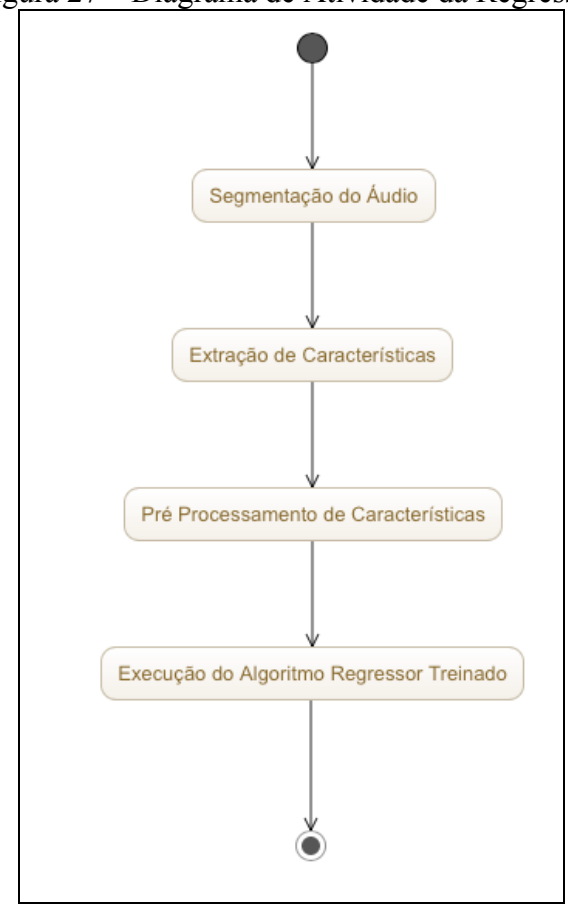
Fonte: Elaborado pelo autor.

Figura 26 – Diagrama de Atividades da Fase de Treinamento



Fonte: Elaborado pelo autor.

Figura 27 – Diagrama de Atividade da Regressão



Fonte: Elaborado pelo autor.

3.3.2 Etapas de desenvolvimento

O desenvolvimento do trabalho pode ser dividido em 4 ciclos: No primeiro ciclo foi feito um estudo para encontrar um *dataset* eficaz na identificação de emoções em músicas. O segundo ciclo, se concentrou em desenvolver um algoritmo para a leitura do *dataset*, a leitura de arquivos de áudio, e para a extração e pré-processamento das características de músicas. No terceiro ciclo foi feita a implementação das fases de treinamento (geração do modelo) e dos testes preditivos. Por fim, o quarto ciclo teve foco na execução de vários perfis de configuração, buscando o perfil que trouxesse a maior assertividade possível na identificação de emoções em músicas.

3.3.2.1 Etapa 1: Base de dados utilizada

O *dataset* utilizado para o treinamento e classificação, extraída de Soleymani (2013), é formada por músicas (arquivos de áudio mp3) e anotações de Valência e Alerta (V/A)

referente as músicas. As músicas são licenciadas pelo Creative Commons[4] e obtidas do repositório Free Music Archive (FMA). As anotações de V/A foram adquiridas através do *crowdsourcing*[5] (serviço de respostas de pesquisas em massa) da Amazon, a Amazon Mechanical Turk (MTurk), sendo que cada música foi anotada por, em media, 10 colaboradores do MTurk.

Das 1000 músicas originalmente presentes, 256 foram removidas por apresentarem semelhanças muito grandes entre si, resultando em efetivamente 744 músicas. O processo de remoção de músicas semelhantes foi feito por Soleymani como parte do trabalho de desenvolvimento do dataset. Cada uma das músicas se encontram padronizadas com 45 segundos de duração e frequência de amostragem de 44.100Hz. O *dataset* possui originalmente 125 músicas de cada um dos seguintes gêneros: Blues, Electronica, Rock, Classica, Folk, Jazz, Country e Pop; sendo que nenhuma delas possui mais de 10 minutos de duração ou menos de 1 minuto de duração. Cada um dos gêneros citados possui músicas de 53 a 100 artistas diferentes.

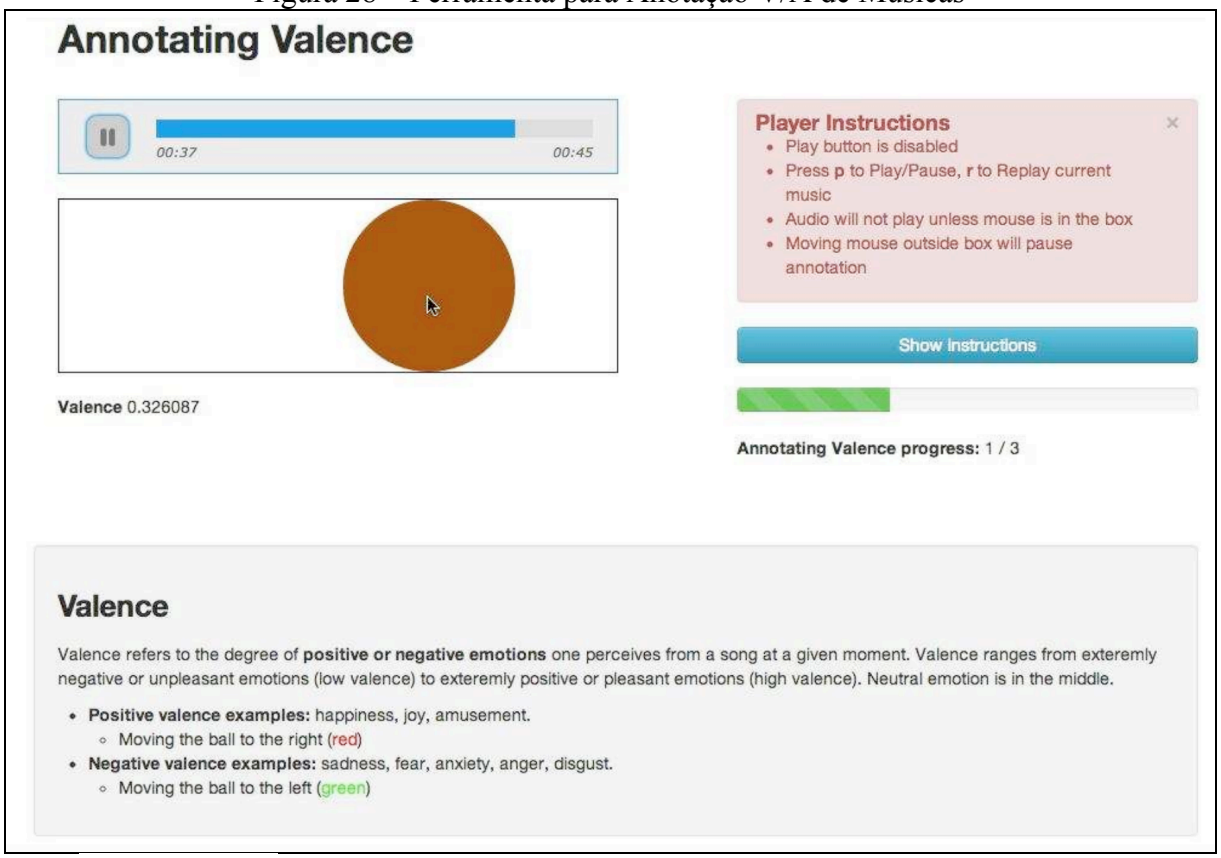
As anotações foram feitas utilizando um software desenvolvido por Soleymani (2013). Este software, exibido na Figura 28, permitia ao colaborador Mturk, chamado de anotador, definir um valor de V/A de forma continua (1 valor de V/A a cada 0.5 segundos) para cada música. Esta definição de valores foi feita movendo o círculo marrom para a esquerda (negativo) ou para a direita (positivo), sendo os valores entre -0.5 e +0.5. Os primeiros 5 segundos de anotações foram descartados devido ao fato de que nos segundos iniciais o anotador está encontrando o local ideal do círculo. Após anotar as músicas de forma contínua, os anotadores davam adicionalmente uma nota única de V/A para todo o trecho de 45 segundos, utilizando uma escala de 0 a 9.

---

[4] Organização sem fins lucrativos que permite o compartilhamento e uso da criatividade e do conhecimento através de instrumentos jurídicos gratuitos.

[5] Modelo de produção que usa de conhecimentos coletivos e voluntários (recrutados especialmente na internet) para solucionar problemas do dia a dia, desenvolver novas tecnologias, criar conteúdo ou prover serviços.

Figura 28 – Ferramenta para Anotação V/A de Músicas



Fonte: Soleymani (2013).

O FMA, repositório de origem das músicas utilizadas no *dataset*, é dirigido pela WFMU8, uma das mais renomadas estações de rádio dos Estados Unidos. Para garantir alta qualidade, cada áudio presente no FMA foi escolhido manualmente por curadores de áudio. Além do fato de que todas as músicas presentes no FMA são protegidas pelo Creative Commons[4], não são publicadas por gravadoras, reduzindo o potencial bias. Este bias pode influenciar na opinião do anotador, que deixa de responder conforme se sente no momento para responder conforme o que já conhece sobre a música. O fato de as músicas não serem publicadas por gravadoras reduz sua popularidade e visibilidade, o que por sua vez reduz a chance de ser conhecido por algum anotador.

3.3.2.2 Etapa 2: Desenvolvimento do algoritmo de IO e extração e pré-processamento de características.

Para o desenvolvimento desta fase do algoritmo foram utilizadas as bibliotecas Librosa, Pandas e Numpy para prover funcionalidades de extração e pré-processamento de características, juntamente com funcionalidades de carga do *dataset* e de arquivos de áudio. A carga do *dataset* é a funcionalidade mais simples, devido ao fato do formato do *dataset* ser um

CSV (Comma Separated Value) e do uso da biblioteca Pandas. O Quadro 5 demonstra o código de carga do *dataset*.

Quadro 5 – Rotina de Carga do Dataset

```
1   def load_annotations(annotations_path):
2       log.debug('Loading annotations from path %s' % annotations_path)
3       annotations = pd.read_csv(filepath_or_buffer=annotations_path,
4   delimiter=',')
5       res = pd.DataFrame(data={'ID': annotations['song_id'], 'AROUSAL':
6   annotations['mean_arousal'],
7                                'VALENCE': annotations['mean_valence']},
8   columns=['ID', 'AROUSAL', 'VALENCE'])
9       res = res.set_index('ID')
10      log.debug('Annotations successfully loaded')
11      return res
```

Fonte: Elaborado pelo autor.

A linha 3 demonstra a chamada do método `read_csv` da biblioteca Pandas. Este método recebe um caminho (annotations_path) apontando para o *dataset* e retorna uma instância de DataFrame, que se traduz para uma matriz 2x2 contendo os dados do *dataset* e expondo várias funcionalidades sobre os dados. Então, da linha 5 à 9 o DataFrame inicial é transformado em uma matriz contendo apenas 3 colunas: ID, AROUSAL e VALENCE, onde o ID equivale a um número de identificação de uma determinada música, e AROUSAL e VALENCE são os respectivos valores de valência e alerta da música.

A rotina de carregamento de áudio faz uso da biblioteca Librosa demonstrado no Quadro 6. As linhas 10 e 11 demonstram o uso da biblioteca Librosa para a carga de áudio, onde é informado o caminho do arquivo de áudio, o *sampling rate* (quantidade de amostras de áudio por segundo), um booleano indicando modo mono ou estéreo e a duração a ser carregada do áudio. O retorno deste método é uma matriz 2x2 onde os índices principais da matriz equivalem aos segundos da música e os índices secundários equivalem às informações de pressão de ar, sendo seu tamanho dependente do *sampling rate* definido. O tamanho padrão dos segmentos carregados das músicas é de 15 segundos. Este valor para a duração dos segmentos foi escolhido arbitrariamente, buscando um equilíbrio entre tamanho do vetor de características final (quanto maior o segmento, maior o vetor) e tamanho do erro.

Quadro 6 – Rotina de Carga de Arquivos de Áudio

```
1  def load_audio(path, duration_seconds=None, sr=22050, mono=True):
2      try:
3          if duration_seconds:
4              duration_seconds = int(duration_seconds)
5          sr = int(sr)
6      except ValueError as e:
7          raise ValueError('Invalid audio duration/sr value: %s, %s' %
8  (duration_seconds, sr))
9      log.debug('Loading audio time series from path %s' % path)
10     audio_time_series, sr = librosa.load(path, sr=sr, mono=mono,
11 duration=duration_seconds)
12     log.debug('Audio time series successfully loaded')
13     return audio_time_series, sr
```

Fonte: Elaborado pelo autor.

A rotina de extração de características também é feita utilizando a biblioteca Librosa, conforme demonstrado no Quadro 7. Cada uma das funções demonstradas é responsável por fazer a extração de uma característica, sendo possível parametrizar a extração de cada uma delas. As características extraídas são: Mel-Frequency Cepstral Coefficients (`extract_mfcc`), Spectral Centroid (`extract_spectral_centroid`), Chromagram (`extract_chromagram`), e tempogram (`extract_tempogram`). As parametrizações são expostas através do arquivo de configurações, demonstrado no Quadro 8.

Quadro 7 – Rotinas de Extração de Características

```
1  def extract_mfcc(self, audio_time_series):
2      """
3      Compute the mel-frequency cepstral coefficients
4      """
5      mfcc = lr.feature.mfcc(y=audio_time_series,
6  sr=self.configs.audio_sampling_rate,
7                             n_mfcc=self.configs.feature_n_mfcc_bands)
8      return mfcc
9
10 def extract_spectral_centroid(self, power_spectogram):
11     contrast = lr.feature.spectral_centroid(S=power_spectogram,
12 sr=self.configs.audio_sampling_rate)
13     return contrast
14
15 def extract_chromagram(self, power_spectogram):
16     chroma = lr.feature.chroma_stft(S=power_spectogram,
17 sr=self.configs.audio_sampling_rate)
18     return chroma
19
20 def extract_tempo(self, audio_time_series):
21     onset_env = lr.onset.onset_strength(audio_time_series,
22 sr=self.configs.audio_sampling_rate)
23     tempo = lr.beat.estimate_tempo(onset_env)
24     return tempo
25
26 def extract_tempogram(self, audio_time_series):
27     tempogram = lr.feature.tempogram(y=audio_time_series,
28 sr=self.configs.audio_sampling_rate)
29     return tempogram
```

Fonte: Elaborado pelo autor.

O Quadro 8 demonstra o arquivo de configurações utilizado pela ferramenta. Este arquivo é dividido em sete seções, sendo elas: `dataset`, `audio`, `model`, `feature_extraction`, `cache`, `database`, `execution`.

a) `dataset`: reúne configurações pertinentes a base de dados de treinamento;
b) `áudio`: reúne configurações relacionadas a rotinas de manipulação de áudio.
c) `model`: agrupa configurações pertinentes ao modelo de treinamento.
d) `feature_extraction`: agrupa configurações relacionadas às rotinas de extração de características.
e) `cache`: reúne configurações utilizadas pelas rotinas de cache.
f) `database`: agrupa configurações de acesso a base de dados de armazenamento de métricas.
g) `execution`: agrupa *flags* de modos de execução.

Quadro 8 – Arquivo de Configurações

```
1  [DATASET]
2  audios_dir = ./resources/clips_45seconds
3  y_path = ./resources/static_annotations.csv
4
5  [AUDIO]
6  length_in_seconds = 45
7  mono = True
8  sampling_rate = 44100
9
10
11 [MODEL]
12 algorithm = scikit.svr
13 name = 'default_model'
14
15
16 [FEATURE_EXTRACTION]
17 extract_mfcc = True
18 extract_spectral_centroid = True
19 extract_chromagram = True
20 extract_tempo = True
21 extract_tempogram = True
22 extract_zero_crossing_rate = True
23
24 transform_mfcc = True
25 transform_spectral_centroid = True
26 transform_chromagram = True
27 transform_tempogram = True
28 transform_zero_crossing_rate = True
29
30 n_mfcc_bands = 20
31
32
33 [CACHE]
34 dir = ./cache_dir
35 create_audio_files_cache = True
36 create_y_cache = True
37 create_x_cache = True
38 create_yhat_cache = True
39
40 [DATABASE]
41 initialization_script_path = ./resources/sql/init.sql
42
43 [EXECUTION]
44 cache = False
45 metrics = True
```

Fonte: Elaborado pelo autor.

O pré-processamento das características é feito após elas serem extraídas e se resume em aplicar o algoritmo Principal Component Analisys (PCA) e um algoritmo de normalização, conforme demonstrado no Quadro 9. A linha 6 demonstra a chamada do método apply_scaling, da biblioteca Scikit Learn. Este método escala os dados para se encaixarem em uma distribuição normal, onde os dados tenham média 0 e desvio padrão de 1. Dados nesta escala são recomendados para alguns algoritmos de regressão, como o Support Vector Regression (SVR) com kernel Radial Basis Function (utilizado neste trabalho), por exemplo, que pressupõe que todos os dados tenham uma distribuição normal.

A linha 10 demonstra a aplicação de um redutor de dimensionalidade, o PCA. O PCA é responsável por resumir os dados para uma versão menor com a menor perda de valor (significado do dado) possível.

Quadro 9 – Rotina de Pré-processamento de Características

```
1  def transform_features(self, X, pca=True, pca_numcomponents=180,
2  normalize=True):
3      # Support Vector Machines assume that all features are centered
4  around zero and have variance in the same order
5      if normalize:
6          X = self.apply_scaling(X)
7      if pca:
8          # reduce the array dimensionality to match the double audio
9  duration in seconds
10         X = self.apply_pca(X, pca_numcomponents)
11     # transform to 1d array
12     return self.ndarray_to_1dlist(X)
```

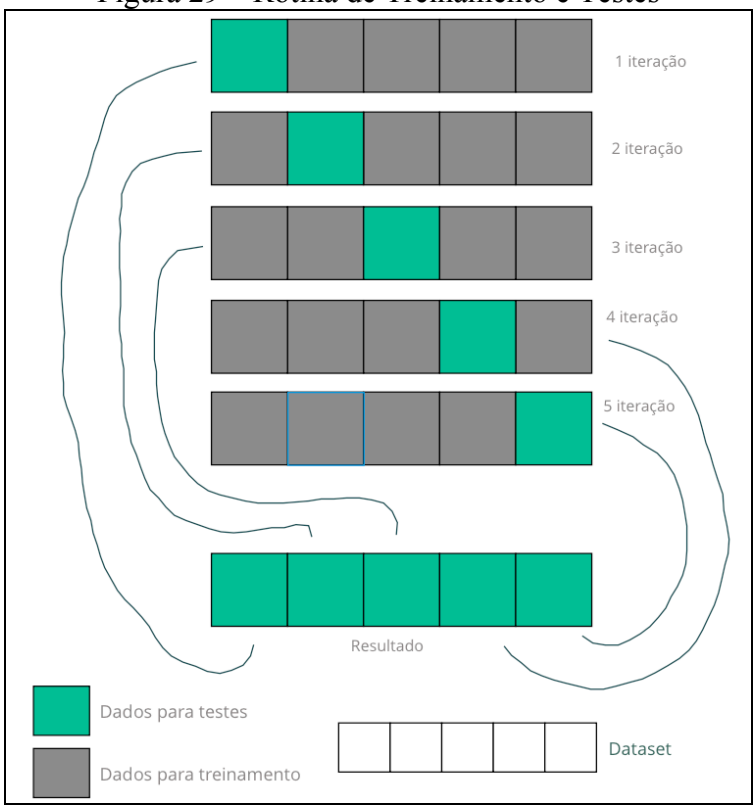
Fonte: Elaborado pelo autor.

3.3.2.3 Etapa 3: Rotinas de Treinamento e Testes de Algoritmos Regressores

A rotina de treinamento, exibida no Quadro 10, é responsável por instanciar os algoritmos de SVR, executar o treinamento destes algoritmos utilizando os dados do *dataset* de treinamento e executar a regressão. O loop for da linha 3 divide os dados do *dataset* utilizando o método de validação cruzada k-fold da biblioteca Scikit Learn, chamados de X_df e y_df. O X_df é um dataframe contendo as características extraídas das músicas do *dataset* de treinamento e o y_df é um dataframe contendo os valores de valência/alerta de cada uma das músicas. Utilizando o valor padrão do argumento k-fold (que é 5, conforme exibido na linha 1), o loop for irá iterar 5 vezes sobre o *dataset*, usando uma parte do *dataset* (quatro quintos, 4/5) para treinamento e outra parte (um quinto, 1/5) para testes. Este processo é realizado k vezes, neste caso 5 vezes, alternando de forma circular os subconjuntos de teste e de treinamento.

Em cada iteração, o dataframe df_list, instanciado na linha 2, é populado contendo os valores preditos pelo algoritmo SVR. A Figura 29 demonstra o funcionamento desta rotina de forma gráfica: a cada iteração uma parte do *dataset* é utilizada para treinar e outra para executar a regressão, os valores resultados da regressão são adicionados à um dataframe de resultado (representado na Figura 29 com o elemento Resultado).

Figura 29 – Rotina de Treinamento e Testes



Fonte: Elaborado pelo autor.

Quadro 10 – Rotina de Treinamento e Predição

```
1   def train_predict(X_df, y_df, kfolds=5):
2       df_list = []
3       for idx, X_train, X_test, y_train_valence, y_train_arousal,
4   y_test_valence, y_test_arousal in k_folds(X_df, y_df, kfolds):
5           clf_valence = svm.SVR()
6           clf_arousal = svm.SVR()
7           clf_valence.fit(X_train, y_train_valence)
8           clf_arousal.fit(X_train, y_train_arousal)
9
10          df = utils.create_yhat_df(idx,
11  clf_valence.predict(X_test).ravel(),
12  clf_arousal.predict(X_test).ravel())
13
14          df_list.append(df)
15
16      res = pd.concat(df_list)
17      return res
```

Fonte: Elaborado pelo autor.

3.3.3 Operacionalidade da implementação

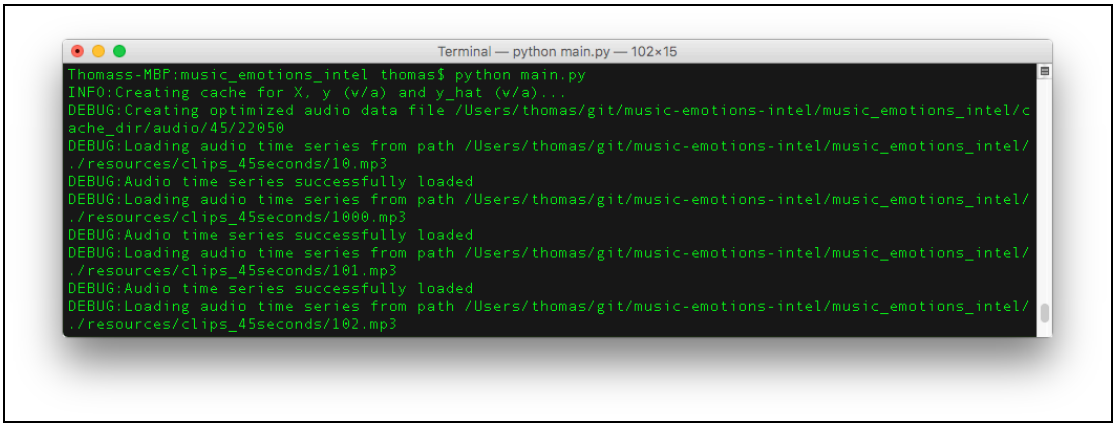
Esta seção demonstra a utilização da ferramenta desenvolvida neste trabalho a nível operacional, na perspectiva de um usuário e simulando um caso de uso. O caso de uso abordado é a detecção da lista de emoções presentes em uma música fornecida pelo usuário. Para atingir este objetivo é necessário primeiramente executar a fase de treinamento da

ferramenta e após isso a predição. A execução do treinamento é feita observando os seguintes passos:

a) baixar a ferramenta através do GitHub (serviço de versionamento e hospedagem de arquivos);
b) navegar até o diretório root do repositório (music-emotions-intel);
c) executar o comando `python main.py`.

A execução levará em consideração o perfil definido no arquivo de configurações `configs.ini` encontrado no diretório root do repositório. A Figura 30 demonstra a execução desta etapa.

Figura 30 – Utilização da ferramenta para treinamento



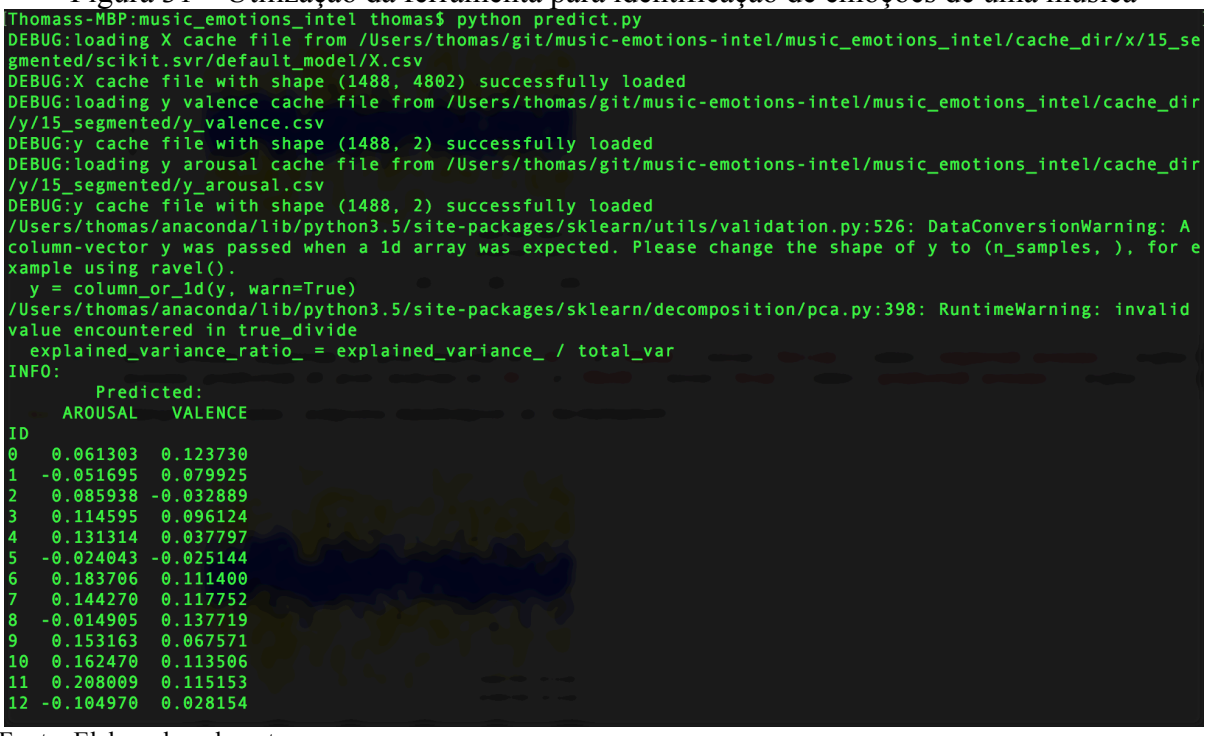
Fonte: Elaborado pelo autor.

Após o treinamento ter terminado com sucesso, é possível executar a identificação de emoções a partir de arquivos de áudio. Para tanto, siga os passos:

a) executar o comando `python predict.py -m CAMINHO_ARQUIVO_MUSICA`;
b) observar os resultados apresentados no terminal.

Um exemplo de resultado da execução de identificação de emoções de um arquivo de áudio pode ser visto na Figura 31, a qual apresenta, ao final da execução, uma tabela contendo as colunas ID, VALENCE e AROUSAL. A coluna ID contém os identificadores de cada segmento da música, a coluna AROUSAL contém os valores de alerta e a coluna VALENCE contém os valores de valência. Cada segmento tem a duração de 15 segundos e os valores limites para valência e alerta variam entre -0.5 e +0.5. A música processada na Figura 31 contém 13 segmentos, que resultam em 195 segundos.

Figura 31 – Utilização da ferramenta para identificação de emoções de uma música

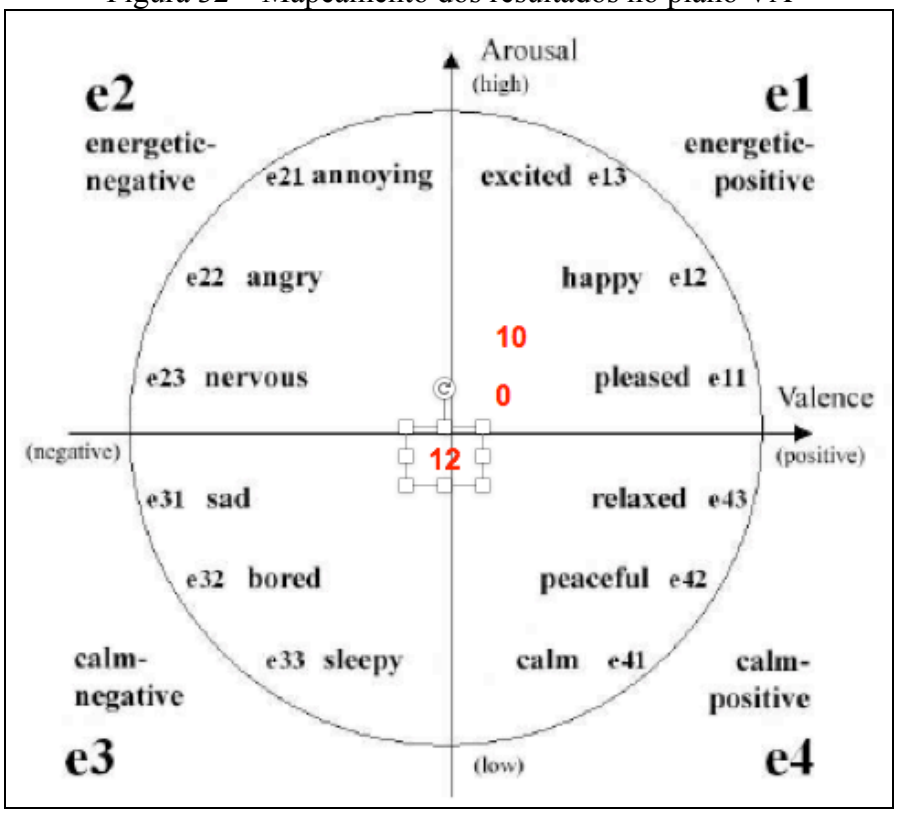


```
Thomass-MBP:music_emotions_intel thomas$ python predict.py
DEBUG:loading X cache file from /Users/thomas/git/music-emotions-intel/music_emotions_intel/cache_dir/x/15_se
gmented/scikit.svr/default_model/X.csv
DEBUG:X cache file with shape (1488, 4802) successfully loaded
DEBUG:loading y valence cache file from /Users/thomas/git/music-emotions-intel/music_emotions_intel/cache_dir
/y/15_segmented/y_valence.csv
DEBUG:y cache file with shape (1488, 2) successfully loaded
DEBUG:loading y arousal cache file from /Users/thomas/git/music-emotions-intel/music_emotions_intel/cache_dir
/y/15_segmented/y_arousal.csv
DEBUG:y cache file with shape (1488, 2) successfully loaded
/Users/thomas/anaconda/lib/python3.5/site-packages/sklearn/utils/validation.py:526: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for e
xample using ravel().
  y = column_or_1d(y, warn=True)
/Users/thomas/anaconda/lib/python3.5/site-packages/sklearn/decomposition/pca.py:398: RuntimeWarning: invalid
value encountered in true_divide
  explained_variance_ratio_ = explained_variance_ / total_var
INFO:
        Predicted:
     AROUSAL   VALENCE
ID
0   0.061303  0.123730
1  -0.051695  0.079925
2   0.085938 -0.032889
3   0.114595  0.096124
4   0.131314  0.037797
5  -0.024043 -0.025144
6   0.183706  0.111400
7   0.144270  0.117752
8  -0.014905  0.137719
9   0.153163  0.067571
10  0.162470  0.113506
11  0.208009  0.115153
12 -0.104970  0.028154
```

Fonte: Elaborado pelo autor.

Para obter uma representação emocional destes valores numéricos é necessário colocá-los no plano de Russel (1980), demonstrado na Figura 32. Por exemplo, o segmento com `ID 0` foi identificado com 0.06 de alerta e 0.12 de valência. O valor de alerta fica praticamente no centro do plano e o valor de valência ficaria levemente para a direita, na direção positiva. O resultado seria uma emoção quase neutra, pendendo para contente. Já o valor obtido no segmento 10 resulta em uma emoção mais próxima de feliz, que pode ser traduzido como uma versão com mais energia da emoção “contente”. O segmento 12 resulta em uma emoção tendenciando para “relaxado”.

Figura 32 – Mapeamento dos resultados no plano VA



Fonte: Elaborado pelo autor.

3.4 ANÁLISE DOS RESULTADOS

A análise de resultados foi dividida em duas partes: quantitativa e qualitativa. O cálculo da análise quantitativa é feito comparando-se dois números: o Mean Absolute Error (MAE) do *baseline*; e o MAE dos valores preditos pelo algoritmo, onde o MAE dos valores preditos deve ser maior que o MAE do *baseline*. A análise qualitativa demonstra porque a abordagem de identificação de emoções de músicas tendo como resultado um conjunto de emoções é mais eficaz que a abordagem de identificação de apenas uma única emoção.

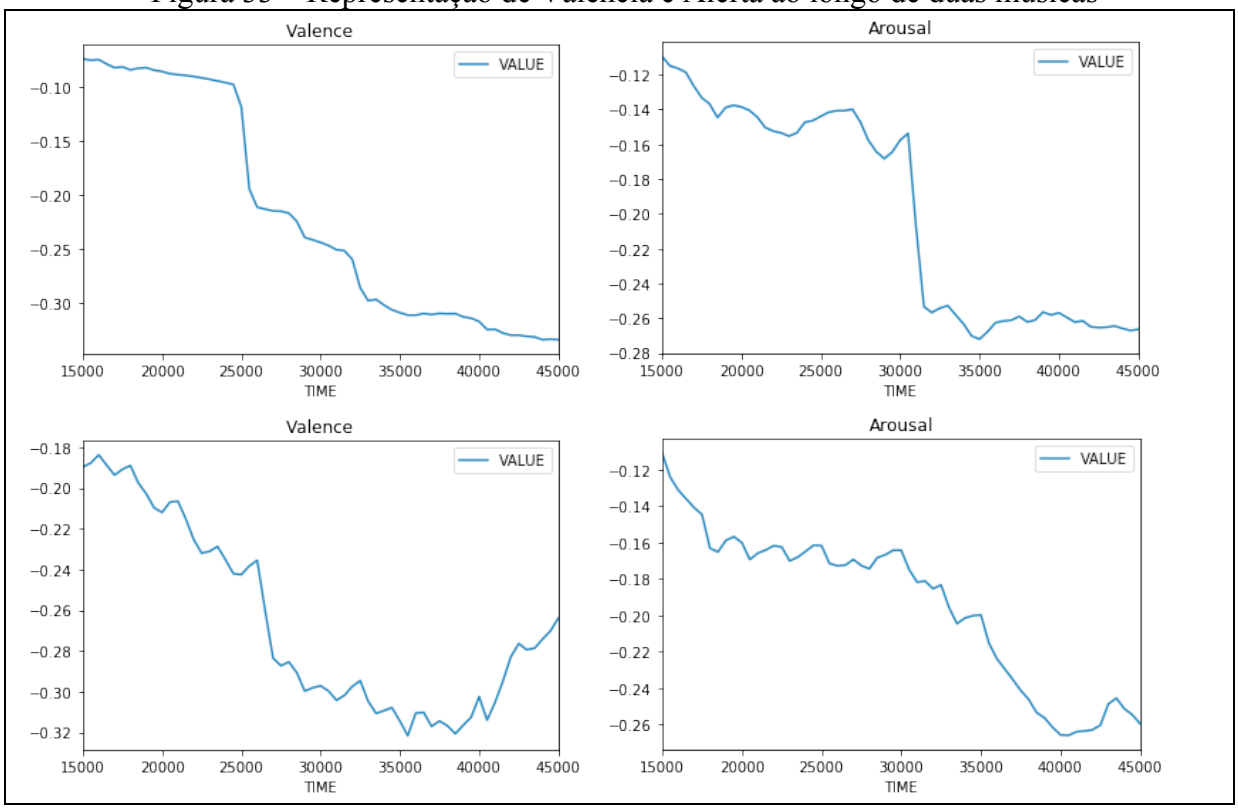
O *baseline* define a assertividade mínima para identificação de emoções, sendo composto pela média dos valores de valência e alerta de todas as músicas, enquanto o valor esperado é o valor de valência e alerta real de cada música. O MAE é uma medida de assertividade utilizada em algoritmos de regressão, servindo para medir o tamanho do desvio entre um dado valor com outro valor esperado.

3.4.1 Análise Qualitativa

A abordagem de identificação de emoções em músicas proposta por este trabalho difere dos trabalhos correlatos em um ponto principal: a representação de emoções de uma música. Os trabalhos de Trohidis (2008) e Wieczorkoowska (2006) utilizam uma abordagem de classificação, que por si só, se considerando o contexto de MER, acaba sendo menos eficaz que a regressão (KIM, 2010). Os trabalhos de Han (2009) e Schmidt, Tunrbull, Kim (2010) optaram por utilizar regressão, identificando também apenas uma emoção para cada música testada.

Independente do tipo de algoritmo, classificação ou regressão, o maior problema se encontra no fato de afirmar que uma música possui um sentimento. Segundo Meyer (1997), a música foi ganhando complexidade ao longo dos séculos, essa complexidade se traduz na quantidade de sentimentos expressada pela música e culmina na heterogeneidade de gêneros e estilos musicais atuais. Uma evidência a ser considerada é o próprio *dataset* utilizado para o treinamento do algoritmo de regressão utilizado neste trabalho: em uma mesma música a variação de VA é intensa. A Figura 33 exibe a variação de valência e alerta de duas músicas escolhidas aleatoriamente, onde o eixo Y representa o valor de valência no caso dos gráficos de *valence* e alerta no caso dos gráficos de *arousal* e o eixo X representa a duração da música.

Figura 33 – Representação de Valência e Alerta ao longo de duas músicas



Fonte: Elaborado pelo autor.

Ao observar os valores do eixo X, nota-se que estes gráficos demonstram a variação de VA de apenas 30 segundos das músicas (de 15.000 milissegundos à 45.000 milissegundos). Se a música inteira fosse considerada a variação seria ainda maior. O que os trabalhos correlatos fazem é definir um único valor para uma representação composta por vários valores.

A abordagem utilizada por este trabalho utiliza regressão para encontrar o conjunto de principais emoções identificadas em uma música. Esta abordagem é mais eficaz que a alternativa escolhida pelos trabalhos correlatos por poder representar mais assertivamente a natureza complexa de emoções das músicas.

3.4.2 Análise Quantitativa

Para realizar a análise quantitativa fez-se uso da medida Mean Absolute Error (MAE) a fim de mensurar a assertividade da ferramenta. O MAE, definido pela Figura 34, provê uma medida do tamanho do erro entre os valores preditos e os valores alvo. É definida como a média das diferenças absolutas dos valores esperados com os valores obtidos.

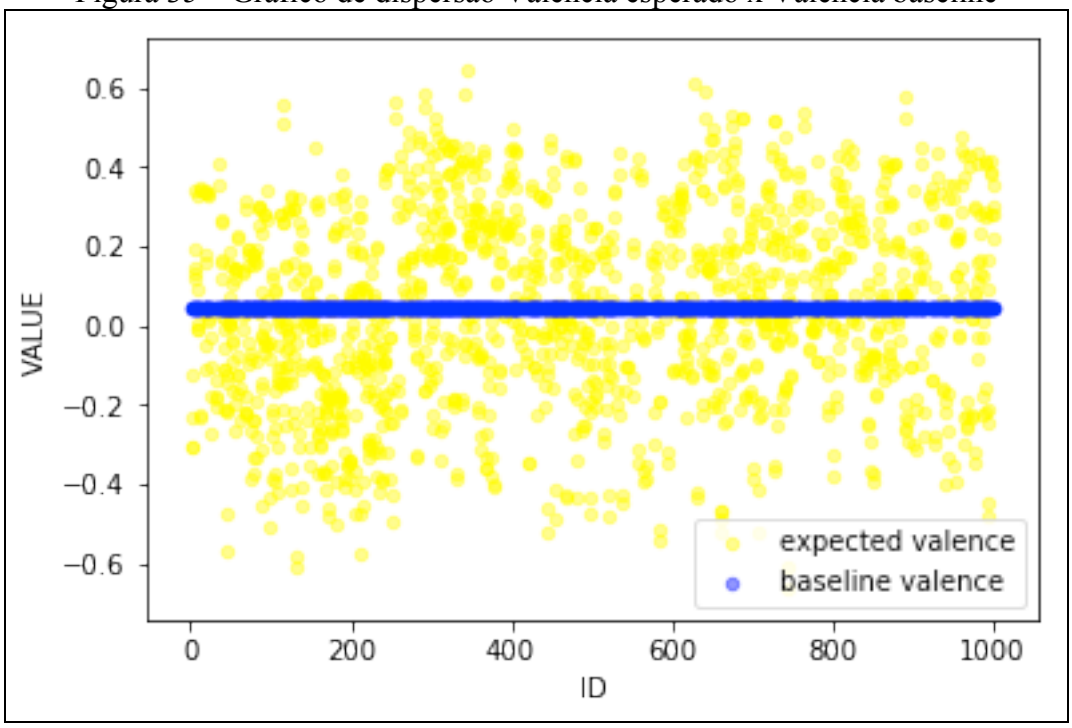
Figura 34 – Formula MAE

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |x_i - x|$$

Fonte: Elaborado pelo autor

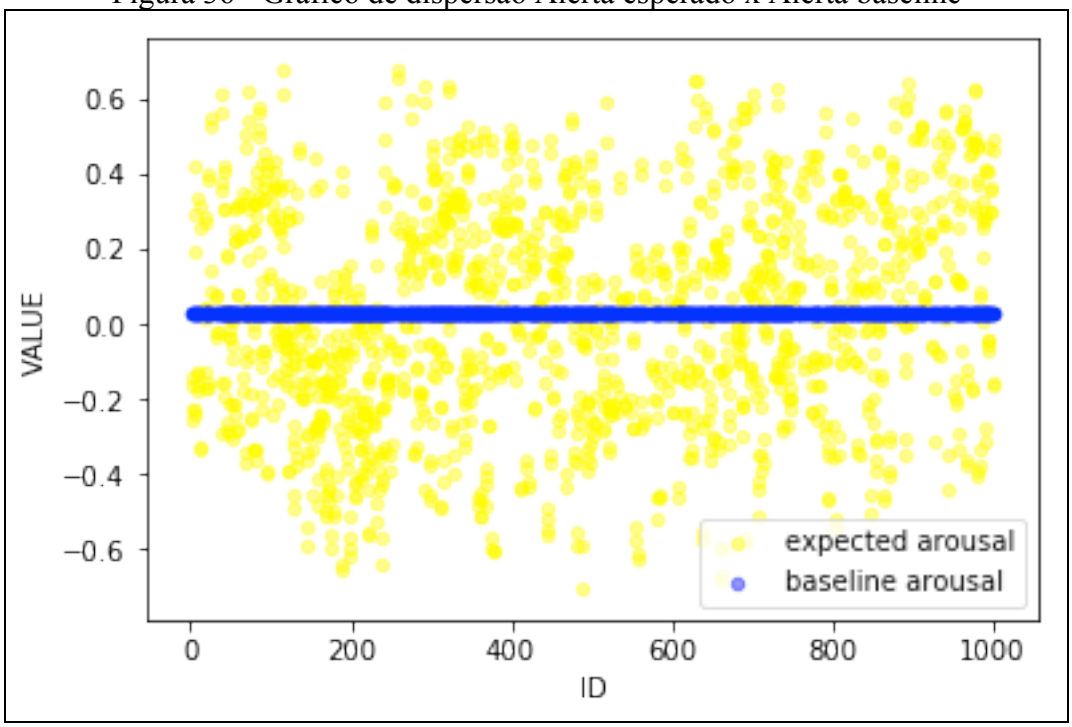
A Figura 35 exibe o gráfico de dispersão dos valores de valência esperados (*expected valence*) versus os valores de valência do *baseline* (*baseline valence*), proporcionando uma melhor visualização do tamanho do desvio entre os pontos esperados e o *baseline*. A Figura 36 demonstra a mesma informação, porém em relação ao alerta. Os gráficos de dispersão exibidos na Figura 35, Figura 36, Figura 37 e Figura 38 são compostos pelos valores de valência e alerta de cada um dos segmentos de áudio extraídos do dataset. Cada ponto é um segmento contendo um valor de valência ou alerta.

Figura 35 – Gráfico de dispersão Valência esperado x Valência baseline



Fonte: Elaborado pelo autor.

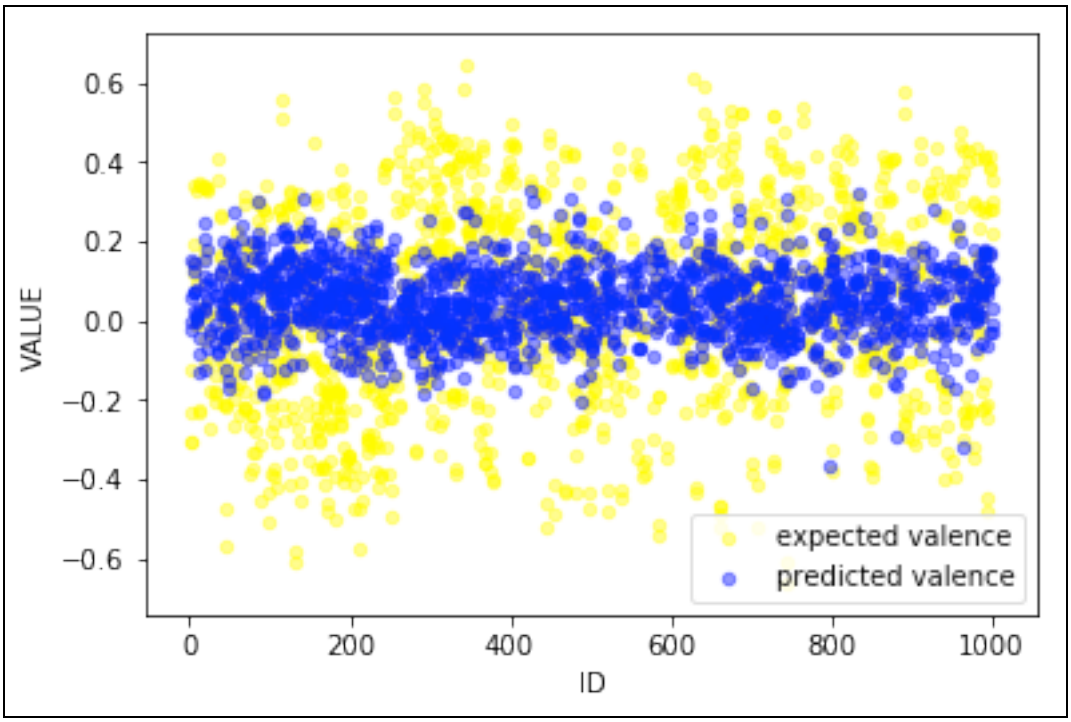
Figura 36 - Gráfico de dispersão Alerta esperado x Alerta baseline



Fonte: Elaborado pelo autor.

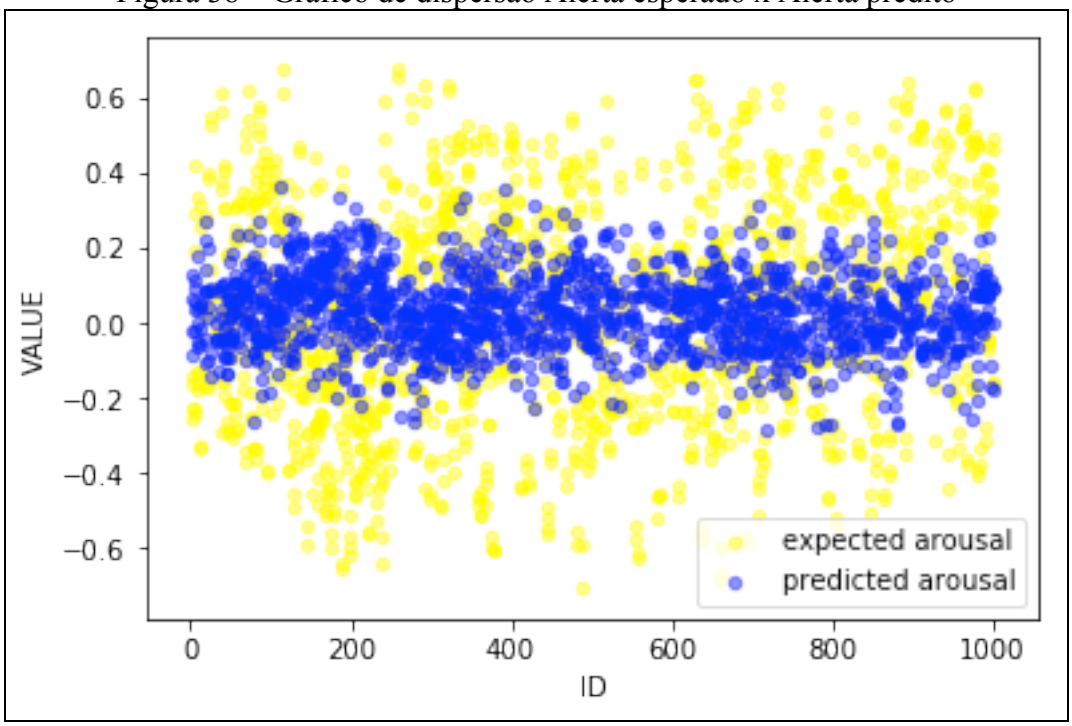
A Figura 37 exibe o gráfico de dispersão dos valores esperados (*expected*) e dos valores preditos pela ferramenta (*predicted*). A Figura 38 exibe a mesma informação, porém para o alerta. Estes gráficos dão uma ideia visual do tamanho do erro entre os valores preditos e os valores anotados (esperados).

Figura 37 – Gráfico de dispersão Valência esperada x Valência predita



Fonte: Elaborado pelo autor.

Figura 38 – Gráfico de dispersão Alerta esperado x Alerta predito



Fonte: Elaborado pelo autor.

A Tabela 4 compara os resultados obtidos por este trabalho com os resultados obtidos pelo trabalho de Huq (2010). A coluna Fonte define qual é a fonte dos resultados: HUQ para o trabalho de Huq (2010) e MEI (Music Emotions Intel) para este trabalho. As colunas B.

Valência e B. Alerta representam os valores de MAE dos *baselines* de valência e alerta, as colunas Valência e Alerta representam os valores de MAE dos valores preditos pelos algoritmos. As colunas mais importantes para a interpretação dos resultados são, respectivamente, % B.Valência e % B.Alerta. Cada uma destas colunas exibe qual a porcentagem de diminuição do erro do MAE dos valores preditos com o MAE dos *baselines*. O algoritmo de HUQ conseguiu diminuir 16.10% do MAE do *baseline* de valência e 57.26% do MAE do alerta *baseline*. Este trabalho conseguiu um resultado de 7.5% maior de erro para a valência em relação ao *baseline* e 8.097% maior para o alerta em relação ao *baseline*.

O principal motivo que fez com que a assertividade deste trabalho ficasse abaixo do *baseline* é o baixo número de experimentos feitos. Durante a elaboração deste trabalho foram executados diversos experimentos utilizando os algoritmos de regressão linear e de SVR, onde foram testados os diferentes kernels. Porém, a opção com maior assertividade, corroborando com a conclusão de Huq (2010), foi algorítmo de SVR com kernel RBF.

Tabela 4 – Comparação entre MAE

Fonte	B.Valência	B.Alerta	Valência	Alerta	% B.Valência	% B.Alerta
HUQ (2010)	0.236	0.365	0.198	0.156	16.102	57.260
MEI	0.200	0.247	0. 215	0. 267	-7.5	- 8.097

Fonte: elaborado pelo autor.

A Tabela 5 faz uma relação dos perfis de execução experimentados. A coluna bitrate informa qual o número de amostras por segundo foram utilizadas para representar as músicas. A coluna PCA indica se o pré-processamento PCA foi aplicado sobre os vetores de características. A coluna Escala indica se foi aplicado o pré-processamento de escala nos vetores de característica, que transforma todos os valores para uma distribuição normal (média 0 e desvio padrão 1). As colunas Valência e Alerta indicam qual o MAE obtido utilizando-se os perfis relacionados. A coluna Tempo indica qual o tempo demandado pela execução. Por fim, as colunas % B.Valência e % B.Alerta indicam a porcentagem de diminuição do MAE obtido pelo perfil indicado em relação ao MAE obtido pelo *baseline*.

É possível verificar que as execuções com uma diminuição maior do erro são dos perfis 3 e 6. O perfil 3, com bitrate de 22050, apresenta um erro 0.92% maior para valência e 1.11% maior para alerta, se comparado com o perfil 6, com bitrate de 44100. Em contrapartida, o tempo de execução do perfil 6 é 70.20% maior. Nota-se também que quando utilizado um bitrate de 44100, o tempo de execução aumenta consideravelmente em todas as execuções, sem ganhos palpáveis na diminuição do erro. Esta tabela mostra que o fator que

mais influencia na diminuição de erro é a aplicação de Escala, seguido da aplicação de PCA e por fim pela representação dos áudios em 44100.

Tabela 5 – Comparação entre perfis de execução

Perfil	Bitrate	PCA	Escala	Valência	Alerta	Tempo	% B.Valência	% B.Alerta
1	22050	N	N	0.301	0.354	1’12’’	-50.5	-43.32
2	22050	S	N	0.278	0.331	57’’	-39	-34.008
3	22050	S	S	0.215	0.267	40’’	-7.5	- 8.097
4	44100	N	N	0.308	0.359	4’4’’	-54	-45.344
5	44100	S	N	0.281	0.334	3’18’’	-40.5	-35.222
6	44100	S	S	0.217	0.270	2’31’’	-8.5	-9.312

Fonte: elaborado pelo autor.

A Tabela 6 exibe um comparativo entre os trabalhos correlatos e este trabalho. É possível verificar que este trabalho é o único que utilizou o conceito de análise da música inteira. O trabalho de Huq (2010), salvo o fato de considerar apenas um segmento da música e possuir um vetor de características menor, possui todos os pontos comparados iguais a este trabalho. O único trabalho que não utilizou o plano de valência e alerta foi o de Wieczorkowsk (2009). Os trabalhos de Yang, Liu, Chen (2006) e Wieczorkowsk (2009) não utilizaram SVR e, finalmente, este trabalho possui o maior número de característica dentre todos. Isso é outro grande fator que pode influenciar na baixa assertividade quantitativa: o *dataset* possui apenas 744 músicas utilizadas para o treinamento, porém o vetor de características possui 4801 características. Esta é uma descrição precisa de uma situação de *overfitting*. *Overfitting* faz com que o algoritmo preditor não consiga generalizar bem, gerando baixa performance em dados nunca vistos e performando bem apenas nos dados de treinamento.

Tabela 6 – Comparação entre os trabalhos correlatos

Característica	YANG; LIU; CHEN, 2006	WIECZO-RKOWSK, 2009	YANG, 2008	HUQ, 2010	Este Trabalho
Considera músicas inteiras					X
Utiliza plano de Russel	X		X	X	X
Tipo de algoritmo	Classificação difusa	Classificação multi-classes	Regressão	Regressão	Regressão
Algoritmo	RAKEL	FKN e FNM	SVR	SVR	SVR
Numero de características	72	15	114	160	4801

Fonte: Elaborado pelo autor.

4 CONCLUSÕES

Os objetivos principais deste trabalho foram o desenvolvimento de um *script* para extração de características relevantes para o reconhecimento de emoções em músicas e a utilização de uma abordagem de interpretação das músicas considerando sua duração total. Esta abordagem consiste em identificar um conjunto de emoções por música, diferente da forma tradicional utilizada, que identifica uma emoção por música. Este objetivo foi atingido, onde as músicas são divididas em segmentos de 15 segundos e os valores de valência e alerta são extraídos a partir destes segmentos.

Apesar do objetivo principal ter sido atingido, a assertividade final obtida com o algoritmo de regressão utilizado foi baixa, isto é, com um resultado inferior da assertividade utilizando o *baseline*. Isso se deve ao baixo número de experimentos feitos. Um motivo que contribuiu para o baixo número de experimentos foi o esforço gasto em rotinas desnecessárias da ferramenta, como a rotina de persistência das métricas em uma base de dados. Inicialmente entendeu-se que seria valiosa a funcionalidade de persistir métricas em uma base de dados, em contraste de apenas escreve-las na console, porém ao final do trabalho notou-se que a rotina foi pouco usada e tomou grande parte do tempo de desenvolvimento.

Além do tempo investido erroneamente, a demora no tempo de treinamento do algoritmo também impactou para o baixo número de experimentos executados. As execuções de treinamento levavam de 45min à 4h em uma máquina com as seguintes configurações: CPU Core i7 2.5GHz, HD SSD, 16 GB de LPDD3 1866MHz, MacOS X. Esta duração variava de acordo com número de características utilizadas e com as configurações do algoritmo de redução de dimensionalidade (PCA). O problema de baixa assertividade pode ser resolvido fazendo-se mais experimentos com configurações variadas, como, por exemplo: além dos kernels testados (RBF e Sigmoid), kernels de SVR diferentes, algoritmos diferentes, configurações de redução de dimensionalidade diferentes, características adicionais, etc.

A contribuição principal deste trabalho é o modelo de representação emocional composto, em contraste do modelo de representação singular. Conforme observado na análise de resultados qualitativa, uma mesma música possui várias emoções ao longo de sua duração. Com este modelo é possível cobrir este aspecto dinâmico das emoções em músicas, podendo servir de base para diferentes algoritmos de aprendizado de máquina relacionados a área, bem como ser estendido para outras abordagens deste domínio.

4.1 EXTENSÕES

O trabalho apresentado nesta monografia atingiu os objetivos propostos. Entretanto,

alguns aspectos podem ser melhorados para a continuidade do trabalho:

a) fazer experimentos com características adicionais, como High Frequency Content, Sones, Overall Loudness, Root Mean Square, TriadSeq e Complex Domain Onset Detection;
b) fazer experimentos utilizando o áudio em modo estéreo;
c) fazer experimentos com tamanhos de segmentos diferentes dos 15 segundos estipulados neste trabalho;
d) testar abordagens de redução de dimensionalidade diferentes, ou ainda utilizar configurações adicionais de PCA;
e) experimentar algoritmos diferentes de SVR, como, por exemplo redes neurais;
f) experimentar diferentes kernels do RBF no SVR, como, por exemplo: linear ou polinomial.

REFERÊNCIAS

APPALACHIAN STATE UNIVERSITY. **Sound waves.** Disponível em: <http://www.appstate.edu>. Acesso em: 01 maio 2017.

BACHU, R. G. et al. Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal. In: **American Society for Engineering Education (ASEE) Zone Conference Proceedings**. 2008. p. 1-7.

BERG, Richard E.. **The Physics of Sound.** New York: Prentice Hall College Div, 1982. 370 p.

BISPHAM, John. Rhythm in Music: What is it? Who has it? And Why? **University Of California Press.** California, p. 125-134. dez. 2006.

COOK, John D. **Cepstrum, quefrency, and pitch.** 2016. Disponível em: <https://www.johndcook.com/blog/2016/05/18/cepstrum-quefrency-and-pitch/>. Acesso em: 20 maio 2017.

CROSSLEY-HOLLAND. **Rhythm.** 2016. Disponível em: <https://www.britannica.com/art/rhythm-music#toc64635>. Acesso em: 05 maio 2014.

GROSCHE, Peter; MÜLLER, Meinard. Tempogram toolbox: Matlab implementations for tempo and pulse analysis of music recordings. In: **Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)**, Miami, FL, USA. 2011.

HAN, Byeong-jun et al. SMERS: Music Emotion Recognition Using Support Vector Regression. In: MUSIC INFORMATION RETRIEVAL CONFERENCE, 1., 2009, Japan. **Proceedings of the 10th International Society for Music Information Retrieval Conference.** Japan: Dblp, 2009.

HASAN, Md. Rashidul et al. Speaker Identification Using Mel Frequency Cepstral Coefficients. In: **International Conference On Electrical & Computer Engineering**, 3., 2004**.** Dhaka: Icece, 2004. p. 28 - 30.

HEVNER, Kate. The affective character of the major and minor modes in music. **The American Journal Of Psychology.** Illinois, p. 103-118. jan. 1935.

HUQ, Arefin; BELLO, Juan Pablo; ROWE, Robert. Automated Music Emotion Recognition: A Systematic Evaluation. **Journal Of New Music Research.** New York, p. 227-244. jan. 2010.

IFTENE, Adrian; RUSU, Andrei; LEAHU, Alexandra. Music Identification Using Chroma Features. In: **CLEF (Notebook Papers/Labs/Workshop)**. 2011.

IZARD, Carroll E. **Human emotions**. Springer Science & Business Media, 2013.

IZARD, Carroll E.. Stability of emotion experiences and their relations to traits of personality. **Journal Of Personality And Social Psychology.** New York, p. 847-860. maio 1993.

KIM, Youngmoo e et al. **Music emotion recognition: A state of the art review.** Philadelphia: Ismir, 2010.

KINSLER, Lawrence E. et al. **Fundamentals of Acoustics.** 4. ed. New Jersey: Wiley, 1999. 560 p.

MENON, Vinod et al. Neural correlates of timbre change in harmonic sounds. **Neuroimage**, v. 17, n. 4, p. 1742-1754, 2002.

MEYER, Leonard B.. **Style and music: Theory, history, and ideology.** Chicago: University Of Chicago Press, 1997. 385 p.

NAKAYAMA, T.. **Speech Recognition based on phoneme.** 2008. Disponível em: <http://www.geocities.jp/voiceofkaijin/>. Acesso em: 01 maio 2017.

NAM, Unjung. **Spectral Centroid.** 2001. Disponível em: <https://ccrma.stanford.edu/~unjung/AIR/areaExam.pdf>. Acesso em: 15 maio 2017.

PROCOPIO, Joe. **Basic Music Theory.** New Jersey: Bookbaby, 2016. 80 p.

RUSSEL, J. A. A circumplex model of affect. **J. Personality and Social Psychology**, v. 39, p. 1161-78, 1980.

RUSSELL, James A.; BARRETT, Lisa Feldman. Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. **Journal of personality and social psychology**, v. 76, n. 5, p. 805, 1999.

SCHMELING, Paul. **Berklee Music Theory.** 2. ed. Boston: Berklee, 2011. 120 p.

SCHUBERT, Emery; WOLFE, Joe. Does Timbral Brightness Scale with Frequency and Spectral Centroid? **Acta Acustica United With Acustica.** Stuttgart, p. 820-825. out. 2006.

SCHOENBERG, Arnold. **Fundamentals of Musical Composition.** Londres: Faber & Faber, 1999. 240 p.

SCHMIDT, Erik M.; TURNBULL, Douglas; KIM, Youngmoo E.. Feature selection for content-based, time-varying musical emotion regression. In: INTERNATIONAL CONFERENCE ON MULTIMEDIA INFORMATION RETRIEVAL, 1., 2010, Philadelphia. **Proceedings of the international conference on Multimedia information retrieval.** New York: Acm, 2010. v. 1, p. 267 - 274.

SOLEYMANI, Mohammad et al. 1000 songs for emotional analysis of music. In: CROWDMM, 1., 2013, Barcelona. **Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia.** New York: Acm, 2013. v. 1, p. 1 - 6.

THAUT, Michael. **Rhythm, music, and the brain:** Scientific foundations and clinical applications. Abingdon: Routledge, 2005. 272 p.

TOM HENDERSON (Illinois). **Pitch and Frequency.** 2016. Disponível em: <http://www.physicsclassroom.com/class/sound/Lesson-2/Pitch-and-Frequency>. Acesso em: 20 maio 2017.

TOM HENDERSON (Illinois). **Sound is a Pressure Wave.** 2016. Disponível em: <http://www.physicsclassroom.com/Class/sound/u1111c.html>. Acesso em: 20 maio 2017.

TOMKINS, Silvan S.. **Affect Imagery Consciousness: The Positive Affects.** New York: Springer Publishing Company, 1962. 522 p.

TROHIDIS, Konstantinos et al. **Multi-Label Classification of Music into Emotions**. In: ISMIR. 2008. p. 325-330.

TUTOR VISTA. **Longitudinal Wave.** Disponível em: <http://physics.tutorvista.com/waves>. Acesso em: 30 maio 2017.

SCHMIDT, Erik M.; TURNBULL, Douglas; KIM, Youngmoo E. Feature selection for content-based, time-varying musical emotion regression. In: **Proceedings of the international conference on Multimedia information retrieval**. ACM, 2010. p. 267-274.

YANG, Yi-Hsuan; LIU, Chia-Chu; CHEN, Homer H. **Music emotion classification: a fuzzy approach**. In: Proceedings of the 14th ACM international conference on Multimedia. ACM, 2006. p. 81-84.

WIECZORKOWSKA, Alicja; SYNAK, Piotr; RAŚ, Zbigniew W. **Multi-label classification of emotions in music**. In: Intelligent Information Processing and Web Mining. Springer Berlin Heidelberg, 2006. p. 307-315.