

UNIVERSIDADE FEEVALE

GREGÓRIO CHALINSKI KUSOWSKI

REPRESENTAÇÃO VISUAL DE SEGMENTOS DE ÁUDIO

Novo Hamburgo

2016

GREGÓRIO CHALINSKI KUSOWSKI

REPRESENTAÇÃO VISUAL DE SEGMENTOS DE ÁUDIO

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel em
Ciência da Computação pela
Universidade Feevale

Orientador: João Batista Mossmann

Novo Hamburgo

2016

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização deste trabalho de conclusão, em especial:

A minha esposa, por todo apoio incondicional em tantos momentos da vida, do curso e do desenvolvimento do trabalho. Aos familiares, que suportaram e entenderam meu sumiço. Aos amigos e às pessoas que convivem comigo quase que diariamente, pela paciência, contribuições e apoio durante toda essa jornada. Ao time de profissionais da Universidade Feevale, que estiveram presentes nessa caminhada tanto acadêmica quanto profissional, e contribuíram com toda formação obtida.

Enfim, agradeço a todos aqueles que tiveram a oportunidade de cruzar o meu caminho e foram capazes de mudar minha vida em algum sentido, seja com uma ação conjunta que durou meses, ou seja com um sorriso no rosto. Considero estas pessoas especiais, e espero ser especial para quem cruzar o meu caminho.

RESUMO

Com o aumento constante no montante de informações geradas diariamente nas mais diversas áreas do conhecimento humano, juntamente com a ineficiência do ser humano na interpretação de grandes volumes de dados, verifica-se uma dificuldade para análise e gestão do conteúdo capturado de forma rápida. Neste sentido, a utilização de gráficos para exibição das informações favorece o reconhecimento e a percepção de características. No contexto de coleções musicais, algumas ferramentas exibem representações de ondas do conteúdo de músicas em listas, possibilitando ao usuário uma prévia visual dos conteúdos de áudio. Entretanto, as formas de ondas de um segmento de som não são capazes de representar muita informação sobre o mesmo. Neste contexto, o trabalho aborda o estudo da interpretação de segmentos digitais de áudio e técnicas de visualização de dados. Sendo assim, investigou-se um modelo de representação visual de áudio a partir da interpretação de sinais e técnicas de visualização. No contexto deste estudo, foi realizada uma avaliação com 52 usuários com objetivo de analisar a capacidade de expressar visualmente informações sobre um dado segmento de áudio.

Palavras-chave: Visualização Computacional. Recuperação de Informação de Música. Visualização. Análise de Áudio. Processamento de Áudio.

ABSTRACT

With the constant growing in the creation rate of all information on diverse human knowledge areas, joined with the human inefficiency on interpreting huge volumes of data, a difficult to analyze and manage the gathered data in a fast manner is checked. This way, the use of graphics to show data favors the recognition and perception of characteristics. In the context of musical collections, some tools show wave representations of each music in lists, enabling the users to have a visual preview of its audio content. However, the waveform of a sound segment is not able to represent much information about it. In this context, the work addresses the study of the interpretation of digital audio segments and data visualization techniques. Therefore, it investigates a model that visually represents audio, based on signal interpretation and visualization techniques. In the context of this study, a survey was applied to 52 users in order to analyze the model capability to visually manifest information about a given audio segment.

Keywords: Computational Visualization. Music Information Retrieval. Visualization. Audio Analysis. Audio Processing.

LISTA DE FIGURAS

Figura 1.1 – Segmentos de Jazz apresentados pelo Improviz.....	22
Figura 1.2 – Quatro segundos de áudio são transformados em um objeto em três dimensões.....	23
Figura 1.3 – Stacked Graphs.....	25
Figura 3.1 – <i>Flickr Flow</i>	34
Figura 3.2 – <i>Pipeline</i> para Visualização de Áudio	36
Figura 3.3 – Proposta <i>Cubehelix</i> para escala de cores	38
Figura 4.1 – Mapa de calor gerado a partir da análise <i>amplitudeSpectrum</i>	52
Figura 4.2 – Mapa de calor gerado a partir da análise <i>powerSpectrum</i>	52
Figura 4.3 – Mapa de calor contendo valores positivos da análise MFCC.....	53
Figura 4.4 – Mapa de calor contendo o módulo de valores negativos da análise MFCC.....	53
Figura 4.5 – Gráfico empilhado com linha base zero a partir da análise <i>amplitudeSpectrum</i>	55
Figura 4.6 – Gráfico empilhado equilibrado simetricamente a partir da análise <i>amplitudeSpectrum</i>	55
Figura 4.7 – Gráfico empilhado equilibrado simetricamente a partir da análise <i>powerSpectrum</i>	56
Figura 4.8 – Gráfico empilhado equilibrado simetricamente a partir da análise MFCC.....	56
Figura 4.9 – Um conjunto de dados não ordenados, onde todas alterações abruptas causam uma alteração geométrica significativa.....	56
Figura 4.10 – O mesmo conjunto de dados, com ordenação simples dos dados, causando efeito de faixas diagonais.....	57
Figura 4.11 – O mesmo conjunto de dados, com ordenação que utiliza a estratégia de balanceamento “de dentro para fora”, com resultado esteticamente mais agradável	57
Figura 4.12 – <i>Radial Streamgraph</i> do <i>amplitudeSpectrum</i> de um segmento de áudio de 2 segundos.....	60
Figura 4.13 – <i>Radial Streamgraph</i> do <i>powerSpectrum</i> de um segmento de áudio de 2 segundos.....	61
Figura 4.14 – <i>Radial Streamgraph</i> do MFCC de um segmento de áudio de 2 segundos.....	62
Figura 4.15 – Camadas semitransparentes de um <i>Radial Streamgraph</i> do MFCC de um segmento de áudio de 2 segundos.....	63
Figura 4.16 – Camadas semitransparentes ordenadas decrescentemente de um <i>Radial Streamgraph</i> do MFCC de um segmento de áudio de 2 segundos.....	64

Figura 4.17 – <i>Radial Streamgraph</i> do MFCC de um segmento de áudio de 2 segundos com camadas ordenadas decrescentemente	66
Figura 5.1 –Explicação do modelo utilizada na primeira etapa da pesquisa	68
Figura 5.2 – Explicação sobre a animação de ponteiro utilizada no treinamento.....	69
Figura 5.3 – Gráfico que demonstra a relação dos participantes com Tecnologia da Informação	71
Figura 5.4 – Gráfico que demonstra a relação dos participantes com Áudio	71
Figura 5.5 – Gráfico que demonstra a relação dos participantes com Música	72
Figura 5.6 – Gráfico que exhibe comparativo entre o nível escolar dos participantes	72
Figura 5.7 – Gráfico que soma a quantidade de respostas por questão da avaliação dos participantes	75
Figura 5.8 – Gráfico que mostra a soma da quantidade de participantes pela quantidade de acertos nas questões avaliativas	76
Figura 5.9 – Gráfico que soma a quantidade de respostas por questão da avaliação dos participantes	77
Figura 5.10 – Gráfico que exhibe a média do total de acertos por faixa etária	77
Figura 5.11 – Gráfico que exhibe a média de acertos por faixa etária para cada questão avaliativa.....	77
Figura 5.12 – Gráfico que exhibe a média do total de acertos por Nível de Experiência em TI.....	78
Figura 5.13 – Gráfico que exhibe a média de acertos por Nível de Experiência em TI para cada questão avaliativa.....	78
Figura 5.14 – Gráfico que exhibe a média do total de acertos por Nível de Experiência em TI considerando apenas participantes que apontaram experiência profissional.....	79
Figura 5.15 – Gráfico que exhibe a média do total de acertos por Nível de Experiência em Áudio.....	79
Figura 5.16 – Gráfico que exhibe a média de acertos por Experiência em Áudio para cada questão avaliativa.....	79
Figura 5.17 – Gráfico que exhibe a média do total de acertos por Experiência em Música.....	80
Figura 5.18 – Gráfico que exhibe a média de acertos por Experiência em Música para cada questão avaliativa.....	80
Figura 5.19 – Gráfico que exhibe a média do total de acertos por nível escolar	80
Figura 5.20 – Gráfico que exhibe a média de acertos por nível escolar para cada questão avaliativa.....	81

LISTA DE QUADROS

Quadro 4.1 - Exemplo de código na linguagem JavaScript com a biblioteca Meyda para coletar características de áudio	50
--	----

LISTA DE TABELAS

Tabela 5.1 – Resultados das respostas do questionário para a representação 1.....	73
Tabela 5.2 – Resultados das respostas do questionário para a representação 2.....	73
Tabela 5.3 – Resultados das respostas do questionário para a representação 3.....	74
Tabela 5.4 – Resultados das respostas do questionário para a representação 4.....	74
Tabela 5.5 – Resultados das respostas do questionário para a representação 5.....	74
Tabela 5.6 – Resultados das respostas do questionário para a representação 6.....	75
Tabela 5.7 – Resultados das respostas do questionário para a representação 7.....	75

LISTA DE ABREVIATURAS E SIGLAS

AD	Analógico Digital
API	<i>Application Programming Interface</i>
CLI	<i>Command Line Interface</i>
CSS	<i>Cascading Style Sheets</i>
D3	<i>Data-Driven Documents</i>
DCT	<i>Discrete Cosine Transform</i>
DFT	<i>Discrete Fourier Transform</i>
DOM	<i>Document Object Model</i>
DSL	<i>Domain Specific Language</i>
DSP	<i>Digital Signal Processing</i>
FFT	<i>Fast Fourier Transform</i>
GUI	<i>Graphical User Interface</i>
HTML	<i>HyperText Markup Language</i>
MAM	<i>Music Animation Machine</i>
MFCC	<i>Mel Frequency Cepstral Coefficient</i>
MIDI	<i>Musical Instrument Digital Interface</i>
MIR	<i>Music Information Retrieval</i>
NPM	<i>Node Package Manager</i>
PCM	<i>Pulse Code Modulation</i>
SVG	<i>Scalable Vector Graphics</i>
TI	Tecnologia da Informação
TTT	<i>Task Type by Taxonomy</i>

SUMÁRIO

INTRODUÇÃO.....	13
1 VISUALIZAÇÃO DE DADOS.....	18
1.1 Etapas para gerar Representações Visuais de Dados.....	19
1.2 A pesquisa para visualização de informação	20
1.2.1 Visualização de áudio	21
1.2.2 Trabalhos relacionados	22
1.3 A Pesquisa na Arte.....	23
1.4 Geometria e Estética	24
2 ANÁLISE DE ÁUDIO.....	27
2.1 Conversão de Áudio.....	27
2.2 Propriedades e Características do Som	29
2.3 Extração de Características	31
2.4 Utilização de Características.....	32
3 PROPOSTA DE DESENVOLVIMENTO	34
3.1 Definição de <i>pipeline</i>	35
3.2 Proposta para Implementação	36
4 DESENVOLVIMENTO DO PROTÓTIPO.....	42
4.1 Requisitos.....	42
4.2 Seleção das ferramentas de desenvolvimento e prototipação	43
4.2.1 Extração de características de áudio	44
4.2.2 Apoio para prototipação	46
4.2.3 Visualização de Dados	47
4.3 Implementação.....	50
4.4 Consolidação da Implementação	64
5 AVALIAÇÃO.....	67
5.1 Metodologia	67
5.2 Perfil dos participantes.....	70
5.3 Avaliação	72
5.4 Análise geral	76
6 CONSIDERAÇÕES FINAIS.....	82
6.1 Contribuições e Aprendizado.....	83

6.2 Trabalhos futuros	84
REFERÊNCIAS BIBLIOGRÁFICAS.....	86
ANEXO 1 – E-MAIL DE CONFIRMAÇÃO DA SUBMISSÃO DE ARTIGO PARA A	
REVISTA COLLOQUIUM EXACTARUM	90
APÊNDICE 1 – APRESENTAÇÃO DA PESQUISA DE AVALIAÇÃO E	
TREINAMENTO SOBRE O MODELO PROPOSTO.....	91
APÊNDICE 2 – QUESTIONÁRIO PARA AVALIAÇÃO DO MODELO	97

INTRODUÇÃO

Com o aumento constante da automação dos processos nas mais diversas áreas do conhecimento humano, verifica-se a geração de um volume muito grande de informação. Sabe-se da ineficiência do ser humano na interpretação de grandes volumes de dados, sejam na forma numérica ou textual. Neste contexto, a exibição de dados em forma gráfica facilita o reconhecimento e a percepção de um conjunto grande de características, por estarem apresentados de maneira mais organizada (BOTELHO, 2002).

Neste sentido, a visualização de informação pode estimular naturalmente a percepção e inteligência humana. Ela permite que os seres humanos, ao interpretarem um grande volume de dados, apliquem uma combinação de flexibilidade, criatividade e conhecimento geral para apoio em tomadas de decisão. Logo, o resultado a partir de grandes bases de dados, quando exibido em forma gráfica, faz com que a capacidade de análise em gráficos do ser humano seja aplicada na descoberta de informações (BOTELHO, 2002).

Dentre as técnicas utilizadas para a exibição de dados, uma das mais aplicadas é a projeção geométrica, que consiste na plotagem de atributos e valores em um plano bidimensional. Para atuar com vários atributos, também existe a técnica de apresentação icônica, que visa mapear cada item em um ícone que será exibido em conjunto com as outras entidades, porém este formato possui uma limitação relacionada à quantidade de dados (BOTELHO, 2002). Semelhante à forma icônica, existe a exibição orientada a pixel, que se baseia na separação de uma tela em janelas para cada dimensão/atributo, em que cada registro é representado como um pixel em cada janela, exibindo seu atributo correspondente (BRANCO, 2003).

Conforme Freitas *et al.* (2001), a consideração inicial para a escolha de uma técnica de visualização é a caracterização dos dados. Essa etapa é composta por três critérios, o primeiro trata sobre a classe de informação a qual um atributo pertence, de forma a possibilitar o enquadramento da entidade em uma categoria ou grupo. O segundo, refere-se ao tipo de dado que um atributo possui, seja numérico, alfanumérico ou simbólico. O terceiro critério diz respeito à dimensão e natureza de onde os atributos são definidos, seja pela dimensionalidade, no caso de domínio ou pela classificação de dados quantitativos em discretos ou contínuos, no caso de natureza.

Além da importância da escolha da técnica de visualização, baseada na classificação ou categorização do conjunto de dados, a escolha das cores utilizadas nas representações de dados é importante, conforme destaca Branco (2003). Tutida (1998 *apud* BRANCO, 2003) afirma que

a seleção de cores adequadas é um aspecto crítico em visualização. Branco (2003) ainda cita um exemplo no qual as cores que compõem a representação visual são capazes de determinar o sucesso dos resultados de uma técnica.

Uma forma muito comum de visualizar dados é por meio de gráficos baseados em tempo. Alencar (2007) define Séries Temporais como um conjunto de observações sequenciadas no tempo. Ehlers (2009) ainda ressalta que uma característica relevante nesse tipo de representação é a dependência entre os dados que estão adjacentes, uma vez que a ordem dos mesmos é crucial para uma série temporal.

Neste contexto, um exemplo de Séries Temporais com um grande volume de dados, são os segmentos de mídia, como músicas. Ono (2015) afirma que a representação visual de músicas visa facilitar a interpretação, exibindo tons, notas, acordes, harmonia e outras propriedades. Entendendo que a extração de características é a base para qualquer tipo de análise de áudio, Tzanetakis *et al.* (2002 *apud* ONO, 2015) explicam que o sinal sonoro pode ser expresso através de um cálculo de representação numérica. Percebe-se, então, que com a representação numérica de um conjunto temporal de sinais, é possível gerar uma visualização do resultado.

A forma analógica do sinal de áudio pode ser definida como uma função real sobre uma variável que representa o tempo. Porém, em forma digital ele é definido como uma função que mapeia apenas um intervalo de tempo relacionado à amostragem utilizada. O processamento de um conjunto de sinais é feito sequencialmente, e cada valor dessa sequência recebe o nome de amostra. Para tal, existe o ciclo de processamento do sinal digital, ciclo DSP (*Digital Signal Processing*), que consiste na obtenção e processamento de amostras com o intuito de que para cada valor de entrada, exista um valor de saída (BIANCHI, 2014).

A representação visual de um segmento de áudio pode ser realizada em forma de gráfico, tendo tempo e frequência especificados em seus eixos, e utilizando cores para a magnitude das frequências. Essa representação é conhecida como espectrograma (ONO, 2015). Chen *et al.* (2010) ainda explicam que a simples exibição de onda no domínio do tempo não demonstra muito sobre o seu conteúdo, em vista que a distribuição de frequência é o principal portador de informação dos sinais para análise musical.

Conforme Paulus, Müller e Klapuri (2010), a diferença entre sons aleatórios e música não é clara, pois o que pode ser uma sequência de ruídos para alguns, pode ser uma engenhosa composição musical para outros. Por outro lado, os autores entendem em comum acordo que eventos sonoros isolados não são capazes de representar um significado musical. Para isso, é necessária uma estrutura, ou então um relacionamento entre vários eventos de som.

Smith e Williams (1997) reconhecem que a forma mais popular de visualizar música, é através da notação musical. Por meio dela, músicos experientes são capazes de conhecer o resultado sonoro sem de fato ouvir uma melodia. Um formato digital que se aproxima dessa notação, no sentido de conter cada símbolo e seu respectivo tempo de ocorrência, é o formato MIDI. Embora seja um dos formatos mais populares, o MIDI (*Musical Instrument Digital Interface*) não é utilizado para a representação de performances, tal como uma gravação sonora em específico, mas sim para o mapeamento da informação existente naquele segmento definido.

Para a representação visual de arquivos MIDI, *softwares* como o MIDITrail e o *Music Annotation Machine* podem ser utilizados (ONO, 2015). Os símbolos ou notas são exibidos em uma plotagem baseada em tempo. Dado que o formato MIDI também possibilita a separação de instrumentos por canais, as ferramentas de visualização podem, então, representar visualmente informações sobre as notas contidas no arquivo em uma disposição semelhante a uma partitura, dado que utiliza um domínio discreto com informação booleana.

Em outro sentido, aplicativos como o *Sonic Visualiser* (CANNAM; LANDONE; SANDLER, 2010) buscam exibir as informações de sinais de áudio. Nele, formatos digitais, geralmente formados pela digitalização de conteúdo analógico, são exibidos por meio de técnicas de visualização em representações de espectrograma, formas de ondas e picos de frequências, e a possibilidade de utilização de *plugins* pode acrescentar mais métodos de análise aos já existentes.

Com o objetivo de fazer uma exibição durante a execução de um segmento de áudio, o MAM, *The Music Animation Machine* (MALINOWSKI; TURETSKY, 2016), traz uma evolução histórica nas abordagens deste assunto. Iniciando em 1974, quando é relatada uma alucinação em que notas dançavam em uma página enquanto ouvia-se música, passando pela disponibilização de vídeos de músicas com suas respectivas animações gráficas no *YouTube* e recentemente um aplicativo para *iPad*. Além deste, algumas ferramentas, inclusive *players* comerciais possuem opções para visualização momentânea do que está sendo reproduzido sonoramente.

Atuando na representação de áudio em forma de objetos reais, existem projetos como os do "*Sculpteur de Voix*" (escultor de voz), Gilles Azzaro (AZZARO, 2016), que reproduzem discursos através de impressões em três dimensões, e o projeto *Frozen Waves* (VAN MELLE; MARKS, 2014), que faz uma ligação artística entre som e objeto físico.

Com base neste contexto, o objetivo geral deste trabalho é **investigar um modelo para a representação visual de áudio baseada na interpretação de sinais e técnicas de**

visualização, possibilitando assim, a apresentação dos dados de áudio de forma gráfica mais agradável e rica em informação ao usuário final.

Diante do objetivo geral, os objetivos específicos do presente estudo são:

- **Estudar os principais formatos digitais de áudio, considerando parâmetros que possam ser diferenciais na representação visual de sua análise.**
- **Apresentar uma abordagem de representação visual que utilize os conceitos estudados.**
- **Desenvolver uma representação visual através da análise de áudio.**
- **Validar os resultados obtidos durante os experimentos.**

Além dos objetivos que definem a meta do trabalho, a metodologia é baseada nos conceitos apresentados por Prodanov e Freitas (2013). Logo, o trabalho caracteriza-se, quanto a sua natureza, como pesquisa aplicada, visto que seu produto visa solucionar um problema específico. Devido ao método avaliativo utilizado sobre a conformidade da solução com o problema proposto, este estudo é qualificado como pesquisa qualitativa.

Com base na investigação, levantamento bibliográfico e análise de exemplos e experiências práticas para a formulação da solução, quanto aos seus objetivos pode-se caracterizar o trabalho como pesquisa exploratória. O estudo também caracteriza-se como pesquisa bibliográfica e experimental no que se refere a procedimentos técnicos, pois aborda o estudo das técnicas e conceitos que serão utilizados como referencial teórico e utiliza de experimentação para desenvolvimento e observação dos resultados.

Assim, o volume é estruturado em seis capítulos. O primeiro, intitulado “Visualização de Dados”, aborda conceitos e técnicas relacionadas a visualização de dados no sentido técnico, relacionando tipos de dados com tipos de visualização e entendendo como ocorre a estruturação de um processo de visualização. O conceito de *pipeline* no contexto da visualização é abordado, e após perpassar pela pesquisa relacionada a arte, uma abordagem de visualização que combina geometria e estética é apresentada.

Com o título “Análise de Áudio”, o segundo capítulo apresenta de forma geral a conversão de áudio do formato analógico para o digital, embasando uma sequência que destaca as propriedades presentes no som. Descreve ainda as principais propriedades que podem ser extraídas do áudio em formato digital e exemplifica como uma análise espectral é realizada.

O terceiro capítulo intitulado “Proposta de Desenvolvimento”, utiliza os conceitos abordados de visualização de dados e análise de áudio para apresentar um *pipeline* e uma

proposta para desenvolvimento de um modelo de visualização de áudio como um esquema para implementação, baseado em referências estudadas.

O quarto capítulo, “Desenvolvimento do protótipo”, descreve o desenvolvimento do modelo com base na proposta existente, com a definição de um escopo composto por requisitos, justificativa sobre escolha de ferramentas e relato de como se deu a implementação.

A avaliação do modelo é realizada no quinto capítulo, “Avaliação”, apresentando as análises e resultados obtidos a partir dos resultados da pesquisa feita com base no modelo gerado, com objetivo de entender a assertividade do mesmo, assim como fatores que a influenciam.

O sexto e último capítulo aborda as considerações finais sobre o estudo, apresentando resultados e aprendizado, além de indicar etapas e sugestões que podem ser utilizadas em estudos futuros.

1 VISUALIZAÇÃO DE DADOS

Com o aumento constante da automação dos processos nas mais diversas áreas do conhecimento humano, verifica-se a geração de um volume muito grande de informação (BOTELHO, 2002). Além disso, as melhorias tecnológicas presentes na área de comunicação e computação, possibilitam que os usuários tenham acesso a grande parte dessa informação abertamente, através de diversas bases de dados (FREITAS *et al.*, 2001). No ano de 2009 na conferência do TEDxSP, Fabiana Viégas, cientista brasileira que atua com visualização de dados, comenta que se vive um momento na história em que toda essa informação não está mais limitada aos especialistas, mas disponível aos usuários "comuns" e algumas ferramentas já possibilitam a democratização de tecnologias como a visualização de dados.

Wurman (1991) cunhou o termo “Ansiedade de Informação”, se referindo à lacuna existente entre os dados e o conhecimento. No ponto de vista da análise dos dados, sabe-se que eles descrevem eventos ou entidades, e as representações gráficas de conjuntos, por sua vez, visam facilitar a interpretação pelos usuários, porém sem omitir a quantidade de informação ali presente, por maior que seja (FREITAS *et al.*, 2001).

Apoiando essa ideia, Shneiderman (1996) utiliza o ditado popular de que uma imagem vale mais que mil palavras para expressar que algumas tarefas se tornam mais fáceis quando utilizando imagens ao invés de descrições textuais ou relatos falados. O autor sustenta que a exploração de informações de forma gráfica, apoiada pelas habilidades de percepção humana, torna eficiente a detecção de mudanças em cores, formas, tamanho, movimento e textura e ainda sumariza o processo de visualização como um ritual, conhecido como *Visual Information Seeking Mantra*, que consiste em: visão geral; *zoom* e filtro; detalhamento sob demanda.

Dessa forma, entende-se que a análise gráfica é capaz de facilitar a extração de conhecimento destes dados. Neste contexto, o presente capítulo busca averiguar as etapas e técnicas envolvidas na criação de representações visuais de dados. Assim, a primeira seção aborda as etapas utilizadas para gerar visualizações por meio da abstração conhecida como *pipeline*. Diante do modelo apresentado, a pesquisa em visualização de dados é analisada tanto pelo ponto de vista do usuário como dos tipos de dados a serem exibidos na segunda seção, que também levanta em suas subseções a visualização de áudio, assim como trabalhos relacionados. A terceira seção verifica classificações e guias da pesquisa na arte, que são utilizadas na atual proposta. Uma abordagem que associa geometria e estética é apresentada na quarta seção.

1.1 Etapas para gerar Representações Visuais de Dados

Na área da computação, a abstração mais utilizada em aplicações e bibliotecas de visualização é o que se conhece por *visualization pipeline* (MORELAND, 2013). Ela é representada por uma estrutura composta por módulos executáveis, que ligados entre si por junções direcionadas, apresentam o fluxo dos dados como uma rede, especificando as etapas que compõem o processamento que transforma dados em imagens. O autor ainda explica que este fluxo é composto por conexões e pela gestão da execução, além dos módulos citados anteriormente. Ele os classifica em três tipos: *source*, *filter* e *sink*. *Source* é o tipo que representa a entrada de dados de fora do processo, como por exemplo, a leitura de um arquivo. Estes componentes ainda contam com uma saída, que é utilizada para alimentar o restante do sistema. Definidos como *filter*, outros módulos do *pipeline* contam com entrada e saída de dados, sejam as entradas provenientes de um *source* ou até mesmo de outro *filter*, enquanto sua saída pode alimentar outro *filter* ou ser depositada em um *sink*. Já os *sinks* representam as etapas finais dentro do fluxo de dados do *pipeline*, e devem ser implementados como saídas de dados.

As conexões são as ligações direcionais entre módulos que representam o fluxo de dados do *pipeline*. É possível entender então, que em um contexto de grafos, a rede definida como *pipeline* possui como nodos os módulos, e como arestas as conexões. Já que a simples união de módulos por conexões serve apenas como uma definição do fluxo de dados, isso ainda não é suficiente para o funcionamento do *pipeline*. Para essa demanda, a gestão de execução é responsável por determinar quando e como os módulos devem ser executados (MORELAND, 2013).

Em termos de desenvolvimento, essa abstração é o equivalente a uma rotina declarada em alto nível, e o formato de *pipeline* traz como intenção facilitar a composição de módulos intercambiáveis, que baseados em entradas e saídas em formato genérico possibilitem mudanças na ordem do fluxo de dados sem necessidade de ajuste nas implementações. A conexão entre dois módulos pode ocorrer desde que o formato de saída de um seja compatível com o de entrada de outro (MORELAND, 2013).

Com uma abordagem mais interativa, Holliman (2015) explica a existência de estágios em um cenário de *pipeline*, e cada um destes, por sua vez, refina os dados de forma a deixa-los cada vez mais próximos da visualização final, além de poder aceitar eventos de interação para atuar com comportamento diferente. Os estágios básicos elencados pelo autor são a seleção e análise, mapeamento, renderização e interface com o usuário. Os estágios podem ser definidos

conforme a tarefa que estiver sendo executada pelo usuário, juntamente com o processo durante a visualização, conforme analisa-se a seguir.

1.2 A pesquisa para visualização de informação

Com objetivo de melhorar o entendimento de toda riqueza presente nos variados conjuntos de visualizações de informação que já foram propostos, Shneiderman (1996) apresenta sete tipos de dados que refletem abstrações da realidade e, sete atividades abstratas que representam as ações relacionadas a informação que usuários pretendem executar durante a análise de visualizações. Os tipos de dados, definidos como *Type by Task Taxonomy* (TTT, Tipo por Taxonomia de Tarefa) pelo autor, podem ser utilizados diretamente ou por meio de combinações entre si, e essa taxonomia é útil desde que facilite discussões e conduza para descobertas. Já as atividades, são pontuadas como uma definição aperfeiçoada do Mantra (*Visual Information Seeking Mantra*) apresentado no início do presente capítulo, e podem ser refinadas conforme necessidade.

Como primeiro membro da TTT, existe o tipo de dado unidimensional. Esse tipo de dado caracteriza informações lineares, como textos, código-fonte de programas e listas. Como questões relacionadas, para criação de visualizações, considera-se a seleção de fontes, cores, tamanhos, informações que deverão ser apresentadas na visão geral e como a navegação e seleção poderão ser disponibilizadas. Para os usuários também existem algumas questões, entre as quais atenta-se para o número de itens presentes, filtrar o resultado para visualizar apenas o que for relevante a uma certa condição, ou ver um item com todas as suas propriedades (SHNEIDERMAN, 1996).

Mapas, plantas-baixas e até mesmo a diagramação de jornais são dados bidimensionais. Nas coleções dispostas neste tipo de representação, cada item representa uma parte da área total, e geralmente possui informações referentes ao domínio da tarefa relacionada (nome, valor, etc.), assim como propriedades do domínio da *interface* (tamanho, cor, opacidade, etc.). Em cenários como mapas, alguns sistemas utilizam uma abordagem multicamada, em que cada camada é representada de forma bidimensional. Por sua vez, os usuários buscam por localização de itens, caminhos entre eles, e interagem com contagem, filtragem e detalhamento (SHNEIDERMAN, 1996).

Objetos do mundo real podem ser observados por meio de representações em três dimensões e algumas tarefas exigem o entendimento de adjacências entre peças. Alguns contextos de visualização buscam explorar este tipo de relacionamento, e o usuário precisa estar

ciente da posição e orientação durante a visualização. As soluções neste contexto utilizam técnicas como visão geral, ponto de referência, perspectiva, visualização estéreo, transparência e codificação de cores (SHNEIDERMAN, 1996).

Para acompanhamento de eventos através do tempo, existe a representação de dados temporais. Utilizada em relatórios médicos, gerência de projetos e apresentações históricas, distingue-se do tipo de dados unidimensional devido ao fato de que seus itens podem possuir momentos de início e fim, o que pode causar sobreposição em alguns cenários (SHNEIDERMAN, 1996).

Dados multidimensionais também estão presentes na taxonomia definida por Shneiderman (1996). A representação geralmente é efetuada em gráficos de dispersão de duas dimensões, em que as outras dimensões são geralmente controladas pelo usuário por barras deslizantes. Também é possível representá-los por meio de dispersão em três dimensões, entretanto pode-se sofrer com desorientação por parte do ponto de vista do usuário e até mesmo obstrução dos dados.

Estruturas hierárquicas são coleções de dados nas quais os itens possuem um vínculo com seu superior. Quando organizadas na forma de árvores, é possível representar os itens com ligações entre si, formando um desenho estruturado, na maioria das vezes de forma cônica, onde cada nó pode conter informações mais detalhadas (SHNEIDERMAN, 1996).

Quando em um conjunto de itens, os relacionamentos não seguem uma estrutura simples como em árvore, é conveniente utilizar uma visualização de rede. Além das informações referentes a cada componente do grupo, os caminhos traçados para conectar dois itens, ou para atravessar a rede inteira também são relevantes (SHNEIDERMAN, 1996).

Considerando o objetivo do atual trabalho de representar áudio, é possível combinar tipos de dados temporais com dados multidimensionais segundo a taxonomia estudada.

1.2.1 Visualização de áudio

A forma mais popular de visualizar música é por intermédio da notação musical (SMITH; WILLIAMS, 1997). Com ela, músicos experientes são capazes de conhecer o resultado sonoro sem de fato ouvir uma melodia.

Chan (2007) diferencia a visualização de música e visualização estrutural de música afirmando que a primeira se baseia em intensidade e frequência do espectro da música em tempo real, como é possível verificar em aparelhos de rádio ou sistemas de mídia, enquanto a estrutural

não depende de animação. Tendo qualquer música como um segmento de áudio, é possível utilizar a mesma classificação dada pelo autor nas visualizações de segmentos de onda de som.

Embora Chan (2007) utilize a diferenciação entre visualizações, neste trabalho utiliza-se os termos representação e visualização, estrutural ou não, tanto para áudio como para música para a representação de um segmento temporal, e não a exibição em tempo real do áudio sendo tocado.

1.2.2 Trabalhos relacionados

Embora existam inúmeros trabalhos relacionados a visualização de dados, inicialmente apresenta-se investigações que buscam realizar a representação visual de áudio. Neste sentido, buscando explorar improvisações em músicas de Jazz, o *Improviz* (SNYDAL; HEARST, 2005) propõe duas visualizações: panoramas melódicos para mostrar os contornos de um fraseado e paletas harmônicas, para representar a tendência de utilização de uma combinação de notas em uma parte da música. Um diferencial deste trabalho é que as representações foram realizadas a partir de transcrições musicais, e não arquivos com áudio em formato digital.

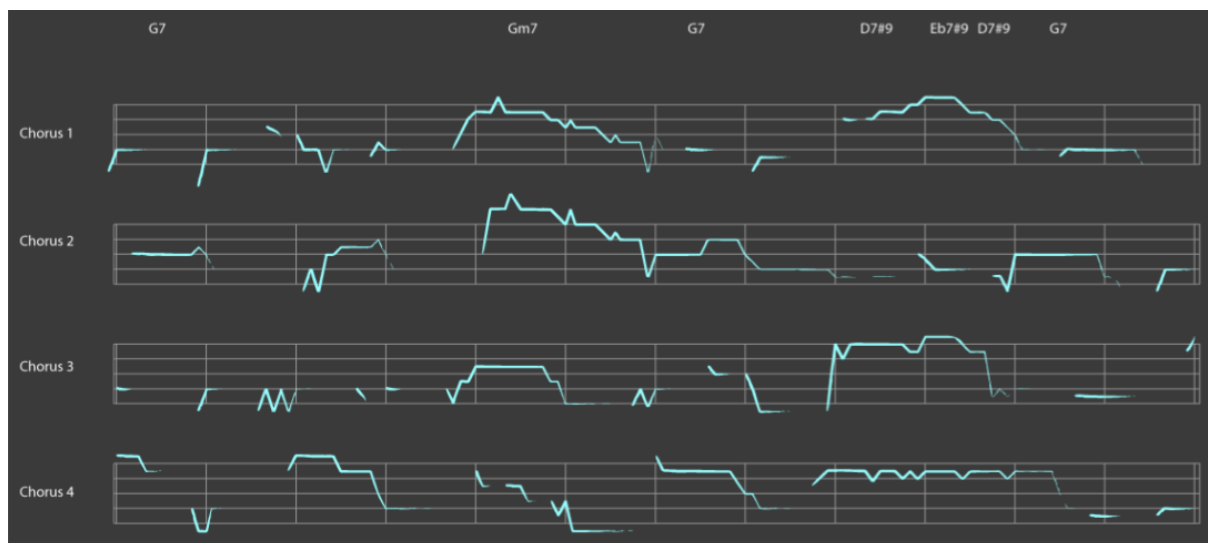


Figura 1.1 – Segmentos de Jazz apresentados pelo Improviz.

Fonte: Snyder e Hearst (2005).

Com uma abordagem não apenas visual, mas também física, Van Melle e Marks (2014) apresentam o projeto *Frozen Waves* como uma ideia de que objetos são processos em mudança contínua no tempo. Assim, por meio de interações contínuas de gravações de som, análise espectral de áudio, o projeto realiza a criação de um modelo e posteriormente sua impressão física em três dimensões, promovendo representações físicas do mundo acústico.

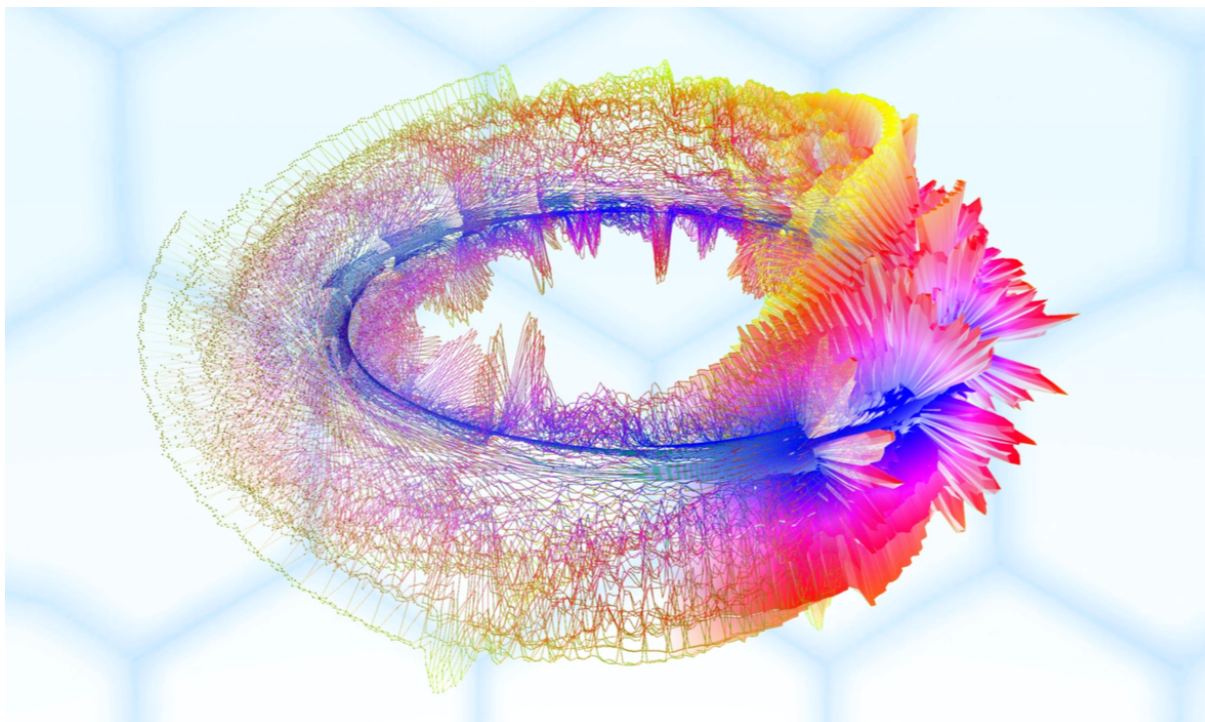


Figura 1.2 – *Frozen Waves* – Quatro segundos de áudio são transformados em um objeto em três dimensões.

Fonte: Van Melle e Marks (2014).

Os objetos criados podem então ser revertidos nos sons que os criaram, isto é, o som pode ser transformado em um objeto físico, e este objeto físico, por sua vez, pode ser convertido no mesmo som que o gerou. As criações também podem ser consideradas protótipos de vestuário ou decoração (VAN MELLE; MARKS, 2014).

Ainda no contexto de representações físicas de áudio, é notável o trabalho de Azzaro (2016). Autointitulado como "*Sculpteur de Voix*" (escultor de voz), o artista reproduz discursos através de impressões em três dimensões. Enquanto um trabalho parecido é empregado por Fischer (2010), em que uma faixa musical é composta em um panorama.

Embora as pesquisas anteriores investiguem a representação de áudio, no contexto deste trabalho é importante destacar a pesquisa apresentada por Viégas e Wattenberg (2016) que desenvolve uma visualização conhecida como *Flickr Flow*, em que a exibição de dados de um período é representada em forma circular, com distribuição temporal em sentido horário. Utilizando um algoritmo que foi criado para a *WIRED Magazine*, os autores trabalharam com a extração de cores de fotos nos períodos do ano para formar a representação.

1.3 A Pesquisa na Arte

A organização das informações através de elementos visuais que representam conteúdos sobre dados de diferentes naturezas aproxima-se da arte. Um exemplo é a representação muito

utilizada no jornalismo, atualmente, que se intitula de infográfico (GIANNELLA, 2014). Ela consiste na apresentação de informações por meio de elementos visuais integrados com informações textuais sintéticas, unindo os sistemas de comunicação verbal e visual (BOTTENTUIT JUNIOR; LISBÔA; COUTINHO, 2011). Este modelo possui o objetivo de ilustrar as informações da melhor maneira possível, utilizando técnicas de Design de Informação, que segundo Horn (2000) é a arte ou ciência empregada no aumento da eficiência e eficácia da utilização da informação. Dessa forma, o autor do presente trabalho aborda a pesquisa na arte e no design dada a importância do viés artístico em elementos visuais.

Segundo Frayling (1993), a relação entre arte e pesquisa pode ser classificada nas categorias: 1- Pesquisa em arte (*Research into art and design*); 2- Pesquisa através da arte (*Research through art and design*); 3- Pesquisa para arte (*Research for art and design*).

A mais comum delas, pesquisa em arte, é geralmente composta por estudo histórico, pesquisa em estética ou percepção e investigação de outras perspectivas sobre a arte, como social, econômica, política, ética e cultural. É considerada a mais direta, pois seus estudos buscam derivar a arte em regras e procedimentos (FRAYLING, 1993).

Já a pesquisa através da arte, consiste no que é alcançado e comunicado por meio das atividades de arte e criação. Isto é, um trabalho executado na área artística envolve o desenvolvimento de um relato substancial com nível de pesquisa enquanto é executado (FRAYLING, 1993).

Frayling (1993) ainda explica que a pesquisa para arte é aquela que gera subsídios para a arte, como, por exemplo, a busca de materiais de referência. O objetivo é gerar um artefato de conhecimento de sentido representativo. O autor ainda diferencia os indivíduos da área, entre aqueles que produzem conhecimento e aqueles que produzem arte, com argumento de que não se oferece pesquisa quando arte "fala por si", dado que a meta é produzir arte ao invés de conhecimento e entendimento.

1.4 Geometria e Estética

Ao tratar de arte, é comum relacioná-la à estética. Com base nesta afirmação, o trabalho de Byron e Wattenberg (2008) aborda decisões de *design* e algoritmos que estão por trás de gráficos que foram utilizados no jornal New York Times e foram efetivos em mostrar um grande volume de dados para uma enorme audiência. Estas visualizações, chamadas pelos autores de *Stacked Graphs* são por eles utilizadas com base em dados históricos, em que o

tempo possui enorme significância. Portanto, os métodos de visualização baseados em séries temporais são facilmente aplicados ao contexto das técnicas estudadas por eles.

O primeiro deles, é uma visualização em que Byron desenvolveu para identificar tendências pessoais em músicas (BYRON; WATTENBERG, 2008). A partir de dados coletados no Last.fm, uma espécie de rede social que atua com um serviço capaz de coletar o histórico de músicas de seus usuários. O autor apresentou os históricos em um gráfico de barras, exibindo a quantidade de vezes que cada artista teve suas músicas ouvidas, agrupado por semana. No contexto de seu trabalho, um problema proposto foi buscar a melhor forma de exibir os dados, pois cada usuário teria o seu gráfico pessoal, e uma forma convencional estudada por Byron foi o uso de um gráfico de pilha por camadas, em que cada camada representasse o histórico de um artista.

No entanto, seria problemático para Byron apresentar uma abordagem que prejudicasse a legibilidade de camadas individuais em casos onde os usuários tivessem um grande número de artistas a ser exibido. O objetivo de seu trabalho não era apenas estatístico, mas também emocional, para encarnar visualmente a conexão que ouvintes possuem com a música que ouvem (BYRON; WATTENBERG, 2008).

Neste contexto foi criada uma nova forma de gráficos de pilha, o *Streamgraph*, que organizando as camadas de uma forma mais orgânica, enfatiza a legibilidade das mesmas. Como parte de um projeto acadêmico, este *design* recebeu atenção tanto de entusiastas de visualização como de amantes de música. Este destaque fez com que o trabalho também recebesse prestígio do *The New York Times*, que publicou tanto em versão impressa, como também disponibilizou online uma visualização interativa, onde 7500 filmes foram dispostos em um período de 21 anos apresentando suas respectivas receitas (BYRON; WATTENBERG, 2008).

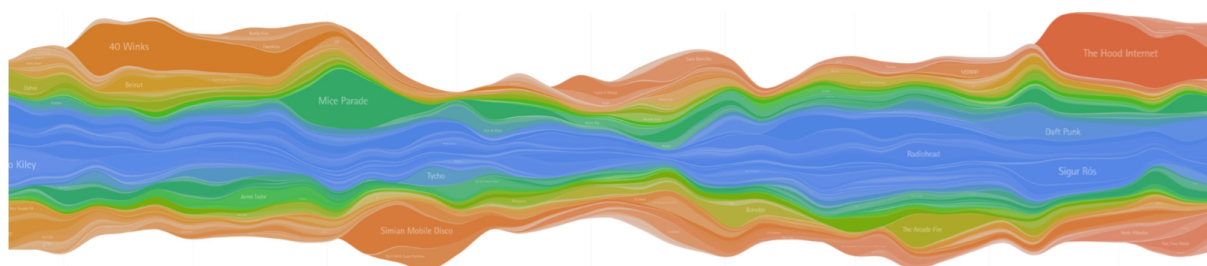


Figura 1.3 – Stacked Graphs

Fonte: Byron e Wattenberg (2008).

A apresentação de dados, independente de particularidades, pode ser orientada pelos princípios da excelência gráfica (TUFTE, 1986). Eles consistem na comunicação de ideias complexas com clareza, precisão e eficiência, de forma que deem ao usuário (visualizador) o

maior número de ideias, no menor tempo possível, com a menor utilização de pintura no menor espaço. Sustentando estes princípios, o autor também apresenta as micro e macro leituras, nas quais propõe que a uma determinada distância seja possível entender o conjunto, assim como tendências, enquanto de forma aproximada seja possível examinar individualmente cada elemento. Estes objetivos também devem ser alcançados ao mostrar várias séries temporais individuais no contexto de gráficos empilhados, pois deve ser possível identificar componentes isoladamente, ao mesmo tempo que a soma de vários componentes também esteja evidente.

2 ANÁLISE DE ÁUDIO

Como a fonte de dados para as representações visuais do presente trabalho são sons, é necessário compreender quais são suas propriedades e quais características podem ser extraídas destas. Neste contexto, fisicamente, sons são os resultados de oscilações ou variações de pressão em um meio elástico (HANSEN, 2001). Por exemplo, quando estes eventos são propagados no ar, essas alterações ocorrem em torno da pressão atmosférica. Inclusive, do ponto de vista acústico, a diferenciação entre som e ruído é subjetiva ao ouvinte, dado que ambos são causados pelo mesmo fenômeno (HANSEN, 2001).

Estes eventos por si, não estão diretamente preparados para que sejam trabalhados e analisados como dados. Para isso, Zolzer (2008) explica como funciona o processo de gravação, onde microfones captam o sinal analógico, que após convertido é armazenado em formato digital.

2.1 Conversão de Áudio

Este processo é conhecido como conversão *AD* (Analógico Digital), dada a transformação do áudio do meio físico para o digital. A gravação do sinal digital pode ocorrer em diversos canais, e é distribuída em um conjunto de sistemas. Estes por sua vez seguem uma taxa de **amostragem** que é estabelecida e controlada por um sistema central de tempo (ZOLZER, 2008). Se tratando de tecnologia digital de áudio, as taxas geralmente utilizadas são: 48 *kHz* para estúdios profissionais, 44.1 *kHz* para discos e 32 *kHz* para transmissões. Conforme Tzanetakis (2011), a taxa de amostragem representa a quantidade de intervalos, por segundo, em que um segmento de áudio é dividido em amostras. Assim, por exemplo, um áudio gravado a 48 *kHz*, possui 48 mil amostras por segundo.

Além da amostragem que ocorre na taxa explicada acima, a conversão de sinal analógico para digital possui outra operação básica, conhecida como **quantização** (ZOLZER, 2008). Esta operação consiste na digitalização de um segmento de sinal analógico pela atribuição de um valor a ele. Por este processo, entende-se que baseado na taxa de amostragem, o sinal em um espaço no tempo terá sua amplitude extraída em uma forma discreta, pré-estabelecida dentro de um conjunto finito de níveis. Para finalizar a digitalização de um sinal analógico, Fernandes e Panazio (2009) indicam como última etapa a **codificação**, onde os valores extraídos no passo anterior são designados conforme o domínio de aplicação.

Dessa forma, as operações de amostragem, quantização e codificação são abordadas a seguir:

Amostragem: Como visto anteriormente, existem diversas taxas de amostragem já consolidadas no mercado, e é interessante ressaltar como elas foram alcançadas de forma não empírica. Antes mesmo da utilização do termo amostragem, Nyquist (1928) elencou diversos tópicos na transmissão por telégrafos, entre eles a proporcionalidade entre a banda de frequência e velocidade de transmissão de sinal.

Com base nesta relação e estudos matemáticos, Shannon (1949) evidencia o **Teorema da Amostragem** na área de comunicação ao afirmar que se uma função $f(t)$, que descreve um sinal contínuo, possuir frequências dentro de uma banda W em ciclos por segundo, de zero até seu máximo, ela pode ser definida por uma série de pontos, desde que eles estejam espaçados $1/2W$ segundos entre si. Essa afirmação é baseada na justificativa de que entre dois pontos temporalmente adjacentes, não pode haver uma mudança substancial em um período de tempo menor do que metade do ciclo da maior frequência possível. Dessa forma, para uma função que descreve frequências de máximo valor W , em um segmento de T segundos, entende-se que um conjunto de $2TW$ valores é suficiente para descrevê-la de forma que o dado amostrado possa ser recuperado sem perdas significativas.

Quantização: Após conhecer a fatia de tempo que cada amostra representa, a operação de quantização é a responsável pela atribuição de valores discretos a cada intervalo (BLACK; EDSON, 1947). Esta atribuição corresponde geralmente a um número, que pode ser definido pela densidade ou até mesmo pela média daquele intervalo.

O desafio nestas operações está em conjunto com a amostragem, em determinar o menor número de intervalos de tempo a passarem pelo processo de quantização, e a precisão com a qual a quantização será feita sem causar distorção ou granularidade aparentes. Segundo o estudo de Black e Edson (1947) sobre PCM (Modulação de Códigos de Pulsos), a quantização de ondas de voz deve ter tamanho de 5 a 7 bits por amostra, o que garante de 32 a 128 níveis.

Codificação: Segundo Zolzer (2008), os sinais que já foram segmentados no processo de amostragem e discretizados na quantização podem então passar pelo método de codificação, baseados em diversas formas de compressão que a área de PCM investigou. O autor apresenta os dois tipos aplicados a áudio, *lossy* e *lossless*, com e sem perdas, respectivamente.

O primeiro, *lossy*, é baseado em um modelo da percepção humana para efetuar uma quantização diferenciada para comprimir o sinal, e suas aplicações são variadas, como otimização para transmissão ou armazenamento. Enquanto o segundo, *lossless*, utiliza métodos

estatísticos que embora não tenham a mesma performance de compressão do método anterior, não implicam em perdas no sinal (ZOLZER, 2008).

Então, dessa forma, é possível compreender como a **amostragem**, a **quantização** e a **codificação** se aplicam no formato digital: O ser humano é capaz de perceber frequências de até 20 *kHz*, portanto baseado no Teorema da Amostragem, entende-se que aproximadamente 40 *kHz* sejam suficientes para uma taxa de amostragem sem perdas, e próximo disso temos a frequência utilizada em discos, que é de 44.1 *kHz*. Após esta definição, a quantificação busca atribuir valores discretos para todos estes intervalos e posteriormente a codificação visa otimizar tal atribuição. Concluídas essas etapas, o segmento de áudio em formato digital já é capaz de fornecer dados suficientes para extração de informações de mais alto nível.

2.2 Propriedades e Características do Som

Neste sentido, o estudo de Orio (2006) indica que as três propriedades básicas de um som musical são a frequência, que define o quão grave ou agudo ele é; a intensidade, sendo sua amplitude ou potência; e o timbre, que torna possível a diferenciação entre dois sons de mesma frequência e amplitude, como por exemplo vozes ou instrumentos. Estas propriedades estão presentes não apenas em músicas, conforme apresentado pelo autor, mas em qualquer segmento de som.

Uma base para análise de áudio parte da extração de suas características. Este processo consiste na geração de representações mais compactas, contendo informações mais específicas sobre o seu conteúdo (TZANETAKIS, 2011). Neste sentido, na área de tecnologia em áudio, focado no contexto musical, a área de Recuperação de Informação de Música (MIR, *Music Information Retrieval*) tem evoluído para a descoberta de informações por meio de diversas técnicas e ferramentas (ORIO, 2006). Dessa forma, como o atual trabalho busca estudar a extração de características de áudio em geral, e algumas informações de cunho exclusivo musical não estão presentes em qualquer segmento de áudio, algumas ferramentas e técnicas de MIR baseado em conteúdo são estudadas para análise de som.

No contexto de visualização de áudio, uma etapa muito importante é a seleção de quais características deverão ser apresentadas. Em seu estudo sobre visualização de séries temporais, Alencar (2007) detalha etapas relacionadas ao processo de mineração de dados. Entre elas, o pré-processamento contém operações que podem ser aplicadas em cenários com este tipo de dados, que são a suavização, normalização e remoção da sazonalidade. Assim, esta seção

aborda as características que podem ser extraídas de segmentos de áudio e operações relevantes a cada uma.

Outras informações como notas, que são específicas de músicas e podem não ser encontradas em qualquer segmento de áudio não serão utilizadas neste trabalho, dado que se pretende trabalhar com qualquer segmento de áudio.

Como exemplo desta distinção de características, é possível observar no projeto *jMIR*, como McKay (2010) utilizou diferentes componentes para extração de características. Entre eles estão o *jAudio*, para extração de informações diretamente de áudio, *jSymbolic* para análise de arquivos MIDI, o *jWebMiner*, que captura dados culturais através da internet e o *jLyrics* que busca letras de músicas na web. Seguindo o objetivo do atual estudo, apenas componentes de análise como o *jAudio* serão abordados, tendo em vista sua relação com som independente de contexto musical.

Analisando os recursos de extração de informações presentes no *jAudio*, é possível identificar algumas características como as apresentadas a seguir, além de diversas variações que podem ser derivadas das abordadas.

1. *Zero Crossing*
2. *Fraction of Low Amplitude Frames*
3. *Spectral Flux*
4. *Spectral Rolloff*
5. *Compactness*
6. *Method of Moments*
7. *2D Method of Moments*
8. MFCC
9. *Beat Histogram*

Além destas características, existem diversas outras, exploradas por demais ferramentas, dentre as quais é relevante citar a biblioteca *librosa* (MCFEE *et al.*, 2015), pois seus recursos para este tipo de processamento estão divididos em dois grupos: informações espectrais e rítmicas. No entanto, como as análises rítmicas estão diretamente ligadas à música, elas não serão consideradas no presente momento. Este tipo de diferenciação apoia e reforça a escolha das características.

2.3 Extração de Características

Embora diversas ferramentas já implementem e forneçam o trabalho de interpretar o áudio digital e processá-lo para extrair as características de alto nível, é necessário conhecer como este processo ocorre. Para isso, a presente seção investiga a transformação do formato PCM em espectrograma. O *Pulse Code Modulation* (PCM) consiste em um formato não comprimido que representa o formato de onda do som através de uma sequência de números. Cada número neste formato retrata a intensidade da onda em um período de tempo, que é definido conforme a taxa de amostragem utilizada. Já o espectrograma, é uma representação bastante utilizada, que em duas dimensões retrata tempo e frequência em seus eixos, enquanto utiliza cores para a energia de cada frequência naquele determinado momento (ORIO, 2006).

Segundo Osgood (2007), todo sinal possui um espectro e pode ser analisado no domínio de tempo ou de frequência. Com o intuito de representar e analisar fenômenos periódicos, existem as Séries de Fourier. Baseadas no período do sinal, as séries apresentam uma forma para extrair um conjunto de senoides de frequências diferentes, cuja soma equivale ao sinal original. Estas senoides são os elementos que montam o espectro, representando cada uma sua respectiva frequência, assim como sua energia.

No contexto do presente trabalho, onde a análise parte de áudio, sabe-se que as ondas sonoras em sua grande maioria não são periódicas, portanto as Séries de Fourier não são capazes de extrair as frequências conforme necessário. Neste sentido, considerando as Análises de Fourier como um conjunto de técnicas, tem-se uma evolução das Séries, que estende sua visão para fenômenos não periódicos. Esta análise é conhecida como Transformadas de Fourier e pode ser considerada como uma transição a partir das Séries, pois gera um espectro contínuo ao apresentar as frequências, enquanto as Séries de Fourier geram um conjunto discreto de frequências. Computacionalmente, é preferível processar valores discretos do que sinais ou espectros contínuos, portanto no atual cenário onde o sinal de áudio já é disponibilizado em formato discreto seguindo uma taxa de amostragem, uma variação das Transformadas de Fourier adaptada para o processamento computacional, conhecida como Transformadas Discretas de Fourier (DFT - *Discrete Fourier Transform*) (OSGOOD, 2007), demonstra-se ser mais adequada.

Mesmo com toda velocidade computacional existente, a aplicação direta da DFT não é encontrada com muita frequência em aplicações devido ao tempo exorbitante necessário para executá-la. Porém, a análise científica teve uma revolução com o desenvolvimento das Transformadas Rápidas de Fourier (FFT - *Fast Fourier Transform*). A FFT apresenta os

resultados da DFT, porém seguindo uma outra proposta de algoritmo, isto é, um método diferente para efetuar as computações. Esta escolha faz com que a FFT tenha destaque em velocidade, se comparada a outros algoritmos para cálculo da DFT (BERGLAND, 1969). Assim sendo, a FFT é encontrada com frequência em diversas ferramentas da área de DSP, facilitando e disseminando sua utilização.

2.4 Utilização de Características

Como base para entender as análises de sinal no ponto de vista de utilização e implementação, Tzanetakis (2011) expõe a utilização de sistemas classificados como *filterbanks*. Inclusive, é possível considerar a FFT como um destes, pois o autor atribui o termo a qualquer sistema que separe o sinal de entrada em diferentes faixas, cada um referente a um subconjunto das frequências presentes no escopo. Embora existam diversas formas de fazer a decomposição de sinal que geram representações que possuam subconjuntos para relacionar dados, todas elas podem ser consideradas *filterbanks*, independente de características e restrições, como impossibilidade de reverter operações ou sobreposição de valores.

Tzanetakis (2011) aponta uma das utilizações mais comuns de um *filterbank* em processamento de áudio como sendo a aproximação de como o sistema humano de audição processa sons. Esta aproximação é dada pela escala Mel de frequência, onde os subconjuntos de frequências são espaçados igualmente até 1 kHz, e após ele seguem um espaçamento sequencial logarítmico.

Este espaçamento é praticado pois o ser humano não percebe os sons na mesma escala física na qual os eventos sonoros ocorrem. Logo, este recurso é utilizado vastamente na área de reconhecimento de voz. Logan (2000) descreve como ocorre o processo de criação de MFCC (*Mel Frequency Cepstral Coefficients*) a partir do áudio em formato de onda. As etapas que compõem este processo são iniciadas pela separação do sinal em pequenos quadros, através de uma função janela que divide o sinal em intervalos de tempos iguais.

O processo continua com a aplicação das Transformadas de Fourier em cada segmento que foi separado, que retorna um conjunto de espectros de amplitude. Estes espectros são utilizados em uma nova etapa, responsável por calcular apenas seu logaritmo, dado que a sonoridade percebida de um sinal é aproximadamente logarítmica. Após isso, é realizada a suavização do espectro de um domínio igualmente intervalado para a escala Mel, com objetivo de enfatizar as frequências de forma mais significativa à percepção humana (LOGAN, 2000).

Essa suavização é alcançada pela redistribuição e agrupamento dos componentes espectrais no domínio de frequência em uma quantidade menor de componentes. Logan (2000) utiliza um exemplo onde 256 intervalos de frequência igualmente espaçados em Hz são convertidos para 40 intervalos espaçados na escala mel. Este ajuste não torna a coleção de dados apenas mais enxuta, mas também mais relevante para análise.

Considerando o contexto de cada quadro individualmente, os resultados obtidos até então estão altamente acoplados, portanto o último passo deste processo é a utilização do DCT (*Discrete Cosine Transform*), técnica que a comunidade de processamento de fala utiliza para descorrelacionar os componentes de cada quadro (LOGAN, 2000). Esta técnica também reduz o número de elementos presentes em cada quadro para aproximadamente 13.

Dessa forma, fica evidente que conforme o objetivo a ser alcançado é possível optar por *filterbanks* diferentes. Como o trabalho atual envolve a representação visual, e esta por sua vez tem como objetivo apresentar resultados para humanos, nada mais adequado que exibir dados de uma forma mais compreensível, baseado em uma relação entre a visualização e a percepção auditiva. Portanto, dentre as ferramentas disponíveis para análise espectral, é possível notar na MFCC a apresentação de dados em uma escala baseada na audição. O simples uso de FFT, embora contemplasse o mesmo conjunto de dados, não o faria com efeito tão adaptado ao estudo.

Assim, com base no estudo sobre extração de características de áudio presente neste capítulo, em conjunto com as técnicas de visualização presentes no capítulo 1, tem-se o necessário para embasar uma proposta de desenvolvimento, que é apresentada no próximo capítulo.

3 PROPOSTA DE DESENVOLVIMENTO

Objetiva-se o desenvolvimento de um sistema de visualização capaz de gerar representações estruturais de áudio que sejam visualmente semelhantes ao projeto *Flickr Flow* (VIÉGAS; WATTENBERG, 2016). Baseado nesta referência é possível identificar similaridades com a taxonomia definida por Shneiderman (1996) através de uma combinação de tipos de dados temporais e n-dimensionais, dependendo da quantidade de atributos a serem extraídos do segmento de áudio a ser representado.

Desse modo, consideramos semelhanças importantes entre o *Flick Flow* e a proposta deste trabalho, uma vez que ambos representam uma visualização radial em que está implícito o uso de séries temporais.



Figura 3.1 – *Flickr Flow*
Fonte: Viégas e Wattenberg (2016).

Uma etapa necessária para a construção do constructo pretendido é a análise de referências visuais e artísticas que se assemelham ao trabalho de Viégas e Wattenberg (2016). Assim, dado que antes da implementação, referências e guias para a criação são estudadas, esta

etapa do trabalho enquadra-se em pesquisa para a arte segundo a classificação de Frayling (1993), já descrita no capítulo 1 deste trabalho.

Seguindo na estruturação da proposta de desenvolvimento do sistema de visualização de áudio, procura-se apoiar a definição de um processo de pesquisa através de quatro questões genéricas, apresentadas por Gray e Malins (2004), que podem ser adaptadas para cada cenário: “O quê?”; “Por quê?”; “Como?”; “E então?”.

Com base nos questionamentos descritos anteriormente, o estudo é moldado com o seguinte conjunto de perguntas e respostas: **O quê?** Gerar representações visuais que tornem evidentes as características dos dados, fazendo com que o usuário reconheça um segmento de áudio ou compreenda seu estilo sem necessariamente ouvir. **Por quê?** Porque modelos conhecidos atualmente são muito técnicos ou muito artísticos. **Como?** Com o estudo de visualização de dados e base em alguns parâmetros da pesquisa para a arte, gerar um modelo que padronize a representação visual de ondas. **E então?** Avaliar o resultado do modelo de modo a verificar o quão assertiva e representativa a visualização é em relação ao segmento de áudio que a originou, também fornecendo a proposta como um esquema para outras implementações e estudos.

3.1 Definição de *pipeline*

Este trabalho utiliza a definição de *pipeline* e seus elementos, apresentados por Moreland (2013) e estudados no capítulo 1. Logo, a proposta de *pipeline* deste trabalho foi definida com componentes e fluxo de dados, conforme apresentada na Figura 3.1. Ela é composta por um componente do tipo *Source*, cinco do tipo *Filter*, e um do tipo *Sink*. O componente de entrada, denominado como ***audio source***, é responsável pela inserção do áudio a ser representado em formato digital dentro do *pipeline*. As transformações, são realizadas através dos componentes ***audio color extractor***, ***audio outlines***, ***audio shapes***, ***painter*** e ***arc mapping***, e, por fim a saída é realizada pelo componente ***render***. Assim, cada componente realiza uma determinada tarefa, conforme destacado:

- audio source***: o módulo de entrada do *pipeline* simboliza a leitura de um arquivo de áudio em formato digital. Sua saída no contexto do fluxo de dados são os dados do próprio arquivo;

- audio color extractor***: realiza a extração de um conjunto de cores a partir de um segmento de áudio;

- audio outlines**: efetua a criação de contornos que delimitem propriedades do áudio dispostas temporalmente;

- audio shapes**: propriedades do segmento são agrupadas e dispostas em formas vetoriais;

- painter**: com base nas cores fornecidas como entrada, preenche as formas e colore os contornos em uma matriz de cores;

- arc mapping**: reposiciona as informações temporais em um arco que representa a linha do tempo;

- render**: finalizando o processo, é responsável pela serialização final do trabalho.

A combinação dos componentes é sugerida tendo como ponto de partida o consumo do áudio pelo componente **audio source**. Este, por sua vez, alimenta os componentes **audio color extractor**, **audio outlines** e **audio shapes**. O resultado dessas três transformações gera subsídio para a etapa **painter**, que por sua vez, fornece um mapa de cores para **arc mapping**, responsável por reposicioná-lo em formato circular, que finalizando o processo depositará seu resultado no componente **render**, responsável pela serialização final do trabalho.

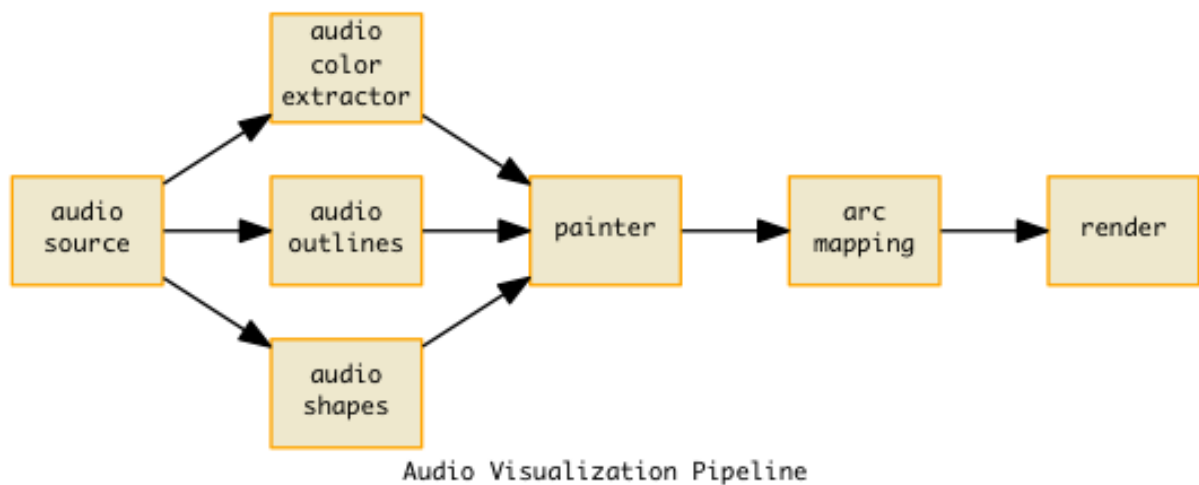


Figura 3.2 – *Pipeline* para Visualização de Áudio

Fonte: elaborado pelo próprio autor.

3.2 Proposta para Implementação

Com base na investigação realizada em visualização de dados e pesquisa para a arte, a proposta e o *pipeline* gerado são disponibilizados como um esquema para implementação, para que em conjunto com a análise de áudio sejam capazes de representar segmentos de áudio de forma visualmente agradável e informativa. Como expectativa, tem-se a capacidade de gerar resultados semelhantes ao projeto *Flickr Flow* (VIÉGAS; WATTENBERG, 2016).

Com este objetivo, o desenvolvimento de um protótipo pode seguir o *pipeline* como um esquema base, que segundo Moreland (2013) pode ser entendido como uma abstração. Se tratando de uma abstração, algoritmos podem ser encapsulados e utilizados em conjunto de diversas formas, independente de sua implementação. O autor ainda reforça que o *pipeline* é um fluxo lógico de dados, porém tanto os dados como o controle do funcionamento podem ocorrer de formas variadas através da rede, e este desvio pode ser considerado um detalhe de implementação. Do ponto de vista do usuário, identificar o que é entrada e saída, é suficiente. Portanto no contexto deste trabalho, entende-se que o objetivo do *pipeline* é gerar representações visuais baseadas em áudio.

Concebido de forma a gerar visualizações semelhantes ao trabalho *Flickr Flow*, o *pipeline* apresentado tem seus componentes do tipo *Filter* baseados em características existentes em sua inspiração. Contemplando tais características, a proposta de implementação parte da leitura de áudio digital, e recomenda-se a utilização do formato PCM nesta etapa. Já que, segundo Black e Edson (1947), tal formato permite recriar a onda original de forma fiel, e embora nenhuma taxa de amostragem seja definida como obrigatória, cabe a implementação fazer uso de taxas semelhantes aquelas utilizadas para reprodução de música, baseadas na capacidade humana de percepção, mesmo que a proposta não aborde a audição de nenhum segmento de áudio.

Com isso, baseado no teorema da amostragem (SHANNON, 1949), torna-se possível garantir que o resultado visual, a faixa de frequências presentes no áudio, e o alcance auditivo do ser humano estejam relacionados. Definida a proposta para o componente ***audio source***, a utilização dos dados de áudio em componentes *Filter* torna-se padronizada no ponto de vista da entrada de dados, ou seja, o recebimento de seus parâmetros.

Tendo os dados padronizados, a etapa ***audio color extractor*** possui a responsabilidade de extrair uma escala de cores, que deve ser utilizada em conjunto com o resultado de formas e contornos, das etapas ***audio outlines*** e ***audio shapes***. A aplicação das cores presentes nesta escala deve obedecer a frequência de cada segmento disposto na visualização, assim, a escala deve fornecer cores como em uma sequência numérica com objetivo de evidenciar a informação disposta no local onde for utilizada.

Um exemplo da aplicação de escala de cores descrito por Green (2011), é a exibição de imagens na área da astronomia, onde é comum a necessidade de exibir intensidade de alguma fonte. Em seu trabalho, o autor evidencia que uso de mais ou menos brilho em um mesmo tom de cor para demonstrar intensidade, faz com que a percepção varie conforme a cor base

escolhida. Isto é, ao utilizar apenas brilho para trabalhar a escala, um tom de amarelo pode aparentar ter maior intensidade que um tom de azul que possua o mesmo brilho.

Para solucionar esta possível lacuna na percepção visual de escalas de cores, Green (2011) apresenta o esquema *Cubehelix* de cores. Ele é elaborado com base em um cubo, onde seus eixos representam três cores: R (vermelho), G (verde) e B (azul), e a composição de valores nas coordenadas escolhidas formam as cores da escala. Com domínio de 0 a 1 em cada eixo, a escala consiste em um trajeto que inicia em (0, 0, 0) e resulta na ausência total das três cores, ocasionando a cor preta, e tem fim em (1, 1, 1), exibindo como resultado a soma das três cores que é branco.

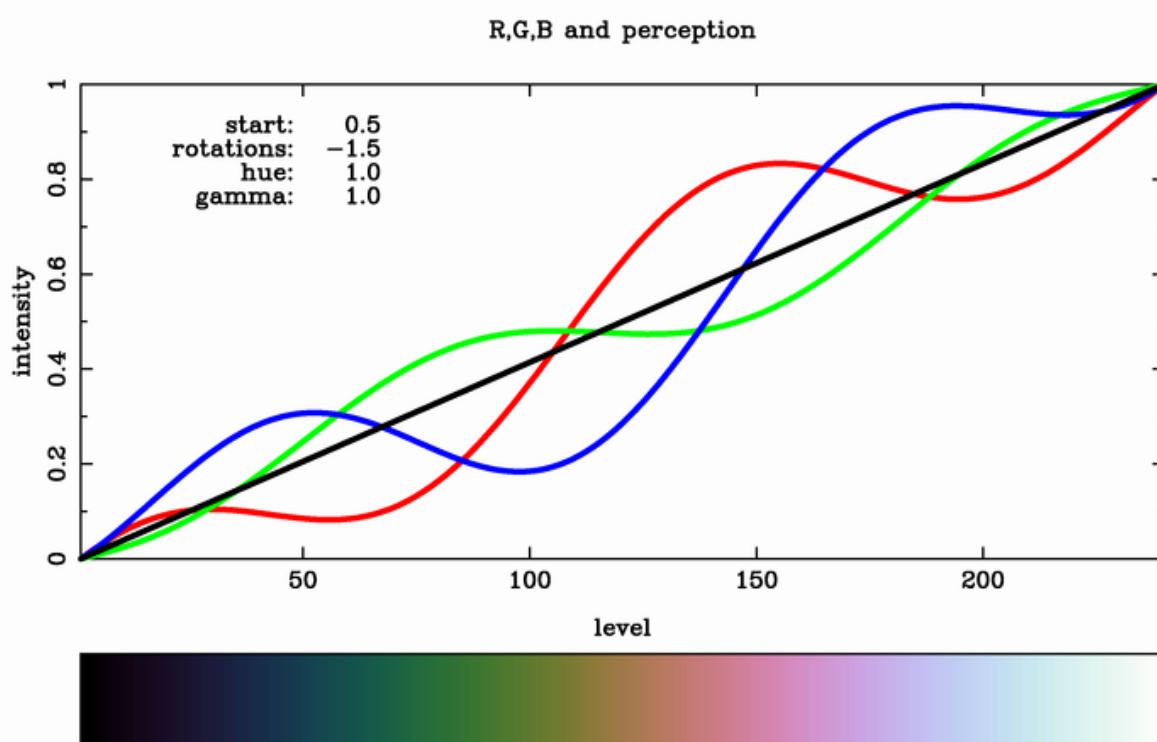


Figura 3.3 – Proposta *Cubehelix* para escala de cores

Fonte: Green (2011).

Porém, este trajeto não é uma simples linha reta que vai de ponto a ponto. Green (2011) cita um esquema de cores que foi utilizado na década de 1980 por John Fielden, no qual o trajeto elaborado consistia em um espiral ao redor da suposta diagonal. Baseado nesta proposta, Green então adaptou e otimizou este esquema com foco na percepção de cada uma das cores. Assim sendo, dentro da aplicação da proposta de visualização, é necessário utilizar ou elaborar uma escala de cores capaz de representar diferentes valores dentro de um intervalo sequencial, portanto recomenda-se o uso de alguma escala de intensidade conforme esta apresentada anteriormente. Ainda que a escala *Cubehelix* trate de intensidade, é necessário ressaltar que na

presente proposta, as cores serão utilizadas para representar as faixas de frequência, enquanto a intensidade do áudio será explicada em mais detalhes na definição de formas.

Realizada a escolha ou extração de cores, a etapa *painter* fará o preenchimento das mesmas sobre a definição de contorno e formas que formarão o resultado visual do estudo. A definição das formas, abstraída na etapa *audio shapes* não é feita diretamente com o PCM, dado que a coleção de valores fornecida pelo PCM por si, já gera o formato de onda. Portanto, é necessário utilizar uma ou mais das técnicas abordadas em MIR, e a sugestão de utilização inicial parte de uma formatação com dados distribuídos nos eixos de tempo, intensidade e frequência.

Diferente de um espectrograma em três dimensões, onde há uma largura fixa no eixo que concorre com o tempo, a definição das formas do atual trabalho é feita a partir de uma interpretação do histograma supostamente implementado no *Flickr Flow*, para isso buscamos como tal geralmente é apresentado. Neves e Pelaes (2008) descrevem um histograma como um conjunto de números, que no contexto de uma imagem em tons de cinza, indica o percentual de pontos que são encontrados em cada nível de cor. Este conjunto então, é geralmente apresentado em representações de duas dimensões, com os níveis de cinza distribuídos horizontalmente e quantidade de ocorrências indicadas verticalmente. É interessante ressaltar que a soma do conjunto totalizará a o percentual total ou quantidade de pixels presentes na imagem, dependendo de como a quantidade de ocorrências estiver sendo calculada. Considerando que imagens são usadas para gerar a representação *Flickr Flow*, a representação de séries temporais, considerando percentual de cores pode sempre exibir uma largura constante, dado que o histograma de uma imagem sempre totalizará a imagem por completo.

No entanto, como este trabalho utiliza-se de faixas de frequência e intensidade, sabe-se que os sons são compostos por variações de amplitudes e frequências, portanto não existe nenhuma garantia de que a soma de todas amplitudes através de todas as faixas resulte em um valor constante durante todos intervalos do áudio analisados. Isso faz com que a representação proposta, para manter a fidelidade do som, não possua esta característica.

Assim sendo, a representação em duas dimensões pode ser considerada como uma restrição para a definição das formas na etapa *audio shapes*, e ao caracterizar o resultado de *filterbanks* neste contexto como dados capazes de gerar um gráfico empilhado, a situação favorece a utilização de *Streamgraphs*, dado seu potencial de exibir séries temporais de forma legível e agradável. Então, a utilização deste tipo de representação para definir as formas que representarão o áudio aproxima o resultado esperado ao objetivo.

Logo, os dados utilizados para tal demarcação serão os de algum *filterbank*, pois eles consideram a presença de várias faixas, que são apresentadas em diversos momentos ou continuamente. Dessa forma, o desafio de representar as três conhecidas dimensões de espectrogramas em apenas duas é resolvido com essa abordagem. Exemplificando em comparação ao espectrograma tridimensional, um suposto pico em numa faixa de frequência terá sua representação como uma alteração de largura, dada a restrição dimensional.

Definidas as formas da representação gráfica, é momento de definir os contornos com a implementação da etapa ***audio outlines***, que utilizando cores semelhantes às já utilizadas ou tons neutros, conforme visto no trabalho *Flickr Flow*, servem para separar visualmente um faixa da outra. Estas transições devem ser baseadas nas formas já definidas na etapa ***audio shapes***.

Tendo as informações de formas e contornos, assim como as cores a serem utilizadas, é possível partir para a etapa ***painter***. Com objetivo de fazer uma reprodução em duas dimensões, expondo dados em dois eixos, tempo e intensidade, é momento de efetuar o preenchimento das formas com suas respectivas cores, a pintura dos contornos baseados em seus tons, e a aplicação de ambos sobre um mapa de coordenadas. Esta impressão, ainda em formato digital, não precisa necessariamente ser persistida, nem mesmo ser fornecida visualmente como parte do *pipeline*. Seu objetivo é ter um mapa de coordenadas de pixels, ou um conjunto de vetores, que sejam capazes de definir em um retângulo um gráfico de curvas no estilo *Streamgraph*. Identifica-se este resultado, dentro da definição de proposta, como plotagem linear.

Para finalizar o processo dos *Filters*, cabe à etapa de ***arc mapping*** realizar a transcrição da plotagem linear para uma representação em forma de arco. A sugestão para implementação, é efetuar uma conversão do eixo do tempo, supostamente horizontal, de distância para graus ou radianos. Juntamente com essa transformação, os valores verticais referentes a amplitude, podem permanecer os mesmos, porém sua utilização será feita em relação ao centro do arco para determinar a distância.

Isto é, em um exemplo, enquanto o *Streamgraph* ocupa uma altura de 200 pixels e uma largura de 1000 pixels, a proposta deste componente do *pipeline* solicita a transformação destes 1000 pixels em 360°. Efetuando o início da transformação em 0°, e considerando que este seja uma reta vertical acima do centro do arco, entendemos que os 200 pixels presentes na primeira coluna da plotagem linear, estarão dispostos no primeiro conjunto de coordenadas convertidas para arco, no grau 0. Proporcionalmente, as demais colunas deverão ser dispostas no arco até completarem os 360° exatamente na última coluna. Isso faz com que o resultado da proposta possa ser apresentado de forma muito semelhante ao trabalho que o inspira.

Por questões estéticas, uma sugestão para a implementação, é a aplicação de uma compensação no eixo vertical da plotagem linear. Este ajuste faz com que a representação fique um pouco mais afastada do eixo central, tornando-a mais semelhante à obra em que se baseia.

Por fim, a etapa ***render*** é responsável pela saída do pipeline de forma que o usuário possa visualizá-la como imagem.

Disponibilizados como resultados deste capítulo a proposta de implementação juntamente ao *pipeline*, tem-se o conjunto necessário para iniciar a implementação do modelo que é apresentada no próximo capítulo.

4 DESENVOLVIMENTO DO PROTÓTIPO

Com base na proposta de desenvolvimento, um protótipo foi elaborado de forma a atender a pretensão do atual trabalho e seu desenvolvimento é descrito neste capítulo. Dessa forma é abordada a escolha de ferramentas, desde a linguagem base, bibliotecas para análise de áudio e *frameworks* de visualização, assim como o desenvolvimento, decisões tomadas e restrições que surgiram durante a implementação.

4.1 Requisitos

Para apoiar as decisões tomadas no durante planejamento e execução do desenvolvimento, alguns requisitos foram levantados e são considerados nas etapas do processo. Eles são elencados na lista a seguir e são descritos e atendidos neste capítulo.

RQ1) O modelo deve gerar representações visuais estáticas.

Sabe-se da importância da interação presente nas ferramentas de visualização de dados, contudo o presente trabalho não aborda a interação do usuário com a representação gerada. Portanto a única interação presente deve ser a entrada de áudio no início do processo. Isto remete a visualização estática que se encontra no trabalho *Flickr Flow* (VIÉGAS; WATTENBERG, 2016).

RQ2) A fonte de dados do modelo deve ser áudio.

As representações geradas deverão ser baseadas em áudio, conforme objetivo geral do trabalho. Formatos intermediários podem e devem ser utilizados internamente na construção do modelo, porém não precisam ser aceitos como entrada, nem expostos como saída. A disponibilização do modelo poderá posteriormente ser adaptada em trabalhos futuros.

RQ3) As partes do modelo devem poder ser substituídas sem grande impacto.

Seguindo os conceitos de *pipeline* vistos no capítulo 1, os módulos de um fluxo de dados podem ser intercambiáveis (MORELAND, 2013). No desenvolvimento do modelo proposto, tornar todos os elementos do *pipeline* intercambiáveis pode não ser aplicável devido aos formatos de dados com qual cada um trabalha, porém, o presente requisito exige uma separação em pelo menos três módulos: Extração de Características de Áudio, Processamento de Dados e Geração de Representação Visual.

RQ4) Cores devem ser utilizadas para facilitar o reconhecimento de características.

Conforme exposto no *pipeline* definido no capítulo 3, a etapa *audio color extractor* deve ser capaz de extrair um conjunto de cores do segmento de áudio analisado. É necessário ter o cuidado de avaliar o quão relevante esta escala de cores é no resultado do modelo.

RQ5) A informação exibida deverá estar em formato radial.

Baseado no trabalho *Flickr Flow* (VIÉGAS; WATTENBERG, 2016), sabe-se que os dados do ano analisado estão dispostos sequencialmente em ângulos que vão de 0° a 360°. A representação temporal do segmento de áudio deverá apresentar dados em formato radial, expondo a linha temporal ao redor do eixo central, e assim como o trabalho de referência, deverá concluir a exibição nas coordenadas onde iniciou.

Definidos os requisitos, a escolha das ferramentas para o desenvolvimento da proposta é abordada na seguinte seção.

4.2 Seleção das ferramentas de desenvolvimento e prototipação

Embora a linguagem *JavaScript* tenha sido criada exclusivamente para *Web* em 1995 (SEVERANCE, 2012), Ryan Dahl desenvolveu em 2009 uma abordagem chamada de *Node.js*, que era capaz de executar o código na linguagem em servidores. Não foi a primeira implementação capaz disso, pois criado em 1997, e depois liberado pela *Mozilla* em 1998, o *Rhino* também é uma ferramenta que executa código *JavaScript* em servidores (MOZILLA, 2016). A diferença está na arquitetura utilizada por Ryan, que baseada na V8, mecanismo que executa *JavaScript* e é utilizado no Google Chrome, alcançou uma performance excepcional. Sua implementação em C e C++, ainda possibilita que desenvolvedores façam uso de ferramentas já existentes em outras linguagens, sem que seja necessária uma implementação própria (SEVERANCE, 2012).

Como resultado, *JavaScript* alcançou potencial para a criação de aplicações robustas para servidores, e a limitação de execução em navegadores deixou de existir. Isto fez com que a linguagem se popularizasse ainda mais, o que é possível perceber dado que a mesma se encontra em sexto lugar no índice TIOBE (TIOBE, 2016) e tenha o maior número de repositórios ativos no *Github* (GITHUT, 2016), uma rede social que hospeda projetos de código aberto. Com base nestas constatações, é possível perceber a popularidade da linguagem no meio de desenvolvimento de *software*. Ainda assim, para o contexto do atual trabalho é necessário averiguar a viabilidade da implementação do modelo com esta linguagem, e para isso deve-se verificar como pode ser efetuada a interpretação de áudio, o processamento, e por fim, a

representação visual. A evidências que comprovam a possível utilização da linguagem *JavaScript* são descritas nas próximas seções do capítulo.

4.2.1 Extração de características de áudio

Em um estudo sobre diversas ferramentas de extração de características de áudio, Moffat, Ronan e Reiss (2015) avaliaram 10 opções (ferramentas) seguindo quatro dos seis critérios do modelo de Cranfield, assim revisando cobertura, esforço, apresentação e latência de cada um dos sistemas aplicados na obtenção de informações. Dessa forma o modelo de Cranfield adaptado por Moffat, Ronan e Reiss (2015) considera as seguintes características:

1. **Cobertura:** a quantidade de características extraídas em conjunto com pré e pós-processamento.
2. **Esforço:** interface com o usuário tendo como parâmetro a facilidade de uso e documentação. Moffat, Ronan e Reiss (2015) o descrevem como sendo uma forma de definir o quão desafiador um sistema pode ser para seu usuário.
3. **Apresentação:** os formatos de saída e consistência dos resultados, que consistem em como a informação recuperada pela execução do sistema sob avaliação é apresentada ao usuário, e no contexto do presente trabalho se resume ao formato de saída das análises.
4. **Latência:** se refere a eficiência computacional de cada uma das ferramentas.

Dentre bibliotecas disponíveis em linguagens mais consolidadas no meio de processamento de dados, como C, C++, Matlab e Python, destaca-se no contexto deste trabalho a Meyda, pelo fato de ser escrita em *JavaScript* (MOFFAT; RONAN; REISS, 2015). Assim, os próximos parágrafos explicam como os estes quatro critérios, de Cranfield, adaptados em (MOFFAT; RONAN; REISS, 2015) são atendidos pela biblioteca Meyda.

Sabe-se que a Meyda foi concebida com objetivo de atender a extração de características em tempo real, no contexto da *Web Audio API* em navegadores. Dentre suas funcionalidades, considera-se recursos que são úteis em aplicações diversas, incluindo recuperação de informação, visualizações complexas de áudio e jogos (RAWLINSON; SEGAL; FIALA, 2015).

No contexto de **cobertura**, a implementação da Meyda foi baseada em outras bibliotecas, e cita-se a YAAFE, que originalmente foi desenvolvida em C++ com integrações para Python e Matlab, como uma das suas referências. Elaborada com foco em eficiência computacional e processamento em lotes, a YAAFE possui extração de características que

conforme o estudo de Moffat, Ronan e Reiss (2015), são apenas de baixo nível. A diferenciação entre alto e baixo nível é de que recursos de baixo nível são aqueles adquiridos diretamente do sinal de áudio, geralmente em uma análise quadro a quadro, como por exemplo a taxa de cruzamento em zero (*zero-crossing rate*), o centro espectral ou energia de sinal, enquanto tem-se acordes e notas como elementos de alto nível.

Sendo assim, no cenário de **cobertura** de funcionalidades, mesmo não sendo a ferramenta com maior número de características disponíveis para extração, o conjunto de recursos fornecido pela Meyda atende a análise espectral presente na proposta, dado que ela disponibiliza diversos *filterbanks* baseados no domínio da frequência, como Espectro de Amplitude, Espectro de Potência, além de uma implementação de MFCC (RAWLINSON; SEGAL; FIALA, 2015).

Tendo conhecimento de que os recursos disponíveis na biblioteca atendem a proposta, parte-se para o segundo item da adaptação do modelo de Cranfield, denominado **esforço**. A ferramenta Meyda apresenta uma documentação rasa, conforme apontado por Moffat, Ronan e Reiss (2015), e foi catalogada como uma biblioteca de *software*, já que ela apresentava apenas uma API – *Application Programming Interface*, para implementação e exigia a escrita de um programa para utilização. Sendo assim, ela não possuía outras formas de interação como recursos de GUI - *Graphical User Interface* e CLI, *Command Line Interface*, que também faziam parte da avaliação dos autores.

Contudo, um estudo conduzido pelo autor do presente trabalho averiguou que em novembro de 2015, a Meyda deixou de ser apenas uma API ao fornecer suas análises espectrais também em pelo uso de CLI (GITHUB, 2016). Isto é, a restrição de uso, que anteriormente exigia a escrita de um programa para utilização da biblioteca deixa de existir e é possível então extrair características de um dado segmento de áudio sem precisar necessariamente escrever um programa.

No contexto de **apresentação**, conforme a adaptação do modelo de Cranfield, é possível observar que na própria documentação da ferramenta Meyda são citados exemplos para utilização de seus recursos em navegadores e por meio da CLI, gerando a saída em arquivos. Os formatos de áudio e vídeo geralmente utilizados na web são suportados por meio da *Web Audio API*, enquanto a CLI está preparada apenas para áudio no formato wav (GITHUB, 2016). Como saída de dados, a utilização da biblioteca como API faz com que a informação já seja fornecida como estruturas de dados da própria linguagem *JavaScript*. Não muito diferente desta estrutura fornecida pela API, os comandos executados via CLI podem gerar arquivos no formato JSON, formato que nada mais é que a notação de objetos *JavaScript*. Além deste

formato, também é possível gerar arquivos no formato CSV (*comma separated values*), que são conjuntos de valores separados por vírgulas (RAWLINSON; SEGAL; FIALA, 2015).

Assim sendo, a utilização da ferramenta torna-se viável para análise dentro e fora de navegadores utilizando estruturas de dados comuns na linguagem *JavaScript*. A restrição ainda existente é sobre o formato de entrada de arquivos, e para este cenário é possível utilizar alguma solução de conversão de arquivos.

Um ponto de atenção para tal característica, é que a documentação do formato de saída é uma das barreiras mais significantes ao se trabalhar com GUI's e CLI's para integrar soluções de Recuperação de Informações de Música (MOFFAT; RONAN; REISS, 2015). O mesmo não é tão preocupante ao se trabalhar com API's, dado as estruturas utilizadas geralmente já são implementadas ou estão disponíveis na linguagem utilizada, portanto formatos de arquivos tornam-se irrelevantes nestes casos.

Finalizando as características revisadas pelo modelo de Cranfield, a **latência** consiste no conhecimento do tempo necessário para realizar uma tarefa, e o estudo de comparação utilizado como referência (MOFFAT; RONAN; REISS, 2015) discute a complexidade computacional das ferramentas de extração de características além de identificar quando elas são métodos de tempo real, ou seja, que são capazes de extrair informações enquanto o áudio sob estudo está sendo executado, e os métodos off-line, que geram um resultado estático a partir de uma entrada de dados. A importância deste tipo de comparação, no caso de métodos off-line é relevante em cenários onde performance é um requisito importante, e tanto velocidade, como processamento distribuído podem ser cruciais.

Embora o estudo de Moffat, Ronan e Reiss (2015) não tenha dados da biblioteca Meyda para performance *off-line*, dado que tal funcionalidade só veio a estar implementada em dezembro do mesmo ano (GITHUB, 2016), conforme descrito anteriormente, este indicador não impacta no contexto do trabalho atual, dado que eficiência computacional não faz parte dos requisitos. Assim sendo, se conclui que baseado na adaptação do modelo de Cranfield, que foi aplicado na comparação de ferramentas, a biblioteca Meyda atende as necessidades de trabalho vistas até o momento.

4.2.2 Apoio para prototipação

Tendo uma opção escolhida para a extração de características de análise de áudio, é possível considerar que resta ainda o processamento e geração das representações visuais. O processamento dos dados disponibilizados pela biblioteca Meyda pode ser realizado

diretamente em *JavaScript*, sem necessariamente ter o uso de bibliotecas, porém para agilizar o desenvolvimento do protótipo, optou-se por utilizar algumas ferramentas disponíveis para programação que não são obrigatórias. Estas ferramentas são o Budo e o Ramda.

O Budo é um servidor de aplicação para desenvolvimento, que serve para agrupar arquivos *JavaScript* e possibilita o uso de novos recursos da linguagem, facilitando a utilização de módulos através da central NPM (*Node Package Manager*) de módulos (NPM, 2016). Já o Ramda, é uma biblioteca de uso geral, que visa facilitar tarefas com uma abordagem simples e funcional com o uso de uma API bem simples (RAMDA, 2016). Ela possibilita ao desenvolvedor trabalhar com as estruturas de dados já existentes na linguagem *JavaScript* com abordagens da programação funcional, trazendo vantagens como performance e imutabilidade, além de utilizar incentivar o uso de composição, ao invés de agregação ou herança, como geralmente é utilizado em linguagens orientadas a objetos.

4.2.3 Visualização de Dados

Para visualização de dados, considerando as tecnologias e padrões existentes para a web, sabe-se que designers utilizam diversas ferramentas simultaneamente. Os trabalhos são geralmente uma combinação que envolve técnicas e ferramentas diferentes, como por exemplo HTML (*HyperText Markup Language*) para o conteúdo das páginas, CSS (*Cascading Style Sheets*) para estilos, aparência e estética, *JavaScript* para interação e SVG (*Scalable Vector Graphics*) para gráficos vetoriais. A utilização da web como plataforma traz como um de seus grandes trunfos a cooperação de tais tecnologias graças a existência de uma representação compartilhada conhecida como DOM (*Document Object Model*). Esta representação expõe de forma hierárquica o conteúdo de uma página, permitindo que as diversas ferramentas atuem referenciando e manipulando os objetos ali presentes (BOSTOCK, 2011).

Apesar desta estrutura em comum, que possibilita a interoperabilidade de funções, diversas ferramentas focadas em visualização acabam por encapsular a DOM de maneira especializada, não exigindo de seus desenvolvedores o conhecimento em padrões, instrumentos e recursos. Embora este encapsulamento possa ser considerado interessante, dado que reduz significativamente o esforço e tornando equipes de desenvolvimento mais **eficientes**, a flexibilidade que se perde ao depender de tais abstrações pode causar outros problemas (BOSTOCK, 2011).

Além da existente curva de aprendizado para desenvolvedores, como exemplo destes problemas, há um possível custo relacionado a adoção, utilização da ferramenta por parte dos

desenvolvedores, e em seu trabalho, Bostock (2011) refere-se a esta característica como *accessibility* – **acessibilidade**. Então, para os usuários da ferramenta, isso significa que dificuldades no aprendizado de uma representação, podem impedir seu entendimento a fundo e limitar o potencial existente. Além disso, o acúmulo de camadas de abstração, pode dificultar a depuração, dado que algumas estruturas fundamentais só são expostas em momentos de falha (BOSTOCK, 2011).

Ainda é possível considerar a utilização de recursos de forma não otimizada ou até mesmo sendo bloqueados da visualização, e Bostock (2011) cita o uso de recursos como fontes, que podem ser especificadas através de CSS, sendo extinguidas de visualizações por conta de limitações de uma ferramenta, e não da plataforma. Outro exemplo são recursos gráficos que podem existir, porém não estão expostos para implementação direta. Para contornar este tipo de questão, geralmente extensões da biblioteca podem ser criadas, mas exigem um conhecimento profundo sobre o funcionamento da abstração. Estas questões resultam na redução de **expressividade**, que impacta o resultado do desenvolvedor ao utilizar representações intermediárias (BOSTOCK, 2011).

Com base nos pontos levantados, anteriormente, 1 - expressividade, 2 - eficiência e 3 - acessibilidade, Bostock (2011) cita o projeto Protovis, que durante dois anos coletou milhares de observações de usuários. Tais opiniões tornaram possível a ele refinar as preocupações em três objetivos específicos: Compatibilidade, Depuração e Performance – explicados a seguir.

Sobre **compatibilidade**, o autor explica que ferramentas não existem isoladas, mas em um ecossistema de componentes relacionados, e o reuso de tecnologia aproveita não apenas ferramentas, mas também conhecimento e material de referência existente. Assim sendo, tal aproveitamento impacta diretamente em acessibilidade. Por outro lado, a composição de ferramentas, pode fazer com que tarefas sejam atribuídas a ferramentas especializadas, tornando o processo como um todo mais eficiente e evitando complexidade de abordagens mais genéricas.

Durante o processo de desenvolvimento, a tentativa e erro é uma parte fundamental, portanto ferramentas acessíveis devem ser capazes de fornecer **depuração** quando o inevitável ocorre (BOSTOCK, 2011). Neste sentido, algumas já fornecem facilidade para explorar resultados interativamente e em tempo real, tornando tal característica um diferencial para usuários. Um exemplo disso, são as ferramentas de desenvolvedor que existem em navegadores e possibilitam interações rápidas, como ajuste de cores, adição de eventos, e inspeção dos elementos presentes.

Com foco na experiência que o usuário terá com visualizações, Bostock (2011) cita interação e animação como pontos que a tornam mais rica, porém este tipo de recurso, quando utilizado em abstrações de alto nível pode ser até mesmo inviável ao desenvolvedor, dado que para executar pequenas alterações interativas na visualização, talvez ele precise tornar o processamento da visualização redundante a ponto de torná-la computacionalmente ineficiente. Por este motivo, a **performance** é um item que possui bastante importância.

Com base nestes três pontos discutidos, Bostock (2011) apresenta o projeto *Data-Driven Documents* (D3 ou D3.js, como é conhecido na comunidade de desenvolvimento), que propõe uma linguagem específica de domínio para transformar a DOM (*Document Object Model*, que é base para páginas web) com base em dados e operações. Com esta abordagem, designers e desenvolvedores vinculam dados e estilos a elementos da árvore DOM, e dinamicamente a biblioteca executa transformações para gerar ou modificar conteúdo.

Por se tratar de uma linguagem específica de domínio (DSL, *Domain Specific Language*), o D3 funciona sobre a linguagem *JavaScript* e introduz a manipulação eficiente de elementos baseado nos dados, ao invés de propor uma abstração que devesse ser amplamente estudada antes de sua utilização e impactasse em todos os pontos de uso. Desta forma, todo conhecimento prévio em CSS, *JavaScript*, SVG e funcionamento da DOM é aproveitado sem impacto causado pela ferramenta (BOSTOCK, 2011). Além disso, o autor explica que o D3 não funciona como um *framework* tradicional, mas sim uma biblioteca. Isto se dá ao fato de que ele não interfere nem limita as decisões que um desenvolvedor pode tomar durante o desenvolvimento de um projeto.

Em um comparativo onde Bostock (2011) relacionou o D3 com outras ferramentas de visualização é possível verificar que em alguns cenários, a ferramenta não é superior a outras de mesmo objetivo. Em seu estudo, a performance medida pela taxa de atualização de tela em quadros por segundo de outras ferramentas é mais alta em relação ao D3. Por outro lado, a pontuação é inversa quando o quesito é tempo de inicialização das mesmas, e o D3 fica em primeiro.

Bostock (2011) ainda relata que a biblioteca está bem preparada para possibilitar o uso de futuros recursos, a serem incorporados, em novas implementações dos navegadores. Ainda, o trabalho de (Bostock, 2011) afirma que a **acessibilidade** (neste contexto entende-se a facilidade no aprendizado da ferramenta) poderá ser aferida conjuntamente com a adoção do D3 pela comunidade de desenvolvedores.

Baseado no estudo sobre o *Flickr Flow* (VIÉGAS; WATTENBERG, 2016) e sobre *Streamgraphs* (BYRON; WATTENBERG, 2008), nota-se que Martin Wattenberg está presente

em ambos os trabalhos. Ao estudar mais a fundo a criação dos *Streamgraphs* em sua pesquisa, o uso do D3 é explicado, e verificando a documentação da própria ferramenta, ela dispõe um algoritmo de balanceamento específico para *Streamgraphs*. Além disso, ao optar por uma biblioteca ao invés de um *framework*, o trabalho não se limita a uma gama de visualizações fornecida por uma ferramenta de mercado. Analisando estes pontos, conjuntamente com as decisões e vantagens apresentadas nesta seção, o trabalho opta pela utilização do D3 para visualização no desenvolvimento do modelo de visualização.

4.3 Implementação

Conforme apresentado anteriormente, optou-se pela biblioteca Meyda (Seção 4.2.1) para extração de características de áudio, enquanto a visualização utiliza a ferramenta D3, conforme justificado na Seção 4.2.3 deste trabalho. A implementação como um todo se dá baseada na linguagem *JavaScript*, utilizando como apoio recursos citados na Seção 4.2.2, tal como Budo, Ramda e NPM.

Escolhidas as ferramentas, o processo de desenvolvimento é iniciado, ocupando-se com a extração dos dados relacionados aos segmentos de áudio. O processamento destes dados será desenvolvido de forma a torná-los interpretáveis de forma simples pela ferramenta de visualização. Esta ferramenta, por sua vez, possui a responsabilidade de transformar os dados em uma visualização gráfica.

Em um protótipo, fez-se o uso da biblioteca Meyda com o recurso de extração de dados em tempo real, e sua utilização foi realizada a partir de um navegador, com o auxílio de um componente de mídia criado via *JavaScript*. Assim, um arquivo de áudio pode ser carregado e executado, enquanto uma função declarada para coletar os dados é chamada sempre que a biblioteca informar a captura e processamento de mais uma etapa concluída.

A configuração utilizada no analisador da biblioteca Meyda pode ser vista no quadro a seguir e consiste nos componentes de áudio presentes nos navegadores, no tamanho de *buffer* que define quantas amostras devem ser analisadas por vez, um conjunto de definição de quais características devem ser extraídas e a função que deve ser chamada sempre que a ferramenta finaliza a análise de uma janela de amostras.

Quadro 4.1 - Exemplo de código na linguagem JavaScript com a biblioteca Meyda para coletar características de áudio

```
1 var Meyda = require('meyda');  
2 var context = new AudioContext();  
3 var tune = new Audio('/audios/brazil.mp3');
```

```

4  var source = context.createMediaElementSource(tune);
5  source.connect(context.destination);
6
7  var data = [];
8  var meydaAnalyzer = Meyda.createMeydaAnalyzer({
9    "audioContext":context,
10   "source":source,
11   "bufferSize": 4096,
12   "featureExtractors": ["amplitudeSpectrum"],
13   "callback": function(features){
14     data.push(features.amplitudeSpectrum);
15   }
16 });
17
18 meydaAnalyzer.start(["amplitudeSpectrum"]);
19 source.mediaElement.play();

```

Fonte: elaborado pelo próprio autor.

No exemplo anterior, uma lista chamada *data* agrupa os resultados do analisador, que é iniciado pela função *start*, recebendo como parâmetros as características a serem extraídas. Mais ao final do exemplo de código, há uma chamada para iniciar a execução do áudio. É interessante ressaltar que a execução do áudio pode ser acompanhada no navegador pela reprodução audível dos dados.

As primeiras características a serem extraídas foram as análises espectrais de amplitude e potência, conhecidas dentro do Meyda como *amplitudeSpectrum* e *powerSpectrum*. Estas, baseadas em análises de Fourier, são extraídas por conjuntos de amostras, que possuem seu tamanho definido na linha 11 do Quadro 4.1. Cada conjunto de amostras interpretado e processado é então coletado para formação de um espectrograma. O tamanho do conjunto de amostras, ou *buffer*, deve ser configurado como alguma potência de 2, por exemplo 1024, 2048 ou 4096.

A escolha da quantidade de amostras está diretamente relacionada ao teorema da amostragem (SHANNON, 1949) – apresentado na Seção 2.1, que relaciona frequência e taxa de amostragem, obedecendo uma escala de 1 para 2. Ou seja, o resultado de cada conjunto de 1024 amostras, é uma lista com 512 valores, cada um referente a amplitude ou potência em uma diferente faixa de frequência. Quanto maior o tamanho do *buffer*, mais faixas de frequências são adquiridas, porém menos conjuntos de amostras são retornados.

Então, neste contexto, dada a dificuldade em verificar uma matriz de dados composta por valores numéricos distribuídos em milhares de colunas e centenas de linhas, notou-se a necessidade de visualizar os dados de uma forma mais eficiente na presente implementação. Portanto, para entendimento na visualização dos dados, utilizou-se um *plugin* do D3 capaz de gerar mapas de calor durante o desenvolvimento da extração de características.

Os mapas de calor foram utilizados como instrumento de visualização para entendimento de **três características** extraídas de um pequeno segmento de áudio. As análises espectrais sob estudo são *amplitudeSpectrum*, *powerSpectrum* e MFCC. O espectro de amplitude trouxe dados referentes a amplitude de onda capturada diretamente do segmento de áudio. O espectro de potência, é uma transformação do espectro de amplitude, onde cada valor é igual ao quadrado do seu valor em amplitude, de forma a intensificar as diferenças, e atenuar valores mais baixos. Já o MFCC, conforme estudado no capítulo 2, é apresentado em dois mapas de calor, o primeiro com os valores positivos e o segundo com o módulo de seus valores negativos. Esta abordagem foi utilizada já que o mapa de calor é melhor apresentado seguindo uma escala de valores positivos, uma vez que os valores extraídos através da MFCC retornam valores tanto positivos quanto negativos devido a utilização da DCT.

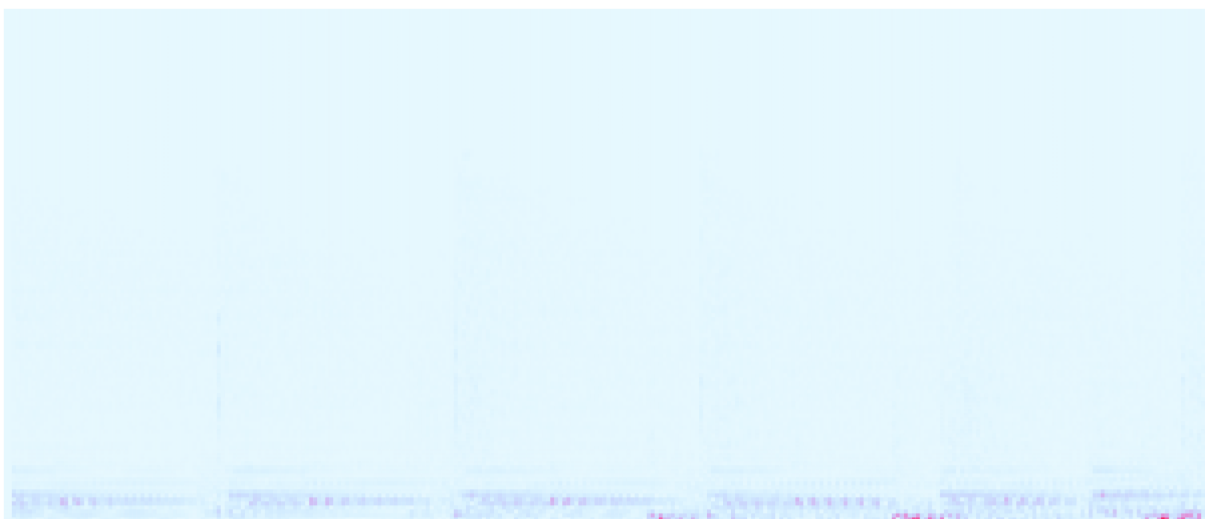


Figura 4.1 – Mapa de calor gerado a partir da análise *amplitudeSpectrum*

Fonte: elaborado pelo próprio autor.

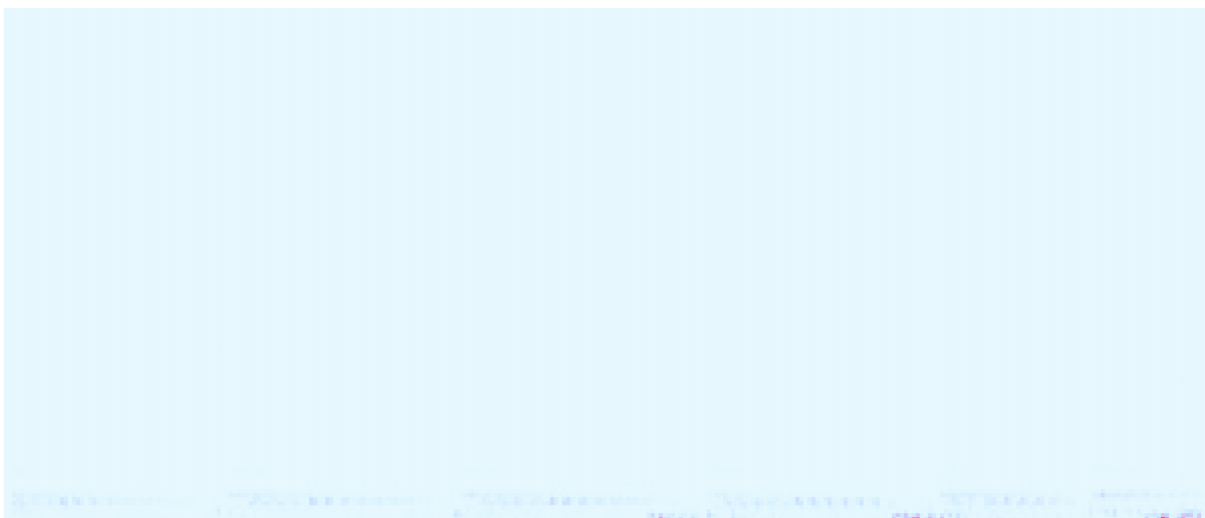


Figura 4.2 – Mapa de calor gerado a partir da análise *powerSpectrum*

Fonte: elaborado pelo próprio autor.



Figura 4.3 – Mapa de calor contendo valores positivos da análise MFCC

Fonte: elaborado pelo próprio autor.



Figura 4.4 – Mapa de calor contendo o módulo de valores negativos da análise MFCC

Fonte: elaborado pelo próprio autor.

Assim, a extração de características de áudio, que gera os dados que podem ser visualizados nas Figuras 4.1, 4.2, 4.3 e 4.4 demonstra que dados de áudio são capazes de gerar representações visuais, dessa forma atendendo ao **RQ2**. Analisando o resultado visual através de mapas de calor, verifica-se que a Figura 4.1 apresenta uma grande quantidade de dados representados pela cor azul claro, que estão relacionados ao valor zero, uma vez que a análise *amplitudeSpectrum* retorna um grande número de faixas de frequência. Já a Figura 4.2 apresenta a análise *powerSpectrum*, que utiliza os valores calculados em *amplitudeSpectrum* e os eleva ao quadrado evidenciado ou atenuando-os. Assim sendo, o resultado apresentado por estas duas análises é semelhante ao analisar a enorme quantidade de valores zerados.

Já a análise MFCC é apresentada nas Figuras 4.3 e 4.4. Devido ao fato de que seus valores podem ser tanto positivos como negativos, optou-se por exibir duas representações, uma contendo os valores iguais ou superiores a zero e outra com os inferiores ou iguais a zero. Com estas representações, foi possível verificar que a MFCC apresenta uma quantidade menor de valores zerados ao compará-la com as análises *amplitudeSpectrum* e *powerSpectrum*. Isto a torna mais interessante que as demais para a implementação de visualizações e pode ser justificado pela utilização da escala Mel de frequência, que reduz o número de faixas extraídas, conforme visto na Seção 2.4.

Dessa forma, o **RQ3** é parcialmente atendido, dado que a operação de extração de características já está isolada, e desde que gere uma matriz de números conforme as análises espectrais vistas, pode ser substituída por outras implementações.

Consolidada a extração de características, é necessário processar os dados de forma que eles sejam capazes de posteriormente retratar as visualizações desejadas. Para tal, é interessante compreender como os *Stacked Graphs*, vistos na seção 1.4, são construídos, e aplicar os conceitos envolvidos para os resultados da etapa de extração de características. A construção deste tipo de representação é baseada em 4 itens ordenados a seguir de acordo com a sua importância segundo Byron e Wattenberg (2008), e são explicados juntamente a exemplos ao decorrer da presente seção:

1. **Formato geral da representação**, dado que este possui influencia na curvatura de todas camadas individuais.
2. **Ordenação das camadas**, pois a ordem pode ser escolhida para atender critérios estéticos.
3. **Rótulos ou etiquetas** podem ser apresentados para dar mais informações sobre cada faixa.
4. **Escolha de cores**, que além de crítica, apoia os visualizadores a diferenciar uma faixa da outra.

Baseado no primeiro item citado por Byron e Wattenberg (2008), **formato geral da representação**, a geometria de um *Stacked Graph* é baseada no formato gerado pelo conjunto de todas as camadas. Assim sendo, no contexto da atual implementação cada camada corresponde a uma faixa da análise de áudio efetuada previamente.

As camadas da visualização devem ser apresentadas sem espaço entre si, e para cada momento apresentado na série, a altura total do formato geral deve equivaler à soma das camadas do conjunto. Dessa forma, cada componente também representa sua própria altura individualmente. Para atender estas duas colocações sobre espaçamento entre camadas e alturas, o formato geral começa a ser definido pela linha base, que é a linha inferior na qual a camada mais baixa será distribuída (BYRON; WATTENBERG, 2008). Concluída a distribuição de uma camada, as demais podem ser colocadas uma sobre a outra, formando o chamado **formato geral da representação**. Este então, acaba por ser definido pelos contornos presentes na parte inferior da primeira faixa, e superior da última.

Em gráficos empilhados convencionais, como o da Figura 4.5, é comum que a linha base seja zero constante. É este o motivo pelo qual o formato geral visto possui oscilações apenas no topo, enquanto a linha inferior é uma reta. O formato geral dos *Streamgraphs* é montado a partir da aplicação de um ajuste nesta linha base, e diferentes algoritmos são propostos para obter um resultado, que para Byron e Wattenberg (2008), seja mais agradável visualmente. O algoritmo mais simples para alcançar uma variação destes resultados, visa balancear a forma geral em torno de um eixo horizontal central, tornando a representação simétrica. Os demais algoritmos não seguem a simetria da abordagem simples, pois priorizam a estética e tendem a reduzir e suavizar as curvas de cada camada não obedecendo necessariamente a simetria no eixo.

O resultado da aplicação do algoritmo de simetria, citado anteriormente, pode ser visualizado na Figura 4.6. Ele é implementado com o apoio da biblioteca D3. A opção utilizada para realizar este balanceamento se chama *stackOffsetSilhouette* e faz a distribuição simétrica das camadas. Porém, se necessário, a biblioteca também fornece a opção *stackOffsetWiggle*,

que é proposta com base nos algoritmos que tendem a criar uma representação com faixas mais suaves (D3, 2016).

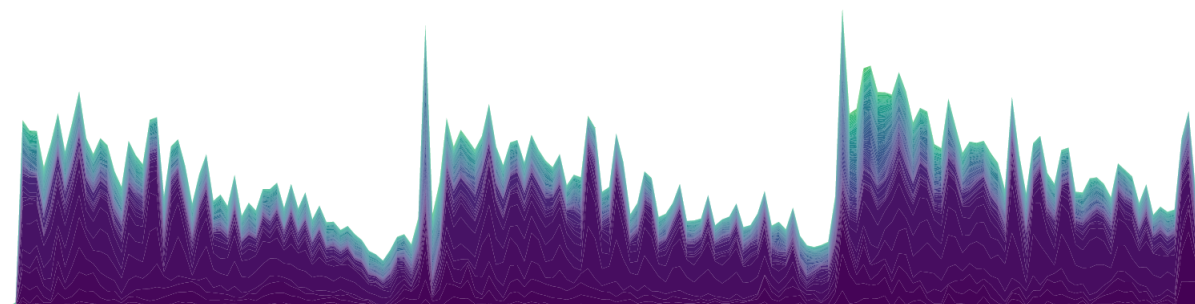


Figura 4.5 – Gráfico empilhado com linha base zero a partir da análise *amplitudeSpectrum*
Fonte: elaborado pelo próprio autor.

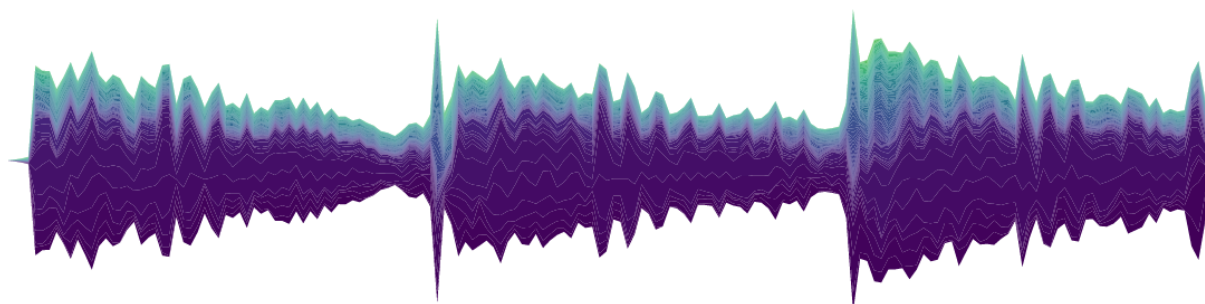


Figura 4.6 – Gráfico empilhado equilibrado simetricamente a partir da análise *amplitudeSpectrum*
Fonte: elaborado pelo próprio autor.

Conhecendo as operações que são possíveis na linha base de gráficos empilhados, os algoritmos presentes na biblioteca D3 são aplicados sobre os dados das análises espectrais geradas, obtendo dados que ao serem expostos visualmente resultam em representações semelhantes ao *Flickr Flow* (VIÉGAS; WATTENBERG, 2016), conforme pode-se notar nas Figuras 4.6, 4.7 e 4.8. Considerando a Figura 4.6, a grande quantidade de faixas na análise *amplitudeSpectrum*, assim como muitos valores zerados mostra como os dados diferentes de zero se concentram em poucas camadas. No cenário da Figura 4.7, que utiliza a análise *powerSpectrum*, isto torna-se mais evidente.

Com uma exibição de cores diferente em relação as Figuras 4.6 e 4.7, a representação da análise MFCC, vista na Figura 4.8 exibe também tons de amarelo e verde. Este comportamento ocorre, pois, a análise retorna um número menor de faixas, resultando em um conjunto reduzido de camadas. Além disso, conforme explicado sobre os mapas de calor da análise MFCC nas Figuras 4.3 e 4.4, há também um menor número de valores zerados.

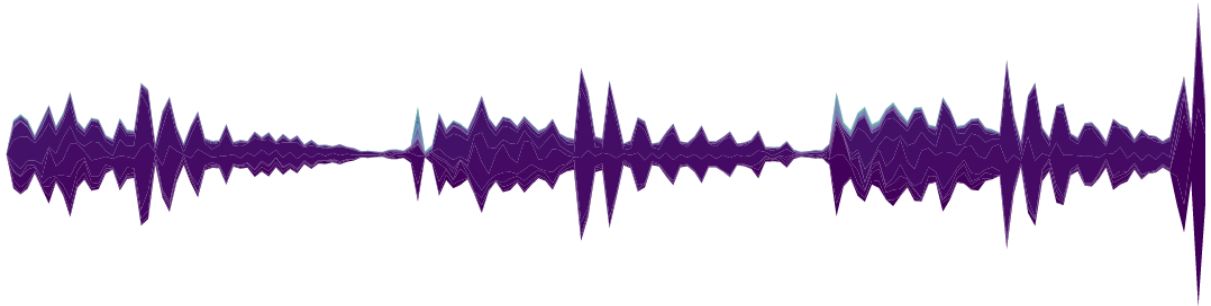


Figura 4.7 – Gráfico empilhado equilibrado simetricamente a partir da análise *powerSpectrum*
 Fonte: elaborado pelo próprio autor.

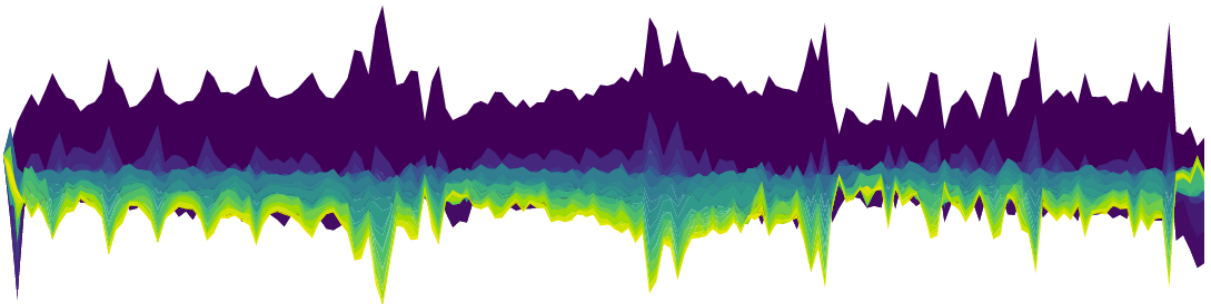


Figura 4.8 – Gráfico empilhado equilibrado simetricamente a partir da análise MFCC
 Fonte: elaborado pelo próprio autor.

A **ordenação das camadas** é dependente do contexto em que se encontra a visualização, assim como do seu objetivo (BYRON; WATTENBERG, 2008). O impacto que a ordenação possui com o apoio da estratégia utilizada sobre a linha base, como equilíbrio dos dados pode ser constatado nas Figuras a seguir, que representam o mesmo conjunto de dados.

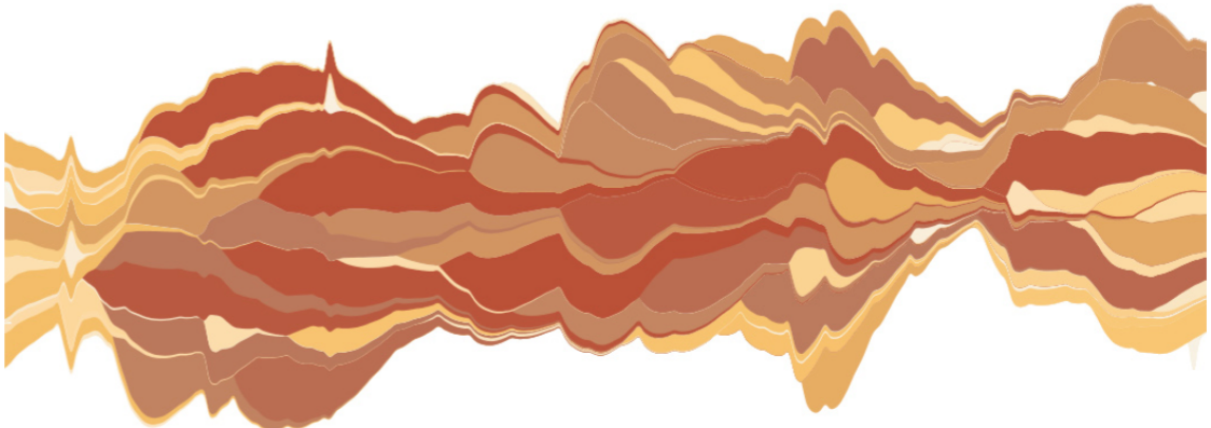


Figura 4.9 – Um conjunto de dados não ordenados, onde todas alterações abruptas causam uma alteração geométrica significativa
 Fonte: Byron e Wattenberg (2008).

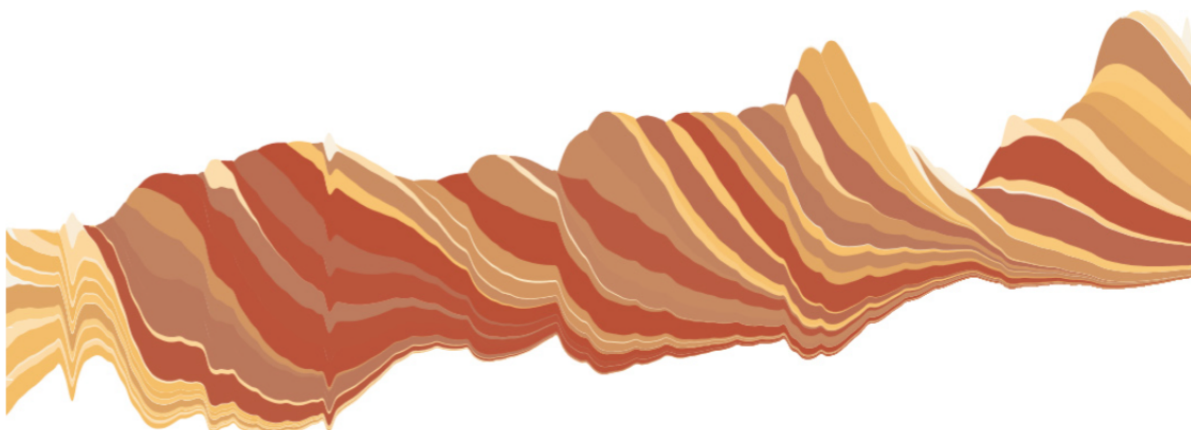


Figura 4.10 – O mesmo conjunto de dados, com ordenação simples dos dados, causando efeito de faixas diagonais

Fonte: Byron e Wattenberg (2008).



Figura 4.11 – O mesmo conjunto de dados, com ordenação que utiliza a estratégia de balanceamento “de dentro para fora”, com resultado esteticamente mais agradável

Fonte: Byron e Wattenberg (2008).

Assim, é possível verificar que a Figura 4.9, ao apresentar dados não ordenados pode possuir uma forma geral idêntica às Figuras 4.10 e 4.11, porém as camadas individualmente não seguem nenhuma ordenação, e a medida que ordenações são aplicadas, as visualizações podem ser mais agradáveis, como é o caso da Figura 4.11.

Com conhecimento destas tratativas, ao analisar o trabalho *Flickr Flow* (VIÉGAS; WATTENBERG, 2016), nota-se que existe uma ordenação sequencial, que acompanha diversos tons sem existir uma troca abrupta entre cores adjacentes na escala utilizada. Isto ocorre, pois, as faixas existentes tanto no *Flickr Flow* como no atual trabalho permanecem com valores durante toda a linha temporal, enquanto a ordenação utilizada por Byron e Wattenberg (2008) é utilizada em cenários onde faixas possuem intervalo finito dentro da representação.

Além disso, considerando o contexto de um conjunto de histogramas de cores do trabalho *Flickr Flow* (VIÉGAS; WATTENBERG, 2016) e da análise espectral de um segmento de áudio, entende-se que ambos tratam, cada um de apenas uma coleção de dados, enquanto a

visualização proposta pelo *Streamgraph* (BYRON; WATTENBERG, 2016) pode representar dados que não estão relacionados diretamente, como categorias diferentes. Portanto o uso de ordenação e mesmo cores diferentes pode não afetar o resultado final. Dessa forma, entende-se que a ordenação no contexto da atual implementação pode ser mantida com base na ordem já existente nas faixas espectrais.

Outro parâmetro considerado nos *Streamgraphs* é a **escolha de cores**. Com objetivo de possibilitar aos usuários a distinção de camadas, exaltar a dimensão dos dados ou trazer mais informações, a seleção da coloração utilizada é considerada crítica por Byron e Wattenberg (2008). No contexto do atual trabalho, assim como na visualização do *Flickr Flow* (VIÉGAS; WATTENBERG, 2016), as cores não possuem responsabilidade de distinção entre camadas, e sim de representação de dados. No trabalho de Viégas e Wattenberg (2016) verifica-se que os tons utilizados em cada camada representam suas respectivas cores. Ainda representando informação, porém de forma um pouco diferente, as camadas do trabalho atual utilizam cores em uma escala sequencial para representar qual a ordem da faixa exibida.

Este fator, de encontro com o objetivo do módulo *audio color extractor* presente na proposta de *pipeline* do capítulo 3, reforça a criticidade da extração de cores. Byron e Wattenberg (2008) explicam que a escolha de cores não é utilizada apenas para diferenciação ou contextualização, e aplicada ao atual trabalho a informação representada pela cor é de grande importância. Por este motivo, o modelo utilizará uma escala de cores fixa, e cada faixa exibida fará a seleção sequencial de uma cor dentro deste intervalo. Assim sendo, a implementação deverá considerar cores intermediárias dentro da escala para todas as camadas. Um exemplo, considerando uma escala de tons de cinza, seria a distribuição de cores começando em branco e gradualmente alcançando o tom preto.

Para facilitar a utilização de escalas de cores, a própria ferramenta D3 fornece escalas prontas para utilização, e a própria escala *Cubehelix* (GREEN, 2011) está disponível. No contexto da implementação atual da proposta, as visualizações utilizam o esquema *Viridis*, que também está presente na própria biblioteca (D3, 2016). Assim sendo, a utilização de cores atende ao requisito proposto, **RQ4**.

Em seu estudo, Byron e Wattenberg (2008) citam a **utilização de rótulos** para a informação aplicada em *Streamgraphs*. De modo geral, este item requer atenção pois trabalhos podem exibir centenas ou milhares de camadas. Já no trabalho *Flickr Flow*, utiliza-se um detalhamento sob demanda através da interação do usuário, que exibe algumas das fotos que representam o momento selecionado (VIÉGAS; WATTENBERG, 2008), porém, na

implementação atual nenhuma das abordagens será utilizada, dado o interesse de fazer a própria representação ter a informação.

Considerando o uso de *Streamgraphs*, além da própria seleção de parâmetros, algoritmos para balanceamento e resultado estético geral, nota-se a ausência de um trabalho em contornos na implementação do atual capítulo. No entanto, visualmente é possível notar que o *Flickr Flow* implementa alguns detalhes entre as camadas em tons neutros. O item *audio outlines* é citado na proposta, porém não implementado nesta fase do atual trabalho.

Considerando o processamento realizado até então, juntamente com a possibilidade de facilmente efetuar a representação visual dos dados por meio do uso da biblioteca D3, entende-se que as etapas *audio color extractor*, *audio outlines* e *audio shapes* estão concluídas, assim como a etapa *painter* do *pipeline*. Para implementação da etapa *arc mapping*, propõe-se uma adaptação da etapa *painter*, considerando duas pequenas alterações nas concepções de eixos e distâncias.

A primeira, é considerar o tempo total do segmento de áudio T , que na representação horizontal de *Streamgraphs* era equivalente a largura do resultado, como 360° . Isso faz com que a visualização radial termine no ponto onde começou, e para cada instante t em T , em torno do eixo C , existam dados para serem exibidos. Isto é, para um segmento qualquer de áudio, sua representação gráfica estará distribuída de forma constante ao redor de um eixo, até que alcance o ângulo em que iniciou.

A segunda adaptação, é em relação ao eixo vertical, que é representado pela altura nas representações de *Streamgraphs*. Na etapa *arc mapping* as distâncias das coordenadas que antes eram medidas em relação ao eixo horizontal agora devem ser calculadas em relação ao centro C da representação. Juntamente com a primeira adaptação, isso faz com que os dados presentes em cada instante t em T estejam representados a uma distância do centro C , que anteriormente era a distância em relação ao eixo horizontal.

Este conjunto de ações, aplicado ao resultado obtido anteriormente pela extração de cores, definição de contornos e formas torna a implementação apta a atender o requisito **RQ5**. Como resultado de tal adaptação, obteve-se um tipo de visualização denominado *Radial Streamgraph* pelo autor do trabalho. As mesmas análises do segmento de áudio, *amplitudeSpectrum*, *powerSpectrum* e MFCC, que são representadas com o *Streamgraph* convencional nas Figuras 4.6, 4.7 e 4.8 são representadas utilizando a técnica *Radial Streamgraphs* a seguir, nas Figuras 4.12, 4.13 e 4.14.

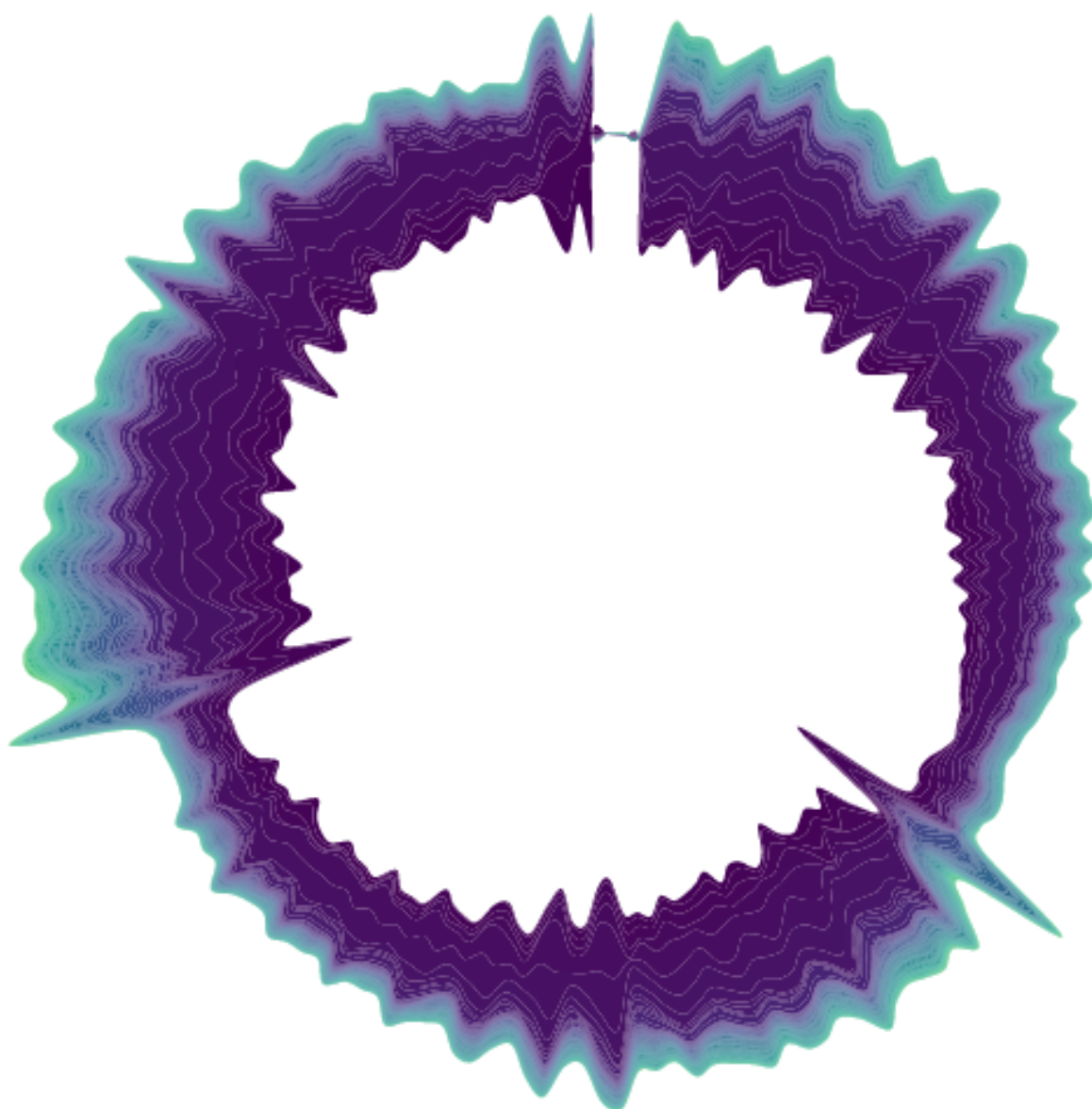


Figura 4.12 – *Radial Streamgraph* do *amplitudeSpectrum* de um segmento de áudio de 2 segundos

Fonte: elaborado pelo próprio autor.

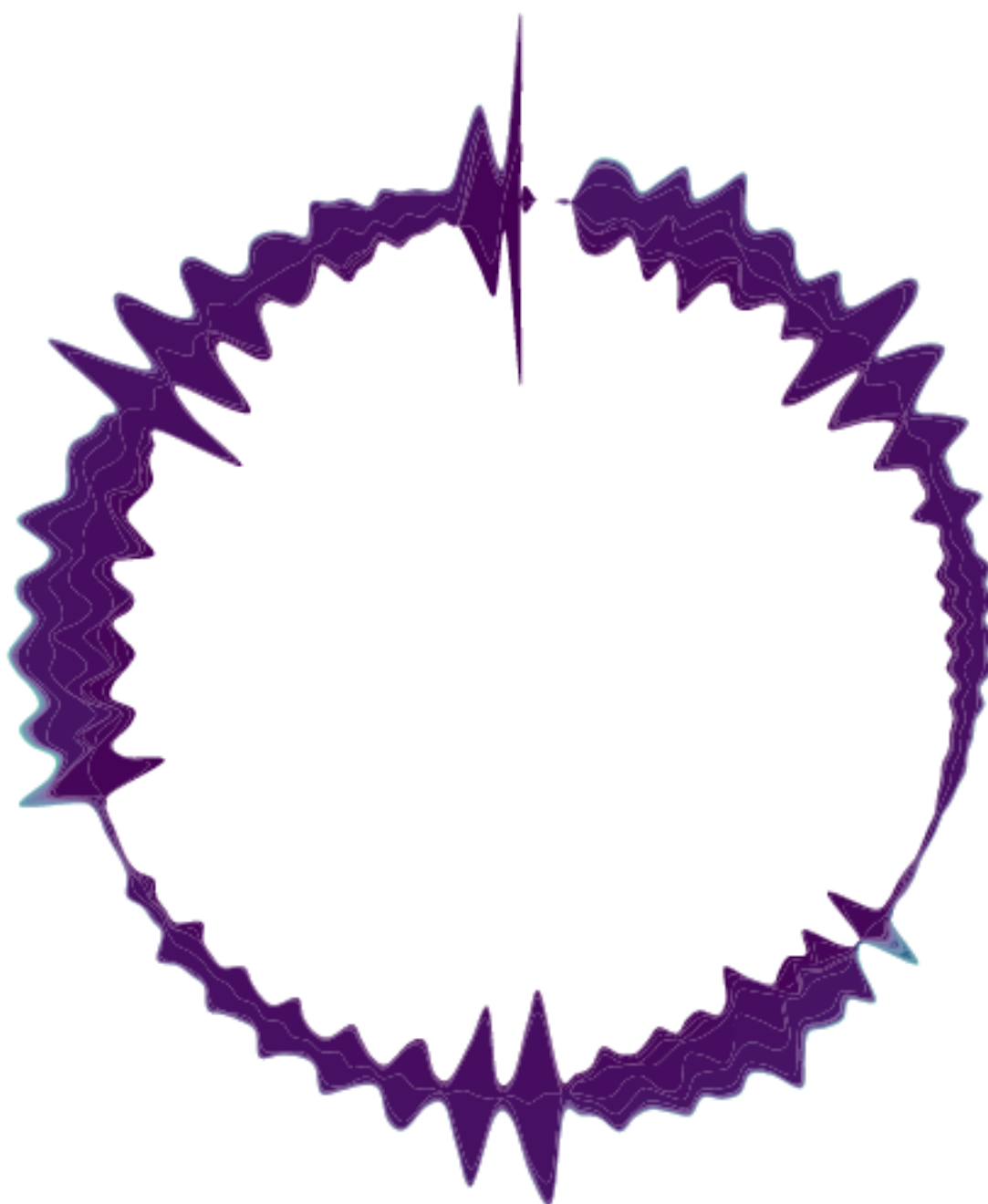


Figura 4.13 – *Radial Streamgraph* do *powerSpectrum* de um segmento de áudio de 2 segundos

Fonte: elaborado pelo próprio autor.

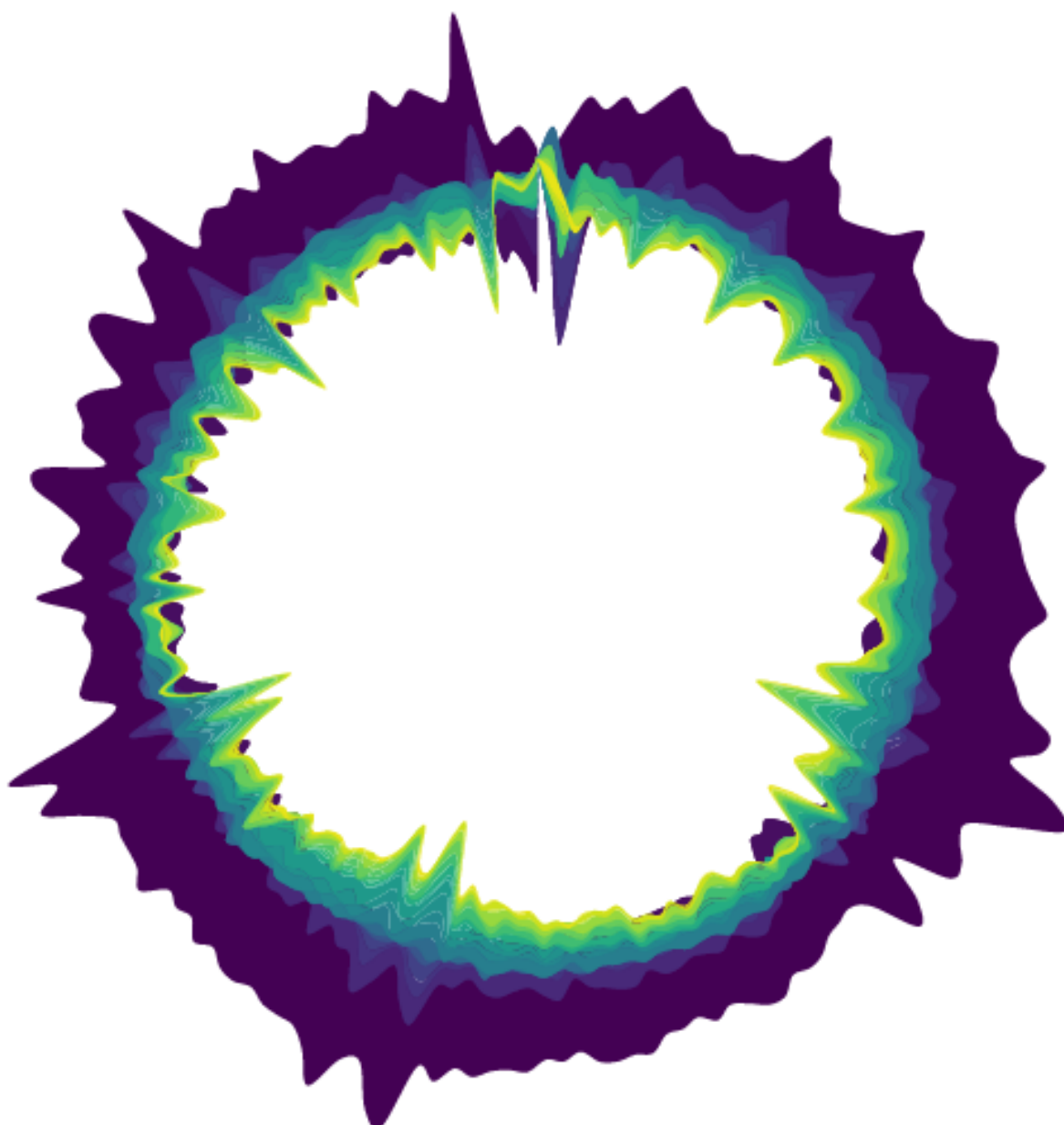


Figura 4.14 – *Radial Streamgraph* do MFCC de um segmento de áudio de 2 segundos

Fonte: elaborado pelo próprio autor.

Dessa forma, a representação em formato radial de *Streamgraphs* tornou-se possível. Considerando a observação feita sobre sobreposição de dados feita por Moreland (2013), uma análise comparativa de cada uma das representações radiais foi feita com camadas semitransparentes de mesma cor. Através de tal verificação, constatou-se que a MFCC posicionava camadas sobrepostas. Diante do objetivo do trabalho, este problema causava um resultado crítico, pois informação deixava de ser exibida ao usuário, como é possível constatar na Figura 4.15.

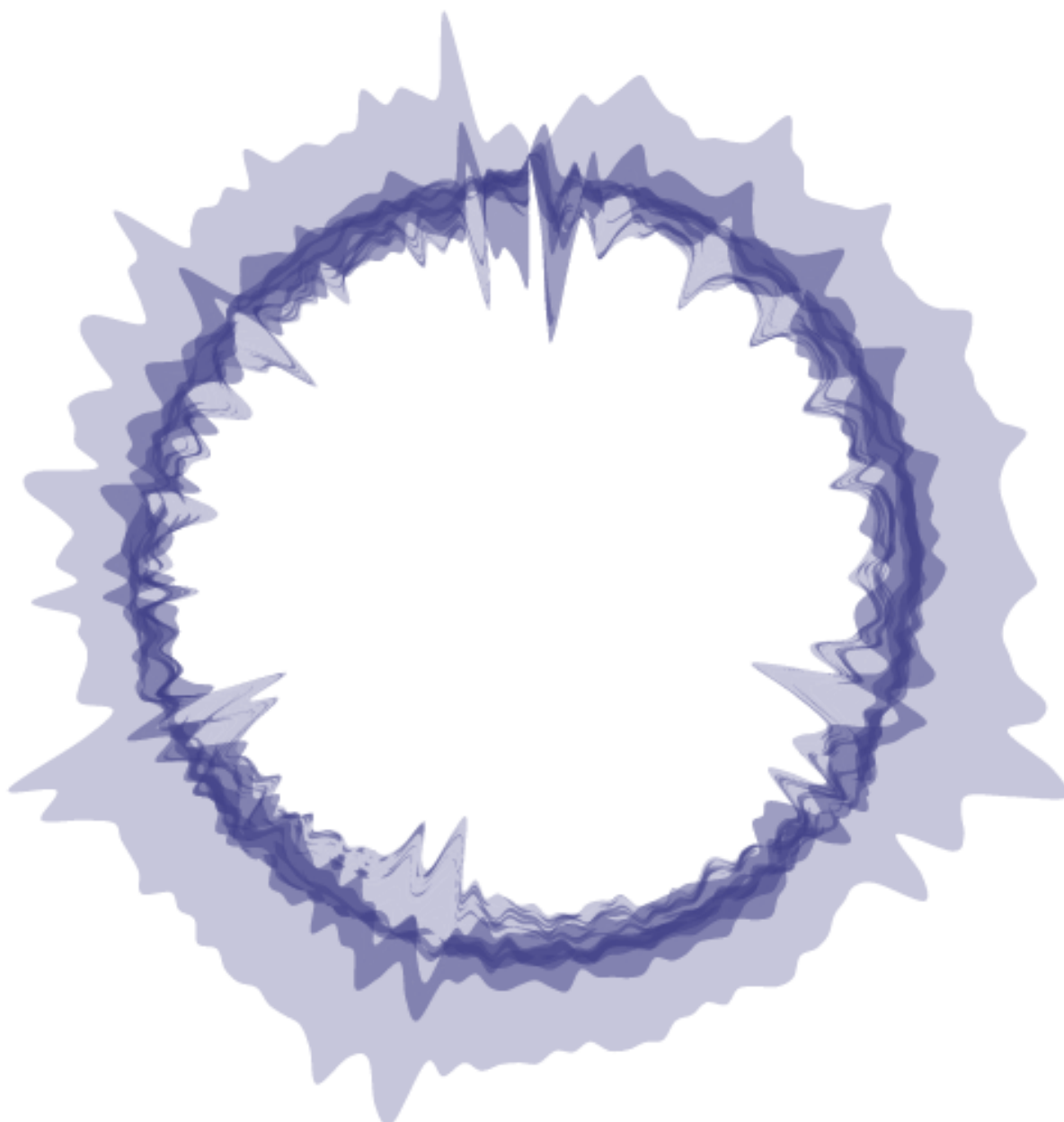


Figura 4.15 – Camadas semitransparentes de um *Radial Streamgraph* do MFCC de um segmento de áudio de 2 segundos

Fonte: elaborado pelo próprio autor.

Tal situação ocorre devido as operações que são efetuadas no MFCC. A transformação senoidal faz com que os valores de amplitude/potência, anteriormente positivos, tenham valores negativos durante o processamento, e no caso de não possibilidade de empilhamento, conforme visto na seção 4 do capítulo 2, uma técnica melhor deve ser utilizada para representações radiais de MFCC. Ao estudar possíveis soluções dentro da biblioteca D3, encontrou-se a opção de ordenação decrescente, que reduziu significativamente a sobreposição, mas ainda assim, não a extinguiu, conforme observado na Figura 4.16.

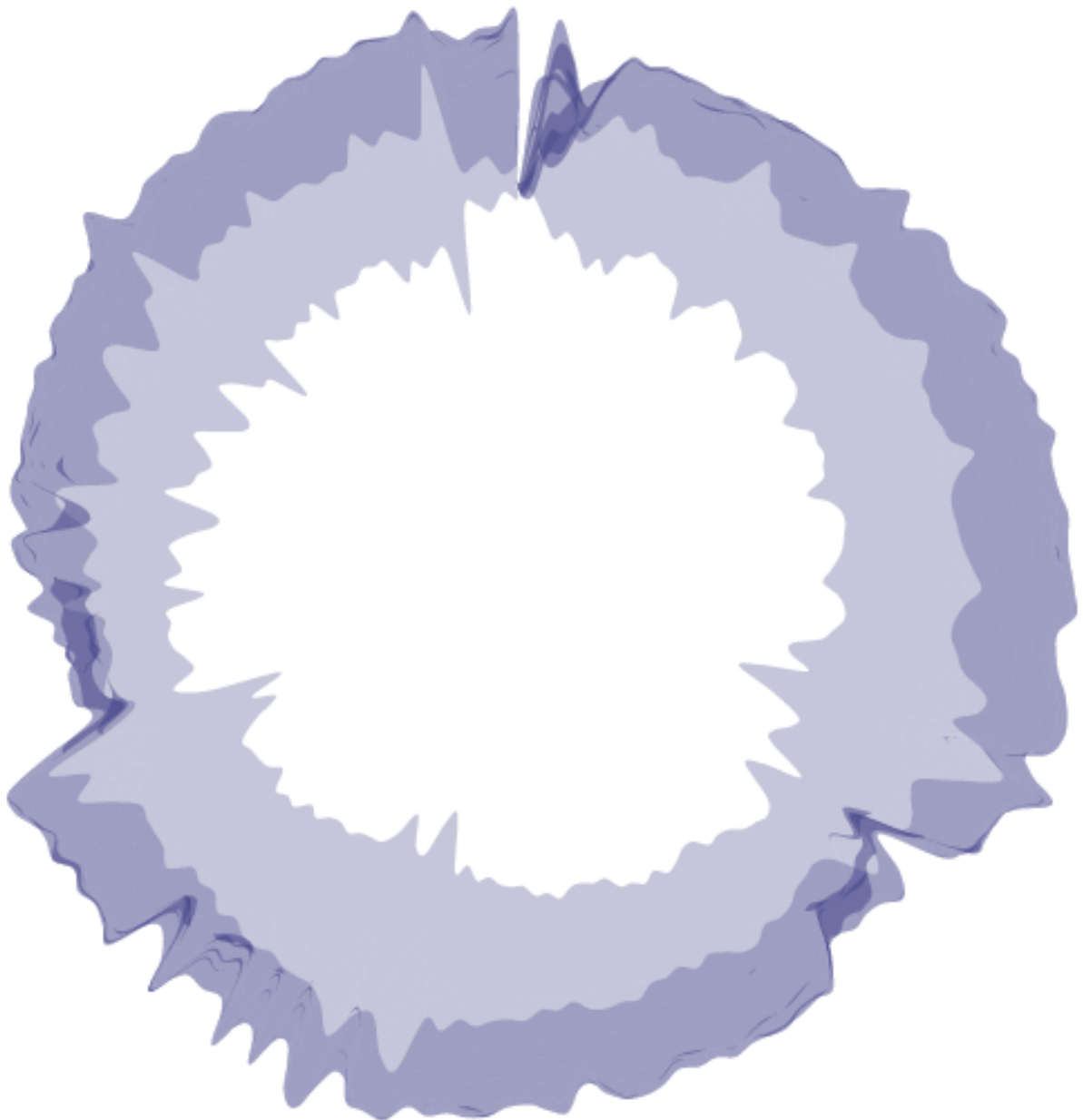


Figura 4.16 – Camadas semitransparentes ordenadas decrescentemente de um *Radial Streamgraph* do MFCC de um segmento de áudio de 2 segundos

Fonte: elaborado pelo próprio autor.

4.4 Consolidação da Implementação

Considerando os itens abordados no desenvolvimento, é possível verificar como os requisitos são atendidos. O modelo implementado gera representações estáticas (**RQ1**), como é possível verificar nas visualizações elaboradas pelo autor que representam as análises espectrais em formato radial (**RQ5**). A biblioteca Meyda foi responsável pela interpretação de arquivos de áudio, e a partir deles forneceu os dados utilizados para os experimentos (**RQ2**). A utilização da composição de bibliotecas e etapas, onde a extração de características foi feita em

uma matriz de valores, um processamento intermediário efetuava balanceamento de valores, e uma etapa final posicionava o resultado gráfico em formato radial, caracteriza a possibilidade de separar partes do modelo (**RQ3**). Além disso, a escala de cores é utilizada de forma sequencial, com objetivo de exibir informações sem precisar de legendas específicas para cada camada exibida (**RQ4**).

Dessa forma, detalhando o primeiro requisito é possível consolidar a implementação em três partes. Do ponto de vista da extração de características de áudio, a biblioteca Meyda atendeu este objetivo tanto por meio da linha de comando como pela exposição de sua API no navegador para análise em tempo real. A implementação utilizou apenas três das diversas características disponíveis para consumo pela ferramenta: *amplitudeSpectrum*, *powerSpectrum* e MFCC.

A questão de processamento, feita sobre os dados fornecidos pela extração de características de áudio, foi tratada com o apoio do D3, que embora seja uma ferramenta de visualização, trata o equilíbrio de dados, além da ordenação de camadas. Os dados trabalhados podem ser exibidos então tanto pelo próprio D3 como por qualquer outra ferramenta de visualização. Tal flexibilidade torna-se evidente pois durante o desenvolvimento, pode-se verificar o uso progressivo de um mapa de calor, gráfico de camadas empilhadas, *Streamgraph*, e por fim o *Radial Streamgraph*.

A parte de visualização ficou sob responsabilidade da biblioteca D3. Os dados obtidos e já processados foram passados para uma implementação de *Streamgraph* modificada, que baseada em adaptações de eixos foi capaz de gerar as representações radiais com semelhanças ao trabalho *Flickr Flow*. O resultado das etapas em conjunto, aplicado sobre a análise MFCC pode ser visto a seguir.

Optou-se pela utilização de MFCC pelo fato da mesma utilizar a escala Mel de frequências, sendo assim mais próxima a percepção sonora humana. É possível verificar que nas demais representações de *amplitudeSpectrum* e *powerSpectrum* a variação de cores não era tão significativa quando a presente no MFCC, e este ponto também foi considerado.

O resultado da implementação é então apresentado na Figura 4.17, podendo ser definido como um *Radial Streamgraph*, que utiliza dados de MFCC e ordenação decrescente da biblioteca D3.

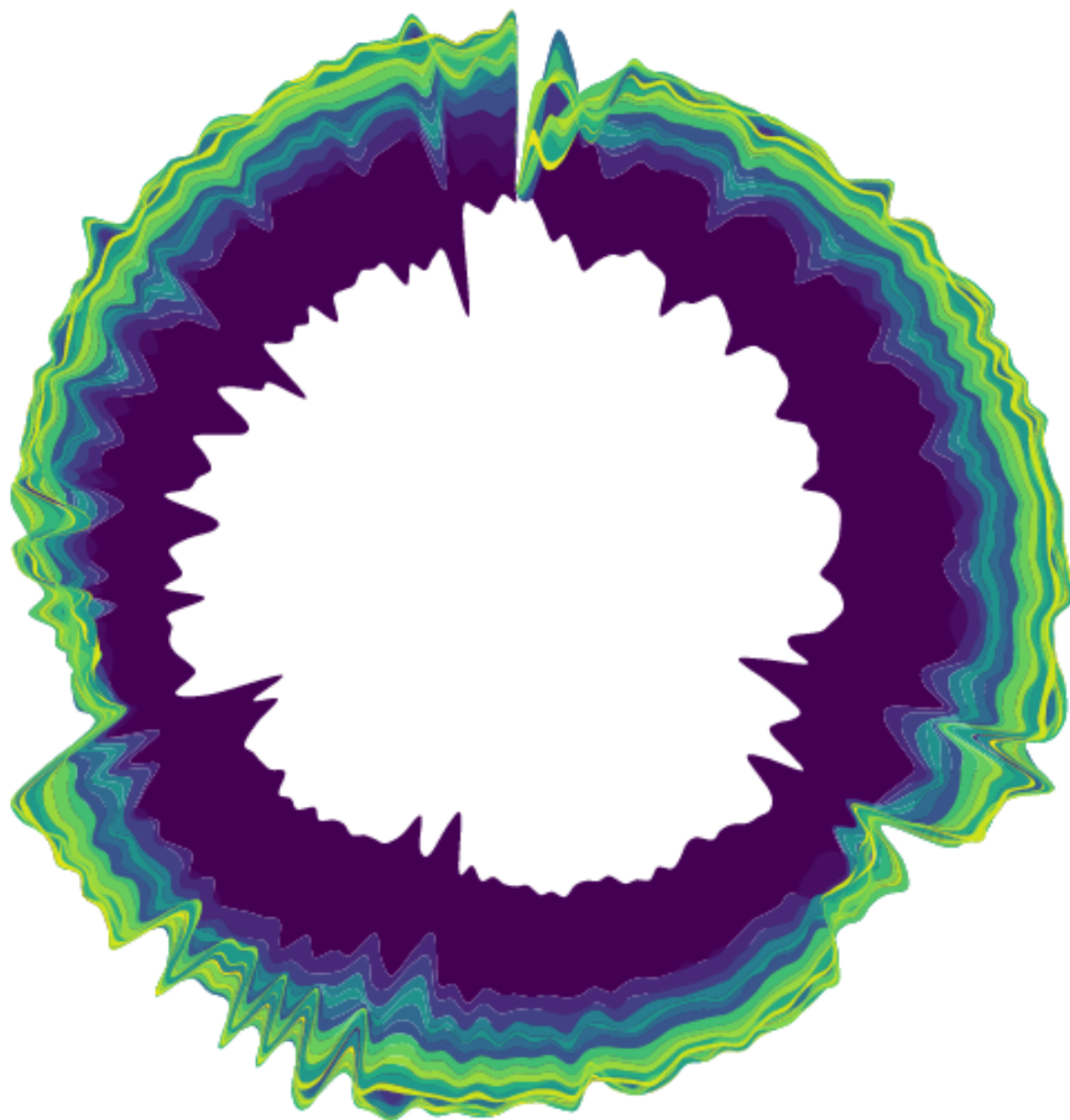


Figura 4.17 –*Radial Streamgraph* do MFCC de um segmento de áudio de 2 segundos com camadas ordenadas decrescentemente

Fonte: elaborado pelo próprio autor.

Concluído o desenvolvimento da proposta, evidenciado pela representação vista na Figura 4.17, o capítulo seguinte visa avaliar a representatividade da visualização gerada.

5 AVALIAÇÃO

Com objetivo de **avaliar** o modelo de visualização de áudio utilizando *Radial Streamgraphs* e **analisar** sua capacidade de expressar informação sobre o segmento sonoro utilizado, foi executado um processo de avaliação, com usuários, dividido em duas etapas, sendo elas:

1. **Treinamento para interpretação do modelo:** tem como objetivo apresentar aos usuários as informações exibidas no modelo de visualização, relacionando som com imagem;
2. **Aplicação de questionário para verificação da eficácia do modelo:** examinar, do ponto de vista do usuário, o quanto as visualizações geradas se relacionam com os segmentos de áudio analisados, sendo capaz de representá-los sem necessidade de som.

O treinamento aos participantes da pesquisa foi oferecido juntamente a uma introdução ao contexto do trabalho com objetivo de conduzir o público na pesquisa proposta. As próximas seções descrevem os objetivos da avaliação, a metodologia utilizada, o perfil dos participantes e a análise dos resultados.

5.1 Metodologia

Com a finalidade de verificar a eficiência das representações visuais de áudio geradas pelo modelo proposto, um treinamento precedeu um questionário que envolvia dados do perfil de participantes e sete visualizações para os usuários classificarem. A avaliação ficou disponível online em português no endereço <https://gregoriokusowski.github.io/audio-viz/survey> e em inglês no endereço <https://gregoriokusowski.github.io/audio-viz/survey/index-en.html> entre os dias 12 de setembro e 12 de outubro de 2016. Foi utilizado conteúdo estático para as etapas iniciais, e uma ferramenta de formulários para tornar possível a captura de respostas. Além de uma contextualização inicial, ela consistiu em quatro etapas: 1) Explicação, 2) Treinamento, 3) Questionário sobre Perfil e 4) Questionário de Avaliação do Modelo.

A **etapa de explicação** consiste na apresentação do estudo do trabalho como um todo, do modelo implementado e do objetivo com a atual pesquisa. A relação do áudio com o modelo gerado foi explicada de uma forma simplificada aos usuários, com base na Figura 5.1, exibida a seguir.

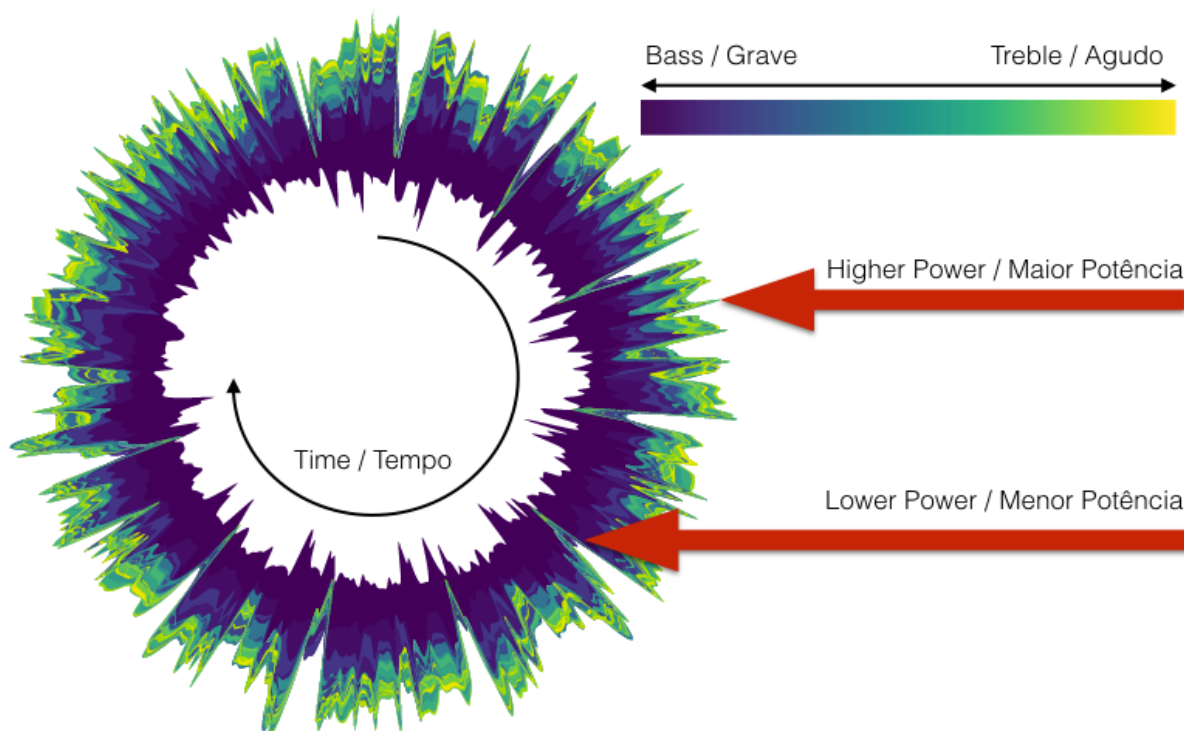


Figura 5.1 –Explicação do modelo utilizada na primeira etapa da pesquisa

Fonte: elaborado pelo próprio autor.

Após apontar as características do áudio presentes na visualização, a etapa de treinamento foi composta pela apresentação de 5 segmentos de áudio de estilos musicais diferentes, cada um contendo 10 segundos de som e seu respectivo modelo gerado. O resultado visual dos segmentos é então apresentado ao usuário, e a partir de um botão é possível interagir com a representação. A interação consiste na execução do áudio, juntamente com um ponteiro que demonstra o relacionamento do tempo com a música em execução.

A escolha das músicas foi feita com base em um acervo de amostras fornecido pela *Technische Universität Dortmund* (HOMBURG *et al.* 2005). A coleção é separada em categorias musicais, onde é possível optar pelo acesso aos áudios em si ou seus metadados, para o caso de estudos focados na área de *MIR*. Embora o foco dos dados fornecidos seja para aprendizagem de máquina e mineração de dados, os exemplos obtidos e a separação por categorias serviram como um cenário interessante para avaliar o modelo de visualização proposto. Assim sendo, o treinamento é composto pela visualização do segmento de áudio juntamente com a interação que executa o áudio juntamente com um ponteiro que indica o momento que está sendo tocado.

As cinco faixas escolhidas para treinamento, cada qual dentro de seu estilo musical, foram escolhidas aleatoriamente. Embora o acervo conte com nove estilos diferentes, optou-se

pelo uso de apenas 5, sendo eles de músicas eletrônicas, jazz, pop, rock e rap. Os trechos escolhidos, juntamente com seus respectivos estilos foram:

- Eletrônica: Migg-Slow_Motion.mp3
- Jazz: Coolface-Checkmate.mp3
- Pop: Sugarbuzz-Born_Again.mp3
- Rock: Tommy_Merry-Above_and_beyond.mp3
- Rap: Area_51_MC_s-Area_51_Show_Sampler.mp3

Este conjunto formou o treinamento, e julga-se necessário informar que a interação apresentada junto à visualização que executa o áudio e exibe a animação de ponteiro é usada apenas para o treinamento e não faz parte do modelo. Ao pressionar o botão *PLAY*, o ponteiro que se move em torno do centro da visualização, é iniciado em 12” e desloca-se em sentido horário, conforme exibido na Figura 5.2.

ELECTRONIC

PLAY

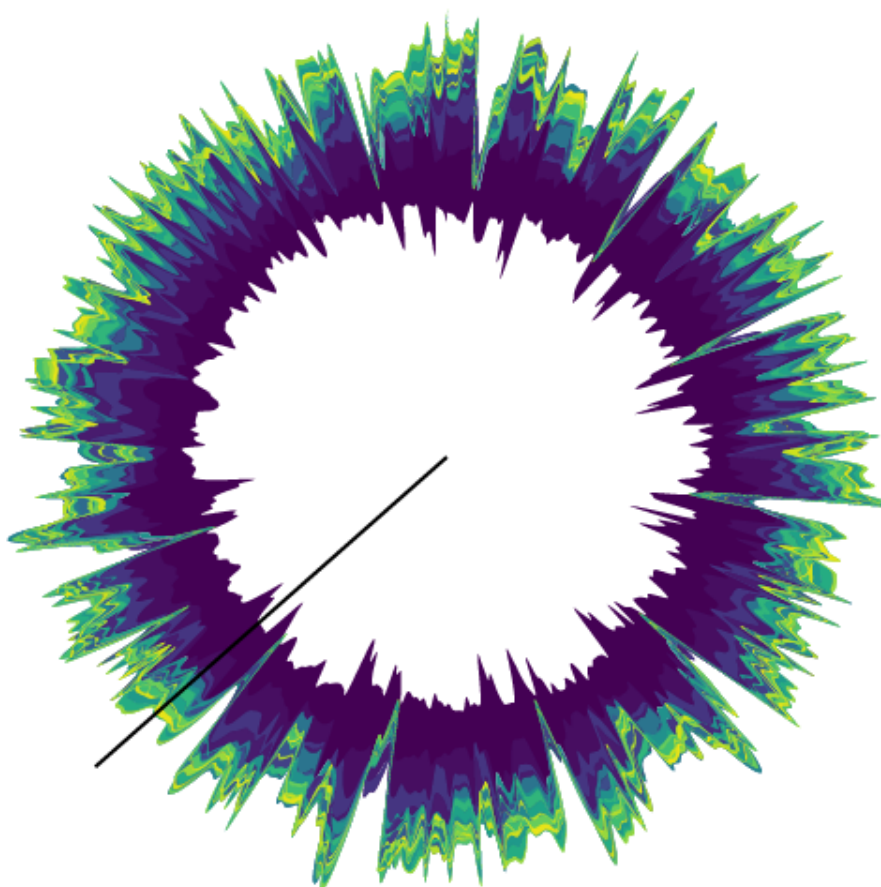


Figura 5.2 – Explicação sobre a animação de ponteiro utilizada no treinamento
Fonte: elaborado pelo próprio autor.

Para classificar os usuários e proporcionar uma análise mais completa dos resultados, um questionário de dados de perfil foi planejado e aplicado. Ele conta com questões que envolvem idade, gênero, experiência e formação. Em seguida, o último questionário exibiu sete representações geradas pelo modelo de visualização implementado, e solicitou aos participantes que classificassem cada item em um dos cinco estilos musicais presentes no treinamento. O objetivo deste é então, investigar juntamente com o conjunto de dados coletados de perfil, a assertividade do modelo conforme características dos participantes.

Alguns participantes contribuíram informando o tempo médio de resposta aproximado de 5 minutos, medido e fornecido por eles mesmos sem ferramenta ou acompanhamento. A pesquisa foi divulgada através de e-mail e publicada em duas comunidades da rede social *Reddit*. Uma delas focada em visualização de dados, e outra de músicos.

5.2 Perfil dos participantes

O questionário contou com respostas de 52 pessoas, e a única restrição para a participação seria alguma dificuldade de visão, que impedisse que o participante pudesse avaliar os modelos gerados ou a impossibilidade de ouvir os segmentos de áudio durante o treinamento, esta última não impeditiva. Salvas as exceções que não são cobertas no estudo, não existiu nenhuma restrição para a participação.

A faixa etária dos participantes ficou entre 17 e 55 anos. Com média 29,92 e mediana 28. Entre eles, 43 são do sexo masculino e 9 do sexo feminino.

O nível de experiência com Tecnologia da Informação (TI), Áudio e Música pode ser classificado nas seguintes categorias através de múltipla seleção:

1. Nenhum
2. Passatempo
3. Acadêmica
4. Profissional

Para o nível de Experiência com TI, obteve-se uma contagem de 48 respostas com nível profissional, 19 com nível acadêmico, 7 como passatempo e 2 participantes informaram não ter nenhuma das relações sugeridas. Dentre estas respostas, apenas um participante possui apenas experiência acadêmica, e outro apenas experiência como passatempo. No entanto, além dos 30 usuários que apontaram apenas experiência profissional na área, 12 apontaram possuir conhecimento acadêmico e profissional enquanto 6 afirmaram se enquadrar nas três categorias

não nulas: Passatempo, Acadêmica e Profissional. As respostas agrupadas podem ser verificadas na Figura 5.3.

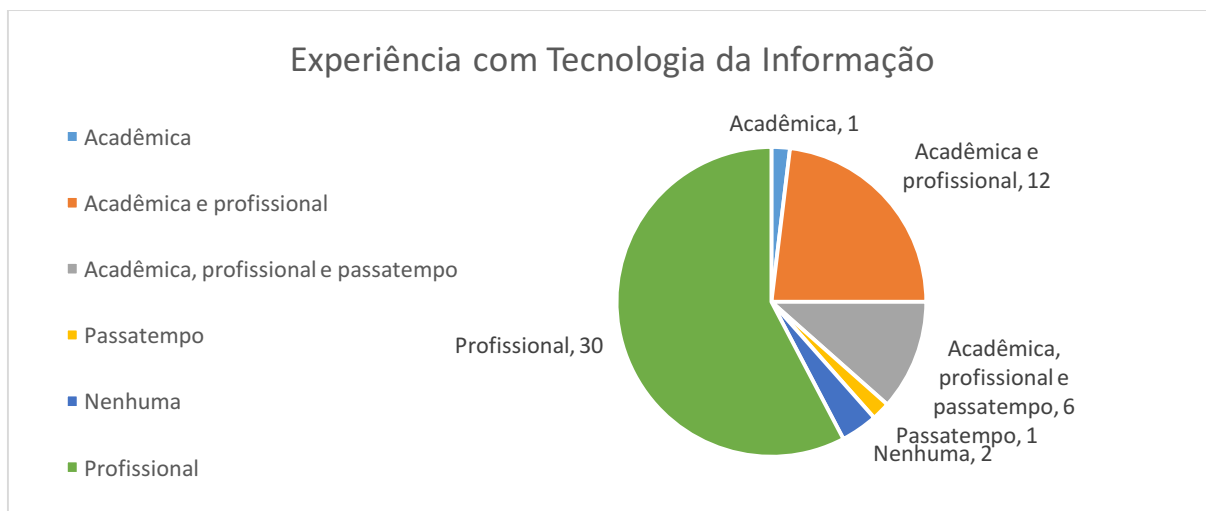


Figura 5.3 – Gráfico que demonstra a relação dos participantes com Tecnologia da Informação

Fonte: elaborado pelo próprio autor.

Utilizando o mesmo tipo de classificação, outra questão foi formulada para experiência com Áudio, e obteve-se 25 respostas para nenhuma experiência, 25 para passatempo, 2 para acadêmica e 1 para profissional. Considerando a combinação de múltiplas respostas por participante, apenas um indicou ter conhecimento na área acadêmica e profissional. Todo o restante optou apenas por uma categoria. A distribuição pode ser vista na Figura 5.4.



Figura 5.4 – Gráfico que demonstra a relação dos participantes com Áudio

Fonte: elaborado pelo próprio autor.

Outra classificação que utilizou a mesma categorização é a experiência com Música. Embora também se contempla múltipla escolha, possibilitando combinação de respostas, todas respostas foram únicas, contemplando uma soma de 37 participantes com experiência como passatempo, 3 com relação acadêmica, 1 profissional e 11 sem experiência. É possível verificar

na Figura 5.5 que a maior parte dos participantes classifica sua experiência em música como passatempo.

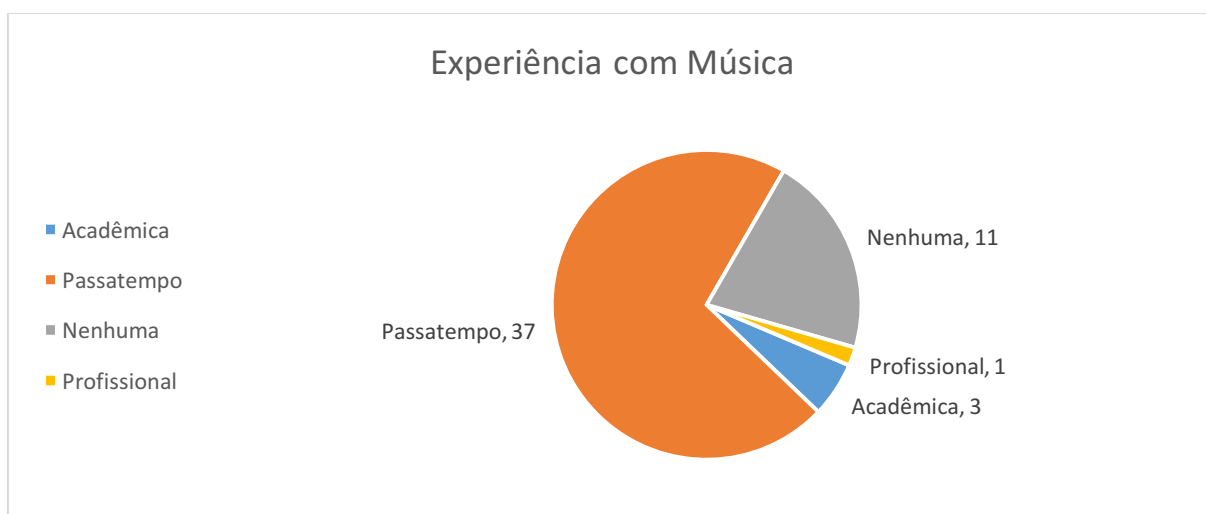


Figura 5.5 – Gráfico que demonstra a relação dos participantes com Música

Fonte: elaborado pelo próprio autor.

Para relacionar o conhecimento acadêmico apontado anteriormente, o grau de escolaridade dos participantes também foi pesquisado através de uma questão de seleção única. A questão consistiu em classificá-los em Ensino Médio, Ensino Superior – Graduação, Ensino Superior – Mestrado e Ensino Superior – Doutorado. Considerando as respostas dos participantes, 5 possuem Ensino Médio e 37 Graduação. Além disso, contou-se com a presença de 8 mestres e 2 doutores, conforme a Figura 5.6.

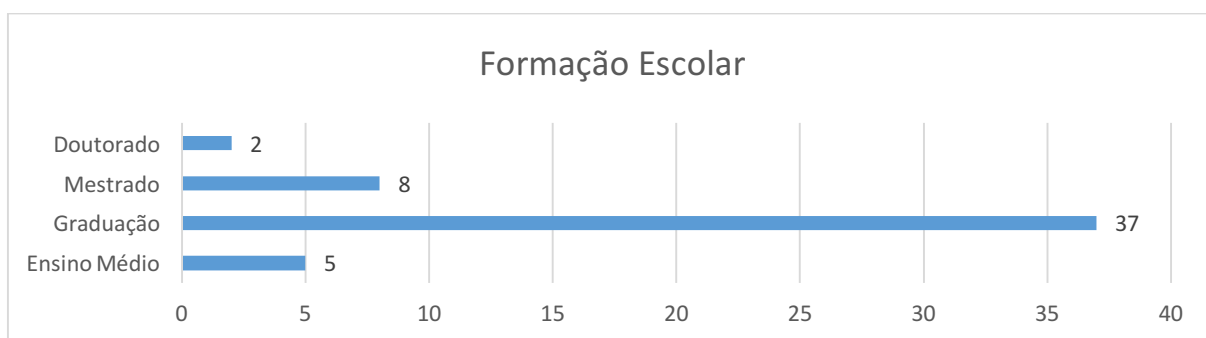


Figura 5.6 – Gráfico que exibe comparativo entre o nível escolar dos participantes

Fonte: elaborado pelo próprio autor.

5.3 Avaliação

Concluída a primeira parte do questionário que define o perfil dos participantes envolvidos, a etapa de avaliação foi realizada com a exibição das representações de sete segmentos de música de diferentes estilos aos participantes. Cada participante então, classificou cada visualização a uma determinada categoria, conforme julgamento após o treinamento.

Utilizando trechos de músicas de dez segundos retirados da coleção de amostras da *Technische Universität Dortmund* (HOMBURG *et al.* 2005), a mesma utilizada para as visualizações presentes no treinamento, as representações exibidas foram as das seguintes categorias, com as respectivas músicas, nesta ordem:

1. Eletrônica: *Effective_Now-Complete.mp3*
2. Jazz: *Jim_Crew-Neckbone.mp3*
3. Pop: *Ave_nue-Free.mp3*
4. Pop: *James_Albin_Band-Alright.mp3*
5. Rap: *Nero_Ceasar_and_Peace_of_Mind-Saul.mp3*
6. Rap: *TYCE-GONE_WIT_DAT.mp3*
7. Rock: *The_Sundogs-Oasis.mp3*

A representação da primeira faixa, *Effective_Now-Complete.mp3* da categoria eletrônica, obteve 29 respostas classificando-a corretamente. No restante, 16 respostas foram Pop, 4 Rock, 2 Jazz e 1 Rap, conforme a tabela 5.1.

Tabela 5.1 – Resultados das respostas do questionário para a representação 1.

Categoria	Total
Electronic	29
Jazz	2
Pop	16
Rap	1
Rock	4
Total	52

Fonte: Elaborado pelo próprio autor.

A segunda faixa apresentada, *Jim_Crew-Neckbone.mp3*, de Jazz, obteve 22 respostas corretas. Além destes acertos, 11 participantes a classificaram como Rap, 7 Pop, 6 Rock e 6 eletrônica.

Tabela 5.2 – Resultados das respostas do questionário para a representação 2.

Categoria	Total
Electronic	6
Jazz	22
Pop	16
Rap	11
Rock	6
Total	52

Fonte: Elaborado pelo próprio autor.

A categoria Pop teve duas faixas representadas nas questões avaliativas. A primeira, *Ave_nue-Free.mp3*, obteve uma quantidade de 12 acertos. Essa mesma quantia de respostas foi considerada pelos participantes nas categorias eletrônica e rock. Jazz recebeu 10 respostas enquanto Rap, 6.

Tabela 5.3 – Resultados das respostas do questionário para a representação 3.

Categoria	Total
Electronic	12
Jazz	10
Pop	12
Rap	6
Rock	12
Total	52

Fonte: Elaborado pelo próprio autor.

A segunda amostra representada para Pop, *James_Albin_Band-Alright.mp3*, contou com apenas 5 acertos. As respostas dos participantes foram focadas em Rap, contabilizando uma soma de 19, seguidas por 12 em Jazz, 11 em Rock e 5 em eletrônica.

Tabela 5.4 – Resultados das respostas do questionário para a representação 4.

Categoria	Total
Electronic	5
Jazz	12
Pop	5
Rap	19
Rock	11
Total	52

Fonte: Elaborado pelo próprio autor.

Além da categoria Pop, Rap também teve duas faixas na avaliação. A primeira questão, que exibiu a visualização da faixa *Nero_Ceasar_and_Peace_of_Mind-Saul.mp3*, recebeu 16 respostas corretas. 11 participantes responderam Rock, seguidos por 10 que optaram por eletrônica, 9 Pop e 6 Jazz.

Tabela 5.5 – Resultados das respostas do questionário para a representação 5.

Categoria	Total
Electronic	10
Jazz	6
Pop	9
Rap	11
Rock	16
Total	52

Fonte: Elaborado pelo próprio autor.

A segunda faixa de Rap exibida foi *TYCE-GONE_WIT_DAT.mp3*. Ela teve 8 acertos, e o restante das respostas foi distribuído da seguinte forma: 15 em eletrônica, 10 em Pop, 10 em Rock e 9 em Jazz.

Tabela 5.6 – Resultados das respostas do questionário para a representação 6.

Categoria	Total
Electronic	15
Jazz	9
Pop	10
Rap	8
Rock	10
Total	52

Fonte: Elaborado pelo próprio autor.

Por fim, a última visualização para avaliação contemplava o estilo musical Rock. O trecho de 10 segundos da faixa *The_Sundogs-Oasis.mp3* obteve 23 acertos por parte dos participantes, seguido por 15 respostas na categoria Pop, 7 em Jazz, 4 em eletrônica e 3 em Rap.

Tabela 5.7 – Resultados das respostas do questionário para a representação 7.

Categoria	Total
Electronic	4
Jazz	7
Pop	15
Rap	3
Rock	23
Total	52

Fonte: Elaborado pelo próprio autor.

Com os resultados da avaliação, que podem ser vistos na Figura 5.7, é possível ter uma visão geral sobre a quantidade de acertos em cada questão avaliativa. O conjunto de respostas para cada pergunta é apresentado com uma separação por categorias, contabilizando a soma de respostas em cada uma delas, além da soma geral.

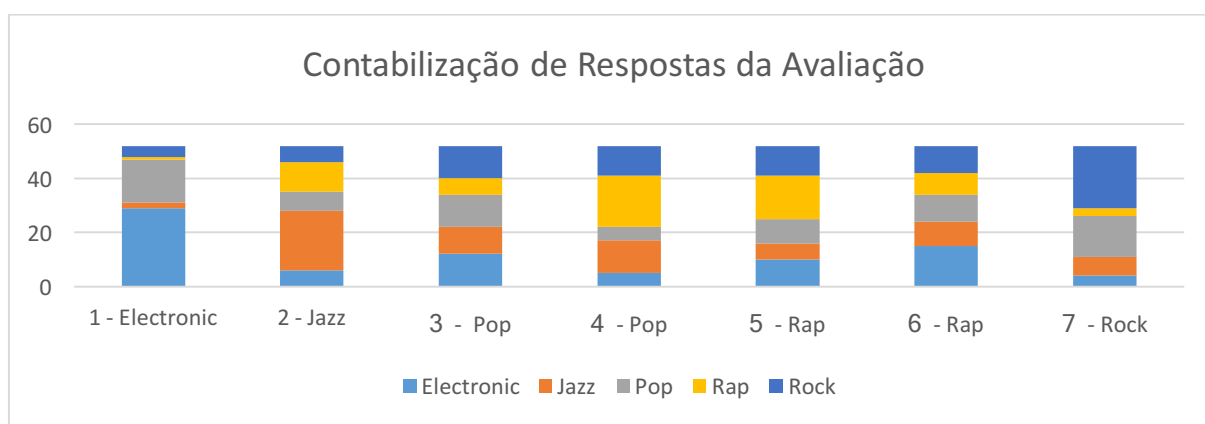


Figura 5.7 – Gráfico que soma a quantidade de respostas por questão da avaliação dos participantes

Fonte: elaborado pelo próprio autor.

5.4 Análise geral

Com objetivo de compreender melhor a assertividade do modelo, juntamente com a relação entre os acertos e os perfis dos participantes, a atual seção busca analisar os dados de forma mais detalhada, propondo diferentes visões sobre o mesmo conjunto de dados obtidos. O conhecimento adquirido sobre as pessoas que participaram da pesquisa pode apoiar a interpretação de resultados e embasar possíveis melhorias no modelo, além de proporcionar novas possibilidades para trabalhos futuros.

Verificando que nenhum dos estilos musicais obteve totalidade de acertos segundo os participantes, é possível buscar a informação da quantidade de acertos que cada participante teve nas diversas questões avaliativas. Neste quesito, houveram 7 participantes que não acertaram nenhuma das 7 questões. Com 100% de acerto, não houve nenhum participante, e apenas um classificou corretamente 6 das 7 visualizações. Houveram 11 integrantes que tiveram apenas 1 resposta correta, 11 com 2, 14 com 3, 6 com 4 e apenas 2 com 5 acertos conforme a Figura 5.8.

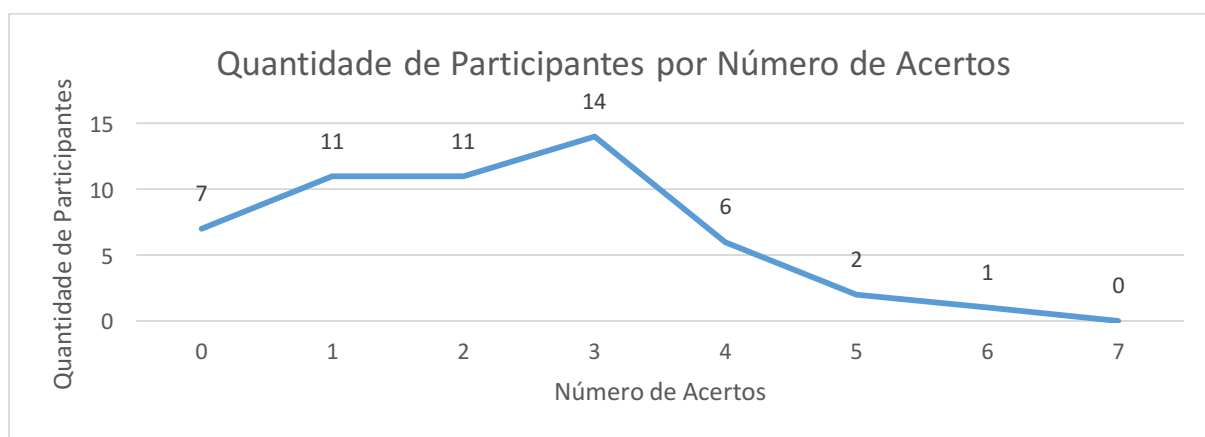


Figura 5.8 – Gráfico que mostra a soma da quantidade de participantes pela quantidade de acertos nas questões avaliativas

Fonte: elaborado pelo próprio autor.

Assim sendo, o número de acertos por usuário teve uma média de 2,21. A densidade de acertos por questão é apresentada na Figura 5.9. Assim sendo, a pesquisa contabilizou uma média de 16,42 acertos por questão, e os três estilos mais reconhecidos pelos participantes durante a execução da pesquisa foram o Eletrônico, o Rock e o Jazz.

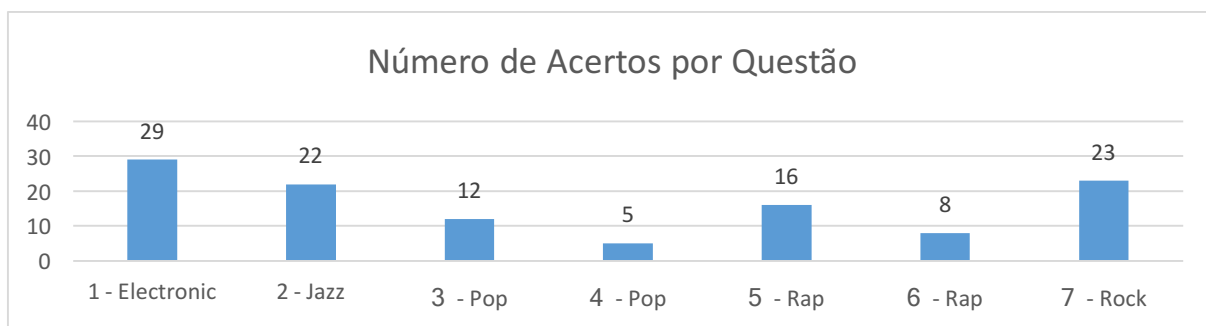


Figura 5.9 – Gráfico que soma a quantidade de respostas por questão da avaliação dos participantes

Fonte: elaborado pelo próprio autor.

A primeira questão para classificação de perfil utilizada no instrumento de pesquisa foi de idade, e uma classificação de faixas etárias foi elaborada para visualizar sua relação com acertos, tanto no resultado geral como individual por questão avaliativa, conforme exibido nas Figuras 5.10 e 5.11.



Figura 5.10 – Gráfico que exibe a média do total de acertos por faixa etária

Fonte: elaborado pelo próprio autor.

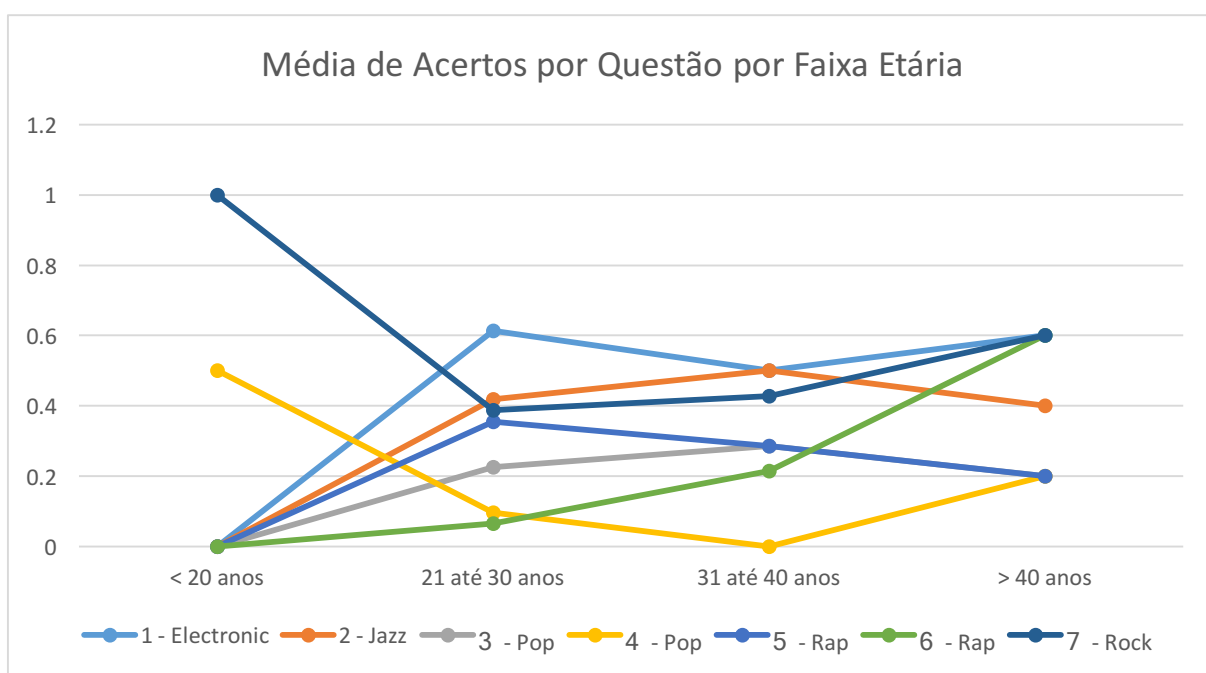


Figura 5.11 – Gráfico que exibe a média de acertos por faixa etária para cada questão avaliativa

Fonte: elaborado pelo próprio autor.

A partir dos dados coletados, é possível constatar um crescente na média geral de acertos conforme a faixa etária dos participantes avança. No entanto, avaliando a média das faixas ao separar as questões avaliativas, nota-se que as faixas 4 - Pop e 7 - Rock obtiveram uma média bastante alta na faixa etária de até 20 anos. Este valor significativo ocorreu devido a quantidade de apenas dois participantes na categoria. Também é possível verificar que a representação 1 – Electronic recebeu uma média de acertos superior ou igual a 50%, caso desconsidere-se a categoria de até 20 anos.

A relação entre experiência com Tecnologia da Informação também pode ser vinculada ao número de acertos. Dessa forma é possível averiguar se o conhecimento adquirido na área influencia o resultado da seleção do participante, e se a classificação do mesmo também faz variar a escolha das respostas. A Figura 5.12 demonstra essa relação no contexto geral da avaliação, enquanto a Figura 5.13 mostra a média de acertos por questão.

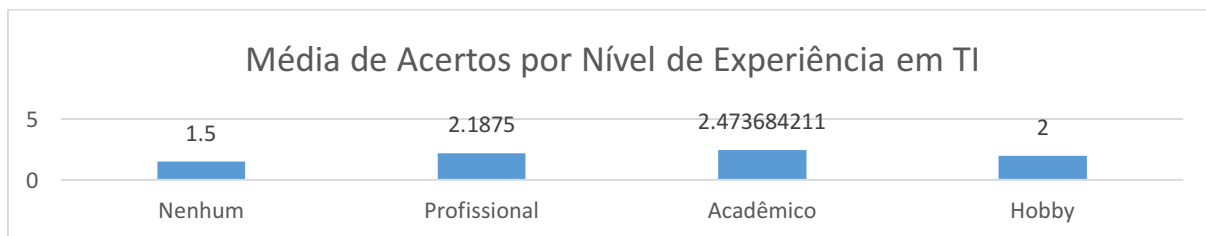


Figura 5.12 – Gráfico que exhibe a média do total de acertos por Nível de Experiência em TI

Fonte: elaborado pelo próprio autor.

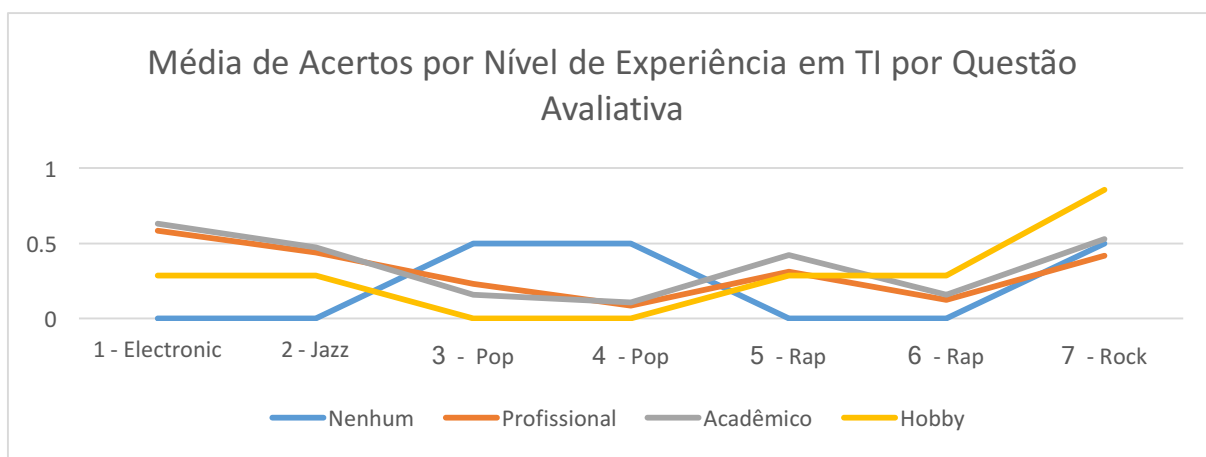


Figura 5.13 – Gráfico que exhibe a média de acertos por Nível de Experiência em TI para cada questão avaliativa

Fonte: elaborado pelo próprio autor.

Considerando que a grande maioria dos participantes possui conhecimento profissional em Tecnologia da Informação, entende-se que é relevante averiguar também as combinações de conhecimento dos participantes que tenham indicado a experiência profissional na área. Conforme a Figura 5.14, nota-se que entre os pesquisados a presença de conhecimento acadêmico influencia na atribuição correta das visualizações aos estilos de música.



Figura 5.14 – Gráfico que exibe a média do total de acertos por Nível de Experiência em TI considerando apenas participantes que apontaram experiência profissional

Fonte: elaborado pelo próprio autor.

A experiência em áudio dos participantes também é comparada ao resultado das questões avaliativas, e tem-se que o apontamento de qualquer experiência dentre os participantes está ligado a um impacto negativo na média geral, conforme visto na Figura 5.15 onde a maior média foi obtida por aqueles que não possuíam qualquer experiência em áudio. Um comparativo de média de acertos por cada uma das questões também é exibido na Figura 5.16.



Figura 5.15 – Gráfico que exibe a média do total de acertos por Nível de Experiência em Áudio

Fonte: elaborado pelo próprio autor.

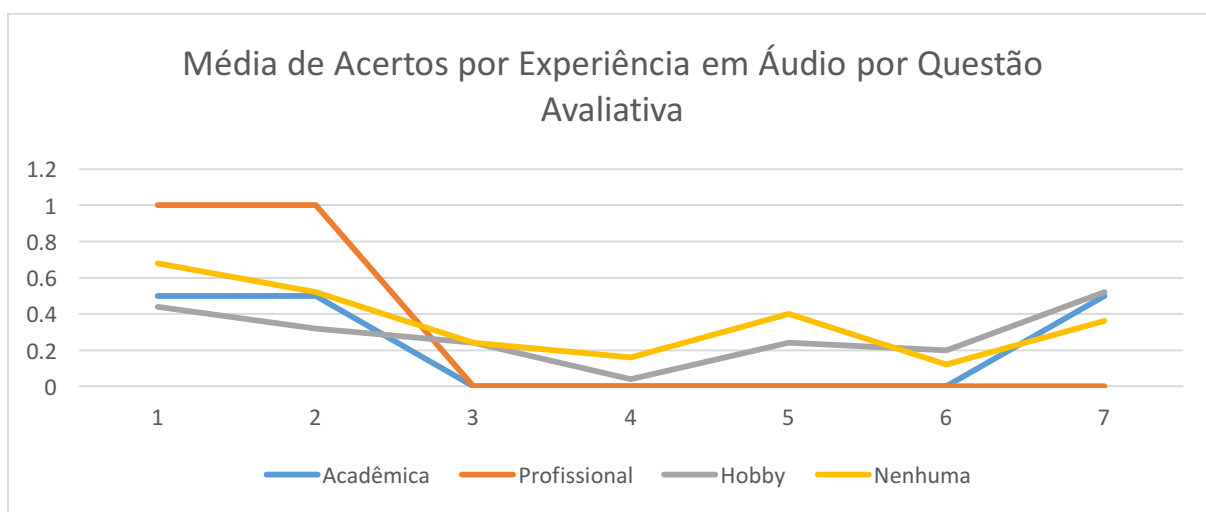


Figura 5.16 – Gráfico que exibe a média de acertos por Experiência em Áudio para cada questão avaliativa

Fonte: elaborado pelo próprio autor.

Da mesma forma, a experiência em música dos participantes foi analisada, porém contrastando ao comparativo anterior, nota-se que a experiência profissional com música obteve

uma média de acertos superior ao restante, seguido pelos participantes sem experiência musical. A média geral pode ser vista na Figura 5.17 enquanto uma abordagem que vincula a experiência à média de acertos por questão avaliativa também é exibida na Figura 5.18.

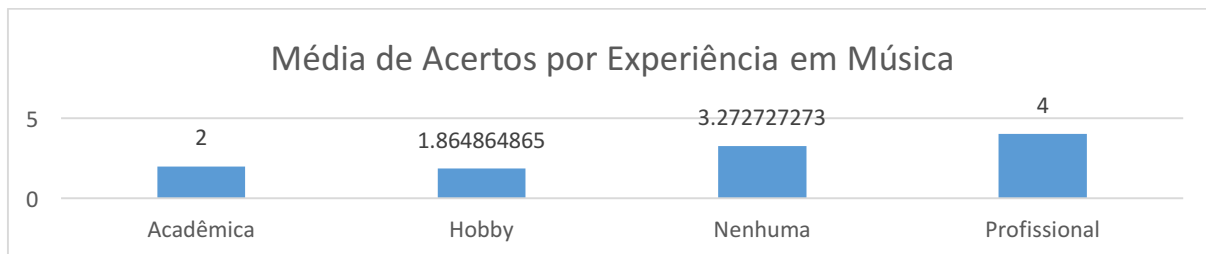


Figura 5.17 – Gráfico que exhibe a média do total de acertos por Experiência em Música
Fonte: elaborado pelo próprio autor.

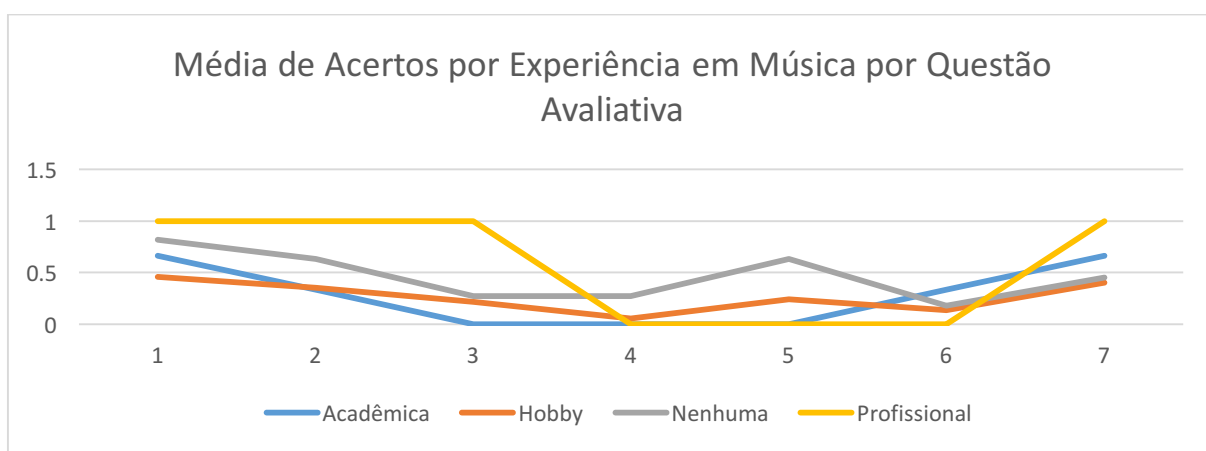


Figura 5.18 – Gráfico que exhibe a média de acertos por Experiência em Música para cada questão avaliativa
Fonte: elaborado pelo próprio autor.

Outra questão utilizada na classificação de perfil dos participantes, a formação escolar, pode ser vinculada à média de acertos para que seja verificada a relação existente entre o nível de instrução e capacidade de reconhecimento de características do modelo. A média de acertos geral pode ser verificada na Figura 5.19, enquanto a média de acertos por questão avaliativa é exibida na Figura 5.20.

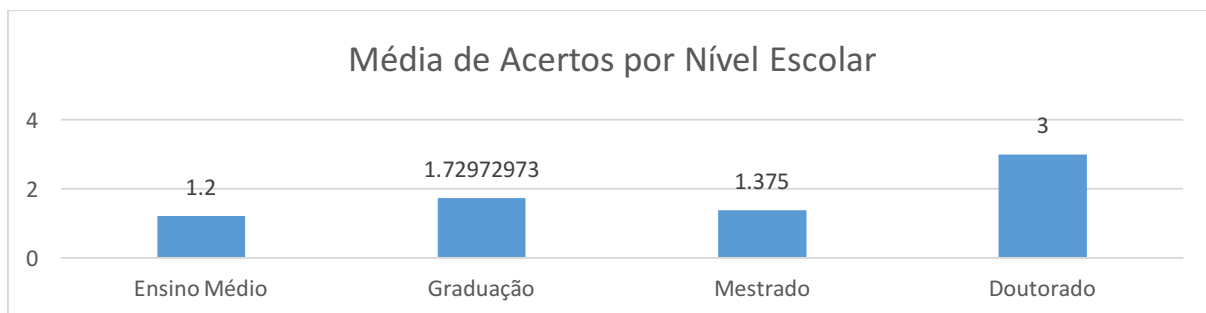


Figura 5.19 – Gráfico que exhibe a média do total de acertos por nível escolar
Fonte: elaborado pelo próprio autor.

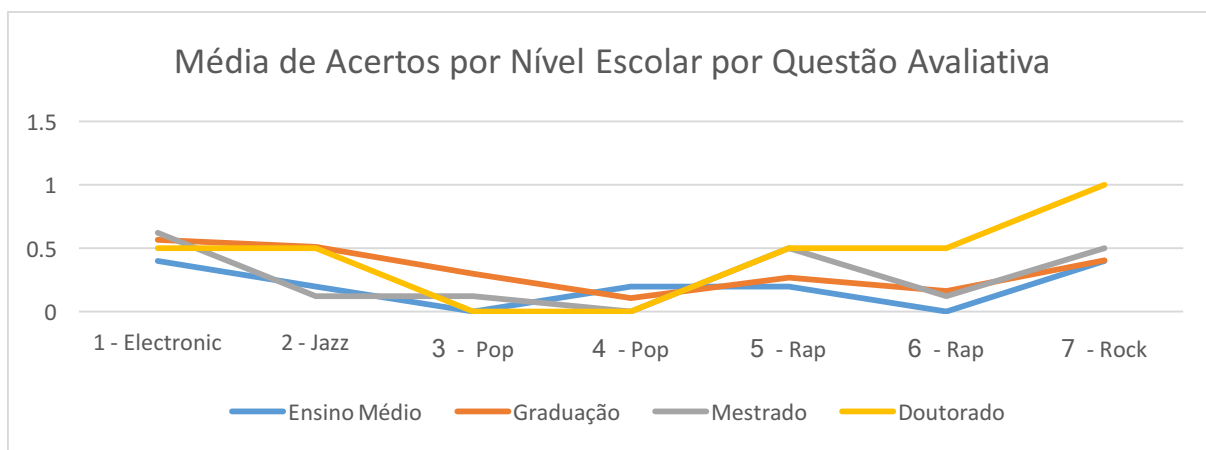


Figura 5.20 – Gráfico que exhibe a média de acertos por nível escolar para cada questão avaliativa

Fonte: elaborado pelo próprio autor.

Dessa forma, por meio dos dados obtidos sobre o perfil dos usuários e suas respostas nas questões avaliativas, é possível estabelecer de forma geral algumas relações entre perfil e resultados. Assim, segundo o estudo traçado, dentro do grupo de participantes da pesquisa em questão, a faixa etária está ligada à média de acertos, de forma que os participantes com idade mais avançada obtiveram maior quantidade de acertos.

O nível de experiência com tecnologia da informação se mostrou importante, principalmente no âmbito acadêmico, dado que a média de acertos dos participantes se mostrou superior conforme o vínculo de conhecimento e experiência era existente e se relacionava à academia. A ausência de vínculo com TI obteve a menor média de todas, indicando dentro do contexto que experiência nesta área, embora não requerida é um diferencial.

Ao contrário do que se viu para tecnologia da informação, os participantes sem experiência alguma em áudio obtiveram a maior média de acertos. E de forma diferente destes dois cenários, a experiência em música, quando relacionada a média de acertos não resultou em conclusão clara a respeito da experiência, pois a experiência profissional obteve maior média geral, seguida pela ausência de experiência, acadêmica e passatempo.

O nível escolar, no entanto, indicou um avanço na média geral não uniforme conforme aumentava, tendo a menor média o público que possui apenas ensino médio, seguido por mestrado e graduação, enquanto doutorado obteve a maior média geral.

Assim, o estudo sobre os resultados deste questionário, dentro do contexto de participantes no qual foi aplicado, traça algumas relações entre características de perfil com a assimilação da visualização, e o trabalho é finalizado com as considerações finais a seguir.

6 CONSIDERAÇÕES FINAIS

A proposta de visualização de áudio foi implementada e avaliada, e o presente capítulo visa analisar o trabalho como um todo.

Neste sentido, o estudo verificou a importância da visualização de dados em contextos com grandes volumes de informação. Assim, o primeiro capítulo abordou os conceitos para definição de *pipelines* de visualização, o processo que envolve a análise e entendimento dos dados por parte do usuário, a relação entre os tipos de dados e tipos de gráficos, além de citar trabalhos relacionados a visualização de dados e áudio. Um modelo de visualização também visto é o *Streamgraph*, que combina estética e geometria ao representar conjuntos temporais de dados. Assim, este modelo e os conceitos de *pipeline* abordados no capítulo foram vastamente utilizados na definição da proposta.

A análise de áudio é abordada no segundo capítulo, inicialmente pela transformação de áudio analógico em digital. Após compreender como o áudio é, tanto fisicamente como evento quanto digitalmente como conjunto de dados, inicia-se uma verificação das propriedades que podem ser extraídas de segmentos digitais. Além disso, verifica-se como processamentos podem ser feitos sobre propriedades, para que características relevantes ao contexto do trabalho estejam disponíveis e possam ser usadas no desenvolvimento da visualização. Dessa forma, o capítulo agrega ao trabalho com o entendimento de análises, entre elas a MFCC, que passou a ser utilizada no desenvolvimento da proposta.

Com base no estudo de visualização e análise de áudio, no terceiro capítulo uma proposta é montada e fornecida juntamente a um *pipeline* que busca gerar representações visuais de segmentos de áudio. Esta proposta reúne e utiliza os *Streamgraphs* vistos no capítulo um, juntamente com as análises de áudio abordadas no capítulo dois.

A implementação do protótipo foi realizada e descrita no capítulo quatro, onde cinco requisitos foram definidos para guiar o desenvolvimento. Para cumprir tais premissas, escolheu-se um conjunto de ferramentas, que aplicadas de acordo com a proposta de implementação, retomaram conceitos vistos anteriormente e realizaram a extração de características de áudio e exibição das mesmas visualmente na forma de *Radial Streamgraphs*.

O quinto capítulo retrata uma abordagem para verificação da assertividade do modelo baseado no perfil de participantes. Por meio de uma pesquisa realizada após uma explicação e treinamento sobre a visualização implementada, os participantes foram convidados a classificar visualizações geradas a partir de segmentos de música ao estilo musical de cada uma delas com base apenas em imagens, sem som.

Deste modo, o objetivo geral do trabalho foi alcançado, já que um modelo de representação visual de áudio foi proposto e desenvolvido a partir do estudo conjunto de técnicas de visualização e interpretação de sinais de áudio. Constatou-se, no entanto, que as visualizações geradas ainda não apresentam uma fidelidade representativa que as relacione com os segmentos de áudio que as geram a ponto de tornar a identificação de sons possível sem reprodução audível.

Contudo a avaliação realizada nesse trabalho não objetivava a comparação entre técnicas já utilizadas para visualização de áudio, não sendo assim possível avaliar se a técnica proposta possui uma representação visual superior às demais já existentes, nesta situação a seção de trabalhos futuros, do presente capítulo, pontua um novo estudo nesse contexto.

Quanto aos objetivos específicos, o estudo dos principais formatos digitais de áudio não precisou ser efetuado, pois conforme visto na seção 2.3 as análises espectrais utilizam o formato PCM como base para processamento, e a partir dele que temos a extração de características como espectro de amplitude, espectro de potência e MFCC, que foram utilizadas no desenvolvimento da visualização.

Outro objetivo específico visava apresentar uma abordagem de representação visual que utilizasse os conceitos estudados, e conforme visto no capítulo 3, este foi atendido plenamente, dado que um *pipeline* de visualização e uma proposta de desenvolvimento foram elaborados e fornecidos para implementação.

O desenvolvimento do protótipo, relatado no capítulo 4 atende ao terceiro objetivo específico, que era desenvolver uma representação visual através da análise de áudio. Assim, com base em todas as guias do capítulo 3, foi possível elaborar a visualização que utiliza dados de áudio para gerar gráficos do tipo *Radial Streamgraphs*.

O último objetivo específico, validar os resultados obtidos durante os experimentos, foi atendido no capítulo 5, através da pesquisa efetuada com 52 participantes onde o perfil deles foi relacionado com o número de acertos na classificação das visualizações, e verificou-se então que a faixa etária dos participantes, além da experiência dos mesmos com Tecnologia da Informação possui relação dentro do contexto de participantes com o entendimento da visualização.

6.1 Contribuições e Aprendizado

No decorrer deste trabalho as seguintes contribuições foram obtidas:

- O *pipeline* de visualização, a proposta de desenvolvimento e os requisitos utilizados no protótipo podem ser utilizados em outros contextos de visualização de áudio ou dados como ponto de partida.
- Um resumo que aborda a visualização de áudio, verificando as etapas envolvidas na sua geração foi submetido e apresentado na Feira de Iniciação Científica INOVAMUNDI em 2016.
- Um artigo sobre visualização de áudio, que trata dos aspectos de visualização envolvidos na concepção e estruturação e propõe um pipeline para gerar visualizações semelhantes ao *Flickr Flow* (VIÉGAS; WATTENBERG, 2016) foi submetido para a revista *Colloquium Exactarum* em 2016 conforme Anexo 1 e aguarda avaliação.
- A modularização, representada pelo RQ3 da seção 4.1 torna possível a prototipação de variações de visualizações, sejam referentes a áudio, sinais ou qualquer outro dado, assim como com representações que estejam ou não relacionadas ao objetivo do trabalho.
- A biblioteca D3 apresentou uma enorme flexibilidade para visualização, tornando possível a implementação do *Radial Streamgraph*, sendo assim recomendada para representações que abordem exibição estática como a proposta no atual trabalho.
- Notou-se que com os navegadores atuais de internet, através do uso da biblioteca Meyda para análise de áudio, a extração de características como a MFCC até mesmo em tempo real é uma realidade acessível.

Assim, entende-se que o presente trabalho agrega não apenas com a implementação capaz de gerar visualizações de áudio, mas também como um conjunto de recursos que foram estudados e fornecidos que poderão ser utilizados em trabalhos futuros, conforme sugestões na seguinte seção.

6.2 Trabalhos futuros

No que se refere a trabalhos futuros, considera-se importante:

- Comparar a representação gerada com outras ferramentas de visualização de áudio tradicionais, com objetivo de ter um melhor entendimento sobre a interpretação das visualizações. Podem ser consideradas além das citadas no

capítulo 1, representações utilizadas por profissionais de áudio, como formato de onda, histograma e outros.

- Propor ou implementar uma solução definitiva para a questão de sobreposição na visualização, citada no capítulo 4, dado que a atual abordagem foi capaz de reduzir, porém não eliminar a exibição de uma camada sobre outra.
- Apresentar o conceito de *Radial Streamgraph* para as comunidades envolvidas em âmbito acadêmico ou de visualização de dados, através de publicações e uma possível biblioteca, conforme o próximo item.
- Publicar uma biblioteca que atue como *plugin* para o D3 que facilite a geração de *Radial Streamgraphs*.
- Utilizar outras análises espectrais de áudio como base para a geração das representações. Já que o atual estudo focou na utilização da MFCC, entende-se que outras características, sejam de mais alto ou baixo nível podem ser utilizadas, e uma sugestão deixada é que o espectro de amplitude seja utilizado e adaptado a escala Mel de frequência, tornando-se assim mais próximo dos níveis de escala da audição humana, mas sem processamento adicional que utiliza a análise inversa de Fourier e a função senoidal que resultou em valores negativos.

REFERÊNCIAS BIBLIOGRÁFICAS

ALENCAR, Aretha Barbosa. **Mineração e visualização de coleções de séries temporais**. 2007. Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação.

AZZARO, Gilles. **Gilles AZZARO - Sculpteur de Voix**. Disponível em: <<http://www.gillesazzaro.com/>>. Acesso em 31 abr. 2016.

BERGLAND, G. D. **A guided tour of the fast Fourier transform**. IEEE spectrum, v. 6, n. 7, p. 41-52, 1969.

BIANCHI, André Jucovsky. **Processamento de áudio em tempo real em dispositivos computacionais de alta disponibilidade e baixo custo**. 2014. Tese de Doutorado. Universidade de São Paulo.

BLACK, Harold S.; EDSON, J. O. **Pulse code modulation**. Transactions of the American Institute of Electrical Engineers, v. 1, n. 66, p. 895-899, 1947.

BOSTOCK, Michael. **D3.js**. Data Driven Documents, 2012.

BOTELHO, Elisângela. **Visualização de dados como ferramenta de Classificação em sistemas de Bases de Dados para Data Mining**. 2002. Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação.

BOTTENTUIT JUNIOR, João Batista; LISBOA, Eliana Santana; COUTINHO, Clara Pereira. **O infográfico e as suas potencialidades educacionais**. 2011.

BRANCO, Vinícius Marques Alves. **Visualização como suporte à exploração de uma base de dados pluviométricos**. 2003. Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação.

BYRON, Lee; WATTENBERG, Martin. **Stacked graphs—geometry & aesthetics**. IEEE transactions on visualization and computer graphics, v. 14, n. 6, p. 1245-1252, 2008.

CANNAM, Chris; LANDONE, Christian; SANDLER, Mark. **Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files**. In: Proceedings of the 18th ACM international conference on Multimedia. ACM, 2010. p. 1467-1468.

CHAN, Winnie Wing-Yi; QU, Huamin. **A report on musical structure visualization**. Leonardo, Department of Computer Science and Engineering Hong Kong University of Science and Technology Clear Water Bay, Kowloon, Hong Kong, 2007.

CHEN, Qing et al. **Analysis of Music Representations of Vocal Performance Based on Spectrogram**. In: Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on. IEEE, 2010. p. 1-4.

D3. **Data-Driven Documents**. Disponível em: <<https://d3js.org/>>. Acessado em 4 nov. 2016.

EHLERS, Ricardo S. **Análise de séries temporais**. Laboratório de Estatística e Geoinformação. Universidade Federal do Paraná, 2007.

FERNANDES, Tales Gouveia; PANAZIO, Aline Neves. **DO ANALÓGICO AO DIGITAL: AMOSTRAGEM, QUANTIZAÇÃO E CODIFICAÇÃO**. 2009.

FISCHER, Andreas Nicolas. **Reflection II**. Disponível em: <<http://anf.nu/reflection-ii>>. Acesso em: 23 mai. 2016.

FRAYLING, Christopher. **Research in art and design**. 1993.

FREITAS, Carla Maria Dal Sasso et al. **Introdução à visualização de informações**. Revista de informática teórica e aplicada. Porto Alegre. Vol. 8, n. 2 (out. 2001), p. 143-158, 2001.

GIANNELLA, J. R. **Dispositivo infovis: interfaces entre visualização da informação, infografia e interatividade em sites jornalísticos**. 2014. 189 f. 2014. Tese de Doutorado. Dissertação (Mestrado em Comunicação)–Escola de Comunicações e Artes, Universidade de São Paulo.

GITHUB. **Github**. How people build software. Disponível em <<https://github.com/hughrawlinson/meyda/commit/a3113a1c6ed2f908e6cde38575064f9aafe2cfd2#diff-22d5d2b57c69d2e1ac054c4d0f744ff1>>. Acesso em 4 nov. 2016.

GITHUT. **Github**. Disponível em <<http://githut.info/>>. Acesso em 4 nov. 2016.

GRAY, Carole; MALINS, Julian. **Visualizing research: A guide to the research process in art and design**. Burlington, VT: Ashgate, 2004.

GREEN, D. A. **A colour scheme for the display of astronomical intensity images**. arXiv preprint arXiv:1108.5083, 2011.

HANSEN, Colin H.; SEHRNDT, C. H. **Fundamentals of acoustics**. Occupational Exposure to Noise: Evaluation, Prevention and Control. World Health Organization, 2001.

HOLLIMAN, Nick; WATSON, Paul. **Scalable Real-Time Visualization Using the Cloud**. Cloud Computing, IEEE, v. 2, n. 6, p. 90-96, 2015.

HOMBURG, Helge et al. **A Benchmark Dataset for Audio Classification and Clustering**. In: ISMIR. 2005. p. 528-31.

HORN, Robert E. **Information design: Emergence of a new profession**. Information design, p. 15-33, 1999.

LOGAN, Beth et al. **Mel Frequency Cepstral Coefficients for Music Modeling**. In: ISMIR. 2000.

MALINOWSKI, Stephen; TURETSKY, L. **Music animation machine**. Disponível em: <<http://www.musanim.com/>>. Acesso em 31 abr. 2016.

MCFEE, Brian et al. **librosa: Audio and music signal analysis in python**. In: Proceedings of the 14th Python in Science Conference. 2015.

MCKAY, C. 2010. **Automatic music classification with jMIR**. Ph.D. Thesis. McGill University, Canada.

MOFFAT, David; RONAN, David; REISS, Joshua D. **AN EVALUATION OF AUDIO FEATURE EXTRACTION TOOLBOXES**. 2015.

MORELAND, Kenneth. **A survey of visualization pipelines**. Visualization and Computer Graphics, IEEE Transactions on, v. 19, n. 3, p. 367-378, 2013.

MOZILLA. **Mozilla**. Disponível em <<https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino/History>>. Acesso em 4 nov. 2016.

NEVES, Samuel Clayton Maciel; PELAES, Evaldo Gonçalves; Laboratório de Processamento de Sinais. **Estudo e implementação de técnicas de segmentação de imagens**. Revista Virtual de Iniciação Acadêmica da UFPA-Universidade Federal do Pará-Departamento de Engenharia Elétrica e de Computação, v. 1, n. 2, 2008.

NPM. **Node Package manager**. Installs, publishes and manages node programs. Disponível em <<https://www.npmjs.com/>>. Acesso em 12 nov. 2016.

NYQUIST, Harry. **Certain topics in telegraph transmission theory**. 1928.

ONO, Jorge Henrique Piazzentin. **Visualização de similaridades em bases de dados de música**. Tese de Doutorado. Universidade de São Paulo.

ORIO, Nicola. **Music retrieval: A tutorial and review**. now publishers Inc, 2006.

OSGOOD, Brad. **The Fourier transform and its applications**. Lecture Notes for EE, 2007.

PAULUS, Jouni; MÜLLER, Meinard; KLAPURI, Anssi. **State of the Art Report: Audio-Based Music Structure Analysis**. In: ISMIR. 2010. p. 625-636.

PRODANOV, Cleber Cristiano; DE FREITAS, Ernani Cesar. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico – 2ª Edição**. Editora Feevale, 2013.

RAMDA. **Ramda**. A practical functional library for Javascript programmers. Disponível em <<http://ramdajs.com/>>. Acesso em 12 nov. 2016.

RAWLINSON, Hugh; SEGAL, Nevo; FIALA, Jakub. **Meyda: an audio feature extraction library for the Web Audio API**. In: Proceedings of the 2015 Web Audio Conference. 2015.

SEVERANCE, Charles. **Javascript: Designing a language in 10 days**. Computer, v. 2, n. 45, p. 7-8, 2012.

SHANNON, Claude Elwood. **Communication in the presence of noise**. Proceedings of the IRE, v. 37, n. 1, p. 10-21, 1949.

SHNEIDERMAN, Ben. **The eyes have it: A task by data type taxonomy for information visualizations.** In: Visual Languages, 1996. Proceedings., IEEE Symposium on. IEEE, 1996. p. 336-343.

SMITH, Stephen M.; WILLIAMS, Glen N. **A visualization of music.** In: Visualization'97., Proceedings. IEEE, 1997. p. 499-503.

SNYDAL, Jon; HEARST, Marti. **Improviz: visual explorations of jazz improvisations.** In: CHI'05 extended abstracts on Human factors in computing systems. ACM, 2005. p. 1805-1808.

TIOBE. **TIOBE.** Disponível em <<http://www.tiobe.com/tiobe-index/>>. Acesso em 4 nov. 2016.

TUFTE, Edward R. **The visual display of quantitative information.** 1986.

TZANETAKIS, George. **Music data mining.** CRC Press, 2011.

VAN MELLE, Gerbrand; MARKS, Stefan. **Frozen waves: exploring the transformation between sound and object.** 2014.

VIÉGAS, Fernanda; WATTENBERG, Martin. **Flickr flow.** HINT. FM. Disponível em <<http://hint.fm/projects/flickr/>>. Acesso em 23 mai. 2016.

VIÉGAS, Fernanda. **TEDxSP 2009 - Fernanda Viegas.** Disponível em: <https://www.youtube.com/watch?v=DUV_oC_Rvy4>. Acesso em 23 mai. 2016. 2009.

WURMAN, Richard Saul. **Ansiedade de informação.** Cultura Editores Associados, 1991.

ZÖLZER, Udo. **Digital audio signal processing.** John Wiley & Sons, 2008.

ANEXO 1 – E-MAIL DE CONFIRMAÇÃO DA SUBMISSÃO DE ARTIGO PARA A REVISTA COLLOQUIUM EXACTARUM



Gregório Kusowski <gregorio.kusowski@gmail.com>

[CE] Agradecimento pela Submissão

Robson Augusto Siscoutto <robson@unoeste.br>
To: Gregório Chalinski Kusowski <gregorio.kusowski@gmail.com>

Wed, Jun 1, 2016 at 11:51 PM

Gregório Chalinski Kusowski,

Agradecemos a submissão do seu manuscrito "VISUALIZAÇÃO DE DADOS: UMA PROPOSTA DE PIPELINE PARA VISUALIZAÇÃO DE ÁUDIO" para Colloquium Exactarum. Através da interface de administração do sistema, utilizado para a submissão, será possível acompanhar o progresso do documento dentro do processo editorial, bastando logar no sistema localizado em:

URL do Manuscrito:

<http://revistas.unoeste.br/revistas/ojs/index.php/ce/author/submission/1529>

Login: kusowski

Em caso de dúvidas, envie suas questões para este email. Agradecemos mais uma vez considerar nossa revista como meio de transmitir ao público seu trabalho.

Robson Augusto Siscoutto
Colloquium Exactarum

Colloquium Exactarum

<http://revistas.unoeste.br/revistas/ojs/index.php/ce>

APÊNDICE 1 – APRESENTAÇÃO DA PESQUISA DE AVALIAÇÃO E TREINAMENTO SOBRE O MODELO PROPOSTO

Representação Visual de Segmentos de Áudio

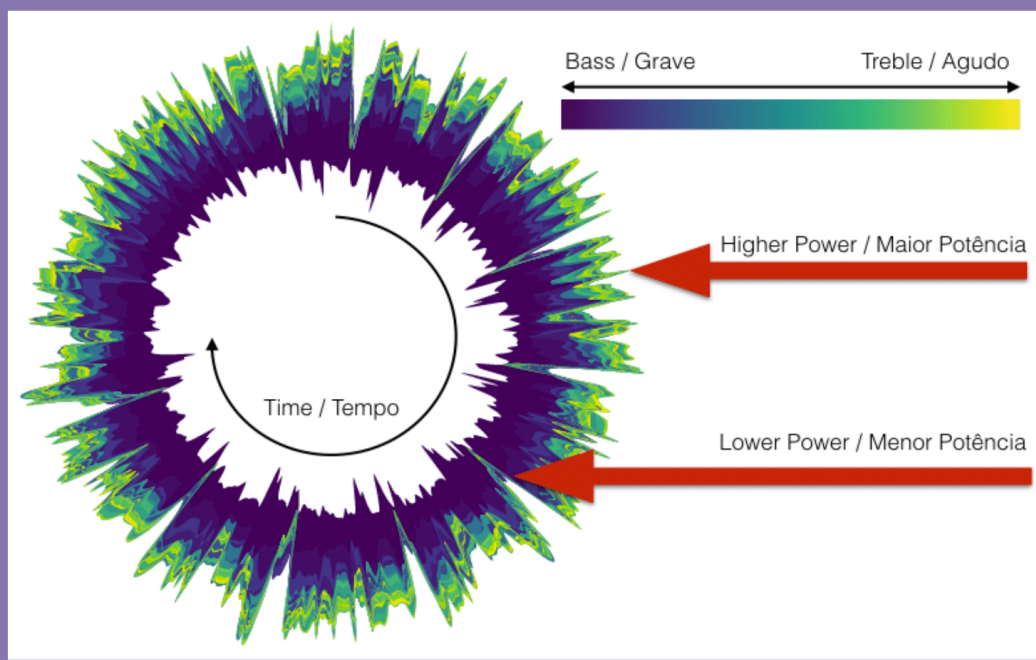
Pesquisa para Trabalho de Conclusão
Gregório Chalinski Kusowski - Universidade Feevale

Como parte do meu trabalho de conclusão, a atual pesquisa busca avaliar um modelo criado para representar visualmente informações de um dado segmento de áudio. O trabalho como um todo, tem como objetivo principal investigar um modelo de representação visual de áudio a partir da interpretação de sinais e técnicas de visualização. Assim sendo, a pesquisa consiste nos seguintes itens e dura entre 5 e 10 minutos:

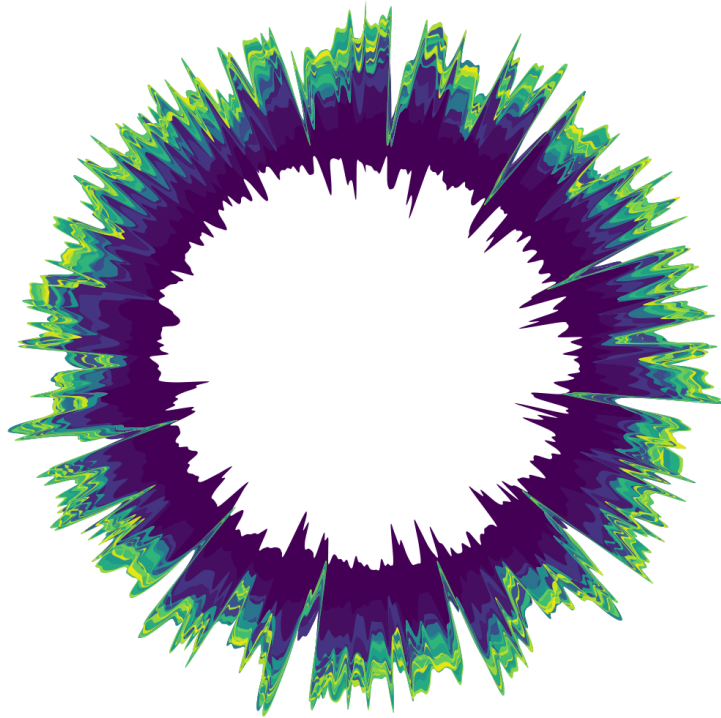
- Explicação - É explicado como o modelo é gerado e como está relacionado com o áudio.
- Treinamento - 5 segmentos de áudios de 10 segundos cada ficam disponíveis, e podem ser ouvidos e vistos através do modelo proposto. Cada segmento é um pedaço de uma música de um estilo musical diferente.
- Qualificação - Seus dados/perfil, contendo questões rápidas sobre você.
- Questões finais - 7 representações visuais de músicas são exibidas, e para cada uma delas, você irá classificar em qual gênero musical ela está inserida, sem ouvir o áudio.

Dado o uso da ferramenta de pesquisa, você iniciará pelo treinamento. Os dados aqui coletados não serão expostos publicamente.

O modelo utilizado é apresentado em formato radial, conforme a imagem, onde as cores representam a frequência (mais grave ou agudo), o tamanho a potência, e o ângulo de cada segmento o momento em que o áudio é analisado. Para conhecer os estilos, navegue abaixo em cada um deles e pressione play para uma visualização interativa. Após conhecer os 5 estilos apresentados, você pode acessar a pesquisa através do link no fim da página.

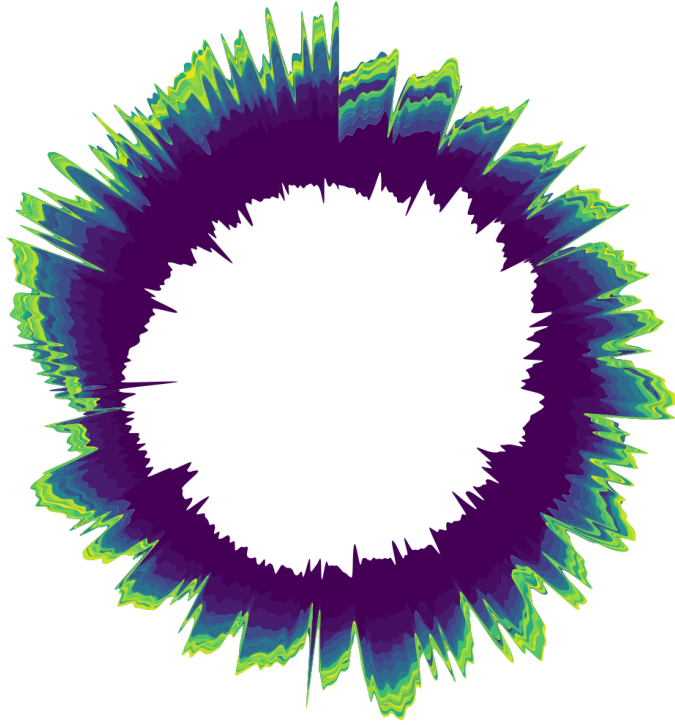


PLAY



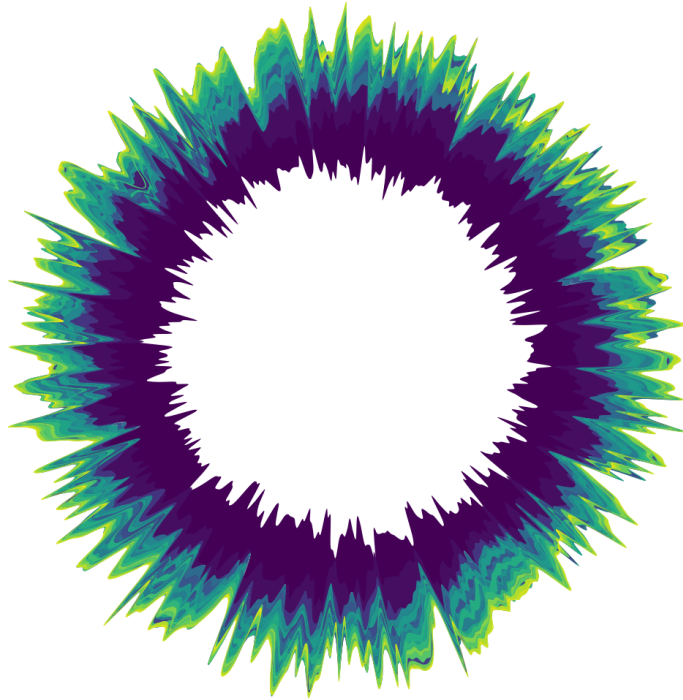
ELECTRONIC

PLAY



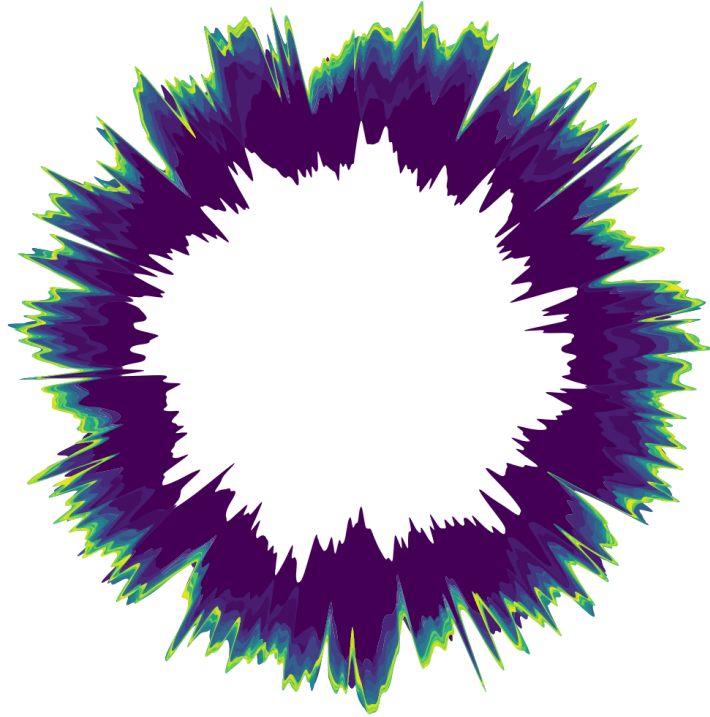
JAZZ

PLAY



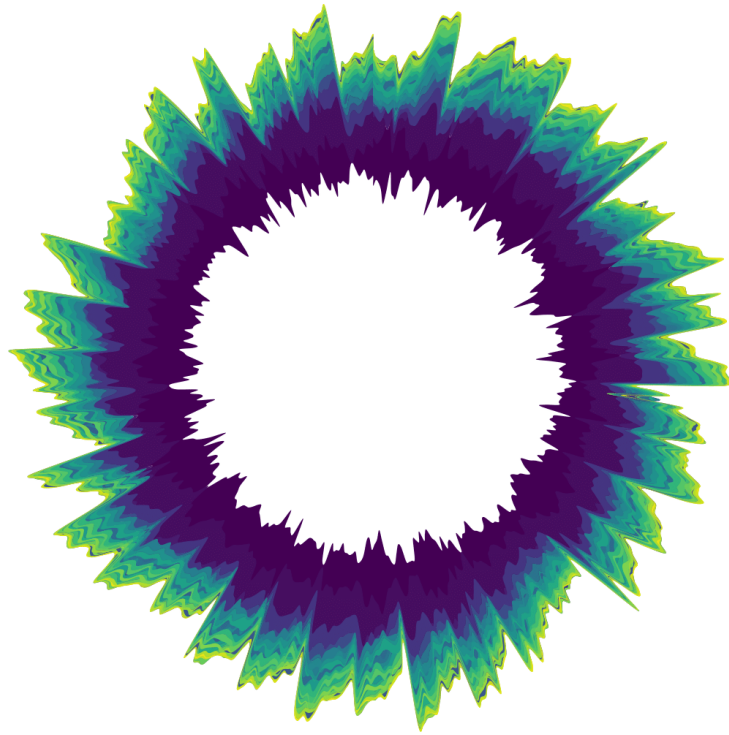
POP

PLAY



RAP

PLAY



ROCK

[Clique aqui para acessar a pesquisa.](#)

APÊNDICE 2 – QUESTIONÁRIO PARA AVALIAÇÃO DO MODELO

Visual Representation of Audio Segments / Representação Visual de Segmentos de Áudio

About you / Sobre você

***Obrigatório**

1. Age / Idade *

.....

2. Gender / Gênero *

Marcar apenas uma oval.

☐ Female / Feminino

☐ Male / Masculino

☐ Other / Outro

3. Experience with IT / Experiência com TI

Marque todas que se aplicam.

☐ None / Nenhuma

☐ Academic / Acadêmica

☐ Professional / Profissional

☐ Hobby / Passatempo

4. Experience with Audio / Experiência com Áudio

Marque todas que se aplicam.

☐ None / Nenhuma

☐ Academic / Acadêmica

☐ Professional / Profissional

☐ Hobby / Passatempo

5. Experience with Música / Experiência com Música

Marque todas que se aplicam.

☐ None / Nenhuma

☐ Academic / Acadêmica

☐ Professional / Profissional

☐ Hobby / Passatempo

6. Academic Formation / Formação Escolar

Marcar apenas uma oval.

- ☐ Basic School / Ensino Fundamental
- ☐ High School / Ensino Médio
- ☐ Bachelor Degree / Ensino Superior - Graduação
- ☐ Master Degree / Ensino Superior - Mestrado
- ☐ Doctorate Degree / Ensino Superior - Doutorado

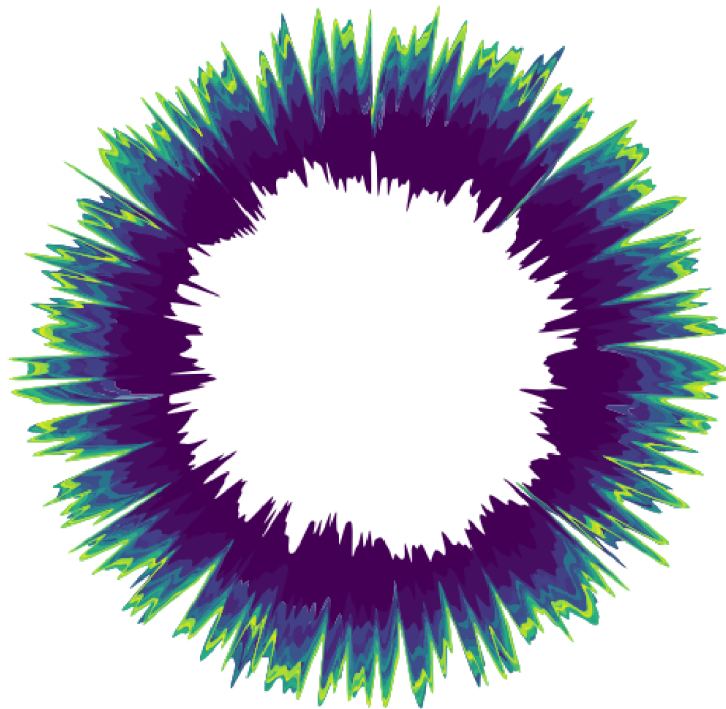
7. You e-mail if you want to get feedback from the survey results / Disponibilize seu e-mail se desejar receber feedback da pesquisa

.....

Final questions / Questões finais

Please select the music genre that fits with the correspondent image / Por favor escolha o gênero musical de acordo com a imagem exibida

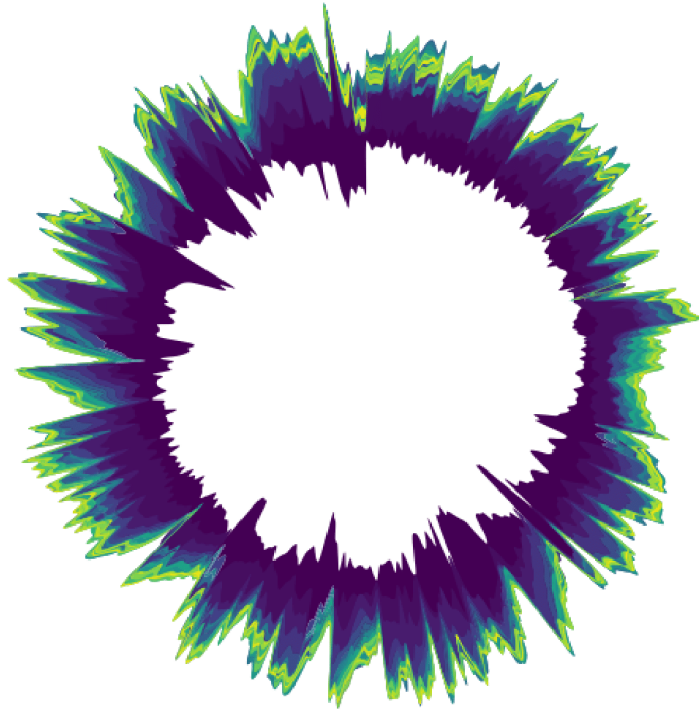
8. 1 *



Marcar apenas uma oval.

- ☐ Electronic
- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Rap

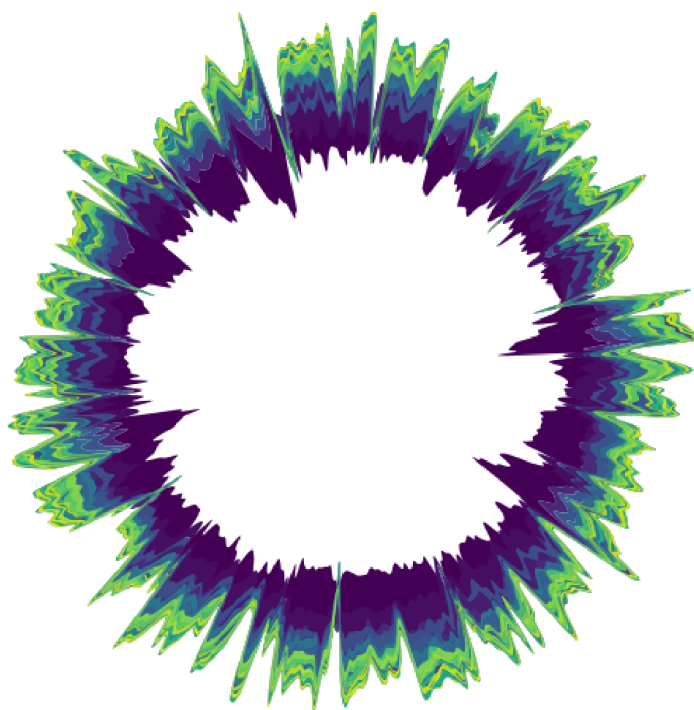
9.2 *



Marcar apenas uma oval.

- ☐ Electronic
- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Rap

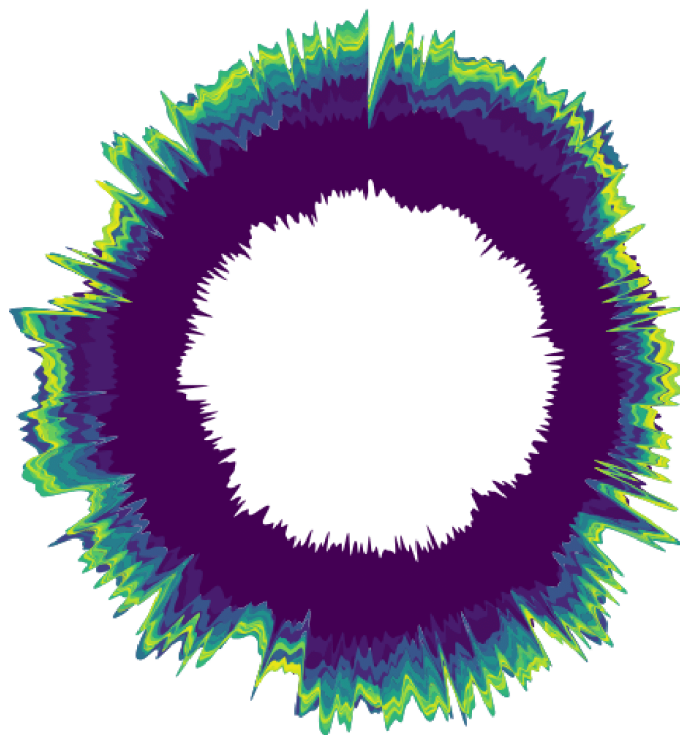
10.3 *



Marcar apenas uma oval.

- ☐ Electronic
- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Rap

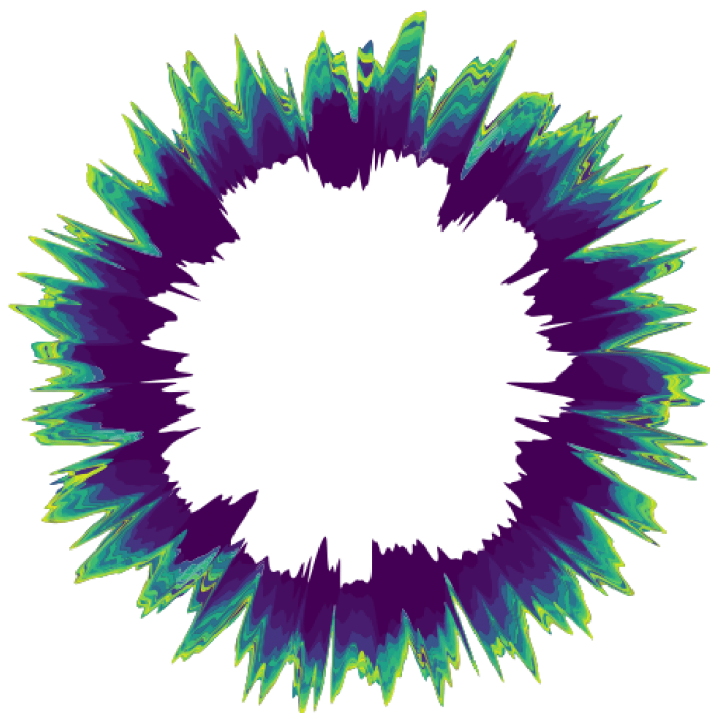
11.4 *



Marcar apenas uma oval.

- ☐ Electronic
- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Rap

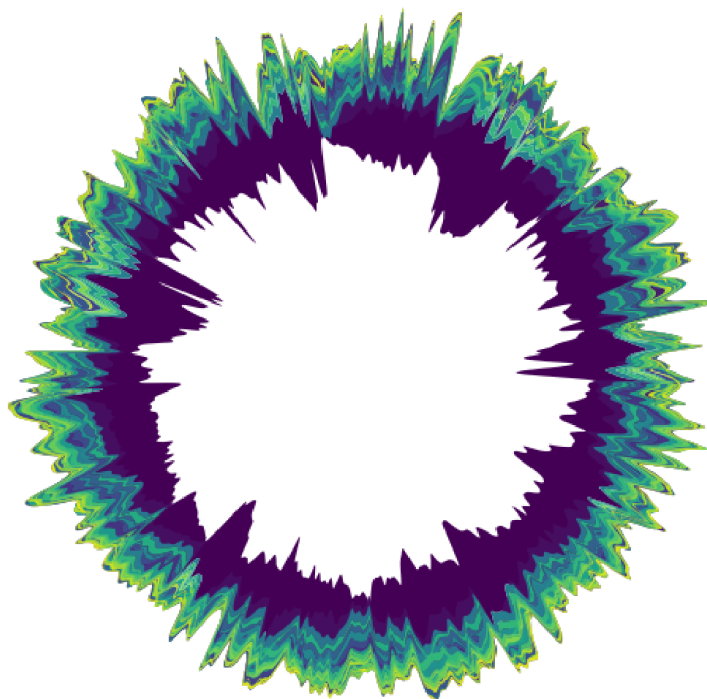
12. 5 *



Marcar apenas uma oval.

- ☐ Electronic
- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Rap

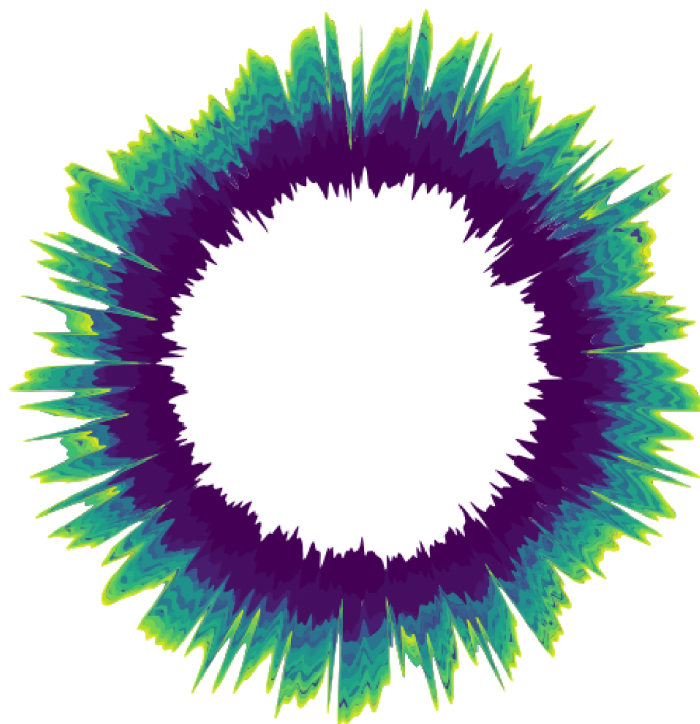
13. 6 *



Marcar apenas uma oval.

- ☐ Electronic
- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Rap

14. 7 *



Marcar apenas uma oval.

- ☐ Electronic
- ☐ Jazz
- ☐ Pop
- ☐ Rock
- ☐ Rap