# Sumário


**1 INTRODUÇÃO . . . 3**
**1.1 Fundamentação Teórica . . . 4**
1.1.1 A etapa de Pré-Processamento . . . 4
1.1.2 A etapa de Processamento . . . 6
1.1.2.1 O algoritmo Needleman & Wunsch (NW) . . . 7
1.1.2.2 O algoritmo Dynamic Time Warping (DTW) . . . 8
1.1.2.3 O algoritmo Smith Waterman (SW) . . . 9
1.1.3 A etapa de Pós-Processamento . . . 10
**1.2 Questão de Pesquisa e Hipótese . . . 11**
**1.3 Objetivo desta Pesquisa . . . 12**

**2 ANÁLISE DO PROBLEMA . . . 13**
**2.1 Contorno Melódico , Harmônico e Rítmico . . . 14**
**2.2 A atitude do Músico . . . 15**
**2.3 Convergências Conceituais . . . 15**
**2.4 Conclusão da Análise . . . 16**

**3 A MATRIZ MUSSIX (MU6) . . . 17**
**3.1 Cálculos da Mu6 . . . 18**
3.1.1 Definição de Match/Mismatch da Mu6 . . . 18
3.1.2 Definição de Penalidades da Mu6 . . . 18
**3.2 O Algoritmo Mu6 . . . 20**
**3.3 O Attitude-Gram . . . 21**
3.3.1 As Características Attitude-Gram . . . 23

**4 PROTOCOLO EXPERIMENTAL . . . 25**
**4.1 Passos para realização de uma Busca Query-By . . . 25**
4.1.1 Passos 1 a 3 - O Pré-Processamento das Buscas . . . 26
4.1.1.1 A Transposição das Músicas para escala Única . . . 28
4.1.1.2 A Categorização de Músicas . . . 29
4.1.1.3 A Montagem de Listas para Busca . . . 30
4.1.2 Passos 4 a 6 - O Processamento de Uma Busca . . . 30
4.1.2.1 Passos para uma Comparação Musical . . . 30
4.1.2.2 A Inicialização da Comparação . . . 31
4.1.2.3 O Alinhamento da Comparação . . . 32
4.1.2.4 Parametros de Processamento do Alinhamento . . . 34

4.1.2.4.1 Parametrização Fixa - Smith Waterman Original . . . 35
4.1.2.4.2 Parametrização MUSSUM . . . 35
4.1.2.4.3 Parametrização Mussix (Mu6) . . . 35
4.1.2.5 O Caminhamento reverso - Pontuação de Similaridade . . . 36
4.1.2.6 A Classificação de uma música de comparação . . . 39
4.1.3 Passo 07 - O Pós-Processamento de uma Busca . . . 40

**5 AVALIAÇÃO DOS RESULTADOS . . . 42**
**5.1 Experimentos . . . 42**
5.1.1 Experimento 1 - Avaliação de Metricas . . . 42
5.1.2 Experimento 2 - Comparação de Pontuação de Similaridade . . . 43
5.1.3 Experimento 3 - Variação de Balanço de Precisão . . . 43
5.1.4 Experimento 4 - Análise de Tempo . . . 43
**5.2 Resultados Obtidos . . . 44**
5.2.1 Resultado 01 - Metricas de Avaliação . . . 44
5.2.2 Resultado 02 - Comparação de Pontuação . . . 45
5.2.3 Resultado 03 - Variação de Balanço de Precisão . . . 46
5.2.4 Resultado 04 - Análise de Tempo . . . 48
**5.3 Análise dos Resultados . . . 49**

**6 CONSIDERAÇÕES FINAIS . . . 51**
**6.1 Relevância do Estudo . . . 51**
**6.2 Contribuições da Pesquisa . . . 51**
**6.3 Limitações da Pesquisa . . . 51**
**6.4 Trabalhos Futuros . . . 51**

**REFERÊNCIAS . . . 52**

# 1 Introdução

Music Information Retrieval (MIR) é um campo de pesquisa multidisciplinar que se baseia nas tradições, metodologias e técnicas de uma variedade notavelmente ampla de disciplinas incluindo Acústica,PsicoAcústica,Musicologia,Ciência da Computação,Aprendizagem de Máquina, entre outros.

O principal objetivo da pesquisa MIR, independentemente do paradigma disciplinar sob o qual é realizada, é proporcionar o acesso ao vasto conteúdo musical existente atualmente em um nível igual ou superior ao realizado pelos mecanismos de pesquisa baseados em texto.

Uma música por si só, é uma junção complexa de fenômenos acústicos, rítmicos, harmônicos, estruturais e culturais, onde o músico expressa sua arte através da sua atitude musical, ou seja , a maneira como o Músico se expressa através de seu instrumento musical, seja ele qual for.

O grande desafio da pesquisa MIR é o desenvolvimento de sistemas de recuperação que sejam capazes de interpretar este conteúdo por si só, Ou seja, os pesquisadores do MIR se esforçam para criar sistemas de recuperação em que a própria música, representada na forma de áudio (por exemplo, MP3s, WAV, etc.) ou de maneira simbólica (por exemplo, MIDI, partituras etc.) , seja o principal mecanismo pelo qual os usuários/músicos interagem com tais sistemas.

Em termos simples, a pesquisa do MIR quer desenvolver sistemas que permitam aos usuários pesquisarem conteúdo musical usando consultas estruturadas musicalmente.

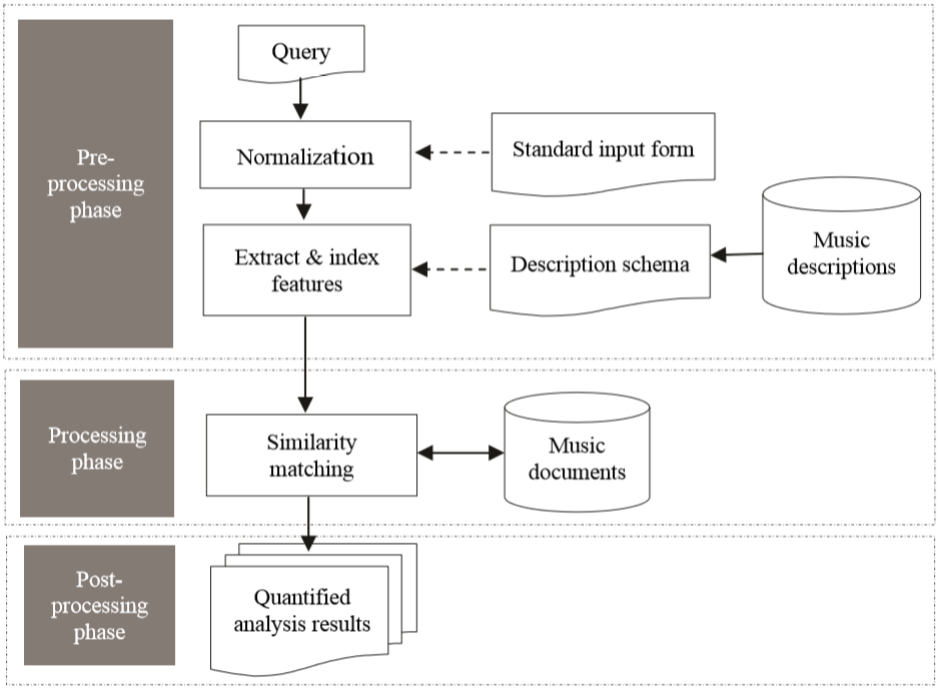
Tais consultas podem ser estruturadas musicalmente de diversas maneiras, onde comumente em MIR vêm sido identificadas como consultas "Query-By", onde podem incluir técnicas como consulta por canto, consulta por exemplo (por exemplo, enviar um MP3 conhecido para encontrar peças semelhantes) e / ou consulta por notação (por exemplo, colocar notas em um quadro musical para formar a consulta), etc.

Este documento descreve a pesquisa realizada no intuito de fazer com que buscas Query-By-Playing , possam realizar estas consultas por similaridade , utilizando dados simbólicos musicais, de maneira mais precisa melhorando os resultados destas buscas, tornando-as mais Musicais do que apenas utilizar a similaridade musical calculada através da análise tonal da música(Notas Musicais).

## 1.1 Fundamentação Teórica

De uma maneira geral um sistema MIR funciona em etapas distintas responsáveis pelo processamento do som e detalhadas no artigo (Kuldeep Gurjar, 2018)(Figura 1).

Figura 1 – MIR System: Estrutura de Processamento




Fonte: (Kuldeep Gurjar, 2018)

### 1.1.1 A etapa de Pré-Processamento

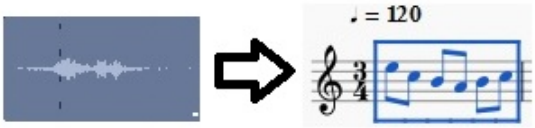
A grande maioria dos dados musicais utilizados, sendo arquivos de áudio, ou arquivos com dados musicais simbólicos são armazenados procurando seguir a mesma estrutura musical original, onde uma música é considerada como o encadeamento sequencial de notas musicais, que de maneira cadenciada forma a sua estrutura fundamental.

Na etapa de pré-processamento de uma Query , a informação recebida para a busca, é extraida , convertida e estruturada, para que possa ser utilizada na etapa posterior de processamento.

De maneira geral, esta conversão ocorre de acordo com o tipo e os propósitos da consulta sendo realizada.

Como identificado por (MARTINIANO; N. Silla, 2017), a grande maioria dos sistemas de consultas musicais, acaba convertendo dados de áudio , para dados simbólicos como exemplificado na Figura 2.

Figura 2 – Pré-Processamento: Áudio para Dados Simbólicos

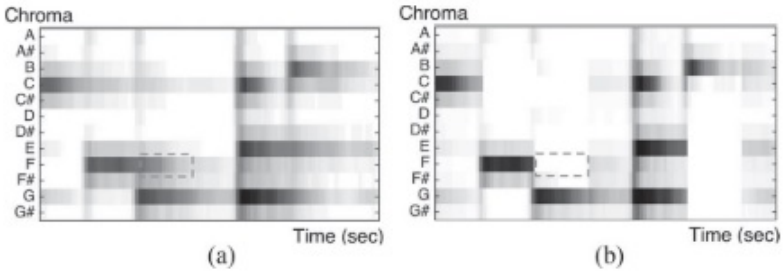



Fonte: Autoria Própria

Outros tipos de conversão podem ser realizados nesta etapa, como por exemplo em Em 2013 no artigo (PARK; LEE, 2013), Min Woo Park & Eui Chul Lee , através de uma abordagem de processamento de dados simbólico converteram faixas MIDI em um vetor unidimensional numerico contendo apenas algumas características musicais. A base desta pesquisa foi a proposta de um algoritmo de conversao das notas musicais simbolicas MIDI (pitch e tempo), em um vetor unidimensional de notas numericas.

Nesta etapa os dados podem ser estruturados de forma diferente ao convencional onde por exemplo, Em 2016, no artigo (LI; DUAN, 2016), é apresentado uma forma de processar o calculo de score de similaridade convertendo dados de Áudio em um Cromagrama para possibilitar o mapeamento de variações tonais causadas por alavancas tonais presentes nos teclados musicais.

Figura 3 – Pré-Processamento: Áudio para Chromagrama

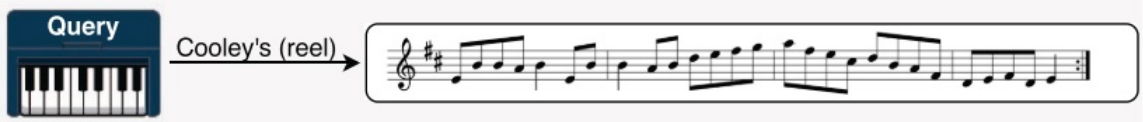



Fonte: (LI; DUAN, 2016)

Em 2017 no artigo (MARTINIANO; N. Silla, 2017)"BIRITS: A Music Information Retrieval System Using Query-by-Playing Techniques", Martimiano & Silla apresentam um metodo de Query-by-Playing em dois estágios utilizando dados simbólicos da musica.Em seu pré-processamento é realizado a conversão de músicas simbólicas MIDI em cadeias de strings , para o processamento de calculo de similaridade, conforme a Figura 4.

A conversão de dados MIDI em strings possibilitou a redução das notas para o calculo de similaridade a 12 notas pré-definidas presentes na escala diatônica ocidental, convertidas dentro da escala de Dó(Escala 5 MIDI).

Figura 4 – Pré-Processamento: MIDI para Strings




Fonte: (MARTINIANO; N. Silla, 2017)

### 1.1.2 A etapa de Processamento

A etapa de processamento caracteriza o núcleo da funcionalidade de busca (Query) e em um sistema MIR abrange primordialmente o cálculo de pontuação de similaridade entre as musicas envolvidas na busca.

Da mesma maneira que no Pré-Processamento as técnicas e algoritmos utilizados nesta etapa variam de acordo com o tipo de busca sendo realizada, que precisa estar em concordância com os dados recebidos pelo pré-processamento.

Várias técnicas para avaliar músicas simbólicas similares foram introduzidas nos últimos anos, desde algoritmos geométricos,que consideram representações geométricas de melodias e calculam a distância entre estas músicas, até utilização de aprendizagem de máquina e métodos probabilisticos como redes de Markov para determinar esta similaridade.

Em 2013, Ching-Hua Chuan no artigo (CHUAN, 2013)"A multimodal approach to song-level style identification in pop/rock using similarity metrics"é utilizado a geração de n-grams a partir de sequencias de acordes realizados por musicos para a identificação do seu estilo individual(Signature Style).Neste artigo foram gerados três(3) n-grams gerados a partir de calculos de Similaridade Melódica,Perfil de Harmonia e Características acústicas. e que alimentam um Classificador Binario em SVM(Support Vector Machines) para classificar o estilo de cada musico.

No mesmo ano, Alessio Degani, Marco Dalai, Riccardo Leonardi e Pierangelo Migliorati no artigo (DEGANI et al., 2013)"A Heuristic for Distance Fusion in Cover Song Identification"detalham uma pesquisa de fusão de duas features musicais que otimizam os resultados de calculo de similaridade entre musicas.Para os experimentos são utilizadas a combinação de duas features (Salience Feature e Harmonic Pitch Class Profile - HPCP) sendo utilizadas em dois algoritmos de calculo de distancia (Dinamic Time Warping e Qmax Cross Recurrence Plot).

Em 2017 no artigo (MARTINIANO; N. Silla, 2017)"BIRITS: A Music Information Retrieval System Using Query-by-Playing Techniques", Martimiano & Silla utilizam um calculo de similaridade por strings de notas e propõe uma Matriz de Pesos contendo parametros de Acertos/Erros baseada na distãncia em semi-tons entre as notas musicais para a otimização dos resultados do alinhamento local chamada MUSSUM(Music Substitution Matrix).

É importante notar que apesar destes métodos e algoritmos se apliquem ao cálculo da medida de similaridade, os únicos algoritmos que levam em consideração as características principais de uma música(Encadeamento sequencial de notas no tempo), são os algoritmos de alinhamento de sequências Global e Local.

#### 1.1.2.1 O algoritmo Needleman & Wunsch (NW)

Como descrito em (FERRARO; HANNA, 2007) , em 1970 Needleman & Wunsch (NEEDLEMAN; WUNSCH, 1970) e, em seguida, Wagner & Fisher (1974) propuseram algoritmos que computam uma medida de similaridade entre duas cadeias de Símbolos como a sequência mínima de pontuação das operações elementares necessárias para transformar uma das cadeias na outra.

Dadas duas cadeias de símbolos S1 e S2 com seus respectivos comprimentos |S1 |e |S2 |, ou seja, o número de notas em melodias , um conjunto de operadores elementares em strings, chamados de operações de edição e uma pontuação associada a cada operação de edição, uma pontuação entre essas duas strings é definida como a pontuação da sequência de operações de edição transformando S1 em S2 com uma pontuação mínima. Esta medida de similaridade faz uso do princípio de programação dinâmica para alcançar um algoritmo com complexidade quadrática, ou seja, em O (|S1|x |S2| ).

A idéia principal do algoritmo de alinhamento é mapear o numero de operações de Substituição,Alteração ou Exclusão necessárias para transformar S1 em S2, onde o score de similaridade seria provindo do número de operações realizadas para cada item da string, armazenados durante o andamento do algoritmo em uma matriz de distancias(numero de operacoes) ou score.

Mesmo utilizando-se do algoritmo e uma matriz de distâncias , executando os algoritmos anteriores obtemos apenas o score de similaridade. Caso seja necessário obter também a string resultante deste alinhamento, é necessário armazenar todas as operações realizadas para que posteriormente possamos reconstruir o caminho percorrido na matriz.

Esta reconstrução do caminho é comumente chamada de Backtracing, onde traçamos o caminho percorrido das operações e determinarmos o seu resultado. Este resultado final determina o seu nome de algoritmo, pois através do alinhamento das distancias e com o backtracing das operações resultando em uma string alinhada em sua totalidade, o nome alinhamento global foi adotado para este tipo de algoritmo.

**Algoritmo 1:** O Algoritmo Needleman & Wunsch

**Result:** Matriz de Scores Calculada(Bottom-Up).

```
Inicialização;
D(i,0) = i;
D(0,j) = j;
Calculo de Scores;
for i ← 1 to M do
    for j ← 1 to N do
        D(i,j)= Min(i,j);
    end
end
```

**Algoritmo 2:** A Funcao Min

**input** : A posicao atual $i$,$j$

**output** : Distancia Minima

**Result:** O resultado sera o score minimo de distancia.

```
embaixo = M[i-1,j ] + 1;
lado = M[i ,j-1] + 1;
diagonal = M[i-1,j-1] + ScoredeErro : Acerto;
retorno = Minimo(embaixo,lado,diagonal)
```

Figura 5 – Processamento NW: a) Scores e b) Backtracing

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| N | 9 | 8 | 9 | 10 | 11 | 12 | 11 | 10 | 9 | 8 |
| O | 8 | 7 | 8 | 9 | 10 | 11 | 10 | 9 | 8 | 9 |
| I | 7 | 6 | 7 | 8 | 9 | 10 | 9 | 8 | 9 | 10 |
| T | 6 | 5 | 6 | 7 | 8 | 9 | 8 | 9 | 10 | 11 |
| N | 5 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 10 |
| E | 4 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 9 |
| T | 3 | 4 | 5 | 6 | 7 | 8 | 7 | 8 | 9 | 8 |
| N | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 7 | 8 | 7 |
| I | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 6 | 7 | 8 |
| # | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | # | E | X | E | C | U | T | I | O | N |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| n | 9 | ↓8 | ↙←↓9 | ↙←↓10 | ↙←↓11 | ↙←↓12 | ↓11 | ↓10 | ↓9 | ↙8 |
| o | 8 | ↓7 | ↙←↓8 | ↙←↓9 | ↙←↓10 | ↙←↓11 | ↓10 | ↓9 | ↙8 | ←9 |
| i | 7 | ↓6 | ↙←↓7 | ↙←↓8 | ↙←↓9 | ↙←↓10 | ↓9 | ↙8 | ←9 | ←10 |
| t | 6 | ↓5 | ↙←↓6 | ↙←↓7 | ↙←↓8 | ↙←↓9 | ↙8 | ←9 | ←10 | ←↓11 |
| n | 5 | ↓4 | ↙←↓5 | ↙←↓6 | ↙←↓7 | ↙←↓8 | ↙←↓9 | ↙←↓10 | ↙←↓11 | ↙↓10 |
| e | 4 | ↙3 | ←4 | ↙←5 | ←6 | ←7 | ←↓8 | ↙←↓9 | ↙←↓10 | ↓9 |
| t | 3 | ↙←↓4 | ↙←↓5 | ↙←↓6 | ↙←↓7 | ↙←↓8 | ↙7 | ←↓8 | ↙←↓9 | ↓8 |
| n | 2 | ↙←↓3 | ↙←↓4 | ↙←↓5 | ↙←↓6 | ↙←↓7 | ↙←↓8 | ↓7 | ↙←↓8 | ↙7 |
| i | 1 | ↙←↓2 | ↙←↓3 | ↙←↓4 | ↙←↓5 | ↙←↓6 | ↙←↓7 | ↙6 | ←7 | ←8 |
| # | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | # | e | x | e | c | u | t | i | o | n |

Fonte: Autoria própria

#### 1.1.2.2 O algoritmo Dynamic Time Warping (DTW)

O algoritmo DTW é um algoritmo de alinhamento global comumente utilizado no processamento de músicas polifônicas por possuir um cálculo geométrico de distância (Distância Euclideana) na construção da matriz de pesos.

Tecnicamente a única diferença entre os algoritmos NW e o DTW consiste na soma do peso atribuído ao score mínimo em cada iteração do algoritmo, onde no algoritmo NW

esta soma ocorre apenas quando um match(diagonal) ocorre.

**Algoritmo 3:** O Algoritmo DTW

**Result:** Matriz de Scores Calculada(Bottom-Up).

```
initialization;
D(i,0) = infinito;
D(0,j) = infinito;
D(0,0) = 0;
for i ← 1 to M do
    for j ← 1 to N do
        custo = EuclideanDistance(i,j);
        D(i,j)= custo + Min(i,j);
    end
end
```

**Algoritmo 4:** A Funcao Min

**input** : A posicao atual $i$,$j$

**output** : Distancia Minima

**Result:** O resultado sera o score minimo de distancia.

```
embaixo = M[i-1,j ];
lado = M[i ,j-1];
diagonal = M[i-1,j-1];
retorno = Min(embaixo,lado,diagonal)
```

#### 1.1.2.3 O algoritmo Smith Waterman (SW)

Como descrito também em (FERRARO; HANNA, 2007),Em muitas aplicações, duas cadeias podem não ser muito semelhantes na sua totalidade, mas podem conter regiões altamente semelhantes (substrings). Nesse caso, a tarefa é localizar e extrair um par de regiões, uma a uma das duas cadeias onde existem altas similaridades. Isto é chamado de alinhamento local ou problema de similaridade local definido por Smith & Waterman em 1981 (SMITH; WATERMAN, 1981) e é definido como: dada duas sequências S1 e S2, encontrar as substrings p1 e p2 de S1 e S2 respectivamente, cuja similaridade é máxima sobre todos os pares de substrings de S1 e S2.

Como no algoritmo de alinhamento global, o algoritmo de alinhamento local, também precisa realizar um backtracing na sua matriz (Chamada de Substitution Matrix), e que resulta em diferenças significativas.

A principal diferença está nas repetições de score máximo , onde para cada score encontrado, uma substring local é gerada.

Outra diferença significativa no algoritmo relativo a seu antecessor é a possibilidade de processamento dos Gaps entre estas substrings, permitindo que uma tolerância a erros possa ser mapeada.

**Algoritmo 5:** O Algoritmo Smith Waterman

**Result:** Matriz de Scores Calculada(Top-Down).

```
Inicialização;
D(i,j) = 0;
Calculo de Scores;
for i ← 1 to M do
    for j ← 1 to N do
        D(i,j)= Max(i,j);
    end
end
```

**Algoritmo 6:** A Funcao Max

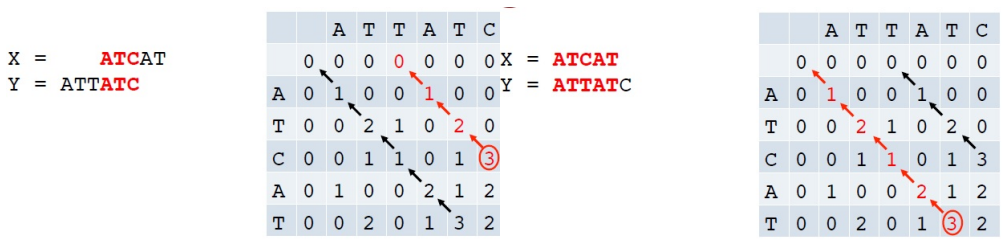
**input** : A posicao atual $i$,$j$

**output** : Distancia Maxima

**Result:** O resultado sera o score maximo de distancia.

```
lado = M[i-1,j ] + gap;
topo = M[i ,j-1] + gap;
diagonal = M[i-1,j-1] + ScoredeErro : Acerto;
retorno = Maximo(topo,lado,diagonal)
```

Figura 6 – Processamento SW: Backtracing




Fonte: Autoria própria

### 1.1.3 A etapa de Pós-Processamento

De maneira geral na etapa de pós-processamento os resultados do processamento de similaridade musical são coletados.

É na etapa de pós-processamento que dados estatísticos como acuracia e ranqueamento de similaridade são coletados para uma posterior análise quantitativa das informações geradas em cada busca MIR.

Em 2007 no artigo (FERRARO; HANNA, 2007) "Optimizations of Local Edition for Evaluating Similarity Between Monophonic Musical Sequences", Ferraro & Hanna realizam comparações entre os algoritmos de similaridade utilizados até então.Nestas comparações são propostos novos parâmetros de configuração dos algoritmos baseado em regras musicais para a melhoria da acuracia destes algoritmos.Para a avaliação dos resultados foi criado pós-processamento onde a técnica Average Dynamic Recall (ADR), é utilizada para avaliar todos os experimentos.

No artigo (WANG; TSAI; SU, 2014)"Note-based alignment using score-driven non-negative matrix factorisation for audio recordings"onde Tien-Ming Wang, Pei-Yin Tsai & Alvin W. Y. Su, apresentam a divisão de faixas de áudio em notas simbólicas MIDI e colocadas em uma estrutura de conversao chamada Piano-Roll feature.Todo o Pós-Processamento das comparações levaram em conta a acuracia resultante como medida de comparação.

Conforme exposto no artigo (DOWNIE et al., 2010)“The Music Information Retrieval Evaluation eXchange: Some Observations and Insights”, Stephen Downie coloca “a seleção ou criação de métricas de avaliação apropriadas é crucial para a avaliação científica apropriada do desempenho do sistema MIR.”

A seleção das métricas de avaliação também possui um forte componente emocional, à medida que os participantes se esforçam para mostrar o sucesso de seus algoritmos e sistemas da melhor maneira possível.

Ainda no mesmo artigo é apresentado uma lista de métricas para a avaliação de sistemas MIR, estas métricas são utilizadas na competição de algoritmos MIR, chamada MIREX - Music Information Retrieval Evaluation eXchange, e seguem abaixo elencadas as principais métricas utilizadas para avaliação de algoritmos MIR.

1. Precision, Recall e F-Measure
2. Mean Reciprocal Rank
3. Average Accuracy

## 1.2 Questão de Pesquisa e Hipótese

Para este trabalho temos a seguinte questão de pesquisa:

1. É Possível obter resultados mais "Musicais", que se aproxime da forma como o ser humano identifica Músicas?

A hipótese desta pesquisa é:

1. Existe uma forma de obter scores de similaridade mais musicais através de parametrização de alinhamento de sequencia local entre músicas.

## 1.3 Objetivo desta Pesquisa

O Cálculo de Pontuação de Similaridade , em todos os trabalhos de MIR , desempenha papel central, onde através dele são elencadas as musicas sejam para análise de resultados , posterior classificação, ou até mesmo para a preparação de dados para futura utilização.

Portanto o objetivo deste trabalho é otimizar o cálculo de pontuação de similaridade para buscas de músicas simbólicas (SMS) Query-By-Playing, a fim de obtermos maior precisão nos resultados de um modo geral.

# 2 Análise do Problema

Em todas as pesquisas analisadas para este trabalho, quando o cálculo de similaridade leva em conta a estrutura de uma música, a mesma realiza este calculo de pontuação levando-se em conta apenas sua estrutura tonal(notas musicais), onde outros aspectos musicais que podem auxiliar no aumento da precisão destes resultados não é utilizada e quando utilizada, a mesma é feita após ou antes do calculo de pontuação, onde não foi encontrado nenhuma, em que outras estruturas musicais fossem utilizadas durante o processamento de calculo.

Isto se deve em alguns casos, por não haver a necessidade de uma precisão alta para a seleção das músicas, em outros casos foi identificado que devido aos algoritmos de alinhamento de sequência garantirem o melhor alinhamento, o maior score sempre é resultante da música correta não havendo necessidade de precisão maior.

Em Query-By-Playing além de possuirmos apenas fragmentos de músicas para alinhamento, a precisão no retorno de tais músicas deve ser refletida em toda a lista de músicas e não somente na primeira música mais similar, onde a ocorrência de um número alto de repetições de pontuação descaracterizam o resultado final da busca.

Por Exemplo, no artigo (MARTINIANO; N. Silla, 2017)"BIRITS: A Music Information Retrieval System Using Query-by-Playing Techniques", Martimiano & Silla apresentam um metodo de Query-by-Playing em dois estágios utilizando dados simbólicos da musica.Em seu trabalho, no segundo estágio de processamento é aplicado um filtro SVM , para a diminuição de repetições de pontuação de similaridade,onde mesmo com a aplicação de filtro SVM com 87% de acurácia, a efetividade na filtragem dos resultados foi em torno de 1% dos resultados.

No trabalho BIRITS , foi obtido resultados significativos relativos a forma de utilização do algoritmo de alinhamento e consequentemente no calculo de pontuação de similaridade, onde fora criada a matriz MUSSUM - Music Substitution Matrix.A idéia principal desta matriz foi a utilização da distância em semi-tons das notas da escala diatônica para a atribuição das penalidades (Gaps) no algoritmo de alinhamento local Smith-Waterman (SW).

Isto abriu o precedente principal da hipótese deste trabalho, onde viu-se que, seria possível otimizar os resultados destes alinhamentos e obter pontuação de similaridade mais precisa e com um menor indice de repetições.

Aliado a isto, devido a entrada da consulta musical ser realizada por músicos, a precisão do retorno deve dar a impressão ao ser humano, que não apenas o acerto de notas

está sendo levada em consideração, onde a utilização de estruturas musicais independentes da característica tonal da música precisa ser levada em conta também.

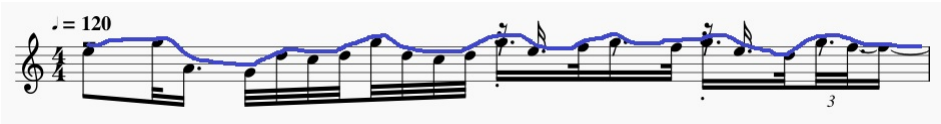
## 2.1 Contorno Melódico , Harmônico e Rítmico

Um contorno de maneira geral pode ser descrito como o desenho do relevo formado ao longo de uma ou mais medidas distribuidas através de uma sequência.

Para a definição de uma música, existem três componentes fundamentais, a melodia , a Harmonia e o Rítmo.

Uma melodia pode ser considerada como o encadeamento sequencial de notas musicais, que de maneira cadenciada forma a estrutura fundamental da Música(Figura 7).

Figura 7 – Contorno Melódico de uma Música.




Fonte: Autoria Própria

Cada nota emitida por um instrumento musical possui uma frequencia específica de acordo com a escala tonal do instrumento musical, e as mesmas são cadenciadas na melodia através do tempo de vibração de cada nota formando o Ritmo Musical.

Cada uma das notas musicais da escala diatônica (Dó, Ré , Mi , Fá , Sol, La , Si) distanciam-se umas das outras através de uma medida de Tons e Semi-Tons, onde a menor dela é o Semi-Tom.

Portanto um Contorno Melódico pode ser definido como o numero de semi-tons entre uma nota e sua nota subsequente na melodia. Conforme citado por (DUREY; CLEMENTS, 2003), o contorno melódico é encontrado quando calculamos o numero de semi-tons de uma nota para a próxima.

Extendendo esta teoria para as demais estruturas musicais , o Contorno Harmônico é o contorno desenhado pelas frequências harmônicas de cada uma das notas da cadência, e o Contorno Rítmico é a Duração de cada nota da cadência e que define o ritmo da música.

Foi percebido em (MARVIN, 1988) que a percepção do contorno para o ser humano é mais imediata do que a do tom e Parece provável que a percepção do contorno, em algum nível, tenha um papel na maneira como os ouvintes podem diferenciar melodias, associar relações temáticas, determinar formalidades esquemas e assim por diante, pois atendem a uma composição não-tonal.

Aplicações analíticas da teoria dos contornos, portanto, modelam alguns aspectos da percepção musical do ouvinte que a análise de pitch sozinha não aborda.

## 2.2 A atitude do Músico

Outro aspecto importante no intuito de simular a percepção humana na detecção de similaridades musicais é relativo ao modo de reconhecimento de padrões ritmicos e harmônicos provindos da atitude do músico na composição musical.

De acordo com o estudo de (MARVIN, 1988):

"Finalmente, a generalização da teoria do contorno em outros domínios permite ao analista comparar diversas facetas da estrutura musical em uma única escala seqüencial. Os exemplos analíticos mostraram maneiras pelas quais a análise de sucessões de duração como contornos rítmicos esclarece alguns aspectos da linguagem musical de um compositor."

Mesmo com excelentes resultados em seu trabalho, boa parte do estudo de MIR busca reproduzir a reação humana em identificar a similaridade do ouvinte , porém quase nenhum estudo foi encontrado na tentativa de tentar reproduzir como o músico tenta expressar seus sentimentos e expressões através de sua música.

Portanto chega-se a conclusão que, não basta apenas a verificação por similaridade Tonal somada ao Contorno Melódico, mas também as características de Atitude do Músico na composição musical também possuem um grau significativo da percepção musical do ser humano, já que acabam expressando aspectos culturais e sociais do mesmo.

## 2.3 Convergências Conceituais

Apesar de parecerem bastante promissores os resultados avaliados por (MARVIN, 1988), em 2007 no artigo (FERRARO; HANNA, 2007) "Optimizations of Local Edition for Evaluating Similarity Between Monophonic Musical Sequences", Ferraro & Hanna realizam comparações entre os algoritmos de similaridade utilizados até então e identificaram que além de haver diferenças entre resultados provindos de alinhamentos locais e Globais, a utilização de contorno melódico foi avaliada da seguinte forma:

"Os resultados mostram claramente que a representação do contorno leva aos piores resultados. A diferença de acurácia é muito significativa: 38%, enquanto os demais resultados são maiores que 50%. Isso se justifica principalmente pela falta de informações contidas nessa representação. É óbvio que várias melodias podem ser representadas pelo mesmo contorno melódico. Assim, o vocabulário proposto parece ser muito limitado para esta aplicação musical. As representações relativas absolutas e principais levam aproximadamente às mesmas medidas médias de RAM. No entanto, as principais representações

relativas e de intervalo apresentam a grande vantagem de propor um escore de similaridade invariável na transposição."

Através de uma análise do trabalho de (CHUAN, 2013), foi possível verificar também o mapeamento dos sentimentos intuitivos de similaridade na perspectiva do ouvinte, concentrando-se na criação de características inseridas em um n-gram de valores numéricos, para treinamento de um Classificador SVM , identificando o estilo musical de um Músico.

Como resultado Ching-Hua Chuan mostra que a abordagem alcança o melhor resultado na taxa correta de 85% usando apenas sete características de similaridade.

Portanto é possível concluir que as soluções utilizando a estrutura tonal da música possuem a limitação da dependência da escala tonal da música para transposição, e que as soluções que propoe a utilização de estruturas independentes de transposição como o contorno melódico, apesar de possuírem esta vantagem , possuem baixo indice de acertos na pontuação de similaridade, bem como , além disto a atitude do músico em suas composições oferecem promissoras informações para a identificação de similaridade sem que a verificação nota por nota precise ser feita.

## 2.4 Conclusão da Análise

Levando em consideração os trabalhos realizados anteriormente e os conceitos musicais levantados para a análise do problema , fica aparente que, é necessário encontrar uma forma de cálculo de pontuação de similaridade que seja capaz de unir a estrutura tonal da música , com a estrutura invariante de contornos musicais, bem como o mapeamento da atitude do músico, para atingirmos o resultado esperado por este trabalho.

# 3 A Matriz MUSSIX (Mu6)

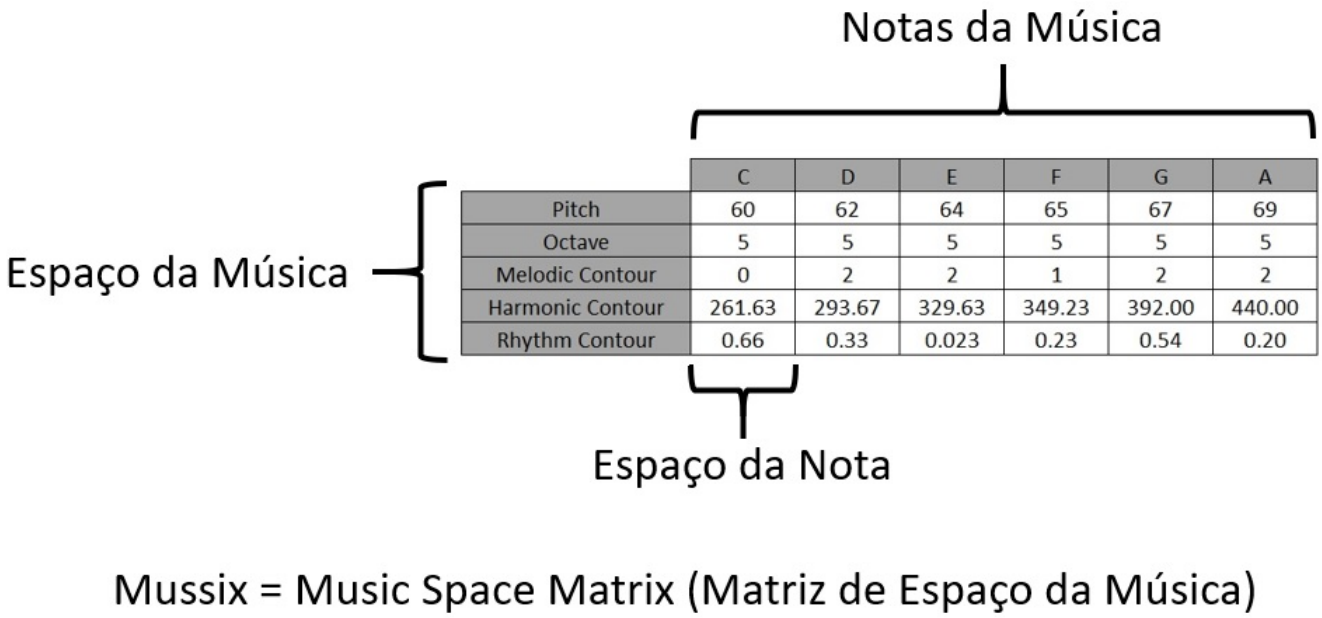
A matriz MUSSIX , também apelidada de Mu6 , surgiu no intuito de se obter uma matriz de parametrização do cálculo de similaridade capaz de retornar scores de músicas próximas da forma que o ser humano identifica a similaridade entre as músicas, ou seja , resultados mais "Musicais".

Inspirada na matriz MUSSUM (MARTINIANO; N. Silla, 2017) , a Mu6 procura expandir o cálculo de Match/Mismatch e Gaps baseados apenas nas notas musicais (realizado pela MUSSUM), e passa a utilizar outros aspectos da percepção musical, como Melodia , Harmonia e Ritmo.

Para isto, expandiu-se o conceito de notas musicais, para notas musicais Multi-Dimensionais, onde cada nota da música possui um espaço específico dentro do espaço musical contido em uma música.

Isto possibilitou aplicar os conceitos de contornos musicais durante o processamento de calculo do score de similaridade(Figura 8).

Figura 8 – A Matriz MUSSIX (Mu6).




| | C | D | E | F | G | A |
|---|---|---|---|---|---|---|
| Pitch | 60 | 62 | 64 | 65 | 67 | 69 |
| Octave | 5 | 5 | 5 | 5 | 5 | 5 |
| Melodic Contour | 0 | 2 | 2 | 1 | 2 | 2 |
| Harmonic Contour | 261.63 | 293.67 | 329.63 | 349.23 | 392.00 | 440.00 |
| Rhythm Contour | 0.66 | 0.33 | 0.023 | 0.23 | 0.54 | 0.20 |

Fonte: Autoria Própria

## 3.1 Cálculos da Mu6

O valor final de pontuação de similaridade é proveniente do alinhamento local realizado pelo algoritmo, e seu valor é diretamente influenciado pelos valores de Match/Mismatch e Penalidades (Gaps) aplicados durante o processamento.

Diferente do algoritmo original Smith-Waterman (SMITH; WATERMAN, 1981) com valores de Math/Mismatch e Gaps Fixos, bem como do calculo realizado pela MUSSUM (MARTINIANO; N. Silla, 2017) que utiliza a distancia em semi-tons entre as notas para a definição das penalidades, a matriz Mu6 utiliza o espaço determinado pelos valores de Pitch,Contorno Melódico e a Oitava da Nota para a definição do valor de Match/Mismatch, e utiliza os valores de contorno Harmônico e Rítmico para determinar a penalidade a ser atribuída.

### 3.1.1 Definição de Match/Mismatch da Mu6

Durante o processamento convolutivo do algoritmo de alinhamento local , é realizada uma comparação de Match/Mismatch para determinar a pontuação e direção a qual este valor foi originado (Esquerda , Diagonal ou Topo).

É neste momento em que é aplicado o calculo de match/mismatch da matriz Mu6, utilizando um cálculo de distância entre os componentes Pitch,Oitava e Contorno melódico da nota do Sinal A , com o Sinal B.

O cálculo de distância sendo aplicado atualmente é a distância Manhattan , onde considerando A e B como as notas multi-dimensionais sendo comparadas, $x$ sendo considerada como o valor de pitch da nota, $y$ sendo o valor da oitava da nota e $z$ como sendo o contorno melódico da nota, temos especificado a equação 3.1.

$$d(A, B) = |Ax - Bx| + |Ay - By| + |Az - Bz| \tag{3.1}$$

Após calculado a distância entre as notas, é comparado se a distância é maior que zeros, caso a distância for zero temos um Match, caso contrário temos um Mismatch. O valor final atribuído ao resultado do Match e Mismatch é a Oitava da nota para um Match e o Valor da Oitava negativado em caso de um Mismatch.

### 3.1.2 Definição de Penalidades da Mu6

Ainda durante o processamento convolutivo do algoritmo de alinhamento local , nos casos em que ocorre um Mismatch um cálculo de penalidade é aplicado e atribuído aos parâmetros de funcionamento.

O cálculo de penalidade foi inspirado na reação que um músico tem ao errar uma nota em seu instrumento musical, onde ao errar uma nota , por reflexo tende-se a interromper a execução da nota.

Portanto, dificilmente um músico que cometa um erro de execução irá manter a nota sendo tocada por um longo periodo, ou seja, a penalidade precisa ser inversamente proporcional ao tempo decorrido da nota errada.

Outro detalhe importante,é a característica dos instrumentos musicais em adaptar-se as limitações físiológicas do ser humano, referente a localização das oitavas, onde o erro ocorre frequentemente na mesma escala onde o contorno harmonico está sendo desenhado, e dificilmente um músico será capaz de errar uma nota que não esteja dentro da oitava atual.

Conceitualmente o cálculo de penalidades da Mu6 pode ser considerado como o valor normalizado do contorno Harmônico(dentro da escala atual), versus o tempo em que o erro ocorre. Como cada escala possui um intervalo fixo de frequências, foi possível calcular os valores mínimos e máximos da fórmula de normalização no mesmo momento em que a duração da nota ocorre.

Considerando $NormHz$ como a Frequência Normalizada, $Hz$ como a frequência da nota executada, $MinHz$ a Frequência Mínima da Escala e $MaxHz$ a frequência máxima da escala temos a equação de Normalização 3.2.

$$NormHz = (Hz - MinHz)/(MaxHz - MinHz) \tag{3.2}$$

Figura 9 – Calculo de Penalidades (Mu6).

Fonte: Autoria Própria

Consecutivamente considerando $P$ como a Penalidade a ser aplicada, $NormHz$ como a Frequência previamente normalizada e $Td$ como a duração em milisegundos da nota executada, temos a equação do cálculo de penalidades da Mu6 3.3.

$$P = NormHz/Td \tag{3.3}$$

## 3.2 O Algoritmo Mu6

Para o processamento do cálculo de pontuação de similaridade foram feitas adaptações no algoritmo de alinhamento Local (SW) para o processamento correto da Mu6.

Estas adaptações consistem na conversão de utilização de strings de notas musicais, para a utilização da Mu6 em cada trecho musical, e a inclusão do cálculo de distância e cálculo de penalidades.

Para isto a estrutura principal do algoritmo SW foi preservada, convertendo-se a estrutura em strings, para a matriz Mussix possibilitando que as notas multi-dimensionais sejam acessadas a cada iteração do algoritmo.

**Algoritmo 7:** O Algoritmo Mu6

```
Result: Matriz de Scores Calculada(Top-Down).
Inicialização;
D(i,j) = 0;
Calculo de Scores;
for i ← 1 to M do
    for j ← 1 to N do
        D(i,j)= Max(i,j);
    end
end
```

A cada iteração as notas multi-dimensionais A e B são comparadas através do cálculo de distância Manhattan, utilizando os componentes Pitch, Oitava da Nota e Contorno Melódico.

**Algoritmo 8:** A Funcao Distancia

**input** : Notas MultiDimensionais $A$ , $B$

**output** : Distancia entre as Notas Multi-Dimensionais

**Result:** O resultado sera a distancia Manhattan entre as notas.

```
tDistances[0] = A.Pitch - B.Pitch;
tDistances[1] = A.Contour - B.Contour;
tDistances[2] = A.Octave - B.Octave;
for i ← 1 to 3 do
    tResultDistance += tDistances[i];
end
retorno = tResultDistance
```

Após o calculo de Similaridade entre as Notas, é aplicado então os valores de Match/Mismatch ou Penalidades de acordo com o definido pela análise conceitual de funcionamento da Matriz Mu6.

**Algoritmo 9:** A Funcao DistanciaSimilaridade

**input** : A posicao atual $i$,$j$

**output** : Distancia de Similaridade

**Result:** O resultado sera a distancia de similaridade entre as notas.

```
A = Mu6A[i-1];
B = Mu6B[j-1];
if Distancia(A,B) = 0 then
    Acerto = A.Octave;
else
    ScoredeErro = −A.Octave;
    NormHz = (A.Frequency - MinHz)/(MaxHz - MinHz);
    gap = NormHz / A.Duration;
end
retorno = Maximo(topo,lado,diagonal)
```

## 3.3 O Attitude-Gram

Inspirado conceitualmente em identificar como a atitude de um musico pode ser traduzida em uma serie de características e tecnicamente pelo trabalho proposto por (CHUAN, 2013) , o Attitude-Gram é um vetor de características que é extraído da Mu6,no intuito de mapear a atitude do músico durante a performance musical e compará-la com a musica de comparação antes de realizar o alinhamento e calculo de pontuação de similaridade.

A idéia principal do Attitude-Gram é atribuir uma pontuação zerada, para toda música que estiver distante do Attitude-Gram, ou seja, somente músicas com a mesma atitude mapeada na musica sendo tocada será alvo de calculo de similaridade.

Com isto é possível otimizar e filtrar a lista final de músicas a serem retornadas pelo sistema ao Músico/Usuário.

No total foram extraídas oito (8) características da Mu6 com o objetivo principal de identificar similaridades entre a atitude empregada pelo músico no momento da composição musical.

Como Exemplo, podemos citar uma dupla de músicos que realizam uma performance com 2Cielos utilizando duas músicas vindas de músicos aparentemente sem relação nenhuma (5º Sinfonia de Bethoven e a música Whole Lotta Love da Banda de Rock Led Zeppelin) ilustrado na Figura 10.

Figura 10 – Attitude-Gram - Bethoven vs Led Zeppelin.

Fonte: <https://www.youtube.com/watch?v=x8yymm3DtVA>

No vídeo, podemos identificar diversas características que fazem com que as duas musicas sejam similares baseado principalmente na repetição de notas, ritmos e contorno harmônico, e foi baseado neste vídeo que serviu de inspiração para a definição das 8 características do Attitude-Gram.

### 3.3.1 As Características Attitude-Gram

No Total foram calculadas oito(8) características extraídas da Mu6 listadas na tabela 1.

Tabela 1 – Características do Attitude-Gram

| **Campo** | **Descrição** |
|---|---|
| Oitavas | Número de Oitavas utilizada durante a performance do músico. |
| Bemois | Número de Bemóis utilizado durante a performance. |
| Média de Duração Notas | A média de duração de todas as notas da música. |
| Nota Mais Utilizada | A nota da Escala diatônica mais utilizada durante a performance. |
| Notas Únicas | Quantidade das notas da escala diatônica utilizadas durante a performance. |
| Repetições Consecutivas | A maior quantidade de repetições consecutivas de uma nota durante a performance. |
| BPM Médio | A quantidade de Batidas por Minuto Média da performance. |
| Atitude do Músico | Valor Composto das caracteristicas anteriores representando a Atitude do músico durante a performance. |

Fonte: Autoria Própria

O objetivo principal do Attitude-Gram é mapear a atitude do músico durante a performance musical ou sua composição, e para tal realiza uma análise rítmica da obra mapeando principalmente repetições de notas , a média de velocidade na execução da obra e notas mais utilizadas.

O valor final de atitude do músico é calculado através da divisão entre a duração média das notas e o número de repetições consecutivas da nota mais utilizada. Considerando $Tm$ como a duração média das notas , $Rt$ como a maior quantidade de repetições da nota base e $Att$ como sendo a atitude do músico temos a seguinte equação 3.4

$$Att = Tm/Rt \tag{3.4}$$

Podem ocorrer casos em que não há repetições de notas mais usadas onde para se mapear a atitude do músico foi utilizado a quantidade de notas únicas da escala diatônica, portanto considerando $Nu$ como a quantidade de notas unicas utilizadas na composição , $Tm$ como a duração média das notas e $Att$ como sendo a atitude do músico temos a seguinte equação 3.5

$$Att = Tm/Nu \tag{3.5}$$

Outra característica importante para a definição do Attitude-Gram é o cálculo de Batidas por Minuto Médio, onde é possível estabelecer uma área comum entre as frequentes transições no andamento da performance musical, que em conjunto com a atitude do músico

conseguem determinar a distancia de atitude entre uma música e outra.Considerando $Tm$ como a duração média das notas e $MedBPM$ como sendo as batidas por minuto médias da performance chegou-se a equação 3.6.

$$MedBPM = 60000/Tm \tag{3.6}$$

Para se determinar se os músicos de cada composição, a da performance de entrada e da música pré-existente sendo consultada, um cálculo de distância Manhattan é realizado entre as oito características extraídas conforme o algoritmo de Cálculo de Distância Attitude-Gram.

**Algoritmo 10:** A Funcao Distancia Attitude-Gram

**input :** Attitude-Gram $A$ , $B$

**output :** Distancia entre dois Attitude-Grams

**Result:** O resultado sera a distancia Manhattan entre as atitudes.

```
tDistances[0] = A.Octave - B.Octave;
tDistances[1] = A.Bemois - B.Bemois;
tDistances[2] = A.DurMed - B.DurMed;
tDistances[3] = A.RNote - B.RNote;
tDistances[4] = A.UNote - B.UNote;
tDistances[5] = A.CNote - B.CNote;
tDistances[6] = A.MedBPM - B.MedBPM;
tDistances[7] = A.Attitude - B.Attitude;
for i ← 0 to 7 do
    tResultDistance += tDistances[i];
end
retorno = tResultDistance
```

# 4 Protocolo Experimental

O protocolo experimental desta pesquisa será realizado seguindo-se as três etapas de uma busca por similaridade Query-By, definidas na fundamentação teórica deste documento.

O objetivo principal deste protocolo experimental é possibilitar que as diversas técnicas de calculo de similaridade possam ser medidas e avaliadas seguindo os mesmos passos de execução independente do experimento a ser realizado.

Todas as informações sobre o Dataset Musical, Métricas de Avaliação utilizadas, bem como parâmetros de funcionamento foram distribuídos dentro desta estrutura de execução com o intuito de podermos criar uma conexão entre o item do protocolo experimental e em que momento/local do processamento experimental este item se relaciona.

Neste protocolo experimental estará sendo referenciado com frequência Músicas de Consulta e Músicas de comparação, portanto precisamos estabelecer primeiramente suas diferenças descritas abaixo:

**Música de Consulta -** "*Uma música de consulta neste trabalho pode ser qualquer fragmento musical, até mesmo uma música completa, que será enviada para ser comparada com as músicas contidas no dataset musical. Estes fragmentos de música buscam simular as situações que ocorrem durante uma performance musical.*"

**Música de Comparação -** "*Uma música de comparação neste trabalho é toda e qualquer música pré-processada contida no dataset musical de consulta, e que receberá uma pontuação de similaridade relativa a música de consulta, bem como em cada busca será classificada momentaneamente para avaliação dos resultados de uma busca.*"

Outro detalhe importante sobre o protocolo experimental deste trabalho, é que o mesmo se refere a uma execução única de uma busca Query-By-Playing, onde nos experimentos será detalhado o número de execuções e parametros para a realização do experimento.

## 4.1 Passos para realização de uma Busca Query-By

Para a realização de uma busca query-by é necessário a realização de alguns passos que caracterizam o funcionamento de uma busca completa.

Cada um destes passos pertence a uma determinada etapa de processamento divididas em Pré-Processamento, Processamento e Pós-Processamento conforme descritos na tabela 2.

Tabela 2 – Protocolo Experimental - atividades por etapa de processamento.

| **Passo** | **Descrição** | **Etapa** | **Há Repetição?** |
|---|---|---|---|
| Passo 01 | Carregar Dataset Musical. | Pré-Processamento | Não |
| Passo 02 | Criar Lista de Musicas para Consulta. | Pré-Processamento | Não |
| Passo 03 | Escolher Musica Para Consulta. | Pré-Processamento | Não |
| Passo 04 | Escolher Musica Para Comparação. | Processamento | Sim |
| Passo 05 | Comparar Musica A com a Musica B | Processamento | Sim |
| Passo 06 | Classificar Musica de Comparação. | Processamento | Sim |
| Passo 07 | Calcular Métricas da Busca. | Pós-Processamento | Não |

Fonte: Autoria Própria

Um detalhe importante relativo aos passos para a realização de uma busca Query-By, está na repetição de passos necessários até que uma busca completa seja realizada.

Para a etapa de Processamento cada uma das músicas presentes no dataset musical, será escolhida e processada , bem como a cada comparação de similaridade a musica selecionada(Música para Comparação) será classificada de acordo com a pontuação de similaridade obtida no processo de comparação.

### 4.1.1 Passos 1 a 3 - O Pré-Processamento das Buscas

Nesta etapa os passos 01 , 02 e 03 do processamento geral de uma busca (Definidos na Tabela 2) serão realizados.

Estes passos envolvem a extração e preparação de todas as informações musicais contidas no dataset musical.

Para a realização deste trabalho está sendo utilizado uma lista de 10.198 músicas tradicionais irlandesas em formato MIDI utilizadas no artigo (MARTINIANO; N. Silla, 2017).

Todas as músicas apresentam-se separadas em folders de gênero musical , onde cada música apresenta uma média de 350 notas por música e uma média de 1000 músicas por gênero musical.

Conforme definido na Tabela 3, Apenas a informação de Pitch e Duração da nota estão presentes em um arquivo MIDI, onde todas as outras informações precisam ser criadas , preparadas ou convertidas, para a utilização nas buscas de acordo com o algoritmo sendo utilizado.

Esta preparação é necessária, pois os diferentes algoritmos sendo utilizados utilizam dados diferentes para a etapa de processamento, onde os algoritmos SW Padrão e o Algoritmo utilizando a matriz de parametrização MUSSUM, utilizam notas musicais em formato de cadeias de caracteres e o algoritmo Mu6 utiliza cadeias de notas Multi-

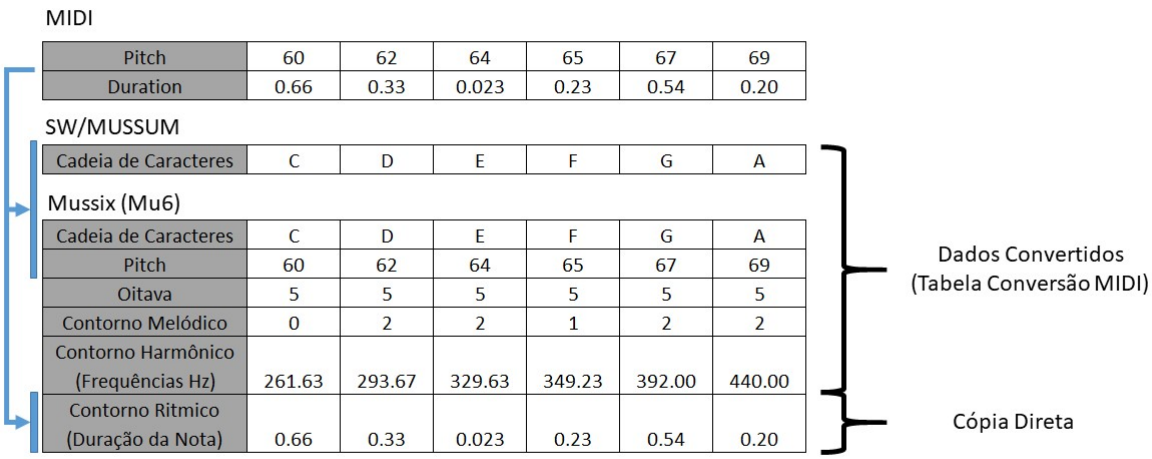
Tabela 3 – Informações Musicais - Conversão MIDI para utilização nos Algoritmos.

| **Campo** | **MIDI** | **MUSSUM/SW** | **Mu6** |
|---|---|---|---|
| Nota Caracteres | Não | Sim | Sim |
| Pitch | Sim | Não | Sim |
| Duração | Sim | Não | Sim |
| Oitava da Nota | Não | Não | Sim |
| Contorno Melódico | Não | Não | Sim |
| Contorno Ritmico | Não | Não | Sim |
| Contorno Harmônico | Não | Não | Sim |
| Attitude-Gram | Não | Não | Sim |

Fonte: Autoria Própria

Dimencionais, conforme Figura 11.

Figura 11 – Dados Musicais - Conversão MIDI.




Fonte: Autoria Própria

Para a realização desta conversão foi criada uma tabela fixa de conversão MIDI contendo todas as informações necessárias de uma nota, no qual para cada nota MIDI lida, será criada uma nota em uma cadeia de notas nova a ser utilizada posteriormente pelos algoritmos de alinhamento local, onde podemos visualizar na Tabela 4 as escalas 4,5 e 6 (Si , Dó , Ré respectivamente).

Tabela 4 – Mapeamento MIDI - Tabela de Conversão MIDI.

| **Pitch** | **Nota Caracter** | **Oitava** | **Frequência** |
|---|---|---|---|
| 48 | C | 4 | 130.81 |
| 49 | Db | 4 | 138.59 |
| 50 | D | 4 | 146.83 |
| 51 | Eb | 4 | 155.56 |
| 52 | E | 4 | 164.81 |
| 53 | F | 4 | 174.61 |
| 54 | Gb | 4 | 185.00 |
| 55 | G | 4 | 196.00 |
| 56 | Ab | 4 | 207.65 |
| 57 | A | 4 | 220.00 |
| 58 | Bb | 4 | 233.08 |
| 59 | B | 4 | 246.94 |
| 60 | C | 5 | 261.63 |
| 61 | Db | 5 | 277.18 |
| 62 | D | 5 | 293.67 |
| 63 | Eb | 5 | 311.13 |
| 64 | E | 5 | 329.63 |
| 65 | F | 5 | 349.23 |
| 66 | Gb | 5 | 369.99 |
| 67 | G | 5 | 392.00 |
| 68 | Ab | 5 | 415.30 |
| 69 | A | 5 | 440.00 |
| 70 | Bb | 5 | 466.16 |
| 71 | B | 5 | 493.88 |
| 72 | C | 6 | 523.25 |
| 73 | Db | 6 | 554.37 |
| 74 | D | 6 | 587.33 |
| 75 | Eb | 6 | 622.25 |
| 76 | E | 6 | 659.26 |
| 77 | F | 6 | 698.46 |
| 78 | Gb | 6 | 739.99 |
| 79 | G | 6 | 783.99 |
| 80 | Ab | 6 | 830.61 |
| 81 | A | 6 | 880.00 |
| 82 | Bb | 6 | 932.33 |
| 83 | B | 6 | 987.77 |

Fonte: Autoria Própria

#### 4.1.1.1 A Transposição das Músicas para escala Única

Além da conversão de notas MIDI para cadeias de caracteres e notas Multi-Dimensionais, como podemos ver na tabela 4 de conversão MIDI , uma nota convertida

para strings vinda de um arquivo MIDI pode estar presente em diversas escalas tonais (oitavas diferentes) causando falsos acertos dentro do algoritmo de alinhamento e causando falhas no cálculo de pontuação de similaridade.

Isto afeta as metricas de avaliação na comparação entre os algoritmos , onde foi decidido se transpor todas as músicas MIDI para a escala de Dó (Escala MIDI 05).

Utilizando uma mesma escala tonal permite a realização de uma comparação mais justa entre os algoritmos, já que os algoritmos de alinhamento baseados apenas na escala tonal de uma música (SW e MUSSUM) possuem esta limitação.

#### 4.1.1.2 A Categorização de Músicas

Além da conversão das cadeias de notas musicais , para cada folder detectado no dataset musical, foi definido uma categoria (com o mesmo nome do folder) no qual as músicas do dataset musical serão segmentadas conforme Tabela 5.

Tabela 5 – Categorias - Categorias de Músicas no Dataset Musical.

| **Nome Categoria** | **Identificador Categoria** |
|---|---|
| barndance | 0 |
| hornpipe | 1 |
| jig | 2 |
| mazurka | 3 |
| polka | 4 |
| reel | 5 |
| slide | 6 |
| slip-jig | 7 |
| strathspey | 8 |
| three-two | 9 |
| waltz | 10 |

Fonte: Autoria Própria

A idéia principal da categoria de uma música é definir grupos de música que estejam agrupadas de modo mais revelante, para que na etapa de processamento , mais precisamente na classificação de similaridade, a categoria da música possa ser utilizada para definir se uma música é ou não relevante.

A categoria de uma música é uma informação que será propagada também na criação das listas de teste para o experimento, onde esta informação é copiada para a música nova , derivada de uma música original.Isto possibilita que mesmo tendo uma música com uma série de notas randômicas, podemos determinar sua relevância através de sua origem.

Neste trabalho não há foco de se realizar a aprendizagem de máquina para a detecção da relevância de uma música, pois a mesma é dependente do contexto do problema (Identificação de Gêneros Musicais, por exemplo) a ser resolvido, portanto independente do problema ou forma como as musicas estão organizadas será considerado relevante uma música de consulta que possua a mesma categoria da música de comparação.

#### 4.1.1.3 A Montagem de Listas para Busca

Adicionalmente aos passos 01 e 02 desta etapa , o passo 03 envolve a seleção de uma música para consulta. A música de consulta tem papel importante no processo, pois busca simular uma situação real de Query-By-Playing onde um músico está tocando seu instrumento.

Para simular situações reais de performance musical e poder avaliar a reação dos diversos algoritmos de alinhamento em diversas situações, foram geradas músicas para consulta baseadas nas musicas originais do dataset musical variando-se o numero de notas musicais e incluindo-se notas randômicas nas mesmas.

Como o cerne de processamento da Mu6 é a inclusão dos contornos musicais para o processamento de alinhamento e devido a maioria das musicas seguir o padrão musical de tríades para a formação de uma tonalidade (Tônica , Terça e Quinta da escala sendo tocada) foi optado por se gerar randomicamente , a cada três notas da música original, uma nota randômica pertencente a escala diatônica ocidental conforme mostrado na Figura 12.

Para ser possível avaliar apropriadamente os resultados em conjunto com a geração randômica de notas , a música de consulta possuirá um número limite de notas de 10 a 100 notas para que trechos musicais diferenciados pelo tamanho sejam avaliados.

### 4.1.2 Passos 4 a 6 - O Processamento de Uma Busca

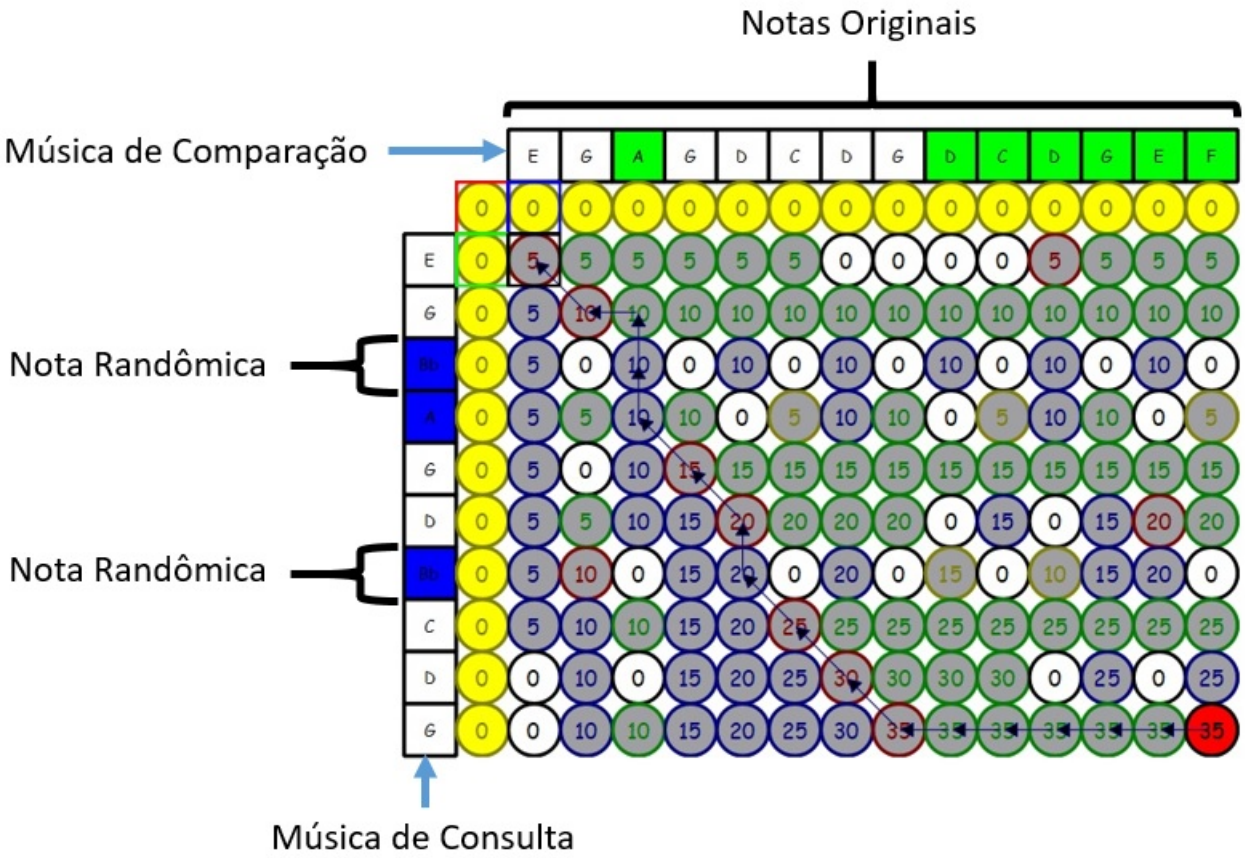
Na etapa de Processamento será executada recursivamente os passos 4,5 e 6 deste protocolo , envolvendo basicamente a seleção da Música de Comparação (Passo 4), Comparar a música selecionada no passo 03 com a Música de Comparação e após isto , através da pontuação obtida desta comparação, classificar a música de comparação.

Estes passos devem ser repetidos até que não haja mais músicas no dataset musical para comparação, ou seja, para uma unica busca estes passos serão repetidos 10.198 vezes.

#### 4.1.2.1 Passos para uma Comparação Musical

Analogamente aos passos necessários para este protocolo experimental, para podermos realizar uma comparação musical e assim obter a pontuação de similaridade , foi estabelecido alguns passos para sua realização.

Figura 12 – Notas Randômicas - Inclusão a cada Tríade.




Fonte: Autoria Própria

O processo de comparação Musical foi dividido em três passos principais respectivamente nomeados de Inicialização, Alinhamento e BackTracing.

#### 4.1.2.2 A Inicialização da Comparação

A primeira etapa de processamento de uma comparação é a inicialização de um alinhamento.

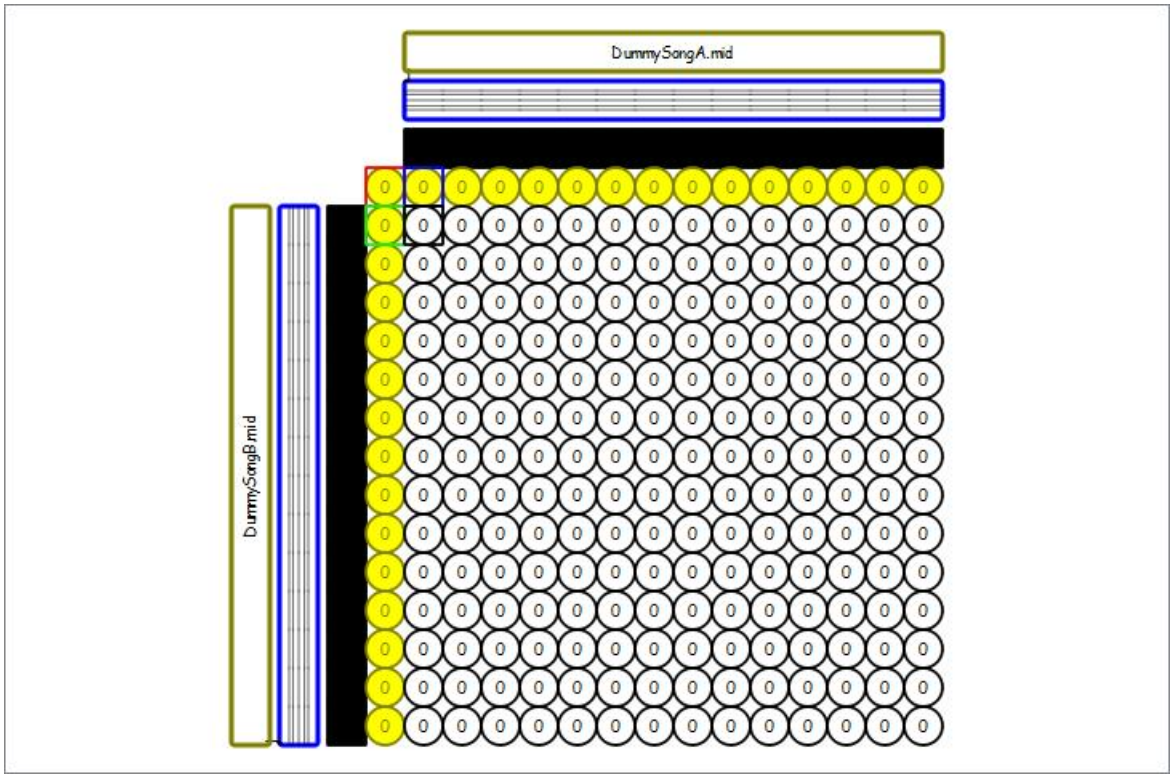
primordialmente a inicialização de um alinhamento visa configurar valores iniciais na matriz de pesos. No caso de um algoritmo de alinhamento, esta matriz é chamada de Substitution Matrix.

A inicialização da Matriz pode ser representada pelas seguintes formulas:

$$\begin{aligned} \text{M(i,0)} &= 0 \ , \ 0 \leq i \leq m \\ \text{M(0,j)} &= 0 \ , \ 0 \leq i \leq n \end{aligned} \tag{4.1}$$

Para um alinhamento local adequado, sempre a substitution matrix será inicializada com o Tamanho da música(A e B) + 1 , para garantir uma coluna e uma linha com valores zerados e assim evitarmos de realizar o teste de valores inválidos na borda da matriz conforme Figura 13.

Figura 13 – Inicialização de Uma Substitution Matrix.




Fonte: Autoria Própria

#### 4.1.2.3 O Alinhamento da Comparação

Após a inicialização da substitution matrix, iniciamos o caminhamento convolutivo do algoritmo de alinhamento local.

Para cada valor (i,j) contido na matriz,utilizando a Máscara de Processamento, executamos fórmula de convolução definida pelo algoritmo de alinhamento escolhido (SW , MUSSUM ou Mu6).

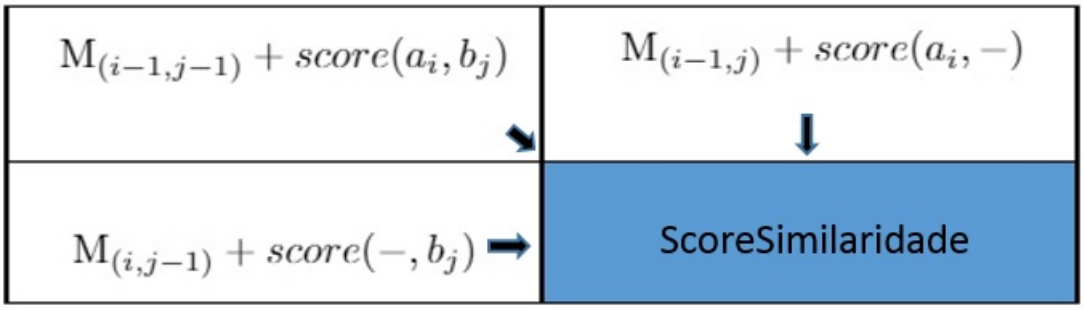
A diferença entre a execução de uma formula de convolução normal e a convolução realizada por um algoritmo de alinhamento, consiste em que para o alinhamento de sequencia cada valor da mascara de processamento(Núcleo de Convolução) é realizado dinâmicamente um calculo, baseado em parametros de processamento pré definidos(Match/Mismatch e Gap).

A formula de calculo da pontuação de similaridade do alinhamento de sequencia local pode ser representada da seguinte forma:

$$ScoreSimilaridade(i,j) = max \begin{cases} \mathrm{M}_{(i-1,j-1)} + s(a_i, b_j) \\ \mathrm{M}_{(i-1,j)} + s(a_i, -) \\ \mathrm{M}_{(i,j-1)} + s(-, b_j) \\ 0 \end{cases} \quad (4.2)$$

A transposição de cada uma destas formulas pode ser visualizada graficamente na Figura 14.

Figura 14 – Transposição de Formulas no Núcleo de Convolução.




Fonte: Autoria Própria

Uma atenção especial deve ser dada as formulas parciais $score(a_i, b_j)$ , $score(a_i, -)$ e $score(-, b_j)$ , responsáveis pelo teste de similaridade entre o valor da nota da música de consulta , com a nota da música de comparação.

Estas formulas parciais representam os valores de Match , Mismatch e Gap parametrizados no algoritmo de alinhamento e que em conjunto com a direção do valor máximo definem o calculo final da pontuação de similaridade para cada item da matriz.

A função $Max()$ utilizada no algoritmo de alinhamento pode ser representada através de pseudo código no algoritmo 11.

**Algoritmo 11:** A Funcao Max

**input :** A posicao atual $i$,$j$

**output :** Pontuação de Similaridade

**Result:** O resultado sera a pontuação maxima de similaridade.

```
lado = M[i-1,j ] + gap;
topo = M[i ,j-1] + gap;
diagonal = M[i-1,j-1] + ScoredeErro : Acerto;
retorno = Maximo(topo,lado,diagonal)
```

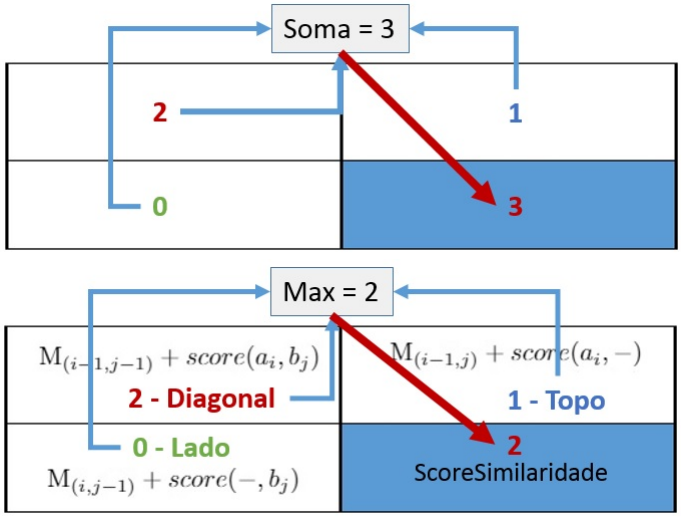
Outra diferença significativa entre um núcleo de convolução normal e um núcleo de convolução para alinhamento de sequências, é relativo a operação matemática realizada entre os vizinhos para a geração do valor resultante da convolução.

Em um calculo de convolução pela regra geral, o valor resultante da soma entre os seus vizinhos é atribuído ao centro do núcleo de convolução.

Em um calculo de convolução para o alinhamento de uma sequencia local, o valor resultante do maior valor entre os vizinhos, define o valor a ser atribuído ao centro do nucleo de convolução, e define a Direção em que este valor foi originado.Estas diferenças

podem ser visualizadas na Figura 15.

Figura 15 – Comparativo de Núcleo: a) Núcleo Normal , b) Núcleo Alinhamento




É baseado no registro da direção resultante de cada valor calculado pelo núcleo de processamento, que é possível posteriormente realizar o caminhamento reverso do algoritmo.

#### 4.1.2.4 Parametros de Processamento do Alinhamento

Como vimos anteriormente nesta seção, as formulas parciais de alinhamento são responsáveis pelos parâmetros the Match, Mismatch e Gap utilizados pelo algoritmo para o calculo da pontuação de similaridade projetada na substitution Matrix.

Portanto para cada busca sendo realizada pelo protocolo experimental, será escolhido de acordo com o experimento, qual é a parametrização a ser usada durante a busca.

Os três tipos de parametrização sendo utilizadas neste trabalho são a parametrização Smith-Waterman padrão (Fixa) , parametrização utilizando a Matriz MUSSUM e a parametrização utilizando a Mu6.

Dentro da teoria musical existem inumeras regras musicais que podem ser representadas através de uma matriz, como por exemplo , criarmos uma matriz de tamanho igual ao numero de notas da escala diatônica, e para cada valor desta matriz, calcularmos a distância em semi-tons entre uma e outra(Matriz MUSSUM). Adicionalmente podemos utilizar a diagonal principal da matriz para conter dados adicionais, que podem ser utilizados durante o processamento do cálculo de pontuação de similaridade.

A vantagem na utilização de matrizes ocorre devido a termos todos os valores pré-calculados sem que seja necessário o calculo durante o processamento do algoritmo, e dependendo da parametrização obtermos resultados de pontuação de similaridade melhores.

#### 4.1.2.4.1 Parametrização Fixa - Smith Waterman Original

A parametrização Fixa é a parametrização original Smith Waterman, ou seja, os valores utilizados nas formulas de calculo de pontuação de similaridade Match , Mismatch e Gap são determinados antes da execução do algoritmo e não mudam de acordo com o andamento convolutivo.

Os valores de parâmetros sendo utilizados neste trabalho podem ser vistos na tabela 6 , e conforme detalhado pelo autor em (SMITH; WATERMAN, 1981), para uma maximização dos resultados, um valor de match deve ser positivo, e os valores de Mismatch e Gap negativos.

Tabela 6 – Tabela de Parametrização - Smith-Waterman Padrão.

| **Parâmetro** | **Valor** |
|---|---|
| Match | 2 |
| Mismatch | -2 |
| Gap Penalty | -1 |

Fonte: Autoria Própria

#### 4.1.2.4.2 Parametrização MUSSUM

Conforme o autor em (MARTINIANO; N. Silla, 2017), a Matriz de parametrização MUSSUM foi definida da seguinte forma:

"Para um alinhamento mais preciso, propomos uma nova matriz de substituição chamada MUSSUM (MUSic SUBstitution Matrix). Os valores para a incompatibilidade foram calculados de acordo com a distância entre cada semitom da escala C maior. A diagonal principal foi calculada considerando a frequência de cada nota no banco de dados. As notas que obtiveram a maior frequência foram pontuadas com valores mais baixos do que aquelas com pequenas frequências. Isso foi feito porque a correspondência entre as notas que têm baixa frequência no banco de dados apresenta uma chance maior de ser a sequência desejada. A representação da matriz é mostrada na Figura 16."

Neste trabalho , quando ocorre uma incompatibilidade (Mismatch) , o valor da diagonal negativado está sendo atribuído para o parametro e o valor de penalidade está sendo atribuido de acordo com a distância de semi-tons definidos na MUSSUM.

#### 4.1.2.4.3 Parametrização Mussix (Mu6)

O Objetivo principal da utilização da matriz de parametrização Mu6 para o funcionamento do alinhamento local nesta pesquisa, foi a de podermos aproximar a

Figura 16 – A Matriz MUSSUM.

| | C | C# | D | D# | E | F | F# | G | G# | A | A# | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | 8 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -10 | -11 |
| C# | -1 | 8 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 | -10 |
| D | -2 | -1 | 4 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9 |
| D# | -3 | -2 | -1 | 10 | -1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 |
| E | -4 | -3 | -2 | -1 | 5 | -1 | -2 | -3 | -4 | -5 | -6 | -7 |
| F | -5 | -4 | -3 | -2 | -1 | 5 | -1 | -2 | -3 | -4 | -5 | -6 |
| F# | -6 | -5 | -4 | -3 | -2 | -1 | 6 | -1 | -2 | -3 | -4 | -5 |
| G | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 5 | -1 | -2 | -3 | -4 |
| G# | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 10 | -1 | -2 | -3 |
| A | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 4 | -1 | -2 |
| A# | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 10 | -1 |
| B | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 5 |

Fonte: Autoria Própria

pontuação de similaridade do algoritmo de alinhamento das caracteristicas musicais que estabelecem uma relação musical comum entre uma nota da música de consulta e uma nota da música de comparação.

De maneira diferente da MUSSUM, a matriz Mu6 não é um parâmetro único que é consultado durante o alinhamento, mas sim parâmetros contidos em cada uma das notas musicais, tanto da música de comparação , como na música de consulta.

Portanto, a matriz Mussix propriamente dita, é construída durante o pré-processamento da consulta e consiste na utilização de cadeias de notas Multi-Dimensionais que, durante o andamento convolutivo do algoritmo de alinhamento, são utilizadas para o cálculo de Match(Compatibilidade) , Mismatch(Incompatibilidade) e Gap Penalty(Penalidade).

#### 4.1.2.5 O Caminhamento reverso - Pontuação de Similaridade

Após concluído todo o caminhamento convolutivo da substitution matrix (matriz de convolução) , é realizado o caminhamento reverso dos itens desta matriz para a construção do melhor alinhamento local resultante do processamento do algoritmo.

É nesta etapa do processamento em que a pontuação de similaridade que será atribuída á musica de comparação.

Para este trabalho, a soma simples de acertos no alinhamento final será feita sem a necessidade de processarmos os outros tipos de ocorrencia presentes no algoritmo de alinhamento (Inclusão / Alteração e Exclusão).

Como o objetivo é verificar qual música é mais similar, a quantidade de ocorrências de igualdades é mais relevante do que a identificação das operações necessárias para transformar a música de consulta , na música de comparação.

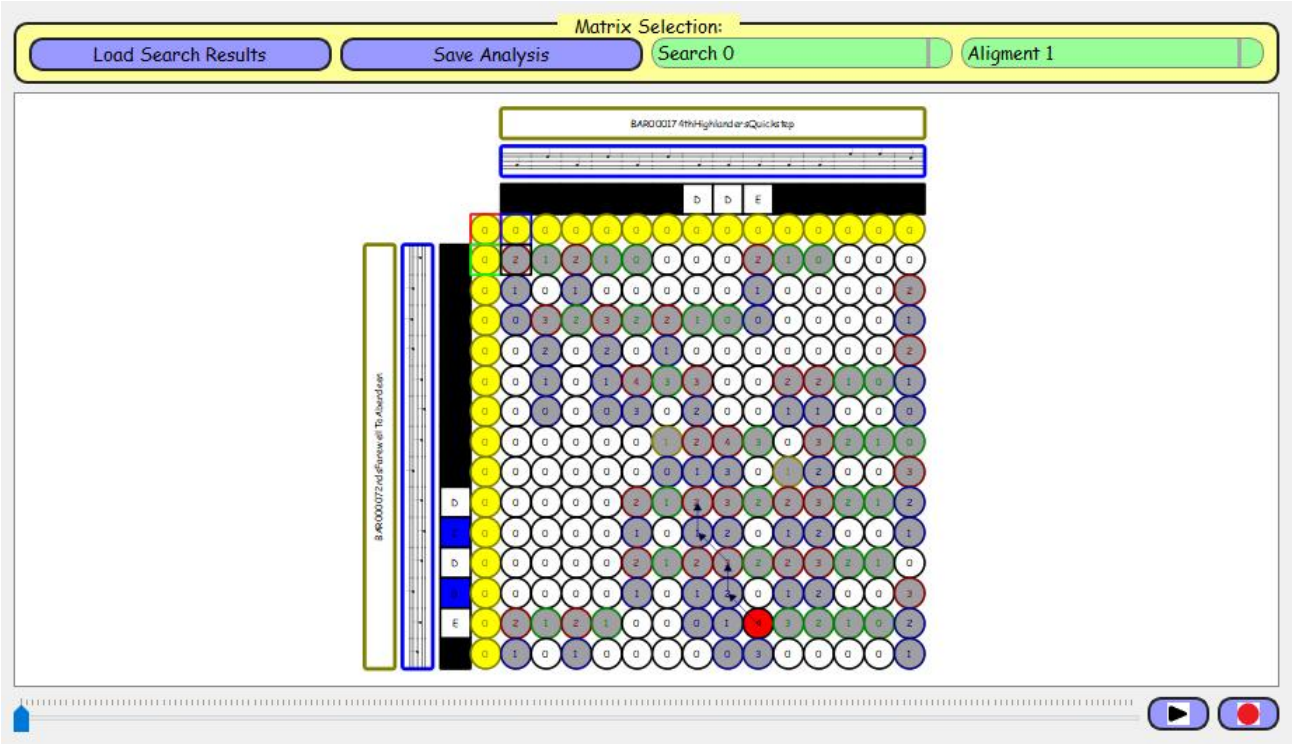
Isto é realizado através da execução sequencial de 3 passos:

1. Obtenção do maior score contido na matriz.
2. A partir da posição do maior item, percorrer a matriz utilizando as direções coletadas durante o processamento, até que se encontre um valor de score zerado(Caminho Bloqueado).
3. Calcular o Score Final de Similaridade da música de comparação atribuindo 1 ponto para cada valor do alinhamento resultante de um Match(Diagonal).

Estes 3 passos podem ser representados através de pseudo-código no algoritmo 12 utilizado para a realização do caminhamento reverso.

Para facilitar o entendimento do processamento e verificação do seu funcionamento, foi criada uma animação passo a passo do algoritmo Smith-Waterman entre duas musicas e que pode ser verificado na Figura 17 (Vídeo Youtube).

Figura 17 – Animação de Alinhamento Local - Algoritmo Smith-Waterman(SW).




Fonte: <https://youtu.be/E_lMlcPMmuE>

Esta animação demonstra todo o caminho percorrido pela máscara de processamento de alinhamento local e a atualização ocorrida em cada um dos elementos da matriz de substituição utilizada pelo algoritmo.

**Algoritmo 12:** O Algoritmo de Caminhamento Reverso

**Result:** Construção do Alinhamento Local.

```
81  GetMax(IdxMaxX,IdxMaxY);
82  while !TraceEnd do
83      IScore = GetVal(IdxMaxX, IdxMaxY, iDirection);
84      switch (iDirection) do
85          case Diagonal do
86              IdxMaxX-=1; IdxMaxY-=1;
87          end
88          case Lado do
89              IdxMaxY-=1;
90          end
91          case Topo do
92              IdxMaxX-=1;
93          end
94      end
95  end
96  CalculaScore();
```

A função que retorna o maior valor na matriz de substituição pode ser verificada no algoritmo 13 representado também através de pseudo código.

**Algoritmo 13:** A Função GetMax

**Result:** Retorna os Indices X,Y com o Maior Valor na Matriz.

```
97  iMaxValue = 0;
98  iMaxValueX = 0;
99  iMaxValueY = 0;
100 for j ← 0 to SizeY do
101     for i ← 0 to SizeX do
102         if Matrix[Idxj][Idxi].iValue ≥iMaxValue then
103             iMaxValue = Matrix[Idxj][Idxi].iValue;
104             iMaxValueX = Idxi;
105             iMaxValueY = Idxj;
106         end
107     end
108 end
109 //Retorno;
110 x = iMaxValueX;
111 y = iMaxValueY;
```

Finalmente a função que calcula o score final, percorre o caminho traçado no pos-processamento e atribui 1 ponto a cada Match encontrado (Representado no algoritmo 14).

**Algoritmo 14:** A Função CalculaScore

**Result:** Calcula e Retorna a Pontuação de Similaridade Final.

```
iTraceScore = 0;
for i ← 0 to iPathSize do
    if vTraceBack[i].iDirection == Diagonal then
        iTraceScore++;
    end
end
```

#### 4.1.2.6 A Classificação de uma música de comparação

A classificação de uma música de comparação é o último passo da etapa de Processamento de uma busca e seu objetivo principal é, através da pontuação de similaridade e da categoria da música , classificar cada uma das músicas de consulta como TP - Verdadeiro Positivo, FP - Falso Positivo , TN - Verdadeiro Negativo ou FN - Falso Negativo.

Para realizar esta classificação é utilizado a categoria da música para definir se a música é relevante ou não através de um teste de igualdade entre as duas músicas.

Após a verificação de relevância da música para determinarmos a classificação final da música é realizado um cálculo que está sendo chamado neste trabalho de Balanço de Precisão conforme definido abaixo:

**Balanço de Precisão -** "O balanço de Precisão é um cálculo de percentual de acertos relativo a música de consulta, que define um limiar entre músicas TP - Verdadeiro Positivo e FN - Falso Negativo e Músicas FP - Falso Positivo e TN - Verdadeiro Negativo."

Simplificando é o balanço de precisão (De zero(0%) a Cem(100%)) que , em conjunto com a categoria de uma música, irá classificar a música de comparação, para futuramente calcularmos as métricas de avaliação.

Após o cálculo da classificação da música, a mesma deve ser incluída em uma fila de prioridades , para o controle da quantidade de músicas que serão retornadas e que será utilizada para o cálculo das métricas de avaliação.

O objetivo principal de incluir as músicas em uma lista de prioridades é , além de controlar o numero de músicas final, não precisar ordenar as músicas pelo score no pós-processamento.

O processamento da classificação de uma música de comparação pode ser definido através do algoritmo 15.

**Algoritmo 15:** Determina a Classificação do Score

```
118 if A.Categoria = B_i.Categoria then
119     if SearchScore >= (B_i.NumeroNotas * BalançoPrecisão)/100 then
120         B_i.Classificação ← TP
121         AdicionaFilaMusicasRetorno(B_i.Score)
122     else
123         B_i.Classificação ← FN
124         AdicionaFilaMusicasRetorno(B_i.Score)
125 else
126     if SearchScore >= (B_i.NumeroNotas * BalançoPrecisão)/100 then
127         B_i.Classificação ← FP
128         AdicionaFilaMusicasRetorno(B_i.Score)
129     else
130         B_i.Classificação ← TN
131         AdicionaFilaMusicasRetorno(B_i.Score)
```

### 4.1.3 Passo 07 - O Pós-Processamento de uma Busca

Após toda a fase de processamento ter sido concluída iniciamos a ultima etapa de processamento.

O Pós-Processamento neste trabalho é responsável em realizar todos os cálculos de métricas utilizados nesta pesquisa e utilizam a classificação obtida na etapa de processamento para realizar estes cálculos.

As métricas de avaliação calculadas neste trabalho são:

- Precisão , Cobertura e Média Harmônica
- Acurácia
- Ranking Recíproco Médio.
- Tempo Percorrido (Total , Pré Processamento,Processamento e Pós-Processamento)

A escolha na utilização de diversas métricas de avaliação ocorre com o objetivo de avaliar não somente a precisão de uma busca, mas também medir em qual posição um acerto ocorre, bem como podermos avaliar os impactos que mudanças no algoritmo afetam as diversas medidas.

A sumarização de classificações pode ser especificada conforme o algoritmo 16.

**Algoritmo 16:** A Função Calculo de Métricas

**Result:** Calcula as metricas de Avaliação.

```
132 for i ← 0 to MusicasRetornadas do
133     if MusicasRetornadas[i].Classificação == TP then
134         QtdTP++;
135     end
136     if MusicasRetornadas[i].Classificação == FP then
137         QtdFP++;
138     end
139     if MusicasRetornadas[i].Classificação == TN then
140         QtdTN++;
141     end
142     if MusicasRetornadas[i].Classificação == FN then
143         QtdFN++;
144     end
145     if MusicasRetornadas[i].Nome == MusicaConsulta.NomeOriginal then
146         MRR = 1/position;
147     end
148 end
149 Precisão = ((QtdTP/(QtdTP + QtdFP)));
150 Cobertura = ((QtdTP/(QtdTP + QtdFN)));
151 MediaHarmonica = (2 * Precisão * Cobertura)/(Precisão + Cobertura);
152 Acuracia = (QtdTP + QtdFP)/(QtdTP + QtdFP + QtdTN + QtdFN);
```

Todas estas medidas referem-se a apenas uma busca realizada, onde as medidas finais de avaliação serão calculadas de acordo com o experimento calculando a media obtida destas métricas.

# 5 Avaliação dos Resultados

Neste capítulo serão apresentados e discutidos os resultados da utilização da matriz Mu6 em comparação com a utilização do algoritmo Smith-Waterman original e utilizando a matriz MUSSUM.

Para a avaliação de resultados os seguintes itens foram foco de atenção durante os experimentos.

- Quais são as diferenças de pontuação de similaridade resultante em cada algoritmo?
- Quais são os reflexos nas métricas de avaliação com a variação do balanço de precisão?
- Quais são os reflexos nas métricas de avaliação com a variação do tamanho de uma música de consulta?
- Quais são os tempos demandados por cada etapa da busca?
- Quais foram as músicas retornadas na lista final? Ocorreram coincidências?

## 5.1 Experimentos

Foram desenhados alguns experimentos específicos para o levantamento das métricas e respostas as questões a serem avaliadas na análise de resultados. Estes experimentos foram executados em uma ferramenta desenvolvida para este trabalho chamada QBPLab, com o intuito de possibilitar a reprodução destes resultados e eventual re-análise , caso necessário.

Cada busca realizada nestes experimentos seguiu rigorosamente todas as etapas do protocolo experimental e suas métricas serão avaliadas de acordo com o objetivo de cada experimento.

Todas as métricas de avaliação coletadas nas buscas serão consolidadas em um relatório único de médias resultantes de processamento.

### 5.1.1 Experimento 1 - Avaliação de Metricas

Neste experimento o foco principal é poder extrair as diversas métricas calculadas neste trabalho para avaliação.

Para tal este experimento consiste em executar 4 buscas Query-By-Playing por algoritmo utilizando 4 músicas de consulta selecionadas aleatoriamente.

O Balanço de precisão a ser utilizado é de 80% do Tamanho total da Música de comparação, e o tamanho do buffer deve variar entre 20 a 80 notas a cada consulta respectivamente.

Após a execução de todas as buscas , coletar as metricas obtidas e calcular a média de cada uma em um relatório único para avaliação.

### 5.1.2 Experimento 2 - Comparação de Pontuação de Similaridade

Após uma Consulta efetuada, uma lista de músicas resultante ordenadas pela pontuação de similaridade é retornada, e cada uma destas músicas possui uma pontuação específica.

A tarefa deste experimento é realizar 1(uma) busca para cada algoritmo , escolhendo a mesma música original e coletar esta lista de músicas resultante com suas respectivas pontuações e classificações.

O objetivo principal deste experimento é poder comparar a pontuação de similaridade e classificação atribuida na lista de músicas de retorno para cada algoritmo utilizado.

### 5.1.3 Experimento 3 - Variação de Balanço de Precisão

A tarefa deste experimento é realizar diversas buscas, utilizando músicas de consultas aleatórias, alterando o balanço de precisão de 0 até 100 , para cada algoritmo.

A idéia deste experimento é poder identificar os reflexos na precisão da busca com a alteração deste cálculo, pois o mesmo é responsável pela classificação das músicas de consulta sendo processadas.

Simplificando, o objetivo principal é identificar em qual balanço de precisão cada algoritmo apresenta os resultados mais precisos, bem como determinar se o mesmo mantém esta precisão a partir de alguma medida de balanço.

### 5.1.4 Experimento 4 - Análise de Tempo

A tarefa deste experimento é realizar diversas consultas , utilizando o balanço de precisão que gerou a maior precisão para todos os algoritmos (Obtido no Experimento 3).

Para cada etapa de processamento de uma busca, o tempo inicial e final de processamento deve ser calculado, ou seja , O tempo geral de processamento, o tempo demandado para o Pré-Processamento, bem como para o Processamento e Pós-Processamento.

Após todas as buscas serem executadas deve-se calcular a media de tempo subdividida por algoritmo e etapa de processamento.

## 5.2 Resultados Obtidos

Após a execução dos experimentos foram coletados todos os resultados e confeccionados os relatórios contendo as métricas de avaliação.

### 5.2.1 Resultado 01 - Metricas de Avaliação

Todas as buscas apresentaram diferenças significativas relativo as metricas de avaliação.

Notou-se uma diferença significativa na Precisão de buscas utilizando a Mu6 em relação a buscas utilizando SW e MUSSUM.

Notou-se também que aparentemente precisão e acurácia foram mutuamente exclusivas comparando-se algoritmos SW/MUSSUM e Mu6.

Foi elaborado relatório contendo as medias das métricas de avaliação para cada um dos algoritmos avaliados na Figura 18.

Figura 18 – Métricas de Avaliação - Médias de Avaliação.

Fonte: Autoria Própria

### 5.2.2 Resultado 02 - Comparação de Pontuação

O resultado obtido através da execução do experimento 2 é a lista de músicas resultante do processamento de cada algoritmo juntamente com sua respectiva pontuação de similaridade e classificação.

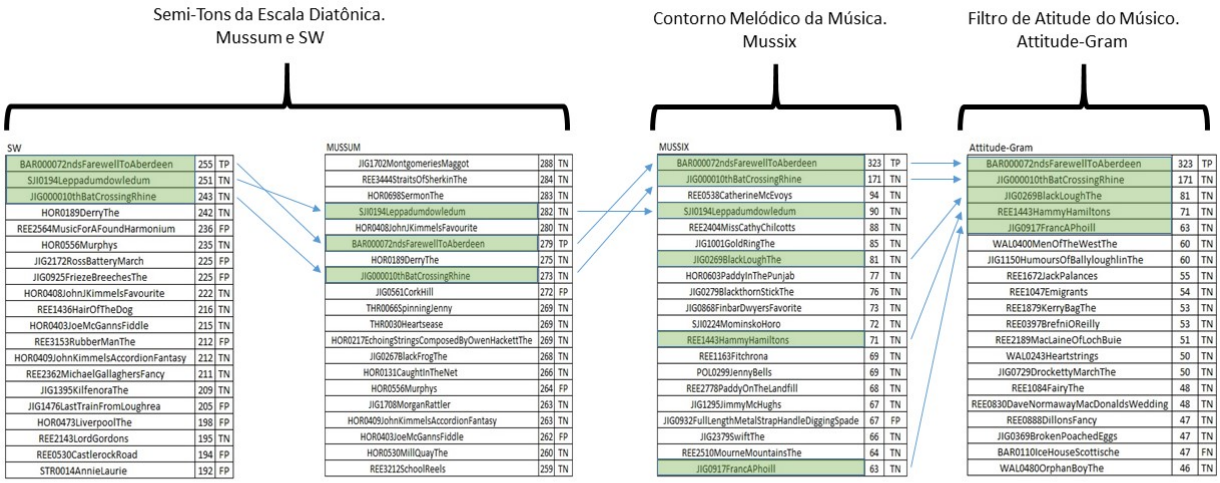
A Música aleatória de comparação escolhida para este experimento foi a música original ***Barndance*** , e a mesma foi utilizada para a realização das buscas com o algoritmo SW, MUSSUM , Mu6 e Mu6/Attitude-Gram.

O Balanço de Precisão utilizado para as buscas foi de 50% e o tamanho da Música de comparação foi a mesma da música original.

O Objetivo principal deste experimento é identificar as diferenças de pontuação entre os diversos algoritmos utilizados, bem como quais são as coincidências que ocorrem, já que é sabido que o algoritmo smith-waterman sempre retorna o melhor alinhamento local possível.

Para demostrar os resultados obtidos neste experimento foram desenhadas 4 listas de músicas, uma ao lado da outra representadas na Figura 19

Figura 19 – Experimento 02 - Comparação de Pontuação.




Fonte: Autoria Própria

Foi realizada uma checagem item a item de cada lista e realizado a ligação entre as mesmas em caso de coincidência de resultado.

Neste experimento foi detectado uma sensibilidade no algoritmo MUSSUM relativo

ao balanço de precisão utilizado , causando queda na precisão dos resultados, onde comparativamente com o SW, o algoritmo utilizando a matriz MUSSUM teve resultados MRR piores.

Relativo ao valor da pontuação de similaridade e ocorrência de duplicidades, foi detectado uma diferença significativa entre o valor de pontuação provindo da Mu6 e as pontuações retornadas pelo SW e MUSSIX.

Em todas as comparações a pontuação retornada pela Mu6, o decrécimo de pontuação apresentou-se acentuado indo da maior pontuação 323 para a segunda com 171 e após isto pontuações decrescendo até 46 na última música.

As pontuações retornadas pelos algoritmos SW e MUSSUM seguiram um mesmo perfil de decrécimo onde no SW o maior valor foi de 255 indo para 251 na segunda música decrementando suavemente a pontuação até um valor mínimo de 192.

Comparativamente podemos visualizar estas diferenças através dos quadros comparativos de pontuação da Figura 20.

Figura 20 – Experimento 02 - Decrescimo de Pontuação por Algoritmo.

Fonte: Autoria Própria

Foi detectado também diferenças significativas relativo as músicas retornadas pelos algoritmos, onde as musicas retornadas pelos algoritmos SW e MUSSUM, com excessão das musicas de alta pontuação de similaridade, são diferentes das retornadas pela Mu6.

Adicionalmente também foi detectado diferença significativa relativo as músicas retornadas pelo algoritmo Mu6, e as musicas retornadas utilizando a combinação Mu6/Attitude-Gram, onde as músicas com maior score acabam permanecendo na primeira posição e músicas diferentes completam o grupo de músicas retornado.

### 5.2.3 Resultado 03 - Variação de Balanço de Precisão

Após a execução do experimento 03 foi confeccionado um relatório evolutivo da precisão das buscas a medida que o balanço de precisão é alterado.

Para a execução deste experimento utilizando trechos de 15 notas com músicas originais notou-se uma evolução consistente, onde a matrix Mu6 precisou de um balanço de

precisão menor para atingir resultados de precisão máxima conforme podemos visualizar na Figura 21

Figura 21 – Experimento 03 - Evolução Musicas Originais 15 Notas.

Fonte: Autoria Própria

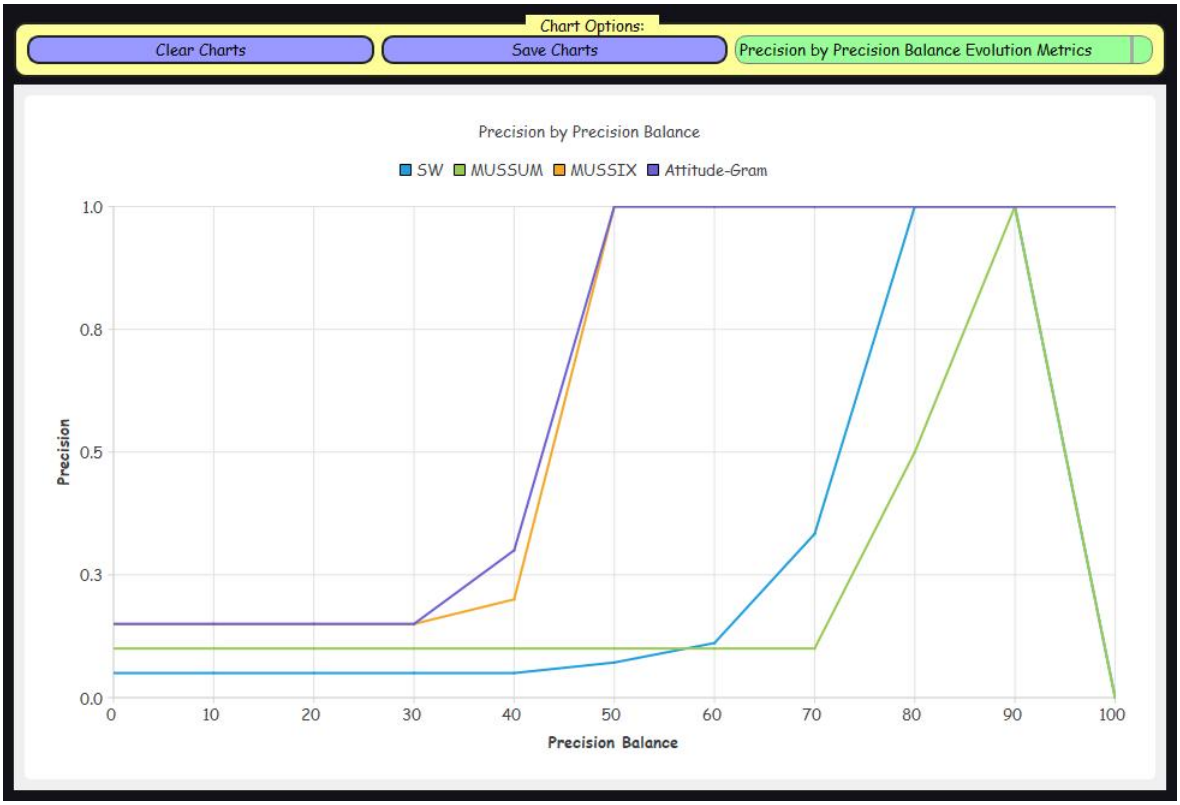
Após isto foi executado o mesmo experimento utilizando notas randômicas a cada 3 intervalos em uma musica completa e notou-se durante a evolução da execução uma instabilidade na precisão no algoritmo que utiliza a matriz MUSSUM (Acima de 70% a precisão cresce até chegar ao máximo com 90%, onde cai até zero de 90% a 100%).

O algoritmo Mu6 utilizou um balanço de precisão menor (50%) para atingir a precisão máxima nas buscas e manteve este resultado por toda a evolução do balanço de precisão, conforme podemos verificar na Figura 22.

Como resultado deste experimento foi possível notar que na média os resultados retornados pela utilização da Mu6 apresentam-se mais consistentes relativos a precisão de resultados, porém em contrapartida os resultados de acurácia parecem mutuamente exclusivos quando comparados aos resultados SW e MUSSUM.

Percebe-se também que a utilização de notas randômicas , mesmo com precisão de 90%, tanto o algoritmo SW quanto o algoritmo utilizando a matriz MUSSUM algumas consultas retornando ranking recíproco menor que 100% , onde a Mu6 apresentou Ranking

Figura 22 – Experimento 03 - Evolução Musicas Randomicas Completas.




Fonte: Autoria Própria

Máximo em todas as consultas.

### 5.2.4 Resultado 04 - Análise de Tempo

Com os resultados obtidos do experimento 03 foi notado que o balanço de precisão de 90% gerou uma precisão de 100% na lista de músicas final para todos os algoritmos utilizados.

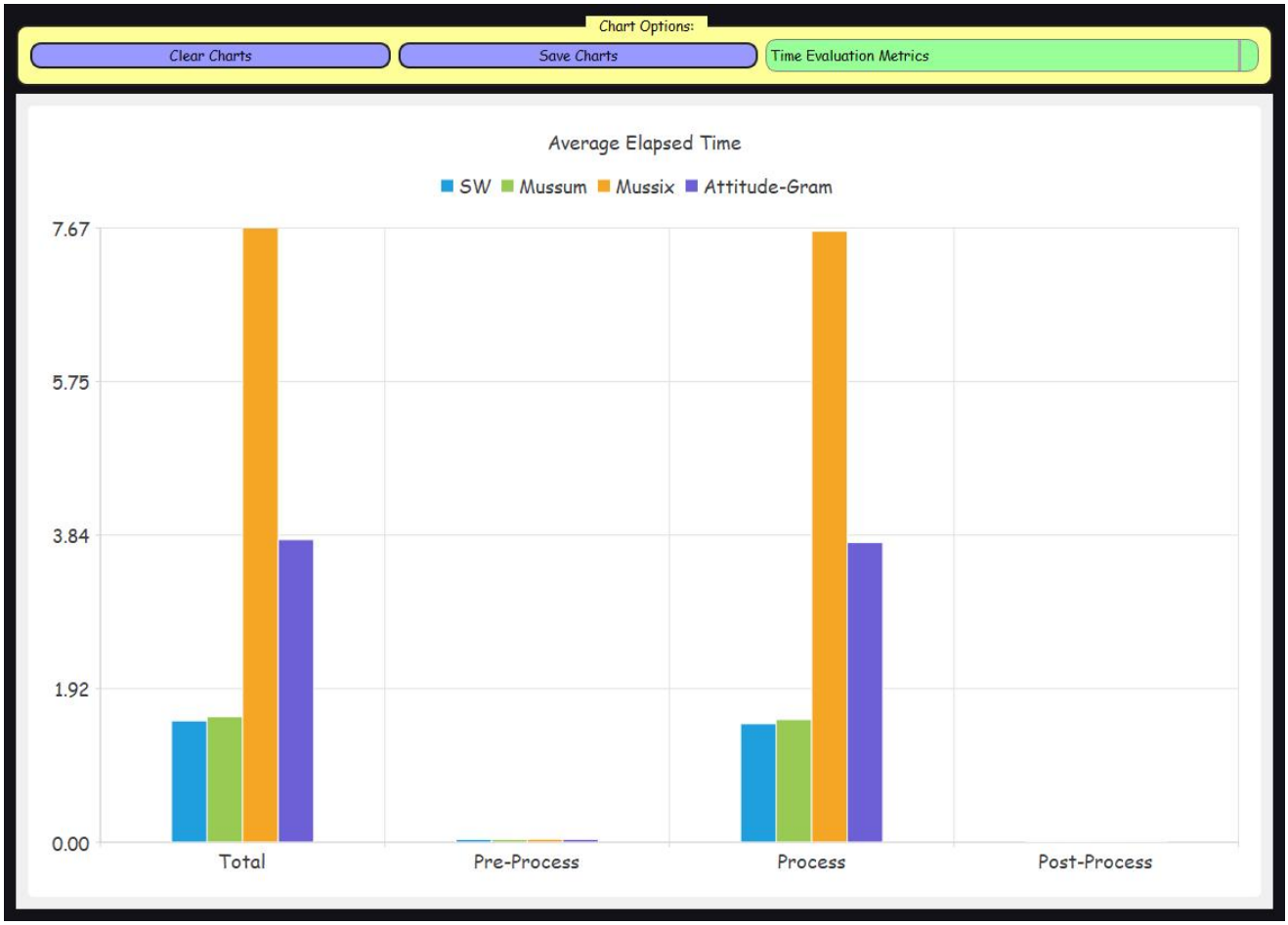
Portanto para este experimento foi configurado um balanço de precisão de 90% e foram executadas 4 buscas por algoritmo , utilizando musicas de consulta escolhidas aleatoriamente.

Após a execução das buscas, todos os tempos de execução foram coletados e foi gerado o relatório de análise de Tempo da Figura 23.

Notou-se nestes resultados que a grande parte do tempo demandado foi utilizado na etapa de processamento, onde o algoritmo Mu6 consumiu mais que o dobro de tempo para realizar o cálculo de similaridade, mostrando-se mais complexo que os algoritmos SW e MUSSUM.

Foi notado também que além de melhorar os resultados do processamento Mu6, a utilização da filtragem Attitude-Gram reduziu em mais de 50% o tempo demandado na

Figura 23 – Métricas de Avaliação - Avaliação de Tempo.




Fonte: Autoria Própria

utilização Mu6.

Não foi notado diferenças significativas de diferença de tempo de execução entre os algoritmos SW e MUSSUM , onde aparentemente demonstram a mesma complexidade de processamento.

## 5.3 Análise dos Resultados

Após a execução de todos os experimentos e a coleta de todos os resultados foi possível realizar uma análise dos resultados em vista das questões foco deste trabalho.

As diferenças de pontuação de similaridade mostraram uma diferença significativa em termos de precisão , onde através da análise do decréscimo de pontuação entre uma busca e outra, a Mu6 mostrou-se mais precisa em toda a lista de músicas, não somente na primeira música retornada.

Através da variação do Balanço de precisão também foi identificado que a Matriz Mu6 precisa de menos notas para atingir uma precisão máxima e que os algoritmos SW

e MUSSUM além de precisarem de mais notas para atingirem uma precisão máxima mostraram-se instáveis na precisão dependendo do balanço adotado.

Por exemplo, no caso da utilização da Matriz MUSSUM, qualquer balanço de precisão menor que 90% ou maior que 90% , irá gerar resultados menos precisos, onde com a utilização da Matriz Mu6 , a partir de 50% de balanço a precisão aumenta chegando ao máximo em 80% mantendo-se assim até o balanço máximo de precisão.

O mesmo resultado aconteceu quando variou-se os tamanhos das músicas de consulta, onde a matriz Mu6 precisou de um número menor de notas para chegar na precisão máxima.

Relativo ao tempo de processamento demandado pela Mu6 , os resultados tornaram-se promissores quando aliado ao Attitude-Gram atingiu-se um tempo de processamento 50% menor aproximando-se dos tempos utilizados pelos algoritmos SW e MUSSUM.

Por último, relativo as músicas retornadas pelas buscas, todos os algoritmos mostraram resultados diferentes de músicas, onde coincidiram apenas as quatro primeiras músicas, onde ocorreu a dificuldade de não haver um Ground-Thru para validar se as músicas realmente são mais similares ou não sob a perspectiva de um músico.

Portanto, na média de todas as execuções de buscas nos experimentos a matriz Mu6 mostrou-se 50% mais precisa em comparação com os algoritmos SW e MUSSUM atingindo os resultados esperados por este trabalho.

# 6 Considerações Finais

## 6.1 Relevância do Estudo

## 6.2 Contribuições da Pesquisa

## 6.3 Limitações da Pesquisa

## 6.4 Trabalhos Futuros

# Referências


CHUAN, C. H. A multimodal approach to song-level style identification in pop/rock using similarity metrics. *2013 12th International Conference on Machine Learning and Applications*, v. 2, p. 321–324, 2013. Disponível em: <http://ieeexplore.ieee.org/document/6786128/>. Citado 3 vezes nas páginas 6, 16 e 21.

DEGANI, A. et al. A Heuristic for Distance Fusion in Cover Song Identification. *2013 (WIAMIS)*, 2013. Citado na página 6.

DOWNIE, J. S. et al. The music information retrieval evaluation eXchange: Some observations and insights. *Studies in Computational Intelligence*, v. 274, p. 93–115, 2010. ISSN 1860949X. Citado na página 11.

DUREY, A.; CLEMENTS, M. Direct estimation of musical pitch contour from audio data. *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, v. 5, p. V–561–4, 2003. ISSN 1520-6149. Disponível em: <http://ieeexplore.ieee.org/document/1200031/>. Citado na página 14.

FERRARO, P.; HANNA, P. Optimizations of local edition for evaluating similarity between monophonic musical sequences. *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, n. 1970, p. 64–69, 2007. Citado 4 vezes nas páginas 7, 9, 11 e 15.

Kuldeep Gurjar, Y.-S. M. A comparative analysis of music similarity measures in music information retrieval systems. *Journal of Information Processing Systems · January 2018*, n. January, 2018. ISSN 2092805X. Citado na página 4.

LI, B.; DUAN, Z. An Approach to Score Following for Piano Performances with the Sustained Effect. *IEEE/ACM Transactions on Audio Speech and Language Processing*, v. 24, n. 12, p. 2425–2438, 2016. ISSN 23299290. Citado na página 5.

MARTINIANO, L.; N. Silla, C. BIRITS: A Music Information Retrieval System Using Query-by-Playing Techniques. In: *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2017. p. 535–542. ISBN 978-1-5386-3876-7. Disponível em: <https://ieeexplore.ieee.org/document/8371990/>. Citado 8 vezes nas páginas 4, 5, 6, 13, 17, 18, 26 e 35.

MARVIN, E. W. A generalized theory of musical contour: its application to melodic and rhythmic analysis of non-tonal music and its perceptual and pedagogical implications. 1988. Disponível em: <http://www.mendeley.com/research/a-generalized-theory-of-musical-contour-its-application-to-melodic-and-rhythmic-analysis-of-nontonal>. Citado 2 vezes nas páginas 14 e 15.

NEEDLEMAN, S. B.; WUNSCH, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, v. 48, n. 3, p. 443–453, 1970. ISSN 00222836. Citado na página 7.

PARK, M. W.; LEE, E. C. Similarity measurement method between two songs by using the conditional euclidean distance. *WSEAS Transactions on Information Science and Applications*, v. 10, n. 12, p. 381–388, 2013. ISSN 17900832. Citado na página 5.

SMITH, T.; WATERMAN, M. Identification of common molecular subsequences. *Molecular Biology*, v. 147, p. 195–197, 1981. ISSN 0022-2836. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/7265238>. Citado 3 vezes nas páginas 9, 18 e 35.

WANG, T.-M.; TSAI, P.-Y.; SU, A. W.-Y. Note-based alignment using score-driven non-negative matrix factorisation for audio recordings. *IET Signal Processing*, v. 8, n. 1, p. 1–9, 2014. ISSN 17519675. Disponível em: <http://digital-library.theiet.org/content/journals/10.1049/iet-spr.2012.0157>. Citado na página 11.