



Query by Playing

O Problema

Será que existe uma maneira de fazer o computador parecer entender a arte musical de um músico?

Buscar Músicas em uma Base... tocando uma música....

Será Possível?

O Objetivo

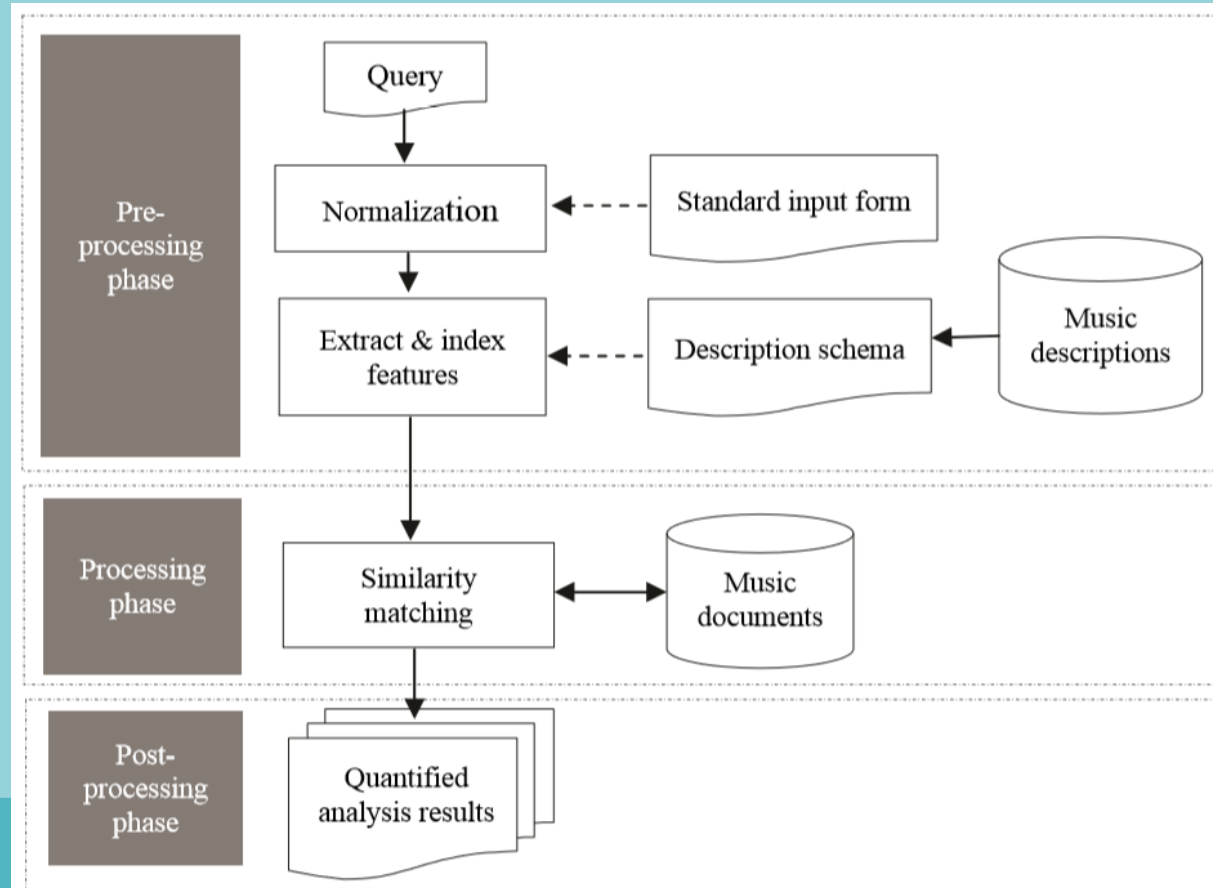
Este trabalho surgiu da curiosidade em saber como pode-se fazer que o computador possa entender a arte musical, onde o usuário possa ter a sensação de que realmente o sistema está compreendendo sua música.

O principal aspecto desta inteligência é o computador conseguir disponibilizar ao músico as músicas mais similares ao que ele está tocando.

Music Information Retrieval - MIR

No meio Científico e Acadêmico, a área de pesquisa responsável pelo estudo deste tipo de busca é chamado de Music Information Retrieval – MIR.

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



Query by?

Em **Music Information Retrieval (MIR)**, um tópico bastante ativo atualmente na comunidade acadêmica é a realização de Buscas de Músicas por Similaridade.

Em cada tipo de busca há diferentes tipos de Input destes dados , bem como o processamento de similaridade necessário também.

Tipos Comuns de Busca:

- Query by Humming – Busca de Músicas “Cantarolando”
- Query by Example – Busca de Músicas utilizando Músicas de Exemplo.
- Query by Tapping – Buscas de Músicas com a mesma “Batida” rítmica.

QBP – Query by Playing?

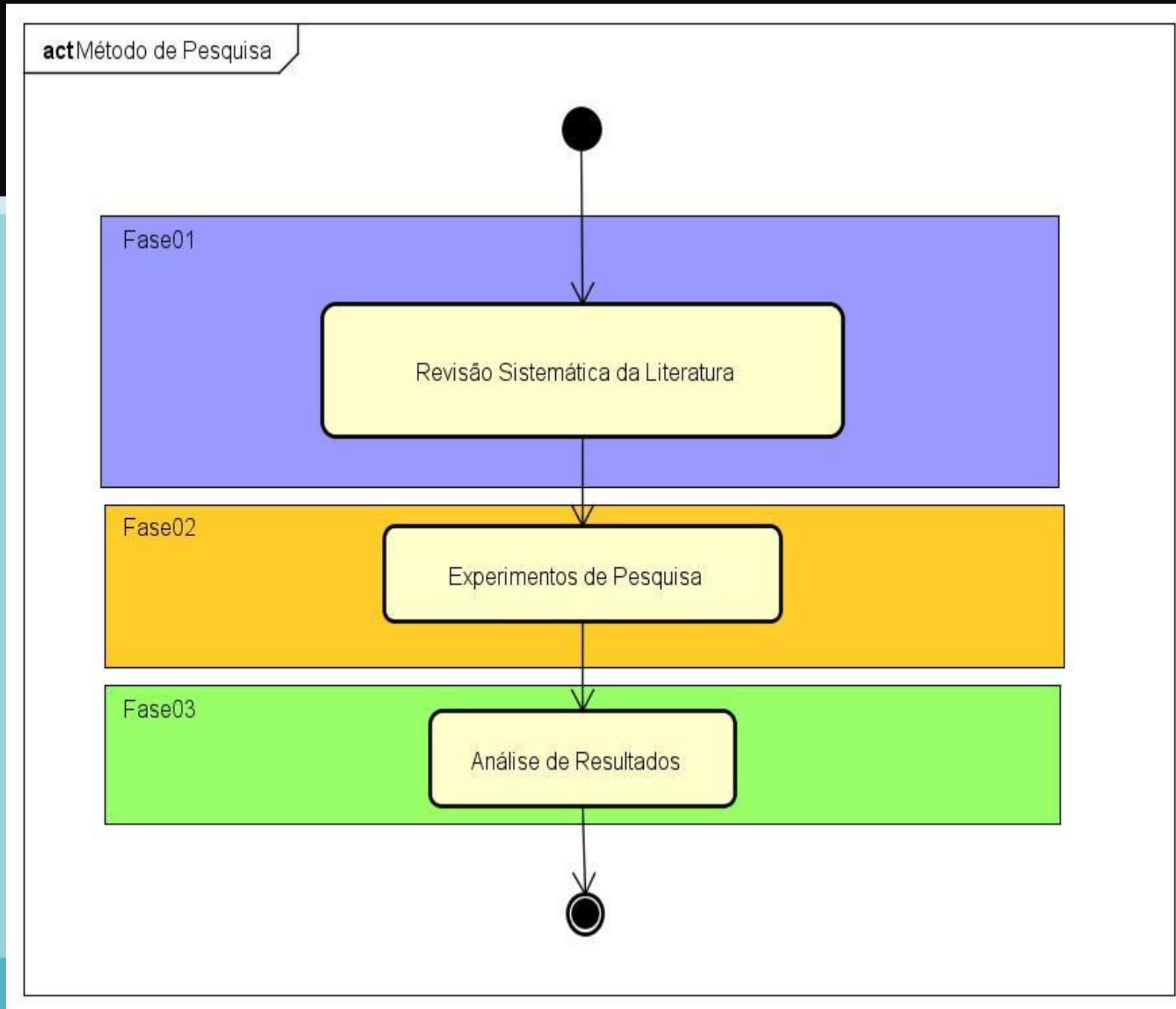
A grande maioria de pesquisas envolvendo buscas de musicas por similaridade vem sendo nomeada com o prefixo "Query-by" onde o complemento do nome identifica geralmente o Input de dados sendo realizado na Query, mas...

O Que faz Query-By-Playing ser diferente?

- Instrumentos Musicais.
 - Real-Time.
 - Intuitiva.

A Pesquisa

1. Revisão Sistemática
2. Experimentos
3. Análise de Resultados



Abordagem de Pesquisa

String Matching

Algoritmo Smith Waterman

Tabela 3 – Score de Pesquisa.

Abordagem	Algoritmos	Qtd. Pesquisa	Ref. Relevantes	Relação Direta	Score
String	NW SW	72	13	8	31
Geometric	DTW , LBDM	34	23	14	23
N-Grams	—	24	10	3	12
Probabilistic	HMM , SVM	75	4	4	27
Hybrid	—	51	17	5	24

Fonte: Autoria Própria

O Algoritmo de Similaridade SW

Critério de Escolha:

- Baseado em Strings.
- Realiza Alinhamento Local.
- Possibilita Alinhamento de trechos musicais.

Algoritmo 3: O Algoritmo Smith Waterman

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

```
14 Inicialização;  
15  $D(i,j) = 0$ ;  
  
16 Calculo de Scores;  
17 for  $i \leftarrow 1$  to  $M$  do  
18   | for  $j \leftarrow 1$  to  $N$  do  
19   |   |  $D(i,j) = \text{Max}(i,j)$ ;  
20   |   end  
21 end
```

Algoritmo 4: A Funcao Max

input : A posicao atual i,j

output : Distancia Maxima

Result: O resultado sera o score maximo de distancia.

```
22  $lado = M[i-1,j] + gap$ ;  
23  $topo = M[i,j-1] + gap$ ;  
24  $diagonal = M[i-1,j-1] + \text{Score de Erro : Acerto}$ ;  
25  $retorno = \text{Maximo}(topo, lado, diagonal)$ 
```

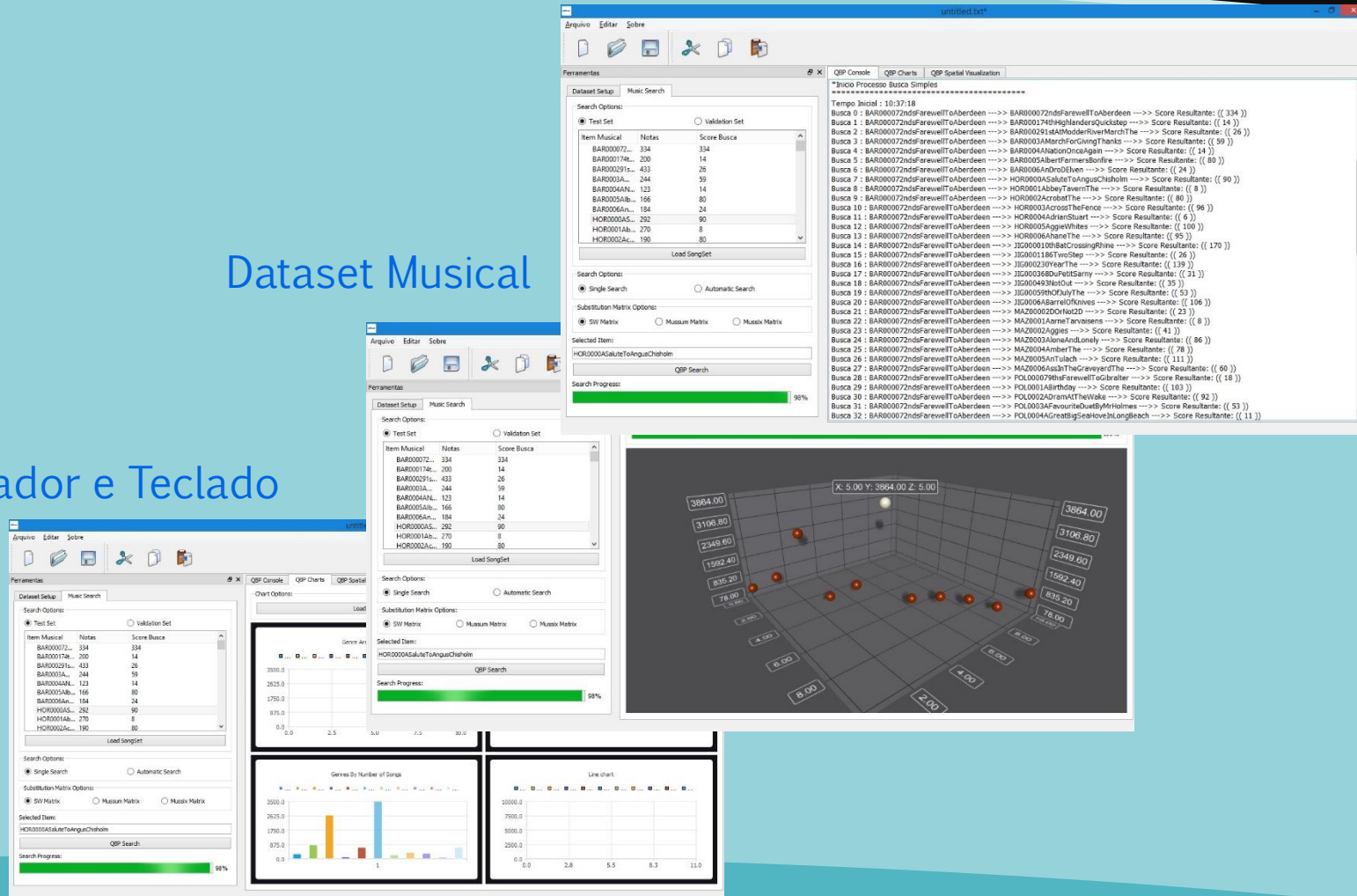
O Ambiente Computacional

Para a realização deste trabalho foi necessário um ambiente computacional que possibilite Organizar , converter e processar os experimentos de pesquisa, além de coletar os resultados de cada experimento para sua consecutiva avaliação.

Dataset Musical

Sintetizador e Teclado Virtual

Sistema de Busca.



O Dataset Musical

Para a realização deste trabalho está sendo utilizado uma lista de 10.198 músicas tradicionais irlandesas em formato MIDI 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.

Tabela 6 – Build do Dataset Musical.

Descrição	Resultado
Criação Dataset	24 Horas para 10.199 Musicas MIDI
Carregamento de Dataset	10 segundos.
Tempo de Busca QBP	0.5 Milisegundos para uma musica completa
Acuracia Classificador	30% Custom e 82.48% com JSymb. e Weka.
Features Extraidas	8 Custom e 156 Jsymbolic
Algoritmos Implementados	3 variações do SW

Fonte: Autoria Própria

O Teclado Virtual

Neste trabalho este item computacional está sendo chamado de VirtualKeyboard e o mesmo compreende a abstração do Sistema Operacional para a conexão com instrumentos musicais, a síntese de áudio necessária para manipulação de notas musicais, bem como a reprodução destes resultados em saídas de áudio configuradas no sistema

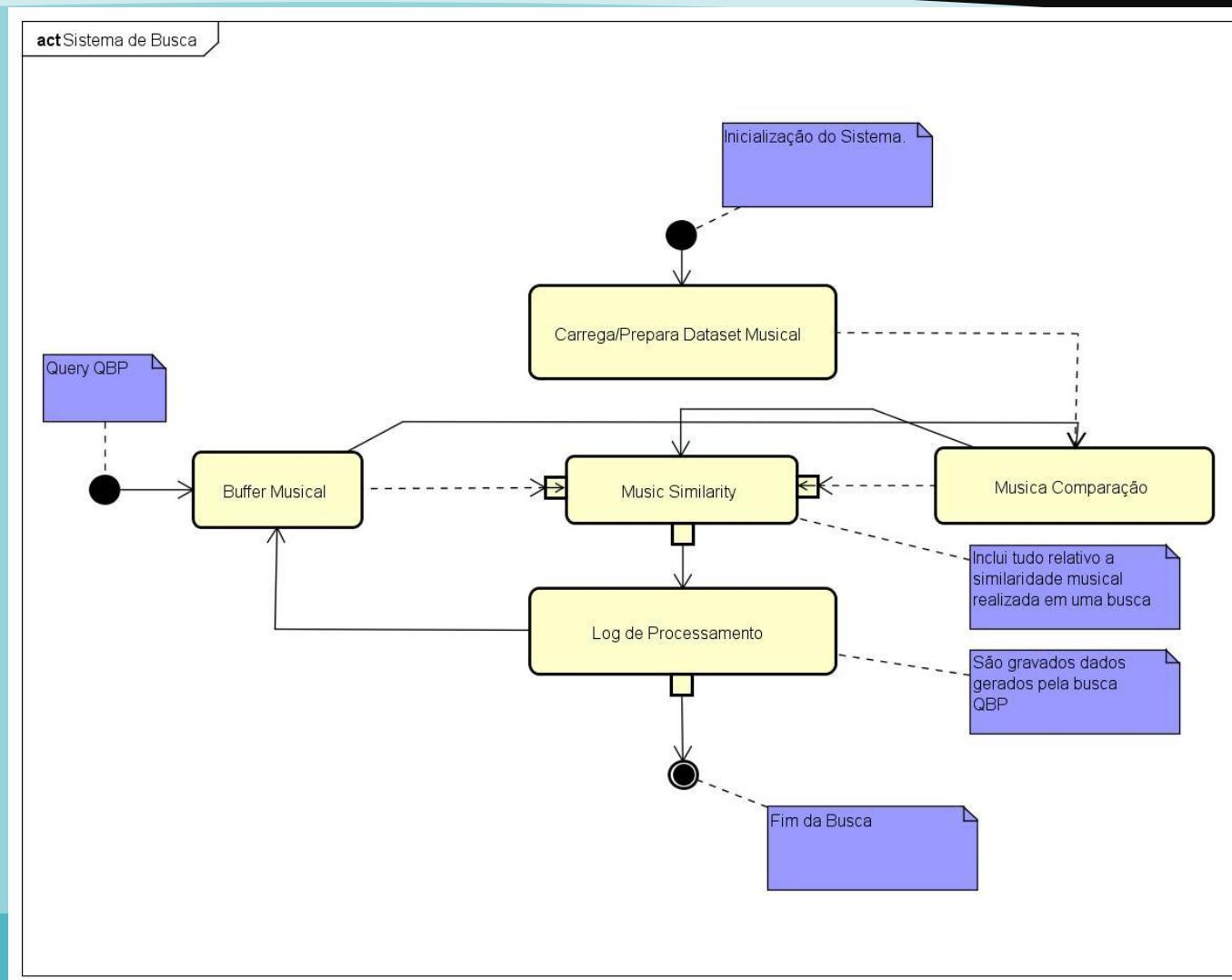
Tabela 7 – Tabela de Síntese.

Campo	Descrição	Existe na Definição MIDI?
Note Name	Nome da Nota Simbólica (Ex.: Ab , B , C , D)	Não
isBemol	Informa se esta nota é um Bemol na Escala	Não
isSharp	Informa se esta nota é um Sustenido na Escala	Não
Pitch	Valor Numérico da Nota MIDI do Instrumento	Sim
Frequency	Valor Numérico da Frequencia(Hz) da Nota MIDI	Não
Duration	Duração em Milisegundos da Nota MIDI	Sim
Octave	Oitava da Escala na nota MIDI	Não
Shortmessage	Mensagem Hexadecimal de Execução MIDI	Não

Fonte: Autoria Própria

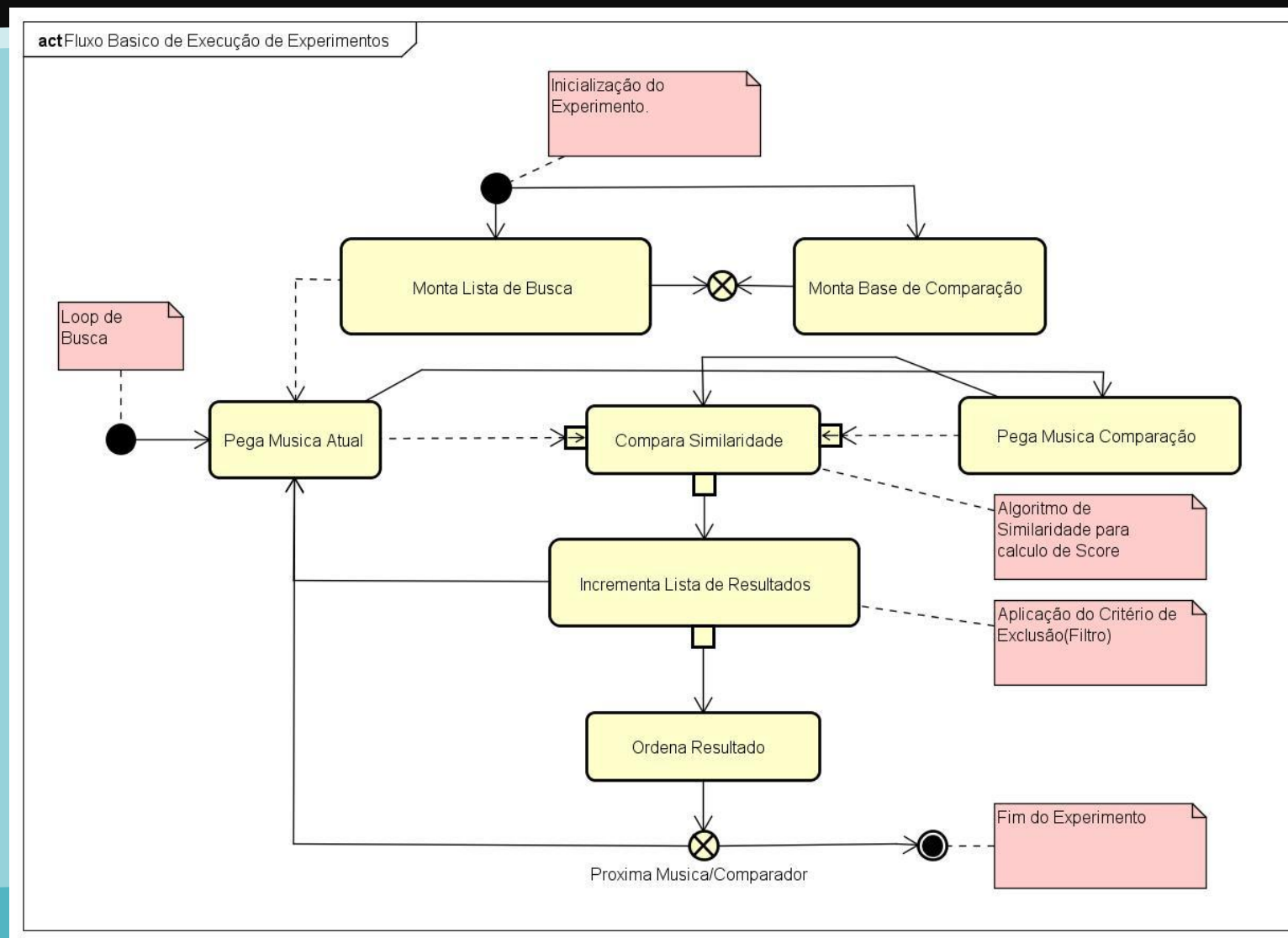
O Sistema de Busca

Para orquestrar a execução da consulta ao dataset musical e realizar a etapa de processamento (Music Similarity) é necessário um sistema de busca para que estes itens computacionais gerem os resultados esperados.



Experimentos de Pesquisa

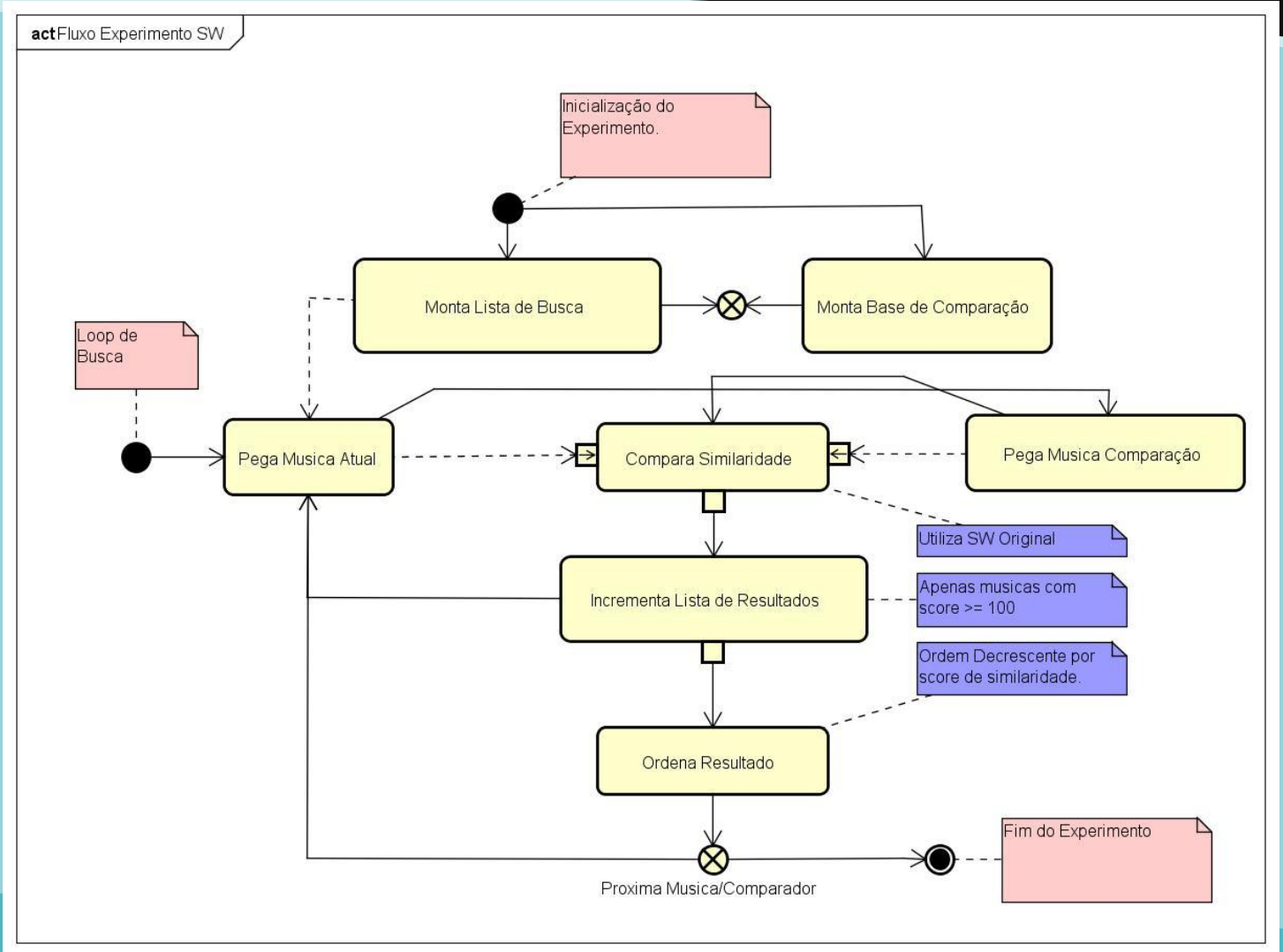
Todos os experimentos de pesquisa foram executados seguindo um fluxo básico de execução de forma que podemos identificar os pontos chave de processamento



Experimento SW (Smith Waterman Padrão)

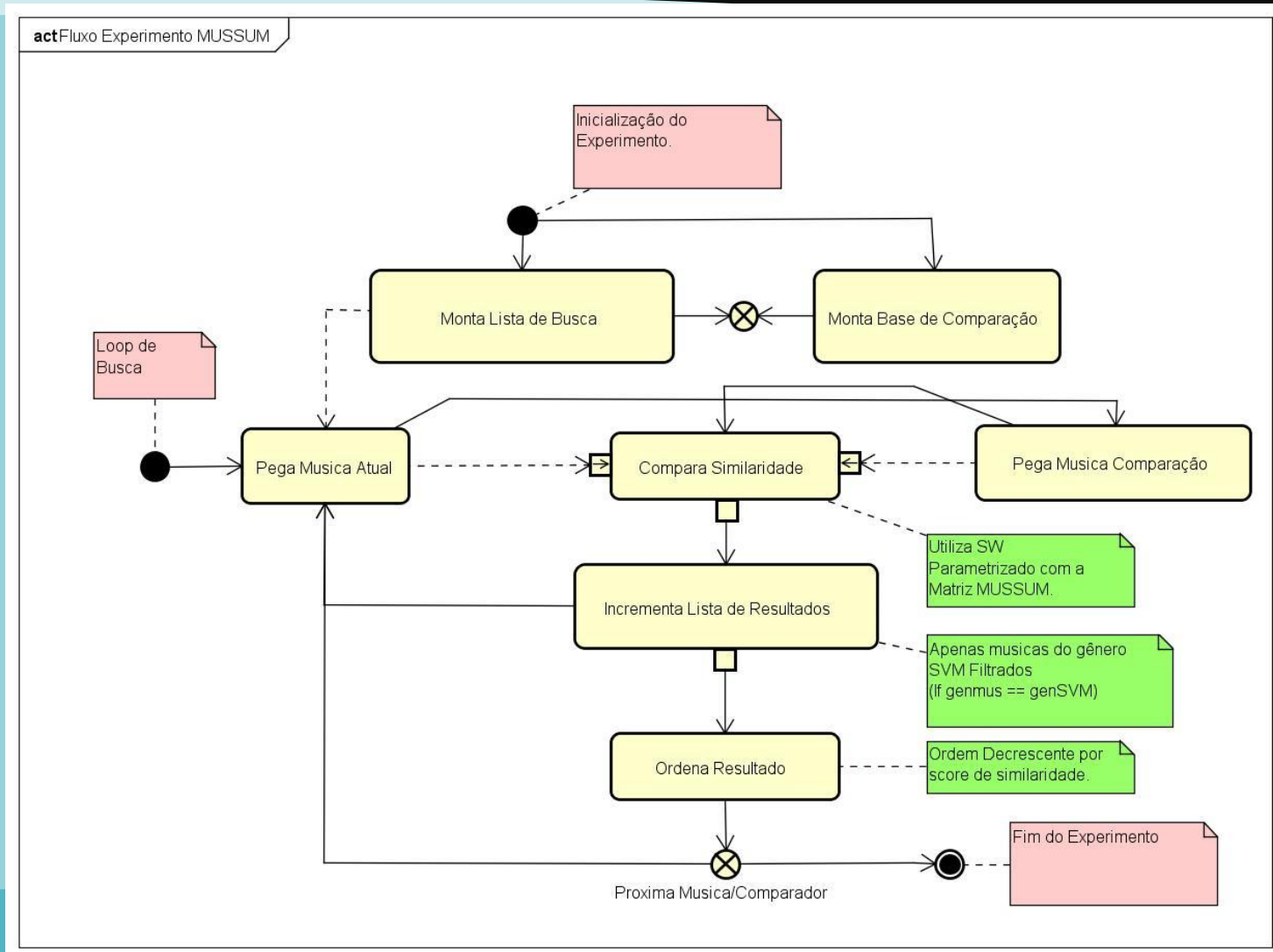
No primeiro experimento foi realizado dez buscas (1 música randômica por gênero musical) sobre todo o dataset musical.

O processo executado pode ser visualizado na Figura e compreende na execução dos testes de similaridade utilizando o algoritmo original Smith Waterman.



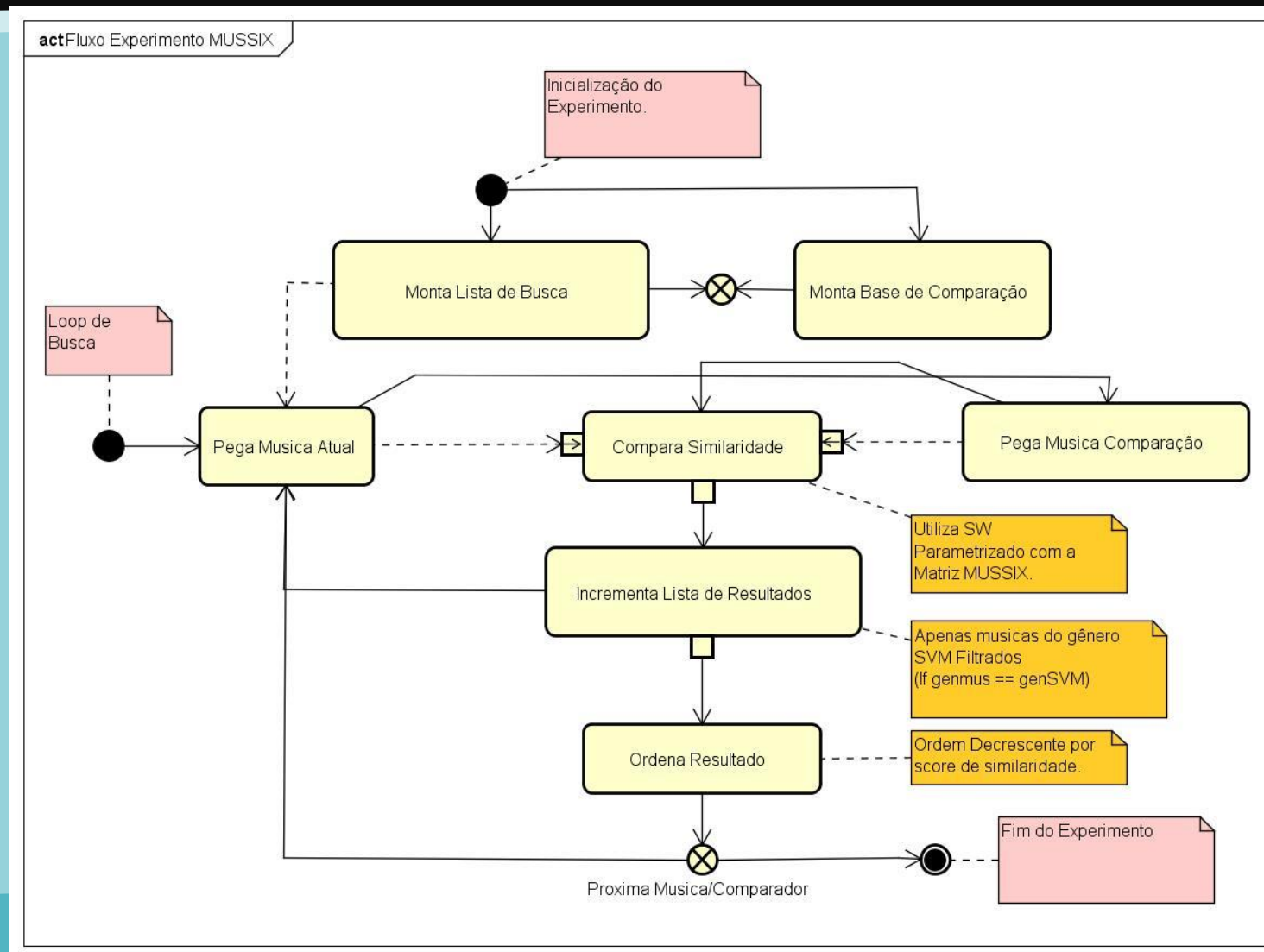
Experimento MUSSUM

O Segundo experimento desta pesquisa compreendeu em executar a pesquisa das mesmas dez musicas do experimento inicial, porém utilizando uma matriz de pesos parametrizando o algoritmo SW, chamada MUSSUM - Music Substitution Matrix, proposta no artigo (MARTINIANO, 2017) "BIRITS: A Music Information Retrieval System Using Query-by-Playing Techniques".



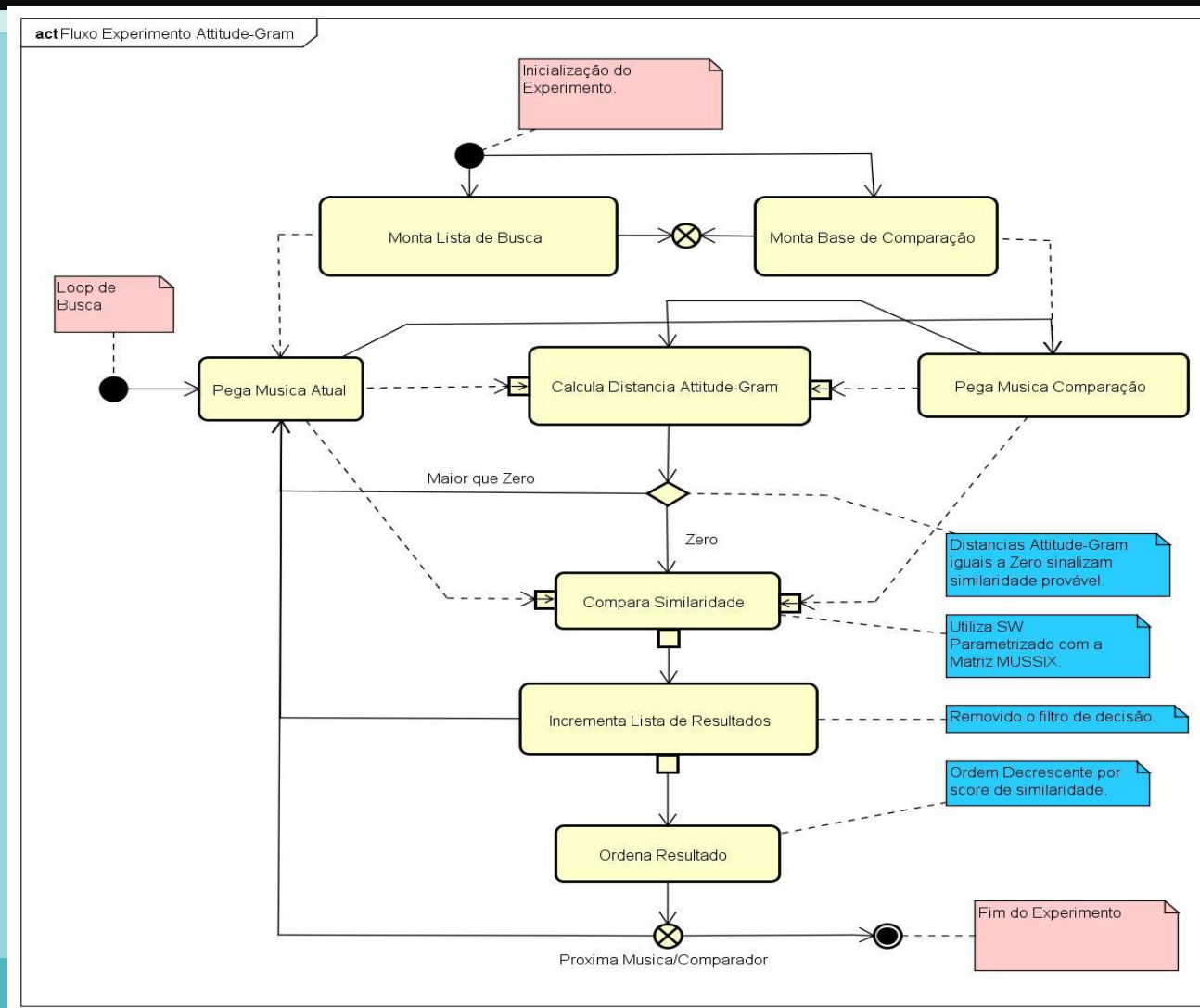
Experimento MUSSix

O terceiro experimento foi realizado a mesma pesquisa de dez músicas utilizada nos experimentos anteriores, porém utilizando uma nova matriz de parametrização chamada MUSSIX - Music Space Matrix.



Experimento Attitude-Gram

O quarto experimento foi realizado a mesma pesquisa de dez músicas utilizada nos experimentos anteriores, porém foi incluído uma atividade ,anterior a verificação de similaridade musical, para a determinação da relevância da música. Esta nova atividade representa o calculo de distancia Manhattan entre o AttitudeGram presente em todas as musicas do dataset.



Métricas de Avaliação

Acurácia

$$AcuráciaLocal(\%) = \frac{(N - P)}{N} \times 100$$

$$AcuráciaGlobal(\%) = \frac{A}{T} \times 100$$

N - número total de músicas

P - posição da música

A - numero de acertos

T - total de buscas

Eficiência

$$EficiênciaLocal(\%) = \frac{\frac{(L-R)}{L}}{T} \times 100$$

$$EficiênciaGlobal(\%) = \frac{\sum_{n=0}^S EL}{S}$$

L - número total de músicas

R - musicas retornadas

T - Tempo transcorrido

EL - eficiência local

S - total de buscas

Resultados Preliminares

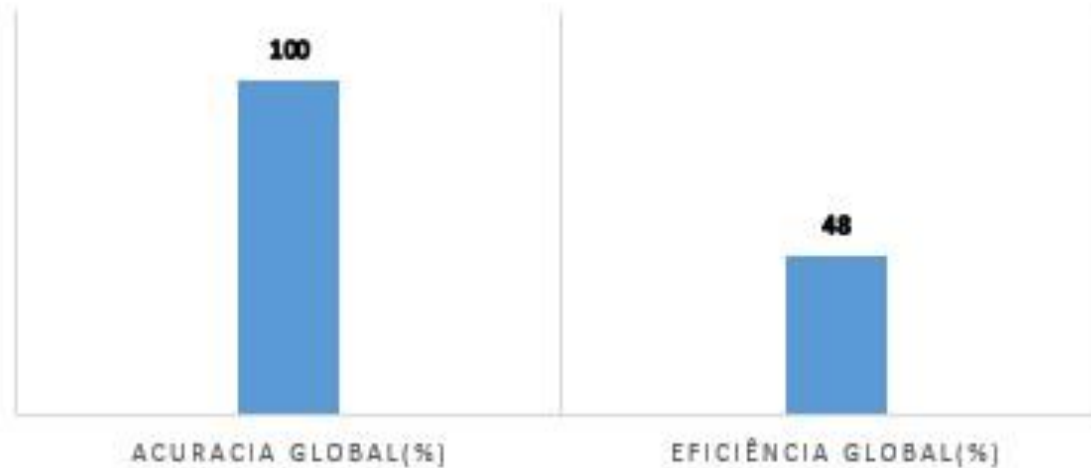
Eficiencia Local(%)



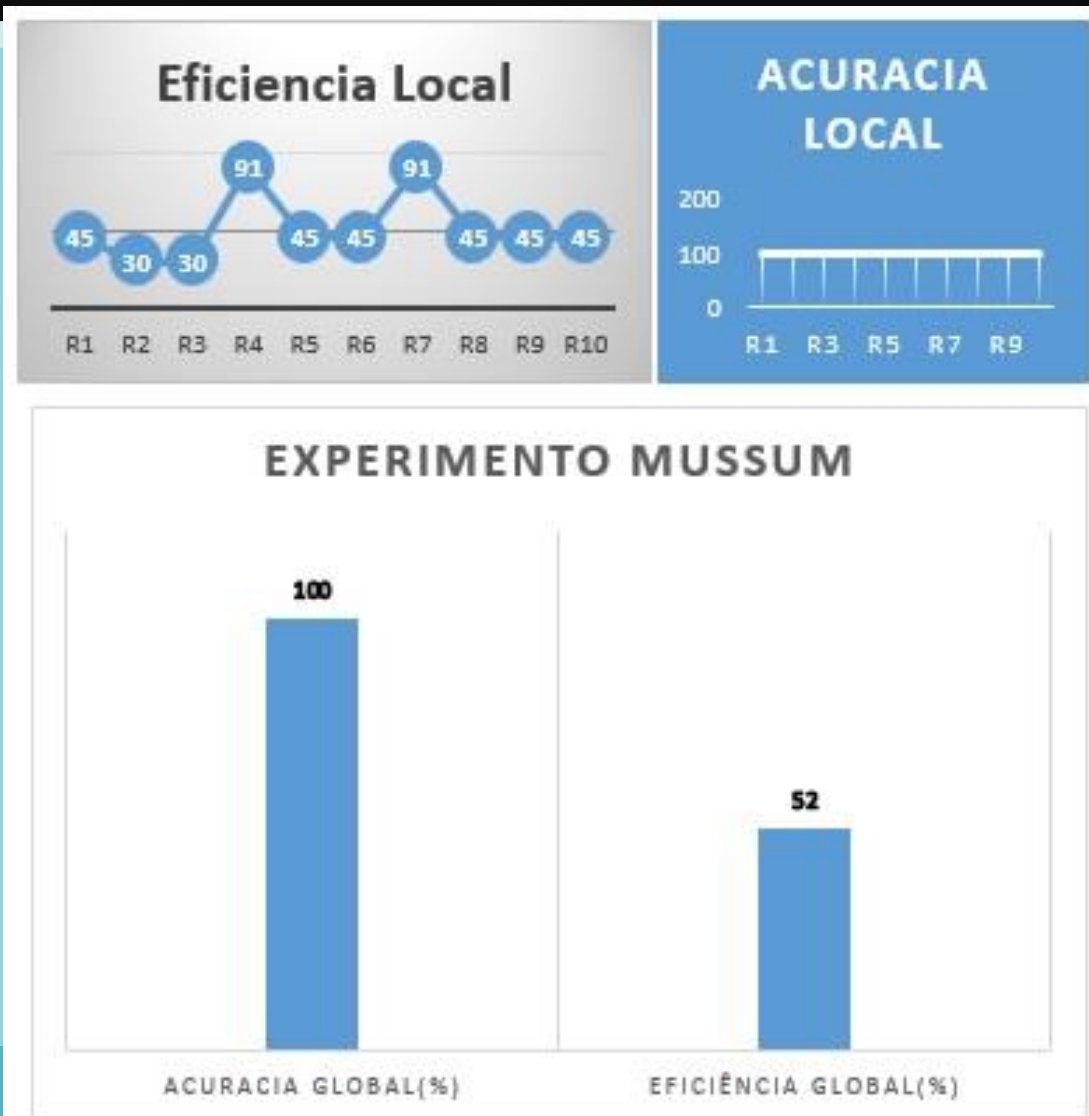
ACURACIA LOCAL



EXPERIMENTO SW



Resultados Preliminares

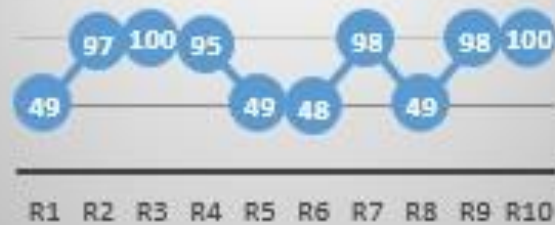


Resultados Preliminares



Resultados Preliminares

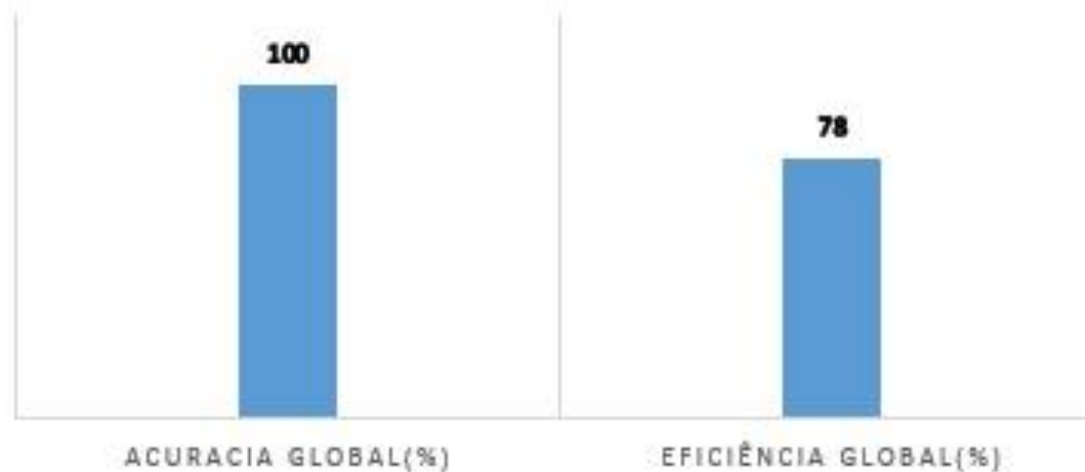
Eficiência Local



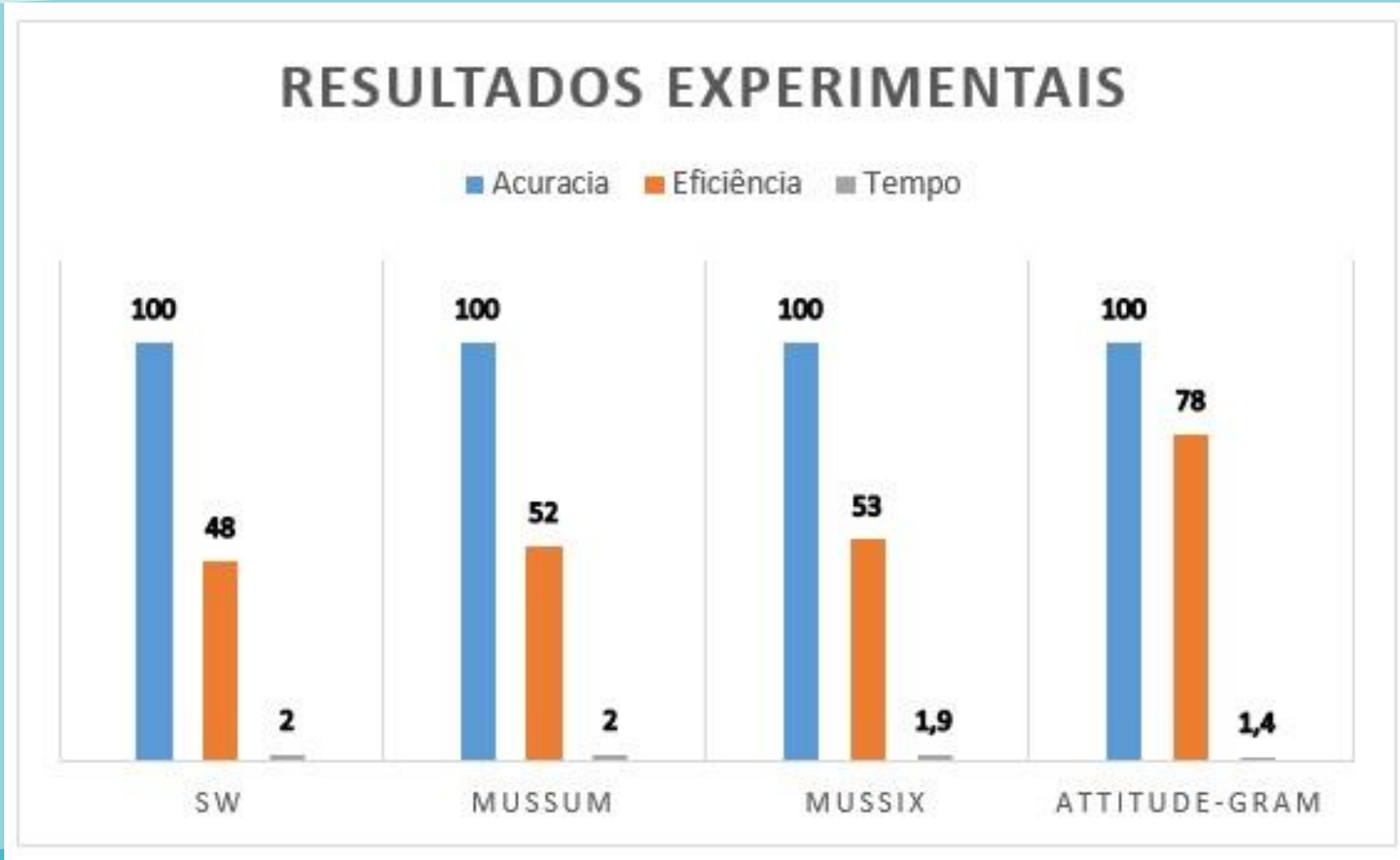
ACURACIA LOCAL



EXPERIMENTO ATTITUDE-GRAM



Resultados Preliminares



Análise dos Resultados Preliminares

Após a execução dos experimentos, os resultados de acurácia para todos os experimentos resultou em 100% para todas as buscas efetuadas.

Nos experimentos a maior variação de resultados ocorreu relativo a eficiência da busca em cada experimento onde as buscas utilizando o Attitude-Gram apresentaram uma eficiência de 25% a 30% maior que os outros experimentos.

Notou-se também que apesar de termos um filtro de pós-processamento baseado em gêneros musicais, existem diferenças significativas dentro do próprio gênero musical que puderam ser exploradas com a utilização também do Attitude-Gram possibilitando resultados mais relevantes musicalmente.

Através de testes de audição foi comparado os resultados das musicas na segunda e terceira posição do ResultSet de Busca, e em todos os experimentos a utilização da Matriz MUSSIX apresentou resultados mais musicais que a utilização da MUSSUM e SW.

Considerações

Este trabalho é dedicado a todos os cientistas músicos que, algum dia, sonharam em fazer a máquina entender sua arte.

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

Algoritmo 3: O Algoritmo Smith Waterman

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

```

14 Inicialização;
15 D(i,j) = 0;
16 Cálculo de Scores;
17 for i ← 1 to M do
18   for j ← 1 to N do
19     D(i,j) ← Max(i,j);
20   end
21 end

```

Algoritmo 4: A Função Max

input : posição atual i,j

output : Distância Máxima

Result: O resultado será o score máximo de distância.

```

22 lado = M[i-1,j] + gap;
23 topo = M[i,j-1] + gap;
24 diagonal = M[i-1,j-1] + Score de Erro : Acerto;
25 retorno = Maximo(topo,lado,diagonal)

```

A Matriz MUSSix

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

Algoritmo 3: O Algoritmo Smith Waterman

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

```

14 Inicialização;
15  $D(i,j) = 0$ ;
16 Calculo de Scores;
17 for  $i \leftarrow 1$  to  $M$  do
18   for  $j \leftarrow 1$  to  $N$  do
19      $D(i,j) = \text{Max}(i,j)$ ;
20   end
21 end

```

Algoritmo 4: A Funcao Max

input : A posicao atual i,j

output : Distancia Maxima

Result: O resultado sera o score maximo de distancia.

```

22  $lado = M[i-1,j] + gap$ ;
23  $topo = M[i,j-1] + gap$ ;
24  $diagonal = M[i-1,j-1] + \text{Score de Erro : Acerto}$ ;
25  $\text{retorno} = \text{Maximo}(topo, lado, diagonal)$ 

```

O n-Gram Attitude

