

Orientando: Adriano Luiz Moretti

Orientador: Paulo Cesar Rodacki Gomes



UNIVERSIDADE
REGIONAL DE BLUMENAU

PROTÓTIPO DE UM SOFTWARE PARA O RECONHECIMENTO DE NOTAS MUSICAIS

Roteiro

- Introdução
- Objetivos
- Teoria da música
- Propriedades físicas do Som
- Áudio Digital
- Sistemas MIDI
- Arquivos WAVE
- Transformada de Fourier
- Transformada Rápida de Fourier
- Desenvolvimento do Protótipo
 - Especificação
 - Implementação do Kaco Acordes
- Resultados
- Conclusões
- Extensões
- Protótipo KACO ACORDES
- Questionamentos

Introdução

- Como percepção musical, vários músicos não conseguem identificar qual a nota musical soada
- Técnicas para o reconhecimento de notas musicais
- A utilização de arquivos WAVE
- Os pioneiros da matemática musical
- Jean-Baptiste Joseph Fourier demonstrou que qualquer função física, pode ser expressa como uma série de somas e termos

Objetivos

O objetivo principal deste trabalho é desenvolver um software que faça o reconhecimento de notas musicais para auxiliar músicos que tenham uma percepção musical baixa

Objetivos específicos:

- Abstrair os dados de um arquivo WAVE
- Utilizar a transformada rápida de Fourier

Teoria da Música

- Notas musicais naturais:

Dó	Ré	Mi	Fá	Sol	Lá	Si
----	----	----	----	-----	----	----

- Sustenidos #:

Dó	Dó#	Ré	Ré#	Mi	Fá	Fá#	Sol	Sol#	Lá	Lá#	Si	Dó
----	-----	----	-----	----	----	-----	-----	------	----	-----	----	----

- Bemóis b:

Dó	Si	Si \flat	Lá	Lá \flat	Sol	Sol \flat	Fá	Mi	Mi \flat	Ré	Ré \flat	Dó
----	----	------------	----	------------	-----	-------------	----	----	------------	----	------------	----

Teoria da Música

INTERVALOS

- Distância entre duas notas musicais
- Intervalo Simples (dentro de uma oitava)
- Intervalo Composto(alem de uma oitava)
- Exemplos de um Intervalo para o outro:
 - nota Dó para o Ré – Intervalo de um tom;
 - nota Mi para o Fá – Intervalo de um semitom.

Teoria da Música

- ESCALAS

Seqüência de um conjunto de notas musicais

- ACORDES

Composta por três ou mais notas musicais

Melódica:

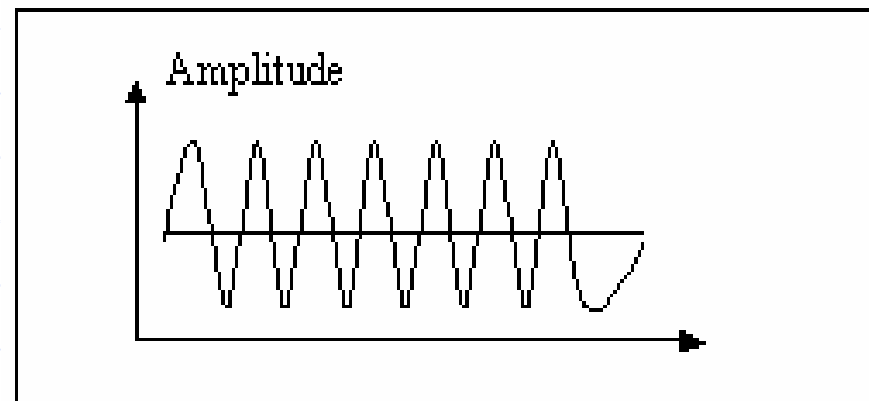
Acorde de forma arpegiada, ou seja, nota após nota

Harmônica:

Intervalos pertencentes ao acorde tocado simultaneamente

Propriedades Físicas do Som

- O ouvido do ser humano recebe vibrações que o ar transporta
- Equipamentos Transdutores:
 - De sinais elétricos para sinais acústicos: alto-falantes
 - De sinais acústicos para sinais elétricos: microfones
- Representação física:
 - onda senoidal (tempo x amplitude)



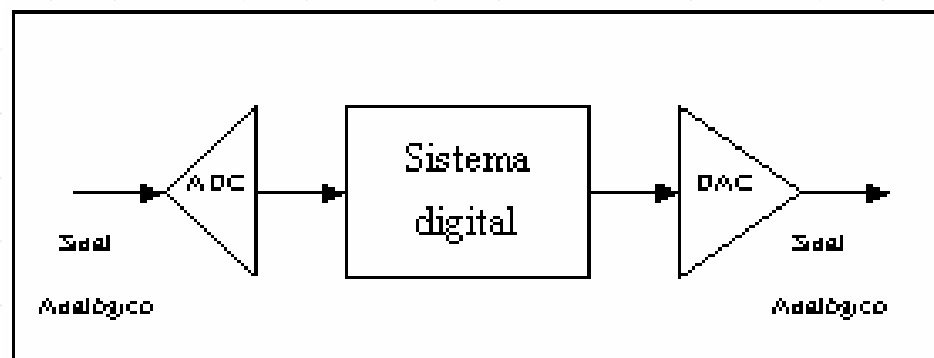
Parâmetros Perceptuais do Som

- **INTENSIDADE:** Qualidade que distingue sons fortes de sons fracos referente a volume
- **ALTURA:** Qualidade que distingue sons graves de agudo
- **TIMBRE:** Qualidade que distingue sons da mesma intensidade e altura, quando emitidos por instrumentos diferentes

Áudio Digital

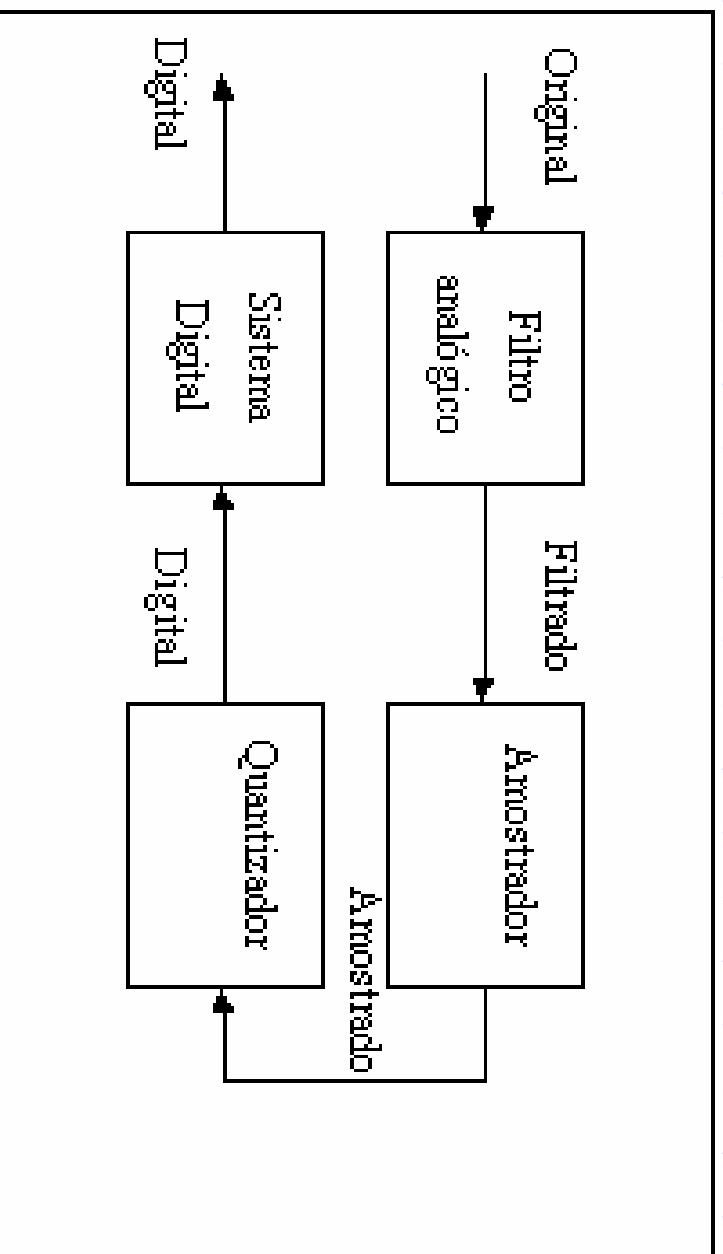
DIGITALIZAÇÃO DO SOM

- ADC – Conversor Analógico Digital
- DAC – Conversor Digital Analógico



Áudio Digital

PROCESSAMENTO DE DIGITALIZAÇÃO



Áudio Digital

OPERAÇÕES DE PROCESSAMENTO DIGITAL DO SOM

- No Domínio do Tempo
- No Domínio da Frequência

Sistema MIDI

- Protocolo MIDI
- Tipos de Instrumentos: teclados, acordeões
- Tecnologia MIDI: possui cinco conectores
 - MIDI IN*
 - MIDI OUT*
 - MIDI THRU*
 - MIDI ON*
 - MIDI OFF*
- Arquivos MIDI: armazenamento de dados musicais

MIDI X ÁUDIO

Vantagens:

- * Tamanho
- * Seqüenciadores
- * Troca na associação
- * Tempo Real

Desvantagens:

- * Reproduz apenas músicas
- * Sons de Instrumentos MIDI não são iguais aos originais

Arquivos *WAVE*

Desenvolvido pela *Microsoft*

RIFF(Resource Interchange File Format)

Estrutura do cabeçalho – *PCM Stereo*

Seus dados - *.data* contendo valores de amplitude

Número de canais

Transformada de Fourier

- O operador Fourier Transforma uma função $h(t)$ definida no domínio do tempo para o domínio da frequência
- Suas Propriedades:
 - * Linearidade
 - * Escalonamento
 - * Deslocamento no tempo
 - * Deslocamento na frequência
 - * Funções pares e ímpares

Transformada Rápida de Fourier

- Coleção de algoritmos de grande eficiência computacional
- TDF – Transformada Discreta de Fourier
- Dizimação no tempo
- Funcionamento:
número de amostras = potências de 2

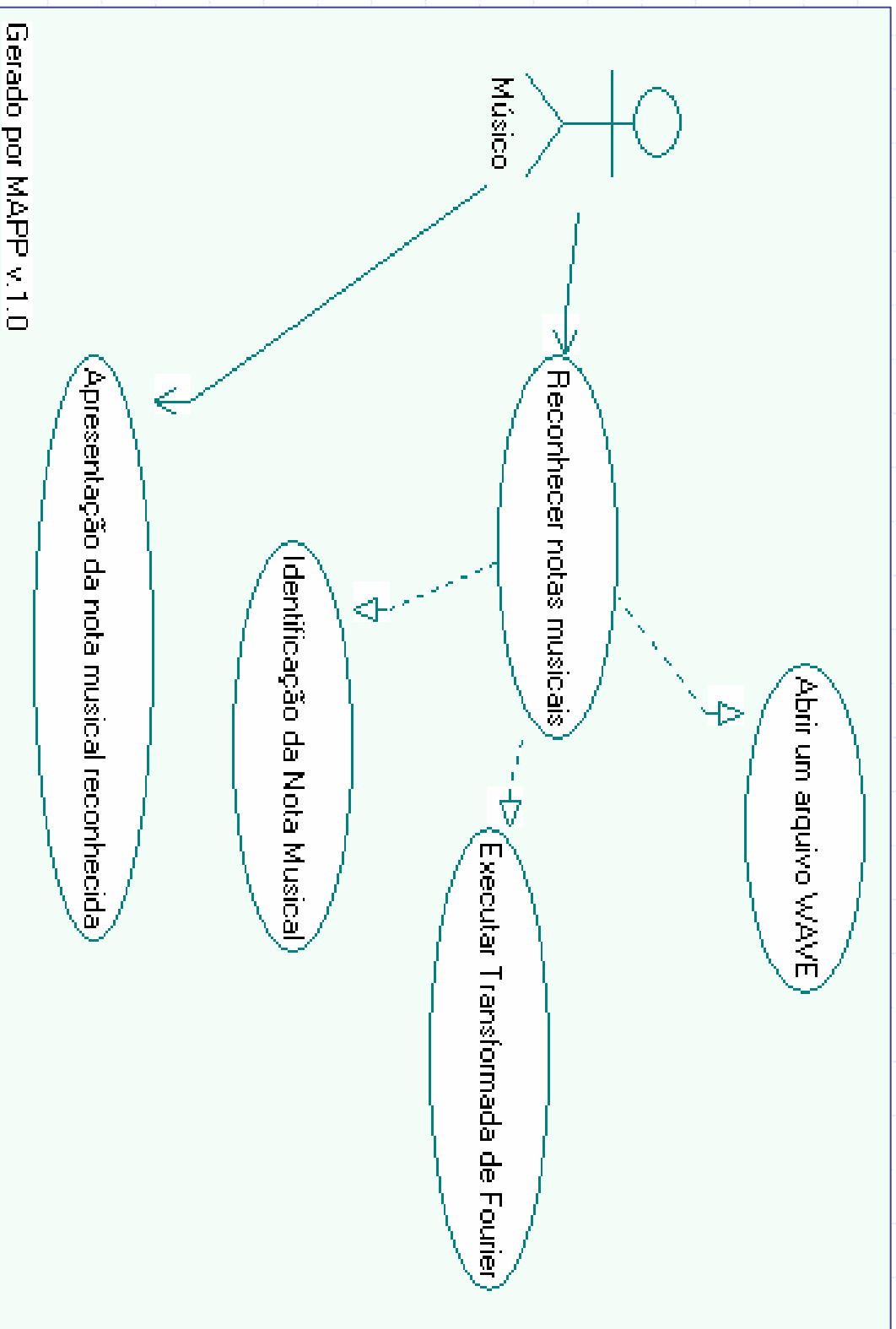
Desenvolvimento do protótipo

Requisitos:

Reconhecer uma nota musical qualquer a partir de um arquivo WAVE

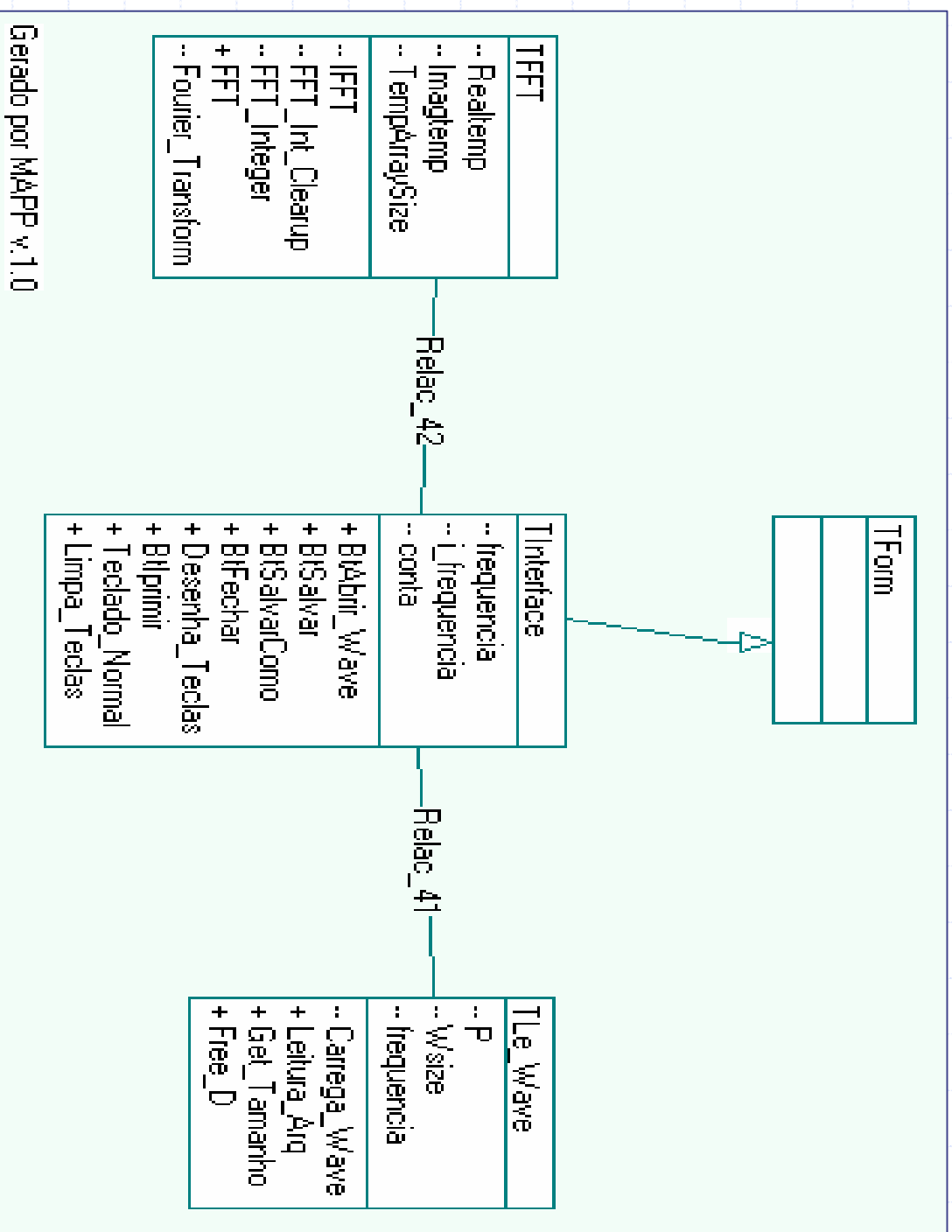
Especificação

Diagrama de Caso de Uso



Especificação

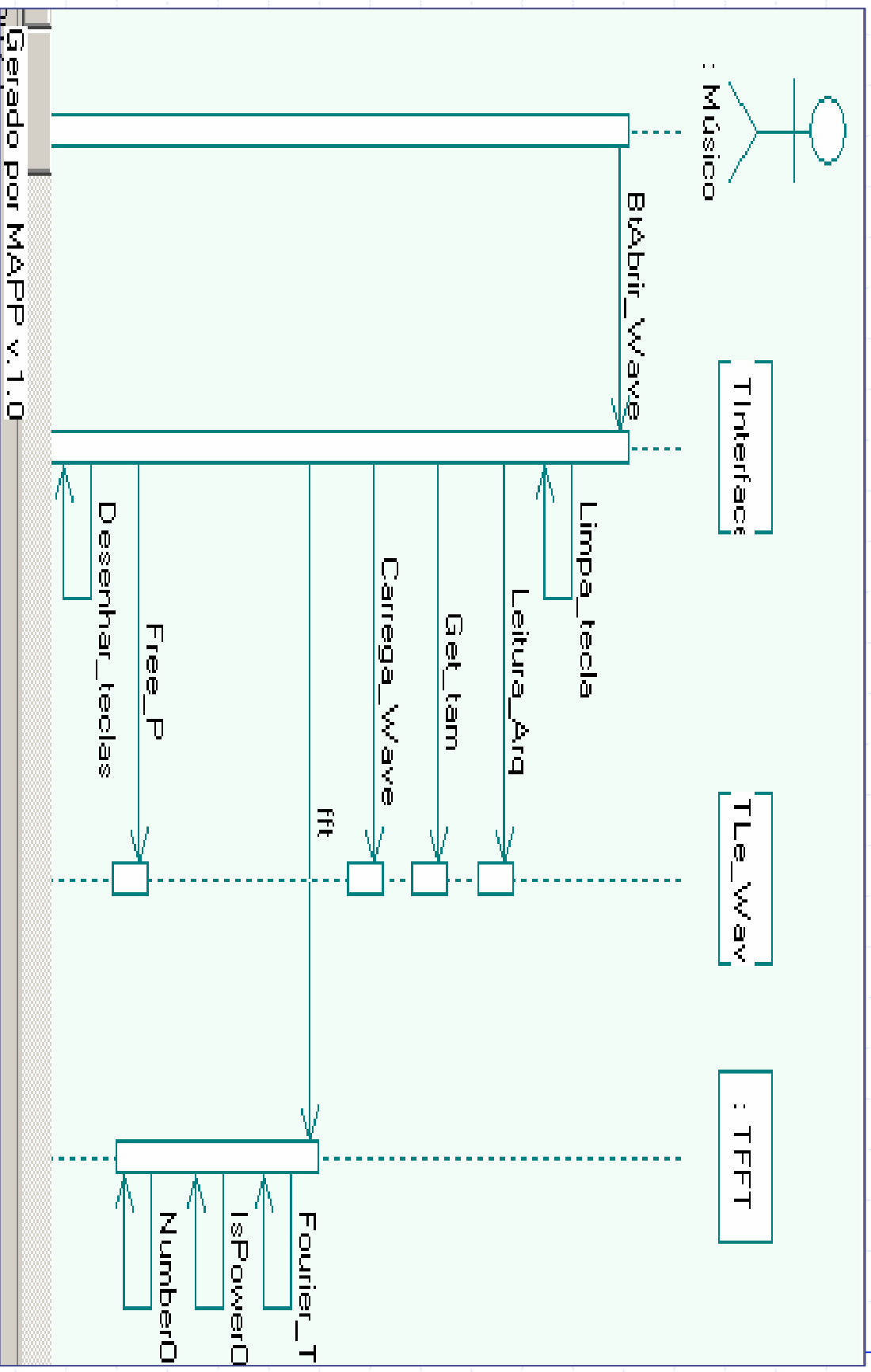
Diagrama de Classes



Gerado por MAPP v.1.0

Especificação

Diagrama de Sequência



Desenvolvimento

- Cálculo da frequência
- Média ponderada
- Transformada de Fourier
- Transformada Rápida de Fourier

Desenvolvimento

Cálculo da Freqüência:

Fórmula da Freqüência

$$F = \frac{PosCanal * Fa}{4410}$$

“Se freqüência ≥ 520 e freqüência ≤ 529 então
nota = Dó”

Desenvolvimento

Média Ponderada:

Vetor de Frequência das Notas Musicais

```
// valores reais das frequencias das notas musicais
const
    frequ_notas: array[0..41] of real = (65.00,
    174.61,185.00,196.00,207.65,220.00,233.08,246.94,261.63,277.18,
    293.66,311.13,329.63,349.23,369.99,392.00,415.30,440.00,466.16,
    493.88,523.25,554.37,587.33,622.25,659.26,698.46,739.99,783.99,
    830.61,880.00,932.33,987.77,1046.50,1108.73,1174.66,1244.51,
    1318.51,1396.91,1478.98,1567.98,1661.22,1760.00);
```

Desenvolvimento

Média Ponderada:

Cálculo da média Ponderada

```
soma_max := 0;
for vetor_f := low(frequ_notas) to high(frequ_notas) do
  begin
    pos_nota := (frequ_notas[vetor_f] + 4096) / 22050;
    j := round(pos_nota);
    soma := (Twx[j-2]*0.25) + (Twx[j-1]*0.5) + Twx[j] +
            (Twx[j+1]*0.5) + (Twx[j+2]*0.25);
    if soma > soma_max then begin
      soma_max := soma;
      resultado := vetor_f + 52;
    end;
  end;
setlength(frequencia, length(frequencia)+1);
frequencia[high(frequencia)] := resultado;
```

Desenvolvimento

Transformada de Fourier

Algoritmo da Transformada de Fourier

```
procedure Tfrmgeral.TF (var X: array of byte; var TFX: array of
real);
var
  k,n,tam : integer;
  TFR, TFI : real;
begin
  tam := Length(X);
  for n := 0 to tam-1 do begin
    Application.ProcessMessages;
    TFR := 0; TFI := 0;
    for k := 0 to tam-1 do
      begin
        TFR := TFR + X[k]*cos(2*pi*n*k/tam);
        TFI := TFI + X[k]*sin(2*pi*n*k/tam);
      end;
    TFX[n] := abs(sqrt(TFR*TFR + TFI*TFI));
  end;
end;
```

Desenvolvimento

Transformada Rápida de Fourier Aloca memória para FFT

```
procedure TTF.fft_integer(NumSamples: word; var RealIn,  
  ImagIn: array of integer; var RealOut, ImagOut: array of  
  double);  
var  
  i: word;  
begin  
  if NumSamples > TempArraySize then begin  
    //libera memória caso seja limpa  
    fft_integer_cleanup;  
    // aloca memória para a proxima amostragem  
    GetMem ( RealTemp, NumSamples * sizeof(double) );  
    GetMem ( ImagTemp, NumSamples * sizeof(double) );  
    // atualiza esta variável com o número de amostra 4096  
    TempArraySize := NumSamples;  
  end;  
  for i := 0 to NumSamples-1 do begin  
    RealTemp^[i] := RealIn[i]; // numeros de amostra  
    ImagTemp^[i] := ImagIn[i]; // valores zero para imagem  
  end;  
  //chama a procedure para o calculo da fft  
  FourierTransform (2*PI, NumSamples, RealTemp^, ImagTemp^,  
    RealOut, ImagOut );  
end;
```

Desenvolvimento

Transformada Rápida de Fourier Libera memória para FFT

```
procedure TTF.fft_integer_cleanup;  
begin  
  if TempArraySize > 0 then begin //verifica se amostra  
    if RealTemp <> MILL then begin  
      FreeMem ( RealTemp, TempArraySize *  
        sizeof(double) );  
      RealTemp := MILL; // foi liberada a memoria da  
        amostra  
    end;  
    if ImagTemp <> MILL then begin  
      FreeMem ( ImagTemp, TempArraySize *  
        sizeof(double) );  
      ImagTemp := MILL;  
    end;  
    TempArraySize := 0;  
  end;  
end;
```

Resultados

- Foram feitos vários testes com inúmeras notas musicais
- Todas foram reconhecidas com SUCESSO!

Conclusões

Principais conclusões:

- Protótipo específico para acordeão, pode ser adaptado para outros instrumentos
- O Reconhecimento de uma nota foi possível e é viável
- O reconhecimento de seqüências de notas apresentou problemas
- A música é uma ciência exata, isto facilita o desenvolvimento de ferramentas computacionais para o músico

Limitações

Principais limitações do protótipo:

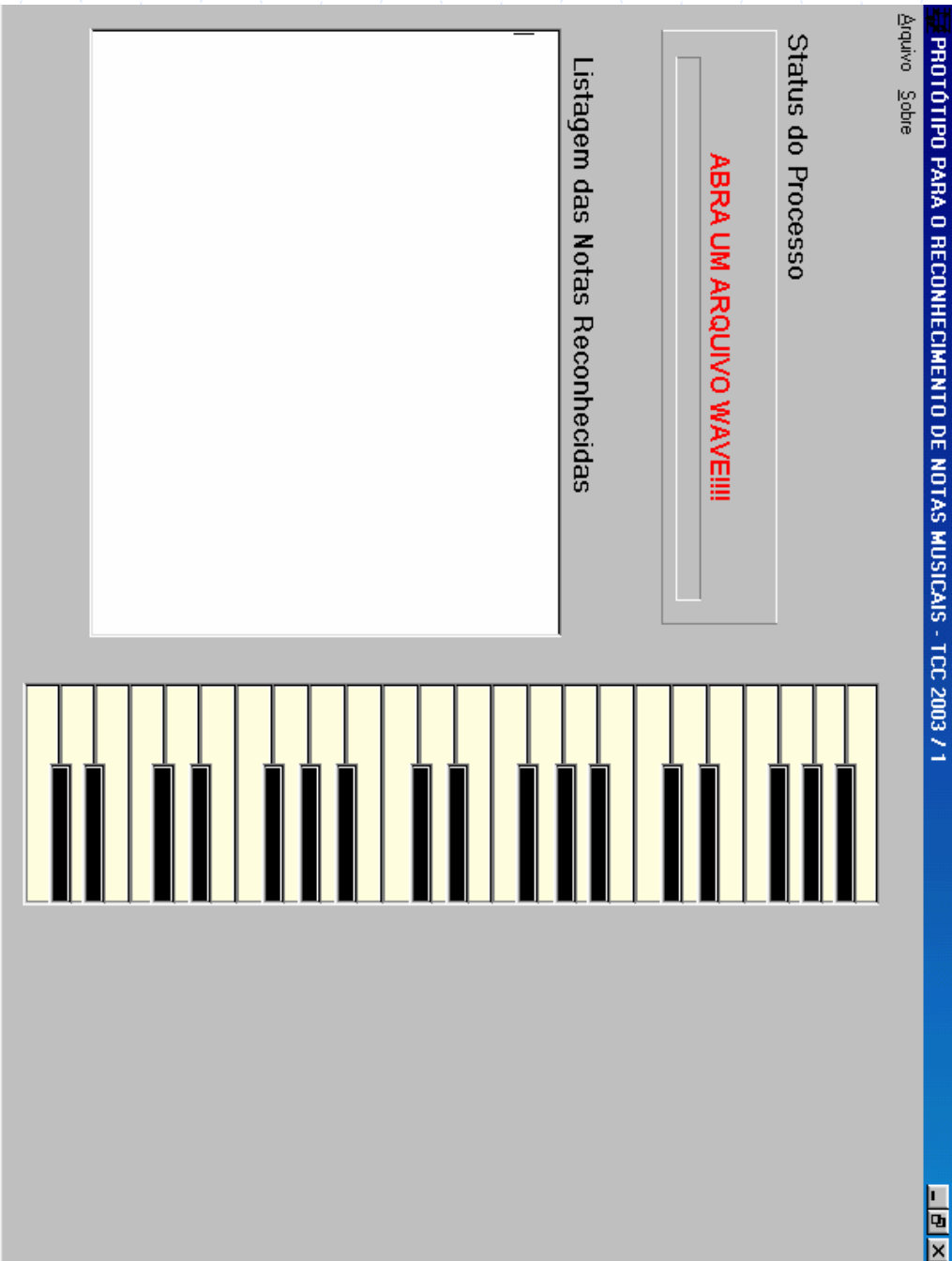
- Não reconhece seqüências de notas musicais
- Não reconhece timbre, apenas notas

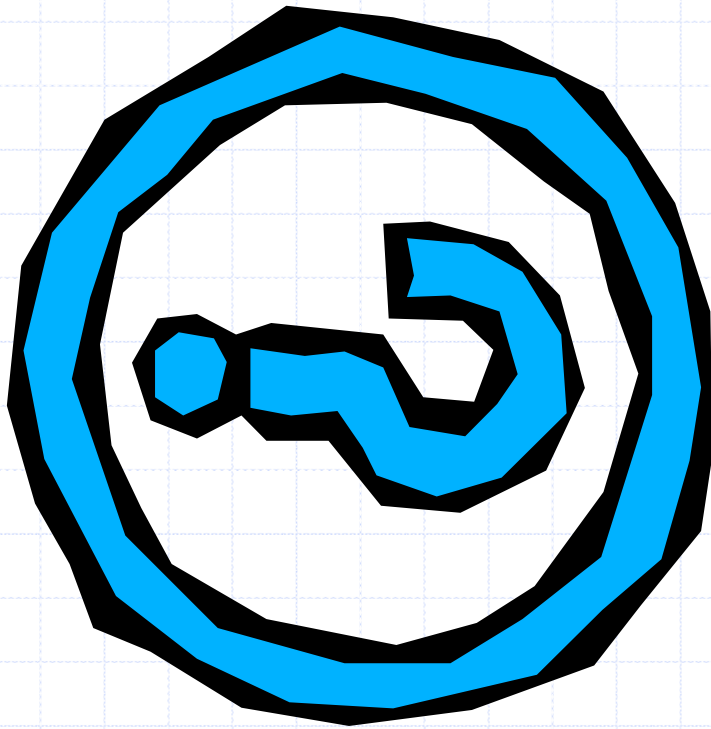
Extensões

Sugestões para trabalhos futuros:

- Reconhecer acordes musicais
- Ferramenta de afinação de instrumentos musicais utilizando PDA's
- Adaptação da interface do software para outros instrumentos

Protótipo





Questionamentos