

CENTRO UNIVERSITÁRIO FACVEST
CURSO DE CIÊNCIA DA COMPUTAÇÃO
ANDREI DE MOURA MACHADO

**APLICAÇÃO DA INFORMÁTICA AO ESTUDO DE CONCEITOS
MUSICAIS**

**LAGES
2012**

ANDREI DE MOURA MACHADO

**APLICAÇÃO DA INFORMÁTICA AO ESTUDO DE CONCEITOS
MUSICAIS**

Trabalho de conclusão de curso apresentado ao Centro Universitário FACVEST como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profº. Msc. Márcio José Sembay

LAGES

2012

ANDREI DE MOURA MACHADO

**APLICAÇÃO DA INFORMÁTICA AO ESTUDO DE CONCEITOS
MUSICAIS**

Trabalho de conclusão de curso apresentado ao Centro Universitário FACVEST como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profº. Msc. Márcio José Sembay

Lages, SC ____/____/2012. Nota _____

Márcio José Sembay

Profº. Msc. Márcio José Sembay

LAGES

2012

AGRADECIMENTOS

Primeiramente a DEUS, sem ele nunca conseguiria alcançar meus objetivos. Minha família, que sempre esteve junto comigo nos momentos mais difíceis. Aos companheiros de sala de aula e meus amigos, que nunca deixaram de me apoiar. E por último aos professores e mestres que dedicaram seu tempo a nos apoiar.

APLICAÇÃO DA INFORMÁTICA AO ESTUDO DE CONCEITOS MUSICAIS

Andrei de Moura Machado ¹

Márcio José Sembay ²

RESUMO

Este trabalho tem como objetivo desenvolver um sistema relacionado com percepção musical, com base na identificação de intervalos musicais. Ele se caracteriza como uma ferramenta de auxílio à aprendizagem a ser utilizado por alunos, professores, profissionais e por qualquer pessoa que se interesse por música. Como forma de alcançar este objetivo foi realizada vários estudos em áreas como computação musical, informática e ensino da música. Através de entrevistas com profissionais da área de música foram analisadas várias técnicas de aprendizagem, que contribuíram no desenvolvimento do sistema, destacando os problemas enfrentados entre alunos e professores e como pode ser melhorada essa questão. Após estes estudos, conseguiu-se desenvolver um software que auxilia na percepção musical. Foi desenvolvido um sistema que seja capaz de apresentar intervalos musicais para o usuário e propor exercícios sobre os mesmos, visando o treinamento auditivo do mesmo. O software gera aleatoriamente, dentro do módulo escolhido, intervalos que são apresentados para o usuário. Em seguida, também de forma aleatória, são gerados exercícios. Neste momento, apenas a primeira nota é tocada e, através do teclado virtual, o aluno deve tocar a segunda nota. O diferencial deste sistema em relação aos semelhantes é que ele já tem definida a sequência de apresentações dos intervalos. O software pode ser utilizado por qualquer usuário. Pretendo me aprofundar ao máximo para futuras implementações.

¹ Acadêmico da 9ª fase do Curso de Ciência da Computação, do Centro Universitário FACVEST.

² Mestre

Palavras-Chave: intervalo musical; percepção musical; informática na educação; ensino de música.

APPLICATION OF COMPUTING THE STUDY OF MUSICAL CONCEPTS

Andrei de Moura Machado¹

Márcio José Sembay²

ABSTRACT

This goal of this work is to develop a system related to musical perception based on the identification of musical intervals. It is featured as a helping tool for learning, and is suitable for use by students, professors, professionals, and any person interested in music. To reach this objective, several studies were performed in areas such as musical computing, information technology, and music teaching. Through interviews with music professionals, many learning techniques that contributed for the system development were analyzed, highlighting problems faced by students and teachers, as well as how this matter can be improved. After these studies, a *software* system that helps on musical perception was developed. A system capable of showing musical intervals to the user and offering exercises about them was developed, aiming to train the user hearing ability. The *software* generates intervals within the chosen module at random, which are shown to the user. Next, also at random, exercises are generated. At this moment, only the first tone is played and, through a virtual keyboard, the student is supposed to play the second tone. The difference between this system and other similar ones is that the interval sequence of this system is already defined. Any person can use the software. I intend to deepen the maximum for future implementations.

Keywords: musical interval; musical perception; information technology on education; music teaching.

¹ Acadêmico da 9ª fase do Curso de Ciência da Computação, do Centro Universitário FACVEST.

² Márcio José Sembay

Sumário

INTRODUÇÃO	11
1. TEMA	13
1.1. TEMÁTICA	13
1.2. JUSTIFICATIVA	13
1.3. OBJETIVOS	14
1.4. OBJETIVO GERAL	14
1.5. OBJETIVOS ESPECÍFICOS	14
1.6. METODOLOGIA	14
2. REFERENCIAL TEÓRICO	15
2.1. Conceitos básicos da música	15
2.2. Intervalo musical	15
2.3. Tipos de Intervalos Musicais	16
2.4. Intervalo Simples e Composto	16
2.5. Intervalo melódico	16
2.6. Intervalo harmônico	17
2.7. Nomes	17
3. PERCEPÇÃO MUSICAL	19
4. METODOLOGIA DE ENSINO.....	20

5. SOFTWARES SIMILARES.....	21
5.1 SOLFEGE	21
5.2 GUIDO 1.03	23
5.3 Tipos de estudo	24
5.4 Modo estudo <i>versus</i> modo teste	24
5.5 Modo de teste seletivo.....	24
5.6 Opções do programa	25
5.7 Considerações sobre o <i>software</i> Guido 1.03	26
5.8 Conclusão	26
6. FERRAMENTAS PARA DESENVOLVIMENTO.....	27
6.1 MIDI – Musical Instrument Digital Interface	27
6.2 Mensagens MIDI	28
6.3 Mensagens que dependem de canal MIDI	28
6.4 Mensagens “transparentes”.....	29
6.5 Codificação das notas em MIDI	29
6.6 Numeração de timbres	31
6.7 Codificando uma sequência MIDI	34
7. JAVA	37
7.1 Java e multimídia	37
7.2 O pacote Java Sound	38
7.3 Classes do pacote Java Sound	38
7.4 Classe Synthesizer	39
7.5 Classe ShortMessages.....	39
7.6 Classe Receiver	39
7.7 Classe MidiSystem.....	40
7.8 Conclusão	40

8. MODELAGEM DO SISTEMA.....	41
8.1 UML (<i>Unified Modeling Language</i>)	41
8.2 Especificação dos requisitos funcionais	41
8.3 Diagrama de <i>Use Case</i>	42
8.4 Descrição dos cenários	43
8.5 Diagrama de classes.....	43
8.6 Estrutura dos exercícios.....	45
8.7 Conclusão	47
9. APRESENTAÇÃO DO SISTEMA	47
9.1 Interface do sistema.....	47
9.2 Módulo exemplo	49
9.3 Módulo exercício.....	51
9.4 Verificação do desempenho	52
9.5 Conclusão	53
CONSIDERAÇÕES FINAIS.....	53
REFERÊNCIAS	55

INTRODUÇÃO

Hoje, a revolução que os equipamentos digitais provocam está baseada, entre outros efeitos, na disponibilidade de informação, na facilidade como que se pode conseguir bom material de estudo em qualquer área. A quantidade e qualidade de assuntos e de produções aparecem na multiplicidade de mídias (CDs, DVD, CDRoms etc.), de meios de comunicação, como por exemplo, no aumento do número de TVs abertas e TVs a cabos, ou de publicações especializadas nos mais diversos assuntos em bancas de jornais, até se ampliar praticamente ao infinito com as possibilidades da world Wide web e outras redes da internet. Antes de qualquer coisa, a forma e a qualidade de informação são mais importantes agora que sua simples existência material. Se antes eram comuns os pontos de vista únicos disponíveis, as atividades e materiais iguais para todos, os modos de ensino rígidos e formatados, hoje tudo é variedade, possibilidade de escolha, disponibilidade de novos pontos de vista.

Assim, acredito que o aprendizado em música e informática deva ser menos voltado para uma apresentação de “como fazer o quê” em determinados programas de computador, num simples processo de repetição do que o professor indica, e mais voltada para a compreensão dos princípios importantes de funcionamento, para a criação de novas idéias do que o programa possa fazer, de modos úteis e eficientes de produzir o que se deseja em termos musicais, seja uma gravação, seja uma partitura, seja simplesmente descobrir a letra ou o nome de uma música.

O que pretendo com as aulas de música e computador é apresentar as possibilidades mais importantes de uso do computador em termos musicais, indo desde operações básicas (como, por exemplo, simplesmente escutar um CD no IO aparelho do computador) até manipulações e informações avançadas. As aulas deverão ter tanto um conteúdo teórico, falando também de princípios úteis de acústica (ciência do som) e teoria musical, quanto uma realização prática no equipamento disponível, na escola ou na vida particular do aluno. Uma interação com a internet será fundamental, como fonte de informação e de ferramentas; os principais programas de computador vistos no curso, por exemplo, serão programas disponíveis gratuitamente (freeware) na World Wide Web, dispensando a necessidade do aluno já possuir de antemão programas caros e difíceis de serem conseguidos, sem diminuir a qualidade da compreensão e da realização.

Entre os conceitos mais importantes, estarão os tipos de arquivos musicais (ou sonoros) nos computadores, noções sobre a natureza do som (frequência, amplitude, espectro etc.), dicas de uso de recursos da Internet, noções de arranjo musical e gravação multi-pista

etc. Entre os programas principais, estarão o WINAMP (reprodutor de músicas com muitos recursos úteis e interessantes), o AUDACITY (editor multi-pista de arquivos sonoros), o ANVIL (editor de MIDI muito prático), o NOTEPAD (editor de partituras da mesma empresa que produz o Finale), o BUCKET ON TAB (para tablaturas de guitarra e baixo), o EMULE (para conseguir arquivos de músicas na Internet) etc., todos disponíveis para download gratuito na Internet. A partir deles, é claro, o uso de programas similares pode ser mais aprofundado, ao mesmo tempo que dúvidas mais específicas sobre um programa ou outro também podem ser resolvidas.

Conhecimentos mínimos exigidos: noções de teoria musical (algumas) e leitura de partituras (um pouco); noções de ambiente Windows (uso do mouse, informática elementar etc.). Note-se que os próprios recursos e programas envolvidos no curso possibilitarão também o desenvolvimento destes conteúdos.

1. TEMA

Aplicação da Informática ao Estudo de Conceitos Musicais

1.1. TEMÁTICA

A sensibilidade do ser humano à música é um fator importante no desenvolvimento da percepção musical, facilitando o aprendizado de conceitos e teorias. Uma forma de aprimorar esse conhecimento é através de exercícios repetitivos. Organizar estes exercícios de forma que o aluno sintase cada vez mais motivado a seguir o estudo, maximizando seu aprendizado, é um desafio para qualquer professor de música. A sequência e a forma com que os exercícios são apresentados devem ser definidos com base na avaliação realizada a cada resposta do aluno, o que exige muita perícia do professor e dificulta a implementação de um sistema informatizado para tal fim.

1.2. JUSTIFICATIVA

A Informática tem se apresentado não apenas como uma ferramenta de auxílio no desenvolvimento de tarefas, mas sim como uma tendência em nível de mundo. Vem interferindo agressivamente em todas as áreas do conhecimento, bem como nos variados setores profissionais, sejam públicos ou privados. Direta ou indiretamente, todos fazem uso de algum serviço sobre o qual a tecnologia dos computadores está sendo utilizada. Este fator, entre outros, comprova o fato de que o princípio do uso da informática na música deve ser precedido, sem dúvida, da preparação do docente. É o professor que assume o papel de maestro no processo de troca de paradigmas por que passa o aluno quando se depara com os novos conceitos e softwares relacionados à informática musical, na sua interpretação mais genérica.

A tarefa da percepção musical muitas vezes é difícil de compreender num primeiro contato, por isso são necessários exercícios repetitivos para aprimorar o desempenho dos alunos. São poucas as pessoas que possuem essa técnica de audição, por isso é primordial a necessidade de uma série de exercícios. Através do uso da tecnologia, o aluno além frequentar aulas de música, adquire mais uma ferramenta que poderá auxiliar no seu estudo. A importância do uso de computadores na educação musical não se limita simplesmente ao uso

de uma nova mídia, mas também à divulgação de uma nova forma de se fazer arte: a cultura da tecnologia.

1.3. OBJETIVOS

1.4. OBJETIVO GERAL

- Tem por finalidade desenvolver um software para o ensino de conceitos musicais, utilizando ferramentas necessárias para a prática de exercícios que visam desenvolver esta importante habilidade.

1.5. OBJETIVOS ESPECÍFICOS

- Definir a melhor sequência de apresentação dos intervalos musicais visando um maior aproveitamento do aluno;
- Desenvolver um sistema que seja capaz de apresentar intervalos musicais para o usuário e propor exercícios sobre os mesmos;
- Criar um mecanismo para avaliar o aproveitamento do aluno e indicar se o mesmo está apto a estudar novos intervalos.

1.6. METODOLOGIA

Inicialmente foram estudados conceitos de teoria, percepção musical e como é o processo tradicional ou mais comum de ensino destas disciplinas. Em seguida, foram feitas entrevistas com profissionais da área, para analisar pontos que vão ajudar no desenvolvimento do sistema. Logo após, realizou-se testes com softwares semelhantes analisando suas vantagens e desvantagens para com o usuário.

Depois de levantados o requisito acima citado foi feita a modelagem do software. A implementação do mesmo foi feita em Java, utilizando todos os procedimentos e funções do padrão MIDI (Musical Instrument Digital Interface) para geração de sons. Em seguida foi implementado um banco de dados usando como ferramenta o MySQL. Além disso, foram feitos testes durante a implementação do software, possibilitando que o desenvolvedor verificasse erros ou falhas na aplicação.

2. REFERENCIAL TEÓRICO

2.1. Conceitos básicos da música

Mousikē é um conceito grego que significa “a arte das musas”. E, deste, deriva a palavra música, que define a arte de organizar sensível e logicamente uma combinação coerente de sons e silêncios. Para isso, são usados os princípios fundamentais da melodia, a harmonia e o ritmo.

Enquanto manifestação artística, a música é um produto cultural que procura suscitar uma experiência estética no ouvinte. Também constitui um estímulo no campo perceptivo do homem, uma vez que pode assumir várias funções, como é o caso do entretenimento, do ambiente criado ou da comunicação.

A música como dizia acima, é composta por dois elementos básicos, que são os sons e os silêncios. O som é a sensação captada pelo ouvido, através das variações de pressão produzidas pelo movimento vibratório dos corpos sonoros e que se transmitem pelo ar. A ausência do som é o silêncio, o qual nunca é absoluto, tendo em conta a existência da atmosfera.

O som engloba quatro parâmetros fundamentais: a altura (o resultado da frequência produzido por um corpo sonoro, que determina se o som é grave ou agudo), a duração (o tempo que duram as vibrações que produzem um som), a intensidade (a força com que se produz um som, representada pela amplitude) e o timbre (a qualidade que permite distinguir entre os diferentes instrumentos ou vozes). A organização da música, por sua vez, depende de vários aspectos, entre os quais destacaremos a melodia (o conjunto de sons que soam de forma sucessiva e que são percebidos com uma identidade e sentido próprio), a harmonia (que regula a concordância dos sons e cuja unidade básica é o acorde) e o ritmo (a pauta de repetição dos sons e silêncios).

2.2. Intervalo musical

Intervalo musical é a relação entre as frequências de duas notas. São classificados quanto à simultaneidade ou não dos sons e à distância (altura) entre eles. Na música ocidental, os intervalos são estudados a partir da divisão diatônica da escala. As unidades de medida de intervalos, baseadas na escala logarítmica, são o tom e o semitom. Para intervalos menores que um semitom, são utilizados o savart e o cent (o mais utilizado atualmente). Segundo o

exposto anteriormente, a homossexualidade esteve presente nas diversas culturas das civilizações antigas, bem como, ininterruptamente seguiu a evolução da sociedade. Obviamente, essas relações também estiveram presentes no processo da história brasileira, descobrindo aqui, semelhança e aversão a sua prática.



FIGURA 1- Intervalo de DÓ a SOL

(Fonte: PINTO, 2000)

2.3. Tipos de Intervalos Musicais

Na escala diatônica, a primeira classificação de um intervalo é quanto à ocorrência de simultaneidade em sua execução. Assim, o intervalo será melódico quando os sons aparecerem em sucessão um ao outro, ou harmônico, caso sejam executados no mesmo instante.

2.4. Intervalo Simples e Composto

O intervalo também pode ser simples ou composto, dependendo da distância entre uma e outra nota:

- ♣ Simples: Quando se acha contido dentro de uma oitava.
- ♣ Composto: Quando ultrapassa uma oitava.

2.5. Intervalo melódico

Pode ser classificado quanto:

- ♣ A posição do segundo som em relação ao primeiro. Assim, o intervalo será ascendente se o segundo som for de maior frequência (mais agudo) que o primeiro e será descendente caso o segundo som seja de menor frequência (mais grave) que o primeiro.
- ♣ A distância entre os dois sons. Será conjunto o intervalo que dista de um ou dois semitons (somente o intervalo de segunda) entre as notas e serão disjuntos todos os outros.

2.6. Intervalo harmônico

O intervalo harmônico pode ser classificado somente quanto à distância entre os dois sons.

2.7. Nomes

Os nomes dos intervalos da escala diatônica são dados pela distância vertical entre duas notas – intervalo de segunda entre duas notas seguidas (distância de 2 notas), terça ou terceira entre duas notas cuja distância é de 3 notas, quarta quando a distância é de 4 notas, etc. – mais o designativo que indica se a frequência entre os intervalos são mais ou menos consonantes – intervalo justo ou perfeito, menor, maior, aumentado, diminuto, super aumentado ou super diminuto – chamado também de "qualidade" do intervalo.

QUADRO 1 - Intervalos

Número de semitons	Intervalo diatônico	Abreviatura	Intervalo cromático	Abreviatura	Nomenclatura latina	Abreviatura
0	Uníssonos	1 ^a P	Segunda diminuta	2 ^a d		
1	Segunda menor	2 ^a m	Uníssonos aumentado	1 ^a A	Semitom	S
2 (1 tom)	Segunda maior	2 ^a M	Terceira diminuta	3 ^a d	Tom	T
3	Terceira	3 ^a m	Segunda	2 ^a A		

	menor		aumentada			
4 (2 tons)	Terceira maior	3 ^a M	Quarta diminuta	4 ^a d		
5	Quarta perfeita	4 ^a P	Terceira aumentada	3 ^a A		
6 (3 tons)	Quarta aumentada			4 ^a A	Trítono	TT
	Quinta diminuta			5 ^a d		
7	Quinta perfeita	5 ^a P	Sexta diminuta	6 ^a d		
8 (4 tons)	Sexta menor	6 ^a m	Quinta aumentada	5 ^a A		
9	Sexta maior	6 ^a M	Sétima diminuta	7 ^a d		
10 (5 tons)	Sétima menor	7 ^a m	Sexta aumentada	6 ^a A		
11	Sétima maior	7 ^a M	Oitava diminuta	8 ^a d		
12 (6 tons)	Oitava perfeita	8 ^a P	Sétima aumentada	7 ^a A		

(Fonte: RATTON, 2001)

Nota:

- Um intervalo menor, quando decrescido de um semitom, se transforma em um intervalo diminuto.
- Um intervalo maior, quando acrescido de um semitom, se transforma em um intervalo aumentado.
- Um intervalo diminuto, quando decrescido de um semitom, se transforma em um intervalo super diminuto.
- Um intervalo aumentado, quando acrescido de um semitom, se transforma em um intervalo super aumentado.

- Um intervalo justo ou perfeito, quando decrescido de um semitom, se transforma em um intervalo diminuto.
- Um intervalo justo ou perfeito, quando acrescido de um semitom, se transforma em um intervalo aumentado.

Os intervalos com a mesma distância em semitons mas com nomes diferentes, como por exemplo a quarta aumentada e a quinta diminuta, ou a terça diminuta e a segunda maior, têm o nome de intervalos enarmônicos.

Existe um meio mais racional e fácil de saber a qualidade de um dado intervalo sem ter de contar o número de semitons entre as notas. Basta ter em conta o fato de, na escala diatônica ou natural, a distância entre todas as notas é de um tom, exceto entre as notas Mi e Fá, e Si e Dó, onde o intervalo é de um semitom - são os chamados semitons naturais.

Uma vez identificado onde se localizam esses semitons naturais, basta ter em conta que:

- ♣ Nos intervalos de segunda e terceira são maiores os que não possuem, isto é, não "passam por" nenhum semitom natural.
- ♣ Nos intervalos de sexta e sétima, são maiores os que possuem apenas um semitom natural.
- ♣ Os intervalos de quarta e quinta são todos perfeitos, com exceção do trítone (quarta aumentada ou quinta diminuta).

3. PERCEPÇÃO MUSICAL

A **percepção musical** é a capacidade de perceber as ondas sonoras como parte de uma linguagem musical. Envolve especialmente a identificação dos atributos físicos do som, como volume, timbre e afinação (percepção sonora), mas também elementos musicais como melodia (percepção melódica) e ritmo (percepção rítmica).

Visto como indispensável para musicistas foram desenvolvidos uma série de métodos destinados a aumentar esta capacidade em crianças e adultos. Cursos de música (em universidades, conservatórios ou mesmo em escolas regulares), tipicamente, reservam várias aulas com este propósito.

O termo percepção musical é, muitas vezes, usado como sinônimo de percepção sonora, nesse caso desconsiderando melodia, ritmo e elementos de linguagem musical.

Altos níveis de percepção musical são sinais de apurada capacidade de análise sonora. Ajudam muito, mas não garantem a musicalidade do indivíduo, visto que não tem relação

direta com capacidades de produção sonora. Algumas tarefas esperadas de uma alta percepção musical incluem a identificação de escalas a partir de melodias; acordes e progressões de acordes em trechos musicais; nuances interpretativas; harmônicos; vozes em meio a uma polifonia; e até ruídos indesejáveis em meio a música. O chamado "ouvido absoluto" refere-se à capacidade de identificar e nomear notas musicais apenas pela audição de sons correspondentes.

Dimensão do tempo métrico, ou medido, onde se ouve um tempo 'de metrônomo', um tempo contado por unidades de igual duração (tempo de relógio). Nesta dimensão, o conhecimento técnico, analítico prevalece; pensa-se em cada material da música, como por exemplo, no compasso (binário conta 1 2; no ternário conta 1 2 3), no andamento (andante, rápido, lento), na acentuação, na agógica (acelerando, rallentando), e outros. Dimensão de tempo não métrico, que pode ser vivenciado por meio das palavras da canção, seguindo mais os acentos, entonações das frases do texto do que propriamente da métrica definida do compasso. O tempo é mais flexível, elástico, com componentes de expressividade do conteúdo semântico.

Dimensão da corporeidade, que pode ser métrico ou não, mas que seguirá a intuição auditiva; o corpo 'pensa' e responde gestual mente; a percepção é orientada pelos componentes mental (ou cognitivo), afetivo (ou psicológico) e físico-corporal. O corpo expressa a percepção do tempo que pode suspender e/ou deixar o 'métrico', para realçar outro componente (melodia, acorde, uma suspensão, uma palavra, etc.) que foi sentido ou percebido como significativo para o intérprete e/ou ouvinte.

Dimensão expressivo-individual que é essencialmente não métrico, ou, se há métrica, com certeza está livre da medida 'contada'. É essencialmente flexível e elástica a vivência temporal nesta dimensão. Um exemplo Que acontece frequentemente é quando o ouvinte tem toda a música na mente, e com isso tem a capacidade de 'parar' em uma determinada melodia, evento, ou sonoridade, e lá fica repetindo, voltando, refazendo, curtindo o mesmo em um tempo não-métrico. O tempo, neste caso, é tão variável quanto variável é a concepção do tempo musical.

4. METODOLOGIA DE ENSINO

A missão do educador é indicar os caminhos a ser percorrido, mostrar onde está a informação e como ela pode ser utilizada. Certas áreas do estudo da técnica musical exigem repetições mecânicas contínuas, e ali o professor tem um papel importante incentivando,

avaliando e reportando progressos. É necessário que a informação certa chegue até o aluno certo, dentro de seu nível de capacitação e entendimento.

A educação musical tem nas novas tecnologias, portanto, um aliado vital na tarefa de disponibilizar essas informações. E, apesar de meios como a Internet ou o vídeo serem novidades no ensino da música, a influência que a tecnologia exerce sobre a mesma pode ser notada fortemente ao longo da história. A percepção musical refere-se ao treinamento auditivo. A metodologia de ensino mais adotada é a de conduzir o usuário a aprender a ouvir as notas musicais e reconhecê-las, através de um árduo trabalho de estudo e várias repetições de exercícios (GOULART, 2002).

Segundo pesquisas feitas com profissionais da área musical, a princípio são utilizados apenas intervalos melódicos (reconhecimento das notas de uma melodia, uma a uma) como exercícios, pelo fato de ser mais simples para usuários que estão tendo um primeiro contato com a música. Os exercícios são iniciados por intervalos consonantes (quintas e quartas justas), onde o aluno tem mais facilidade de percepção, seguindo depois com os intervalos de terças e sextas (maiores, menores). Após um bom desempenho nos intervalos, o aluno passa para exercícios dissonantes, que seriam, segundas e sétimas (maiores, menores, etc.) e intervalos diminutos e aumentados. Dentro de cada intervalo devem ser realizados exercícios com intervalos ascendentes e descendentes.

Além disso, é importante que se tenha mudanças de timbres, para que o aluno não fique limitado apenas no mesmo, alternações de oitavas, exercitando ainda mais sua percepção. Após a conclusão desta etapa, o aluno está apto a trabalhar com sons simultâneos (intervalos harmônicos) e ritmos, porém estes aspectos não serão tratados neste trabalho.

5. SOFTWARES SIMILARES

A seguir serão mostrados softwares similares relacionados com percepção musical.

5.1 SOLFEGE

Solfège é um programa livre para treino de audição. O programa é parte do Projeto GNU. Possui versão para diversos sistemas operacionais.

Uma das ideias deste programa é que você possa estender o programa sem ter que mergulhar no código fonte.

Com este programa você poderá treinar diversas atividades musicais como:

- ♣ Intervalos - harmônico-melódicos, ascendente/descendente e até cantar os intervalos;
- ♣ Acordes - acordes na posição fundamental e invertidos, cantar os acordes, etc.;
- ♣ Escalas - diversas escalas (maior, menores, modos gregos, bebop, jazz, etc.);
- ♣ Ritmo - você poderá tocar os ritmos ou escrevê-los após ouvir um exercício;
- ♣ Ditados, progressões harmônicas, etc.
- ♣ Testes;
- ♣ etc.

É um programa completo para o desenvolvimento da percepção musical. Uma ferramenta e tanto para ajudar os alunos e professores no estudo do dia-a-dia



FIGURA 2-Interface do *software* Solfege.

(Fonte: www.solfege.org, acessado em 23 de Março de 2012)

Identificar o intervalo

Solfege irá desempenhar um intervalo harmônico ou melódico, e você deve identificá-lo.

Sing intervalo

Solfege mostra um intervalo na notação de música. A primeira nota é tocada. Você tenta cantar a segunda. A resposta correta é jogado para que você possa comparar.

Ritmo

Solfege brinca com você quatro batidas aleatórias células rítmicas. Você tenta reconhecê-los.

Sing acorde

Solfege mostra-lhe uma corda em notação musical. Um A (440 Hz) é tocada. Você tenta cantar o resto do acorde. A resposta correta é jogado para que você possa comparar.

Balança

Solfege tem mais de 80 exercícios diferentes escalas praticar mais de 70 diferentes escalas.

5.2 GUIDO 1.03

Este software proporciona ao aluno/usuário uma série de opções de estudo, tais como: percepção de intervalos, tríades, tétrades, modos e notas. Além disso, o aluno/usuário tem o privilégio de escolher a nota de início, a velocidade e o volume, facilitando-o no seguimento do exercício. A seguir serão mostradas as principais funções do uso do software. A figura 1 apresenta a tela para o estudo de intervalos deste sistema.

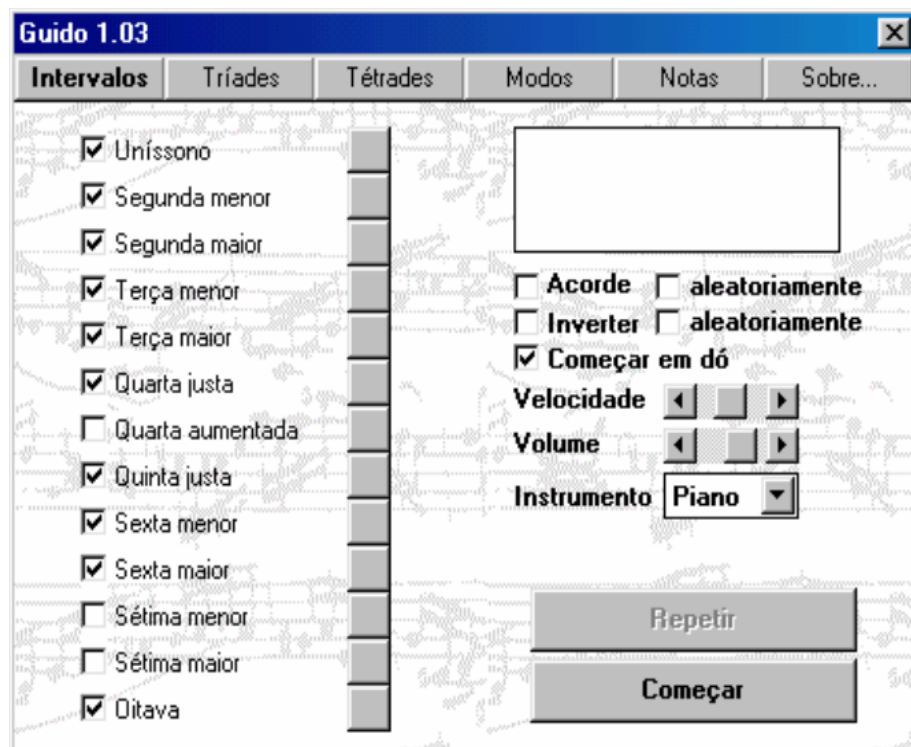


FIGURA 3 - Interface do *software* Guido 1.03.
(Fonte: www.solfege.org, acessado em 23 de Março de 2012)

5.3 Tipos de estudo

Na parte superior da janela, apresentada na figura 2, o usuário pode selecionar o tipo de estudo que vai realizar, entre as opções "Intervalos", "Tríades", "Tétrades", "Modos" e "Notas". Ao ser selecionado um tipo de estudo, a coleção de "checkboxes" correspondente vai aparecer na janela, com as opções possíveis para o estudo. Dependendo do tipo de estudo escolhido, essa operação pode influenciar também outros parâmetros de configuração do programa.

5.4 Modo estudo *versus* modo teste

No modo "estudo", o usuário pode ouvir os intervalos, acordes, modos ou notas, pressionando o botão em frente ao checkbox indicativo. No modo "teste", que tem início quando o usuário pressiona o botão "começar", os botões servem às opções de respostas do estudante para os intervalos, acordes, modos ou notas tocados aleatoriamente pelo computador, que ele deve identificar. Caso o elemento seja identificado corretamente, o computador sorteará um novo, executando-o imediatamente. Em caso de erro, a opção correta será realçada na tela e o aluno deverá então selecioná-la para prosseguir. Os erros do usuário alteram progressivamente o algoritmo de seleção empregado na escolha dos novos elementos, de forma que o computador passa a insistir nos elementos em que o usuário demonstrou maior dificuldade em reconhecer. Durante o modo "estudo", o usuário deverá selecionar nos checkboxes, os intervalos, acordes, modos ou notas que deverão ser perguntados a ele durante o modo "teste". O modo "teste" apenas termina, voltando ao modo "estudo", quando se é pressionado o botão "terminar".

5.5 Modo de teste seletivo

Este modo de teste é selecionado quando apenas um dos checkboxes está selecionado. O computador seleciona qualquer dos intervalos, acordes, modos ou notas, conforme o tipo de teste, e o executa para o usuário, que apenas deverá pressionar o botão correspondente ao checkbox escolhido, caso reconheça tal elemento. Caso não seja pressionado o botão após um período de tempo um novo elemento será escolhido. Por exemplo, se no modo de estudo de intervalos apenas o checkbox referente a "terça menor" estiver selecionado, o usuário deverá apenas reconhecer terças menores, pressionando o botão quando o computador executar tal intervalo.

5.6 Opções do programa

- **Velocidade**

Pode-se executar as notas mais rapidamente ou mais lentamente, conforme a posição desse botão. Testes em maior velocidade são mais difíceis.

- **Volume**

É possível ajustar a intensidade do som com esse controle.

- **Instrumento**

Pode-se selecionar o timbre que o sistema executará as notas. O timbre pode ter influência significativa no resultado dos testes. É interessante variar os timbres para acostumar o ouvido a separar os conceitos de timbre e de altura.

- **Começar em dó**

Quando esta opção está selecionada, a primeira nota de cada intervalo, acorde ou modo é sempre a nota dó. No caso do estudo de notas, a nota dó é executada em primeiro lugar. Caso a opção não seja selecionada, a nota inicial é sorteada pelo computador. É conveniente iniciar o estudo com esta opção, passando progressivamente a não empregá-la, na medida em que o aluno progride no estudo.

- **Acorde**

Esta opção vale apenas para os estudos de intervalos, tríades e tétrades. Quando esta opção está selecionada no estudo de intervalos, são tocados intervalos harmônicos (ambas as notas são tocadas simultaneamente). Quando esta opção não está selecionada nos estudos de tríades ou tétrades, o computador gerará arpejos com as notas dos acordes selecionados.

Quando a opção "aleatoriamente" também está selecionada, o computador escolherá aleatoriamente entre tocar intervalos melódicos ou harmônicos ou entre tocar acordes ou arpejos.

- **Inverter**

Esta opção não vale para o modo de estudo de notas. Quando selecionada, produção simultânea de sons; harmonia. Acordes de sons sucessivos em instrumentos de cordas.

Esta opção provoca a seleção de intervalos descendentes, seja no modo de estudo de intervalos, seja nos modos de estudo de tríades e tétrades em arpejos, seja no estudo de modos. Quando a opção "aleatoriamente" também está selecionada, o computador escolherá aleatoriamente entre tocar intervalos ascendentes ou descendentes.

5.7 Considerações sobre o *software* Guido 1.03

Este software, além de ser bastante completo, traz como vantagens ao aluno/usuário, a facilidade de execução para usuários leigos, seja na área de música ou informática, através de sua interface amigável, mostrando ao aluno/usuário uma série de opções de escolha nos menus, podendo fazer seu próprio controle de estudo. Entretanto, este software traz como principal desvantagem a falta de um tutorial didático, como por exemplo, indicando ao aluno/usuário qual é a forma mais correta para iniciar os exercícios, quais conceitos devem ser aprendidos primeiro e a melhor sequência a seguir. Além disso, não traz nenhuma explicação teórica sobre os exercícios em questão.

5.8 Conclusão

Este capítulo abordou tópicos de teoria musical relacionados com o ensino da mesma e o computador. Cada item tem um significado importante no desenvolvimento do trabalho, tais como: noções de teoria musical, percepção e ensino, ou seja, entender como esses itens são ensinados para modelar o sistema que, por sua vez, são itens fundamentais para usuários com pouca experiência em relação à música. Sendo assim, no momento da execução dos exercícios propostos do software, o aluno/usuário terá todos os conceitos necessários para obter um bom desempenho final. Neste capítulo também foram analisados softwares semelhantes identificando pontos fortes e fracos para aproveitar as ideias no desenvolvimento deste trabalho. Dentre as características positivas encontradas em alguns dos sistemas destacam-se a

interface amigável, como no caso do software memória (figura 3) e diversos menus de opções relacionados com os exercícios como como software Guido 1.03 (figura 1). Como principais desvantagens tem-se uma sequencia incorreta na ordem dos exercícios, falta de um tutorial explicativo, dicas de desempenho e variedades de timbres a serem executados, sendo estas algumas das características que serão incorporadas ao software desenvolvido neste trabalho.

6. FERRAMENTAS PARA DESENVOLVIMENTO

Neste capítulo são apresentadas as ferramentas para o desenvolvimento do software, suas principais funções e quais as vantagens de usá-las neste trabalho.

6.1 MIDI – Musical Instrument Digital Interface

A Musical Instrument Digital Interface (MIDI) é o formato padrão para a música eletrônica. Pode-se criar música MIDI através de um instrumento digital, como um teclado eletrônico ou através de um software. A interface MIDI permite criar música digital sintetizada, ou seja, uma interface que reproduz a música original. Os músicos podem compartilhar suas criações musicais com aficionados por música do mundo inteiro. Um sintetizador MIDI é um dispositivo que pode reproduzir sons e música MIDI (DEITEL, 2003). Quando um arquivo MIDI é lido, a placa de som toca a música, nota por nota, no volume certo e na seqüência correta. Como MIDI é uma linguagem para computador, ela funciona melhor em equipamentos eletrônicos como teclados/sintetizadores. Sintetizador é um instrumento digital ou analógico que processa diversos parâmetros dos sons gerados internamente e os reproduz acionados por teclas ou por comandos MIDI (RATTON, 2001).

A interpretação de dados MIDI varia entre os sintetizadores, de modo que a execução de um arquivo pode soar bem diferente quando reproduzido em sintetizadores diferentes daquele onde foi criado. Os sintetizadores suportam quantidades e tipos de sons instrumentais diferentes. Usualmente, os sintetizadores baseados em hardware (teclados) são capazes de produzir música sintetizada de qualidade mais alta do que os sintetizadores baseados em software (DEITEL, 2003). Depois de cerca de alguns anos, desde o surgimento do MIDI e após muita confusão e reclamação dos usuários, os fabricantes de MIDI, através da International MIDI Association (hoje substituída pela MIDI Manufacturers Association), chegaram a um consenso sobre um formato universal para arquivamento de dados MIDI (sequências), que pudesse ser usado por todos os sequenciadores, tanto para escrita (salvar),

quanto para leitura (carregar). Dessa forma, o arquivo SMF é o meio de intercâmbio de sequências de dados MIDI entre diferentes softwares MIDI e diferentes equipamentos MIDI que implementem sequenciamento (RATTON, 2001).

6.2 Mensagens MIDI

As mensagens MIDI são sequências curtas de bytes que representam eventos musicais. Os eventos estão associados a notas musicais e operações de controle dos instrumentos e da rede. As mensagens de rede comandam principalmente aspectos de sincronização entre instrumentos, enquanto as mensagens de instrumentos são dirigidas a instrumentos individuais (PAULA FILHO, 2000).

6.3 Mensagens que dependem de canal MIDI

São as mensagens que, quando enviadas, afetam apenas o canal MIDI selecionado.

São elas:

- Note On — comando para tocar uma determinada nota, com uma determinada velocidade — velocidade, que na prática significa intensidade.
- Note Off — comando para desligar a nota que está sendo tocada. Também há dentro do comando a ‘velocidade’ de desligamento da nota.
- Aftertouch - um comando que dispara ao colocar pressão na tecla em que toca. Nota-se que, para cada tecla pode ser colocada uma pressão diferente, gerando, portanto, mensagens diferentes.
- Control Change — comandos que podem controlar diversas partes do teclado, como o volume (control 7) e o pedal de sustain (control 64). Basicamente, há 128 comandos para serem usados, sem contar algumas combinações — por exemplo, poderia se dizer que o Control 0 e Control 32 em conjunto formam um ‘Control 129’. É por causa dessa enorme possibilidade de combinações de controles que o padrão MIDI não ficou obsoleto.
- Program Change (patch) — o comando que muda de um timbre para outro, dos que estão dentro do instrumento. Este comando tem um alcance de, no máximo, 128 timbres.

- Channel Pressure — Outra forma de Aftertouch, mas que é ‘monofônica’, isto é, atua indistintamente para todas as notas do teclado. Por ser mais barato de se construir, é a forma de aftertouch mais comum nos teclados mais baratos, e também nos que têm ‘peso’, isto é, imitam a ação do piano acústico.

6.4 Mensagens “transparentes”

São mensagens que normalmente não são manipuladas pelo músico, mas que existem para controle de diversos sistemas:

- Midi Clock — mensagem usada para que dois sequenciadores andem no mesmo andamento. Basicamente, há 24 delas para cada semínima⁵. Numa música de andamento constante, elas irão aparecendo também a uma velocidade constante. Mas num ralentando⁶, por exemplo, elas irão ter um espaçamento maior entre uma e outra. A maior desvantagem dessa mensagem é, que depois de gravar uma música com sincronismo baseado em Midi Clock, qualquer alteração de andamento será impossível. Hoje em dia, ela quase não é mais usada, em favor de outra mensagem.
- Midi Time Code — mensagem que carrega em si o tempo ‘absoluto’ da música. Isto quer dizer que ela vai aparecendo a intervalos regulares, independente do andamento da música. A sua grande vantagem é que, se quiser mudar este andamento depois, ela não atrapalhará em nada, pois estava apenas “marcando o tempo” e o sequenciador poderá se mover livremente por cima dessa marcação. Seu uso mais comum é em conjunto com o padrão SMPTE, de sincronismo de vídeo.

Figura musical de valor metade de uma mínima.

Tornar-se menos compacto.

- Song Position Pointer — Quando tem-se dois sequenciadores funcionando juntos — por exemplo, um computador e uma bateria eletrônica com o ritmo programado internamente, essa mensagem faz com que, ao tocar uma música no meio dela no sequencer mestre, o outro comece a tocar sua parte no mesmo compasso daquele.

6.5 Codificação das notas em MIDI

Quando é pressionada uma tecla no teclado de um instrumento MIDI, este gera um código que identifica a ação (pressionar a tecla) e transmite uma mensagem contendo aquele código. Esta mensagem (ou comando) é chamada de Note On. No caso exemplificado (ato de pressionar uma tecla), o código gerado é dividido em três bytes: um byte de status, que designa o tipo de código ou mensagem, e dois bytes de dados, que contêm os parâmetros da mensagem. Os dois bytes de dados indicam, respectivamente, o número da tecla pressionada (ou seja, qual a nota que foi executada) e qual a intensidade (força) com que ela foi pressionada. O número da nota pode ser qualquer valor desde 0 a 127 (128 notas no total; cobrindo mais do que 10 oitavas), sendo que a nota dó₄, que é o dó abaixo do lá central (A440) do piano, tem o número 60 (ou seja: o dó₃ é igual a 48, e o dó₅ é igual a 72). O códigos e o valores de algumas notas podem ser visto no quadro 2. O valor da intensidade com que a nota foi pressionada é conhecido pelo nome de "key velocity" (velocidade da tecla), porque quanto mais rápido se pressionar a tecla, mais forte será o seu som. De fato, os teclados eletrônicos detectam a velocidade do movimento da tecla, dentro de uma escala que vai de 0 a 127 e a usam para designar a intensidade do toque.

QUADRO 2 - Relação de algumas notas com seus respectivos valores

Nota	Nº Oitava	Nº Nota
Dó	3	48
--	--	--
Dó	4	60
Dó# / Réb	4	61
Ré	4	62
Ré# / Mib	4	63
Mi	4	64
Fá	4	65
Fá# / Solb	4	66
Sol	4	67
Sol# / Láb	4	68
Lá	4	69
Lá# / Sib	4	70
Sí	4	71

Dó	5	72
-----------	----------	-----------

(Fonte: RATTON, 2001)

Da mesma forma que ao se pressionar uma tecla é gerado um código correspondente, quando se solta a tecla do instrumento é gerado um outro código, que identifica qual nota foi solta e o valor da velocidade com que ela foi solta (key-off velocity). A maioria dos teclados, no entanto, não detecta a key-off velocity, e adota o valor 0 para este parâmetro. A mensagem gerada quando se solta a tecla é chamada de note off (RATTON, 2001). É através destas mensagens que o programador/usuário pode gerar sons via MIDI em um computador que possua uma placa de som.

6.6 Numeração de timbres

Outro aspecto importante do padrão MIDI é a definição de uma ordem fixa para os timbres do equipamento. A falta de uma ordem como essa sempre foi um enorme empecilho para o intercâmbio de músicas em formato MIDI, pois se a sequência continha comandos de troca de timbres (program change), os números dos timbres existentes nesses comandos acabavam selecionando timbres errados em outros equipamentos. Preocupada com esse sério problema, a Roland⁷, tratou de criar seu padrão de numeração, que foi adotado em praticamente toda a sua linha de instrumentos, tornando-os plenamente compatíveis entre si e possibilitando então, a execução correta de músicas MIDI em qualquer sintetizador Roland. Embora os números dos programas MIDI sejam ordenados de 0 a 127, no painel dos instrumentos os timbres são numerados de formas diferentes, como mostra o quadro 3.

QUADRO 3 - Relação entre os fabricantes e as ordens dos timbres

Numeração MIDI	0 a 127
E-mu Proteus	0 a 127
Korg N5	0 a 99
Korg NS5R	1 a 128
Roland JX-305	A11 a A18; A21 a A28; A31 a A38; etc.

Roland XP – 60	1 a 128
-----------------------	----------------

(Fonte: RATTON, 2001)

A ordenação de timbres adotou certos critérios que, embora criticados por alguns, têm uma lógica bem clara. Em primeiro lugar, foram criadas dezesseis “famílias” (bancos de timbres), cada uma com oito timbres, são elas: pianos, percussão cromática, órgãos, violões e guitarras, baixos, instrumentos individuais de orquestra, conjuntos, metais, palhetas, flautas, sons sintéticos para solo, sons sintéticos para “cama” (pads), efeitos sintéticos, timbres étnicos, timbres percussivos, e efeitos sonoros. No quadro 4, tem-se a lista de alguns dos timbres e de algumas famílias do padrão MIDI, sendo importante observar que uma lista completa contém 128 timbres, numerados de 0 a 127.

QUADRO 4 - Alguns n° de timbres, com suas respectivas descrições

<i>Piano</i> Cód. / Descrição do Timbre	<i>Guitar</i> Cód. / Descrição do Timbre	<i>Ensemble</i> Cód. / Descrição do Timbre
0 = Acoustic Grand Piano 1 = Bright Acoustic Piano 2 = Electric Grand Piano 3 = Honky-tonk Piano 4 = Rhodes Piano 5 = Chorused Piano 6 = Harpsichord 7 = Clavinet	24 = Acoustic Guitar (nylon) 25 = Acoustic Guitar (steel) 26 = Electric Guitar (jazz) 27 = Electric Guitar (clean) 28 = Electric Guitar (muted) 29 = Overdriven Guitar 30 = Distortion Guitar 31 = Guitar Harmonics	48 = String Ensemble 1 49 = String Ensemble 2 50 = SynthStrings 1 51 = SynthStrings 2 52 = Choir Aahs 53 = Voice Oohs 54 = Synth Voice 55 = Orchestra Hit
<i>Chromatic Percussion</i> Cód. / Descrição do Timbre	<i>Bass</i> Cód. / Descrição do Timbre	<i>Brass</i> Cód. / Descrição do Timbre
8 = Celesta	32 = Acoustic Bass	56 = Trumpet

9 = Glockenspiel	33 = Electric Bass (finger)	57 = Trombone
10 = Music Box	34 = Electric Bass (pick)	58 = Tuba
11 = Vibraphone	35 = Fretless Bass	59 = Muted Trumpet
12 = Marimba	36 = Slap Bass 1	60 = French Horn
13 = Xylophone	37 = Slap Bass 2	61 = Brass Section
14 = Tubular Bells	38 = Synth Bass 1	62 = Synth Brass 1
15 = Dulcimer	39 = Synth Bass 2	63 = Synth Brass 2
<i>Organ</i>	<i>Strings</i>	<i>Reed</i>
Cód. / Descrição do	Cód. / Descrição do	Cód. / Descrição do
Timbre	Timbre	Timbre
16 = Hammond Organ	40 = Violin	64 = Soprano Sax
17 = Percussive Organ	41 = Viola	65 = Alto Sax
18 = Rock Organ	42 = Cello	66 = Tenor Sax
19 = Church Organ	43 = Contrabass	67 = Baritone Sax
20 = Reed Organ	44 = Tremelo Strings	68 = Oboe
21 = Accordion	45 = Pizzicato Strings	69 = English Horn
22 = Harmonica	46 = Orchestral Harp	70 = Bassoon
23 = Tango Accordion	47 = Timpani	71 = Clarinet

(Fonte: RATTON, 2001)

O padrão MIDI também determina que o equipamento tenha 16 partes timbrais simultâneas, que devem ser capazes de receber comandos MIDI por 16 canais diferentes (sendo obrigatório que todas as 16 partes timbrais reconheçam ajustes de volume via MIDI). O padrão ainda determina que o canal de MIDI nº.10 seja usado exclusivamente para bateria e/ou percussão. Isso significa que, numa sequência feita para ser tocada por instrumentos compatíveis com padrão MIDI, a execução da bateria/percussão deve estar sempre no canal 10. Além disso, o padrão MIDI também estipulou a posição dos instrumentos de percussão ao longo do teclado, isto é, determinou quais as teclas que disparam os instrumentos de percussão. A numeração das notas MIDI vai de 0 a 127 (total de 128 notas), sendo que a nota nº 36 é o Dó 2. As posições dos instrumentos de percussão no teclado General MIDI estão listadas no quadro 5, porém, este não é o objetivo do nosso trabalho.

QUADRO 5 - Ordem dos instrumentos de percussão no teclado

<i>Percurssion</i> Cód. / Descrição do Timbre	<i>Percurssion</i> Cód. / Descrição do Timbre	<i>Percurssion</i> Cód. / Descrição do Timbre
35 = Acoustic Bass Drum	50 = High Tom	65 = High Timbale
36 = Bass Drum 1	51 = Ride Cymbal 1	66 = Low Timbale
37 = Side Kick	52 = Chinese Cymbal	67 = High Agogo
38 = Acoustic Snare	53 = Ride Bell	68 = Low Agogo
39 = Hand Clap	54 = Tambourine	69 = Cabasa
40 = Electric Snare	55 = Splash Cymbal	70 = Maracas
41 = Low Floor Tom	56 = Cowbell	71 = Short Whistle
42 = Closed High-Hat	57 = Crash Cymbal 2	72 = Long Whistle
43 = High Floor Tom	58 = Vibrastrap	73 = Short Guiro
44 = Pedal High Hat	59 = Ride Cymbal 2	75 = Claves
45 = Low Tom	60 = High Bongo	76 = High Wood Block
46 = Open High Hat	61 = Low Bongo	77 = Low Wood Block
47 = Low-Mid Tom	62 = Mute High Conga	78 = Mute Cuica
48 = High-Mid Tom	63 = Open High Conga	79 = Open Cuica
49 = Crash Cymbal 1	64 = Low Conga	80 = Mute Triangle
		81 = Open Triangle

(Fonte: RATTON, 2001)

6.7 Codificando uma sequência MIDI

As figuras apresentadas a seguir foram extraídas do sequenciador Cakewalk8. Na figura 6 é apresentada uma sequência de notas em um compasso quaternário.



FIGURA 4 - Exemplo partitura.**(Fonte: PINTO, 2000)**

A maioria dos sequenciadores mostra a nota e lugar que ela começa (em termos de compasso, tempo e a subdivisão de tempo, chamada tick) e quanto esta nota dura (também em termos de compasso, tempo e ticks). O sequenciador deveria entender: o Dó no primeiro tempo, o Mi no segundo e, finalmente, um acorde de três notas que dura os dois últimos tempos do compasso. Esta sequência de notas é representada de acordo com a figura 7.

Event List - Track 1							
Trk	Hr:Mn:Sc:Fr	Meas:Beat:Tick	Chn	Kind	Values		
1	00:00:00:00	1:1:000	1	Note	C 5	83	119
1	00:00:00:18	1:2:000	1	Note	E 5	95	119
1	00:00:01:06	1:3:000	1	Note	E 5	86	2:000
1	00:00:01:06	1:3:000	1	Note	C 5	101	2:000
1	00:00:01:06	1:3:000	1	Note	G 5	101	2:000

FIGURA 5 - Lista de eventos sob uma única trilha**(Fonte: RATTON, 2001)**

A seguir é detalhado o conteúdo de cada coluna:

- Values: indica o nome da nota, a velocidade e finalmente sua duração. O primeiro Dó então dura 119 espaços, praticamente igual a uma semínima, bem como o Mi;
- Kind: tipo de evento apresentado, no caso Note (nota);
- Chn: canal MIDI em que esses comandos estão gravados;
- Meas, Beat, Tick: compasso atual, tempo dentro do compasso e tick (na figura 6, uma nota no começo do primeiro, outra no segundo e três no terceiro tempo);
- Hr, Mn, Sc, Fr: mostra em termos de tempo de relógio, isto é, horas, minutos, 30 segundos e frames, quando as notas aparecem;
- Trk: track 1 (trilha 1) do sequenciador . Suponha que se deseja mudar o som do instrumento na hora em que fosse tocar o acorde: isso faz-se com o comando program change. Na figura 8, o timbre é alterado para o número 2 (que neste teclado corresponde ao som chamado 'Celesta').

Event List - Track 1						
Trk	Hr:Mn:Sc:Fr	Meas:Beat:Tick	Chn	Kind	Values	
1	00:00:00:00	1:1:000	1	Note	C 5	83 119
1	00:00:00:18	1:2:000	1	Note	E 5	95 119
1	00:00:01:06	1:3:000	1	Patch	2 Celesta	
1	00:00:01:06	1:3:000	1	Note	E 5	86 2:000
1	00:00:01:06	1:3:000	1	Note	C 5	101 2:000
1	00:00:01:06	1:3:000	1	Note	G 5	101 2:000

FIGURA 6 - Lista de eventos sob program change.

(Fonte: RATTON, 2001)

Esta mensagem permite que se escolha um dentre um grupo de 128 timbres (quando a especificação de MIDI foi criada, em 1983, achava-se que os instrumentos teriam no máximo 128 timbres armazenados em sua memória interna). Essa mensagem de program change contém uma indicação do canal de MIDI em que ela está sendo transmitida e o número do timbre (patch) que está sendo selecionado. A seguir, a figura 9 define a coluna que identifica o tipo de mensagem que está especificado contrl (Kind), ou seja, tipo de evento, no caso control change. Tem em seguida sua identificação (no 7, ou seja, volume) seguida do valor. No caso o volume está no máximo e sendo baixado (PINTO, 2000).

Event List - Track 1						
Trk	Hr:Mn:Sc:Fr	Meas:Beat:Tick	Chn	Kind	Values	
1	00:00:00:00	1:1:000	1	Note	C 5	83 119
1	00:00:00:18	1:2:000	1	Note	E 5	95 119
1	00:00:01:06	1:3:000	1	Patch	2 Celesta	
1	00:00:01:06	1:3:000	1	Note	E 5	86 2:000
1	00:00:01:06	1:3:000	1	Note	C 5	101 2:000
1	00:00:01:06	1:3:000	1	Note	G 5	101 2:000
1	00:00:01:06	1:3:000	1	Contrl	7-Volume	127
1	00:00:01:08	1:3:016	1	Contrl	7-Volume	114
1	00:00:01:10	1:3:028	1	Contrl	7-Volume	103
1	00:00:01:14	1:3:050	1	Contrl	7-Volume	91

FIGURA 7 - Lista de eventos sob control change

(Fonte: RATTON, 2001)

Dessa forma, usando o control change nº 7, pode-se transmitir via MIDI uma faixa de 128 valores possíveis de volume, podendo estar de 0 (volume mínimo) a 127 (volume máximo). Entretanto, embora seja possível ajustar 128 valores via MIDI, isso não quer dizer que o equipamento receptor possua 128 níveis de volume (embora a maioria possua). Isso quer dizer que para um equipamento onde o volume pode ter 64 níveis diferentes, cada dois valores de control change nº 7 designam um único valor de volume.

7. JAVA

Java é tanto uma linguagem de programação de alto nível quanto uma plataforma. É uma linguagem orientada a objetos, independente de arquitetura, portátil, robusta, segura, interpretada e distribuída. Além disso, Java é tanto compilada quanto interpretada (DEITEL, 2003). Dentre algumas das vantagens da linguagem java estão suas bibliotecas em tempo de execução que visam proporcionar independência de plataforma: é possível usar o mesmo código nos sistemas Windows 95/98/NT, Solaris, Unix, Macintosh, etc. Isto certamente é necessário para a programação na Internet. Outra vantagem de programação é que a linguagem Java tem uma sintaxe similar a C++ (HORSTMANN e CORNELL, 2001). O Java é uma linguagem orientada a objetos, mais ainda do que o C++ (DEITEL, 2003). Tudo em Java, com exceção de alguns poucos tipos básicos como números, são objetos. O projeto orientado a objetos substituiu técnicas estruturadas anteriores porque ele tem muitas vantagens ao lidar com projetos mais complexos.

7.1 Java e multimídia

A “nova onda” de Java é a multimídia: como usar o som, imagens gráficas e vídeo para fazer os aplicativos “ganharem vida”. A linguagem Java, por meio de pacotes de classes, que são uma parte do mundo de programação de Java, fornece extensas instalações multimídia que permitirão ao usuário começar a desenvolver imediatamente aplicativos de multimídia poderosos. Existem também extensões padrões do Java, uma delas é denominada Java Media Framework (JMF), que fornece melhor processamento de imagens e reprodução de áudio aprimorada, além de suportar muitos formatos de áudio atuais. Versões futuras do

JMF incluirão muitos recursos novos como a capacidade de gravar áudio e vídeo (DEITEL, 2003).

7.2 O pacote Java Sound

Muitos dos programas de computador atualmente atraem a atenção do usuário com recursos de áudio. Até mesmo applets⁹ e aplicativos básicos podem melhorar a experiência do usuário com sons ou clipes simples de música. Com interfaces de programação de som, os desenvolvedores podem criar aplicativos que reproduzem sons em resposta às interações do usuário. Por exemplo, em muitos aplicativos, quando ocorre um erro e uma caixa de diálogo aparece na tela, esta é frequentemente acompanhado por um som. Os usuários, portanto, recebem tanto indicações visuais quanto auditivas de que ocorreu um problema. Como outro exemplo, os programadores de jogos usam recursos de áudio extensivos para melhorar a interação com os jogadores. A API Java Sound é uma maneira de incorporar mídia de áudio em 9 Programas em Java que podem ser embutidos em documentos HTML. aplicativos mais simples do que o Java Media Framework. A API (Application Program Interface) Java Sound vem junto com o Java 2 Software Development Kit versão 1.3. A API JavaSound consiste em quatro pacotes: `javax.sound.midi`, `javax.sound.midi.spi`, `javax.sound.sampled` e `javax.sound.sampled.spi`. Os pacotes `javax.sound.midi` e `javax.sound.sampled` fornecem classes e interfaces para acessar, manipular e reproduzir áudio no formato MIDI e como amostra (formatos tipo `.wav`, `.au`, `.aiff`) respectivamente. Os pacotes que terminam com `.spi` fornecem aos desenvolvedores ferramentas para adicionar o suporte ao Java Sound a formatos de áudio adicionais. A API Java Sound fornece acesso a Java Sound Engine, que cria áudio digitalizado e captura mídia a partir dos dispositivos de som. O Java Sound exige uma placa de som para reproduzir áudio. Caso contrário, um programa que usa tal API irá disparar uma exceção se ele acessar recursos de áudio do sistema (DEITEL, 2003).

7.3 Classes do pacote Java Sound

A seguir serão descritos alguns conceitos e métodos de algumas das principais classes do pacote Java Sound.

7.4 Classe Synthesizer

A classe Synthesizer fornece recursos para gerar sons e transmití-los para outros dispositivos MIDI, como gravadores. Ela usa um objeto que implementa a interface Synthesizer (uma subinterface de MidiDevice), para acessar a geração de sons, instrumentos, recursos de canal e bancos de som padrão do sintetizador. Um sintetizador (Synthesizer) inclui métodos para incluir vários timbres de instrumentos que estão em um pacote chamado SoundBank. Um SoundBank é uma coleção de instrumentos, organizada pelo banco e pelo número do programa (através do objeto Patch do instrumento). A seguir, alguns dos métodos utilizados nessa classe:

- `getChannels ()`: Obtém o conjunto dos canais de MIDI controlados por este synthesizer;
- `loadAllInstruments (Soundbank soundbank)`: contém todos os timbres de instrumentos armazenados em um soundbank;
- `loadInstrument (Instrument instrument)`: carrega um instrumento em particular do soundbank.

7.5 Classe ShortMessages

Os métodos dessa classe são instruções musicais explícitas, como as notas específicas a tocar e mudanças de frequência. A seguir, alguns dos métodos utilizados nessa classe:

- `note_On`: valor do comando para a nota na mensagem;
- `note_Of`: valor do comando para a nota fora da mensagem;
- `start`: byte de status para a mensagem do começo.

7.6 Classe Receiver

Receiver é um receptor de eventos MIDI. A seguir, alguns dos métodos utilizados nessa classe:

- `close ()`: Indica que a aplicação terminou de usar o receptor, e que recursos limitados podem ser liberados ou estarem disponíveis;
- `send (MidiMessage message, long timeStamp)`: emite uma mensagem e um timeStamp (tempo exato) de MIDI a este receptor.

7.7 Classe MidiSystem

MidiSystem é uma classe que fornece o acesso aos recursos de sistema instalados de MIDI, incluindo dispositivos como synthesizers, sequencers, e a entrada e saída de portas MIDI. Esta classe tem também métodos para ler arquivos URLs que contêm dados ou soundbanks padrão de MIDI. A seguir, alguns dos métodos utilizados nessa classe:

- `getMidiDevice(MidiDevice.Info info)`: obtém o dispositivo pedido de MIDI;
- `getMidiFileFormat(File file)`: obtém o formato de um arquivo MIDI especificado;
- `getReceiver()`: obtém um receptor de MIDI de uma porta externa ou de outro de dispositivo MIDI do sistema;
- `getSequence(File file)`: obtém uma seqüência de MIDI, de um arquivo especificado;.
- `getSoundbank(File file)`: constrói um Soundbank lendo um arquivo MIDI especificado;
- `getSynthesizer()`: obtém o synthesizer do efeito.
- `getTransmitter()`: obtém um transmissor de MIDI de uma porta externa ou de outro dispositivo MIDI do sistema.

7.8 Conclusão

Neste capítulo foram estudados conceitos sobre MIDI e Java. Cada um dos tópicos citados neste capítulo tem extrema importância no desenvolvimento do sistema, como por exemplo, noções sobre mensagens MIDI, como suas principais funções, numeração de timbres do padrão MIDI, codificação de notas em MIDI utilizando como exemplo um software musical para melhor entendimento. Além disso, foram estudados classes e métodos prontos da API Java Sound, utilizados para geração de som, tornando-se um ponto central da programação orientada a objetos melhorando assim a portabilidade do programa. Sendo que, na execução do software o usuário usufruirá diretamente das técnicas utilizadas na criação do software e será através destes objetos é que serão gerados os sons correspondentes nos exercícios.

8. MODELAGEM DO SISTEMA

Neste capítulo são apresentados os requisitos funcionais do sistema, através de diagramas Use-Case e pela descrição dos cenários. Um diagrama de classes apresentará arquitetura do sistema, descrevendo as classes utilizadas na implementação com seus relacionamentos.

8.1 UML (*Unified Modeling Language*)

Esta é uma linguagem padrão para elaboração da estrutura de projetos de software. A UML pode ser empregada para visualização, especificação, construção e documentação de artefatos que façam uso de sistemas complexos de software (BOOCH, RUMBAUGH e JACOBSON, 2000).

8.2 Especificação dos requisitos funcionais

Problema: Existe uma necessidade por parte de usuários aprender a reconhecer as notas através da percepção musical.

Objetivo: Desenvolver um software que auxilie os usuários a reconhecer intervalos musicais através dos sons das notas. O software de percepção de intervalos deverá ser capaz de:

- Gerar o som das notas;
- Elaborar exercícios com diferentes graus de dificuldade;
- Avaliar o aluno e verificar se o mesmo assimilou os conceitos envolvidos nos exercícios executados;
- Permitir que o aluno defina o grau de dificuldade que deseja iniciar, definindo os tipos de intervalos, números de oitavas e timbres dos instrumentos envolvidos;
- Em caso de término dos exercícios, o sistema deverá gravar em um banco de dados, exatamente em que grau de dificuldade o aluno/usuário terminou;
- Gerar um relatório que mostra o desempenho final do aluno.

Os requisitos funcionais foram definidos levando-se em consideração a observação de conceitos de teoria musical que o sistema deve possuir. Além disso, foram utilizadas informações obtidas em entrevistas com profissionais da área.

8.3 Diagrama de *Use Case*

Com base nas funcionalidades do sistema foi definido o Diagrama de Use- Case apresentado na figura 10, permitindo um melhor entendimento do sistema com relação a quem vai utilizá-lo.

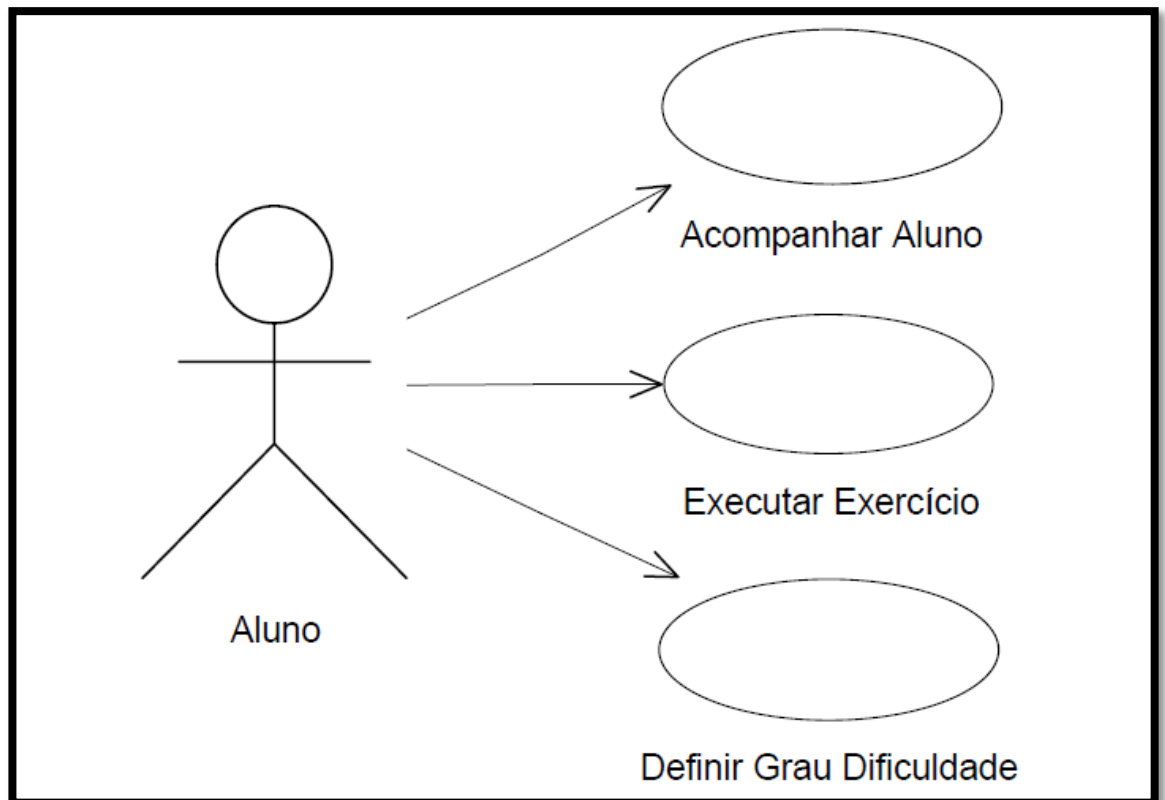


FIGURA 8 - Especificação dos requisitos do usuário aluno através de Use Cases

(Fonte: Próprio Autor, 2012)

O usuário Aluno tem um papel fundamental na operacionalização do sistema, uma vez que ele é o responsável pela inclusão de seus dados e pela execução dos exercícios.

8.4 Descrição dos cenários

Os cenários se caracterizam como uma descrição textual dos requisitos funcionais. Assim, abaixo são representadas as descrições dos processos identificados, onde os fluxos principais representam a sequência padrão de ações no cenário e os fluxos alternativos representam a sequência de ações que deverão ser executadas caso alguma exceção aconteça.

- **Cenário: Acompanhar Aluno**

Pré-Condição: O aluno já deve estar cadastrado. Fluxo Principal:

1. O aluno digita seu nome;
2. O sistema gera um relatório com o desempenho final do aluno referente aos exercícios já executados

- **Cenário: Executar Exercícios**

Fluxo Principal:

1. O aluno escolhe o módulo executar exercício;
2. O sistema gera exemplos de intervalos de acordo com o módulo em questão;
3. O sistema inicia a execução dos exercícios;
4. O aluno responde os exercícios;
5. O sistema registra o resultado da avaliação daquele exercício.

- **Cenário: Definir Grau Dificuldade**

Fluxo Principal:

- O aluno seleciona o grau de dificuldade antes de começar os exercícios;
- O sistema carrega o exercício escolhido.

8.5 Diagrama de classes

Segundo MATOS (2002), em qualquer aplicação orientada a objetos, classes são importantes por quatro motivos:

- Na verdade, são reais estruturas de dados, ou seja, são o objetivo final para o início da etapa de programação;
- Dar ao programador a noção do domínio do problema;
- Projetar uma solução de implementação ao problema;
- Traçar perspectivas de escalabilidade ao sistema (onde e como o sistema pode crescer em termos funcionais). Com base nos conceitos acima o diagrama de Classes ficará como apresentado na figura 11.

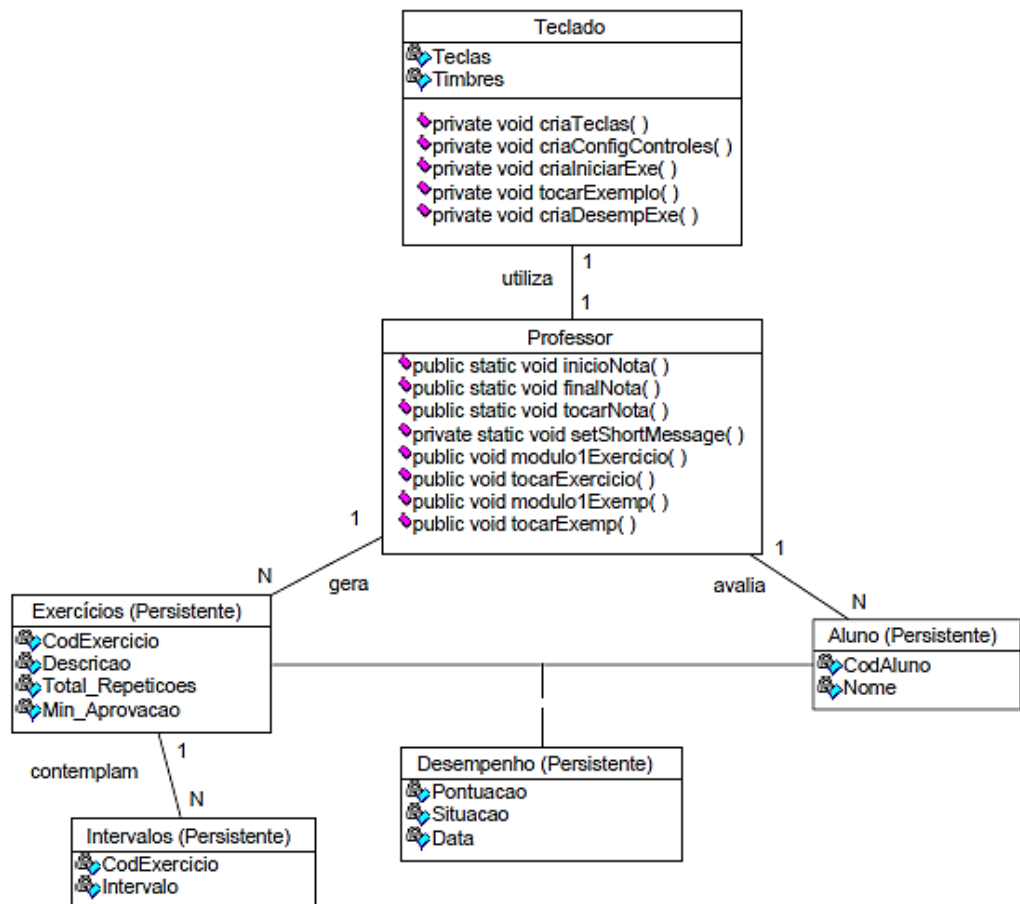


FIGURA 11 - Diagrama de classes.

(Fonte: Próprio Autor, 2012)

As descrições de cada classe serão descritas do quadro 6 a seguir.

QUADRO 6 - Descrição das classes.

Nome da Classe	Descrição
Teclado	Classe que desenha as teclas do teclado virtual e faz com que gere som sobre as mesmas.
Professor	Classe que implementa todos os exercícios referentes aos módulos
Exercícios	Classe que fornece dados referentes aos exercícios.
Desempenho	Classe que fornece dados referentes aos resultados.
Aluno	Classe que armazena dados referentes ao usuário/aluno.
Intervalos	Classe fornece dados sobre os tipos de intervalos.

(Fonte: Próprio Autor, 2012)

8.6 Estrutura dos exercícios

Com base em entrevistas com professores e profissionais na área de teoria musical, os exercícios serão divididos em 40 módulos, de acordo com o grau de dificuldade, do mais “fácil” para o mais “difícil”. Em cada módulo o aluno terá o privilégio de ouvir os sons dos intervalos respectivos, que servirá como uma ajuda, antes de iniciar o exercício. A partir do momento que o aluno atingir sua pontuação (meta especificada pelo programa) ele passará automaticamente para o exercício seguinte, caso contrário, deverá reiniciar o exercício. Todos os exercícios serão executados da seguinte forma:

- O aluno iniciará o exercício ouvindo uma apresentação dos intervalos das notas, que ele deverá aprender. Logo após serão executados os testes. A cada teste o aluno terá duas chances para “acertar” o intervalo tocado. Cada módulo terá uma quantidade de exercícios diferentes bem como diferentes pontuações mínimas para a aprovação. Acertando na primeira chance, o aluno acumulará 10 pontos. Acertando na segunda chance, o aluno acumulará 7 pontos. Não acertando em nenhuma das chances disponíveis, o aluno não acumulará nenhum ponto naquele exercício.
- Se o aluno for aprovado ele passará para o próximo módulo, caso contrário, o sistema registra seus dados e repete o exercício do módulo atual. Os módulos serão organizados da seguinte forma:
- Módulo 1: reconhecimento de intervalos de quintas. O objetivo deste módulo é fazer com o que o aluno seja capaz de reconhecer e diferenciar quintas (justa, aumentada e diminuta), dividido em sub-módulos, trabalhando inicialmente com intervalos ascendentes, em seguida descendentes e, por fim, alternados.

- Módulo 2: reconhecimento de quartas. O objetivo deste módulo é também fazer com que o aluno seja capaz de reconhecer e diferenciar quartas (justa e aumentada), dividido em sub-módulos, iniciando com intervalos ascendentes, em seguida descendentes e por fim alternados.
- Módulo 3: reconhecimento de intervalos de oitava. O objetivo é fazer com que o aluno seja capaz de reconhecer um intervalo de oitava.
- Módulo 4: reconhecimento de intervalos de quartas, quintas e oitava. O objetivo deste módulo é fazer com o que aluno reconheça e diferencie os intervalos de quartas, quintas e oitavas, em todas suas possibilidades (justas, aumentadas e diminutas).
- Módulo 5: reconhecimento de intervalos de terças. O objetivo deste módulo é também fazer com que o aluno seja capaz de reconhecer e diferenciar terças (maior e menor). Também dividido em sub-módulos, trabalhando apenas com intervalos ascendentes, em seguida descendentes e, por fim, alternados.
- Módulo 6: reconhecimento de intervalos de sextas: o processo é semelhante ao do módulo de reconhecimento de terças.
- Módulo 7: reconhecimento de intervalos de terças, quartas, quintas, sextas e oitavas, todas alternadas.
- Módulo 8: reconhecimento de intervalos de segundas. O objetivo deste módulo é também fazer com que o aluno seja capaz de reconhecer e diferenciar segundas (maior, menor e aumentada), dividido em sub-módulos, trabalhando apenas com intervalos ascendentes, em seguida descendentes e, por fim, alternados.
- Módulo 9: reconhecimento de intervalos de sétimas. O objetivo deste módulo é fazer com que o aluno seja capaz de reconhecer e diferenciar sétimas (maior e menor), dividido em sub-módulos, trabalhando apenas com intervalos ascendentes, em seguida descendentes e, por fim, alternados.
- Módulo 10: terminado esta série de módulos o aluno passará para este módulo, que é o reconhecimento de todos os intervalos, justos, ascendentes, descendentes e alternados. É importante destacar que, nestes dez módulos, os intervalos serão gerados apenas em duas oitavas e com timbre fixo. Passando deste módulo, o aluno iniciará a etapa seguinte que é a repetição dos 10 primeiros módulos, porém utilizando agora 4 oitavas e com timbre fixo. Logo após a etapa seguinte, serão repetidos todos os intervalos utilizando quatro oitavas, mas com timbre livre, e por fim, a última etapa consiste na execução dos 10 módulos com timbre livre e 6 oitavas. Desta forma, o aluno deverá

cumprir 40 módulos para terminar o curso. Durante a execução ou após o término de todos os módulos o aluno poderá obter seu desempenho final através de um relatório gerado pelo sistema.

8.7 Conclusão

Neste capítulo foram estudados e implementados métodos para a modelagem do sistema. Dentre eles, conceito e utilização da linguagem UML para elaboração da estrutura de um projeto, diagrama de Use-Case e Diagrama de Classes, para melhor visualização da estrutura e do funcionamento do software. Além disso, um dos tópicos importantes neste capítulo é a estrutura dos exercícios, que serve como um tutorial explicativo para melhor entendimento dos usuários do sistema. A estrutura definida para os exercícios procura guiar o aluno durante seu estudo, visando facilitar o seu aprendizado com base na experiência dos professores da área. Isto se torna um diferencial deste programa em relação a outros com o mesmo objetivo, já que nestes últimos muitas vezes o aluno começa o estudo por intervalos de mais difícil assimilação, tornando o processo de aprendizagem mais lento, se desmotivando e por fim desistindo.

9. APRESENTAÇÃO DO SISTEMA

Este capítulo trata da implementação do sistema, apresentando suas telas com comentários sobre os módulos que compõe o sistema.

9.1 Interface do sistema

Na figura 12 é apresentada a tela principal do software e as funções que podem ser executadas através dela

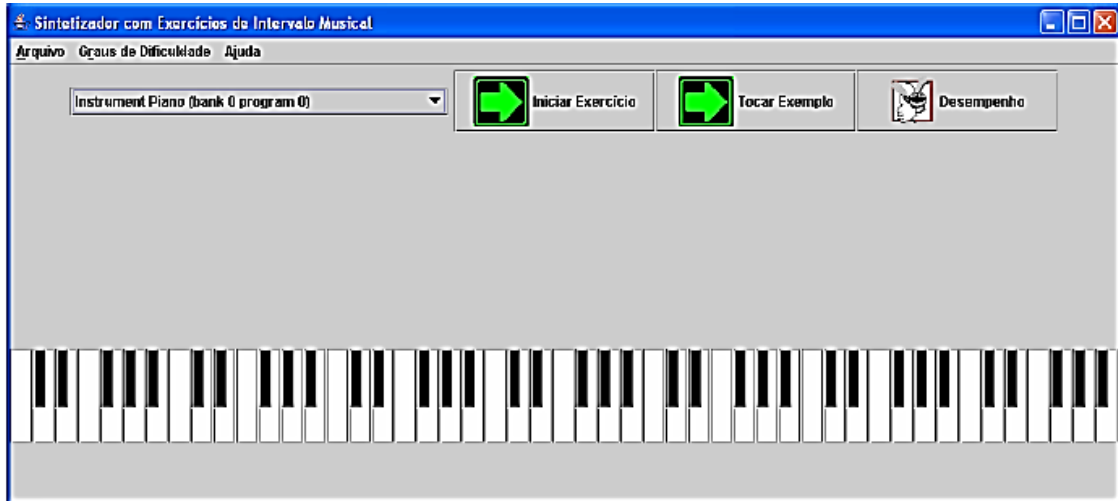


FIGURA 10 - Tela principal do *software*

(Fonte: Próprio Autor, 2012)

Através do teclado virtual, que se encontra na parte inferior da tela principal, o aluno tem a possibilidade de tocar as notas para, simplesmente, ouvi-las como em um teclado real. Além disso, através desta interface, o aluno pode:

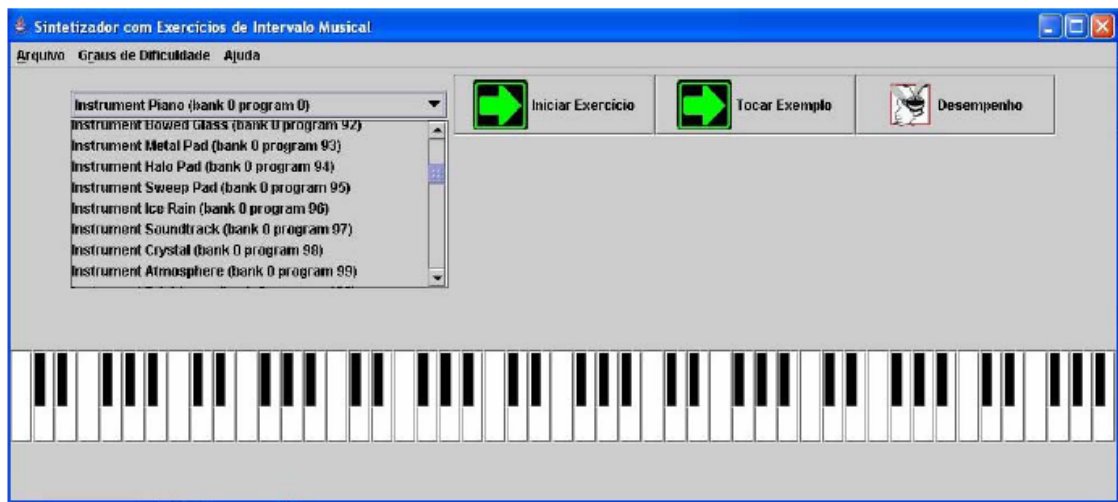


FIGURA 11 - Amostra de timbres inclusos no software.

(Fonte: Próprio Autor, 2012)

- Escolher o timbre a ser tocado a partir da lista de timbres (figura 13);
- Cadastrar o aluno através do menu Arquivo;
- Alterar os Graus de Dificuldade. Este menu possui 4 submenus denominados “Etapa I”, “Etapa II”, “Etapa III”, “Etapa IV”, onde cada um desses submenus possui mais 10

subitens denominados “módulos” (figura 14), utilizados para geração dos intervalos e execução dos exercícios;

- Iniciar Exercício: após a seleção do módulo escolhido, o software toca o intervalo escolhido;
- Iniciar Exemplo: quando pressionado este botão, o software toca os intervalos referentes ao módulo escolhido, em todas as suas formas diminuto, justo, aumentado ou maior e menor.
- Desempenho: mostra o desempenho atual do aluno referente aos exercícios.

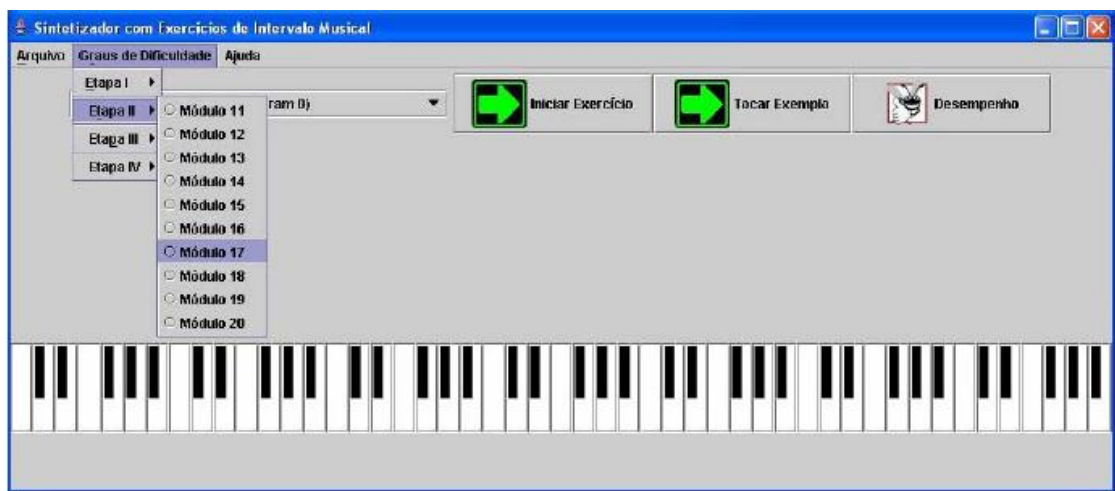


FIGURA 12 – Graus de dificuldade do software.

(Fonte: Próprio Autor, 2012)

9.2 Módulo exemplo

Todos os exemplos (elaboração algorítmica dos mesmos) são praticamente iguais variando apenas a nota base (variável min). Isto vale para intervalos divididos da mesma forma, quartas e quintas de uma forma, segunda, terça, sexta e sétima de outro e oitava isolada. Sendo assim desenvolveu-se um algoritmo que receba como parâmetro a nota base do intervalo e gere os exemplos de todos os intervalos com classificação semelhante. O quadro 7 mostra, através do algoritmo, a execução apenas de intervalos de quintas, para intervalos de quartas basta mudar o parâmetro (min) para um inteiro igual a 4, já para outros intervalos como de segundas, terças, etc, deverá ser criada outra estrutura do código fonte, tendo em vista que a classificação destes intervalos é diferente. A nota mínima (variável min) define o intervalo a ser tocado, por exemplo, 4 define que é de quarta e 6 de quinta etc (quantidade de

semitons entre as notas), sendo fixo, pois a definição está no momento da chamada do método. O que varia é apenas a variável (i) para contar quantos exemplos foram feitos e serve para definir se o intervalo será justo, aumentado ou diminuto.

Algoritmo Módulo Exemplo.

```

public void modulo1Exemp()
{
    tocarExemp(6,24);
}
public void tocarExemp(int min, int oitava)
{
    int x = 0;
    for (int i = 0; i <= 2; i++)
    {
        if (i != 2)
        {
            x = i + 1;
        }
        else
        {
            x = 0;
        }
        for (int j = 0; j <= 2; j++)
        {
            pri = 0 + (int) (Math.random() * oitava) + 48;
            seg = pri + min + x;
            tocarNota (pri, 1000);
            tocarNota (seg, 2000);
        }
    }
    for (int i = 0; i <= 2; i++)
    {
        pri = 0 + (int) (Math.random() * oitava) + 48;
        seg = pri + min + i;
        tocarNota (pri, 1000);
        tocarNota (seg, 2000);
    }
}

```

Sendo assim a variável (seg) receberá a primeira nota (pri) mais a variável (min) que é fixa mais a variável (x) que definirá o tipo de intervalo. O argumento que é recebido (variável oitava) servirá para definir a quantidade de oitavas que pode ser usada no sorteio, por exemplo, 24 define que o número sorteado estará em apenas duas oitavas, 48 quatro oitavas e assim sucessivamente. O método (tocarNota) recebe como argumento a primeira ou segunda nota e sua duração em milissegundos. Sendo assim este exemplo tocará 12 intervalos: três justos, três aumentados, três diminutos e depois um de cada respectivamente.

9.3 Módulo exercício

O processo de elaboração do algoritmo módulo exercício é semelhante ao algoritmo módulo exemplo, variando também apenas a nota base (variável min). O quadro 8 mostra o algoritmo do módulo exercício. Entretanto, este algoritmo serve apenas para os intervalos de quartas e quintas com suas variações (justas, aumentadas ou diminutas). Na linha em destaque do quadro 8 é calculada a segunda nota a ser tocada, definindo o tipo do intervalo a ser gerado.

Algoritmo Módulo Exercício

```
public void modulo1Exercicio()
{ tocarExercicio(6,24); }
public void tocarExercicio(int min, int oitava)
{
pri = 0 + (int) (Math.random() * oitava) + 48;
seg = pri + min + 0 + (int) (Math.random() * 3);
tocarNota (pri, 1000);
tocarNota (seg, 2000);
PianoBigasII.teclas_branças_e_pretas[pri -
24].setBackground(Color.blue);
}
```

A diferença é que este método toca o intervalo apenas uma única vez sorteando a nota em uma das duas oitavas predefinidas. Como ajuda ao aluno, o sistema “pinta” a primeira nota tocada, exigindo que o aluno pressione a segunda nota corretamente através de sua percepção.

9.4 Verificação do desempenho

Na figura 13 é apresentada a tela desempenho

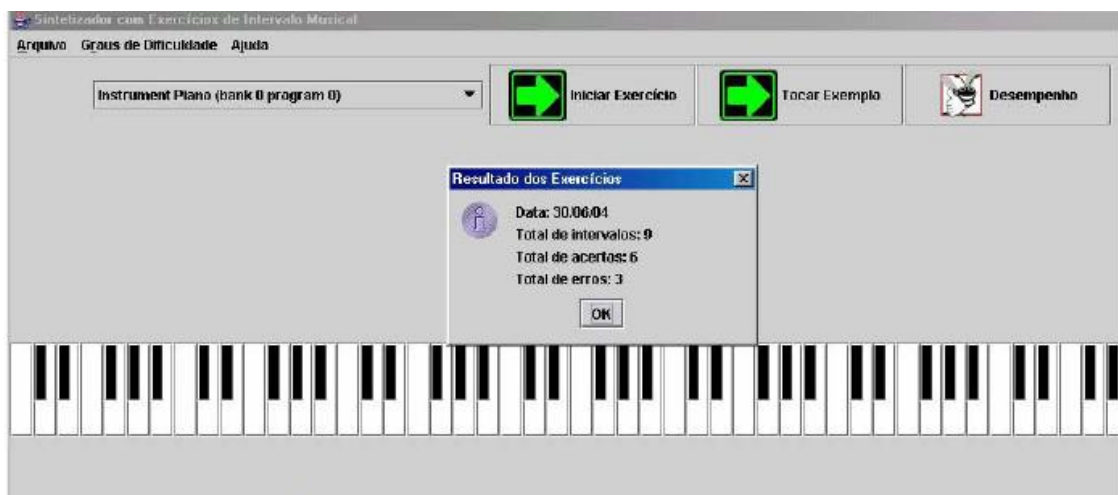


FIGURA 13 - Tela desempenho

(Fonte: Próprio Autor, 2012)

Logo após a execução de um ou vários intervalos, o aluno tem a opção de clicar no botão “Desempenho” para mostrar sua avaliação atual. No campo “Data” o programa mostra a data atual em que o aluno executou os exercícios, o campo “Total de intervalos” mostra a quantidade de intervalos executados até aquele presente momento, já o campo “Total de acertos” indica quantos acertos o aluno adquiriu também até aquele presente momento, e por fim, o campo “Total de erros” indica quantos erros o aluno cometeu.

9.5 Conclusão

Neste capítulo foram apresentadas algumas das interfaces do software com suas repetitivas funções. Foram também apresentados através de algoritmos o funcionamento dos exercícios referentes aos módulos e de como o usuário deve utilizar o software para execução dos exercícios. Além disso, foi apresentado o processo como aluno pode verificar o seu desempenho.

CONSIDERAÇÕES FINAIS

A ideia de projetar e desenvolver o sistema com exercícios de intervalos musicais surgiu, inicialmente, do desejo de utilizar a informática como uma ferramenta de auxílio a educação. Após algumas pesquisas, percebeu-se que a ideia seria bem vinda no ensino, principalmente na área de música. A partir disto, procurou-se fazer visitas a escolas de músicas da região, apresentando a ideia inicial do projeto e coletando dados com profissionais da área sobre como é a estrutura do ensino da música. Essas visitas foram de grande importância para a criação de um esboço do projeto e auxiliaram na elaboração de novas idéias para o sistema. Com base nas entrevistas realizadas conseguiu-se desenvolver um software com exercícios de intervalo musical, definindo assim a sequência correta dos intervalos, partindo dos mais fáceis para os mais difíceis. Logo após, foram implementados os exercícios de percepção musical, cujo objetivo seria treinar a audição do aluno com exercícios repetitivos. Entretanto, havia necessidade de obter materiais teóricos mais específicos e consistentes para a criação e desenvolvimento do software com exercícios de intervalos Musicais. Estes foram obtidos através de leituras de textos sobre o uso do computador na educação e o ensino da música. Enfrentou-se uma grande dificuldade para encontrar bibliografias relacionadas na área da música, principalmente, sobre ensino de percepção musical. Percebeu-se que a percepção ainda é tratada como um “dom” e não uma habilidade a ser desenvolvida. Esta falta de material com indicações sobre a melhor sequência para o estudo de intervalos musicais e conseqüente desenvolvimento da percepção melódica dos

alunos ressaltam a importância do sistema, que traz este conhecimento embutido nos graus de dificuldades dos seus módulos que é aumentado gradativamente, de acordo com os resultados do aluno. Houve algumas dificuldades na implementação tratando-se da avaliação do aluno. Não foi possível criar um mecanismo de aproveitamento indicando se o aluno estaria apto a seguir para o módulo seguinte (novos intervalos) ou não, sendo que a única informação dada é o número de acertos e erros do mesmo, ficando assim sob sua responsabilidade a definição do módulo a ser estudado. Como sugestões de pesquisas e atividades futuras pretende-se dar continuidade ao software incluindo outras técnicas de teoria musical, como percepção harmônica e rítmica. Além disso, há a possibilidade de melhorias no sistema já existente, como transformar o aplicativo em um applet para deixá-lo disponível na internet para qualquer usuário usufruir o mesmo. Ainda assim poderá ser melhorado o processo de avaliação do software, podendo identificar o tipo de intervalo com maior dificuldade e direcionando mais os exercícios para aquele intervalo.

REFERÊNCIAS

- 1 BOOCH, G.; RUMBAUGH, J.; JACOBSON, J. **UML, Guia do Usuário**. Trad. Fábio Freitas da Silva. Rio de Janeiro: Campus, 2000. 472 p.
- 2 CARDOSO, B.; MASCARENHAS, M. **Curso Completo de Teoria e Solfejo**. 15. ed. São Paulo: Irmãos Vitale, 1996. 192 p. v. 1.
- 3 DEITEL, H. M. **Java Como Programar**. Trad. Carlos Arthur Lang Lisbôa. 4. ed. Porto Alegre: Bookman, 2003. 1383 p.
- 4 FONTES, M. **ABC da Música**. Trad. Mariana Czertok. 2. ed. São Paulo: Irmãos Vitale, 1998. 284 p.
- 5 IZZO, M. **Noções Elementares de Música**. 4. ed. São Paulo: Irmãos Vitale, 1946. 225.
- 6 MASLAKOWSKI, M. **Aprenda em 21 dias MySQL**. Trad. Edson Furmankiewics, Joana Figueiredo. Rio de Janeiro: Campos, 2000. 458 p.
- 7 PAULA FILHO, W. P. **Multimídia Conceitos e Aplicações**. Rio de Janeiro: Livros Técnicos e Científicos Editora, 2000. 321 p.
- 8 PEDRON, C. **Método Completo de Divisão Musical**. São Paulo: Musicalia S/A. Cultura Musical, 1976. 76 p.
- 9 RATHBONE, A. **Multimídia & CD-ROMs para leigos**. Trad. Pedro Cesar de Conti. São Paulo: Berkeley Brasil, 1996. 368 p.

10 THOMPSON, M. **Java 2 && banco de dados**. São Paulo: Érica, 2002. 198 p.

Sites Consultados:

ANGÉLICO, H. Apontamentos de Teoria Musical 2. 2002. Apresenta textos sobre Teoria Musical.

Disponível em:

<<http://www.cl-ultramarino-n-srapaz.rcts.pt/material/edmusical/teoria02.htm#>>. Acesso em: 11 abr. 2011.

GOHN, D. As Novas Tecnologias e a Educação Musical. 2000. Apresenta textos sobre Tecnologia e Educação Musical.

Disponível em: <<http://cdchaves.sites.uol.com.br/educamusical.htm#>>.

Acesso em: 12 marc. 2012.