

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**PROTÓTIPO DE UM SISTEMA PARA AUXILIAR NA
COMPOSIÇÃO MUSICAL UTILIZANDO REGRAS DE
HARMONIA**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

ROBERTO CARLOS MASKE

BLUMENAU, DEZEMBRO/2000

2000/2-46

PROTÓTIPO DE UM SISTEMA PARA AUXILIAR NA COMPOSIÇÃO MUSICAL UTILIZANDO REGRAS DE HARMONIA

ROBERTO CARLOS MASKE

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Roberto Heinzle — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Roberto Heinzle

Prof. Dalton Solano dos Reis

Prof. Carlos Eduardo N. Bizzotto

*Este trabalho de conclusão de curso
é dedicado à minha esposa **Heidy**,
pelo amor, carinho, apoio e incentivos
recebidos ao longo destes anos de graduação.*

AGRADECIMENTOS

Ao professor e Orientador Roberto Heinzle pelo acompanhamento e incentivo na realização do trabalho.

A todos os amigos, professores e colegas do curso de Ciências da Computação pelo incentivo, ajuda, apoio e compreensão recebidos durante os anos de graduação.

SUMÁRIO

AGRADECIMENTOS	IV
SUMÁRIO	V
LISTA DE FIGURAS.....	VIII
RESUMO.....	X
ABSTRACT	XI
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS	2
1.2 ORGANIZAÇÃO DO TRABALHO	3
2 MÚSICA	4
2.1 ORIGENS DA MÚSICA	4
2.2 TEORIA ELEMENTAR DA MÚSICA	5
2.3 HARMONIA	13
2.3.1 DEFINIÇÃO	13
2.3.2 ACORDES.....	14
3 MIDI	17
3.1 CONCEITOS	17
3.2 PROTOCOLO	20
3.2.1 MENSAGENS.....	20
3.2.2 RUNNING STATUS	22
3.3 FORMATO.....	25
3.4 EVENTOS.....	27
4 SISTEMAS ESPECIALISTAS	30
4.1 CONCEITOS	30
4.2 COMPONENTES DE UM SISTEMA ESPECIALISTA	31

4.2.1	BASE DE CONHECIMENTOS	32
4.2.2	MECANISMO DE APRENDIZAGEM E AQUISIÇÃO DO CONHECIMENTO	33
4.2.3	MOTOR OU MÁQUINA DE INFERÊNCIA.....	33
4.2.4	SISTEMA DE CONSULTA.....	34
4.2.5	SISTEMA DE JUSTIFICAÇÃO	35
4.2.6	QUADRO NEGRO	35
4.3	REPRESENTAÇÃO DO CONHECIMENTO	36
4.4	A SHELL – EXPERT SINTA.....	37
4.4.1	ARQUITETURA DE UM SISTEMA ESPECIALISTA NO EXPERT SINTA	37
4.4.2	UTILIZANDO REGRAS DE PRODUÇÃO NO EXPERT SINTA SHELL	38
4.4.3	O MÉTODO DE EXTRAÇÃO DE CONHECIMENTO DO EXPERT SINTA SHELL.....	40
4.4.4	VARIÁVEIS UNIVALORADAS E VARIÁVEIS MULTIVALORADAS	41
4.4.5	FATORES DE CONFIANÇA.....	41
4.4.6	O CÁLCULO DE PROBABILIDADE NO EXPERT SINTA	42
4.4.7	GERENCIANDO BASES DE DADOS.....	44
4.5	FORMALIZAÇÃO DO CONHECIMENTO	44
4.6	EXPERT SINTA VISUAL COMPONENT LIBRARY (VCL)	44
4.6.1	COMPONENTES DO VCL.....	45
4.6.2	RELAÇÃO ENTRE OS COMPONENTES	47
5	O PROTÓTIPO DESENVOLVIDO.....	49
5.1	INTRODUÇÃO	49
5.2	ESPECIFICAÇÃO	49
5.2.1	ATRIBUTOS E VALORES	50
5.2.2	CLÁUSULAS E PREDICADOS.....	51
5.2.3	REGRAS	51
5.2.4	FATOR DE CERTEZA.....	52
5.3	MODELAGEM ESSENCIAL	52

5.4 PLATAFORMA DE DESENVOLVIMENTO	53
5.5 OPERACIONALIDADE DO PROTÓTIPO	53
6 CONCLUSÕES E SUGESTÕES.....	64
6.1 CONCLUSÕES	64
6.2 LIMITAÇÕES.....	64
6.3 SUGESTÕES PARA TRABALHOS FUTUROS	65
ANEXO 1	66
REFERÊNCIAS BIBLIOGRÁFICAS	112

LISTA DE FIGURAS

FIGURA 1 – PAUTA MUSICAL.....	5
FIGURA 2 – LINHAS E ESPAÇOS SUPLEMENTARES	6
FIGURA 3 – CLAVE DE SOL E DE FÁ.	7
FIGURA 4 – CLAVES MAIS USADAS.....	7
FIGURA 5 – DIVISÃO PROPORCIONAL DOS VALORES.....	8
FIGURA 6 – PARTES DE UMA NOTA.....	8
FIGURA 7 – PONTO DE AUMENTO	9
FIGURA 8 – TOM E SEMITOM	9
FIGURA 9 – FIGURAS SIMPLES E COMPOSTAS.....	11
FIGURA 10 – TONALIDADES COM BEMÓIS.....	12
FIGURA 11 – TONALIDADES COM SUSTENIDOS.	12
FIGURA 12 – NOTAÇÃO.	13
FIGURA 13 – TABELA DE INTERVALOS.....	14
FIGURA 14 – TIPOS DE TRÍADE.	15
FIGURA 15 – TRÍADE EXTENDIDA.....	15
FIGURA 16 – TRÍADE COM NOTA ADICIONADA.	16
FIGURA 17 – TRANSMISSÃO DE BYTES.....	17
FIGURA 18 – MENSAGENS “NOTE ON” E “NOTE OFF”.	18
FIGURA 19 – FORMATO DA MENSAGEM MIDI.	19
FIGURA 20 – TRÊS BYTES DE STATUS.....	23
FIGURA 21 – UM BYTE DE STATUS.	23
FIGURA 22 – TOCANDO E ABAFANDO UMA NOTA.	24
FIGURA 23 – QUADRO DE QUANTIDADE DE TEMPO TRADUZIDOS PARA VALORES DE 32 BITS	28

FIGURA 24 - COMPONENTES BÁSICOS DE UM SISTEMA ESPECIALISTA.....	32
FIGURA 25 – ARQUITETURA DE UM SISTEMA ESPECIALISTA.....	38
FIGURA 26 – REGRA DE PRODUÇÃO.	44
FIGURA 27 – BARRA DE FERRAMENTAS NO DELPHI COM A BIBLIOTECA (<i>EXPERT SINTA VCL</i>).....	45
FIGURA 28 – RELACIONAMENTO ENTRE OS COMPONENTES DO EXPERT SINTA VCL.	48
FIGURA 29 – DIAGRAMA DE CONTEXTO	52
FIGURA 30 – TELA DE ABERTURA	54
FIGURA 31 – OPÇÃO PARA ABRIR O ARQUIVO MIDI.....	55
FIGURA 32 – ABRINDO O ARQUIVO MIDI	55
FIGURA 33 – EXEMPLO DE UMA PARTITURA CRIADA NO ENCORE 4.21	56
FIGURA 34 – BASE DE CONHECIMENTO GERADA NO <i>EXPERT SINTA SHELL</i>	57
FIGURA 35 – EXEMPLO DE CONSULTA NO <i>EXPERT SINTA SHELL</i>	57
FIGURA 36 – OPÇÃO DE ESCOLHA DO ACORDE	58
FIGURA 37 – SOLUÇÃO DA CONSULTA.....	58
FIGURA 38 – HISTÓRICO DA SOLUÇÃO.....	59
FIGURA 39 – TODOS OS VALORES DA REFERIDA CONSULTA	59
FIGURA 40 – A SOLUÇÃO NO PROTÓTIPO.....	60
FIGURA 41 – OPÇÃO PARA SALVAR O ARQUIVO MIDI APÓS A SOLUÇÃO	60
FIGURA 42 – OPÇÃO PARA EXECUTAR O ORIGINAL.....	61
FIGURA 43 – TOCANDO O ARQUIVO ORIGINAL.....	61
FIGURA 44 – OPÇÃO PARA EXECUTAR O ARQUIVO MIDI APÓS A SOLUÇÃO	62
FIGURA 45 – EXECUTANDO ARQUIVO MIDI APÓS A SOLUÇÃO.....	62
FIGURA 46 – RESULTADO NO ENCORE 4.21 APÓS A SOLUÇÃO	63

RESUMO

A finalidade deste trabalho de conclusão de curso é apresentar um estudo sobre sistema especialista e considerações sobre a interface MIDI – *Musical Instrument Digital Interface*, com o objetivo de desenvolver um protótipo de software para complementar uma segunda voz, utilizando regras de harmonia musical. As ferramentas utilizadas para a construção do sistema serão a *Shell “Expert SINTA”*, que utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidades, e a linguagem de programação *Delphi 3.0*.

ABSTRACT

The objective of this final paper is to present a study about specialist system and MIDI - Musical Instrument Digital Interface. It includes the development of a software prototype to perform a second music voice, using musical harmony rules.

The Expert Sinta Shell is a tool used in this work in order to manipulate the production rules set of the knowledge base. The programming language used is Delphi 3.0.

1 INTRODUÇÃO

Devido ao grande avanço tecnológico, a maneira de se fazer música mudou. Por essa razão, a composição musical também deve assimilar a essas mudanças. A busca de métodos não tradicionais, enriquecidos de criatividade, abrem novas perspectivas no campo da música ([FRI1994]). Tais métodos combinados aos métodos tradicionais, quando apropriados, podem levar a um resultado ainda melhor no processo de composição, principalmente com a utilização do computador. Hoje em dia, não basta apenas tocar certo, limitando-se aos padrões tradicionais. É preciso escolher o estilo e a forma, sem o que, a música perde a sua principal finalidade, que é fazer refletir a personalidade do artista.

Conforme [PRI1994], música é a arte dos sons, combinados de acordo com as variações da altura, proporcionados segundo a sua duração e ordenados sob as leis da estética. São três os elementos fundamentais de que se compõe a música: melodia, ritmo e harmonia. A melodia consiste na sucessão dos sons formando sentido musical. O ritmo é o movimento dos sons regulados pela sua maior ou menor duração e a harmonia consiste na execução de vários sons ouvidos ao mesmo tempo, observados as leis que regem os agrupamentos dos sons simultâneos, formando acordes. O conceito de acorde diz que três ou mais notas que soam simultaneamente formam um acorde ([ARC1986]).

Os sons musicais são representados graficamente por sinais chamados notas, e a escrita da música dá-se o nome de notação musical. As notas são sete: dó-ré-mi-fá-sol-lá-si. Estas notas são escritas na pauta musical ou pentagrama, que é a reunião de cinco linhas horizontais, paralelas e equidistantes, formando entre si quatro espaços.

Dada uma partitura musical com apenas uma voz melódica, o protótipo se encarrega de gerar uma segunda voz, de acordo com regras de harmonia, facilitando assim, o trabalho de composição e harmonização de qualquer peça musical. Um exemplo, é tendo a nota sol em uma partitura, com o acorde de dó maior, esta pode ser acrescida por uma segunda voz usando a nota mi ou dó, pois estas notas fazem parte do acorde de dó maior.

No protótipo foi utilizado o padrão MIDI, que segundo [FER1995], é a comunicação entre um computador e um periférico musical. No *stream* de dados MIDI, descreve-se cada um dos eventos musicais desejados (volumes, notas, tempos e outros), necessários para

execução da música em questão; não se guarda a forma de onda sonora em si, mas a seqüência de “comandos” musicais necessários para reproduzi-la com certo grau de acurácia. Trata-se de uma espécie de partitura musical eletrônica que, passada a um sintetizador MIDI, será transformada na melodia em questão.

O protótipo implementado é um sistema especialista que, de acordo com [HEI1995], “são sistemas computacionais projetados e desenvolvidos para solucionarem problemas que normalmente exigem especialistas humanos com conhecimento na área de domínio da aplicação”. Este conhecimento será armazenado em um banco de conhecimentos através de fatos e regras.

Propõe-se a especificação e implementação de um Sistema Especialista que auxiliará o músico para adição de uma segunda voz musical. As ferramentas utilizadas para a construção do sistema serão a *Shell “Expert SINTA”*, que utiliza um modelo de representação do conhecimento em regras de produção e probabilidades, e a linguagem de programação Delphi 3.0.

O método de especificação utilizado no presente trabalho será expresso pela notação BNF (*Backus-Naur Form*).

1.1 OBJETIVOS

O trabalho proposto tem como objetivo a especificação e implementação de um protótipo de um sistema especialista que auxiliará e facilitará o trabalho de músicos para adicionar uma segunda voz musical. As principais características do protótipo são:

- a) músico entra apenas com a primeira voz musical e o acorde de sua preferência, gerado num editor musical;
- b) rapidez na inclusão da segunda voz musical;
- c) possibilidade de ouvir a música após o término da inclusão;
- d) facilidade de inclusão de novas regras na base de conhecimento;
- e) não é necessário o músico ter muito conhecimento de harmonia musical.

1.2 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado em seis capítulos, descrevendo:

- Capítulo 1 – Introdução ao trabalho, com breve descrição do contexto desde, seus objetivos e sua organização;
- Capítulo 2 – Fundamentação teórica relativa a Música, abrangendo desde conceituação, harmonia;
- Capítulo 3 – Fundamentação teórica sobre MIDI (*Musical Instrument Digital Interface*), compreendendo conceituação, formato, especificação;
- Capítulo 4 – Fundamentação teórica relativa a Sistemas Especialistas, abrangendo desde conceituação, ferramenta *Expert SINTA Shell* até formalização do conhecimento;
- Capítulo 5 – Apresentação do protótipo desenvolvido;
- Capítulo 6 – Conclusões e sugestões para futuros trabalhos.

2 MÚSICA

Segundo [JAC1956], a música é uma arte. Isto quer dizer que em sua criação intervém, de modo principal, a intuição e a sensibilidade, ajustadas, como em toda obra humana, pela inteligência e a vontade. A sua importância para a sociedade é um fato consumado. São três os elementos fundamentais de que se compõe a música:

- Melodia: consiste na sucessão dos sons formando sentido musical.
- Ritmo: é o movimento dos sons regulados pela sua maior ou menor duração.
- Harmonia: consiste na execução de vários sons ouvidos ao mesmo tempo, observadas as leis que regem os agrupamentos dos sons simultâneos.

Com a união destes 3 elementos, pode-se obter uma combinação infinita de resultados, os quais se pode sentir em qualquer momento ou lugar. Desde o hino do país até cerimônias religiosos, percebe-se a importância que esta forma de expressão deu ao mundo.

2.1 ORIGENS DA MÚSICA

Para [LAR1977], procurar as origens da música parece temerário. O canto precedeu a palavra? A palavra, ao contrário, induziu os homens a cantar? No homem primitivo a música foi uma espécie de expansão fisiológica. Com o desenvolvimento de sua inteligência e o aperfeiçoamento de sua sensibilidade, foi tornando-se uma arte. Segundo [JAC1956], a música seguiu dois caminhos, um dos povos orientais, e outro, o que havia de levar ao desenvolvimento portentoso da música atual, caminho esse seguido preferencialmente pelo povos orientais.

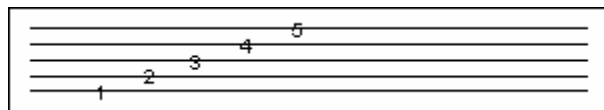
Ainda seguindo [JAC1956], o conhecimento completo e consciente do sistema musical não chegou a ser obtido até uma época muito avançada na Idade Média. Foi nessa ocasião que o homem ocidental se deu conta de um elemento maravilhoso da arte da música, que até então não havia descoberto: a harmonia, o prazer da consonância. Até aquele momento não

havia percebido que sons distintos podiam ser executados de uma só vez e dar lugar a uma música nova e inteligente, ou melhor, à verdadeira música. Num tempo em que já todas as artes haviam chegado ao cume da perfeição.

2.2 TEORIA ELEMENTAR DA MÚSICA

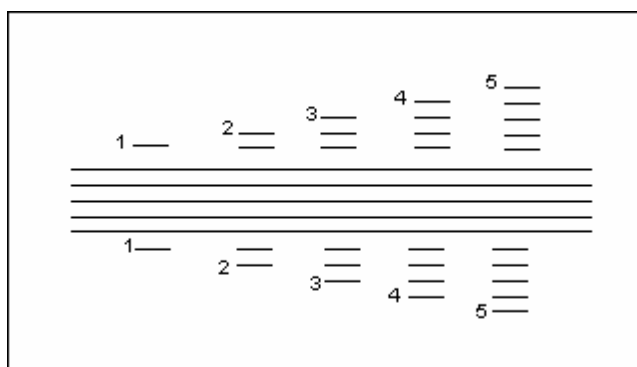
De acordo com [ARC1986], a música se escreve em pautado de 5 linhas formando entre si 4 espaços que se denomina pauta. A pauta é também chamada de pentagrama. São utilizadas as linhas e os espaços. É nas linhas e nos espaços da pauta que se escrevem as notas. As linhas, bem como os espaços da pauta, são contadas de baixo para cima (figura 1).

FIGURA 1 – PAUTA MUSICAL



A pauta, entretanto, não é suficiente para conter todos os sons musicais que o ouvido pode apreciar. Por esse motivo, usam-se linhas chamadas suplementares superiores ou suplementares inferiores, quando são colocadas, respectivamente, acima ou abaixo da pauta. Usa-se também escrever notas nos espaços formados por essas linhas (espaços suplementares superiores ou inferiores). As linhas e espaços suplementares contam-se de baixo para cima quando superiores e de cima para baixo quando inferiores. O número de linhas ou espaços suplementares não é limitado, contudo, não é comum empregar-se mais de 5 (figura 2).

FIGURA 2 – LINHAS E ESPAÇOS SUPLEMENTARES



Fonte: [CAM2000]

Para representar a altura e a duração dos sons, são usados sinais de forma oval, que, pelas posições tomadas na pauta, indicam sons mais graves ou mais agudos, designando-se o tempo de prolongação pela variedade de forma. Designando a altura dos sons, os distinguimos com os monossílabos: do, ré, mi, fá, sol, lá, si, repetidos em séries contínuas do grave ao agudo. Essas 7 notas ouvidas sucessivamente formam uma série de sons à qual se dá o nome de escala. Quando essa série de sons segue sua ordem natural (dó-ré-mi-fá-sol-lá-si) temos uma escala ascendente; seguindo em ordem inversa (si-lá-sol-fá-mi-ré-dó) temos uma escala descendente. A escala estará completa se for terminada a série ascendente ou iniciada a descendente com a nota dó. Chama-se de **cifra** a representação das notas musicais através de letras. A=lá, B=si, C=dó, D=ré, E=mi, F=fá, G=sol.

Conforme [PRI1994], para determinar o nome da nota e a sua altura na escala coloca-se no princípio da pauta um sinal chamado clave. Cada clave dá seu nome à nota escrita em sua linha. Nos espaços e nas linhas subsequentes, ascendentes ou descendentes, as notas vão sendo nomeadas sucessivamente, de acordo com a ordem já referida (figura 3). Há três sinais de claves: clave de sol, clave de fá e clave de dó (figura 4).

FIGURA 3 – CLAVE DE SOL E DE FÁ.

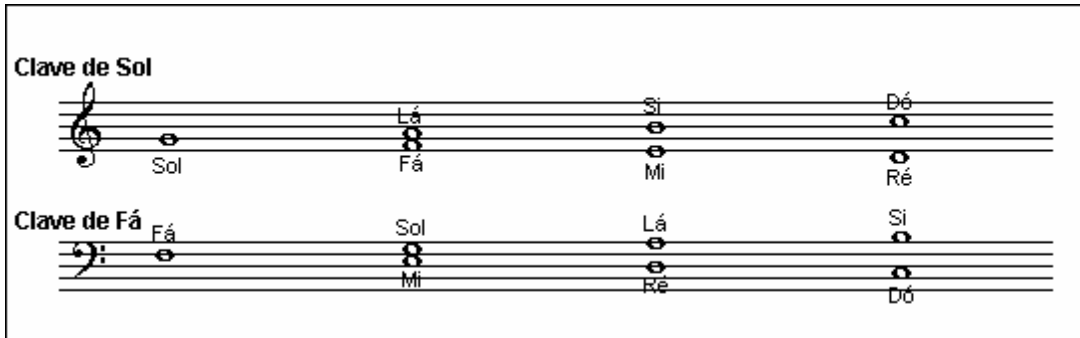




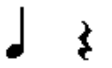




FIGURA 4 – CLAVES MAIS USADAS.



Fonte: [CAM2000]

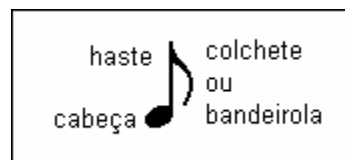
Nem todas as notas tem a mesma duração. Para representar as várias durações dos sons musicais as notas são escritas sob formas diferentes. Essas diversas formas das notas são chamadas figuras ou valores. Essas figuras representam os sons os quais são chamadas valores ou ainda, figuras de som. Pausas são figuras que indicam duração de silêncio entre sons. Cada figura de som tem sua respectiva pausa que lhe corresponde ao tempo de duração. Também são sete os valores mais usados na escrita musical moderna, com o qual indicamos a duração dos sons (figura 5).

FIGURA 5 – DIVISÃO PROPORCIONAL DOS VALORES

Figura	Nota e pausa
Semibreve	
Mínima	
Semínima	
Colcheia	
Semicolcheia	
Fusa	
Semifusa	

Uma nota é dividida em: cabeça, haste e colchete (figura 6).

FIGURA 6 – PARTES DE UMA NOTA

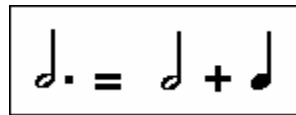


Fonte: [CAM2000]

Quando se escrevem duas ou mais colcheias, semicolcheias, fusas ou semifusas consecutivas, usa-se também substituir os colchetes por barras horizontais, ficando as notas unidas em grupos.

Pode-se acrescentar um ponto numa nota que prolongará a duração na metade de seu valor. Inclusive pode haver pontos duplos ou triplos. Cada um deles aumenta a duração na metade do valor anterior, razão pela qual, qualquer nota pode receber um número infinito de pontos (contudo, nunca chegará a duplicar o valor da nota original). É por isso chamado ponto de aumento (figura 7).

FIGURA 7 – PONTO DE AUMENTO



Fonte: [CAM2000]

De acordo com [OLI2000], o intervalo é a distância de freqüência sonora que existe entre duas notas. O menor intervalo possível entre duas notas é de meio tom (um semitom). Por exemplo: o intervalo entre as notas C e D é de 1 tom, ou 2 semitons (figura 8).

FIGURA 8 – TOM E SEMITOM

do#	re#		mi#	fa#	sol#	la#		si#
reb	mib	fab		solb	lab	sib	dob	
DO	RE	MI	FA	SOL	LA	SI	DO	

Fonte: [CAM2000]

Como vimos, o intervalo entre C e D é de 1 tom, e o menor intervalo possível entre duas notas é de meio tom. Logo, entre C e D existe uma terceira nota. Esta nota pode ser

chamada de C# (dó **sustenido**) ou de Db (ré **bemol**). Estas notas que ficam entre as notas naturais são chamadas de acidentadas. C# (dó sustenido) é a nota dó elevada em meio tom e Db é a nota ré baixada em meio tom, logo, são a mesma nota (o que chamamos "enarmonia", que é um mesmo som com nomes diferentes). Quando temos uma nota acidentada e queremos que ele volte a ser uma nota natural, usa-se o **bequadro** que anula o efeito de qualquer acidente precedente.

De acordo com [OLI2000], **escala cromática** é aquela formada por todas as 7 notas naturais e mais os acidentes que existem entre essas notas. São 12 notas no total:

- Ascendente: C, C#, D, D#, E, F, F#, G, G#, A, A#, B;
- Descendente: B, Bb, A, Ab, G, Gb, F, E, Eb, D, Db, C.

Quando a escala é ascendente usam-se sustenidos e quando descendente, bemóis. Observe que não existe E# nem B#. Por isso é dito que o intervalo entre E e F é de meio tom, assim como entre B e C.

A **escala diatônica** possui apenas 7 notas, e não 12 como no caso da cromática. As pessoas às vezes pensam que a escala diatônica é formada somente pelas notas naturais (as teclas brancas de um piano ou teclado).

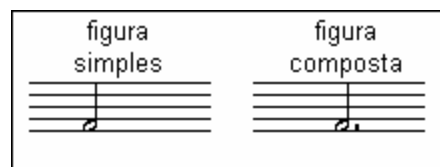
Os tempos são agrupados em porções iguais, de dois em dois, de três em três ou de quatro em quatro, constituindo unidades métricas as quais se dá o nome de compasso.

- Os compassos de 2 tempos são chamados **binários**.
- Os compassos de 3 tempos são chamados de **ternários**.
- Os compassos de 4 tempos são chamados de **quaternários**.

Cada grupo de tempos, isto é, cada compasso, é separado do seguinte por uma linha vertical travessão. Na terminação de um trecho musical usa-se colocar dois travessões denominados de travessão duplo ou pausa final (se a terminação for absoluta, isto é na finalização do trecho). Em qualquer compasso, a figura que preenche um tempo chama-se **unidade de tempo** e a figura que preenche um compasso chama-se **unidade de compasso**. Os compassos se dividem em duas categorias: simples e compostos. São representados por uma fração ordinária colocada no princípio da pauta, depois da clave.

Compassos Simples: são aqueles cuja unidade de tempo é representada por uma figura divisível por 2. Tais figuras são chamadas simples, isto é, são figuras não pontuadas (figura 9). Vejamos por exemplo: um compasso qualquer (binário, ternário, ou quaternário) no qual a unidade de tempo seja a semínima ou a colcheia. A semínima vale 2 colcheias, e a colcheia vale 2 semicolcheias, logo, ambas são divisíveis por 2; por conseguinte como unidade de tempo serão compassos simples. Analisemos os termos das frações que representam os compassos simples. O **numerador** determina o número de tempos do compasso. Os algarismos que servem para numerador dos compassos simples são: 2 (para binário), 3 (para o ternário) e 4 (para o quaternário). O **denominador** indica a figura que representa a unidade de tempo.

FIGURA 9 – FIGURAS SIMPLES E COMPOSTAS.



Fonte: [CAM2000]

O sistema tonal é formado por 2 modos: o modo maior e o modo menor. O modo de uma escala é distinguido pela colocação dos semitons.

- No modo maior, os semitons acham-se do terceiro para o quarto e do sétimo para o oitavo grau.
- No modo menor há três tipos de escalas. A escala natural ou forma antiga, os semitons se situam entre o segundo e o terceiro e entre o quinto e o sexto grau.

De acordo com [ARC1986], tonalidade é o conjunto de sons constituintes da escala em relação com a sua tônica. Na figura 10 temos todas as tonalidades formadas com bemóis.

FIGURA 10 – TONALIDADES COM BEMÓIS.

Nº de acidentes	Acidentes	Tonalidade maior	Tonalidade menor	Cifra maior	Cifra menor
0	0	Dó	lá	C M	A m
1	Si	Fá	ré	F M	D m
2	Si-mi	Sib	sol	Bb M	G m
3	Si-mi-lá	Mib	dó	Eb M	C m
4	Si-mi-lá-ré	Láb	fá	Ab M	F m
5	Si-mi-lá-ré-sol	Réb	Sib	Db M	Bb m
6	Si-mi-lá-ré-sol-dó	Solb	mib	Gb M	Eb m
7	Si-mi-lá-ré-sol-dó-fá	Dób	láb	Cb M	Ab m

Na figura 11 temos todas as tonalidades formadas com sustenidos.

FIGURA 11 – TONALIDADES COM SUSTENIDOS.

Nº de acidentes	Acidentes	Tonalidade maior	Tonalidade menor	Cifra maior	Cifra menor
0	0	Dó	Lá	C M	A m
1	Fá	Sol	Mi	G M	E m
2	Fá-dó	Ré	Si	D M	B m
3	Fá-dó-sol	Lá	Fá#	A M	F# m
4	Fá-dó-sol-ré	Mi	Dó#	E M	C# m
5	Fá-dó-sol-ré-lá	Si	Sol#	B M	G# m
6	Fá-dó-sol-ré-lá-mi	Fá#	Ré#	F# M	D# m
7	Fá-dó-sol-ré-lá-mi-si	Dó#	Lá#	C# M	A# m

Indica-se por meio da armadura de compasso, que mostra quantos tempos estão agrupados. Aparece no pentagrama inicial junto à armadura de clave. A armadura de compasso também indica um sistema de acentuação: geralmente o primeiro tempo de cada compasso será o mais forte. Além disso, há outros símbolos que indicam outros aspectos da música. A notação completa de uma peça musical se intitula partitura (figura 12).

FIGURA 12 – NOTAÇÃO.

O diagrama mostra uma partitura musical com duas staves (treble e bass clef) em 3/4 compasso. O tempo é marcado como *Allegro molto e vivace*. A dinâmica é indicada por *fff* e *con forza*. Outros símbolos incluem *Sustenido* e *Pausas*. As legendas apontam para: Pentagrama, Compasso, Indicador de tempo, Claves, Armadura de clave, Notas, Indicadores de dinâmica, Sustenido e Pausas.

Fonte: [REG2000]

2.3 HARMONIA

2.3.1 DEFINIÇÃO

Segundo [MAT2000], **harmonia** é a relação vertical das notas que são executadas num mesmo momento. A harmonia pode ser ternária (sons formados pelo intervalo de terças, ex. Dó/Mi/Sol ou Dó/Mib/Sol), quaternária (formada por intervalos de quarta, ex. Fá/Si/Mi ou Fá/Sib/Mi), quinária (intervalo de quinta, inversão do de quarta, ex. Si/Fá/Dó), intervalo de segunda (ex. Dó/Ré/Mi) e assim por diante. É irrelevante se estes intervalos são maiores ou menores ou mesmo aumentados. É básico para o estudo da Harmonia e para a composição,

que se tenha em mente a seguinte tabela de intervalos, que são os graus dos tons com relação a uma nota fundamental, neste exemplo (Dó) (figura 13).

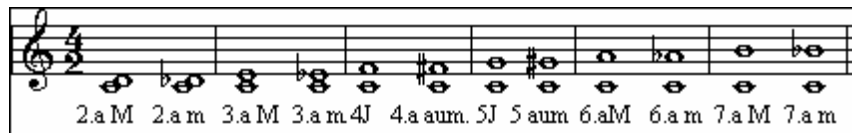


FIGURA 13 – TABELA DE INTERVALOS

Fonte: [MAT2000]

Neste exemplo, Ré# ou Mib, forma com respeito à Dó, um intervalo de terça menor (3m).

2.3.2 ACORDES

Acordes são 3 ou mais notas tocadas ao mesmo tempo. Os acordes mais simples são formados por apenas 3 notas (tríades).

- O acorde "Maior" é formado pelas notas dos graus 1, 3 e 5. Assim, o acorde de C maior (C) é formado pelas notas C, E e G.
- O acorde "Menor" é semelhante ao Maior, porém a nota do grau 3 é reduzida em meio tom. Assim, o acorde de C menor (Cm) é: C, Eb e G.
- O acorde "Diminuto" é semelhante ao Menor, porém a nota do grau 5 também é reduzida em meio tom. Assim, o acorde C Diminuto (Cdim) é: C, Eb e Gb.
- O acorde "Aumentado" é semelhante ao Maior, porém a nota do grau 5 é aumentada em meio tom. Assim, o acorde C Aumentado (C+) é: C, E e G#.
- O acorde com 7ª (também chamado "Dominante") é feito adicionando-se a nota do grau 7b ao acorde original. Assim o C com 7ª (C7) é: C, E, G e Bb. Este tipo de acorde é o mais encontrado no Blues, Rock, Country e Pop.

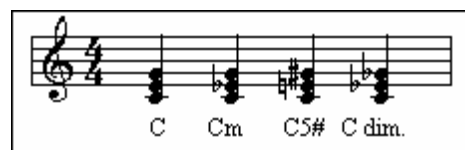
Arpejos são as notas de um acorde tocadas separadamente, em seqüência, ao invés de todas juntas. Existe um número muito limitado de acordes disponíveis na gaita diatônica, mas com o uso de *bends* e *overbends*, qualquer *arpeggio* pode ser tocado. Conhecer os arpejos é muito útil para a improvisação e para manter a improvisação consistente com os acordes do acompanhamento.

Os acordes mais utilizados em uma canção são os do grau 1, 4 e 5. Ou seja, se o tom da música é C, os acordes mais utilizados serão C (C,E,G), F (F,A,C) e G (G,B,D). Frequentemente os acordes são utilizados com acréscimo da 7ª. Ex: C (C,E,G,Bb).

A tríade pode assumir 4 formas distintas (figura 14):

- Maior,
- Menor,
- Diminuta,
- Alterada.

FIGURA 14 – TIPOS DE TRÍADE.

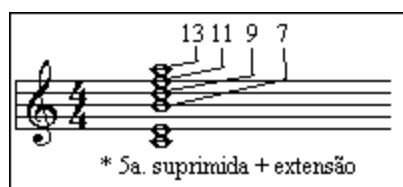


Fonte: [MAT2000]

Na harmonia a quatro vozes é habitual suprimir-se a 5ª. na tríade extendida. Neste tipo de acorde as notas principais são (figura 15):

- Fundamental - indica a tonalidade,
- Terça - Qualidade do acorde (maior, menor, etc.),
- Sétima - indicação da extensão,
- A dissonância característica.

FIGURA 15 – TRÍADE EXTENDIDA.

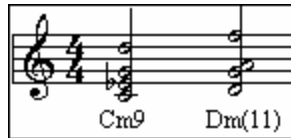


Fonte: [MAT2000]

A diferença entre a Tríade Extendida e a Tríade com Nota Adicionada é que a segunda sempre terá uma (figura 16):

- Fundamental,
- Terça,
- Quinta,
- A adição.

FIGURA 16 – TRÍADE COM NOTA ADICIONADA.



Fonte: [MAT2000]

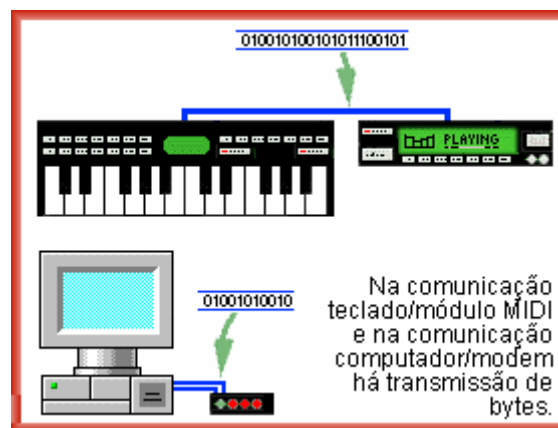
3 MIDI

3.1 CONCEITOS

MIDI é a sigla de *Musical Instrument Digital Interface*, que representa um padrão de comunicação de dados criado em 1983, fruto de um acordo entre diversos fabricantes de instrumentos musicais norte-americanos e japoneses, para possibilitar a transferência de informações entre instrumentos musicais e computadores ([RAT1995]).

Segundo [GON2000], MIDI é uma "linguagem eletrônica", composta de um grande conjunto de comandos musicais, com a qual instrumentos eletrônicos controlam uns aos outros. Da mesma forma que o *modem* passa bytes de dados entre o computador e o provedor Internet, usando um protocolo chamado TCP/IP, instrumentos eletrônicos passam bytes de dados usando o protocolo MIDI, conforme mostra a figura 17.

FIGURA 17 – TRANSMISSÃO DE BYTES



Fonte: [GON2000]

Este protocolo define várias séries diferentes de bytes. MIDI é, assim, um protocolo para transferência de informações (informações musicais em sua maioria). Estas informações tomam a forma de sinais eletrônicos que os instrumentos passam entre si.

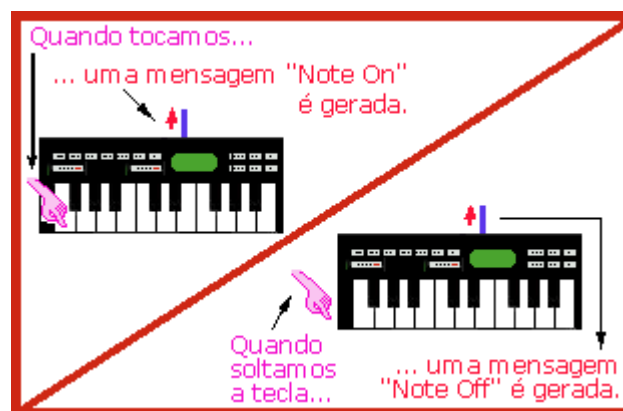
Há certas coisas que todo instrumento musical faz. Em qualquer hora, o músico pode fazer um instrumento começar a fazer um som. Por exemplo, um músico pode pressionar uma tecla de um piano para iniciar um som, ou beliscar uma corda de um violão etc. A ação de iniciar um som denomina-se "*Note On*".

Muitos instrumentos permitem que o músico "pare" o som em determinado tempo. Por exemplo, um músico pode soltar uma tecla de um piano, e interromper seu som. Esta ação de interromper o som denomina-se "*Note Off*".

As ações acima são ações musicais. E MIDI define, para cada ação musical (como as duas mencionadas acima), uma certa série de bytes chamadas de mensagens.

Assim, por exemplo, MIDI define como um instrumento deve iniciar um som ou seja, executar a ação "*Note On*" com uma mensagem chamada "*Note On*". Da mesma forma, MIDI diz ao instrumento para interromper o som usando uma mensagem chamada "*Note Off*" (figura 18).

FIGURA 18 – MENSAGENS “*NOTE ON*” E “*NOTE OFF*”.



Fonte: [GON2000]

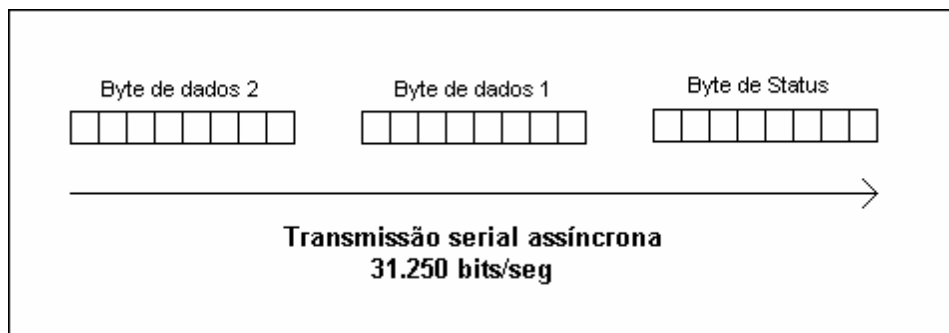
Mas MIDI é mais do que mensagens "*Note On*" e "*Note Off*". Há muitas mensagens mais, uma para cada ação musical.

Há uma mensagem que diz a um instrumento que mova sua Roda de Modulação ("*Pitch Wheel*") e quanto deve movê-la. Há outra mensagem que diz ao instrumento

pressionar ou soltar o pedal de sustentação; outra que diz ao instrumento que deve mudar de volume e o quanto fazê-lo; outra que diz para o instrumento mudar seu timbre (órgão para violão, por exemplo); etc. Pode-se dizer que MIDI pode fazer com um instrumento eletrônico tudo que um humano é capaz de fazer, e mesmo algumas coisas que humanos não podem.

Segundo [RAT1995], para transferir as informações através do cabo MIDI, o instrumento codifica essas informações sob a forma de números binários, que são enviados serialmente como mostra a figura 19. A transmissão ocorre a uma velocidade de 31.250 bits por segundo. Como a maioria das mensagens MIDI possui apenas um, dois, ou três bytes de tamanho, estas mensagens MIDI "voam" pelo cabo MIDI rapidamente fazendo com que o ouvido humano pense que os dois instrumentos estão realmente tocando em uníssono.

FIGURA 19 – FORMATO DA MENSAGEM MIDI.



Pode-se conectar um cabo MIDI na porta *MIDI OUT* (ou *MIDI THRU*) do segundo instrumento à uma porta *MIDI IN* de um terceiro, e então o segundo instrumento repassaria para o terceiro as mensagens que recebeu do primeiro. Assim, todos os três instrumentos podem tocar em uníssono. Poderia conectar um quarto, quinto, sexto, ..., instrumentos. Chama-se isso de ligar instrumentos "em série" (ou *daisy-chaining*).

Entretanto, instrumentos ligados em série não precisam, necessariamente, tocar em uníssono sempre. Cada um pode tocar sua própria parte musical mesmo que todas as mensagens MIDI que controlam todos os instrumentos passe por cada um deles.

Existem, em MIDI, 16 "canais". Todos eles existem em um só cabo MIDI, que liga dois ou mais instrumentos (e talvez um computador) em conjunto. Por exemplo, a mensagem MIDI para o Dó central poderia ser enviada no "canal 1", ou no "canal 2", etc.

A maioria dos instrumentos MIDI permite ao músico selecionar a que canal(is) o instrumento deve responder e qual(is) ignorar. Por exemplo, para configurar um instrumento para responder apenas às mensagens MIDI no canal 1, e enviar uma mensagem MIDI "*Note On*" pelo canal 2, então o instrumento não tocaria a nota. Assim, se um músico possui diversos instrumentos ligados em série (o *daisy-chaining* citado acima), ele poderia configurar todos os instrumentos para responder a diferentes canais cada um. Assim, o músico pode ter diversos instrumentos tocando em diversos canais, e pode controlar o que cada um deles vai tocar e com qual timbre sonoro irá tocar, a partir de um só instrumento.

Quando se quer gravar uma só nota, há só duas mensagens envolvidas: uma *Note On*, para iniciar o som da nota, de tamanho igual a 3 bytes, e uma *Note Off* (outros 3 bytes), para terminá-la após algum tempo. Esta segunda mensagem, *Note Off*, não é gerada até que se solte a tecla da nota. Isto soma 6 bytes. Na verdade, pode-se pressionar a nota por uma hora, que ainda teria apenas 6 bytes: uma mensagem *Note On* e outra *Note Off*. Pois na mensagem *Note On* existe: "a nota foi tocada no tempo x". Em *Note Off*, existe: "a nota foi solta no tempo y".

3.2 PROTOCOLO

3.2.1 MENSAGENS

O protocolo MIDI é composto de mensagens. Uma mensagem consiste em uma *string* (isto é, uma série) de bytes de 8 bits. MIDI tem muitas mensagens definidas. Algumas mensagens consistem em somente 1 byte. Outras mensagens tem 2 bytes. Outras tem 3 bytes. Um certo tipo de mensagem MIDI pode ter um número ilimitado de bytes.

Uma coisa que todas as mensagens têm em comum é que o primeiro byte da mensagem é o byte de *status*. Este é um byte "especial", porque é o único byte que tem o bit 7 setado.

Assim, sempre que for encontrado um byte destes, será detectado o começo de uma mensagem MIDI. Este será um byte de *status* entre 0x80 a 0xFF. Os bytes restantes da mensagem (isto é, os bytes de dados, se existirem) estarão na escala 0x00 a 0x7F (para indicar o uso do sistema numérico hexadecimal, de base 16, usou-se a convenção da linguagem C de iniciar um valor com o 0x).

Os bytes de *status* de 0x80 a 0xEF são para as mensagens que podem ser transmitidas em qualquer dos 16 canais MIDI. Por causa disto, estes são chamados "*Voice Messages*" (ou "Mensagens de Voz").

Para estes bytes de *status*, quebra-se o byte de 8 bits em 2 conjuntos de 4 bits. Por exemplo, um byte de *status* de valor 0x92 pode ser quebrado em 2 conjuntos, com valor de 9 (conjunto superior) e de 2 (conjunto inferior). O conjunto superior diz-lhe que tipo é a mensagem MIDI.

Estão aqui os valores possíveis para o conjunto superior, e a que tipo de *Voice Messages* cada um representa:

- 8 = Note Off,
- 9 = Note On,
- A = *AfterTouch* (também conhecida como "*Key Pressure*"),
- B = Control Change,
- C = Program Change,
- D = Channel Pressure,
- E = Pitch Wheel.

Assim, o exemplo de 0x92, mostra que seu tipo de mensagem é "*Note On*" (já que o conjunto superior é 9). E o que é o 2 do conjunto baixo? Isto significa que a mensagem está no canal MIDI 2. Há 16 canais (lógicos) possíveis em MIDI, sendo 0 o primeiro. Assim, esta mensagem é um "*Note On*" no canal 2.

O byte de *status* para especificar uma mudança do programa no canal 0, o conjunto superior necessitaria ser C, para um *Program Change*, e o conjunto baixo necessitaria ser 0 para o canal 0. Assim, o byte de *status* seria 0xC0. Embora o byte de *Status* MIDI contém os 16 canais em números 0 a F (15), todo equipamento MIDI (incluindo softwares de computador) indicam um número de canal ao músico como 1 a 16. Assim, um byte de *status* emitido no canal 0 MIDI é considerado estar no "canal 1" para o músico.

Portanto, bytes de *status* de 0x80 a 0xEF são para as mensagens de voz. Os bytes de *status* de 0xF0 a 0xFF são para as mensagens que não estão em nenhum canal particular. Estes bytes de *status* são usados para as mensagens que carregam uma informação que interessa a todos os dispositivos MIDI, tais como sincronização de todos os dispositivos de *playback* em determinado momento. Estes bytes de *status* são divididos em duas categorias. Os bytes de *status* de 0xF0 a 0xF7 são chamados de "Mensagens Comuns Ao Sistema" (ou "*System Common Messages*"). Os bytes de *status* de 0xF8 a 0xFF são chamados de "Mensagens de Tempo Real do Sistema" (ou "*Realtime System Messages*").

Determinados bytes de *status* dentro desta escala não são definidos pela especificação MIDI, e são reservadas para uso futuro. Por exemplo, os bytes de *status* de 0xF4, 0xF5, e 0xFD não são usados. Se um dispositivo de MIDI receber tal byte de *status*, deve ignorar essa mensagem.

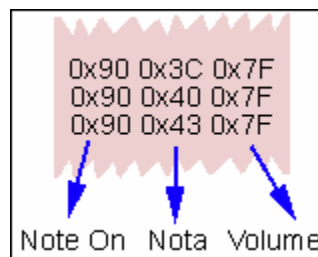
3.2.2 RUNNING STATUS

A especificação MIDI permite que uma mensagem MIDI seja emitida sem seu byte de *status* (ou seja, apenas seus bytes de dados serão transmitidos), desde que a mensagem anteriormente transmitida tenha o mesmo *status*. Isto é chamado de "*running status*". O *running status* é simplesmente um esquema inteligente para maximizar a eficiência da transmissão MIDI (através da remoção de bytes de *status* redundantes). A idéia básica do *running status* é que um dispositivo deve sempre se lembrar do último byte de *status* que recebeu (à exceção de *RealTime*), e, se não receber um byte de *status* em mensagens subsequentes, deve supor que está tratando de uma situação "*running status*". Um dispositivo

que gera mensagens MIDI deve sempre se lembrar do último byte de *status* que emitiu (à exceção de *RealTime*), e se precisar emitir outra mensagem que possua o mesmo *status*, o byte de *status* pode ser omitido.

Abaixo um exemplo de um dispositivo que cria um fluxo de mensagens MIDI; o dispositivo emite 3 mensagens *Note On* no canal 0 (figura 20).

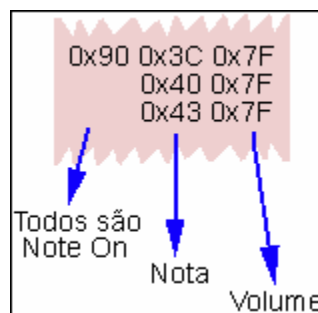
FIGURA 20 – TRÊS BYTES DE *STATUS*.



Fonte: [GON2000]

Observa-se que os bytes de *status* de todas as 3 mensagens são os mesmos no caso, *Note On*, canal 0. Conseqüentemente, o dispositivo podia executar um *running status* para as últimas 2 mensagens, emitindo os seguintes bytes (figura 21):

FIGURA 21 – UM BYTE DE *STATUS*.



Fonte: [GON2000]

Isto permite que o dispositivo poupe tempo, já que há 2 bytes a menos para transmitir. Claro, se a mensagem que o dispositivo enviou antes destas 3 também é um *Note On* no canal 0, então ele poderia ter omitido o *status* da primeira mensagem acima, também.

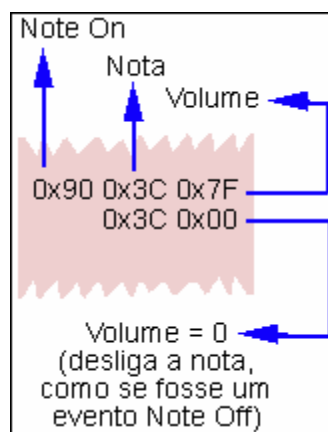
Ele recebe o *status* da primeira mensagem (0x90) e pensa: "aqui temos uma mensagem *Note On* no canal 0. Preciso me lembrar desse byte de *status*. Sei que há mais 2 bytes de dados vindo, vou esperá-los". E então ele recebe os 2 bytes de dados. Então, recebe o byte de dados da segunda mensagem (0x40).

E aqui o dispositivo pensa: "eu não estava esperando mais bytes de dados! Eu esperava por um novo byte de *status* de alguma outra mensagem. Bem, então esta deve ser uma mensagem usando *running status*. O último byte de *status* recebido era 0x90, então supõe-se que esta nova mensagem tenha este mesmo *status*. Portanto, este 0x40 é o primeiro byte de dados de uma outra mensagem *Note On* no canal 0".

Uma mensagem *Note On* com uma velocidade 0 é considerada como sendo uma *Note Off*. Com isto, pode-se emitir um fluxo inteiro de mensagens de nota desligando ou ligando notas, sem precisar enviar um novo byte de *status*, além do presente na primeira mensagem. Todas as mensagens terão o mesmo *status* de *Note On*, mas as mensagens que desligam notas terão a velocidade zero.

O exemplo abaixo mostra como tocar e abafar uma nota, utilizando este esquema de *running status* (figura 22).

FIGURA 22 – TOCANDO E ABAFANDO UMA NOTA.



Fonte: [GON2000]

As mensagens da categoria de *RealTime* (*status* de 0xF8 a 0xFF) não efetuam *running status* de forma nenhuma. Pois uma mensagem de *RealTime* consiste em somente 1 byte, e pode ser recebida a qualquer momento (inclusive dentro de outra mensagem), portanto deve ser tratada de forma "transparente".

As mensagens da categoria Comuns ao Sistema (de *status* 0xF0 a 0xF7) cancelam todo *running status*. Ou seja, a mensagem após uma Mensagem Comum Ao Sistema deve começar com um byte de *status*.

O *running status* é executado somente para mensagens da Categoria de Voz (cujo *status* é 0x80 a 0xEF).

3.3 FORMATO

O formato MIDI padrão (*Standard MIDI File*, ou SMF) é um formato de arquivos especificamente projetado para armazenar os dados que um seqüenciador (software ou hardware) grava e reproduz.

Este formato armazena as mensagens MIDI padrão (bytes de *status*, com os bytes de dados apropriados) mais um *timestamp* para cada mensagem (isto é, uma série de bytes que representam quantos pulsos de relógio aguardar antes de "reproduzir" o evento). O formato permite salvar informações sobre tempo, número de pulsos em resolução de semínima (ou resolução expressa em divisões de segundo, chamada configuração SMPTE), duração e tom da música, e nomes de trilhas e padrões. Pode armazenar múltiplos padrões e trilhas, de forma que qualquer aplicativo pode preservar estas estruturas quando carregar o arquivo.

O formato foi projetado para ser genérico, de forma que qualquer seqüenciador poderia ler ou escrever tal arquivo, sem perder os dados mais importantes, e flexível o suficiente para um determinado aplicativo armazenar seus dados próprios ("extras") de forma que outro aplicativo poderia carregá-lo sem problemas, e poderia ignorar com segurança esses dados extras que não precisasse. O formato de arquivos MIDI salvam dados em blocos (grupos de bytes precedidos por um identificador e tamanho) que podem ser analisados, carregados, saltados etc.

Dados são sempre salvados dentro de um bloco. Pode haver vários blocos dentro de um arquivo MIDI. Cada bloco pode ter um tamanho diferente (onde o tamanho refere-se a quantos bytes o bloco contém). Os bytes de dados em um bloco estão relacionados de certa forma. Um bloco é simplesmente um grupo de bytes relacionados.

Cada bloco começa com um identificador de 4 bytes ASCII, que diz que "tipo" de bloco é. Os próximos 4 bytes (assumindo que um byte é igual a 8 bits), indicam o tamanho em bytes do bloco (este valor é expresso em um número de 32 bits). Todos os blocos devem começar com estes dois campos (estes 8 bytes), os quais são referidos como cabeçalho do bloco.

O tamanho não inclui o cabeçalho de 8 bytes. Simplesmente diz quantos bytes de dados estão no bloco seguindo o cabeçalho. Exemplo de um cabeçalho (com bytes expressados em hexadecimal):

- 4D 54 68 64 00 00 00 06.

Os primeiros 4 bytes formam o identificador ASCII "MThd", isto é, os primeiros 4 bytes são valores ASCII para "M", "T", "h" e "d". Os próximos 4 bytes nos dizem que deve haver mais 6 bytes de dados no bloco (e após isso encontrados o próximo cabeçalho de bloco ou o fim do arquivo). Todo arquivo MIDI começa com esse cabeçalho MThd e é assim que se sabe que este arquivo é um arquivo MIDI.

Os primeiros dois bytes de dados (que seguem os 8 bytes do cabeçalho), nos dizem o formato ou tipo. Há hoje 3 tipos diferentes de arquivos MIDI.

O **tipo 0** significa que o arquivo contém uma só trilha contendo dados MIDI em talvez todos os 16 canais MIDI. Um arquivo de **tipo 1** significa que o arquivo contém uma ou mais trilhas simultâneas (isto é, todas começam de um tempo assumido como zero), talvez cada uma em uma em um canal MIDI. Juntas, todas estas trilhas são consideradas uma seqüência ou padrão. Um arquivo do **tipo 2** significa que o arquivo contém um ou mais padrões de uma só trilha, mas seqüencialmente independentes.

Os dois próximos bytes (chamados Número de Trilhas, ou "*NumTracks*") dizem quantas trilhas estão armazenadas no arquivo. Para arquivos de formato 0, ele é sempre igual a 1. Para os outros dois formatos, pode haver diversas trilhas. Os dois últimos bytes indicam em quantos Pulsos por Semínima (ou "*Pulses per Quarter Note*", abreviado como PPQN) a resolução das batidas de tempo são baseados, chamado de Divisão ("*Division*").

Aqui está um exemplo de um bloco MThd completo (com o cabeçalho):

- 4D 54 68 64 Identificador MThd,
- 00 00 00 06 O tamanho de um bloco MThd é sempre 6,
- 00 01 O Formato é 1,
- 00 02 Há 2 blocos MTrk neste arquivo,
- 00 F0 Cada incremento de tempo representa 1ms.

Após o bloco MThd, deve-se encontrar um Bloco MTrk, já que este é o único outro bloco definido atualmente (se encontrar outro identificador de bloco, pode ser proprietário de algum outro programa, então devemos saltá-lo ignorando o número de bytes de dados indicados no campo Tamanho).

Um bloco MTrk contém todos os dados MIDI (com bytes de tempo), mais alguns dados não-MIDI opcionais para uma trilha. Deve-se encontrar tantos blocos MTrk existentes no arquivo, quantos o byte *NumTracks* do bloco MThd indicar.

O cabeçalho MTrk inicia com o identificador MTrk, seguido pelo tamanho (número de bytes de dados a serem lidos para esta trilha). O tamanho será provavelmente diferente para cada trilha.

Nos arquivos de formato 0, alterações nos sinais de compasso e tempo são passados através de um MTrk. No formato 1, o primeiro MTrk deve consistir apenas de eventos sinais de compasso e tempo de forma que possa ser lido por um dispositivo capaz de gerar um "mapa de tempos". No formato 2, cada MTrk deve iniciar com ao menos um evento de tempo e compasso iniciais. Se não há eventos de sinal de tempo e compasso no arquivo MIDI, é assumido 120 BPM (batidas por minuto) e compasso 4/4.

3.4 EVENTOS

O primeiro evento na trilha pode ser soar uma nota Dó central. O segundo pode ser tocar o Mi acima do Dó central. Estes dois eventos podem acontecer ao mesmo tempo. O terceiro evento pode ser liberar a nota Dó central. Este evento pode ocorrer alguns compassos após os primeiros dois eventos (isto é, o dó central é mantido pressionado por alguns compassos). Cada evento possui um "tempo" para ocorrer, e os eventos são arranjados dentro de "blocos" de memória na ordem em que ocorrem.

Em um arquivo MIDI, o "tempo" de um evento precede os bytes de dados que criam o evento. Em outras palavras, os bytes que formam a "batida do tempo" vêm primeiro. Uma dada batida de tempo é referenciada pelo evento prévio. Por exemplo, se um evento ocorre 4 tempos após o início da reprodução, seu "tempo delta" é 04. Se o próximo evento ocorre simultaneamente com este primeiro evento, seu tempo é zero. Assim, um "tempo delta" é a duração (em tempos) entre um evento e o evento precedente.

Um tempo delta é armazenado como uma série de bytes que são chamados de quantidade de tamanho variável. Apenas os primeiros 7 bits de cada byte são significantes (alinhados à direita, como um byte ASCII). Então, com um tempo-delta de 32 bits, deve-se desempacotá-lo em uma série de bytes de 7 bits. Se terá um número variável de bytes dependendo do seu tempo-delta. Para indicar qual é o último byte da série, deixa o bit 7 "limpo". Em todos os bytes precedentes, seta-se o bit 7. Assim, se um tempo-delta está entre 0 e 127, ele pode ser representado por um byte. O maior tempo-delta é 0FFFFFFF, que se traduz em um tamanho variável de 4 bytes. Aqui estão alguns exemplos de tempo-delta como valores de 32 bits, e as quantidades de tempo variáveis para a qual eles são traduzidos (figura 23).

FIGURA 23 – QUADRO DE QUANTIDADE DE TEMPO TRADUZIDOS PARA VALORES DE 32 BITS

Número	Quantidade
00000000	00
00000040	40
0000007F	7F
00000080	81 00
00002000	C0 00
00003FFF	FF 7F
00004000	81 80 00
00100000	C0 80 00
001FFFFFFF	FF FF 7F
00200000	81 80 80 00
08000000	C0 80 80 00
0FFFFFFF	FF FF FF 7F

Fonte: [GON2000]

Os primeiros (1 a 4) byte(s) em um MTrk serão os primeiros eventos de tempo-delta de quantidade de tamanho variável. O próximo byte de dados é atualmente o primeiro byte do evento em si. O chamaremos de *Status* do evento. Para eventos MIDI, este será o byte de *Status* MIDI atual (ou o primeiro byte de dados MIDI em caso de "*running status*"). Por exemplo, se o byte vale 90 hexa, então este evento é um "*Note On*" no canal 0 MIDI. Se por exemplo, o byte vale 23 hexa, você precisa verificar o *status* do evento anterior ("*MIDI running status*"). Obviamente, o primeiro evento MIDI em MTrk deve ter um byte de *status*. Após um byte de *status* temos um ou 2 bytes de dados (dependendo do *status* - algumas mensagens MIDI possuem apenas um byte de dados subsequente). Após isso tem-se o tempo-delta do próximo evento (de quantidade variável) e o início do processo de leitura do próximo evento.

4 SISTEMAS ESPECIALISTAS

4.1 CONCEITOS

Um Sistema Especialista (SE) de acordo com [FEI1983], é um programa inteligente de computador que usa conhecimentos e procedimentos inferenciais, para resolver problemas que são bastante difíceis, de forma a requererem para sua solução, muita perícia humana. O conhecimento necessário para atuar a esse nível, mais os procedimentos inferenciais empregados, pode considerar-se um modelo da perícia aos melhores profissionais do ramo. O conhecimento de um sistema especialista consiste em fatos e heurísticas. Os fatos constituem um corpo de informação que é largamente compartilhado, publicamente disponível e geralmente aceito pelos especialistas em um campo.

Um Sistema Especialista é aquele que é projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano. É capaz de emitir uma decisão, apoiado em conhecimento justificado, a partir de uma base de informações, tal qual um especialista de determinada área do conhecimento humano ([FAV2000]). Conforme [RIB1987], um SE deve, além de inferir conclusões, ter capacidade de aprender novos conhecimentos e, desse modo, melhorar o seu desempenho de raciocínio e a qualidade de suas decisões. Para tomar uma decisão sobre um determinado assunto, um especialista o faz a partir de fatos que encontra e de hipóteses que formula, buscando em sua memória um conhecimento prévio armazenado durante anos, no período de sua formação e no decorrer de sua vida profissional, sobre esses fatos e hipóteses. E o faz de acordo com a sua experiência, isto é, com o seu conhecimento acumulado sobre o assunto e, com esses fatos e hipóteses, emite a decisão. Durante o processo de raciocínio, vai verificando qual a importância dos fatos que encontra comparando-os com as informações já contidas no seu conhecimento acumulado sobre esses fatos e hipóteses. Neste processo, vai formulando novas hipóteses e verificando novos fatos; e esses novos fatos vão influenciar no processo de raciocínio. Este raciocínio é sempre baseado no conhecimento prévio acumulado. Um especialista com esse processo de raciocínio pode não chegar a uma decisão se os fatos de que dispõe para aplicar o

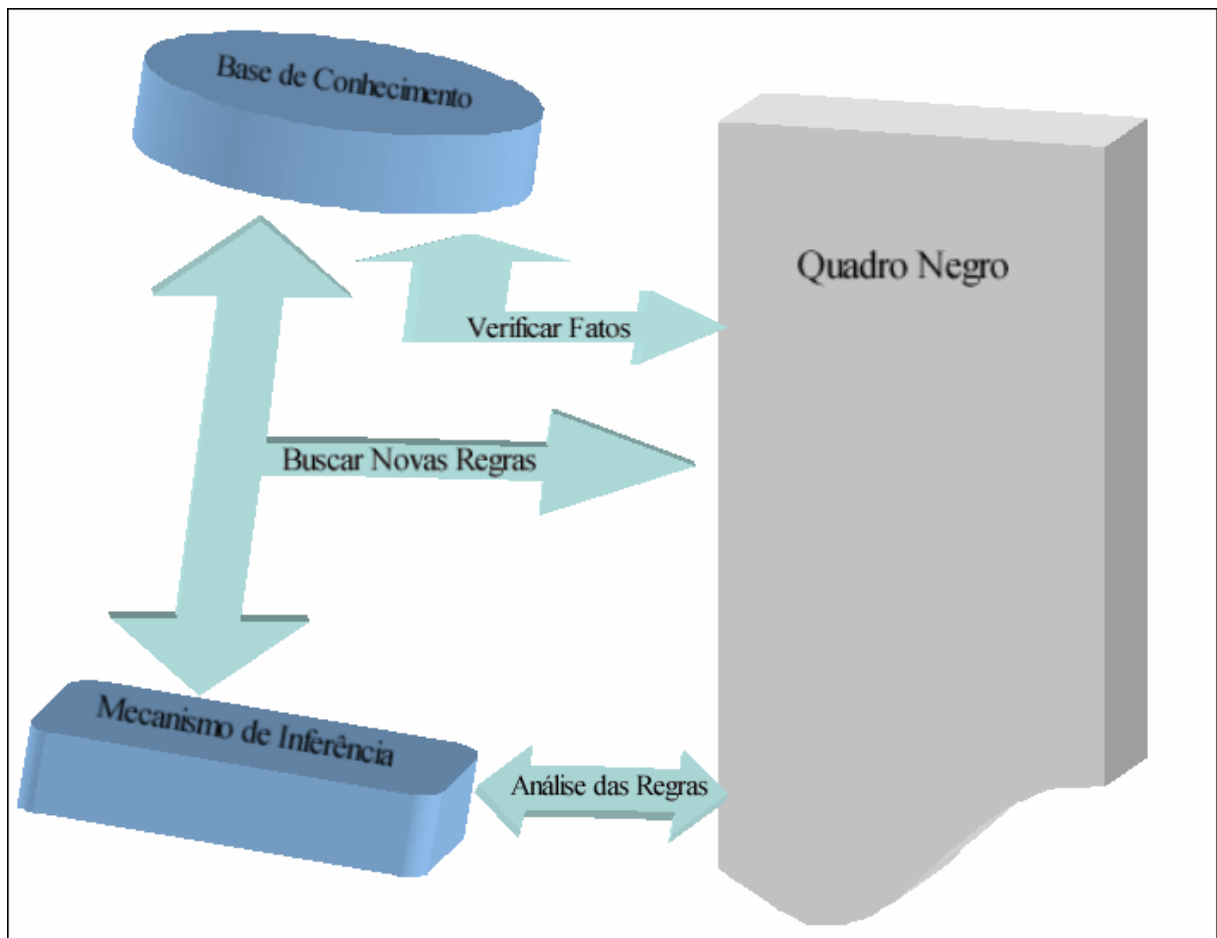
seu conhecimento prévio não forem suficientes. Pode, por este motivo, inclusive chegar a uma conclusão errada; mas este erro é justificado em função dos fatos que encontrou e do seu conhecimento acumulado previamente.

4.2 COMPONENTES DE UM SISTEMA ESPECIALISTA

Os sistemas especialistas empregam informações nem sempre completas manipulando-as através de métodos de raciocínio simbólico sem seguir modelos numéricos, para produzir aproximações satisfatórias ou aproximações úteis. Sendo assim, quanto mais completa e corretamente estiver representado o conhecimento, melhor será a saída do sistema. Para tanto faz necessário a aquisição de conhecimento, uso de heurísticas, de métodos de representação de conhecimento e de máquinas de inferência ([FAV2000]).

Segundo [RAB1995], “os componentes de um sistema especialista sofrem influências as mais variadas, desde a generalidade pretendida, os objetivos do mesmo, a representação do conhecimento e as ferramentas usadas na implementação”. Na figura 24 é mostrado o modelo geral da arquitetura de um sistema especialista apresentada por vários autores.

FIGURA 24 - COMPONENTES BÁSICOS DE UM SISTEMA ESPECIALISTA.



Fonte: [FAV2000].

4.2.1 BASE DE CONHECIMENTOS

A base de conhecimentos é o local onde se armazenam fatos e regras. Este conhecimento é passado ao sistema pelo especialista e armazenado de uma forma própria que permitirá ao sistema fazer posteriormente o processo de inferência. Um novo fato pode modificar todo o processo de inferência de acordo com as regras existentes sobre ele que estão sendo aplicadas e também sobre os novos fatos gerados pela avaliação dessas regras ([RIB1987]).

A qualidade do conhecimento armazenado é determinante no potencial do sistema especialista. A fase de construção da base de conhecimentos é uma das mais complexas na implementação do sistema especialista, pois o conhecimento de um especialista não se encontra formalizado, precisando de um trabalho prévio. A base de conhecimentos está interligada com quase todos os elementos do sistema, especialmente com a máquina de inferência, o mecanismo de aprendizagem e aquisição do conhecimento e o quadro negro ([HEI1995]).

4.2.2 MECANISMO DE APRENDIZAGEM E AQUISIÇÃO DO CONHECIMENTO

Segundo [RAB1995], “obter o conhecimento é a parte mais complexa da construção de um sistema especialista. A aquisição do conhecimento tende a caracterizar áreas de pesquisa específicas nas universidades e centros de pesquisas, geralmente ligadas a engenharia do conhecimento”.

Para [LEV1988], “a fase de aquisição do conhecimento é a que apresenta maior dificuldade na construção de um sistema especialista. Esta dificuldade advém do fato de não existir uma linguagem comum do entendimento entre as partes envolvidas no projeto”. O especialista não tem suas idéias organizadas utilizando processos indutivos e dedutivos na obtenção das soluções. Desta forma cabe ao engenheiro de conhecimento tentar organizar esses elementos e obter as informações necessárias.

4.2.3 MOTOR OU MÁQUINA DE INFERÊNCIA

Para [RAB1995], a máquina de inferência não é normalmente um único módulo de programa. É em geral, entendido como compreendendo o interpretador de regras e o escalonador das regras, quando o sistema especialista envolve regras de produção.

O processo de inferência está associado com a estrutura utilizada para o armazenamento do conhecimento na base de dados. De forma geral, pode-se afirmar que o processo envolve um encadeamento lógico que permite tirar conclusões a partir do conhecimento existente. Conforme [HEI1995], o motor de inferência é o responsável pela ação repetitiva de buscar, analisar e gerar novos conhecimentos.

[RIB1987], escreve que “o mecanismo de inferência depende de como se está representando o conhecimento. Nos sistemas de avaliação de regras, o mecanismo de inferência busca as regras na base do conhecimento e as avalia. Essa busca depende dos fatos e das hipóteses que existem e que se quer determinar a cada momento. Os objetivos a serem determinados pelo sistema de inferência devem ser relacionados com uma determinada ordem. A busca de regras é feita de maneira automática para que uma meta seja atingida.

Entretanto, existem casos em que a resposta pode ser obtida de maneira imediata e, nesses casos são estabelecidas estratégias de avaliação imediata, evitando todo o processo natural de busca e avaliação de regras. Outra estratégia usada consiste em o mecanismo de inferência proceder antes à busca das novas regras que foram causadas pela necessidade de se atender a uma meta, e avaliar essas regras a serem pesquisadas. Como os atributos são encontrados em diversas regras, o valor de uma cláusula já pode ter sido estabelecido. Esse valor, sozinho, permite determinar antecipadamente que a premissa da regra é falsa, e que não há razões para novas buscas. As estratégias de busca e avaliação de regras dependem do tipo de representação para o conhecimento e da arquitetura das próprias regras.

4.2.4 SISTEMA DE CONSULTA

O usuário é geralmente, alguém que não participou da elaboração do sistema, sendo, portanto, natural que não conheça as estruturas do sistema e, que, provavelmente, não esteja familiarizado com as forma de representação do conhecimento adotadas. Para que os potenciais usuários possam acessar com proveito e sem maiores dificuldades o sistema especialista, é preciso muni-lo de recursos para consulta.

Conforme [HEI1995], “a maioria dos sistemas existentes usam técnicas simples de interação com o usuário, quase sempre utilizando perguntas já pré-formatadas e respostas tipo múltipla escolha. Outra técnica é a definição de uma forma sintética simples com um vocabulário restrito e limitado, própria para utilização do sistema”.

4.2.5 SISTEMA DE JUSTIFICAÇÃO

Para [HEI1995], “o módulo de justificação é na verdade um recurso de questionamento fornecido ao usuário”. E portanto, o módulo de justificação é obrigatório nos sistemas especialistas, tendo, geralmente a capacidade de responder às seguintes perguntas:

- Como chegou a esta conclusão?
- Por que chegou a esta conclusão?
- Por que não chegou a outra conclusão?

[RIB1987] descreve que “este módulo interage com o usuário esclarecendo-os de como o sistema chegou a determinada conclusão ou por que está fazendo determinada pergunta. Utiliza diversos recursos estruturas próprias para atender ao seu objetivo, mostrando que regra e que fatos foram usados na base de conhecimento, sempre que isso for solicitado por quem usa o sistema”.

4.2.6 QUADRO NEGRO

Segundo [RAB1995], “o quadro-negro é a área de trabalho do sistema especialista. Armazena informações, fatos e estruturas de suporte ao funcionamento do sistema, quando este efetua raciocínios. Embora todos os sistemas especialistas usem o quadro-negro, nem todos o têm como um componente explícito”.

4.3 REPRESENTAÇÃO DO CONHECIMENTO

Para [HEI1995], “para um sistema especialista resolver problemas é imprescindível que esteja associado a ele um razoável volume de conhecimentos relativos ao domínio do problema. Este conhecimento deve ser transformado em estruturas de dados organizadas de modo a permitir a sua utilização pelo computador”.

A representação do conhecimento é a formalização do conhecimento do sistema. Para que isto seja possível existem técnicas que permitem modelar o conhecimento de forma eficiente, sendo as principais descritas a seguir:

a) **lógica das proposições e dos predicados:** na lógica das proposições, será atribuído o valor lógico verdadeiro se as informações disponíveis permitem tirar esta conclusão a respeito de uma proposição, caso contrário é atribuído o valor falso. Para se trabalhar com várias proposições, utilizações compostas e aumentar a capacidade de expressão. Estes operadores são: AND, NOT, OR, IMPLIES, EQUIVALENT ([HEI1995]);

b) **regras de produção:** sua estrutura constitui-se basicamente de uma premissa, ou conjunto de premissas, e uma conclusão, ou conjunto de conclusões. As regras são armazenadas como uma coleção de declarações SE-ENTÃO (SE <premissas> ENTÃO <conclusões>). Onde a parte condicional consiste de uma expressão proposicional ou simplesmente um termo;

c) **redes semânticas:** foram inicialmente desenvolvidas para modelagem psicológica da memória humana, constituindo-se agora num método de representação padrão. São estruturas formadas por nós, conectados, entre si através de arcos rotulados. Os nós representam objetos, conceitos, situações ou ações, e os arcos representam relações entre os nós;

d) **quadros ou frames:** esta forma de representação do conhecimento, organiza conhecimento de maneira a tornar evidente a compreensão de como a inferência pode ser feita. Um *frame* é constituído por um nome, uma coleção de atributos, chamados de escaninhos ou *slots*, e valores associados a eles.

4.4 A SHELL – EXPERT SINTA

O *Expert SINTA* é uma ferramenta computacional que utiliza técnicas de inteligência artificial para a geração automática de sistemas especialistas. Esta ferramenta utiliza um modelo de representação do conhecimento baseado em regras de produção e probabilidades, tendo como objetivo principal simplificar o trabalho de implementação de sistemas especialistas através do uso de uma máquina de inferência compartilhada, do tratamento probabilístico das regras de produção e da utilização de explicações sensíveis ao contexto da base de conhecimento modelada.

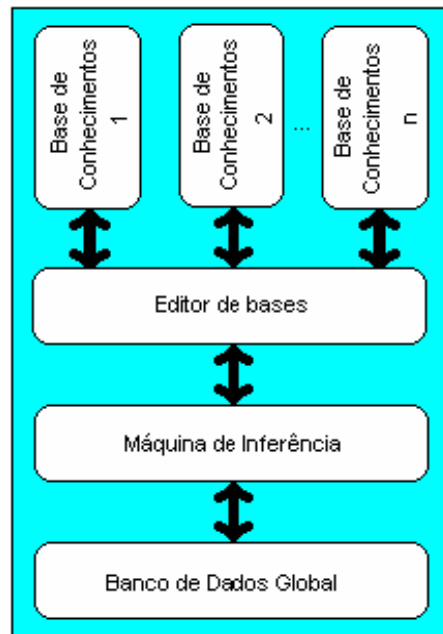
As principais características do *Expert SINTA*, conforme [LIA2000] são:

- a) utilização do encadeamento para trás (*backward chaining*);
- b) utilização de fatores de confiança;
- c) ferramentas de depuração;
- d) possibilidade de incluir ajudas *on-line* para cada base.

4.4.1 ARQUITETURA DE UM SISTEMA ESPECIALISTA NO EXPERT SINTA

Os sistemas especialistas que utilizam o *Expert SINTA* possuem a seguinte arquitetura, conforme mostra a figura 25:

FIGURA 25 – ARQUITETURA DE UM SISTEMA ESPECIALISTA



Fonte: [LIA2000]

Onde:

- a) **base de conhecimentos** representa a informação (fatos e regras) que um especialista utiliza;
- b) **editor de bases** é o meio pelo qual a *shell* permite a implementação das bases desejadas;
- c) **máquina de inferência** o motor de inferência é o responsável pela ação repetitiva de buscar, analisar e gerar novos conhecimentos;
- d) **banco de dados global** são as evidências apontadas pelo usuário do Sistema Especialista durante uma consulta.

O objetivo do *Expert SINTA* é simplificar ao máximo as etapas de criação de um Sistema Especialista completo. Para tanto, já oferece uma máquina de inferência básica, fundamentada no encadeamento para trás (*backward chaining*) ([LIA2000]).

4.4.2 UTILIZANDO REGRAS DE PRODUÇÃO NO EXPERT SINTA SHELL

A representação do conhecimento por regras de produção é baseada nas propostas do matemático *Emil Post* (1943) que via nos sistemas de produção um modelo computacional geral de produção é a forma mais utilizada em sistemas especialistas. A justificativa é a naturalidade que representa para o homem pois, o par condição-ação para raciocinar e decidir, também é usado pela mente humana ([HEI1995]).

Para [LIA2000], as regras de produção possuem as seguintes vantagens:

- Modularidade: cada regra, por si mesma, pode ser considerada uma peça de conhecimento independente;
- Facilidade de edição: novas regras podem ser acrescentadas e antigas ser modificadas com relativa independência;
- Transparência do sistema: garante maior legibilidade da base de conhecimentos.

Por exemplo:

SE tem combustível no tanque	(Premissas)
E tem combustível no carburador	(Premissas)
ENTÃO o motor recebe combustível	(conclusões da regra)

Para o projetista que cria bases utilizando o *Expert SINTA*, o seguinte critério para definições deve ser seguido.

A estrutura das premissas devem obedecer o seguinte modelo:

<atributo>	<operador>	<valor>	<conectivo>
------------	------------	---------	-------------

- Atributo: é uma variável capaz de assumir uma ou múltiplas instanciações no decorrer da consulta à base de conhecimento.

- Operador: é a ligação entre o atributo e o valor da premissa que define qual a comparação a ser realizada. São operadores tais como: =, >, <=, >=, <> entre outros.
- Valor: é um item de uma lista a qual foi previamente criada e relacionada a um atributo.
- Conectivo: **NÃO, E, OU**, sua função é unir a sentença ao conjunto das premissas que formam a seção de antecedentes de uma regra.

A estrutura da conclusão deve obedecer o seguinte modelo:

<atributo>	=	<valor>	<grau de confiança>
------------	---	---------	---------------------

- atributo: equivale ao atributo das premissas;
- “=” é um operador de **atribuição e não de igualdade**;
- valor: equivale ao valor utilizado nas premissas;
- grau de confiança: é um percentual indicando qual a confiabilidade da conclusão da regra. Varia de 0% a 100%.

4.4.3 O MÉTODO DE EXTRAÇÃO DE CONHECIMENTO DO EXPERT SINTA SHELL

O projetista deve incluir na definição da base quais os atributos que devem ser encontrados. A máquina de inferência encarrega-se de encontrar uma atribuição para o atributo desejado nas conclusões das regras. Para que a regra seja aprovada, suas premissas devem ser satisfeitas, obrigando à máquina a encontrar os atributos das premissas para que possam ser julgadas, acionando um encadeamento recursivo. Caso o atributo procurado não

seja encontrado em nenhuma conclusão de regra, uma pergunta direta é feita ao usuário ([LIA2000]).

4.4.4 VARIÁVEIS UNIVALORADAS E VARIÁVEIS MULTIVALORADAS

Quando a máquina de inferência está atrás de encontrar instanciações para uma variável univalorada, ela irá procurar até encontrar um valor ou então esgotar todas as possibilidades da base de conhecimento. Se durante a busca de outra variável, uma variável univalorada receber um valor quando já possuía outro, esse valor antigo será descartado, vigorando o outro valor. Uma única variável pode receber vários valores em uma única consulta ao sistema ([LIA2000]).

A busca de valores para variáveis multivaloradas prossegue até que toda a base de conhecimento seja explorada. Os valores permanecem acumulados. Nesta hora é preciso ter cuidado com as contradições presentes na base ([LIA2000]). As variáveis numéricas são tratadas como univaloradas.

4.4.5 FATORES DE CONFIANÇA

Sabe-se que o conhecimento humano não é determinístico. Não há especialista que esteja sempre em condições de afirmar determinada conclusão com certeza absoluta. Graus de confiança são geralmente atribuídos às respostas, principalmente quando existe mais de uma, sendo este um dos pontos mais críticos na elaboração de uma representação computacional do saber humano.

Observam-se dificuldades para representar a confiabilidade das informações:

- a) especialistas não se sentem confortáveis em pensar em termos de probabilidades. Suas estimativas não precisam corresponder àquelas definidas matematicamente;
- b) tratamentos rigorosamente matemáticos de probabilidade utilizam informações nem sempre disponíveis ou simplificações que não são claramente justificáveis em aplicações práticas.

4.4.6 O CÁLCULO DE PROBABILIDADE NO EXPERT SINTA

A atribuição dos fatores de confiança na presente versão do *Expert Sinta* é a seguinte ([LIA2000]):

- a) **quando deseja-se saber qual o valor final atribuído às variáveis na conclusão de uma regra:** deve-se seguir os seguintes passos: seja c_1 o grau de confiança atribuído ao resultado final da premissa de uma regra r . Na conclusão de r , deve-se ter expressões como $var = value \text{ CNF } c_2$, onde var é uma variável, $value$ é um termo qualquer que pode ser atribuído a uma variável, c_2 é um real pertencente entre zero e cem $[0,100]$ que representa o grau de confiança da atribuição. Mas c_2 é apenas uma referência, pois o valor final é dependente do resultado da premissa. Assim sendo, será realizado a operação $var = value \text{ CNF } c_1 * c_2$. Exemplo de uma aplicação para o cálculo de probabilidades: SE `fumagina = sim ENTÃO suspeita de praga = mosca branca`, grau de confiança (CNF) 70%. Assim sendo, supondo que o grau de confiança da igualdade `fumagina = sim` é 80%, teremos que a variável `suspeita de praga` será atribuído o valor `mosca branca`, com o respectivo grau de confiança $0.80 * 0.70 = 0.56 = 56\%$.
- b) **Cálculo do grau de confiança com o operador E:** se possuí-se duas igualdades $var_1 = value_1$ e $var_2 = value_2$, com os respectivos graus de confiança c_1 e c_2 , têm-se que a sentença $var_1 = value_1$ e $var_2 = value_2$ retornará como valor de confiança $c_1 * c_2$. Exemplo de aplicação: SE `estados das folhas = esfarelam facilmente E presença de manchas irregulares = sim...` Se o grau de confiança da igualdade `estados`

das folhas = esfarelam facilmente é 80% e o grau de confiança da igualdade presença de manchas irregulares = sim é 70%, temos que a conjunção das duas sentenças retornará um valor CNF de 56%, pois esse é o produto dos dois valores;

- c) **Cálculo do grau de confiança com o operador OU:** se possui-se duas igualdades $var1 = value1$ e $var2 = value2$, com os respectivos graus de confiança $c1$ e $c2$, têm-se que a sentença $var1 = value1$ OU $var2 = value2$ retornará com o valor de confiança $c1 + c2 - c1 * c2$. Exemplo de aplicação: SE besouros vermelhos = sim OU larvas marrons = sim... Se o grau de confiança da igualdade besouros vermelhos = sim é 80% e o grau de confiança da igualdade larvas marrons = sim é 70%, temos que a disjunção das duas sentenças retornará um valor CNF de $0.70 + 0.80 - 0.70 * 0.80 = 1.50 - 0.56 = 0.94 = 94\%$.
- d) **Quando uma variável recebe duas vezes o mesmo valor em pontos diferentes da consulta:** em momentos diferentes de uma consulta, uma mesma variável var pode receber o mesmo valor v , sendo que até à penúltima instanciação esta variável possuía grau de confiança $c1$, e a última atribuiu um CNF $c2$. Sendo assim, têm-se que o valor final de confiança para $var = v$ será dado através da fórmula $c_a + c_n - c_a * c_n$, onde c_a representa o grau de confiança antes da última mudança e o c_n representa o último grau de confiabilidade atribuído. Exemplo de aplicação: a variável doença possuía valor $mofo\ preto$ com grau de confiança 60%. Após a aplicação de outras regras chegou-se a uma outra atribuição $doença = mofo\ preto$, desta vez com CNF 50%. O cálculo se dá de maneira semelhante à aplicação da regra OU: doença terá como um dos valores $mofo\ preto$, com respectivo grau de confiança $0.60 + 0.50 - 0.60 * 0.50 = 1.10 - 0.30 = 0.80 = 80\%$.

O sistema admite 50% como valor mínimo de confiança para que uma igualdade seja considerada verdadeira, mas esse valor pode ser modificado. O intervalo de grau de confiança varia de 0 a 100.

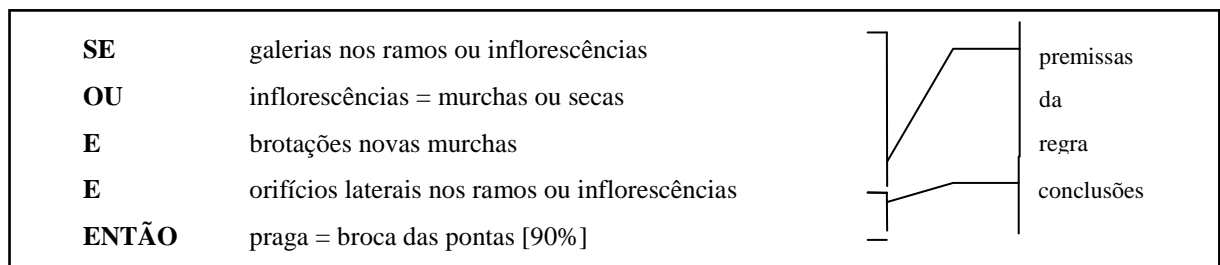
4.4.7 GERENCIANDO BASES DE DADOS

Como padrão, o *Expert SINTA* grava as bases de conhecimento geradas em arquivos com extensões *.BCM. Ao salvar a base pela primeira vez, uma caixa de diálogo aparecerá para que seja indicado o nome do arquivo na qual a base será gravada e qual pasta será encontrado este arquivo ([LIA2000]).

4.5 FORMALIZAÇÃO DO CONHECIMENTO

Conforme [LIA2000], o *Expert SINTA* utiliza regras de produção para a representação do conhecimento. Essas regras são um conjunto de condições no estilo SE...ENTÃO..., com a possibilidade de inclusão de conectivos lógicos. Relacionando, deste modo, os atributos no escopo do conhecimento e o uso de probabilidades, como se pode observar na figura 26.

FIGURA 26 – REGRA DE PRODUÇÃO.



Fonte: [LIA2000]

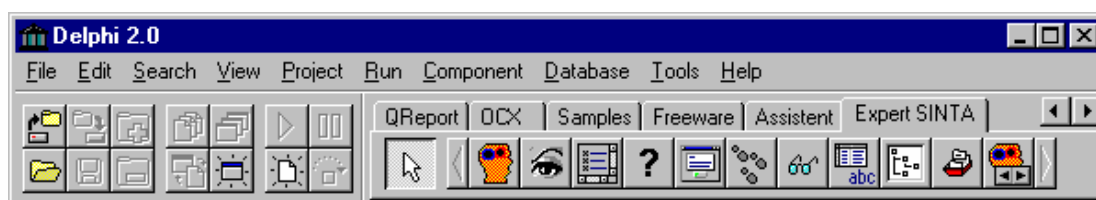
4.6 EXPERT SINTA VISUAL COMPONENT LIBRARY (VCL)

De acordo com [LIA2000], a *Expert SINTA VCL* é uma biblioteca de componentes para programação de Sistemas Especialistas. Na figura 27 temos a barra de ferramentas da *Expert SINTA VCL* para Delphi. De uma forma geral, esta biblioteca de componentes torna possível a criação de aplicações para bases de conhecimento geradas com o *Expert SINTA*.

Esta biblioteca de componentes vem sendo utilizada na construção de Sistemas Especialistas. Os principais motivos da utilização do *Expert SINTA VCL* são:

- a) o *Expert SINTA (shell)* não provém toda a funcionalidade necessária a certos Sistemas Especialistas;
- b) não há meios de aproveitar os dados obtidos com o *shell* em outros programas;
- c) é inviável o acréscimo de vários recursos de interface e intercâmbio de dados na ferramenta em si;
- d) os Sistemas Especialistas devem ser compilados em uma dada linguagem de programação e utilizados de forma totalmente independente do *Expert SINTA*;
- e) é possível reaproveitar milhares de linhas de código já escritas na construção do *shell*.

FIGURA 27 – BARRA DE FERRAMENTAS NO DELPHI COM A BIBLIOTECA (*EXPERT SINTA VCL*).



De uma forma geral, a *Expert SINTA VCL* torna possível a criação de *front-ends* para bases de conhecimento geradas com o *Expert SINTA*. Entre as tarefas desempenhadas por esta VCL, há:

- a) encapsulamento da máquina de inferência e a estrutura de dados que representa o conhecimento (regras de produção);
- b) fornecimento de mecanismos para entrada de dados do usuário;
- c) fornecimento de mecanismos de depuração;
- d) permitir a personalização da aplicação final.

4.6.1 COMPONENTES DO VCL

Os componentes nativos da *Expert SINTA VCL* podem ser divididos em categorias, conforme descrito por [LIA2000]. A seguir, são relacionadas estas categorias dos componentes:



TExpertSystem – encapsula a máquina de inferência e a estrutura de dados que representa a base de conhecimento.



TRuleView – exibe as regras de um base de conhecimento referenciada por um componente *TExpertSystem*.



TExpertPrompt – menu para entrada de dados do usuário em resposta a uma determinada pergunta efetuada pelo sistema.



TLabelQuestion – a única opção de personalização de interface integrada no *shell* Expert SINTA é a possibilidade de mudança da mensagem que aparece em uma pergunta para cada variável.



TValuesGrid – exibe as instâncias (valores) de uma dada variável por ordem decrescente de grau de confiança.



TWhyDialog – caixa de diálogo que exibe uma explicação para a necessidade de uma dada pergunta.



TDebugPanel – semelhante a *TRuleView*, exibe as regras da base de conhecimento de um sistema especialista em um painel, mas indica também qual premissa (ou conclusão) está sendo analisada pela máquina de inferência em determinado ponto de uma consulta.



TWatchPanel – de forma semelhante a opção *Watch* de um ambiente de programação, exibe as instâncias (valores atribuídos durante uma consulta) de todas variáveis através de dois painéis: o superior lista todas as variáveis; o inferior, as instâncias da variável selecionada no painel superior.



TConsultTree – este componente pode criar e exibir de forma hierárquica todos passos seguidos do começo ao fim de uma consulta.



TAllVars – ao contrário de *TwatchPanel*, este componente são se atualiza automaticamente para cada nova instância criada pela máquina de inferência.



TExNavigator - um navegador que controla o fluxo da consulta em conjunto com as respostas entradas pelo usuário e outros componentes de interface acrescentados pelo desenvolvedor da aplicação.

4.6.2 RELAÇÃO ENTRE OS COMPONENTES

De acordo com [LIA2000], existe outra forma de classificar a VCL: componentes da atualização automática, os quais modificam-se automaticamente sempre que um fato relevante ocorre durante uma consulta e componentes passivos, que precisam da chamada de um método para exibir funcionalidade.

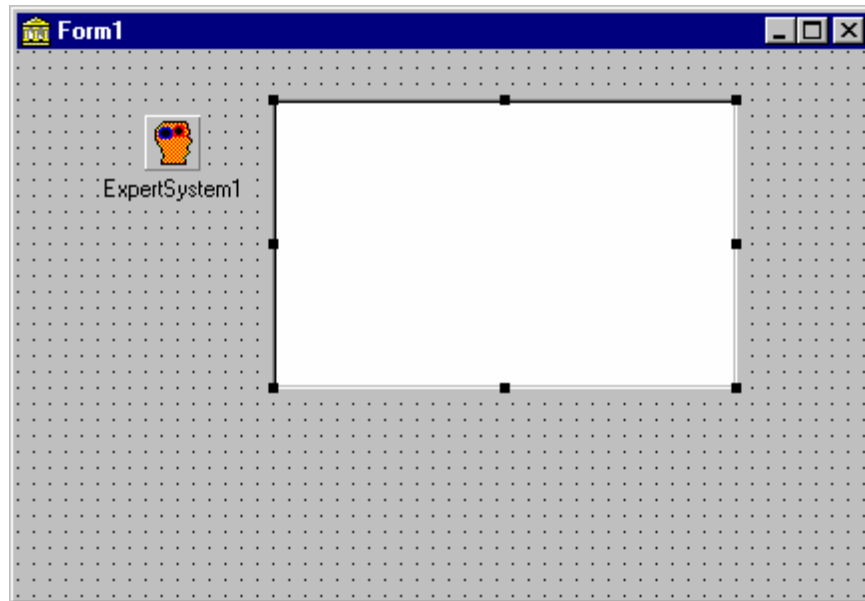
Basicamente, todos os componentes, à exceção de *TConsultiTree* e *TAllVars*, são automáticos. Para que componentes automáticos procedam como tal, é preciso relacioná-los a um componente *TExpertSystem*. Para isso, existe a propriedade *ExpertSystem*. Através do *Object Inspector* de um ambiente visual como o Delphi, pode ser atribuído um sistema especialista para cada controle automático (figura 28). Cada componente reage de acordo com a mudança feita. Um exemplo é o *TRuleView*, que reage a mudanças do arquivo da base de conhecimento, mas nenhum componente nativo da *Expert SINTA VCL* é notificado sobre mudanças realizadas diretamente na estrutura de dados, como, por exemplo, uma alteração de nome de variável feita através do *shell*.

Outro parâmetro que aparece constantemente nos componentes da *Expert Sinta VCL* são os códigos de variáveis. Por exemplo, o componente *TExpertPrompt* monta automaticamente um menu de entrada de dados para que o usuário marque valores de uma dada variável. Logo, a variável é um parâmetro básico deste componente. A variável referenciada pelo componente é indicada através de seu código.

Cada variável criada através do *Expert SINTA* recebe um código interno que nunca muda (a não ser, óbvio, que esta seja apagada e inserida novamente – categoricamente nem se quer seria a mesma variável), conforme mostrado no anexo. Assim, o uso de códigos é o modo mais estável de referência de variáveis.

Para obter os códigos criados pelo *Expert SINTA*, deve-se abrir a base de conhecimento no *Shell* e selecionar o menu Arquivo/Exportar/Códigos. A seguir, digita-se o nome do arquivo texto (.txt) onde sairão os resultados. Pode-se abrir posteriormente este arquivo em um editor de textos quando precisar saber o código de uma dada variável ou valor ([LIA2000]).

FIGURA 28 – RELACIONAMENTO ENTRE OS COMPONENTES DO EXPERT SINTA VCL.



Fonte: [LIA2000]

5 O PROTÓTIPO DESENVOLVIDO

5.1 INTRODUÇÃO

A proposta de construir um protótipo de um sistema especialista utilizando a ferramenta *Expert SINTA Shell* para auxílio a músicos, objetiva apoiar o especialista na realização de suas tarefas. O desenvolvimento de um protótipo deve necessariamente possuir uma especificação, onde se define os requisitos da aplicação.

O passo inicial da fase de especificação do sistema é um levantamento de dados e informações para nortear o desenvolvimento do protótipo. Esse levantamento é feito contatando-se o músico profissional, e elaborando-se uma descrição textual, por exemplo. A partir desta descrição, parte-se para a etapa de modelagem do protótipo. E com essas informações pôde-se realizar a especificação através da notação BNF (*Backus-Naur Form*).

5.2 ESPECIFICAÇÃO

A forma de representação do conhecimento escolhida para o sistema especialista são as regras de produção (ver seção 4.5). A estrutura das regras é representada abaixo, expressa pela notação BNF (*Backus-Naur Form*):

```

<regra> ::= SE <condição> ENTÃO <cláusula> <certeza>

<condição> ::= <cláusula> | <cláusula> E <cláusula>

<cláusula> ::= <atributo> <predicado> <valor>

<atributo> ::= <cadeia>

<predicado> ::= = | <>

<valor> ::= <cadeia>

<cadeia> ::= {letra minúscula}

<letra minúscula> ::= a | b | c | d | ..... | v | x | z ||

<certeza> ::= 1 | 2 | 3 | 4 | ..... | 98 | 99 | 100

```

Como se observa na BNF, a regra é formada pelas partes SE e ENTÃO. Na parte SE, ou antecedente, da regra estão as premissas (caudas). Ela é formada por uma ou mais cláusulas. Já na parte ENTÃO, ou conseqüente (cabeça), está a conclusão da regra que é formada por uma cláusula única.

Uma cláusula é uma combinação de um atributo, um predicado e um valor. Os atributos e valores são cadeias de caracteres que o usuário cria livremente. Os predicados utilizados na regra são “=” (igual) e “<>” (diferente). Para cada regra existe ainda um fator de certeza, este fator de certeza expressa o grau de certeza do conjunto de cláusulas que compõem a regra ([LIA2000]).

5.2.1 ATRIBUTOS E VALORES

Os atributos das regras são classificados em dois tipos: intermediário ou conclusivo. Um atributo conclusivo é aquele que representa uma solução para o problema, ou seja, existindo uma resposta para o atributo existirá uma resposta para o problema como um todo.

Já os atributos intermediários são os que representam uma conclusão parcial que exigirá continuidade ao processo de busca da solução do problema.

Existe um domínio de respostas que cada atributo pode assumir. Estas possíveis respostas que um determinado atributo pode assumir são denominadas de valores. Veja alguns exemplos abaixo:

- Atributo: estações do ano,
- Valores: {verão, inverno, outono, primavera}, e,
- Atributo: impressora não imprime,
- Valores: {falta papel, falta cartucho de impressão, impressora desligada}.

A criação de atributos e valores para posterior utilização no sistema é livre. Cabe ao usuário do sistema fazer o cadastramento prévio dos mesmos para utilizá-los na elaboração das regras.

5.2.2 CLÁUSULAS E PREDICADOS

As cláusulas das regras são formadas pela ligação de um atributo com um valor através do uso de um predicado. Os predicados disponíveis no *Expert SINTA Shell* são igualdade (“=”) e desigualdade (“<>”):

- Atributo: Estações do Ano,
- Predicado :=,
- Valor: verão.

5.2.3 REGRAS

São elas que compõem o conhecimento do sistema especialista, sendo construídas através da conjunção de cláusulas. Cada regra é composta por uma cláusula obrigatória, na sua parte antecedente e uma cláusula, também obrigatória, na sua parte conseqüente. Alternativamente a cláusula pode possuir na sua parte antecedente qualquer número de cláusulas ligadas pelos conectivos lógicos “=” e “<>”.

5.2.4 FATOR DE CERTEZA

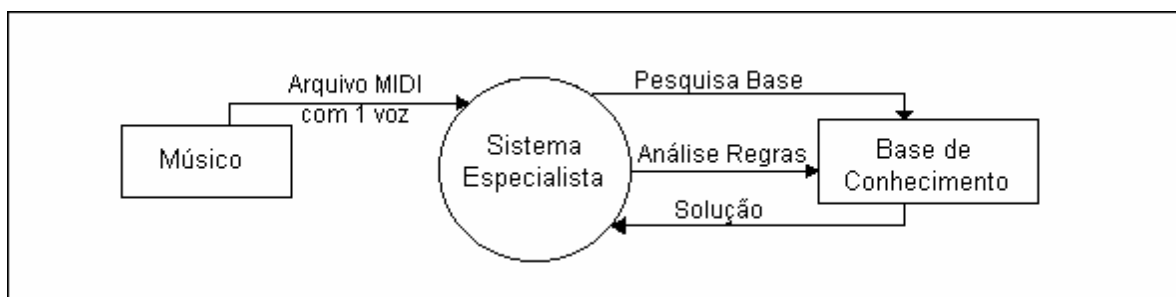
Existe um fator de certeza associado para cada regra na base de conhecimentos do sistema especialista. O fator de certeza é um número de 1 à 100 que demonstra o percentual de certeza do conhecimento descrito pela regra. Deve-se observar que o grau de certeza está relacionado à regra como um todo, ou seja, ao conjunto de cláusulas que compõem a regra.

5.3 MODELAGEM ESSENCIAL

Esta modelagem mostra, como o próprio nome diz, a essência do sistema a ser desenvolvido. É composta pelos modelos ambiental e comportamental. O modelo ambiental visa mostrar como o sistema interage com o ambiente externo e o modelo comportamental indica o que o sistema deve fazer para interagir com o ambiente externo. No presente trabalho somente foi desenvolvido o modelo ambiental.

O modelo ambiental é composto por um diagrama de contexto, que representa o fluxo de dados, e por uma lista de eventos, que representa as tarefas que devem ser executadas no sistema. A figura 29 mostra o diagrama de contexto.

FIGURA 29 – DIAGRAMA DE CONTEXTO



Os eventos do protótipo são:

1. Músico entra com um arquivo MIDI gerado num seqüenciador.
2. Sistema Especialista pesquisa dados na base de conhecimento.
3. Sistema Especialista analisa dados da base de conhecimento.
4. Base fornece a solução (conhecimento) para o sistema especialista.

5.4 PLATAFORMA DE DESENVOLVIMENTO

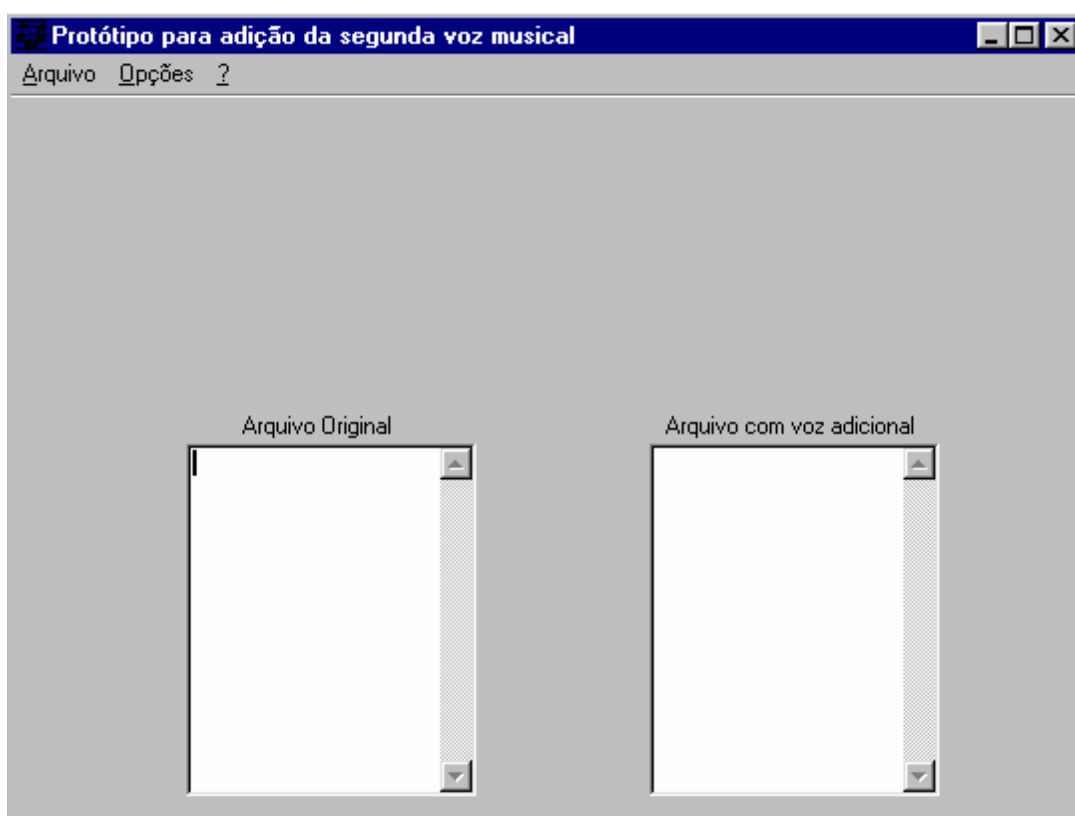
O protótipo foi desenvolvido para a plataforma PC em ambiente Windows, utilizando-se um notebook pentium 133 Mhz, e 16 Mb de memória RAM. Para implementar o protótipo optou-se pelo ambiente de programação visual Borland Delphi, em sua versão 3.0, pois a biblioteca *VCL Expert SINTA Shell* na versão utilizada para o desenvolvimento do protótipo foi desenvolvida para ser utilizada com Delphi 3.0. O ambiente de programação Delphi possui algumas características merecedoras de destaque, como: “abordagem baseada em formulários e orientada a objetos, suporte a banco de dados, integração com a programação em Windows e sua tecnologia de componentes” ([CAN1998]).

5.5 OPERACIONALIDADE DO PROTÓTIPO

A seguir, será apresentado o funcionamento do protótipo. Nesta apresentação, serão mostradas as telas do protótipo, bem como as características de cada uma delas.

A figura 30 é a primeira tela, sendo a abertura do sistema.

FIGURA 30 – TELA DE ABERTURA



Na opção de menu Arquivo -> Abrir (figura 31) é aberto o arquivo no qual se deseja adicionar uma segunda voz musical (figura 32). Este arquivo pode ser gerado em qualquer software musical, desde que for salvo no formato MIDI. O arquivo deverá ser gerado no formato 1, contendo três blocos **MTrk**. Sendo no primeiro bloco os eventos relacionados ao sinal de compasso, a velocidade da música, a armadura da clave, etc., no segundo bloco deverá conter somente a melodia, ou seja, a primeira voz musical, armazenados no canal 1 e no terceiro bloco, armazenados no canal 2, a harmonia musical (contendo as três notas musicais, que formam o acorde). Observa-se na figura 33, que foi utilizado o editor musical **Encore 4.21**, para a criação de uma partitura musical. Nota-se que o exemplo se refere a partitura da música noite feliz, e a melodia encontra-se no canal 1 e a harmonia no canal 2. A primeira nota musical é a nota de número 43 em hexadecimal, que convertendo em decimal, teremos o número 67.

FIGURA 31 – OPÇÃO PARA ABRIR O ARQUIVO MIDI

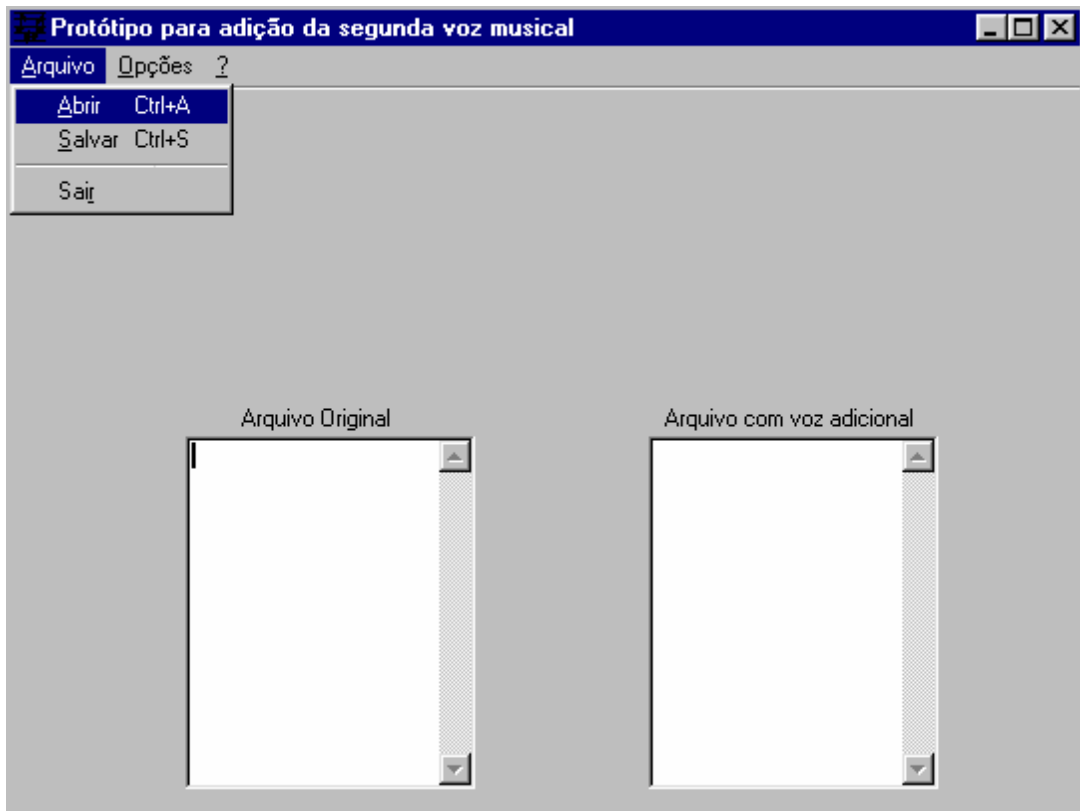


FIGURA 32 – ABRINDO O ARQUIVO MIDI

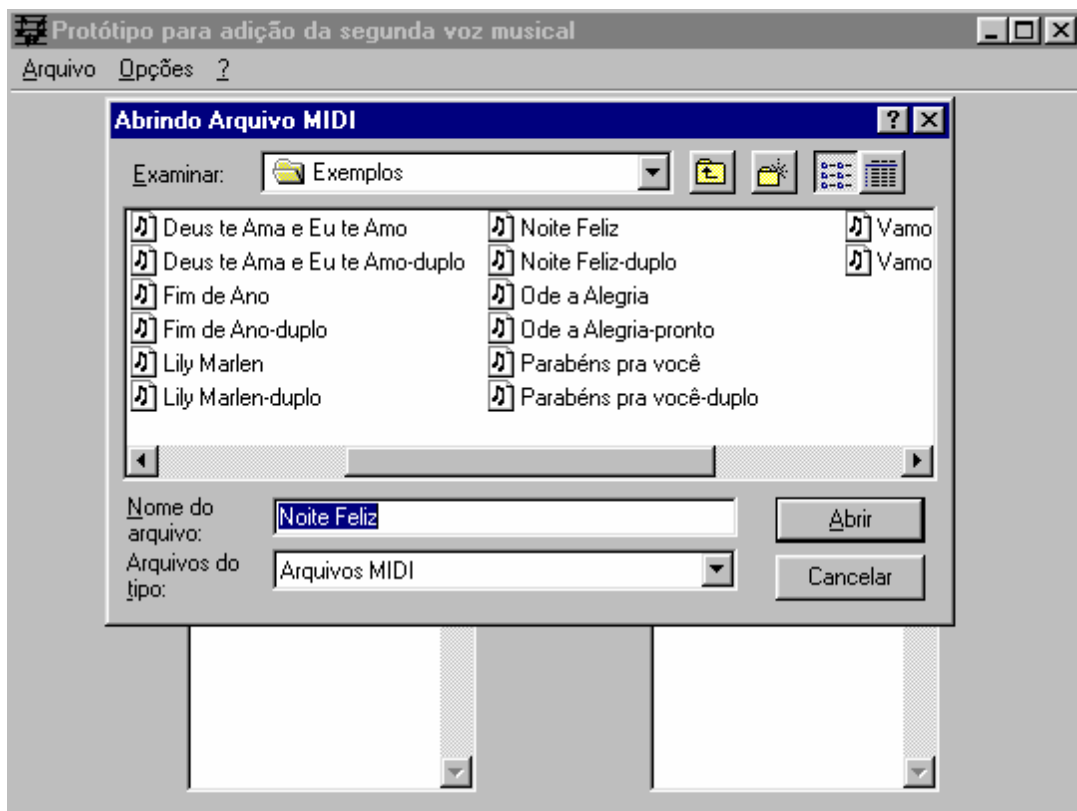


FIGURA 33 – EXEMPLO DE UMA PARTITURA CRIADA NO ENCORE 4.21

No momento em que o arquivo estiver aberto, o protótipo se encarregará de consultar a base de conhecimento que foi gerada no *Expert Sinta Shell* (figura 34) e automaticamente verificará todas as notas que compõem a melodia e incluirá de acordo com a harmonia, a nota que mais se enquadra para formar a segunda voz musical.

Na figura 35 tem-se um exemplo de como é feito a consulta na ferramenta *Expert Sinta Shell*, usando o número “67” na entrada, que equivale a nota “Sol”, que é a primeira nota da música noite feliz. Convertendo em hexadecimal o número é “43”. Após ter sido entrado com o número da nota em decimal, deve-se selecionar qual o acorde (figura 36) que se encontra debaixo da nota, no canal 2. Após isso o sistema percorrerá toda a base de conhecimento e retornará a solução de acordo com as regras. Neste exemplo retornou como solução a nota de número “40” em hexadecimal, que representa a nota “mi” (figura 37).

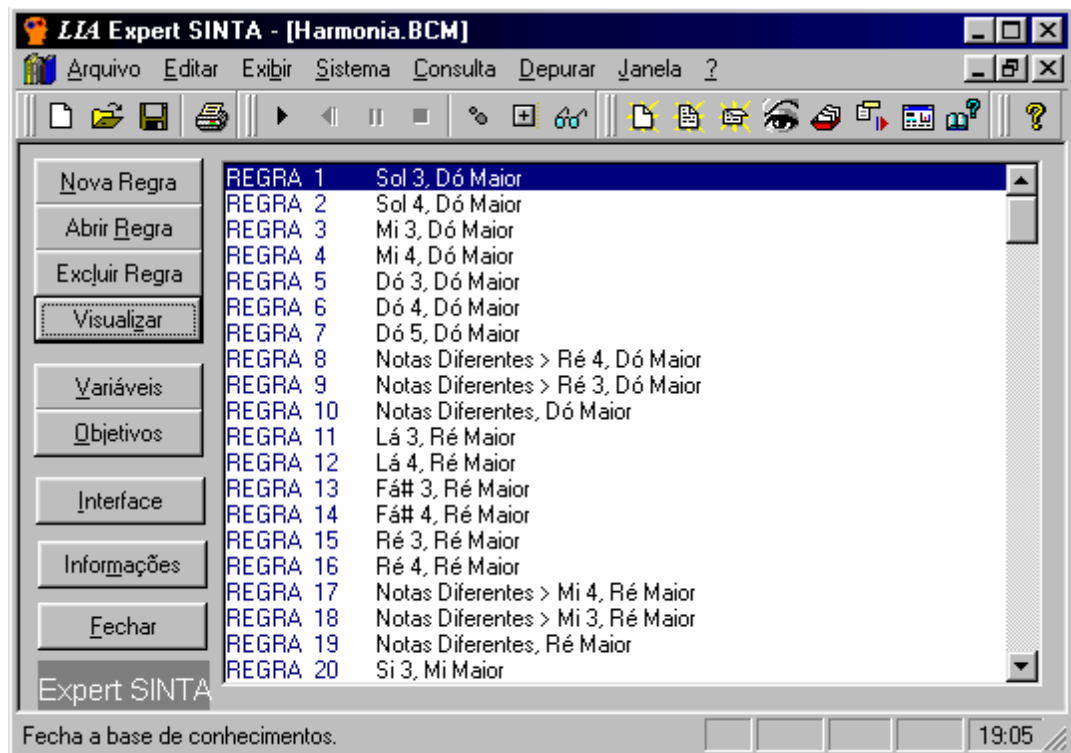
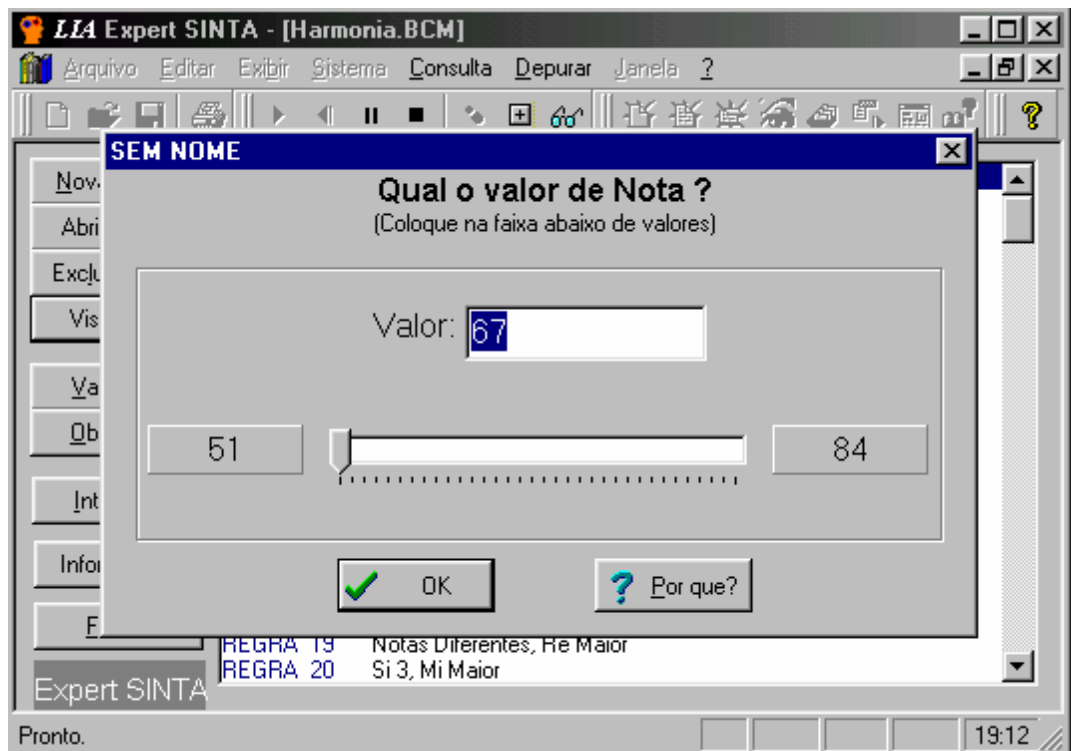
FIGURA 34 – BASE DE CONHECIMENTO GERADA NO *EXPERT SINTA SHELL*FIGURA 35 – EXEMPLO DE CONSULTA NO *EXPERT SINTA SHELL*

FIGURA 36 – OPÇÃO DE ESCOLHA DO ACORDE

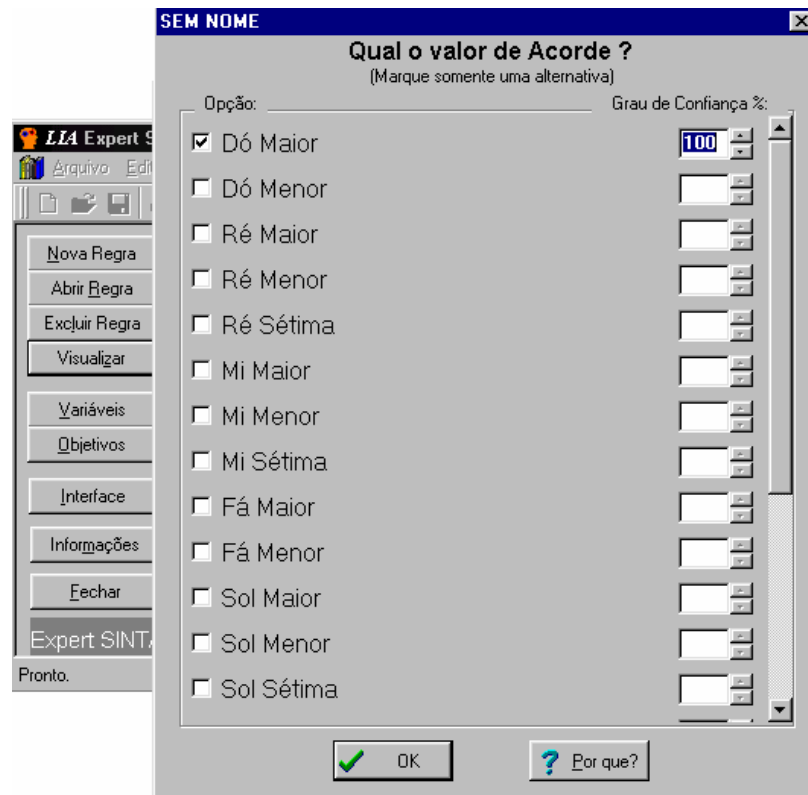


FIGURA 37 – SOLUÇÃO DA CONSULTA



Na figura 38 temos o histórico de como ele percorreu a base de conhecimento e na figura 39 uma amostragem de todos os valores de entrada e solução.

FIGURA 38 – HISTÓRICO DA SOLUÇÃO

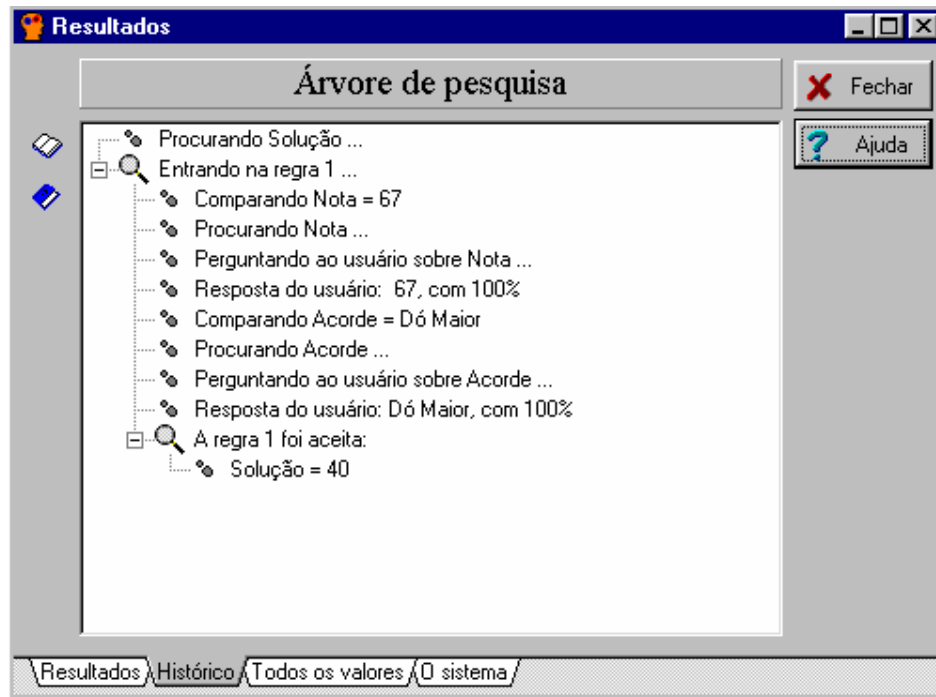


FIGURA 39 – TODOS OS VALORES DA REFERIDA CONSULTA



Na figura 40 temos no uma descrição detalhada de como foi adicionada a segunda voz musical no protótipo, bem como o evento e o tempo de nota. Após o sistema especialista ter feito a verificação em todas as notas musicais, o arquivo estará pronto para ser salvo na opção de menu Arquivo -> Salvar (figura 41).

FIGURA 40 – A SOLUÇÃO NO PROTÓTIPO

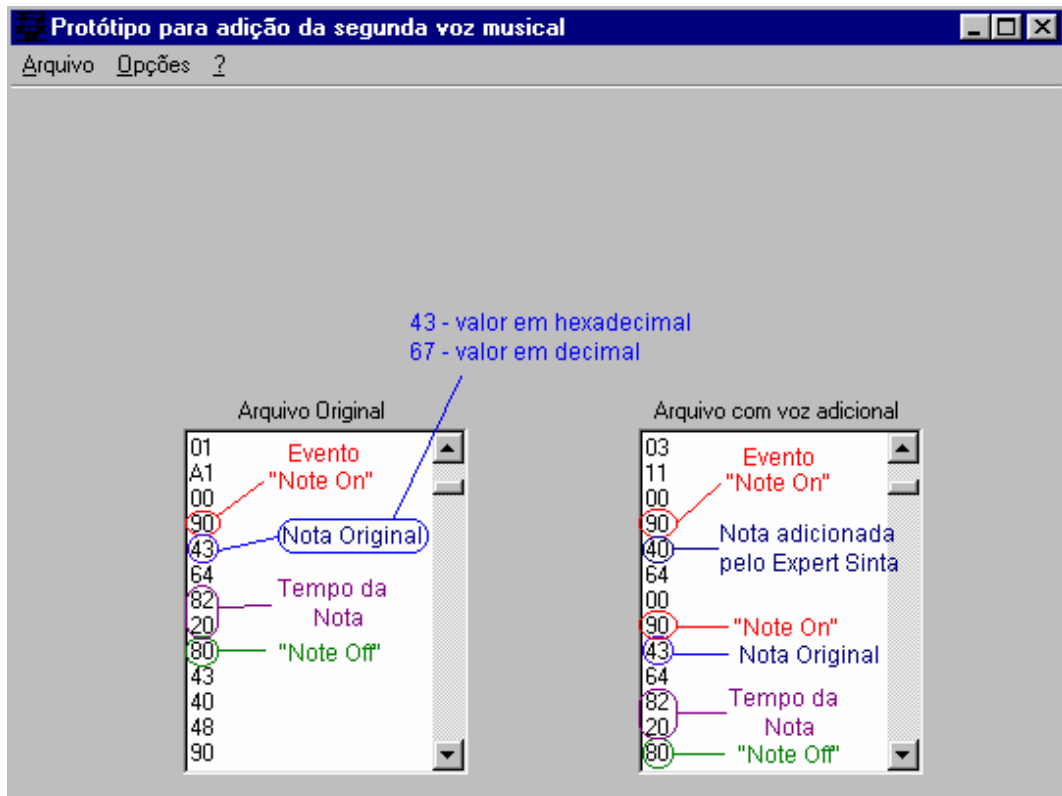
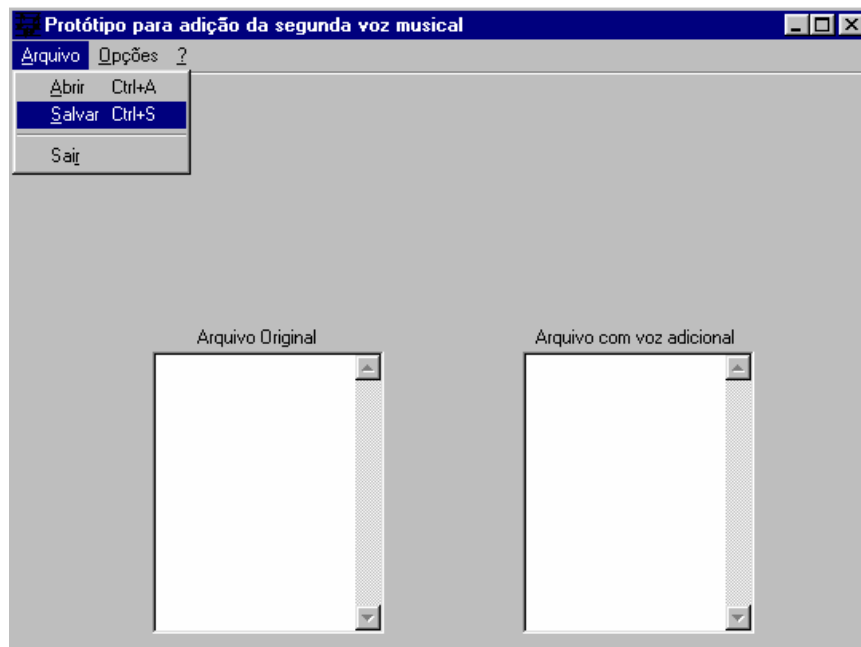


FIGURA 41 – OPÇÃO PARA SALVAR O ARQUIVO MIDI APÓS A SOLUÇÃO



Na opção de menu Opções -> Executar Original, poderá ser ouvido o arquivo MIDI antes de conter a segunda voz (figura 42), e na figura 43 tem-se o arquivo MIDI original executando. Na opção de menu Opções -> Executar Atual, poderá ser ouvido o arquivo MIDI

após ter sido feita a complementação da segunda voz musical, através do sistema especialista (figura 44), e na figura 45 tem-se o arquivo MIDI executando, após a adição da segunda voz musical.

FIGURA 42 – OPÇÃO PARA EXECUTAR O ORIGINAL

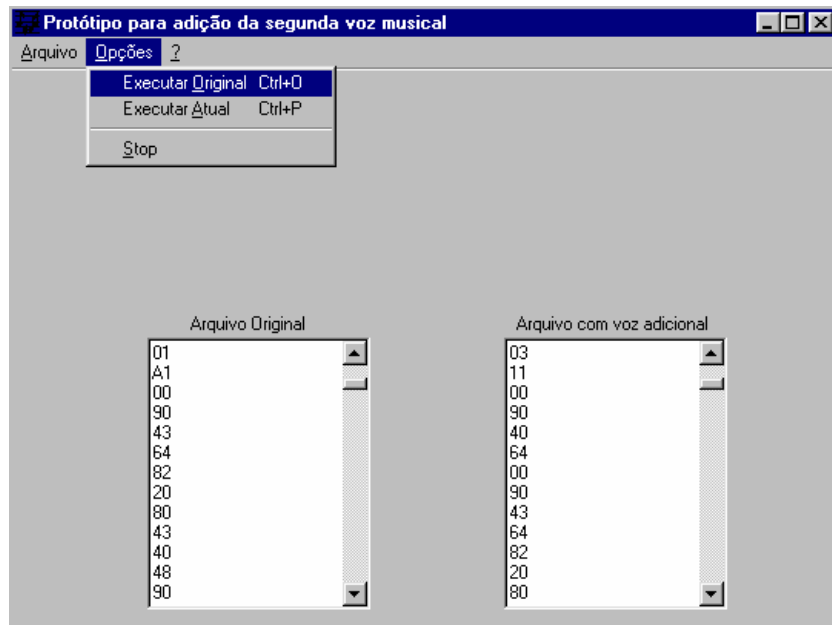


FIGURA 43 – TOCANDO O ARQUIVO ORIGINAL

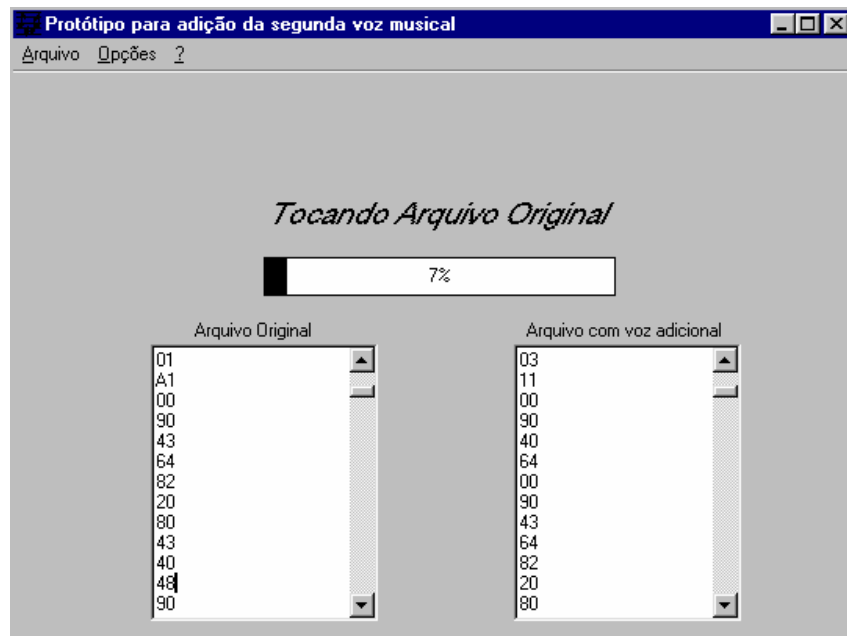


FIGURA 44 – OPÇÃO PARA EXECUTAR O ARQUIVO MIDI APÓS A SOLUÇÃO

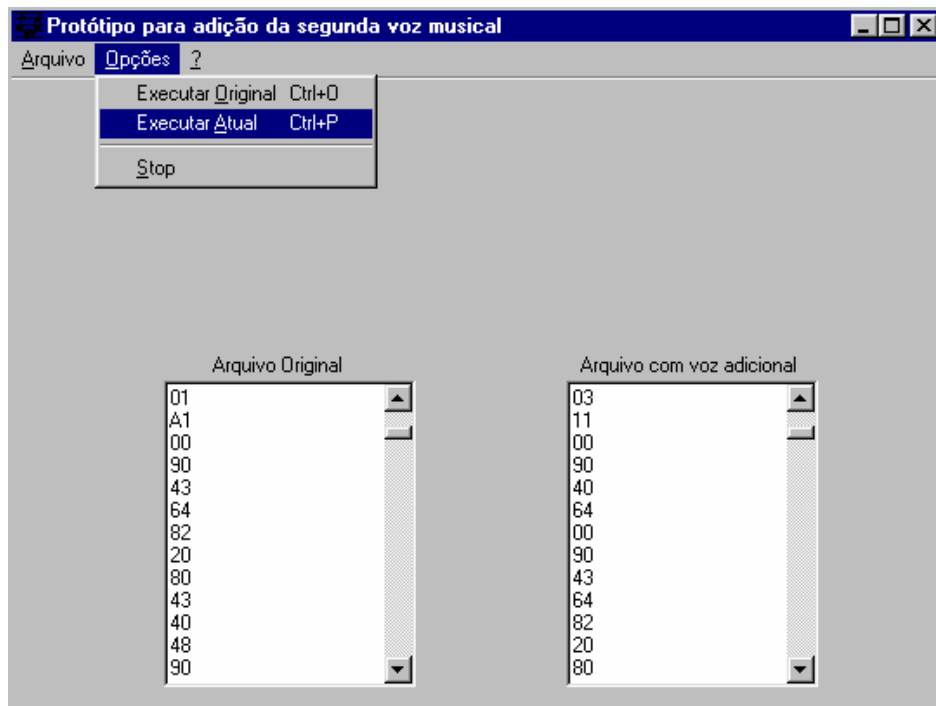
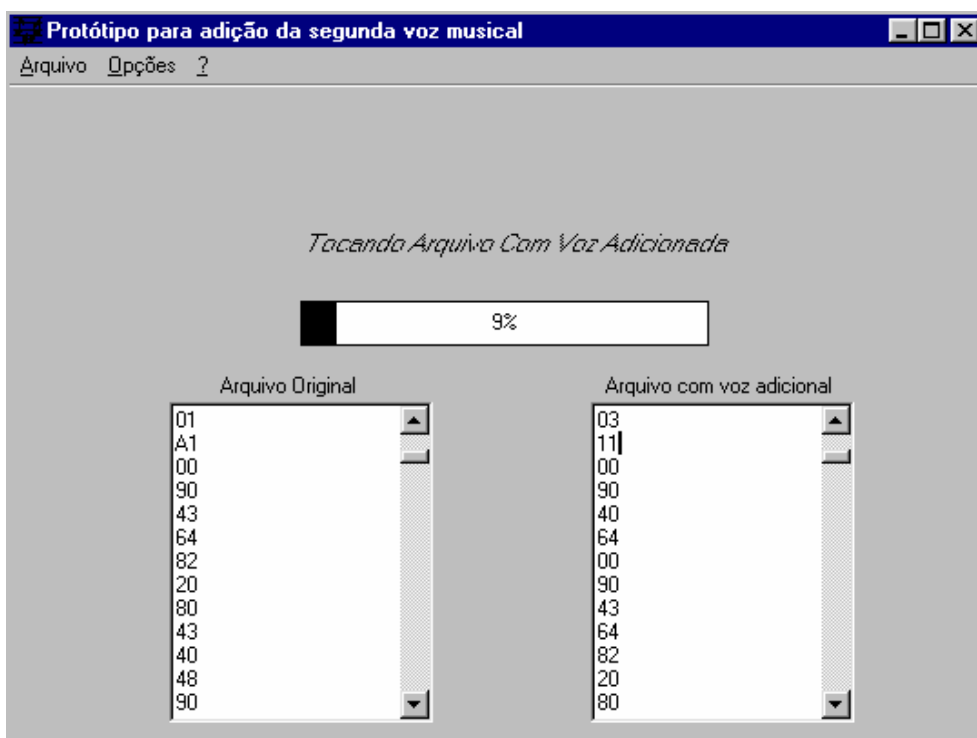


FIGURA 45 – EXECUTANDO ARQUIVO MIDI APÓS A SOLUÇÃO



Após a protótipo ter concluído a consulta em todas as notas, o arquivo salvo foi aberto no editor musical Encore 4.21 e pôde-se observar a segunda voz musical adicionada abaixo da primeira voz musical (figura 46).

FIGURA 46 – RESULTADO NO ENCORE 4.21 APÓS A SOLUÇÃO

The image shows a screenshot of the Encore software interface. The window title is "Encore - [C:\ATCC\Exemplos\Noite Feliz-duplo.MID - MIDI File]". The menu bar includes "File", "Edit", "Notes", "Measures", "Score", "View", "Windows", "Setup", and "Help". The toolbar contains various icons for playback and editing, including "Voice -", "M2", and navigation arrows. The main area displays four staves of musical notation in 3/4 time. The notation consists of chords and single notes across the staves. The taskbar at the bottom shows the "Iniciar" button and the application name "Encore - [C:\ATCC\Exemplos\Noite Feliz-duplo.MID - MIDI File]". The system clock in the bottom right corner shows "15:43".

O conjunto completo das regras que compõem a base de conhecimento encontra-se no Anexo 1 do presente relatório.

6 CONCLUSÕES E SUGESTÕES

6.1 CONCLUSÕES

A proposta do presente trabalho foi de demonstrar a utilização da tecnologia dos sistemas especialistas, mais precisamente a ferramenta *Expert Sinta Shell*, aplicados na área da música, objetivando auxiliar o músico na inclusão de uma segunda voz de uma forma rápida e eficiente na escrita de partituras para uma banda ou orquestra.

O sistema especialista no domínio do conhecimento no qual foi construído, demonstrou que os objetivos propostos foram alcançados. A técnica *Expert Sinta Shell* demonstrou a possibilidade de utilização de sistemas especialistas na área da música, na forma de uma ferramenta de apoio à decisão, liberando o especialista para outras atividades correlacionadas.

O sistema especialista desenvolvido pode trazer a músicos uma maior agilidade na escrita de partituras, pois muitos músicos amadores enfrentam dificuldades na adição de uma segunda voz musical, principalmente quando não se tem conhecimento de harmonia musical.

6.2 LIMITAÇÕES

O protótipo construído possui algumas limitações. Pode-se citar algumas:

- a) foram utilizados somente acordes maiores, menores e de sétima, e, acordes sem acidente, devido ao aumento excessivo de regras;
- b) limitou-se a entrada de arquivo MIDI ao uso do formato 1, pois este formato separa os canais em blocos diferentes;

- c) neste protótipo foi utilizado somente a nota de um acorde, que melhor soava juntamente com a nota da primeira voz musical.

6.3 SUGESTÕES PARA TRABALHOS FUTUROS

Para as tecnologias apresentadas neste estudo, inúmeros caminhos poderão se abrir, mostrando um vasto número de aplicações possíveis. No caso específico deste trabalho, poderia ser feito um estudo mais aprofundado sobre harmonia musical para que fosse utilizado todas as maneiras possíveis de notas cabíveis com a primeira voz musical.

Poderia ser utilizado outras técnicas de inteligência artificial que também podem chegar a uma solução, visto que em uma música, poderá ter algumas notas musicais adicionadas que não soem tão bem, como em outra música que tenha a mesma nota musical.

ANEXO 1

REGRAS DA BASE DE CONHECIMENTO

* Valores em decimal na variável **nota**

* Valores em hexadecimal na variável **solução**

Regra 1

SE Nota = 67

E Acorde = Dó Maior

ENTÃO Solução = 40 CNF 100%

Regra 2

SE Nota = 79

E Acorde = Dó Maior

ENTÃO Solução = 4C CNF 100%

Regra 3

SE Nota = 64

E Acorde = Dó Maior

ENTÃO Solução = 3C CNF 100%

Regra 4

SE Nota = 76

E Acorde = Dó Maior

ENTÃO Solução = 48 CNF 100%

Regra 5

SE Nota = 60

E Acorde = Dó Maior

ENTÃO Solução = 34 CNF 100%

Regra 6

SE Nota = 72

E Acorde = Dó Maior

ENTÃO Solução = 40 CNF 100%

Regra 7

SE Nota = 84

E Acorde = Dó Maior

ENTÃO Solução = 4C CNF 100%

Regra 8

SE Nota <> 84

OU Nota <> 79

OU Nota <> 76

E Nota > 74

E Acorde Anterior <> Dó Maior

E Acorde = Dó Maior

ENTÃO Solução = 48 CNF 100%

Regra 9

SE Nota <> 72

OU Nota <> 67

OU Nota <> 64

E Nota > 62

E Acorde Anterior <> Dó Maior

E Acorde = Dó Maior

ENTÃO Solução = 3C CNF 100%

Regra 10

SE Nota <> 84

OU Nota <> 79

OU Nota <> 76

OU Nota <> 72

OU Nota <> 67

OU Nota <> 64

OU Nota <> 60

E Acorde Anterior = Dó Maior

E Acorde = Dó Maior

ENTÃO Solução = Nota Anterior CNF 100%

Regra 11

SE Nota = 69

E Acorde = Ré Maior

ENTÃO Solução = 42 CNF 100%

Regra 12

SE Nota = 81

E Acorde = Ré Maior

ENTÃO Solução = 4E CNF 100%

Regra 13

SE Nota = 66

E Acorde = Ré Maior

ENTÃO Solução = 3E CNF 100%

Regra 14

SE Nota = 78

E Acorde = Ré Maior

ENTÃO Solução = 4A CNF 100%

Regra 15

SE Nota = 62

E Acorde = Ré Maior

ENTÃO Solução = 36 CNF 100%

Regra 16

SE Nota = 74

E Acorde = Ré Maior

ENTÃO Solução = 42 CNF 100%

Regra 17

SE Nota <> 81

OU Nota <> 78

E Nota > 76

E Acorde Anterior <> Ré Maior

E Acorde = Ré Maior

ENTÃO Solução = 4A CNF 100%

Regra 18

SE Nota <> 74

OU Nota <> 69

OU Nota <> 66

E Nota > 64

E Acorde Anterior <> Ré Maior

E Acorde = Ré Maior

ENTÃO Solução = 3E CNF 100%

Regra 19

SE Nota <> 81

OU Nota <> 78

OU Nota <> 74

OU Nota <> 69

OU Nota <> 66

OU Nota <> 62

E Acorde Anterior = Ré Maior

E Acorde = Ré Maior

ENTÃO Solução = Nota Anterior CNF 100%

Regra 20

SE Nota = 71

E Acorde = Mi Maior

ENTÃO Solução = 44 CNF 100%

Regra 21

SE Nota = 83

E Acorde = Mi Maior

ENTÃO Solução = 50 CNF 100%

Regra 22

SE Nota = 68

E Acorde = Mi Maior

ENTÃO Solução = 40 CNF 100%

Regra 23

SE Nota = 80

E Acorde = Mi Maior

ENTÃO Solução = 4C CNF 100%

Regra 24

SE Nota = 64

E Acorde = Mi Maior

ENTÃO Solução = 38 CNF 100%

Regra 25

SE Nota = 76

E Acorde = Mi Maior

ENTÃO Solução = 44 CNF 100%

Regra 26

SE Nota <> 83

OU Nota <> 80

E Nota > 78

E Acorde Anterior <> Mi Maior

E Acorde = Mi Maior

ENTÃO Solução = 4C CNF 100%

Regra 27

SE Nota <> 76

OU Nota <> 71

OU Nota <> 68

E Nota > 66

E Acorde Anterior <> Mi Maior

E Acorde = Mi Maior

ENTÃO Solução = 40 CNF 100%

Regra 28

SE Nota <> 83

OU Nota <> 80

OU Nota <> 76

OU Nota <> 71

OU Nota <> 68

OU Nota <> 64

E Acorde Anterior = Mi Maior

E Acorde = Mi Maior

ENTÃO Solução = Nota Anterior CNF 100%

Regra 29

SE Nota = 60

E Acorde = Fá Maior

ENTÃO Solução = 39 CNF 100%

Regra 30

SE Nota = 72

E Acorde = Fá Maior

ENTÃO Solução = 45 CNF 100%

Regra 31

SE Nota = 84

E Acorde = Fá Maior

ENTÃO Solução = 51 CNF 100%

Regra 32

SE Nota = 69

E Acorde = Fá Maior

ENTÃO Solução = 41 CNF 100%

Regra 33

SE Nota = 81

E Acorde = Fá Maior

ENTÃO Solução = 4D CNF 100%

Regra 34

SE Nota = 65

E Acorde = Fá Maior

ENTÃO Solução = 39 CNF 100%

Regra 35

SE Nota = 77

E Acorde = Fá Maior

ENTÃO Solução = 45 CNF 100%

Regra 36

SE Nota <> 84

OU Nota <> 81

E Nota > 79

E Acorde Anterior <> Fá Maior

E Acorde = Fá Maior

ENTÃO Solução = 4D CNF 100%

Regra 37

SE Nota <> 77
OU Nota <> 72
OU Nota <> 69
E Nota > 67
E Acorde Anterior <> Fá Maior
E Acorde = Fá Maior
ENTÃO Solução = 41 CNF 100%

Regra 38

SE Nota <> 65
OU Nota <> 60
E Nota > 59
E Acorde Anterior <> Fá Maior
E Acorde = Fá Maior
ENTÃO Solução = 39 CNF 100%

Regra 39

SE Nota <> 84
OU Nota <> 81
OU Nota <> 77
OU Nota <> 72
OU Nota <> 69
OU Nota <> 65
OU Nota <> 60
E Acorde Anterior = Fá Maior
E Acorde = Fá Maior
ENTÃO Solução = Nota Anterior CNF 100%

Regra 40

SE Nota = 62

E Acorde = Sol Maior

ENTÃO Solução = 3B CNF 100%

Regra 41

SE Nota = 74

E Acorde = Sol Maior

ENTÃO Solução = 47 CNF 100%

Regra 42

SE Nota = 71

E Acorde = Sol Maior

ENTÃO Solução = 43 CNF 100%

Regra 43

SE Nota = 83

E Acorde = Sol Maior

ENTÃO Solução = 4F CNF 100%

Regra 44

SE Nota = 67

E Acorde = Sol Maior

ENTÃO Solução = 3B CNF 100%

Regra 45

SE Nota = 79

E Acorde = Sol Maior

ENTÃO Solução = 47 CNF 100%

Regra 46

SE Nota <> 83
E Nota > 81
E Acorde Anterior <> Sol Maior
E Acorde = Sol Maior
ENTÃO Solução = 4F CNF 100%

Regra 47

SE Nota <> 79
OU Nota <> 74
OU Nota <> 71
E Nota > 69
E Acorde Anterior <> Sol Maior
E Acorde = Sol Maior
ENTÃO Solução = 43 CNF 100%

Regra 48

SE Nota <> 67
OU Nota <> 62
E Nota > 61
E Acorde Anterior <> Sol Maior
E Acorde = Sol Maior
ENTÃO Solução = 3B CNF 100%

Regra 49

SE Nota <> 83
OU Nota <> 79
OU Nota <> 74
OU Nota <> 71

OU Nota <> 67

OU Nota <> 62

E Acorde Anterior = Sol Maior

E Acorde = Sol Maior

ENTÃO Solução = Nota Anterior CNF 100%

Regra 50

SE Nota = 64

E Acorde = Lá Maior

ENTÃO Solução = 3D CNF 100%

Regra 51

SE Nota = 76

E Acorde = Lá Maior

ENTÃO Solução = 49 CNF 100%

Regra 52

SE Nota = 61

E Acorde = Lá Maior

ENTÃO Solução = 39 CNF 100%

Regra 53

SE Nota = 73

E Acorde = Lá Maior

ENTÃO Solução = 45 CNF 100%

Regra 54

SE Nota = 69

E Acorde = Lá Maior

ENTÃO Solução = 3D CNF 100%

Regra 55

SE Nota = 81

E Acorde = Lá Maior

ENTÃO Solução = 49 CNF 100%

Regra 56

SE Nota <> 81

OU Nota <> 76

OU Nota <> 73

E Nota > 71

E Acorde Anterior <> Lá Maior

E Acorde = Lá Maior

ENTÃO Solução = 45 CNF 100%

Regra 57

SE Nota <> 69

OU Nota <> 64

OU Nota <> 61

E Nota > 60

E Acorde Anterior <> Lá Maior

E Acorde = Lá Maior

ENTÃO Solução = 39 CNF 100%

Regra 58

SE Nota <> 81

OU Nota <> 76

OU Nota <> 73

OU Nota <> 69

OU Nota <> 64

OU Nota <> 61

E Acorde Anterior = Lá Maior

E Acorde = Lá Maior

ENTÃO Solução = Nota Anterior CNF 100%

Regra 59

SE Nota = 65

E Acorde = Sib Maior

ENTÃO Solução = 3E CNF 100%

Regra 60

SE Nota = 77

E Acorde = Sib Maior

ENTÃO Solução = 4A CNF 100%

Regra 61

SE Nota = 62

E Acorde = Sib Maior

ENTÃO Solução = 3A CNF 100%

Regra 62

SE Nota = 74

E Acorde = Sib Maior

ENTÃO Solução = 46 CNF 100%

Regra 63

SE Nota = 70

E Acorde = Sib Maior
ENTÃO Solução = 3E CNF 100%

Regra 64

SE Nota = 82
E Acorde = Sib Maior
ENTÃO Solução = 4A CNF 100%

Regra 65

SE Nota <> 82
OU Nota <> 77
OU Nota <> 74
E Nota > 72
E Acorde Anterior <> Sib Maior
E Acorde = Sib Maior
ENTÃO Solução = 46 CNF 100%

Regra 66

SE Nota <> 70
OU Nota <> 65
OU Nota <> 62
E Nota > 60
E Acorde Anterior <> Sib Maior
E Acorde = Sib Maior
ENTÃO Solução = 3A CNF 100%

Regra 67

SE Nota <> 82
OU Nota <> 77

OU Nota <> 74

OU Nota <> 70

OU Nota <> 65

OU Nota <> 62

E Acorde Anterior = Sib Maior

E Acorde = Sib Maior

ENTÃO Solução = Nota Anterior CNF 100%

Regra 68

SE Nota = 66

E Acorde = Si Maior

ENTÃO Solução = 3F CNF 100%

Regra 69

SE Nota = 78

E Acorde = Si Maior

ENTÃO Solução = 4B CNF 100%

Regra 70

SE Nota = 63

E Acorde = Si Maior

ENTÃO Solução = 3B CNF 100%

Regra 71

SE Nota = 75

E Acorde = Si Maior

ENTÃO Solução = 47 CNF 100%

Regra 72

SE Nota = 71
E Acorde = Si Maior
ENTÃO Solução = 3F CNF 100%

Regra 73

SE Nota = 83
E Acorde = Si Maior
ENTÃO Solução = 4B CNF 100%

Regra 74

SE Nota <> 83
OU Nota <> 78
OU Nota <> 75
E Nota > 73
E Acorde Anterior <> Si Maior
E Acorde = Si Maior
ENTÃO Solução = 47 CNF 100%

Regra 75

SE Nota <> 71
OU Nota <> 66
OU Nota <> 63
E Nota > 61
E Acorde Anterior <> Si Maior
E Acorde = Si Maior
ENTÃO Solução = 3B CNF 100%

Regra 76

SE Nota <> 83

OU Nota <> 78

OU Nota <> 75

OU Nota <> 71

OU Nota <> 66

OU Nota <> 63

E Acorde Anterior = Si Maior

E Acorde = Si Maior

ENTÃO Solução = Nota Anterior CNF 100%

Regra 77

SE Nota = 67

E Acorde = Dó Menor

ENTÃO Solução = 3F CNF 100%

Regra 78

SE Nota = 79

E Acorde = Dó Menor

ENTÃO Solução = 4B CNF 100%

Regra 79

SE Nota = 63

E Acorde = Dó Menor

ENTÃO Solução = 3C CNF 100%

Regra 80

SE Nota = 75

E Acorde = Dó Menor

ENTÃO Solução = 48 CNF 100%

Regra 81

SE Nota = 60

E Acorde = Dó Menor

ENTÃO Solução = 33 CNF 100%

Regra 82

SE Nota = 72

E Acorde = Dó Menor

ENTÃO Solução = 3F CNF 100%

Regra 83

SE Nota = 84

E Acorde = Dó Menor

ENTÃO Solução = 4B CNF 100%

Regra 84

SE Nota <> 84

OU Nota <> 79

OU Nota <> 75

E Nota > 74

E Acorde Anterior <> Dó Menor

E Acorde = Dó Menor

ENTÃO Solução = 48 CNF 100%

Regra 85

SE Nota <> 72

OU Nota <> 67

OU Nota <> 63

E Nota > 62

E Acorde Anterior <> Dó Menor

E Acorde = Dó Menor

ENTÃO Solução = 3C CNF 100%

Regra 86

SE Nota <> 84

OU Nota <> 79

OU Nota <> 75

OU Nota <> 72

OU Nota <> 67

OU Nota <> 63

OU Nota <> 60

E Acorde Anterior = Dó Menor

E Acorde = Dó Menor

ENTÃO Solução = Nota Anterior CNF 100%

Regra 87

SE Nota = 69

E Acorde = Ré Menor

ENTÃO Solução = 4I CNF 100%

Regra 88

SE Nota = 81

E Acorde = Ré Menor

ENTÃO Solução = 4D CNF 100%

Regra 89

SE Nota = 65

E Acorde = Ré Menor

ENTÃO Solução = 3E CNF 100%

Regra 90

SE Nota = 77

E Acorde = Ré Menor

ENTÃO Solução = 4A CNF 100%

Regra 91

SE Nota = 62

E Acorde = Ré Menor

ENTÃO Solução = 35 CNF 100%

Regra 92

SE Nota = 74

E Acorde = Ré Menor

ENTÃO Solução = 41 CNF 100%

Regra 93

SE Nota <> 81

OU Nota <> 77

E Nota > 76

E Acorde Anterior <> Ré Menor

E Acorde = Ré Menor

ENTÃO Solução = 4A CNF 100%

Regra 94

SE Nota <> 74

OU Nota <> 69

OU Nota <> 65

OU Nota <> 62
E Nota > 64
E Acorde Anterior <> Ré Menor
E Acorde = Ré Menor
ENTÃO Solução = 3E CNF 100%

Regra 95

SE Nota <> 81
OU Nota <> 77
OU Nota <> 74
OU Nota <> 69
OU Nota <> 65
OU Nota <> 62
E Acorde Anterior = Ré Menor
E Acorde = Ré Menor
ENTÃO Solução = Nota Anterior CNF 100%

Regra 96

SE Nota = 71
E Acorde = Mi Menor
ENTÃO Solução = 43 CNF 100%

Regra 97

SE Nota = 83
E Acorde = Mi Menor
ENTÃO Solução = 4F CNF 100%

Regra 98

SE Nota = 67

E Acorde = Mi Menor

ENTÃO Solução = 40 CNF 100%

Regra 99

SE Nota = 79

E Acorde = Mi Menor

ENTÃO Solução = 4C CNF 100%

Regra 100

SE Nota = 64

E Acorde = Mi Menor

ENTÃO Solução = 37 CNF 100%

Regra 101

SE Nota = 76

E Acorde = Mi Menor

ENTÃO Solução = 43 CNF 100%

Regra 102

SE Nota <> 83

OU Nota <> 79

E Nota > 78

E Acorde Anterior <> Mi Menor

E Acorde = Mi Menor

ENTÃO Solução = 4C CNF 100%

Regra 103

SE Nota <> 76

OU Nota <> 71

OU Nota <> 67
E Nota > 66
E Acorde Anterior <> Mi Menor
E Acorde = Mi Menor
ENTÃO Solução = 40 CNF 100%

Regra 104

SE Nota <> 83
OU Nota <> 79
OU Nota <> 76
OU Nota <> 71
OU Nota <> 67
OU Nota <> 64
E Acorde Anterior = Mi Menor
E Acorde = Mi Menor
ENTÃO Solução = Nota Anterior CNF 100%

Regra 105

SE Nota = 60
E Acorde = Fá Menor
ENTÃO Solução = 38 CNF 100%

Regra 106

SE Nota = 72
E Acorde = Fá Menor
ENTÃO Solução = 44 CNF 100%

Regra 107

SE Nota = 84

E Acorde = Fá Menor

ENTÃO Solução = 50 CNF 100%

Regra 108

SE Nota = 68

E Acorde = Fá Menor

ENTÃO Solução = 41 CNF 100%

Regra 109

SE Nota = 80

E Acorde = Fá Menor

ENTÃO Solução = 4D CNF 100%

Regra 110

SE Nota = 65

E Acorde = Fá Menor

ENTÃO Solução = 38 CNF 100%

Regra 111

SE Nota = 77

E Acorde = Fá Menor

ENTÃO Solução = 44 CNF 100%

Regra 112

SE Nota <> 84

OU Nota <> 80

E Nota > 79

E Acorde Anterior <> Fá Menor

E Acorde = Fá Menor

ENTÃO Solução = 4D CNF 100%

Regra 113

SE Nota <> 77

OU Nota <> 72

OU Nota <> 68

E Nota > 67

E Acorde Anterior <> Fá Menor

E Acorde = Fá Menor

ENTÃO Solução = 41 CNF 100%

Regra 114

SE Nota <> 65

OU Nota <> 60

E Nota > 59

E Acorde Anterior <> Fá Menor

E Acorde = Fá Menor

ENTÃO Solução = 38 CNF 100%

Regra 115

SE Nota <> 84

OU Nota <> 80

OU Nota <> 77

OU Nota <> 72

OU Nota <> 68

OU Nota <> 65

OU Nota <> 60

E Acorde Anterior = Fá Menor

E Acorde = Fá Menor

ENTÃO Solução = Nota Anterior CNF 100%

Regra 116

SE Nota = 62

E Acorde = Sol Menor

ENTÃO Solução = 3A CNF 100%

Regra 117

SE Nota = 74

E Acorde = Sol Menor

ENTÃO Solução = 46 CNF 100%

Regra 118

SE Nota = 70

E Acorde = Sol Menor

ENTÃO Solução = 43 CNF 100%

Regra 119

SE Nota = 82

E Acorde = Sol Menor

ENTÃO Solução = 4F CNF 100%

Regra 120

SE Nota = 67

E Acorde = Sol Menor

ENTÃO Solução = 3A CNF 100%

Regra 121

SE Nota = 79

E Acorde = Sol Menor
ENTÃO Solução = 46 CNF 100%

Regra 122

SE Nota <> 82
E Nota > 81
E Acorde Anterior <> Sol Menor
E Acorde = Sol Menor
ENTÃO Solução = 4F CNF 100%

Regra 123

SE Nota <> 79
OU Nota <> 74
OU Nota <> 70
E Nota > 69
E Acorde Anterior <> Sol Menor
E Acorde = Sol Menor
ENTÃO Solução = 43 CNF 100%

Regra 124

SE Nota <> 67
OU Nota <> 62
E Nota > 61
E Acorde Anterior <> Sol Menor
E Acorde = Sol Menor
ENTÃO Solução = 3A CNF 100%

Regra 125

SE Nota <> 82

OU Nota <> 79

OU Nota <> 74

OU Nota <> 70

OU Nota <> 67

OU Nota <> 62

E Acorde Anterior = Sol Menor

E Acorde = Sol Menor

ENTÃO Solução = Nota Anterior CNF 100%

Regra 126

SE Nota = 64

E Acorde = Lá Menor

ENTÃO Solução = 3C CNF 100%

Regra 127

SE Nota = 76

E Acorde = Lá Menor

ENTÃO Solução = 48 CNF 100%

Regra 128

SE Nota = 60

E Acorde = Lá Menor

ENTÃO Solução = 39 CNF 100%

Regra 129

SE Nota = 72

E Acorde = Lá Menor

ENTÃO Solução = 45 CNF 100%

Regra 130

SE Nota = 84

E Acorde = Lá Menor

ENTÃO Solução = 51 CNF 100%

Regra 131

SE Nota = 69

E Acorde = Lá Menor

ENTÃO Solução = 3C CNF 100%

Regra 132

SE Nota = 81

E Acorde = Lá Menor

ENTÃO Solução = 48 CNF 100%

Regra 133

SE Nota <> 84

OU Nota <> 81

OU Nota <> 76

OU Nota <> 72

E Nota > 71

E Acorde Anterior <> Lá Menor

E Acorde = Lá Menor

ENTÃO Solução = 45 CNF 100%

Regra 134

SE Nota <> 69

OU Nota <> 64

OU Nota <> 60

E Nota > 59
E Acorde Anterior <> Lá Menor
E Acorde = Lá Menor
ENTÃO Solução = 39 CNF 100%

Regra 135

SE Nota <> 84
OU Nota <> 81
OU Nota <> 76
OU Nota <> 72
OU Nota <> 69
OU Nota <> 64
OU Nota <> 60
E Acorde Anterior = Lá Menor
E Acorde = Lá Menor
ENTÃO Solução = Nota Anterior CNF 100%

Regra 136

SE Nota = 65
E Acorde = Sib Menor
ENTÃO Solução = 3D CNF 100%

Regra 137

SE Nota = 77
E Acorde = Sib Menor
ENTÃO Solução = 49 CNF 100%

Regra 138

SE Nota = 61

E Acorde = Sib Menor

ENTÃO Solução = 3A CNF 100%

Regra 139

SE Nota = 73

E Acorde = Sib Menor

ENTÃO Solução = 46 CNF 100%

Regra 140

SE Nota = 70

E Acorde = Sib Menor

ENTÃO Solução = 3D CNF 100%

Regra 141

SE Nota = 82

E Acorde = Sib Menor

ENTÃO Solução = 49 CNF 100%

Regra 142

SE Nota <> 82

OU Nota <> 77

OU Nota <> 73

E Nota > 72

E Acorde Anterior <> Sib Menor

E Acorde = Sib Menor

ENTÃO Solução = 46 CNF 100%

Regra 143

SE Nota <> 70

OU Nota <> 65
OU Nota <> 61
E Nota > 60
E Acorde Anterior <> Sib Menor
E Acorde = Sib Menor
ENTÃO Solução = 3A CNF 100%

Regra 144

SE Nota <> 82
OU Nota <> 77
OU Nota <> 73
OU Nota <> 70
OU Nota <> 65
OU Nota <> 61
E Acorde Anterior = Sib Menor
E Acorde = Sib Menor
ENTÃO Solução = Nota Anterior CNF 100%

Regra 145

SE Nota = 66
E Acorde = Si Menor
ENTÃO Solução = 3E CNF 100%

Regra 146

SE Nota = 78
E Acorde = Si Menor
ENTÃO Solução = 4A CNF 100%

Regra 147

SE Nota = 62

E Acorde = Si Menor

ENTÃO Solução = 3B CNF 100%

Regra 148

SE Nota = 74

E Acorde = Si Menor

ENTÃO Solução = 47 CNF 100%

Regra 149

SE Nota = 71

E Acorde = Si Menor

ENTÃO Solução = 3E CNF 100%

Regra 150

SE Nota = 83

E Acorde = Si Menor

ENTÃO Solução = 4A CNF 100%

Regra 151

SE Nota <> 83

OU Nota <> 78

OU Nota <> 74

E Nota > 73

E Acorde Anterior <> Si Menor

E Acorde = Si Menor

ENTÃO Solução = 47 CNF 100%

Regra 152

SE Nota <> 71
OU Nota <> 66
OU Nota <> 62
E Nota > 61
E Acorde Anterior <> Si Menor
E Acorde = Si Menor
ENTÃO Solução = 3B CNF 100%

Regra 153

SE Nota <> 83
OU Nota <> 78
OU Nota <> 74
OU Nota <> 71
OU Nota <> 66
OU Nota <> 62
E Acorde Anterior = Si Menor
E Acorde = Si Menor
ENTÃO Solução = Nota Anterior CNF 100%

Regra 154

SE Nota = 60
E Acorde = Ré Sétima
ENTÃO Solução = 39 CNF 100%

Regra 155

SE Nota = 72
E Acorde = Ré Sétima
ENTÃO Solução = 45 CNF 100%

Regra 156

SE Nota = 84

E Acorde = Ré Sétima

ENTÃO Solução = 51 CNF 100%

Regra 157

SE Nota = 69

OU Nota = 66

OU Nota = 62

E Acorde = Ré Sétima

ENTÃO Solução = 3C CNF 100%

Regra 158

SE Nota = 81

OU Nota = 78

OU Nota = 74

E Acorde = Ré Sétima

ENTÃO Solução = 48 CNF 100%

Regra 159

SE Nota <> 84

OU Nota <> 81

OU Nota <> 78

E Nota > 76

E Acorde Anterior <> Ré Sétima

E Acorde = Ré Sétima

ENTÃO Solução = 4A CNF 100%

Regra 160

SE Nota <> 74
OU Nota <> 72
OU Nota <> 69
OU Nota <> 66
E Nota > 64
E Acorde Anterior <> Ré Sétima
E Acorde = Ré Sétima
ENTÃO Solução = 3E CNF 100%

Regra 161

SE Nota <> 84
OU Nota <> 81
OU Nota <> 78
OU Nota <> 74
OU Nota <> 72
OU Nota <> 69
OU Nota <> 66
OU Nota <> 62
E Acorde Anterior = Ré Sétima
E Acorde = Ré Sétima
ENTÃO Solução = Nota Anterior CNF 100%

Regra 162

SE Nota = 62
E Acorde = Mi Sétima
ENTÃO Solução = 3B CNF 100%

Regra 163

SE Nota = 74

E Acorde = Mi Sétima

ENTÃO Solução = 47 CNF 100%

Regra 164

SE Nota = 71

OU Nota = 68

OU Nota = 64

E Acorde = Mi Sétima

ENTÃO Solução = 3E CNF 100%

Regra 165

SE Nota = 83

OU Nota = 80

OU Nota = 76

E Acorde = Mi Sétima

ENTÃO Solução = 4A CNF 100%

Regra 166

SE Nota <> 83

OU Nota <> 80

E Nota > 78

E Acorde Anterior <> Mi Sétima

E Acorde = Mi Sétima

ENTÃO Solução = 4C CNF 100%

Regra 167

SE Nota <> 76

OU Nota <> 74

OU Nota <> 71

OU Nota <> 68
E Nota > 66
E Acorde Anterior <> Mi Sétima
E Acorde = Mi Sétima
ENTÃO Solução = 40 CNF 100%

Regra 168

SE Nota <> 83
OU Nota <> 80
OU Nota <> 76
OU Nota <> 74
OU Nota <> 71
OU Nota <> 68
OU Nota <> 64
OU Nota <> 62
E Acorde Anterior = Mi Sétima
E Acorde = Mi Sétima
ENTÃO Solução = Nota Anterior CNF 100%

Regra 169

SE Nota = 65
E Acorde = Sol Sétima
ENTÃO Solução = 3E CNF 100%

Regra 170

SE Nota = 77
E Acorde = Sol Sétima
ENTÃO Solução = 4A CNF 100%

Regra 171

SE Nota = 62

E Acorde = Sol Sétima

ENTÃO Solução = 3B CNF 100%

Regra 172

SE Nota = 74

OU Nota = 71

OU Nota = 67

E Acorde = Sol Sétima

ENTÃO Solução = 41 CNF 100%

Regra 173

SE Nota = 83

OU Nota = 79

E Acorde = Sol Sétima

ENTÃO Solução = 4D CNF 100%

Regra 174

SE Nota <> 83

E Nota > 81

E Acorde Anterior <> Sol Sétima

E Acorde = Sol Sétima

ENTÃO Solução = 4F CNF 100%

Regra 175

SE Nota <> 79

OU Nota <> 77

OU Nota <> 74

OU Nota <> 71
E Nota > 69
E Acorde Anterior <> Sol Sétima
E Acorde = Sol Sétima
ENTÃO Solução = 43 CNF 100%

Regra 176

SE Nota <> 67
OU Nota <> 65
OU Nota <> 62
E Nota > 61
E Acorde Anterior <> Sol Sétima
E Acorde = Sol Sétima
ENTÃO Solução = 3B CNF 100%

Regra 177

SE Nota <> 83
OU Nota <> 79
OU Nota <> 77
OU Nota <> 74
OU Nota <> 71
OU Nota <> 67
OU Nota <> 65
OU Nota <> 62
E Acorde Anterior = Sol Sétima
E Acorde = Sol Sétima
ENTÃO Solução = Nota Anterior CNF 100%

Regra 178

SE Nota = 67

E Acorde = Lá Sétima

ENTÃO Solução = 40 CNF 100%

Regra 179

SE Nota = 79

E Acorde = Lá Sétima

ENTÃO Solução = 4C CNF 100%

Regra 180

SE Nota = 64

E Acorde = Lá Sétima

ENTÃO Solução = 37 CNF 100%

Regra 181

SE Nota = 76

OU Nota = 73

OU Nota = 69

E Acorde = Lá Sétima

ENTÃO Solução = 43 CNF 100%

Regra 182

SE Nota = 81

E Acorde = Lá Sétima

ENTÃO Solução = 4F CNF 100%

Regra 183

SE Nota <> 81

OU Nota <> 79

OU Nota <> 76
OU Nota <> 73
E Nota > 71
E Acorde Anterior <> Lá Sétima
E Acorde = Lá Sétima
ENTÃO Solução = 45 CNF 100%

Regra 184

SE Nota <> 69
OU Nota <> 67
OU Nota <> 64
OU Nota <> 61
E Nota > 60
E Acorde Anterior <> Lá Sétima
E Acorde = Lá Sétima
ENTÃO Solução = 39 CNF 100%

Regra 185

SE Nota <> 81
OU Nota <> 79
OU Nota <> 76
OU Nota <> 73
OU Nota <> 69
OU Nota <> 67
OU Nota <> 64
OU Nota <> 61
E Acorde Anterior = Lá Sétima
E Acorde = Lá Sétima
ENTÃO Solução = Nota Anterior CNF 100%

Regra 186

SE Nota = 69

E Acorde = Si Sétima

ENTÃO Solução = 42 CNF 100%

Regra 187

SE Nota = 81

E Acorde = Si Sétima

ENTÃO Solução = 4E CNF 100%

Regra 188

SE Nota = 66

OU Nota = 63

E Acorde = Si Sétima

ENTÃO Solução = 39 CNF 100%

Regra 189

SE Nota = 78

OU Nota = 75

OU Nota = 71

E Acorde = Si Sétima

ENTÃO Solução = 45 CNF 100%

Regra 190

SE Nota = 83

E Acorde = Si Sétima

ENTÃO Solução = 51 CNF 100%

Regra 191

SE Nota <> 83
OU Nota <> 81
OU Nota <> 78
OU Nota <> 75
E Nota > 73
E Acorde Anterior <> Si Sétima
E Acorde = Si Sétima
ENTÃO Solução = 47 CNF 100%

Regra 192

SE Nota <> 71
OU Nota <> 69
OU Nota <> 66
OU Nota <> 63
E Nota > 61
E Acorde Anterior <> Si Sétima
E Acorde = Si Sétima
ENTÃO Solução = 3B CNF 100%

Regra 193

SE Nota <> 83
OU Nota <> 81
OU Nota <> 78
OU Nota <> 75
OU Nota <> 71
OU Nota <> 69
OU Nota <> 66
OU Nota <> 63

E Acorde Anterior = Si Sétima

E Acorde = Si Sétima

ENTÃO Solução = Nota Anterior CNF 100%

REFERÊNCIAS BIBLIOGRÁFICAS

- [ARC1986] ARCHANJO, Samuel. **Lições elementares de teoria musical**. São Paulo: Ricordi, 1986.
- [CAM2000] CAMPOS, Antônio Carlos Neves. **Teoria elementar da música**. Correio Eletrônico <http://www.geocities.com/cdmcct/teoria/index.html>. Data da consulta: 12/11/2000.
- [CAN1998] CANTU, Marco. **Dominando o Delphi 3 – A Bíblia**. São Paulo: Makron Books, 1998.
- [FAV2000] FÁVERO, Alexandre José. **Sistemas especialistas**. Endereço Eletrônico <http://www.din.uem.br/ia/especialistas/>. Data da consulta: 05/11/2000.
- [FEI1983] FEIGENBAUM, Edward A. e McCORDUCK, P. **The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World**. Addison-Wesley Publishing Company, 1983.
- [FER1995] FERREIRA, Josemar Dias. **Multimídia para programadores e analistas**. Rio de Janeiro: Infobook, 1995.
- [FRI1994] FRITSCH, Eloi Fernando. **CAMM – Compositor automático de melodias musicais**. Porto Alegre, 1994. Tese (Pós-Graduação em Ciências da Computação) Instituto de Informática, Universidade Federal do Rio Grande do Sul.
- [GON2000] GONTIJO, Christian Haagensen. **Sítio Web sobre Midi**. Endereço Eletrônico: <http://www.geocities.com/SiliconValley/Campus/7086/m.htm>. Data da consulta: 04/10/2000.
- [HEI1995] HEINZLE, Roberto. **Protótipo de uma ferramenta para criação de sistemas especialistas baseados em regras de produção**. Florianópolis: UFSC,

1995. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção e Sistemas.

- [JAC1956] **Enciclopédia prática Jackson.** São Paulo: Jackson Editores, 1956.
- [LAR1977] **Enciclopédia Delta-Larousse.** São Paulo: Larousse, 1977.
- [LEV1988] LEVINE, Robert I.; DRANG, Diane E.; EDELSON, Barry. **Inteligência artificial e sistemas especialistas.** São Paulo: McGraw-Hill, 1988.
- [LIA2000] LIA. **Sistemas inteligentes aplicados.** Endereço Eletrônico <http://www.lia.ufc.br/~bezerra/exsinta/index.html>. Data da consulta: 05/10/2000.
- [MAT2000] MATTOS, Fernando Lewis. **Técnicas de composição musical.** Endereço Eletrônico <http://www.portoweb.com.br/compor>. Data da consulta: 05/11/2000.
- [OLI2000] OLIVEIRA, Reinaldo G.. **Teoria musical.** Endereço Eletrônico <http://www.geocities.com/gaitifera/teoria.html>. Data da consulta: 17/11/2000.
- [PRI1994] PRIOLLI, Maria Luisa de Mattos. **Princípios básicos da música para a juventude.** Rio de Janeiro: Casa Oliveira de Músicas Ltda, 1994.
- [RAB1995] RABUSKE, Renato Antônio. **Inteligência artificial.** Florianópolis: Editora da UFSC, 1995.
- [RAT1995] RATTON, Miguel. **Criação de música e sons por computador.** Rio de Janeiro: Campus, 1995.
- [REG2000] REGAUER, Charles. **Teoria da música.** Endereço eletrônico <http://www.geocities.com/crodilo/teoria.htm>. Data da consulta: 01/11/2000.
- [RIB1987] RIBEIRO, Horácio da Cunha e Souza. **Introdução aos sistemas especialistas.** Rio de Janeiro – LTC : Livros Técnicos e Científicos Editora, 1987.