

Dissertação de Mestrado
Aluno: Giordano Ribeiro Eulálio Cabral
Orientador: Geber Lisboa Ramalho

D'Accord Guitar
Um Sistema Para Execução Violonística

Recife
Junho 2002

Dedicatória

Dedico este trabalho a minha família.

A família Ribeiro,

a família Cabral,

a família CIn

e a família D'Accord.

Agradecimentos

Agradeço a todos que contribuíram para que eu desenvolvesse este trabalho. A Geber, pelo carinho, incentivo e confiança. A Malu, sem a qual eu não teria começado. A todos os meus amigos, pelo companheirismo e força que nunca faltaram. Em especial, a meus amigos da D'Accord, sem os quais este projeto não teria chegado aonde chegou. A Bel, por me ajudar a crescer em todos os sentidos. E à minha família, por tudo o que ela representa para mim.

Resumo

Existe uma grande lacuna de softwares para o aprendizado de peças musicais que sejam apropriados para notações musicais simples, como as músicas cifradas, principalmente no caso do violão e da guitarra. Visando superar estas limitações, foi desenvolvido o *D'Accord Guitar*, cujo princípio básico é apresentar a informação da maneira mais próxima possível ao processo cognitivo do usuário, facilitando o aprendizado de acompanhamentos musicais no violão e na guitarra.

O desenvolvimento de tal software trouxe consigo algumas dificuldades. Além de ser um sistema em tempo real, que exige uma execução de grande precisão e a utilização de tecnologia multimídia específica, foi preciso criar interfaces gráficas que aprimorassem o processo de aprendizagem e de edição, bem como definir um formato que descrevesse de forma mais completa a informação musical. Em particular, destacamos a modelagem do raciocínio envolvido no processo de interpretação de um conjunto de cifras de acordes por um músico. Este problema é extremamente complexo devido à sua ambigüidade e sua subjetividade, e ainda não foi tratado pela comunidade acadêmica.

O *D'Accord Guitar* provê um conjunto de ferramentas úteis para diversos tipos de violonistas e guitarristas em diferentes níveis. Ele vem sendo utilizado por mais de 100.000 usuários, obtendo um retorno bastante positivo.

Abstract

There is a great lack of software to learn musical pieces based on simple musical notations, such as ciphered songs, specially when applied to guitar studies. We developed *D'Accord Guitar* in order to achieve this goal. The fundamental principle in *D'Accord Guitar* is to present the information as close as possible to the guitarist learning process, making the assimilation of guitar accompaniments easier.

Such a development brought some difficulties: high precision involved in implementing real time features, use of specific multimedia technology, creation of a practical graphical interface, and definition of a file format with a more complete musical description.

Particularly, we highlight the modeling of the rational process involved in interpreting ciphered songs by a musician. This problem is extremely complex due to its ambiguity and subjectivity, and has not been related yet in the scientific academic literature.

D'Accord Guitar provides a set of useful tools, appropriate to different levels of guitarists. *D'Accord Guitar* has been used for more than 100.000 users, obtaining very positive responses.

Índice

Dedicatória.....	2
Agradecimentos	3
Resumo	4
Abstract.....	5
Índice.....	6
Índice de Figuras	8
1. Introdução	12
1.1. Sistemas Automáticos	15
1.2. Objetivo.....	16
1.3. D'Accord Guitar	17
2. Entre a cifra e a partitura: uma solução por computador?	19
2.1. Métodos e ferramentas convencionais para o aprendizado de peças musicais	19
2.1.1. Partituras.....	19
2.1.2. Tablaturas	21
2.1.3. Músicas Cifradas.....	21
2.1.4. Cifras X Partituras.....	23
2.1.5. Sistemas Combinados	25
2.1.6. Imitação	27
2.2. Métodos e ferramentas computacionais para o aprendizado de peças musicais	28
2.3. Dificuldades encontradas	29
2.3.1. Interpretação de Cifras	30
O Processo decisório	36
2.4. Conclusão	19
3. Sistemas Computacionais para o Aprendizado de Peças Musicais.....	38
3.1. Sistemas de Simulação Instrumental	38
3.2. Sistemas de Acompanhamento Automático(SAA)	40
3.3. Editores de Tablaturas.....	43
3.4. Dicionários Interativos de Acordes	44
3.5. Sistemas de Karaokê	44
3.6. Conclusão	45
4. <i>D'Accord Guitar</i>	47
4.1. Princípio	48
4.1.1. Separação entre acompanhamento e melodia.....	48
4.1.2. Representação direta	51
4.2. Modos de interação.....	51
5. <i>D'Accord Guitar Player</i>	53
5.1. Interface.....	53
5.1.1. Instrumento Virtual	53
5.1.2. Música cifrada.....	54
5.1.3. Acordes.....	55
5.2. Funcionalidades.....	55
5.2.1. Manipulação de arquivos.....	55
5.2.2. Controle de execução	56
5.2.3. Funções inteligentes	57
5.2.4. Configuração.....	59
5.3. Arquitetura	61
5.4. Concepção e implementação	62

5.4.1.	Representação musical	62
5.4.2.	Sequenciador.....	66
5.4.3.	Funções Inteligentes.....	68
5.5.	Discussão.....	72
6.	<i>D'Accord Guitar Editor</i>	75
6.1.	Interface e funcionalidades.....	75
6.2.	Arquitetura	79
6.3.	Concepção e implementação	81
6.3.1.	Interface de Gravação.....	81
6.3.2.	Playback	91
6.3.3.	Reedição	93
6.3.4.	Parsers	93
6.4.	Discussão.....	98
7.	<i>D'Accord Guitar Browser</i>	99
7.1.	Interface e funcionalidades.....	99
7.1.1.	Braço Interativo	100
7.1.2.	Acesso à base de acordes.....	101
7.2.	Arquitetura	101
7.3.	Concepção e implementação	102
7.3.1.	Criação da Base de Acordes	102
7.3.2.	Reconhecimento de acordes, notas e intervalos	103
7.3.3.	Digitação.....	105
7.4.	Discussão.....	106
8.	Interpretação de Cifras.....	107
8.1.	Estado da Arte	107
8.2.	Metodologia utilizada	108
8.3.	Etapa 1: construção de acordes.....	109
8.3.1.	Posicionamento.....	110
8.3.2.	Digitação.....	114
8.4.	Etapa 2: critérios de avaliação de posicionamento e digitação	115
8.4.1.	Valoração.....	117
8.4.2.	Quadro Comparativo	124
8.5.	Etapa 3: escolha multi-critérios	125
8.5.1.	Flexibilidade	128
8.5.2.	Facilidade do posicionamento	132
8.5.3.	Frequência de uso e sonoridade	135
8.5.4.	Facilidade de transição.....	135
8.5.5.	Linha de baixo e melódica.....	137
8.6.	Escolha final de posições e digitações	137
8.6.1.	Algoritmo de busca	138
8.6.2.	Gatilhos	140
8.6.3.	Perfis de usuários	141
8.6.4.	Modelo Final.....	142
8.7.	Estudos de caso.....	142
9.	Conclusões	146
9.1.	Resultados	146
9.2.	Contribuições.....	147
9.3.	Perspectivas	148
10.	Bibliografia	150
	Anexo 1 - Representação Utilizada.....	156

Índice de Figuras

Figura 1.1 – Exemplo de uma partitura da música “Garota de Ipanema”.....	13
Figura 1.2 – Exemplo da música “Garota de Ipanema” cifrada.	14
Figura 1.3 – Exemplo de um SSI (Desktop Guitarist).	16
Figura 2.1 – Exemplo de uma partitura.	20
Figura 2.2 – Uma tablatura e uma partitura da mesma música	21
Figura 2.3 – Exemplo de um trecho de uma música cifrada, com diagramas das posições dos acordes em diferentes instrumentos	22
Figura 2.4 – Diferentes modelos de cifragem.....	23
Figura 2.5 – comparação entre a partitura e uma cifra de um trecho da música “Garota de Ipanema”	24
Figura 2.6 – exemplos de combinações de notações para obter um sistema mais completo ...	26
Figura 2.7 – aprendizado por imitação.....	27
Figura 2.8 – Unindo-se os diversos recursos computacionais é possível desenvolver um método de leitura, escrita e aprendizado mais eficiente.	29
Figura 2.9 – Caso ideal: partindo de um conjunto de cifras, gerar uma execução real.....	30
Figura 2.10 – Indicação de tempo em uma cifra.....	30
Figura 2.11 – Indicação de tempo em grades de acordes.	31
Figura 2.12 – diferentes escolhas de vozes para um C7M no piano e em partitura.....	32
Figura 2.13 – três posições consonantes em uma representação do braço do violão	32
Figura 2.14 – diferentes escolhas de posições	33
Figura 2.15 – Diferentes coordenadas que podem ser utilizadas para montar um Fm7(11). ...	33
Figura 2.16 – Possíveis posições e digitações de um Fm7(11)	34
Figura 2.17 – exemplos de posições do acorde Am no violão e sua respectiva escolha de vozes	35
Figura 2.18 – Possíveis fingerings de uma posição de Dm7	36
Figura 3.1 – Braço de violão virtual.....	38
Figura 3.2 – Tela principal do SSI “The Guitarist”	39
Figura 3.3 – Arpejo de um Am7 em um SSI convencional.....	40
Figura 3.4 – Telas dos softwares Band-In-A-Box e Harmony Assistant	41
Figura 3.5 – Edição da grade de acordes no Band-In-A-Box.....	42
Figura 3.6 – Associação entre posições de acordes e estilos musicais	42
Figura 3.7 – Braço de violão representado a partir da versão 10 do Band-In-A-Box	43
Figura 3.8 – Janela principal do editor de tablaturas Guitar Pro	43
Figura 3.9 – Janela principal do software Chord Wizard.....	44
Figura 3.10 – Janela principal do software MIDIMaster Karaoke.	45
Figura 4.1 – Arpejo de um Am7 e de um D7 em um SSI convencional.....	49
Figura 4.2 – Arpejo de um Am7 e de um D7 no D’ Accord Guitar.	49
Figura 4.3 – Edição musical baseada em harmonia, ritmo e melodia.....	50
Figura 4.4 – Representação de uma posição do acorde Am7.....	51
Figura 4.5 – Relacionamento entre os modos de interação e os conjuntos de funcionalidades disponíveis nas classes básicas.....	52
Figura 5.1 – Interface do D’ Accord Guitar Player.....	53
Figura 5.2 – Braço do violão virtual	54
Figura 5.3 – Letra e cifras animadas	54
Figura 5.4 – Indicação do próximo acorde	55
Figura 5.5 – Exemplo de música a ser impressa.....	55
Figura 5.6 - Exemplo de uma música impressa.	56

Figura 5.7 – Controles de execução	56
Figura 5.8 – Linha de tempo e outros indicadores	56
Figura 5.9 – Barra de controle da velocidade	57
Figura 5.10 - Parâmetros para a escolha de acordes	58
Figura 5.11 - Configuração de restrições sobre os acordes a serem gerados	58
Figura 5.12 - Função de transposição.....	59
Figura 5.13 – Trecho de uma melodia, convertida em solo.	59
Figura 5.14 – Configuração do volume e instrumento relativos a cada canal.....	60
Figura 5.15 – Escolha do driver MIDI	60
Figura 5.16 – As 4 orientações possíveis para exibir o braço do violão	61
Figura 5.17 – Arquitetura do D' Accord Guitar Player.....	61
Figura 5.18 – Reprodução de uma melodia a partir de um arquivo MIDI	63
Figura 5.19 – Hierarquia das classes relativas a eventos em DV3	64
Figura 5.20 – Conjunto de eventos de diversos tipos com indicação de tempo em DV3	65
Figura 5.21 – Acompanhamento e Melodia	65
Figura 5.22 – Aplicação de um ritmo a um conjunto de acordes, gerando um acompanhamento.....	66
Figura 5.23 – Comparação das unidades de medida de tempo.....	67
Figura 5.24 – Transposição de uma melodia MIDI	68
Figura 5.25 – Efeito da transposição de texto.....	69
Figura 5.26 – Deslocamento das seleções de letra, efetuado na transposição.....	70
Figura 5.27 – Processo de transposição de solos	71
Figura 5.28 – Interpretação de cifras para o violão	72
Figura 5.29 – Exibição da execução em um SSI convencional (The Jazz Guitarist, abaixo) e no D' Accord Guitar Player (no alto).	73
Figura 5.30 – Exibição da execução violonística em um SSI convencional (The Jazz Guitarist, à esquerda) e no D' Accord Guitar Player (à direita).....	74
Figura 6.1 – Janela inicial e de gravação do D' Accord Guitar Editor	75
Figura 6.2 – Tela de edição da música cifrada	76
Figura 6.3 – Tela de edição da melodia.....	76
Figura 6.4 – Tela de edição da harmonia	77
Figura 6.5 – Tela de gravação do ritmo.....	78
Figura 6.6 – Tela de edição da letra (karaokê)	79
Figura 6.7 - Tela de edição das informações sobre a música	79
Figura 6.8 – Arquitetura do D' Accord Guitar Editor.....	80
Figura 6.9 – Processo geral de gravação	80
Figura 6.10 - Grade de compassos e os eventos que ocorrem em cada um deles.....	82
Figura 6.11 - Associação entre a melodia e as seleções de letra utilizadas para o karaokê	82
Figura 6.12 – Processo de gravação da melodia.....	83
Figura 6.13 – Associação entre letra e melodia	83
Figura 6.14 – Processo de gravação da animação da letra	84
Figura 6.15 – Processo de separação de letra	84
Figura 6.16 – Associação da letra com a melodia	85
Figura 6.17 – Gravação em tempo real	85
Figura 6.18 – Processo de gravação da harmonia.....	86
Figura 6.19 – Sincronização da harmonia através da grade de acordes	86
Figura 6.20 – Sincronização da harmonia pela gravação em tempo real	87
Figura 6.21 – Aplicação de vários ritmos dentro de uma música.	87
Figura 6.22 – Captação da informação rítmica.....	88
Figura 6.23 – Processo de gravação de trocas de ritmos.....	89

Figura 6.24 – Exemplo de trecho onde não há formação clara de acordes	89
Figura 6.25 – Processo de gravação de solos	90
Figura 6.26 – Processo de gravação de eventos de controle	91
Figura 6.27 – Conversão MIDI → DV3.....	91
Figura 6.28 – Punch In, Punch Out, Overdubbing.....	93
Figura 6.29 – Parsers necessários	94
Figura 6.30 – Parser para separar letra e cifra	95
Figura 6.31 – Música cifrada de 5 maneiras diferentes.....	95
Figura 6.32 – BNF do formato de músicas cifradas do D' Accord Guitar.....	96
Figura 6.33 – BNF do formato de cifras do D' Accord Guitar.....	97
Figura 6.34 – Exemplo de letra a ser animada.....	97
Figura 7.1 – Tela principal do D' Accord Guitar Browser.....	100
Figura 7.2 – Informações sobre uma posição de D7	101
Figura 7.3 – Diversas posições e digitações para um Am7	101
Figura 7.4 – Arquitetura do D' Accord Guitar Browser	102
Figura 7.5 – Processo de reconhecimento dos dados relacionados ao acorde.....	103
Figura 7.6 – Posição a ser reconhecida	105
Figura 8.1 – Duas posições possíveis para um Am7 associadas, respectivamente, aos estilos bossa-nova e rock	107
Figura 8.2 – Processo geral de interpretação	109
Figura 8.3 – Construção de acordes	109
Figura 8.4 - Posicionamento	110
Figura 8.5 – Reconhecimento de acordes.....	110
Figura 8.6 – Coordenadas que podem ser utilizadas para formar o acorde.....	111
Figura 8.7 – Algumas posições encontradas para um Fm7(11).....	111
Figura 8.8 – Algumas posições de acordes onde mais de 4 cordas são pressionadas.....	112
Figura 8.9 - Figura com uma armadura da mão direita desaconselhável para tocar batidas de Rock.....	113
Figura 8.10 - Interface de configuração de restrições	113
Figura 8.11 - Exemplos de posições de acorde passíveis de sofrer restrições.....	114
Figura 8.12 – Digitação para uma posição de Db7	114
Figura 8.13 – Possíveis digitações de uma posição de Dm7	115
Figura 8.14 – Processo decisório para encontrar a melhor sequência de posições para uma sequência de cifras.....	117
Figura 8.15 – Duas posições. O Gm é considerado mais fácil por encontrar-se numa região de menor tensão	118
Figura 8.16 – Dois exemplos de posições de um Am7	118
Figura 8.17 – Comparação de duas posições quanto à flexibilidade	119
Figura 8.18 – Mesmo exemplo da Figura 8.17, com a identificação das coordenadas alcançáveis.	119
Figura 8.19 – Um possível digitação para uma posição de Em7	121
Figura 8.20 – Comparação de duas posições quanto à similaridade com uma primeira.....	122
Figura 8.21 – Comparação de 2 posições quanto à facilidade de transição com a primeira..	122
Figura 8.22 – Exemplo de duas posições que possuem uma coordenada como ponte	123
Figura 8.23 – Exemplo de duas posições que mantêm as cordas pressionadas por cada dedo	123
Figura 8.24 – Comparação entre 4 posições quanto à similaridade com a primeira.....	124
Figura 8.25 - Intervalos dos valores possíveis para cada critério	126
Figura 8.26 – Equiparação dos níveis de classificação para os diversos critérios	126

Figura 8.27 – Idéia central do aprendizado de máquina supervisionado: utilizar exemplos avaliados por um ser humano para avaliar novos exemplos.....	127
Figura 8.28 – Possíveis limiares entre os níveis de classificação.....	128
Figura 8.29 – Classificação manual	128
Figura 8.30 – Histograma de ocorrências de um conjunto de dados e classificação baseada na divisão em 5 intervalos iguais	129
Figura 8.31 – Classificação baseada na divisão em 5 intervalos igualmente povoados	130
Figura 8.32 – Histograma com o número de ocorrências em cada nível	130
Figura 8.33 – Classificação baseada na divisão em 5 grupos povoados de acordo com sua probabilidade.....	131
Figura 8.34 – Classificação baseada na minimização do erro	131
Figura 8.35 – Uma possível base de exemplos.....	133
Figura 8.36 – Matriz com a base de exemplos gerada	134
Figura 8.37 – Exemplo de classificação da similaridade entre algumas mudanças de posições	136
Figura 8.38 - Exemplo de classificação (quanto à linha de baixo) entre algumas posições ..	137
Figura 8.39 – Algoritmo de busca utilizado	138
Figura 8.40 – Exemplo de árvore de busca gerada	139
Figura 8.41 - Posições e digitações escolhidas para uma determinada cifra.....	140
Figura 8.42 – Exemplos de gatilhos.....	141
Figura 8.43 - Exemplos de perfis e os pesos associados a cada critérios.....	142
Figura 8.44 – Modelo implementado	142
Figura 8.45 – Posições e digitações mais comuns para a sequência de acordes dada	143
Figura 8.46 – Posições de acordes seguindo a melhor linha de baixo para a sequência de acordes dada.	143
Figura 8.47 - Posições e digitações encontradas levando em consideração todos os critérios	143
Figura 8.48 - Posições e digitações encontradas levando em consideração a facilidade de posicionamento, de transição e a flexibilidade	144
Figura 8.49 - Posições e digitações encontradas de acordo com o perfil solista	144
Figura 8.50 - Posições e digitações levando em consideração flexibilidade e linha melódica, relaxando restrições de inversões e de número de repetições permitidas.....	144
Figura A.51 - Braço do violão na horizontal e na vertical.	156
Figura A.52 - Diagrama e sua relação com o violão.....	156

1. Introdução

Existe hoje em dia um grande número de pessoas interessadas em uma maneira simples de aprender a tocar um instrumento, ou de aumentar o seu repertório. Pessoas interessadas principalmente em música popular, que desejam acompanhar-se enquanto cantam ou enquanto outras pessoas cantam, e que não estão completamente satisfeitos com as possibilidades de aprendizado existentes.

A maneira mais antiga e aparentemente mais direta de aprendizado de peças musicais é através da imitação (Dowling 1989). Esta abordagem é muito comum entre os iniciantes, enquanto ainda não aprenderam a ler música em nenhuma notação. O professor mostra como uma música é tocada, e o aluno tenta reproduzir a sua execução. Embora seja geralmente considerada a maneira mais eficiente, ressaltamos aqui dois problemas. O primeiro é a necessidade da presença constante do professor, e o segundo é que o aluno reproduz a execução sem saber exatamente o que está tocando. Se o aluno soubesse quais padrões harmônicos e rítmicos está tocando, aprenderia mais rapidamente novas músicas.

Em função disto, a aprendizagem musical conta normalmente com as notações musicais, que diminuem a necessidade da presença de um professor, configurando-se como a linguagem onde é descrita a teoria musical e que serve de base para todo o processo de leitura e de escrita musical. Cada notação procura encontrar o melhor compromisso entre simplicidade e completude, entre riqueza de detalhes e facilidade de leitura, entre ser genérica ou específica para um instrumento. Quanto mais rica a notação, mais precisa ela pode ser, porém torna-se também menos legível. Por outro lado, quanto mais simplificada a notação, a sua leitura exige um maior nível de interpretação. Esta balança entre simplicidade e riqueza de detalhes pode ser bem ilustrada com as duas notações mais utilizadas atualmente: as partituras (Michels 1988) e as músicas cifradas (Sher 1991).

As partituras são consideradas bastante precisas e, conseqüentemente, pouco flexíveis, por representarem cada nota a ser tocada. A sua leitura é uma atividade não trivial, em função do grande volume de informação a tratar. Boa parte da educação musical tradicional trata apenas de como ler uma partitura, o que freqüentemente desestimula os alunos. Além disso, por não tratarem peculiaridades de cada instrumento, precisam ser transcritas, o que é especialmente problemático no caso dos instrumentos de corda. Na Figura 1.1 é exibida a partitura do acompanhamento de violão da música *Garota de Ipanema*.

Garota de Ipanema

Tom Jobim
Vinícius de Moraes

doce ♩ = 60

Arr. Aílton Martins da Cunha

Figura 1.1 –Exemplo de uma partitura da música “Garota de Ipanema”.

Já as músicas cifradas são mais adequadas ao público citado no primeiro parágrafo. Elas suprimiram parcialmente as carências deste público, por apresentarem uma notação extremamente simplificada, voltada para o acompanhamento de música popular, e tornaram-se um grande sucesso de mercado. Hoje em dia é extremamente comum encontrar revistas e livros de músicas cifradas, assim como diversos *sites* especializados, na *Internet*, servindo como repositórios de tais músicas.

Nas cifras, é suposto que o músico já conheça a melodia, a letra e o ritmo da música. Desta forma ela representa apenas o componente harmônico (os acordes), através de uma associação destes com a letra da música. Isto pode ser visto na Figura 1.2. Para ilustrar o grau de simplificação alcançado, ela mostra a cifra da mesma música exibida na Figura 1.1. Como necessitam de um volume menor de informação, são mais simples de ler e escrever.

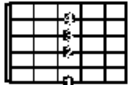
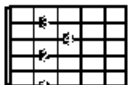
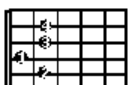
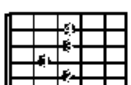

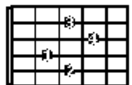
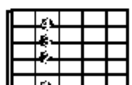
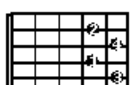
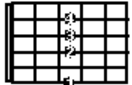
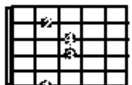
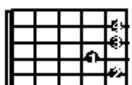

<p>F7M G7(13) <u>Olha</u> que coisa mais linda, mais cheia de <u>graça</u> Gm7 É ela menina que vem e que <u>passa</u> Gb7(#11) Am7 Ab7 Db7M(9) Gb7(#11) Num doce <u>balanço</u>, caminho do <u>mar</u> _ _ _</p> <p>F7M G7/13 <u>Moça</u> do corpo dourado, do sol de Ipanema Gm7 O seu <u>balançado</u> é mais que um <u>poema</u> Gb7(#11) F7M É a coisa mais <u>linda</u> que eu já vi <u>passar</u></p> <p>F#7M B7(9) <u>Ah</u>, por que estou tão <u>sozinho</u>? F#m7 D7(9) <u>Ah</u>, por que tudo é tão <u>triste</u>? Gm7 Eb7(9) <u>Ah</u>, a beleza que <u>existe</u></p> <p>Am7 D7(b9) A beleza que não é só <u>minha</u> Gm7 C7(b9) <u>Que</u> também <u>passa</u> <u>sozinha</u></p> <p>F7M G7(13) <u>Ah</u>, se ela soubesse que quando ela <u>passa</u> Gm7 O mundo sorrindo se enche de <u>graça</u> Gb7(#11) F7M E fica mais <u>lindo</u> por causa do <u>amor</u> (Gb7(#11) F7M } x _ _ _ Por causa do <u>amor</u></p>	<p>Gm7</p>  <p>Ab7</p>  <p>B7(9)</p>  <p>Eb7(9)</p>  <p>↑6ª casa</p>	<p>Gb7(#11)</p>  <p>Db7M(9)</p>  <p>F#m7</p>  <p>D7(b9)</p> 	<p>Am7</p>  <p>↑5ª casa</p> <p>F#7M</p>  <p>D7(9)</p>  <p>C7(b9)</p> 
---	--	--	---

Figura 1.2 – Exemplo da música “Garota de Ipanema” cifrada.

Além disso, embora elas também sejam genéricas, elas normalmente trazem desenhos indicando a posição de cada acorde no instrumento desejado, como também pode

ser visto na Figura 1.2. Na grande maioria dos casos, este instrumento é o violão (ou a guitarra), em função da grande popularidade de que desfruta.

Porém, as músicas cifradas possuem limitações. Em primeiro lugar, são incompletas, não possibilitando a indicação do ritmo, do instante em que cada evento ocorre, ou de arranjos que o músico possa realizar (e.g., solos). Em segundo lugar, não são padronizadas. Cada um pode utilizar uma sintaxe diferente para escrevê-las. Em terceiro lugar, são ambíguas. Cada acorde pode ser posicionado de diversas maneiras, a critério do músico. O músico precisa, então, interpretar cada cifra à sua maneira, decidindo o ritmo a aplicar e as posições a colocar para cada acorde, por exemplo. Portanto, para que a cifra seja bem tocada, é preciso que o músico tenha conhecimento e experiência.

Por estas razões, ainda há uma grande insatisfação com recursos disponíveis hoje em dia para auxiliar o aprendizado de peças musicais. As músicas cifradas são bastante utilizadas, mas em grande parte devido ao fato da outra alternativa (as partituras) ser muito complexa, e por possuírem um maior acervo de músicas do gênero popular.

O objetivo deste trabalho foi satisfazer de forma mais completa este público, desenvolvendo um sistema capaz de unir a precisão das partituras, a simplicidade das cifras e a eficiência e objetividade da imitação.

1.1. Sistemas Automáticos

Vários esforços foram realizados no sentido de utilizar os recursos de multimídia dos computadores para aprimorar os processos de escrita, leitura e aprendizado musicais (Roads 1996). Porém, ainda há carência de uma solução adequada ao público em questão. A seguir serão descritas brevemente as soluções existentes atualmente e as limitações que apresentam.

Um destes esforços são os chamados *Sistemas de Simulação Instrumental* (SSI), que mostram diretamente a música sendo tocada em um instrumento virtual (Beck 1996), exibida na tela do computador (Figura 1.3). Desta forma, o processo de leitura é bastante facilitado. As principais limitações desses sistemas são a falta de um ambiente apropriado para edição musical, a impossibilidade de interação com o instrumento virtual, a falta de preocupação didática, e a inadequação para quem baseia sua leitura musical na notação de músicas cifradas. Esta inadequação é verificada principalmente nos SSIs específicos para violão/guitarra.

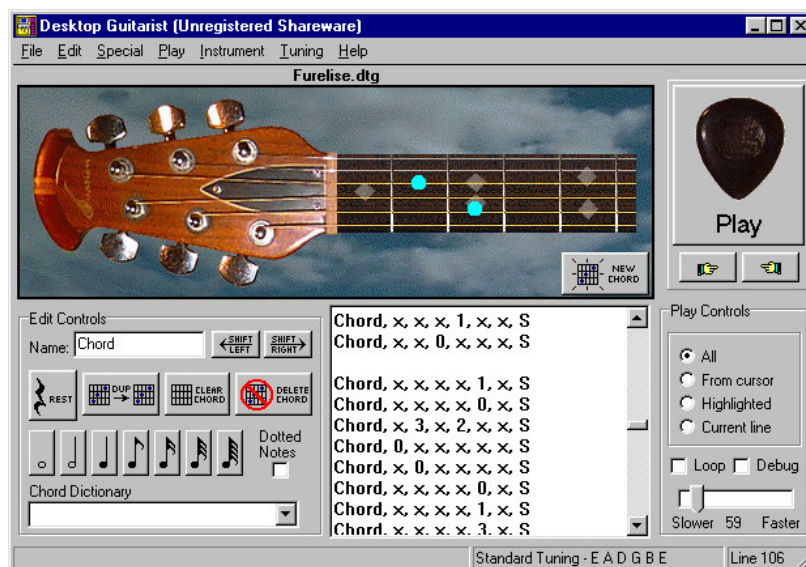


Figura 1.3 – Exemplo de um SSI (Desktop Guitarist).

Há ainda diversas ferramentas para escrita musical, como editores e seqüenciadores. Com tais sistemas é possível gravar diretamente a execução através de um instrumento musical MIDI (Hill 1994), ou escrever a música utilizando alguma notação. Porém, alguns problemas são encontrados. Na maioria deles, a música é apenas tocada, mas não é exibida em um instrumento virtual. Além disso, embora haja um grande público interessado, pouquíssimos desses sistemas são baseados em cifras. Uma exceção são os chamados *Sistemas de Acompanhamento Automático*. Entretanto, o objetivo destes não é educacional e suas soluções para instrumentos de corda harmônicos, como o violão, é muito precária.

Em resumo, além de falhas pontuais, existem problemas gerais nas soluções atuais. São eles: a falta de integração dos processos de escrita, leitura e aprendizado; a falta de adequação a instrumentos de corda harmônicos (e.g., violão e guitarra), por não tratarem das particularidades destes instrumentos (Cabral 2001a); e a inadequação ao público usuário de músicas cifradas.

1.2. Objetivo

O objetivo deste trabalho foi prover um software que sirva de suporte para o aprendizado de músicas de uma maneira que reúna precisão, simplicidade e eficiência. Para isto, foram aproveitados alguns conceitos presentes nos diversos sistemas existentes, e acrescentados outros conceitos necessários. A idéia central é frisar o acompanhamento musical, como nas cifras, e exibir a execução de forma direta em um instrumento virtual (simulando o processo de imitação), tudo isto seqüenciado e animado de maneira precisa.

Para conseguir uma solução onde o aprendizado seja eficiente, a idéia é incorporar o conceito de um instrumento virtual, como nos SSIs. O instrumento musical escolhido foi o violão (e, conseqüentemente, a guitarra) por dois motivos. Em primeiro lugar, em função da sua popularidade, o que pode ser comprovado nas publicações, como revistas e livros (Chediak 1999), de músicas cifradas. Em segundo lugar, pelo desafio envolvido. O violão é o instrumento que apresenta mais dificuldades quanto à interpretação de uma música cifrada. Por isso, acredita-se que resolvendo o problema para este, será possível resolver para os demais.

Para que seja possível ao usuário tocar ao mesmo tempo em que a execução é mostrada no instrumento virtual, e para facilitar a memorização, é preciso exibir a música cifrada de maneira sincronizada à execução. Por utilizar a notação das músicas cifradas, pretende-se obter o mesmo grau de simplicidade. Porém, efetivamente executando a música no computador é possível apresentar informações que a cifra não representa, como os ritmos, a melodia associada, as posições dos acordes, os solos e os tempos em que cada um destes ocorre.

Além disso, pretende-se estender o conceito de instrumento virtual dos SSIs, tornando-o interativo. Enquanto navega pelo ambiente, o usuário pode aprender conceitos musicais como ocorre, por exemplo, em *Dicionários Interativos de Acordes*.

É necessário, também, incorporar um ambiente de gravação e edição ao sistema, resgatando as principais funcionalidades das ferramentas de escrita musical existentes. Para isto, devem ser estudadas as peculiaridades do violão, o que demanda a criação de uma representação mais completa (incluindo informações específicas ao instrumento, como dedos das mãos direita e esquerda que tocam cada nota, e em que casa e corda eles estão posicionados) e a automatização de atividades racionais do músico: como a interpretação e a transposição de cifras, assim como a transcrição de solos no violão.

1.3. *D'Accord Guitar*

Como resultado deste trabalho, foi criado o *D'Accord Guitar*. Este software procura suprir uma enorme demanda de pessoas que não possuem ou não estão satisfeitas com as ferramentas de aprendizado existentes. O *D'Accord Guitar* é adequado aos músicos que utilizam músicas cifradas, e procura utilizar os recursos multimídia do computador (Roads 1996) para atingir um maior grau de satisfação deste público.

O *D'Accord Guitar* possui 3 modos de interação: o *D'Accord Guitar Player* é voltado à leitura musical e pode ser visto como um Sistema de Simulação Violonística. Porém, para poder ler tais músicas, é necessário antes escrevê-las. O *D'Accord Guitar Editor* é um sistema de edição musical apto a lidar com músicas cifradas. Tal sistema permite que a cifra musical editada seja interpretada de acordo com as preferências do usuário, incorporando para isso um sistema inteligente. O *D'Accord Guitar Browser* pode ser visto como um Dicionário de Acordes Interativo, e é voltado ao aprendizado de conceitos musicais, como o de formação de acordes.

Nos capítulos 2 e 3 serão apresentados os problemas de representação musical e os esforços computacionais já despendidos para aprimorar os métodos de escrita e leitura musical. Nos capítulos 4, 5, 6 e 7 será descrito o *D'Accord Guitar* e seus três modos de interação, através das suas arquiteturas e das maneiras como foram abordadas e solucionadas as diversas dificuldades encontradas. No capítulo 8 será visto em detalhes o problema da interpretação de cifras no violão. No capítulo 9 serão apresentados e discutidos os resultados. No capítulo 10 será feita uma análise das contribuições oferecidas pelo estudo efetuado e serão realizadas conclusões e considerações sobre as perspectivas observadas como trabalhos futuros.

A implementação do *D'Accord Guitar* envolveu a resolução de diversos problemas, de naturezas bem diferentes. Isto inclui problemas de representação de conhecimento (West, Howell & Cross 1991), de concepção de uma interface de aquisição de informações eficiente (Lavrač 1990), de definição de um formato de arquivos mais apropriado (Nugroho & Sajeev 1995), de implementação de um seqüenciador em tempo-real, de sincronização de eventos (Miner & Caudel 1998) e de criação de um sistema especialista violonista (Farret 1996).

A interpretação de cifras merece destaque especial. Ela é necessária pois é preciso um mecanismo para preencher automaticamente o que é incompleto e para selecionar a melhor opção quando ocorre ambigüidade nas cifras. Isto é o que faz um músico quando interpreta uma música cifrada, e para isto ele usa critérios oriundos da teoria musical, da sua experiência, do seu gosto pessoal, além de questões anatômicas. Este estudo não possui referências disponíveis na literatura científica atualmente e é considerada a característica mais inovadora do presente trabalho. Um primeiro protótipo foi disponibilizado na *Internet*, onde recebeu cerca de 100.000 *downloads* em cerca de 1 ano, tendo obtido uma resposta bastante positiva do público.

2. Entre a cifra e a partitura: uma solução por computador?

Neste capítulo, iremos descrever os métodos existentes para aprendizado musical, e como o computador pode ser utilizado para aperfeiçoá-los. Cabe salientar que estamos lidando especificamente com o caso do violão, que se situa na categoria de instrumentos de corda dedilhados harmônicos. Tais instrumentos são muito usados para acompanhamento e podem ser citados, além do violão, a guitarra, o cavaquinho, a viola, o banjo e o bandolim. As considerações feitas para o violão, portanto, podem ser estendidas para os demais instrumentos desta categoria, especialmente para a guitarra, que possui a mesma estrutura musical do violão (número de cordas, afinação das cordas). Para uma maior praticidade, passaremos a chamar esta categoria simplesmente de instrumentos de corda.

2.1. Métodos e ferramentas convencionais para o aprendizado de peças musicais

Durante a história, vários sistemas foram criados de forma a permitir ao ser humano descrever músicas. As chamadas notações musicais são linguagens que a maioria dos músicos utilizam para se comunicar. Através de uma determinada notação, compositores podem escrever músicas e os músicos são capazes de executá-las. Numa época em que não havia meios de gravação, esta era a única maneira de registrá-las.

Poucos destes sistemas de representação musical, ou simplesmente notações musicais, consolidaram-se entre os músicos. Os mais conhecidos e utilizados hoje em dia são as partituras e as músicas cifradas. No caso específico dos instrumentos de corda, há também as tablaturas (Michels 1994).

2.1.1. Partituras

As partituras eram até meados do século XX o único sistema de representação musical largamente utilizado, e são precisas o suficiente para que dois músicos reproduzam a mesma música de forma satisfatoriamente similar. Através delas foi possível, por exemplo, que certas peças tenham sobrevivido a vários séculos sem nunca terem sido gravadas. Na Figura 2.1 é mostrado um exemplo de partitura.

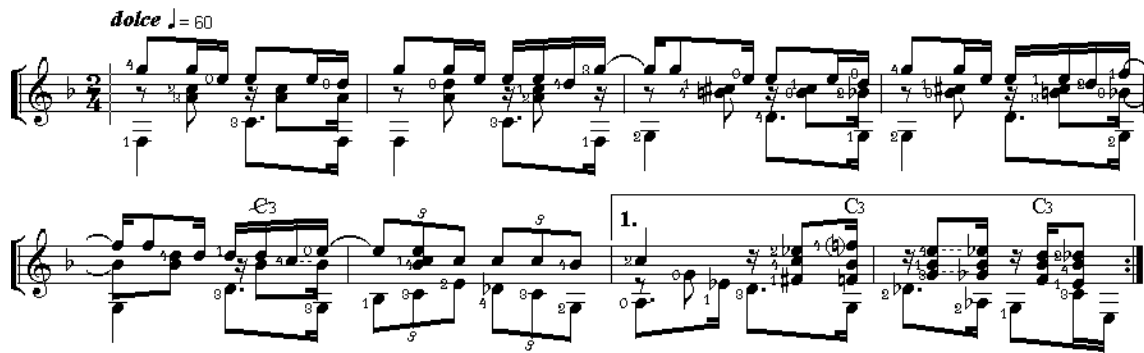


Figura 2.1– Exemplo de uma partitura.

As partituras são geralmente consideradas bastante precisas e pouco flexíveis, por representarem informações sobre cada nota a ser tocada. A Figura 2.1 exibe um exemplo de uma partitura do acompanhamento da música *Garota de Ipanema*, onde é visível a complexidade envolvida na sua leitura, já que há uma informação detalhada sobre cada nota a ser tocada. As partituras são bastante completas, permitindo a descrição de um grande número de elementos. Apesar disso, ressaltamos três problemas. Em primeiro lugar, esta riqueza de detalhes dificulta sua legibilidade, o que limita o público que a utiliza. A leitura de uma partitura é algo que exige estudo e dedicação. Embora seja altamente justificável e recomendável o seu aprendizado, a dificuldade envolvida desestimula bastante, principalmente os iniciantes. Desta forma, as partituras se enquadram perfeitamente a músicos profissionais e/ou eruditos. Porém, não se adequam a um público muito grande de pessoas que procuram um método mais simples para aprender a tocar um instrumento.

Em segundo lugar, as partituras não são totalmente adequadas para instrumentos de corda. Em uma partitura, são representadas as notas, mas o músico precisa saber como reproduzir cada nota no seu instrumento. Como nos instrumentos de corda cada nota pode ser produzida em diferentes posições, o músico precisa interpretar a partitura, adaptando-a ao seu instrumento. Esta tarefa é comumente denominada de transcrição (Michels 1988).

Em terceiro lugar, sua riqueza de detalhes também dificulta o processo de escrita. Como só as pessoas que sabem ler uma partitura sabem escrevê-las, o acervo musical em partituras fica vinculado à preferência musical destas pessoas. Por esta razão, é bastante reduzido seu acervo de música popular, por exemplo. Além disso, esta riqueza de detalhes deixa a escrita mais trabalhosa, o que também contribui para reduzir o acervo. Em diversos gêneros musicais é virtualmente impossível encontrar partituras disponíveis.

2.1.2. Tablaturas

As tablaturas são conhecidas como partituras para instrumentos de corda. Elas fazem uma analogia com as partituras, onde a pauta representa as cordas do instrumento e cada nota a ser tocada é representada por um número. O número e a linha indicam a casa e a corda a serem utilizadas. Na Figura 2.2 pode ser vista uma tablatura e uma partitura da mesma execução.

Fugue
3 = F#

Arr. John Francis J. S. Bach



Figura 2.2 – Uma tablatura e uma partitura da mesma música

A tablatura é mais direta do que a partitura, por fazer uma analogia com a execução no instrumento. Deste modo, não é necessária a transcrição citada anteriormente. Além disso, embora seja feita uma analogia com as partituras, as tablaturas são mais simplificadas, sem exigir a representação de intensidade, de andamento, de tonalidade ou uma indicação precisa dos tempos e durações dos eventos. Mesmo sendo mais simples e direta que as partituras, ainda são bastante detalhadas, tornando-as frequentemente inadequadas ao público iniciante.

2.1.3. Músicas Cifradas

Em virtude da dificuldade encontrada com as partituras e tablaturas, e da falta de necessidade de um nível tão alto de detalhamento depois da invenção de mecanismos de gravação, outras notações foram desenvolvidas. Elas usam um nível maior de abstração, visando simplificar o processo de escrita e leitura. Entre elas, destacam-se as cifras musicais¹.

As cifras são mais simples também por basearem-se no fato de que as pessoas normalmente desejam aprender a tocar músicas que já conhecem. Como a melodia, o ritmo, as variações de andamento e a letra já são conhecidos, este sistema representa apenas o componente harmônico. Ou seja, ao invés de representar cada nota tocada, representa apenas

¹ Neste trabalho utilizaremos algumas nomenclaturas diferentes para esta mesma notação: músicas cifradas, cifras musicais ou simplesmente cifras.

os acordes que compõem a harmonia da música. A ocorrência de acordes é bem menor do que a de notas em uma música, e através da harmonia é possível ao músico realizar um acompanhamento ou uma improvisação de forma bem mais direta e prática. Além disso, ao invés de basear sua representação temporal na divisão em compassos, ela associa o tempo de entrada dos acordes com a letra da música, como pode ser visto na Figura 2.3.

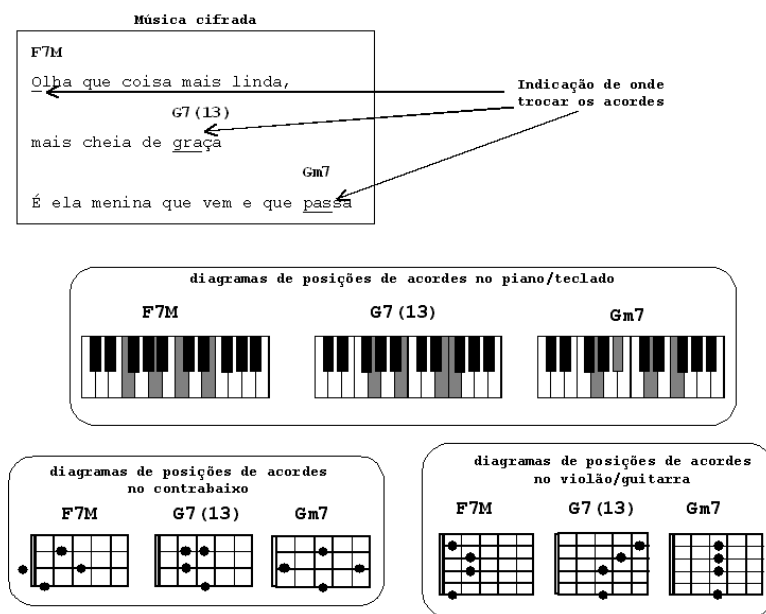


Figura 2.3 – Exemplo de um trecho de uma música cifrada, com diagramas das posições dos acordes em diferentes instrumentos

Em resumo, o método é bastante simples e intuitivo. Em geral, a indicação da duração de cada acorde é feita pelo ponto na letra onde ele aparece, e a posição como cada acorde é montado em um instrumento é mostrada em um diagrama (na Figura 2.3 há diagramas para piano, baixo e violão).

Entretanto, as músicas cifradas são incompletas, imprecisas e ambíguas. Incompletas, pois pressupõem que o músico conheça a música (especialmente a melodia e o ritmo), omitindo diversos elementos. Ambíguas, pois mais de uma cifra pode representar o mesmo acorde, já que não é uma notação padronizada. Por exemplo: para alguns, um Am9 pode representar um “Lá menor com nona”, e para outros pode representar um “Lá menor com sétima e nona”. Imprecisas, porque representam acordes que podem ser executados em diversas posições. Mesmo quando há diagramas das posições dos acordes (o que é mais comum no caso do violão), são apenas sugestões. Além disso, frequentemente há a indicação apenas das casas, mas não dos dedos.

Na Figura 2.4 são vistas cinco variações de uma mesma música cifrada. Os casos ilustram as diferenças de representação de cada acorde (um mesmo acorde com nomes diferentes, como *Bm* e *Bmin*) e de diagramação da cifra (local onde cada cifra é posicionada em relação ao texto).

<p>(a) Bm A4/7 A7 Lua, Lua, Lua</p> <p>D7MF# F°(b13) F° Guia-me</p> <p>Em7 A4/7 A7 Bm E eu te dou a minha voz</p>	<p>(b) Bmin A4 A7 Lua, Lua, Lua</p> <p>Dmaj7/F# Fdim(b13) Fdim Guia-me</p> <p>Emin7 A4 A7 Bmin E eu te dou a minha voz</p>
<p>(c) Bm / A7/4 A7 Lua, Lua, Lua D7MF# / F°(b13) F° Guia-me Em7 / A7/4 / A7 / Bm E eu te dou a minha voz</p>	<p>(d) Bm / A4.7 A7 Lua, Lua, Lua</p> <p>D7+/F# / Fo(b13) Fo Guia -me</p> <p>Em7 / A4.7 A7 Bm E eu te dou a mi nha voz</p>
<p>(e) [Bm]Lua, [A4/7]Lua, [A7]Lua [D7MF#]Guia-[F°(b13)]me[F°] [Em7]E eu te dou a [A4/7]mi[A7]nha [Bm]voz</p>	

Figura 2.4 – Diferentes modelos de cifragem

As músicas cifradas são muito adequadas para quem deseja uma maneira rápida de efetuar um acompanhamento musical em algum instrumento. Porém, a qualidade deste acompanhamento, assim como a de uma possível improvisação, depende diretamente da competência do músico. Como são imprecisas, a capacidade de interpretação do músico é o mais importante. Tanto um leigo quanto um músico profissional podem dar suas interpretações de uma cifra, porém, provavelmente serão completamente diferentes. Há um conhecimento que vai sendo acumulado por um músico que o permite saber qual a melhor forma de interpretá-la.

2.1.4. Cifras X Partituras

Na Figura 2.5 é visível a diferença do grau de dificuldade de leitura (assim como de escrita) de uma partitura e de uma cifra. Para uma melhor comparação, a partitura e a cifra referem-se à mesma música. Na primeira, é necessário dedicação e estudo apenas para

compreender o que deve ser tocado. Já a segunda é bastante intuitiva e direta, permitindo até a um leigo executar o acompanhamento.

Partitura

Música Cifrada

F7M **G7(13)**
 Olha que coisa mais linda, mais cheia de graça
Gm7
 É ela menina que vem e que passa
Gb7(#11) **Am7** **Ab7** **Db7M(9)** **Gb7(#11)**
 Num doce balanço, caminho do mar

F7M

G7(13)

Gm7

Gb7(#11)

Am7

Ab7

Db7M(9)

↑5ª casa
↑14ª casa

Figura 2.5 – comparação entre a partitura e uma cifra de um trecho da música “Garota de Ipanema”

Hoje em dia, estas duas notações convivem com outras (e.g., tablaturas), inclusive com suas variações e combinações, cada sistema se enquadrando melhor a um determinado público alvo ou a um determinado propósito. À escola tradicional de música melhor se enquadram as partituras, principalmente em casos onde é difícil fazer a distinção entre o que é arranjo e o que é a composição em si. Em peças clássicas e instrumentais, por exemplo, o arranjo é rígido. Desta forma é imprescindível que haja a indicação precisa de todas as ações executadas por cada instrumento.

Já as cifras musicais são menos precisas, provendo mais flexibilidade na execução. Isto pode ser visto como vantagem ou desvantagem, dependendo do público alvo e do seu propósito. São muito adequadas em música popular (canções), mas também em alguns gêneros de música instrumental, como o jazz, por permitirem uma maior flexibilidade acerca do arranjo e por valorizarem a improvisação.

Na tabela abaixo, compararemos estas três notações principais, através de alguns atributos que indicam a adequação de cada uma:

Notação Musical	Público Característico	Estilo	Precisão	Flexibilidade	Facilidade / simplicidade de escrever / ler?	Adequado para improvisação?	Interpretação direta no violão?
Partituras	Profissional	Clássico, instrumental	Alta	Baixa	Baixa	Não	Não
Tablaturas	Profissional ou amador	Popular, instrumental	Média	Baixa	Média	Não	Sim
Cifras	Profissional ou amador	Popular, jazz	Baixa	Alta	Média	Sim	Sim

Tabela 1 – Comparação entre partituras, tablaturas e cifras.

2.1.5. Sistemas Combinados

Existem sistemas que combinam as diferentes notações, descrevendo a melodia da música através de uma partitura e o acompanhamento através da música cifrada. Este sistema é utilizado, por exemplo, em songbooks (livros com compilações de canções), tornando-os úteis para diferentes níveis e perfis de músicos.

modelo 1: diagramas junto com a partitura

GAROTA DE IPANEMA

Tom Jobim

modelo 2: música cifrada separada da partitura

GAROTA DE IPANEMA

Antônio Carlos Jobim
Vinícius de Moraes

bossa nova

F7M
Olha que coisa mais linda
G7(13)
mais cheia de graça
Gm7
É ela menina que vem e que passa
Gb7(#11) **F#m7**
Num doce balanço, caminho do mar
Ab7 **Db7M(9)** **Gb7(#11)**

F7M **G7(13)** **Gm7** **Gb7(#11)**

F#m7 **Ab7** **Db7M(9)**

↑5ª casa ↑4ª casa

Figura 2.6 – exemplos de combinações de notações para obter um sistema mais completo

Esta separação dos diversos componentes aprimora muito o processo de aprendizado, já que cada componente pode ser aprendido isoladamente. No exemplo da Figura 2.6, a melodia (caso não seja conhecida) é mostrada em uma pauta. A harmonia é representada pelos nomes dos acordes, com suas respectivas posições no instrumento desejado. O tempo em que ocorrem os acordes pode ser verificado através da partitura ou pela posição em que aparecem na letra da música. Caso o músico já conheça a canção (no caso da Figura 2.6, *Garota de Ipanema*), ele necessita apenas da cifra para se acompanhar enquanto canta, ou para servir de base para solos e improvisações. Caso ele deseje acompanhar-se, basta posicionar os acordes

mostrados na grade e aplicar o ritmo adequado (segundo a indicação do modelo 2 na Figura 2.6, bossa-nova).

Os sistemas combinados tentam suprir uma carência de representação melódica e temporal das músicas cifradas. Entretanto, os mesmos problemas destacados anteriormente perduram. Com estes sistemas, é preciso saber ler as duas notações para tirar proveito. Como a melodia é representada por uma partitura, o público que não sabe lê-las não obtém nenhum ganho. Quanto ao acompanhamento, sofre as limitações, a imprecisão e a ambigüidade das cifras (falta de representação do ritmo, dos solos e, freqüentemente, dos dedos a utilizar).

2.1.6. Imitação

Um processo de aprendizado musical muito simples e eficiente é o da imitação (Clarke 1993). Por exemplo, quando um aluno tenta reproduzir o que um professor está tocando (Figura 2.7), o professor, além de fornecer a teoria musical e verificar o aprendizado do aluno, demonstra a execução real do que está escrito. Desta forma, é possível captar todas as nuances que não são descritas na notação, como a intensidade de cada nota e a maneira de posicionar as mãos e os dedos, por exemplo.

A imitação, no entanto, exige a presença de um professor, o que freqüentemente é inviável. Mesmo quando se aprende a tocar um instrumento com um professor, é necessário estimular a independência do aluno. Em resumo, embora a simples imitação configure um processo extremamente eficiente de assimilação de informações, ele não se configura como um método pedagógico viável para o grande público, pois exige a presença constante do professor.



Figura 2.7 – aprendizado por imitação

2.2. Métodos e ferramentas computacionais para o aprendizado de peças musicais

Os computadores possuem recursos extremamente valiosos para a aprendizagem musical (Sonnenschein 2000). São capazes de gravar e reproduzir som e vídeo, de realizar animações gráficas e de criar uma realidade virtual. Dentre os recursos tecnológicos existentes, merece destaque a tecnologia MIDI (Moog 1986). Com ela é possível ao computador comunicar-se com instrumentos MIDI, registrando e executando uma performance musical. Estes recursos (existentes na grande maioria dos equipamentos atuais) podem aprimorar os processos de leitura e de escrita de peças musicais (Woody 2000). O objetivo é utilizá-los para transpor o abismo entre simplicidade e completude discutido no início do capítulo.

A idéia principal é desenvolver aplicações que se aproximem do processo de imitação. É possível tirar proveito da capacidade de emissão de seqüências de sons e imagens sincronizadas (animações) para simular no computador uma execução musical. Em outras palavras, é possível aproximar-se da forma como um professor “ensina” uma música a um aluno, demonstrando-a na sua frente. Isto pode ser conseguido, por exemplo, com um vídeo digital ou com uma animação gráfica que simule esta execução. Deste modo, é possível obter uma evolução no método de aprendizado. Aplicações que ao mesmo tempo exibam a música sendo executada diretamente no instrumento em questão e sua representação de acordo com alguma notação musical seriam um grande avanço no processo de leitura musical.

O objetivo deste trabalho é aprimorar o processo de leitura de peças musicais no violão e na guitarra com um sistema computacional baseado em cifras, mas que obtenha uma precisão ainda superior ao de uma partitura, através de uma simulação de uma execução musical diretamente no computador (Figura 2.8).

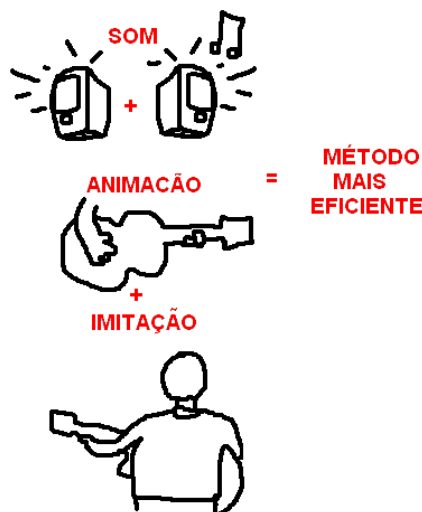


Figura 2.8 – Unindo-se os diversos recursos computacionais é possível desenvolver um método de leitura, escrita e aprendizado mais eficiente.

Sendo baseado em cifras, a leitura musical é simples. Em função da simulação, o aprendizado é mais direto. Ao mesmo tempo, a escrita musical é também simplificada, já que o usuário escreve apenas a cifra e o computador é capaz efetivamente de executá-la, baseando-se em parâmetros configuráveis pelo próprio usuário.

Desta forma, este trabalho pretende prover uma solução completa, aprimorando tanto o processo de escrita quanto o de leitura musical, baseando-se em uma notação simples, para atender a um público que não é restrito aos músicos profissionais.

2.3. *Dificuldades encontradas*

Para alcançar tal objetivo, algumas dificuldades são encontradas. Um software que efetue uma animação é uma aplicação em tempo real, onde a precisão é extremamente importante (Dannenberg & Rubine 1995). Isto envolve sincronização de eventos e comunicação direta com o sistema operacional. Para um software que dê suporte a cifras, é preciso implementar tarefas que já estão resolvidas para partituras, por exemplo, como o seqüenciamento e a criação de um formato de arquivos apropriado (Dannenberg 1993).

Além disso, esta execução precisa ser previamente gravada de alguma forma. A música, para ser lida, precisa ser escrita. Portanto, é preciso também transpor o abismo entre simplicidade e precisão quanto à escrita. Para isto, é necessária uma boa interface de aquisição da informação musical, idealmente com uma escrita simples como a das cifras, mas que de alguma forma seja executável, como as partituras (Figura 2.9).

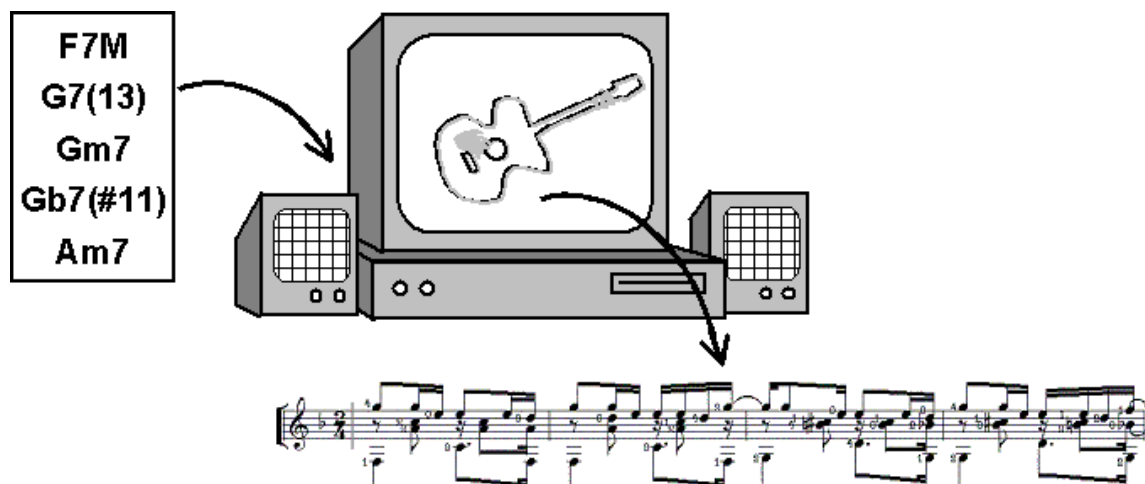


Figura 2.9 - Caso ideal: partindo de um conjunto de cifras, gerar uma execução real

Isto, no entanto, é um problema não trivial. É preciso simular o raciocínio de um músico ao interpretar uma cifra, o que frequentemente leva em consideração não só aspectos de teoria e técnica musical, como aspectos subjetivos e/ou culturais. Tais considerações são apresentadas a seguir.

2.3.1. Interpretação de Cifras

Quando um músico lê uma cifra, ele cria um acompanhamento para a melodia da música, que normalmente já é conhecida. Este acompanhamento consiste em montar acordes e aplicar um ritmo a eles. São sugeridas posições para montar cada acorde em um diagrama, e o ritmo é aplicado geralmente de acordo com o que foi escutado na gravação da música, ou através de alguma indicação na música cifrada. Os tempos podem ser indicados de duas formas. Na primeira, posiciona-se o acorde sobre o ponto, na letra, onde ele inicia (Figura 2.10). Na segunda, há uma indicação de tempo explícita (Figura 2.11), o que é muito útil quando não há uma letra para servir de referência, como é comum em canções de jazz, por exemplo.

```

F7M
Olha que coisa mais linda
                G7(13)
mais cheia de graça
                Gm7
É ela menina que vem e que passa
                Gb7(#11)      Am7
Num doce balanço, caminho do mar
                Ab7  Db7M(9)  Gb7(#11)
—      —      —

```

Figura 2.10 - Indicação de tempo em uma cifra.

F7M	/	/	/
	/	/	/
G7(13)	/	/	/
	/	/	/
Gm7	/	/	/
Gb7(#11)	/	/	/
Am7	/	Ab7	/
Db7M(9)	/	Gb7(#11)	/

Figura 2.11 - Indicação de tempo em grades de acordes.

Esta representação é muito vaga e exige que o músico faça uma interpretação da cifra. Tal interpretação depende de seu conhecimento, experiência e preferência. Não há uma melhor forma de interpretar uma cifra, pois se trata de uma escolha pessoal e, portanto, desenvolver um sistema que interprete uma cifra com qualidade é uma tarefa bastante difícil. É preciso encontrar alguma forma de automatizar estas escolhas nos espectros rítmico e harmônico.

Quanto ao aspecto rítmico, esta interpretação é diferente nos diversos instrumentos, já que cada um tem uma maneira própria de impor o ritmo e de construir os acordes. Em instrumentos de teclado, como piano e órgão, o ritmo é imposto pela pressão dos dedos nas teclas, o que permite a utilização simultânea dos 10 dedos do músico. Já os instrumentos de sopro são melódicos, ou seja, o músico executa apenas uma nota a cada momento. O ritmo é imposto pela respiração efetuada pelo músico sobre o instrumento. Os instrumentos de percussão sequer são melódicos e, portanto fogem ao escopo de interpretação de cifras. Nos instrumentos de corda, o ritmo é imposto por uma das mãos do músico, enquanto a outra constrói os acordes.

Quanto ao aspecto harmônico, o problema refere-se a como construir cada acorde e como escolher um entre eles. Cada acorde é formado por um conjunto de notas separadas por determinados intervalos. Por esta razão, nos instrumentos melódicos, que permitem a execução de apenas uma nota por vez, não é possível construírem-se acordes. Já nos instrumentos harmônicos, onde várias notas podem ser tocadas simultaneamente, tal construção é possível. Para os instrumentos de teclado, tal processo é relativamente simples, pois nestes instrumentos cada nota pode ser tocada em apenas uma posição. Sabendo-se quais notas compõem cada acorde, basta ao músico escolher em que ordem elas vão aparecer. Esta

tarefa é normalmente denominada escolha de vozes, ou *voicing*. Na Figura 2.12, é possível ver diversas escolhas de vozes para o acorde C7M no piano e em uma partitura.

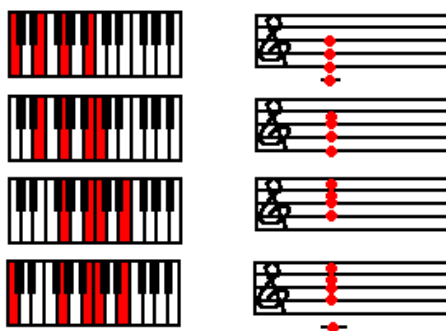


Figura 2.12 – diferentes escolhas de vozes para um C7M no piano e em partitura

Um caso bastante característico, no entanto, é o dos instrumentos de corda (harmônicos), como o violão e a guitarra. Nestes instrumentos, cada frequência pode ser gerada em vários pontos diferentes do instrumento. Na Figura 2.13 é possível ver uma ilustração do braço de um violão, que é a região onde são posicionados os acordes. Nesta figura, são exibidos 3 pontos diferentes onde pode ser tocado um C5 (Dó, na quinta oitava). Cada uma destas posições é aqui denominada de coordenada.

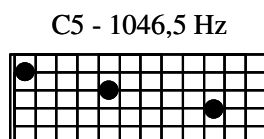


Figura 2.13 – três posições consonantes em uma representação do braço do violão

Nestes tipos de instrumento, portanto, a construção de acordes é mais difícil. É preciso escolher que posições e que dedos serão usados para cada acorde. Estas duas escolhas são comumente chamadas de *posicionamento* e *digitação*, vistas a seguir.

Posicionamento

O primeiro problema consiste em encontrar as posições de um determinado acorde no braço do violão. Na Figura 2.14, por exemplo, são exibidas 3 escolhas diferentes de posições para os acordes C7M, Dm7 e G7.

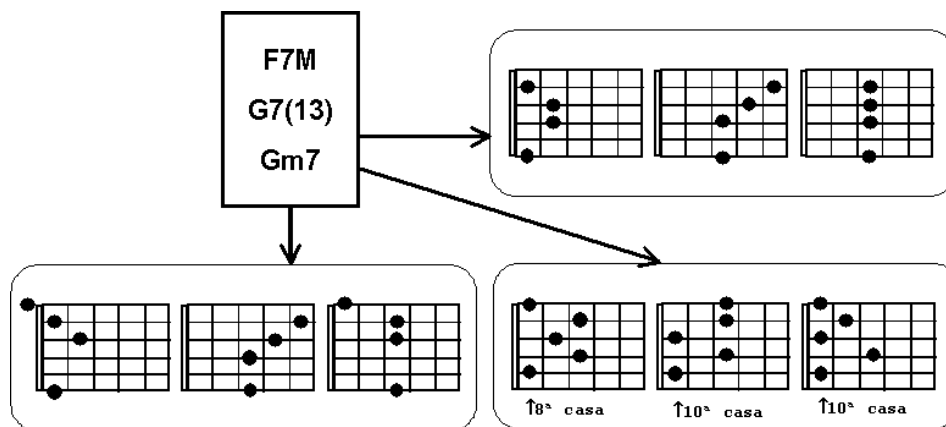


Figura 2.14 – diferentes escolhas de posições

Cada acorde é formado por um conjunto de notas, que podem aparecer em qualquer ordem de apresentação. Elas podem inclusive ser repetidas ou omitidas. A posição de um acorde no violão é um conjunto de coordenadas no braço deste instrumento, onde em cada coordenada uma nota é tocada. O problema de posicionamento consiste em encontrar posições no braço que sejam exequíveis pelo músico, e cujas coordenadas condigam com as notas que compõem o acorde. Neste ponto, um grave problema refere-se ao grande número de posições possíveis. Alguns cálculos podem ser feitos para ilustrar este fato.

Há uma variação no número de trastes existentes em violões e guitarras, normalmente variando de 19 a 24. Para simplificar os cálculos, digamos que o músico possa apenas montar acordes nas primeiras 12 casas. Desta forma, uma oitava inteira é coberta. Isto equivale a dizer que em cada corda podem ser tocadas todas as notas de um acorde. Dado um acorde composto de n notas, existem no mínimo $6*n$ coordenadas a serem utilizadas, que podem ser combinadas de diversas formas para montar o acorde. A Figura 2.15 mostra as 30 coordenadas disponíveis nas 12 primeiras casas para se montar o acorde Fm7(11), dado que ele é formado pelas notas F, Ab, C, Eb e Bb.

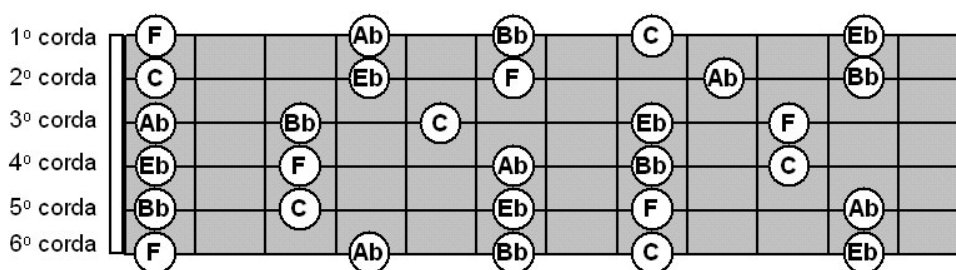


Figura 2.15 - Diferentes coordenadas que podem ser utilizadas para montar um Fm7(11).

Dependendo do número de notas que compõem o acorde, pode haver mais de 100.000 combinações possíveis destas coordenadas. A grande maioria delas, entretanto, não forma o acorde desejado, de acordo com a teoria musical. Além disso, dentre as combinações que efetivamente formam um acorde, uma grande parte não é exequível pelo músico. É preciso encontrar as combinações destas coordenadas que formem o acorde em questão seguindo restrições anatômicas, da teoria musical e da técnica violonística.

Na Figura 2.16 pode ser visto um exemplo. Das 5 posições mostradas, a *a* é a única posição correta. A posição *b* não se caracteriza como um *Fm7(11)*, em virtude de não possuir as notas *Eb* e *Ab*. A posição *c* possui uma digitação que contraria a técnica violonística. A posição *d* necessita que sejam utilizados mais dedos do que o possível. A posição *e* é inviável anatomicamente, em virtude da grande distância entre os dedos.

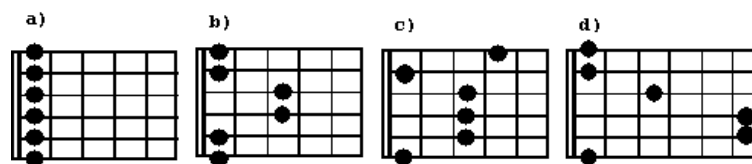


Figura 2.16 – Possíveis posições e digitações de um *Fm7(11)*

Neste exemplo, do acorde *Fm7(11)*, existem 43750 combinações possíveis das 30 coordenadas a serem usadas para montá-lo nas 12 primeiras casas do violão (assumindo que tal acorde pode ser construído utilizando-se 6, 5 ou 4 cordas). Destas, 5880 efetivamente formam um *Fm7(11)*. Destas, apenas 283 são exequíveis pelo músico.

Dentre estas 283 encontradas, é preciso levar em consideração diversos critérios para escolher determinadas posições em detrimento de outras. Estes critérios são o maior desafio a enfrentar no problema de posicionamento, e são discutidos em maiores detalhes no capítulo 8.

Certos critérios são oriundos da teoria musical, e através deles certas escolhas de vozes são preferidas a despeito de outras. Por exemplo, foi dito que certas notas que compõem um acorde podem ser repetidas ou omitidas. Porém, de acordo com a teoria musical, são preferidas as posições que não repitam e/ou omitam determinadas notas (Fowler 1984a). Tais critérios são relativamente simples de avaliar, já que a teoria musical é conhecida. Por exemplo, de acordo com a Figura 2.17 a primeira posição é considerada inferior às demais por repetir a nota C. A quinta posição é também considerada inferior mas, neste caso, por dobrar a nota E.

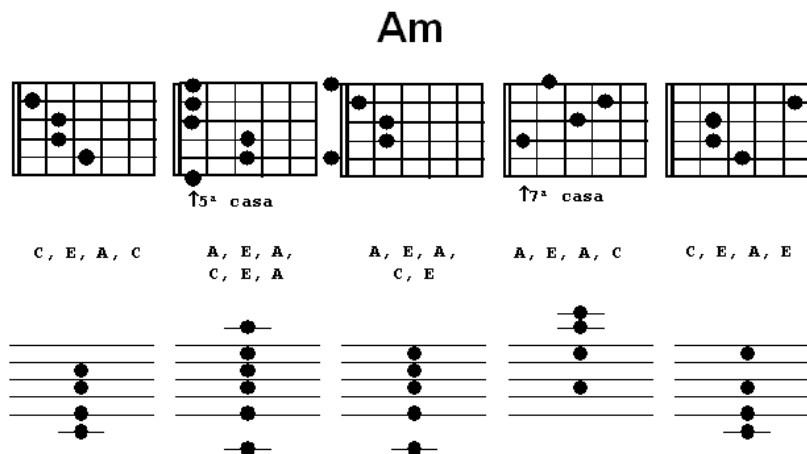


Figura 2.17 – exemplos de posições do acorde Am no violão e sua respectiva escolha de vozes

Existem também critérios culturais, que normalmente não possuem uma explicação objetiva. Por exemplo, um músico pode escolher determinadas posições apenas porque são as mais comumente utilizadas. Tais critérios são extremamente difíceis de tratar, uma vez que não há a possibilidade de definir regras para descrevê-los. Por exemplo, na Figura 2.17, a terceira posição é normalmente considerada a mais comum.

Outros critérios são anatômicos, tratando das escolhas que os músicos fazem em função da dificuldade em tocar cada posição, ou nas restrições que cada posição pode acarretar. Por exemplo, a primeira e a quarta posições da Figura 2.17 não deixam nenhum dedo livre. Desta forma, o músico fica impossibilitado de, por exemplo, fazer um arranjo melódico enquanto realiza o acompanhamento. Ainda na Figura 2.17, normalmente a terceira posição é preferida por ser considerada a mais confortável. Descobrir que posições são mais confortáveis ou flexíveis, no entanto, também é uma tarefa não trivial. É preciso fazer uma modelagem física da mão esquerda do músico, assim como do braço do violão.

Além de todos estes critérios de avaliação, a escolha das posições deve levar em consideração o encadeamento de acordes. Com isto, cada escolha depende das anteriores e das posteriores, o que torna o problema extremamente complexo. Por exemplo, frequentemente um músico prefere seqüências de posições onde haja a menor variação na armadura dos dedos. É comum, também, preferir uma seqüência que possua uma linha de baixo mais suave (ou seja, onde as notas mais graves de cada acorde sejam mais próximas).

Digitação

O segundo problema na interpretação de cifras consiste em definir quais dedos utilizar em cada corda. Em outras palavras, consiste em encontrar as possíveis digitações para uma determinada posição. A quantidade de combinações envolvidas neste caso é pequena quando comparada com o problema de posicionamento, pois todas as configurações possíveis para uma mesma posição envolvem apenas quatro dedos. No entanto, a tarefa de reconhecer quais dentre estas configurações de dedos são exeqüíveis pelo músico é mais difícil.

Também no caso da digitação existe uma dificuldade de aquisição de um conhecimento não explícito do músico. Além disso, a escolha da melhor digitação a aplicar em cada situação varia de músico para músico e de estilo para estilo. Um fator complicador é a possibilidade de utilização de pestana, principalmente nos casos em que sua utilização é facultativa. Uma pestana é a utilização de um único dedo (tipicamente, o indicador) para pressionar várias cordas ao mesmo tempo. A Figura 2.18 mostra algumas das possibilidades de digitações para uma posição de Dm7, tanto com a utilização de pestana (digitações *b* e *f*) quanto sem a utilização de pestana (demais digitações). Destas, apenas as digitações *a*, *b*, *e*, *f* e *g* são permitidas.

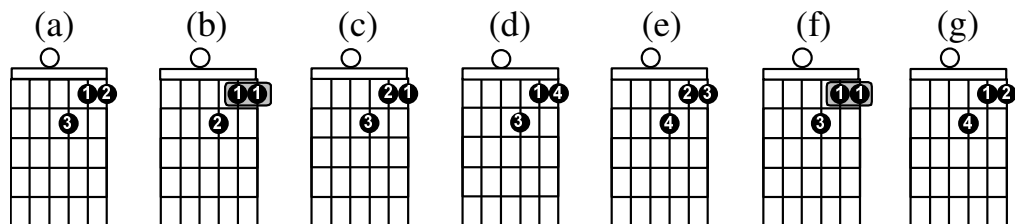


Figura 2.18 – Possíveis fingerings de uma posição de Dm7

Assim como acontece com o problema do posicionamento, na digitação também é necessário um processo de avaliação para permitir que alguma, dentre as digitações encontradas, seja escolhida, em detrimento das demais. Normalmente, os músicos fazem tal escolha baseando-se em critérios anatômicos, culturais e de técnica violonística, muitas vezes difíceis de elicitar. A seguir, será descrito o processo decisório relativo à interpretação de cifras de uma maneira geral, englobando critérios de posicionamento e de digitação.

O Processo decisório

De acordo com o conhecimento e a experiência acumulados por um músico, ele lança mão dos diversos critérios citados (musicais, anatômicos, culturais, limitações técnicas) para escolher que posições utilizar para cada acorde. Para citar um exemplo, um músico

iniciante frequentemente conhece apenas uma posição para cada acorde. Mesmo quando conhece mais de uma, prefere posições mais confortáveis, já que não tem habilidade suficiente para executar posições mais difíceis. Neste caso, o acompanhamento geralmente é pobre. Já um músico experiente prefere posições onde a escolha de vozes seja mais rica, ou onde haja uma transição mais suave.

Os valores de alguns dos critérios citados podem ser automaticamente encontrados ou sugeridos. Outros precisam ser atribuídos diretamente por um ser humano, como por exemplo as posições mais utilizadas. Alguns precisam de uma modelagem física, outros de definições de regras vindas da teoria musical. A importância que o músico dá a cada critério também varia de músico para músico e de situação para situação (e.g., se ele pretende inserir linhas melódicas ao acompanhamento). Outro grande problema é, portanto, como avaliar todos estes critérios em conjunto para prover uma interpretação de uma cifra de acordo com as preferências de um músico.

2.4. *Conclusão*

O software apresentado neste trabalho possui como requisitos a facilidade e a simplicidade de leitura e de escrita, de forma a facilitar o aprendizado de peças musicais. Para isto, baseia-se em músicas cifradas, por serem simples e de larga utilização. Ao mesmo tempo, precisa ser flexível para permitir a entrada de componentes que não são descritos na notação de músicas cifradas. Além disso, deve ser apto a efetivamente tocar a música gerada, o que demanda a utilização de técnicas de inteligência artificial para preencher a lacuna de representação desta notação. O capítulo seguinte apresenta as principais categorias de software que de alguma forma facilitam a leitura e escrita musical, e quais conceitos são aproveitados para conceber o *D'Accord Guitar*.

3. Sistemas Computacionais para o Aprendizado de Peças Musicais

Neste capítulo, serão vistas as principais iniciativas que já foram tomadas para utilizar o computador como ferramenta de auxílio aos processos de escrita, leitura e aprendizado de peças musicais. Serão classificados os diversos sistemas apresentados e será avaliado o modo como eles atendem aos requisitos levantados no capítulo anterior, suas qualidades e deficiências, reaproveitando as boas idéias para conceber nosso sistema. É importante salientar que não foram encontrados detalhes de implementação destes sistemas, uma vez que são aplicações comerciais.

3.1. *Sistemas de Simulação Instrumental*

Provavelmente, a maneira mais fácil de aprender como tocar um instrumento musical é pela observação de alguém o tocando. Baseado nesta suposição, alguns sistemas exibem um instrumento virtual onde é possível visualizar uma música sendo tocada.

Os *sistemas de simulação instrumental* (SSI) (Cabral, 2001b) são sistemas que, ao invés de mostrar uma música em uma partitura, mostram a sua execução em um instrumento virtual ao mesmo tempo em que ela é tocada. Um instrumento virtual é uma imagem da parte pertinente do instrumento em questão. No caso do piano, por exemplo, são exibidas as teclas. No caso do violão, é exibido seu braço, já que nele são mostrados os acordes (Figura 3.1). A ação do músico também é exibida através de animações gráficas, onde os dedos do músico são mostrados como figuras estilizadas.

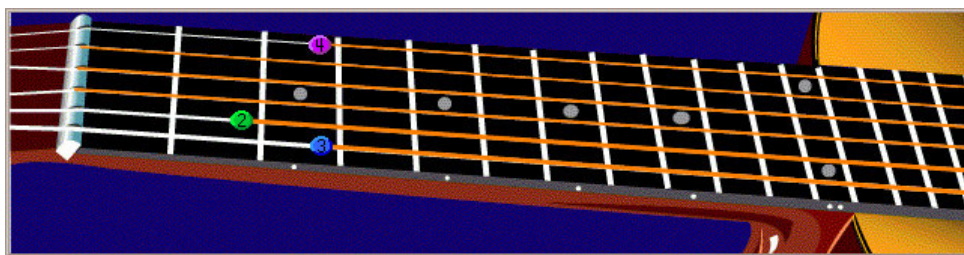


Figura 3.1 – Braço de violão virtual

Os SSIs permitem aumentar a precisão, a completude e a simplicidade da notação musical utilizada. Geralmente eles também oferecem algumas funções complementares. Por exemplo, alterar a tonalidade e o tempo, procurar músicas na *Internet* e exibir informações

extra sobre a música (e.g., autores, gênero, ano de gravação, comentários). *The Pianist*², *The Guitarist*³ (Figura 3.2), *iSong*⁴, e *Desktop Guitar*⁵ são exemplos de SSIs.

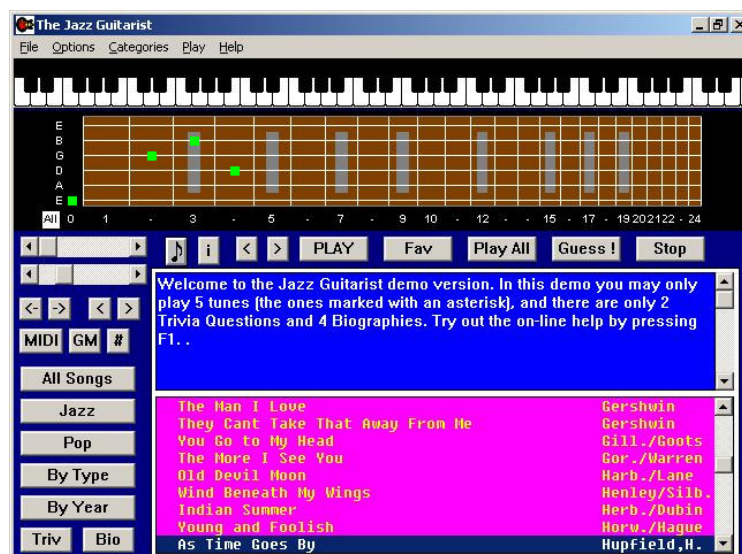


Figura 3.2 – Tela principal do SSI “The Guitarist”

Algumas críticas podem ser feitas aos SSIs. Em primeiro lugar, é preciso ressaltar que seu principal objetivo é simular a execução musical de um humano tocando uma determinada música em um determinado instrumento. Eles visam ensinar canções específicas a pessoas já com alguma experiência com o instrumento, e por isto não tomam muito cuidado com teoria musical ou técnicas instrumentais. Com isto, é difícil para um músico iniciante utilizá-los.

Em segundo lugar, o nível de interatividade é muito baixo. O usuário torna-se normalmente um mero espectador (executando apenas comandos gerais como tocar, parar, trocar de tonalidade ou mudar a velocidade), sem poder interagir com o instrumento virtual como se fosse um instrumento real.

Em terceiro lugar, em praticamente todos os SSIs atuais, não há a possibilidade de edição. Desta forma, os usuários não podem criar ou alterar suas próprias músicas. As exceções são os sistemas da Desktop Music (e.g., *Desktop Guitar*, *Desktop Banjo*). Neles, o usuário pode escrever uma música no violão, desde que informe o tempo e a coordenada de cada nota tocada, tornando o processo de edição praticamente inviável. Por esta razão, o

² Software fabricado pela empresa PG Music (www.pgmusic.com)

³ Software fabricado pela empresa PG Music (www.pgmusic.com)

⁴ Software fabricado pela empresa iSong (www.isong.com)

⁵ Software fabricado pela empresa Desktop Music (www.desktopmusic.com)

acervo musical disponível nos SSIs é bastante reduzido e, conseqüentemente, são pouco utilizados. A implementação de um recurso de edição poderia torná-los ferramentas de uso comum entre os músicos.

Em quarto lugar, vários SSIs não incluem a exibição da letra da música. De acordo com a tendência atual (que pode ser constatada nos diversos *songbooks* disponíveis no mercado), o fato de utilizar a letra da música como referência para posicionar os acordes e a melodia parece facilitar a leitura da música. A maioria dos SSIs, entretanto não as mostra. Mesmo quando são mostradas, freqüentemente não são sincronizadas com as cifras dos acordes e com a melodia.

Em quinto lugar, a interface dos SSIs para instrumentos de corda (em particular o violão), não mostra a informação apropriadamente, complicando a interpretação da música sendo tocada no braço do violão. De fato, os SSIs atuais dão ênfase à informação melódica (solo), encobrendo a harmonia (acordes) subjacente. Por exemplo, a Figura 3.3 mostra uma seqüência de notas em um braço de violão de um SSI convencional. Vendo assim, não é imediato saber que se trata do arpejo de um *Am7*.

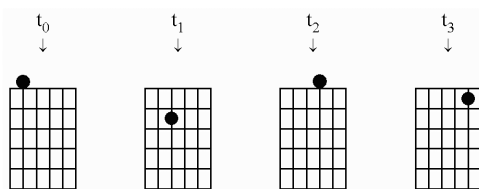


Figura 3.3 – Arpejo de um *Am7* em um SSI convencional

Já que nem a cifra do acorde é sincronizada com a execução da música, nem os dedos da mão esquerda são mostrados em cada coordenada, é muito difícil para o usuário reconhecer a harmonia. Na execução de uma música completa, este reconhecimento é quase impossível, já que vários arpejos são encadeados e tocados rapidamente (Holdsworth 1998).

Outros tipos de sistemas parecem superar ao menos parcialmente algumas destas deficiências, e serão descritos a seguir. Incorporando alguns recursos existentes em tais sistemas, é possível desenvolver um SSI bem mais completo.

3.2. *Sistemas de Acompanhamento Automático(SAA)*

Os *Sistemas de Acompanhamento Automático* (Ramalho, Rolland & Ganascia 1999) são sistemas que geram um acompanhamento musical completo (para todos os instrumentos desejados) baseados simplesmente em uma grade de acordes e em um estilo musical

particular. São geradas linhas melódicas (e.g., saxofone ou baixo), rítmicas (e.g., bateria) e/ou posições de acordes (e.g., piano).

São muito freqüentemente utilizados por arranjadores e músicos profissionais. Por arranjadores, para não precisarem se preocupar com o arranjo de todos os instrumentos (por exemplo, para criar o arranjo de cordas, mas sem se preocupar com o arranjo do baixo e da bateria). Por músicos, para servir de assistente de ensaio (por exemplo, para gerar um acompanhamento que servirá de base para um solo). Como exemplos de SAAs, podemos citar o *Band-In-A-Box*⁶ e o *Harmony Assistant*⁷.

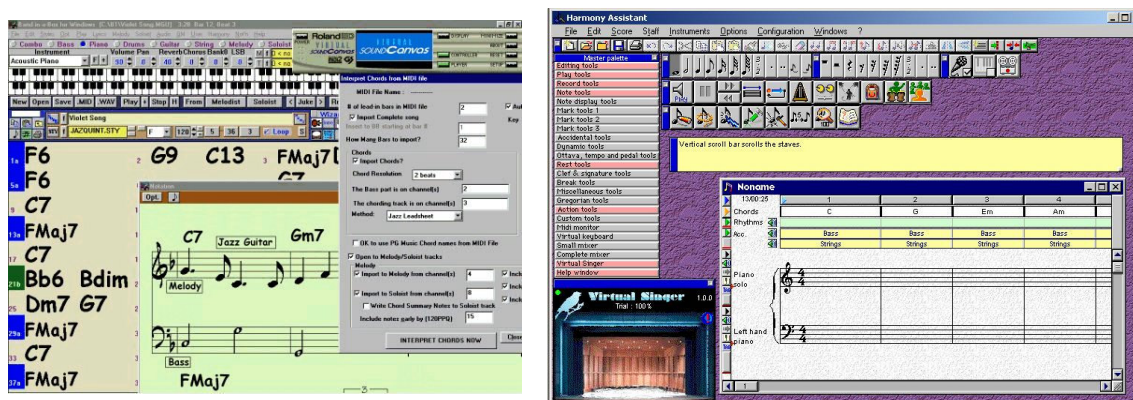


Figura 3.4 – Telas dos softwares *Band-In-A-Box* e *Harmony Assistant*

Como foi dito, praticamente em todos os SSIs é impossível aos usuários escreverem suas próprias músicas. Mesmo quando possível, a realização da escrita é inviável, pois o processo é exaustivo. Nos SAAs, o processo de edição é bem mais eficiente, o que poderia ser reaproveitado nos SSIs. Eventualmente, tais sistemas podem até incluir um módulo SSI para exibir a música gerada em um instrumento virtual (normalmente, um teclado), resolvendo parcialmente alguns dos problemas discutidos aqui. Por exemplo, eles mostram cifras de acordes sincronizadamente com o restante da música, facilitando a interpretação das notas tocadas (coordenadas exibidas). E, principalmente, o usuário pode editar as cifras da música em uma grade de acordes e associá-la à melodia da música (Figura 3.5).

⁶ Fabricado pela empresa PG Music (www.pgmusic.com)

⁷ Fabricado pela empresa Myriad Software (www.myriad-online.com)

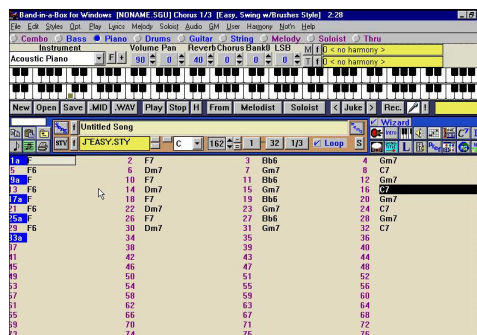


Figura 3.5 – Edição da grade de acordes no *Band-In-A-Box*

No entanto, o principal objetivo dos SSAs é gerar seqüências MIDI. Portanto, para estes sistemas não é tão importante o tratamento de certos detalhes de execução. Isso se agrava bastante quando se trata de instrumentos de corda, pela complexidade de modelagem (explicada no capítulo anterior). Em geral, não há indicação das posições possíveis de um acorde e, quando há, a abordagem é bastante simplista: cada acorde é limitado a apenas uma posição (normalmente, a posição mais comum). A única exceção encontrada foi o *Band-In-A-Box*, a partir da versão 10. Nele, é feita uma associação entre posições de acordes e estilos musicais. Isto é uma solução grosseira, uma vez que a escolha de posições é uma tarefa que leva em consideração diversos outros fatores. A Figura 3.6 mostra uma instância desta associação, para o acorde *Am7*. Nela, a posição *a* é associada ao estilo *bossa-nova*, enquanto a posição *b* é associada ao estilo *rock*. Isto quer dizer que quando um usuário editar uma música que contenha o acorde *Am7* no estilo *bossa-nova* a posição *a* será escolhida, o que torna o resultado final bastante artificial.

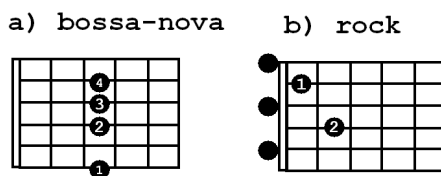


Figura 3.6 – Associação entre posições de acordes e estilos musicais

Além disso, os SAAs atuais não tratam as digitações no braço do violão, como pode ser visto na Figura 3.7. Desta forma, é mais difícil ao usuário compreender como deve executar cada nota. Estas limitações inibem um uso apropriado destes sistemas para instrumentos de corda, mesmo com a popularidade de que desfrutam.

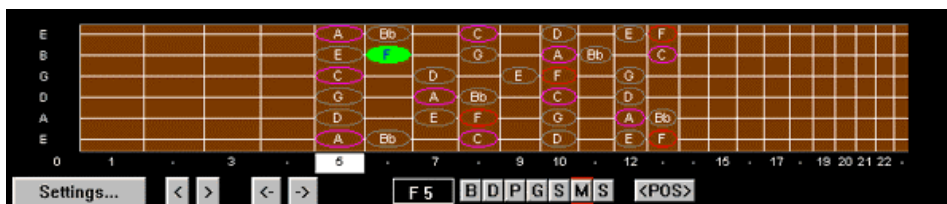


Figura 3.7 – Braço de violão representado a partir da versão 10 do Band-In-A-Box

Outro problema encontrado é o baixo grau de flexibilidade dos SAAs. Na maioria destes sistemas é impossível editar um ritmo e/ou as posições dos acordes. Assim, o usuário precisa aceitar o acompanhamento gerado sem efetuar ajustes.

Os SAAs podem ser vistos como ferramentas complementares aos SSIs, por visarem um processo de escrita mais simplificado, da mesma forma que os SSIs visam simplificar o processo de leitura. A incorporação deste recurso aos SSIs parece ser o caminho natural para difundir seu uso. Entretanto, para o caso dos instrumentos de corda, é necessário resolver antes os problemas apresentados. Em outras palavras, é preciso automatizar o processo de interpretação de cifras no violão, o que é um problema não enfrentado até o presente momento.

3.3. Editores de Tablaturas

Os editores de tablaturas são similares aos editores de partituras, e resolvem parcialmente o problema de edição para instrumentos de cordas, por utilizar uma notação já apropriada a eles. Como exemplo, pode ser citado o *Guitar Pro*⁸ (Figura 3.8). Existe um ambiente gráfico que possibilita a edição da música diretamente na tablatura exibida na tela do computador. A edição também pode ser feita através de uma guitarra ou de arquivo MIDI. No entanto, a falta da exibição de um instrumento virtual dificulta a sua compreensão.

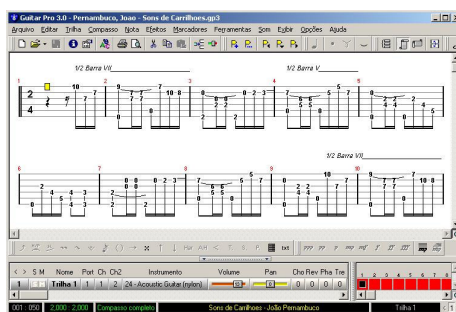


Figura 3.8 – Janela principal do editor de tablaturas Guitar Pro

⁸ Fabricado pela empresa Guitar Pro (www.guitar-pro.com)

3.4. Dicionários Interativos de Acorde

Outro tipo de sistema relacionado ao nosso tema de estudo são os *Dicionários Interativos de Acorde* (DIAs). Eles provêm um conceito que também merece ser aproveitado: a interação com o instrumento virtual. Tais sistemas são micro-mundos⁹ (Nichol 1988), onde o usuário pode aprender conceitos musicais enquanto navega no ambiente. Pode-se consultar as possíveis posições de um determinado acorde, verificar que notas o compõem e onde encontrá-las no braço do violão. Além disso, é possível interagir diretamente com o braço do violão para definir a posição que o usuário desejar, e o sistema reconhece a que acorde se refere tal posição. *Chord Wizard*¹⁰ (Figura 3.9) e *Violão Virtual*¹¹ são exemplos de dicionários de acordes.

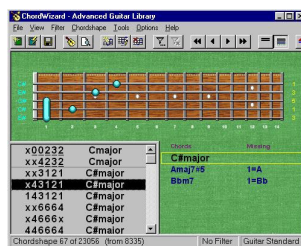


Figura 3.9 – Janela principal do software Chord Wizard

Apesar de altamente interativos, o objetivo de tais sistemas não é exibir uma execução musical num instrumento virtual. Em outras palavras, a informação é estática e, portanto, não é possível utilizá-los para exibir uma música. A integração desta interatividade em um SSI, entretanto, permitiria que fosse aprimorado de forma sensível o processo de aprendizado.

3.5. Sistemas de Karaokê

Por fim, há uma categoria que não é propriamente voltada ao aprendizado, mas que possui um conceito reaproveitável: são os *Sistemas de Karaokê* (Figura 3.10). Eles são sistemas simples, que sincronizam a exibição da letra da música com a música sendo tocada. A incorporação de um sistema de karaokê em um SSI seria bastante interessante, animando tanto a letra quanto a cifra dos acordes. Isto seria bastante útil para um grande número de

⁹ Um micromundo é definido como um ambiente onde o usuário adquire conhecimento não pela exposição explícita da informação, seja na forma de explicações ou de demonstrações. O conhecimento é adquirido pela simples navegação neste ambiente. Esta abordagem tem demonstrado ser a mais indicada para o ensino utilizando o computador, através de ferramentas interativas, que despertem o interesse pelo aprendizado sem torná-lo cansativo.

¹⁰ Fabricado pela Chord Wizard (www.chordwizard.com)

¹¹ Fabricado pela MGB Informática (www.violaovirtual.com.br)

músicos, principalmente iniciantes, que têm dificuldade em saber o instante em que cada acorde entra. Mesmo assim, poucos SSIs efetivamente implementam isso.

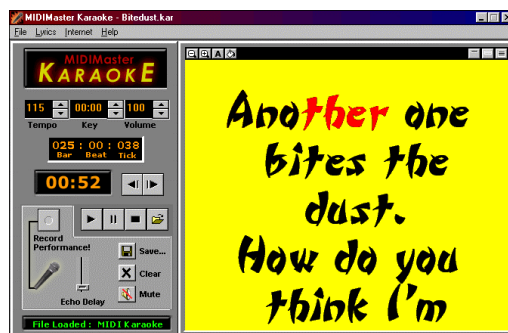


Figura 3.10 – Janela principal do software MIDIMaster Karaoke¹².

3.6. Conclusão

Na tabela abaixo, são comparados os diversos tipos de sistemas apresentados aqui, em função dos recursos que possuem.

	SSI	SAA	Editores de Tablaturas	DIA	Sistemas de Karaokê
<i>Baseado em Imitação (Instrumento Virtual)</i>	SIM	pode incorporar um instrumento virtual	NÃO	SIM	NÃO
<i>Permite edição</i>	Quando permite, é inviável	SIM	SIM	NÃO	NÃO
<i>Apto para iniciantes</i>	NÃO	NÃO	NÃO	SIM	SIM
<i>Interativo</i>	Muito pouco	na edição	na edição	SIM	NÃO
<i>Sincronização com a letra da música</i>	Raramente	SIM	NÃO	NÃO	SIM
<i>Propício para instrumentos de corda</i>	Existem SSIs específicos, mas não são satisfatórios	Apenas o Band-In-A-Box 10, mas ainda não é satisfatório	SIM	SIM	NÃO

Tabela – comparação dos diversos sistemas (SSI = Sistemas de Simulação Instrumental, SSA = Sistemas de Acompanhamento Automático, DIA = Dicionários Interativos de Acordes)

Para obter um sistema que ao mesmo tempo aprimore o processo de leitura, aprendizado e escrita musical, há a necessidade de integrar diversas das funcionalidades apresentadas. Isto não sido efetuado ainda, aparentemente pela grande dificuldade de concepção inerente.

¹² Fabricado pela empresa *Blaze Audio* (www.singingelectronics.com)

Para aprimorar o processo de leitura, o sistema implementado neste trabalho fundamentou-se nos SSIs, mostrando a música sendo tocada diretamente no violão e/ou guitarra. Além disso, ele deve se basear na notação de cifras, que é mais simples. Desta forma, será possível sincronizar a execução musical com a letra (já devidamente cifrada), animada como em um sistema de karaokê. Como foi visto, no entanto, os atuais SSIs ainda não possuem soluções satisfatórias para os instrumentos de corda, uma vez que não representam apropriadamente a informação musical. É necessária, portanto, a criação de um sistema de representação que incorpore as particularidades destes instrumentos.

Para aprimorar o processo de aprendizado, o sistema deve incluir a possibilidade de interação existente nos DIAs, para que seja possível aprender sobre formação de acordes e escalas.

Para aprimorar o processo de escrita, o sistema deve utilizar um processo de edição semelhante ao dos SAAs. Deste modo é possível uma escrita musical simplificada, baseando-se em uma notação já de uso comum (cifras) para gerar um acompanhamento automático no violão e/ou guitarra de acordo com preferências do usuário. Entretanto, como foi visto, o acompanhamento gerado nos SAAs atuais para instrumentos de cordas é muito simplista, já que não levam em consideração detalhes característicos da sua execução. É preciso, então, modelar o raciocínio de um violonista/guitarrista ao interpretar uma música cifrada. Este é um problema ainda não resolvido na literatura científica e espera-se com este trabalho oferecer uma grande contribuição.

4. *D'Accord Guitar*

Este capítulo descreve o *D'Accord Guitar*, um sistema específico para violão e guitarra, que busca aperfeiçoar o processo de leitura, aprendizado e escrita musical. Integrando funcionalidades das aplicações citadas no capítulo anterior e acrescentando outras, o *D'Accord Guitar* busca sobrepor as falhas apontadas, e seu desenvolvimento exigiu, além de um grande trabalho de implementação, um grande trabalho de concepção.

É difícil enquadrar o *D'Accord Guitar* em uma categoria convencional de aplicativo musical. De certa forma, ele pode ser visto como um sistema de simulação instrumental editável, específico para violão, incorporando o processo de edição dos sistemas de acompanhamento automático para escrever músicas cifradas. O software é acrescido com funcionalidades de dicionários de acordes, tutoriais de música e sistemas de karaokê. Ele provê ainda um módulo inteligente, que pode ser visto como um sistema especialista ou um sistema de apoio a decisões para violonistas.

De fato, o *D'Accord Guitar* engloba um sistema de simulação instrumental, onde o instrumento virtual é um braço de violão (como o braço é idêntico para o violão e para a guitarra, o sistema serve para estes dois instrumentos). Ele pode ser visto, portanto, como um SSI, mas com uma interface gráfica que permita uma representação mais rica do que a destes (incluindo a indicação dos dedos, das casas e das cordas a utilizar).

Da mesma forma, o *D'Accord Guitar* engloba um dicionário interativo de acordes, possibilitando a interação direta com o instrumento virtual, tornando-se um micromundo musical. Nele, o usuário pode aprender conceitos sobre formação de acordes e navegar pela informação existente. O problema encontrado nos DIAs, sobre a impossibilidade de criação de animações e, conseqüentemente, de execuções virtuais, obviamente é resolvido no *D'Accord Guitar*, já que ele incorpora um SSI.

O *D'Accord Guitar* também incorpora um sistema de karaokê, onde a música cifrada (e não somente a letra da música) é animada.

Por fim, o *D'Accord Guitar* incorpora o processo de edição dos sistemas de acompanhamento musical automático, podendo ser visto também como um editor musical específico para violão e apto a editar músicas cifradas.

Este capítulo discute os princípios e a arquitetura geral do *D'Accord Guitar*. Nos capítulos seguintes, serão detalhadas a abordagem utilizada, as dificuldades encontradas e as

principais decisões tomadas durante o desenvolvimento desta ferramenta, assim como serão propostas soluções para alguns dos problemas previamente discutidos.

4.1. Princípio

O princípio básico do *D'Accord Guitar* é apresentar a informação de forma mais próxima da maneira como um aluno aprende músicas em um instrumento. Neste trabalho, estamos interessados no aprendizado de acompanhamento de músicas no violão. Neste caso, há duas idéias básicas: a primeira é a de que uma separação apropriada dos diversos elementos musicais (West, Howel & Cross 1991) melhora o aprendizado do usuário, por deixá-lo focar em um elemento particular por vez; a segunda é que, para uma notação musical ser precisa, a ponto de indicar exatamente como a música deve ser tocada, é preciso tratar as particularidades de cada instrumento. A seguir serão descritas as razões de ter-se adotado este tipo de abordagem.

4.1.1. Separação entre acompanhamento e melodia

Como estamos interessados em aprimorar o aprendizado de um acompanhamento, é necessário separar o que é melodia e o que é acompanhamento. Porém, em uma partitura, harmonia, melodia e ritmo são representados da mesma forma: uma sequência de notas e durações, sem distinguir entre um arranjo solístico e a harmonia da música, por exemplo. Desta forma, é complicado representar um acompanhamento através de uma partitura, ao contrário das cifras. Ao ler uma cifra, os músicos lidam apenas com o acompanhamento (assume-se que a melodia já é conhecida), aprendendo separadamente o ritmo e a harmonia da música.

Pretendeu-se, neste trabalho, desenvolver um sistema onde o aprendizado aproxime-se deste modelo, pois temos como hipótese de trabalho que esta separação facilita o aprendizado e o processo de memorização, por exigir o aprendizado de padrões rítmicos e harmônicos, ao invés de sequências de notas. No *D'Accord Guitar* é feita uma distinção clara entre acompanhamento e melodia. Acompanhamento é a parte executada pelo violão e é exibido no instrumento virtual. Melodia é a parte cantada, associada à letra da música, e é apenas escutada de forma sincronizada à animação da letra (karaokê).

O acompanhamento de acordes, por sua vez, é visto não como um conjunto de notas tocadas sequencialmente, mas como um ritmo sendo aplicado a uma sequência de acordes. Desta forma, em vez de aprender uma grande sequência de notas, o usuário pode focar no reduzido número de acordes, associados a um padrão rítmico conhecido. A representação

explícita da harmonia subjacente, por exemplo, pode resolver um dos problemas ilustrados no capítulo anterior: reconhecer as posições dos acordes a tocar quando são mostrados diferentes arpejos encadeados. Abaixo é exibido um arpejo de uma posição de um *Am7* e de um *D7* no *D'Accord Guitar*, em contraste com a maneira padrão de exibi-lo, como ilustrado no capítulo anterior.

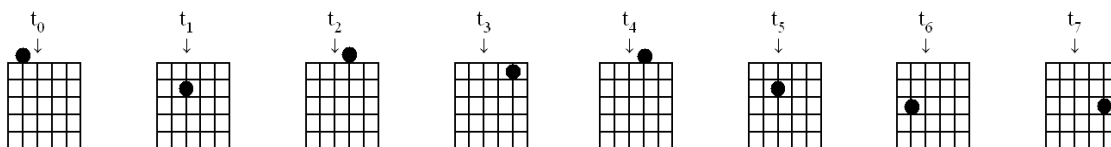


Figura 4.1 – Arpejo de um *Am7* e de um *D7* em um SSI convencional

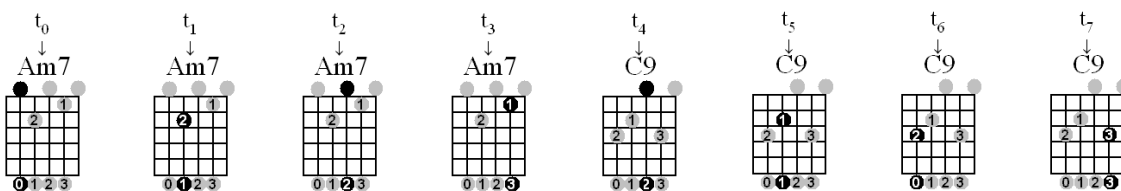


Figura 4.2 – Arpejo de um *Am7* e de um *D7* no *D'Accord Guitar*.

Enquanto é tocado um arpejo de um acorde, o *D'Accord Guitar* não mostra a animação nota a nota, como um SSI tradicional. A posição do acorde é mantida e as notas sendo tocadas são destacadas, como ilustrado na Figura 4.2. A separação entre harmonia e ritmo é indicada pelos círculos no alto e na base. Os círculos no alto indicam dedos da mão esquerda (harmonia), e os círculos na base indicam dedos da mão direita (ritmo). Isto se aproxima bastante da visualização de uma execução real, e facilita a leitura. De fato, percebendo a presença dos acordes em adição à das notas individuais, é possível obter uma visão mais clara do acompanhamento.

Analogamente ao processo de leitura, o processo de escrita também é simplificado com esta separação. É possível, inclusive, omitir alguns elementos e gerar alguns automaticamente. Por exemplo, o usuário pode especificar uma grade de acordes, importar a melodia de um arquivo MIDI, e especificar um padrão rítmico para ser aplicado aos acordes, como ilustrado na Figura 4.3. Neste exemplo, não é necessário preocupar-se com a letra da música nem com a gravação da melodia da música, tornando o processo bastante direto.

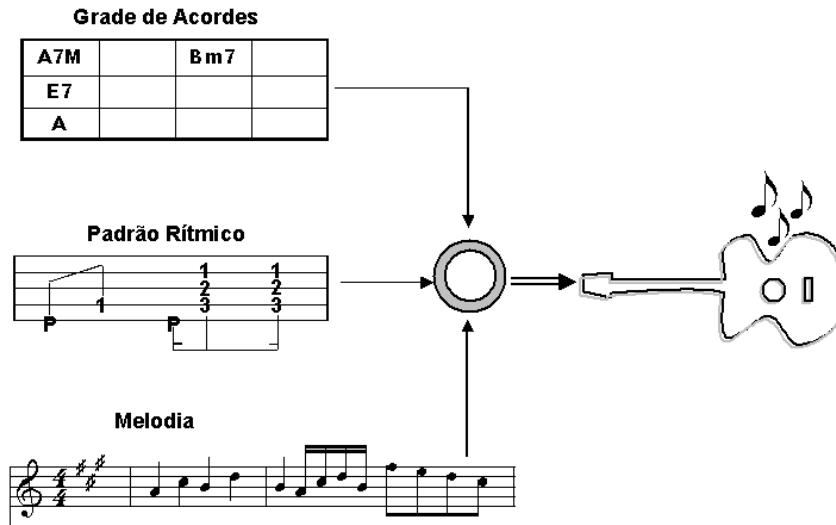


Figura 4.3 – Edição musical baseada em harmonia, ritmo e melodia

Na Tabela 4.1, podem ser vistos quais elementos são utilizados em algumas situações. Quando se deseja apenas aprender como efetuar um acompanhamento, basta representar-se a grade de posições de acordes e os ritmos a serem aplicados. No caso das músicas cifradas, este acompanhamento é sincronizado com a letra da música. A melodia e o ritmo não são indicados, porque se assume que o músico já os conheça. Nos *songbooks*, a melodia da música é indicada. O ritmo, quando indicado, é feito de modo superficial.

Situação	Elementos utilizados
Acompanhamento	Harmonia + Ritmo
Música Cifrada	Harmonia + Letra
<i>Songbooks</i>	Harmonia + Letra + Melodia
Execução real	Harmonia + Ritmo + Letra + Melodia + Solo + Outros

Tabela 4.1 – Exemplos dos elementos utilizados em algumas situações

O *D'Accord Guitar* engloba a representação de informações harmônicas, rítmicas, melódicas, de solo, de letra e de variação de andamento. Ele está apto a editar e executar músicas de forma mais completa que as cifras musicais ou os *songbooks*, tentando aproximar-se da representação de uma execução real (em uma execução real existem outros elementos como, por exemplo, os solos que são tocados para dar uma melhor qualidade ao acompanhamento). Porém, isto possui um preço: é preciso sincronizar todos os eventos, uma atividade não trivial, como será visto no próximo capítulo.

4.1.2. Representação direta

A segunda idéia básica do *D'Accord Guitar* é que a leitura musical é mais direta se houver uma analogia entre a notação e o instrumento real. Isto é o que ocorre com os desenhos de posições de acordes nas músicas cifradas e com a diagramação das tablaturas, por exemplo. O que buscamos no *D'Accord Guitar* foi uma simulação virtual do violão, de forma que o aprendizado se aproximasse ao máximo do processo de imitação.

O *D'Accord Guitar* escolheu o violão como instrumento base por duas razões primordiais. Em primeiro lugar, por sua popularidade e relevância cultural (principalmente no Brasil) e pela falta de soluções apropriadas. Em segundo lugar, por ser o tipo de instrumento que apresenta o maior grau de dificuldade com relação à representação (como visto no capítulo 2). Acredita-se que resolvendo o caso para o violão, será possível resolver para os demais instrumentos.

Para isso, a representação visual da informação precisa ser específica para o violão, tratando de suas peculiaridades. No *D'Accord Guitar*, ao invés de haver apenas a indicação da nota a ser tocada (como ocorre em geral), há a indicação da posição de cada dedo no braço do violão virtual. No momento em que tal nota é tocada, os dedos correspondentes (das mãos direita e esquerda) são destacados. Desta forma, não é necessário uma interpretação da notação. Na Figura 4.4, os círculos pretos indicam as cordas que estão efetivamente sendo tocadas, e os círculos cinzentos indicam dedos que não estão tocando, mas que estão posicionados.

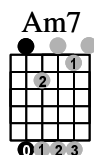


Figura 4.4 – Representação de uma posição do acorde Am7

É possível ao usuário verificar não só auditivamente quanto visualmente a maneira como a música é tocada, percebendo-a e memorizando-a melhor.

4.2. Modos de interação

O *D'Accord Guitar* pode ser usado em três modos de interação diferentes, cada uma tendo em vista um determinado público. São eles o *D'Accord Guitar Player*, o *D'Accord Guitar Editor* e o *D'Accord Guitar Browser*.

O *D'Accord Guitar Player* pode ser visto como o sistema de simulação instrumental do *D'Accord Guitar*, utilizado para tocar as músicas no violão virtual. Com este sistema, o usuário pode executar e efetuar não só comandos gerais sobre a canção (como tocar, parar, repetir, adiantar e retroceder a música, e mudar o andamento), mas também comandos mais especializados (como transpor a música ou escolher posições para os acordes). O *D'Accord Guitar Player* é o carro-chefe, voltada para o grande público interessado no aprendizado de acompanhamentos musicais no violão.

O *D'Accord Guitar Browser* pode ser visto como dicionário interativo de acordes do *D'Accord Guitar*, e permite interagir diretamente com o instrumento virtual e consultar a base de acordes. Ele é voltado ao aprendizado de teoria sobre formação de acordes, sendo útil também como obra de referência.

O *D'Accord Guitar Editor* é o modo de interação que permite a escrita de músicas para serem tocadas no *D'Accord Guitar Player*, sendo voltado para a parcela do público interessada em escrever músicas para violão baseada em cifras.

Embora cada modo de interação tenha sua própria arquitetura, eles compartilham uma mesma base de classes, seguindo as regras de reutilização provenientes da engenharia de software (Pressman 1997). Tais classes básicas fornecem um conjunto de funções de 4 tipos: *de execução, especialistas, de gravação e de interação*. A Figura 4.5 mostra quais categorias de funções são utilizadas por cada modo de interação (linhas contínuas indicam a utilização de todas as funções da categoria e linhas tracejadas indicam a utilização de parte das funções da categoria).

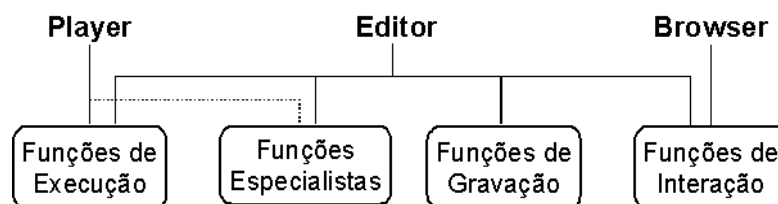


Figura 4.5 – Relacionamento entre os modos de interação e os conjuntos de funcionalidades disponíveis nas classes básicas

O *D'Accord Guitar Player*, *D'Accord Guitar Editor* e o *D'Accord Guitar Browser* são 3 módulos de um mesmo sistema, chamado *D'Accord Guitar*. Nos próximos capítulos, iremos descrevê-los em maiores detalhes, mostrando as funcionalidades existentes e o modo como foram abordados os problemas encontrados.

5. D'Accord Guitar Player

O objetivo principal do *D'Accord Guitar Player* é exibir uma música sendo tocada no violão da maneira mais próxima possível à maneira como um músico vê outro músico tocando. Neste capítulo, será visto o modo como este sistema representa as informações, seu funcionamento, sua arquitetura e a metodologia utilizada para resolver os problemas encontrados.

5.1. Interface

O projeto de interface do *D'Accord Guitar* visa principalmente à simplicidade, já que é um requisito do software que sua utilização seja a mais direta e intuitiva possível, para não desestimular o usuário. Na Figura 5.1, pode ser vista a janela principal do software.



Figura 5.1 – Interface do D'Accord Guitar Player

O *D'Accord Guitar Player* exibe em separado a música cifrada (na parte inferior da janela), o acorde atual e o seguinte (no canto superior esquerdo), e a execução violonística propriamente dita (no violão virtual, no centro). Tal saída gráfica é sincronizada com a execução sonora, o que faz o usuário associar a animação à música.

5.1.1. Instrumento Virtual

Na parte central da janela há o braço do violão virtual indicando de maneira mais detalhada a performance das mãos direita e esquerda. São ilustrados a posição do acorde e os

dedos das duas mãos. O braço é exibido, a princípio, do ponto de vista do músico ao tocar um violão (a cabeça do violão à esquerda, a primeira corda em cima e a sexta corda em baixo).

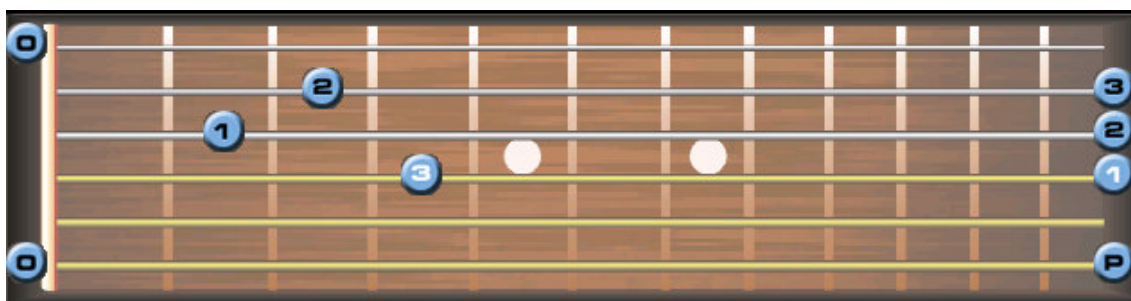


Figura 5.2 – Braço do violão virtual

A Figura 5.2 mostra um exemplo, para ilustrar como tais informações são exibidas. Nela, está sendo mostrada uma posição comum do acorde *Mi Maior com Quarta, Sétima e Nona*, cuja cifra é *E4/7(9)*. A posição do acorde é exibida da seguinte forma: em primeiro lugar, ela é montada utilizando os dedos 1, 2 e 3 (indicador, médio e anular) sobre as cordas 3, 2 e 4, respectivamente. Tais dedos estão posicionados, também respectivamente, sobre as casas 2, 3 e 3. As cordas 1 e 6 são tocadas soltas (sem nenhum dedo pressioná-las).

Quando uma corda é tocada, as figuras dos respectivos dedos são destacadas. Na Figura 5.2, o dedo 1 da mão direita e, conseqüentemente, o dedo 3 da mão esquerda são tocados. Com esta representação, acredita-se conseguir estabelecer-se uma relação direta entre o que é executado e o que é exibido, tornando o aprendizado e a leitura mais intuitivos.

5.1.2. Música cifrada

No canto inferior da janela é exibida a música cifrada (letra e cifras dos acordes), animada como em sistemas de karaokê. Para muitos músicos, apenas esta animação da música cifrada já é o suficiente para efetuar o acompanhamento. Embora seja bastante simples, tal recurso enriquece o processo de leitura mesmo de músicos experientes.

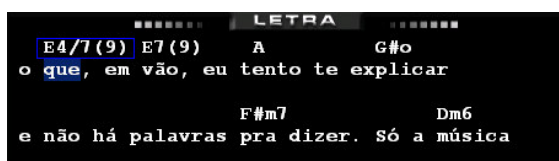


Figura 5.3 – Letra e cifras animadas

5.1.3. Acordes

Além da animação da música cifrada como em um karaokê, o *D'Accord Guitar* exibe claramente a harmonia, através da cifra tanto do acorde sendo tocado quanto do próximo (Figura 5.4).

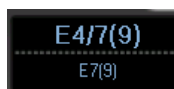


Figura 5.4 – Indicação do próximo acorde

5.2. Funcionalidades

As funcionalidades do *D'Accord Guitar Player* incluem *tocar*, *parar*, *abrir*, *salvar*, *imprimir* e *transpor uma música*, *alterar configurações* do ambiente, entre outras. Elas podem ser acessadas através do menu (situado no canto superior esquerdo da janela) ou da barra de controles de execução (situada na parte superior central da janela).

5.2.1. Manipulação de arquivos

No *D'Accord Guitar*, cada música é armazenada em um arquivo em formato proprietário, com extensão DV3. As funções de manipulação de arquivo são funções básicas, como *abrir*, *salvar* e *imprimir*. Para escolher a música a ser tocada, deve-se *abrir* o arquivo correspondente. A opção de *salvar* armazena as alterações feitas. Na Figura 5.5 é exibida a janela da função *Visualizar Impressão*, e na Figura 5.6 é mostrada a impressão final.

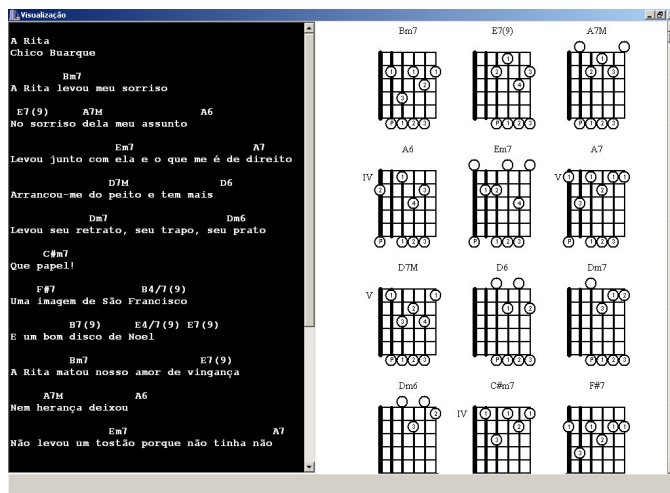


Figura 5.5 – Exemplo de música a ser impressa.

A Rita
Chico Buarque

<p>Em7 A Rita levou meu sorriso</p> <p>E7(9) A7#6 A6 Ho sorriso dela meu encanto</p> <p>Em7 A7 Levou junto com ela e o que me é de direito</p> <p>D7#9 D6 Arrancou-me do peito e tem mais</p> <p>Dm7 Dm6 Levou seu retrato, seu tempo, seu gesto</p> <p>C#m7 Que papel?</p> <p>F#7 B4/7(9) Uma imagem de São Francisco</p> <p>E7(9) B4/7(9) E7(9) E um bom disco de Noel</p> <p>Em7 E7(9) A Rita matou nosso amor de vingança</p> <p>A7#6 A6 Hes herança deixou</p> <p>Em7 A7 Não levou um toado porque não tinha não</p> <p>D7#9 D6 Hes causou perdas e danos</p> <p>Dm7 Dm6 Levou os meus planos, meus pobres enganos</p> <p>C#m6 F#7(b13) Os meus vícios seus, o meu coração</p> <p>E7 E7(9) A6 E além de tudo me deixou tudo o violão</p>	<p>Bm7</p> <p>(P) 1(2)3</p> <p>E7(9)</p> <p>(P) 1(2)3</p> <p>A7M</p> <p>(P) 1(2)3</p> <p>A6</p> <p>IV</p> <p>(P) 1(2)3</p> <p>Em7</p> <p>(P) 1(2)3</p> <p>A7</p> <p>V</p> <p>(P) 1(2)3</p> <p>D7M</p> <p>V</p> <p>(P) 1(2)3</p> <p>D6</p> <p>(P) 1(2)3</p> <p>Dm7</p> <p>(P) 1(2)3</p> <p>Dm6</p> <p>(P) 1(2)3</p> <p>C#m7</p> <p>IV</p> <p>(P) 1(2)3</p> <p>F#7</p> <p>(P) 1(2)3</p> <p>B4/7(9)</p> <p>(P) 1(2)3</p> <p>B7(9)</p> <p>(P) 1(2)3</p> <p>B4/7(9)</p> <p>(P) 1(2)3</p> <p>C#m6</p> <p>III</p> <p>(P) 1(2)3</p> <p>F#7(b13)</p> <p>(P) 1(2)3</p> <p>B7</p> <p>(P) 1(2)3</p>
---	---

Figura 5.6 - Exemplo de uma música impressa.

5.2.2. Controle de execução

As funções de controle de execução fazem analogia a um aparelho de som (Figura 5.7). Além dos comandos padrão (*tocar*, *parar*, *interromper*, *retroceder* e *avançar*), existe o *loop*, que permite que o usuário defina um trecho a ser repetido indefinidamente (geralmente um trecho onde o usuário esteja experimentando alguma dificuldade), aprimorando seu aprendizado.



Figura 5.7 – Controles de execução

Durante a execução musical, vários indicadores mostram o instante corrente da música. Com a opção de *linha de tempo* é possível ir para um ponto específico, caso seja desejado.



Figura 5.8 – Linha de tempo e outros indicadores

Por fim, é possível alterar a velocidade (andamento) da música, através da *barra de velocidade* (Figura 5.9), que permite que a música seja tocada mais lenta ou mais rapidamente

do que o normal. Ela é útil para que o usuário possa perceber com maior precisão detalhes de execução.



Figura 5.9 – Barra de controle da velocidade

5.2.3. Funções inteligentes

Algumas funcionalidades do sistema envolvem atividades realizadas por músicos, como interpretar uma cifra de acordo com critérios do usuário, transpor a música ou converter a melodia em um solo.

Escolha de posições de acordes

As funções de interpretação harmônica são a maior contribuição do *D'Accord Guitar*, por seu aspecto inovador, modelando uma atividade fundamental entre os músicos quando utilizam notações como a de cifras. Grosso modo, as funções de interpretação harmônica tratam de gerar um acompanhamento para violão baseado em um conjunto de cifras de acordes e em preferências escolhidas pelo usuário.

As funções de interpretação harmônica são chamadas no *D'Accord Guitar Player* através da opção *escolha de posições de acordes*. Ela permite ao usuário atribuir critérios para serem usados para que o sistema escolha a melhor maneira de posicionar os acordes. Esta é talvez a principal decisão que um músico precisa tomar ao interpretar uma cifra, e por isso é de extrema relevância.

A escolha de critérios pode ser feita em diferentes níveis. Em um nível macro, o usuário simplesmente escolhe o perfil a que se adequa (e.g., iniciante, profissional, solista), e o sistema automaticamente ajusta os parâmetros de todos os critérios de acordo com este perfil. Num nível mais detalhado, o usuário pode ajustar pessoalmente os parâmetros de cada critério a ser levado em consideração para escolher as melhores posições (e.g., facilidade, frequência de utilização, flexibilidade). Na Figura 5.10, pode ser vista a interface utilizada para configurar os parâmetros desta escolha.



Figura 5.10 - Parâmetros para a escolha de acordes

Num nível ainda mais detalhado, é possível ao usuário definir restrições quanto aos acordes a serem gerados, e dar preferência a determinadas seqüências de posições (gatilhos). A Figura 5.11 mostra a janela onde é possível restringir aos acordes a serem considerados.



Figura 5.11 - Configuração de restrições sobre os acordes a serem gerados

Sendo configurável, esta escolha de posições é útil para diferentes tipos e níveis de músicos, interpretando a música de acordo com suas preferências. Ela é não trivial, e não se encontra nos demais softwares existentes no mercado. A implementação desta funcionalidade vai além do aprimoramento do processo de leitura ou aprendizado. Ela pode servir como um sistema tutor, auxiliando o músico em sua atividade cotidiana.

Transposição

A opção de *transposição* permite ao músico alterar a tonalidade da música, e é usada, por exemplo, para que fique mais fácil posicionar os acordes ou para que a melodia localize-

se numa região mais confortável para se cantar. Para utilizar a opção de transposição, basta ao usuário informar em quantos tons ele deseja elevar ou rebaixar a música (Figura 5.12).

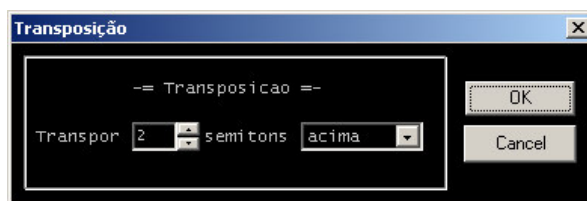


Figura 5.12 - Função de transposição

Assim como a funcionalidade anterior, a transposição de músicas no violão também é uma tarefa não trivial, e não costuma estar coberta entre as funcionalidades disponíveis nas aplicações existentes atualmente.

Alternar entre harmonia e solo

Outra função importante trata da visualização da informação: *alternar entre harmonia e solo*. Como padrão, o *D'Accord Guitar* assume que a informação a ser visualizada no braço do violão seja a harmonia (as posições dos acordes). Porém, através desta opção é possível visualizar a maneira de tocar a melodia no violão, convertendo-a em um solo (Wang & Li 1997), como pode ser visto na Figura 5.13.

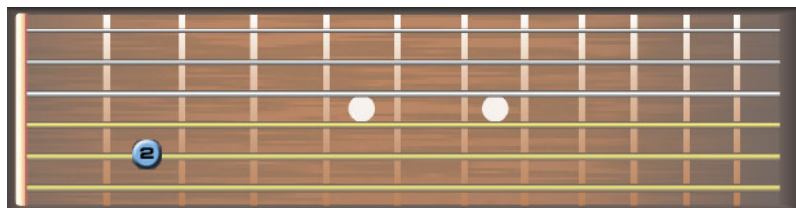


Figura 5.13 – Trecho de uma melodia, convertida em solo.

Isto é muito útil quando, por exemplo, dois músicos estão tocando em conjunto. Neste caso, freqüentemente um deles executa o acompanhamento enquanto o segundo executa um solo. Esta opção, assim como as anteriores, vai além do aperfeiçoamento da notação, servindo como um sistema de apoio a decisão para violonistas.

5.2.4. Configuração

As funções de configuração servem para ajustar a saída MIDI e a orientação do braço do violão. As opções MIDI referem-se ao ajuste do volume e do instrumento relativo a cada canal. Este ajuste encontra-se na opção *Configurações/Instrumentos* (Figura 5.14). Existem

dois canais principais: um que executa as notas do violão e outro que executa as notas da melodia. Para alterar o volume destes canais, há uma opção na parte inferior da tela.

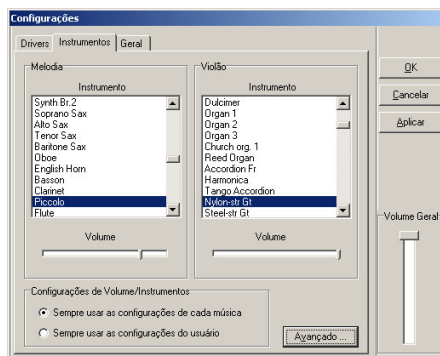


Figura 5.14 – Configuração do volume e instrumento relativos a cada canal

A opção *Configurar* serve ainda para alterar o driver MIDI utilizado como saída (Figura 5.15), de forma semelhante aos demais aplicativos musicais.

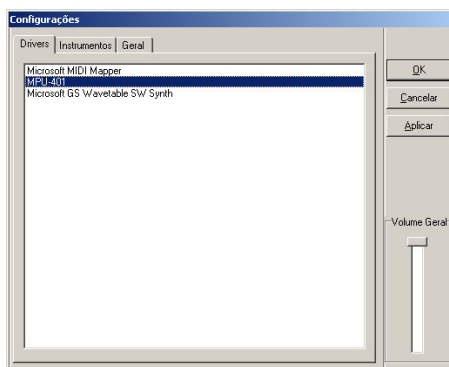


Figura 5.15 – Escolha do driver MIDI

É possível, ainda, configurar o braço do violão, de forma que exiba a informação da maneira mais intuitiva ao usuário. Abaixo são apresentadas as 4 maneiras possíveis. A opção *Padrão* mostra o violão como o próprio músico o vê, de cima. As cordas mais graves aparecem em baixo e as mais agudas em cima. A opção *Vista frente* mostra o violão como uma pessoa vendo outra tocá-lo. É útil para quem já é acostumado a aprender vendo outra pessoa tocando.

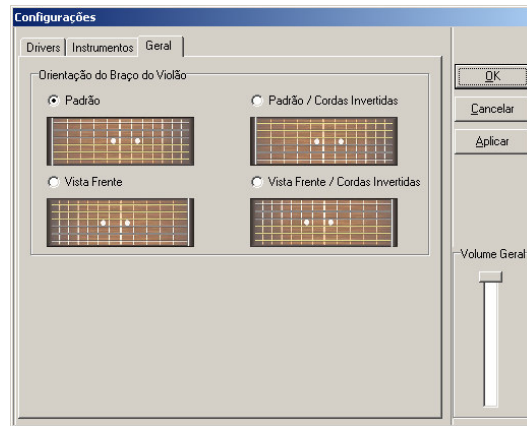


Figura 5.16 – As 4 orientações possíveis para exibir o braço do violão

5.3. Arquitetura

Na figura a seguir (Figura 5.17), pode ser vista a arquitetura do *D'Accord Guitar Player*. O usuário interage com o sistema através das diversas funcionalidades existentes, que estão dispostas em 4 grupos: funções de manipulação de arquivo, funções de controle de execução, funções de configuração e funções musicais inteligentes. O sistema responde ao usuário através das saídas gráficas e sonora.

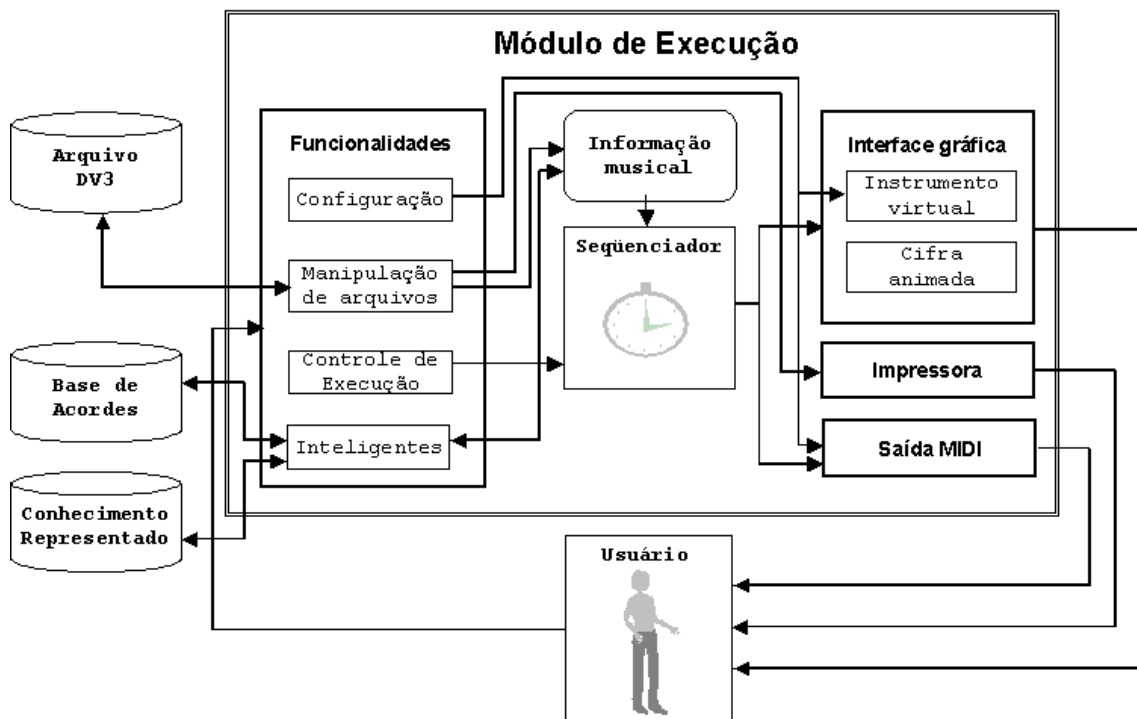


Figura 5.17 – Arquitetura do D'Accord Guitar Player

As funções *abrir* e *salvar* (funções de manipulação de arquivo) acessam arquivos no formato DV3, carregando em memória a *informação musical*, e vice-versa. A função de *impressão* (também de manipulação de arquivos) leva esta *informação musical* à impressora.

As funções de configuração alteram as propriedades gráficas do ambiente ou as propriedades sonoras da saída MIDI.

As funções de controle de execução tratam efetivamente de executar o que está contido na *informação musical* (a música, devidamente estruturada), sincronizando os eventos através de um *seqüenciador*.

As funções musicais inteligentes utilizam o conhecimento armazenado na *base de conhecimento* e na *base de acordes* para modelar o raciocínio de um músico nas atividades de interpretação de cifras, transposição e transcrição de solos. Através destas funções, é possível alterar a *informação musical*, de acordo com as *preferências* escolhidas pelo usuário.

5.4. *Concepção e implementação*

Alguns detalhes relativos ao desenvolvimento do *D'Accord Guitar Player* merecem destaque e serão descritos a seguir. Eles referem-se à criação de uma representação própria para a descrição das músicas, às dificuldades envolvidas no processo de seqüenciamento e à concepção das funções inteligentes.

5.4.1. **Representação musical**

O primeiro problema encontrado é o de representação da informação musical, seja em arquivo, seja em memória.

Arquivos DV3

O único formato de arquivo atualmente consolidado e que é suportado por bibliotecas de programação é o MIDI (Robbins 1995). Os arquivos MIDI, no entanto, possuem apenas informação melódica. Os demais formatos, que porventura pudessem ser mais completos e apropriados, são proprietários e não possuem qualquer espécie de suporte. Portanto, a única solução encontrada foi criar nosso próprio formato de arquivo.

O formato de arquivo usado no *D'Accord Guitar* foi batizado de DV3 e usa como base a idéia de eventos dos arquivos MIDI padrão. Porém, os eventos especificados no protocolo MIDI são limitados. Existem apenas eventos melódicos e de controle. Aqui não nos ateremos aos eventos de controle (eventos que não se aplicam a notas específicas, mas a todas

as notas), por fugirem ao escopo da discussão. Quanto aos eventos melódicos, cada nota a ser tocada é representada pela sua altura, intensidade e instante em que deve ser tocada (ver Figura 5.18).

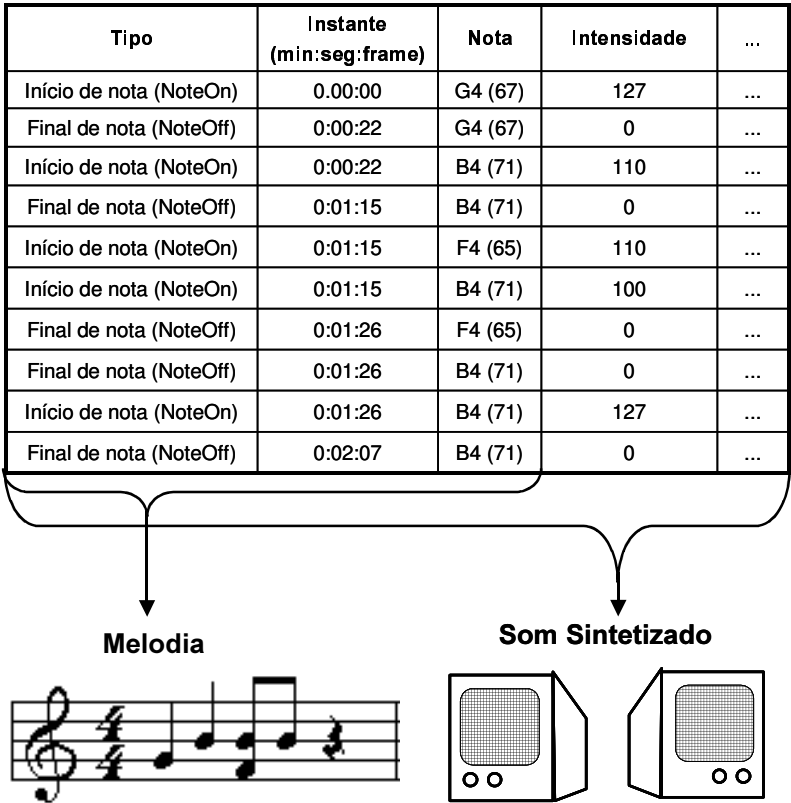


Figura 5.18 – Reprodução de uma melodia a partir de um arquivo MIDI

Nos arquivos DV3, no entanto, existem também eventos para:

- Acordes, e suas respectivas posições.
- Melodia.
- Letra, e os trechos a serem animados.
- Solos.
- Ritmos.

Na Figura 5.19 pode ser vista a estrutura de classes destes eventos musicais.

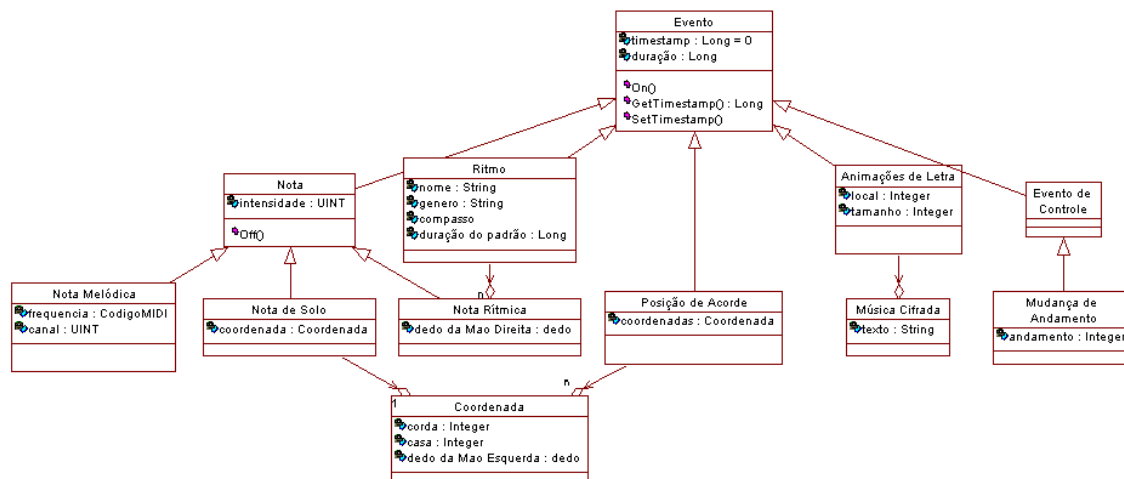


Figura 5.19 – Hierarquia das classes relativas a eventos em DV3

Os eventos de nota são análogos aos eventos de nota existentes em MIDI. Eles são divididos em 3 tipos: as *notas da melodia*, as *notas rítmicas* e as *notas do solo*. As notas da melodia são idênticas às notas MIDI, incluindo informações de altura, canal e intensidade. As notas de solo incluem ainda informações específicas para a sua visualização no braço do violão, como o dedo, a casa e a corda a utilizar. As notas rítmicas são eventos que simulam o toque de um dedo da mão direita em alguma corda. Logicamente, armazenam a informação do dedo utilizado, assim como da intensidade.

Os eventos de letra são usados para efetuar as seleções na letra, de forma que o texto da música cifrada funcione como um karaokê e, portanto, armazenam a coordenada e o tamanho da seleção.

O único evento de controle definido é o de mudança de andamento. Ele é definido como o número de batidas por minuto (*bpm*), como nos arquivos MIDI (Honing 2001), e afeta a velocidade com que a música é executada.

Os eventos de posição de acorde servem para montar os acordes, simulando a movimentação da mão esquerda do músico. Eles indicam, além da cifra do acorde, a coordenada de cada dedo da mão esquerda, as cordas que são ou não são tocadas, e as cordas onde estão posicionados os dedos da mão direita.

Esta indicação da posição dos dedos da mão direita no evento de posição do acorde serve como base para a aplicação de um ritmo aos acordes. Cada ritmo é composto por um conjunto de eventos de nota rítmica, que efetivamente disparam o som do violão, realizando o acompanhamento.

Objetos musicais executáveis

A *informação musical* que será utilizada pelo seqüenciador é nada mais do que os mesmos eventos acima, carregados em memória a partir dos arquivos DV3. São o que chamamos de *objetos musicais executáveis* que, por terem estampados em si os tempos em que devem ser disparados, permitem descrever uma música. Na Figura 5.20, é exibido um conjunto de eventos que servem de exemplo.

Tipo	Instante (compasso: semí-fusas)	Evento	...
Início de Ritmo (RitmoOn)	0:00	"Samba"	...
Início de Acorde (AcordeOn)	0:00	"A7M"	...
Início de nota (NoteOn)	0:00	G4	...
Final de nota (NoteOff)	0:16	G4	...
Início de nota (NoteOn)	0:16	C5	...
Início de nota (NoteOn)	0:16	E5	...
Final de nota (NoteOff)	0:32	C5	...
Final de nota (NoteOff)	0:32	E5	...
Início de Acorde (AcordeOn)	0:32	"Bm7"	...
Início de nota (NoteOn)	0:32	B4	...
...
Final de nota (NoteOff)	85:00	G4	...

Figura 5.20 – Conjunto de eventos de diversos tipos com indicação de tempo em DV3

O acompanhamento, por exemplo, é formado pela aplicação de eventos de nota rítmica em acordes devidamente posicionados. Posições estas que, por sua vez, são trocadas no momento de disparo dos eventos de posições de acordes. Quando tais eventos são seqüenciados, é obtido exatamente o efeito da movimentação das mãos esquerda e direita do músico, e o acompanhamento do violão toma forma (Figura 5.21).

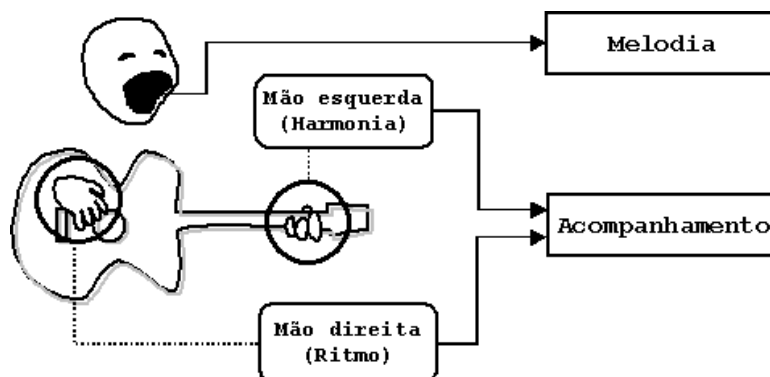


Figura 5.21 – Acompanhamento e Melodia

Cada nota do ritmo dispara uma nota no acorde. O ritmo é aplicado repetida e ciclicamente durante a música, resultando no acompanhamento. Na Figura 5.22, pode ser

visto este processo. O acompanhamento gerado está representado na figura através de uma tablatura.

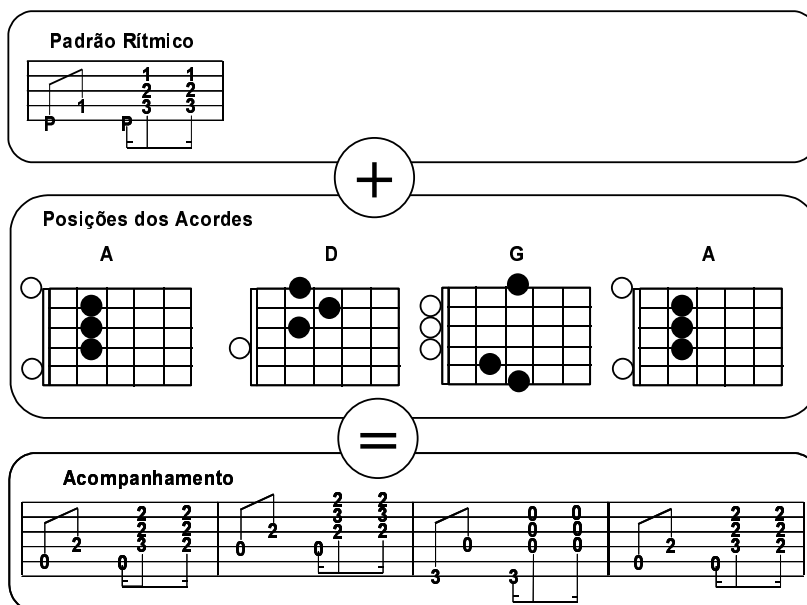


Figura 5.22 – Aplicação de um ritmo a um conjunto de acordes, gerando um acompanhamento.

5.4.2. Seqüenciador

O seqüenciamento sincronizado dos eventos de todos os tipos citados efetivamente executa a música, provendo o efeito gráfico (animação) e sonoro (música) desejados. Esta sincronização é feita através de um componente chamado *seqüenciador* (Roads 1996).

O *seqüenciador* é um relógio que acessa estes objetos musicais e os executa no tempo especificado. Esta execução necessita ser bastante precisa, uma vez que um ser humano consegue perceber variações mínimas na execução de uma música, na ordem de milissegundos, o que exige um sistema de tempo real eficiente.

Representação do tempo

O *D'Accord Guitar* utiliza uma representação de tempo diferente da utilizada nos arquivos MIDI. Em MIDI, o andamento é definido como o número de *batidas por minuto* (bpm), e cada batida é dividida em um número variável de *ticks*, que é a unidade de tempo utilizada para temporizar os eventos (MMA 1996). Na Figura 5.23, pode ser vista uma comparação entre estas unidades de tempo. Na parte superior da figura é exibida uma partitura. Abaixo, são mostrados os instantes em que cada evento existente na partitura é executado, através de 3 representações de tempo.

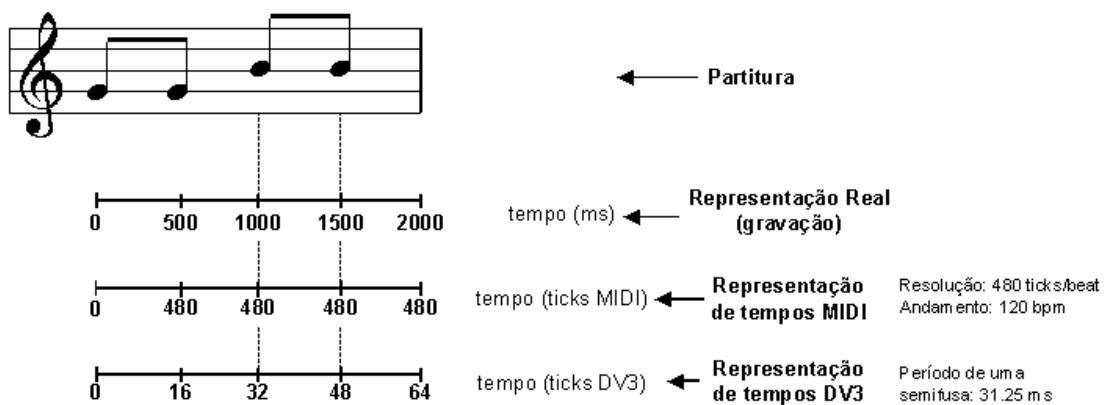


Figura 5.23 – Comparação das unidades de medida de tempo

No caso da Figura 5.23, os eventos estão uniformemente espaçados, tendo duração de 500 ms cada. Com uma resolução de 480 *ticks/batida* e um andamento de 120 *batidas/minuto*, cada evento está espaçado por pouco mais de 1 ms. Este valor é calculado através da fórmula abaixo:

$$\text{tempo de duração de cada tick (em ms)} = \frac{60.000}{\text{andamento} \times \text{resolução}}$$

Em MIDI, o número de *ticks por batida* pode chegar a 960. Como o andamento mais rápido possível é de 250 bpm, os eventos podem estar separados por um intervalo de tempo de até 0,25 ms. No *D'Accord Guitar*, o número de *ticks* existentes em uma batida é fixo, sendo definido como a divisão suficiente para representar uma semífusa. Os eventos no *D'Accord Guitar* podem estar separados por até 15 ms. Este valor como o período do seqüenciador do *D'Accord Guitar* parece ser razoável, pois embora a resolução MIDI seja bem mais alta, os seqüenciadores comumente não tocam os eventos de maneira tão precisa, pois se baseiam em relógios com período de 10 ms. Desta forma, é obtida uma pequena diferença real para os seqüenciadores MIDI tradicionais.

Como o número de *ticks* por batida é fixo no *D'Accord Guitar*, a variação de andamento é bastante simples de implementar, bastando alterar o período do ciclo do seqüenciador. Quanto maior o período, mais lenta a música se torna. Quanto menor o período, mais rápida a música se torna. No caso da Figura 5.23, como é escolhido um andamento de 120 *batidas/minuto*, o período do relógio é de 31.25 ms.

Embora não seja utilizada uma representação MIDI, este relógio foi implementado usando uma biblioteca MIDI, chamada *Maximum MIDI* (Messick 1997), obtendo uma precisão que satisfaz o nível de exigência do ouvido humano.

5.4.3. Funções Inteligentes

O *D'Accord Guitar* possui uma base de conhecimento que modela o conhecimento do violonista quanto à teoria e à técnica musical, na forma de regras, restrições, exemplos e heurísticas. A seguir, será descrito como estas regras, restrições e heurísticas foram utilizadas para modelar o raciocínio do violonista nas 3 atividades citadas anteriormente (transposição, transcrição e interpretação).

Transposição

Quando é feita uma transposição no *D'Accord Guitar*, os seguintes elementos são alterados: os acordes (tanto a cifra quanto a posição), a música cifrada (o texto da letra e as cifras dos acordes), a melodia da música e os solos existentes. Como existe a separação entre os elementos da música, cada um pode ser transposto individualmente. Quanto à melodia, a solução é trivial, bastando transpor cada nota MIDI executada (Figura 5.24). Quanto aos demais elementos, a solução é discutida a seguir.

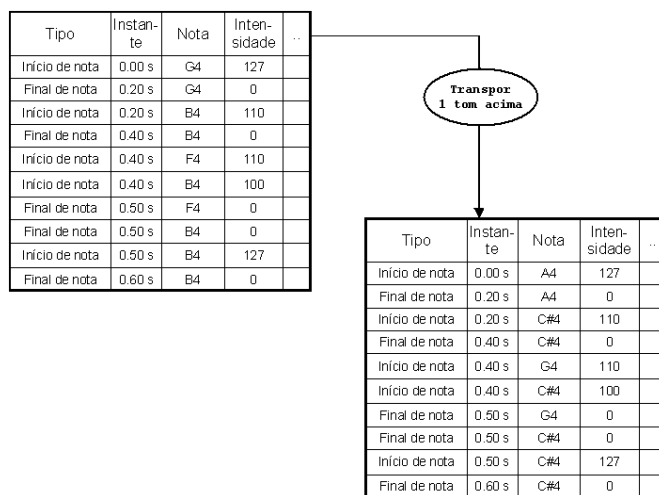


Figura 5.24 – Transposição de uma melodia MIDI

Transposição de Texto

Quanto à transposição do texto da música cifrada, é preciso alterar as cifras dos acordes (Figura 5.25). Para isto, é preciso separar, primeiramente, o que é a cifra e o que é a

letra da música dentro do texto. Esta separação será discutida no próximo capítulo, pois é muito importante no processo de edição.

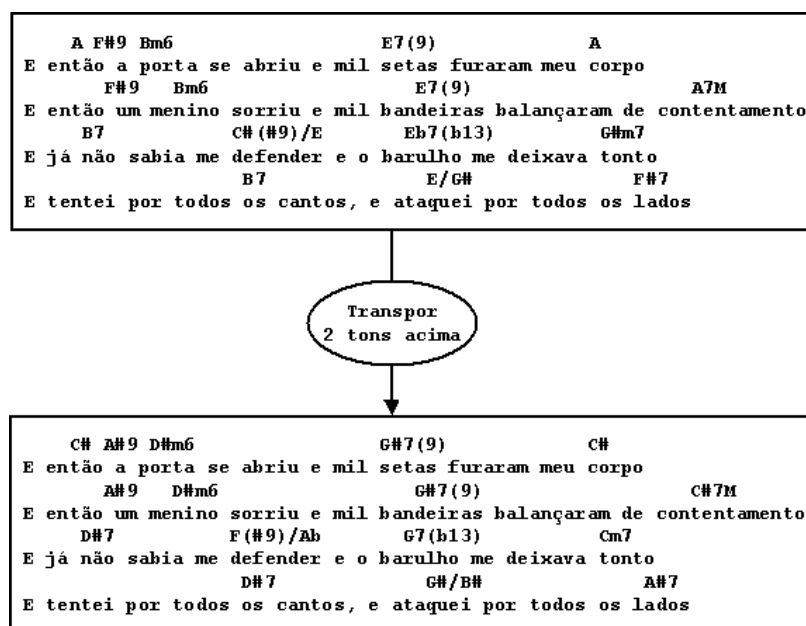


Figura 5.25 – Efeito da transposição de texto

Depois de reconhecidas as cifras dos acordes dentro da música cifrada, é preciso alterá-las, trazendo 2 problemas. O primeiro trata da escolha das cifras a utilizar. Por exemplo, dado um acorde de A7M transposto 6 semitons acima, qual cifra deve resultar: D#7M ou Eb7M? A solução encontrada foi incluir no cabeçalho do arquivo a tonalidade da música. De posse da tonalidade, é possível definir qual cifra se adequa melhor, de acordo com as regras da teoria musical (Cambouropoulos 2001).

O segundo problema trata do deslocamento da cifra no texto. Por exemplo, ao transpor o acorde Am/C em 1 semitom, ele torna-se um A#m/C#. Com isso o número de caracteres é alterado, o que interfere tanto na posição do acorde sobre a letra quanto nas coordenadas utilizadas pelos eventos de seleção de letra. O sistema precisa então verificar o ponto na letra em que cada cifra ocorre e alterar o texto da música cifrada. Da mesma forma, precisa atualizar todos os eventos de seleção de letra.

A	F#9	Bm6		E7(9)		A					
E	então	a	porta	se	abriu	e	mil	setas	furaram	meu	corpo

C#	A#9	D#m6		G#7(9)		C#					
E	então	a	porta	se	abriu	e	mil	setas	furaram	meu	corpo

Figura 5.26 – Deslocamento das seleções de letra, efetuado na transposição

A transposição do texto de uma música cifrada é, *per si*, uma aplicação de bastante interesse pela comunidade musical.

Transposição de Posições de Acordes

Quanto à transposição das posições de acordes, o problema é ainda mais complexo. A principal dificuldade é que no violão existem inúmeras maneiras de se posicionar cada acorde. Desta forma, é requerido um processo decisório para escolher quais as novas posições dos acordes. Este problema é justamente o encontrado no processo de interpretação e, portanto, será exposto neste tópico.

Transposição de Solos

A dificuldade encontrada na transposição de solos, assim como na transposição de posições de acordes, deve-se à possibilidade de posicionar uma nota em mais de uma coordenada no violão. Para transpor um solo, o *D'Accord Guitar* transforma-o inicialmente em uma melodia, suprimindo as informações específicas do violão. Esta melodia é transposta (um processo direto) e depois reconvertida para o violão, utilizando a função de transcrição de solos, que será explicada a seguir. Este modelo pode ser observado na Figura 5.27.

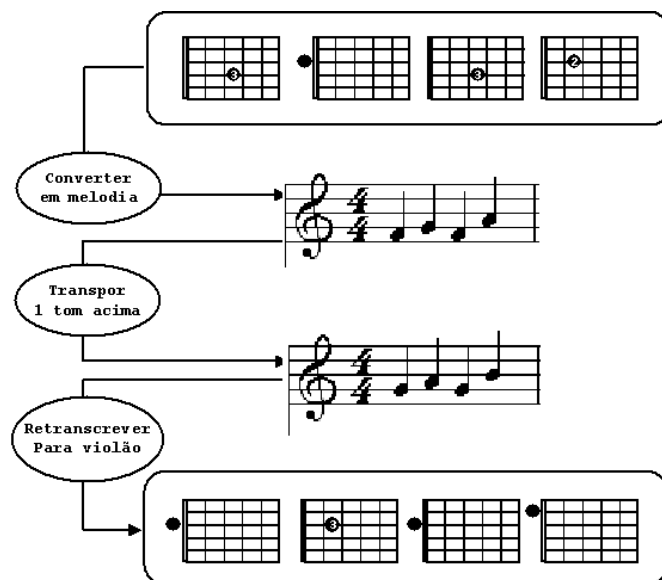


Figura 5.27 – Processo de transposição de solos

Transcrição de Solos

Este problema consiste em gerar uma execução no violão, com informações de que dedos utilizar em que casas e cordas, baseado em informações MIDI (Cambouropoulos 2000). Ele pode ser visto, também, como o problema de transcrever uma partitura para o violão (Klapuri 1998). Este problema já foi estudado por diversos grupos de pesquisa. No *D'Accord Guitar* foi adaptada uma solução existente na literatura científica (Sayegh 1989; Cordier 1995).

Esta solução funciona de maneira simples. Dada uma sequência de notas, inicialmente são calculadas as possíveis coordenadas no violão para tocar cada nota. É gerado, portanto, um grafo (Manber 1994), e deseja-se encontrar a melhor sequência de coordenadas. É definido como custo entre duas coordenadas a soma da distância entre as suas casas com a distância entre suas cordas.

Interpretação Harmônica

A geração de um acompanhamento baseado em uma grade de cifras de acordes é alvo de um considerável número de estudos (e.g. Ramalho 1997). No entanto, o acompanhamento específico para violão é completamente diferente. Ao invés de levar em consideração uma sequência de notas, leva em consideração uma sequência de coordenadas no violão, mudando completamente o espectro do problema (Figura 5.28). De forma mais

detalhada, as funções de interpretação harmônica foram projetadas para resolver problemas como digitação das mãos direita e esquerda, escolha de vozes e posicionamento de acordes.

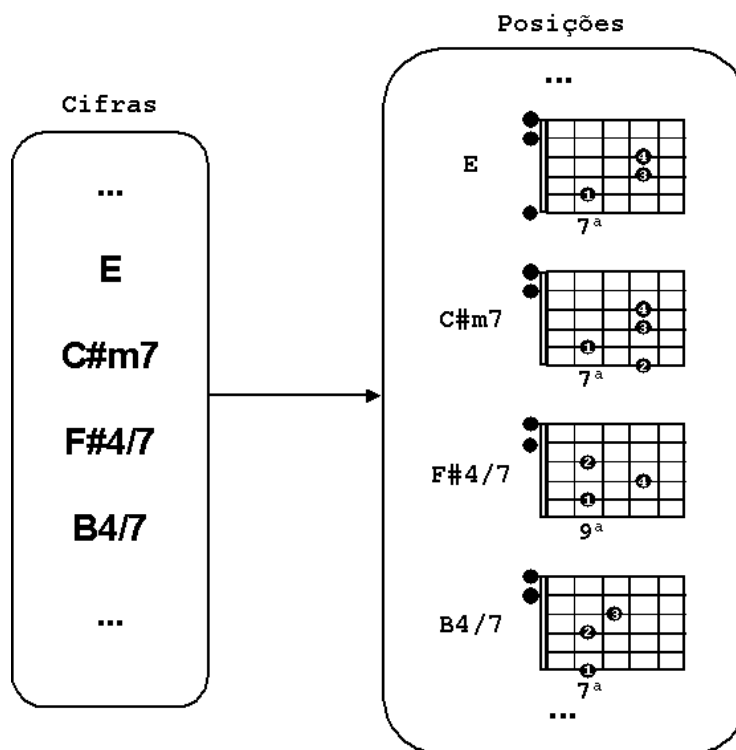


Figura 5.28 –Interpretação de cifras para o violão

Para realizar tais tarefas, é preciso elicitar o conhecimento existente entre os músicos, pois existe um grande conhecimento em torno das escolhas de posições no violão, que transcende a busca normal por escolhas de vozes satisfatórias e linhas melódicas. Ele é fruto da experiência e da maturidade do músico, e muitas vezes envolve gostos pessoais e critérios culturais (Fowler 1984b). Tal conhecimento, entretanto, ainda não foi trabalhado computacionalmente.

Por este aspecto inovador, as funções de interpretação harmônica configuram-se como a maior contribuição científica deste trabalho. Por sua relevância, elas serão descritas em um capítulo à parte (capítulo 8).

5.5. Discussão

Algumas características podem ser observadas para fazer uma comparação entre o *D'Accord Guitar Player* e os demais *Sistemas de Simulação Instrumental*.

Em primeiro lugar, a informação no *D'Accord Guitar Player* é mais rica e apropriada ao instrumento. Os SSIs atuais baseiam-se em arquivos MIDI, ou em pequenas variações dos mesmos, e portanto limitam-se à informação de qual a frequência da nota tocada. O *D'Accord Guitar Player* utiliza um formato próprio, que permite uma descrição muito mais completa, indicando os dedos que executam cada nota, por exemplo.

Em segundo lugar, o *D'Accord Guitar Player* permite a exibição de mais informações que os SSIs tradicionais, através de um instrumento virtual mais visível. Na Figura 5.29 isto pode ser visto. No *The Jazz Guitarist* (imagem inferior), é impossível exibir qualquer outro elemento que não seja a coordenada tocada, dado o tamanho reduzido da área de trabalho.

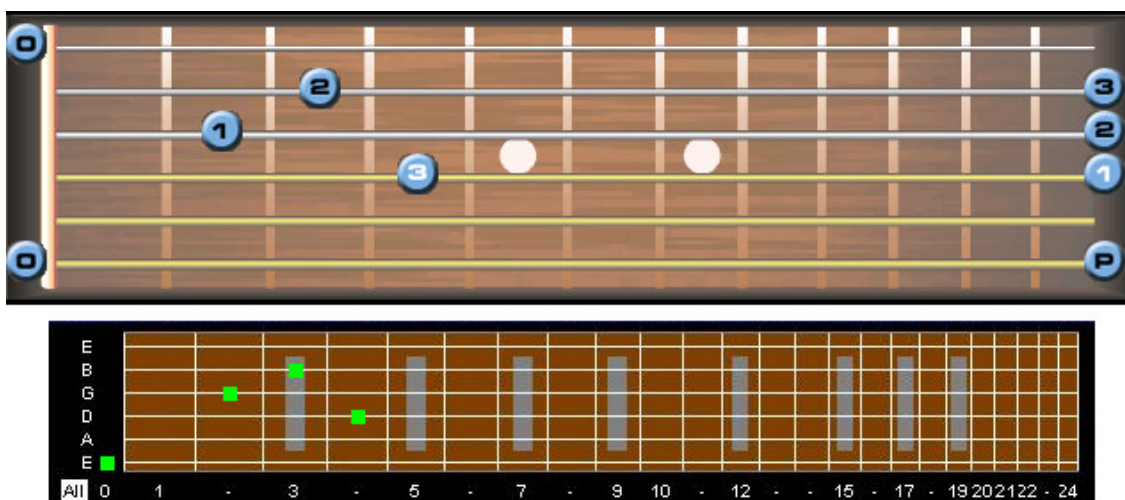


Figura 5.29 – Exibição da execução em um SSI convencional (*The Jazz Guitarist*, abaixo) e no *D'Accord Guitar Player* (no alto).

O formato do arquivo utilizado e o projeto de interface gráfica servem, portanto, como diferenciais do *D'Accord Guitar Player*, por permitir uma descrição mais detalhada e precisa da execução registrada. Além disso, há diversas funcionalidades existentes em outras aplicações que foram incorporadas, como o karaokê e os controles de repetição. O *D'Accord Guitar Player* diferencia-se, ainda, ao inserir novas funcionalidades, como a transposição e a interpretação melódica.

O principal diferencial, contudo, é separação apropriada dos diversos elementos musicais, tornando o processo de leitura como um todo muito mais simples. Em especial, a exibição clara da harmonia facilita o entendimento e a memorização da execução musical. Da

mesma forma que no capítulo 4, pode ser feita uma comparação entre a exibição da execução em um SSI convencional e no *D'Accord Guitar Player* (Figura 5.30).

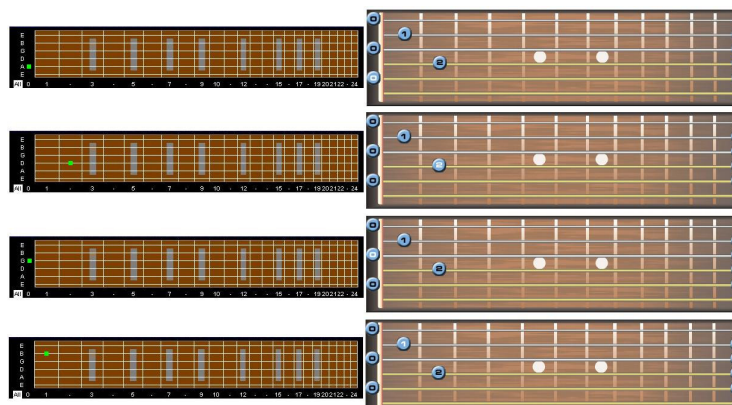


Figura 5.30 – Exibição da execução violonística em um SSI convencional (*The Jazz Guitarist*, à esquerda) e no *D'Accord Guitar Player* (à direita)

No *D'Accord Guitar Player* há a exibição da harmonia subjacente. Em outras palavras, são exibidas não só as notas sendo tocadas, mas as posições montadas, mesmo que parte delas não esteja sendo tocada no momento. Esta forma de exibição permite que o usuário, além de simplesmente ver “o que” está sendo tocado, veja “como” está sendo tocado. A exibição das posições ajuda a compreender de forma muito mais satisfatória a maneira como uma determinada sequência de notas deve ser tocada, e são um salto extraordinário no processo de leitura e aprendizagem.

Por fim, o processo de sugestão de posições é inovador. Durante as pesquisas da literatura específica e dos softwares comerciais existentes, não encontramos aplicações que resolvessem tal problema de maneira séria.

6. D'Accord Guitar Editor

O *D'Accord Guitar Editor* serve para gravar músicas no formato DV3. Ele possui várias funcionalidades em comum com o *D'Accord Guitar Player*, mas a seguir serão vistas apenas as funções que são específicas de gravação.

6.1. Interface e funcionalidades

A interface do *D'Accord Guitar Editor* é mais complexa que a dos demais. A janela inicial é praticamente idêntica à do *D'Accord Guitar Player*, como pode ser visto na Figura 6.1. A partir do botão *gravar* aparece a janela de gravação.

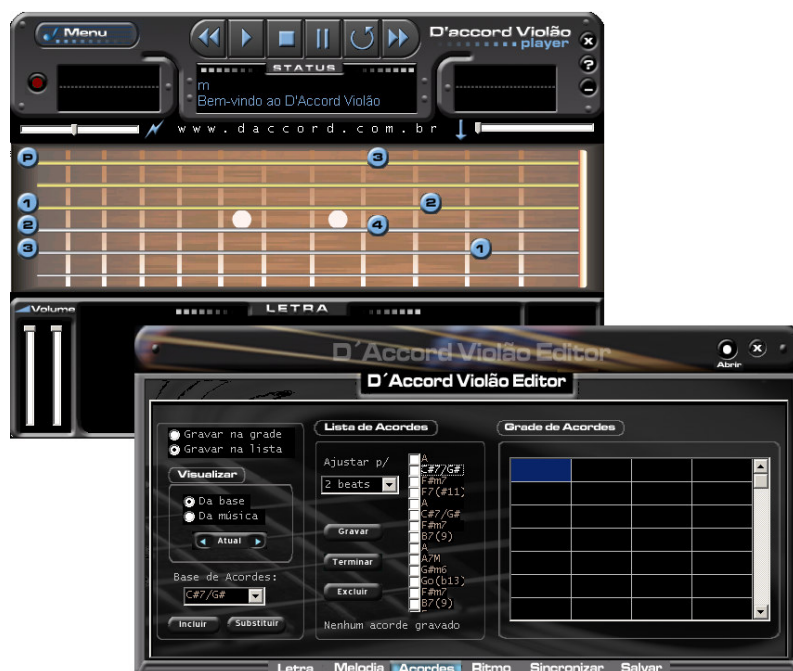


Figura 6.1 – Janela inicial e de gravação do D'Accord Guitar Editor

Esta janela de gravação alterna entre diversos modos, cada um para gravar ou editar um determinado conjunto de elementos. Estes elementos podem teoricamente ser gravados em qualquer ordem. Faz sentido, contudo, obedecer a certas regras. Por exemplo, a edição da música cifrada antes da gravação da harmonia e da letra. Da mesma forma, é importante gravar a melodia antes da letra, para servir de base para a sincronização.

Um primeiro elemento a editar é a música cifrada. Ela é escrita ou copiada em uma caixa de texto, seguindo a sintaxe descrita no padrão DV3 (especificada na seção 6.3.4). A partir desta janela, pode-se exportar a letra e o conjunto de acordes a serem gravados. Na

Figura 6.2 pode ser vista a janela de edição da música cifrada. No lado esquerdo, há a caixa de edição, e à direita os botões para a exportação da letra e das cifras.

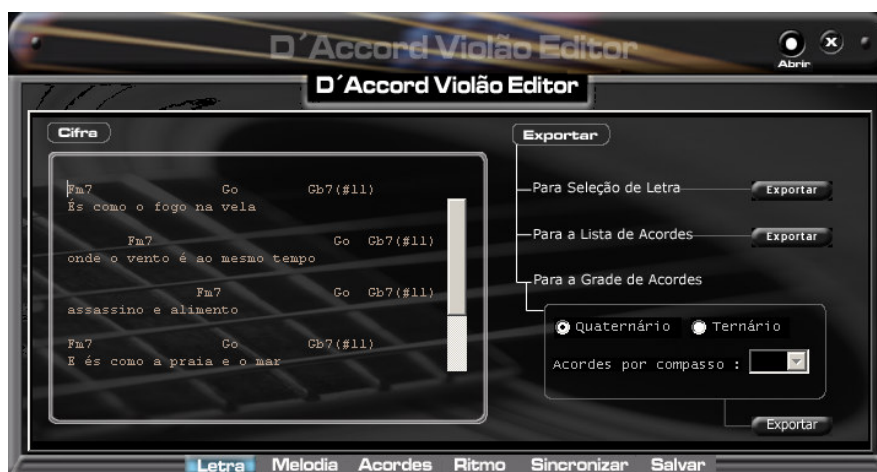


Figura 6.2 – Tela de edição da música cifrada

Um segundo elemento a gravar é a melodia da música. Ela é gerada a partir de um arquivo MIDI ou gravada diretamente de um instrumento MIDI. Caso seja escolhida a geração a partir de um arquivo, uma das trilhas é escolhida como trilha de voz. Esta trilha será utilizada para sincronização com a letra da música. O resto do arquivo MIDI, todavia, pode ser incorporado ao DV3, se desejado.

Na Figura 6.3 é exibida a janela de gravação da melodia. Os botões da parte superior servem para escutar o arquivo MIDI. Os controles na parte inferior servem para escolher as trilhas associadas à letra e as trilhas que serão apenas escutadas.



Figura 6.3 – Tela de edição da melodia

Um terceiro elemento a gravar é a harmonia. Dada a lista de acordes, capturada da música cifrada, é preciso indicar o tempo em que cada um ocorre e sua posição no violão. Os tempos em que ocorrem são gravados através do teclado, em tempo real, ou editados em uma grade de acordes. Quanto às posições, o usuário pode escolher diretamente na base, ou pode utilizar a sugestão oferecida pelo sistema.

A Figura 6.4 exibe a janela de gravação de acordes. Na parte central há a lista de acordes importados da música cifrada. Caso o músico deseje editar o tempo em que eles ocorrem, utilizará a *grade de acordes*, à direita. Caso queira gravar em tempo real, selecionará o botão *gravar*.

Caso o usuário queira especificar manualmente as posições dos acordes, ele pode utilizar a base de acordes. Neste caso, será aberta uma janela igual à do *D'Accord Guitar Browser* (capítulo 7), onde o usuário poderá interagir ou navegar na base para escolher uma posição. Caso deseje que o sistema sugira posições, estas serão baseadas na função de interpretação harmônica. Esta opção será discutida em detalhes em capítulo próprio (capítulo 8), em virtude de sua relevância.

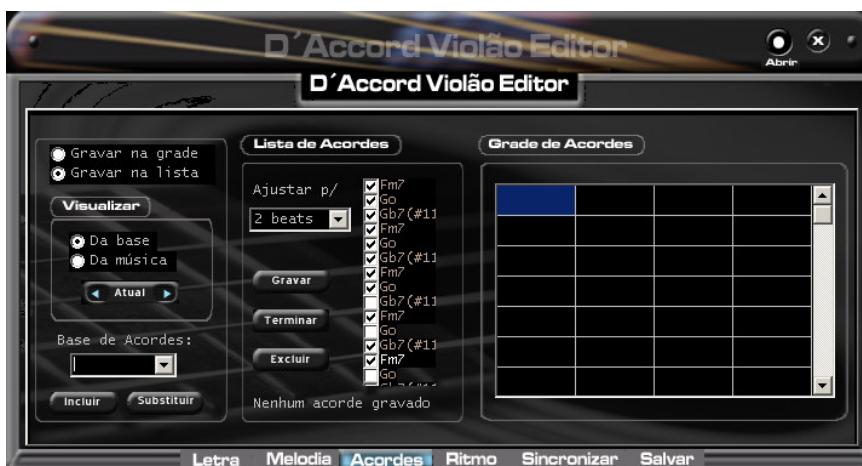


Figura 6.4 – Tela de edição da harmonia

O próximo elemento a gravar é o ritmo. Cada ritmo pode ser gravado ou simplesmente escolhido da base. Caso ele seja escolhido na base, é possível escutar uma amostra de como ficará o resultado final. Caso seja gravado, isto é feito através do teclado do computador ou de um instrumento MIDI. Há, ainda, a possibilidade de gravar a execução inteira da música, simulando a execução real da mão direita do músico, ao invés de um ritmo a ser aplicado repetidamente.

Além disso, o usuário pode utilizar mais de um ritmo em uma mesma música, precisando gravar, neste caso, as trocas de ritmo. Na Figura 6.5 é exibida a janela de gravação de ritmos. O usuário pode preferir utilizar ritmos pré-definidos na base de ritmos (do lado esquerdo) ou gravar seus próprios ritmos (ao centro). No lado direito, são apresentados todos os ritmos disponíveis para serem utilizados.



Figura 6.5 – Tela de gravação do ritmo

Um quinto elemento a gravar é a animação da letra (karaokê). Há duas maneiras de gravá-la. Na primeira, ela é gravada em tempo real. Na segunda, a letra da música é sincronizada com a melodia. No primeiro caso, as seleções de letra são inicialmente delimitadas pelo usuário. Depois, o tempo em que cada separação ocorre é gravado em tempo real. Na segunda, a cada separação feita, sua respectiva nota na melodia é tocada. A Figura 6.6 exibe a janela de edição dos trechos de letra a serem selecionados, onde as barras "/" delimitam os trechos.

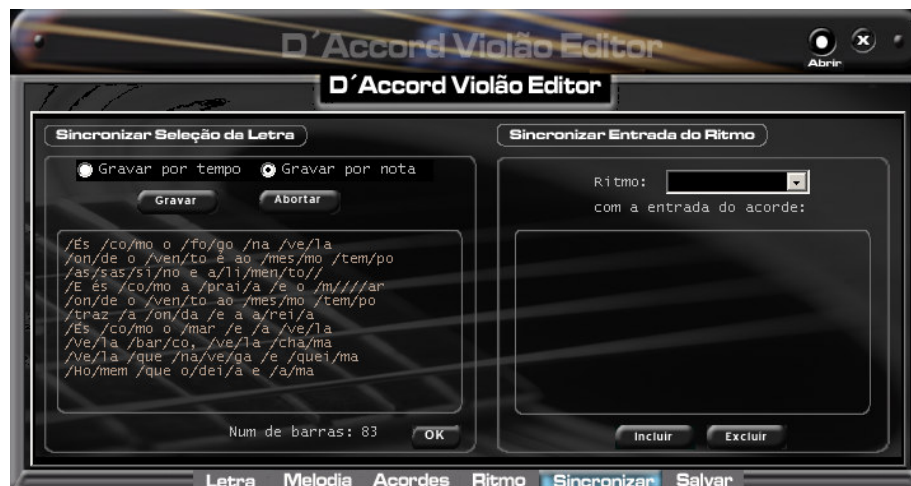


Figura 6.6 – Tela de edição da letra (karaokê)

Um último elemento a gravar são as informações de direitos autorais. É possível ao usuário preencher um campo com informações sobre a música, como título, gênero, autor, intérprete, ano em que foi gravado, etc. Com base nestes dados, pretende-se implementar um mecanismo de busca para arquivos DV3. Na Figura 6.7 é exibida a tela onde são solicitados dados sobre a música.

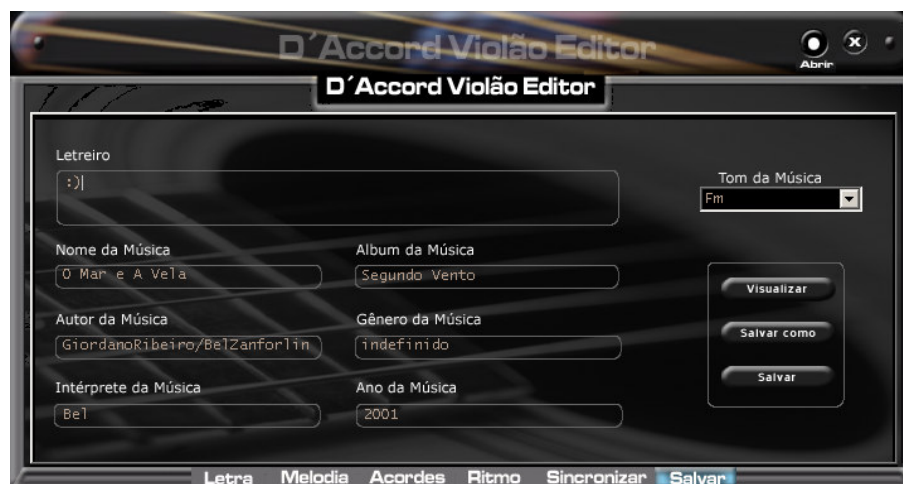


Figura 6.7 - Tela de edição das informações sobre a música

6.2. Arquitetura

A arquitetura do *D'Accord Guitar Editor* é semelhante à do *D'Accord Guitar Player*, com um seqüenciador que, ao invés de preocupar-se em tocar uma determinada informação musical no tempo certo, preocupa-se em detectar os instantes em que os eventos musicais são tocados. Esta arquitetura é mostrada na Figura 6.8.

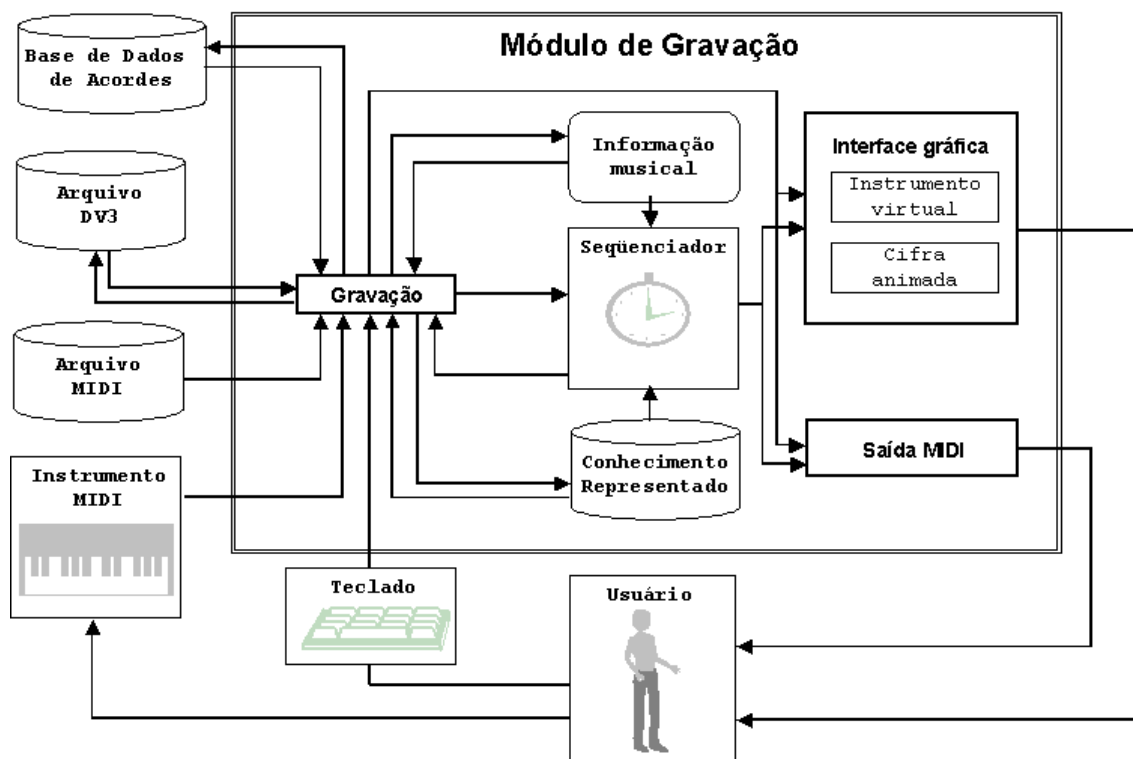


Figura 6.8 – Arquitetura do D'Accord Guitar Editor

Para efeito de simplificação, a Figura 6.8 mostra apenas as funções específicas de gravação, e o modo como elas interagem com os demais componentes. O processo de gravação no *D'Accord Guitar* resume-se à gravação em separado de cada elemento musical, seguindo o processo descrito na Figura 6.9. Nos tópicos a seguir, será descrito o processo de gravação de cada elemento.

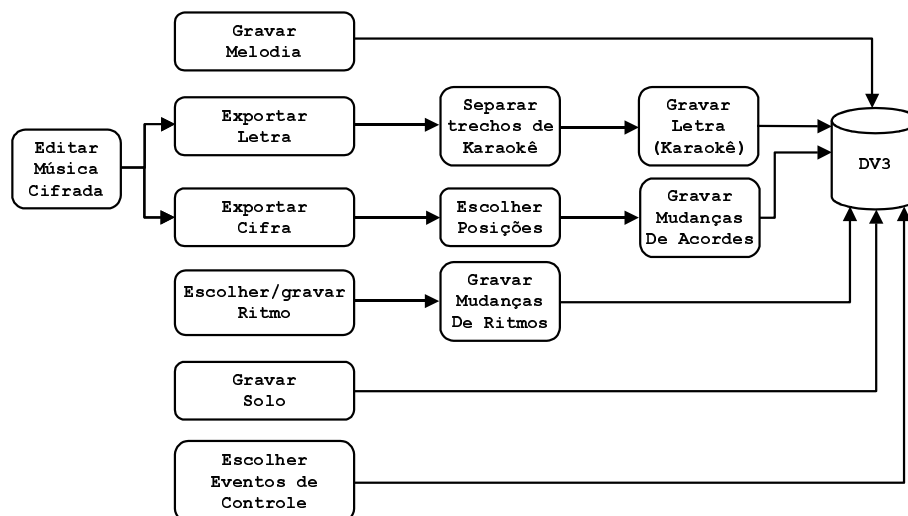


Figura 6.9 – Processo geral de gravação

6.3. *Concepção e implementação*

Os problemas encontrados no *D'Accord Guitar Player* (representação musical e seqüenciamento) também existem no *D'Accord Guitar Editor* e, portanto, não serão apresentados novamente (ver capítulo 5). A seguir serão descritos apenas os problemas específicos a este modo de interação. Essencialmente, há problemas envolvidos com a criação de uma *interface de gravação*, com a execução em *playback* do que já foi gravado enquanto grava-se um novo elemento, com a *reedição* do que já foi gravado ou editado e com o tratamento do texto.

6.3.1. **Interface de Gravação**

O problema de interface de gravação é complicado. Cada elemento tem uma forma diferente de ser gravado. Além disso, cada elemento é gravado separadamente. Como sincronizá-los, então? Nós observamos duas possibilidades: edição através de um ambiente gráfico ou gravação direta.

Caso seja provida uma edição gráfica, é necessária uma correlação da representação gráfica com alguma notação conhecida, uma boa interface de entrada de dados, e a utilização de uma unidade de medida de tempo que sirva de referência à sincronização dos eventos. A unidade adotada no *D'Accord Guitar* foi a de compassos. Isto quer dizer que cada evento é colocado em uma posição específica dentro de algum compasso, como visto na Figura 6.10, abaixo.

Compasso	1ª batida	2ª batida	3ª batida	4ª batida
1	Andamento: 120 bpm Ritmo: Samba Acorde: A7M (1/1)	(2/1)	Acorde: Bm7 (3/1)	(4/1)
2	Acorde: E7 (1/2)	(2/2)	Acorde: A (3/2)	(4/2)
3	Ritmo: Samba2 Acorde: A7 (1/3)	(2/3)	Acorde: D (3/3)	Acorde: Bm7 (4/3)
4	Acorde: E7 (1/4)	(2/4)	Acorde: A (3/4)	(4/4)
5	Ritmo: Samba Acorde: A7M (1/5)	Acorde: A7 (2/5)	Acorde: D (3/5)	Acorde: E7 (4/5)
6	Andamento: 120 bpm Acorde: A (1/6)	(2/6)	(3/6)	(4/6)

Figura 6.10 - Grade de compassos e os eventos que ocorrem em cada um deles

Há, ainda, uma outra forma de indicar o instante em que um evento ocorre, que é através da sua correlação com outros eventos. Por exemplo, existe uma correspondência entre os eventos de seleção de letra e os eventos de melodia. Portanto, é possível associar cada seleção de letra a uma nota da melodia (Figura 6.11), e não a um compasso.

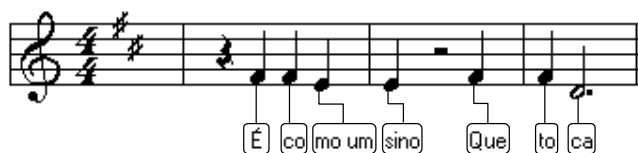


Figura 6.11 - Associação entre a melodia e as seleções de letra utilizadas para o karaokê

Adicionalmente, o músico pode “marcar” os instantes em que cada evento ocorre de forma mais prática e intuitiva, através da gravação direta, em tempo real, com os elementos previamente gravados sendo tocados em *playback*. Esta execução em *playback*, no entanto, pode acarretar alguns problemas, que serão vistos numa seção própria (6.3.2).

Cada elemento musical utiliza uma notação diferente. A seguir será descrito como cada elemento é gravado.

Melodia

A melodia é uma sequência de notas MIDI, com a indicação dos instantes em que ocorrem. Para obtê-la, é possível gravar a partir de um instrumento MIDI ou importar de um arquivo MIDI padrão.

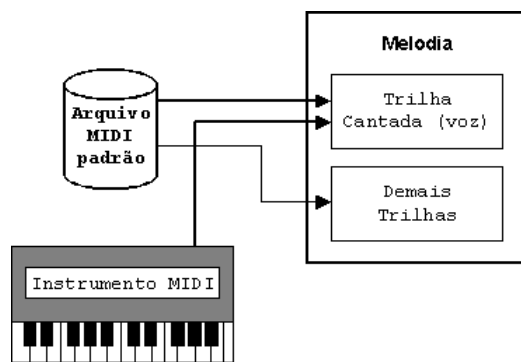


Figura 6.12 – Processo de gravação da melodia

É possível, inclusive, importar um arquivo MIDI completo (contendo, por exemplo, melodia e acompanhamento de bateria, baixo, piano, etc.). Desta forma, é escutado todo o arranjo da música, dando um efeito mais realista à sua execução e permitindo reconhecê-la melhor. Neste caso é necessário escolher qual das trilhas corresponde à letra da música, para posteriormente sincronizá-la com a animação da letra.

Letra

Há dois elementos diferentes referentes à edição da letra da música: a música cifrada e as seleções realizadas sobre a letra, que dão o efeito de animação (karaokê).

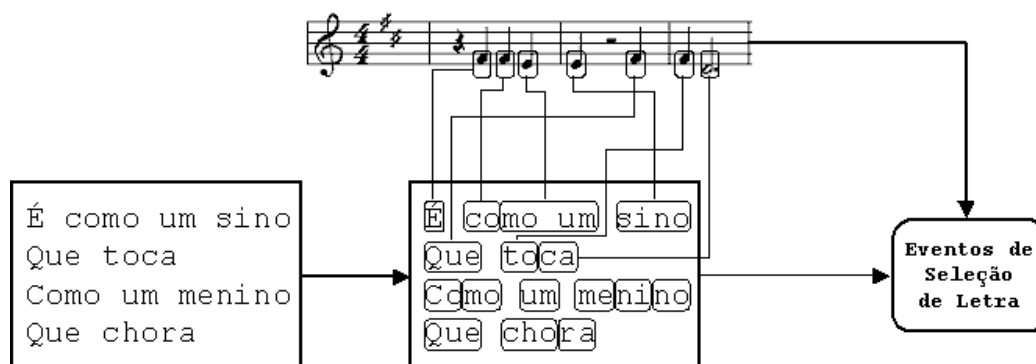


Figura 6.13 – Associação entre letra e melodia

A música cifrada é formada pela letra da música em conjunto com as cifras dos acordes utilizados. Esta música cifrada pode ser digitada ou copiada de algum arquivo pronto

(como uma página na Internet, por exemplo), sendo necessário apenas que ela obedeça ao padrão *D'Accord Guitar*. Obtendo a letra, é possível separar o que é cifra e o que é letra através de um *parser*, que será explicado em seção à parte (6.3.4). As cifras podem então ser exportadas para a gravação da harmonia. A letra é usada para a gravação do karaokê.

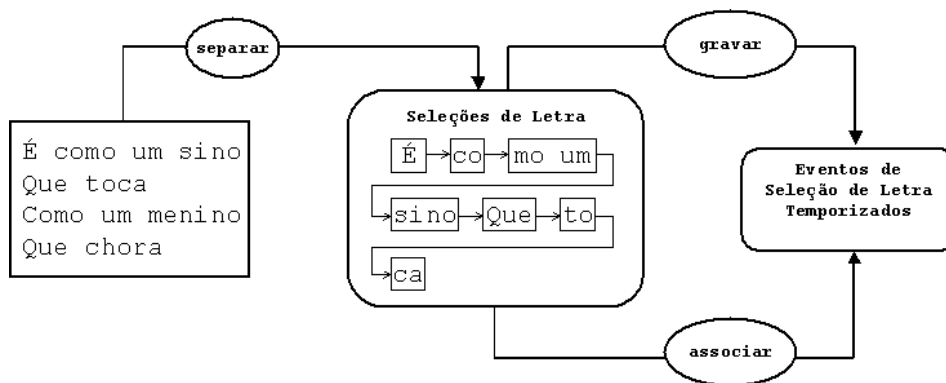


Figura 6.14 – Processo de gravação da animação da letra

Esta gravação acontece da seguinte forma: dada a letra de uma música, cada trecho a ser animado é separado por um caractere especial, como na Figura 6.15. Esta etapa demandou a criação de outro *parser*, que também será visto na seção 6.3.4.

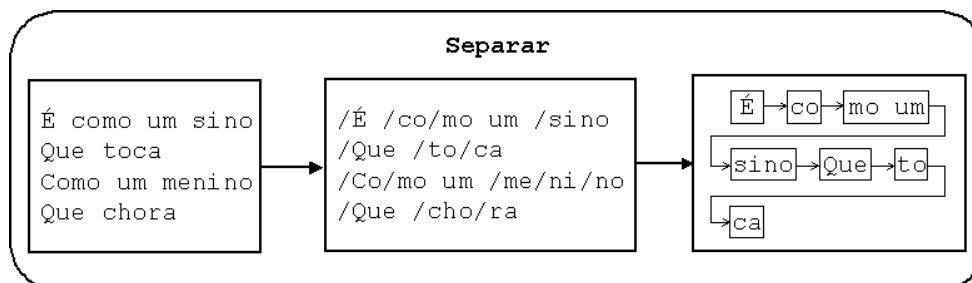


Figura 6.15 – Processo de separação de letra

O próximo passo trata de definir o instante em que cada um destes trechos aparecerá na animação. Isto é feito através da sincronização de cada trecho com a melodia, que pode ser feita de 2 formas. Na primeira, cada trecho é associado a uma nota (no exemplo da Figura 6.16, pode ser visto como é feita esta associação). Desta forma, é possível saber o instante em que cada trecho aparecerá na animação, de acordo com o instante em que cada nota é tocada na melodia.

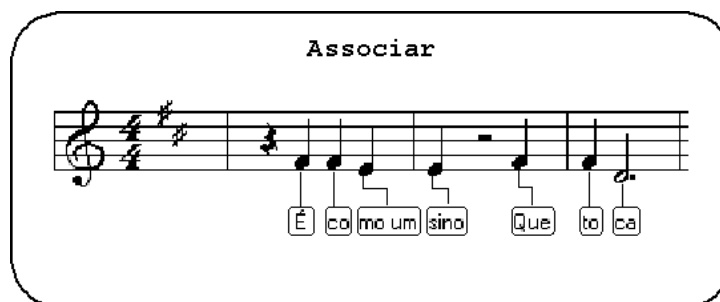


Figura 6.16 – Associação da letra com a melodia

Na segunda, os instantes são definidos através de gravação direta, em tempo real, através do teclado no computador. A cada toque do usuário, é marcado o instante corrente, que é então associado ao trecho em questão (Figura 6.17).

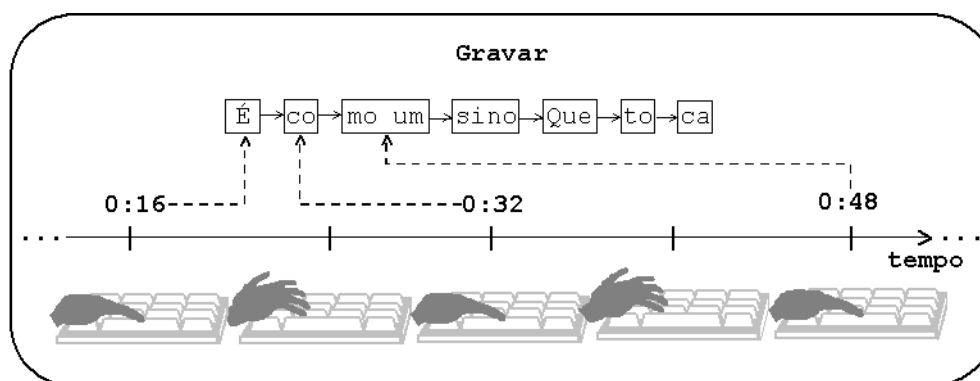


Figura 6.17 – Gravação em tempo real

Harmonia

A informação harmônica é formada por uma sequência de posições de acordes e pelos instantes em que ocorrem. Para obterem-se tais posições, o primeiro passo é definir a sequência de cifras de acordes da musica. Como explicado no tópico anterior, esta lista é importada a partir da música cifrada (o texto da letra, em conjunto com as cifras dos acordes). O próximo passo trata da escolha de posições para cada acorde.

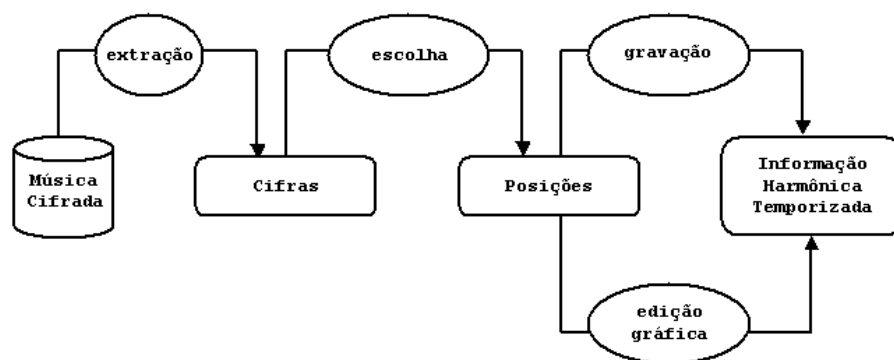


Figura 6.18 – Processo de gravação da harmonia

Dada esta sequência de posições de acordes, é preciso informar em que instante cada um deles ocorre. Em uma cifra, este instante é baseado no compasso em que o acorde ocorre. Em uma execução real, este instante pode ser medido também em segundos. No *D'Accord Guitar* estas 2 formas são possíveis.

Para indicar em que compasso o acorde entra, o usuário digita a sua cifra em uma tabela (Figura 6.19). Nela, cada linha representa um compasso e cada coluna delimita uma batida deste compasso. O conteúdo da célula indica o evento a ser disparado neste momento. Para indicar em tempo real o instante em que o acorde entra, é possível gravar cada troca de acorde, de forma similar à utilizada para gravar as animações de letra (Figura 6.20).

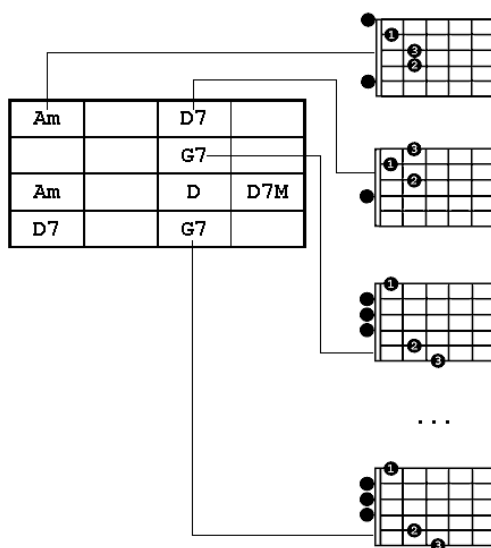


Figura 6.19 – Sincronização da harmonia através da grade de acordes

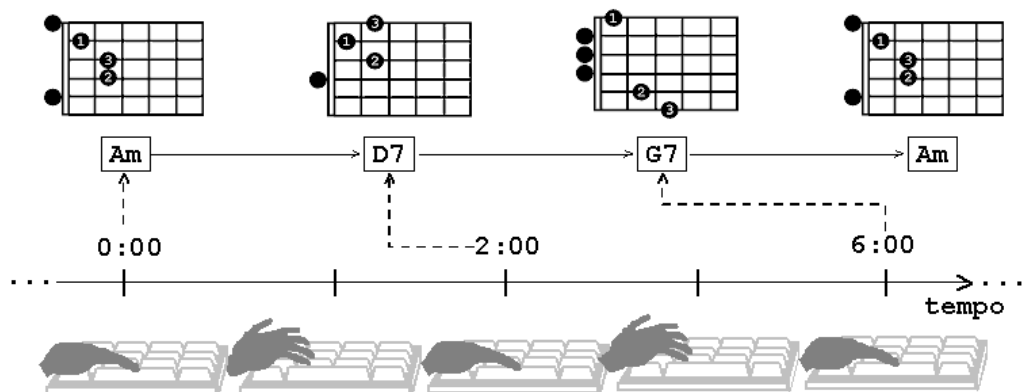


Figura 6.20 – Sincronização da harmonia pela gravação em tempo real

Ritmo

Um ritmo é composto por um conjunto de eventos referentes à execução da mão direita do músico. Este ritmo, quando aplicado às posições de acordes, gera o acompanhamento, como visto na Figura 6.21. Uma música no *D'Accord Guitar* pode conter vários ritmos diferentes que se alternem (Figura 6.21). Portanto, há duas informações a serem gravadas ou editadas: os ritmos propriamente ditos e as trocas de ritmos.

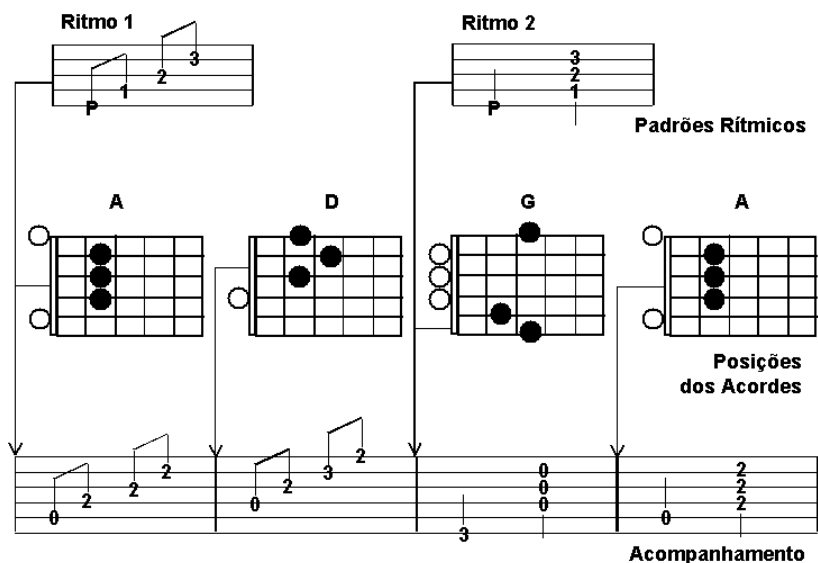


Figura 6.21 – Aplicação de vários ritmos dentro de uma música.

É possível escolher um ritmo dentro da base de dados ou gravá-lo em tempo real. A escolha de um ritmo da base é feita utilizando o *Rhythm Browser*. Com o *Rhythm Browser*, é mostrado um *preview* do ritmo escolhido. O *preview* é feito com a aplicação contínua do

ritmo a um acorde fixo. O usuário pode então ouvi-lo antes de aplicá-lo à sequência de posições.

Caso o usuário deseje efetuar uma gravação em tempo real, ela pode ser feita a partir de um instrumento MIDI (e.g., teclado ou guitarra) ou do teclado do computador (Figura 6.22).

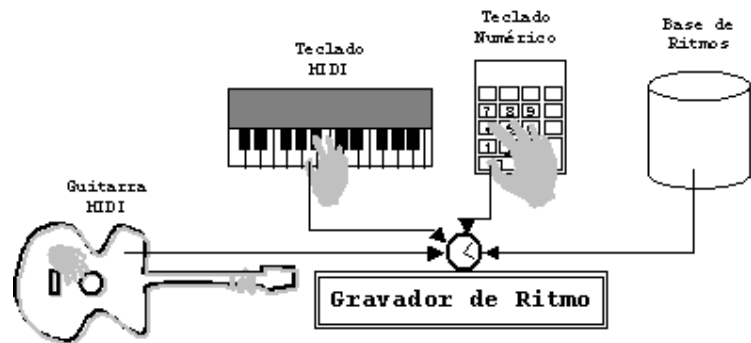


Figura 6.22 – Captação da informação rítmica.

Cada evento existente no ritmo informa o dedo da mão direita que está sendo utilizado, a intensidade que está sendo aplicada e o instante em que ocorre. A gravação estabelece uma analogia entre cada evento recebido e um dos dedos da mão direita. Isto é ilustrado na tabela abaixo.

	Polegar	Indicador	Médio	Anular
Teclado	C4	D4	E4	F4
Guitarra MIDI	5ª corda (A3)	4ª corda (D4)	3ª corda (G4)	2ª corda (B4)
Teclado do Computador	Tecla 0	Tecla 1	Tecla 2	Tecla 3

Tabela 2 – correlação entre os eventos e os dedos da mão direita

O processo de gravação de trocas de ritmo funciona de forma semelhante ao de trocas de acordes: o instante de cada troca pode ser gravado em tempo real ou indicado de acordo com o compasso em que ocorre (Figura 6.23).

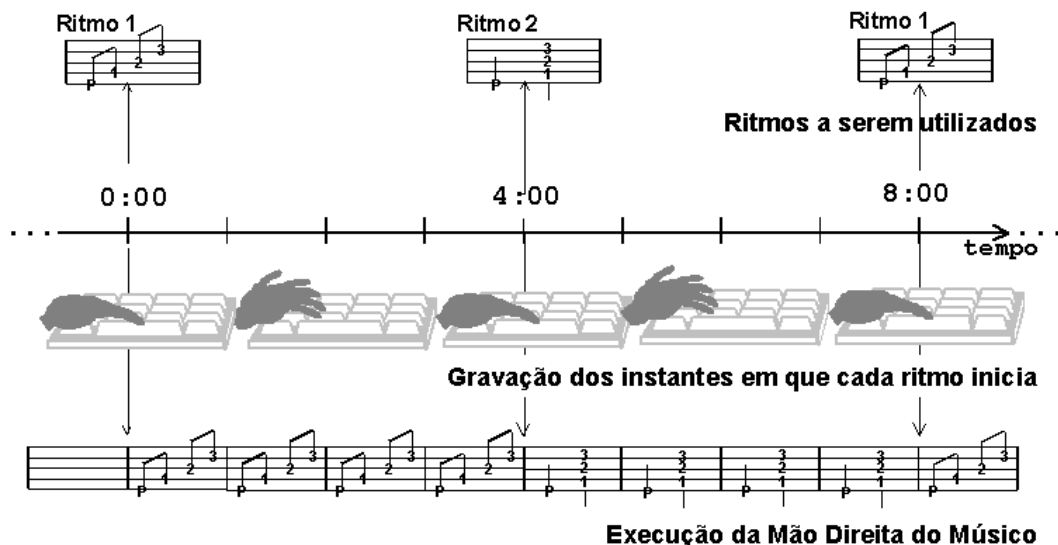


Figura 6.23 – Processo de gravação de trocas de ritmos

Entretanto, um mesmo ritmo (ou um pequeno conjunto de ritmos), quando aplicado durante a música inteira, pode deixar o acompanhamento monótono. O que convencionalmente se chama *suíngue* (*swing*) é justamente as alterações e variações feitas em torno de um determinado ritmo, o que é bastante difícil de ser realizado automaticamente (Ramalho & Ganascia 1994). Nos casos em que o usuário queira um ritmo mais variado, é possível gravar a execução do ritmo de uma música inteira, e não de um padrão a ser repetido ciclicamente. Isto fornece uma maior flexibilidade para quem vai editar a música, gravando a execução da mão direita da maneira que este quiser, efetuando variações e trocas de andamento de forma bastante prática.

Solo

O solo cuida da execução que foge ao acompanhamento. No *D'Accord Guitar*, ele é formado por um conjunto de coordenadas tocadas sequencial ou simultaneamente, e é definido como toda a execução onde não são configurados acordes. Embora tenhamos dado preferência à edição de músicas onde os acordes sejam bem formados, procuramos projetar uma solução flexível, que permitisse que partes de músicas que não se enquadrassem na cifragem de acordes fossem aceitas. Na Figura 6.24 pode ser visto um exemplo.

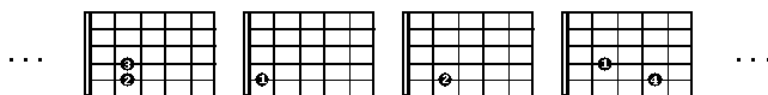


Figura 6.24 – Exemplo de trecho onde não há formação clara de acordes

A Figura 6.25 ilustra 3 possibilidades para a gravação de solos no *D'Accord Guitar*. No primeiro caso, cada evento de solo (na Figura 6.24, existem 4 acordes) é posicionado no braço interativo e os tempos em que eles ocorrem são gravados de forma análoga à gravação das trocas de acordes e ritmos. Alternativamente, o solo pode ser convertido diretamente de um arquivo MIDI padrão, usando o mesmo algoritmo de reconhecimento descrito no capítulo 5 (utilizado na função *alternar entre melodia e solo*). Embora ainda não tenha sido implementada, está prevista também a gravação de solos através de uma guitarra MIDI.

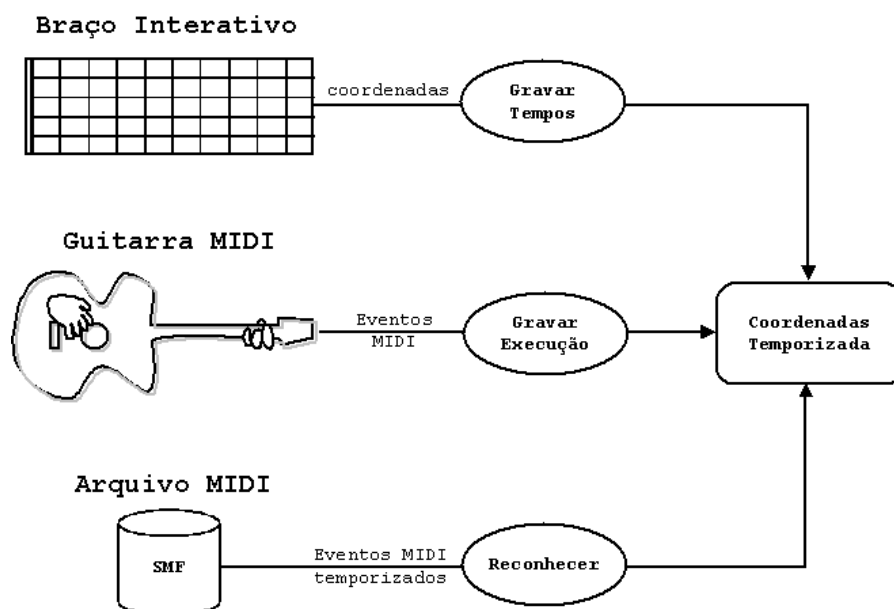


Figura 6.25 – Processo de gravação de solos

Eventos de Controle

Eventos de controle são meta-eventos musicais, como mudanças de andamento e indicações de pontos de repetição. Os pontos de repetição são definidos diretamente no texto da música cifrada, portanto o único tipo de evento de controle a ser gravado é o de mudança de andamento. Para gravá-los, é definida a lista de mudanças de andamento, e os instantes em que ocorrem são gravados de forma análoga aos explicados anteriormente, através do teclado do computador.

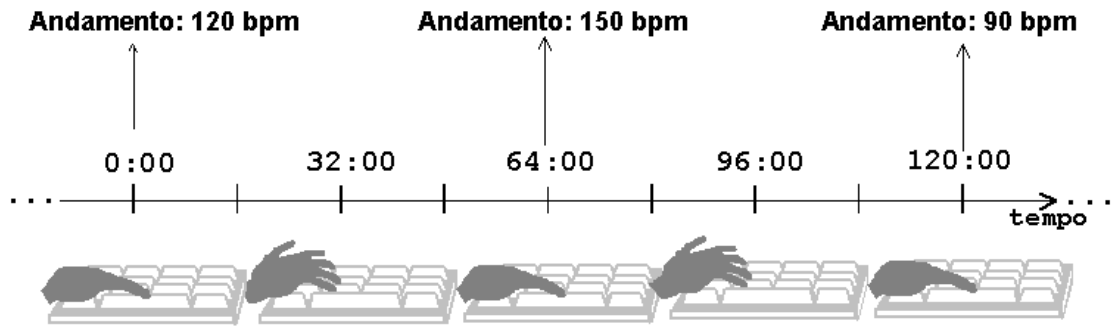


Figura 6.26 – Processo de gravação de eventos de controle

6.3.2. Playback

Uma execução *playback* é definida como uma execução em segundo plano. No caso do processo de gravação de algum elemento da música, trata-se da execução dos demais elementos, enquanto o elemento em questão é gravado. Algumas dificuldades são encontradas quanto a este *playback* durante a gravação (Hamakawa & Rekimoto 1994) e serão descritas a seguir.

Melodia

A primeira dificuldade refere-se à melodia. Como foi visto anteriormente, o *D'Accord Guitar* utiliza um sistema de representação de tempo diferente do utilizado em MIDI. Como escutar um arquivo MIDI, então, utilizando o seqüenciador *D'Accord Guitar*? A solução encontrada foi converter toda a melodia do arquivo MIDI para o formato DV3 no momento em que é importada. O processo realizado é o seguinte:

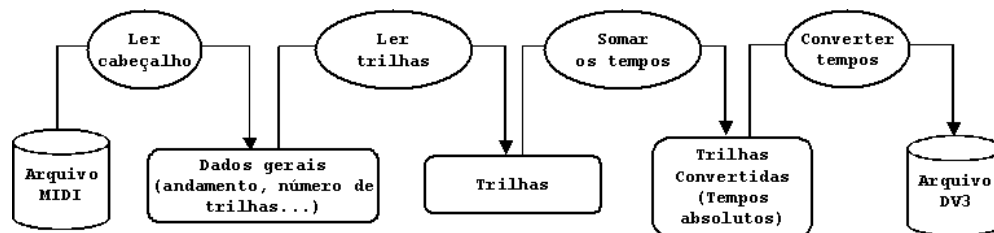


Figura 6.27 – Conversão MIDI → DV3

Em primeiro lugar, é preciso saber qual a resolução utilizada no arquivo MIDI. Isto é lido através do cabeçalho do arquivo. De posse desta resolução, é possível fazer uma relação entre os intervalos de tempo em MIDI e os intervalos de tempo em DV3.

Em seguida, todas as trilhas selecionadas para importação são varridas, gerando uma única trilha, combinada. Nela, os tempos MIDI são convertidos para uma representação absoluta, como a utilizada no *D'Accord Guitar*.

Por fim, os valores são convertidos para a unidade de tempo do *D'Accord Guitar*. Nesta conversão, provavelmente existirão valores fracionários. É feita então uma quantização por aproximação simples (Desain & Honing 1992).

Harmonia

Outra dificuldade encontrada refere-se à harmonia. Como tocá-la, no momento da gravação, se o ritmo ainda não foi escolhido? Isto pode ser melhor mentalizado se imaginarmos a seguinte situação: o usuário num primeiro instante grava a melodia da música, e escolhe os acordes e suas respectivas posições. Ele deseja, então, gravar as trocas de acordes. Ou seja, deseja indicar em que instante cada acorde vai entrar, de forma sincronizada com a melodia. Neste momento, o ritmo ainda não foi selecionado. Porém, este ritmo é que informa como tocar cada acorde. Como tocar os acordes escolhidos, então, sem um ritmo definido?

A solução utilizada toma proveito do fato de que mesmo sem ritmo, é possível “sentir” se as trocas de acordes estão ocorrendo no tempo certo apenas com um toque do músico em todas as cordas do violão, a cada troca. Assim sendo, no instante em que o usuário troca de acorde, todas as notas que compõem a posição são tocadas.

Ritmo

A gravação de um ritmo que será aplicado ciclicamente não precisa de *playback*, pois não existe uma correlação entre o ritmo e os acordes tocados. Pode-se dizer que o ritmo sobrevive sozinho. Prova disto é que existe no *D'Accord Guitar* uma *base de ritmos*, independente de música. Em outras palavras, ao gravar-se um ritmo, o usuário não necessita escutar a música ao fundo. Em verdade, o usuário precisa escutar a música quando ele vai gravar as trocas de ritmo. Neste caso, há uma relação direta com a música, já que é necessário saber em que ponto cada ritmo vai entrar.

É necessário escutar a música em *playback* também se o usuário selecionar a gravação do ritmo inteiro de uma música, e não de ritmos a serem repetidos. Neste caso, também há uma relação direta da gravação com a música.

6.3.3. Reedição

A reedição é importante para consertar os erros que quase sempre ocorrem na etapa de gravação, sem ter que regravar toda a execução. Existem, basicamente, duas informações a serem reeditadas: os eventos propriamente ditos, e os instantes em que ocorrem. Quanto aos eventos, é possível reeditar a lista de acordes, as posições relativas a cada acorde, a lista de ritmos utilizados, o valor do andamento, a letra da música e os trechos da letra a serem animados. Estes eventos são reeditados pela interface gráfica, da mesma forma com que foram editados.

Para alterar os instantes em que os eventos ocorrem são utilizados mecanismos de gravação de estúdio, como *overdubbing*, *punch in* e *punch off* (Keating & Anderton, 1998), de forma que um novo elemento seja gravado em tempo real enquanto os outros previamente gravados são tocados, sincronizadamente. A implementação deste processo de “regravação” em tempo real é bastante complexo, e seu funcionamento é ilustrado na Figura 6.28. Nesta figura, a primeira pauta (no alto) representa a gravação original, a pauta ao centro representa o trecho regravado e a pauta abaixo representa o resultado final (o trecho que foi regravado sobrescreve o trecho equivalente na gravação original).

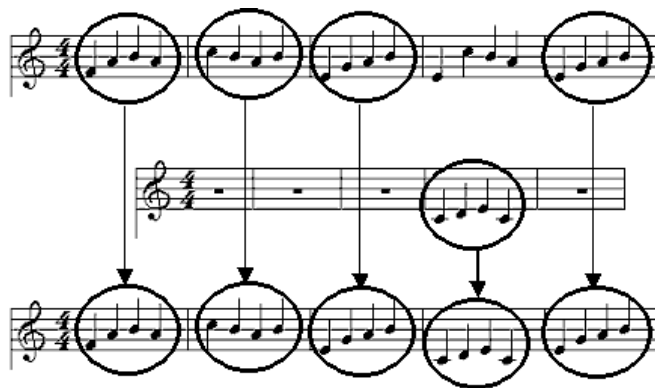


Figura 6.28 – Punch In, Punch Out, Overdubbing

6.3.4. Parsers

Um *parser* (Aho, Sethi & Ulmann 1986) é um analizador sintático, um algoritmo capaz de reconhecer seqüências (geralmente de texto) escritas de acordo com uma determinada sintaxe. Para cada sintaxe, é preciso criar um *parser* para reconhecê-la. Por exemplo, para reconhecer as *TAGS* em um arquivo *HTML*, é necessário implementar um *parser*, assim como é necessário para que um compilador “compreenda” o que está escrito em um arquivo fonte, em uma linguagem de programação.

Nesta seção, chamamos de *parsers* os algoritmos implementados para tratamento de texto no *D'Accord Guitar Editor*. Foram desenvolvidos 3 *parsers*, em duas etapas de processamento, como pode ser visto na Figura 6.29.

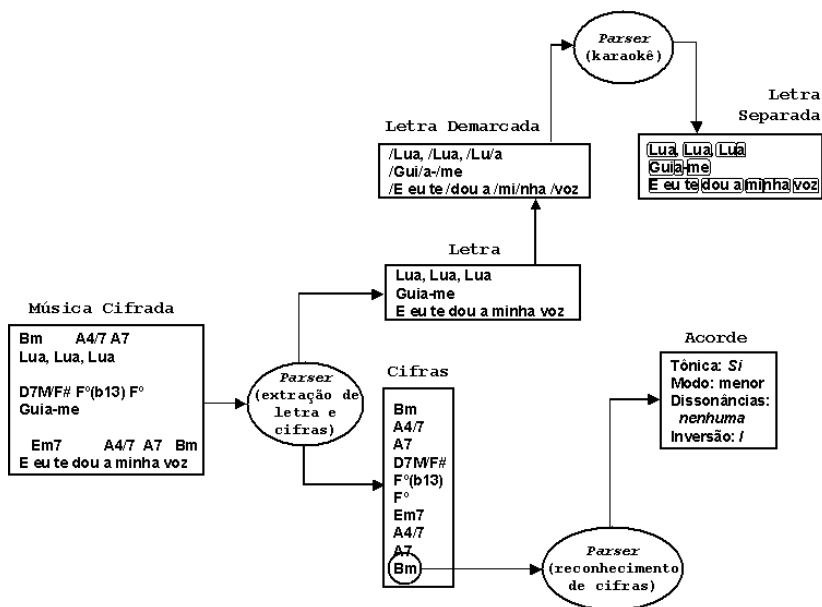


Figura 6.29 – Parsers necessários

A primeira etapa de processamento encontra-se no momento em que a letra e as cifras são exportadas da música cifrada para a gravação do karaokê e da harmonia. Neste caso é necessário primeiro um *parser* para distinguir o que é letra e o que são cifras dentro da música cifrada. Segundo, é necessário um *parser* para reconhecer as cifras dos acordes. A segunda etapa encontra-se no processo de edição da letra no karaokê, onde é necessário tratar as separações do texto. A seguir será estudado cada um destes 3 *parsers*.

Separação de Letra e Cifra

O primeiro parser trata de separar o que é letra e o que é cifra dentro de uma música cifrada, como na Figura 6.30.

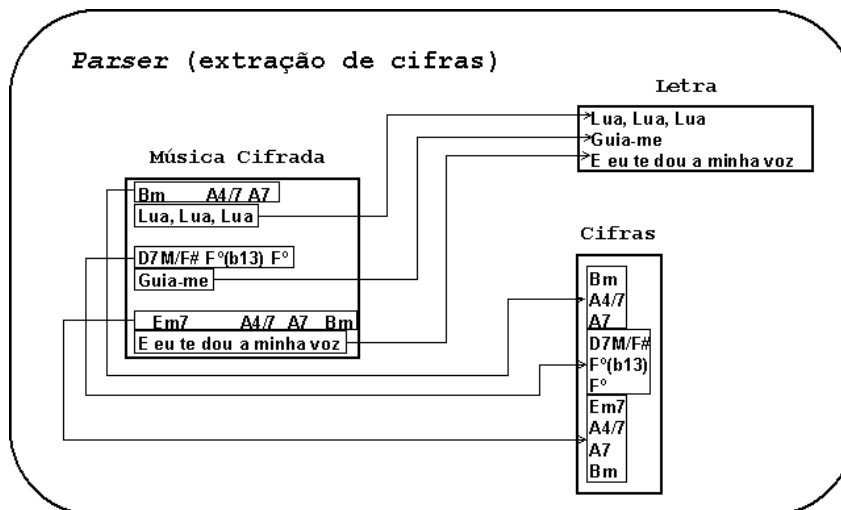


Figura 6.30 – Parser para separar letra e cifra

Este problema é especialmente complexo em virtude da falta de padronização das músicas cifradas. Na Figura 6.31, podem ser vistos alguns exemplos de cifras para uma mesma música. Tanto as cifras dos acordes variam de caso a caso, quanto o modo como elas se relacionam com a letra da música.

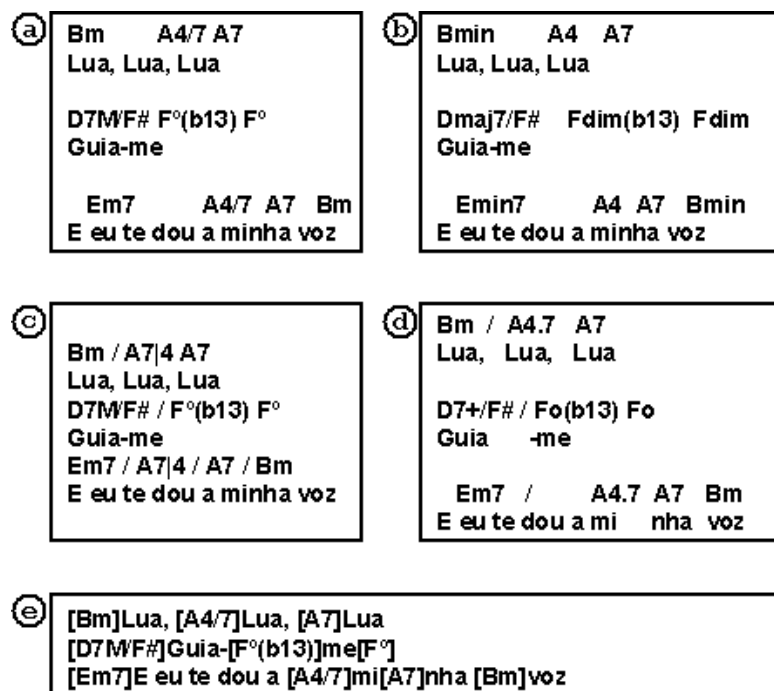


Figura 6.31 – Música cifrada de 5 maneiras diferentes

Uma heurística a utilizar para efetuar esta separação é perceber que normalmente as cifras e a letra localizam-se em linhas separadas (Sher 1991). Teoricamente, é possível

desenvolver um sistema que classifique cada linha e, desta forma, filtre os 2 elementos. Foi implementado um protótipo deste sistema que, porém, encontrou 2 dificuldades. A primeira é que esta regra não é absoluta. Existe público e até mesmo softwares que escrevem letra e cifras mescladas, como na letra *e* da Figura 6.31. A segunda é que cada cifra também não é padronizada. Desta forma, é possível desenvolver um sistema “bom”, mas praticamente impossível criar um sistema “perfeito”.

A solução encontrada foi criar uma padronização própria. No *D’Accord Guitar*, as músicas cifradas são compostas por blocos de linhas de acordes, linhas de letra e linhas em branco, alternadas. Abaixo, pode ser visto a representação em BNF deste formato:

```
<musica cifrada> = <blocos>
<blocos> = <blocos> + <linhas> | <linhas>
<linhas> = <linha acordes> + <linha letra> + <linha em branco>
<linha acordes> = <cifras> | <cifras> + <controle>
<cifras> = <cifra> + <cifra> | <cifra>
<controle> = <numero> + "X"
```

Figura 6.32 – BNF do formato de músicas cifradas do *D’Accord Guitar*

Uma linha de acordes é uma linha que contém apenas palavras reconhecidas como cifras (ver abaixo). Adicionalmente, as linhas de acordes podem conter informações de controle de repetição, indicando o número de vezes que um determinado conjunto de acordes deva ser repetido. As linhas de letra podem conter qualquer conjunto de palavras. As linhas em branco são usadas apenas para separar graficamente os blocos. Esta *música cifrada* constitui o texto que será animado.

Reconhecimento de Cifras

Após separar as cifras, é necessário reconhecer a que acorde cada uma corresponde. Este é um problema relativamente bem estudado (Pachet, Ramalho & Carrive 1996), porém sempre esbarra na ambigüidade e na falta de padronização das cifras. Por exemplo, existem várias formas de escrever um Lá maior com sétima maior: A7M, AM7, A7+, Amaj7, entre outros. A solução, portanto, foi adotar novamente uma padronização. Esta padronização segue indicações existentes na literatura musical para que seja a mais universal possível (Chediak 1984). Abaixo pode ser visto o BNF do formato de cifras utilizado no *D’Accord Guitar*.


```

<cifra> = <tonica> + <modo> + <dissonancias> + <inversao>

<tonica> = <nota>

<modo> = "m" | "o" | Ø

<dissonancias> = (<lista de dissonancias>) | <dissonancia> | Ø

<lista de dissonancias> = <lista de dissonancias> + <dissonância> |
                           <dissonancia>

<dissonancia> = "4" | "b5" | "#5" | "6" | "7" | "7M" | "b9" | "9" | "#9" |
                "11" | "#11" | "b13" | "13"

<inversao> = "/" + <baixo> | Ø

<baixo> = <nota>

<nota> = <sigla> | <sigla> + <acidente>

<sigla> = "A" | "B" | "C" | "D" | "E" | "F" | "G"

<acidente> = "#" | "b"

```

Figura 6.33 – BNF do formato de cifras do D'Accord Guitar

Caso alguma cifra encontrada em uma linha de acordes não seja reconhecida no padrão *D'Accord Guitar*, é escolhida a cifra que mais se aproxime dela.

Animação de letra

O último *parser* serve para tratar das separações da letra que serão animadas. Por exemplo, dada a música cifrada mostrada na Figura 6.34, com as separações indicadas pelas barras verticais, o que deverá ser selecionado para ser animado pelo karaokê? Elas devem selecionar os espaços em branco e vírgulas? Em quais situações?

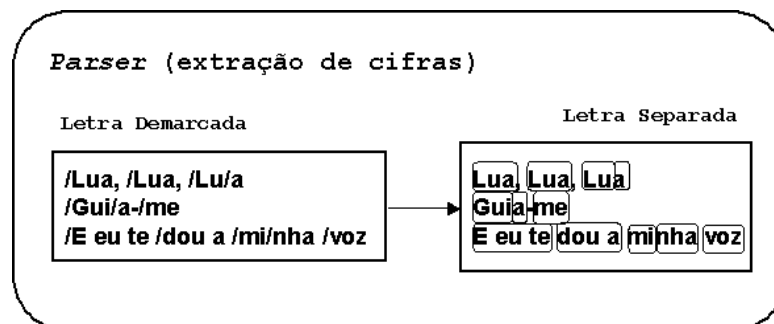


Figura 6.34 – Exemplo de letra a ser animada

Este *parser* é bastante simples, seguindo uma regra trivial. Caso as extremidades da separação sejam espaços em branco, caracteres de pontuação ou quebras de linha, o *parser* as despreza. Caso contrário, ele as seleciona.

6.4. *Discussão*

Alguns pontos podem ser apreciados sobre o *D'Accord Guitar Editor*, de forma a analisá-lo em contraponto aos demais softwares existentes hoje em dia, em especial aos softwares de edição musical.

Em primeiro lugar, o *D'Accord Guitar Editor* é um sistema que permite a edição de músicas cifradas. Existem alguns poucos sistemas que a fazem, hoje em dia, mas lidam com a música cifrada apenas como texto. Desta forma, são de muito pouca relevância. O *D'Accord Guitar Editor* efetua uma edição e gravação executável da música cifrada, o que torna o arquivo gerado um registro real da música, não apenas uma descrição harmônica textual.

Em segundo lugar, ele engloba funcionalidades que, mesmo quando vistas independentes do software, são interessantes ao grande público. Entre elas está um editor de arquivos de karaokê, com um modo de utilização extremamente prático, em contraste com os editores atuais, de interface bastante complexa.

Entre as ferramentas que o *D'Accord Guitar Editor* provê, está também um editor de ritmos. Estes ritmos podem ser trocados entre os usuários, da mesma forma que as músicas DV3. É possível a criação de bases de ritmos personalizadas, de modo expansível. Deixando esta geração a cargo dos usuários, é possível ter bases muito mais abrangentes.

Por fim, a sincronização de eventos através da gravação em tempo real é uma idéia que pode ser aproveitada em outros tipos de software.

7. *D'Accord Guitar Browser*

O *D'Accord Guitar Browser* permite ao usuário navegar através da base de dados de acordes, e interagir diretamente com o braço do violão, visando o aprendizado de teoria musical. Especialmente, a teoria sobre construção de acordes.

A tarefa de construir acordes é uma das mais importantes que um músico realiza ao ler uma cifra. Dado que a música cifrada indica apenas o nome do acorde, é preciso saber como montá-lo no instrumento específico. Como foi visto anteriormente, para os instrumentos de corda (como o violão) esta tarefa é bem mais complexa.

Normalmente, músicos começam seu aprendizado sobre construção de acordes decorando algumas posições. Quando aparece algum acorde desconhecido, os diagramas de posições acordes (que normalmente acompanha a música cifrada) são consultados. Com isso, tais músicos passam a saber tocar (mesmo que de forma precária) uma música cifrada. Dependendo do nível de exigência do músico, tal aprendizado já é suficiente.

Entretanto, por vários motivos, músicos profissionais aprendem a construir acordes. Em primeiro lugar, para saber como posicionar “qualquer” acorde, para que saibam executar uma música mesmo quando não houver um diagrama de posições de acordes, o que é cada vez mais comum com a difusão das músicas cifradas na Internet. Em segundo lugar, para encontrar novas posições que tragam uma nova sonoridade. Conhecendo várias posições para cada acorde, é possível escolher as que mais se encaixam a cada trecho da música.

Para dar suporte a tal aprendizado, no entanto, o *D'Accord Guitar Browser* não assume a forma de um tutorial, e sim a de um micromundo. Em um tutorial, o conhecimento é passado de maneira unidirecional, na forma de textos, imagens, sons, explicações e demonstrações. O aprendizado é então dito passivo, pois o usuário não participa da construção do conhecimento. Em um micromundo, o processo é interativo (o usuário interage com um ambiente que lhe dá respostas), tornando a atividade cognitiva mais lúdica e menos cansativa (Chung & Chung 1998, Davidson 1993).

7.1. *Interface e funcionalidades*

O *D'Accord Guitar Browser* centra-se no aprendizado que pode ser adquirido através da exibição de informações sobre cada posição de acorde no violão. Na Figura 7.1 é possível ver sua janela principal. No centro há o violão virtual onde, para cada posição escolhida, são mostradas informações como a cifra do acorde correspondente, as notas que compõem esse

acorde, os dedos utilizados e o intervalo destas para a nota tônica. Deste modo, o usuário pode entender que intervalos compõem cada tipo de acorde, que notas relacionam-se com cada cifra, que notas são encontradas em cada coordenada, que dedos escolher para cada posição, etc.



Figura 7.1 – Tela principal do D'Accord Guitar Browser

No *D'Accord Guitar Browser*, há basicamente duas funcionalidades: interação com o braço virtual e acesso à base de acordes. Elas serão explicadas a seguir.

7.1.1. Braço Interativo

Através da interação direta com o braço do violão virtual, é possível apreender conceitos musicais sobre formação de acordes. Esta interação é feita da seguinte forma: quando o usuário seleciona o braço do violão virtual com o mouse, a coordenada selecionada é escolhida. Desta forma o usuário pode montar a posição que quiser, e o *D'Accord Guitar Browser* exibirá as informações sobre ela (cifra, notas, intervalos e dedos a utilizar), como mostra a Figura 7.2. O usuário pode efetivamente tocar o acorde escolhido, posicionando o mouse sobre a extremidade direita do braço¹³.

¹³ Este local foi definido em analogia à execução real do músico, já que é o local onde ele posiciona a sua mão direita.

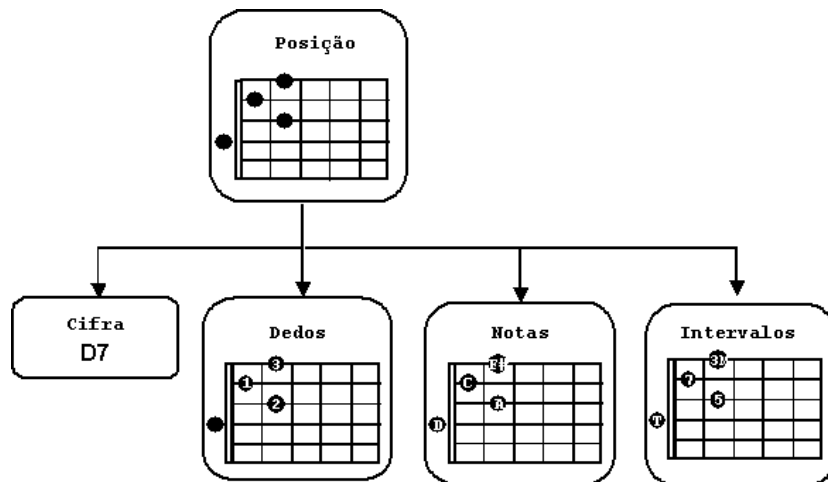


Figura 7.2 – Informações sobre uma posição de D7

7.1.2. Acesso à base de acordes

Através das funções de *acesso ao BD* o usuário pode verificar quais posições e digitações existem para cada acorde, através de uma consulta à base de dados de acordes existente no *D'Accord Guitar*. Ele pode também adicionar novas posições que ele tenha montado no braço interativo, caso deseje. Desta forma, o *D'Accord Guitar Browser* serve também como o editor da base de dados utilizada.

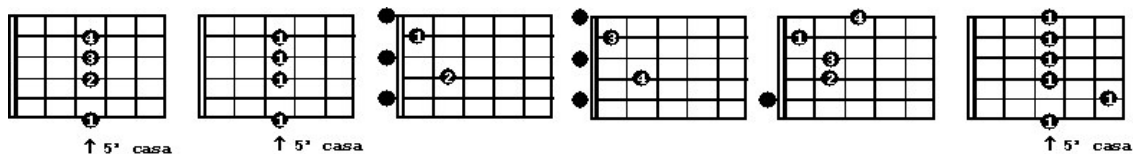


Figura 7.3 – Diversas posições e digitações para um Am7

7.2. Arquitetura

Na figura a seguir (Figura 7.4), pode ser vista a arquitetura do *D'Accord Guitar Browser*. O usuário interage com o sistema através das funções de *acesso ao banco de dados* ou através da *interação com o instrumento virtual*. Quando utiliza as funções de acesso ao banco de dados, estas se comunicam com a base de acordes. Quando utiliza a *interação com o instrumento virtual*, são chamadas as funções de *reconhecimento* para descobrir quais são os *dados* (cifra, dedos, notas, intervalos) referentes à posição e às suas coordenadas. Para efetuar tal reconhecimento, o sistema acessa uma *base de conhecimento* contendo os intervalos que constituem cada acorde, e são verificadas as *preferências* escolhidas pelo usuário (e.g., qual a

Esta base de acordes, no entanto, não é capaz de armazenar todas as posições possíveis de acordes, já que, como foi visto, este número é muito grande. A tarefa contrária, portanto, é não trivial. Trata-se de recuperar as informações de um acorde baseado em uma posição, o que exige um processo de reconhecimento da informação desejada.

7.3.2. Reconhecimento de acordes, notas e intervalos

Estas funções tratam do reconhecimento da cifra da posição de acorde escolhida, juntamente com as notas, intervalos e dedos relativos a cada uma das coordenadas que formam esta posição. O reconhecimento dos três primeiros elementos seguem um único processo, ilustrado na Figura 7.5.

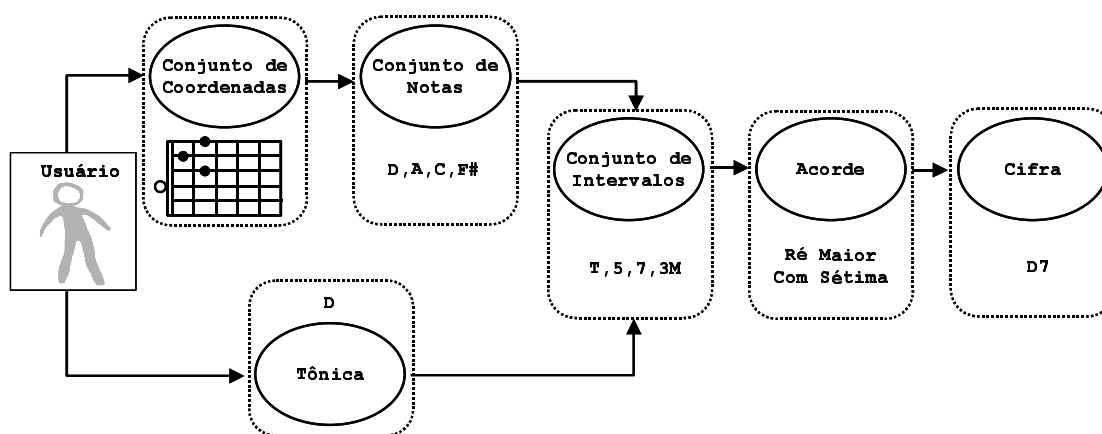


Figura 7.5 – Processo de reconhecimento dos dados relacionados ao acorde

Notas

Embora cada nota possa ser posicionada em várias coordenadas diferentes, como explicado no capítulo 2, cada coordenada corresponde a apenas uma nota. Logo, a conversão de uma posição escolhida pelo usuário em um conjunto de notas (linha B na Figura 7.5) é um processo direto.

Intervalos

O passo seguinte, a conversão deste conjunto de notas em um conjunto de intervalos, é menos trivial. Um intervalo é definido como a distância (medida em semitons) entre duas notas. Na Tabela 3 podem ser vistos os intervalos entre cada dupla de notas.

	A	A#/Bb	B	C	C#/Db	D	D#/Eb	E	F	F#/Gb	G	G#/Ab
A	T	b9	9	3m #9	3M	4 11	b5 #11	5	#5 b6 b13	6 13	7	7M
A#/Bb	7M	T	b9	9	3m #9	3M	4 11	b5 #11	5	#5 b6 b13	6 13	7
B	7	7M	T	b9	9	3m #9	3M	4 11	b5 #11	5	#5 b6 b13	6 13
C	6 13	7	7M	T	b9	9	3m #9	3M	4 11	b5 #11	5	#5 b6 b13
C#/Db	#5 b6 b13	6 13	7	7M	T	b9	9	3m #9	3M	4 11	b5 #11	5
D	5	#5 b6 b13	6 13	7	7M	T	b9	9	3m #9	3M	4 11	b5 #11
D#/Eb	b5 #11	5	#5 b6 b13	6 13	7	7M	T	b9	9	3m #9	3M	4 11
E	4 11	b5 #11	5	#5 b6 b13	6 13	7	7M	T	b9	9	3m #9	3M
F	3M	4 11	b5 #11	5	#5 b6 b13	6 13	7	7M	T	b9	9	3m #9
F#/Gb	3m #9	3M	4 11	b5 #11	5	#5 b6 b13	6 13	7	7M	T	b9	9
G	9	3m #9	3M	4 11	b5 #11	5	#5 b6 b13	6 13	7	7M	T	b9
G#/Ab	b9	9	3m #9	3M	4 11	b5 #11	5	#5 b6 b13	6 13	7	7M	T

Tabela 3 – Tabela de intervalos entre notas

Dentro de um acorde, o conjunto de intervalos corresponde à distância entre cada nota que o compõe e a nota tônica. Tônica é a nota mais importante do acorde, e que lhe empresta seu nome. Por exemplo, dado o conjunto de notas existentes na Figura 7.2, caso a tônica seja o Ré (D), as notas Lá (A), Dó (C) e Fá Sustenido (F#) estão a intervalos de quinta (5), sétima (7) e terça maior (3M), respectivamente.

O problema nesta etapa do reconhecimento é descobrir qual destas notas é a tônica. No *D'Accord Guitar*, é sugerida como tônica a nota posicionada na corda mais grave utilizada, denominada baixo do acorde. No exemplo da Figura 7.2, a tônica sugerida é o Ré (D) posicionado na quarta corda. Esta sugestão baseia-se no fato de que o baixo do acorde normalmente obtém destaque no violão. O usuário pode, contudo, definir que a tônica seja outra, que não seja a do baixo do acorde. Neste caso, é dito que ocorreu uma inversão. Por exemplo, se na posição representada na Figura 7.2 o usuário escolher como tônica o Lá (A), o Ré (D) estará a um intervalo de quarta (4), o Dó (C) a um intervalo de terça menor (3m) e o Fá Sustenido (F#) a um intervalo de sexta (6).

Acordes

Os acordes são compostos por 3 ou mais notas, onde uma delas é necessariamente a tônica (T) e outras duas são a terça (3 ou 3M) e a quinta (5). As demais notas existentes no acorde são chamados de dissonâncias (Bennett 1986). Dados o conjunto de intervalos, a tônica e o conhecimento sobre que intervalos compõem cada acorde, é possível reconhecê-lo. No exemplo da Figura 7.5, o acorde é um Ré Maior com Sétima.

Entretanto, existe ainda um problema, que é comumente chamado de *chord spelling*, ou nomenclatura de acordes (Pachet, Ramalho & Carrive 1996). Este problema refere-se ao fato de que algumas tônicas podem ter mais de um nome (e.g. A# e Bb) e de que alguns intervalos também podem ter mais de um nome (e.g. #5, b6, b13). Deste modo há um problema, por exemplo, em decidir como será escrito o acorde da Figura 7.6: A#(7/b13), A#(7/#5), Bb(7/b13), Bb(7/#5)?

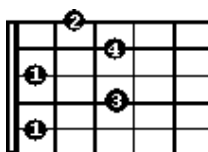


Figura 7.6 – Posição a ser reconhecida

Por enquanto, a solução adotada é reconhecer sempre as mesmas tônicas e os mesmos intervalos em cada situação. No caso da posição da Figura 7.6, é reconhecida a cifra A#(7/b13). Na forma de ajuda, o software expõe ao usuário as tônicas e os intervalos que possuem o mesmo nome, e desta forma ele pode associar a cifra mostrada a qualquer cifra que ele conheça. Com esta solução, a tarefa de determinar que cifra corresponde a um acorde é bastante direta.

É possível, portanto, reconhecer a maior parte das informações sobre um acorde. Para realizar este reconhecimento, o *D'Accord Guitar* precisou representar conhecimentos provindos da teoria musical e violonística, o que foi feito na forma de dados devidamente estruturados e de regras (West, Howell & Cross 1991).

7.3.3. Digitação

O problema de digitação trata do reconhecimento de que dedos serão utilizados em cada coordenada. Este processo de reconhecimento é mais complexo e, por fazer parte da solução do *módulo especialista*, será descrito em seu respectivo tópico, a seguir.

7.4. *Discussão*

O *D'Accord Guitar Browser* é, portanto, um software educativo ao mesmo tempo em que é uma obra de referência. Ele se distingue dos demais *Dicionários Interativos de Acordes* por duas razões. A primeira é por descrever a informação de maneira mais detalhada. Ele mostra, por exemplo, os dedos utilizados e intervalos de cada nota à nota tônica. A segunda é por integrar-se ao formato DV3, utilizando a mesma base de dados do *D'Accord Guitar Player* e do *D'Accord Guitar Editor*, servindo inclusive como ferramenta de edição desta base.

8. Interpretação de Cifras

O problema de interpretação harmônica trata de escolher a melhor sequência de posições no violão para uma sequência de acordes, de acordo com um conjunto de preferências. Nosso objetivo, neste trabalho, foi conseguir uma solução inicial para automatizar esta interpretação, descobrindo as razões que levam os músicos a escolherem determinadas posições e digitações em detrimento de outras. Como não há um consenso sobre tais razões (cada músico dá importância a determinados critérios), e como houve pouco tempo para desenvolver esta solução (pois a maior parte do tempo já foi ocupada pelo desenvolvimento do *D'Accord Guitar* como um todo), tivemos que estabelecer convenções de forma a viabilizar sua implementação. Este capítulo descreve a solução implementada e mostra alguns estudos de caso.

8.1. Estado da Arte

Atualmente, não existem soluções computacionais para este problema. A única solução encontrada é a do sistema de acompanhamento automático Band-In-A-Box¹⁴, a partir de sua décima versão. Nela, as posições e digitações de acordes são associadas a estilos musicais. Desta forma, é dito, por exemplo, que a posição *a* na Figura 8.1 é a posição de *Am7* utilizada em bossa-nova, enquanto a posição *b* na Figura 8.1 é a posição de *Am7* no estilo Rock.

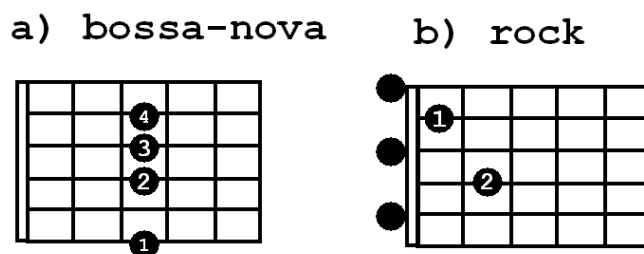


Figura 8.1 – Duas posições possíveis para um *Am7* associadas, respectivamente, aos estilos bossa-nova e rock

Esta solução, contudo, é extremamente artificial. O músico toma esta decisão não de acordo com o estilo, mas de acordo com uma série de fatores, muitas vezes contraditórios. Estamos aqui interessados em descobrir uma maneira automática de simular o que um músico pensa quando ele tem que tomar tal decisão.

¹⁴ PG Music (www.pgmusic.com)

Idealmente, desejaríamos utilizar sistemas que aprendessem a resolver problemas e a lidar com eles em situações novas, apenas com a experiência. Por exemplo, pela observação de como um humano realiza tais tarefas. Os métodos de aprendizagem de máquina supervisionados são técnicas computacionais para fazer isto, tendo como estratégia a generalização (indução) empírica da observação específica (Michalski 1983). Entre os métodos normalmente utilizados em aprendizado de máquina, encontram-se as Redes Neurais (Haykin 1999), ID3 (Mitchell 1997a), Raciocínio Baseado em Casos (Kolodner 1993), Redes Bayesianas (Mitchell 1997b) e métodos estatísticos. Estes métodos dividem-se, tradicionalmente, em 3 etapas: aquisição de exemplos, aprendizado e execução. A aquisição de exemplos trata de inserir a informação a ser aprendida. A etapa de aprendizado trata da execução do mecanismo de aprendizagem. A etapa de execução informa um resultado para exemplos novos, baseado no que foi aprendido com os exemplos dados.

8.2. Metodologia utilizada

Para solucionar o problema de interpretação de cifras, entretanto, nem sempre foi possível utilizar tais técnicas, em virtude do tempo reduzido ou da dificuldade de aquisição de algumas informações. A dificuldade de aquisição deveu-se algumas vezes à necessidade de um grande volume de informação, e outras vezes ao fato de haver informações culturais, sem haver (ao menos teoricamente) justificativas para as escolhas tomadas. Nas seções a seguir, será descrita a solução criada para o *D'Accord Guitar*, descrevendo as técnicas e os métodos utilizados.

Para resolver o problema de interpretação de cifras, o *D'Accord Guitar* utiliza a estratégia de "*dividir para conquistar*", onde o problema é recursivamente quebrado em sub-problemas. A interpretação de cifras, como pode ser visto na figura Figura 8.2, é dividida em 3 etapas. A primeira trata de encontrar todas as posições e digitações possíveis para cada cifra de acorde. Estas posições e digitações constituirão o espaço de busca a ser utilizado nas etapas seguintes. A segunda etapa trata de avaliar cada posição e digitação, conferindo valores a atributos, como facilidade e flexibilidade para as posições. Por vezes os valores são calculados pelo programa, por vezes são lidos diretamente da base de acordes. A última etapa trata de ponderar estes atributos, de forma a selecionar a melhor seqüência. Nas seções a seguir, cada etapa será descrita em maiores detalhes.

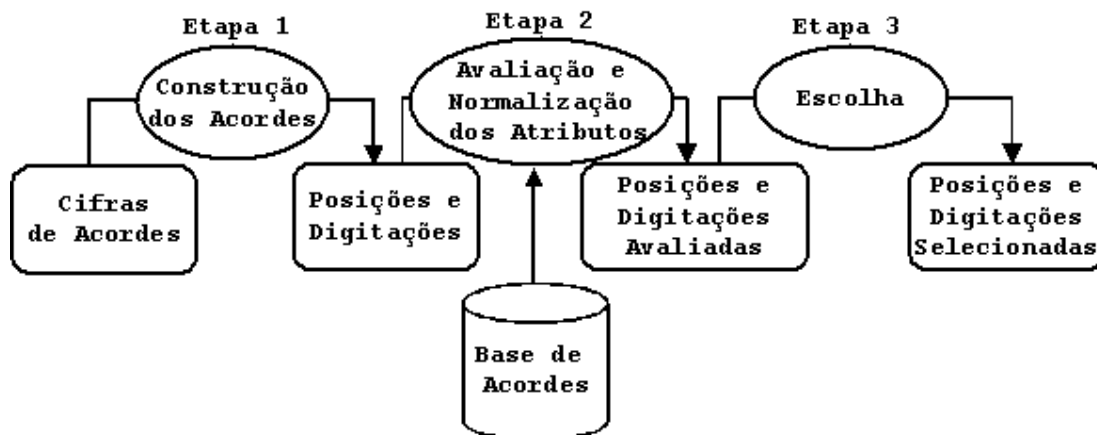


Figura 8.2 – Processo geral de interpretação

8.3. Etapa 1: construção de acordes

Seguindo a estratégia de "*dividir para conquistar*", a construção de acordes divide-se em duas sub-etapas: encontrar as coordenadas que configurem uma posição como correspondente a um dado acorde (posicionamento) e encontrar os dedos a utilizar para montar tal posição (digitação), como discutido nas seções 2.4.1.3 e 2.4.1.4.

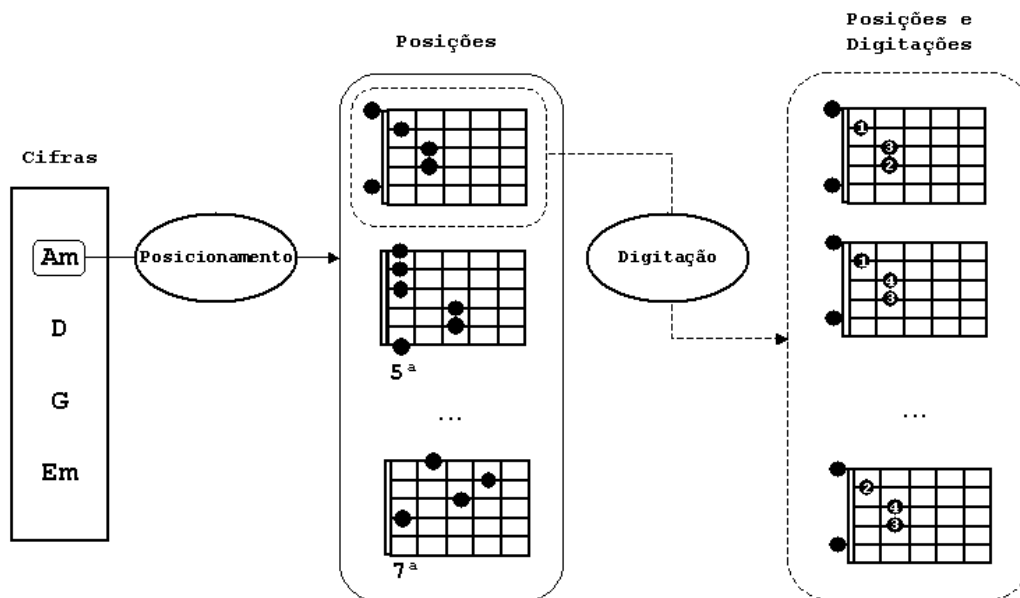


Figura 8.3 – Construção de acordes

As soluções sugeridas e implementadas no módulo especialista do *D'Accord Guitar* para os problemas de posicionamento e digitação são descritas a seguir.

8.3.1. Posicionamento

Seguindo a estratégia de "*dividir para conquistar*", o problema de posicionamento também é dividido em sub-etapas, como visto na figura abaixo.

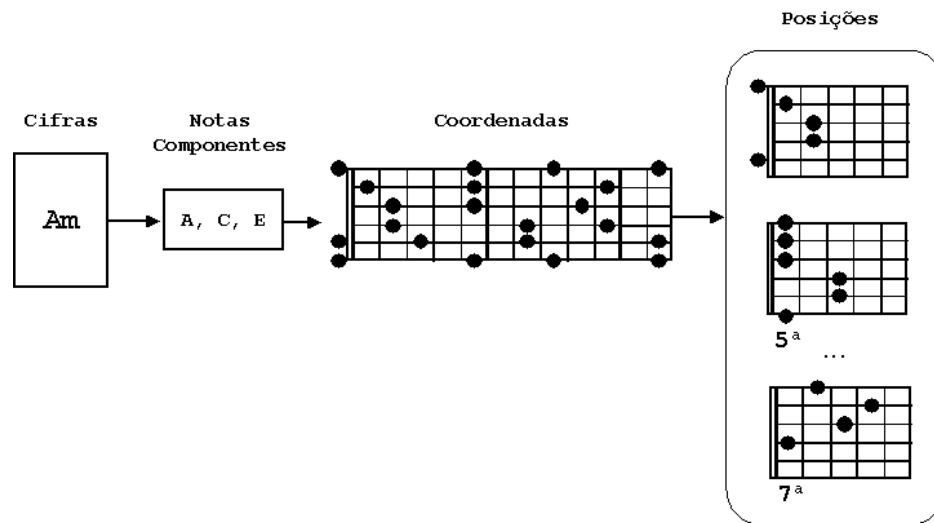


Figura 8.4 - Posicionamento

A primeira etapa trata de buscar o conjunto de notas que compõem cada cifra dada pelo usuário. Isto é feito através de um *parser* para cifras, que reconhece qual a tônica, o modo, as dissonâncias e a inversão utilizadas (Figura 8.5).



Figura 8.5 – Reconhecimento de acordes

A segunda etapa trata de buscar as coordenadas no violão para este conjunto de notas. Esta etapa é bastante simples, uma vez que o programa tem conhecimento sobre a nota em que cada corda inicia e sobre o intervalo entre as notas.

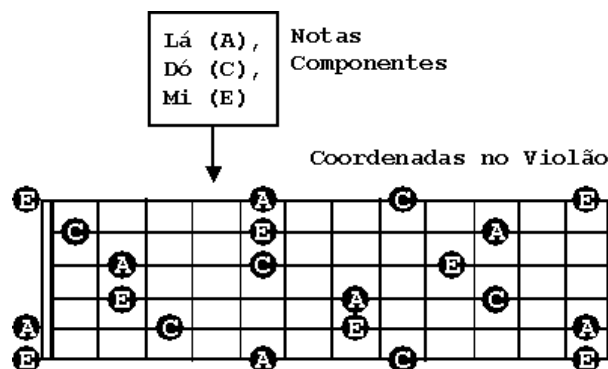


Figura 8.6 – Coordenadas que podem ser utilizadas para formar o acorde

Por fim, busca-se o conjunto de todas as combinações destas notas encontradas que obedeçam a determinadas restrições (Rayward-S, Osman & Reeves 1996). Cada combinação encontrada deve obedecer a restrições musicais, anatômicas e técnicas. Algumas são escolhidas pelo usuário, outras são internas ao sistema. Em função delas, o problema de posicionamento pode ser visto como um problema de satisfação de restrições (Tsang 1993, Pachet & Roy 1995).

Restrições musicais

As restrições musicais referem-se à escolha de vozes dos acordes. São usadas para garantir que a combinação encontrada efetivamente forme o acorde desejado. Existem restrições internas ao sistema, que exigem que sejam encontrados acordes bem formados, seguindo regras da teoria musical. Através destas, não são permitidos acordes que omitam a tônica, por exemplo. A Figura 8.7 mostra algumas das posições encontradas para um Fm7(11):

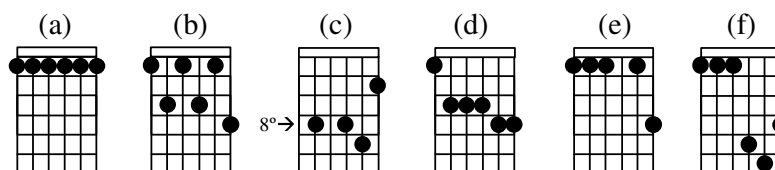


Figura 8.7 – Algumas posições encontradas para um Fm7(11).

Restrições anatômicas da mão esquerda

As restrições anatômicas são usadas para garantir que o acorde possa ser fisicamente tocado pelo músico. Neste caso, é levado em consideração o número de dedos da mão esquerda utilizados e a distância máxima entre eles. A distância máxima é configurável pelo usuário, e tem como valor padrão 4 casas. Desta forma, posições de acorde como a mostrada

na Figura 8.7f é descartada. O número máximo de dedos a utilizar é definido como 4 (indicador, médio, anular e mínimo). Em função desta restrição, posições de acorde como a mostrada na Figura 8.7d são descartadas (já que necessita de 6 dedos para ser tocada).

Mesmo sendo possível usar apenas 4 dedos, no máximo, é possível montar acordes em posições onde mais de 4 cordas sejam pressionadas ao mesmo tempo. O número de cordas pressionadas pode ser maior que 4, caso seja usada uma pestana. Uma pestana é o posicionamento de um dedo (normalmente o indicador) sobre mais de uma corda, como nos exemplos 1 e 4 (respectivamente G e C/E) da Figura 8.8. Nesta figura, círculos pretos indicam um dedo pressionando uma corda. Círculos brancos indicam uma corda tocada solta. Cordas sem círculos indicam cordas não que não devem ser tocadas. Os retângulos em cinza indicam as possíveis pestanas.

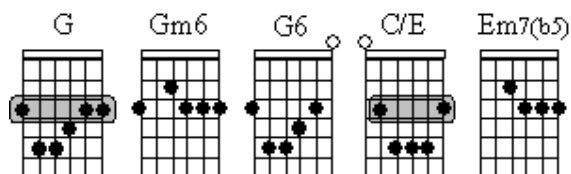


Figura 8.8 – Algumas posições de acordes onde mais de 4 cordas são pressionadas.

Existem casos em que só é possível executar a posição se for usada uma pestana, outros em que não há a possibilidade de se usar pestana, e casos onde sua utilização é opcional. O problema, então, é saber quando uma pestana é aplicável, já que ela nem sempre pode ser utilizada. No *D'Accord Guitar*, convencionaram-se duas regras para detectar se ela pode ser utilizada. Em primeiro lugar, apenas o dedo indicador pode ser utilizado para pestana, portanto ela estará sempre localizada na casa mais à esquerda da posição do acorde. Em segundo lugar, a pestana inicia sempre na corda mais aguda (1ª corda) e se estende até uma das demais cordas. Como padrão, o *D'Accord Guitar* encontra todas as posições que sejam exeqüíveis, usando ou não pestana.

Restrições técnico-culturais da mão direita

Há também restrições sobre a execução da mão direita do músico. Elas servem para restringir acordes onde o conjunto de cordas usadas não seja usual. Na verdade, são posições exeqüíveis, cujas notas efetivamente caracterizam o acorde, porém que vão de encontro à técnica violonística. Em função disto, são chamadas de restrições técnico-culturais. Como padrão, é suposto que o violonista usa um mínimo de 3 cordas para compor um acorde, e

sempre posiciona os dedos indicador, médio e anular em cordas consecutivas. Desta forma, posições de acorde como a da Figura 8.7e (onde a 3ª corda não é tocada) são descartadas.

Configuração de restrições

As restrições podem ser utilizadas pelo usuário como uma maneira fácil de descartar posições de acordes que ele considere indesejáveis. Por exemplo, se ele deseja tocar a música com uma batida de *rock*, que não seja dedilhada, ele pode escolher restringir as posições como a da Figura 8.9, onde há cordas que não podem ser tocadas (no caso da figura, as cordas 1 e 5).

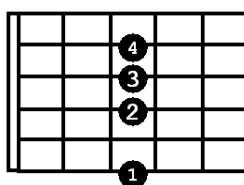


Figura 8.9 - Figura com uma armadura da mão direita desaconselhável para tocar batidas de Rock.

O *D'Accord Guitar* oferece uma interface onde o próprio usuário pode configurar as restrições (Figura 8.10), eliminando uma série de posições de acordo com suas propriedades.

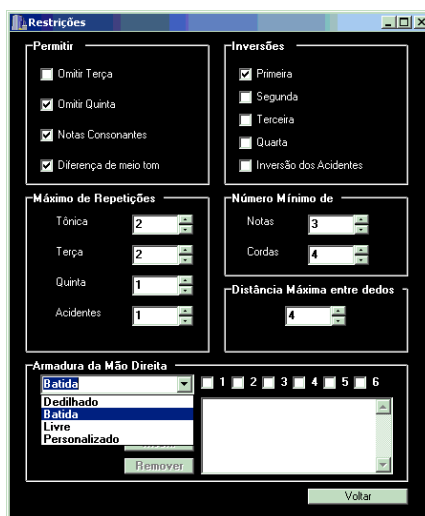


Figura 8.10 - Interface de configuração de restrições

No canto superior esquerdo, o usuário pode restringir ou permitir posições de acordes que omitam a quinta ou a terça nota em casos onde elas são facultativas. Além disso, ele pode restringir ou permitir posições que possuam coordenadas consonantes (na mesma frequência)

ou com uma diferença de meio tom. Na Figura 8.11 são exibidas algumas posições de acordes com estas propriedades.

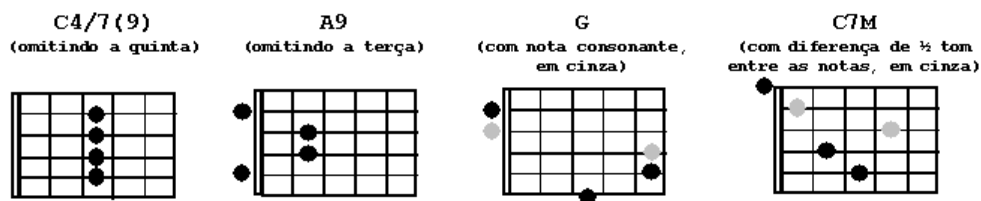


Figura 8.11 - Exemplos de posições de acorde passíveis de sofrer restrições

O usuário pode, também, escolher quais inversões podem ser feitas, quantas ocorrências podem haver de cada nota que compõe o acorde, e restringir a distância máxima entre os dedos. Esta configuração permite ao usuário informar sua preferência.

8.3.2. Digitação

A modelagem da mão do músico já é necessária desde a etapa de posicionamento, mas torna-se fundamental na etapa de digitação. Para cada posição encontrada, é preciso saber que dedos podem ser posicionados em cada corda, de maneira exequível. Este também pode ser visto como um problema de satisfação de restrições, onde todas as combinações que satisfaçam às restrições são encontradas. Embora envolva apenas quatro dedos, a tarefa de reconhecer quais dentre estas configurações de dedos são exequíveis pelo músico é bastante difícil. A principal razão é que é necessário um conhecimento não explícito, e por vezes controverso, por parte dos músicos.

Embora tenha sido difícil para nós defini-las, as regras que especificam as restrições de digitação são simples. Por exemplo, se existem duas cordas pressionadas em diferentes casas, a corda que usa a casa mais próxima à pestana¹⁵ do violão, usa o dedo de menor número. Se existem duas cordas pressionadas em uma mesma casa, a corda mais grave usa o dedo de menor número. Na Figura 8.12, o dedo indicador é colocado na 2ª corda e o dedo médio é colocado na 4ª corda em função da primeira regra. O dedo anelar é colocado na 5ª corda e o dedo mínimo é colocado na 3ª corda em função da segunda regra.

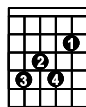


Figura 8.12 – Digitação para uma posição de Db7

¹⁵ Neste caso, pestana refere-se à parte do violão situada na extremidade do braço do violão.

Um fator complicador é a possibilidade de utilização de pestana, principalmente nos casos em que sua utilização é facultativa. Neste caso, é necessário reconhecer se é possível a sua utilização, como visto na seção 8.3.1. A Figura 8.13 mostra algumas das possibilidades de digitação para uma posição de Dm7, tanto com a utilização quanto sem a utilização de pestana. Neste exemplo, as digitações *a*, *b*, *e*, *f* e *g* obedecem às restrições. Já a digitação *c*, embora possa ser executada, vai de encontro à técnica violonística (o dedo indicador normalmente seria colocado na 2ª corda) e a digitação *d* não é sequer exequível.

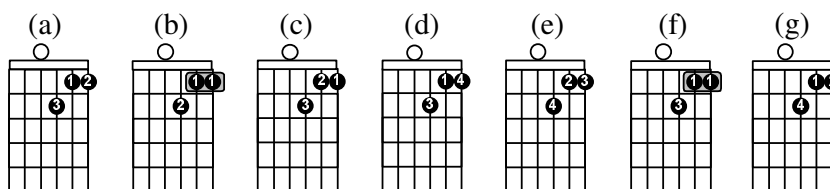


Figura 8.13 – Possíveis digitações de uma posição de Dm7

8.4. Etapa 2: critérios de avaliação de posicionamento e digitação

Nosso objetivo é encontrar as *melhores* posições e digitações de uma determinada seqüência de cifras. A etapa de construção de acordes gera um conjunto de posições e digitações, e é preciso agora alguma maneira de achar quais as melhores entre elas. Em outras palavras, é preciso saber como avaliá-las. Acreditamos que o primeiro passo seja a definição de quais critérios são levados em consideração para considerar uma posição melhor do que outra. O segundo passo é descobrir como calcular automaticamente valores para estes critérios.

A definição dos critérios a serem utilizados baseou-se na leitura da literatura da área musical (e.g. Fowler 1984a, Fowler 1984b, Holdsworth 1998) e em conversas informais com um grupo de 8 especialistas, de diferentes formações (amadores e profissionais, autodidatas e com educação clássica). Os critérios selecionados foram:

- Facilidade do acorde: associado ao conforto em montar um acorde em uma determinada posição com uma determinada digitação.
- Flexibilidade: dando preferência a posições que permitam improvisações e arranjos solísticos.
- Frequência de uso: indicando o quão comumente uma posição e uma digitação são utilizadas.

- Sonoridade: associado à sensação provocada por uma determinada escolha de vozes.
- Facilidade da transição: freqüentemente mencionado como *similaridade* entre posições, indica o grau de dificuldade (ou facilidade) em passar de uma determinada posição para uma outra.
- Linha de baixo e linha melódica: associados à melodia efetuada pela nota mais grave (baixo) ou mais aguda do violão, durante o acompanhamento.

Os músicos iniciantes costumam escolher posições e digitações de acordo com o nível de dificuldade exigido. Além disso, enquanto ainda estão no processo de memorização de posições, preferem as mais comuns. Já músicos experientes costumam tomar estas decisões baseados em outros critérios, como na escolha de vozes de cada posição ou nas posições que suavizem as linhas melódicas das cordas mais graves e/ou agudas. Quando se deseja fazer arranjos de solo simultaneamente à execução do acompanhamento, é comum escolher posições que deixem dedos livres e com o maior alcance possível. Tal tipo de posição é muito útil, por exemplo, quando se deseja improvisar sobre o acompanhamento de base. É senso comum, ainda, escolher posições pela “sonoridade” e pela “facilidade de transição”. A sonoridade é um critério altamente subjetivo, e bastante influenciado pela escolha de vozes. É uma representação ao mesmo tempo qualitativa e quantitativa, simbolizando o quanto o som gerado por uma determinada posição é bonito, o quanto é tenso, se provoca suspense ou relaxamento, entre outras. A facilidade de transição, por outro lado, é um critério mais objetivo. Músicos usualmente escolhem uma determinada posição por ser mais próxima à anterior, sendo mais fácil passar de uma para a outra.

Definidos os critérios, é preciso encontrar uma maneira de calcular valores para eles, de maneira que seja possível comparar qualquer par de posições de acordo com qualquer destes critérios. Como nenhum destes critérios é objetivo, nós fizemos novamente uma divisão em subcritérios. A idéia foi reduzir cada subcritério a subcritérios até encontrar parâmetros objetivos, que pudessem ser computados por uma máquina. A Figura 8.14 mostra o processo decisório inicialmente definido.

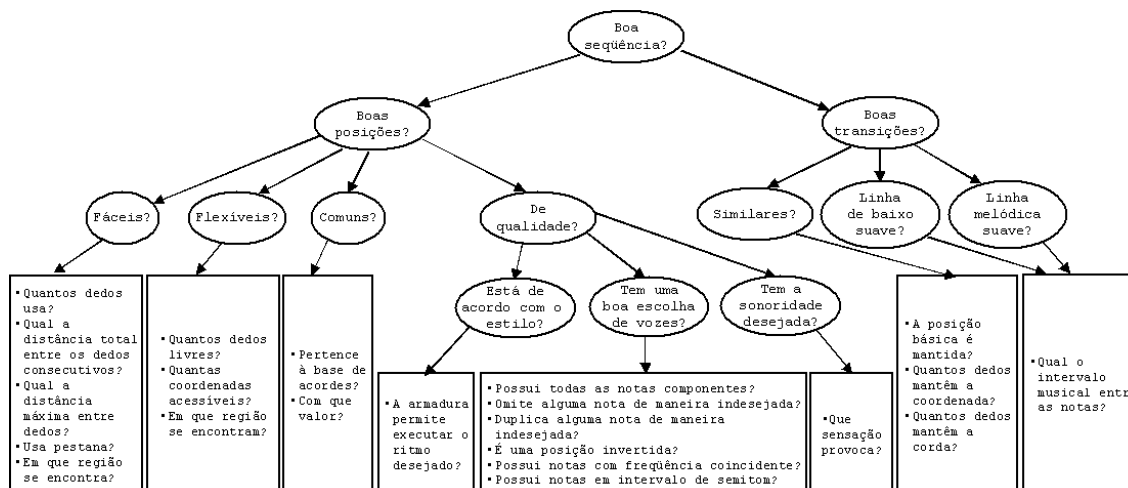


Figura 8.14 – Processo decisório para encontrar a melhor sequência de posições para uma sequência de cifras

Deste modelo inicial, foi visto que alguns critérios eram complexos demais para se lidar, e por isso algumas simplificações foram feitas. Critérios dinâmicos, como a adequação da armadura ao ritmo aplicado, e questões subjetivas, como "que sensação a posição do acorde provoca" não foram tratados a princípio. A seguir, será descrita a maneira com que o cálculo do valor de cada critério foi feito. Esta etapa é chamada, neste trabalho, de etapa de *valoração*.

8.4.1. Valoração

Dados os critérios, a próxima etapa para a implementação do processo de avaliação trata da valoração de cada um deles, individualmente. Alguns são extremamente subjetivos, alguns são culturais, outros são objetivos. A seguir, será explicado como foram calculados os valores para cada um.

Facilidade

O primeiro critério estudado trata do nível de dificuldade envolvido na execução de uma determinada posição. De forma oposta, pode ser visto como a facilidade (ou o conforto) inerente a cada posição. Embora seja um critério bastante subjetivo, algumas propriedades podem ser observadas para comparar duas posições de acordo com a "facilidade". Podendo comparar-se duas posições, pode-se também fazer uma ordenação de todas as posições e, portanto, achar a sequência mais fácil de executar.

Estas propriedades são: o número de dedos utilizados, as distâncias entre estes dedos, a utilização ou não de pestana e a região em que se encontra. Por exemplo, na Figura 8.15, a

posição A é considerada mais fácil do que a posição B porque a região em que se encontra exige menos força para pressionar as cordas.

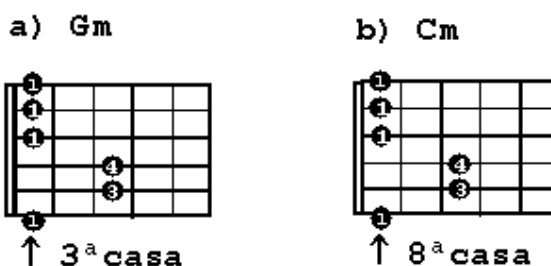


Figura 8.15 – Duas posições. O Gm é considerado mais fácil por encontrar-se numa região de menor tensão

Estas propriedades, em separado, podem ser usadas para comparar posições. O que fazer, no entanto, para lidar com estas propriedades em conjunto? Por exemplo, na Figura 8.16, como dizer qual posição é mais fácil?

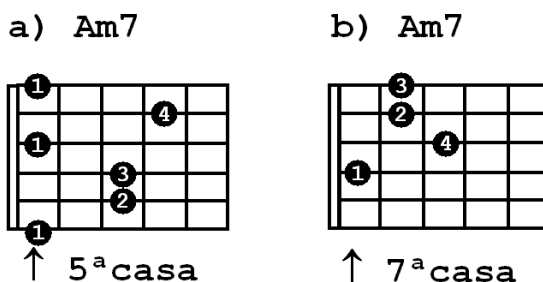


Figura 8.16 – Dois exemplos de posições de um Am7

A abordagem utilizada para ponderar os valores foi o de atribuir para cada situação uma penalidade. Deste modo, são definidas penalidades para a utilização de pestana, para a distância horizontal entre dedos consecutivos maior que 1 casa e para a distância horizontal entre dedos não consecutivos maior que 2 casas. Somando-se estes valores na Figura 8.16, por exemplo, a posição b sofre uma penalidade menor que a posição a.

O valor final obtido é utilizado como medida de “facilidade”. Entretanto, não foi possível um estudo estatístico dos resultados, verificando as taxas de acerto da ordenação, por exemplo, pois não houve um consenso entre os próprios músicos. É possível apenas uma validação do resultado com especialistas. Os resultados do *D’Accord Guitar* foram validados por 5 músicos, que testaram o software com seus próprios casos. Foi consenso entre eles que os primeiros acordes ordenados são mais fáceis do que os demais, e que a ordenação (ao menos a nível macro) é correta.

Flexibilidade

O segundo critério a avaliar trata da flexibilidade da posição. Quando um músico pretende improvisar ou elaborar um arranjo que exceda o simples posicionamento de acordes, ele deseja posições que deixem dedos livres, com o maior alcance possível. Este é um critério mais objetivo. Segundo as opiniões dos músicos consultados, ele pode ser definido, de forma simplificada, como a região alcançável pelos dedos livres. O maior problema, aqui, é justamente descobrir qual a região alcançável. Por exemplo, na primeira posição da Figura 8.17, em quantas coordenadas o músico pode tocar com os dedos livres? Por exemplo, na figura abaixo a primeira posição é mais flexível do que a segunda, pois o número de coordenadas alcançáveis é maior.

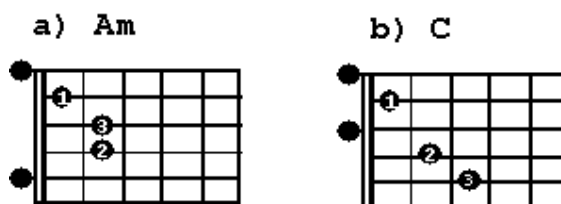


Figura 8.17 – Comparação de duas posições quanto à flexibilidade

Para resolver tal problema, foram reutilizadas as regras que definem as restrições anatômicas de posicionamento em conjunto com as restrições de digitação, como visto na seção 0. Assim, toda casa que forme uma posição exequível é considerada uma coordenada alcançável. Dado uma posição, é possível então encontrar o número de coordenadas alcançáveis. Na Figura 8.18 é mostrado o mesmo exemplo da Figura 8.17, com as coordenadas alcançáveis marcadas em cinza. A posição *a* possui 14 casas alcançáveis, sendo considerada menos flexível que a posição *b*, que possui 16.

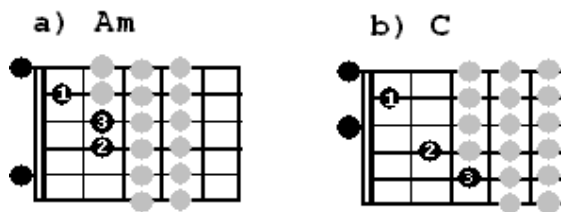


Figura 8.18 – Mesmo exemplo da Figura 8.17, com a identificação das coordenadas alcançáveis.

Frequência de Uso

É importante, dentro do processo de aprendizado, a memorização das posições mais comuns de acordes. Frequentemente, os músicos escolhem tocar determinadas posições apenas por serem as que ele conhece. É disto que trata o terceiro critério: frequência de uso.

Existem posições raras, comuns, clichês, entre outras. Porém, não há atributos que sugiram como comparar duas posições quanto às suas frequências de utilização. É um critério dito cultural, pois se baseia no hábito e na cultura existente. Como calcular automaticamente se uma posição e digitação de acorde encontrada pelos algoritmos de posicionamento e digitação é comum?

Uma possível abordagem seria obter um grande conjunto de músicas editadas e efetivamente contabilizar quantas ocorrências existem de cada posição. Esta abordagem, no entanto, foi considerada inviável a curto prazo. Ao que parece, não é possível calcular este valor, ele tem que ser atribuído de alguma forma por um ser humano. A alternativa escolhida foi redefinir as informações existentes na *base de acordes* do *D'Accord Guitar*, para que incluam atributos extra, indicando explicitamente os valores atribuídos por um ser humano para cada critério. Quem edita a base edita, portanto, o atributo de *frequência de uso*, de forma qualitativa: *raro*, *médio*, *comum*, e *clichê*. O problema é que a base de acordes possui 2888 posições e digitações, um número muito pequeno quando comparado ao número de casos possíveis (casos gerados pela etapa de posicionamento e digitação). É preciso, então, tratar o caso em que a posição não exista na base. Quanto à frequência de uso, esta posição (que não existe na base de acordes) é classificada como desconhecida, ficando um nível ainda abaixo de *raro*.

A abordagem utilizada traz a vantagem da personalização, já que deixa a cargo do usuário a edição dos acordes que utiliza mais frequentemente. Caso o usuário não deseje realizar este trabalho, a base já é inicializada com valores *default*.

Já que se baseia na própria escolha do usuário, é garantida a qualidade da avaliação quanto a esse critério. Além disso, é personalizável, já que cada usuário pode editar os valores de acordo com sua preferência.

Sonoridade

O quarto critério refere-se à qualidade sonora da posição, e configura-se como o mais polêmico, por conter um grande conjunto de justificativas no seu processo decisório. Ele

busca responder à seguinte pergunta: o que é um acorde que soa bem? Ou, mais explicitamente: qual acorde, dentre um conjunto dado, soa melhor?

Este critério inclui conceitos altamente subjetivos, como beleza, sonoridade e nível de tensão. Não somente é um parâmetro quantitativo (o quanto é bom?) quanto qualitativo (que sensação provoca?). Pelo alto grau de dificuldade envolvido na modelagem deste critério, decidimos definir a sonoridade como sendo o gosto pessoal do usuário por determinadas posições existentes na base de acordes. A sua solução segue os mesmos moldes da frequência de uso: inserir um atributo extra na base de acordes, indicando explicitamente o gosto pessoal pela sonoridade de cada posição.

Facilidade de transição

O objetivo final do *D'Accord Guitar* é encontrar a melhor execução violonística de uma sequência de cifras de acordes. Para tanto, não é suficiente observar apenas os critérios referentes a cada posição de acorde isolada. É preciso estender o contexto para englobar as melhores progressões de acordes. Ou seja, é preciso levar em consideração os acordes anteriores e posteriores. Para ilustrar tal fato, tomemos as digitações exibidas na Figura 8.13. Se o próximo acorde é um *Em7* na posição exibida na Figura 8.19, a digitação da Figura 8.13e deve ser utilizado.

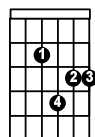


Figura 8.19 – Um possível digitação para uma posição de *Em7*

Os critérios vistos aqui eram inerentes ao acorde, podiam ser calculados acorde por acorde, isoladamente. Porém, esta avaliação isolada de uma posição é algo que não faz muito sentido. Geralmente, em música, mais importante do que cada acorde separadamente é a sua relação com os acordes anterior e posterior (Fowler 1984b). Nesta relação baseia-se boa parte do processo de escolha. Músicos experientes procuram tocar acordes em posições parecidas, para que não precisem fazer mudanças bruscas ou para dar uma sensação de suavidade ao acompanhamento. É um consenso geral que a facilidade de transição é um dos mais importantes critérios no processo de escolha de posições para uma determinada cifra. Esta avaliação, obviamente, não é feita para cada posição de acorde, e sim para cada transição e, portanto, é importante diferenciá-la da facilidade do posicionamento (critério visto na seção 0). A avaliação facilidade de transição trata de responder à seguinte pergunta: dada uma

posição para o primeiro acorde e um conjunto de possíveis posições para o segundo acorde, qual a transição mais fácil de executar? Um exemplo pode ser visto na Figura 8.20.

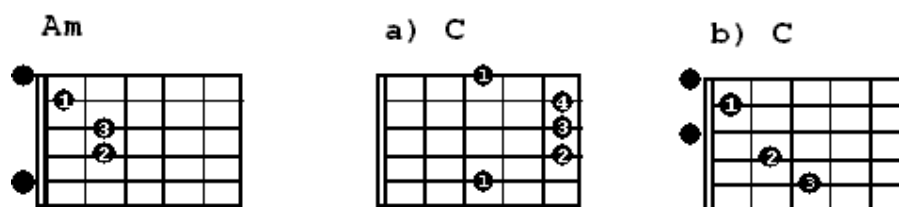


Figura 8.20 – Comparação de duas posições quanto à similaridade com uma primeira

O problema, então, é descobrir quais parâmetros podem ser empregados para sugerir esta facilidade. Os parâmetros encontrados foram:

1. A posição básica é mantida?
 - 1.1. Se sim, na mesma corda ou em cordas diferentes?
 - 1.2. Se não, elas estão a que distância?
2. Quantos dedos mantêm a coordenada?
3. Quantos dedos mantêm a corda?

A pergunta 1 informa que posições como as ilustradas na Figura 8.21 são similares. O F e o G mantêm a mesma posição básica, na mesma corda, e estão a 2 casas de distância. O F e o Bbm mantêm a mesma posição básica, porém em cordas diferentes. A distância entre as casas é nula.

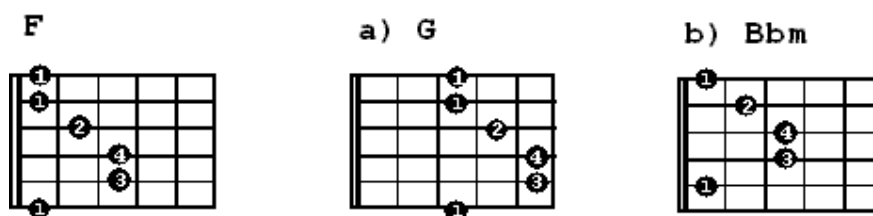


Figura 8.21 – Comparação de 2 posições quanto à facilidade de transição com a primeira

A pergunta 2 indica um outro indicador de facilidade de transição encontrado, por exemplo, nas posições da Figura 8.22. Este indicador é especialmente útil para músicos iniciantes, que freqüentemente usam os dedos que permanecem na mesma coordenada para servirem de “ponte” no momento da troca do acorde. Tais músicos normalmente têm

dificuldade de posicionar rapidamente os dedos nas coordenadas, e com esta ponte eles evitam se perder e trocam de acorde de forma muito mais rápida. Na Figura 8.22 as coordenadas que são mantidas, servindo de ponte, são marcadas.

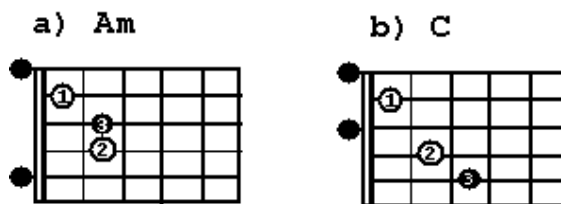


Figura 8.22 – Exemplo de duas posições que possuem uma coordenada como ponte

Estas 3 perguntas denotam 3 níveis de similaridade. No primeiro caso, é mantida a armação da mão, bastando deslocá-la. No segundo caso, são mantidas as coordenadas que servem de ponte. Por fim, a pergunta 3 indica um último grau de similaridade, em que os dedos não são mantidos nas coordenadas, apenas nas cordas. Na Figura 8.23, por exemplo, todos os 4 dedos mantêm-se nas cordas.

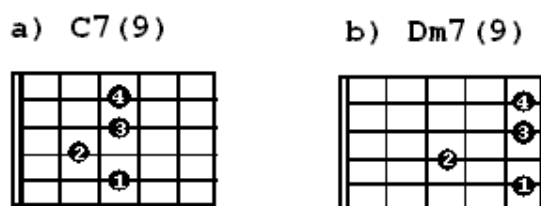


Figura 8.23 – Exemplo de duas posições que mantêm as cordas pressionadas por cada dedo

Com os parâmetros de posição básica, coordenada ponte e número de dedos que se mantêm em coordenadas ou cordas, foi possível especificar regras para definir um algoritmo, de ordem parcial, que comparasse duas posições e dissesse qual tinha a transição mais fácil para uma primeira. Na Figura 8.24 abaixo podem ser vistas 4 posições. Deseja-se saber qual, dentre as posições *b*, *c*, *d* e *e*, possui a transição mais fácil a partir da posição *a*. As posições *b* e *d* mantêm a mesma posição básica, sendo consideradas mais similares do que *c* e *e*. A posição *d* é considerada mais similar a *a* do que a posição *b* porque a distância é menor. Da mesma forma, a posição *c* é considerada mais similar do que a posição *e*, através da ponderação dos 3 atributos citados anteriormente. A ordenação das posições segundo sua similaridade é, portanto: *d*, *b*, *c*, *e*.

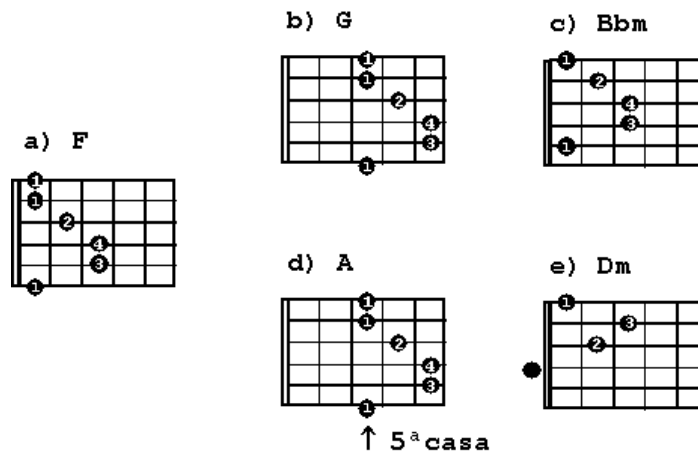


Figura 8.24 – Comparação entre 4 posições quanto à similaridade com a primeira

Linha de Baixo

O critério de linha de baixo visa dar prioridade a posições onde a linha melódica executada pelo baixo (nota mais grave do acorde) seja a mais suave possível. Este também é um critério envolvendo a transição de acordes, e não acordes isolados. O objetivo é encontrar acordes onde seus baixos mantenham-se em notas próximas. Portanto, o atributo a verificar é o de distância entre os baixos. Esta distância refere-se à distância musical, sendo medida como o número de semitons entre as notas.

Linha Melódica

A linha melódica é um problema mais complexo do que a escolha de linha de baixo, pois geralmente leva em consideração não só a proximidade entre as notas, quanto a própria melodia gerada. No *D'Accord Guitar* foi feita uma simplificação, onde a linha melódica assemelha-se à linha de baixo, porém ao invés de observar a nota mais grave, observa a nota mais aguda de cada posição.

8.4.2. Quadro Comparativo

Abaixo é feito um quadro comparativo dos diversos critérios utilizados:

Critério	Escopo	Tipo	Método
Facilidade de Posicionamento	Do acorde	Anatômico	Penalidades sobre todos os subcritérios
Flexibilidade	Do acorde	Anatômico	Único subcritério: número de coordenadas alcançáveis
Freqüência de Uso	Do acorde	Cultural	Base de dados
Sonoridade	Do acorde	Cultural/Musical	Base de dados

Facilidade de Transição	Da transição	Anatômico/Musical	Regras sobre os subcritérios
Linha de Baixo	Da transição	Musical	Único subcritério: distância musical
Linha Melódica	Da transição	Musical	Único subcritério: distância musical

Tabela 4 – comparação dos diversos critérios em relação a seu escopo, tipo, método com foi resolvido e modelo de valoração

Os diversos critérios envolvem conhecimentos musicais, culturais e uma modelagem anatômica do músico, e foram estimados utilizando diferentes técnicas. Os critérios de facilidade, flexibilidade, frequência de uso e sonoridade são propriedades dos acordes. Encontrar as melhores posições para uma sequência de cifras de acordes usando um desses critérios resume-se a encontrar a melhor posição para cada cifra, isoladamente. O algoritmo funcionaria da seguinte forma: para cada cifra dada, seriam geradas suas posições e digitações. Cada par [posição, digitação] seria avaliado segundo os critérios discutidos aqui (a *facilidade de posicionamento*, a *flexibilidade*, a *facilidade de transição* e as os critérios de *linha de baixo* e *melódico*, automaticamente, e a *frequência de uso* e a *sonoridade* através da base de dados).

8.5. *Etapa 3: escolha multi-critérios*

Com as duas etapas anteriores cumpridas, é possível escolher as melhores posições e digitações para uma sequência de cifras, desde que observe apenas um único critério. Porém, este é um caso artificial. Na prática, a escolha é baseada em diversos critérios em conjunto, alguns com maior importância, outros com menor. Porém, como levar em consideração mais de um critério ao mesmo tempo?

A idéia básica é que seja configurado um peso para cada critério, e este peso sirva para gerar uma medida ponderada de “qualidade” do acorde. Porém, cada critério utiliza uma medida diferente, que varia em intervalos diferentes, como mostra a Figura 8.25. Nesta figura é visto que, enquanto a penalidade da facilidade varia entre 0 e 22, o valor encontrado para a flexibilidade varia entre 0 e 72.

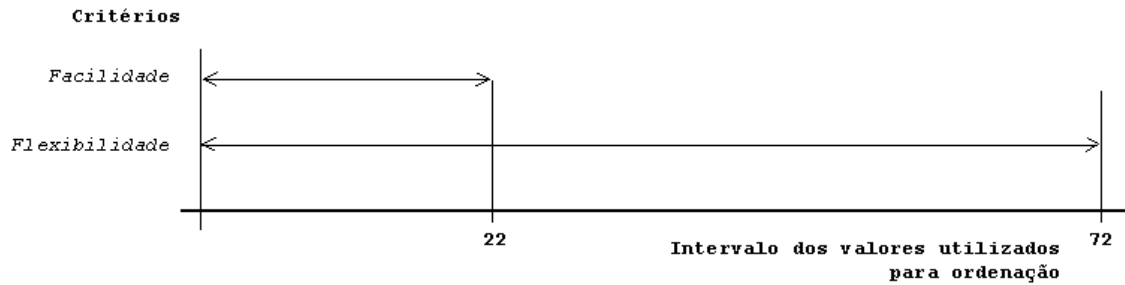


Figura 8.25 - Intervalos dos valores possíveis para cada critério

Além disso, a ordenação de alguns critérios, como o de facilidade de transição, é de ordem parcial, e não de ordem total. Isto quer dizer que as posições podem ser comparadas, mas não há um valor numérico para a facilidade transição que possa ser normalizado. Adicionalmente, os critérios de frequência de uso e de sonoridade foram classificados manualmente, em 5 níveis. A abordagem utilizada para “normalizar” estes critérios foi o de discretizar todos eles em 5 níveis equiparáveis (do nível 0 ao nível 4), para que possam ser ponderados. A Figura 8.26 mostra uma possível equiparação entre a facilidade e a flexibilidade. Para equiparar todos os critérios é necessário normalizar os valores dos diversos critérios, idealmente de forma automática.

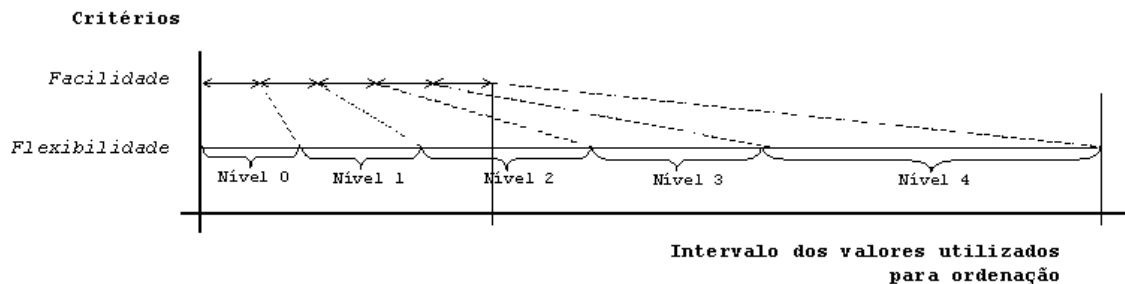


Figura 8.26 – Equiparação dos níveis de classificação para os diversos critérios

Para ilustrar o funcionamento desejado, tomemos como base a Figura 8.27, que exhibe possíveis valores dos critérios citados para as posições e digitações de uma posição de *Am7*. Os exemplos *a* e *b* são diferentes digitações para uma mesma posição, assim como os exemplos *c* e *d*. Dado que os níveis estão equiparados, é possível ordenar as posições e digitações levando em consideração os diversos critérios em conjunto.

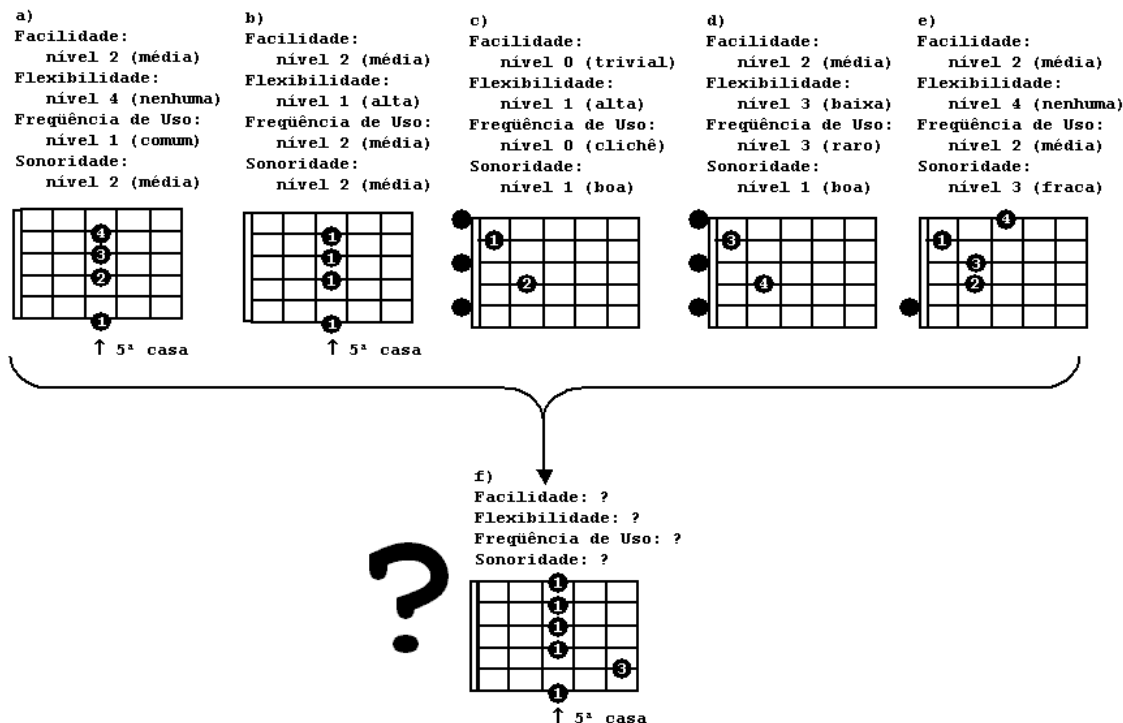


Figura 8.27 – Idéia central do aprendizado de máquina supervisionado: utilizar exemplos avaliados por um ser humano para avaliar novos exemplos

O problema encontra-se, então, em como descobrir automaticamente em que nível cada posição (e digitação) está. Acreditamos que o procedimento ideal seja preencher um grande conjunto de exemplos com a classificação feita por especialistas humanos sobre os diversos critérios, em várias posições, e usar um algoritmo de aprendizado de máquina para que novas posições possam ser classificadas. O aprendizado seria útil, inclusive, para permitir que o processo de interpretação fosse personalizável e adaptável. Em outras palavras, para que o usuário pudesse editar critérios sobre as diversas posições de acordes e o sistema pudesse captar seu processo de avaliação. Porém, algumas dificuldades foram encontradas e, para que conseguíssemos terminar uma primeira versão do sistema em tempo hábil, nem sempre o aprendizado de máquina foi utilizado.

Alguns critérios são culturais, aparentemente sem justificativa para cada escolha. São os critérios de frequência de uso e de sonoridade, que representam respectivamente um hábito e um gosto pessoal. A classificação de tais critérios foi baseada exclusivamente nas escolhas existentes na base de acordes.

Além disso, alguns critérios exigem a aquisição de um volume de dados de treinamento muito grande para que sejam rodados algoritmos de aprendizado de máquina. São

os critérios referentes à transição (facilidade de transição, linha de baixo e linha melódica) e foram classificados de acordo com regras explícitas definidas por especialistas.

Nos demais critérios, foi testada a idéia de utilizar aprendizado de máquina, confrontando-a com a abordagem de definição explícita de regras. Para isto, foram implementados algoritmos simples para aprender a classificar, baseando-se em escolhas dos usuários. A seguir será vista a implementação dos classificadores automáticos de posições de acorde de acordo com cada critério, as técnicas concebidas e a solução final. Planejamos um estudo sistemático, em breve, utilizando outros métodos de aprendizado.

8.5.1. Flexibilidade

Deseja-se, com a classificação da flexibilidade, detectar que uma posição está em um dos seguintes níveis: trivial, fácil, médio, difícil e inviável. Tal classificação toma um único atributo como critério: o número de coordenadas alcançáveis pelos dedos que estão livres. O objetivo é encontrar uma função f , onde x é o número de casas, e $f(x)$ é o nível desejado. Para efetuar esta classificação, é preciso encontrar os limiares entre os níveis, como na Figura 8.28.

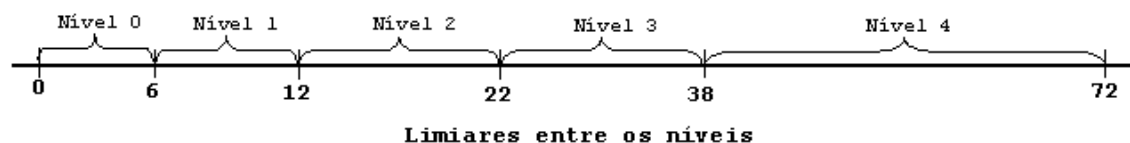


Figura 8.28 – Possíveis limiares entre os níveis de classificação

Uma primeira abordagem, sem aprendizado, trata de estipular manualmente (pelo engenheiro de conhecimento) valores de limite para o número de coordenadas alcançáveis. De acordo com o intervalo em que está, a posição é classificada em um dos níveis desejados. Estes valores foram definidos a partir de um conjunto de dados, constando de 150 posições e digitações de acordes, preenchidos com as classificações em um dos níveis dados, feitas por 4 músicos consultados. A definição de como é feita a classificação foi feita por um ser humano, e baseou-se na análise estatística dos dados, estipulando valores de forma a maximizar a taxa de acerto. Os dados encontrados foram os seguintes:

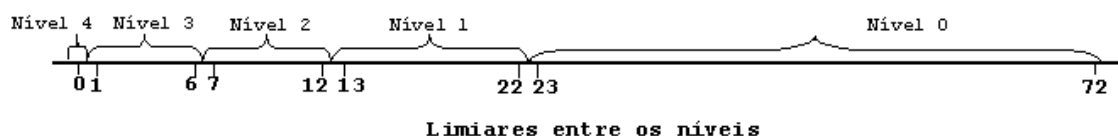


Figura 8.29 – Classificação manual

Uma segunda abordagem trata de automatizar este processo, implementando um sistema que extraia os limiares do conjunto de dados. Seguindo esta abordagem, visualizamos 4 técnicas que poderiam ser usadas. A primeira técnica é extremamente simples: divide-se a penalidade por 5, encontrando 5 intervalos iguais (Figura 8.30). Neste caso, para haver um bom funcionamento, é suposto que há uma distribuição uniforme dos dados, o que não ocorre nos casos observados. O número de posições consideradas sem nenhuma flexibilidade, por exemplo, é muito grande. Na Figura 8.30 é mostrado um exemplo, na forma de histograma.

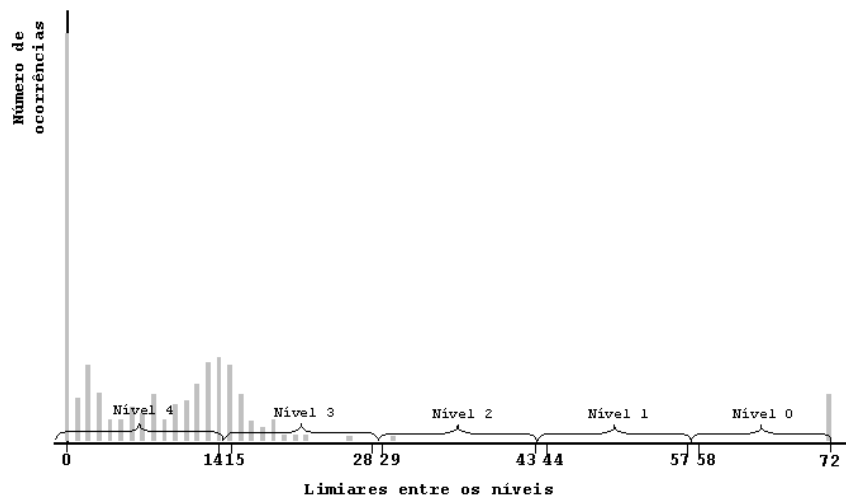


Figura 8.30 – Histograma de ocorrências de um conjunto de dados e classificação baseada na divisão em 5 intervalos iguais

A segunda técnica divide o conjunto de casos em 5 grupos de igual tamanho (Figura 8.31). O limiar é então “aprendido”, de acordo com os casos, e generalizado. O limiar encontrado é usado para os novos casos apresentados.

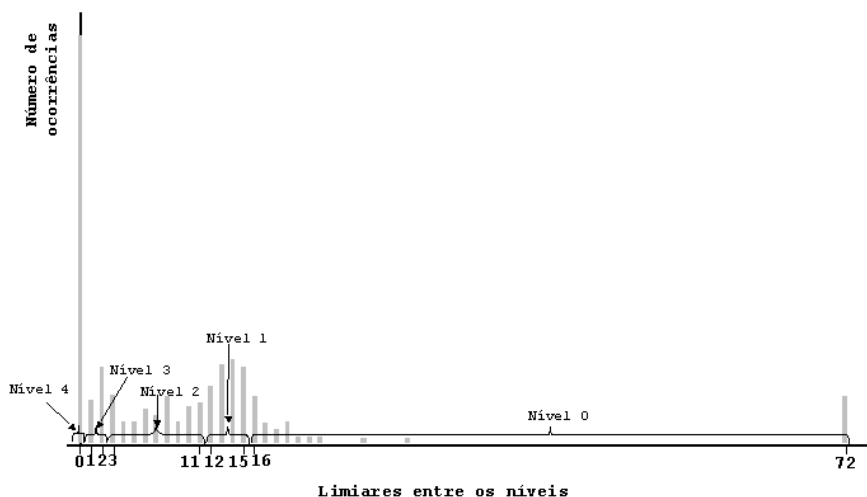


Figura 8.31 – Classificação baseada na divisão em 5 intervalos igualmente povoados

Porém, neste caso, é assumido que há o mesmo número de casos em cada nível, o que não ocorre na prática. Na Figura 8.32 pode-se ver que o nível 4, por exemplo, possui muito mais casos do que os demais, pois como boa parte das posições de acordes utiliza todos os dedos, tornando o número de coordenadas alcançáveis igual a 0.

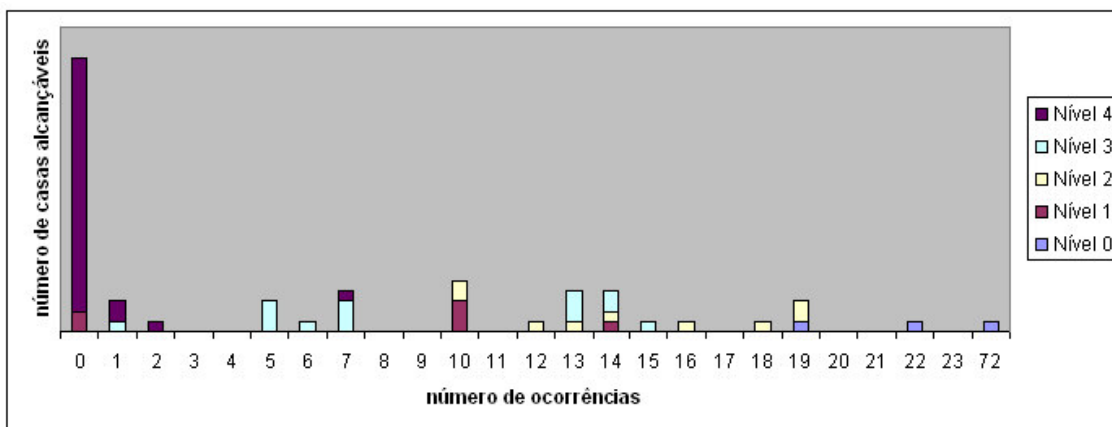


Figura 8.32 – Histograma com o número de ocorrências em cada nível

A terceira técnica computa as porcentagens de casos em cada um dos níveis e usa isto como probabilidade de ocorrência. Portanto, ao invés de dividir em 5 grupos com o mesmo número de casos, divide proporcionalmente à sua probabilidade. A Figura 8.33 ilustra os limiares encontrados para os mesmos exemplos anteriores.

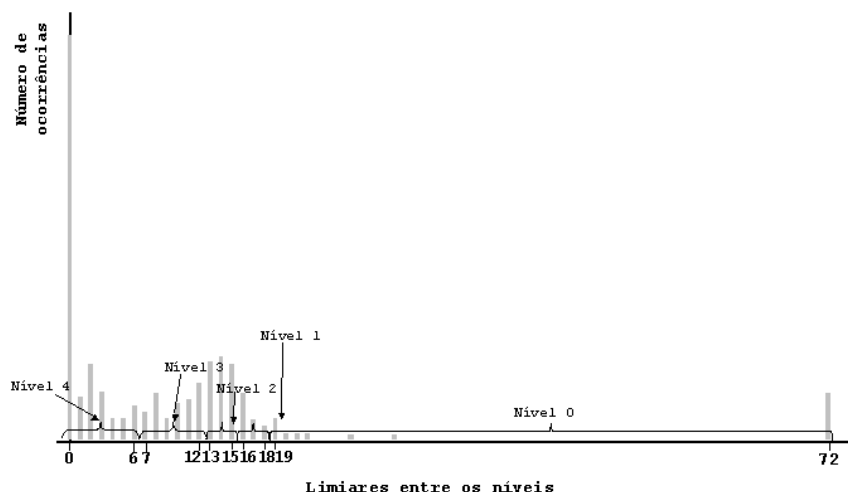


Figura 8.33 – Classificação baseada na divisão em 5 grupos povoados de acordo com sua probabilidade

Uma quarta técnica possível é efetuar uma busca para encontrar limiares que maximizem o número de casos classificados corretamente. A Figura 8.34 ilustra os limiares encontrados para os mesmos exemplos anteriores.

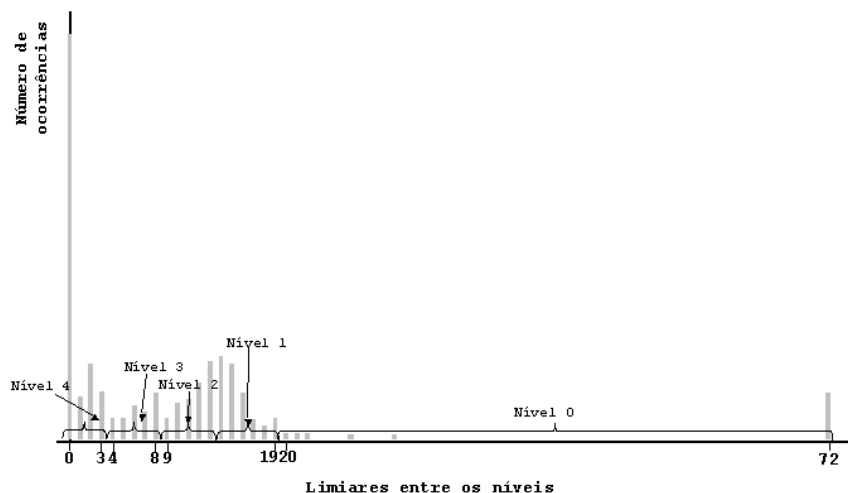


Figura 8.34 – Classificação baseada na minimização do erro

Quadro comparativo

Foi efetuado um teste sobre cada uma das abordagens, com todas as posições da base de acordes que possuíam como tônica o Lá. Os exemplos foram separados em conjunto de treinamento e conjunto de teste de tamanhos iguais. O conjunto de treinamento consistiu das demais posições existentes na base. Abaixo, é mostrada uma tabela com os erros de classificação encontrados.

	Grupos Iguais	Grupos Iguualmente Povoados	Grupos Probabilística mente Povoados	Minimização do Erro
Classificado Corretamente	62,20%	63,41%	67,07%	76,83%
Erro de classificação igual a 1	18,29%	10,98%	19,51%	18,29%
Erro de classificação igual a 2	10,98%	17,07%	8,54%	2,44%
Erro de classificação igual a 3	8,54%	8,54%	4,88%	2,44%
Erro de classificação igual a 4	0,00%	0,00%	0,00%	0,00%

Tabela 5 – Comparação dos resultados obtidos com os diferentes métodos de classificação

Na primeira linha é exibida a porcentagem de exemplos classificados corretamente. Na segunda linha é exibida a porcentagem de exemplos que foram classificados incorretamente, mas cuja diferença entre o resultado obtido e o resultado esperado foi igual a 1. Da mesma forma, a terceira, quarta e quinta linhas mostram as porcentagens de exemplos com erros de classificação iguais a 2, 3 e 4, respectivamente.

De acordo com os resultados, o método de minimização do erro possui um desempenho superior, principalmente quando se leva em consideração o reduzidíssimo número de exemplos onde o erro de classificação é grosseiro. O *D'Accord Guitar* utiliza este método.

8.5.2. Facilidade do posicionamento

A facilidade segue um caso semelhante ao da flexibilidade, onde é preciso encontrar limiares que dividam os valores encontrados nos cinco grupos. Novamente, é possível utilizar a abordagem de cálculo do limiar, tal qual ocorreu com a flexibilidade. Porém, é feita uma ressalva: as técnicas citadas na classificação da flexibilidade são baseadas na penalidade. No entanto, o próprio cálculo do valor da facilidade do posicionamento levou em consideração 4 subcritérios. Portanto, faz mais sentido que a classificação da facilidade não seja feita baseada na penalidade, e sim nos subcritérios que foram utilizados para calculá-la: distância horizontal entre dedos, distância vertical entre dedos, utilização de pestana e região em que a posição se encontra.

Neste cenário são extremamente adequadas as técnicas tradicionalmente denominadas de aprendizado de máquina, como *redes neurais* (RN), *ID3*, *redes bayesianas* e *raciocínio baseado em casos* (CBR), que usam diferentes paradigmas para resolver um problema em comum: aprender, baseado em exemplos. A idéia é utilizar os exemplos classificados manualmente pelos músicos para que o sistema aprenda a classificar novos exemplos, em algum dos níveis possíveis. O algoritmo utilizado foi uma variação do KNN

(Dasarathy 1991, Wettschereck 1994), que embora seja extremamente simples, serve para comparar os resultados encontrados com as duas abordagens (aprendizado de máquina e engenharia do conhecimento).

O algoritmo funciona da seguinte maneira: dado um conjunto de k critérios, podendo assumir valores em um dos diferentes níveis e uma base de exemplos manualmente classificados, é gerada uma matriz de dimensão $k+1$, armazenando o número de classificações em cada nível para cada combinação de valores dos diversos critérios. Por exemplo, caso existam apenas 2 critérios, x e y , podendo ser classificados em 5 níveis (de 0 a 4), e a seguinte base de exemplos:

```
Exemplo 1 : x = 1, y = 1, classificação = nível 2
Exemplo 2 : x = 1, y = 2, classificação = nível 4
Exemplo 3 : x = 0, y = 2, classificação = nível 0
Exemplo 4 : x = 0, y = 0, classificação = nível 1
Exemplo 5 : x = 3, y = 1, classificação = nível 2
Exemplo 6 : x = 2, y = 1, classificação = nível 4
Exemplo 7 : x = 1, y = 0, classificação = nível 1
Exemplo 8 : x = 2, y = 1, classificação = nível 1
Exemplo 9 : x = 1, y = 2, classificação = nível 0
Exemplo 10 : x = 3, y = 2, classificação = nível 3
Exemplo 11 : x = 0, y = 1, classificação = nível 2
Exemplo 12 : x = 1, y = 0, classificação = nível 2
Exemplo 13 : x = 1, y = 1, classificação = nível 0
Exemplo 14 : x = 1, y = 1, classificação = nível 2
Exemplo 15 : x = 1, y = 0, classificação = nível 2
```

Figura 8.35 – Uma possível base de exemplos

A matriz gerada é mostrada a seguir. Cada célula contém um gráfico que indica o número de ocorrências de cada classificação para o par x, y .

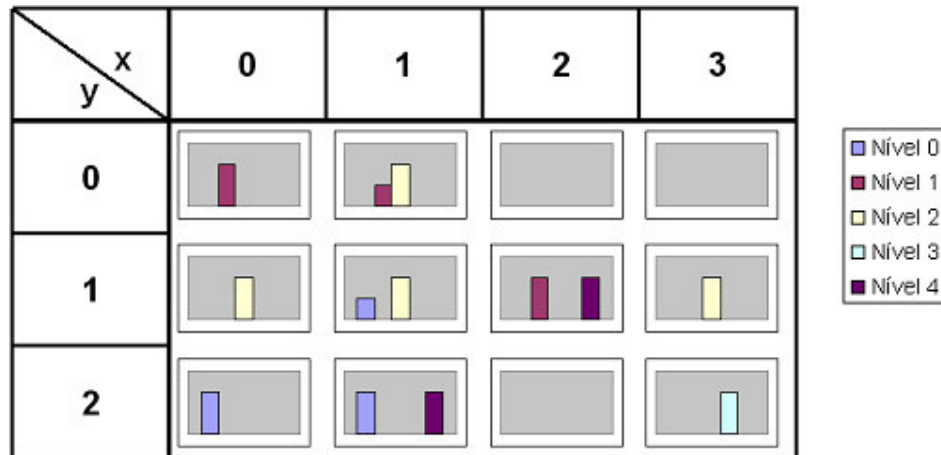


Figura 8.36 – Matriz com a base de exemplos gerada

Dada esta matriz, um novo exemplo pode ser classificado pela simples indexação nesta tabela. O nível que tiver maior ocorrência é retornado como a classificação do novo exemplo. Caso exista mais de um nível com o mesmo número de ocorrências, são consultados os seus vizinhos a uma distância k , que inicialmente é igual a 1 (células adjacentes na matriz). O valor deste k aumenta a cada interação até ocorrer um desempate.

No exemplo da Figura 8.36, caso surja um novo exemplo onde o x seja igual a 0 e o y igual a 1, ele será classificado como nível 2, pois é a única resposta existente. Caso o x seja igual a 1 e o y igual a 1, ele será classificado também como nível 2, já que há duas ocorrências desta classificação, contra uma classificação como nível 0. Caso o x seja igual a 1 e o y igual a 2, inicialmente (com k igual a 0) ocorrerá um empate entre os níveis 0 e 4. O k é, então, incrementado para 1, e serão observados seus vizinhos, o que classificará o novo exemplo como nível 0.

Quadro comparativo

Foram efetuados experimentos também com a classificação da *facilidade*, utilizando a mesma metodologia da *flexibilidade*. Foi comparada a abordagem de cálculo do limiar (baseado na penalidade encontrada, utilizando o algoritmo de minimização do erro, tal qual ocorre com a *flexibilidade*) com a abordagem de aprendizado de máquina, resultando nas seguintes taxas de erro e acerto.

	Limiar	Aprendizado
Classificado Corretamente	50,58%	54,22%
Erro de classificação igual a 1	41,28%	43,37%
Erro de classificação igual a 2	6,40%	2,41%
Erro de classificação igual a 3	1,74%	0,00%
Erro de classificação igual a 4	0,00%	0,00%

Tabela 6 - Comparação dos resultados obtidos com os diferentes métodos de classificação

Os resultados comprovam que, mesmo com um algoritmo de aprendizado extremamente simples, é possível um ganho considerável na qualidade da classificação. Não só o número de exemplos classificados corretamente aumentou, quanto o número de classificações grosseiras tornou-se mínimo.

8.5.3. Frequência de uso e sonoridade

Estes dois critérios são, no mínimo, muito complicados de automatizar, por serem “culturais”, frutos do hábito ou de gosto pessoal de cada um. São subjetivos, sem justificativas claras para cada escolha. A alternativa implementada no *D'Accord Guitar* foi a de utilizar os valores definidos para estes atributos, na base de acordes, pelos especialistas, para efetuar esta classificação. Como os valores definidos pelos músicos já são uma classificação, a solução é bastante simples. O valor do critério de frequência de uso é o valor encontrado na base de dados (*raro*, *médio*, *comum* e *clichê*). Caso o valor não seja encontrado na base, ele é definido como *desconhecido*, situando-se abaixo do nível *raro*. No caso da sonoridade, os valores possíveis na base de exemplos são: *péssimo*, *regular*, *bom* e *excelente*. Quando ele não existe na base, é classificado como *desconhecido*, ficando entre os níveis *péssimo* e *regular*. Caso o usuário não deseje considerar os casos desconhecidos, ele pode escolher a opção *utilizar apenas acordes existentes na base*.

8.5.4. Facilidade de transição

As técnicas de aprendizado são muito apropriadas, mas enfrentam frequentemente um problema bastante grave: a dificuldade de aquisição de exemplos. No caso dos critérios “do acorde”, esta aquisição é simples, bastando classificar manualmente cada acorde. No caso de critérios “da transição”, esta aquisição é mais complexa, pois cada critério precisa ser configurado para pares de posições de acorde. O número de possibilidades, portanto, é igual ao quadrado do número de posições existentes. Para obter um conjunto suficientemente grande para generalizar a classificação é preciso um volume de dados impraticável com a interface de aquisição atual. Para os critérios de facilidade de transição, linha de baixo e linha

melódica, precisamos definir regras explícitas para classificar as posições e digitações em níveis, de forma a obtermos uma primeira solução implementada. Como trabalho futuro está a utilização de aprendizado de máquina também para estes critérios. A seguir será vista a abordagem utilizada para a classificação automática de posições e digitações quanto à facilidade de transição.

A classificação da facilidade de transição entre duas posições informa se a posição, as coordenadas ou as cordas são mantidas. Estes parâmetros, em conjunto como número de casas de dist foram então utilizados para classificar as posições em 5 níveis de similaridade. O resultado foi uma classificação nos seguintes níveis:

- Nível 0: perto, mantendo a posição básica, na mesma corda
- Nível 1: perto, mantendo coordenadas ou cordas
- Nível 2: perto, sem manter nenhuma relação de semelhança
- Nível 3: longe, mantendo a posição básica, coordenadas ou cordas
- Nível 4: longe, sem manter nenhuma relação de semelhança

Nos níveis acima, perto é definido como uma distância menor do que 5 casas, e longe uma distância maior ou igual a 5 casas. Na figura abaixo podem ser vistas as posições ilustradas na seção 0, e as classificações do nível de similaridade para cada par.

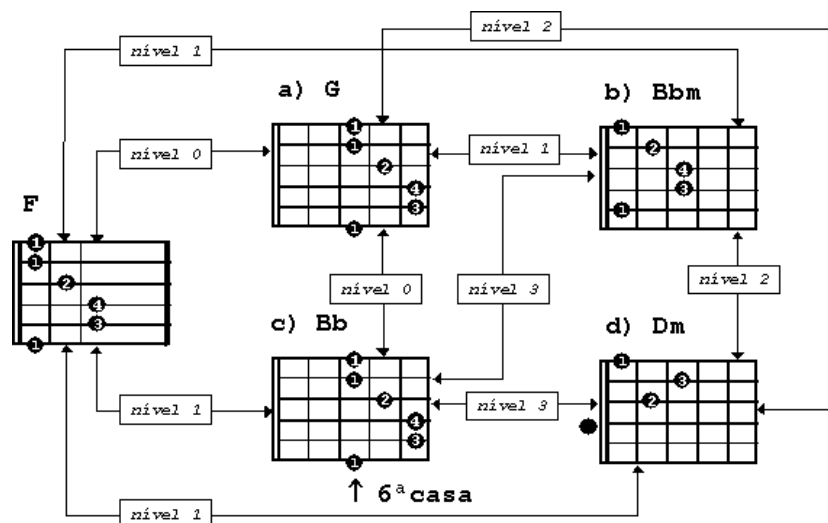


Figura 8.37 – Exemplo de classificação da similaridade entre algumas mudanças de posições

8.5.5. Linha de baixo e melódica

Como os critérios de linhas de baixo e linha melódica baseiam-se em um único critério, foi necessário apenas encontrar os limiares que dividiam os níveis de classificação. Nós definimos os valores destes limiares, a princípio, explicitamente. Como trabalho futuro está prevista a utilização de aprendizado de máquina também para esta classificação.

- Nível 0: (distância = 0)
- Nível 1: ($1 \leq \text{distância} \leq 4$)
- Nível 2: ($5 \leq \text{distância} \leq 9$)
- Nível 3: ($10 \leq \text{distância} \leq 14$)
- Nível 4: ($15 \leq \text{distância}$)

Na Figura 8.38 são mostradas os níveis em que são classificados, com relação à linha de baixo, os mesmos pares de posições da figura anterior.

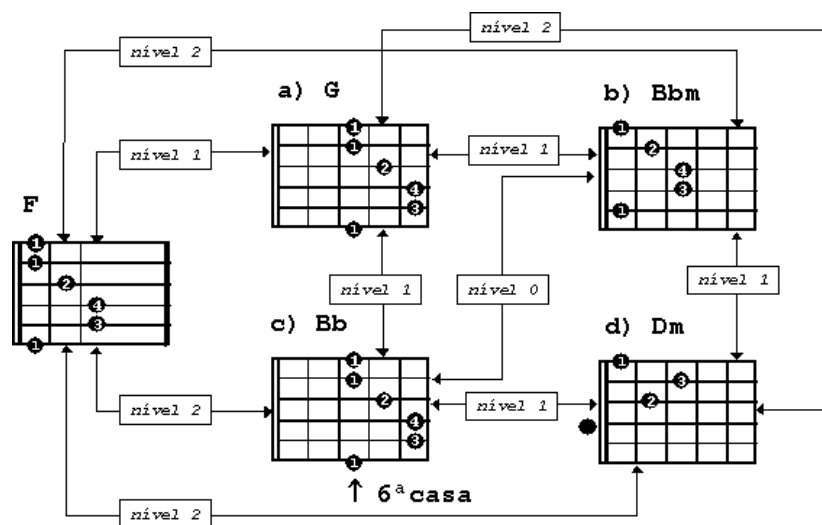


Figura 8.38 - Exemplo de classificação (quanto à linha de baixo) entre algumas posições

8.6. Escolha final de posições e digitações

Tomando com base a avaliação e a normalização de cada atributo, foi possível implementar um algoritmo para buscar as melhores posições e digitações, de acordo com a preferência do usuário sobre certos critérios. A seguir será visto o algoritmo de busca inicial, englobando as etapas descritas até aqui (geração, avaliação de posições e digitações) e, logo após, algumas melhorias realizadas para gerar o modelo final.

8.6.1. Algoritmo de busca

Na Figura 8.39, é mostrado o funcionamento do algoritmo geral. Linhas tracejadas são interações opcionais, podendo ocorrer ou não no processo. Linhas contínuas são interações que sempre ocorrem.

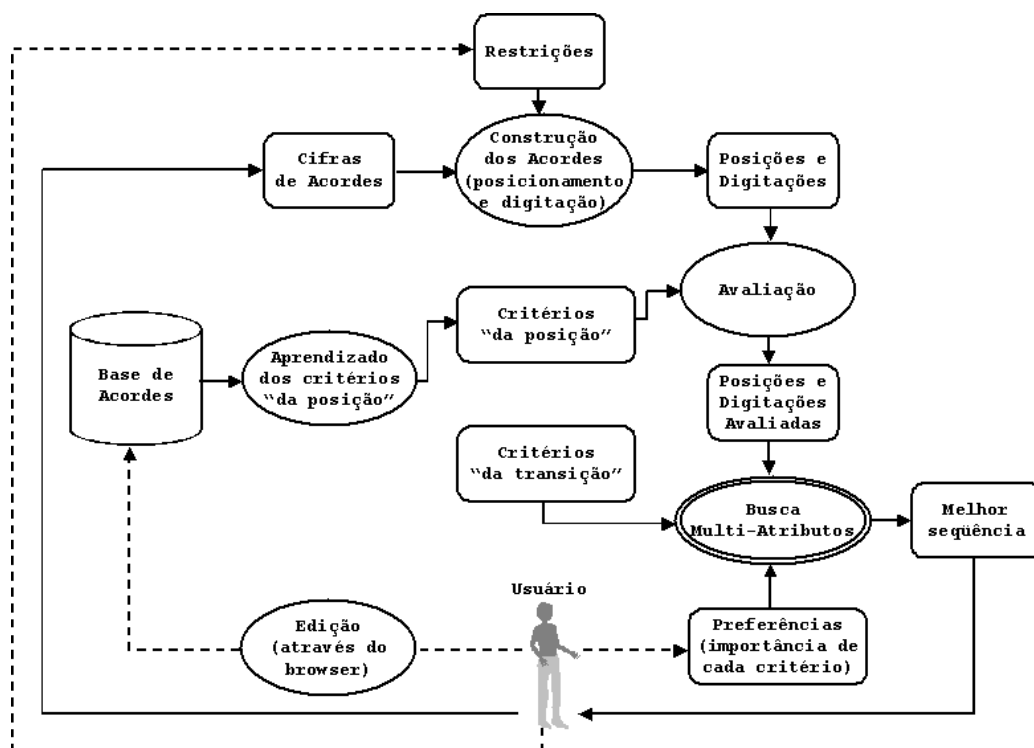


Figura 8.39 – Algoritmo de busca utilizado

No início, é fornecida uma *seqüência de cifras* e, se desejar, o *usuário* pode alterar as *restrições default* sobre os acordes a serem gerados. Para cada cifra são procuradas todas as posições que obedeçam a estas restrições e, para cada posição, são encontradas suas digitações. Cada par <posição, digitação> tem sua facilidade, flexibilidade, freqüência de uso e sonoridade classificadas (em 1 dos 5 níveis existentes) e normalizadas de acordo com os exemplos existentes na *base de acordes*.

Neste ponto é possível realizar a busca pelas melhores posições e digitações, através de outra busca (representada na Figura 8.39 como busca *multi-atributos*). O custo utilizado nesta busca é calculado como o custo dos nós (custo dos critérios “*da posição*”) somado ao custo das transições (custo dos critérios “*da transição*”). Na Figura 8.40 é dado um exemplo de espaço de busca para a seqüência de acordes *Am*, *D7*, *G*, e *Em*. São mostradas algumas das posições e digitações encontradas, cada uma com 4 números que indicam, respectivamente, os valores dos critérios de *facilidade*, *flexibilidade*, *freqüência de uso* e *sonoridade* (seqüências

de 4 números, com fonte em estilo normal). São mostrados também os valores dos critérios de *facilidade*, *linha de baixo* e *linha melódica* para algumas transições (seqüências de 3 números, com fonte em itálico). As reticências indicam que há muitas outras possibilidades.

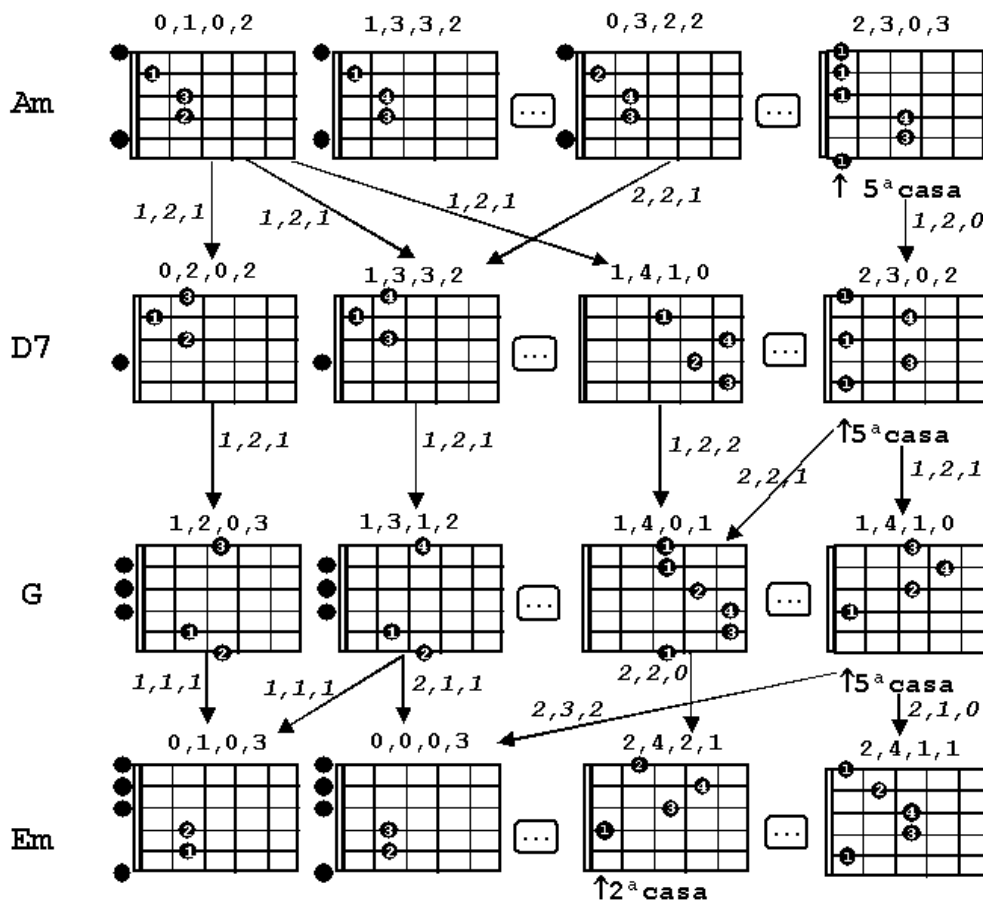


Figura 8.40 – Exemplo de árvore de busca gerada

A busca é chamada de *multi-atributos* pois os custos são, também, ponderados por pesos que o *usuário* define para os diversos critérios, através de suas *preferências*. O resultado desta busca é uma seqüência de posições de acordes que são dadas como resposta ao usuário (Figura 8.41).

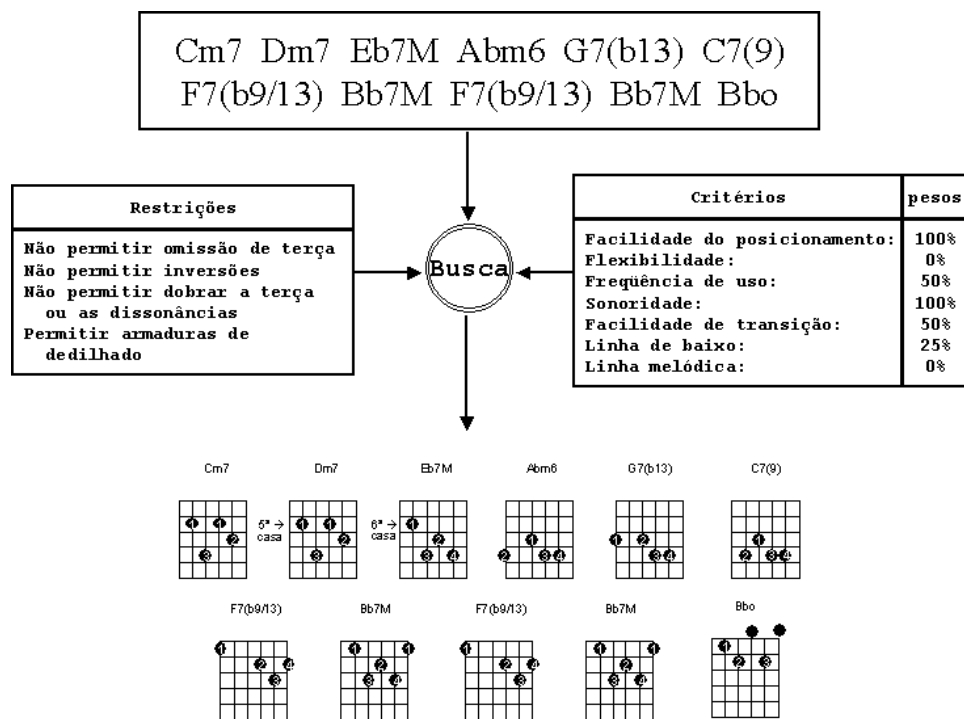


Figura 8.41 - Posições e digitações escolhidas para uma determinada cifra

8.6.2. Gatilhos

Um dos grandes problemas dos algoritmos de busca que utilizam funções de avaliação multi-critérios é que, freqüentemente, o acúmulo dos valores de um conjunto de atributos de menor importância termina sobrepondo um de maior importância (Ramalho, Rolland, & Ganascia 1999). Para resolver este problema, a solução encontrada foi utilizar gatilhos (*triggers*). Os gatilhos servem como regras em nível macro, supervisionando a execução da busca. Eles especificam situações onde um determinado critério torna-se mais importante do que os demais e, sozinho, justifica uma escolha.

No *D'Accord Guitar*, os gatilhos são representados por pares de posições de acordes (Figura 8.42). Caso este mesmo par apareça na cifra, quando a primeira posição for encontrada no algoritmo de busca, a próxima não será mais procurada (ficará restrita à posição especificada). No caso da Figura 8.42, quando a cifra contiver um *Am7* e um *D7(9)* imediatamente depois, o algoritmo de busca não gerará nós *D7(9)* com outras posições que sejam filhos desta posição de *Am7*.

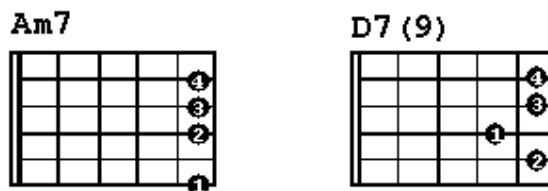


Figura 8.42 – Exemplos de gatilhos

Embora ainda não tenha sido implementado, a idéia é que a base de gatilhos utilizada seja editável, para que tais associações possam ser configuradas pelo usuário. Isto consta como mais um trabalho futuro.

8.6.3. Perfis de usuários

O processo de escolha definido é bastante personalizável, já que o usuário pode ajustar com razoável precisão os diversos critérios e subcritérios utilizados. Porém, tal precisão nem sempre é desejada, pois a maioria dos usuários não tem consciência da influência de cada critério no resultado geral. Por exemplo, o usuário pode querer apenas definir-se como iniciante, e deixar para o sistema a definição de que pesos atribuir a cada critério para que os acordes sejam apropriados para usuários iniciantes. Para possibilitar tal tipo de interação, foi empregado o conceito de perfis.

Os perfis são usados para que o usuário possa atualizar, de maneira simples, tanto os valores dos pesos dos diversos critérios quanto as diversas restrições sobre os acordes a serem gerados. De acordo com o perfil escolhido, todos os parâmetros são automaticamente configurados. Por exemplo, quando se escolhe o perfil *Iniciante*, os valores dos critérios de facilidade e frequência de uso são acrescidos, em detrimento aos de similaridade e flexibilidade. O usuário, obviamente, também pode reajustar os valores, se desejar. Desta forma, é possível um ajuste fino nos parâmetros ao mesmo tempo em que se obtém uma interface simples. Na Figura 8.43 são mostrados os pesos dos critérios relativos aos perfis *iniciante* e *solista*.

Perfil: Iniciante		Perfil: Solista	
CrITÉRIOS	pesos	CrITÉRIOS	pesos
Facilidade do posicionamento:	100%	Facilidade do posicionamento:	0%
Flexibilidade:	0%	Flexibilidade:	100%
Frequência de uso:	50%	Frequência de uso:	0%
Sonoridade:	0%	Sonoridade:	50%
Facilidade de transição:	100%	Facilidade de transição:	75%
Linha de baixo:	0%	Linha de baixo:	50%
Linha melódica:	0%	Linha melódica:	50%

Figura 8.43 - Exemplos de perfis e os pesos associados a cada critérios

8.6.4. Modelo Final

O modelo final pode ser verificado na Figura 8.44. A diferença para o modelo anterior é que a *busca multi-atributos* sofre restrições vindas da *base de gatilhos*, que é idealmente configurável pelo usuário. Além disso, as preferências podem ser ajustadas diretamente pelo usuário ou de acordo com o perfil que ele desejar.

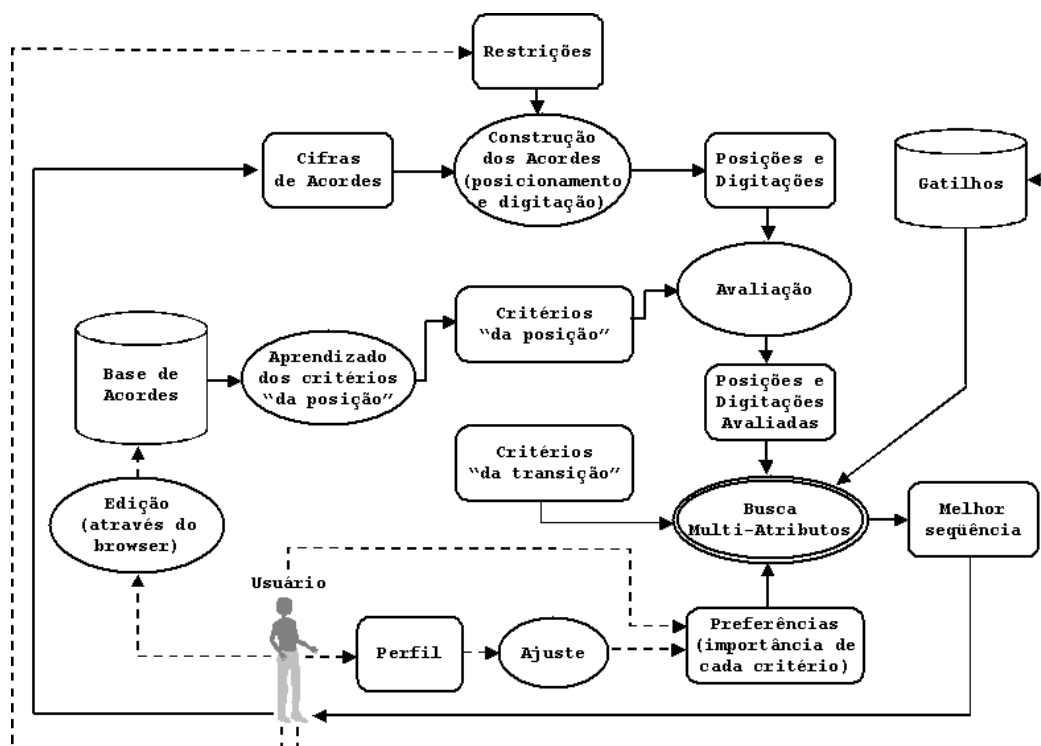


Figura 8.44 – Modelo implementado

8.7. Estudos de caso

A seguir são apresentados alguns estudos de caso simples efetuados quanto à interpretação harmônica do *D'Accord Guitar*. No primeiro estudo de caso, é dada a sequência

de cifras de acordes: Em / C7M / D7 / G. São procuradas as melhores posições e digitações levando em consideração apenas um critério e usando restrições que permitam inversões. Os resultados encontrados quando o atributo escolhido é a frequência de uso (Figura 8.45) são diferentes dos encontrados quando são escolhidos acordes com uma linha de baixo suave, mesmo que se usem posições pouco comuns (Figura 8.46).

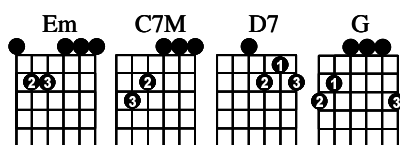


Figura 8.45 –
Posições e
digitações mais
comuns para a
seqüência de
acordes dada

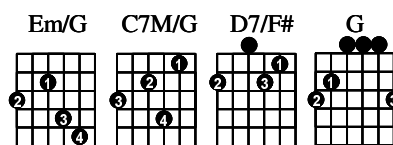


Figura 8.46 –
Posições de
acordes seguindo a
melhor linha de
baixo para a
seqüência de
acordes dada.

A partir do segundo estudo de caso, são utilizados mais de um critério. Na Figura 8.47, é apresentada as posições e digitações escolhidas para a seqüência Am / D7 / G / Em, levando em consideração todos os critérios com 100% de peso, seguindo as restrições *default*. Por coincidência, é escolhida a seqüência de acordes mais comumente usada. Este é um fato recorrente nos diversos testes realizados, e acreditamos dever-se ao fato de que as posições de acordes tornaram-se comuns justamente por trazerem consigo várias qualidades, como facilidade, flexibilidade e boa escolha de vozes.

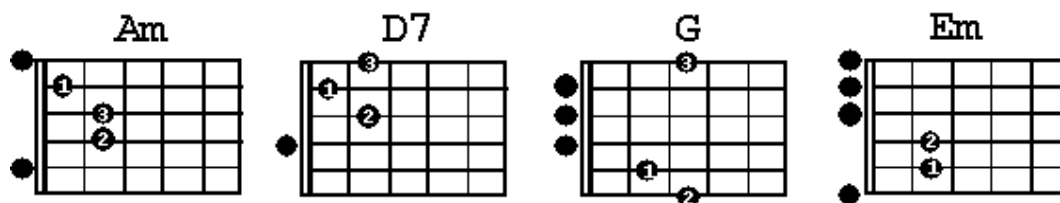


Figura 8.47 - Posições e digitações encontradas levando em consideração
todos os critérios

No terceiro estudo de caso, é utilizada a mesma seqüência de acordes e são levados em consideração os critérios de *facilidade* (com peso de 100%), *facilidade de transição* (com peso de 100%) e *flexibilidade* (com peso de 100%). O resultado é mostrado na Figura 8.48. As posições e digitações encontradas pecam na linha de baixo e (à exceção do D7) não são as mais comuns. No entanto, além de fáceis são extremamente flexíveis.

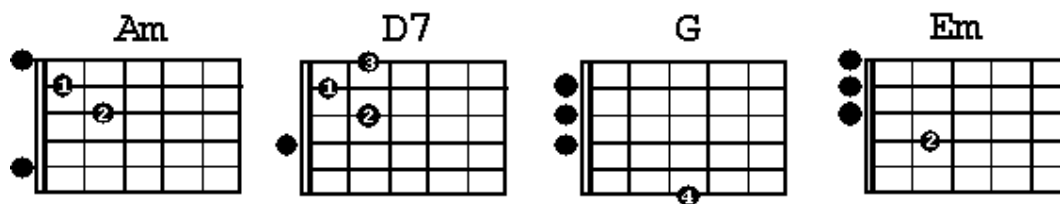


Figura 8.48 - Posições e digitações encontradas levando em consideração a facilidade de posicionamento, de transição e a flexibilidade

Por fim, foram testados os perfis, com a mesma sequência de cifras, mas utilizando as preferências definidas nos perfis. Aqui, são apresentados os perfis *iniciante* e *solista*. O perfil iniciante resultou nas mesmas posições e digitações do segundo estudo de caso (Figura 8.47). O perfil solista resultou na seguinte escolha:

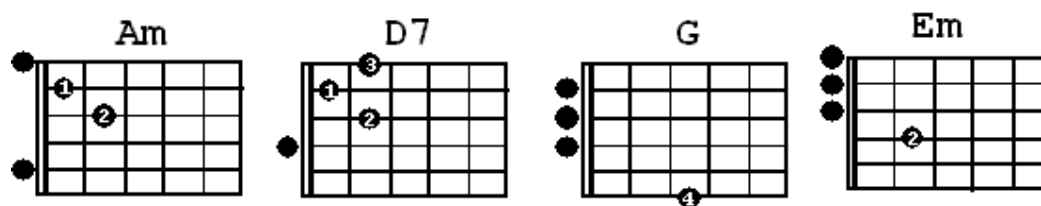


Figura 8.49 - Posições e digitações encontradas de acordo com o perfil solista

É importante ressaltar que o usuário pode escolher utilizar somente acordes da base, e pode alterar as restrições sobre as posições e digitações a serem geradas. Ao escolher os acordes da base, os resultados são mais previsíveis, com posições e digitações tradicionais. Ao escolher relaxar algumas restrições (como a possibilidade de inversões, por exemplo), surgem exemplos bastante interessantes, como o da Figura 8.50.

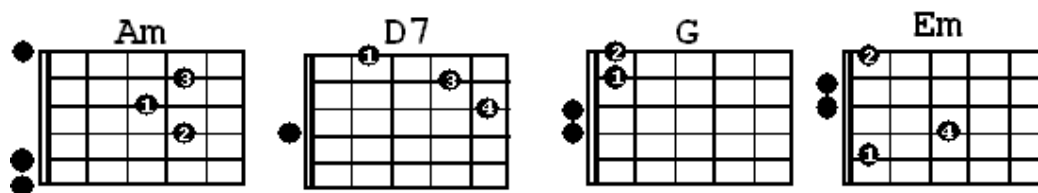


Figura 8.50 - Posições e digitações levando em consideração flexibilidade e linha melódica, relaxando restrições de inversões e de número de repetições permitidas

Os resultados não são conclusivos ainda, pois ainda há muito que refinar e ainda é necessária uma validação sistemática por especialistas. Entretanto, os primeiros resultados são

encorajadores, indicando que esta abordagem pode nos levar a uma solução completa e flexível para o problema apresentado.

Acreditamos que o presente estudo acrescentou algum conhecimento inclusive à área de teoria musical, por explicitar processos de decisão subjetivos do músico.

9. Conclusões

Este trabalho apresentou o *D'Accord Guitar*, um sistema para o aprendizado de músicas no violão/guitarra, incorporando funcionalidades de diversos tipos de software, como *sistemas de simulação instrumental*, *sistemas de acompanhamento automático*, *dicionários interativos de acordes* e *sistemas de karaokê*. Os problemas enfrentados foram apresentados e as suas respectivas soluções foram descritas. Entre eles, destaca-se a implementação de um sistema inteligente que modela o raciocínio de um violonista em atividades como interpretação de cifras, transposição e transcrição de solos.

O *D'Accord Guitar* exigiu um grande esforço, tanto de concepção quanto de implementação. Tal realização só foi possível graças à colaboração de outras pessoas, alguns alunos de iniciação científica (Hugo Pimentel de Santana, Izabel Zanforlin Santana e Rodrigo Queiroz da Costa Lima), e outros colaboradores voluntários e contratados (Thiago Souto Maior Cordeiro de Farias, José Leovigildo de Melo Coelho Filho, Jarbas Jacome de Oliveira Júnior, Andréa Fabrícia Nogueira da Silva, Augusto Cesar de Mello Cabral Pedroza) que colaboraram contínua ou pontualmente ao longo de três anos. Este grupo foi coordenado por mim, com a colaboração administrativa de Armando Pereira da Pontes Júnior e orientado por Geber Ramalho.

9.1. Resultados

Como o esforço de implementação do *D'Accord Guitar* foi enorme, não restou tempo para uma validação sistemática do software, principalmente devido ao fato de que é muito difícil uma análise quantitativa dos seus resultados. Como saber, por exemplo, se o processo de aprendizagem foi aprimorado, e o quanto foi aprimorado? É necessário fazer uma validação com um grande conjunto de pessoas que utilizem o software, questionando se foi criado um método mais eficiente do que os anteriores para leitura, escrita e aprendizagem musical. Para isso, o *software*, nos seus 3 modos de interação, tem sido disponibilizado na *Internet* para que usuários de todo o Brasil (não somente músicos) possam testá-los.

Existe, atualmente, uma comunidade de mais de 30.000 pessoas utilizando o *D'Accord Guitar*, incluindo pessoas de outros pontos do mundo, como Japão, Portugal, Suíça e Estados Unidos. Foi realizada uma pesquisa com esta comunidade, perguntando como o usuário classificaria o *D'Accord Guitar Player* (*fraco*, *razoável*, *bom* ou *muito bom*). A

resposta dos usuários é mostrada no Gráfico 9.1. Somando-se as porcentagens das classificações *bom* e *muito bom*, foi obtido um nível de aceitação de cerca de 95%.

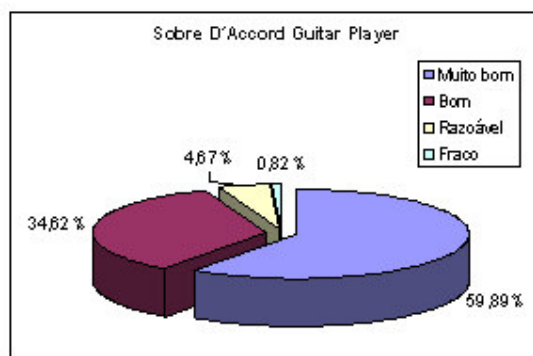


Gráfico 9.1 – Avaliação dos usuários quanto ao D'Accord Guitar Player

Além disso, a enquete abriu espaço para que os usuários citassem os problemas encontrados e fizessem comentários. Os problemas citados foram técnicos, como incompatibilidade com *drivers* ou problemas de instalação. Os comentários em sua grande maioria foram positivos, não havendo comentários conceituais negativos. De acordo com os comentários, a maior parte do público pesquisado considerou o software um auxiliar bastante útil ao processo de aprendizado de músicas no violão.

Quanto ao *D'Accord Guitar Editor* e ao *D'Accord Guitar Browser*, a validação tem sido feita de forma localizada, por enquanto, contando apenas com os músicos que auxiliaram o projeto. Embora o nível de aceitação também tenha sido alto, a validação não é expressiva. Estes softwares também estão sendo disponibilizados e pretendemos realizar uma primeira validação com eles.

9.2. Contribuições

No que concerne os objetivos iniciais do projeto *D'Accord*, enxergamos duas contribuições principais. A primeira refere-se à concepção e ao desenvolvimento de uma nova categoria de software musical, atendendo a um apelo do público por soluções simples de aprendizado de músicas para violão e guitarra. A segunda refere-se à explicitação de um conhecimento implícito e subjetivo, sobre a interpretação de cifras, automatizando este processo com resultados satisfatórios. Este é um problema de grande relevância, difícil, e que ainda não havia sido tratado. Os resultados obtidos são contributivos não só na área computacional, quanto permitem *insights* na própria área de teoria musical.

Além disso, o *D'Accord Guitar* deu origem a uma linha de pesquisa relacionada à computação musical no Centro de Informática da UFPE, envolvendo 2 mestrandos e 2 alunos de iniciação científica, servindo de plataforma e de suporte para uma série de estudos complementares. Entre eles, o aprendizado de estilos violonísticos, a aquisição e reconhecimento de execuções reais, o tratamento dinâmico do ritmo, a extração de padrões, a personalização, a extensão da interpretação harmônica, e a adaptação a outros instrumentos.

Há, também, uma perspectiva comercial com a criação da empresa *D'Accord Music Software*. É um caso de união entre aplicação comercial e projeto científico de pesquisa, contando com alunos, professores e mestrandos de informática, administração, música e programação visual.

Por estas razões, acreditamos ter dado, com este trabalho, algumas contribuições tanto à comunidade científica quanto à comunidade em geral.

9.3. *Perspectivas*

O grande interesse demonstrado e o grande nível de aceitação apontam para a necessidade de maiores estudos na área e para a continuidade do trabalho desenvolvido. Nós pretendemos prosseguir o projeto *D'Accord Guitar* de diversas formas.

Em primeiro lugar, através da criação de outros aplicativos relacionados. Por exemplo, a criação de variações do *D'Accord* para outros instrumentos (em especial outros instrumentos de corda harmônicos, por acreditarmos ser possível desenvolvê-los de forma semelhante). Além disso, pretendemos produzir uma maior sinergia entre a comunidade de usuários do *D'Accord Guitar*. Para isto estamos implementando uma comunidade virtual, que pode servir para troca de músicas, ritmos e acordes, de forma semelhante ao que ocorreu com o MP3.

Em segundo lugar, através da extensão do software. Pretendemos desenvolver aplicativos semelhantes para outros instrumentos, ampliar a interpretação de cifras para que leve em consideração tanto o ritmo quanto a harmonia, e implementar as funções que já foram concebidas, mas ainda não foram implementadas, como a aquisição do acompanhamento musical diretamente via guitarra MIDI.

Em terceiro lugar, através de uma validação sistemática acerca das soluções propostas, tanto do software como um todo quanto das funções de interpretação harmônicas, em particular.

Em quarto lugar, através do refinamento das soluções atuais. As funções de interpretação harmônica são um caso em que é fortemente necessário um refinamento. É preciso fazer uma análise comparativa das melhores técnicas de aprendizado de máquina a utilizar, tratar critérios ainda não modelados (como as restrições rítmicas) e tratar da dinamicidade do processo de escolha, já que a importância dada aos diferentes critérios pode variar no decorrer da música.

Para tal prosseguimento deste trabalho, o grupo *D'Accord* conta com mais um aluno de mestrado, possivelmente um aluno de doutorado e dois alunos de iniciação científica.

10. Bibliografia

- Aho, A. V., Sethi, R. & Ulmann, J. D. (1986). *Copmilers: Principles, Techniques and Tools*. Cambridge, MA: Addison-Wesley.
- Anchor Press, Garden City. Desain, P. and Honing, H. (1992). *Music, Mind, and Machine: Studies in Computer Music, Music Cognition, and Artificial Intelligence*. Thesis Publishers, Amsterdam.
- Ausubel, D. P. (2000). *The Acquisition and Retention of Knowledge: A Cognitive View*. Boston, MA: Kluwer.
- Beck, S. D. (1996). The Virtual Instrument Paradigm: An Alternative Approach to Interactive Computer Music. In *Computers & Mathematics with Applications*, 32(1), pp. 5-10.
- Bennett, R. (1986). *Forma e Estrutura na Música*. Rio de Janeiro: Jorge Zahar.
- Bratko, I. (1994). Machine Learning and Qualitative Reasoning. In *Machine Learning*. 14(3). Pp. 305-312.
- Cabral, G., Zanforlin, I., Santana, H., Lima, R. & Ramalho, G. (2001a). “Da Cifra Para o Braço: Estudo dos Problemas de Execução Musical em Violão e Guitarra”. In *proceedings of the VIII SBC&M*.
- Cabral, G., Zanforlin, I., Santana, H., Lima, R. & Ramalho, G. (2001b) “D’Accord Guitar: an Innovative Guitar Performance System”. *Les Actes des Huitièmes Journées d’Informatique Musicale*. 127-137.
- Cabral, G., Zanforlin, I., Santana, H., Lima, R. & Ramalho, G.(2001c) “Playing Along with D’Accord Guitar”. In *Proceedings of the VIII SBC&M*, Fortaleza.
- Cambouropoulos, E. (2000). From MIDI to Traditional Musical Notation. In *Proceedings of the AAAI'2000 Workshop on Articial Intelligence and Music*, 17th National Conference on Articial Intelligence (AAAI'2000), Austin, TX Menlo Park, CA: AAAI.
- Cambouropoulos, E. (2001). Automatic Pitch Spelling: From Numbers to Sharps and Flats. *Proceedings of the VIII Brazilian Symposium on Computer Music*, Fortaleza.
- Chediak, A. (1984). *Dicionário de acordes cifrados: harmonia aplicada à música popular*. São Paulo: Irmãos Vitale.
- Chediak, A. (1999). *Songbook of Chico Buarque (vol 1-4)*. Rio de Janeiro: Lumiar.

- Chung, S., Chung, S. (1998). A Multimedia System for Interactive Learning of Organ Literature. In *Proceedings on the 1998 ACM Symposium on Applied Computing*. New York, NY: ACM.
- Clarke, E. F. (1993). Imitating and evaluating real and transformed musical performances. *Music Perception*, 10(3), 317-341.
- Cordier, M.-O. (1995) Doigtage intelligent d'une partition de guitare. N° 23, 46-48. Bulletin de L'AFIA.
- Dannenberg, R. B. & D. Rubine. (1995). Toward Modular, Portable, Real-Time Software. In the *Proceedings of the 1995 International Computer Music Conference*, ICMA. Pp. 65-72.
- Dannenberg, R. B. (1993). Software Design for Interactive Multimedia Performance. *Interface-Journal of New Music Research*, 22(3):213-218.
- Dasarathy, B. V. (1991). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press.
- Date, C. J. (1991). *Introdução a sistemas de bancos de dados*. Rio de Janeiro: Campus.
- Davidson, J. W. (1993). Visual perception of performance manner in the movements of solo musicians. *Psychology of Music*. Vol 21. Pp. 103-113.
- Desain, P., & Honing, H. (1992). The quantization problem: traditional and connectionist approaches. In Balaban, M., Ebcioglu, K., & Laske, O. (eds.), *Understanding Music with AI: Perspectives on Music Cognition*. Cambridge, MA: MIT. Pp 448-463.
- Dowling, W. J. (1989). Simplicity and Complexity in Music and Cognition. *Contemporary Music Review*, (s.l.), Vol 4. Pp. 247-53.
- Farrett, P. W. (1996). Intelligent musical instrument systems. *Computer and Mathematics with Applications*, 32(1), pp. 37-48.
- Fowler, W. (1984a) *Chord Voicing Systems*. Fowler Music Enterprises.
- Fowler, W. (1984b) *Chord Progression Systems*. Fowler Music Enterprises.
- Friedman, J. H. (1994). Flexible Metric Nearest Neighbor Classification. Technical Report, Stanford University, Department of Statistics.
- Hamakawa, R. & Rekimoto. (1994). Object Composition and Playback Models for Handling Multimedia Data. In *Multimedia Systems*. 2(1). Pp.26-3.

- Haykin, S. (1999). *Neural Networks – A Comprehensive Foundation*. Englewood Cliffs: Prentice Hall.
- Hill, B. (1994). *MIDI for Musicians: Buying, Installing, and Using Today's Electronic Music-Making Equipment*. Atlanta, GA. A Cappella.
- Holdsworth, A. (1998) *Melody Chords for Guitar*. New York: Hal Leonard.
- Honing, H. (2001). From time to time: The representation of timing and tempo. *Computer Music Journal*, 25(3) 50-61.
- Keating, C. & Anderton, C. (1998) *Digital Home Recording - Tips, Techniques, and Tools for Home Studio Production*. Backbeat Books.
- Klapuri, A. (1998). Automatic Transcription of Music. Dissertação de Mestrado, Tampere University of Technology, Department of Information Technology. Tampere, Finland.
- Kolodner, J. L. (1993). *Case-Based Learning*. Boston: Kluwer Academic Publishers.
- Lavrač, N. (1990). *Principles of Knowledge Acquisition in Expert Systems*. PHd Thesis, Faculty of Technical Sciences, University of Maribor.
- Létourneau, S., Matwin, S. & Famili, F. (1998). A Normalization Method for Contextual Data: Experience from a Large-Scale Application. In *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, Chemnitz. Pp. 49-54
- Manber, U. (1994). *Introduction to Algorithms - A Creative Approach*. Cambridge, MA: Addison-Wesley.
- Messick, P. (1997). *Maximum MIDI: Music Applications in C++*. Greenwich, CO: Manning.
- Michalski, R. S. & Tecuci, G. (1994). *Machine Learning: A Multistrategy Approach (Vol 4)*. Morgan Kaufmann, São Francisco, CA.
- Michels, U. (1988). *Guide Illustré de la Musique*. Rennes: Fayard.
- Miner, N. & Caudel, T. (1998). *Computational Requirements and Synchronization Issues for Virtual Acoustic Displays*. (s.l.): Presence, 7(4). pp.396 -409.
- Miranda, E. (2000). Readings in Music and Artificial Intelligence. *On the Potential of Machine Learning for Music Research*. London: Harwood Academic Publishers.
- Mitchell, T. (1997a). *Machine learning*. New York, NW: McGraw-Hill. Pp. 52-81.
- Mitchell, T. (1997b). *Machine learning*. New York, NW: McGraw-Hill. Pp. 154-200.

- MMA – MIDI Manufacturers Association. (1996). *The Complete MIDI 1.0 Detailed Specification*. Los Angeles, CA.
- Moog, B. (1986) MIDI: Musical Instrument Digital Interface. *Journal of the Audio Engineering Society*, 34(5), 394-404.
- Nichol, J. (1988). Models, micro-worlds and minds. In J. Nichol, J. Briggs, & J. Dean (Eds.), *Prolog, children and students*. London: Kogan Page
- Nugroho, L. E. & Sajeev, A. S. M. (1995). Persistence in Music Data Structures. In *Proceedings of the 1995 ACM Symposium on Applied Computing*. pp 27-31.
- Pachet, F. & Roy, P. (1995). Integrating constraint satisfaction techniques with complex object structures. *15th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, ES'95. Cambridge. Pp. 11-22.
- Pachet, F., Ramalho, G., & Carrive, J. (1996). Representing Temporal Musical Objects and Reasoning in the MusES System. *Journal of New Music Research*. 25(3). Pp. 253-73.
- Pressman, R. S. (1997). *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill.
- Ramalho, G. (1997). *Construction d'un agent rationnel jouant du jazz*. Thèse de doctorat, Université Paris VI.
- Ramalho, G., & Ganascia, J.-G. (1994). Simulating Creativity in Jazz Performance. In Twelfth National Conference on Artificial Intelligence - AAAI'94. Seattle: AAAI. Pp. 108-13.
- Ramalho, L., Rolland, P-Y., Ganascia, J-G. (1999). An Artificially Intelligent Jazz Performer. *Journal of New Music Research*. 28(2), 105-129. Cambridge, MA: Swets & Zeitlinger.
- Ramalho, G. (2001). *Systèmes d'accompagnement musical en temps réel*. Série Spéciale "Informatique Musicale". Paris: Hermès. Forthcoming.
- Rayward-Smith, V., Osman, L & Reeves, R. (1996) *Modern Heuristic Search Methods*. John Wiley & Son.
- Read, R. C. (1997). Combinatorial Problems in the Theory of Music. DMATH: Discrete Mathematics. Vol 167.
- Roads, C. (1996). *The Computer Music Tutorial*. Cambridge, MA: MIT.

- Robbins, S. (1995). A MIDI Primer For Computer Scientists. UTSA Technical Report CS-95-11.
- Rothstein, J. (1992). MIDI: A Comprehensive Introduction. Madison, Wis.: A-R Editions.
- Rowe, P. (1993). *Interactive Music Systems: Machine Listening and Composing*. Cambridge, MA: MIT.
- Rowe, R. (1993). *Interactive music systems: Machine Listening and Composing*. Cambridge, MA: MIT.
- Russel, S & Norvig, P. (1994) *Artificial Intelligence: A Modern Approach*. Englewood Cliffs: Prentice Hall.
- Sayegh, S. (1989) Fingering for String Instruments with the Optimum Path Paradigm. *Computer Music Journal*. 13(3).
- Shelemay, K. K. (1990). *Musical Transcription. The Garland Library of Readings in Ethnomusicology*. Vol 4. City of Garland: Garland.
- Sher, C. (1991) *The New Real Book* (vol. 1 and 2). Berkeley: Sher Music.
- Sonnenschein, D. (2000). Developing Online-Based Instruction: Technology as A Tool For Attaining Music Cognition. In *Technology & Teamwork*, Cambridge, Ma: IEEE.
- Tsang, E. (1993). *Foundations of Constraint Satisfaction*. London: Academic Press.
- Ueno, K., Furukawa, K. & Bain, M. (2000). Motor Skill As Dynamic Constraint Satisfaction. *Lunköping Electronics Articles in Computer and Information Science*. 5(36). Linköping: Lunköping.
- Ueno, K., Higuchi, F., Furukawa, K. (1997). Towards the Understanding of the Technique for Playing String Instruments. *6th IFAC Symposium On Automated Systems Based On Human Skill*.
- Wang, J-F & Li, T-Y. (1997). Generating Guitar Scores from a MIDI Source. In *Proceedings of the 1997 International Symposium on Multimedia Information Processing*.
- West, R., Howell, P., & Cross, I. (1991). Musical Structure and Knowledge Representation. In P. Howell, R. West, & I. Cross (Eds.), *Representing Musical Structure* (pp. 1-30). London: Academic Press.

- West, R., Howell, P., & Cross, I. (1991). Musical Structure and Knowledge Representation. In P. Howell, R. West, & I. Cross (Eds.), *Representing Musical Structure*. London: Academic Press. Pp. 1-30.
- Wettschereck, D. (1994). A Hybrid Nearest-Neighbor and Nearest-Hyperrectangle algorithm. *Machine Learning: ECML-94. European Conference on Machine Learning. Proceedings*. Catania, Italy:Springer-Verlag. Pp. 323-35.
- Widmer, G. (1993). Combining Knowledge-Based and Instance-Based Learning to Exploit Qualitative Knowledge. *Informatica*. 17(4), Special Issue on Multistrategy Learning. Pp.371-385.
- Widmer, G. (1995). Modeling the Rational Basis of Musical Expression. *Computer Music Journal*. 19(2). Cambridge, MA: MIT. Pp.76-96.
- Widmer, G. (2001). Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report. *AI Communications*. 14(3). Pp. 149-162.
- Woody, R. H. (2000). Technology-Assisted Research in Music Cognition: Enhancing Instruction in Expressive Music Performance. In *Proceedings of the Seventh International Technological Directions in Music Learning Conference*. San Antonio, TX: Institute for Music Research Press.
- Youngblood, P. (1989). MIDI Guitar and Synthesis: The Basics of Guitar Synthesis. Milwaukee, WI: Hall Leonard Pub. Co.

Anexo 1 - Representação Utilizada

A representação gráfica utilizada durante a dissertação segue o padrão da maioria das publicações de músicas cifradas. O braço do violão é representado por 6 linhas horizontais e 6 linhas verticais. O braço do violão pode ser representado na horizontal ou na vertical, de forma a ajustar-se à diagramação do restante do texto.

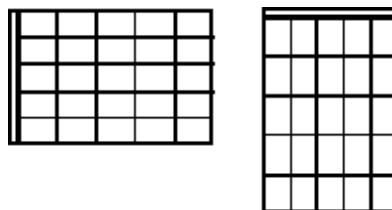


Figura A.51 - Braço do violão na horizontal e na vertical.

O retângulo na extremidade do braço é chamado de pestana, e indica em que lado encontra-se a cabeça do violão (Figura A.52). Esta pestana não deve ser confundida com a utilização de um dedo para pressionar mais de uma corda, que também é chamado de pestana. As linhas delimitam as casas e cordas do violão. Os círculos indicam as coordenadas a serem tocadas. Círculos do lado de fora do braço, ao lado da pestana do violão, representam notas a serem tocadas com a corda solta. Números indicam os dedos a serem utilizados. Cordas sem nenhum círculo são cordas que não devem ser tocadas.

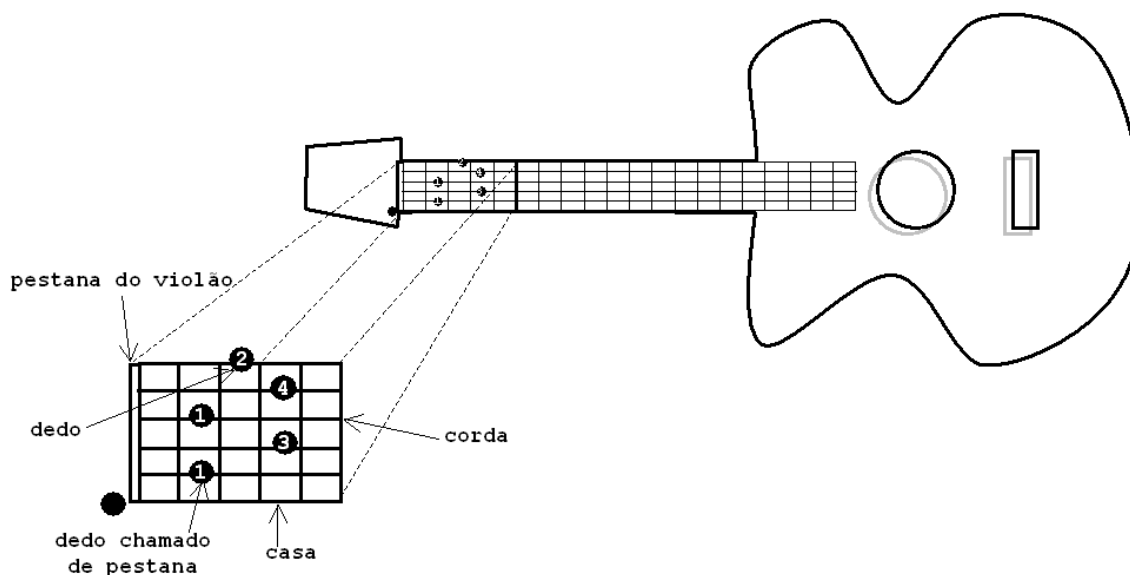


Figura A.52 - Diagrama e sua relação com o violão

