

---

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DISSERTAÇÃO DE MESTRADO

**ARGO – Um Ambiente Adaptativo de  
Consultas Visuais e *Browsing* a Bancos de  
Dados Relacionais**

---

Por

Ana Isabel Pinheiro da Silva

Campina Grande, agosto de 1998

---

UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ANA ISABEL PINHEIRO DA SILVA

**ARGO – Um Ambiente Adaptativo de  
Consultas Visuais e *Browsing* a Bancos de  
Dados Relacionais**

*Dissertação apresentada ao Curso de Mestrado em  
Informática da Universidade Federal da Paraíba, em  
cumprimento às exigências para obtenção do Grau de  
Mestre.*

**Orientador:** *Prof. Marcus da Costa Sampaio, Dr.*

Campina Grande, agosto de 1998



S586a Silva, Ana Isabel Pinheiro da.  
ARGO : um ambiente adaptativo de consultas visuais e  
browsing a bancos de dados realcionais / Ana Isabel  
Pinheiro da Silva. - Campina Grande, 1998.  
98 f.

Dissertação (Mestrado em Informática) - Universidade  
Federal da Paraíba, Centro de Ciências e Tecnologia, 1998.  
"Orientação : Prof. Dr. Marcus da Costa Sampaio".  
Referências.

1. Banco de Dados. 2. Ambiente Adaptativo de Consultas  
Visuais e Browsing (ARGO). 3. ARGO. 4. Dissertação -  
Informática. I. Sampaio, Marcus da Costa. II. Universidade  
Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.65(043)

**ARGO - UM AMBIENTE ADAPTATIVO DE CONSULTAS  
VISUAIS E BROWSING A BANCOS DE DADOS RELACIONAIS**

**ANA ISABEL PINHEIRO DA SILVA**

**DISSERTAÇÃO APROVADA EM 12.08.1998**



**PROF. MARCUS COSTA SAMPAIO, Dr.**  
**Orientador**



**PROF. BERNARDO LULA JÚNIOR, Dr.**  
**Examinador**



**PROF<sup>a</sup> MARIA DE FÁTIMA Q. V. TURNELL, Ph.D**  
**Examinadora**

**CAMPINA GRANDE - PB**

# Agradecimentos

Agradeço a todas as pessoas que passaram por minha vida e que contribuíram, de alguma forma, para o meu crescimento pessoal e/ou profissional. Algumas delas me influenciaram direta e profundamente, outras indiretamente, mas com tamanha intensidade que jamais serão esquecidas.

Acima de tudo, e de todos, agradeço a Deus, pela Vida.

Aos meus pais, Valentim e Socorro, pela perseverança e esforço dedicados à educação dos seus nove filhos. Aos meus irmãos, pela paciência e compreensão.

Ao meu orientador, professor Marcus Sampaio, pelas críticas, sugestões e correções sempre meticulosas que foram de grande importância durante o desenvolvimento deste trabalho.

Aos funcionários da COPIN, LabCom e Miniblibio, pelo atendimento e atenção prestados, em especial a Aninha, Vera, Lilian e Zeneide. Ao pessoal da cantina, Inês, Jô e Romildo, pelo "cafezinho".

Aos amigos, pelos agradáveis momentos de descontração, pelas farras, "shopinhos" no velho e bom BIG MIX (quantas recordações!!!), amanhecer na feira da Prata, viagens, passeios de barco... E, principalmente, pela troca de experiências que me proporcionou

um crescimento moral, intelectual e humano de extrema valia. Em especial, gostaria de citar: Luiza, João, Maysa, Márcia, Avellar, Verônica, Tatiana, Norma, Michelle, Lívia, Márcio, Milena, Guga, Ismênia, Nícia, Adolfo, Sandro, Rute, Cassandra, Mônica, Alvinho, Álvaro, Stênio, Adriano, Petúnia, Ivette, Gustavo, Érika, Josenildo, Trícia, Ianna.

Enfim, agradeço a todos que amei, que amo e que amarei, como já disse Carlos Drummond "Amar se aprende amando, sem omitir o real cotidiano, também matéria de poesia".

# Resumo

O objetivo desta dissertação é desenvolver um ambiente de consultas visuais e *browsing* a bancos de dados relacionais. Este ambiente, denominado ARGO, se enquadra na abordagem de interfaces de consultas visuais, seja pelo uso de representações visuais (icônica, tabular), seja por permitir a manipulação direta do conteúdo, freqüentemente gerando as consultas de forma dinâmica. O objetivo fundamental deste ambiente é o de oferecer a um usuário a interface que lhe seja mais adequada, de forma que, tanto um usuário iniciante quanto um mais experiente se sintam à vontade para interagir com o sistema, cada um com suas exigências particulares. O ARGO é composto por quatro interfaces denominadas: **Básica**, **Mediana**, **Avançada** e **SQL**, sendo cada uma delas apropriada a um tipo específico de usuário (**Iniciante**, **Mediano**, **Experiente** e **Especialista**, respectivamente). O perfil do usuário pode ser modificado ao longo do tempo, para refletir as mudanças nas suas formas de interações com o BD; conseqüentemente, a interface que lhe é adequada em um momento poderá deixar de sê-lo em outro momento. O ARGO “se adapta” a estas mudanças de perfil, fornecendo a cada momento a interface adequada.

## *Abstract*

The objective of this work is to develop a visual querying and browsing environment for a relational database. Such environment, called ARGO, follows an interface approach for visual queries, using visual representation (icons, table) and allowing direct manipulation, producing frequently queries in dinamic way. The fundamental concern in the environment is to offer an interface that suits the user's needs, allowing that a novice user as well as an expert one feel comfortable to use the system. ARGO is composed by four interfaces: **Basic**, **Medium**, **Experient** and **Specialist**. An user category can be updated as the user gets more knowledge about the system, since an interface that is more adequate to an user at one especific moment may not be adequate at another moment. ARGO adapts itself to those changes in the user behaviour or expertise, providing the more appropriate interface to the user.

# Sumário

<b>Capítulo I - Introdução</b>	<b>1</b>
1.1 Objetivos da dissertação.....	4
1.2 Estrutura da dissertação.....	5
<b>Capítulo II – Trabalhos Relacionados</b>	<b>7</b>
2.1 HIBROWSE.....	9
2.1.1 Características da <i>interface</i> .....	9
2.1.2 Consulta e <i>browsing</i> ao BD.....	10
2.1.3 Vantagens e desvantagens do HIBROWSE.....	11
2.2 MEDUSA – Multimode Database User interface with computer Supported learning Aid.....	12
2.2.1 Visão geral da MEDUSA.....	12
2.2.2 A <i>interface</i> .....	13
2.2.3 Auxílio ao aprendizado.....	15
2.2.4 Vantagens e desvantagens da MEDUSA.....	15
2.3 Conclusão.....	15
<b>Capítulo III – Arquitetura do ARGO</b>	<b>17</b>
3.1 A arquitetura do ARGO.....	17
3.1.1 Módulo Gerenciador das <i>interfaces</i> .....	18
3.1.2 Módulo Gerenciador dos Modelo de Usuário.....	19
3.1.3 Módulo Gerenciador dos Metabancos de Dados.....	19

3.1.4 Módulo Mapeador dos SGBDs.....	20
--------------------------------------	----

## **Capítulo IV – Especificação das *interfaces* do ARGO** **22**

4.1 Padrão de apresentação das <i>Interfaces</i> .....	22
4.1.1 Barra de Menu.....	23
4.1.2 Barra de Ícones.....	26
4.1.3 Seleção do BD.....	27
4.1.4 Janela de Esquema.....	28
4.1.5 Janela de Conteúdo.....	28
4.1.6 Barra de Botões.....	28
4.1.7 Barra de Condições.....	31
4.2 <i>Interface</i> Básica.....	33
4.2.1 <i>Scrolling</i> .....	35
4.4.2 Exemplos de utilização da <i>interface</i> .....	36
4.2.3 Análise da <i>interface</i> Básica.....	38
4.3 <i>Interface</i> Mediana.....	40
4.3.1 Apresentação das Janelas e seus conteúdos.....	41
4.3.2 Projeção de campos.....	43
4.3.3 Exemplos de utilização da <i>interface</i> .....	43
4.3.4 Análise da <i>interface</i> Mediana.....	48
4.4 <i>Interface</i> Avançada.....	50
4.4.1 Janela de Referências.....	51
4.4.2 Projeção de campos.....	53
4.4.3 Uso do E e do OU.....	53
4.4.4 Uso de junções explícitas e implícitas.....	56
4.4.5 Exemplos de utilização da <i>interface</i> .....	57
4.4.6 Análise da <i>interface</i> Avançada.....	69
4.5 <i>Interface</i> SQL.....	71
4.5.1 Análise da <i>interface</i> SQL.....	72

## **Capítulo V – O Metabanco de Dados e sua *Interface*** **73**

5.1 <i>Interface</i> para manutenção dos Metabancos e cadastro de Usuários.....	74
5.1.1 Parametrização do Diretório dos metabancos.....	75
5.1.2 Parametrização dos Campos Relevantes.....	76
5.1.3 Parametrização dos Campos das Tabelas dos BDs.....	78
5.1.4 Cadastro de Usuários.....	79

## **Capítulo VI – Implementação das *interfaces* do ARGO** **80**

6.1 Ambiente de desenvolvimento do ARGO.....	80
6.2 Estrutura do módulo gerenciador das <i>interfaces</i> .....	81
6.2.1 Tabela de Usuários.....	81
6.2.2 Tabela de Sessões.....	82
6.2.3 Tabela de Históricos.....	82
6.3 Estrutura do módulo gerenciador dos metabancos de dados.....	84
6.3.1 Definição do metabanco para a aplicação a ser usada como exemplo.....	86
6.4 Definição da tabela de Diretórios para o metabanco.....	88
6.5 Algoritmos das <i>interfaces</i> .....	89
6.5.1 Algoritmos para a Barra de Ícones.....	89
6.5.2 Algoritmo para o botão Pesquisar.....	91
6.5.3 Algoritmo para o botão Ordenar.....	92
6.5.4 Algoritmo para a Barra de Condições.....	92
6.5.5 Algoritmo para seleção de Campo/Valor na Janela de Conteúdo.....	93
6.5.6 Algoritmo para <i>interface</i> Básica.....	93
6.5.7 Algoritmos para <i>interface</i> Mediana.....	94
6.5.8 Algoritmos para <i>interface</i> Avançada.....	96

## **Capítulo VII – Aspecto Adaptativo do ARGO** **99**

7.1 Elementos para construção do Modelo de Chang.....	101
7.1.1 Estereótipos de usuário.....	101
7.1.2 Assinaturas de usuário.....	103
7.2 Modelos de usuário do ARGO.....	103

7.2.1 Definição dos estereótipos de usuário.....	104
7.2.1.1 Critérios para determinação de um estereótipo.....	105
7.2.2 Definição das assinaturas de usuário.....	107
7.3 Construção de um modelo de usuário.....	109
7.3.1 Estrutura do módulo gerenciador dos modelos de usuário.....	113
7.3.1.1 Tabela de Interações.....	113
7.3.1.2 Tabela de Estereótipos.....	114
7.3.1.3 Tabela de Classificação das interações.....	114
7.3.2 Cálculo dos critérios que determinam um estereótipo.....	114
7.3.2.1 Critério <u>freqüência</u> de interação.....	114
7.3.2.2 Critério <u>repetitividade</u> da consulta.....	115
7.3.2.3 Critério <u>familiaridade</u> com o BD.....	116
7.3.2.4 Critério <u>complexidade</u> de formulação da consulta.....	118
7.3.3 Cálculo final do estereótipo.....	122
7.4 Exemplos de mudança de estereótipo.....	123
<b>Capítulo VIII – Conclusões e Perspectivas</b>	<b>125</b>
8.1 Considerações finais.....	125
8.2 Perspectivas de trabalhos futuros.....	127

# Lista de Figuras

Fig.I.1 - Tipos de usuário e suas respectivas <i>interfaces</i> .....	4
Fig.II.1 - Interface do HIBROWSE para hotéis.....	10
Fig.II.2 - Elaborando uma consulta no HIBROWSE para hotéis.....	11
Fig.II.3 - Uma visão geral da interface MEDUSA.....	13
Fig.II.4 - Um exemplo da MEDUSA com alguns diálogos abertos.....	14
Fig.III.1 - Arquitetura do ARGO.....	18
Fig-IV.1 - Definição de janelas e botões.....	23
Fig.IV.2 - Seleção da opção <i>Arquivo/Abrir</i> do Barra de Menu.....	24
Fig.IV.3 - Seleção da opção <i>Ir Para</i> da Barra de Menu.....	25
Fig.IV.4 - Seleção da opção <i>Interface</i> da Barra de Menu.....	26
Fig.IV.5 - Barra de ícones.....	26
Fig.IV.6 - Ícones que identificam as <i>interfaces</i> .....	27
Fig.IV.7 - Tela para entrar com padrão de pesquisa, ao ser selecionado o botão <i>Pesquisar</i> .....	29
Fig.IV.8 - Tela para informar o tipo da ordenação do campo selecionado.....	30
Fig-IV.9 - Interface Básica.....	35
Fig.IV.10 - Pesquisa os Cds que possuem no seu título a letra “B”.....	37
Fig.IV.11 - Resultado da consulta baseada numa busca por padrão.....	37
Fig.IV.12 - Resultado da consulta baseada na seleção de valores exatos.....	38
Fig.IV.13 - Seleção da tabela CD na <i>interface</i> Mediana.....	41
Fig.IV.14 - Seleção da tabela Empréstimo na <i>interface</i> Mediana.....	42
Fig.IV.15 - Seleção do campo Data_Empréstimo para ordenação.....	44

Fig.IV.16 – Resultado: tabela Empréstimo ordenada por Data_Empréstimo.....	44
Fig.IV.17 - Resultado da consulta à tabela Empréstimo para obter os Cds emprestados e não devolvidos.....	45
Fig.IV.18 - Resultado da “navegação” para tabela PESSOA.....	46
Fig.IV.19 - Exemplo de uma projeção: retirar o campo Cod_Pessoa da Janela de Conteúdo.....	47
Fig.IV.20 - Resultado da exclusão do campo Cod_Pessoa da consulta.....	48
Fig.IV.21 - Aspectos iniciais da <i>interface</i> Avançada.....	51
Fig.IV.22 - Resultado da seleção da tabela CD na Janela de Esquema.....	52
Fig.IV.23 - Resultado da consulta usando E e OU.....	55
Fig.IV.24 - Resultado da seleção da tabela Empréstimo.....	57
Fig.IV.25 - Selecionando campo da tabela Cd para exibi-la na Janela de Conteúdo.....	58
Fig.IV.26 - Exibição da Barra de Condições ao “clique” em Data_Devolução p/ escolher o operador “=” .....	59
Fig.IV.27 - Resultado da consulta aos CD’s emprestados.....	60
Fig.IV.28 - Resultado da consulta com a retirada do campo Data_Devolução.....	61
Fig.IV.29 - Inclusão dos campos Cod_Gravadora e Cod_Genero à Janela de Conteúdo.....	62
Fig.IV.30 - Exibição das tabelas referências das tabelas EMPRESTIMO e CD.....	63
Fig.IV.31 - Junção entre as tabelas CD e GENERO, a partir de EMPRESTIMO.....	64
Fig.IV.32 - Resultado da consulta do exemplo 4.....	65
Fig.IV.33 - Consulta ordenada por Título e Cod_Pessoa.....	66
Fig.IV.34 - Resultado da consulta formulada para o exemplo 6.....	67
Fig.IV.35 - Resultado da seleção do BD Fornec.....	68
Fig.IV.36 - Exemplo de uma theta-junção.....	69
Fig.IV.37 - Interface SQL.....	71
Fig.V.1 - Tela inicial com as opções parametrização do Metabanco de dados.....	75
Fig.V.2 - Janela para cadastro dos diretórios.....	76
Fig.V.3 - Janela para definição dos campos relevantes do usuário.....	77

Fig.V.4 - Janela para criação do metabanco que independe do usuário.....	78
Fig.V.5 - Janela para cadastro dos usuários do ARGO.....	79
Fig.VI.1 - Estrutura componente do módulo gerenciador das <i>interfaces</i> .....	81
Fig.VI.2 - Meta tabela MT_CD_DBCds.....	87
Fig.VI.3 - Meta tabela MT_EMPRESTIMO_DBCds.....	87
Fig.VI.4 - Meta tabela MT_GENERO_DBCds.....	87
Fig.VI.5 - Meta tabela MT_GRAVADORA_DBCds.....	87
Fig.VI.6 - Meta tabela MT_MUSICA_DBCds.....	87
Fig.VI.7 - Meta tabela MT_PESSOA_DBCds.....	88
Fig.VI.8 - Meta-tabela com os campos relevantes por usuário.....	88
Fig. VII.1 - Mapeamento dos critérios para determinação do estereótipo entre o Modelo Chang e o modelo do ARGO.....	105
Fig. VII.2 - Formação de um estereótipo e classificação da <i>interface</i> para um usuário.....	110
Fig. VII.3 – Tabela de classificação das <i>interfaces</i> de acordo com o estereótipo.....	111
Fig. VII.4 - Tabela de classificação das interações do usuário.....	118
Fig.VII.5 - Exemplo do <i>log</i> das interações de dois usuários fictícios.....	119
Fig.VII.6 - Exemplo do <i>log</i> das interações ordenado por usuário, forma de acesso à tabela, forma de projeção/seleção e forma de junção.....	120
Fig.VII.7 - Amostragem das interações classificadas e com os pesos atribuídos.....	121
Fig.VIII.1 - Comparação de algumas funcionalidade principais das <i>interfaces</i> .....	126

# Capítulo I

## Introdução

A necessidade de interfaces de acesso a Banco de Dados (BD) de fácil manipulação (*easy-to-use*) e fácil aprendizado (*easy-to-learn*) tem sido identificada como uma área chave de pesquisa. Ainda hoje, um dos principais problemas para os usuários de um BD, ocasionais e/ou iniciantes<sup>1</sup>, é a falta de conhecimento das informações que estão armazenadas no BD, assim como sobre sua representação conceitual. Se, portanto, a interface for difícil de manipular, e exigir muito treinamento dos usuários, a interação pode ficar comprometida.

Uma das formas mais comuns de interação com o BD é através de **linguagens de consulta analíticas**, mas é também o modo mais complicado. Esse tipo de linguagem exige um amplo conhecimento do esquema conceitual sobre o qual o BD se baseia, principalmente se levados em consideração os tipos de usuários que são conhecedores apenas do domínio da aplicação e não do esquema do BD. Um outro problema é que linguagens de consulta analíticas são muito susceptíveis a erros, uma vez que exigem o conhecimento da sintaxe dos comandos para a formulação da consulta.

---

<sup>1</sup> Usuários **ocasionais** são aqueles que possuem uma baixa frequência de interação com o sistema. Os usuários **iniciantes**, podendo ser chamados também de novatos, são os que não possuem experiência com o BD e/ou com a sua interface

Com o advento de consultas em **linguagem natural**, pensou-se em ter amenizado esse problema, dados o grande poder expressivo e a informalidade desses tipos de linguagens. No entanto, a experiência com linguagem natural tem provado, ao longo desses anos, que ela é inerentemente ambígua, tanto na sua estrutura quanto na sua sintaxe [Hend77], tornando-se difícil para o computador interpretar as intenções do usuário, exigindo, em certas situações, um longo e complexo diálogo até que o sistema “entenda” a consulta do usuário. Segundo J. Littlehales e P. Hancox em [Littlehales92], linguagens naturais são fáceis de usar por pessoas comunicando-se com outras pessoas, mas elas não foram, ainda, suficientemente desenvolvidas para permitir interfaces robustas para usuários de um BD.

Como alternativa às linguagens de consulta analíticas e naturais, **linguagens de consultas visuais** foram propostas ([MacDonald75], [Ramos92]). Elas são linguagens de consultas baseadas no uso de representações visuais para descrever o domínio de interesse e expressar as requisições relacionadas. As representações visuais são caracterizadas por um formalismo visual (tais como ícones, diagramas, formulários ou tabelas, ou formalismo híbrido) que conduzem a um modo amigável e efetivo de interação com o sistema, fazendo uso de mecanismos de manipulação direta<sup>2</sup>.

Sistemas que implementam linguagens de consultas visuais são chamados Sistemas de Consultas Visuais [Batini91]. Esses sistemas incluem uma linguagem para expressar uma consulta de forma visual, e uma variedade de funcionalidades para facilitar a interação usuário-sistema. Como tal, eles são orientados a um largo espectro de usuários com habilidades técnicas limitadas e geralmente ignorantes no que tange ao conhecimento da estrutura interna de um BD.

As interfaces de consultas visuais, assim como as linguagens de consultas visuais, provêm meios para expressar o domínio da consulta de uma forma interativa e gráfica (visual, ilustrativa), fazendo uso de formalismo visual. A diferença é que a linguagem de consulta visual é mais abrangente, por envolver a especificação formal da sua sintaxe e

---

<sup>2</sup> Manipulação direta é uma forma rápida, reversível e incremental de apresentação da informação durante a interação usuário-sistema, evitando a ansiedade do usuário em ver o resultado (já que ele é apresentado de imediato) e possibilitando que o aprendizado seja obtido mais facilmente [Shneid87].

semântica. Podemos dizer que uma linguagem de consulta visual é um subconjunto de uma linguagem de programação visual.

As interfaces visuais de consulta propiciam grandes vantagens para o usuário ocasional por permitir a formulação de consultas de forma aleatória (não há uma ordem predefinida para tal), além do que permitem ao usuário uma visão dos resultados na tela à medida em que a consulta é construída. Refinamentos da consulta podem ser feitos, passo a passo, até que o resultado obtido seja satisfatório.

Muitas interfaces de consultas visuais foram empregadas para os mais variados modelos de dados existentes, modelos relacionais<sup>3</sup>, modelos semânticos<sup>4</sup>, modelos orientados a objeto<sup>5</sup>, etc. Ultimamente, a preocupação maior tem sido voltada para os dois últimos tipos de modelos, devido à necessidade de se lidar com informações estruturadas de forma muito complexa. No entanto, devido ao modelo relacional ser o mais difundido em aplicações comerciais e também devido à simplicidade do emprego e uso da representação tabular (representação visual usada em tal modelo de dados), construções de interfaces voltadas para esse modelo continuam sendo utilizadas.

Uma tendência na área de interfaces visuais é de embutir “inteligência” nas mesmas (reconhecendo as preferências, os objetivos, as metas e o próprio conhecimento do usuários sobre o sistema), de forma que a comunicação usuário-sistema se dê o mais cooperativamente possível, ou seja, a interface satisfaça às necessidades do usuário adaptando-se as mesmas. A estratégia usada para isto é o emprego de modelos de usuário ([Errico97], [Chang93]). Um modelo de usuário pode incorporar o conhecimento, a experiência e o objetivo de uma classe de indivíduos que interage com o sistema, para que uma melhor interação usuário-sistema seja alcançada.

---

<sup>3</sup> Nos modelos relacionais o esquema conceitual é representado por relações (denominadas de tabelas), sendo cada relação composta por tuplas (registros de tabelas) e atributos (campos de tabelas), onde cada atributo da relação possui um nome único.

<sup>4</sup> Os modelos semânticos possuem maior poder de expressão do que o relacional, uma vez que a representação das entidades se dá de uma forma mais aproximada do mundo “real”, pois possuem mecanismos de abstração que permitem modelar entidades complexas através de objetos complexos e relacionamentos hierárquicos entre classes de objetos.

<sup>5</sup> Os modelos orientados a objetos surgiram a partir dos modelos semânticos, herdando portanto características dos mesmos acrescidos dos conceitos da programação orientada a objeto, tais como: identidade de objetos, classes, métodos, mensagens, encapsulamento, herança, polimorfismo.

## 1.1 Objetivos da dissertação

O objetivo desta dissertação é a implementação de um **ambiente de consulta** a Bancos de Dados (BD) relacionais que permite ao usuário fazer *browsing*, “navegação” e elaboração de consultas visuais. É um ambiente que oferece portabilidade, uma vez que permite acesso aos dados gerenciados por qualquer Gerenciador de Banco de Dados Relacional. É genérico, ou seja, base de dados diversas (desde que relacionais) podem ser consultadas. Um outro objetivo, importantíssimo, é o seu aspecto adaptativo, pois este ambiente se presta a oferecer aos usuários a interface que lhes seja mais adequada, de forma que, tanto um usuário iniciante quanto um mais experiente se sintam à vontade para interagir com o BD, cada um com suas exigências particulares.

Para possibilitar o caráter adaptativo do ambiente, foram implementadas quatro interfaces de consultas visuais, as quais denominamos de: **Básica, Mediana, Avançada e SQL**, para os tipos de usuários iniciante, mediano, experiente e especialista, respectivamente (Fig.I.1). Inicialmente a cada tipo de usuário é atribuído a interface adequada. Como o perfil de um usuário pode ser modificado, para refletir as mudanças em suas interações com o BD, a interface adequada ao usuário também poderá ser modificada. Desta forma, dizemos que o ambiente se “adapta” as mudanças de perfil dos usuários.

Tipo Usuário	Interface
Iniciante	Básica
Mediano	Mediana
Sofisticado	Avançada
Especialista	SQL

**Fig.I.1 - Tipos de usuário e suas respectivas interfaces**

Cada uma dessas interfaces possui um diferencial que a torna apropriada a seu respectivo tipo de usuário. Quanto mais iniciante for o usuário, mais processamento interno é realizado pelo sistema internamente, para evitar que o usuário precise tomar conhecimento a respeito da estrutura do BD. Quanto mais especialista for o usuário, mais liberdade é oferecida para que ele decida quanto esforço vai despende na elaboração das suas consultas.

Chamamos de iniciante um usuário que não possui familiaridade com o BD, ou seja, não tem conhecimento sobre o esquema conceitual do BD e/ou sobre o domínio da informação armazenada no BD. Este tipo de usuário normalmente é infreqüente (interage esporadicamente com o BD) e gera consultas de forma simples, que não exijam muito esforço na elaboração da mesma. Um outro fator que caracteriza o tipo de usuário iniciante é o de não possuir familiaridade com o uso do computador.

O tipo de usuário especialista, ao contrário do iniciante, é um profundo conhecedor de BD, tanto com relação ao esquema conceitual quanto às informações contidas no BD. Estes usuários são ditos profissionais, ou seja, conhecem bem uma linguagem de consulta analítica (tipo SQL), possivelmente são projetistas de BDs, e até mesmo são ABDs (Administradores de Bancos de Dados).

Nas faixas intermediárias estão os tipos de usuário mediano (mais próximo do iniciante) e experiente (mais próximo do especialista). O tipo de usuário mediano engloba as características do iniciante, só que num grau de intensidade menor, ou seja, eles podem não ser tão ignorantes quanto às informações contidas no BD, até porque são usuários que possuem uma freqüência de interação maior do que os iniciantes e são mais adaptados ao uso do computador.

O tipo de usuário experiente, por sua vez, possui algumas das características do especialista, também com um menor grau de intensidade. O seu conhecimento sobre o BD, por exemplo, não é tão refinado quanto o do especialista, mas ele possui conhecimentos suficientes para realizar consultas com maior poder de expressão, se comparado aos tipos de usuário iniciante e mediano.

## **1.2 Estrutura da dissertação**

Esta dissertação é composta de oito capítulos. O primeiro capítulo é esta introdução.

No capítulo II, apresentamos alguns trabalhos relacionados ao tema desta dissertação.

A arquitetura do ambiente de consulta é apresentada no capítulo III, seus módulos componentes são descritos, ou seja, seus objetivos e funcionalidades.

No capítulo IV, as especificações das interfaces são detalhadamente descritas, e ilustradas com alguns exemplos.

No capítulo V, são apresentadas as especificações do metabanco de dados e da interface de apoio à construção do metabanco.

As estruturas de dados necessárias às interfaces e às meta-informações do BD, como também os principais algoritmos que manipulam essas estruturas, são apresentados no capítulo VI.

O aspecto adaptativo das interfaces, onde são tratados os componentes para a construção do modelo de usuários, assim como as estruturas de dados necessárias, são tratados pelo capítulo VII.

Enfim, no capítulo VIII, apresentamos nossas considerações finais e propostas para trabalhos futuros.

## Capítulo II

### Trabalhos Relacionados

Além do QBE (*Query-by-Example*) [Zloof77], pioneira na representação tabular, que na época foi projetado como sendo uma interface de alto nível para BDs relacionais, e ainda hoje se ser uma das mais conhecidas interfaces visuais de acesso a BD, muitas outras interfaces foram desenvolvidas e fundamentadas em modelos de dados diversos, citamos algumas delas:

- VQL [Mohan90]: interage com o modelo de dados orientado a objeto SSONET, suas consultas são graficamente especificadas usando um conjunto de primitivas visuais (sintaxe formal e estrutura semântica para criação das consultas gráficas)
- IQL [Ramos92]: voltada para BD relacional, fornece interação gráfica na formulação da consulta, e ainda se encarrega de realizar a otimização da mesma;
- MEDUSA [Hietala93]: uma interface inteligente para BD capaz de assistir usuários no seu aprendizado;
- QBB [Dix95]: usa técnicas de *machine learning* para gerar consultas a BDs a partir de exemplos fornecidos pelo usuário;

- HIBROWSE [Ellis95]: usa abordagem de consulta dinâmica<sup>6</sup> na manipulação dos dados de um BD relacional;
- PESTO [Carey96]: interface para BDs orientados a objeto, que permite “navegação” sobre os objetos de um BD e suporta o paradigma *query-in-place* [Carey96] na realização das consultas ao BD.

Todas as interfaces citadas se focam na idéia de que um ambiente de consulta a BD deve ser de fácil manipulação, não apenas por usuários que possuam experiência mas também por usuários iniciantes. Estas interfaces, além de permitir a elaboração de consultas visuais, com um certo poder de expressividade, normalmente permitem fazer *browsing* nos dados e/ou na estrutura de um BD.

A preocupação com o usuário não-profissional (iniciante e/ou ocasional) é notória. Normalmente, buscam-se formas de minimizar a exigência do conhecimento do usuário sobre o BD, através da utilização de interfaces de consulta visual, por serem de fácil manipulação e requererem pouco tempo de aprendizado.

Nosso objetivo, neste capítulo, é apresentar alguns trabalhos realizados na área de interfaces visuais para BDs. Escolhemos, dentre os vários trabalhos acima citados, as interfaces HIBROWSE e MEDUSA, e as descrevemos a seguir. O motivo da escolha dessas interfaces se deu (1) pelas semelhanças entre algumas das abordagens usadas por essas interfaces e as nossas interfaces; (2) como o nosso objetivo principal não é o de fazer uma comparação das nossas interfaces com todas as demais interfaces existentes, nos restringimos a apresentar apenas duas delas, como exemplo; (3) por serem as interfaces HIBROWSE e MEDUSA bastante referenciadas em vários artigos.

## 2.1 HIBROWSE

O sistema HIBROWSE ([Ellis95], [Pollitt93]) apresenta uma visão do banco de dados que reflete o interesse do usuário. O acesso é realizado manipulando o conteúdo do BD

---

<sup>6</sup> Consulta dinâmica envolve o controle interativo do usuário sobre os parâmetros visuais de consulta (exemplo: *sliders bar*) que geram uma rápida e animada apresentação visual do resultado do BD [Shneid87], fazendo uso, desta forma, de mecanismos de manipulação do conteúdo ao invés da manipulação da estrutura.

mostrado e não a consulta baseada na estrutura. Esta abordagem tem várias vantagens para o usuário. O acesso não requer o entendimento da estrutura do banco de dados, tornando-o mais intuitivo. Conseqüentemente, consultas podem ser produzidas mais rapidamente do que com as técnicas de acessos tradicionais. Adicionalmente, o usuário pode fazer *browsing* no conteúdo do banco de dados.

O HIBROWSE não é uma interface genérica, ou seja, ele foi desenvolvido para uma aplicação específica, no caso hotéis [Ellis95].

### 2.1.1 Características da interface

A principal característica do HIBROWSE é apresentar uma visão do conteúdo do banco de dados para que o usuário consulte e faça *browsing* selecionando o conteúdo. A apresentação dessa informação é obtida usando algumas janelas (ver Fig.II.1), cada uma delas apresentando:

- **dados base** (ou seja, os dados reais de uma tabela do BD apresentados de acordo com a visão do usuário), como na janela *hotels* que mostra o nome, a cidade, o número de quartos e o número de estrelas de cada hotel
- **informações resumidas da tabela base** (ou seja, dados obtidos com base na visão apresentada na janela de dados base), como nas janelas *AA star rating*, *rooms* e *tourist board areas*, que apresentam algumas informações sobre hotéis
- **dados relacionados com a tabela base** (ou seja, os dados da tabela que se relaciona com a tabela base), como na janela *companies* que mostra as companhias que têm participação nos hotéis

As janelas de informações resumidas no HIBROWSE para hotéis apresentam dados do banco de dados de diferentes formas: a janela *AA star rating* (Fig.II.1), exibe a contagem do número de hotéis e seus respectivos números de estrelas; a janela de *rooms* mostra a contagem dos hotéis dentro da faixa de quartos desejada; a janela *tourist board area* apresenta os dados baseado na localização geográfica.

HIBROWSE for HOTELS				
hotels				
Name	City	rooms	rating	
22 Jernyn Street	London	18	U	
Abbey Court	London	22	U	
Abbey Grang Hotel	London	8	U	
Abbey Hotel	Great Malvern - Mal ...	106	3	
Abbey Hotel <SALE>	Wymondhan	26	2	
Abbey House Hotel	Reading	22	U	
Abbey Lodge Hotel	London	20	U	
Abbey Park Resort Hotel	York	85	U	
Abbeylee Hotel	Chesterfield	11	1	
3451		207927		
Tourist board areas				
Hotels	name			
3451	UK			
33	Channel Islands			
811	Midlands			
638	North England			
10	Northern Ireland			
531	Scotland			
1207	South England			
220	Wales			
All companies				
Name	hotels	rooms		
ABBLEY GROUP	4	113		
ADNAMS HOTELS LTD	4	81		
ALLOA HOTELS IN SCOTLAND	11	185		
ALNMOUTH HOTELS	2	40		
AMBERLEY CASTLE & INM ON THE ...	2	34		
ANSELLS HOTELS	10	349		
APPOLO HOTELS	3	107		
AQUARIUS HOTELS	4	268		
252		3929		
rooms				
Range	hotels			
> 1000	1			
500-999	19			
200-499	133			
100-199	432			
50-99	698			
30-49	671			
20-29	418			
10-19	638			
AA star rating				
Rating	hotels			
5 star	24			
4 star	200			
3 star	983			
2 star	505			
1 star	50			
lodge	136			
Accom only	0			
Unclassified	1553			

Fig.II.1 – Interface do HIBROWSE para hotéis

### 2.1.2 Consulta e *browsing* ao BD

O usuário realiza a consulta ao BD simplesmente realçando ou marcando os registros requeridos para fazer parte da consulta (Fig.II.2). Nenhuma pesquisa ocorre até que a seta de execução, no canto inferior direito da tela, seja selecionada, e neste caso uma nova apresentação das janelas do HIBROWSE é mostrada, dando ao usuário a visão do banco de dados de acordo com o critério escolhido.

Ao ser executada a consulta, com base nos registros selecionados, todas as janelas são automaticamente atualizadas para refletir a mudança causada pela consulta.

HIBROWSE for HOTELS				
hotels				
Name	City	rooms	rating	
22 Jernyn Street	London	18	U	
Abbey Court	London	22	U	
Abbey Grang Hotel	London	8	U	
Abbey Hotel	Great Malvern - Mal ...	106	3	
Abbey Hotel <SALE>	Wymondhan	26	2	
Abbey House Hotel	Reading	22	U	
Abbey Lodge Hotel	London	20	U	
Abbey Park Resort Hotel	York	85	U	
Abbeyle Hotel	Chesterfield	11	1	
3451		207927		
All companies				
Name	hotels	rooms		
ABBLEY GROUP	4	113		
ADNAMS HOTELS LTD	4	81		
ALLOA HOTELS IN SCOTLAND	11	185		
ALNMOUTH HOTELS	2	40		
AMBERLEY CASTLE & INM ON THE ...	2	34		
ANSELLS HOTELS	10	349		
APPOLO HOTELS	3	107		
AQUARIUS HOTELS	4	268		
252		3929		
Tourist board areas				
Hotels	name			
3452	UK			
34	Channel Islands			
812	Midlands			
639	North England	*		
11	Northem Ireland			
532	Scotland	*		
1208	South England			
220	Wales			
AA star rating				
Rating	hotels			
5 star	24			
4 star	290			
3 star	983			
2 star	505			
1 star	50			
lodge	136			
Accom only	0			
Unclassified	1553			
rooms				
Range	hotels			
<1000	1			
500-999	19			
200-499	133			
100-199	432			
50-99	698			
30-49	671			
20-39	418			
10-19	638			

Fig.II.2 – Elaborando uma consulta no HIBROWSE para hotéis

Além de elaborar consultas visuais, o usuário pode fazer *browsing* sob o conteúdo, ou seja, apenas selecionando o conteúdo de algum campo, uma nova janela de dados é aberta (sobrepondo as janelas já em exibição) com conteúdos referente ao conteúdo selecionado.

### 2.1.3 Vantagens e desvantagens do HIBROWSE

Uma das grandes vantagens do HIBROWSE é a abordagem de consulta dinâmica, ou seja, a visão dos dados apresentada em todas as janelas é dinamicamente alterada quando uma consulta é realizada em qualquer uma das janelas. Outra vantagem é a facilidade de realizar uma consulta, através de marcas feitas nos registros desejados, permitindo total independência do conhecimento da estrutura do BD. Tornando este interface essencialmente adequada para usuários iniciantes.

No entanto, o HIBROWSE é uma interface específica para uma aplicação (hotéis). Além disso, o poder de expressão das consultas nesta interface são limitados, não se pode gerar uma consulta usando um padrão de pesquisa como exemplo, ou seja, parte de um valor de

um campo. Projeção de campos, a critério do usuário, também não é permitida. Junções explícitas/implícitas (conjuntivas ou disjuntivas) não são possíveis de realizar.

Esta interface não se adequa aos objetivos, as preferências, as metas, ao nível de conhecimento do usuário, ou seja, não é uma interface adaptativa. Independente do tipo de usuário (do conhecimento e da experiência) o HIBROWSE apresenta os mesmos critérios de interação.

## **2.2 MEDUSA – Multimode Database User interface with computer Supported learning Aid**

Esta interface, além de combinar o poder das linguagens de consulta gráfica e natural, e por isso é chamada de interface multimodal<sup>7</sup>, nela é incluída o auxílio ao aprendizado do sistema. A idéia é diminuir a distância cognitiva entre o usuário e o sistema.

P. Hietala *et al* em [Hietala93] propuseram a MEDUSA a partir da idéia de que ela seria uma interface inteligente para BD, capaz de assistir usuários enquanto o mesmo está aprendendo a usar a interface e posteriormente, sugerindo a forma mais apropriada de interação usuário-sistema.

### **2.2.1 Visão geral da MEDUSA**

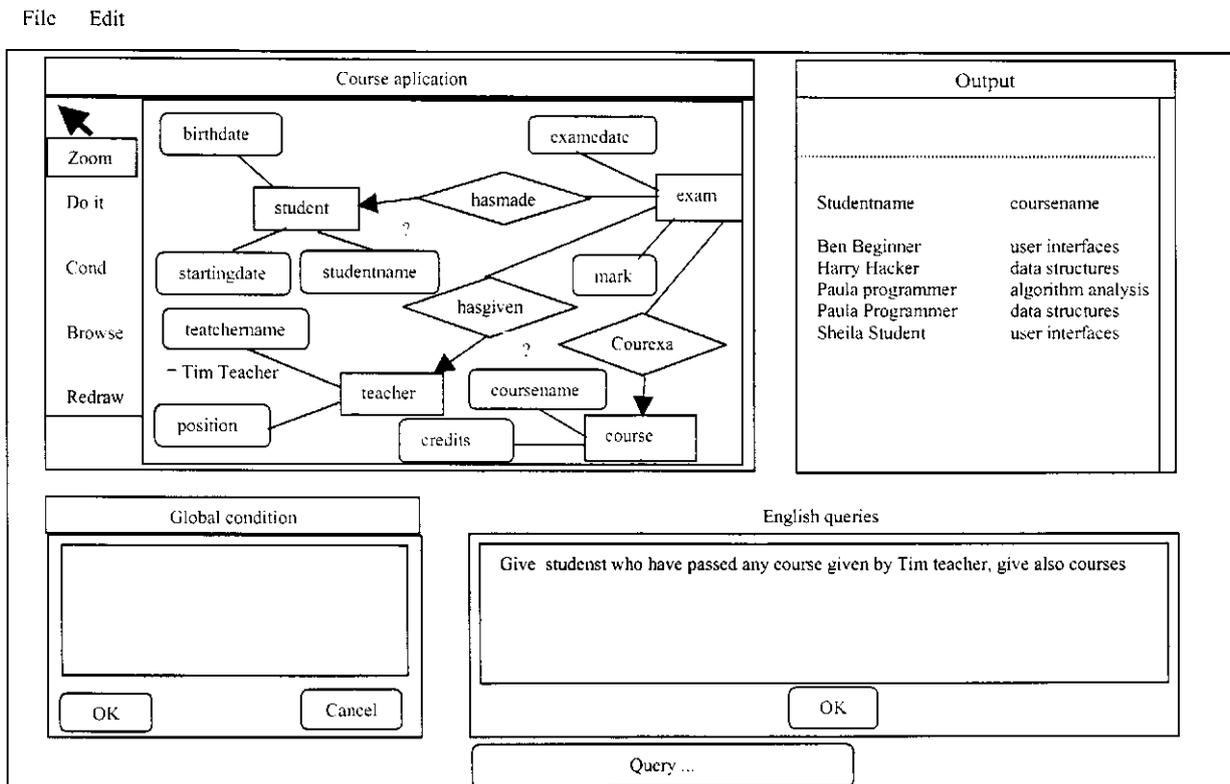
A apresentação gráfica da MEDUSA é baseada no uso de diagramas ER (Entidade-Relacionamento), que descrevem esquemas de BDs. Neste tipo de apresentação, o usuário seleciona o objeto de interesse e, usando alguma forma de diálogo, pode especificar as condições envolvidas.

A interface para a consulta em linguagem natural é bem simples, na realidade, é apenas uma janela de diálogo onde o usuário pode expressar suas consultas entrando com o texto (em inglês, por exemplo).

---

<sup>7</sup> São interfaces que combinam diferentes métodos de representação (diagramas, ícones, textos, tabelas,...), onde estas representações são manipuladas independentemente umas das outras

Uma visão geral da MEDUSA é apresentada na Fig.II.3, onde uma aplicação para controle acadêmico de uma universidade é tratada.



**Fig.II.3 – Uma visão geral da interface MEDUSA**

Uma consulta pode ser realizada em uma das duas representações (gráfica ou natural), ou em ambas, ou seja, pode haver uma combinação das duas para que o usuário obtenha a consulta desejada da forma que melhor lhe convier.

Devido às diferentes alternativas disponíveis para o usuário, a MEDUSA é capaz de analisar diferentes possibilidades de realizar uma mesma operação, e, desta forma, poder fazer comparações entre os diversos tipos de interações, e sugerir o tipo mais adequado.

### 2.2.2 A interface

A interface MEDUSA foi implementada no Apple Macintosh, usando uma *workstation* gráfica. A interface do usuário é baseada no uso de *menus*, diálogos, e janela de esquema gráfico e segue as convenções das aplicações do Macintosh. Na Fig.II.3 são apresentadas

algumas das janelas da interface: janela de saída (denominada *Output*), janela de esquema gráfico (exibindo o diagrama ER da aplicação correspondente), uma janela de diálogo de condição global associado (denominada *Global Condition*), e um diálogo para entrada das expressões em linguagem natural.

Ao selecionar um atributo do esquema ER, o usuário pode optar por abrir uma janela de diálogo e nela especificar as condições de pesquisa para tal atributo; e ainda, informar se os valores do atributo selecionado devem ser exibidos na janela de saída, etc. Um exemplo é apresentado na Fig.II.4, onde uma janela de diálogo para o atributo *coursename* do esquema ER foi aberto.

Há também a possibilidade de editar uma condição global (ou seja, uma condição que não diz respeito apenas a um atributo) na janela de diálogo de condição global (Fig.II.3). Uma das formas de combinar o uso da linguagem gráfica e da linguagem natural é selecionando o objeto de interesse no esquema gráfico, e expressando as condições em linguagem natural na janela de diálogo de condição global.

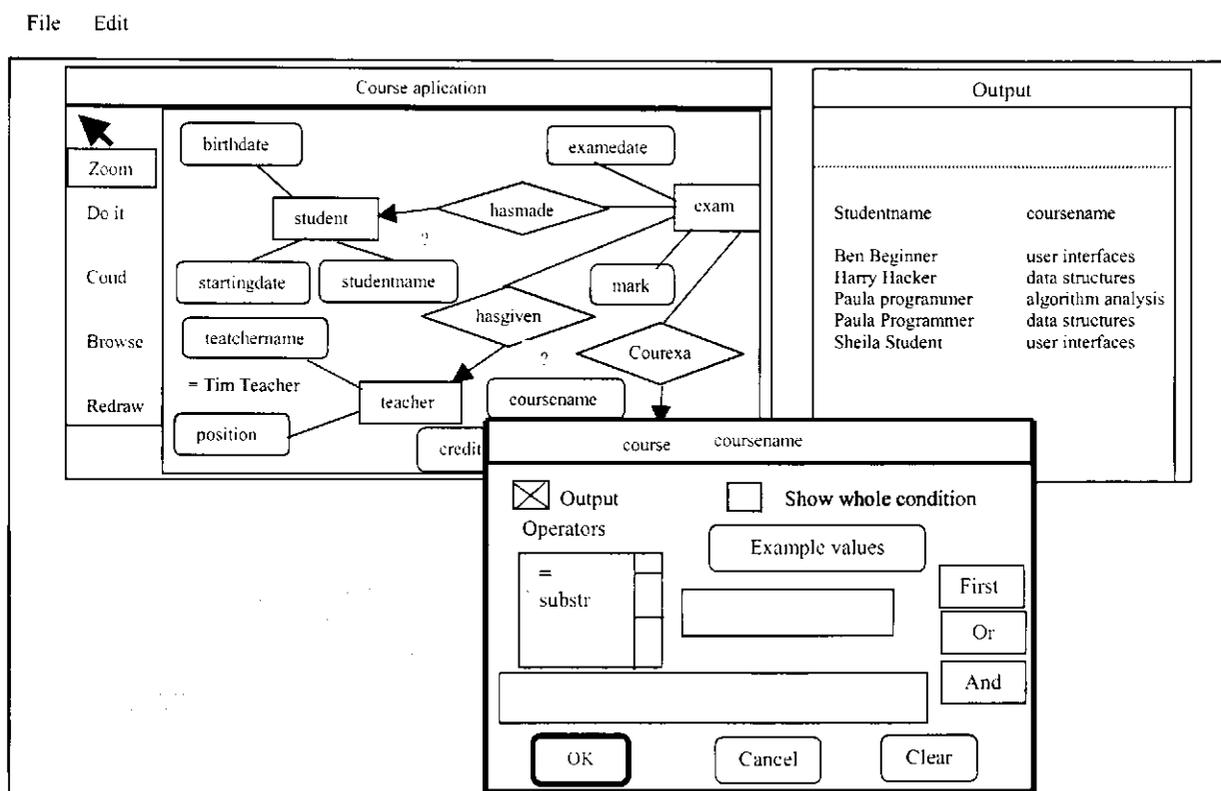


Fig.II.4 – Um exemplo da MEDUSA com janelas de diálogos abertas

### 2.2.3 Auxílio ao aprendizado

O aprendizado automático na MEDUSA é feito em três fases: 1) monitoração do usuário, onde é implementado um rastreamento extensivo das ações do usuário, e a partir dele poder oferecer respostas às ações realizadas; 2) análise das ações do usuário, ou seja, análise de uma seqüência de ações elementares que executam uma tarefa, tornando possível a comparação dos esforços necessários para gerar as diferentes seqüências obtidas; 3) ajuda ao usuário, onde o sistema fornece passos, ou estratégias, necessários à execução de uma tarefa, além de uma explanação da causa dos erros cometidos pelo usuário.

### 2.2.4 Vantagens e desvantagens da MEDUSA

Duas vantagens podem ser observadas na interface MEDUSA: uma delas é sua representação multimodal, que possibilita ao usuário ter a liberdade para escolher a melhor forma de expressar suas consultas, seja usando a representação gráfica, a representação em linguagem natural, ou ambas. A outra vantagem é a capacidade de fornecer auxílio ao aprendizado automático, ou seja, ser capaz de reconhecer as ações dos usuários (a partir do rastreamento das ações), sugerir possibilidades de realização de uma tarefa, ajudas *online* na realização das operações, dentre outras.

Embora esta interface possua três formas diferentes de elaborar suas consultas (forma gráfica, natural, ou ambas), percebemos que nenhuma delas satisfaz plenamente a um usuário iniciante. A representação gráfica é mais difícil de ser usada por representar o esquema do BD em diagramas ER. A linguagem natural, que poderia ser o modo mais “natural” e mais adequado para este tipo de usuário, devido as limitações computacionais ainda hoje existentes neste tipo linguagem, pode gera insatisfação ao tipo de usuário iniciante quando ele tentar formular consultas que não consigam ser bem entendidas pela MEDUSA.

Outros problemas encontrados nesta interface são: não permitir a ordenação (ascendente ou descendentes) dos dados de saídas da consulta; não permitir a seleção de valores a partir de um padrão de entrada fornecido; não realizar a manipulação direta do conteúdo e

sim da estrutura; não se adaptar às necessidades, às exigências e aos conhecimentos dos diversos tipos de usuários que a manipula; não possuir portabilidade.

## 2.3 Conclusão

Algumas das idéias empregadas nas interfaces HIBROWSE e MEDUSA podem ser vislumbradas nas interfaces do ARGO (denominação para o ambiente de consulta desenvolvido nesta dissertação), enquanto que outras detectadas como insuficientes, são encontradas de forma mais completa no ARGO. Na Fig.II.5 é apresentado um quadro comparativo das funcionalidades encontradas nas interfaces HIBROWSE, MEDUSA e nas interfaces do ARGO.

<b>Funcionalidades</b>	<b>HIBROWSE</b>	<b>MEDUSA</b>	<b>ARGO</b>
Manipulação do conteúdo	X		X
Manipulação da estrutura		X	X
Obtenção das consultas de forma dinâmica	X		X*
Seleção de valores marcados	X		X
Seleção de valores informados (digitados)		X	
Seleção de valores por padrão			X
Ordenação de campos (ascendente e descendente)	X		X
Projeção de campos			X
Junções explícitas (conjuntivas e/ou disjuntivas)			X
“Navegação” sobre o conteúdo do BD	X		X
“Navegação” sobre a estrutura do BD		X	
Auxílio ao aprendizado		X	
Adaptativo			X
Portabilidade	X		X
Genérico		X	X

**Fig.II.5 – Quadro comparativo das funcionalidades das interfaces**

\* Os resultados das consultas podem ser exibidos de forma dinâmica ou não, depende da interface que estiver sendo usada e do tipo de interação realizado

Com relação ao HIBROWSE, destacamos o uso da abordagem de manipulação direta do conteúdo e obtenção dos resultados das consultas de forma dinâmica (onde os dados são imediatamente apresentados quando certas ações são realizadas e o conteúdo das janelas são modificados automaticamente). A elaboração das consultas a partir de exemplos “marcados”, também é uma funcionalidade compatível com as interfaces do ARGO, além do fato de ser possível realizar ordenação dos dados nas formas ascendente e descendente; fazer “navegação” sobre o conteúdo do BD, etc. No entanto, o HIBROWSE é uma interface específica, totalmente voltada para uma determinada aplicação, enquanto que as interfaces do ARGO podem ser usadas sobre diversos BDs (relacionais), sendo portanto genéricas.

A interface MEDUSA, por sua vez, se assemelha um pouco com o ARGO no que tange à monitoração das interações dos usuários, sendo a abordagem usada pela MEDUSA voltada para o reconhecimento dos planos (seqüência de ações) dos usuários, sugerindo, em tempo de real, a ação mais adequada para realizar uma determinada operação. A abordagem do ARGO, por sua vez, é baseada no reconhecimento do estereótipo de usuários, e a partir dele sugere a interface mais adequada ao usuário. No entanto, a MEDUSA não é uma interface adaptativa, ela apenas auxilia, em tempo real, o usuário no uso da interface. Enquanto que o ARGO é um ambiente adaptativo, a análise dos estereótipos é feita periodicamente, podendo o mesmo ser alterado à medida em que o usuário for interagindo de diferentes formas com o BD, e desta forma pode ser alterado também o tipo de interface adequado ao usuário.

## Capítulo III

# Arquitetura do Ambiente ARGO

Apresentamos um ambiente adaptativo para consultas visuais, *browsing* e “navegação” a BDs relacionais, o qual denominamos de **ARGO**<sup>8</sup>. Este ambiente é composto por: (1) quatro interfaces de consultas visuais, sendo cada uma delas adequada a um tipo específico de usuário (Fig.I.1); (2) modelos de usuário, que permitem o caráter adaptativo do ambiente; (3) metabancos de dados, criados com a finalidade de prover suporte às interfaces no que se refere à informação a ser apresentada durante as interações usuário-BD.

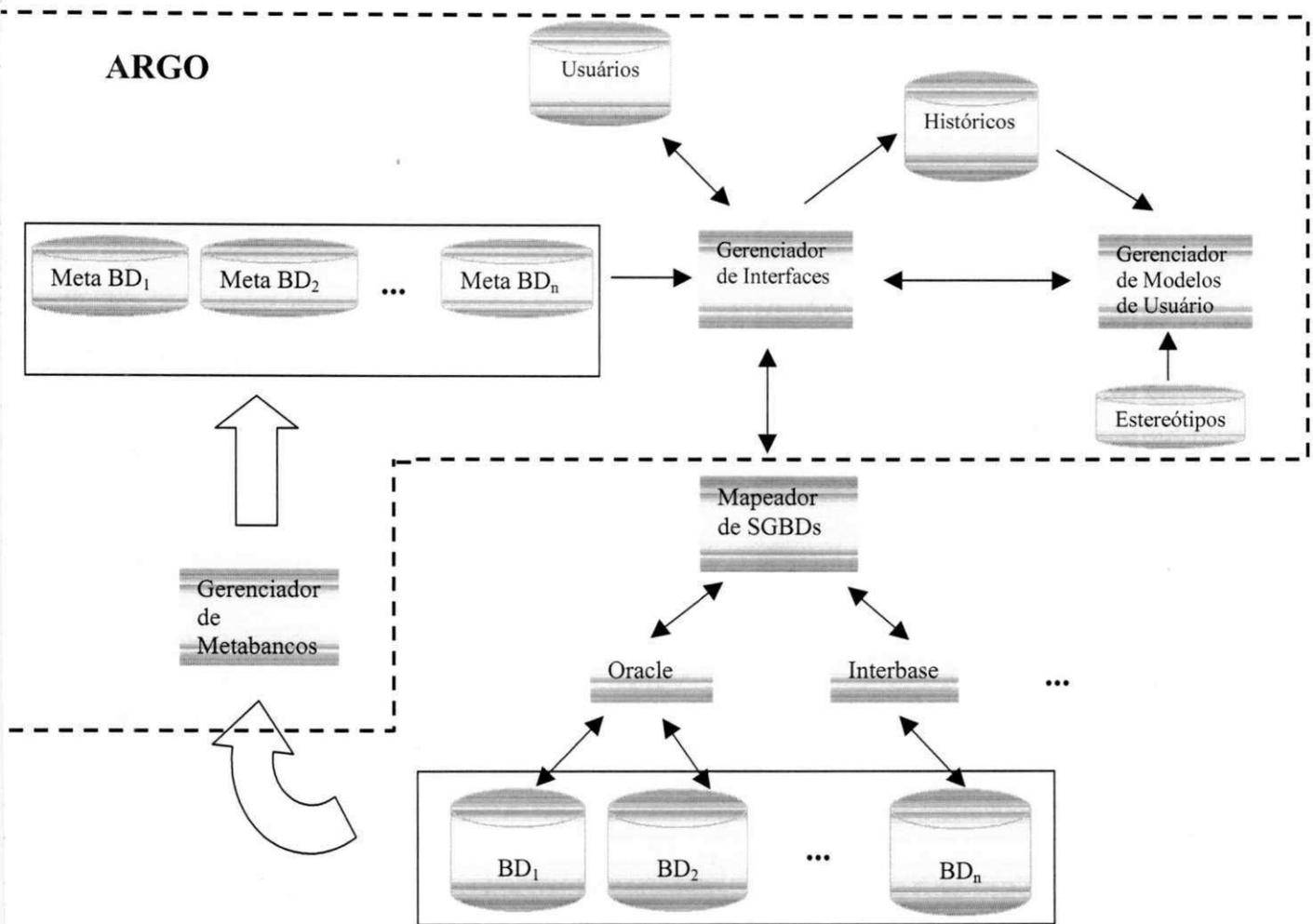
### 3.1 A arquitetura do ARGO

A arquitetura do ARGO é composta por módulos de controle que lidam com as interfaces, os modelos de usuário, o mapeador de SGBDs, e os metabancos de dados (Fig.III.1).

---

<sup>8</sup> ARGO – navio grego no qual os argonautas embarcaram em busca da conquista do Tosão de Ouro. A palavra grega *argos* significa ágil. Minerva (deusa da sabedoria) presidiu a construção deste navio. O mastro foi feito de um carvalho da floresta Dodona, o que fez dizer que o navio Argo pronunciava oráculos, e por isso lhe deram os epítetos de *facundo* e de *sagrado*.

O módulo Gerenciador das Interfaces é responsável por oferecer ao usuário a interface adequada e por manter atualizado um histórico com informações a respeito das formas de interações usadas pelo usuário e das consultas por ele geradas. Este módulo interage com o módulo Gerenciador dos Modelos de Usuário, para a partir dos dados do histórico atualizar o estereótipo do usuário; com o módulo Mapeador dos SGBDs, para obter o acesso aos dados dos BDs; e com os metabancos de dados, para obter informações sobre os campos, das tabelas, que são importantes às consultas a um usuário.



**Fig.III.1 – Arquitetura do ARGO**

Apresentamos uma visão geral dos principais módulos da arquitetura do ARGO nas subseções deste capítulo, os detalhes serão tratados nos capítulos seguintes.

### **3.1.1 Módulo Gerenciador das Interfaces**

Como o ambiente dispõe de quatro interfaces diferentes, é preciso gerenciar o uso das mesmas, ou seja, fornecer a interface correta ao usuário no momento em que ele abrir uma sessão no ARGO, assim como fornecer meios adequados para a identificação do estereótipo, a partir de um histórico das interações e consultas geradas pelo usuário.

O histórico é atualizado durante a interação usuário-sistema. Este histórico é mantido por um tempo pré-definido, e serve para atualizar o modelo de usuário. Nele estão contidas informações a respeito das formas de interação usuário-sistema e sobre as consultas realizadas pelo usuário.

### **3.1.2 Módulo Gerenciador dos Modelos de Usuário**

O módulo de gerenciamento dos modelos de usuário é responsável pela atualização dos estereótipos de usuários, um dos componentes de cada modelo de usuário. Os modelos de usuário refletem o aspecto adaptativo do ambiente ARGO.

O aspecto adaptativo do ARGO se materializa da seguinte forma: a interface apresentada ao usuário, quando ele abre uma sessão no ambiente, será aquela definida como adequada ao seu estereótipo corrente. Com o passar do tempo, e pela análise do conteúdo do histórico, este estereótipo pode ser alterado à medida em que o usuário for interagindo de diferentes formas com o BD. Como a interface adequada ao estereótipo obtido pode ser diferente da anterior, esta nova interface é a que o ARGO apresentará para o usuário nas sessões subsequentes.

### **3.1.3 Módulo Gerenciador dos Metabancos de Dados**

Nos metabancos são registradas informações sobre todas as tabelas do BD, por exemplo os campos relevantes ao usuário. Estas informações são necessárias ao funcionamento do ARGO, uma vez que a exibição dos dados de uma tabela de um BD é feita baseada nas mesmas. Portanto, para que as tabelas de um BD possam ser acessadas pelo ARGO, este BD deverá, obrigatoriamente, possuir um metabanco associado.

### 3.1.4 Mapeador dos SGBDs

O ARGO pode ser usado sobre qualquer SGBDR (Sistema de Gerenciamento a Banco de Dados Relacional), desde que esse gerenciador possa ser configurado no BDE (Borland Database Engine).

Para cada BD a ser consultado pelas interfaces do ARGO, uma configuração apropriada deverá ser feita no BDE. Para tal, a criação de um *alias* para este BD é exigida, e é a partir dele que o BD é acessado através das interfaces.

#### 3.1.4.1 Borland Database Engine (BDE)

É um *software* de conectividade usado pelo Delphi, Delphi Client/Server, Paradox for Windows, e Visual dBase for Windows. Ele oferece um robusto conjunto de características para assistir desenvolvedores de aplicações cliente-servidor, sejam algumas delas:

- API (Application Program Interface) uniforme e consistente para acessar múltiplos bancos de dados, nos mais variados formatos, incluindo: dBase, Paradox, Texto, InterBase, Oracle, Sybase, e Microsoft SQL Server, assim como formato de dados do ODBC (Open DataBase Connectivity) da Microsoft;
- É idealmente apropriado para aplicações cliente-servidor pois permite acesso tanto local quanto remoto a um servidor de dados;
- Suporta totalmente as funcionalidades a 32 bits, incluindo *multi-threading*, *multi-tasking* preemptivo, *universal names convention* (UNC), e nome de arquivos longos;
- Múltiplas consultas podem ser executadas em *background* enquanto usa as características do BDE em *foreground*;
- Múltiplas aplicações podem ser executadas simultaneamente acessando os mesmos arquivos do banco de dados.

- Servidores podem ser acessados pelo *pathname*

Sua arquitetura é baseada em *driver*, ou seja, cada formato de banco de dados distinto requer um *driver* de BDE separado. Estender o BDE para acessar um sistema de banco de dados adicional, é fácil, basta instalar o *driver* apropriado de BDE ou *driver* de ODBC para tal banco de dados.

O BDE fornece transparência de localização da base de dados, que tanto pode ser local quanto remota.

## Capítulo IV

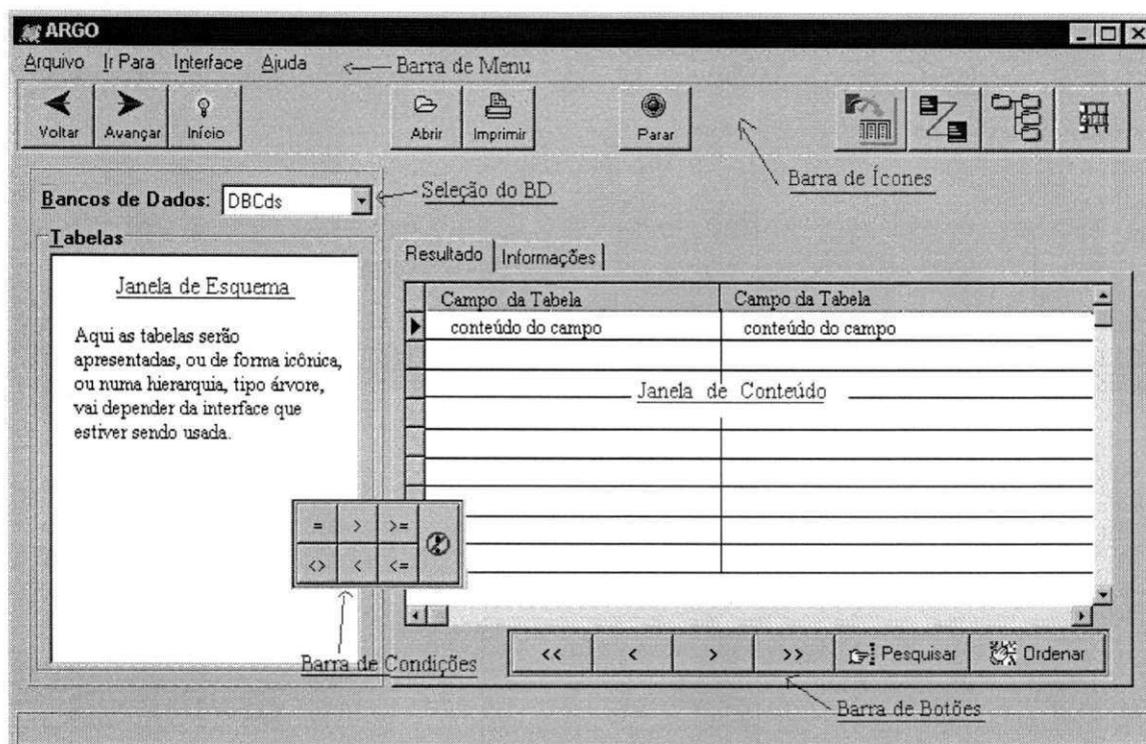
# Especificação das Interfaces do ARGO

Neste capítulo, apresentamos cada uma das interfaces do ARGO (Básica, Mediana, Avançada, SQL) e as funcionalidades que são comuns a todas elas.

### 4.1 Padrão de apresentação das interfaces

As interfaces têm funcionalidades em comum no que diz respeito às opções do *menu*, aos botões principais e, em alguns casos, às janelas de exibição de dados. Podemos dizer que as principais funcionalidades são padronizadas.

Como podemos ver na Fig.IV.1, os conceitos comuns são: Barra de Menu, Barra de Ícones, Barra de Botões e Barra de Condições; além da Janela de Esquema e Janela de Conteúdo.

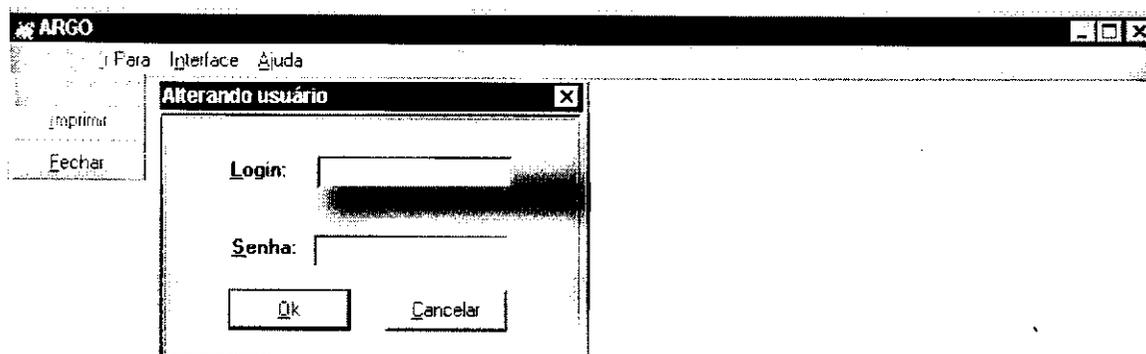


**Fig-IV.1 - Definição de janelas e botões**

Cada um desses componentes é explicado nas subseções seguintes. Como nesta seção estamos tratando dos componentes comuns a todas as interfaces, e havendo casos em que um componente (comum) tem uma funcionalidade a mais (dependendo da interface que estiver sendo usada), os detalhes adicionais de cada uma delas são descritos nas seções correspondentes.

#### 4.1.1 Barra de Menu

Na Barra de Menu são disponibilizadas as opções de abrir e fechar o ambiente ARGO; voltar e avançar sobre as consultas já realizadas; mudar a interface que esta sendo usada para uma outra; e dicas de utilização do ARGO. Algumas das opções possuem *sub-menus* ou janelas para entrada de dados. Segue-se a descrição das opções da Barra de Menu.



**Fig.IV.2 - Seleção da opção *Arquivo/Abrir* da Barra de Menu**

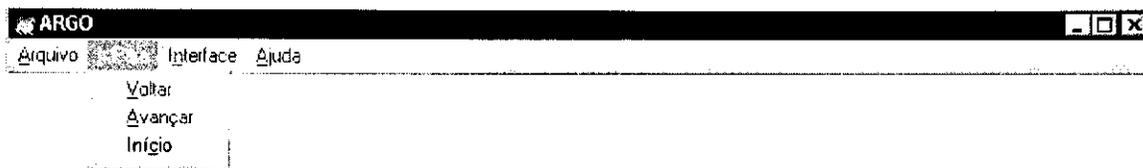
Quando a opção **Arquivo/Abrir** é escolhida, uma janela é aberta e nela é requisitado o *login* e a senha do usuário que deseja usar o ARGO. É necessário que o usuário tenha se cadastrado antes de tentar usar pela primeira vez o ARGO (este cadastro é feito por uma ferramenta de auxílio, disponibilizada à parte, maiores detalhes no capítulo V).

Caso o usuário já esteja cadastrado, a interface que lhe for adequada, de acordo com seu estereótipo (detalhes no capítulo VII), é apresentada.

A opção **Arquivo/Imprimir** permite que o conteúdo da consulta, apresentado na Janela Conteúdo, seja impresso.

A opção **Arquivo/Fechar** finaliza a sessão do usuário. Quando isto ocorre a execução do ARGO é finalizada.

Para que um usuário possa fazer *browsing* nas consultas geradas por ele na sessão corrente, a opção **Ir Para** é disponibilizada. Esta opção possui os *sub-menus* Voltar, Avançar e Início (Fig.IV.3). Antes de explicar cada um deles, vale ressaltar que toda consulta ao BD durante a sessão do usuário é mantida num histórico (capítulo VI, seção 6.2.1.3). Esse histórico é mantido por tipo de interface (isto é, uma consulta obtida a partir de uma interface não é visualizada nas demais).

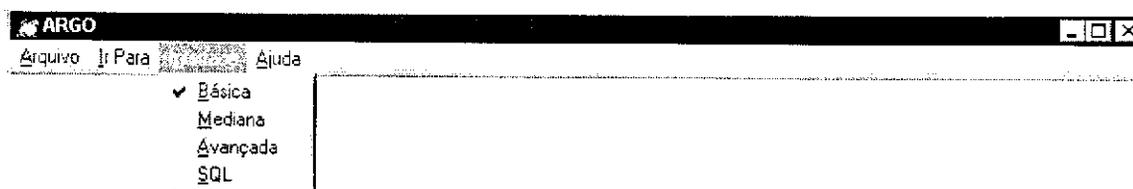


**Fig.IV.3 – Seleção da opção *Ir Para* da Barra de Menu**

Quando a opção **Ir Para/Voltar** é selecionada o resultado é a exibição do conjunto-resposta da consulta obtida imediatamente anterior à consulta corrente. Suponha um exemplo em que as consultas geradas foram: ConsultaAna56, ConsultaAna57, ConsultaAna58, ConsultaAna59, ConsultaAna60 (o ARGO guarda a formulação das consultas no histórico, catalogada como Consulta+login\_usuario+num\_consulta, ver detalhes no capítulo VI). Suponha ainda que a consulta corrente na Janela de Conteúdo seja ConsultaAna58. Tendo sido selecionado o botão Voltar, o resultado obtido é a exibição do conjunto-resposta obtido com a re-execução de ConsultaAna57. O conjunto-resposta de uma consulta pode ser diferente (vai depender do estado corrente do BD), mas a consulta a ser executada é a mesma, isto é, os campos selecionados, as condições impostas, etc.

A opção **Ir Para/Avançar** tem o mesmo propósito da opção Ir Para/Voltar sendo que, ao invés de ser obtida a consulta anterior será obtida a consulta seguinte. Para o exemplo dado acima, sendo ConsultaAna58 a consulta corrente, o resultado da seleção do botão Avançar é a exibição do conjunto-resposta obtido com a re-execução de ConsultaAna59.

Se o usuário desejar ver o conjunto-resposta da primeira consulta que ele gerou na sessão corrente, basta selecionar a opção **Ir Para/Início** que a consulta correspondente será re-executada.



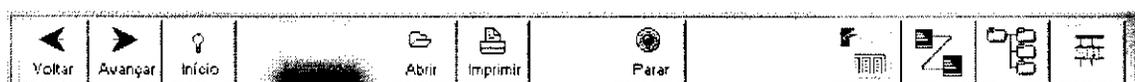
**Fig.IV.4 - Seleção da opção Interface da Barra de Menu**

Embora tenha sido definida *a priori* uma interface como sendo a mais apropriada para o perfil do usuário, existe a possibilidade de que ele queira acessar uma das outras três interfaces, para isso a opção **Interface** é usada. Quando esta opção é escolhida, uma janela é aberta para que o tipo de interface (Básica, Mediana, Avançada ou SQL) seja selecionado.

#### 4.1.2 Barra de Ícones

A **Barra de Ícones** é usada para facilitar a seleção das opções oferecidas pela interface. Ao invés de selecioná-las na Barra de Menu, entrando nos *sub-menus*, os ícones são um meio mais rápido e fácil para tal.

A seguir são descritos todos os ícones que compõem a Barra de Ícones.



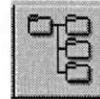
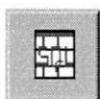
**Fig.IV.5 – Barra de Ícones**

- ⇒ **Voltar:** Retorna à consulta imediatamente anterior a que está sendo exibida.
- ⇒ **Avançar:** Vai para a consulta imediatamente posterior a que está sendo exibida.
- ⇒ **Início:** Mostra a primeira consulta que o usuário realizou na sessão corrente.
- ⇒ **Abrir:** Permite o uso do ARGO por outro usuário, neste caso a sessão do usuário corrente é automaticamente finalizada.
- ⇒ **Imprimir:** Imprime o conteúdo da consulta que está sendo exibida na Janela de Conteúdo.

⇒ **Parar**: Interrompe a formulação de uma consulta e limpa todas as janelas da interface.

Os quatro últimos ícones representam as interfaces que o usuário pode acessar. Embora estejam sendo exibidos quatro ícones na Fig.IV.5, na realidade apenas três estarão exibidos por vez. Sendo a exibição destes ícones dependente da interface ativa no momento, ou seja, se for a Básica, os ícones exibidos são Mediana, Avançada e SQL; se for a Mediana, são exibidos os ícones Básica, Avançada e SQL; e assim por diante.

Os ícones que identificam as interfaces são:

 Ícone para interface Básica	 Ícone para interface Mediana
 Ícone para interface Avançada	 Ícone para interface SQL

**Fig.IV.6 – Ícones que identificam as interfaces**

Quando um destes ícones é selecionado, a interface correspondente é exibida. As consultas realizadas numa interface não são visualizadas nas demais, ou seja, somente as consultas geradas na interface em uso, e na sessão corrente, podem ser visualizadas na mesma.

### 4.1.3 Seleção do BD

Na caixa **Seleção do BD** deve ser informado o banco de dados que o usuário deseja consultar. Uma lista dos *alias* dos BDs, previamente definidos no BDE, é apresentada quando a seta da lateral direita desta caixa for selecionada, para que o *alias* do BD desejado possa ser escolhido.

#### 4.1.4 Janela de Esquema

Na **Janela de Esquema** são exibidas as tabelas do BD selecionado (e seus respectivos campos, no caso da interface Avançada). Nesta janela, a representação das tabelas tanto poderá ser através de ícones (a rigor, o desenho de uma tabela encimando o nome da mesma), quanto de forma hierárquica (representada por uma árvore, onde no primeiro nível ficam as tabelas, e no segundo os campos da tabela). A representação depende da interface utilizada. Em qualquer uma das representações a funcionalidade é a mesma, isto é, permitir a seleção das tabelas (e campos) desejados.

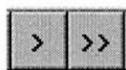
#### 4.1.5 Janela de Conteúdo

Ao ser selecionada uma tabela na Janela de Esquema, os dados para os seus campos são exibidos na **Janela de Conteúdo**. O *default* é a apresentação dos campos relevantes (ou seja, os campos definidos no metabanco de dados como importantes para um usuário), mas o usuário poderá acrescentar ou retirar campos, de acordo com seus objetivos de consulta.

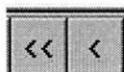
#### 4.1.6 Barra de Botões

Uma **Barra de Botões** é disponibilizada para que as seguintes operações possam ser realizadas sobre os dados da Janela de Conteúdo:

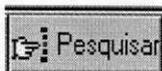
Usado para avançar para o próximo registro, no caso do primeiro botão, ou para o último



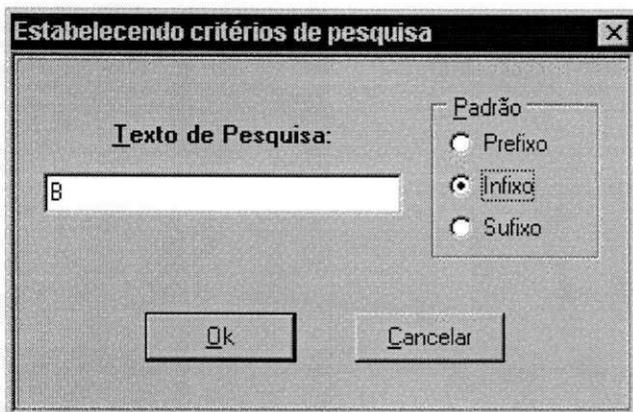
registro apresentado na Janela de Conteúdo, no caso do segundo botão. Quando o cursor já estiver no último registro estes dois botões ficam sem função.



Usado para voltar para o primeiro registro, no caso do primeiro botão, ou para o registro anterior apresentado na Janela de Conteúdo, no caso do segundo botão. Quando o cursor já estiver no primeiro registro estes dois botões ficam sem função.



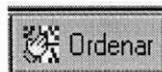
Este botão deve ser usado para selecionar registros na Janela de Conteúdo, a partir de um certo padrão de pesquisa. Para auxiliar a definição do padrão de pesquisa, as opções **prefixo**, **infixo**, **sufixo** são disponibilizadas na janela (Fig.IV.7).



**Fig.IV.7 – Tela para entrar com padrão de pesquisa, ao ser selecionado o botão *Pesquisar***

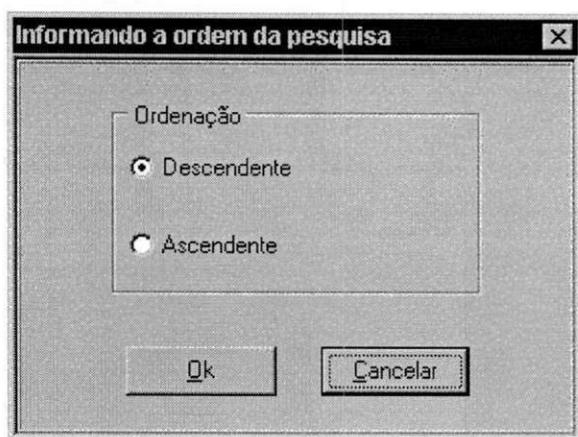
Após ter selecionado um campo na Janela de Conteúdo (a rigor este campo deve ser alfanumérico), o usuário seleciona o botão **Pesquisar** na janela de diálogo que é exibida; entra com o padrão (por exemplo “B”) e seleciona a opção de pesquisa desejada, o *default* sendo a opção **prefixo**. Supondo que foi escolhida a opção *default*, só aparecerão na Janela de Conteúdo, registros cujo conteúdo do campo selecionado começa por “B”.

Mais precisamente, quando se desejar a pesquisa por prefixo, deve-se informar o prefixo desejado e selecionar a opção **Prefixo**; se for por infixo, deve-se informar o infixo desejado e selecionar a opção **Infixo**; se for por sufixo, então informa-se sufixo desejado e seleciona-se a opção **Sufixo** (na Fig.IV.7 é apresentada a janela em que o padrão de pesquisa é informado). Em seguida, escolhe-se Executar ou Cancelar a pesquisa. O resultado da pesquisa executada será exibido na Janela de Conteúdo.



Usado para ordenar os dados na Janela de Conteúdo de acordo com o(s) campo(s) selecionado(s) para ordenação.

Quando o botão **Ordenar** é selecionado, uma janela é aberta contendo duas opções de ordenação (Fig.IV.8), a opção **Ascendente** (para ordenação ascendente) e a opção **Descendente** (para ordenação descendente). Esta janela deixa de ser exibida após a opção desejada ter sido selecionada. O resultado da consulta ordenada é exibido na Janela de Conteúdo.

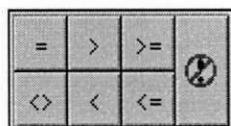


**Fig.IV.8 – Tela para informar o tipo da ordenação desejada**

Antes de pressionar o botão **Ordenar**, o campo desejado para ordenação deve ser selecionado na Janela de Conteúdo. Se a ordenação é por mais de um campo, o procedimento é: escolher o primeiro campo, o botão **Ordenar**, a opção **Ascendente** ou **Descendente**; o mesmo procedimento deve ser realizado para os demais campos desejados. A ordem de ordenação será a mesma em que os campos foram selecionados, ou seja, primeiro campo, segundo campo, etc.

Um campo anteriormente ordenado de forma ascendente pode ser modificado para descendente e vice-versa, basta proceder da forma descrita, ou seja, selecionando o campo, o botão de ordenação e a opção Descendente (ou Ascendente).

#### **4.1.7 Barra de Condições**



A Barra de Condições é usada quando é necessário fazer seleção de registros de uma tabela por valores exatos de alguns dos seus campos.

Em geral, as interfaces só permitem a formulação de consultas usando condições simples (uma condição por vez), com exceção da interface Avançada que usa para tal dois outros botões: **E** e **OU**.

Outra funcionalidade destes botões é o de permitir que junções<sup>9</sup> entre tabelas possam ser realizadas. Para o usuário, não há diferença na utilização destes botões quando o propósito for o de fazer junção, a diferença estando na escolha dos campos (se de uma mesma tabela, ou de tabelas distintas).

Esta barra só é exibida quando o usuário selecionar o campo desejado (seleção com o *mouse* em qualquer lugar na coluna identificada pelo campo, que não seja o cabeçalho) na Janela de Conteúdo, daí o usuário escolhe o operador (=, >, >=, <>, <, <=), e novamente seleciona o valor desejado para estabelecer a condição com o campo selecionado, no caso de estabelecer condições numa mesma tabela. Quando for uma junção, deve ser selecionado o campo desejado na Janela de Referência (ver seção 4.4.1, para maiores detalhes).

Caso o usuário desista de elaborar uma condição, ele deve selecionar o ícone que identifica a opção de cancelamento na Barra de Condições, ou seja, o ícone mais a direita desta barra.

#### **4.1.8 Scrolling**

A Janela de Conteúdo não possui uma largura suficientemente grande para exibir todos os campos ao mesmo tempo, por isso uma barra de *scrolling* horizontal é usada para fazer a rolagem de campos na janela. No exemplo da Fig.IV.9, os campos Cod\_Gravadora e Gravadora não estão visíveis na janela, havendo necessidade de um *scrolling* horizontal para exibi-los.

---

<sup>9</sup> Junção é uma operação da álgebra relacional. Sejam A e B duas tabelas, a junção da tabela A no campo X com a tabela B no campo Y é o conjunto de todas os registros *r* de forma que *r* seja a concatenação de um registro *a* pertencente a A e o registro *b* pertencente a B e o predicado "*a.X theta b.Y*" (onde *theta* é um operador de comparação escalar válido: =, >, <>, >=, <=, etc.) seja avaliado como verdadeiro [Date91].

Da mesma forma, a altura da Janela de Conteúdo não é suficiente para exibir todos os registros para os campos da tabela, neste caso é usado o *scrolling* vertical para rolagem dos registros na Janela de Conteúdo.

## 4.2 Interface Básica

Esta interface é a mais adequada ao usuário *iniciante*, pois não exige o conhecimento da estrutura de um BD. Além de permitir *browsing* nas tabelas, condições de seleção de valores podem ser realizadas, uma vez que propomos um ambiente de consulta visual.

A interface Básica usa a representação icônica/tabular, isto é, ao usuário são disponibilizadas as tabelas do BD selecionado, sendo cada tabela identificada por um ícone. Na realidade, cada tabela tem a mesma representação icônica (que é o desenho de uma tabela) uma vez que, muitas tabelas podendo ser acessadas, tornar-se-ia muito difícil a criação de um ícone personalizado para cada uma. Para fazer a identificação de uma tabela, uma legenda com o nome da mesma sob o ícone é usada (Fig.IV.9).

Ao ser selecionada uma tabela (ou ícone) na Janela de Esquema, os campos relevantes da mesma para o usuário são exibidos, de forma tabular. Havendo campos relevantes que são referências a outras tabelas (chave estrangeira<sup>10</sup>), os campos relevantes das tabelas referenciadas são também exibidos. Este procedimento é recorrente, ou seja, se repete para todas as tabelas referenciadas até que numa certa tabela referenciada não existam campos relevantes que sejam também de referência. O resultado final é a “fusão” de várias tabelas inter-relacionadas.

É importante salientar que as referências são totalmente transparentes aos usuários, uma vez que usuários iniciantes não necessitam ter nenhum conhecimento sobre os relacionamentos entre as tabelas de um BD.

---

<sup>10</sup> Chave estrangeira é um campo, ou uma combinação de campos, numa tabela  $T_2$ , em que cada valor é igual a um valor da chave primária de uma tabela  $T_1$  ( $T_1$  e  $T_2$  não são necessariamente distintas). A chave estrangeira é um *link* básico para outra tabela. A chave primária, por sua vez, é um campo, ou uma combinação de campos de uma tabela  $T$ , onde cada valor identifica de forma única os registros da tabela.

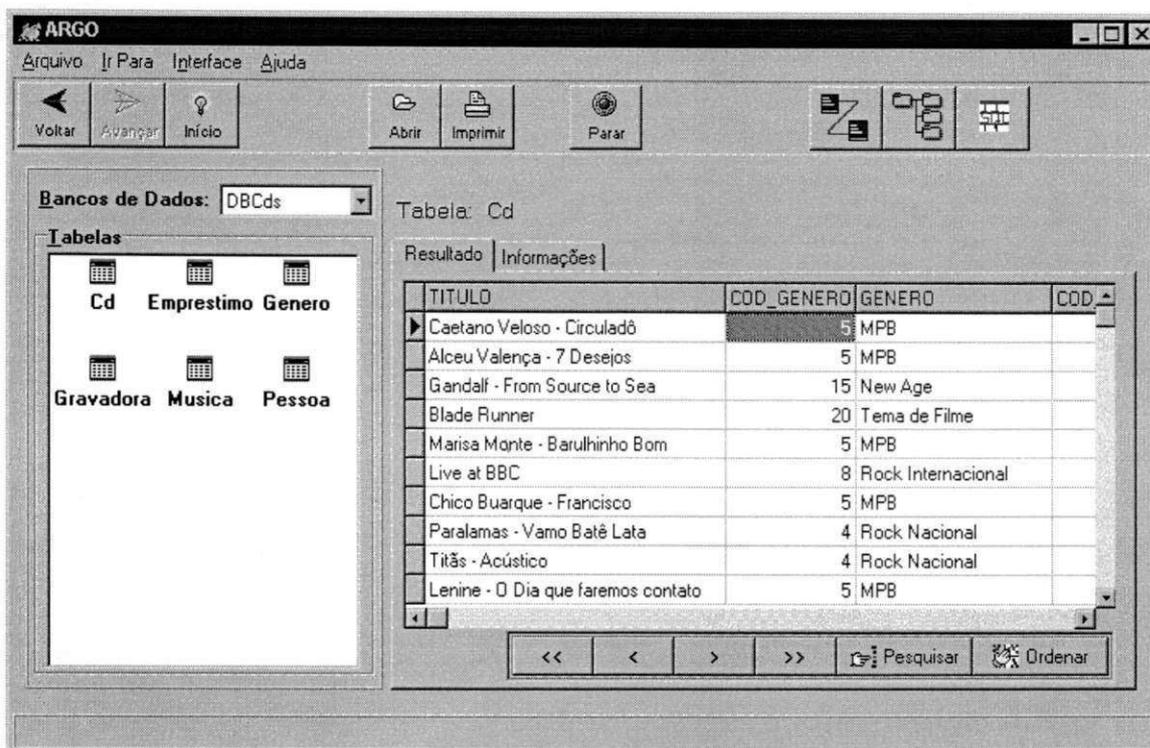
Duas situações limites podem ser detectadas no resultado exibido na Janela de Conteúdo. Uma delas é quando todos os campos que são chaves estrangeiras (em todas as tabelas) tiverem sido definidos como relevantes para um usuário. Em consequência, o resultado obtido pode se aproximar da “fusão” de todas as tabelas do BD. Como se pode perceber, a tabela resultante pode ser muito grande, exigindo, em termos de visualização na tela, que o usuário faça inúmeros *scrollings* horizontais, podendo cansá-lo.

Este problema pode ser contornado se as visões dos usuários do BD estiverem bem definidas. Como se sabe, usuários podem ter visões diferentes de um mesmo BD, ou seja, nem todas as informações são importantes ou permitidas para todos os usuários. Para um mesmo BD existem inúmeras “janelas”, sendo cada uma delas atribuída a certos usuários. Estas “janelas” devem ser muito menores que o BD.

A outra situação limite é quando, para um certo usuário, todos os campos que são chaves estrangeiras não lhe são relevantes. Neste caso, jamais haverá a “fusão” entre as tabelas. Em outras palavras, apenas os campos relevantes da tabela selecionada serão exibidos.

Usamos como exemplo, neste capítulo e nos subseqüentes, uma pequena aplicação que controla os empréstimos e devoluções de CDs de uma CDteca particular, denominado DBCds. Nesta aplicação, estão cadastrados todos os Cds de um proprietário, assim como os gêneros musicais, as gravadoras, as músicas, os cantores e os compositores de cada Cd. Para cada Cd emprestado, informações do tipo: pessoa a quem foi feito o empréstimo, datas (empréstimo e devolução), dentre outras, são registradas.

Como ilustrado na Fig.IV.9, supondo que o usuário que está acessando o ARGO seja Ana (o usuário Ana está cadastrado como sendo do tipo experiente, mas não existe nenhuma restrição quanto ao mesmo poder usar quaisquer das interfaces do ARGO), ao ser selecionada a tabela CD, com base no metabanco definido para Ana, para o esquema do banco de dados DBCds, os campos exibidos da tabela CD na Janela de Conteúdo são aqueles relevantes a Ana. Como a tabela CD faz referências às tabelas GRAVADORA (via campo Cod\_Gravadora) e GENERO (via campo Cod\_Genero), e sendo ambos os campos relevantes a Ana, os campos relevantes de GRAVADORA e GENERO são exibidos também.



**Fig-IV.9 - Interface Básica**

Em resumo, quando Ana selecionar a tabela CD os campos que serão disponibilizados na Janela de Conteúdo são: Título, Cod\_Genero, Cod\_Gravadora da tabela CD, Genero da tabela GENERO e Gravadora da tabela GRAVADORA.

A seguir apresentamos alguns exemplos do uso da interface Básica.

#### 4.2.1 Exemplos de utilização da Interface

Antes de apresentarmos os exemplos, vamos abrir um parêntese para explicar os principais eventos e seus significados.

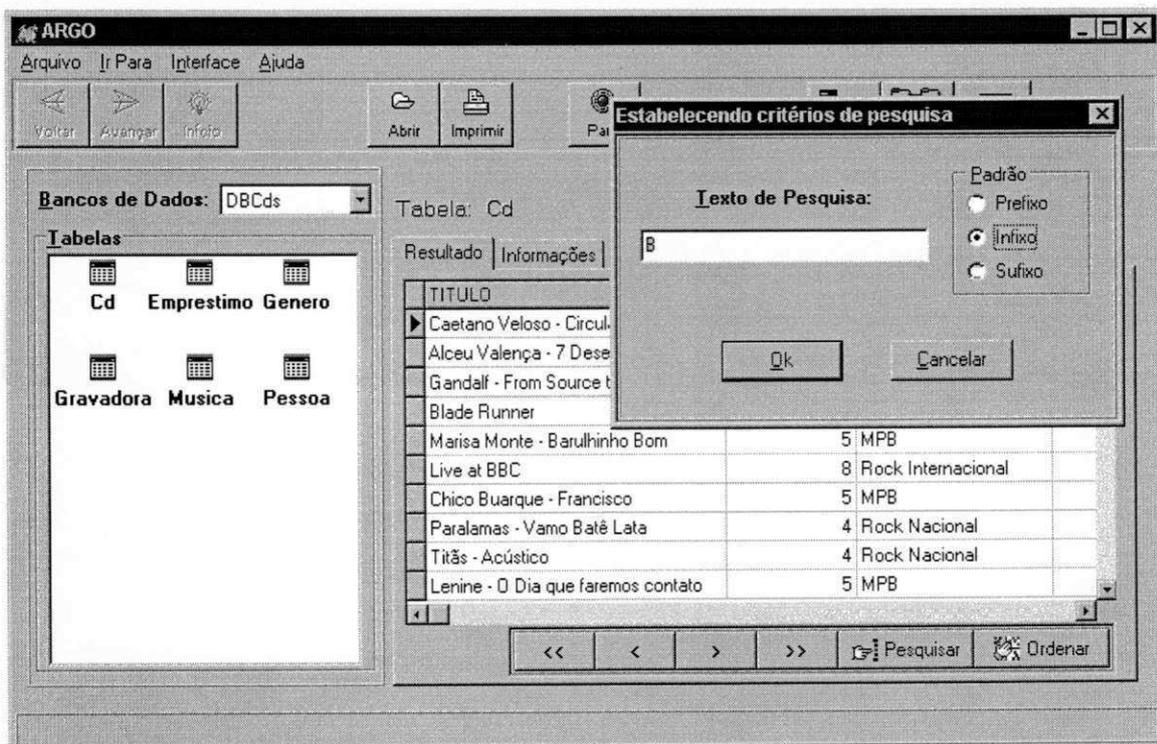
- O evento **duplo “clie”** (seleção com o botão esquerdo do *mouse*) no valor de um campo da Janela de Conteúdo: seleção de um campo ou do valor de um campo. Se o evento ocorreu antes da Barra de Condições ter sido exibida, então um campo foi selecionado; se depois, então o valor de um campo foi selecionado;
- O evento **único “clie”** no conteúdo de um campo da Janela de Conteúdo significa apenas o posicionamento no valor de um campo;

- Qualquer botão e ícone, é selecionado através de um **único “clique”**.

Fechamos aqui o parêntese.

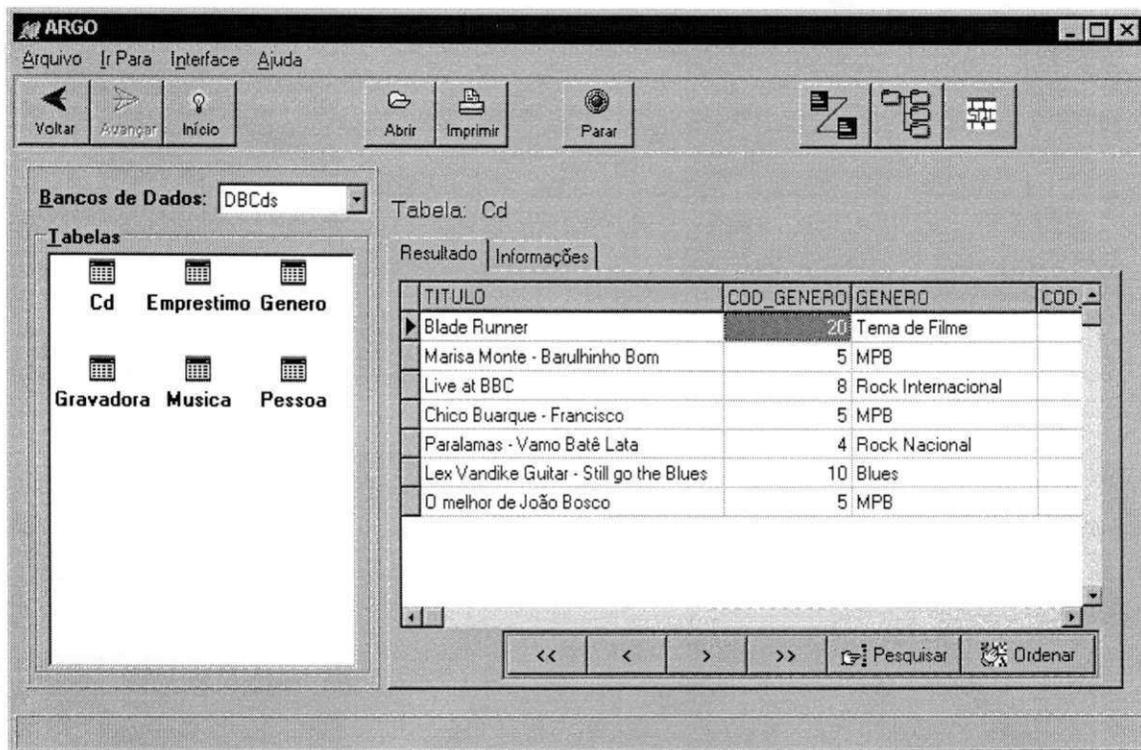
**Exemplo 1:** A partir da Fig.IV.9, obter todos os Cds que contenham no seu título a letra “B”. Os seguintes procedimentos devem ser realizados:

- ↳ Fazer duplo “clique” no valor qualquer do campo Título;
- ↳ Na Barra de Botões selecionar o botão Pesquisar;
- ↳ Digitar “B” no campo da janela que é exibida;
- ↳ Selecionar a opção Infixo, para indicar que o “B” é um infixo, ou seja, deseja-se obter todos os registros que contenham a letra “B” como parte do nome do Título (seja no início, meio, ou fim) (Fig.IV.10)



**Fig.IV.10 - Pesquisa os Cds que possuem no seu título a letra “B”**

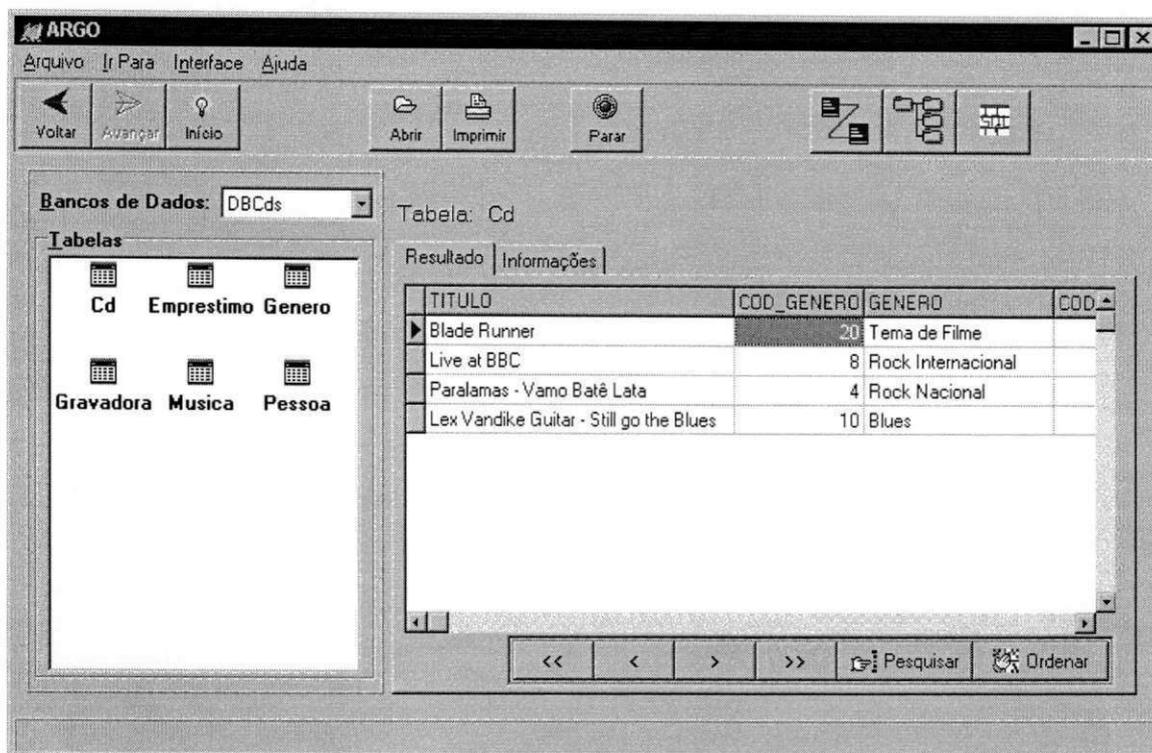
- ↳ Selecionar o botão Executar para obtenção do resultado ( Fig.IV.11)



**Fig.IV.11 - Resultado da consulta baseada numa busca por padrão**

**Exemplo 2:** A partir do resultado obtido na Fig.IV.11, selecionar os Cds cujo Cod\_Genero seja diferente de 5.

- ↳ Fazer duplo “clic” num valor qualquer do campo Cod\_Genero (que não seja no cabeçalho)
- ↳ Na Barra de Condições que é exibida, selecionar o operador “<>” (diferente)
- ↳ Novamente, fazer duplo “clic”, só que desta vez no valor desejado para o campo Cod\_Genero, no caso o valor 5 (Fig.IV.12)



**Fig.IV.12 – Resultado da consulta baseada na seleção de valores exatos**

#### 4.2.2 Análise da interface Básica

Esta interface, como já foi dito, se destina a usuários iniciantes. Ela é simples e de fácil manipulação. Apenas selecionando a tabela desejada, o resultado é uma consulta envolvendo a mesma (exibindo seus campos relevantes), como também os campos relevantes das tabelas que com ela se relacionam, tudo isto de uma forma totalmente transparente ao usuário.

Dentre os operadores definidos para a álgebra relacional, os que são permitidos nesta interface são: Seleção e Junção (este gerado implicitamente).

É importante salientar que a operação de Seleção nesta interface não é realizada na sua totalidade, nela não é permitida a realização de mais de uma condição por vez. Dito de um outro modo, não é possível formular consultas envolvendo seleções múltiplas na forma conjuntiva (E) explicitamente, e a forma disjuntiva (OU) não é suportada.

Uma forma de contornar, em parte, a restrição da operação Seleção, é realizar as consultas a partir de outras já existentes: assim, uma conjunção implícita acaba sendo gerada.

A operação de Projeção não é permitida. O usuário não tem poder para escolher qual campo, da tabela selecionada, ele deseja visualizar. Isto porque a interface não disponibiliza o esquema do BD por completo na sua Janela de Esquema, mas apenas as tabelas.

### 4.3 Interface Mediana

Nesta interface o usuário continua não precisando entender a estrutura do BD, só que agora ele poderá “navegar” pelos *links* das tabelas, ou seja ir de uma tabela para outra apenas selecionando o campo que proporcionar tal ligação.

A informação acessada na Web é estruturada em páginas de documentos HTML (*HyperText Markup Language*) e sua “navegação” se dá através de recursos de hipertexto<sup>11</sup>. Quando o usuário acessa uma página, um documento hipertextual, podendo apresentar recursos gráficos, animação, etc., é exibido. A partir deste documento vários outros documentos relacionados, ou *links*, levam os usuários a “navegar” de uma página para outra.

A idéia da interface Mediana é fornecer um ambiente semelhante ao da Web, no que se refere à “navegação”, sendo neste caso uma “navegação” realizada através das tabelas de um BD. Ao usuário é exigido apenas o reconhecimento de quais *links* proporcionam tal “navegação” nas tabelas de um BD.

Fazendo um paralelo entre os *browsers* Web e a interface Mediana, uma página na Web é equivalente a uma tabela de um BD, enquanto que cada chave estrangeira da tabela de um BD é equivalente a um *link* numa página na Web. A “navegação”, portanto, é realizada quando uma chave estrangeira (*link*), for selecionada por um usuário.

---

<sup>11</sup> Hipertexto oferece uma forma de movimentação de um documento para outro através de *links* de palavras.

Ao usuário não é exigido nenhum conhecimento maior sobre os campos *links*, ou seja, eles não precisam saber que este é um campo chave estrangeira de uma outra tabela: o sistema, internamente, se encarrega de tal nível de detalhamento.

### 4.3.1 Apresentação das Janelas e seus Conteúdos

Nesta interface até duas tabelas podem ser exibidas ao mesmo tempo na tela, permitindo uma visualização simultânea das mesmas, logo duas Janelas de Conteúdos são disponibilizadas.

Quando uma tabela é selecionada na Janela de Esquema, sendo esta a primeira tabela selecionada, os dados dos campos relevantes e dos campos que são chaves estrangeiras são exibidos na Janela de Conteúdo da esquerda. Na Fig.IV.13 a tabela CD foi selecionada e seus dados exibidos na Janela de Conteúdo da esquerda.

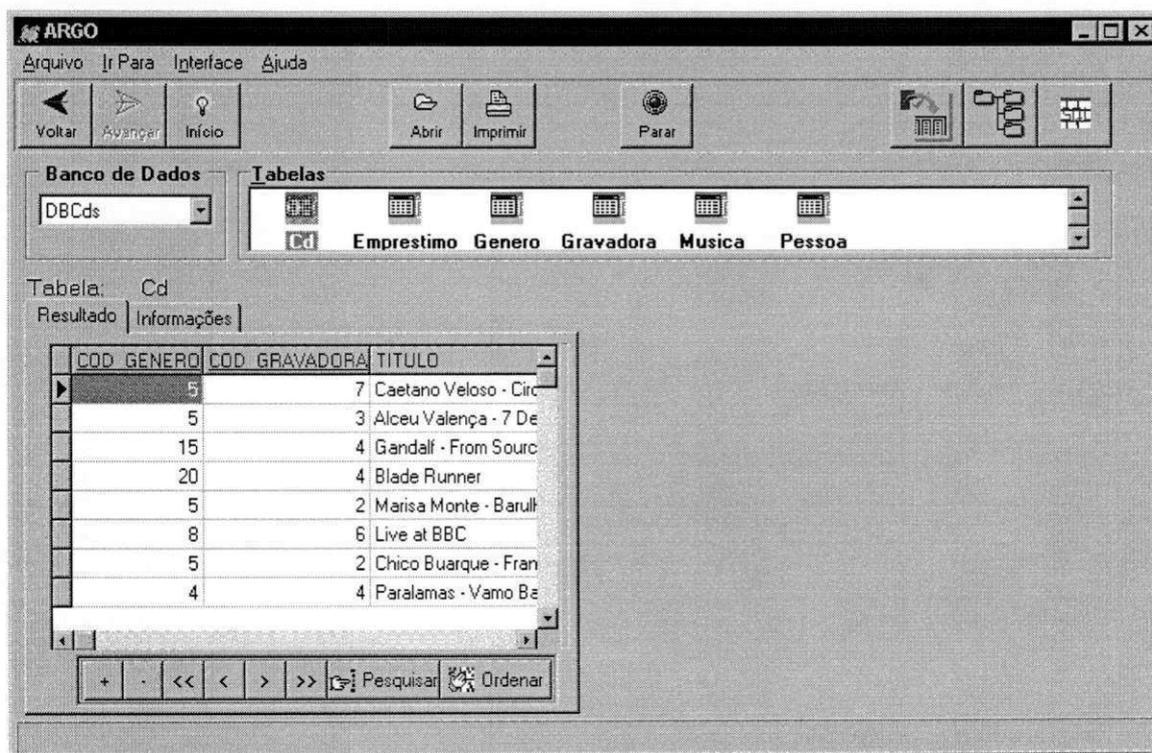


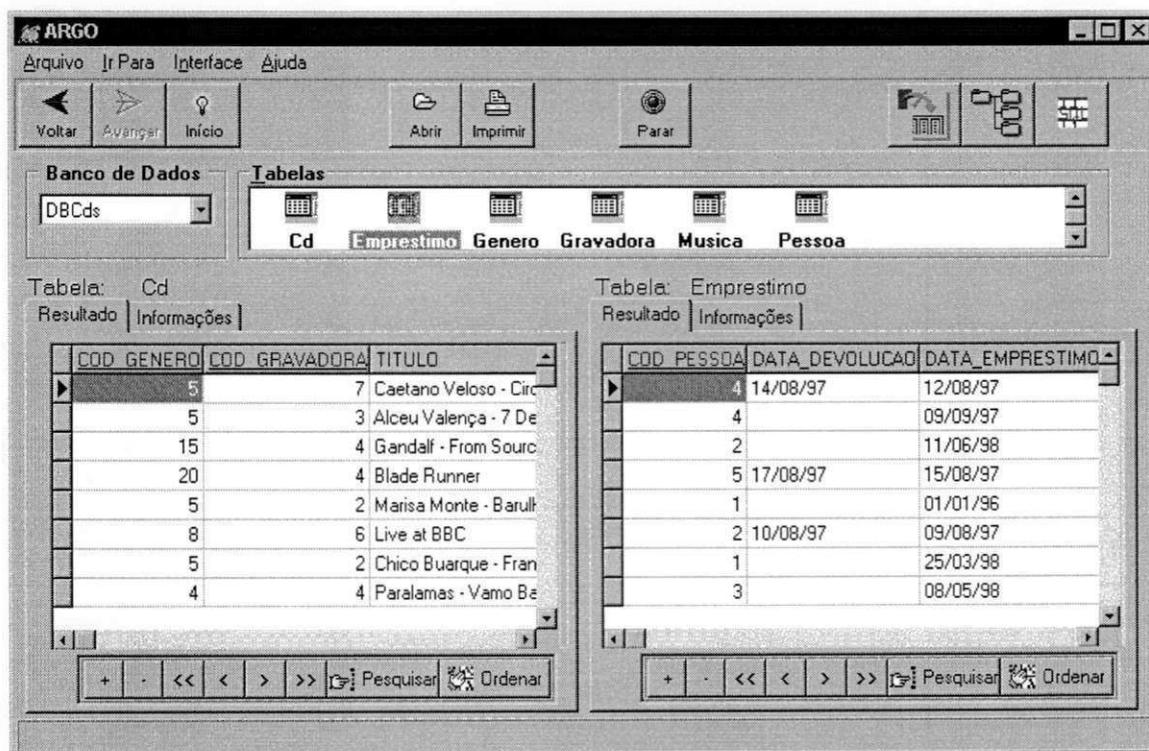
Fig.IV.13 - Seleção da tabela CD na interface Mediana

Se uma outra tabela for selecionada na Janela de Esquema, a segunda Janela de Conteúdo será exibida e nela os dados dos campos relevantes e chaves estrangeiras da tabela selecionada (ver Fig.IV.14, onde a tabela Empréstimo também foi selecionada).

A partir de então, cada nova tabela selecionada na Janela de Esquema terá seus dados exibidos em uma das duas Janelas de Conteúdo. Após a tabela ter sido selecionada, uma janela de diálogo é aberta para que o usuário informe sua preferência sobre a Janela de Conteúdo em que deseja ver os dados da tabela.

Uma identificação (nome) referente à tabela que está sendo exibida em cada Janela de Conteúdo, é apresentada sobre a janela que lhe é correspondente (Fig.IV.14).

Os campos de uma tabela que são chaves estrangeiras são sublinhados, para indicar um *link* e permite que o usuário, ao selecionar este campo (com um único “clique” do *mouse*), “navegue” da tabela atual para a tabela referenciada pelo *link*. Neste caso, a tabela selecionada via *link* tem seus dados exibidos na mesma Janela de Conteúdo em que estavam sendo exibidos os dados da tabela em que o *link* foi selecionado.

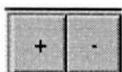


**Fig.IV.14 - Seleção da tabela Empréstimo na interface Mediana**

As tabelas exibidas nas Janelas de Conteúdo, simultaneamente, não precisam ser relacionadas. Por exemplo, um usuário pode selecionar as tabelas GÊNERO e GRAVADORA, sendo elas exibidas, cada uma, numa das Janelas de Conteúdo, embora não possuam relacionamento entre si.

### 4.3.2 Projeção de campos

Para que a projeção de campos (incluir ou retirar da Janela de Conteúdo um determinado campo) seja feita, dois botões a mais foram acrescentados à Barra de Botões desta interface:



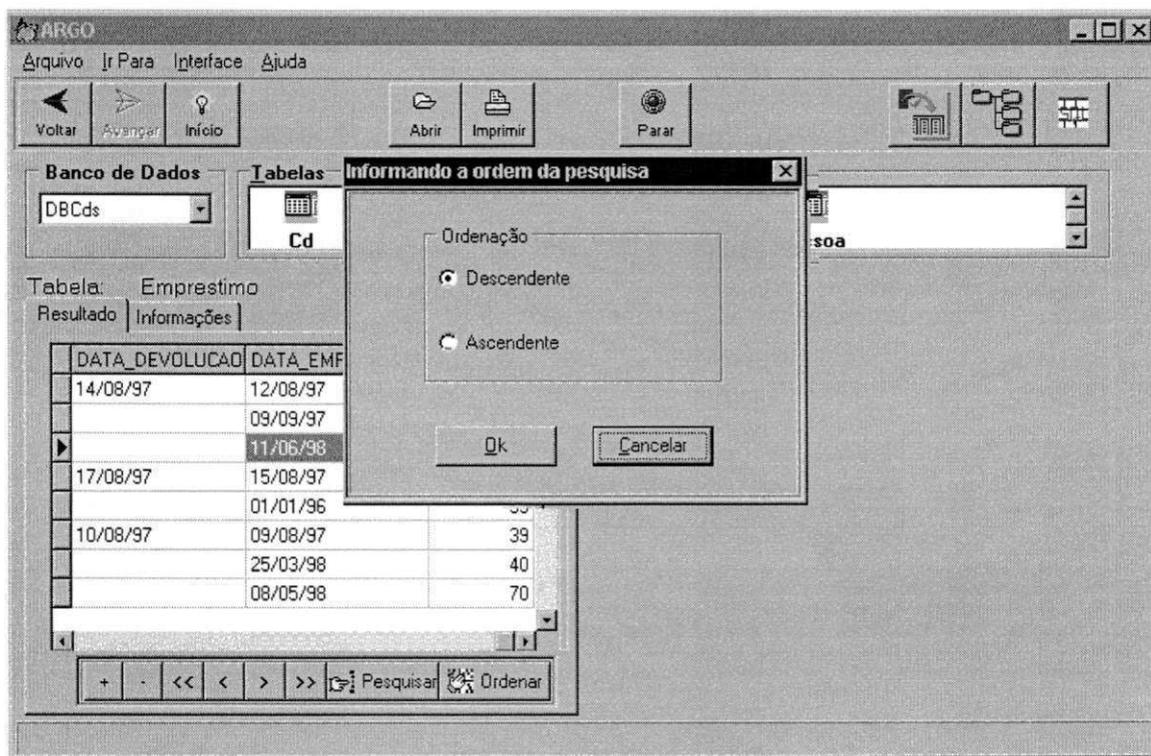
Quando um dos dois botões é selecionado, uma janela de diálogo é apresentada. Nesta janela são exibidos os campos da tabela selecionada para que possam ser escolhidos para inclusão (no caso do primeiro botão) ou exclusão (no caso do segundo botão) os campos desejados. Na Fig.IV.19, é exibida tal janela de projeção.

Para mostrar as funcionalidades da interface Mediana, vamos apresentar na subseção seguinte alguns exemplos ilustrativos

### 4.3.3 Exemplos da interface Mediana

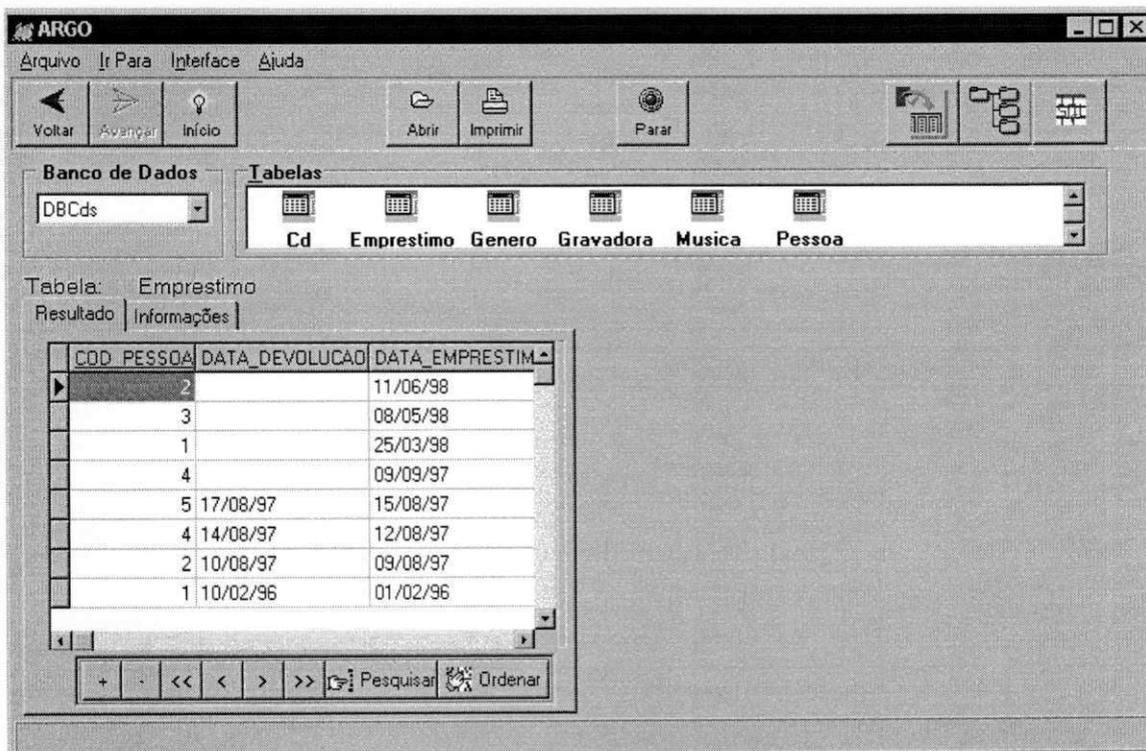
**Exemplo 1:** Selecionar a tabela Empréstimo, e ordenar seus registros por data de empréstimo (ordenação descendente).

- ↳ Selecionar a tabela Empréstimo na Janela de Esquema, com isto os dados são exibidos na Janela de Conteúdo (janela à esquerda)
- ↳ Fazer duplo “clic” em um valor qualquer do campo Data\_Empréstimo
- ↳ Selecionar o botão Ordenar (na Fig.IV.15, é apresentado o resultado obtido)
- ↳ Selecionar a opção desejada para ordenação, no caso a opção descendente e pressionar o botão OK



**Fig.IV.15 – Seleção do campo Data\_Emprestimo para ordenação**

A consulta é processada e o resultado exibido na Janela de Conteúdo (Fig.IV.16).



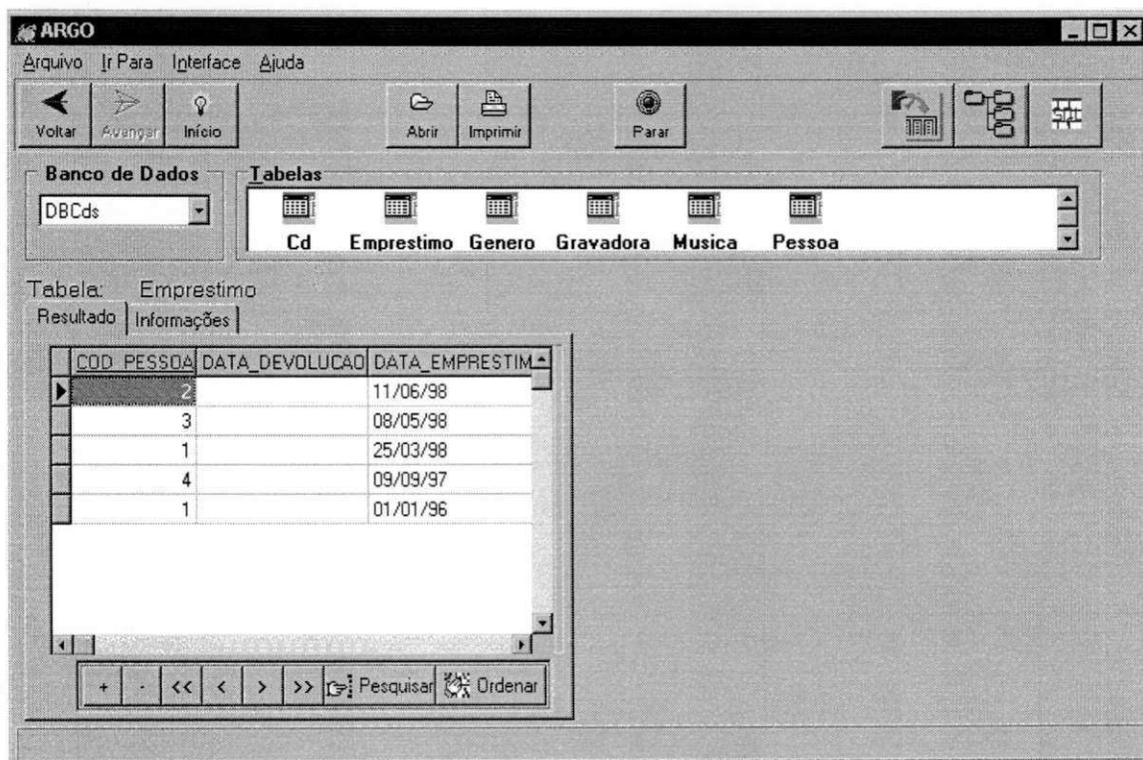
**Fig.IV.16 – Resultado: tabela Empréstimo ordenada por Data\_Empréstimo**

A ordenação pode ser feita por tantos campos quantos o usuário queira, basta selecionar cada campo, individualmente, seguido do tipo de ordenação desejada (Ascendente ou Descendente), como já explicado anteriormente na seção 4.1.6.

**Exemplo 2:** Tomemos o resultado apresentado na Fig.IV.16, a partir dele obter os CDs que estão emprestados.

- ↳ Fazer um duplo “clic” num valor do campo Data\_Devolução
- ↳ Na Barra de Condições apresentada, selecionar o operador de condição “=”
- ↳ Fazer um duplo “clic” no valor desejado para o campo selecionado, ou seja no valor em branco (nulo) para o campo Data\_Devolução

O resultado é em seguida apresentado na mesma Janela de Conteúdo (Fig.IV.17).



**Fig.IV.17 - Resultado da consulta à tabela Empréstimo para obter os CDs emprestados e não devolvidos**

#### **4.3.4 Análise da Interface Mediana**

O diferencial desta interface com relação às demais definidas neste trabalho, é o uso da abordagem de *links*, proporcionando um ambiente de “navegação” entre tabelas de um BD, além de possibilitar a visualização concomitante de duas tabelas, já que possui duas Janelas de Conteúdo.

Assim como a interface Básica, esta também não permite a Seleção de valores com condições múltiplas (nas formas conjuntivas e/ou disjuntivas). A Projeção de campos das tabelas é permitida, diferentemente da interface Básica, na Mediana o usuário tem poder para decidir quais campos deseja visualizar na sua consulta. Qualquer campo da tabela selecionada poderá ser incluído ou excluído da consulta apresentada na Janela de Conteúdo em que for selecionado o respectivo botão de projeção .

A interface preestabelece que os campos relevantes incluem as chaves estrangeiras (mesmo que estas não tenham sido definidas explicitamente como relevantes no metabanco de dados), sendo portanto obrigatoriamente exibidas por representarem os *links* entre as tabelas.

A operação junção não é permitida nesta interface, sob nenhum aspecto (implícita ou explicitamente). Mesmo a interface Mediana não permitindo junções ela tem no geral mais funcionalidades que Básica, tais como “navegação” via *links* e Projeção de campos.

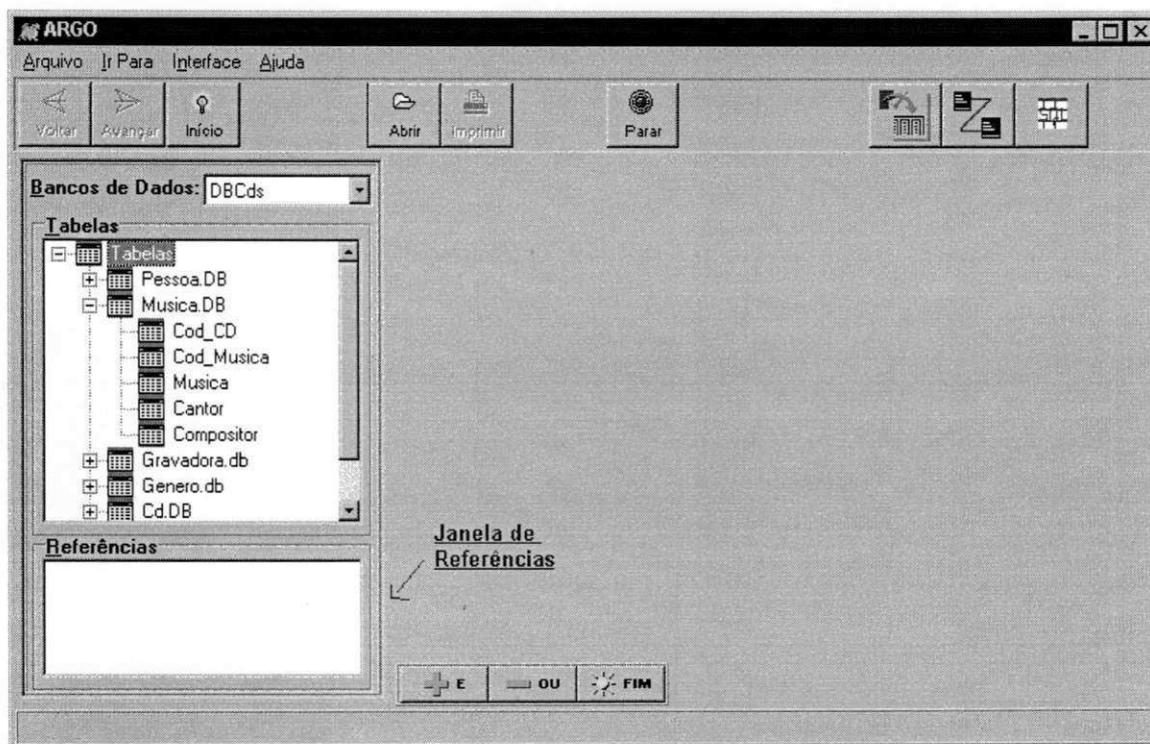
#### **4.4 Interface Avançada**

Esta interface é apropriada para usuários de nível mais avançado que o iniciante e o mediano, no sentido de que eles detêm maiores conhecimentos sobre um BD (conteúdo das tabelas e seus relacionamentos).

Ao contrário das interfaces anteriores, nesta as tabelas referenciadas (aquelas em que a chave primária é equivalente a uma das chaves estrangeiras da tabela selecionada na Janela de Esquema) são explicitamente apresentadas na Janela de Referências (denominação usada para a janela em que as tabelas referenciadas são exibidas). O usuário pode elaborar consultas mais complexas fazendo uso de junções explícitas e

múltiplas comparações entre campos, enriquecendo de forma considerável o poder de expressão da interface.

Na Fig.IV.18 são apresentados alguns aspectos desta interface. Quando um BD é selecionado, o seu esquema é exibido na Janela de Esquema de forma hierárquica, ou seja, em um nível da hierarquia são exibidas as tabelas, em outro nível os campos de cada tabela. A Janela de Referências é exibida, mas a princípio estará vazia, até que uma tabela seja selecionada na Janela de Esquema.



**Fig.IV.18 – Aspectos iniciais da interface Avançada**

As tabelas da Janela de Esquema e da Janela de Referências, só são exibidas quando o usuário selecionar o ícone [+] que aparece ao lado do rótulo **Tabelas** da Janela de Esquema e do rótulo **Referências** na Janela de Referências (no caso de uma tabela ter sido selecionada na Janela de Esquema, caso contrário ela estará vazia). O mesmo ocorre com os campos das tais tabelas, que só são exibidos na hierarquia quando o ícone [+], ao lado do nome da tabela, é selecionado pelo usuário. O procedimento inverso, ou seja, a não visualização de tais tabelas ou campos, se dá quando o ícone [-] (substituto do ícone [+] quando as tabelas ou campos estão exibidos) é selecionado.

#### 4.4.1 Janela de Referências

As tabelas referenciadas pela tabela selecionada na Janela de Esquema são exibidas na **Janela de Referências**. Além de possibilitar um melhor conhecimento e visualização do esquema do BD, esta janela viabiliza o processo de junção entre as tabelas do BD, tanto implícita quanto explicitamente (detalhes na subseção 4.4.4).

Ao ser selecionada uma tabela na Janela de Esquema, a Janela de Conteúdo é exibida e nela são apresentados todos os campos relevantes para o usuário (ver Fig.IV.19, onde a tabela CD foi selecionada). Neste momento, a Janela de Referências é preenchida com as tabelas referenciadas pela tabela selecionada.

Ao ser selecionado um campo de uma tabela referenciada, uma junção implícita é realizada e o campo selecionado passa a fazer parte da consulta exibida na Janela de Conteúdo. Quando, ao invés de um campo, é selecionada uma tabela na Janela de Referências, as tabelas que são referenciadas por esta tabela são exibidas, também, na hierarquia de tabelas da Janela de Referências. Isto para viabilizar que múltiplas junções, incluindo as tabelas de referência das tabelas referenciadas pela tabela selecionada na Janela de Esquema, sejam realizadas.

Sejam as tabelas A, B, C, e D. Suponha que A referencia B, B referencia C, e B referencia D. Quando A for selecionada na Janela de Esquema, B é exibida na Janela de Referências. Quando B é selecionada na Janela de Referências, as tabelas que são referenciadas por B, no caso C e D, também são exibidas na Janela de Referências, para permitir que uma junção entre A, B, C e D possa ser realizada numa mesma consulta, e assim sucessivamente. Após o botão FIM ter sido pressionado, a Janela de Referências passa a conter apenas as tabelas referenciadas pela tabela selecionada na Janela de Esquema.

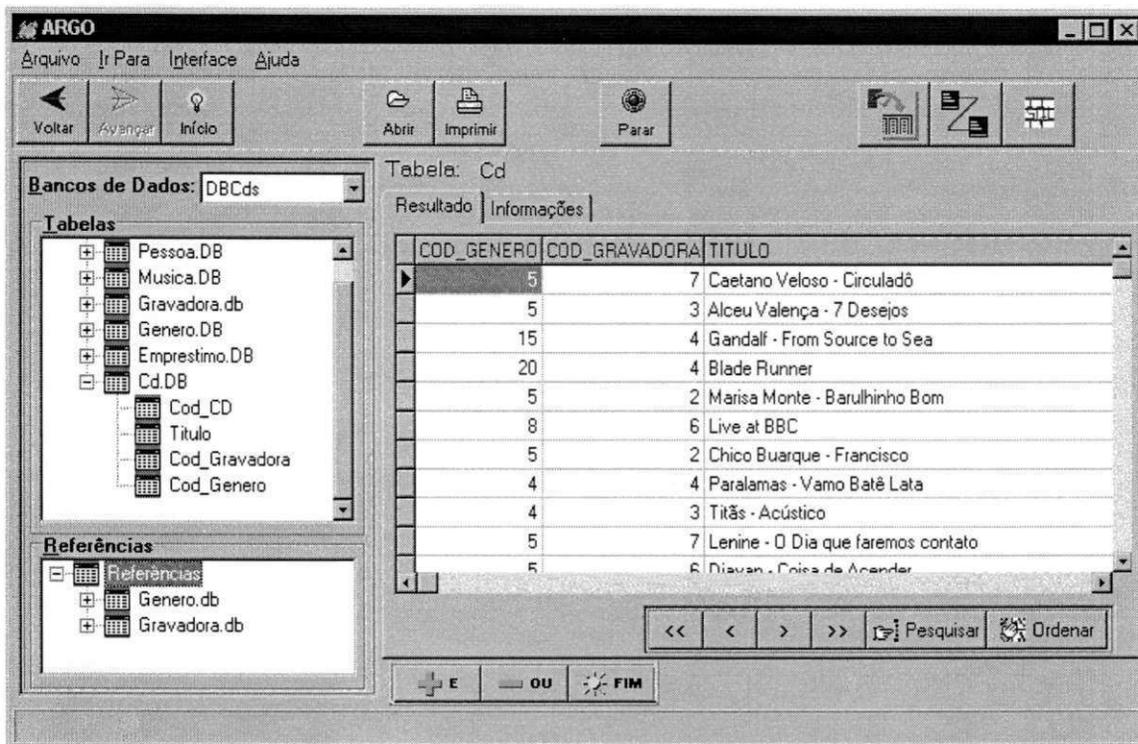


Fig.IV.19 – Resultado da seleção da tabela CD na Janela de Esquema

#### 4.4.2 Projeção de campos

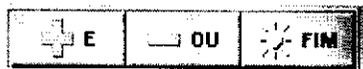
Caso o usuário deseje acrescentar ou retirar algum campo da Janela de Conteúdo, basta selecionar o campo desejado da tabela selecionada na Janela de Esquema ou da tabela referenciada na Janela de Referências. Se este campo ainda não estiver sendo exibido na consulta, automaticamente ele é adicionado à Janela de Conteúdo, caso contrário, ele é retirado.

#### 4.4.3 Uso do E e do OU

A interface Avançada permite a formulação de consultas mais complexas, do tipo em que várias condições devem ser impostas, usando para isso operadores de condições **E** (forma conjuntiva) e **OU** (forma disjuntiva).

Para viabilizar o uso destes operadores de condição, uma nova barra de botões foi incluída nesta interface. É nela que estão contidos os botões referentes aos operadores **E** e

**OU.** Um botão identificador de que o resultado de uma consulta (conjuntiva e/ou disjuntiva) deve ser exibido é também utilizado, ou seja, o botão **FIM**.



O uso dos operadores E e OU é importante quando o usuário desejar selecionar uma faixa de valores, atribuindo mais de uma condição nesta seleção. Também são usados quando for preciso estabelecer múltiplas condições entre tabelas diferentes, caracterizando neste caso junções múltiplas.

Com o intuito de viabilizar a elaboração de consultas envolvendo múltiplas junções explícitas, ou a seleção de faixa de valores, foi definido o botão **FIM**. Se o resultado da consulta fosse exibido de imediato na tela (sem requerer o uso do botão FIM), não seria possível continuar a elaboração de uma consulta deste tipo, uma vez que os dados necessários não mais estariam exibidos na Janela de Conteúdo.

Durante a elaboração de uma consulta envolvendo o uso explícito dos operadores E ou OU, a abordagem de consulta dinâmica não se enquadra: ao contrário, nenhum dado é alterado na Janela de Conteúdo até que o botão FIM seja executado, e só então o resultado da consulta é exibido. Veja o exemplo a seguir:

**Exemplo 1:** Selecionar todos os CDs que sejam do gênero “MPB” cuja gravadora seja “BMG”, ou quaisquer CDs do gênero “Rock Internacional”:

- ↳ Selecionar a tabela CD na Janela de Esquema
- ↳ Na Janela de Referências selecionar o campo Genero da tabela GENERO e o campo Gravadora da tabela GRAVADORA, para que eles sejam acrescentados à Janela de Conteúdo
- ↳ Fazer duplo “clic” num valor qualquer do campo Genero na Janela de Conteúdo
- ↳ Na Barra de Condições que aparece selecionar o operador “=”
- ↳ Fazer duplo “clic” no valor “MPB” do campo Genero

Com isto, a primeira condição foi estabelecida (Genero = "MPB"). A segunda e terceira condições (Gravadora = "BMG", Genero = "Rock Internacional") devem ser estabelecidas, como segue:

- ↳ Selecionar o botão **E**
- ↳ Fazer duplo "clíc" num valor qualquer do campo Gravadora (exceto no cabeçalho) na Janela de Conteúdo
- ↳ Na Barra de Condições que aparece selecionar o operador "="
- ↳ Fazer duplo "clíc" no valor "BMG" do campo Gravadora
- ↳ Selecionar o botão **OU**
- ↳ Fazer duplo "clíc" num valor qualquer do campo Genero (exceto cabeçlho) na Janela de Conteúdo
- ↳ Na Barra de Condições que aparece selecionar o operador "="
- ↳ Fazer duplo "clíc" no valor "Rock Internacional" do campo Genero
- ↳ Selecionar o botão FIM, para indicar fim da preparação da consulta

Após o botão FIM ter sido pressionado, a consulta é então gerada e seu resultado é apresentado na Janela de Conteúdo (Fig.IV.20).

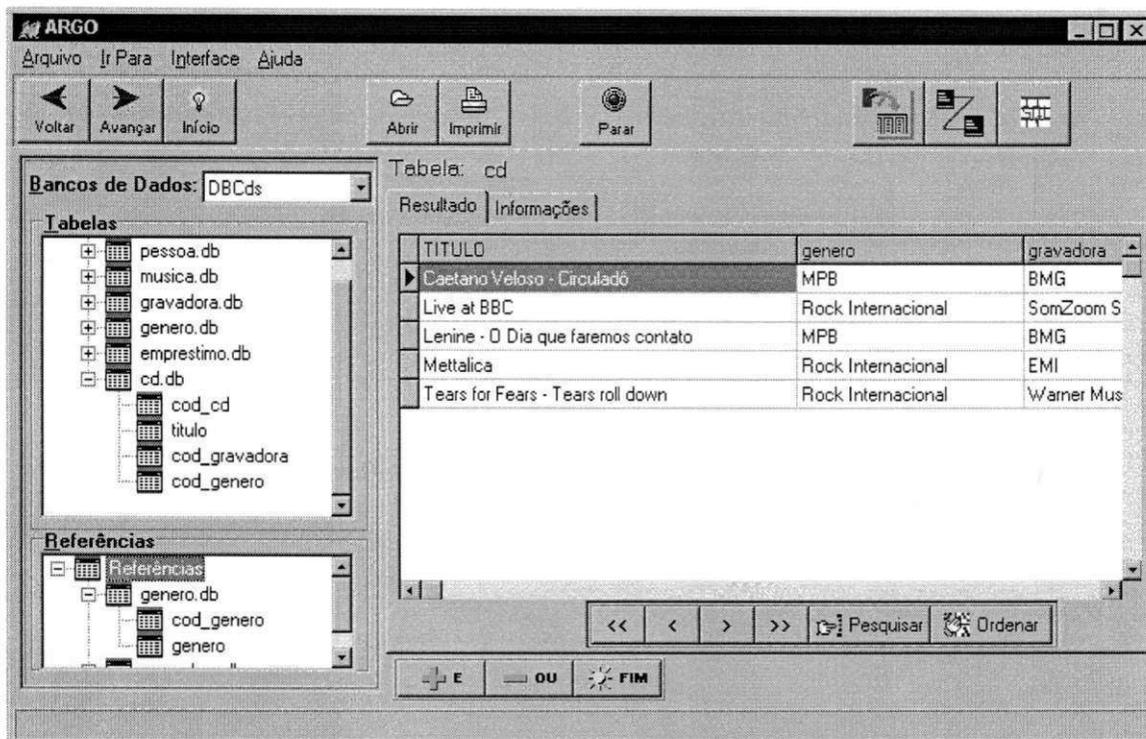


Fig.IV.20 - Resultado da consulta usando E e OU

Caso o usuário desista da consulta (antes de ter pressionado o botão FIM), seja porque errou na elaboração (selecionando um campo/valor indevido, por exemplo) ou simplesmente por não querer mais aquela consulta, ele deve pressionar o ícone **Parar** que fica na Barra de Ícones. Neste caso a Janela de Conteúdo fica sem dados, e na Janela de Esquema nenhuma tabela vai estar selecionada.

#### 4.4.4 Junções explícitas e implícitas

O uso de junção implícita ou explícita é permitido, para isto a Barra de Condições deve ser usada, assim como os botões E, OU e FIM (quando forem múltiplas junções explícitas).

Retornando ao Exemplo 1, podemos perceber que duas junções foram realizadas, embora o usuário não as tenha especificado explicitamente, ou seja, ao selecionar o campo Genero da tabela GENERO e o campo Gravadora da tabela GRAVADORA na Janela de Referências, junções implícitas foram realizadas.

Toda junção implícita é uma equijunção (sempre gera uma junção onde o operador de igualdade é utilizado), e também é sempre conjuntiva, enquanto que uma junção explícita é uma *theta*-junção, ou seja, pode gerar uma junção envolvendo qualquer um dos operadores de comparação (=, >, <, >=, <=, <>), podendo ser conjuntiva ou disjuntiva.

Para que uma junção simples (entre duas tabelas apenas) seja estabelecida, nenhum esforço maior é exigido do usuário, e o procedimento usado pode ser tanto uma junção implícita quanto uma junção explícita, a escolha ficando a critério do usuário.

As junções múltiplas (envolvendo várias tabelas), por sua vez, sempre serão realizadas explicitamente. Neste caso, é preciso usar a Barra de Condições (para selecionar o operador desejado), e os botões **E** ou **OU**, tantas vezes quantas forem o número de junções desejadas, até que o usuário deseje ver o resultado, quando então, o botão FIM é selecionado.

Para ilustrar a elaboração de uma consulta usando junção explícita, veja o exemplo a seguir.

**Exemplo 2:** Obter os CDs que estão emprestados.

↳ Selecionar a tabela EMPRESTIMO (Fig.IV.21)

Como podemos perceber, para este usuário, somente os campos Cod\_Pessoa, Data\_Emprestimo e Data\_Devolucao foram definidos como relevantes. Mas, como pretendemos saber quais são os Cds emprestados (não o seu código), é necessário adicionar o campo Título do Cd à Janela de Conteúdo:

↳ Selecione o campo Título na Janela de Referências para a tabela CD

↳ Quando este campo é selecionado, uma junção (equijunção) é implicitamente realizada entre as tabelas EMPRESTIMO e CD. Portanto, o resultado é uma consulta coerente: os títulos dos respectivos Cds da tabela de EMPRESTIMO (Fig.IV.22).

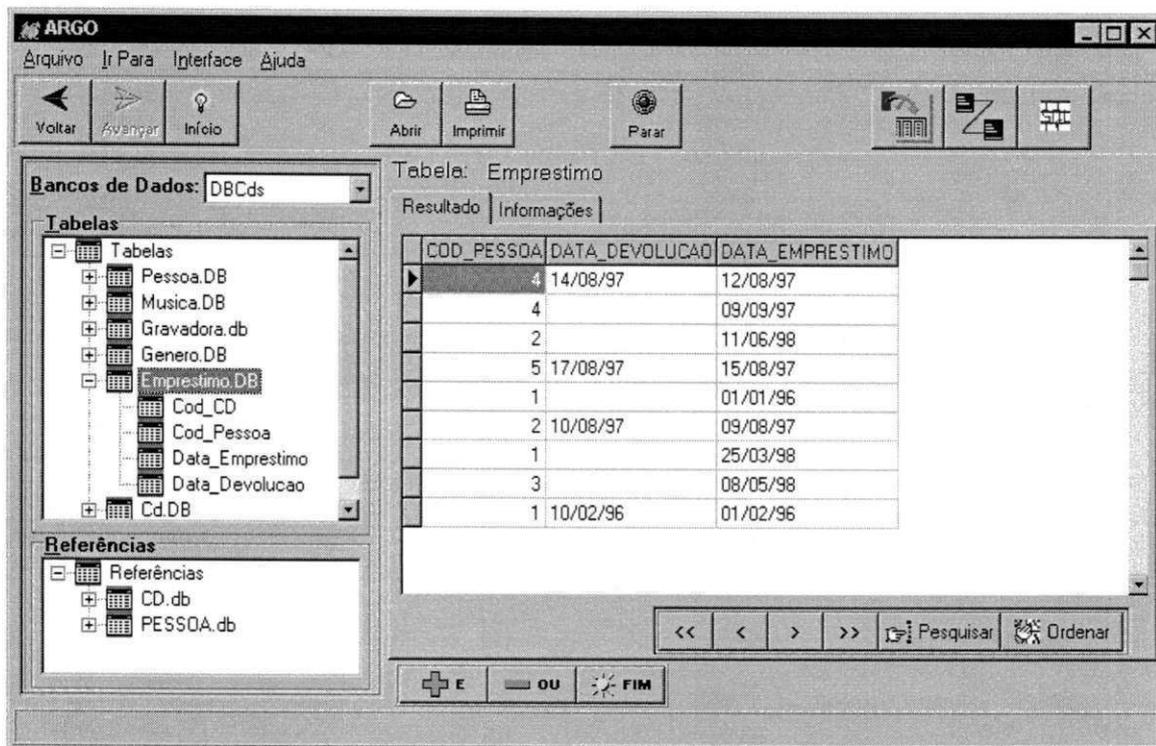


Fig.IV.21 - Resultado da seleção da tabela Empréstimo

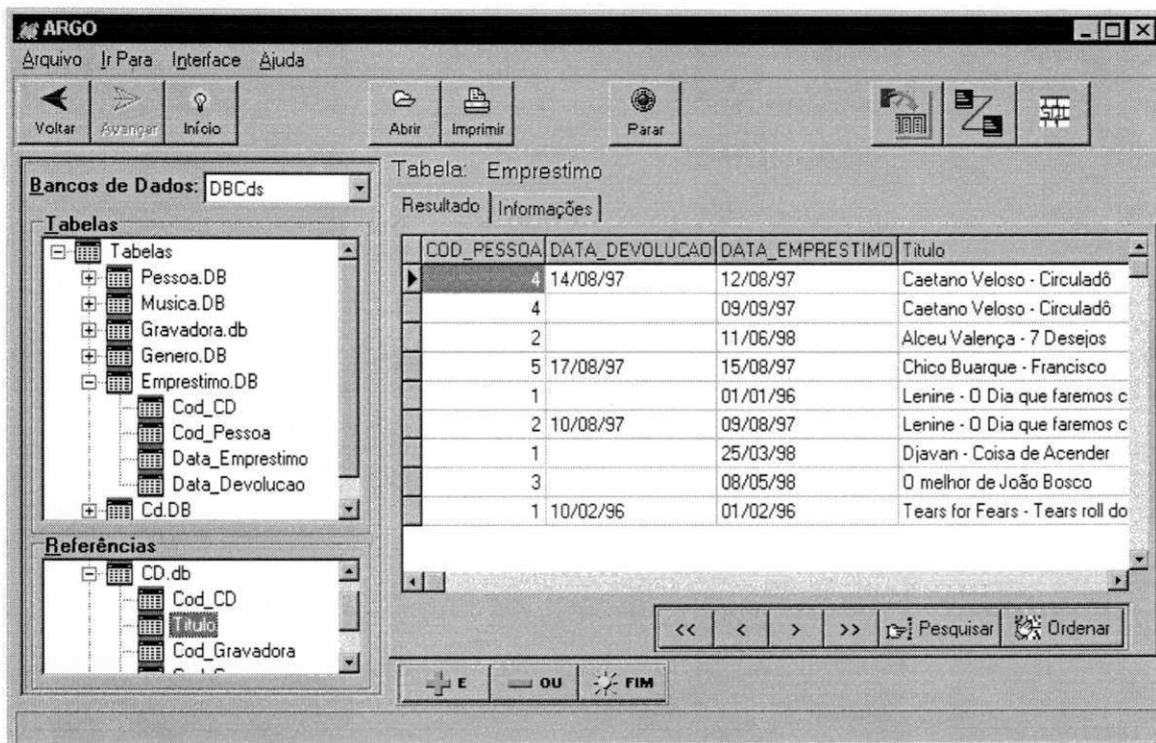
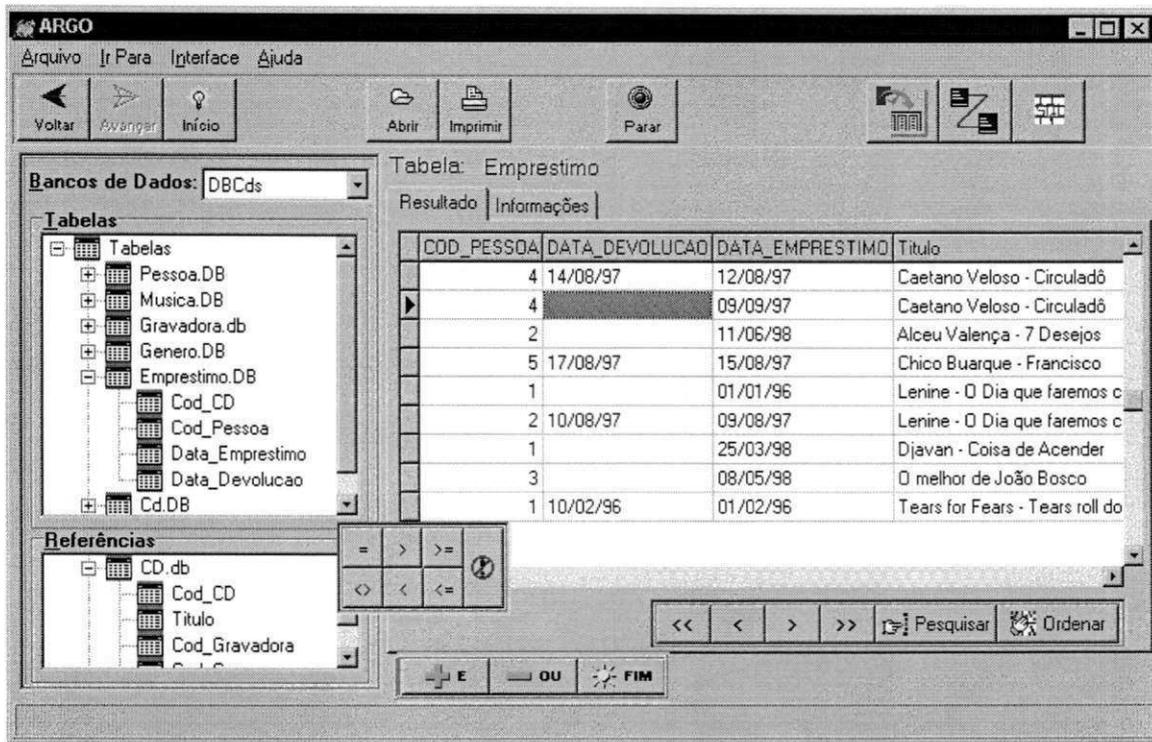


Fig.IV.22 – Selecionando campo da tabela Cd para exibi-lo na Janela de Conteúdo

↵ Fazer um duplo “clíc” no campo Data\_Devolução na Janela de Conteúdo

↳ Na Barra de Condições, que é exibida, selecionar o operador “=” (na Fig.IV.23 é apresentado o resultado destas ações)



**Fig.IV.23 - Exibição da Barra de Condições ao “clique” em Data\_Devolução p/ escolher o operador “=”**

↳ Fazer um duplo “clique” no campo Data\_Devolução que **não** contenha valor preenchido (se o campo está vazio é porque não houve a devolução)

↳ Selecionar o botão FIM

O resultado, apresentado na Fig.IV.24, é a exibição dos registros da tabela EMPRESTIMO, que possuem Data\_Devolução com valor nulo, na Janela de Conteúdo. Como a consulta foi feita a partir do resultado da Fig.IV.23, o campo Título é também apresentado.

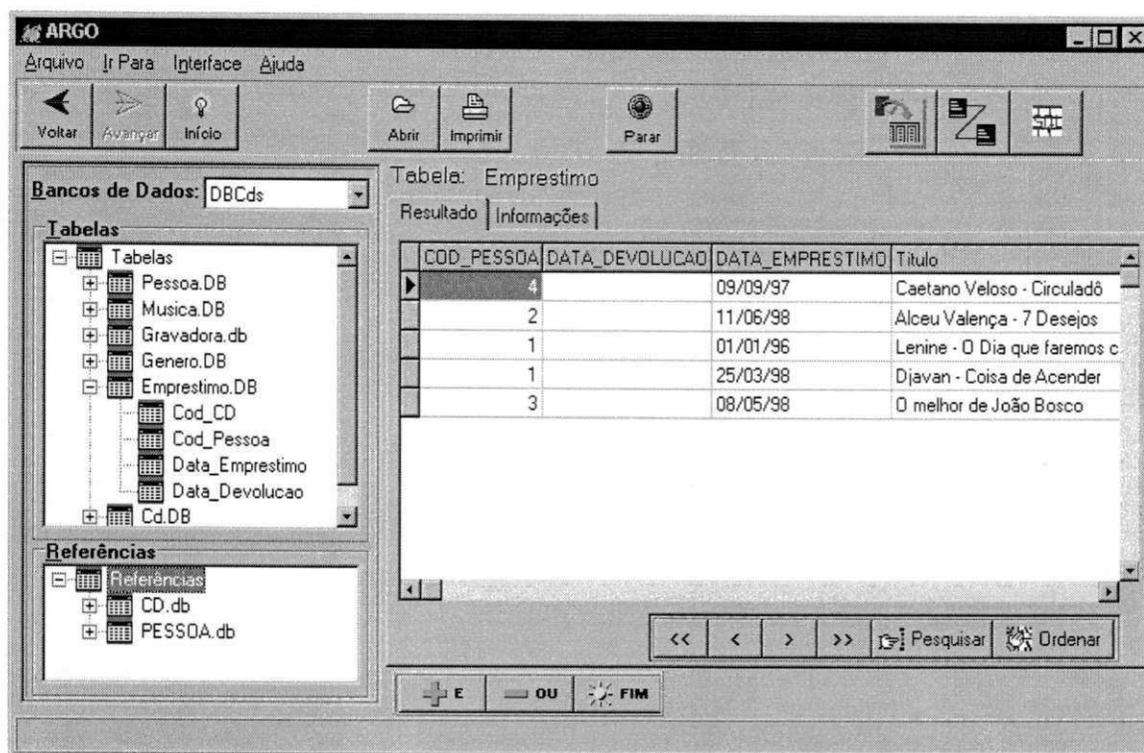


Fig.IV.24 - Resultado da consulta aos CDs emprestados

#### 4.4.5 Análise da interface Avançada

Esta interface, em relação às apresentadas anteriormente, dá maior liberdade para um usuário, considerado experiente, realizar suas consultas. Contudo, ela não pode ser considerada de difícil manipulação, porque muito do esforço computacional é feito pela interface (internamente). A idéia é dar poder de expressão ao usuário, sem exigir muito esforço na formulação. É claro que, para realizar uma consulta mais complexa, o usuário precisa conhecer a estrutura do BD, para isto sendo necessário mais informações sobre a estrutura (ou seja, além das tabelas pertencentes ao BD, os campos de cada tabela também podem ser visualizados).

O ganho de expressividade desta interface em relação as interfaces Básica e Mediana se dá devido à maior flexibilidade das operações de seleção e junção de tabelas. Aqui, tanto Seleção quanto Junção podem ser realizadas nas formas conjuntivas (E) ou disjuntivas (OU), e são realizadas explicitamente por um usuário (*theta*-junção), muito embora conjunções possam também ser realizadas implicitamente (equijunção).

Em suma, com esta interface podemos garantir as seguintes funcionalidades: Projeção de campos, sem restrições; Seleção de valores com condições simples ou múltiplas (fazendo uso dos operadores booleanos E e OU); Junção com igualdade simples ou múltipla; Junção com maior (ou menor) que; Junção com condição adicional; recuperação de valores (tipo *string*) com LIKE (quando pesquisando um padrão); recuperação de valores com NULL; ordenação nas formas ascendentes ou descendentes.

Embora seja uma interface avançada em relação as anteriores, ela possui ainda muitas limitações, no que diz respeito ao poder de expressão, se comparada a uma linguagem de consulta do tipo SQL. Por exemplo, ao contrário da linguagem SQL, não são permitidos: Junção de uma tabela com ela própria; consultas com subconsultas aninhadas; uso de funções embutidas (COUNT, SUM, AVG, MAX, MIN); uso de GROUP BY e HAVING; uso do operador [NOT] EXISTS; uso do operador UNION; uso de NOT LIKE.

#### **4.5 Interface SQL**

Esta interface é indicada para usuários especialistas no sentido de que conhecem a linguagem de consulta SQL, e preferem entrar com comandos SQL ao invés de elaborar visualmente suas consultas.

Para usar esta interface é necessário selecionar o BD desejado na Janela de Banco de Dados e entrar com os comandos SQL na Janela de Comandos (ver Fig.IV.25). Para ver o resultado da consulta, basta selecionar o botão (executar) ao lado da Janela de Comandos, daí a consulta é processada e seu resultado exibido na Janela de Conteúdo.

A Barra de Ícones e a Barra de Menu são utilizadas da mesma forma que nas interfaces anteriores. O usuário pode fazer *browsing* nas consultas geradas nesta interface, através das opções Voltar, Avançar e Início, sem nenhuma mudança nos critérios de uso destas opções em relação as demais interfaces descritas anteriormente.

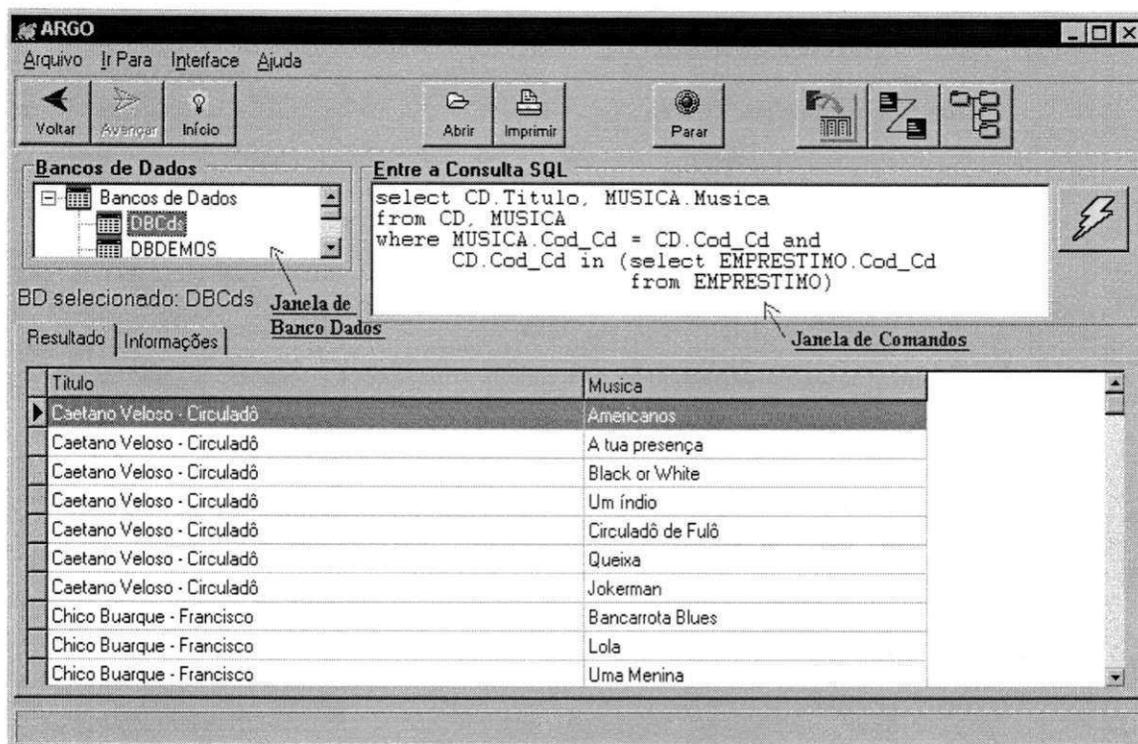


Fig.IV.25 - Interface SQL

#### 4.5.1 Análise da Interface SQL

Todo o poder de expressão da linguagem SQL pode ser aproveitado nesta interface. O conhecimento da sintaxe da linguagem, por parte do usuário, é fundamental para que a consulta possa ser executada com sucesso, assim como o conhecimento do esquema do BD.

Percebemos, então, que esta interface será estritamente utilizada por usuários especialistas (profissionais). E um dos motivos de não nos aprofundar nela, é devido ao fato de que nossa preocupação primordial é com os usuários não-profissionais, que são os verdadeiros usuários de um BD.

# Capítulo V

## O Metabanco de Dados e sua Interface

Todo Sistema de Gerenciamento de Banco de Dados (SGBD) possui um catálogo contendo informações a respeito de suas tabelas. Neste trabalho, informações do tipo chave-estrangeira e sua respectiva tabela de referência são necessárias, e poderiam perfeitamente ser obtidas através do catálogo nativo de um SGBD. Mas, um outro tipo de informação, também necessária, não é encontrada nos catálogos convencionais, qual seja: se o campo é relevante ao usuário. Além do mais, como cada SGBD possui um catálogo proprietário, o fator portabilidade ficaria comprometido. Por estas razões, tornou-se necessário implementar mais um módulo na arquitetura do ARGO, o gerenciador de metabancos de dados.

Dois tipos de controle são tratados pelo metabanco, um trata das informações a respeito dos campos de cada tabela de cada BD, independente dos usuários que as manipulam, enquanto que o outro controle trata das informações a respeito dos campos das tabelas que são relevantes a cada usuário.

As informações que são mantidas no metabanco, para os campos de cada tabela do BD, e independentes do usuário, são: campo(s) chave(s) estrangeira, e campo(s) da tabela referenciada. Embora estas informações possam ser obtidas via catálogo dos SGBDs,

decidimos colocá-las, também, no metabanco para facilitar a obtenção das mesmas, ficando mais independentes dos catálogos proprietários.

Um outro tipo de informação tratada pelo metabanco, que é totalmente dependente do usuário, diz respeito aos campos relevantes. Por campos relevantes entenda-se aqueles que são definidos, de acordo com os critérios de cada usuário, como importantes, ou seja, realmente necessários às suas necessidades, às suas consultas ao BD.

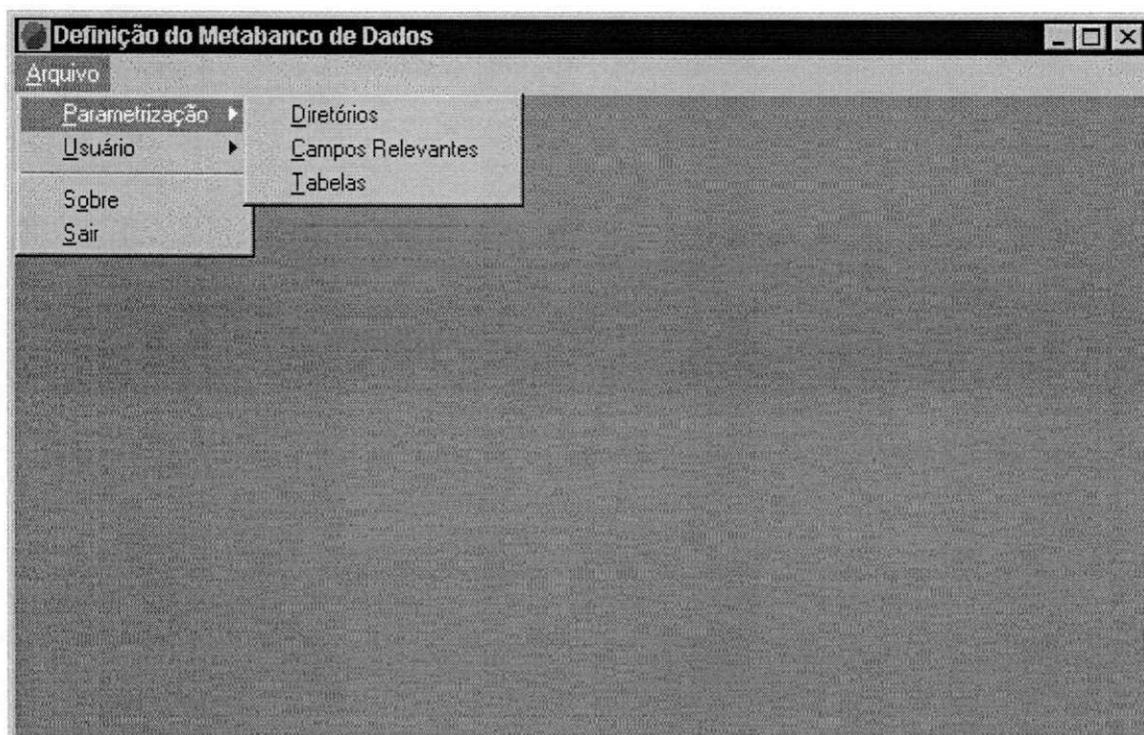
Antes da criação dos metabancos, é feita a parametrização dos diretórios (*path*) onde os metabancos serão criados.

## **5.1 Interface para manutenção dos Metabancos e cadastro de Usuários**

Implementamos uma ferramenta, simplória, com o intuito de agilizar o preenchimento dos metabancos. Nela, são disponibilizadas as opções de parametrização: Diretórios, Campos Relevantes e Tabelas do BD (chaves estrangeiras e campos das tabelas referenciadas), além do próprio cadastro de usuários, que deve existir obrigatoriamente para que o ARGO possa ser usado.

A interface, para fazer a parametrização citada, em muito vai agilizar o trabalho de manutenção dos metabancos. Com exceção da parametrização das tabelas do BD (chaves estrangeira e campos das tabelas referenciadas), que requer a digitação de algumas informações necessárias, o procedimento de preenchimento das tabelas é automático, havendo a necessidade de usar apenas o *mouse* para fazer a seleção dos campos desejados.

A tela inicial é apresentada na Fig.V.1. Nela é apresentado um menu com as opções das parametrizações. A parametrização dos diretórios deve ser feita antes da criação dos metabancos, pois nos diretórios serão mantidas as tabelas do metabanco.



**Fig.V.1 - Tela inicial com as opções de parametrização do Metabanco de dados**

### **5.1.1 Parametrização dos Diretórios dos Metabancos**

Ao selecionar a opção **Parametrização/Diretório**, no *menu* de opções da Fig.V.1, uma janela para cadastro do *path* de cada metabanco é exibida, e nela os campos requisitados devem ser preenchidos. Para os diretórios já cadastrados, consultas ou alterações podem ser realizadas (Fig.V.2).

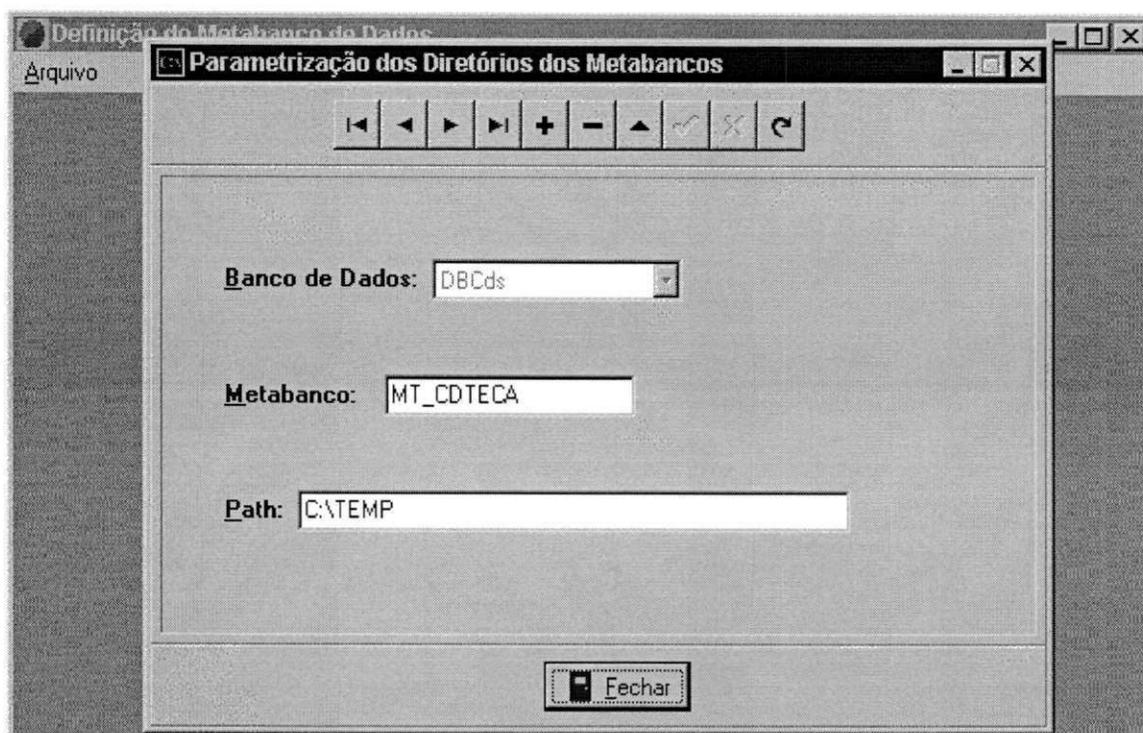


Fig.V.2 - Janela para cadastro dos diretórios

### 5.1.2 Parametrização dos Campos Relevantes

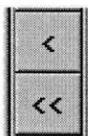
Ao ser selecionada a opção **Parametrização/Campos Relevantes**, a janela apresentada na Fig.V.3 é exibida.

Quando um BD é selecionado seu esquema é exibido. Numa janela apropriada é preciso informar o identificador (*login*) do usuário para quem serão definidos os campos relevantes. Ao selecionar uma tabela do BD, seus campos são exibidos para que sejam selecionados os campos considerados relevantes a um usuário. Para isso, uma barra de botões é usada. Nesta barra, estão contidos os botões (identificando setas direcionadas para a direita e para esquerda) que são descritos a seguir:



Esses dois botões são usados para selecionar os campos relevantes da tabela para o usuário. Com o primeiro botão, seleciona-se um campo por vez na janela onde estão exibidos os campos da tabela selecionada. Este botão, portanto, permite a seleção campo

a campo. O segundo botão, no entanto, seleciona todos os campos da tabela para a janela de campos relevantes, significando que todos os campos da tabela selecionada são relevantes ao usuário.



Estes dois botões têm a funcionalidade inversa dos dois outros acima descritos, ou seja, eles são usados para retirar campos da janela de campos relevantes, significando que o campo passou a ser não relevante ao usuário. O primeiro botão retira os campos selecionados, um a um. O segundo botão retira todos os campos de uma única vez.

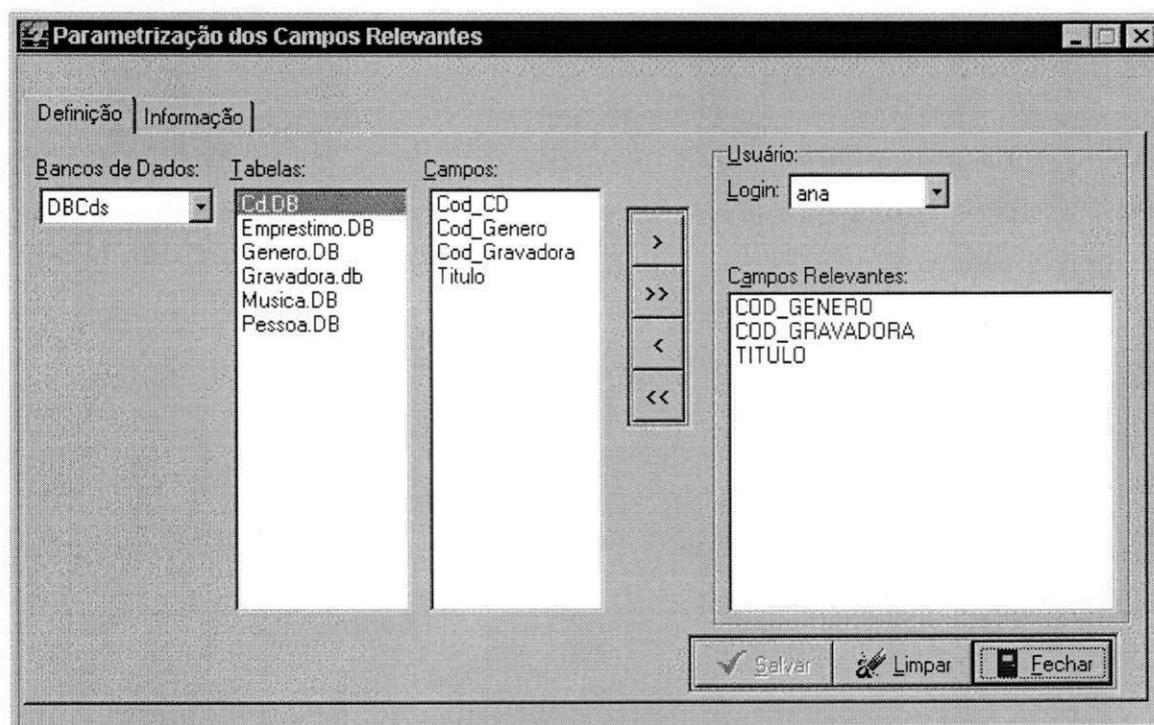


Fig.V.3 - Janela para definição dos campos relevantes do usuário

### 5.1.3 Parametrização dos Campos das Tabelas do BD

Para gerar as tabelas do metabanco, independente de usuário, deve-se selecionar a opção **Parametrização/Tabela**, a janela da Fig.V.4 é exibida e nela deve-se selecionar o BD e a tabela desejada. Para cada tabela selecionada, criar uma meta-tabela correspondente a partir da tabela do BD. A tabela criada terá três campos (ver na seção 6.3 a definição das tabelas do metabanco).

O botão Criar tem a função de gerar a meta-tabela de uma tabela selecionada. Um dos campos da meta-tabela criada, denominada Campo, é automaticamente preenchido, e conterá os nomes dos campos da tabela do BD. Os demais campos da meta-tabela devem ser preenchidos manualmente, tais campos são: Chave-Estrangeira (*False* ou *True*), e caso o valor seja *True* o campo Nome-Campo-Referencia deve ser preenchido com o nome da tabela referenciada seguido de um ponto e do nome do campo (chave primária) da tabela referenciada.

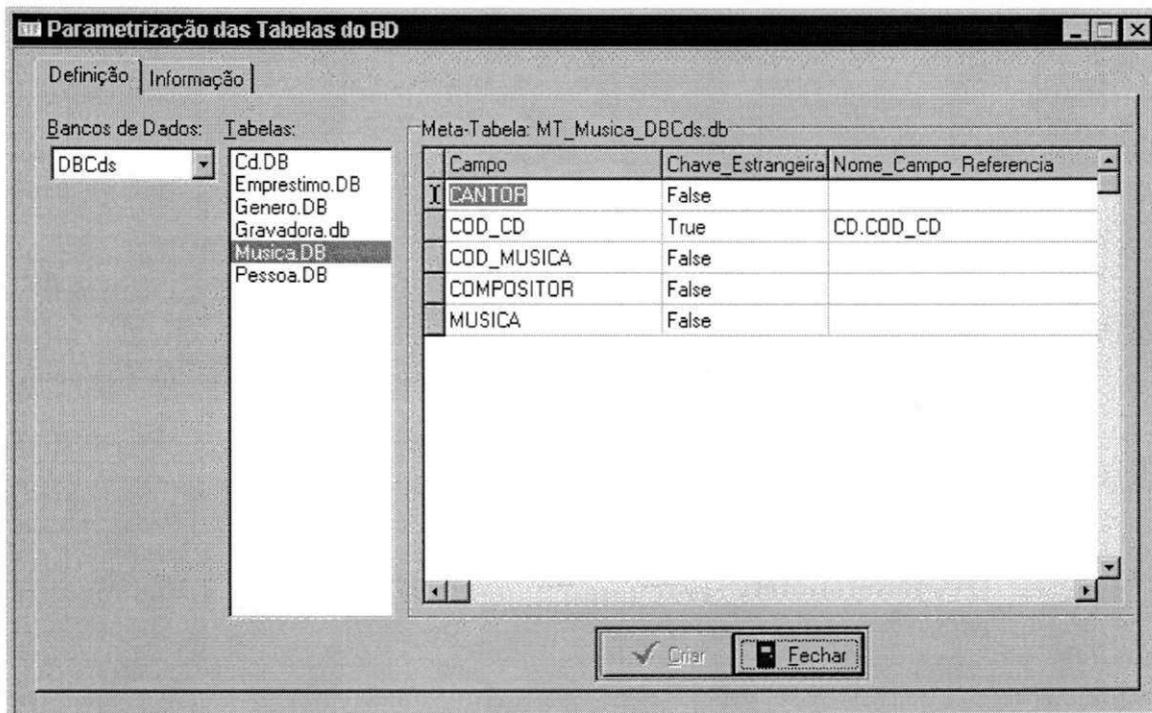
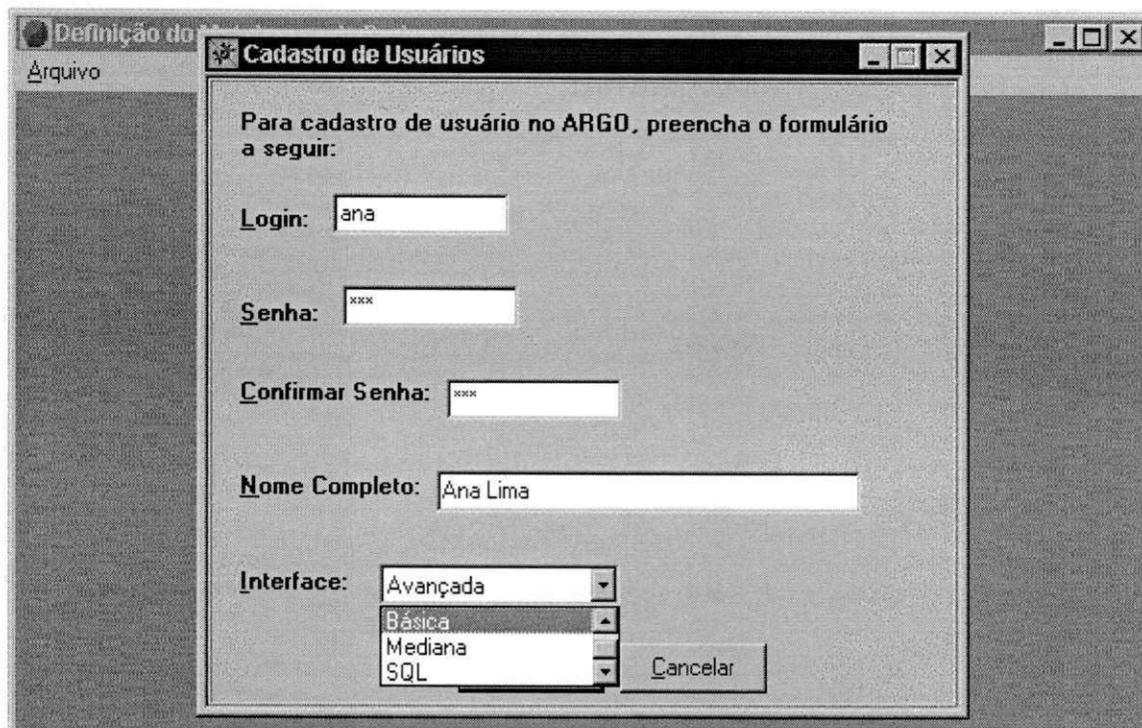


Fig.V.4 - Janela para criação do metabanco que independe do usuário

### 5.1.4 Cadastro de Usuários

Todo usuário deve ser inicialmente cadastrado antes de usar o ARGO pela primeira vez. Como o ARGO é adaptativo, precisamos manter históricos particulares de um usuário para posterior reconhecimento do seu perfil.

Cada usuário tem um *login* e uma senha que o identifica de forma única dentro da tabela de usuário (Fig.V.5). No cadastramento de um usuário, é escolhida a interface (Básica, Mediana, Avançada ou SQL) considerada, a *piori*, mais adequada ao seu perfil.



**Fig.V.5 – Janela para cadastro dos usuários do ARGO**

# Capítulo VI

## Implementação das Interfaces

### do ARGO

O módulo gerenciador da interface, apresentado na Fig.III.1 (capítulo III), é composto por uma estrutura de tabelas, um conjunto de algoritmos mantém essas tabelas. Neste capítulo, apresentaremos tais estruturas e algoritmos, e também a estrutura do módulo gerenciador do metabanco de dados que dá suporte ao ARGO.

O módulo gerenciador da interface e o módulo gerenciador dos metabancos de dados estão concluídos, ou seja, foram especificados e implementados na sua totalidade.

#### **6.1 Ambiente de desenvolvimento do ARGO**

O ARGO foi implementado em Delphi 2.0, sobre o sistema operacional Windows 95. As tabelas internas são do tipo Paradox, por vir embutido no ambiente Delphi. Usamos o BDE (Borland DataBase Engine) para configurar as possíveis bases de dados (relacionais) a serem acessadas pelos usuários, através do ARGO.

#### **6.2 Estrutura do módulo gerenciador das interfaces**

A estrutura de tabelas deste módulo é composta pelas tabelas de Usuários e Históricos.

Um usuário ao abrir uma sessão no ARGO está, neste momento, apto a realizar interações e gerar consultas. Informações sobre as consultas geradas e as formas de interações realizadas são mantidas num histórico.

A seguir apresentamos a estrutura das tabelas componentes ao módulo gerenciador das interfaces, e em seguida a estrutura das tabelas do módulo gerenciador dos metabancos de dados.

### 6.2.1 Tabela de Usuários

Todo usuário precisa de uma identificação para conseguir iniciar uma sessão no ARGO, para isso são usados um identificador de usuário (ou *login* de usuário) e uma senha para permitir o acesso. Um campo identificando o tipo da interface (Básica, Mediana, Avançada ou SQL) é a mais adequada ao usuário, também faz parte desta tabela. Assim sendo, a tabela possui a seguinte estrutura:

*(login, senha, nome\_usuario, tipo\_interface)*

O *login* do usuário é único, portanto é definido como chave primária da tabela. A *senha* é usada para garantir o acesso exclusivo ao usuário cadastrado no ARGO. O *nome\_usuario* é usado como uma informação complementar do usuário. O *tipo\_interface*, como já dito, é usado para indicar a interface adequada ao usuário.

### 6.2.2 Tabela de Históricos

Esta tabela é usada para guardar o histórico (ou *log*) das consultas realizadas por um usuário. Além de ser utilizada na classificação do estereótipo do usuário (capítulo VII), ela é usada pelo ARGO para permitir que o usuário faça “navegação” pelas consultas realizadas na sessão corrente (via ícones Voltar e Avançar das interfaces do ARGO).

A identificação da interface onde a consulta foi gerada é também registrada nesta tabela, isto para impedir que uma consulta realizada numa interface seja visualizada em quaisquer das outras três, já que as interfaces não são totalmente coerentes entre si, ou

seja, há funcionalidades de uma dada interface que não estão presentes nas demais, e vice-versa.

Apenas as consultas geradas na sessão corrente podem ser acessadas pelo usuário durante sua interação com o sistema. Para identificar se uma consulta pertence a sessão corrente, um *flag* é usado. Sendo *True* o valor deste *flag*, a consulta é permanente, portanto não foi realizada na sessão corrente, caso contrário, a consulta foi realizada na sessão corrente. A cada sessão e interação esta tabela é atualizada com novas informações sobre as consultas e as formas de interação, e ao fim da sessão todos esses registros têm seu *status* modificado para *True* (ou seja, passam a ser permanentes).

A estrutura desta tabela é a seguinte:

*(login\_usuario, nome\_BD, nome\_interface, nome\_consulta, tabela\_selecionada, flag\_sessão\_corrente, Dbgrid\_usada, texto\_SQL)*

- <login\_usuario> - identifica o usuário que realizou a consulta
- <nome\_BD> - nome do Banco de Dados usado nas consultas
- <nome\_interface> - nome da interface onde foi gerada a consulta
- <nome\_consulta> - um nome que identifica a consulta do usuário.  
Ele é gerado internamente cada vez que uma consulta é formulada, e segue o padrão:

**Consulta**<login\_usuario><num\_sequenc>

**Consulta** indica uma consulta do usuário e **num\_sequenc** uma numeração começando de zero por usuário e interface.

- <nome\_tabela\_selecionada> - nome da tabela selecionada na Janela de Esquema, se nenhuma tabela estiver selecionada este valor é null
- <flag\_permanente> - se a consulta é da sessão corrente o valor é *False*, caso contrário é *True*
- <Dbgrid\_usado> - este campo se tornou necessário apenas para a interface Mediana, devido ao fato de serem usadas duas Janelas de Conteúdo, e portanto haver a necessidade de saber em qual delas a consulta foi realizada.

<texto\_SQL> - o texto SQL da consulta gerada

### 6.3 Estrutura do módulo gerenciador dos metabancos de dados

Como já dito, no metabanco dois tipos de controles são tratados: um à respeito dos campos de cada tabela de cada BD (independente dos usuários), e outro tratando das informações que são relevantes para os usuários. Para o primeiro tipo de controle, para cada tabela de um BD é criada uma meta-tabela (tabela do metabanco) correspondente, seguindo a padronização: prefixo **MT\_** para identificar que é uma tabela do metabanco; os demais elementos contemplam a tabela do BD e o próprio BD, ou seja:

**MT\_<nome-tabela>\_<nome-BD>**

**MT** - é o prefixo que identifica uma tabela do metabanco

**<nome-tabela>**- é o nome da tabela do BD descrita no metabanco

**<nome-BD>** - é o nome do BD que contém as tabelas descritas no metabanco

A estrutura desta tabela é a seguinte:

**(*nome\_campo*, *chave\_estrangeira*, *nome\_campo\_referência*)**

Onde:

**nome\_campo** - o mesmo nome do campo na tabela do BD

**chave\_estrangeira** - um flag (*True* ou *False*) para indicar se campo é chave estrangeira

**nome\_campo\_referência** - nome do campo na tabela do BD referenciada. Na realidade o conteúdo deste campo deverá ser: nome\_tabela.nome\_campo\_tabela

O campo **nome\_campo** é índice único, não podendo existir numa mesma tabela mais de um campo com o mesmo nome. O campo **nome\_campo\_referência** só é preenchido no caso do nome\_campo ser uma chave estrangeira, caso contrário ele será nulo.

Para o segundo tipo de controle do metabanco, que contempla os campos relevantes para o usuário, uma tabela única é criada para cada BD, e segue o padrão: prefixo **MT\_** para identificar que é uma tabela do metabanco, seguido do nome **Campos\_Relevantes\_Tabelas\_** e do nome do BD correspondente, ou seja:

**MT\_Campos\_Relevantes\_Tabelas\_<nome-BD>**

Cada BD possui uma tabela de campos relevantes distinta, no próprio nome da tabela já está a identificação do BD ao qual ele está associado (<nome-BD>).

A estrutura desta tabela é a seguinte:

*(login\_usuario, nome\_tabela, campo\_relevante)*

<login\_usuario> - o identificador do usuário que usa a tabela

<nome\_tabela> - o nome da tabela do BD, para a qual são definidos os campos relevantes

<campo\_relevante> - o nome do campo da tabela que é relevante para o usuário

Um índice único composto pelos campos login\_usuario, nome\_tabela, campo\_relevante, é usado para garantir que não haverá mais de um registro com os mesmos valores.

### 6.3.1 Definição do metabanco para a aplicação usada como exemplo

Dado o esquema do banco de dados **DBCds** (banco de dados para controle de uma CDteca particular), composto por seis tabelas:

**CD**(Cod\_Cd, Titulo, Cod\_Gravadora, Cod\_Genero)

- Cod\_Cd - Chave primária
- Cod\_Gravadora - Chave estrangeira (da tabela Gravadora)
- Cod\_Genero - Chave estrangeira (da tabela Gênero)

**EMPRÉSTIMO**(Cod\_Cd, Cod\_Pessoa, Data\_Emprestimo, Data\_Devolucao)

- Cod\_Cd+Cod\_Pessoa+Data\_Emprestimo - Chave primária composta
- Cod\_Cd - Chave estrangeira (da tabela CD)
- Cod\_Pessoa - Chave estrangeira (da tabela Pessoa)

**GENERO**(Cod\_Genero, Genero)

- Cod\_Genero - Chave primária

**GRAVADORA**(Cod\_Gravadora, Gravadora)

- Cod\_Gravadora - Chave primária

**MÚSICA**(Cod\_Cd, Cod\_Musica, Musica, Cantor, Compositor)

- Cod\_Cd+Cod\_Musica - Chave primária composta
- Cod\_Cd - Chave estrangeira (da tabela CD)

**PESSOA**(Cod\_Pessoa, Pessoa, Telefone)

- Cod\_Pessoa - Chave primária

O esquema é usado como exemplo no decorrer deste trabalho, assim como o metabanco gerado para o mesmo.

- Meta-tabelas geradas para o banco de dados DBCDs:

Nome_Campo	Chave_Estrangeira	Nome_Campo_Referência(tab.campo)
Cod_Cd	False	null
Titulo	False	null
Cod_Gravadora	True	GRAVADORA.Cod_Gravadora
Cod_Genero	True	GENERO.Cod_Genero

**Fig.VI.1 - Meta tabela MT\_CD\_DBCds**

Nome_Campo	Chave_Estrangeira	Nome_Campo_Referência(tab.campo)
Cod_Cd	True	CD.Cod_Cd
Cod_Pessoa	True	PESSOA.Cod_Pessoa
Data_Emprestimo	False	null
Data_Devolucao	False	null

**Fig.VI.2- Meta tabela MT\_EMPRESTIMO\_DBCds**

Nome_Campo	Chave_Estrangeira	Nome_Campo_Referência(tab.campo)
Cod_Genero	False	null
Genero	False	null

**Fig.VI.3 - Meta tabela MT\_GENERO\_DBCds**

Nome_Campo	Chave_Estrangeira	Nome_Campo_Referência(tab.campo)
Cod_Gravadora	False	null
Gravadora	False	null

**Fig.VI.4 - Meta tabela MT\_GRAVADORA\_DBCds**

Nome_Campo	Chave_Estrangeira	Nome_Campo_Referência(tab.campo)
Cod_Cd	True	CD.Cod_Cd
Cod_Musica	False	null
Musica	False	null
Compositor	False	null

**Fig.VI.5 - Meta tabela MT\_MUSICA\_DBCds**

Nome_Campo	Chave_Estrangeira	Nome_Campo_Referência(tab.campo)
Cod_Pessoa	False	Null
Pessoa	False	Null
Telefone	False	Null

**Fig.VI.6 - Meta tabela MT\_PESSOA\_DBCds**

- Meta-tabela gerada para definição dos campos relevantes, para o usuário Ana, do banco de dados DBCDs:

Login-Usuário	Nome_tabela	Campo_Relevante
Ana	CD	Título
Ana	CD	Cod_Gravadora
Ana	CD	Cod_Genero
Ana	EMPRESTIMO	Cod_Pessoa
Ana	EMPRESTIMO	Data_Emprestimo
Ana	EMPRESTIMO	Data_Devolucao
Ana	GENERO	Genero
Ana	GRAVADORA	Gravadora
Ana	MUSICA	Musica
Ana	MUSICA	Cantor
Ana	MUSICA	Compositor
Ana	PESSOA	Pessoa
Ana	PESSOA	Telefone

**Fig.VI.7- Meta-tabela com os campos relevantes por usuário**

## 6.4 Definição da tabela de Diretórios para o metabanco

As tabelas do metabanco de dados serão gravadas num diretório previamente definido pelo usuário. Sugerimos que seja usado um diretório particular para cada BD, obtendo desta forma uma melhor organização. Uma tabela de diretórios é usada para registrar os diretórios onde estão gravados os metabancos de dados, sua estrutura é a seguinte:

*(Nome-BD, Nome-Metabanco, Path)*

Esta tabela é pesquisada sempre que as informações do metabanco forem necessárias.

## 6.5 Algoritmos das interfaces

Para cada uma das interfaces especificadas são apresentados alguns dos algoritmos para obtenção das funcionalidades propostas. A ordem de apresentação está de acordo com a ordem das seções do capítulo IV.

### 6.5.1 Algoritmos para a Barra de Ícones

#### **Voltar**

*para o registro do log das consultas correspondente a consulta ativa na tela*

*se não for o primeiro registro então*

*início*

*ler no log das consultas o registro imediatamente anterior, por*

*login\_usuario+ nome\_BD+nome\_interface*

*se flag\_permanente = False então /\* a consulta é da sessão corrente \*/*

*início*

*obter o texto\_SQL*

*executar o texto\_SQL*

*inserir um novo registro no log de consultas para a consulta re-executada, neste caso o flag\_permanente é gravado com True (para evitar a exibição repetitiva das consultas re-executadas)*

*se <nome\_interface> for "Avançada" então*

*limpar Janela Referências*

*colocar na Janela de Referências as tabelas referências da atual tabela*

*fim se*

*fim se /\* flag é False \*/*

*fim se*

*se não /\* é o primeiro registro \*/*

*desabilitar o botão Voltar*

## **Avançar**

*para o registro do log das consultas correspondente a consulta ativa na tela*

*se não for o último registro então*

*início*

*ler no log das consultas o registro imediatamente seguinte, por*

*login\_usuario+ nome\_BD+nome\_interface*

*se flag\_permanente = False então /\* a consulta é da sessão corrente \*/*

*início*

*obter o texto\_SQL*

*executar o texto\_SQL*

*inserir um novo registro no log de consultas para a consulta re-executada, neste caso  
o flag\_permanente é gravado com True*

*se <nome\_interface> for "Avançada" então*

*limpar Janela Referências*

*colocar na Janela de Referências as tabelas referenciadas pela atual tabela*

*fim se*

*fim se /\* flag é False \*/*

*fim se*

*se não então /\* é o primeiro registro \*/*

*desabilitar o botão Voltar*

## **Início**

*Para o primeiro registro do log das consultas por login\_usuario+nome\_interface+nome\_BD  
e flag\_permanente = False*

*se não for o primeiro registro então*

*início*

*obter o texto\_SQL*

*executar o texto\_SQL obtido*

*inserir um novo registro no log de consultas para a consulta re-executada, neste caso o flag\_permanente é gravado com True*

*se <nome\_interface> for “Avançada” então*

*início*

*limpar Janela Referências*

*colocar na Janela de Referências as tabelas referências da atual tabela*

*fim se*

*fim se*

*se não então botão Início fica indisponível*

## **Interface**

*se opção\_interface foi selecionada então*

*início*

*fechar a interface anterior*

*exibir a interface escolhida*

*fim se*

## **Abrir**

*finalizar o uso do ARGO para o usuário corrente*

*solicitar o login e a senha para o próximo usuário que deseja abrir uma sessão*

*checar se usuário está cadastrado*

*se sim então*

*abrir sessão para o usuário*

*exibir a interface adequada*

*fim se*

## **6.5.2 Algoritmo para o botão Pesquisar**

Quando o botão Pesquisar tiver sido selecionado, o padrão de pesquisa informado e a condição de pesquisa (prefixo, infixo, sufixo) selecionada, o algoritmo a ser executado é:

*se campo não tiver sido selecionado então*

*enviar mensagem de erro (“Campo para pesquisa não selecionado”)*

*se não*

*para* a consulta SQL corrente, padrão\_pesquisa e condição\_pesquisa informados

*se* cláusula WHERE não existe no SELECT **então**

**acrescentar** a cláusula WHERE

*case* (condição\_pesquisa)

prefixo: **acrescentar** campo\_selecionado LIKE "padrão\_pesquisa%"

infixo: **acrescentar** campo\_selecionado LIKE "%padrão\_pesquisa%"

sufixo: **acrescentar** campo\_selecionado LIKE "%padrão\_pesquisa"

**executar** o comando SELECT final obtido

**inserir** novo registro no log das consultas para a consulta obtida

*fim se*

### 6.5.3 Algoritmo para o botão Ordenar

Quando o botão Ordenar tiver sido pressionado, o campo para ordenação teve sido selecionado e a opção ascendente (ASC) ou descendente (DESC) tiver sido escolhida, o seguinte algoritmo é executado:

*se* campo não tiver sido selecionado **então**

**enviar** mensagem de erro ("Campo para ordenação não selecionado")

*se não*

*para* o texto\_SQL da consulta corrente e o campo\_selecionado na Janela de Conteúdo

*se* a cláusula ORDER BY não existe no SELECT **então**

**acrescentar** a cláusula ORDER BY ao SELECT

*case* (opção\_ordenação)

ascendente: **acrescenta** campo\_selecionado ASC

descendente: **acrescentar** campo\_selecionado DESC

**executar** a consulta\_SQL obtida

**inserir** um novo registro no log das consultas com a consulta obtida

*fim se*

#### 6.5.4 Algoritmo para Barra de Condições

Este algoritmo é executado após um campo ter sido selecionado na Janela de Conteúdo, significando que uma condição entre o campo selecionado e um valor (a ser selecionado após uma condição nesta barra ter sido selecionado), ou uma junção entre o campo selecionado e um outro campo a ser selecionado na Janela de Referências, foi estabelecida. Assim, a cláusula a ser montada ficará em “aberto” até que o outro “lado” da condição seja também selecionado.

*se a cláusula WHERE da consulta SQL corrente não existe então*

*acrescentar o WHERE ao SELECT do texto SQL obtido*

*se não*

*acrescentar AND ao WHERE*

*case (operador\_condição)*

*“=”: acrescentar campo\_selecionado “=”*

*“>”: acrescentar campo\_selecionado “>”*

*“<”: acrescentar campo\_selecionado “<”*

*“>=: acrescentar campo\_selecionado “>=”*

*“<>”: acrescentar campo\_selecionado “<>”*

*flag\_condição = true /\* usado para indicar que uma condição foi selecionada \*/*

#### 6.5.5 Algoritmo para seleção de Campo/Valor na Janela de Conteúdo

*se flag\_campovalor = 1 então /\* um campo está sendo selecionado \*/*

*flag\_campovalor := 2*

*se não /\* valor está sendo selecionado \*/*

*se flag\_condição = true então*

*início*

*flag\_campovalor := 1*

*acrescentar à cláusula WHERE valor\_selecionado*

*se <nome\_interface> for diferente de “Avançada” então*

*início*

*executar a consulta SQL obtida*

*inserir no log das consultas a nova consulta SQL obtida*

*flag\_condição := false*

*fim se /\* se a interface for a Avançada a execução só será feita quando o botão FIM  
for pressionado \*/*

*fim se*

### 6.5.6 Algoritmo para interface Básica

Quando uma tabela for selecionada na Janela de Esquema, o algoritmo utilizado para obtenção dos campos relevantes a serem exibidos na Janela de Conteúdo é:

*Pr\_Campos\_Relevantes(usuário, BD\_usado, tabela\_usada)*

*para a tabela MT\_Campos\_Relevantes\_Tabela\_BD, pesquisando por usuário/tabela*

*montar o SELECT e o FROM parcial para os campos relevantes*

*para cada registro da tabela MT\_<tabela\_usada>\_<BD\_usada>*

*se campo\_tabela é chave estrangeira então*

*se tabela\_usada.campo\_tabela está contida no SELECT então*

*se tabela\_referenciada não está contida na cláusula FROM então*

*início*

*montar condição WHERE*

*chamar Pr\_Campos\_Relevantes(usuário, BD\_usado, tabela-  
referenciada) para obter os campos relevantes da tabela referenciada*

*fim se*

Chamada do algoritmo recursivo:

*chamar Pr\_Campos\_Relevantes(<usuário>, <BD\_selecionado>, <tabela\_selecionada>)*

*executar o comando SELECT final obtido e colocar resultado na Janela de Conteúdo*

*inserir consulta gerada no log das consultas*

### 6.5.7 Algoritmos para interface Mediana

**Algoritmo 1** - Quando uma tabela é selecionada na Janela de Esquema, o algoritmo utilizado para obtenção dos campos relevantes e dos que são chaves estrangeira a serem exibidos na Janela de Conteúdo é:

*para cada registro da tabela MT\_Campos\_Relevantes\_Tabela\_BD por usuário/tabela-  
selecionada*

*montar o comando SELECT e FROM parcial*

*para cada* registro em *MT\_<tabela\_selecionada>\_<BD\_selecionado>*

*se* campo\_tabela é chave estrangeira **então**

*se* campo\_tabela não está contido no *SELECT* **então**

**início**

*montar* a cláusula *SELECT* parcial

*sublinhar* este campo para indicar um link

*montar* a cláusula *FROM* parcial

**fim se**

*se* nenhuma Janela de Conteúdo está Enable **então**

*DBGrid\_usar := DBGrid1*

*se* ambas as janelas (Janela de Conteúdo1 e Janela de Conteúdo 2) estão Enable **então**

**início**

*perguntar* em qual das duas deve ser feita a sobreposição de tabelas (opção\_usuario)

*DBGrid\_usar := opção\_usuario*

**fim se**

*se* apenas a Janela de Conteúdo1 está Enable **então**

*DBGrid\_usar := DBGrid2*

*executar* consulta *SQL* obtida e colocar resultado no *DBGrid\_usar*

*inserir* consulta obtida no log das consultas

**Algoritmo 2** - Quando uma tabela é selecionada via link na Janela de Conteúdo, o algoritmo usado é:

*se* campo\_link é chave estrangeira em *MT\_<tabela\_selecionada>\_<BD\_selecionado>*, por tabela\_selecionada/ campo\_link **então**

**início**

*para cada* registro em *MT\_Campos\_Relevantes\_Tabela\_BD*, por usuário/tabela-referenciada

*montar* o comando *SELECT* e *FROM* parcial

*para cada* registro em *MT\_<tabela\_selecionada>\_<BD\_selecionado>*

**início**

*se* campo\_tabela é chave estrangeira **então**

**início**

*montar* a cláusula *SELECT* parcial

*sublinhar* este campo para indicar um link

*montar* a cláusula *FROM* parcial

*fim se*  
*fim para cada*  
*para a Janela Conteúdo onde o campo\_link foi selecionado*  
*DBGrid\_usar := DBGrid\_corrente*  
*executar consulta SQL obtida e colocar resultado no DBGrid\_usar*  
*inserir a consulta obtida no log das consultas*  
*fim se*

### 6.5.8 Algoritmos para interface Avançada

**Algoritmo 1** - Quando uma tabela é selecionada na Janela de Esquema, o algoritmo usado para obter os campos relevantes, e as tabelas referenciadas na Janela de Referências é:

*para cada registro em MT\_Campos\_Relevantes\_Tabela\_BD, por usuário/tabela-selecionada*  
*montar o comando SELECT e FROM parcial*  
*limpar Janela de Referências*  
*para cada registro em MT\_<tabela\_selecionada>\_<BD\_selecionado>*  
*se campo\_tabela é chave estrangeira então*  
*exibir tabela\_referência na Janela de Referências (montando no TreeView a*  
*hierarquia tabelas/campos)*  
*executar consulta SQL obtida e colocar resultado na Janela de Conteúdo*  
*inserir a consulta obtida no log das consultas*

**Algoritmo 2** - Para **acrescentar** ou **retirar** um campo da Janela de Conteúdo, clicando no campo da tabela selecionada na Janela de Esquema, o algoritmo usado é:

*se campo\_selecionado está no SELECT da consulta SQL\_corrente então*  
*retirar campo\_selecionado do SELECT*  
*se não então*  
*acrescentar ao SELECT o campo\_selecionado*  
*executar a consulta SQL obtida e colocar resultado na Janela de Conteúdo*  
*inserir a consulta obtida no log das consultas*

**Algoritmo 3** - Para **adicionar** ou **retirar** um campo à Janela de Conteúdo, clicando no campo desejado na Janela de Referências, o algoritmo usado é:

```
se campo_referência_selecionado está no SELECT da consulta_SQL corrente então
início
    retirar campo_referência_selecionado do SELECT
    se no SELECT da consulta corrente não existir outro(s) campo(s) da tabela_referenciada
    então início
        retirar tabela_referenciada do FROM
        retirar a condição WHERE que usa campos da tabela_referência
    fim se
fim se
fim se
se não então /* campo não está no SELECT */
início
    acrescentar ao SELECT o campo tabela_referência.campo_referência_selecionado
    se tabela_referência não estiver contida no FROM então
    início
        acrescentar tabela_referência ao FROM
        se a cláusula WHERE não existe na consulta SQL corrente então
            acrescentar à consulta a cláusula WHERE
        se não então
            acrescentar à consulta o operador de boole AND
        acrescentar à consulta o predicado de junção:
            tabela_selecionada_janela_esquema.campo_tabela =
                tabela_referência.campo_referência_selecionado
    fim se /* tabela não está contida no FROM */
fim se /* campo não está no SELECT */
se flag_condição = false então /* A consulta pode ser executada, independe do botão FIM */
início
    executar a consulta SQL obtida e colocar resultado na Janela de Conteúdo
    inserir a consulta obtida no log das consultas
fim se
```

**Algoritmo 4** - Quando uma tabela é selecionada na Janela de Referências (a própria tabela e não os campos da mesma):

```
para cada registro de MT_<tabela_referência_selecionada>_<BD_selecionado>  
se campo_tabela é chave estrangeira então  
exibir tabela_referência na Janela de Referências (montando no TreeView a  
hierarquia tabelas/campos já existentes)
```

**Algoritmo 5** - Quando o botão FIM é pressionado:

```
se flag_condição = true então  
início  
flag_condição := false  
executar a consulta SQL obtida e colocar resultado na Janela de Conteúdo  
inserir a consulta obtida no log das consultas  
fim se
```

**Algoritmo 6** - Quando os botões E ou OU forem pressionados, no caso se condições múltiplas:

```
se flag_condição = true e condição WHERE existe então  
case (operador_selecionado)  
E: acrescentar AND ao WHERE  
OU: acrescentar OR ao WHERE
```

## Capítulo VII

# O Aspecto Adaptativo do

# ARGO

Um sistema de computação não deve sempre impor uma padronização do perfil de seus usuários, mas pode levar em conta o conhecimento do usuário sobre o sistema, assim como as formas específicas de interação usuário-sistema. Para tanto, é necessário que sejam levadas em consideração as diferentes habilidades e necessidades dos usuários durante a interação. Nas últimas duas décadas, pesquisas têm sido feitas na área de Inteligência Artificial (IA) sobre sistemas cuja interação é feita sob medida para os usuários, através de **modelos de usuário** [Errico97]. A idéia desses modelos é tornar a interação usuário-sistema mais personalizada e eficiente, levando em consideração as expectativas do sistema sobre os usuários, dinamicamente adaptando-se aos objetivos, habilidades e preferências dos mesmos.

Uma definição clássica para Modelo de Usuário é proposta por Wahlster e Kobsa, com respeito a sistemas de diálogo [Wahlster86]: “Um modelo de usuário é uma fonte de conhecimento que contém suposições explícitas sobre todos os aspectos do usuário que podem ser relevantes para o comportamento do sistema”. Para que um modelo de usuário possa ser usado, algumas questões básicas devem ser tratadas, tais como: a forma de

obtenção do conhecimento sobre as interações do usuário; a atualização do modelo; e os resultados que podem ser obtidos a partir dele.

Algumas das abordagens usadas para gerar modelos de usuário são baseadas em estereótipos, no reconhecimento dos planos, ou nas preferências de usuário.

A abordagem baseada em estereótipos é usada para modelar o conhecimento do usuário [Rich89], e é a mais usualmente empregada na definição de modelos de usuário. Ela é normalmente empregada em áreas de aplicações que exijam uma rápida, mas não necessariamente exata e completa avaliação sobre o conhecimento de um usuário [Kobsa93]. Nesta abordagem, a pessoa responsável por fazer o modelo de usuário deve se preocupar com três tarefas, quais sejam: (1) a identificação de subgrupos de usuários que possuam características homogêneas e sejam relevantes à aplicação; (2) a identificação de características chave, ou seja, aquelas que permitam a identificação dos membros de um subgrupo de usuário; (3) a representação em estereótipos. As características de um identificado grupo de usuário devem ser formalizadas em um apropriado sistema de representação. A coleção de todas as características de um subgrupo de usuário é chamada de estereótipo de usuário. Estereótipos para grupos de usuários, com diferentes graus de conhecimento, têm sido empregados para adaptar as informações aos usuários.

Recentemente a abordagem de reconhecimento dos planos do usuário tem sido melhor enfatizada, onde um plano é uma seqüência de ações de um usuário que conclui uma certa meta. Os sistemas que reconhecem os planos de um usuário observam as ações efetuadas por este usuário e tentam determinar todos os possíveis planos para os quais as ações observadas podem ser adequadas [Kobsa93].

Para a abordagem baseada nas preferências do usuário, pouca atenção havia sido dada, sendo uma das exceções o sistema HAM-ANS [Morik85]. Recentemente, entretanto, consideráveis esforços têm sido gastos na modelagem das preferências do usuário, particularmente nas preferências das informações. Em [Jennings93] foi desenvolvido um servidor eletrônico de notícias personalizado, baseado na arquitetura de redes connexionistas, em [Kaplan93] foi desenvolvido o sistema HYPERFLEX que recomenda informações para os usuários baseados nas suas necessidades e preferências.

Para o modelo de usuário do ARGO, empregamos a abordagem baseada em estereótipos de usuário a partir da idéia proposta por S. K. Chang *et al* em [Chang93], em que o modelo é formado por três elementos: **estereótipos de usuários, assinaturas de usuário e modelos de sistema**. Os motivos que nos levaram a adotar esta abordagem são principalmente: (1) a abordagem de estereótipos de usuários ainda hoje é a mais usada na construção de modelos de usuário; (2) o modelo de usuário de Chang mostrou-se bastante adequado aos nossos objetivos, uma vez que este modelo se propõe a modelar diferentes classes de usuários de BDs, diferentemente das demais referências encontradas na nossa pesquisa bibliográfica.

O modelo de usuário de Chang é usado em [Chang93] para automaticamente fornecer aos usuários de BDs o tipo de representação visual mais adequada (ou seja: tabular, icônica, diagramática ou híbrida) e sua correspondente modalidade de interação. Daqui por diante, e somente por razões de comodidade, nos referiremos a este modelo como **Modelo de Chang**.

Apresentaremos a seguir os elementos usados para a construção do Modelo de Chang, para somente depois apresentarmos o modelo de usuário do ARGO, o qual foi inspirado no Modelo de Chang.

## **7.1 Elementos para a construção do Modelo de Chang**

### **7.1.1 Estereótipos de usuário**

Num modelo de usuário genérico, uma hipótese razoável que pode ser feita é a de que diferentes pessoas acessando um sistema possuem similaridades no que diz respeito ao uso do sistema. Sendo assim, os usuários pertencem a uma única classe. Neste caso, o modelo de usuário é uma coleção de fatos assumidos como verdadeiros para todos os usuários.

Em muitos casos, entretanto, ao invés de se usar um modelo genérico, é mais apropriado considerar um conjunto de modelos, cada um representando uma classe específica de usuários. As características de cada classe de usuários é denominada **estereótipo**.

Em [Chang93], estereótipos são adotados para modelar diferentes classes de usuários de BD. O esquema de classificação adotado está de acordo com o proposto em [Batini91]. Os usuários são divididos em profissionais ou não profissionais, de acordo com o treinamento que eles tiveram. O enfoque foi voltado para as características dos usuários não profissionais. Para tais usuários foram relacionados quatro critérios para a definição do estereótipo, quais sejam: freqüência de interação, repetitividade da consulta, complexidade estrutural da consulta e familiaridade com o BD. Para cada critério um peso foi estipulado: **H**(*high*, alto) e **L**(*low*, baixo).

**Critério 1:** Freqüência de interação do usuário com o sistema:

- Freqüente (H): O usuário tem uma rotina de interação elevada
- Ocasional (L): O usuário interage esporadicamente

**Critério 2:** Repetitividade da consulta:

- Repetitivo (H): Normalmente, as consultas de um usuário são realizadas de acordo com um padrão similar (com relação aos tipos de operadores envolvidos na consulta).
- Improvisado (L): Consultas de formas diversas, sem semelhanças entre elas com relação aos operadores envolvidos

**Critério 3:** Complexidade estrutural da consulta:

- Sofisticado (H): Riqueza de expressão na elaboração das consultas, normalmente por um usuário que tem algum conhecimento de linguagens de consulta analíticas a BD
- Simples (L): As operações necessárias à formulação das consultas são limitadas. Não requerem da linguagem grande poder de expressão

**Critério 4:** Familiaridade com o conteúdo do BD:

- Especialista (H): O usuário tem conhecimento detalhado do conteúdo do BD e da estrutura de dados, expressando-se, normalmente, através de uma linguagem de consulta analítica
- Não-especialista (L): O usuário tem uma idéia geral do conteúdo do BD, mas mesmo assim precisa que o sistema seja amigável para que ele possa “entender” a realidade que lhe é de interesse

### 7.1.2 Assinaturas de usuário

Sempre que um usuário interage com o sistema, é protocolado o conjunto de consultas formuladas por ele. Chamando de  $Q = (Q_1, Q_2, \dots, Q_n)$  a seqüência de consultas formuladas nos tempos  $t_1, t_2, \dots, t_n$ , respectivamente, diz-se que  $Q$  determina o comportamento do usuário, e que de  $Q$  é possível inferir que tipo de padrão as consultas seguem, quão freqüentemente elas são formuladas, e quão complexas elas são. A partir de  $Q$ , o estereótipo do usuário pode ser atualizado. A idéia não é manter todas as histórias das interações, mas apenas uma história compacta (ou seja, as consultas mais recentemente realizadas) que é definida como sendo a **assinatura do usuário**.

### 7.1.3 Modelos de Sistema

Um modelo de sistema é uma representação do conhecimento de um usuário sobre o conteúdo de um BD e sua estrutura, em outras palavras, é a visão que o usuário tem do BD. O sistema constrói um histórico das visões que deve ser analisado de tempos em tempos, de forma a sugerir ao usuário a visão mais apropriada às suas necessidades.

A vantagem de se explorar modelos de sistemas se torna evidente ao se levar em consideração grandes BDs, porque neles o custo de recuperação da informação é muito alto. Contudo, a informação requerida por um usuário individual em muitos casos é limitada a uma parte pequena e específica do BD. A fim de reduzir o custo de tais operações, Chang e Deng propuseram em [Chang90] representar o espaço limitado que o usuário geralmente acessa numa pequena e dinâmica estrutura denominada Visual Net [Chang90]. Esta estrutura é usada por Chang, em seu modelo de usuário, para representar os modelos de sistema.

## 7.2 Modelos de usuário do ARGO

Com o intuito de dinamicamente adaptar o tipo de interface mais apropriada aos usuários do ARGO, especificado no capítulo IV, construímos nosso modelo de usuário inspirado no Modelo de Chang. Contudo, há diferenças entre os dois modelos, e essas diferenças serão explicitadas na discussão das subseções que se seguem.

Do modelo de Chang empregamos apenas os elementos **estereótipos** e **assinaturas** de usuário para a construção do modelo do ARGO, e os conceituamos da seguinte forma:

- **Estereótipos de Usuário:** são usados para classificar as interfaces, ou seja, dependendo do estereótipo em que um usuário se enquadrar, a interface oferecida ao mesmo será: Básica, Mediana, Avançada ou SQL
- **Assinaturas de Usuário:** são representadas pelos históricos gerados durante as interações usuário-sistema, e servirá para atualizar os estereótipos de usuários

Cada um desses componentes será individualmente tratado nas próximas seções, onde serão explicitadas as formas de obtenção do conhecimento sobre as interações, a manutenção dos modelos, e os resultados possíveis de serem obtidos a partir dos mesmos.

### 7.2.1 Definição dos estereótipos de usuário

Sempre que um usuário é cadastrado para acessar o ARGO, a ele é associado um tipo de interface que melhor se adequa (inicialmente) a seus propósitos. Este cadastro inicial é feito sem nenhum conhecimento prévio do comportamento do usuário. Por exemplo, quando um usuário é cadastrado e a ele atribuído a interface Mediana, nenhuma garantia é dada sobre a capacidade ou não do usuário ser realmente do tipo mediano.

À medida em que o usuário interage com o sistema, sua “marca” acaba sendo registrada, através dos históricos. O certo é que o perfil do usuário vai sendo refinado a cada interação e a cada consulta gerada, podendo ser modificado ao longo do tempo, pois o usuário pode interagir de diferentes formas com o BD.

O objetivo não é montar o perfil de cada usuário separadamente, e sim adquirir conhecimento sobre o usuário e enquadrá-lo, a partir do seu perfil, em um dos estereótipos previamente definidos. Nas próximas seções será detalhada a forma de obtenção do conhecimento do perfil dos usuários, e em qual estereótipo se enquadra. Os tipos de estereótipos a serem tratados também serão mais precisamente definidos.

### 7.2.1.1 Critérios para a determinação de um estereótipo

Esta seção trata dos aspectos relativos à determinação dos critérios para a definição de um estereótipo. As diferenças de nossos critérios em relação aos do Modelo de Chang são ressaltadas na Fig. VII.1.

Critérios	Modelo de Usuário de Chang	Modelo de Usuário do ARGO
Frequência de interação	Número de consultas geradas num tempo $T$	Número de consultas corretas geradas num tempo $T$
Repetitividade da consulta	Consultas com padrão similar (com relação à estrutura e aos operadores envolvidos)	Consultas com padrão similar (com relação ao texto-sql gerado, se usa muitas junções ou zero junções)
Familiaridade com o BD	Conhecimento sobre o conteúdo e a estrutura do BD (obtido através de questionário que o usuário deve responder)	Conhecimento sobre o conteúdo do BD (se acessa as mesmas tabelas freqüentemente), sobre a estrutura (se normalmente gera suas consultas via comandos SQL)
Complexidade da consulta	Consultas estruturalmente mais complexas, com relação ao poder expressivo	A complexidade está na formulação da consulta e não necessariamente na estrutura

**Fig. VII.1 – Mapeamento entre os critérios do Modelo Chang e do modelo ARGO**

Para calcular o critério Frequência de Interação, nós consideramos apenas as consultas que foram realizadas com sucesso<sup>12</sup> num espaço de tempo  $T$ , enquanto que no Modelo de Chang são consideradas todas as consultas (com ou sem sucesso). Uma possível razão para que Chang tenha guardado todas as consultas, seria o de fazer experimentos de usabilidade da interface (considerando como um dos parâmetros o número de erros obtidos na especificação de uma consulta). Como não nos propomos a fazer tal experimento, achamos desnecessário manter um histórico contendo todas as tentativas de formulação de uma consulta, com insucessos.

A Repetitividade Estrutural da Consulta é calculada a partir dos padrões de similaridade entre as consultas realizadas por um usuário. Nós detectamos estes padrões de similaridade com base no número de junções de tabelas usadas nas consultas, se o usuário usa muitas (mais do que cinco junções) ou nenhuma junção. Chang detecta esses padrões baseando-se na estrutura das consultas e nos operadores envolvidos nas mesmas.

<sup>12</sup> Nós nos referimos ao termo consultas realizadas com sucesso para as consultas que, ao seu término, produziram um resultado e não uma mensagem de erro.

Nós adotamos a operação junção para detectar as similaridades entre as consultas, a partir da análise do “comportamento” das interfaces. Vejamos o exemplo de um usuário que usa a interface Básica: ele pode ser repetitivo, pois, nesta interface, o acesso a uma tabela normalmente acarretará uma série de junções (implícitas), sem que o usuário possa intervir no uso ou não das mesmas. No caso de um usuário da interface Mediana, ele sempre é repetitivo, mas **pelo** motivo contrário, ou seja, ele nunca realiza consultas usando junção. Já aos usuários das demais interfaces (Avançada e SQL), é dada a liberdade de ser ou não repetitivo, ficando o uso de junções a seu critério.

Saber se um usuário possui Familiaridade com o BD não é muito trivial, tanto que Chang propõe como método para obtenção deste critério o emprego de um questionário, onde nele o usuário assinala (sim ou não) nas questões nele contidas. O objetivo é fazer com que o usuário “apresente” seu conhecimento sobre o conteúdo e a estrutura do BD. Nossa proposta é de obter tal critério através da detecção do quão freqüente é o acesso às mesmas tabelas do BD num dado período, isto para constatar a familiaridade com o conteúdo. No caso de familiaridade com a estrutura, será obtido tal conhecimento a partir do quão freqüentemente o usuário formula suas consultas via comandos SQL.

Chang determina o critério Complexidade da Consulta analisando as consultas geradas pelo usuário, se elas possuem uma estrutura complexa (com emprego de operadores mais sofisticados) ou uma estrutura simples. O nosso critério tem uma conotação diferente do critério de Chang. Isto porque não seria correto afirmar que um usuário que gera consultas contendo, por exemplo, inúmeras junções seja classificado como um usuário experiente, uma vez que consultas geradas na interface Básica podem conter inúmeras junções (implícitas), sem exigir qualquer conhecimento do usuário sobre o fato. O que vai nos permitir identificar se um usuário é simples ou experiente, quanto ao critério Complexidade, é a forma como ele gera suas consultas, por isso dizemos que o cálculo é da **Complexidade de Formulação da Consulta**. Se o usuário é do tipo que sempre obtém suas consultas conjuntivas (ou disjuntivas, ou mistas) explicitamente (via comandos SQL ou visualmente nas janelas apropriadas, botões, etc.), este pode ser identificado como um usuário conhecedor da estrutura relacional do BD e/ou de SQL.

## 7.2.2 Definição das assinaturas de usuário

Como já dito, a assinatura do usuário determina seu comportamento perante a interface, e para permitir que o sistema tome conhecimento desse comportamento é necessário tê-lo registrado em forma de históricos.

Na realidade, os históricos são a representação de dois tipos de *logs*. Um *log* mantém informações sobre as consultas geradas pelo usuário, e este é denominado de *log* de consultas, ele é atualizado a cada nova consulta gerada. Este *log* não será mantido indefinidamente com todas as consultas: descartamos de tempos em tempos as consultas mais antigas. Sugerimos manter este *log* por um ano.

O segundo *log* mantém informações sobre as interações do usuário com o sistema. Por interação, entenda-se a forma de manipular a interface, ou seja, como o usuário acessa as tabelas (via Janela de Esquema, Janela de Referências, Janela de Comando, ou ainda via *links*); se faz seleção de valores, e neste caso, se o faz de forma simples ou múltipla; se faz projeção de campos e qual a forma usada para tal (lembramos que, para cada interface existe uma forma diferente de fazer projeção), e principalmente se faz junção (implícita ou explicitamente). Este *log* também deve ter suas interações mais antigas descartadas.

Para determinar este segundo *log*, é preciso identificar as formas de interação que de fato contribuem para o reconhecimento do comportamento (ou da mudança de comportamento) do usuário. Analisando as interfaces, foi possível observar que pelo menos quatro formas de interação são relevantes: formas de acesso às tabelas, formas de fazer projeção, formas de fazer seleção, formas de fazer junção.

Para facilitar as referências que faremos às formas de interação no decorrer deste capítulo, denominaremos cada uma delas e os seus respectivos valores. Definimos que a letra **F** representa uma forma de interação, seguida de um número que identifica unicamente cada forma, ou seja, **F1** representa a forma de acesso às tabelas, **F2** a forma de fazer projeção, e assim por diante. O mesmo critério é adotado para os valores, sendo que a letra representativa agora passa a ser o **V**, e a numeração será de acordo com a forma de interação a qual o valor pertencer, ou seja: **V11** identifica que é o primeiro valor

da forma de interação F1, no caso acesso via Janela de Esquema. Abaixo são listadas as quatro formas de interação e seus possíveis valores.

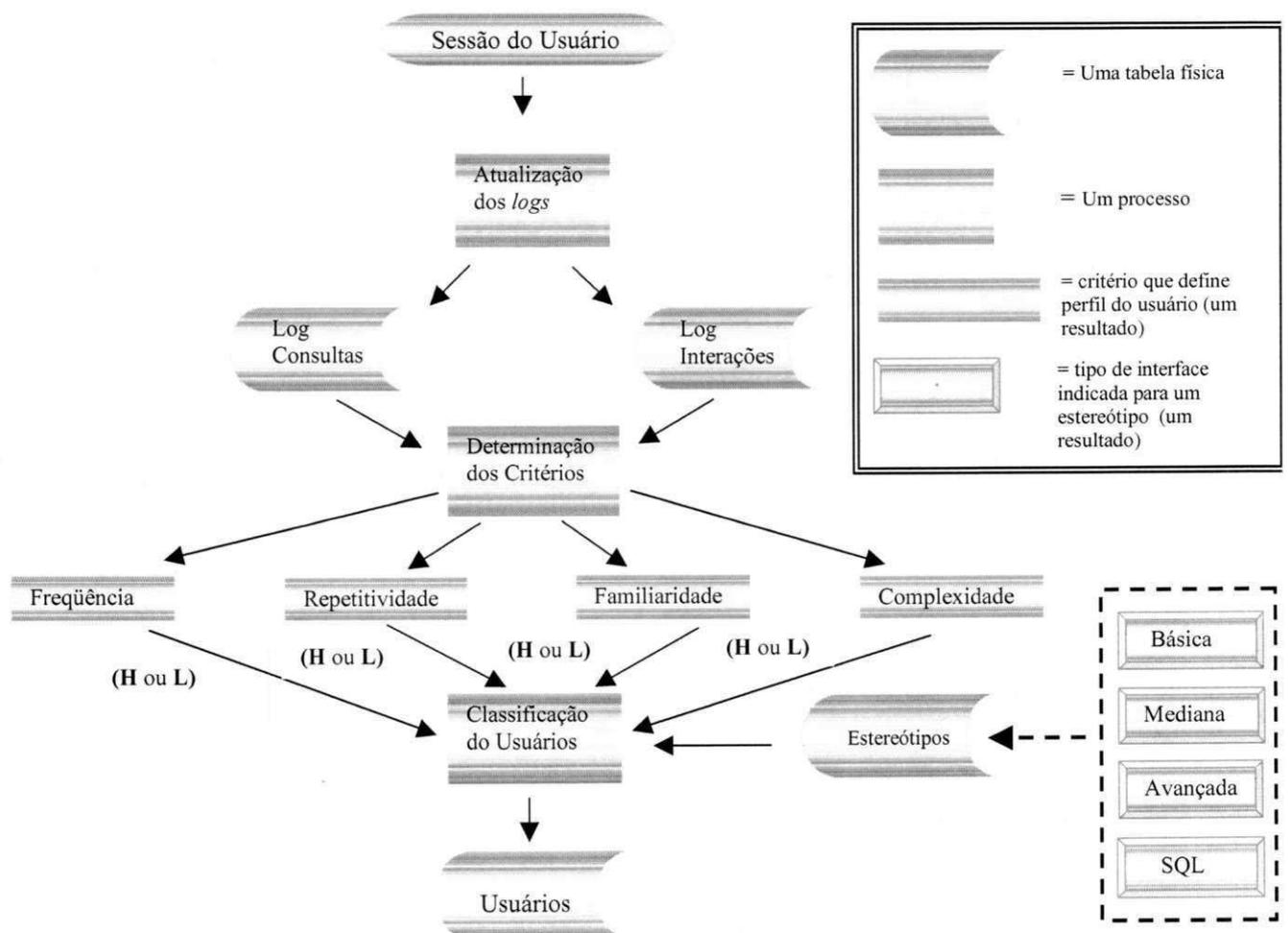
- **(F1) Formas de acessar as tabelas:**
  - (V11) via Janela de Esquema
  - (V12) via *Link*
  - (V14) via Janela de Comando
- **(F2) Formas de fazer projeção de campos:**
  - via Botões para Projeção (adicionar ou retirar)
    - (V21) botão Projeção Adicionar
    - (V22) botão Projeção Retirar
  - (V23) via Janela de Esquema (quando seleciona campos e não tabelas)
  - (V24) via Janela de Comandos
- **(F3) Formas de fazer seleção de valores:**
  - (V31) via botão Pesquisar
  - via Barra de Condição (=, <>, >, <, >=, <=)
    - (V32) condição Simples
    - (V33) condições Múltiplas (através do uso dos botões E ou OU)
  - (V34) via Janela de Comandos
- **(F4) Formas de fazer junção:**
  - (V41) implicitamente (através das informações contidas no meta-banco)
  - (V42) explicitamente (através do uso da barra de condição e, se forem múltiplas junções, dos botões E ou OU, ou via Janela de Comandos)

### 7.3 Construção de um modelo de usuário

A forma de computar inicialmente o modelo de um usuário e suas sucessivas atualizações (e adaptações) é aqui definida. Na realidade, a construção do estereótipo é quem de fato

vai exigir maiores fundamentações. A Fig.VII.2 ilustra os processos envolvidos e os resultados obtidos na construção de um modelo de usuário.

Durante uma sessão, o usuário faz inúmeras interações que resultam em consultas. Tanto as formas de interação quanto as consultas são registradas em históricos (*logs* de consultas e interações, separados), e a partir deles (um ou outro, ou ambos) é possível calcular cada um dos critérios para determinação de um estereótipo. Com base nos valores obtidos em cada um dos critérios (ou seja, H ou L), a tabela da Fig.VII.3 é consultada, e dela é obtida a classificação da interface para o estereótipo obtido.



**Fig. VII.2 – Formação de um estereótipo e classificação da interface para um usuário**

Na tabela da Fig.VII.3 são apresentadas as possíveis combinações para os valores H e L para os critérios que determinam um estereótipo. Considerando todas as possibilidades, dezesseis estereótipos são gerados, desde HHHH (usuário repetitivo, freqüente, sofisticado e especialista) até LLLL (usuário improvisado, ocasional, simples e não-especialista). Dentre estas dezesseis possibilidades, no entanto, existem quatro que são logicamente improváveis de ocorrer, são elas: HLHH, HLLH, LLHH e LLLH, já que nestas quatro possibilidades o valor do segundo critério (freqüência de interação) é L, enquanto que o valor do quarto critério (familiaridade com o BD) é H, o que nos levaria a concluir que um usuário pouco freqüente não possuía, também, familiaridade como BD. Mas, como o critério freqüência é calculado com base no *log* de consultas num período *T* qualquer (logo, somente as consultas deste período são consideradas), e o critério familiaridade é calculado a partir do *log* de consultas completo (ou seja, todas as consultas do *log* são consideradas), é possível que um usuário tenha diminuído sua freqüência naquele período (*T*), sem no entanto ter perdido a familiaridade com o BD. Por esta razão, tratamos todas as possibilidades como possíveis de ocorrer.

Complexidade/Familiaridade Repetitividade/Freqüência	H / H	H / L	L / H	L / L
H / H	SQL	Avançada	Mediana	Básica
H / L	SQL	Avançada	Mediana	Mediana
L / H	SQL	Avançada	Básica	Básica
L / L	SQL	Avançada	Avançada	Básica

**Fig. VII.3 – Tabela de classificação das interfaces de acordo com o estereótipo**

Para cada uma das possibilidades, foi atribuída a interface mais adequada. O preenchimento da tabela de estereótipo foi feito a partir da análise das funcionalidades das interfaces, com relação aos possíveis valores (H ou L) que os critérios podem vir a assumir em cada uma das interfaces. Por exemplo, a interface SQL só pode ser atribuída ao estereótipo que possuir os critérios Complexidade e Familiaridade com valor H; a interface Mediana só pode ser atribuída ao estereótipo que possuir o critério Repetitividade com valor H e Complexidade com valor L. As interfaces Básica, Avançada e SQL, com relação ao critério Repetitividade, são um pouco mais flexíveis

(cada uma a sua maneira): o critério pode obter tanto valor H quanto o L, pois, na Básica as junções implícitas ocorrem dependentes das definições no metabanco e dos relacionamentos existentes entre as tabelas, sendo assim, não é possível afirmar que sempre haverá muitas junções ou zero junções; na Avançada (assim como na SQL), como o usuário tem liberdade total para fazer ou não suas junções, é também impossível fazer tal afirmação. Ainda com relação a interface Básica, pode-se afirmar que o critério Complexidade sempre irá obter o valor L. E assim por diante.

Esta tabela será usada sempre que for feita a atualização dos modelos de usuário. A cada atualização os valores para cada critério são obtidos Dependendo dos valores encontrados o estereótipo do usuário tanto pode ter evoluído (passar de uma interface simples para uma mais sofisticada) como involuído (passar de uma interface sofisticada para uma mais simples), dessa forma, o ARGO se adapta ao estereótipo do usuário fornecendo ao mesmo a interface que lhe é mais adequada ao seu estereótipo atual.

O período para atualização dos estereótipos deverá ser previamente definido. Estamos propondo que seja a cada 30 dias, para todos os usuários. Na nossa proposta de atualização dos modelos foi preciso definir algumas dimensões para a classificação dos estereótipos, ou seja:

- por **frequente** entenda-se a realização de pelo menos **100 (x)** consultas no período
- por **repetitivo** entenda-se a constatação de que pelo menos **60% (y)** das consultas do *log* de consultas (do período de 30 dias) possuem um padrão similar
- por **especialista** (à respeito da familiaridade) entenda-se a constatação de que no mínimo **40% (n)** das tabelas do BD são consultadas, cada uma delas, pelo menos **50% (z)** das vezes (a partir do *log* de consultas completo e não apenas do período de 30 dias); ou, o número de consultas geradas via comandos SQL tenha sido pelo menos **60% (k)** no período (este por sua vez obtido a partir do *log* das interações)

Estas dimensões foram definidas sem nenhum rigor, uma vez que seriam necessários experimentos para validar tais valores com exatidão e segurança.

Nas subsecções seguintes tratamos dos detalhes a serem considerados no cálculo de cada um dos critérios para definição de um estereótipo. Antes, apresentamos a estrutura das

tabelas necessárias à implementação das idéias propostas, cuja responsabilidade é do módulo gerenciador dos modelos de usuário (um dos módulos da arquitetura do ARGO).

### **7.3.1 Estrutura do módulo gerenciador dos modelos de usuário**

O módulo gerenciador dos modelos de usuários obtém dos históricos gerados pelo módulo gerenciador das interfaces as informações necessárias para o cálculo dos critérios que determinam os estereótipos de usuários. Como já sabemos, os históricos representam as assinaturas de usuários, e são formados pelos *logs* de consultas e interações (duas tabelas distintas). O *log* de consultas foi especificado na seção 6.2.3. O *log* das interações é importante e necessário na determinação dos estereótipos de usuários, assim como as tabelas de estereótipos e de classificação das interações. Estas tabelas serão especificadas nas subseções seguintes.

#### **7.3.1.1 Tabela de Interações**

Nesta tabela são registradas as formas de interação do usuário consideradas relevantes na atualização do modelo de usuário. Cada uma das formas de interação (ver na seção 7.2.2, a definição dessas formas de interação) representa um campo da tabela de interações. Apenas as formas de interação seleção e projeção são representadas por um mesmo campo, já que estes valores nunca serão concomitantemente registrados. A estrutura desta tabela é:

*(cod\_inter, login\_usuario, forma\_acesso\_tabela, forma\_projeção\_seleção,  
forma\_junção)*

O preenchimento desta tabela é em tempo real. A cada interação do usuário um registro novo é gerado, ou seja, se uma tabela é selecionada na Janela de Esquema (não importando a interface usada para tal), um registro neste *log* é gerado e contém: (*<cod\_inter>*, *<login\_usuario>*, 'V11', *null*, *null*). Como está descrito na seção 7.2.2, a forma de acesso à tabela "V11" significa que uma tabela foi acessada via Janela de Esquema.

### 7.3.1.2 Tabela de Estereótipos

Esta tabela é formada pelos campos identificadores dos critérios que definem os estereótipos de usuário e do tipo da interface apropriada. Sua estrutura é a seguinte:

*(critério\_Repet, critério\_Freq, critério\_Complex, critério\_Famil, tipo\_interface)*

Os valores para esta tabela são fixos e previamente definidos.

### 7.3.1.3 Tabela de Classificação das interações

A tabela de classificação das interações também tem seus valores fixos e previamente definidos. Ela é necessária no cálculo do critério Complexidade da formulação da consulta. Sua estrutura é a seguinte:

*(forma\_acesso\_tabela, forma\_projeção\_seleção, forma\_junção, classificação)*

## 7.3.2 Cálculo dos critérios que determinam um estereótipo

### 7.3.2.1 Critério freqüência de interação

A freqüência de interação é calculada a partir do *log* das consultas, pois se baseia no número de consultas geradas no período. A idéia é obter todas as consultas geradas, por usuário, durante as sessões realizadas no período estipulado (no caso, nos últimos 30 dias). Se a ocorrência for inferior a  $x$  o critério freqüência é **L**, caso contrário é **H**. O algoritmo a ser usado para fazer este cálculo é o seguinte:

*para cada sessão do usuário no período*

*enquanto houver consultas, no log de consultas, para tal usuário faça*

*acumular número total de consultas do período (consulta\_período)*

*se consulta\_período < x então*

*critério\_freqüência := L*

*senão*

*critério\_freqüência := H*

### 7.3.2.2 Critério repetitividade da consulta

A repetitividade da consulta é calculada com base no *log* de consultas, já que são comparados os textos SQL nele contidos para encontrar similaridades. Os padrões de similaridades que nós definimos são:

- consultas com **zero** junções, em que nenhuma comparação entre campos de tabelas distintas é detectada
- consultas com **muitas** junções, possuem **mais de cinco** comparações entre campos de tabelas distintas

A partir do *log* de consultas o algoritmo calcula o percentual de ocorrência de cada padrão de similaridade, por usuário, no período de 30 dias. Se o percentual for igual ou superior a  $y\%$ , o critério de repetitividade é **H**, caso contrário é **L**. O algoritmo usado é o seguinte:

```
para cada consulta do usuário em questão no log de consultas do período
    acumular a quantidade de consultas (quantidade_consulta)
enquanto houver consultas no log de consultas do período, para o usuário em questão faça
    se não tem cláusula WHERE no texto-SQL então
        acumular pontos para o padrão zero_junções
    se sim então
        início
            acumular número de junções obtidos na consulta (qt_junções)
            caso qt_junções
                zero junções: acumular pontos para zero_junções
                mais de cinco junções: acumular pontos para muitas_junções
        fim se
    fim enquanto
para cada padrão de similaridade computado, com base no total de registro no log do período
    calcular o percentual de ocorrência de cada padrão
para cada um dos percentuais computados
    se um deles possuir valor maior ou igual a  $y\%$ 
        critério_repetitividade := H
```

*senão*

*critério\_repetitividade := L*

### 7.3.2.3 Critério familiaridade com o BD

Este critério é calculado por usuário e BD através do *log* de consultas completo (e não apenas no período de 30 dias), já que o objetivo é detectar o acesso freqüente (de **z%**) às mesmas tabelas de um BD para um percentual de **n%** das tabelas do BD. Apenas no caso de ter sido obtido o valor **L**, após o *log* de consultas ter sido analisado, o *log* de interações é consultado para que a quantidade de consultas geradas (no período de 30 dias), a partir de comandos SQL na Janela de Comandos, seja obtido. Se esta quantidade for igual ou superior a **k%**, o valor obtido é **H**, ou seja este é um usuário especialista, caso contrário, o valor obtido é **L**.

Primeiramente, o *log* de consultas é ordenado por usuário, BD, tabela, nessa ordem. O número total de acessos às tabelas de cada BD é obtido, em seguida é feito o cálculo proporcional do número de acessos a uma única tabela com relação ao número de acessos total ao BD, isto para cada BD e tabelas. Se em pelo menos um dos BDs consultados, **n%** das tabelas tiver sido consultada **z%** das vezes, o algoritmo é encerrado com o valor para o critério igual a **H**, caso contrário, o algoritmo continua para o usuário em questão e demais BDs por ele consultado. Se mesmo assim não tiver sido obtido o valor **H**, só então o *log* de interações é consultado para se obter a quantidade de consultas geradas via Janela de Comandos, se esta quantidade for menor que **k%** o valor do critério é **L**, caso contrário é **H**. O algoritmo usado é o seguinte:

*ordenar o log de consultas por usuário, BD, tabela*

*para o primeiro BD do usuário em questão no log de consultas completo*

*acumular o número total de acessos a este BD (qt\_acessos\_BD)*

*obter quantidade de tabelas existente no BD atual (num\_tabelas\_BD)*

*enquanto existir no log de consultas registro para o usuário em questão faça*

*início*

*se for o mesmo BD então*

*se for a mesma tabela então*

*acumular a quantidade de acessos a uma mesma tabela/BD (qt\_acessos)*

*senão (se mudou de tabela) então*

**início**

*calcular* o percentual de acessos da tabela com relação a *qt\_acesso\_BD* e *qt\_acessos* (*pct\_ocorrência\_tabela*)

*se* *pct\_ocorrência\_tabela*  $\geq$  *z%* **então**

*acumular* quantidade de tabelas (*num\_tabelas*)

**fim se**

**senão** (*se mudou de BD*) **então início**

*obter* quantidade de tabelas existente no BD atual (*num\_tabelas\_BD*)

*obter* o número total de acessos ao BD atual (*qt\_acessos\_BD*)

*se* *num\_tabelas*  $\geq$  (*num\_tabelas\_BD* \* *n%* / 100) **então**

*critério\_familiar* := **H**

*parar* algoritmo

**fim se**

**fim enquanto**

*se* *num\_tabelas*  $\geq$  (*num\_tabelas\_BD* \* *n%* / 100) **então** (*para o último BD lido*)

*critério\_familiar* := **H**

**senão** /\* *checar o log das interações* \*/

**início**

*enquanto* existir no log de interação (*dos últimos 30 dias*) registro do usuário **faça**

**início**

*acumular* a quantidade de interações (*qt\_interação\_período*)

*acumular* a quantidade de vezes que a Janela de Comando foi usada (*qt\_comando*) no período

**fim enquanto**

*se* (*qt\_comando* \* 100% / *qt\_interação\_período*)  $\geq$  *k%* **então**

*critério\_familiar* := **H**

**senão**

*critério\_familiar* := **L**

**fim se**

#### 7.3.2.4 Critério complexidade de formulação da consulta

Na tabela da Fig.VII.4 são apresentadas as possíveis combinações de ocorrências das interações e para cada uma delas a classificação adequada (Simples ou Sofisticada). Com

base nesta tabela e no *log* das interações é feito o cálculo do critério Complexidade. A tabela foi preenchida de acordo com a codificação das formas de interações apresentada na seção 7.2.2.

Interação acesso à tabela	Interação projeção ou seleção	Interação Junção	Classificação da interação
V11	Null	V41	Simple
V11	Null	Null	Simple
V11	Null	V42	Sofisticada
V11	V21	Null	Simple
V11	V22	Null	Simple
V11	V23	Null	Simple
V11	V31	Null	Simple
V11	V32	Null	Simple
V11	V33	Null	Sofisticada
V12	Null	Null	Simple
V12	V21	Null	Simple
V12	V22	Null	Simple
V12	V31	Null	Simple
V12	V32	Null	Simple
V13	Null	Null	Simple
V13	Null	V42	Sofisticada
V13	V23	Null	Simple
V13	V31	Null	Simple
V13	V32	Null	Simple
V13	V33	Null	Sofisticada
V14	(Não importa)	(Não importa)	Sofisticada

**Fig. VII.4 – Tabela de classificação das interações do usuário**

O processo de classificação da complexidade se dá em três etapas. A primeira etapa é a extração dos registros do *log* de interações correspondentes aos últimos 30 dias (período para cálculo do estereótipo) que serão usados como uma amostragem para classificação; a segunda etapa é a classificação desses registros (da amostragem) segundo a tabela de classificação das interações (ver Fig.VII.4); e a terceira e última etapa é a própria classificação da complexidade, a partir da amostragem. Cada uma dessas etapas será detalhada a seguir.

A extração dos registros a serem usados para classificar a complexidade é feita a partir do *log* de interações ordenado pelos atributos: usuário, forma\_acesso\_tabela, forma\_projeção\_seleção e forma\_junção, nessa ordem. Tomemos como exemplo o *log* de interações dos usuários Maria e João, apresentados na Fig.VII.5, para explicar a determinação do critério Complexidade.

Cod-Inter	Usuário	Forma de acesso à tabela	Forma de projeção/seleção	Forma de junção
*	João	V11	Null	V41
*	João	V11	V22	Null
*	João	V11	Null	V41
*	João	V12	V32	Null
*	João	V11	Null	V41
*	João	V11	V22	Null
*	João	V11	Null	V41
*	João	V11	V22	Null
*	João	V11	V31	Null
*	João	V11	V22	Null
*	Maria	V11	Null	Null
*	Maria	V12	Null	Null
*	Maria	V13	V33	Null
*	Maria	V13	V33	Null
*	Maria	V13	V33	Null
*	Maria	V13	V33	Null

**Fig.VII.5 – Exemplo do log de interações de dois usuários fictícios**

Ao ordenar a tabela da Fig.VII.5, obtemos a tabela da Fig.VII.6. Percebemos na Fig.VII.6 que alguns registros possuem valores iguais para os campos login\_usuario, forma\_acesso\_tabela, forma\_projeção\_seleção, forma\_junção. Agrupando estes registros iguais obtemos vários conjuntos (cada conjunto foi identificado pela letra C seguida de um número: C1, C2, C3, ...).

Cod-Inter	Usuário	Forma acesso à tabela	Forma projeção seleção	Forma junção
*	João	V11	Null	V41
*	João	V11	Null	V41
*	João	V11	Null	V41
*	João	V11	Null	V41
*	João	V11	V22	Null
*	João	V11	V22	Null
*	João	V11	V22	Null
*	João	V11	V22	Null
*	João	V11	V31	Null
*	João	V12	V32	Null
*	Maria	V11	Null	Null
*	Maria	V12	Null	Null
*	Maria	V12	Null	Null
*	Maria	V13	V33	Null
*	Maria	V13	V33	Null
*	Maria	V13	V33	Null
*	Maria	V13	V33	Null

→ C1

→ C2

→ C3

→ C4

→ C5

→ C6

→ C7

**Fig.VII.6 – Exemplo do log das interações ordenado por usuário, forma de acesso à tabela, forma de projeção/seleção e forma de junção**

O conteúdo do campo `cod_inter` na Fig.VII.5 e Fig.VII.6 é um asterisco (\*), indicando que o valor destes campos não tem importância na classificação.

De cada conjunto da tabela ordenada (Fig.VII.6) é extraído um registro identificador (amostragem do conjunto), sendo a cada um destes registros atribuído um peso, ou seja, como um conjunto pode ser maior do que outro, é necessário dizer o quanto cada conjunto “pesa” em relação aos registros da tabela no período. Este peso é o número de registros contidos em cada conjunto (Fig.VII.7).

Para cada registro de amostragem obtido, é feita a classificação dos mesmos com base na tabela de classificação das interações (se Simples ou Sofisticado). Uma tabela temporária irá conter estes registros de amostragem, com classificação e peso, e será tratada daqui para frente como sendo um conjunto de treinamento, ou seja, um conjunto de exemplos de entrada a ser usado na classificação do critério Complexidade.

Usuário	Forma acesso à tabela	Forma de projeção seleção	Forma de junção	Classificação	Peso
João	V11	Null	V41	Simples	4
João	V11	V22	Null	Simples	4
João	V11	V31	Null	Simples	1
João	V12	V32	Null	Simples	1
Maria	V11	Null	Null	Simples	1
Maria	V12	Null	Null	Simples	2
Maria	V13	V33	Null	Sofisticado	4

**Fig.VII.7 – Amostragem das interações classificadas e com os pesos atribuídos**

Para o conjunto de treinamento obtido (Fig.VII.7), é feita a ordenação por usuário e classificação; é feito o somatório dos pesos dos registros pertencentes a uma mesma classificação, por usuário; em seguida é feita a comparação: se a soma dos pesos para a classificação Simples for maior, a complexidade é **L**, caso contrário a complexidade é **H**. O algoritmo usado para fazer a classificação da Complexidade é o seguinte:

*ordenar o log de interações do período (de 30 dias) por usuário, forma\_acesso\_tabela, forma\_projeção\_seleção, forma\_junção*

*enquanto existir registro no log de interações do período para o usuário em questão faça se os valores dos registros forem iguais para os quatro campos usados na ordenação do log de interações então*

*acumular peso dos registros iguais*

*senão*

*início*

*obter a classificação do registro anterior a partir da tabela de classificação das interações*

*adicionar ao conjunto de treinamento (tabela temporária) o registro anterior com o peso obtido e a classificação*

*fim se*

*fim enquanto*

*ordenar o conjunto de treinamento por usuário e classificação*

*somar peso dos registros pertencentes a um mesmo usuário e classificação*

*se soma\_peso\_simples >= soma\_peso\_sofisticado então*

*critério\_complexidade := L*

*senão*

*critério\_complexidade := H*

Para o conjunto de treinamento obtido na Fig.VII.7, o resultado da aplicação do algoritmo para o usuário João é **L**, já que ele só obteve classificação Simples; para o usuário Maria é **H**, pois o somatório dos pesos para a classificação Sofisticada foi um ponto maior que a Simples.

### 7.3.3 Cálculo final do estereótipo

Após todos os algoritmos para obtenção dos critérios de classificação do estereótipo terem sido executados, os critérios frequência, repetitividade, familiaridade e complexidade, terão recebido **H** ou **L** como resultado.

A tabela de estereótipos é consultada a partir desses valores, e um tipo de interface será encontrado para os mesmos. Com isso, a tabela de usuário é atualizada apenas atualizando o valor do campo tipo\_interface. A partir desse momento, sempre que o usuário acessar o ARGO a interface que lhe foi atribuída será apresentada.

O algoritmo chamador dos algoritmos gerados para cada critério é o seguinte:

*para cada usuário calcular os critérios chamando os respectivos algoritmos*

*chamar algoritmo critério repetitividade (Rep)*

*chamar algoritmo critério frequência (Freq)*  
*chamar algoritmo critério complexidade(Comp)*  
*chamar algoritmo critério familiaridade (Fam)*  
*para o estereótipo (Rep+Freq+Comp+Fam) encontrado*  
*obter da tabela de estereótipos a interface apropriada*  
*atualizar o campo interface da tabela usuário para o usuário em questão*

## 7.4 Exemplos de mudança de estereótipo

**Exemplo 1:** Tomemos como exemplo o usuário João, usado na seção 7.3.4. Supondo que para este usuário, na última avaliação do seu estereótipo, foram obtidos os valores **HLL** para os critérios repetitividade da consulta, frequência de interação, complexidade na formulação das consultas e familiaridade com o conteúdo ou estrutura do BD, respectivamente. Neste caso, a interface indicada como adequada à João foi a **Básica** (segundo a tabela de classificação das interfaces da Fig.VII.3).

No entanto, João passou a usar mais frequentemente o ARGO, gerando muito mais que  $x$  consultas num período de um mês. Observamos, também, pelo *log* das interações que João usou muitas condições múltiplas (para fazer seleção), através do uso dos botões E ou OU, e também maior número de acesso às tabelas de referências (via Janela de Referências).

No novo cálculo do estereótipo, percebemos as mudanças claramente para os critérios frequência, que embora anteriormente já tivesse sido classificado como **H**, ele aumentou consideravelmente; e o critério complexidade, que antes havia sido classificado como **L**, nessa nova atualização passou a ser **H**, pois suas interações classificadas como Sofisticada, obtiveram o “peso” maior do que as classificadas como Simples. Portanto, o usuário João passou a se enquadrar no estereótipo **HHLL**, e a interface indicada para esse novo estereótipo é a **Avançada**.

Este exemplo dá a idéia da evolução de um estereótipo. Como o ARGO é ambiente adaptativo (evolutivo + involutivo), apresentaremos a seguir um exemplo em que ocorre uma involução de um estereótipo.

**Exemplo 2:** Para este exemplo tomemos o usuário Maria, que antes havia sido classificado com um estereótipo adequado para a interface Avançada, qual seja: **LHHL**.

Quando um novo cálculo do estereótipo foi realizado foram observadas algumas mudanças no “comportamento” deste usuário. Observou-se, por exemplo, que as consultas geradas por Maria, em grande maioria, não continham junções, constatando-se inclusive um percentual de maioria superior a **y%**. Também: o acesso as mesmas tabelas de um mesmo BD (num percentual de **n%**) foi muito superior a **z%**. Um outro fato importante foi observado com relação às formas de elaboração das consultas, pois Maria passou a usar bem menos a Janela de Referências, passando a realizar muito mais interações consideradas Simples do que as consideradas Sofisticadas.

Podemos concluir deste exemplo que Maria teve, nos últimos 30 dias, uma melhora no seu conhecimento sobre o conteúdo de algumas tabelas do BD, obtendo, desta forma mais familiaridade com o BD (logo o critério familiaridade recebeu valor **H**); obteve, também, maior número de consultas similares (com relação às junções) (o critério repetitividade também recebeu valor **H**); contudo, perdeu em complexidade de formulação das consultas, pois obteve mais consultas formuladas de forma Simples do que Sofisticada (o critério complexidade recebeu valor **L**).

O novo estereótipo calculado para Maria, portanto, foi: **HHLH**, sendo para este estereótipo adequado a interface **Mediana**.

# Capítulo VIII

## Conclusões e Perspectivas

### 8.1 Considerações Finais

Nesta dissertação apresentamos um ambiente de consulta visual para acesso a banco de dados relacionais, denominado ARGO, que utiliza mecanismos de manipulação direta, ou seja, usa uma forma rápida, reversível e incremental de apresentar as informações durante a interação usuário-sistema. O ambiente proposto se enquadra na definição de *easy-to-use interfaces*, uma vez que o enfoque foi concentrado no objetivo de possibilitar aos usuários uma interação mais fácil com o sistema. Nesta ambiente, o fator portabilidade e acesso a bases de dados de forma genérica estão embutidos, além do fato de, a partir do reconhecimento das formas de interação usuário-BD, este ambiente poder se adaptar ao usuário, fornecendo-lhe a interface mais adequada .

O ARGO fornece quatro tipos de interfaces (Básica, Mediana, Avançada e SQL), cada uma delas se adequando a um tipo específico de usuário, iniciante, mediano, experiente e especialista, respectivamente. As consultas realizadas possuem padrões (de texto SQL) diferenciáveis de uma interface para outra, por isto, é possível reconhecer uma interface a partir das consultas nela geradas. Também, o poder de expressão das

consultas realizadas em cada interface é diferente, a interface SQL é a que possui maior poder de expressão, em seguida a Avançada, e assim sucessivamente.

Na Fig.VIII.1, apresentamos uma tabela comparativa dos quatro tipos de interfaces. O objetivo desta tabela é deixar claro as diferenças ou semelhanças existentes entre cada uma das interfaces. Para isso, relacionamos numa coluna os tipos de operações normalmente usadas nas consultas (junções, seleções, projeções, ordenação, etc.), nas demais colunas os tipos de interfaces (Básica, Mediana, Avançada, SQL). Para cada interface é apresentada a forma de uso da operação (caso seja permitido).

Operações	Básica	Mediana	Avançada	SQL
Uso de junções	Inúmeras (equijunção)	Não permite	Pode usar ou não (equijunção ou <i>tetha</i> -junção)	Pode usar ou não (quaisquer que sejam)
Seleção por valores exatos	Apenas conjuntivas (implícitas)	Apenas conjuntivas (implícitas)	Conjuntivas ou disjuntivas (implícitas ou explícitas)	Qualquer condição (explícitas)
Projeção de campos	Não permite	Irrestrito	Irrestrito	Irrestrito
Pesquisa por padrão	Permitido apenas para campos do tipo <i>string</i>	Permitido apenas para campos do tipo <i>string</i>	Permitido apenas para campos do tipo <i>string</i>	Permitido
Ordenação	Nas formas ascendente e descendente	Nas formas ascendente e descendente	Nas formas ascendente e descendente	Nas formas ascendente e descendente
Uso de funções embutidas	Não permite	Não permite	Não permite	Permite

**Fig.VIII.1 – Comparação das principais funcionalidade das interfaces**

Podemos perceber, pela Fig.VIII.1, que pode não haver coerência entre as consultas obtidas a partir de diferentes interfaces (umas permitem junção, outras não; umas só permitem equijunção, outras permitem também *tetha*-junção; etc.). Por isso, não permitimos que consultas geradas numa certa interface possam ser executadas em qualquer uma das outras interfaces.

As interfaces Básica e Mediana são estritamente direcionadas para usuários iniciantes e/ou ocasionais, por não exigirem nenhum conhecimento do usuário sobre o esquema conceitual do BD. Embora a interface Avançada possibilite um ambiente amigável para o usuário, para que todo o seu poder de expressão seja usado é necessário que o usuário

tenha um certo conhecimento sobre o BD (a estrutura das tabelas e seus relacionamentos), podendo exigir nesses caso um treinamento adequado para os usuários. De qualquer maneira, as três interfaces (Básica, Mediana e Avançada) são dirigidas para usuários não-experientes. Já a interface SQL é inteiramente direcionada a usuários experientes, de forma a generalizar o uso do ARGO a todo tipo de usuário.

Um aspecto importante do ambiente ARGO é o seu caráter adaptativo, no sentido de que ele “aprende” a conhecer o usuário a partir das interações usuário-BD, podendo desta forma, apresentar ao usuário a interface mais apropriada ao seu estereótipo. Para isto, fazemos uso de modelos de usuário apropriados. O ARGO se adapta para acompanhar o comportamento do usuário, às suas formas de interações e ao estilo das consultas por ele geradas.

As interfaces do ARGO foram especificadas e implementadas, e alguns testes foram realizados usando para tal bases de dados em Oracle, Interbase e Paradox. O modelo de usuário, embora não tenha sido implementado, foi totalmente especificado, incluindo os algoritmos mais importantes para a obtenção dos estereótipos de usuário.

Ainda a respeito da especificação do modelo de usuário, vale salientar que existe a necessidade de se fazer experimentos com as interfaces do ARGO para validar as dimensões dos valores propostos na determinação do estereótipo de usuário, ou seja, os percentuais adotados para se calcular os critérios Frequência de interação, Repetitividade da consulta e Familiaridade com o BD. Neste experimento devem ser envolvidos tipos de usuário compatíveis aos que foram definidos neste trabalho, ou seja, Iniciante, Mediano, Experiente e Especialista.

## 8.2 Perspectivas de Trabalhos Futuros

Possíveis extensões desse trabalho podem ser realizadas. Podemos enumerar:

- ↪ Proporcionar o emprego completo dos operadores da linguagem relacional SQL, como negação (NOT), existência (EXISTS) e agrupamento (GROUP BY e HAVING), dentre outros;
- ↪ Poder usar as funções embutidas do SQL (AVG, MIN, MAX, SUM, COUNT);
- ↪ Permitir, na interface Avançada, que consultas envolvendo junção de uma tabela com ela própria (auto-junção) possa ser especificada (exemplo: obter os empregados que ganham mais que seus gerentes);
- ↪ Possibilitar que a operação de equijunção possa ser feita na interface Mediana; isso incrementaria sobremaneira a capacidade de “navegação” da interface, já que mais *links* poderiam ser disponibilizados;
- ↪ Implementar os algoritmos propostos para a obtenção dos estereótipos de usuário;
- ↪ Fornecer “inteligência” ao ARGO para que ele apresente as tabelas mais freqüentemente acessadas por um usuário, além da interface apropriada, sempre que o usuário abrir uma nova sessão no ARGO
- ↪ Fazer experimentos com as interfaces, podendo, desta forma, obter valores mais adequados para as dimensões utilizadas no cálculo de alguns dos critérios de determinação de um estereótipo

# Referências Bibliográficas

- [**Batini91**] Batini, C., Catarci, T., Costabile, M. F., Levialdi, S. *Visual Query Systems*. Technical Report N.04.91 of Dipartimento di Informatica e Sistemistica, University of Rome “La Sapienza”, 1991.
- [**Bryce86**] Bryce, D., Hull, R. *SNAP: A Graphis-based Schema Manager*, in Proc. of the 2<sup>nd</sup> International Conference on Data Engineering. Los Angeles, p. 151-164, 1986.
- [**Carey96**] Carey, M., Haas, L., Maganty, V., Williams, J. *PESTO: An Integrated Query/Browser for Object Database*. Proc. of the 22<sup>nd</sup> VLDB Conf. Mumbai (Bombay), India, 1996.
- [**Catarci96**] Catarci, T., Chang, S. K., Costabile M. F., Levialdi, S., Santucci, G. *A Graph-based Framework for Multiparadigmatic Visual Access to Databases*. To appear in IEEE Transaction on Data and Knowledge Engineering, 1995-96
- [**Chang90**] Chang, S. K., Deng, Y. *Intelligente Database Retrieviel by Visual Reasoning*. In Proc. IEEE COMPSAC Conference, pp 459-464, Chicago, October 1990.

- [**Chang93**] Chang, S. K., Costabile, M. F., Levialdi, S. *A Framework for Intelligent Visual Interface Design for Database Systems*. In proc. Of the Int'l Workshop on Interfaces to Database Systems (IDS92), R.Cooper ed., Springer-Verlag, pp. 377-391, Glasgow, 1993
- [**Date91**] Date, C. J. *Introdução a Sistemas de Bancos de Dados*. Ed. Campus, (tradução da 4ª edição original), 1991.
- [**Dix95**] Dix, A., Patrick, A. *Query by Browsing*. In Sawier P., editor, *Interfaces to Database Systems*, Lancaster 1994, Workshop in Computing, Springer-Verlag, London, pages 236-248, 1995.
- [**Ellis95**] Ellis, G. P., Finlay, G. E., Pollitt, A. S. *HIBROWSE for Hotels: Bridging the Gap between User and System Views of a Database*. In Sawier P., editor, *Interfaces to Database Systems*, Lancaster 1994, Workshop in Computing, Springer-Verlag, London, pages 46-92, 1995.
- [**Errico97**] Errico, Bruno. *Intelligent Agentes and User Modelling*. Dottorato di Ricerca in Informatica. Università Degli Studi di Roma "La Sapienza", 1997.
- [**Hend77**] Hendrix, G. *Human Engineering for Applied Natural Language Processing*. In: IJCAI-77: Proc. of the 5<sup>th</sup> International Joint Conference on Artificial Intelligence, pp. 183-191, 1977.
- [**Hietala93**] Hietala, P., Nummenmaa, J. *A Multimodal Database User Interface and Framework Supporting User Learning and User Interface Evaluation*. In Cooper R., editor, *Interfaces to Database Systems*, Glasgow 1992, Workshop in Computing, Springer-Verlag, London, pages 392-404, 1993.

- [Jennings93]** Jennings, A., Higuchi, A. *A User Model Neural Network for Personal News Service*. User Modeling and User-Adapted Interaction 3:1-25, 1993
- [Kaplan93]** Kaplan, J., Chen, J. *Adaptative Hypertext Navigation Based on User Goal and Context*. User Modeling and User-Adapted Interaction 3:193-220, 1993
- [Kobsa93]** Kobsa, Alfred. *User Modeling: Recent Work, Prospects and Hazards*. M. Schneider-Hufschmidt, T. Kühme and U. Malinowski (eds.): Adaptive User Interfaces: Principles and Practice. North-Holland, Amsterdam, 1993 (<http://zeus.gmd.de/~kobsa/papers/1993-aui-kobsa.ps>)
- [Littlehales92]** Littlehales, J., Hancox, P. *The Problems of Integrating Interfaces to Publicly Available Databases*. Proc. of the first International Workshop on Interfaces to Database Systems, Glasgow, 1-3 July 1992
- [MacDonald75]** MacDonald, N., Stonebraker, M. *CUPID: A User Friendly Graphics Query Language*, im Proc. ACM-PACIFIC, p. 127-131, April 1975
- [Mohan90]** Mohan, L. *A Framework for Building Knowledge-Intensive Data Models*, in Proc. 2th International Conference on Software Engineering & Knowledge Engineering, p. 33-38, Junho 1990
- [Morik85]** Morik, K. *User Modelling, Dialogue Structures, and Dialogue Strategy in HAM-ANS*. In Proc. of the 2<sup>nd</sup> EACL, pp. 268-273, Geneva, Switzerland, 1985
- [Pollitt86]** Pollitt, A. S. *Query by Menu: A novel DBMS Query Language, a Description and Comparison with QBE*. 8<sup>th</sup> BCS IRSG Research Colloquium on Information Retrieval, University of Strathclyde, 1986

- [Pollitt93]** Pollitt, A. S., Ellis, G. P., Smith, M. P. *HIBROWSE – Adding the Power of Relational Databases of the Traditional IR Architecture – the Future for Graphic User Interfaces*. 15<sup>th</sup> BCS IRSG Research Colloquium on Information Retrieval, University of Strathclyde, 1993
- [Ramos92]** Ramos, H. B. *IQL: An Integrated Graphical User Interfaces for Relational Queries with Genericity*, em Anais do VII Simpósio Brasileiro de Banco de Dados, Porto Alegre, p. 431-446, maio de 1992
- [Rich89]** Rich, E. *Stereotypes and User Modeling*. In A. Kobsa and W. Wahlster (eds.): *User Models in Dialog Systems*, pp. 35-51. Springer, Berlin, Heidelberg, 1989
- [Sneid87]** Sneiderman, Ben. *Designing the User Interface – Strategies for Effective Human-Computer Interaction*. Ed. Addison Wesley, edição maio de 1987
- [Wahlster86]** Wahlster, W., Kobsa, A. *Dialogued-based User Models*. In proc. Of the IEEE, pp. 948-1040, 1986
- [Zloof77]** Zloof, M. *Query by Example*. IBM Systems Journal, 21 (3): 324-343, 1977