

**Instituto de Ciências Matemáticas e de Computação**

ISSN - 0103-2569

**Editor de Esquemas – E<sup>2</sup> SIRIUS  
para o Modelo SIRIUS  
Manual do Usuário**

**MYRIAN RENATA BARROS ARAUJO  
CAETANO TRAINA JR.**

**Nº 88**

**RELATÓRIOS TÉCNICOS DO ICMC**

São Carlos  
Abr./1999

|             |         |
|-------------|---------|
| SYSNO       | 1042394 |
| DATA        | / /     |
| ICMC - SBAB |         |

Instituto de Ciências Matemáticas e de Computação

Editor de Esquemas - E<sup>2</sup>SIRIUS  
para o Modelo SIRIUS  
Manual do Usuário

Myrian Renata Barros Araujo  
e\_mail: mrenata@icmc.sc.usp.br

Caetano Traina Jr.  
e\_mail: caetano@icmc.sc.usp.br

ICMSC - USP  
13560 - 970

Caixa Postal 668  
São Carlos SP

Abril de 1999

# Sumário

---

---

|                                                             |        |
|-------------------------------------------------------------|--------|
| Capítulo 1 .....                                            | 01     |
| Introdução .....                                            | 01     |
| 1.1 Considerações iniciais .....                            | 01     |
| 1.2 Organização do Trabalho .....                           | 04     |
| <br>Capítulo 2 .....                                        | <br>05 |
| Editor de Esquemas - E2SIRIUS .....                         | 05     |
| 2.1 Formulário Objeto .....                                 | 05     |
| 2.2 Formulário Define/Vincula Atributos .....               | 10     |
| 2.3 Remove Atributo .....                                   | 11     |
| 2.4 Formulário Atributo .....                               | 13     |
| 2. 5 Remove SubAtributo .....                               | 16     |
| 2. 6 Formulário SubAtributo .....                           | 17     |
| 2. 7 Formulário Vínculo .....                               | 19     |
| 2. 8 Formulários Editar e Definir Valor de Atributo .....   | 23     |
| 2. 9 Formulário Especialização Restrita/Categorização ..... | 26     |
| 2. 10 Formulário Especialização Restrita .....              | 27     |
| 2. 11 Formulário Generalização Ampla .....                  | 29     |
| 2. 12 Formulário Colônias Abertas .....                     | 31     |
| 2. 13 Formulário Define Tipo Colônia .....                  | 32     |
| 2. 14 Formulário Habita .....                               | 33     |
| <br>Capítulo 3 .....                                        | <br>34 |
| Considerações finais .....                                  | 34     |
| Referências Bibliográficas .....                            | 35     |

# Lista de figuras

---

---

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| <b>Figura 1:</b> Tela de rosto do Editor .....                                       | 01 |
| <b>Figura 2:</b> Módulos do Sistema .....                                            | 01 |
| <b>Figura 3:</b> Formulário Objeto .....                                             | 05 |
| <b>Figura 4:</b> Formulário Define/Vincula Atributos .....                           | 10 |
| <b>Figura 5:</b> Formulário Remover Atributo .....                                   | 11 |
| <b>Figura 6:</b> Formulário Atributo (característica texto) .....                    | 13 |
| <b>Figura 7:</b> Formulário Atributo (característica tupla) .....                    | 14 |
| <b>Figura 8:</b> Formulário Remove SubAtributo .....                                 | 16 |
| <b>Figura 9:</b> Formulário SubAtributo (característica tupla) .....                 | 17 |
| <b>Figura 10:</b> Formulário SubAtributo (característica atributo de atributo) ..... | 18 |
| <b>Figura 11:</b> Formulário Vínculo .....                                           | 19 |
| <b>Figura 12:</b> Formulários para edição de valor(es) de atributo(s) .....          | 23 |
| <b>Figura 13:</b> Formulário Especialização Restrita/Categorização .....             | 26 |
| <b>Figura 14:</b> Formulário Especialização Restrita .....                           | 27 |
| <b>Figura 15:</b> Formulário Generalização Ampla .....                               | 29 |
| <b>Figura 16:</b> Formulário Colônias Abertas .....                                  | 31 |
| <b>Figura 17:</b> Formulário Define Tipo Colônia .....                               | 32 |
| <b>Figura 18:</b> Formulário Habita .....                                            | 33 |



# Capítulo 1

## Introdução

### 1.1 Considerações iniciais

Este relatório apresenta o Editor de Esquemas - E<sup>2</sup>SIRIUS, **Figura 1**, projetado para dar suporte a Metamodelos Orientados a Objetos.

O aplicativo E<sup>2</sup>SIRIUS foi desenvolvido como parte do trabalho de mestrado [Araujo\_98], [Araujo\_98a]. Os conceitos utilizados na construção deste editor aplicam-se a outros modelos elaborados segundo o paradigma de orientação a objetos que tenham a



**Figure 1:** Tela de rosto do Editor.

característica de serem metamodelos. Em particular, o editor desenvolvido apoia-se no modelo SIRIUS [Biajiz\_96].

O documento aqui apresentado descreve parte de um projeto mais abrangente, o Sistema SIRIUS, **Figura 2**, que está sendo desenvolvido pelo grupo GbDI - Grupo de Base de Dados e Imagens do ICMC- USP [<http://gbdi.icmc.sc.usp.br/>]. O Sistema SIRIUS é composto basicamente por três módulos:

- o Modelo SIRIUS;
- o Gerenciador SIRIUS/GO;
- e as Técnicas SIRIUS de Desenvolvimento de Sistemas.



**Figure 2:** Módulos do Sistema

As Técnicas de Desenvolvimento SIRIUS englobam um conjunto de aplicativos que estão sendo projetados e

implementados para auxiliar na construção do sistema como um todo. Porém é importante destacar que tais aplicativos não tenham necessariamente que ser implementados apoiados pelo Gerenciador SIRIUS.

O Gerenciador SIRIUS/GO [Traina\_96] está sendo projetado para atuar em arquitetura cliente/servidor, sendo que o servidor é dividido em duas partes básicas: o núcleo e o nível semântico. O núcleo possui uma estrutura em camadas: a camada superior SG-Atr (subsistema de gerenciamento de atributos); a camada seguinte é a SG-Lóg (subsistema de gerenciamento de registros lógicos); seguida da camada SG-Fis (subsistema de gerenciamento de registros físicos) e a última camada SG-Cach (subsistema de gerenciamento de memória cache). O nível semântico é formado por módulos. Entre eles o módulo de: objeto (objeto conceito), de tupla, de atributo, de *blobs* e de índices. O núcleo atua como um servidor de objetos para as camadas semânticas que compõem os vários módulos semânticos, do nível semântico. O Editor aqui apresentado é uma das ferramentas que constitui o conjunto de utilitários cliente do Gerenciador.

O módulo centralizador do sistema é o modelo SIRIUS desenvolvido a partir de um formalismo de Metamodelos Orientados a Objetos baseado na particularização de abstrações de dados [Biajiz\_96]. Os elementos do modelo SIRIUS estão estruturados segundo as abstrações que o modelo suporta. Os construtores semânticos desse modelo apoiam-se em três abstrações fundamentais: as abstrações de Classificação, de Generalização e de Associação. Esta última é especializada nas abstrações de Agregação e de Composição. Os elementos do modelo são formados principalmente por: objetos, atributos, características de atributos, colônias, tipos de objetos, tipos de colônias e tipos de atributos. Todo objeto é uma instância de um único tipo. Objeto é uma agregação de atributos, cada um possuindo um tipo e um conjunto de valores. Tipos de atributos possuem uma característica que indica entre outras, se o atributo representa um dado estático do objeto, um comportamento, uma associação com outros objetos ou uma forma de interface do sistema com o usuário. Tipos de objetos são por sua vez objetos, e como tais, podem ser instâncias de outros tipos, criando uma hierarquia de classificação. Tipos de objetos podem ser, também, subtipos e supertipos, criando uma rede acíclica de tipos pela abstração de generalização/especialização. É possível definir como os objetos são compostos por outros, em hierarquias de composição. O conjunto de todos os objetos que compõem diretamente um determinado objeto, segundo determinado aspecto, constitui uma colônia de objetos. Todos os objetos habitam exatamente uma colônia e toda colônia é constricta pelo objeto composto. A própria



base de dados é considerada um objeto, que restringe a colônia global, onde habitam todos os objetos que são máximos de uma hierarquia de composição. Toda colônia tem um tipo, que determina o aspecto do objeto que a restringe, que é detalhado pelos objetos que a habitam. Um objeto pode restringir qualquer número de colônias, desde que seja uma de cada tipo [Traina\_97] e [Ferreira\_96].

A confecção do editor teve como objetivo oferecer suporte tanto para a edição de modelagens de situações reais no Gerenciador de Objetos SIRIUS/GO, quanto auxiliar projetistas no aprendizado e treinamento dos conceitos do modelo. Como Editor de Esquemas de SIRIUS/GO, fornece suporte necessário para a especificação detalhada de todos os conceitos do modelo, bem como, recursos para a alimentação de dados de teste em pequenos volumes, tendo também uma interface que procura facilitar o trabalho de concepção e digitação de esquemas. Como ferramenta de auxílio à aprendizagem do modelo, o Editor apresenta recursos para permitir a manipulação e visualização de conceitos próprios de SIRIUS. Entre esses recursos, o Editor explora particularmente o conceito de abstrações de dados, entre elas, a Abstração de Classificação, que no modelo é suportada para a representação de situações de classificação no mundo real, e torna sintaticamente homogênea a criação tanto de tipos quanto de instâncias.

Para a criação do Editor - E<sup>2</sup>SIRIUS, as abstrações do modelo foram agrupadas em subconjuntos de formulários que abrangem apenas uma das abstrações de cada vez. Em outras palavras, os subconjuntos de formulários foram agrupados de modo a obter-se uma coleção mínima de formulários. Em particular, o Editor de Esquemas conta com um pequeno conjunto de telas (onde não existe interação entre o usuário e a base) e um conjunto considerável de formulários (onde existe interação entre o usuário e a base). É através dos formulários que o usuário pode armazenar e/ou consultar e/ou alterar seus dados na base. É importante observar que os formulários e/ou subconjuntos de formulários estão interligados, pois existe uma conexão entre as abstrações. Desta forma, o Editor engloba todas as abstrações do modelo.

Em relação as abstrações é importante ressaltar que a navegação, por elas no Editor, será feita sempre par a par, isto é, classificação/instanciação, generalização/especialização e restringir/habitar. Além do movimento de navegação ser sempre em uma única direção, o usuário dificilmente terá muitos níveis dentro de uma abstração. Desta forma, não existe composição de abstrações, dificultando assim que o usuário se perca durante a navegação.

A base de dados foi construída utilizando-se o gerenciador INTERBASE

[<http://www.interbase.comr>] versão 4.2. A implementação foi construída de maneira portátil, utilizando os recursos básicos da linguagem SQL [Elmasri\_94], de maneira que o aplicativo possa ser conectado em outro servidor, monousuário ou multiusuário, que utilize comandos SQL básicos .

Tendo-se uma base criada, o passo seguinte foi a elaboração de uma interface que, como a base, engloba todos os construtores semânticos do modelo. Na implementação da interface a ferramenta de desenvolvimento utilizada foi o C++Builder Cliente/Server Suite [<http://www.inprise.com/bcppbuilder>] versão 3.0.

## 1.2 Organização do Trabalho

Estes relatório está organizado em 3 capítulos:

- o Capítulo1: descreve de modo geral o Editor de Esquemas, contextualizando-o no Sistema SIRIUS o qual ele pertence;
- o Capítulo2: descreve os formulários elaborados para a confecção de E<sup>2</sup>SIRIUS. Exemplificando assim o próprio funcionamento do aplicativo. Quando se faz necessário alguns conceitos do modelo são explicados para melhorar o entendimento do formulário e conseqüentemente do Editor;
- o Capítulo3: contém as considerações finais.



# Capítulo 2

## Editor de Esquemas - E<sup>2</sup>SIRIUS

Este capítulo descreve os principais formulários que compõe o Editor de Esquemas. Nesta seção, apenas para facilitar a leitura, os interadores que são botões são escritos entre parênteses para diferencia-los dos demais interadores.

### 2.1 Formulário Objeto

O formulário Objeto, **Figura 3**, destaca-se por ser o mais importante e conseqüentemente o mais complexo, é formado por interadores que dão suporte para navegação em todas as hierarquias de

The image shows a screenshot of the 'Objeto' form in the E2SIRIUS editor. The form has a title bar 'Objeto' and a menu bar with 'Abstrato', 'Detalhe', and 'Relacionamento'. It is divided into two main sections: 'Atributos de Classificação' (top) and 'Atributos de Instanciação' (bottom). The 'Atributos de Classificação' section includes a table with columns 'NOME', 'CARACT.', 'NIVEL', 'IDENT.', 'DEFAULT', and 'ORGANIZAÇÃO'. Below this table are fields for 'Tipo do Objeto', 'Meta Tipo Obj', and 'Nome do Objeto'. The 'Atributos de Instanciação' section includes a table with columns 'ATRIBUTO', 'SUB\_ATRIBUTO', 'CARACT.', and 'VALOR'. To the right of these sections are various buttons and controls, including 'Define Colônia', 'Objeto', 'Atributo', 'Topo', 'Abstrações', 'Classificar', 'Generalizar', 'Constringir', 'Instanciar', 'Especializar', 'Habilitar', 'Al. Classif', 'Edita Valor', 'Al. Instanc', 'Edita Valor', 'Def. Valor', 'Def. Atr. Extra', and 'Fechar'. There are also labels 'Atrib.de Classificação', 'Obj. Corrente', and 'Atrib.de Instanciação' pointing to specific parts of the form. On the right side, there are two boxes labeled 'Valore(s) p/ Atributo(s)' and 'Valore(s) p/ Atributo(s)' with arrows pointing to the 'Edita Valor' buttons.

Figure 3: Formulário Objeto.

abstrações do modelo. Este formulário possui a característica de ser um formulário de consulta (o usuário pode visualizar os atributos de classificação e/ou instanciação) e edição (possibilita a criação e manipulação de objetos).

O enfoque principal está centrado no interador Objeto, chamado aqui também de objeto corrente, ou seja, o objeto que está sendo editado ou que já está armazenado na base. Este objeto é composto por três interadores:

- Nome do Objeto: componente onde o usuário irá editar o nome do objeto (quanto este estiver sendo criado) e/ou apenas irá visualizar o(s) objeto(s) que já está(ão) armazenado(s) na base, para isto o usuário deve clicar a direita deste componente;
- Tipo do Objeto: componente que exibe o(s) tipo(s) de objeto(s), este(s) é(são) criado(s) automaticamente pelo sistema quanto o usuário instância um objeto tipo, por meio da abstração de instanciação. Neste caso, a navegação na hierarquia de Classificação/Instanciação é feita de cima para baixo;
- Tipo da Colônia: componente que lista o(s) tipo(s) de colônia(s), este(s) deve(m) ser definido(s) pelo usuário, no formulário Defina Tipo de Colônia (que será visto adiante), este componente define qual a colônia que a(s) instância(s) do objeto corrente irá(ão) habitar.

Em uma modelagem pode(m) existir atributo(s) de classificação e/ou atributo(s) de instanciação vinculado(s) a um objeto, isto é ilustrado em E<sup>2</sup>SIRIUS, através dos segmentos de retas verticais que ligam o objeto ao(s) seu(s) atributo(s). O segmento acima do objeto estabelece a ligação entre o objeto corrente e o(s) atributo(s) de classificação (representado(s) no interador Atributos de Classificação), do mesmo modo é estabelecida a conexão entre o objeto corrente e seu(s) atributo(s) de instanciação (representado(s) no interador Atributos de Instanciação). O interador Atributos de Classificação exibe para cada atributo de classificação informações como:

- nome: onde o usuário irá visualizar o nome do atributo de classificação;
- caract.: campo que contém informação a respeito da característica do atributo de classificação. Lembrando que o modelo suporta quinze características de atributo distintas que são divididas em quatro classes:
  - Estática, englobando as características:
    - texto;



- número e;
- booleano;
- Dinâmica, contendo as características:
  - regra e
  - procedimento;
- Interface, agrupando as características:
  - visualização;
  - som;
  - partitura;
  - imagem e
  - gráfico;
- e finalmente as características da classe Estrutural representadas pelos atributos:
  - estrutura de dado;
  - tempo;
  - tupla;
  - atributo de atributo e
  - relacionamento;
- nível: este campo exibe o nível do atributo, ou seja, um valor numérico que expressa em qual nível da hierarquia de classificação ele (atributo) será instânciado;
- ident: ilustra se o atributo é um identificador ou não. Atributo(s) identificador(es) e (são) representado(s) pela letra S (Sim) e atributo(s) não identificador(es) é (são) representado(s) pela letra N (Não);
- *default*: esta coluna indica qual(is) atributo(s) possui(m) a particularidade de ser(em) atributo(s) *default*, ou seja, atributo(s) que possui(m) um valor pré-definido, onde tal valor pode ou não ser repassado para as instâncias do objeto que está vinculado a este atributo. Atributo(s) *default* é (são) representado(s) pela letra S e atributo(s) não default é (são) representado(s) pela letra N;
- organização: indica se um atributo é monovalorado ou multivalorado.

O botão (Edita Valor), à direita do interador Atributos de Classificação, chama outros formulários permitindo que o usuário defina um valor para o atributo que foi vinculado ao objeto como sendo atributo *default*.



De modo análogo o interador Atributos de Instanciação ilustra informações específicas dos atributos de instanciação, informações como:

- atributo: nome do atributo instanciado;
- sub\_atributo: nome do subatributo (quando o atributo instância possui a característica tupla ou atributo de atributo);
- caract: mostra o qual a característica do atributo, e se existir, qual a característica do subatributo;
- valor: exibe o valor dado ao atributo (ou subatributo) após a sua instanciação.

Tem-se três botões à direita do interador Atributos de Instanciação (exceto o botão (Fechar)) que são:

- o botão (Def.Valor): cuja finalidade é definir e/ou modificar o(s) valor(es) do(s) atributo(s) do objeto que está um nível acima na hierarquia de classificação em relação ao objeto corrente;
- o botão (Edita Valor) cuja funcionalidade é definir e/ou modificar o(s) valor(es) do(s) atributo(s) associado(s) ao objeto corrente e;
- o botão (Def.Atr.Extra) que irá chamar formulários específicos para o usuário definir um atributo extra.

Existem ainda dois outros botões, situados entre os interadores Atributos de Classificação e Atributos de Instanciação, que se referem aos atributos:

- o botão (+ Atributo): que permite o usuário entrar em uma tela onde ele deve optar por adicionar um novo atributo ou fazer um vínculo entre um atributo que já existe na base e o objeto corrente e;
- o botão (-Atributo): que possibilita o usuário remover um atributo da base ou simplesmente remover o vínculo deste atributo com o objeto corrente.

Os formulários que permitem o usuário efetuar tais ações serão melhores detalhados nos próximos tópicos.

Associados diretamente ao objeto corrente existem os botões:

- o botão (+ Objeto): que insere e armazena na base um objeto definido pelo usuário e o;
- o botão (- Objeto): que remove da base o objeto corrente.

Foi criado no formulário Objeto um botão de atalho, para que o usuário navegue na hierarquia de classificação mais rapidamente, ou seja, este botão permite que o usuário volte para o

objeto mais alto na hierarquia de classificação diretamente, sem ter que retornar nível a nível na hierarquia. Este botão é chamado (Topo).

O formulário Objeto é importante no Editor de Esquemas - E<sup>2</sup>SIRIUS pois, além das funções descritas acima, é este formulário que permite que o usuário represente, em uma modelagem, todas as abstrações que o modelo suporta:

- em nível abstrato, através dos botões:
  - (Classificar);
  - (Generalizar) e;
  - (Constringir),
- e em nível de detalhe, através dos botões:
  - (Instanciar);
  - (Especializar) e;
  - (Habitar).

A associação de atributos a objetos, atributos a outros atributos (formando as estruturas: tuplas e atributo de atributo) e as associações entre objetos a objetos (através de atributos com característica de relacionamento que estão associados aos objetos que participam do relacionamento), é a maneira de implementar a abstração de Agregação. Desta forma, os construtores semânticos do modelo estão centralizados no objeto corrente.

Quando uma base de dados é criada, ela é inicializada automaticamente com todos os objetos que pertencem ao Esquema (Base Intencional) definidos pelo modelo SIRIUS, como por exemplo, colônia Global e objeto tipo Meta Tipo Obj. Por exemplo, o objeto Meta Tipo Obj. pode ser exibido no interador Tipo do Objeto. Esta é a forma de manipular objetos que são tipo ou não, isto é possível, porque o Editor de Esquemas - E<sup>2</sup>SIRIUS foi criado de modo que objeto tipo e objetos que não são tipos são tratados de maneira uniforme (homogênea). Pois como o modelo suporta tipos persistentes estes também devem ser tratados, ou seja, criados, manipulados e visualizados [Araujo\_98a].

## 2.2 Formulário Define/Vincula Atributos

O formulário Define/Vincula Atributos, **Figura 4**, é um formulário de consulta, pois permite que o usuário visualize todos os atributos que existem na base de dados, no entanto, o usuário não poderá editar e/ou manipular novos atributos neste formulário. Para efetuar estas últimas operações o usuário deve selecionar o botão (Definie atributo). Este formulário é composto por quatro interadores. São eles:

- Atributos da Base: é este interador que exibe todos os atributos armazenados na base juntamente com suas respectivas características,

possibilitando que o usuário efetue uma consulta aos atributos que já foram criados e que se encontram disponíveis na base (tais atributos podem ou não estar vinculados a um ou mais objetos);

- o botão Define atributo: este interador permite que um novo atributo seja criado na base, caso ele não exista. Para isto, este botão faz uma chamada ao formulário Atributo, que permite a definição de novos atributos (ou subatributos);
- o botão Vincula: uma vez criado(s) o(s) atributo(s) geralmente é necessário que seja(m) efetuado(s) o(s) vínculo(s) deste(s) ao(s) objeto(s) que pertencem a uma modelagem. Para suprir esta necessidade o botão (Vincula) irá permitir que um atributo na base seja vinculado ao objeto corrente. Isto será feito no formulário Vínculo descrito adiante e finalmente o interador;
- o botão Fechar: sai deste formulário e retorna para o formulário que o chamou.

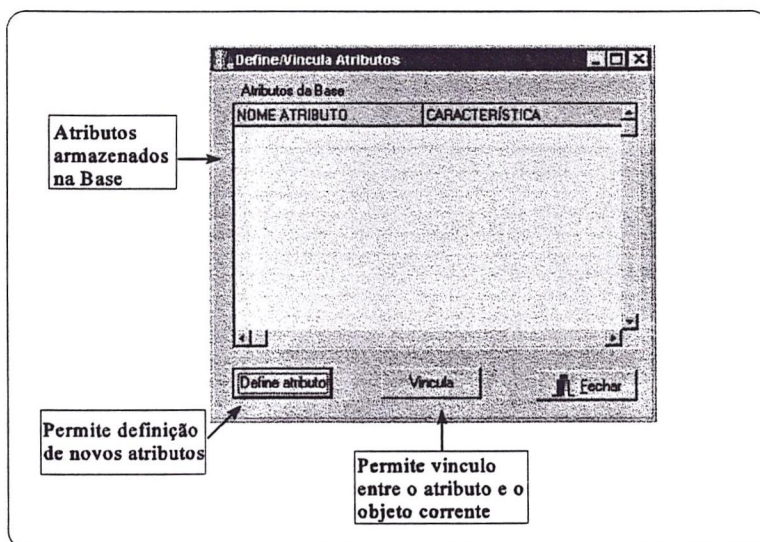


Figure 4: Formulário Define/Vincula Atributos.



## 2.3 Remove Atributo

Existem varias operações que podem ser efetuadas com um atributo. Uma delas é a de remoção. O editor permite que a remoção de um atributo seja feita de dois modos:

- simplesmente removendo o vínculo entre o atributo e o objeto corrente, (deste modo o atributo permanece na base) ou;
- removendo o atributo da base. Um atributo só poderá ser removido da base caso não exista vínculo dele com nenhum objeto ou com nenhum outro atributo.

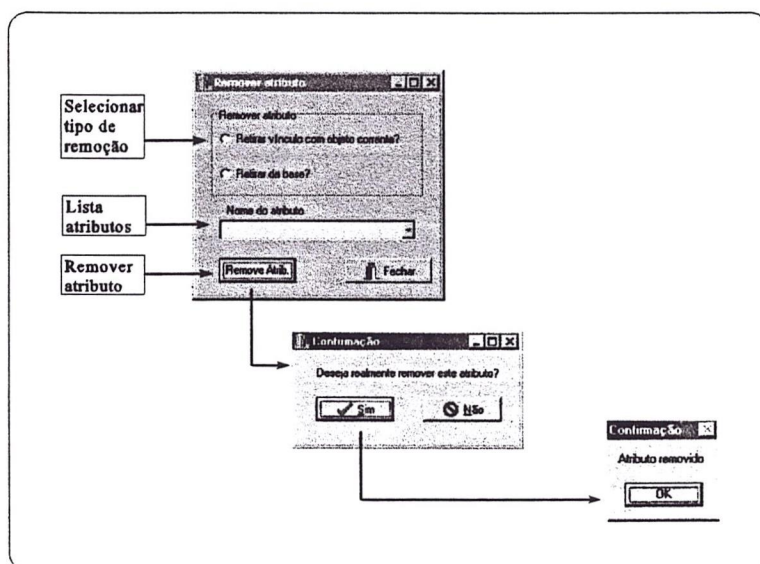


Figure 5: Formulário Remover Atributo.

Para efetuar a operação de remoção foi criado o formulário Remover Atributo, **Figura 5**, que é formado por quatro interadores:

- Remover atributo: onde o usuário poderá selecionar como ele deseja remover um atributo (retirar vínculo com o objeto corrente ou retirar o atributo da base);
- Nome do atributo: este componente irá listar o(s) atributo(s) apropriado(s) dependendo do que o usuário selecionou no interador anterior. Se o usuário optar por retirar o vínculo do atributo com o objeto corrente, serão listados apenas os atributos que já estão vinculados com o objeto corrente, para que o usuário possa escolher qual o atributo que não mais será vinculado ao objeto. Por outro lado, se o usuário optar por retirar o atributo da base, serão listados todos os atributos da base, para que o usuário selecione qual deles será removido;
- o botão Remove Atributo: uma vez escolhido o atributo que será removido este interador irá chamar uma tela onde o usuário deverá confirmar se deseja remover ou não o atributo. Caso deseje, uma ação apropriada (remoção da base ou vínculo) será executada. Se

o atributo selecionado para remoção não puder ser removido e/ou se sua remoção implicar em outras consequências (por exemplo, remoção de um atributo que seja também um subatributo de uma tupla, para removermos este atributo devemos primeiro remove-lo da tupla) o aplicativo irá enviar uma mensagem informando a situação do atributo para que o usuário saiba como direcionar suas opções;

- o botão Fechar: sai deste formulário e retorna para o formulário que o chamou.



## 2.4 Formulário Atributo

O formulário Atributo presta-se tanto para consulta (o usuário pode visualizar se já existe um atributo de uma certa característica) quanto para criação, edição e manipulação de atributos (pois possui campos apropriados para o usuário definir as propriedades de um novo atributo). Este formulário é composto por várias “fichas” sobrepostas. Cada “ficha” representa uma, dentre as quinze, características de atributo que o modelo suporta.

Características de atributos que o modelo suporta

Atributo

Atributo de atributo | Booleano | Estrutura de dados | Gráfico | Imagem | Partitura | Procedimento  
Regra | Relacionamento | Som | Tempo | **Texto** | Tupla | Visualização | Número

Nome do atributo

Lista de atributos de acordo com a característica

Formato

String | ASCII | RTF

Tamanho da string

Formato de dado do atributo

Salvar

Fechar

Armazena atributo

Figure 6: Formulário Atributo (característica texto).

O nome da característica do atributo é situada na “lingueta” de cada ficha e é através desta que se escolhe a característica do atributo a ser criado. Como as características possuem particularidades específicas, estas também são representadas no formulário, ou seja, as “fichas” são distintas umas das outras, possibilitando que o usuário armazene informações apropriadas e específicas para cada característica. Por exemplo, para criar-se um atributo com característica texto, **Figure 6**, o usuário deve editar informações específicas desta característica nos interadores apropriados:



- Nome atributo: este interador lista todos os atributos com característica texto armazenados na base, se o atributo que o usuário necessita representar ainda não existe, ele deve digitar neste componente o nome do novo atributo, que terá neste caso característica texto. Caso já exista um atributo com o mesmo nome, o aplicativo não permite que um novo atributo com o mesmo nome seja criado;
- Formato: além do nome, do novo atributo, o usuário deve também selecionar o formato de dado que este atributo terá;
- o botão (Salvar): após a edição do novo atributo para que este seja, de fato, armazenado na base, o usuário deverá salvá-lo, clicando no interador (Salvar);
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.

No entanto, para definir outros atributos que não tenham a característica texto, os procedimentos serão outros, dependendo da característica escolhida. Como, por exemplo, um

Figure 7: Formulário Atributo (característica tupla).

atributo com característica tupla, **Figura 7**. Neste caso, o usuário deve escolher a “ficha” própria

para a edição de um atributo com característica tupla. A tela de edição, manipulação e visualização de um atributo com característica tupla, contém seis interadores, são eles:

- Nome do atributo: onde serão listados todos os atributos com característica tupla já definidos. Para criação de um novo atributo o usuário deve digitar o nome do novo atributo neste componente, em seguida salvar esta informação, para armazenar na base. Caso já exista um atributo com o mesmo nome, o aplicativo não permite que um novo atributo com o mesmo nome seja criado. Uma vez armazenado na base, os botões (+Subatributo) e (-Subatributo) são habilitados;
- o botão (+Subatributo): tem como função permitir que um novo(s) subatributo(s) seja(m) criado(s). Cabe observar que o(s) subatributo(s) é (são) criado(s) através de um formulário próprio, o formulário Sub Atributo (que será visto no próximo item);
- o botão (-Subatributo): chama o formulário Remover SubAtributo, que como o próprio nome diz, tem a função de remover um subatribuo;
- Subatributos de Classificação: este componente não é um componente editável, o objetivo dele, é mostrar para o usuário quais os subatributos que estão associados ao atributo com característica tupla. Este componente é sempre atualizado de acordo com o atributo selecionado (no interador Nome do atributo). Desta forma, sempre que um atributo com característica tupla tiver um ou mais subatributos, este(s) pode(m) ser visualizado(s), assim como as suas respectivas características;
- o botão (Salvar): permite que um novo atributo com característica tupla seja armazenado na base;
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.

De modo análogo, outros atributos com outras características podem ser criados, basta observar quais as particularidades de cada um, e defini-los apropriadamente.



## 2. 5 Remove SubAtributo

Assim com o atributo, existem várias operações que podem ser efetuadas com um subatributo, uma delas é a remoção. No entanto, a remoção de um subatributo ocorre de modo especial. O formulário Remove SubAtributo, **Figura 8**, não irá remover, de fato, o subatributo da base. Este formulário irá remover o subatributo da estrutura que este estiver associado, ou seja, remove o vínculo entre um subatributo com a estrutura tupla ou com a estrutura atributo de atributo. Para que um subatributo seja removido da base, o usuário primeiro deve remover o vínculo do subatributo com a estrutura que ele estiver conectado,

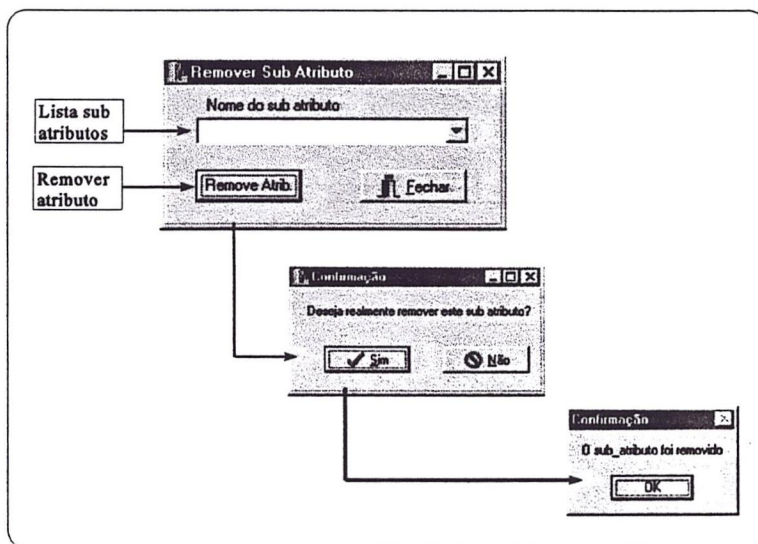


Figure 8: Formulário Remove Sub atributo.

em seguida, para removê-lo (o subatributo) da base o usuário deverá utilizar o botão (-Atributo) do formulário Objeto, e proceder da mesma forma que a remoção de um atributo.

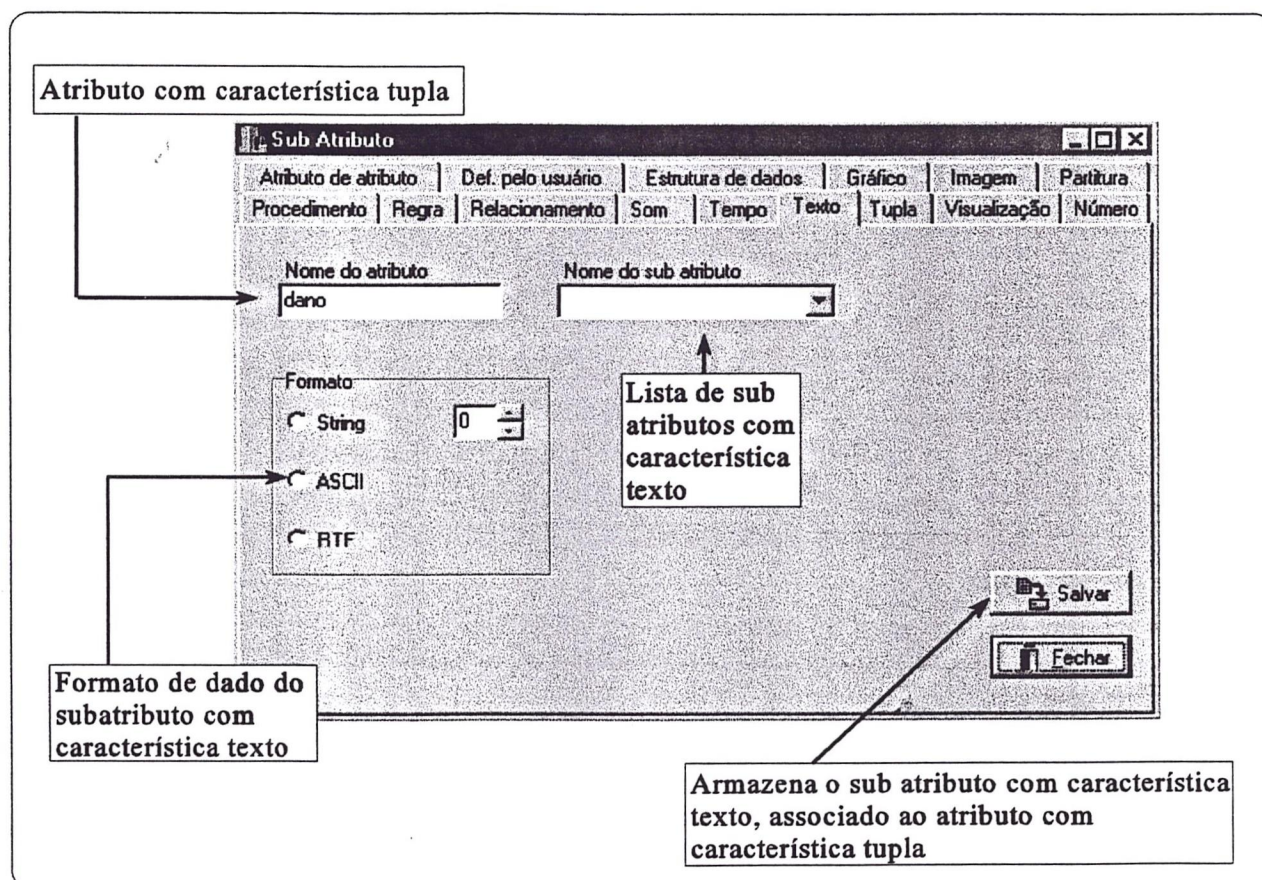
O formulário Remove SubAtributo, possui três interadores:

- Nome do sub atributo: onde o usuário deverá selecionar qual o subatributo será removido;
- o botão (Remove Atrib.): cuja finalidade é chamar uma tela onde o usuário deverá confirmar se deseja realmente efetuar tal operação, caso a opção for afirmativa, o botão (Sim), da tela Confirmação irá executar a operação de remoção do vínculo entre a estrutura tupla ou atributo de atributo e o subatributo selecionado para ser removido;
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.



## 2.6 Formulário SubAtributo

O formulário SubAtributo, **Figura 9**, assim como o formulário Atributo visto anteriormente, possui a característica de ser um formulário de consulta (o usuário pode visualizar qual o atributo corrente, ou seja, o atributo no qual serão vinculados o(s) subatributo(s)) e edição (pois possui campos apropriados para o usuário definir as propriedades de um novo subatributo). Este formulário também é composto por várias “fichas” sobrepostas. Cada “ficha” representa uma, dentre as quinze, características de atributo que o modelo suporta, visto na seção 2.4, mas, neste caso em particular, estas serão as possíveis características dos subatributos, que serão criados, editados e/ou manipulados pelo usuário.



**Figure 9:** Formulário SubAtributo (característica tupla).

A diferença básica entre o formulário Atributo (**Figura 6** ou **Figura 7**) e o formulário SubAtributo é que o último indica não só o nome do subatributo, mas também o nome do atributo ao qual o(s) subatributo(s) está(rão) associado(s). Visualmente, um interador chamado Nome do subatributo é colocado à direita do interador Nome do atributo. Em outras



palavras, cada “ficha” contém, além das informações particulares de cada característica, o interador Nome do subatributo. Cabe observar também, que quando os subatributos manipulados estão vinculados a estrutura atributo de atributo as “fichas” exibidas no formulário SubAtributo contém um interador a mais, o interador Ident. Atributo de Atributo, este interador permite que o usuário defina qual o subatributo que será o identificador da estrutura atributo de atributo.

**Atributo com característica atributo de atributo**

**Sub Atributo**

Atributo de atributo | Def. pelo usuário | Estrutura de dados | Gráfico | Imagem | Partitura  
Procedimento | Regra | Relacionamento | Som | Tempo | Texto | Tupla | Visualização | Número

Nome do atributo:

Nome do sub atributo:

☐ Ident. Atributo de Atributo

Formato:

☐ String

☐ ASCII

☐ RTF

**Formato de dado do sub atributo**

**Lista de sub atributos com característica texto**

**Seleciona o sub atributo, que será identificador**

**Salvar**

**Fechar**

**Armazena o sub atributo com característica texto, associado ao atributo com característica atributo de atributo**

Figure 10: Formulário SubAtributo (característica atributo de atributo).



## 2. 7 Formulário Vínculo

É o formulário Vínculo , **Figura 11**, que irá promover a agregação entre o objeto corrente e um

O formulário **Vínculo** possui as seguintes seções e campos:

- Lista de atributos que podem ser vinculados:** Uma lista suspensa no topo para selecionar o atributo.
- Propriedades do atributo:**
  - Nome do atributo:** Campo de texto para o nome do atributo selecionado.
  - Classif./Instanc.:** Grupo de radio buttons para escolher entre **Classificação** e **Instanciação**.
  - Organização do atributo:** Grupo de radio buttons para escolher entre **Monovalorado**, **Lista**, **Conjunto** e **Vetor**.
  - Identificador:** Caixa de seleção.
  - Fixo:** Caixa de seleção.
  - Default:** Caixa de seleção.
  - Atrib. indefinido:** Caixa de seleção.
  - Valor nulo:** Caixa de seleção.
- Nível de classificação:** Campo de spin para definir o nível, atualmente em 0.
- Norma para lista e vetor:** Campo de spin para definir a norma, atualmente em 0.
- Atributo que fornece a norma do vetor:** Campo de texto para especificar o atributo de referência.
- Botões de ação:** **Salvar** e **Fechar** no canto inferior direito.

As abas principais são **Atributos Classificação/Instanciação** e **Organização do atributo**.

**Figure 11:** Formulário Vínculo.

atributo da base. Este formulário permite consulta (pois lista quais os atributos da base), edição e/ou manipulação (alteração) pois possui campos apropriados para que o usuário efetue o vínculo entre um objeto e um atributo. Além disso fornece recurso de habilitar ou desabilitar alguns componentes dependendo do que o usuário pode ou não fazer, de acordo com imposições do modelo SIRIUS, pois houve a preocupação, na fase, de desenvolvimento, de direcionar o usuário a efetuar operações coerentes.

Existe um conjunto de interadores que compõe este formulário, tais interadores são:

- **Nome do atributo:** componente que lista todos os atributos da base;
  - **Classif./Instanc.:** neste interador deve ser feita a opção vincular um atributo de classificação ao objeto corrente ou vincular um atributo de instanciação ao objeto corrente;
- Interadores que definem propriedades dos atributos:



- **Identificador:** este interador só estará habilitado se o atributo for de classificação. Se este interador for selecionado pelo usuário, o atributo de classificação que será vinculado ao objeto possui a particularidade de ser um identificador para um objeto, ou seja, o valor deste atributo irá corresponder ao objeto que será uma instância do objeto que o atributo identificador estiver vinculado;
- **Fixo:** quando este interador for selecionado, automaticamente é setado a organização monovalorada (para o atributo), pois pela implementação um atributo será fixo se for monovalorado;
- **Default:** esta opção pode ser selecionada tanto para um atributo de classificação como para um atributo de instanciação, no entanto, quanto um atributo for *default* ele não pode ser também atributo indefinido, o que já está previsto pelo aplicativo;
- **Atrib. indefinido:** quando um atributo for indefinido significa que ainda não foi dado nenhum valor para este atributo;
- **Valor nulo:** esta opção indica que o atributo vinculado tem um valor, no caso, o valor é nulo. Portanto, este atributo não pode ser também um identificador, neste caso o interador identificador é desabilitado;
- **Nível:** este componente permite que o usuário defina a qual nível na hierarquia de classificação o atributo estará vinculado. O nível deve ser editado ou selecionado pelo usuário. No entanto, se este optar por um atributo de instanciação o interador nível será desabilitado, pois, neste caso, o nível é armazenado automaticamente como tendo valor zero;
- **Organização:** este componente define se o atributo será monovalorado ou multivalorado;
  - **Monovalorado:** se o usuário selecionar a opção monovalorado, o atributo terá um único valor, este valor poderá ser determinado por meio dos formulários:
    - Editar valor de atributo (Objeto não instanciado),
    - Editar valor de atributo (Objeto instanciado) e
    - Define valor de atributo (Atributo definido no pai). Estes formulários serão vistos na próxima seção.
  - **Multivalorados:** são os atributos que possuem mais de um valor. Os valores associados ao(s) atributo(s) multivalorado(s) também são definidos por meio dos formulários do parágrafo anteriores.
    - **Lista:** este interador representa um atributo multivalorado, atributos multivalorados

possuem a particularidade de terem dimensão. Esta dimensão, no aplicativo, pode ser estabelecida por meio do interador Norma;

- Conjunto: para este tipo de organização a norma representa o número máximo de elementos que pode ter um dado conjunto;
- Vetor: se este tipo de organização for escolhida o usuário têm duas maneiras de determinar o valor da norma. Definindo no interador Norma, a dimensão do vetor ou escolhendo no interador Atributo que fornece a norma do vetor. Este interador permite que a norma de um vetor seja escolhida por um atributo com característica número que possui um valor inteiro.;
- o botão (Salvar): o aplicativo permite que qualquer campo possa ser modificado a qualquer hora, porém quando o usuário desejar armazenar todas as informações referentes ao vínculo, isto deve ser feito através do interador (Salvar). Observe que uma vez armazenadas estas informações, para que uma alteração possa ser feita o usuário deverá remover o vínculo do atributo com o objeto corrente, e fazê-lo novamente;
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.

Resumindo, um atributo que será vinculado ao objeto deve ser escolhido no interador Nome do atributo. Uma vez escolhido o atributo, este pode ser vinculado ao objeto, podendo ser um atributo de classificação ou atributo de instanciação. Conforme a opção escolhida, outros interadores serão ativados pelo aplicativo, de maneira adequada, de acordo com os conceitos teóricos do modelo. Por exemplo, se o usuário desejar vincular um atributo como sendo de classificação, ele deve indicar, no interador Classif./Instanc., a primeira opção. Feito isso, o sistema pré-inicializa o interador Nível para o valor 1 (um). Se esse atributo for um identificador, E<sup>2</sup>SIRIUS desabilita os interadores Fixo, Default, Atrib. Indefinido e Valor Nulo apenas o interador Nível fica habilitado para que o usuário forneça qual é o nível de classificação do atributo que está sendo vinculado. Se o atributo for indefinido, então ele não pode ser um identificador e também não pode ter um valor nulo. Neste caso, os interadores Identificador e Valor Nulo são desabilitados. Se o atributo tiver valor nulo, o interador Identificador é desabilitado.

Considerando-se um atributo vinculado como sendo atributo de instanciação, então os interadores Identificador e Nível são desabilitados, pois não faz sentido para o modelo ter



um atributo instanciado que seja um identificador e/ou que possua nível. Se o atributo for indefinido significa que não será atribuído nenhum valor para ele, portanto, o interador Valor Nulo é desabilitado, para que o atributo não tenha nenhum valor, nem mesmo o valor nulo. O fato de existirem interadores que, ora estão habilitados, ora não, permite impedir, via aplicativo, que o usuário execute operações ilegais. Este recurso foi utilizado também em outros formulários do Editor.

Ao escolher a forma como um atributo será vinculado, o usuário deve escolher também a organização do atributo. Um atributo pode ser monovalorado ou multivalorado. Quando for indicada a organização monovalorado, a norma (ou comprimento) não precisa ser dada, neste caso, foi pré-definida pelo sistema como tendo valor 1 (um). No caso de atributos multivalorados, que podem possuir como organização lista, conjunto ou vetor, a norma deve ser setada pelo usuário. Finalmente, o vínculo é armazenado na base através do interador (Salvar).

## 2.8 Formulários Editar e Definir Valor de Atributo

O Editor de Esquemas E<sup>2</sup>SIRIUS possui três formulários específicos para edição e/ou definição de valor(es) de um atributo. No entanto, cada formulário é empregado em uma determinada situação.

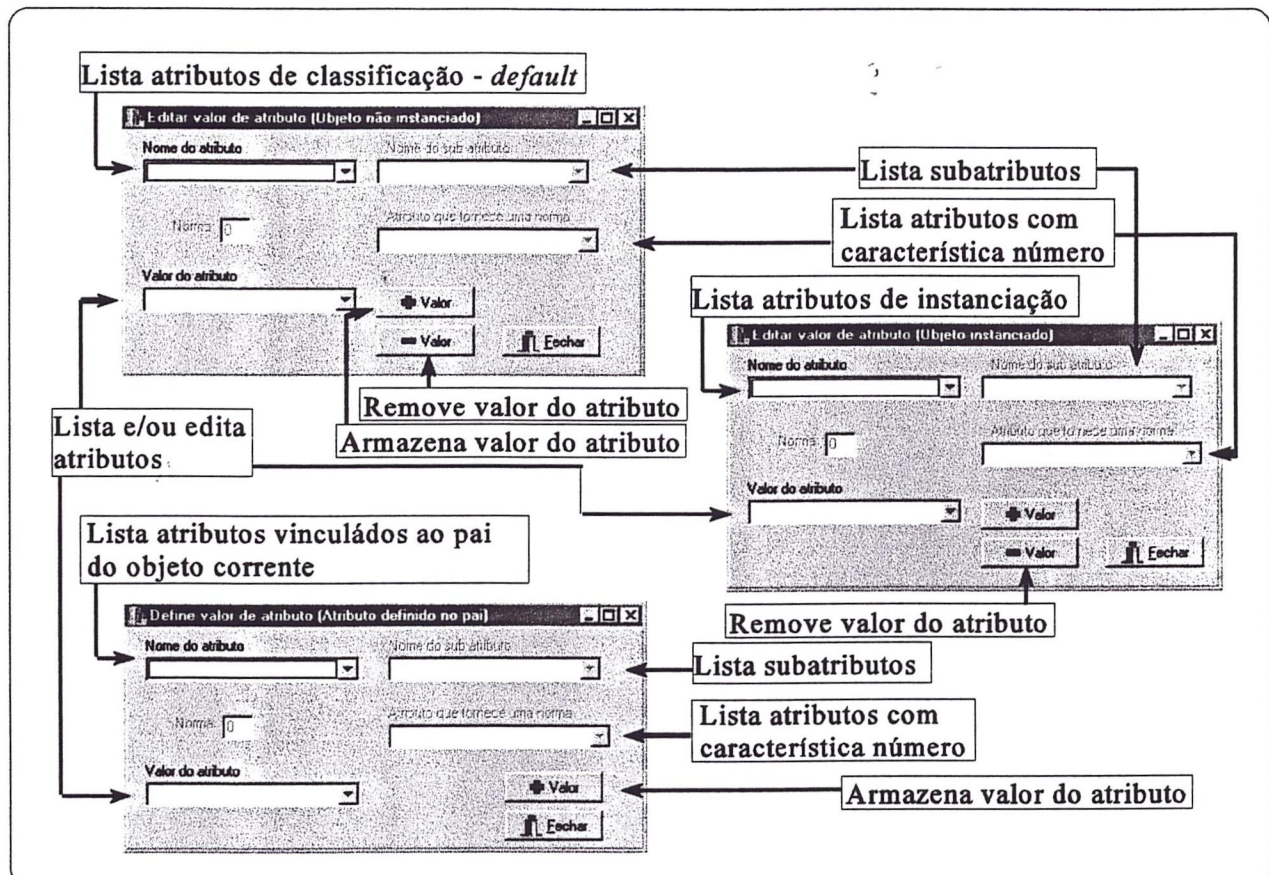


Figure 12: Formulários para edição de valor(es) de atributo(s).

Os formulários são:

- **Editar Valor de Atributo (Objeto não Instanciado):** este permite que o usuário defina um valor para um atributo de classificação vinculado como *default*, ou seja, é através deste formulário que atributo(s) *default* tem(têm) seu(s) valor(es) definido(s) e/ou alterado(s) caso já exista(m);
- **Editar Valor de Atributo (Objeto Instanciado):** possibilita que o valor(es) de atributo(s) de instanciação (nível zero) seja(m) definido(s) e/ou modificado(s);
- **Define Valor de Atributo (Atributo definido no pai):** é este formulário que irá definir e/ou alterar valor de atributo(s) (fixo(s), monovalorado(s) e de nível um) que estão vinculados no objeto que está um nível acima na hierarquia de classificação,



ou seja, no objeto “pai”.

Os formulários acima, são bem parecidos, no entanto cada um possui uma funcionalidade específica. O aplicativo possui os “mesmos” interadores para estes formulários a menos do interador (+Valor) que não esta presente no formulário Define Valor de Atributo (Atributo definido no pai). É importante observar que embora os interadores sejam os mesmos as funcionalidades destes não necessariamente são as mesmas.

Os formulários Editar Valor de Atributo (Objeto não Instanciado) e Editar Valor de Atributo (Objeto Instanciado) possuem nove interadores:

- Nome do atributo: é o único que difere de um formulário para outro, pois ele irá listar o nome do atributo de acordo com a funcionalidade do formulário. No formulário:
  - Editar Valor de Atributo (Objeto não Instanciado): os atributos listados são os atributos de classificação, que também são *default*. No formulário;
  - Editar Valor de Atributo (Objeto Instanciado): são listados todos os atributos de instanciação, atributos com nível zero, associados ao objeto corrente instanciado, por fim no formulário;
  - Define Valor de Atributo (Atributo definido no pai): lista os atributos vinculados ao “pai” do objeto corrente;
- Nome do subatributo: este interador será habilitado somente quando o atributo possuir características tupla ou atributo de atributo. Neste interador serão listados os subatributos, para que o usuário escolha qual subatributo deve ter seu valor editado e/ou alterado;
- Norma: é um interador para consulta que será ativado quando o atributo que irá ter um valor editado e/ou alterado for multivalorado. O valor deste interador já foi estabelecido quando o vínculo entre o atributo e o objeto foi feito. É este valor que será exibido para que o usuário saiba quantos valores ele deverá armazenar, no máximo, para um atributo multivalorado;
- Atributo que fornece a norma do vetor: este interador, assim como o anterior, é para consulta, no entanto este será utilizado somente para vetores (que tiveram a norma determinada desta forma, no formulário Vínculo);
- Valor do atributo: é um interador de consulta e edição, pois o usuário pode verificar os valores de atributos já existentes (no caso de atributos multivalorados) ou pode editar um valor para um atributo (ou subatributo). Este interador permite também que o usuário escolha um valor já definido para alteração se necessário;

- o botão (+Valor): este interador irá armazenar na base o valor do atributo. Após a inserção deste valor, o aplicativo irá enviar uma mensagem para o usuário, avisando que o valor foi inserido na base. É interessante notar que quando se determina um valor para um dado atributo este valor pode ser visto no formulário Objeto, que possui interadores específicos (que são sempre atualizados) para consulta;
- o botão (-Valor): remove da base um valor de atributo;
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.



## 2. 9 Formulário Especialização Restrita/Categorização

O modelo SIRIUS trata a abstração de Generalização de maneira mais abrangente do que normalmente é tratado na literatura. SIRIUS considera a abstração de Generalização Ampla. Esta abstração é especializada em duas outras: a abstração de Especialização Restrita e a abstração de Categorização.

Esta última deve ser melhor desenvolvida, porém, o Editor de Esquemas foi criado de forma a permitir facilmente a inclusão futura de maiores detalhes referentes a esta abstração.

A partir do formulário Objeto, o objeto tipo (objeto corrente) pode ser especializado, para isto, o usuário deve selecionar o botão (Especializar). Este

botão irá exibir o formulário Especialização Restrita/Categorização, criado com a finalidade de tratar os dois tipos de especializações de forma ortogonal. Este formulário possui apenas três interadores:

- o botão (Especializar): chama o formulário Especialização Restrita;
- o botão (Categorização): chama o formulário Categorização;
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.

Nas próximas seções os formulários Especialização Restrita e Generalização serão mais detalhados.

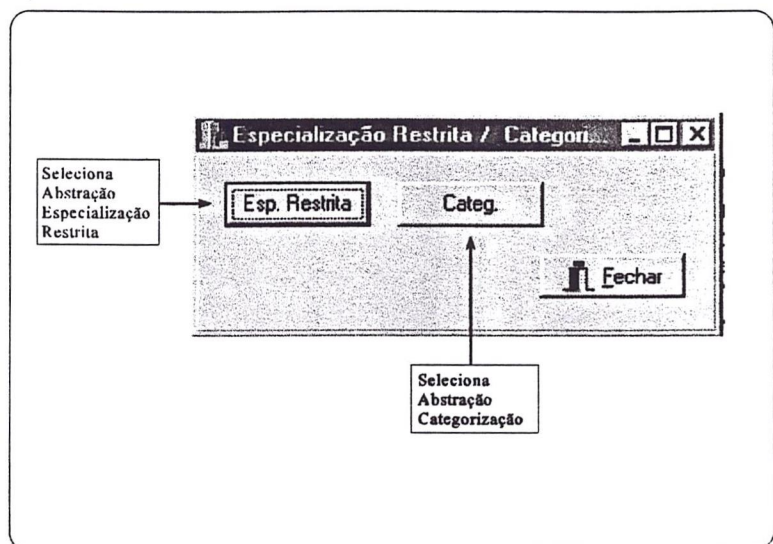


Figure 13: Especialização Restrita/Categorização.

## 2. 10 Formulário Especialização Restrita

O formulário Especialização Restrita, **Figura 14**, é editável o usuário irá utilizá-lo para criar e/ou alterar os objetos específicos. Este formulário é composto por nove interadores:

O diagrama mostra a interface do formulário 'Especialização Restrita' com as seguintes seções e interadores:

- Super tipo - Obj. Generalização:** Aponta para a seção 'Objeto generalização' que contém campos para 'Tipo do Objeto' (Meta Tipo Obj), 'Tipo da Colônia' (Global) e 'Nome do Objeto' (escola).
- Restrições:** Aponta para a seção 'Restrições' com opções de 'Sobreposição' (Disjunção, Sobreposição) e 'Participação' (Total, Parcial).
- Lista critério para especilaização:** Aponta para o campo 'Critério'.
- Sub tipo - Obj. Específico:** Aponta para a seção 'Objeto específico' com campos para 'Tipo do Objeto' (escola), 'Tipo da Colônia' (Global) e 'Nome do Objeto'.
- Predicado:** Aponta para o campo 'Predicado'.
- Inicialização:** Aponta para o campo 'Inicialização'.
- Remove objeto específico:** Aponta para o botão 'Obj. Espec.'.
- Armazena objeto específico:** Aponta para os botões 'Salvar' e 'Fechar'.

**Figure 14:** Formulário Especialização Restrita.

- Objeto generalização: este interador exibe para o usuário o super tipo da hierarquia de generalização. Os componentes Tipo do Objeto e Tipo da Colônia irão exibir as mesmas informações do objeto corrente. Por outro lado, o componente Nome do objeto lista os objetos que possuem o mesmo Tipo do Objeto e o mesmo Tipo da Colônia do objeto corrente, para que o usuário possa optar por qual dentre os objetos ele deseja criar e/ou alterar o(s) objeto(s) específico(s). Estes são criados segundo algumas restrições;
- Restrições: este interador permite que o usuário selecione os tipos de restrições que



devem ser levadas em consideração para a criação dos objetos específicos. As restrições são:

- Sobreposição - que pode ser:
  - Disjunção ou
  - Sobreposição;
- Participação - que pode ser:
  - Total ou
  - Parcial;
- Critério: é um interador editável. Nele o usuário irá digitar o nome do atributo (de classificação) que será o critério da especialização. Vale ressaltar que a implementação (desta primeira versão) foi feita de forma que apenas um atributo seja escolhido como critério para a criação dos objetos específicos. Uma vez armazenado na base, um critério de especialização poder ser listado neste interador para que o usuário crie e/ou altere o(s) objeto(s) específico(s) gerados a partir deste critério. Quando o usuário seleciona um critério na lista de critérios o interador Restrições exibe as restrições que foram armazenadas na base para o critério selecionado;
- Objeto específico: permite a criação e/ou alteração do(s) objeto(s) específico(s). O usuário deve definir apenas o(s) nome(s) do(s) objeto(s) específico(s), pois os interadores Tipo do Objeto e Tipo da Colônia são automaticamente preenchidos com os dados herdados do objeto generalização;
- Predicado: permite a edição do predicado;
- Inicialização: permite a edição da(s) inicialização(ões) caso esta(s) exista(m);
- o botão (-Obj. Espec.): remove o vínculo do objeto específico com o objeto generalização, ou seja, o objeto específico deixa de ser derivado do objeto generalização. No entanto, o objeto específico continua armazenado na base, como um objeto tipo;
- o botão (Salvar): durante a criação do(s) objeto(s) específico(s), o usuário tem a liberdade de mudar as restrições e/ou o critério de especialização, caso seja necessário. Porém uma vez definido como será efetuada a especialização do(s) objeto(s) todas as informações referentes a esta especialização serão armazenadas na base através o interador (Salvar);
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.

## 2. 11 Formulário Generalização Ampla

O formulário Generalização Ampla, **Figura 15**, é um formulário de consulta. Nele o usuário

The screenshot shows a window titled "Generalização Ampla". It contains several input fields and a table. Annotations with arrows point to specific parts of the form:

- Super tipo - Obj. Generalização** points to the "Objeto generalização" section, which includes "Tipo do Objeto", "Tipo da Colônia", and "Nome do Objeto" fields.
- Restrições** points to the "Restrições" section, which has two columns of radio buttons: "Sobreposição" and "Disjunção" in the first column, and "Participação" and "Parcial" in the second.
- Lista os dois tipos de especialização** points to the "Opção da Especialização" dropdown menu.
- Objeto(s) específico(s)** points to the "Objetos Específicos" table.
- Lista critério de especialização** points to the "Critério" dropdown menu.

The "Objetos Específicos" table has the following structure:

| OBJETO ESPECÍFICO | PREDICADO | INICIALIZAÇÃO |
|-------------------|-----------|---------------|
|                   |           |               |
|                   |           |               |
|                   |           |               |

At the bottom right of the window is a "Fechar" button.

**Figure 15:** Formulário Generalização Ampla.

poderá obter informações a respeito a hierarquia de Generalização/Especialização Restrita ou Generalização/Categorização. Por exemplo é possível verificar quais os subtipos (objetos específicos) estão associados ao supertipo (objeto generalização). Para isto, o usuário deverá selecionar o supertipo, a abstração (Especialização Restrita ou Categorização) que foi utilizada para a criação dos objetos específicos (subtipos) e também qual o critério utilizado para a especialização. Feito isto, as restrições que foram utilizadas para criar a hierarquia de Generalização/ Especialização Restrita ou Generalização/Categorização (dependendo do que o usuário quer consultar) são automaticamente exibidas, assim como o(s) objeto(s) específico(s) e seu(s) respectivo(s) predicado(s)



e inicialização(ões).

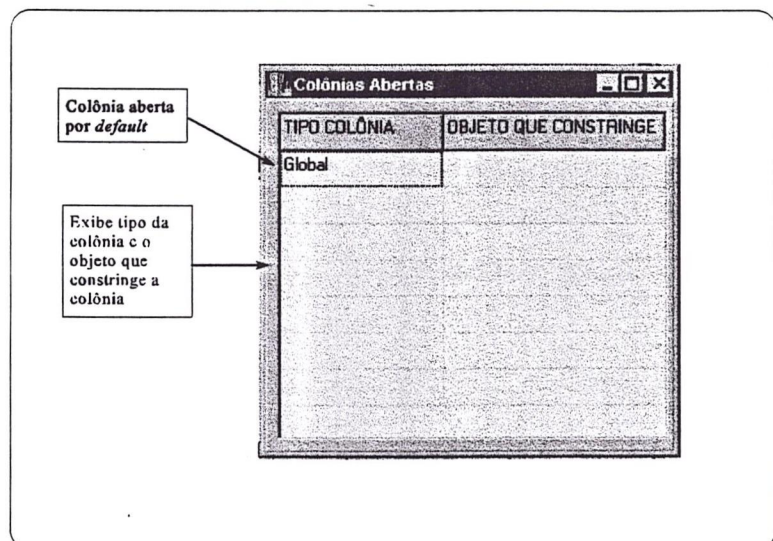
Este formulário é composto por alguns interadores, entre eles:

- Objeto generalização: onde os componentes Tipo do Objeto e Tipo da Colônia são automaticamente preenchidos com os mesmos dados do objeto corrente. Porém o nome do objeto deve ser selecionado pelo usuário. O componente Nome do Objeto lista todos os objetos que tiverem o mesmo Tipo do Objeto e o Tipo da Colônia que estiver sendo mostrado nestes componentes;
- Restrições: ilustra quais as restrições empregadas em uma hierarquia de Generalização;
- Opção de Especialização: este interador permite que o usuário selecione qual a hierarquia que deseja consultar, ou seja, uma hierarquia criada a partir das abstrações Generalização/Especialização Restrita ou Generalização/Categorização;
- Critério: é um interador editável onde o usuário irá informar qual o critério utilizado na hierarquia;
- Objetos Específicos: a partir das informações da hierarquia de generalização (setadas nos interadores anteriores) o usuário poderá consultar os objetos específicos neste interador;
- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.

## 2. 12 Formulário Colônias Abertas

O modelo impõe que os objetos devem ser criados em colônias que estejam abertas em tempo de execução. Sendo assim, o usuário necessita saber quais colônias estão abertas, antes de criar um objeto. Para fornecer esta informação foi criado o formulário Colônias Abertas, **Figura 16**.

Este é um formulário de consulta que está sempre disponível em tempo de execução. Ele contém apenas um interador, uma tabela, que exibe o Tipo da Colônia e o Objeto Que Constringe uma colônia aberta. Por *default*, a colônia do tipo Global sempre estará aberta, portanto sempre será mostrada, porém o objeto que constringe esta colônia não é ilustrado, pois este objeto pertence



**Figure 16:** Formulário Colônias Abertas.

ao MetaEsquema e não precisa ser explicitamente representado.

É o botão (Constringir) do formulário Objeto que irá criar uma colônia, caso ela ainda não exista, e abri-la. Desta forma, sempre que uma colônia for aberta, o formulário Colônias Abertas é atualizado. Este formulário não armazena dados na base, apenas exibe informações que são relevantes em tempo de execução. Se a execução do aplicativo for finalizada, estes dados são perdidos, pois não existirão colônias abertas.

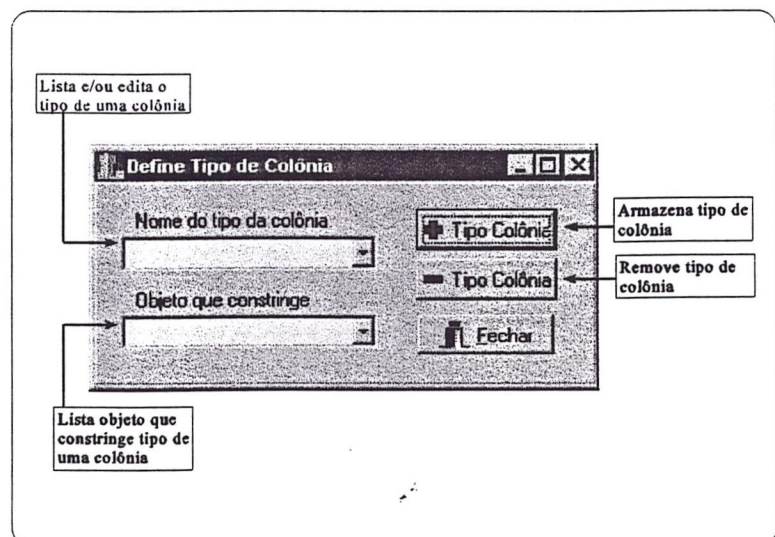


## 2. 13 Formulário Define Tipo Colônia

Uma colônia não tem nome. Ela possui apenas um tipo, pois objetos habitam colônias de um certo tipo, e objetos também constroem colônias de um certo tipo. Uma colônia é identificada por seu tipo e pelo nome do objeto que a constrói, uma vez que cada objeto constrói no máximo uma colônia de um dado tipo (porém colônias do mesmo tipo podem ser construídas por mais de um objeto). O tipo de uma colônia é criado através do formulário, Define Tipo de Colônia,

**Figura 17.** Este formulário possui os interadores:

- Nome do Tipo da Colônia: onde o usuário define o tipo de uma colônia. Os tipos de colônias que já estão definidos podem ser consultados utilizando-se também este interador;
- o botão (+Tipo Colônia): uma vez definido o tipo de uma colônia, este deve ser adicionado na base por meio do botão adiciona tipo colônia;
- o botão (-Tipo Colônia): ao contrário, permite que um tipo de colônia seja removido da base;
- O b j e t o q u e c o n s t r i n g e: este interador, não permite edição. Apenas irá exibir o objeto que constrói uma colônia cujo tipo é mostrado no interador Nome do Tipo da Colônia. O objeto que constrói um o tipo de uma colônia é determinado por meio do botão



**Figure 17:** Formulário Define Tipo de Colônia.

- o botão (Fechar): sai deste formulário e retorna para o formulário que o chamou.

É importante salientar que este formulário permite efetuar operações em nível de DHC (Diagrama Hierárquico de Colônia).

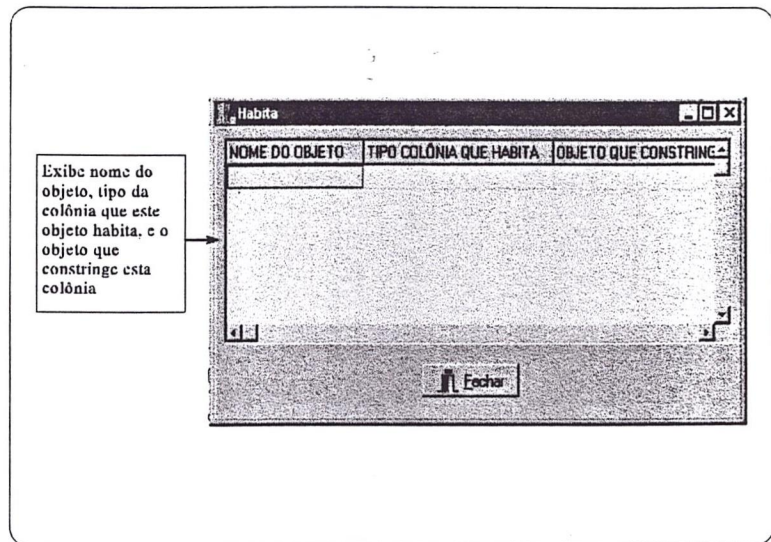
## 2. 14 Formulário Habita

O formulário Habita, **Figura 18**, é um formulário de consulta, cuja finalidade é informar ao usuário em quais colônias os objetos armazenados na base habitam. Estas informações são exibidas por meio de um interador que possui três colunas contendo:

- Nome do objeto: os objetos mostrados podem ser quaisquer armazenados na base, portanto, tanto objetos instâncias como objetos tipo;
- Tipo colônia que habita: exibe qual o tipo da colônia e onde o objeto habita (neste caso habita fisicamente);
- Objeto que restringe:

nome do que restringe uma colônia, em outras palavras, qual o tipo de colônia é constrita por este objeto;

A informação de qual objeto restringe um tipo de colônia é muito importante, uma vez que o modelo permite que exista mais de uma colônia com o mesmo tipo, que é constrita por objetos distintos. Sabendo-se também qual é o objeto que restringe uma colônia, podemos afirmar qual a colônia um dado objeto irá habitar. É importante salientar que este formulário permite consulta em nível de DRI (Diagrama de Representação de Instâncias).



**Figure 18:** Formulário Habita.



# Capítulo 3

## Considerações finais

---

---

O trabalho de implementação do Editor de Esquemas aqui descrito levou em consideração três aspectos principais:

- o suporte às abstrações integrada aos demais conceitos do modelo;
- ser um meio de facilitar a construção de modelagens feitas com um modelo altamente complexo;
- e constituir-se em uma ferramenta de suporte ao desenvolvimento do próprio gerenciador SIRIUS/GO

Editor foi desenvolvido de forma a não ter conjuntos de comandos/formulários diferenciados para uma parte de acesso:

- à definição do esquema (equivalente a uma DDL - *Data Definition Language*);
- e outra para a manipulação de dados (equivalente a uma DML - *Data Manipulation Language*).

Isso porque o modelo de dados suporta a abstração de classificação, permitindo que tipo de objeto seja um objeto com as mesmas propriedades, restrições e suporte do gerenciador de todos os demais objetos.

Vale salientar também que a complexidade inerente ao Editor de Esquemas E<sup>2</sup>SIRIUS torna difícil o treinamento de analistas que possam utilizá-lo para modelar sistemas. No entanto, na concepção deste houve a preocupação de ao mesmo tempo ser possível:

- editar os esquemas manipulando explicitamente os conceitos formais que levaram à especificação do modelo - resultando numa ferramenta de apoio ao aprendizado do modelo, quanto ser possível;
- a edição de esquemas realizada por projetistas já experientes - resultando numa ferramenta que suporta ao mesmo tempo a intuitividade no desenvolvimento de modelagens, quanto permite a edição eficiente de esquemas.

A implementação, da primeira versão do aplicativo, já levou em conta mudanças futuras, e

aumentando a flexibilidade do Editor, que pode ser facilmente configurado para suportar a construção de esquemas para outros modelos de dados que suportem subconjuntos dos conceitos suportados por SIRIUS, como é o caso da maioria dos modelos atuais desenvolvidos sobre o paradigma de orientação a objetos.



# Referências Bibliográficas

---

---

- [Araujo\_98] Araujo, M. R. B. - "Representação de Construtores Semânticos em SIIRIUS e Suporte através de um Editor de Esquemas", Dissertação de Mestrado em Ciências da Computação ICMC - USP, São Carlos - SP, Agosto de 1998.
- [Araujo\_98a] Araujo, M. R. B; Traina, C. Jr - "Editor de Esquemas com Suporte para Hierarquia de Classificação", XIII Simpósio Brasileiro de Banco de Dados, realizado em Maringá, de 12 a 14 de outubro de 1998.
- [Biajiz\_96] Biajiz, M. - "Representação de Modelos de Dados Orientados a Objetos através de Parametrização de Abstrações", Tese de Doutorado, São Carlos - SP, Setembro de 1996.
- [Elmasri\_94] Elmasri, R.; Navathe, S. B. - "Fundamentals of Database Systems", Addison-Wesley Publishing Company, 1994, Second Edition.
- [Ferreira\_96] Ferreira, J. E.; Traina, Jr, C. - "Controle de Compartilhamento e Acesso em GBDOO Baseado em Composição de Objetos", XI Simpósio Brasileiro de Banco de Dados, pp. 143 - 157, São Carlos, Outubro 1996.
- [Traina\_97] Traina Jr, C.; Ferreira, J. E.; Biajiz, M. - "Use of a Semantically Grained Database System for Distribution and Control within Design Environments", Lecture Notes on Computer Science nro 1300, pp. 1130 - 1135, Alemanha, Agosto de 1997.
- [Traina\_96] Traina Jr, C.; Traina A. - "Algoritmo para Criação e Gerenciamento de Identificadores de Objetos em um Gerenciador de Objetos", JAIO - 25<sup>o</sup> Jornadas Argentinas de Informática e Investigación Operativa, pp. 2.41 - 2.52, Septiembre 1996.