UNIVERSIDADE DE SÃO PAULO

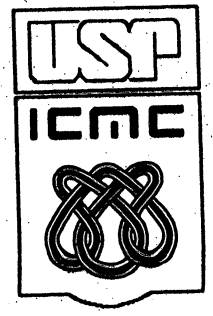
Instituto de Ciências Matemáticas e de Computação

PRIMITIVAS DO NÚCLEO DE GERENCIADOR
DE ESQUEMAS DE DADOS NO MODELO MRO

CAETANO TRAINA JUNIOR.
LUIZ CAMOLESI JUNIOR

Nº 10

RELATÓRIOS TÉCNICOS



São Carlos - SP

Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2509

Primitivas do Núcleo de Gerenciador de Esquemas de Dados no Modelo MRO*

CAETANO TRAINA JÚNIOR

LUIZ CAMOLESI JÚNIOR

Nº 010

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
maio / 1993

SYSNO 848145
DATA / /
ICMC - SBAB

Universidade de São Paulo
Instituto de Ciências Matemáticas de São Carlos
Departamento de Ciências de Computação e Estatística

PRIMITIVAS DO NÚCLEO DO GERENCIADOR DE ESQUEMAS DE DADOS NO MODELO MRO*

Caetano Traina Júnior

Luiz Camolesi Júnior

Endereço : Instituto de Ciências Matemáticas de São Carlos - ICMSC
Av. Dr. Carlos Botelho 1465
CEP. 13560.970 - Centro
São Carlos - São Paulo - Brasil
FAX: (0162)749150 - TEL: (0162)726222 RAMAL 3851

INTRODUÇÃO

O **Modelo de Representação de Objetos - MRO*** foi elaborado, no Departamento de Ciências de Computação e Estatística do ICMSC-USP em São Carlos no final de 1986, com o objetivo de permitir a construção de Sistemas de Gerenciamento de Bases de Dados que atendam às necessidades de manipulação de dados e representação de informações dos sistemas de apoio a projetos tradicionais de engenharia ("Computer Aided Design" - CAD), ou de ambientes de suporte ao desenvolvimento de sistemas ("Computer Aided Software Engineering" - CASE).

O desenvolvimento de um SGBDOO denominado **GErenciador de Objetos - GEO** iniciou-se em meados de 1988, com a incorporação gradativa dos conceitos do MRO*. Atualmente um dos objetivos desse sistema é a validação prática e o estudo de como os conceitos do MRO* e de Sistemas Orientados a Objetos em geral podem ser implantados em ferramentas reais.

Para a tarefa de implementar o Gerenciador de forma que este possa manipular eficientemente a Base Extensional e a Base Intencional, o GEO foi dividido em 5 módulos básicos :

1.) Núcleo de Acesso a Registros : emula um sistema de Gerenciamento de Arquivos Lógicos, fornecendo suporte aos demais módulos. É composto pelo Subsistema de Gerenciamento de Registros (SGR), Subsistema de Gerenc. de Arquivos (SGA) e Subsistema de Gerenciamento de Memória (SGM). Através destes módulos é realizado todo o gerenciamento de memória secundária, simulação de uma memória "cache" para disco em memória e controle de acesso e proteção dos dados, através do gerenciador de transações.

2.) Subsistema Básico de Gerenciamento de Objetos : possibilita a manipulação de Objetos sem restrições de ordem semântica. Envolve o tratamento de Identificadores de Objetos, manipulação de Composições de Objetos, contexto e navegação na Base.

3.) Subsistema Básico de Gerenciamento de Relacionamentos : permite a manipulação de Relacionamentos (binários ou triplos) definidos pelo usuário. O GEO gerencia Relacionamentos Intrínsecos e Relacionamentos Extrínsecos: os intrínsecos são aqueles cuja semântica é conhecida pelo Gerenciador, tal como as Hierarquias de Generalização e de Composição, sendo gerenciados pelo Subsistema Básico de Gerenciamento de Objetos; os extrínsecos são aqueles cuja semântica é definida pelo usuário e são gerenciados por este módulo em questão.

4.) Subsistema Básico de Gerenciamento de Atributos : permite a manipulação de Atributos com características de Propriedades e Comentários, tanto para Objetos quanto para Relacionamentos.

5.) Gerenciador de Esquemas de Dados - GED : consiste nas Primitivas de manipulação, acesso e verificação de Esquemas de Dados, as quais são ativadas pelos demais módulos.

Este manual apresenta uma descrição das primitivas do GED atualmente usadas pelo GEO na manipulação de Bases de Dados. Tais rotinas podem ser utilizadas nas versões GED/Mono-Esquema e GED/Multi-Esquema.

Implementadas em Linguagem C "padrão", tais rotinas foram projetadas para serem transportáveis para vários ambientes: DOS, UNIX, VMS, não necessitando de drásticas alterações em seus códigos fonte.

Como campo de testes das primitivas implementadas foi utilizado um editor experimental denominado GEO_INC. Vale ressaltar que todos os trabalhos envolvendo Bases de Dados do GEO são inicialmente implantados e testados antes de realmente comporem o núcleo do gerenciador e se tornarem disponíveis para a utilização.

Bibliografia Adicional

[CAMOLESI_92] Luiz Camolesi Júnior, "**Suporte a Acesso Multi-Usuário em Bases de Dados Orientadas a Objetos Através de Esquemas Suplementares**", Dissertação de Mestrado, ICMSC-USP, Novembro 1992.

[TRAINA_92] Caetano Traina Jr., Jan F. W. Slaets, "**MRO - Um Modelo de Representação de Objetos**", Notas Internas (104), ICMSC-USP, Fevereiro 1992.

[TRAINA_93] Caetano Traina Jr., Luiz Camolesi Júnior, "**Gerenciador de Esquemas de Dados Baseado no Modelo MRO***", Notas Internas (131), ICMSC-USP, Março 1993.

NOME

CarATR - Indica a característica de um Tipo de Atributo.

SINOPSE

```
#include <meta.h>

pal2 CarATR (Tatr, caract)
coditipo Tatr;
pal2 *caract;
```

DESCRIÇÃO

Esta rotina fornece a caract. de um Tipo de Atributo.

RETORNO

. valor (int) correspondente a característica.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Atributo no Esquema.
. deve existir a característica no Tipo de Atributo.

NOTA

O valor de "*caract" pode ser:

1 - Sinônimo
2 - Propriedade
3 - Comentário
4 - Tempo
5 - Estrutura
6 - Gráfico
7 - Regra
8 - Visualização
9 - Procedimento
10 - Imagem

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
808 - código não corresponde a um Tipo de Atributo.
837 - "característica" inexistente.

VEJA TAMBÉM

CodATR(3GEOSQU), DadPRO(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{       coditipo Tatr;
        pal2 caract,erro;
        Tatr = 63;
        erro = CarATR(Tatr,&caract);
.....
```

NOME

CodATR - Indica o código de um Tipo de Atributo.

SINOPSE

```
#include <meta.h>

pal2 CodATR (nome, Tatr)
cadeia nome;
coditipo *Tatr;
```

DESCRIÇÃO

Esta rotina fornece o código do Tipo de Atributo cujo nome é indicado em "nome".

RETORNO

. valor (long) correspondente ao código do Tipo de Atributo.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Atributo no Esquema.

NOTA

"nome" pode ser tanto uma cadeia normal (formato padrão CCMX&CO - cadeia com campo tamanho máximo e corrente) quanto uma cadeia formato "-.".

ERROS

0 - não há erro.
811 - nome não reconhecido no Esquema.
816 - nome não corresponde a um Tipo de Atributo.

VEJA TAMBÉM

NomATR(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tatr;
        erro = CodATR("-.idade",&Tatr);
.....
```

NOME

CodCOL - Indica o código de um Tipo de Colônia.

SINOPSE

```
#include <meta.h>

pal2 CodCOL (nome, Tcol)
cadeia nome;
coditipo *Tcol;
```

DESCRIÇÃO

Esta rotina fornece o código do Tipo de Colônia cujo nome é indicado em "nome".

RETORNO

. valor (long) correspondente ao código do Tipo de Colônia.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Colônia no Esquema.

NOTA

"nome" pode ser tanto uma cadeia normal (formato padrão CCMX&CO - cadeia com campo tamanho máximo e corrente) quanto uma cadeia formato "-.".

ERROS

0 - não há erro.
811 - nome não reconhecido no Esquema.
812 - nome não corresponde a um Tipo de Colônia.

VEJA TAMBÉM

NomCOL(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tcol;
        erro = CodCOL("-.projeto", &Tcol);
.....
```

NOME

CodMOD1 - Indica o código de uma Modalidade grau 1.

SINOPSE

```
#include <meta.h>

pal2 CodMOD1 (nome, Tmod1)
cadeia nome;
coditipo *Tmod1;
```

DESCRIÇÃO

Esta rotina fornece o código da Modalidade grau 1 cujo nome é indicado em "nome".

RETORNO

. valor (long) correspondente ao código da Modalidade grau 1.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir a Modalidade no Esquema.

NOTA

"nome" pode ser tanto uma cadeia normal (formato padrão CCMX&CO - cadeia com campo tamanho máximo e corrente) quanto uma cadeia formato "-.".

ERROS

0 - não há erro.
811 - nome não reconhecido no Esquema.
814 - nome não corresponde a uma Modalidade grau 1.

VEJA TAMBÉM

NomMOD1(3GEOSQU), OpsMOD1(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tmod1;
        erro = CodMOD1("-.trabalha",
        &Tmod1);
.....
```

NOME

CodMOD2 - Indica o código de uma Modalidade grau 2.

SINOPSE

```
#include <meta.h>

pal2 CodMOD2 (nome, Tmod2)
cadeia nome;
coditipo *Tmod2;
```

DESCRIÇÃO

Esta rotina fornece o código da Modalidade grau 2 cujo nome é indicado em "nome".

RETORNO

. valor (long) correspondente ao código da Modalidade grau 2.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir a Modalidade no Esquema.

NOTA

"nome" pode ser tanto uma cadeia normal (formato padrão CCMX&CO - cadeia com campo tamanho máximo e corrente) quanto uma cadeia formato "-.".

ERROS

0 - não há erro.
811 - nome não reconhecido no Esquema.
815 - nome não corresponde a uma Modalidade grau 2.

VEJA TAMBÉM

NomMOD2(3GEOSQU), CptMOD2(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tmod2;
        erro=CodMOD2("-.utiliza_p_trab",
        &Tmod2);
.....
```

NOME

CodOBJ - Indica o código de um Tipo de Objeto.

SINOPSE

```
#include <meta.h>

pal2 CodOBJ (nome, Tobj)
cadeia nome;
coditipo *Tobj;
```

DESCRIÇÃO

Esta rotina fornece o código do Tipo de Objeto cujo nome é indicado em "nome".

RETORNO

. valor (long) correspondente ao código do Tipo de Objeto.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Objeto no Esquema.

NOTA

"nome" pode ser tanto uma cadeia normal (formato padrão CCMX&CO - cadeia com campo tamanho máximo e corrente) quanto uma cadeia formato "-.".

ERROS

0 - não há erro.
811 - nome não reconhecido no Esquema.
813 - nome não corresponde a um Tipo de Objeto.

VEJA TAMBÉM

NomOBJ(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tobj;
        erro = CodOBJ("-.pessoa",&Tobj);
.....
```

NOME

CodSIN - Indica o código de um Tipo de Sinônimo.

SINOPSE

```
#include <meta.h>

pal2 CodSIN (nome, Tsin)
cadeia nome;
coditipo *Tsin;
```

DESCRIÇÃO

Esta rotina fornece o código do Tipo de Atributo com característica de Sinônimo cujo nome é indicado em "nome".

RETORNO

. valor (long) correspondente ao código do Tipo de Sinônimo.
. erro (vide ERROS).

RESTRIÇÕES

. o Tipo de Sinônimo deve ser um Tipo de Atributo com a característica de sinônimo.
. deve existir o Tipo de Atributo no Esquema.

NOTA

"nome" pode ser tanto uma cadeia normal (formato padrão CCMX&CO - cadeia com campo tamanho máximo e corrente) quanto uma cadeia formato "-.".

ERROS

0 - não há erro.
811 - nome não reconhecido no Esquema.
817 - nome não corresponde a um Tipo de Sinônimo.
838 - "característica" inexistente.

VEJA TAMBÉM

NomSIN(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tsin;
        erro = CodSIN("-.CIC",&Tsin);
.....
```

NOME

CptMOD2 - Indica os componentes da Modal. grau 2.

SINOPSE

```
#include <meta.h>

pal2 CptMOD2(Tmod2,Tmodl_deter,
Tmodl_conj)
coditipo Tmod2, *Tmodl_deter,
        *Tmodl_conj;
```

DESCRIÇÃO

Esta rotina indica os códigos das Modalidades grau 1 que compoem (determina e conjunto) a Modalidade grau 2 fornecida.

RETORNO

. valores (long) correspondentes aos códigos das Modalidades grau 1.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir a Modalidade grau 2 no Esquema.
. A Modalidade grau 2 deve ser composta por duas Modalidades grau 1.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
807 - código não corresponde a uma Modalidade grau 2.
836 - Modalidade grau 2 não é composta por duas Modalidade grau 1.

VEJA TAMBÉM

OpsMOD1(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo   Tmod2,   *Tmodl_deter,
        *Tmodl_conj;
        erro = CptMOD2 (Tmod2,
        &Tmodl_deter, &Tmodl_conj);
.....
```

NOME

CtgCOL - Indica o Tipo de Objeto que constringe o Tipo de Colônia.

SINOPSE

```
#include <meta.h>

pal2 CtgCOL (Tcol, Tobj)
coditipo Tcol, *Tobj;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Objeto que constringe o Tipo de Colônia fornecido.

RETORNO

. valor (long) correspondente ao código do Tipo de Objeto.

. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Colônia no Esquema.

. deve existir o relacionamento "constringe" entre o Tipo de Colônia fornecido e um Tipo de Objeto contido no Esquema.

ERROS

0 - não há erro.

801 - código não corresponde a uma peça do Esquema.

802 - código não corresponde a um Tipo de Colônia.

831 - Tipo de Colônia não é constrito por nenhum Tipo de Objeto.

VEJA TAMBÉM

HabOBJ(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tcol, *Tobj;
        Tcol = 70;
        erro = CtgCOL(Tcol,&Tobj);
.....
```

NOME

DadPRO - Indica o "tipo de dado" de um Tipo de Propriedade.

SINOPSE

```
#include <meta.h>

pal2 DadPRO (Tpro, tipodado)
coditipo Tpro;
pal2 *tipodado;
```

DESCRIÇÃO

Esta rotina indica o "tipo de dado" de um Tipo de Atributo com característica de propriedade.

RETORNO

. valor (int) correspondente ao "tipo de dado".
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Atributo no Esquema
. o Tipo de Atributo deve ter a carac. de propriedade.

NOTA

O valor de "*tipodado" pode ser:

1 - byte
2 - int
3 - long
4 - coditipo
5 - float
6 - double
7 - cadeia
8 - 2dcoordenada
9 - data
10 - hora
11 - valor

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
810 - código não corresponde a um Tipo de Propriedade.
839 - "tipo de dado" inexistente.

VEJA TAMBÉM

CarATR(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{       pal2 erro, tipodado;
        coditipo Tpro;
        erro = DadPRO(Tpro,&tipodado);
```

NOME

HabOBJ - Indica o Tipo de Colônia habitada pelo Tipo de Objeto.

SINOPSE

```
#include <meta.h>

pal2 HabOBJ (Tobj, Tcol)
coditipo Tobj, *Tcol;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Colônia habitada pelo Tipo de objeto fornecido.

RETORNO

. valor (long) correspondente ao código do Tipo de Colônia.

. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Objeto no Esquema.

. o Tipo de Objeto fornecido deve habitar um Tipo de Colônia.

ERROS

0 - não há erro.

801 - código não corresponde a uma peça do Esquema.

803 - código não corresponde a um Tipo de Objeto.

832 - Tipo de Objeto não habita nenhum Tipo de Colônia.

VEJA TAMBÉM

CtgCOL(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tobj, Tcol;
        Tobj = 75;
        erro = HabOBJ(Tobj,&Tcol);
.....
```

NOME

IdfBIN - Indica o Tipo de Objeto identificado pelo Tipo de Relac. Binário.

SINOPSE

```
#include <meta.h>

pal2 IdfBIN (Tbin, Tobj)
coditipo Tbin, *Tobj;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Objeto identificado pelo Tipo de Relacionamento Binário fornecido.

RETORNO

. valor (long) correspondente ao código do Tipo de Objeto.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Relacionamento Binário no Esquema.
. o Tipo de Relac. Binário deve identificar um Tipo de Objeto.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
804 - cód. não corresponde a um Tipo de Relac. Binário
818 - Tipo de Relac. Binário não identifica nenhum Tipo de Objeto.

VEJA TAMBÉM

IdfOBJ(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tbin, Tobj;
        Tbin = 80;
        erro = IdfBIN(Tbin,&Tobj);
.....
```

NOME

IdfOBJ - Indica o Tipo de Relac. Binário que identifica o Tipo de Objeto.

SINOPSE

```
#include <meta.h>

pal2 IdfOBJ (Tobj, Tbin)
coditipo Tobj, *Tbin;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Relacionamento Binário que identifica o Tipo de Objeto fornecido.

RETORNO

. valor (long) correspondente ao código do Tipo de Relac. Binário.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Objeto no Esquema.
. o Tipo de Objeto deve ser identificado por um Tipo de Relac. Binário.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
803 - código não corresponde a um Tipo de Objeto
818 - Tipo de Objeto não é identificado por nenhum Tipo de Relac. Binário.

VEJA TAMBÉM

IdfBIN(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tbin, Tobj;
        Tobj = 85;
        erro = IdfOBJ(Tobj,&Tbin);
.....
```

NOME

IdxCOL - Indica o índice interno de um Tipo de Colônia.

SINOPSE

```
#include <meta.h>

pal2 IdxCOL (Tcol, indice)
coditipo Tcol
pal2 *indice;
```

DESCRIÇÃO

Esta rotina indica o índice utilizado na alocação da Colônia do Tipo fornecido.

RETORNO

. valor (int) correspondente ao índice do Tipo de Colônia.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Colônia no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
802 - código não corresponde a um Tipo de Colônia.
831 - "índice" inexistente.

EXEMPLO

```
#include <meta.h>
main()
{
      pal2 erro, indice;
      coditipo Tcol;
      Tcol = 90;
      erro = IdxCOL(Tcol,&indice);
.....
```

NOME

MulAOB - Indica as multiplicidades de um Tipo de Atributo.

SINOPSE

```
#include <meta.h>

pal2 MulAOB (Tobj, Tatr, min, max)
coditipo Tobj, Tatr;
pal2 *min, *max;
```

DESCRIÇÃO

Esta rotina indica as multiplicidades mínima e máxima do Tipo de Atributo relacionado ("atributo de") com o Tipo de Objeto fornecido.

RETORNO

. valores (int) correspondentes as multiplicidades.
. erro (vide ERROS).

RESTRIÇÕES

. devem existir os Tipos de Objeto e Atributo no Esquema.
. o Tipo de Atributo deve ser atributo do Tipo de Objeto.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
803 - código não corresponde a um Tipo de Objeto.
808 - código não corresponde a um Tipo de Atributo.
818 - Tipo de Atributo não é atributo do Tipo de Objeto.

VEJA TAMBÉM

MulARB(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro,min,max;
        coditipo Tobj, Tatr;
        Tobj = 85; Tatr = 90;
        erro = MulAOB(Tobj, Tatr, &min,
        &max);
.....
```

NOME

MulARB - Indica as multiplicidades de um Tipo de Atributo.

SINOPSE

```
#include <meta.h>

pal2 MulARB (Tbin, Tatr, min, max)
coditipo Tbin, Tatr;
pal2 *min, *max;
```

DESCRIÇÃO

Esta rotina indica as multiplicidades mínima e máxima do Tipo de Atributo relacionado ("atributo de") com o Tipo de Relacionamento Binário fornecido.

RETORNO

. valores (int) correspondentes as multiplicidades.
. erro (vide ERROS).

RESTRIÇÕES

. devem existir os Tipos de Relac. Binário e Atributo no Esquema.
. o Tipo de Atributo deve ser atributo do Tipo de Relac. Binário.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
804 - código não corresponde a um Tipo de Relac. Binário.
808 - código não corresponde a um Tipo de Atributo.
818 - Tipo de Atributo não é atributo do Tipo de Relac. Binário.

VEJA TAMBÉM

MulART(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro,min,max;
        coditipo Tbin, Tobj;
        Tbin = 85; Tatr = 90;
        erro = MulARB(Tbin,Tatr,&min,
        &max);
.....
```

NOME

MulART - Indica as multiplicidades de um Tipo de Atributo.

SINOPSE

```
#include <meta.h>

pal2 MulART (Ttri, Tatr, min, max)
coditipo Ttri, Tatr;
pal2 *min, *max;
```

DESCRIÇÃO

Esta rotina indica as multiplicidades mínima e máxima do Tipo de Atributo relacionado ("atributo de") com o Tipo de Relacionamento Triplo fornecido.

RETORNO

. valores (int) correspondentes as multiplicidades.
. erro (vide ERROS).

RESTRIÇÕES

. devem existir os Tipos de Relac. Triplo e Atributo no Esquema.
. o Tipo de Atributo deve ser atributo do Tipo de Relac. Triplo.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
805 - código não corresponde a um Tipo de Relac. Triplo.
808 - código não corresponde a um Tipo de Atributo.
818 - Tipo de Atributo não é atributo do Tipo de Relac. Triplo.

VEJA TAMBÉM

MulAOB(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro,min,max;
        coditipo Ttri, Tatr;
        Ttri = 85; Tatr = 90;
        erro = MulART(Ttri,Tatr,&min,
        &max);
.....
```

NOME

MulBIN - Indica as multiplicidades de um Tipo de Relac. Binário.

SINOPSE

```
#include <meta.h>

pal2 MulBIN (Tbin, min, max)
coditipo Tbin;
pal2 *min, *max;
```

DESCRIÇÃO

Esta rotina indica as multiplicidades mínima e máxima do Tipo de Relacionamento Binário.

RETORNO

. valores (int) correspondentes as multiplicidades.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Relac. Binário no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
804 - código não corresponde a um Tipo de Relac. Binário.

VEJA TAMBÉM

ObjBIN(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro,min,max;
        coditipo Tbin;
        Tbin = 100;
        erro = MulBIN(Tbin,&min,&max);
.....
```

NOME

MulSIN - Indica as multiplicidades de um Tipo de Sinônimo.

SINOPSE

```
#include <meta.h>

pal2 MulSIN (Tobj,Tsin, min, max)
coditipo Tobj, Tsin;
pal2 *min, *max;
```

DESCRIÇÃO

Esta rotina indica as multiplicidades mínima e máxima do Tipo de Sinônimo associado ("identifica") ao Tipo de Objeto fornecido.

RETORNO

. valores (int) correspondentes as multiplicidades.
. erro (vide ERROS).

RESTRIÇÕES

. devem existir os Tipos de Atributo e Objeto no Esquema.
. Tipo de Atributo deve ser sinônimo de tipo de Objeto.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
803 - código não corresponde a um Tipo de Objeto.
809 - código não corresponde a um tipo de Sinônimo.
818 - Tipo de Sinônimo não identifica o Tipo de Objeto.
838 - "característica" inexistente.

VEJA TAMBÉM

MulAOB(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro,min,max;
        coditipo Tobj, Tsin;
        Tobj = 100; Tsin = 105
        erro = MulSIN(Tobj,Tsin,&min,
        &max);
.....
```

NOME

MulTRI - Indica as multiplicidades de um Tipo de Relac. Triplo.

SINOPSE

```
#include <meta.h>

pal2 MulTRI (Ttri, min, max)
coditipo Ttri;
pal2 *min, *max;
```

DESCRIÇÃO

Esta rotina indica as multiplicidades mínima e máxima do Tipo de Relac. Triplo.

RETORNO

. valores (int) correspondentes as multiplicidades.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Relac. Triplo no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
805 - código não corresponde a um Tipo de Relac Triplo.

VEJA TAMBÉM

ObjTRI(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro,min,max;
        coditipo Ttri;
        Ttri = 100;
        erro = MulTRI(Ttri,&min,&max);
.....
```

NOME

NomATR - Indica o nome do Tipo de Atributo.

SINOPSE

```
#include <meta.h>

pal2 NomATR (Tatr,nome)
coditipo Tatr;
cadeia nome;
```

DESCRIÇÃO

Esta rotina indica o nome do Tipo de Atributo fornecido.

RETORNO

. valor (cadeia) correspondente ao nome do Tipo de Atributo.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Atributo no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
808 - código não corresponde a um Tipo de Atributo.

VEJA TAMBÉM

CodATR(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        cadeia nome;
        pal2 erro;
        coditipo Tatr;
        Tatr = 110;
        erro = NomATR(Tatr,nome);
.....
```

NOME

NomCOL - Indica o nome do Tipo de Colônia.

SINOPSE

```
#include <meta.h>

pal2 NomCOL (Tcol,nome)
coditipo Tcol;
cadeia nome;
```

DESCRIÇÃO

Esta rotina indica o nome do Tipo de Colônia fornecido.

RETORNO

. valor (cadeia) correspondente ao nome do Tipo de Colônia.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Colônia no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
802 - código não corresponde a um Tipo de Colônia.

VEJA TAMBÉM

CodCOL(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        cadeia nome;
        pal2 erro;
        coditipo Tcol;
        Tcol = 110;
        erro = NomCOL(Tcol,nome);
.....
```

NOME

NomMOD1 - Indica o nome da Modalidade grau 1.

SINOPSE

```
#include <meta.h>

pal2 NomMOD1 (Tmod1,nome)
coditipo Tmod1;
cadeia nome;
```

DESCRIÇÃO

Esta rotina indica o nome da Modalidade grau 1.

RETORNO

. valor (cadeia) correspondente ao nome da Modalidade grau 1.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir a Modalidade grau 1 no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
806 - código não corresponde a uma Modalidade grau 1.

VEJA TAMBÉM

CodMOD1(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        cadeia nome;
        pal2 erro;
        coditipo Tmod1;
        Tmod1 = 110;
        erro = NomMOD1(Tmod1,nome);
.....
```

NOME

NomMOD2 - Indica o nome da Modalidade grau 2.

SINOPSE

```
#include <meta.h>

pal2 NomMOD2 (Tmod2,nome)
coditipo Tmod2;
cadeia nome;
```

DESCRIÇÃO

Esta rotina indica o nome da Modalidade grau 2.

RETORNO

. valor (cadeia) correspondente ao nome da Modalidade grau 2.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir a Modalidade grau 2 no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
807 - código não corresponde a uma Modalidade grau 2.

VEJA TAMBÉM

CodMOD2(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        cadeia nome;
        pal2 erro;
        coditipo Tmod2;
        Tmod2 = 110;
        erro = NomMOD2(Tmod2,nome);
.....
```

NOME

NomOBJ - Indica o nome do Tipo de Objeto.

SINOPSE

```
#include <meta.h>

pal2 NomOBJ (Tobj,nome)
coditipo Tobj;
cadeia nome;
```

DESCRIÇÃO

Esta rotina indica o nome do Tipo de Objeto fornecido.

RETORNO

. valor (cadeia) correspondente ao nome do Tipo de Objeto.

. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Objeto no Esquema.

ERROS

0 - não há erro.

801 - código não corresponde a uma peça do Esquema.

803 - código não corresponde a um Tipo de Objeto.

VEJA TAMBÉM

CodOBJ(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        cadeia nome;
        pal2 erro;
        coditipo Tobj;
        Tobj = 110;
        erro = NomOBJ(Tobj,nome);
.....
```

NOME

NomSIN - Indica o nome do Tipo de Sinônimo.

SINOPSE

```
#include <meta.h>

pal2 NomSIN (Tsin,nome)
coditipo Tsin;
cadeia nome;
```

DESCRIÇÃO

Esta rotina indica o nome do Tipo de Atributo com característica de sinônimo.

RETORNO

. valor (cadeia) correspondente ao nome do Tipo de Sinônimo.

. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Atributo no Esquema.

ERROS

0 - não há erro.

801 - código não corresponde a uma peça do Esquema.

809 - código não corresponde a um Tipo de Sinônimo.

838 - "característica" inexistente.

VEJA TAMBÉM

CodSIN(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        cadeia nome;
        pal2 erro;
        coditipo Tsin;
        Tsin = 110;
        erro = NomSIN(Tsin,nome);
.....
```

NOME

ObjBIN - Indica o destino de um Tipo de Relac. Binário.

SINOPSE

```
#include <meta.h>

pal2 ObjBIN (Torigem, Tmod1, Tdestino)
coditipo Torigem, Tmod1,
        *Tdestino;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Objeto destino que se relaciona "origem de" com um Tipo de Relacionamento Binário que está associado com a Modalidade grau 1 e o tipo de Objeto origem.

RETORNO

. valor (long) correspondente ao código do Tipo de objeto destino.
. erro (vide ERROS).

RESTRIÇÕES

. devem existir os Tipos de Objeto origem e Modalidade grau 1 no Esquema.
. deve existir um único Tipo de Objeto destino como possível retorno desta rotina.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
803 - código não corresponde a um Tipo de Objeto.
806 - código não corresponde a uma Modalidade grau 1.
818 - Tipo de Objeto origem não se relaciona com Modalidade grau 1.
819 - existe mais de um Tipo de Objeto destino.

VEJA TAMBÉM

MulBIN(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Torigem,Tdestino,Tmod1;
        Torigem = 110;  Tmod1 = 115;
        erro = ObjBIN(Torigem,Tmod1,
        &Tdestino);
.....
```

NOME

ObjTRI - Indica a divisão 2 de um Tipo de Relac. Triplo.

SINOPSE

```
#include <meta.h>

pal2 ObjTRI (Ttri, Tobj_div2)
coditipo Ttri, *Tobj_div2;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Objeto que se relaciona "divisão 2" com o Tipo de Relacionamento Triplo.

RETORNO

. valor (long) correspondente ao código do Tipo de objeto divisão 2.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir o Tipo de Relac. Triplo no Esquema.
. deve existir o relac. "divisão 2" entre o Tipo de Relac. Triplo e um Tipo de Objeto.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
803 - código não corresponde a um Tipo de Objeto.
805 - código não corresponde a um Tipo de Relac. Triplo.
818 - Tipo de Relac. Triplo não se relaciona com Tipo de Objeto através da "divisão 2".

VEJA TAMBÉM

MulTRI(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Ttri, Tobj_div2;
        Ttri = 110;
        erro = ObjTRI(Ttri,&Tobj_div2);
.....
```

NOME

OpsMOD1 - Indica o oposto de uma Modalidade grau 1.

SINOPSE

```
#include <meta.h>

pal2 OpsMOD1 (Tmod1, Toposto)
coditipo Tmod1, *Toposto;
```

DESCRIÇÃO

Esta rotina indica a Modalidade oposta da Modalidade grau 1 fornecida.

RETORNO

. valor (long) correspondente ao código da Modaldade grau 1 oposta.
. erro (vide ERROS).

RESTRIÇÕES

. deve existir a Modalidade grau 1 no Esquema.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
806 - código não corresponde a uma Modalidade grau 1.
835 - Modalidade oposta não encontrada.

VEJA TAMBÉM

NomMOD1(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tmod1, Toposto;
        Tmod1 = 110;
        erro = OpsMOD1(Tmod1,&Toposto);
.....
```

NOME

RelBIN - Indica o Tipo de Relac. Binário.

SINOPSE

```
#include <meta.h>

pal2 RelBIN (Torigem, Tmodl, Tdestino,
Tbin)
coditipo Torigem, Tdestino, Tmodl,
        *Tbin;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Relacionamento Binário associado aos Tipo de Objeto origem, destino e a Modalidde grau 1.

RETORNO

. valor (long) correspondente ao código do Tipo de Relac. Binário.
. erro (vide ERROS).

RESTRIÇÕES

. devem existir os Tipos de Objeto origem, destino e Modalidade grau 1.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
803 - código não corresponde a um Tipo de Objeto.
806 - código não corresponde a uma Modalidade grau 1.
818 - Tipo de Relacionamento Binário não encontrado.

VEJA TAMBÉM

MulBIN(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo Tmodl, Torigem, Tdestino,
        Tbin;
        Torigem = 120; Tdestino = 130;
        Tmodl = 110;
        erro = RelBIN(Torigem,Tmodl,
        Tdestino, &Tbin);
.....
```

NOME

RelTRI - Indica o Tipo de Relac. Triplo.

SINOPSE

```
#include <meta.h>

pal2 RelTRI (Tobj_div1, Tmod2, Tobj_div3,
Ttri)
coditipo Tobj_div1, Tobj_div3, Tmod2,
        *Ttri;
```

DESCRIÇÃO

Esta rotina indica o Tipo de Relacionamento Triplo associado aos Tipo de Objeto divisão 1, divisão 3 e a Modalidde grau 2.

RETORNO

. valor (long) correspondente ao código do Tipo de Relac. Triplo.
. erro (vide ERROS).

RESTRIÇÕES

. devem existir os Tipos de Objeto divisão 1, divisão 3 e Modalidade grau 2.

ERROS

0 - não há erro.
801 - código não corresponde a uma peça do Esquema.
803 - código não corresponde a um Tipo de Objeto.
807 - código não corresponde a uma Modalidade grau 2.
818 - Tipo de Relacionamento Triplo não encontrado.

VEJA TAMBÉM

MulTRI(3GEOSQU).

EXEMPLO

```
#include <meta.h>
main()
{
        pal2 erro;
        coditipo    Tmod2,    Tobj_div1,
        Tobj_div3, Ttri;
        Tobj_div1 = 120; Tobj_div3 = 130;
        Tmod2 = 110;
        erro = RelTRI(Tobj_div1,Tmod2,
        Tobj_div3,&Ttri);
.....
```

NOME

G_INICIALIZA_ESQUEMA - Inicializa a estrutura do Esquema de Dados.

SINOPSE

```
#include <meta.h>
#include <stresqu.h>

G_INICIALIZA_ESQUEMA()
```

DESCRIÇÃO

Esta rotina inicializa o vetor que indica os Esquemas Suplementares contidos no Esquema de Dados.

RESTRIÇÕES

. deve ser usado somente uma vez quando o usuário inicia seu acesso a Base de Dados.

NOTA

A estrutura de vetor a seguir possui 20 posições nos quais são indicados os Esquemas Suplementares usados pelos usuários da Base de Dados:

```
extern struct Esq_usados {
        codiobjeto codigo;
        int        n_tipos,
                   n_div2,
                   n_atrde;
} VEsquema[20];
```

VEJA TAMBÉM

G_REQUISITA_ESQUEMA(3GEOSQU), G_LIBERA_ESQUEMA(3GEOSQU).

EXEMPLO

```
#include <meta.h>
#include <stresqu.h>
main()
{
        G_INICIALIZA_ESQUEMA();
.....
```

NOME

G_REQUISITA_ESQUEMA - Inclui um Esquema Suplementar no vetor que corresponde ao Esquema de Dados.

SINOPSE

```
#include <meta.h>
#include <stresqu.h>

G_REQUISITA_ESQUEMA(cod_objeto)
codiobjeto cod_objeto;
```

DESCRIÇÃO

Esta rotina inclui o Esquema Suplementar, indicado pelo código do Objeto (cod_objeto) que o constringe, no Esquema de Dados ativo pelo usuário.

RETORNO

. erro (vide ERROS).

RESTRIÇÕES

. deve existir o objeto que constringe o Esquema Suplementar.

. O Esquema de Dados resultante não deve ficar inconsistente.

ERROS

0 - não há erro.

851 - Esquema Suplementar já está em uso.

852 - Requisição não pode ser realizada (Inconsistência)

859 - Requisição não pode ser realizada (Algum Esquema Suplementar protegido).

VEJA TAMBÉM

G_LIBERA_ESQUEMA(3GEOSQU).

EXEMPLO

```
#include <meta.h>
#include <stresqu.h>
main()
{
        pal2 erro;
        codiobjeto cod_objeto;
        cod_objeto = 70;
        erro = G_REQUISITA_ESQUEMA
                        (cod_objeto);
.....
```

NOME

G_LIBERA_ESQUEMA - Retira um Esquema Suplementar do vetor que corresponde ao Esquema de Dados.

SINOPSE

```
#include <meta.h>
#include <stresqu.h>

G_LIBERA_ESQUEMA(cod_objeto)
codiobjeto cod_objeto;
```

DESCRIÇÃO

Esta rotina retira o Esquema Suplementar, indicado pelo código do Objeto (cod_objeto) que o constringe, no Esquema de Dados ativo pelo usuário.

RETORNO

. erro (vide ERROS).

RESTRIÇÕES

. O Esquema Suplementar indicado pelo Objeto deve estar ativo no Esquema de Dados.

. O Esquema de Dados resultante não deve ficar inconsistente.

ERROS

0 - não há erro.

853 - Liberação não pode ser realizada (Inconsistência).

857 - Esquema Suplementar não está em uso.

859 - Liberação não pode ser realizada (Algum Esquema Suplementar protegido).

VEJA TAMBÉM

G_REQUISITA_ESQUEMA(3GEOSQU).

EXEMPLO

```
#include <meta.h>
#include <stresqu.h>
main()
{
        pal2 erro;
        codiobjeto cod_objeto;
        cod_objeto = 70;
        erro = G_LIBERA_ESQUEMA
                        (cod_objeto);
.....
```