



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-10498-PRE/5975

MANUAL DE REFERÊNCIA - LEGAL

Cláudio Clemente Faria Barbosa
Cordeiro, João Pedro Cerveira Cordeiro

INPE
São José dos Campos
2001

A

Manual de Referência - LEGAL

Cláudio Clemente Faria Barbosa

João Pedro Cerveira Cordeiro

Este apêndice apresenta a você a sintaxe dos comandos e as palavras reservadas da linguagem LEGAL.

OBS: As palavras reservadas da linguagem aparecem em negrito.

Declaração

Imagem

Declara uma variável do tipo **Imagem** que será posteriormente associada a um novo PI criado no programa ou um PI já existente no projeto corrente. Esta variável poderá ser utilizada em alguma operação do programa.

Sintaxe:

```
Imagem variável ("categoria");
```

Considerações:

O primeiro termo indica que geo-campos do tipo **Imagem** poderão ser associado à variável durante a execução do programa. A categoria entre aspas e parênteses, indica a categoria definida no banco de dados ativo. Mais de um nome de variável pode ser simultaneamente declarados para uma mesma categoria.

Exemplos:

```
Imagem banda3, banda4, ivdn ("LANDSAT");
```

```
Imagem banda3 ("LANDSAT"), xp2 ("SPOT");
```

```
Imagem B3 ("ImagensTM");
```

Numerico

Declara uma variável do tipo **Numerico** (ou digital) que será posteriormente associada a um novo PI criado no programa ou um PI já existente no projeto corrente. Esta variável poderá ser utilizada em alguma operação do programa.

Sintaxe:

```
Numerico variável ("categoria");
```

Considerações:

O primeiro termo indica que geo-campos do tipo **Numerico** poderão ser associado à variável durante a execução do programa. A categoria entre aspas e parênteses, indica a categoria definida no banco de dados ativo. Mais de um nome de variável pode ser simultaneamente declarados para uma mesma categoria.

Exemplos:

```
Numerico GR1, GR2, GR3 ("Grade_Altimetria");  
Numerico ALTI ("ALTIMETRIA"), exp("EXPOSICAO");  
Numerico grade2 ("Grades_declividade");
```

Tematico

Declara uma variável do tipo **Tematico** que será posteriormente associada a um novo PI criado no programa ou um PI já existente no projeto corrente. Esta variável poderá ser utilizada em alguma operação do programa.

Sintaxe:

```
Tematico variável ("categoria");
```

Considerações:

O primeiro termo indica que geo-campos do tipo **Tematico** poderão ser associado à variável durante a execução do programa. A categoria entre aspas e parênteses, indica a categoria definida no banco de dados ativo. Mais de um nome de variável pode ser simultaneamente declarados para uma mesma categoria.

Exemplos:

```
Tematico USO1, USO2, USO3 ("USO_TERRA");  
Tematico Solo1 ("SOLOS"), GEO("Geologia");  
Tematico dec2 ("Declividade");
```

Objeto

Declara uma variável do tipo **objeto** que será posteriormente usada em uma operação do tipo reclassificação por atributos.

Sintaxe:

```
Objeto variável ("categoria");
```

Considerações:

O primeiro termo indica que entidades do tipo **objeto** poderão ser associado à variável durante a execução do programa. A categoria entre aspas e parênteses, indica a categoria definida no banco de dados ativo. Mais de um nome de variável pode ser simultaneamente declarados para uma mesma categoria.

Exemplos:

```
Objeto lot1, lot2, lot3 ("Lotes_urbanos");  
Objeto CID ("Cidades"), FAZ ("Fazendas");  
Objeto Post ("Postes_eletricos");
```

Cadastral

Declara uma variável do tipo **cadastral** que será posteriormente associada a um novo PI criado no programa ou um PI já existente no projeto corrente. Esta variável poderá ser utilizada em alguma operação do programa.

Sintaxe:

```
Cadastral variável ("categoria");
```

Considerações:

O primeiro termo indica que entidades do tipo **Cadastral** poderão ser associado à variável durante a execução do programa. A categoria entre aspas e parênteses, indica a categoria definida no banco de dados ativo. Mais de um nome de variável pode ser simultaneamente declarados para uma mesma categoria.

Exemplos:

```
Cadastral cad1 ("Cadastro_urbano");  
Cadastral map_AS ("Mapa_Paises");
```

Tabela

Declara uma variável do tipo tabela que será usadas para definir transformações entre campos e objetos de diferentes tipos.

Sintaxe:

```
Tabela variável (tipo_de_transformação);
```

Considerações:

Numa declaração de tabela, o primeiro termo deverá conter a palavra reservada **Tabela**, seguida do nome da variável e de uma palavra chave indicando o tipo de transformação - **Reclassificacao**, **Fatiamento** ou **Ponderacao**. Mais de um nome de variável pode ser simultaneamente declarados para um mesmo tipo de transformação.

Uma tabela de **Reclassificação** define o mapeamento entre classes temáticas de duas diferentes categorias, no **Fatiamento** um campo de uma categoria numérica é transformado em um campo temático, enquanto numa **Ponderação** a transformação é de um campo temático para um campo numérico.

Exemplos:

```
Tabela USO1, USO2, USO3 (Reclassificacao);
```

```
Tabela Solo1(Ponderacao), GEO(Ponderacao);
```

```
Tabela dec2 (Fatiamento);
```

Atributo

Declara uma variável do tipo **atributo** que será posteriormente usada em uma operação do tipo reclassificação por atributos.

Sintaxe:

```
Atributo variável ("categoria_objeto", "atributo");
```

Considerações:

A declaração uma variável especial do tipo **atributo** é necessária quando se deseja efetuar uma reclassificação por atributos, na qual uma operação de transformação gera dados do modelo temático ou numérico, a partir de um mapa cadastral e do valor de atributos de uma categoria de objetos dada.

Numa declaração de **variáveis de atributos** deve ser indicado o nome da categoria de objetos a que o atributo pertence, bem como o nome que o identifica para o ambiente Spring.

Exemplos:

```
Atributo valores ("LOTES", "IPTU");
```

Instanciação

Recupere

Associa um *plano de informação* existente no projeto corrente com uma variável previamente declarada.

Sintaxe:

```
variável = Recupere (Nome = "nome_do_pi")
```

Considerações:

Os tipos tratados pelo operador **Recupere** são: **Imagem**, **numerico**, **Tematico**, **Objetos e Cadastral**. A variável e o plano de informação recuperado devem ser do mesmo tipo. Por exemplo, um plano de informação do tipo temático só pode ser associado a uma variável declarada anteriormente como temática.

Exemplos:

```
// Exemplos de declarações
Tematico Solos ("TIPOS_SOLOS");
Numerico topo ("ALTIMETRIA"),
Imagem banda3 ("LANDSAT"),
// Exemplos de instanciações das declarações acima
Solos = Recupere(Nome = "solos_vale_paraiba");
topo = Recupere(Nome = "altimetria_sjc");
banda3 = Recupere(Nome = "TM3_SJC");
```

Novo

Quando usado com variáveis do tipo **Tematico**, **Imagem** ou **Numerico**, cria um *plano de informação* do mesmo tipo no projeto corrente e associa este plano a variável do lado esquerdo do sinal de atribuição "=". Quando usado com variáveis do tipo **Tabela**, o operador **Novo** cria uma tabela no contexto do programa em execução. Esta tabela pode ser usada nos processamentos baseados em tabelas.

Sintaxe:

Para tipo imagem:

```
variável = Novo (Nome = "nome_do_pi", ResX =
valor,
ResY = valor, Nbits = 8);
```

Para tipo temático:

```
variável = Novo (Nome = "nome_do_pi",
ResX = valor , ResY = valor,
Escala = valor,
```

```
Repres = Vetor ou Raster);
```

Para tipo Numérico:

```
variável = Novo (Nome = "nome_do_pi",
                 ResX = valor , ResY = valor,
                 Escala = valor,
                 Min = valor, Max = valor,
                 Repres = Grade ou Amostra);
```

Para tipo Cadastral:

```
variável = Novo (Nome = "nome_do_pi",
                 ResX = valor , ResY = valor,
                 Escala = valor);
```

Para tipo Tabela de ponderação:

```
variável = Novo (CategoriaIni = "nome_da_categoria",
                 "geoclasse" : valor,
                 "geoclasse" : valor,
                 ...
                 Outros      : valor );
```

Para tipo Tabela de fatiamento:

```
variável = Novo (CategoriaFim = "nome_da_categoria",
                 [valorI, valorF] : "geoclasse",
                 [valorI, valorF] : "geoclasse",
                 ...
                 Outros      : valor );
```

Para tipo Tabela de reclassificação:

```
variável = Novo (CategoriaIni = "nome_da_categoria",
                 CategoriaFim = "nome_da_categoria",
                 "geoclasse" : "geoclasse",
                 "geoclasse" : "geoclasse",
                 ...
                 "geoclasse" : Outros );
```

Considerações:

Para variáveis do tipo Imagem, Numérico, Temático e Cadastral um plano com o nome fornecido é criado no projeto ativo. Este novo plano é usado para armazenar o resultado de uma operação.

Na instanciação de tabelas a palavra reservada “**Outros**” é opcional e define um valor a ser atribuído nos casos não cobertos pelas alternativas anteriores.

Para a instanciação de uma tabela de fatiamento o termo [valorI, valorF] significa a faixa de valores do campo numérico a ser atribuído a uma geoclasse. A faixa vai de valorI inclusive até valorF exclusive.

Parâmetros:

Nome - nome do plano de informação criado.

ResX - resolução horizontal.

ResY - resolução vertical.

Nbits – número de bits da imagem (8, 10, 16, ...).

Escala - escala do plano criado.

Min - valor mínimo permitido na grade criada.

Max - valor máximo permitido na grade criada.

Repres - representação do plano criado.

Vetor ou **Matriz** para Temático.

Grade ou **Amostra** para Numérico.

CategoriaIni - categoria de entrada.

CategoriaFim - categoria de saída.

Exemplos:

```
// Criar um PI Imagem:
ima = Novo (Nome = "ImagemResultante", ResX=30, ResY=30,
           Nbits = 8);

// Criar um PI Temático:
solo = Novo (Nome = "Solos", ResX=50, ResY=50,
           Escala = 100000);

// Criar um PI Numérico:
alti = Novo (Nome = "Altimetria", ResX=50, ResY=50,
           Escala = 1000, Min=0, Max=100);

// Criar um PI Cadastral:
cadast = Novo (Nome = "Plano_Urbano", ResX=50, ResY=50,
           Escala = 1000);

// Criar uma Tabela de Ponderação:
pond = Novo (CategoriaIni = "Tipos_de_solos",
           Le : 0.60,
           Li ; 0.20,
           Ls : 0.35,
           Aq : 0.10);

// Criar uma Tabela de fatiamento:
fatia = Novo (CategoriaFim = "Faixas_de_Declividade",
           [0.0, 5.0] : "baixa",
           [5.0, 15.0] : "media",
           [15.0, 45.0] : "alta");

Criar uma Tabela de reclassificação:
reclass = Novo (CategoriaIni = "Floresta",
           CategoriaFim = "Desmatamento",
           "Floresta_Densa" : "Floresta",
           "Floresta_Varzea" : "Floresta",
           "Rebrota" : "Desmatamento",
           "Area_Desmatada" : "Desmatamento");
```

Operação

Atribua

Gera um plano de informação temático efetuando o mapeamento de geoclasses com base em resultados de expressões lógicas (booleanas). Este mapeamento é executado por operadores lógicos pontuais (e, ou, não), que fazem a sobreposição (overlay) entre as classes dos PIs.

Expressões booleanas são combinadas a partir dos operadores '&&' (e lógico, intercessão), '||' (ou lógico, união) e '!' ou '~' (negação, complemento), ou ainda pela comparação entre pixels de imagens ou valores de grade através dos operadores '<', '>', '<=', '>=', '==' e '!='; ou da comparação entre classes de PIs temáticos através dos operadores '==' e '!='. Pode-se envolver até 40 PIs simultaneamente.

Sintaxe:

```
variável = Atribua(CategoriaFim = "nome_da_categoria")
    {
        "nome_da_geoclasse" : expressao_booleana),
        "nome_da_geoclasse" : expressao_booleana),

        "nome_da_geoclasse" : expressao_booleana),
        ...
        "nome_da_geoclasse" : Outros
    };
```

Alternativamente pode-se usar a sintaxe abaixo:

```
variável = Atribua
    {
        ...
    };
```

Considerações:

As geoclasses temáticas usadas no lado esquerdo do ":" devem ser geoclasses da categoria temática associada à variável temática previamente declarada. A palavra reservada "**Outros**" é opcional e define um valor a ser atribuído nos casos não cobertos pelas alternativas anteriores.

As expressões booleanas operam sobre representações matriciais dos dados envolvidos e são operações pontuais.

Exemplo:

```
temas = Atribua (CategoriaFim = "AptidaoAgricola")
    {
        "Bom" : (solos.tema == "Le" && decl >= 0.05),
        "Medio" : (solos.tema == "Aq" || decl >= 0.10),
        "Ruim" : (solos.tema == "Aq" &! decl >= 0.10),
        "fundo" : Outros ;
    }
```

Fatie

Gera um plano de informação do tipo **Temático** a partir de um dado numérico. As regras de processamento são baseadas em uma tabela de fatiamento criada através de um operador **Novo**.

Sintaxe:

```
pist = Fatie (pien, tabfat)
```

Considerações:

A variável `pist` deve ser um plano de informação do tipo **Temático**. O parâmetro `pien` deve ser um plano de informação do tipo numérico. O parâmetro `tabfat` deve ser uma tabela do tipo fatiamento.

Exemplo:

```
// Início de programa
// Declarações
Temático classes_decl ("Faixas_de_Declividade");
Numerico decli ("Declividade");
Tabela fatia(Fatiamento);
// Instanciações
fatia = Novo (CategoriaFim = "Faixas_de_Declividade",
             [0.0, 5.0] : "baixa",
             [5.0, 15.0] : "media",
             [15.0,45.0] : "alta");

decli = Recupere(Nome = "grade_declividade");
classes_decl = Novo(Nome="decli_fat", ResX = 40,
                   ResY = 40, Escala = 100000);

// Operação
classes_decl = Fatie(decli, fatia);
// Final de programa
```

Pondere

Gera um plano de informação do tipo **Numerico** a partir de um dado temático. As regras de processamento são baseadas em uma tabela de ponderação criada através de um comando usando o operador Novo.

Sintaxe:

```
pisn = Pondere (piet, tabpond)
```

Considerações:

A variável **pisn** deve ser um plano de informação do tipo **Numerico**. O parâmetro **piet** deve ser um plano de informação do tipo **Temático**. O parâmetro **tabpond** deve ser uma tabela do tipo **Ponderacao**.

Exemplo:

```
// Inicio de programa
// Declarações
Temático geom ("Geomorfologia");
Numerico fragil ("fragilidade");
Tabela tab_geo (ponderacao);

// Instanciações
fragil = Novo (Nome = "fragilidade_solo", ResX = 50,
              ResY = 50, Escala = 100000);

tab_geo = Novo (CategoriaIni = "Geomorfologia",
              "SEpt" : 1.2,
              "EspP" : 1.1,
              "Estb" : 1.3,
              "dk" : 2.4);

geom = recupere(Nome = "Geomorf_RADAM");

// Operação
fragil = 0.5 * Pondere(geom, tab_geo);

// Final de programa
```

Reclassifique

Remapea os temas de um plano de informação temático para um novo plano também temático. Este remapeamento é feito segundo regras descritas por uma tabela de reclassificação. Esta é uma operação de generalização, onde temas diferentes de entrada são mapeados para um de saída.

Sintaxe:

```
pisn = Reclassifique (piet ,tabela_reclass)
```

Considerações:

A variável `pisn` deve ser um plano de informação do tipo **Temático**. O parâmetro `piet` deve ser um plano de informação do tipo **Temático**. O parâmetro `tabela_reclass` deve ser uma tabela do tipo **Reclassificacao**, instanciada previamente pelo operador **Novo**.

Exemplos:

```
{
// inicio do programa
// Declarações
Tematico cobertura ("Floresta");
Tematico desmat ("Desmatamento");
Tabela tab_reclass(Reclassificacao);
// Instanciações
tab_reclass = Novo (CategoriaIni = "Floresta",
                    CategoriaFim = "Desmatamento",
                    "Floresta_Densa" : "Floresta",
                    "Floresta_Varzea" : "Floresta",
                    "Rebrota" : "Desmatamento",
                    "Area_Desmatada" : "Desmatamento");

cobertura = Recupere(Nome = "tipos_cobertura");
desmat = Novo (Nome = "desmatamento",
              ResX = 30, ResY = 30, Escala = 100000);
// Operação
desmat = Reclassifique (cobertura,tab_reclass);
// Final de programa
}
```

? :

Implementa uma expressão condicional a partir do resultado de uma expressão booleana avaliada antes do sinal ?. Se o resultado da expressão booleana for verdadeira atribui-se para a variável de saída o resultado da expressão antes do “:”, caso contrario atribui-se o resultado da expressão que vem depois do “:”.

Sintaxe:

```
variável = expressao_booleana ? expressao1 : expressao2
```

Considerações:

O tipo da variável de saída e o da expressão1 e expressao2 devem ser compatível.

Exemplos: Recorte de imagem por temas

```
{
// início do programa
// Declarações
Imagem tm345, tm5M ("Imagens");
Tematico uso ("UsoSolo");
// Instanciações
  uso  = Recupere (Nome = "UsodoSolo");
  tm345 = Recupere (Nome = "TM345");
  tm5M = Novo (Nome = "teste", ResX=30, ResY=30);
// Operação condicional
tm5M = uso.Class=="cidade" ? tm345 : 240;
// Final de programa
}
```

ReclAtrib

Cria um PI temático em função dos atributos definidos para objetos representados em um PI cadastral. O operador executa a transformação sobre a representação vetorial.

Sintaxe:

```
variável = ReclAtrib (objeto, atributo, tab_fat)
SobreMapa pi_cadastral;
```

Exemplos:

```
{
Atributo pop ("Comunidades_O", "populacao");
Objeto com ("Comunidades_O");
Cadastral cad ("Mapa_de_Comunidades");
Tematico tem ("Faixas_tematicas");
Tabela fat (Fatiamento);

cad = Recupere (Nome = "mapa_de_comunidades");
tem = Novo (Nome ="populacao", ResX =100, ResY =100,
Escala =100000, Repres =Raster);

fat = Novo ( CategoriaFim= "Populacao",
[1000, 2000] : "Baixa",
[2000, 5000] : "Media",
[5000, 10000] : "Alta" );
tem = ReclAtrib (com, pop, fat) OnMap cad;
}
```

MediaZonal(*pie*,*lista_de_zonas*)

Calcula o valor médio dentro de cada zona (região) de um plano de informação referenciado pelo argumento *pie*. As zonas são definidas, uma a uma pela *lista_de_zonas*.

Sintaxe:

Resultado em grade Numérica
`pis = MediaZonal(pie, lista_de_zonas)`
 Resultado em um único valor numérico
`vr = MediaZonal(pie, zona)`
 Resultado numérico em Tabela de Atributos de Objetos
`Objeto."atr" = MediaZonal(pie, objeto OnMap pic)`

Considerações:

pie - Plano de informação de entrada que define os valores dentro de cada região geográfica avaliada. O plano de informação *pie* deve ser do tipo **Imagem** ou do tipo **Numerico**.

lista_de_zonas - Identifica as zonas sobre as quais serão calculados os valores médios. Todas as zonas presentes na *lista_de_zonas* devem estar localizadas sobre a região geográfica do *pie*. Uma zona ou região pode ser definida por uma expressão booleana ou por um objeto. Uma *lista_de_zonas* pode ser definida por uma lista de expressões booleanas, ou por todas as classe temáticas presente em um plano de informação temático.

pis plano de informação de saída que deve ser do tipo **Numerico**.

vr - variável do tipo float que recebe o valor médio resultante de uma única região.

Objeto."atr" - Atributo de um objeto representado no plano de informação *pic*.

pic - Plano de informação do tipo cadastral representando um conjunto de regiões.

No caso do resultado em grade numérica, todas as posições na grade numérica dentro de uma zona receberão o valor médio calculado sobre aquela zona.

Exemplos:

```
{
// Início de programa
// Declarações
Tematico zonas      ("Solos_vale_paraiba");
Numerico ph        ("ph_solos_vale_paraiba");
Numerico ph_medio  ("grade_media_zonais_10x10");
Float    ph_terra_roxa;
```

```

// Instanciações
// Dados de entrada
ph = Recupere(Nome = "ph_solos_sjc");
zonas = Recupere(Nome = "solos_sjc");
ph_medio = Novo(Nome = "ph_medio_solos_sjc", ResX= 10,
                ResY= 10, Escala = 100000, Min = 0, Max =
20);

// Operação
// Resultado em campo Numérico para algumas regiões
especificas.
ph_medio = MediaZonal(ph,zonas.Classe == "terra_roxa",
                      zonas.Classe == "latosolo_vermelho",
                      zonas.Classe == "solos_hidromorficos");

// Resultado em campo Numérico para as classe temáticas
// presentes no plano de informação temático "solos_sjc".
// Neste caso cada tipo de solo define uma região.

ph_medio = MediaZonal(ph,zonas.Classe * );

// Resultado em um único valor numérico
ph_terra_roxa = MediaZonal(ph,zonas.Classe ==
"terra_roxa");

}

// Programa de media zonal que atualiza tabela de atributos
{
// Inicio de programa
// Declarações
Objeto          regioes ("municipios");
Cadastral       mapamunicipios ("mapacad");
Digital         altimetria("altime");

mapamunicipios = Recupere (Nome = "municipios");
altimetria     = Recupere (Nome = "alt");

// Resultado numérico em Tabela de Atributos de Objetos
regioes."altitudeM" = MediaZonal (altimetria,regioes OnMap
mapamunicipios);
}

```

MaxZonal(*pie*, *lista_de_zonas*)

Identifica o valor máximo dentro de cada zona (região) de um plano de informação referenciado pelo argumento *pie*. As zonas são definidas, uma a uma pela *lista_de_zonas*.

Sintaxe:

Resultado em grade Numérica.

```
pis = MaxZonal(pie, lista_de_zonas)
```

Resultado em um único valor numérico

```
vr = MaxZonal(pie, zona)
```

Resultado numérico em Tabela de Atributos de Objetos

```
Objeto."atr" = MaxZonal(pie, objeto OnMap pic)
```

Considerações:

pie - Plano de informação de entrada que define os valores dentro de cada região geográfica avaliada. O plano de informação *pie* deve ser do tipo **Imagem** ou do tipo **Numerico**.

lista_de_zonas - Identifica as zonas sobre as quais serão calculados os valores máximos. Todas as zonas presentes na *lista_de_zonas* devem estar localizadas sobre a região geográfica do *pie*. Uma zona ou região pode ser definida por uma expressão booleana ou por um objeto. Uma *lista_de_zonas* pode ser definida por uma lista de expressões booleanas, ou por todas as classe temáticas presente em um plano de informação temático.

Pis - Plano de informação de saída que deve ser do tipo **Numerico**.

vr - Variável do tipo float que recebe o valor máximo dentro de uma única região.

Objeto."atr" - Atributo de um objeto representado no plano de informação *pic*.

pic - Plano de informação do tipo cadastral representando um conjunto de regiões.

No caso do resultado em grade numérica, todas as posições na grade numérica dentro de uma zona receberão o valor máximo calculado sobre aquela zona.

Exemplos:

```
{
// Inicio do Programa
// Variáveis usadas como restrição
```

```

Tematico    zon("tipos_do_Solo");
Imagem     tm5("imagensTM");
Numerico   ph("ph_dos_solos");
Numerico   maximo_ph_zonal("grade10x10");
Float      reflec_max_terra_roxa;

ph           = Recupere(Nome = "ph_solo_sjc");
zon          = Recupere(Nome = "solos_sjc");

maximo_ph_zonal = Novo(Nome = "ph_maximo_zonal_sjc",
ResX = 10, ResY = 10, Escala = 100000, Min = 0 Max =
200);

// Resultado em campo Numérico
maximo_ph_zonal = MaxZonal(hp,zon.Classe ==
"terra_roxa",                               zon.Classe ==
"latosolo_vermelho",
                               zon.Classe == "solos_hidromorficos
);

// Resultado em campo Numérico para as classe temáticas
// presentes no plano de informação temático
"solos_sjc". // Neste caso cada tipo de solo
define uma região.

ph_medio = MaxZonal (ph,zon.Classe * );

// Resultado em um único valor numérico
reflec_max_terra_roxa = MaxZonal(tm5,zon.Classe ==
"terra_roxa");

// Final de Programa
}

// Programa de maximo zonal que atualiza tabela de atributos
{
// Inicio de programa
// Declarações
Objeto           regioes ("municipios");
Cadastral        mapamunicipios ("mapacad");
Digital          altimetria("altime");

mapamunicipios = Recupere (Nome = "municipios");
altimetria     = Recupere (Nome = "alt");

// Resultado numérico em Tabela de Atributos de Objetos
regioes."altitudeMax" = MaxZonal (altimetria,regioes OnMap
mapamunicipios);
}

```

Espacialize (objeto."<atributo>" **OnMap** <cadastral>)

Permite que os valores referentes a um certo atributo de objeto sejam mapeados em um geocampo do modelo Temático ou Digital. No primeiro caso atributos textuais são associados a classes temáticas, no segundo, os valores numéricos do atributo são representados em uma grade. Em ambos os casos regiões de isovalores são definidas à partir das entidades previamente associadas aos objeto em um mapa cadastral.

Sintaxe:

Resultado tipo **Temático** ou **Digital** (geo-campo).

```
tem = Espacialize(obj."atrtxt" OnMap cad)
```

```
dig = Espacialize(obj."atrnum" OnMap cad)
```

Considerações:

tem é o nome de uma variável associada ao plano de informação de saída, declarada previamente no programa. A variável *tem*, deve estar associada a uma certa categoria temática, cujas classes, definidas, também previamente, no esquema conceitual devem ter nomes idênticos aos valores textuais possíveis do atributo considerado para espacialização.

dig é o nome de uma variável associada ao plano de informação de saída, declarado previamente no programa sob uma certa categoria digital.

obj é o nome de uma variável associada a geo-objetos de uma categoria do modelo objetos. Estes geo-objetos, associados previamente a entidades de um mapa cadastral, devem ter seus campos de atributos preenchidos com valores numéricos ou nomes de classes temáticas..

atrtxt é o nome de um atributo do tipo texto que se deseja espacializar; cada valor possível desse atributo deve ser idêntico ao nome de uma classe da categoria temática associada ao plano representado pela variável *tem*.

atrnum é o nome de um atributo do tipo inteiro ou real que se deseja espacializar

cad é uma variável declarada no programa como sendo de uma certa categoria do modelo Cadastral, associada a um plano de informação cujas entidades representam os objetos envolvidos na espacialização.

Exemplo:

```

{
    // Inicio de programa
    // Declarando os nomes das variáveis usadas
    Objeto      limites ("talhoes");
    Cadastral   fazendas ("mapadefazendas");
    Tematico    tem ("solos");
    Numerico    ph_solos("grade_ph")

    // Instanciações
    //Dados de entrada
    fazendas = Recupere (Nome = "mt");
    tem = Novo (Nome="mapa_de_solos", ResX =30, ResY =30,
    Escala =10000, Repres =Vetor);

    Resultado em geo-campo Temático
    tem = Atualize (limites."SOLO" OnMap fazendas);

    Resultado em campo Numérico
    ph_solos = Atualize (limites."ph" OnMap fazendas);
}

```

Atualize (<PI referencia>, objeto **OnMap**
<cadastral>, <operador zonal>)

Realiza operações zonais entre GEO-CAMPOS e GEO-OBJETOS, onde os atributos do GEO-OBJETOS são atualizados a partir do resultado de operações booleanas sobre as classes dos GEO-CAMPOS temáticos de entrada.

Atualiza um atributo dos objetos, previamente associados a entidades em um mapa cadastral, a partir de valores obtidos pela aplicação de operadores zonais sobre um plano de informação de referencia.

Sintaxe:

```
obj."atr" = Atualize (pie, obj OnMap cad, opZ)
```

Considerações:

obj é o nome de uma variável associada a geo-objetos de uma categoria do modelo objetos. Estes geo-objetos, associados previamente a entidades de um mapa cadastral, devem ter seus campos de atributos preenchidos com valores numéricos ou nomes de classes temáticas..

pie é o nome de uma variável associada ao plano de informação de entrada, declarada previamente no programa e associada a uma certa categoria temática ou digital.

A operação zonal utilizada deve estar coerente com o modelo do dado representado pelo PI de referência, isto é: alguns operadores não fazem sentido diante do tipo de dado de referência, por exemplo, uma operação de *MediaZonal* não é viável sobre um plano de modelo temático.

"*atr*" é o nome do atributo que se deseja atualizar referente a cada objeto da categoria de objetos associada à variável *obj* que estejam associados a entidades de um mapa cadastral.

cad é uma variável declarada no programa como sendo de uma certa categoria do modelo Cadastral.

Exemplo:

```
// Programa Atualiza tabela com operacao de media zonal
{
Objeto          regioes ("municipios");
Cadastral      mapamunicipios ("mapacad");
Digital        altimetria("altime");

mapamunicipios = Recuperar (Nome = "municipios");
altimetria = Recuperar (Nome = "alt");
regioes."Altitudemedia" = Atualize ( altimetria, regioes OnMap
mapamunicipios, MedZ) ;
}

// Programa Atualiza tabela com operacao de maximo zonal
{
Objeto          regioes ("municipios");
Cadastral      mapamunicipios ("mapacad");
Digital        altimetria("altime");

mapamunicipios = Recuperar (Nome = "municipios");
altimetria = Recuperar (Nome = "alt");
regioes."Altitudemaxima" = Atualize ( altimetria, regioes
OnMap
mapamunicipios, MaxZ);
}
```

MinZonal(*pie, lista_de_zonas*)

Identifica o valor mínimo dentro de cada zona (região) de um plano de informação referenciado pelo argumento *pie*. As zonas são identificadas, uma a uma pela *lista_de_zonas*.

Sintaxe:

Resultado em grade Numérica.

```
pis = MinZonal (pie, lista_de_zonas)
```

Resultado em um único valor numérico

```
vr = MinZonal (pie, zona)
```

Resultado numérico em Tabela de Atributos de Objetos
Objeto."atr" = **MinZonal** (pie, objeto OnMap *pic*)

VariedadeZonal(*pie, lista_de_zonas*)

Calcula a diversidade de valores ou de temas dentro de cada zona (região) de um plano de informação referenciado pelo argumento *pie*. As zonas são identificadas, uma a uma pela *lista_de_zonas*.

Sintaxe:

Resultado tipo **Numerico** (geo-campo).

```
pis = VariedadeZonal(pie, lista_de_zonas)
```

Resultado em um único valor numérico

```
vr = VariedadeZonal(pie, zona)
```

FaixaZonal(*pie, lista_de_zonas*)

Sintaxe:

Resultado tipo **Numerico** (geo-campo).

```
pis = FaixaZonal(pie, lista_de_zonas)
```

Resultado em um único valor numérico

```
vr = FaixaZonal(pie, zona)
```