Operadores – TerraMA^2

Tipos de Análise

1- Utilitários

2 - Operadores para Análise baseada em Objetos Monitorados

3 - Operadores entre dados matriciais

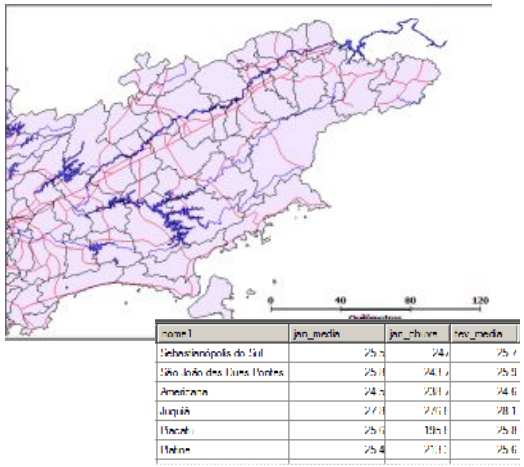
4 – Operadores de PCD

INPE

DPI
Divisão de Processamento de Imagens

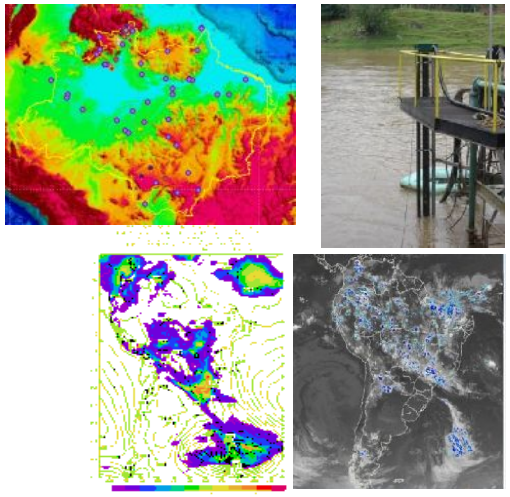
Tipos de Análises

- Análise baseada em Objetos Monitorados
  - **ENTRADA**
    - Requer um mapa vetorial previamente disponível como dado estático;
    - Requer dados dinâmicos cadastrados;
    - Requer um modelo de análise escrito em Python.
  - **SAÍDA** : tabela com os resultados da análise



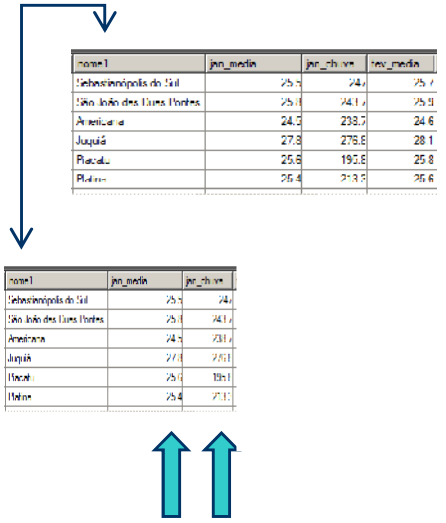
Mapa com áreas a serem monitoradas

+



Dados Ambientais dinâmicos

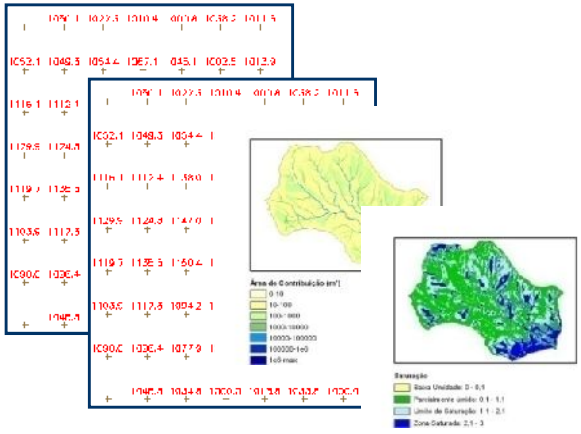
=



Novas colunas com resultados

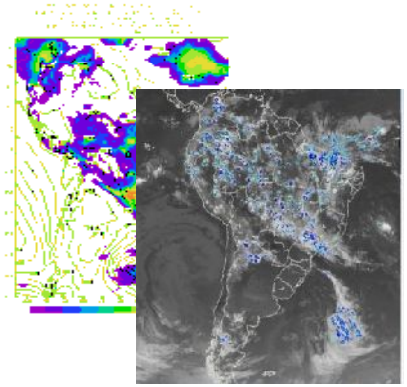
Tipos de Análises

- Análise baseada em Grades
  - **ENTRADA**
    - Mapas estáticos matriciais disponíveis
    - Requer dados dinâmicos matriciais cadastrados **(pelo menos um)**
    - Requer um modelo de análise escrito em Python
  - **SAÍDA** : Dado dinâmico matricial.



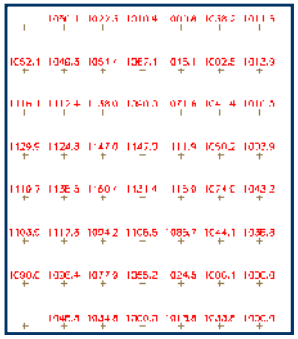
Dados matriciais estáticos

+



Dados dinâmico matricial

=



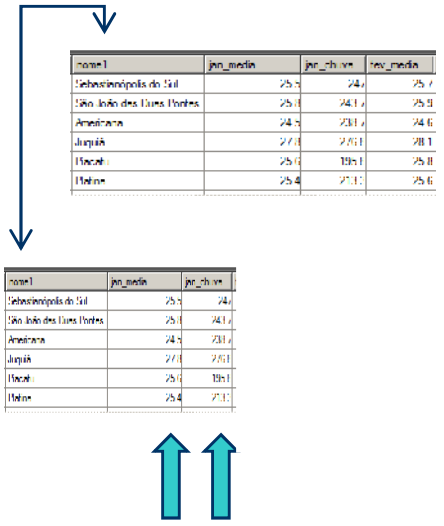
Dados dinâmico matricial

Tipos de Análises

- Análise com PCD
  - **ENTRADA**
    - Requer uma fonte de dados do tipo PCD
    - Requer um modelo de análise escrito em Python
  - **SAÍDA** : tabela com os resultados da análise

PCD´s no campo

=



Novas colunas com resultados

1- Utilitários para as análises

Os seguintes utilitários estão disponíveis

Unidade de distância : unidade utilizada pelo operador “buffer”

Unidade de tempo : unidade utilizada pelo operadores históricos

Buffer : define distâncias ou faixas de distâncias a partir de objetos monitorados (dados estáticos representados por ponto, linha ou polígonos)

Adiciona valor : utilizado para inserir valores aos resultados das análises

Gerais : demais funções

1 - Utilitários - Unidade de distância

Para operadores que utilizam unidades de distância tem-se as seguintes opções:

- “cm”: Centímetros
- “m” : Metros
- “km”: Kilômetros

Exemplo de Uso no operador Buffer

- buffer = Buffer(BufferType.Out_union, 50, "**cm**")
- buffer = Buffer(BufferType.Level, 400, “**m**“, 200, “**m**”)
- buffer = Buffer(BufferType.In_out, 200, “**km**“, 200, “**km**”)

1 - Utilitários - Unidade de tempo

Para filtro de data temos as seguintes unidades de tempo:

- sec: Segundo
- min: Minuto
- h: Hora
- d: Dia
- w: Semana

Exemplo de Uso

- x1 = occurrence.zonal.count("ocorrencias", "**120sec**", buf1, "UF = 'AM'")
- x2 = occurrence.zonal.interval.max("focos", "**30min**", "10min", "Intens", buf1)
- x3 = dcp.zonal.history.min("Serra do Mar", "Pluvio", "**48h**", ids)
- x4 = grid.zonal.forecast.median("ETA15km", "**3d**", buffer_mun)
- x5 = grid.zonal.history.accum.min("hidro", "**1w**", 0, buffer_reg)

1 - Utilitários – Buffer

Criação de buffer a partir de geometrias vetoriais.

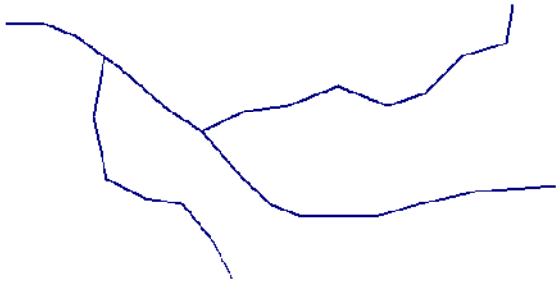
classe : Buffer()

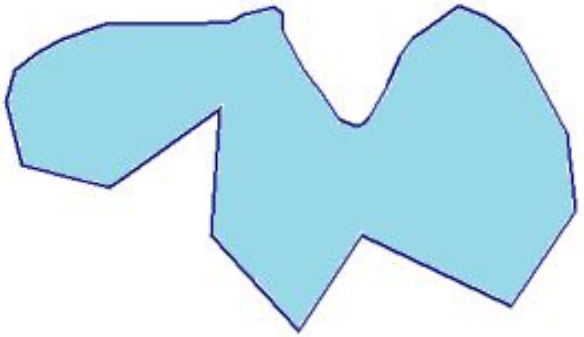
Tipos

BufferType.None : Sem buffer
BufferType.In : Somente a geometria do buffer interno.
BufferType.Out : Somente a geometria do buffer externo.
BufferType.In_out : A união da geometria do buffer externo com a geometria do buffer interno.
BufferType.Out_union : Interior da geometria mais a geometria do buffer externo
BufferType.In_diff : Interior da geometria menos a geometria do buffer, este buffer deve ser interno.
BufferType.Level: A diferença entre a geometria do buffer 1 e a geometria do buffer 2.

1 - Utilitários – Buffer

BufferType.None( ) : Sem buffer





Ponto

Linha

Área do Polígono

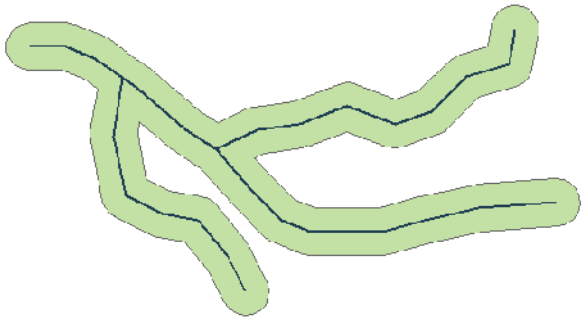
Exemplo: b1 = Buffer(BufferType.None) ou
b1 = Buffer( )

OBS: Considera somente a localização do ponto, linha e área do polígono que fizer interseção com o dado dinâmico

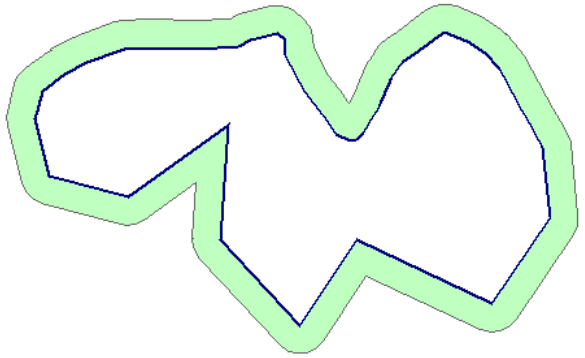
1 - Utilitários – Buffer

BufferType.Out : Somente a área do buffer externo.

Buffer em volta do ponto



Buffer em volta da linha



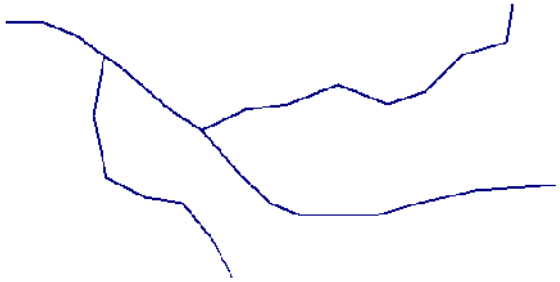
Buffer externo ao polígono

Exemplo: b1 = Buffer(BufferType.Out, 200, “m")

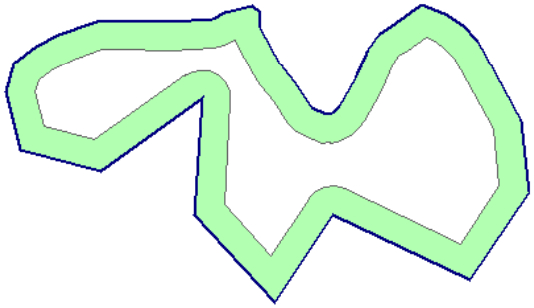
1 - Utilitários – Buffer

BufferType.In : Somente a área do buffer interno.

Ponto



Linha



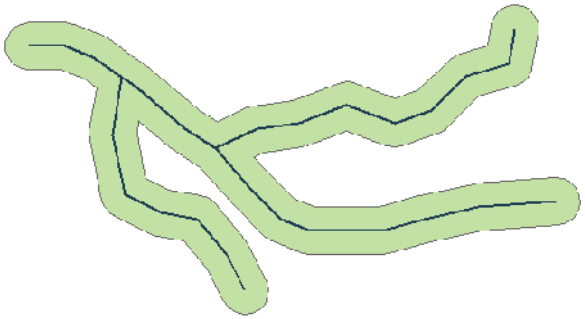
Buffer interno ao polígono

Exemplo: b1 = Buffer(BufferType.In, 200, “m")

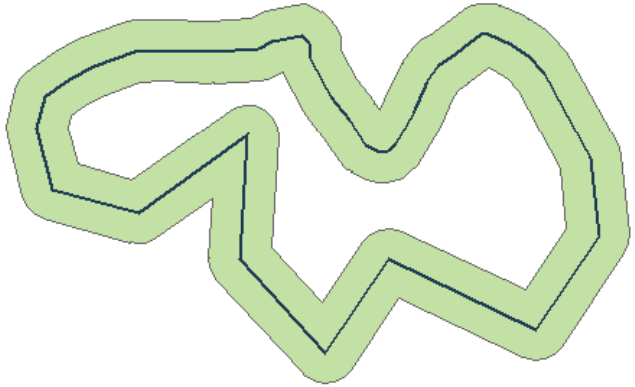
1 - Utilitários – Buffer

Buffer.Type.In_out : A união da geometria do buffer externo com a geometria do buffer interno.

Buffer em volta do ponto



Buffer em volta da linha



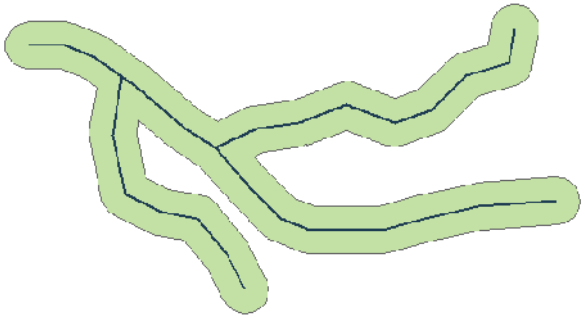
Buffer interno e externo ao polígono

Exemplo: b1 = Buffer(BufferType.In_out, 200, “m“, 200, “m”)

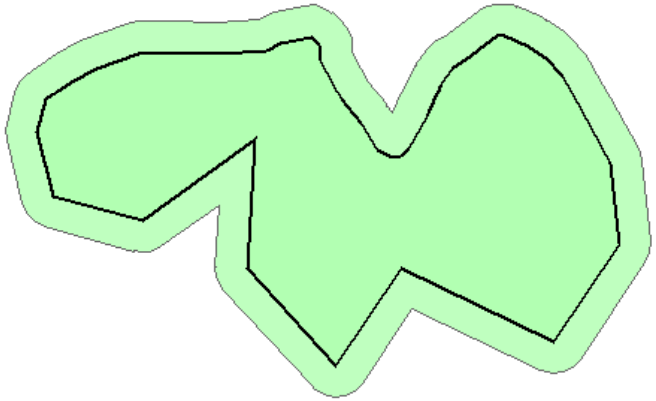
1 - Utilitários – Buffer

BufferType.Out_union : Interior da geometria mais a geometria do buffer externo

Buffer em volta do ponto



Buffer em volta da linha



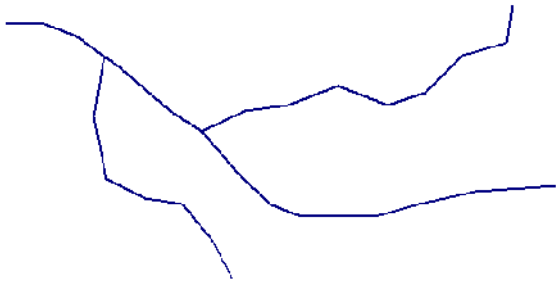
Buffer externo mais área do polígono

Exemplo: b1 = Buffer(BufferType.Out_union, 200, “m“)

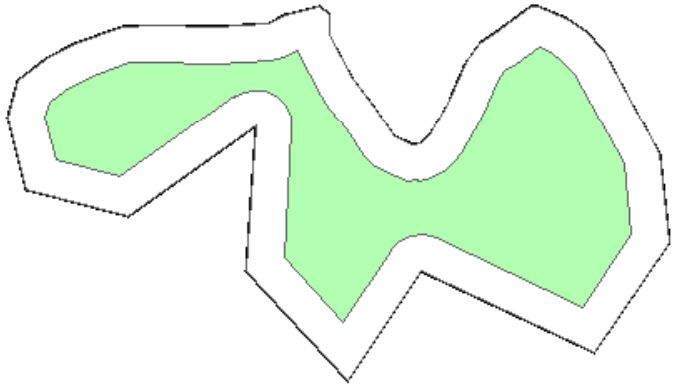
1 - Utilitários – Buffer

BufferType.In_diff : Interior da geometria menos a geometria do buffer interno.

Ponto



Linha

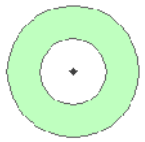


Área do polígono menos buffer interno

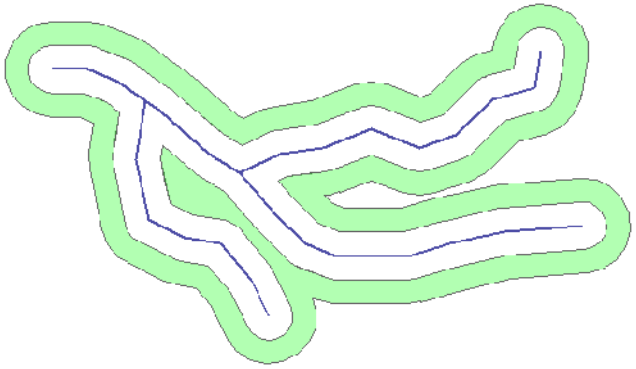
Exemplo: b1 = Buffer(BufferType.In_diff, 200, “m")

1 - Utilitários – Buffer

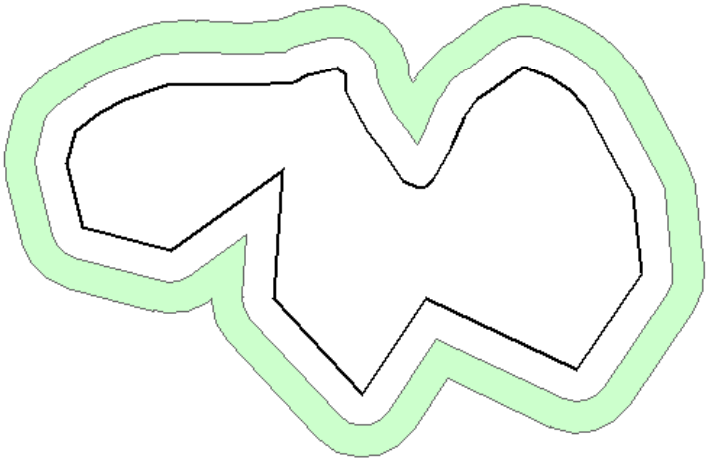
BufferType.Level: A diferença entre a geometria do buffer 1 e a geometria do buffer 2 (distancia de 1 maior que 2)



Buffer1 externo do ponto menos Buffer2 externo



Buffer1 externo da linha menos Buffer2 externo



Buffer1 externo do polígono menos buffer2 externo

Exemplo: b1 = Buffer(BufferType.Level, 400, “m“, 200, “m”)

1 - Utilitários - Adiciona valor

Método utilizado para adicionar o valor ao resultado análise

Assinatura

- add_value("attributeName", value)

Parâmetros

- *attributeName*: String com o nome do atributo que vai armazenar o valor
- *value*: Valor a ser armazenado, deve ser do tipo numérico. (Ex. Integer, Float, Double).

Exemplo de uso

- moBuffer = Buffer(BufferType.object_plus_buffer, 2., "km")
- x = dcp.min("Serra do Mar", moBuffer, "1d", "Pluvio")
- add_value("Minimo", x)

1 - Utilitários - Gerais

Estatística : Funções estatísticas para agregação

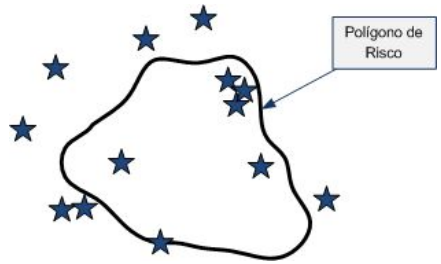
- Statistic.min : valor mínimo de uma lista de valores
- Statistic.max : valor máximo de uma lista de valores
- Statistic.mean : valor médio de uma lista de valores
- Statistic.sum : soma de uma lista de valores
- Statistic.mean : média de uma lista de valores
- Statistic.standard_deviation : desvio padrão de uma lista de valores

2- Operadores para Análise baseada em Objetos Monitorados

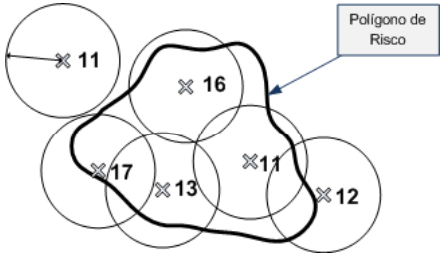
Utiliza operadores zonais com a geometria dos dados estáticos vetoriais de **ponto, linha ou polígonos** para realização de cálculos estatísticos.

Tipos de operadores:

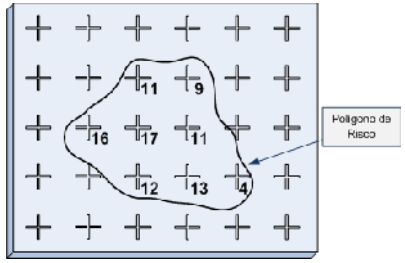
I. Operadores zonais de ocorrência



II. Operadores zonais de PCDs



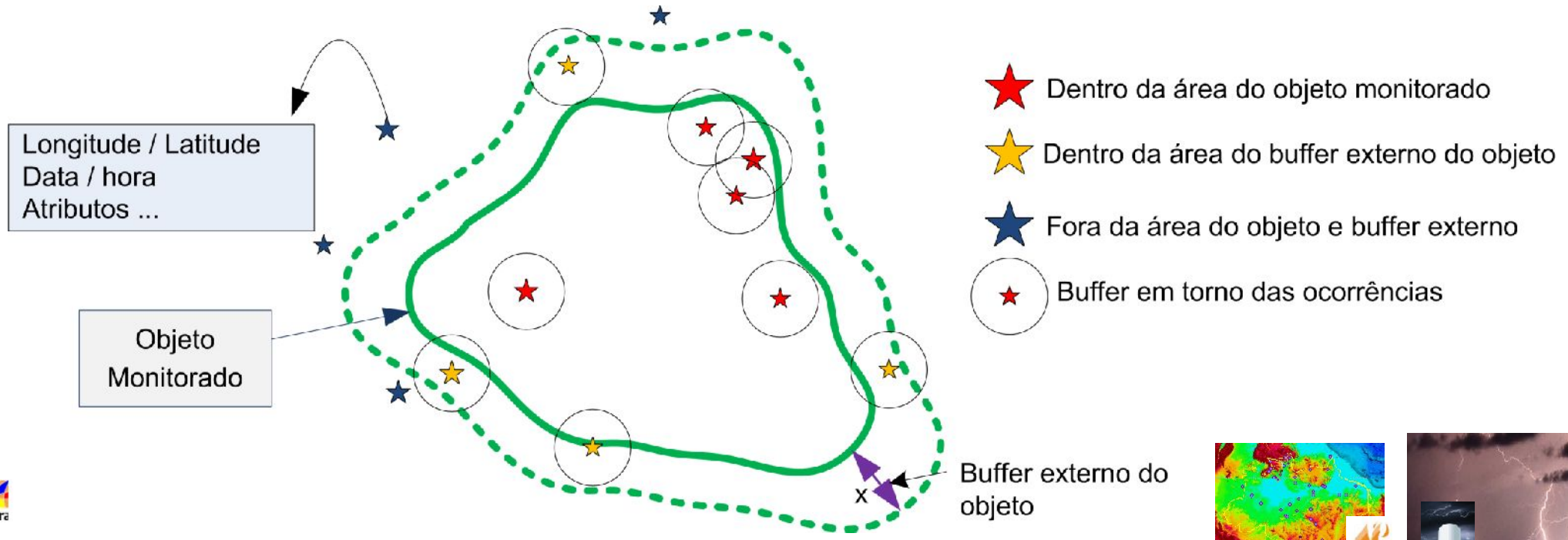
III. Operadores zonais de grades



2.I - Operadores zonais de ocorrência

Operadores utilizados para obter estatísticas sobre as fontes de dados do tipo de ocorrências.

Consideram a localização dos pontos de ocorrências e seus atributos que interceptam todas as geometrias (pontos, linhas ou polígonos) de um mapa (objetos monitorados) ou a área de influência (buffer) dessas geometrias num intervalo de tempo passado.



2.I - Operadores zonais de ocorrência

Tipos de operadores zonais de ocorrência estão divididos em 3 grupos

2.I.A – Zonal

Grupo de operadores que consideram as ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

2.I.B – Zonal por Intervalo

Grupo de operadores que consideram as ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre dois valores de tempo informado no passado.

2.I.C – Zonal por Agregação

Grupo de operadores que consideram as ocorrências agrupadas em torno do mesmo ponto e que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

2.I.A - Operadores zonais de ocorrência - Zonal

occurrence.zonal.<operator>(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <buffer>, “<restriction>”) onde:

- operator : **count, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_occurrence : String com o nome da série de dados de ocorrências
- time: String com o intervalo de tempo, a partir da hora atual, para filtrar as ocorrências. Ver utilitário Unidades de tempo
- attribute : String com o nome do atributo da ocorrência que deve ser utilizado para recuperar os valores, o atributo deve ser do tipo numérico (Ex. Integer, Float, Double, Long). Não usar para operador “count”
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Parâmetro obrigatório somente se o parâmetro seguinte (restriction) for utilizado. Ver utilitário Buffer.
- restriction : [Opcional] String com a restrição SQL a ser aplicada. Não utilizar se não houver restrição.

2.I.A - Operadores zonais de ocorrência
Zonal: **Contagem**

Retorna a quantidade de ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.count(“<dynamic_data_occurrence>”, “<time>”, <buffer>, “<restriction>”)

Exemplo: buf1 = Buffer(BufferType.Level, 400, “m“, 200, “m”)
x = occurrence.zonal.count("ocorrencias", "1d", buf1, "UF = ’AM”")

2.I.A - Operadores zonais de ocorrência
Zonal : **Mínimo**

Retorna o menor valor do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.min(“<dynamic_data_occurrence >”, “<time>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.min("ocorrencias", "1d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.A - Operadores zonais de ocorrência
Zonal : **Máximo**

Retorna o maior valor do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.max(“<dynamic_data_occurrence >”, “<time>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.max("ocorrencias", "1d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.A - Operadores zonais de ocorrência
Zonal : **Média**

Retorna a média dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.mean(“<dynamic_data_occurrence >”, “<time>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.mean("ocorrencias", "1d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.A - Operadores zonais de ocorrência
Zonal : **Soma**

Retorna a soma dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.sum(“<dynamic_data_occurrence >”, “<time>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.sum("ocorrencias", "1d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.A - Operadores zonais de ocorrência
Zonal : **Mediana**

Retorna a mediana dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.median(“<dynamic_data_occurrence >”, “<time>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.median("ocorrencias", "1d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.A - Operadores zonais de ocorrência
Zonal : **Desvio Padrão**

Retorna o desvio padrão dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.standard_deviation(“<dynamic_data_occurrence >”, “<time>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.standard_deviation("ocorrencias", "1d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.A - Operadores zonais de ocorrência
Zonal : **Variância**

Retorna a variância dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

occurrence.zonal.variance(“<dynamic_data_occurrence >”, “<time>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.variance("ocorrencias", "1d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.B - Operadores zonais de ocorrência – Zonal por Intervalo

occurrence.zonal.interval<operator>(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, “<attribute>”, <buffer>, “<restriction>”)
onde:

Operator : **count, min, max, mean, sum, median, standard_deviation, variance**

- dynamic_data_occurrence : String com o nome da série de dados de ocorrências
- time_begin: String inicial (mais antigo) do intervalo de tempo para filtrar as ocorrências.
- time_end: String final (mais recente) do intervalo de tempo para filtrar as ocorrências.
- attribute : String com o nome do atributo da ocorrência que deve ser utilizado para recuperar os valores, o atributo deve ser do tipo numérico (Ex. Integer, Float, Double, Long). Não usar para operador “count”
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Parâmetro obrigatório somente se o parâmetro seguinte (restriction) for utilizado. Ver utilitário Buffer.
- restriction : [Opcional] String com a restrição SQL a ser aplicada. Não utilizar se não houver restrição.

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo: **Contagem**

Retorna a quantidade de ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval.count(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, <buffer>, “<restriction>”)

Exemplo: buf1 = Buffer()
x = occurrence.zonal.count(“raios", “2d", "1d", buf1, "UF = ’AM’")

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo : **Mínimo**

Retorna o menor valor do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval,min(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.interval.min("ocorrencias", “4h", "1h", "Intensidade", buf1, "UF = ’AM’")
```

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo : **Máximo**

Retorna o maior valor do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval.max(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.interval.max(“focos", “30min", "10min", "Intensidade", buf1, "UF = ’AM’")
```

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo : **Média**

Retorna a média dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval.mean(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.interval.mean(“raios", "10d", “5d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo : **Soma**

Retorna a soma dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval.sum(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.interval.sum("ocorrencias", “2h", "1.5d", "Intensidade", buf1, "UF = ’AM’")
```

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo : **Mediana**

Retorna a mediana dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval.median(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.interval.median("ocorrencias", “3w", "1w" "Intensidade", buf1, "UF = ’AM’")
```

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo : **Desvio Padrão**

Retorna o desvio padrão dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval.standard_deviation(
“<dynamic_data_occurrence >”,
“<time_begin>”, “<time_end>”, “<attribute>”,
<buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.interval.standard_deviation(“foco", “20h", "15h", "Intensidade", buf1, "UF = ’AM’")
```

TerraMA²

2.I.B - Operadores zonais de ocorrência
Zonal por intervalo : **Variância**

Retorna a variância dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

occurrence.zonal.interval.variance(“<dynamic_data_occurrence>”, “<time_begin>”, “<time_end>”, “<attribute>”, <buffer>,“<restriction>”)

Exemplo:

```
buf1 = Buffer()
x = occurrence.zonal.interval.variance(“raios", “360sec", “48sec", "Intensidade", buf1, “type = 1")
```

2.I.C - Operadores zonais de ocorrência – Zonal por Agregação

occurrence.zonal.aggregation<operator>(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <aggregation_statistic>, <aggregation_buffer>, <buffer>, “<restriction>”) onde:

operator : **count, min, max, mean, sum, median, standard_deviation, variance**

- **dynamic_data_occurrence** : String com o nome da série de dados de ocorrências
- **time**: String com o intervalo de tempo, a partir da hora atual, para filtrar as ocorrências. Ver utilitário Unidades de tempo.
- **attribute** : String com o nome do atributo da ocorrência que deve ser utilizado para recuperar os valores, o atributo deve ser do tipo numérico (Ex. Integer, Float, Double, Long). Não usar para operador “count”
- **aggregationBuffer** : Objeto Buffer para agregação dos pontos. Ver utilitário Buffer
- **aggregationStatistic** : Tipo de operador estatístico a ser utilizado para selecionar o valor do atributo para as ocorrências agregadas. Não usar para operador “count”
- **buffer** : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Parâmetro obrigatório somente se o parâmetro seguinte (restriction) for utilizado. Ver utilitário Buffer.
- **restriction** : [Opcional] String com a restrição SQL a ser aplicada. Não utilizar se não houver restrição.

2.I.C - Operadores zonais de ocorrência
Zonal por agregação: **Contagem**

Retorna a quantidade de ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto serão contadas um única vez.

occurrence.zonal.aggregation.count(“<dynamic_data_occurrence>”, “<time>”, <aggregation_buffer>, <buffer>, “<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.count(“focos", "1d", bufaggreg, buf1, "UF = ’AM’")
```

2.I.C - Operadores zonais de ocorrência
Zonal por agregação : **Mínimo**

Retorna o menor valor do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto terão um único valor obtido por análise estatística (utilitário estatístico).

occurrence.zonal.aggregation.min(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <aggregation_statistic>, <aggregation_buffer>, <buffer>,“<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.min(“focos", "1d", “risco”, Statistic.min, bufaggreg, buf1, "UF = ’AM’")
```

2.I.C - Operadores zonais de ocorrência
Zonal por agregação : **Máximo**

Retorna o maior valor do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto terão um único valor obtido por análise estatística (utilitário estatístico).

occurrence.zonal.aggregation.max(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <aggregation_statistic>, <aggregation_buffer>, <buffer>,“<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.max(“focos", "1d", “risco”, Statistic.max, bufaggreg, buf1, "UF = ’AM’")
```

2.I.C - Operadores zonais de ocorrência
Zonal por agregação : **Média**

Retorna a média dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto terão um único valor obtido por análise estatística (utilitário estatístico).

occurrence.zonal.aggregation.mean(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <aggregation_statistic>, <aggregation_buffer>, <buffer>,“<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.mean(“focos", "1d", “risco”, Statistic.max, bufaggreg, buf1, "UF = ’AM’")
```

2.I.C - Operadores zonais de ocorrência
Zonal por agregação : **Soma**

Retorna as soma dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto terão um único valor obtido por análise estatística (utilitário estatístico).

occurrence.zonal.aggregation.sum(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <aggregation_statistic>, <aggregation_buffer>, <buffer>,“<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.sum(“focos", "1d", “risco”, Statistic.max, bufaggreg, buf1, "UF = ’AM’")
```

2.I.C - Operadores zonais de ocorrência
Zonal por agregação : **Mediana**

Retorna a mediana dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto terão um único valor obtido por análise estatística (utilitário estatístico).

occurrence.zonal.aggregation.median(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <aggregation_statistic>, <aggregation_buffer>, <buffer>,“<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.median(“focos", "1d", “risco”, Statistic.min, bufaggreg, buf1, "UF = ’AM’")
```

2.I.C - Operadores zonais de ocorrência
Zonal por agregação : **Desvio Padrão**

Retorna o desvio padrão dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto terão um único valor obtido por análise estatística (utilitário estatístico).

occurrence.zonal.aggregation.standard_deviation(
“<dynamic_data_occurrence>”, “<time>”,
<aggregation_statistic>, <aggregation_buffer>,
“<attribute>”, <buffer>,“<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.standard_deviation(“focos", "1d", “risco”, Statistic.max, bufaggreg, buf1, "UF = ’AM’")
```

2.I.C - Operadores zonais de ocorrência
Zonal por agregação : **Variância**

Retorna a variância dos valores do atributo das ocorrências que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Ocorrências agrupadas em torno do mesmo ponto terão um único valor obtido por análise estatística (utilitário estatístico).

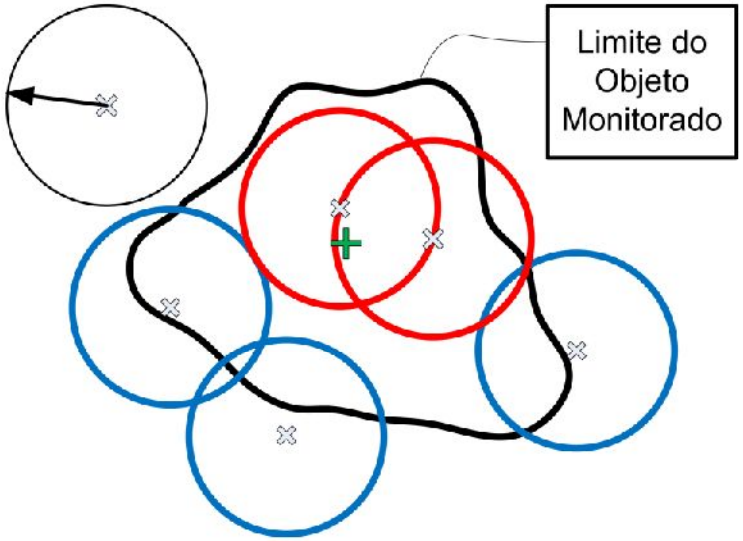
occurrence.zonal.aggregation.variance(“<dynamic_data_occurrence>”, “<time>”, “<attribute>”, <aggregation_statistic>, <aggregation_buffer>, <buffer>,“<restriction>”)

Exemplo:
```
buf1 = Buffer()
bufaggreg = Buffer(BufferType.Out,  200, “m")
x = occurrence.zonal.aggregation.variance(“focos", "1d", “risco”, Statistic.max, bufaggreg, buf1, "UF = ’AM’")
```

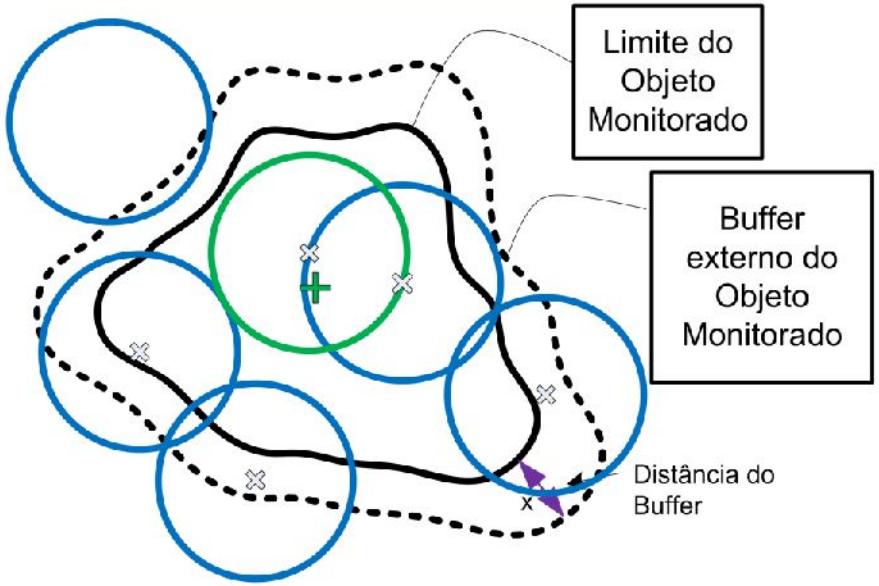
2.II - Operadores Zonais de PCDs

Operadores sobre um conjunto de pontos de PCD, porém a seleção dos pontos obedecem uma regra de influência ou valor informado.

Área de Influência das PCD e área do polígono



Área de Influência das PCD e buffer externo do polígono



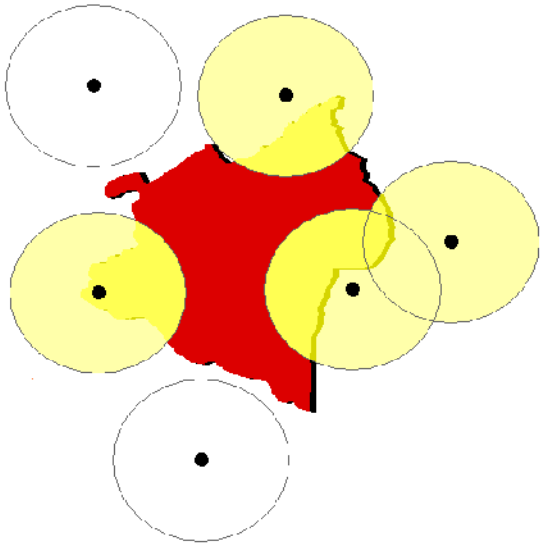
+ - Centro de massa do objeto (polígono) monitorado
○ - Área de influência da PCD não **toca** área do objeto nem engloba o **centro** de massa do objeto.
○ - Área de influência da PCD envolve o **centro** de massa do objeto e **toca** a área do mesmo
○ - Área de influência da PCD **toca** a área do objeto
○ - Área de influência da PCD envolve o **centro** de massa do objeto mas não **toca** a área do buffer externo do mesmo

2.II - Operadores zonais de PCD

Para uso dos operadores com PCD sobre objetos monitorados foi definido um utilitário para obter as PCD que influenciam os objetos monitorados.

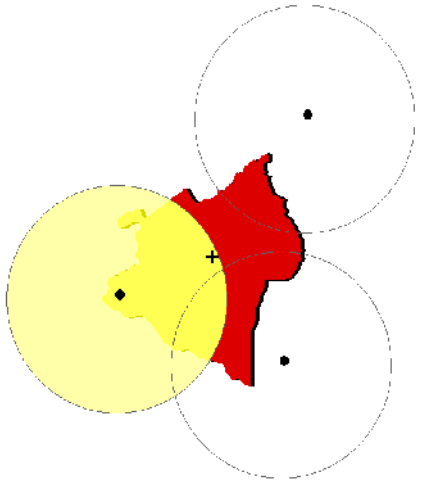
.- Influência por Regra

Raio (toca)



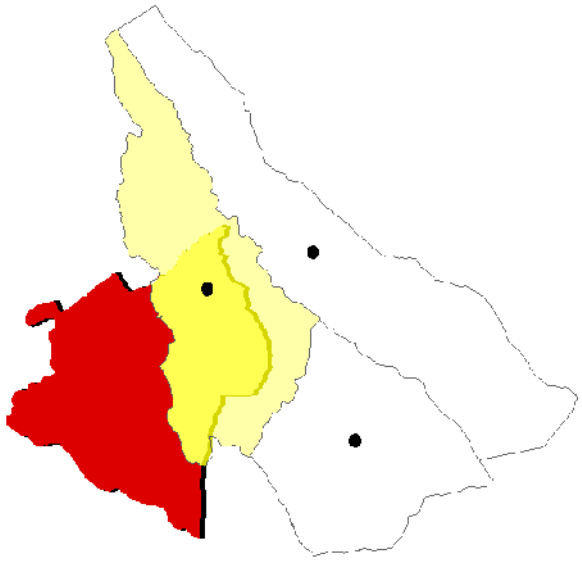
círculo de influência intersecta o polígono

Raio (centro)



círculo de influência precisa conter o centróide do polígono

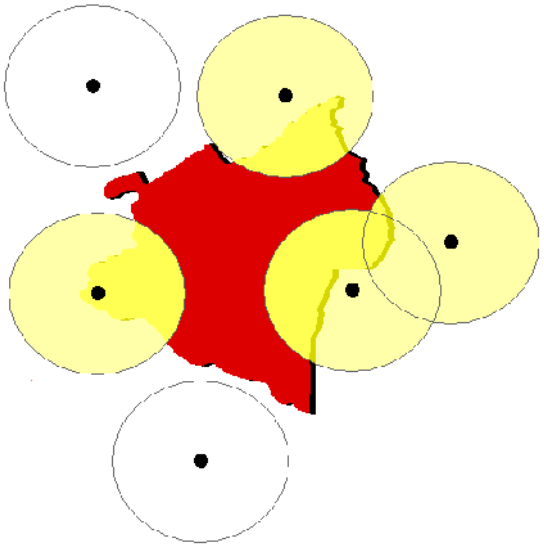
Região



um mapa estático define a área de influência de cada PCD. Um atributo de cada área identifica o código das PCDs.

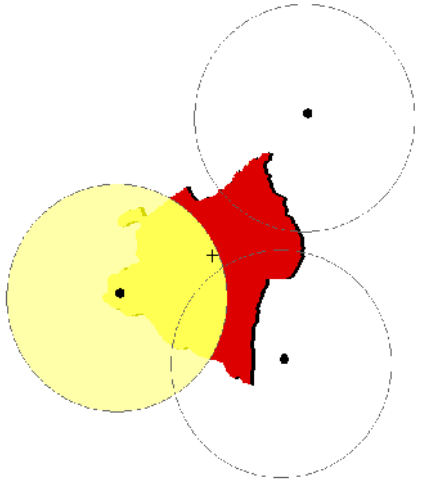
Influência por regra

Raio (toca)



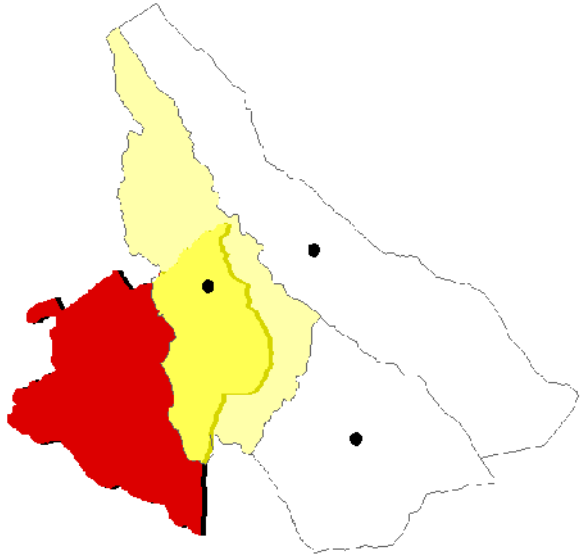
Área de influência intersecta o polígono

Raio (centro)



Área de influência precisa conter o centróide (centro de massa) do polígono

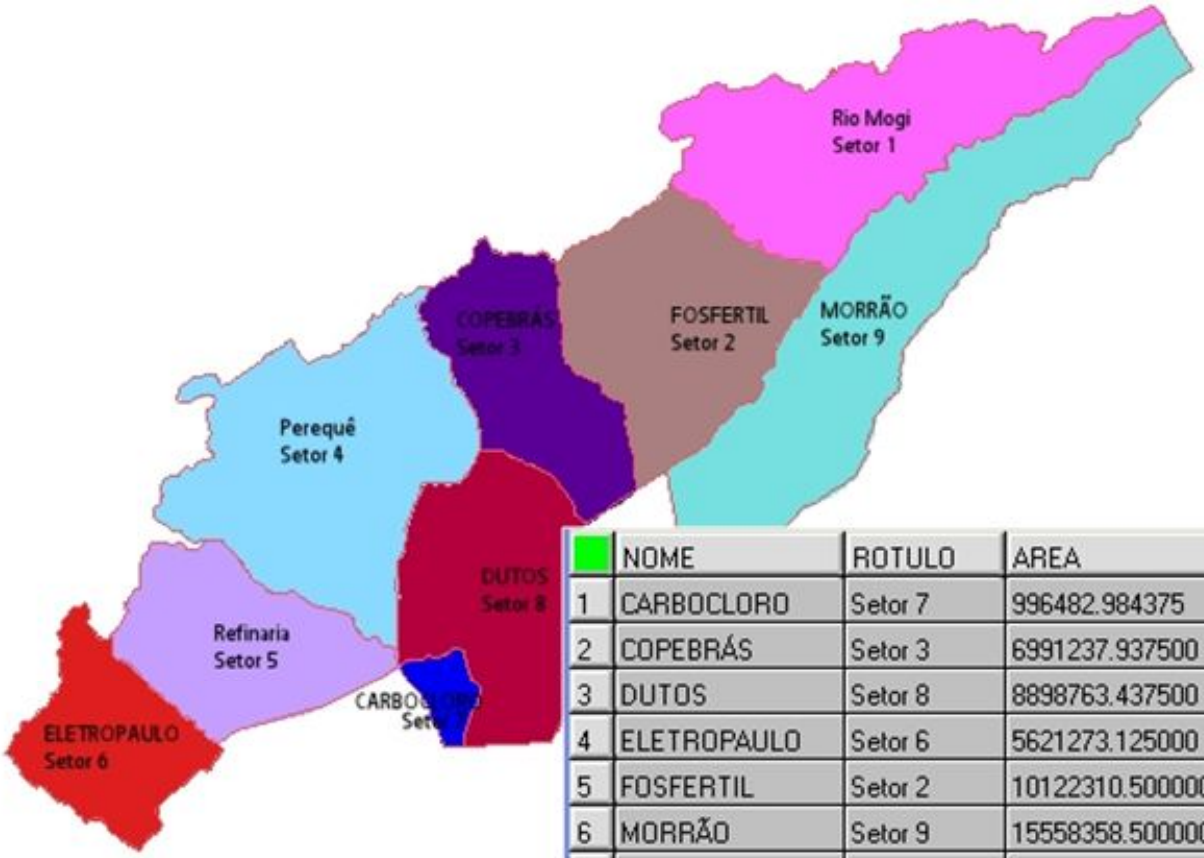
Região



Um mapa estático define a região de influência de cada PCD. Um atributo desse mapa identifica o código das PCDs.

Influência por atributo

Definido pelo atributo do Objeto



	NOME	ROTULO	AREA	K	POSTO1	POSTO2
1	CARBOCLORO	Setor 7	996482.984375	2500.00000		
2	COPEBRÁS	Setor 3	6991237.937500	2033.00000	29	26
3	DUTOS	Setor 8	8898763.437500	2700.00000		
4	ELETROPAULO	Setor 6	5621273.125000	3467.00000	24	
5	FOSFERTIL	Setor 2	10122310.500000	2033.00000	29	26
6	MORRÃO	Setor 9	15558358.500000	3945.00000	26	
7	Perequê	Setor 4	13891532.875000	2357.00000	25	
8	Refinaria	Setor 5	7559360.437500	2603.00000	28	
9	Rio Mogi	Setor 1	12707225.875000	3945.00000	26	

Um ou mais atributos do objeto monitorado (dado estático) especifica quais PCD´s devem ser consideradas.

2.III - Operadores de PCDs

Influência PCD

Operador auxiliar para um criar um vetor com a lista de PCD´s que influenciam o objeto monitorado. Dois tipos:

- Baseado nos atributos do objeto
- Baseado na regra de influência da análise

dcp.influence.by_attribute("**dynamic_data_dcp** “, **list_attribute**)

dcp.influence.by_rule("**dynamic_data_dcp** “, **buffer**) onde:

- **dynamic_data_dcp** : String com o nome da série de dados de PCD.
- **list_attribute**: Parâmetro contendo a lista de atributos do objeto monitorado contendo ID´s das PCD´s que o influenciam. Ex. [att1, att2, att3]
- **buffer** : Buffer para ser aplicado ao objeto monitorado. **Não obrigatório**. Ver utilitário Buffer

2.II - Operadores zonais de PCD

Tipos de operadores zonais de PCD estão divididos em 3 grupos

2.II.A – Zonal

Grupo de operadores que consideram as PCD´s que influenciam o objeto monitorado e utilizam somente a última medida obtidas por cada PCD.

2.II.B – Zonal Histórico

Grupo de operadores que consideram as PCD´s que influenciam o objeto monitorado e utilizam as últimas medidas obtidas por cada PCD, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

2.II.C – Zonal Histórico por Intervalo

Grupo de operadores que consideram as PCD´s que influenciam o objeto monitorado e utilizam as últimas medidas obtidas por cada PCD, no intervalo de tempo definido entre dois valores de tempo informado no passado.

2.II.A - Operadores zonais de PCD – Zonal

dcp.zonal.<operator>(“<dynamic_data_dcp>”, <buffer>, “<attribute>”, <list_dcp>) onde:

- operator : **count, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_dcp : String com o nome da série de dados de PCD
- buffer : Buffer para ser aplicado ao objeto monitorado. Parâmetro obrigatório somente se operador for “count”. Ver utilitário Buffer.
- attribute : String com o nome do atributo da PCD que deve ser utilizado para recuperar os valores. O atributo deve ser do tipo numérico (Ex. Integer, Float, Double, Long). Não usar para operador “count”
- list_dcp: lista contendo a identificação das PCD´s que influenciam o objeto monitorado. Ver operador “Influência PCD”. Não usar se operador zonal for “count”.

2.II.A - Operadores zonais de PCD
Zonal : **Contagem**

Retorna o número de PCD´s que influenciam o objeto monitorado ou sua área de influência.

dcp.zonal.count(“<dynamic_data_dcp>”, <buffer>)

Exemplo:

```
buf1 = Buffer()
x = dcp.zonal.count(“estacoes", buf1)
```

2.II.A - Operadores zonais de PCD
Zonal : **Mínimo**

Retorna o menor valor de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado.

dcp.zonal.min(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.min("Serra do Mar", "Pluvio", ids)
```

2.II.A - Operadores zonais de PCD
Zonal : **Máximo**

Retorna o maior valor de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado.

dcp.zonal.max(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.max("Serra do Mar", "Pluvio", ids)
```

2.II.A - Operadores zonais de PCD
Zonal : **Média**

Retorna a média dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado.

dcp.zonal.mean(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.mean("Serra do Mar", "Pluvio", ids)
```

2.II.A - Operadores zonais de PCD
Zonal : **Soma**

Retorna a soma dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado.

dcp.zonal.sum(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.sum("Serra do Mar", "Pluvio", ids)
```

2.II.A - Operadores zonais de PCD
Zonal : **Mediana**

Retorna a mediana dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado.

dcp.zonal.median(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.median("Serra do Mar", "Pluvio", ids)
```

2.II.A - Operadores zonais de PCD
Zonal : **Desvio Padrão**

Retorna o desvio padrão dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado.

dcp.zonal. standard_deviation(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.standard_deviation("Serra do Mar", "Pluvio", ids)
```

2.II.A - Operadores zonais de PCD
Zonal : **Variância**

Retorna a mediana dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado.

dcp.zonal.variance(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.variance("Serra do Mar", "Pluvio", ids)
```

2.II.B - Operadores zonais de PCD – Zonal Histórico

dcp.zonal.history.<operator>(“<dynamic_data_dcp>”, “<attribute>”, “<time>”, <list_dcp>) onde:

- operator :**min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_dcp : String com o nome da série de dados de PCD
- attribute : String com o nome do atributo da PCD que deve ser utilizado para recuperar os valores. O atributo deve ser do tipo numérico (Ex. Integer, Float, Double, Long).
- time: String com o intervalo de tempo para filtrar os valores de cada PCD no intervalo de tempo desejado. Ver utilitário Unidades de tempo
- list_dcp: lista contendo a identificação das PCD´s que influenciam o objeto monitorado. Ver operador “Influência PCD”.

2.II.B - Operadores zonais de PCD
Zonal Histórico: **Mínimo**

Retorna o menor valor de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

dcp.zonal.history.min(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.min("Serra do Mar", "Pluvio“, “1d”, ids)
```

2.II.B - Operadores zonais de PCD
Zonal Histórico: **Máximo**

Retorna o maior valor de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

dcp.zonal.history.max(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.max("Serra do Mar", "Pluvio“, “1d”, ids)
```

2.II.B - Operadores zonais de PCD
Zonal Histórico: **Média**

Retorna a média dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

dcp.zonal.history.mean(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.mean("Serra do Mar", "Pluvio“, “1d”, ids)
```

2.II.B - Operadores zonais de PCD
Zonal Histórico: **Soma**

Retorna a soma dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

dcp.zonal.history.sum(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.sum("Serra do Mar", "Pluvio“, “1d”, ids)
```

2.II.B - Operadores zonais de PCD
Zonal Histórico: **Mediana**

Retorna a mediana dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

dcp.zonal.history.median(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.median("Serra do Mar", "Pluvio“, “1d”, ids)
```

2.II.B - Operadores zonais de PCD
Zonal Histórico: **Desvio Padrão**

Retorna o desvio padrão dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

dcp.zonal.history.standard_deviation(“<dynamic_data_dcp>” , “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.standard_deviation("Serra do Mar", "Pluvio“, “1d”, ids)
```

2.II.B - Operadores zonais de PCD
Zonal Histórico: **Variância**

Retorna a variância dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

dcp.zonal.history.variance(“<dynamic_data_dcp>”, “<attribute>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.variance("Serra do Mar", "Pluvio“, “1d”, ids)
```

2.II.C - Operadores zonais de PCD – Zonal Histórico por Intervalo

dcp.zonal.history.interval<operator>(“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>) onde:

- operator :**min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_dcp : String com o nome da série de dados de PCD
- attribute : String com o nome do atributo da PCD que deve ser utilizado para recuperar os valores. O atributo deve ser do tipo numérico (Ex. Integer, Float, Double, Long).
- time_begin: String inicial (mais antigo) do intervalo de tempo para filtrar as ocorrências.
- time_end: String final (mais recente) do intervalo de tempo para filtrar as ocorrências.
- list_dcp: lista contendo a identificação das PCD´s que influenciam o objeto monitorado. Ver operador “Influência PCD”.

2.II.C - Operadores zonais de PCD
Zonal Histórico por Intervalo: **Mínimo**

Retorna o menor valor de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

dcp.zonal.history.interval.min(“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.interval.min("Serra do Mar", "Pluvio“, “2d”, “1d”, ids)
```

2.II.C - Operadores zonais de PCD
Zonal Histórico por Intervalo: **Máximo**

Retorna o maior valor de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

dcp.zonal.history.interval.min(“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.interval.max("Serra do Mar", "Pluvio“, “2d”, “1d”, ids)
```

2.II.C - Operadores zonais de PCD
Zonal Histórico por Intervalo: **Média**

Retorna a média dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

dcp.zonal.history.interval.mean(“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.interval.mean("Serra do Mar", "Pluvio“, “2d”, “1d”,  ids)
```

2.II.C - Operadores zonais de PCD
Zonal Histórico por Intervalo: **Soma**

Retorna a soma dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

dcp.zonal.history.interval.sum(“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.interval.sum("Serra do Mar", "Pluvio“, “2d”, “1d”, ids)
```

2.II.C - Operadores zonais de PCD
Zonal Histórico por Intervalo: **Mediana**

Retorna a madiana dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

dcp.zonal.history.interval.median(“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.interval.median("Serra do Mar", "Pluvio“, “2d”, “1d”, ids)
```

2.II.C - Operadores zonais de PCD
Zonal Histórico por Intervalo: **Desvio Padrão**

Retorna o desvio padrão dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

dcp.zonal.history.interval.standard_deviation (“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.interval.standard_deviation("Serra do Mar",
"Pluvio“, “2d”, “1d”, ids)
```

2.II.C - Operadores zonais de PCD
Zonal Histórico por Intervalo: **Variância**

Retorna a variância dos valores de um atributo comum das PCD´s que influenciam cada geometria de um objeto monitorado, no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

dcp.zonal.history.interval.variance(“<dynamic_data_dcp>”, “<attribute>”, “<time_begin>”, “<time_end>”, <list_dcp>)

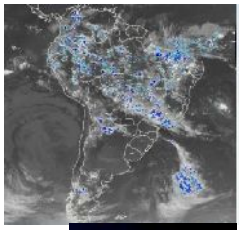
Exemplo:

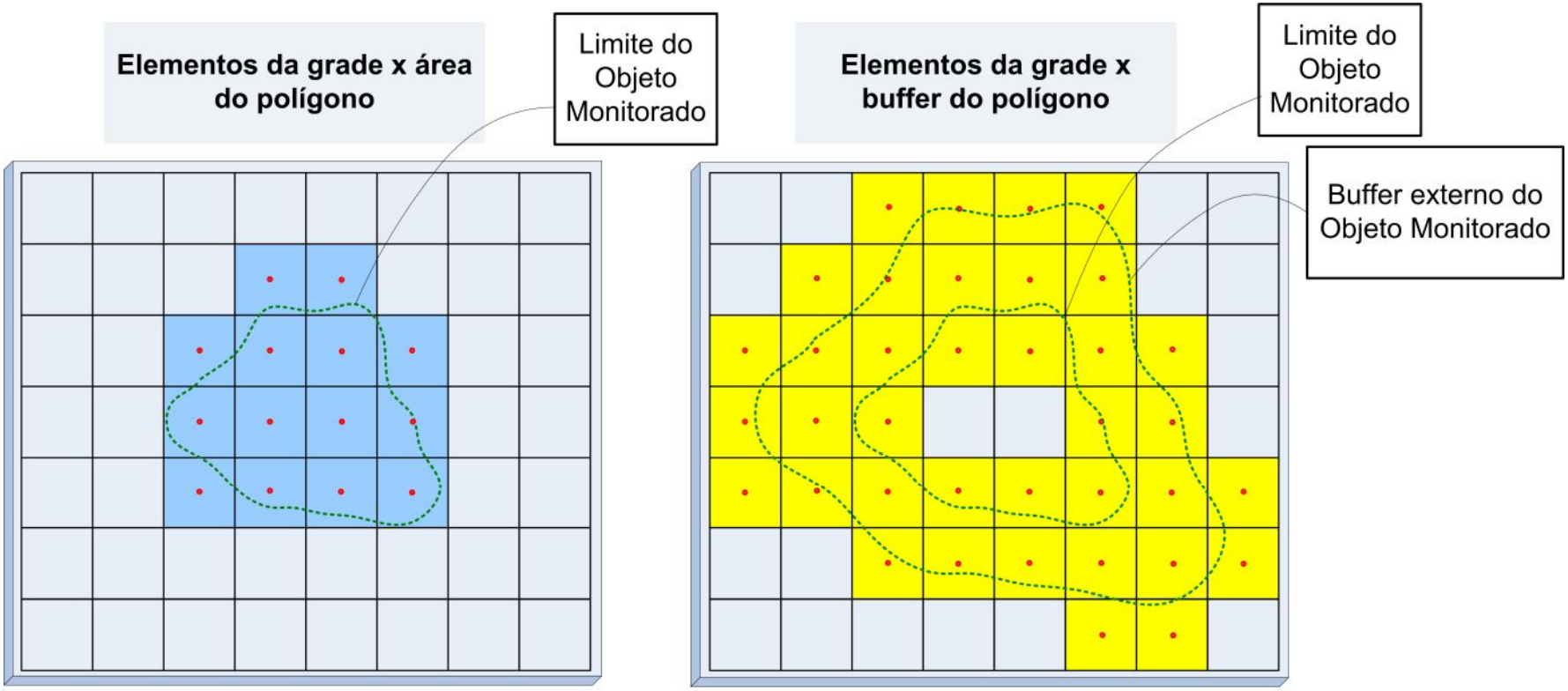
```
b1 = Buffer(BufferType.Out_union, 2, "km")
ids = dcp.influence.by_rule("Serra do Mar", b1)

x = dcp.zonal.history.interval.variance("Serra do Mar", "Pluvio“, “2d”, “1d”, ids)
```

2.III - Operadores zonais de dados matriciais

Retornam valores que fazem interseção dos pontos da grade com o objeto monitorado ou sua área de influência (buffer). O cálculo é realizado sempre que o serviço de coleta obtém uma nova grade ou por programação.





TerraMA²

2.III - Operadores zonais de dados matriciais

Tipos de operadores zonais matriciais estão divididos em 8 grupos

2.III.A – Zonal

Grupo de operadores que consideram somente o último dado dinâmico matricial (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

2.III.B – Zonal Histórico

Grupo de operadores que consideram os últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

2.III.C – Zonal Histórico de Precipitação

Grupo de operadores que consideram os últimos dados dinâmicos matriciais (em mm/h de precipitação) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

.

2.III - Operadores zonais de dados matriciais

Tipos de operadores zonais de grades estão divididos em 8 grupos

2.III.D– Zonal Histórico Acumulado

Grupo de operadores que consideram valores acumulados dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Primeiro é realizada a operação de soma dos pontos da matriz no intervalo de tempo para depois realizar a operação zonal.

2.III.E– Zonal Histórico por Intervalo

Grupo de operadores que consideram intervalo de tempo de valores dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

2.III - Operadores zonais de dados matriciais

Tipos de operadores zonais de grades estão divididos em 8 grupos

2.III.F – Zonal de Previsão

Grupo de operadores que consideram as próximas camadas de dados dinâmicos matriciais de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

2.III.G– Zonal de Previsão Acumulado

Grupo de operadores que consideram as próximas camadas de dados dinâmicos matriciais de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro. Primeiro é realizada a operação de soma dos pontos da matriz no intervalo de tempo para depois realizar a operação zonal

2.III.H– Zonal de Previsão por Intervalo

Grupo de operadores que consideram intervalo de tempo das próximas camadas de dados dinâmicos matriciais de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

2.III.A - Operadores zonais de dados matriciais - Zonal

Grupo de operadores que consideram somente o último dado dinâmico matricial (ou mais atual) que intercepta o objeto monitorado ou sua área de influência (buffer).

grid.zonal.<operator>(“<dynamic_data_grid>”, <band>, <buffer>) onde:

- Operator : **count, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de observação.
- band: [Opcional] Banda da grade ser utilizada. Parâmetro obrigatório somente se o parâmetro seguinte (buffer) for utilizado. Se não informado será considerado a primeira banda (0). Não utilizar com operador “count”.
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.III.A - Operadores zonais de dados matriciais
Zonal : **Contagem**

Retorna a quantidade de “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.count(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

buf1 = Buffer()

x = grid.zonal.count(“hidro", 0, buf1) ou

x = grid.zonal.count(“hidro“) dado que não tem buffer e banda 0

2.III.A - Operadores zonais de dados matriciais
Zonal : **Mínimo**

Retorna o menor valor dentre os “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.min(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
x = grid.zonal.min("hidro", 0, b1)
```

2.III.A - Operadores zonais de dados matriciais
Zonal : **Máximo**

Retorna o maior valor dentre os “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.max(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 400, "m")
x = grid.zonal.max("hidro", 0, b1)
```

2.III.A - Operadores zonais de dados matriciais
Zonal : **Média**

Retorna a média dos valores dentre os “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.mean(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Level,  10, “km“, 5, “km”)
x = grid.zonal.mean("hidro", 0, b1)
```

2.III.A - Operadores zonais de dados matriciais
Zonal : **Soma**

Retorna a soma dos valores dentre os “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.sum(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

grid.zonal.sum("hidro", 0) ou x = grid.zonal.sum("hidro")

2.III.A - Operadores zonais de dados matriciais
Zonal : **Mediana**

Retorna a mediana dos valores dentre os “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.median(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.median("radar", 0, b1)
```

2.III.A - Operadores zonais de dados matriciais
Zonal : **Desvio padrão**

Retorna o desvio padrão dos valores dentre os “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.standard_deviation(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.standard_deviation("radar", 0, b1)
```

2.III.A - Operadores zonais de dados matriciais
Zonal : **Variância**

Retorna a variância dos valores dentre os “pixels” da última matriz (ou mais atual) que interceptam o objeto monitorado ou sua área de influência (buffer).

grid.zonal.variance(“<dynamic_data_grid>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.In_diff, 800, "m")
x = grid.zonal.variance("grade_chuva", 0, b1)
```

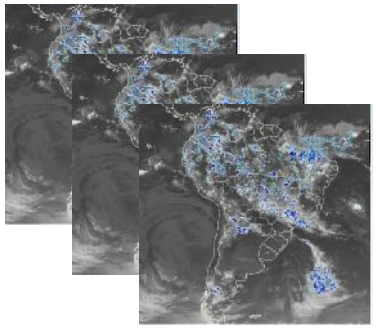
2.III.B - Operadores zonais de dados matriciais – Zonal Histórico

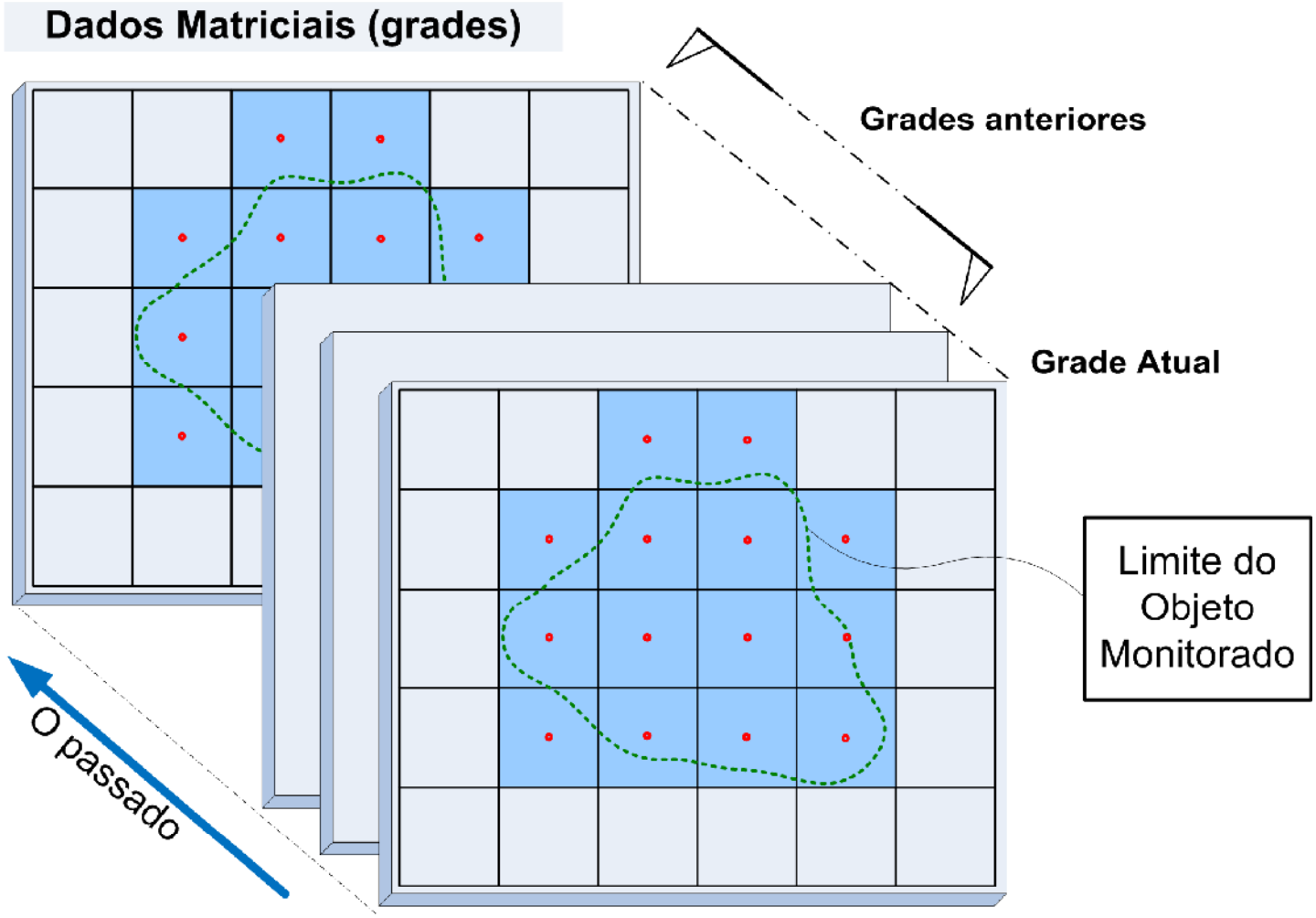
Grupo de operadores que consideram os últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.<operator>(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>) onde:

- Operator : **num, list, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de observação.
- time: String com o intervalo de tempo, a partir da hora atual, para filtrar as matrizes. Ver utilitário Unidades de tempo.
- band: [Opcional] Banda da grade ser utilizada. Parâmetro obrigatório somente se o parâmetro seguinte (buffer) for utilizado. Se não informado será considerado a primeira banda (0). Não utilizar com operador “num e list”.
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.III.B - Operadores zonais de dados matriciais – Zonal Histórico





2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Número**

Retorna o número de matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.num(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
buf1 = Buffer()
x = grid.zonal.history.num(“hidro", “12h”, buf1)
```

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Lista**

Retorna a lista com data/hora das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.list(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
buf1 = Buffer()
x = grid.zonal.history.list(“hidro", “12h”, buf1)
```

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Mínimo**

Retorna o menor valor dentre os pixels de todas as matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.min(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
x = grid.zonal.history.min("hidro", “12h”, 0, b1)
```

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Máximo**

Retorna o maior valor dentre os pixels de todas as matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.max(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 400, "m")
x = grid.zonal.history.max("hidro", “12h”, 0, b1)
```

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Média**

Retorna a média dos valores dentre os pixels de todas as matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.mean(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Level,  10, “km“, 5, “km”)
x = grid.zonal.history.mean("hidro", “12h”, 0, b1)
```

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Soma**

Retorna a soma dos valores dentre os pixels de todas as matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.sum(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

x = grid.zonal.history.sum("hidro", “12h”, 0) ou

x = grid.zonal.history.sum("hidro", “12h”)

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Mediana**

Retorna a mediana dos valores dentre os pixels de todas as matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.median(”<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.median("radar", “12h”, 0, b1)
```

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Desvio Padrão**

Retorna o desvio padrão dos valores dentre os pixels de todas as matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.standard_deviation(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.standard_deviation("radar", “12h”, 0, b1)
```

2.III.B - Operadores zonais de dados matriciais
Zonal Histórico : **Variância**

Retorna a variância dos valores dentre os pixels de todas as matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.variance(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.In_diff, 800, "m")
x = grid.zonal.history.variance("grade_chuva", “12h”, 0, b1)
```

2.III.C - Operadores zonais de dados matriciais – Zonal Histórico de Precipitação

Grupo de operadores que consideram os últimos dados dinâmicos matriciais (em mm/h de precipitação) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history,prec.<operador>(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>) onde:

- Operator : **min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de observação.
- time: String com o intervalo de tempo, a partir da hora atual, para filtrar as matrizes. Ver utilitário Unidades de tempo.
- band: [Opcional] Banda da grade ser utilizada. Parâmetro obrigatório somente se o parâmetro seguinte (buffer) for utilizado. Se não informado será considerado a primeira banda (0).
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.III.C - Operadores zonais de dados matriciais
Zonal Histórico de Precipitação : **Mínimo**

Retorna o menor valor dos últimos dados dinâmicos matriciais (em **mm/h de precipitação**) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.prec.min(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
x = grid.zonal.history.prec.min("hidro", “12h”, 0, b1)
```

2.III.C - Operadores zonais de dados matriciais
Zonal Histórico de Precipitação : **Máximo**

Retorna o maior valor dos últimos dados dinâmicos matriciais (em **mm/h de precipitação**) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.prec.max(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 400, "m")
x = grid.zonal.history.prec.max("hidro", “12h”, 0, b1)
```

2.III.C - Operadores zonais de dados matriciais
Zonal Histórico de Precipitação : **Média**

Retorna a média dos valores dos últimos dados dinâmicos matriciais (em **mm/h de precipitação**) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.prec.mean(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Level,  10, “km“, 5, “km”)
x = grid.zonal.history.prec.mean("hidro", “12h”, 0, b1)
```

2.III.C - Operadores zonais de dados matriciais
Zonal Histórico de Precipitação : **Soma**

Retorna a soma dos valores dos últimos dados dinâmicos matriciais (em **mm/h de precipitação**) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.prec.sum(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

x = grid.zonal.history.prec.sum("hidro", “12h”, 0)

2.III.C - Operadores zonais de dados matriciais
Zonal Histórico de Precipitação : **Mediana**

Retorna a mediana dos valores dos últimos dados dinâmicos matriciais (em **mm/h de precipitação**) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.prec.median(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.prec.median("radar", “12h”, 0, b1)
```

2.III.C - Operadores zonais de dados matriciais
Zonal Histórico de Precipitação : **Desvio Padrão**

Retorna o desvio padrão dos valores dos últimos dados dinâmicos matriciais (em **mm/h de precipitação**) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.prec.standard_deviation(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.prec.standard_deviation("radar", “12h”, 0, b1)
```

2.III.C - Operadores zonais de dados matriciais
Zonal Histórico de Precipitação : **Variância**

Retorna a variância dos valores dos últimos dados dinâmicos matriciais (em **mm/h de precipitação**) que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.prec.variance(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.In_diff, 800, "m")
x = grid.zonal.history.prec.variance("grade_chuva", “12h”, 0, b1)
```

2.III.D - Operadores zonais de dados matriciais – Zonal Histórico Acumulado

Grupo de operadores que consideram valores acumulados dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado. Primeiro é realizada a operação de soma dos pontos da matriz no intervalo de tempo para depois realizar a operação zonal.

grid.zonal.history.accum.<operator>(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>) onde:

- Operator : **min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de observação.
- time: String com o intervalo de tempo, a partir da hora atual, para filtrar as matrizes. Ver utilitário Unidades de tempo.
- band: [Opcional] Banda da grade ser utilizada. Parâmetro obrigatório somente se o parâmetro seguinte (buffer) for utilizado. Se não informado será considerado a primeira banda (0).
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.III.D - Operadores zonais de dados matriciais
Zonal Histórico Acumulado : **Mínimo**

Retorna a **menor soma acumulada** dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.accum.min(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.accum.standard_deviation("radar", “12h”, 0, b1)
```

2.III.D - Operadores zonais de dados matriciais
Zonal Histórico Acumulado : **Máximo**

Retorna a **maior soma acumulada** dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.accum.max(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.accum.standard_deviation("radar", “12h”, 0, b1)
```

2.III.D - Operadores zonais de dados matriciais
Zonal Histórico Acumulado : **Média**

Retorna a **média da soma acumulada** dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.accum.mean(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.accum.standard_deviation("radar", “12h”, 0, b1)
```

2.III.D - Operadores zonais de dados matriciais
Zonal Histórico Acumulado : **Soma**

Retorna a **soma da soma acumulada** dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.accum.sum(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.accum.standard_deviation("radar", “12h”, 0, b1)
```

2.III.D - Operadores zonais de dados matriciais
Zonal Histórico Acumulado : **Mediana**

Retorna a **mediana da soma acumulada** dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.accum.median(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.accum.standard_deviation("radar", “12h”, 0, b1)
```

2.III.D - Operadores zonais de dados matriciais
Zonal Histórico Acumulado : **Desvio Padrão**

Retorna o **desvio padrão da soma acumulada** dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.accum.standard_deviation(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.accum.standard_deviation("radar", “12h”, 0, b1)
```

2.III.D - Operadores zonais de dados matriciais
Zonal Histórico Acumulado : **Variância**

Retorna a **variância da soma acumulada** dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no passado.

grid.zonal.history.accum.variance(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.accum.standard_deviation("radar", “12h”, 0, b1)
```

2.III.E - Operadores zonais de dados matriciais – Zonal Histórico por Intervalo

Grupo de operadores que consideram intervalo de tempo de valores dos últimos dados dinâmicos matriciais que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.<operator>(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>) onde:

- Operator : **num, list, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de observação.
- time_begin: String inicial (mais antigo) do intervalo de tempo para filtrar as matrizes.
- time_end: String final (mais recente) do intervalo de tempo para filtrar as matrizes.
- band: [Opcional] Banda da grade ser utilizada. Parâmetro obrigatório somente se o parâmetro seguinte (buffer) for utilizado. Se não informado será considerado a primeira banda (0). Não utilizar com operador “num e list”.
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Número**

Retorna o número de matrizes no intervalo de tempo que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.num(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

```
buf1 = Buffer()
x = grid.zonal.history.interval.num(“hidro", “12h”, “2h”, buf1)
```

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Lista**

Retorna a lista com data/hora das matrizes no intervalo de tempo que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.list(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

```
buf1 = Buffer()
x = grid.zonal.history.interval.list(“hidro", “12h”, “2h”, buf1)
```

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Mínimo**

Retorna o menor valor dentre os pixels, no intervalo de tempo, das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.min(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
x = grid.zonal.history.interval.min("hidro", “12h”, “2h”, 0, b1)
```

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Máximo**

Retorna o maior valor dentre os pixels, no intervalo de tempo, das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.max(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 400, "m")
x = grid.zonal.history.interval.max("hidro", “12h”, “2h”, 0, b1)
```

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Média**

Retorna a média dos valores dentre os pixels, no intervalo de tempo, das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.mean(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Level,  10, “km“, 5, “km”)
x = grid.zonal.history.interval.mean("hidro", “12h”, “2h”, 0, b1)
```

TerraMA²

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Soma**

Retorna a soma dos valores dentre os pixels, no intervalo de tempo, das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.sum(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>)

Exemplo:

x = grid.zonal.history.interval.sum("hidro", “12h”, “2h”, 0) ou

x = grid.zonal.history.interval.sum("hidro", “12h”, “2h”)

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Mediana**

Retorna a mediana dos valores dentre os pixels, no intervalo de tempo, das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.median(”<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.interval.median("radar", “12h”, “2h”, 0, b1)
```

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Desvio Padrão**

Retorna o desvio padrão dos valores dentre os pixels, no intervalo de tempo, das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

grid.zonal.history.interval.standard_deviation(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.history.interval.standard_deviation("radar", “12h”, “2h”, 0, b1)
```

2.III.E - Operadores zonais de dados matriciais
Zonal Histórico por Intervalo : **Variância**

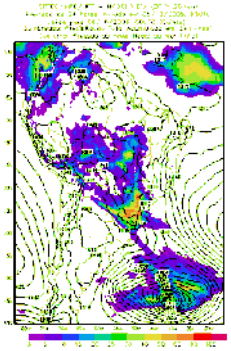
Retorna a variância dos valores dentre os pixels, no intervalo de tempo, das matrizes que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no passado em função da data/hora atual.

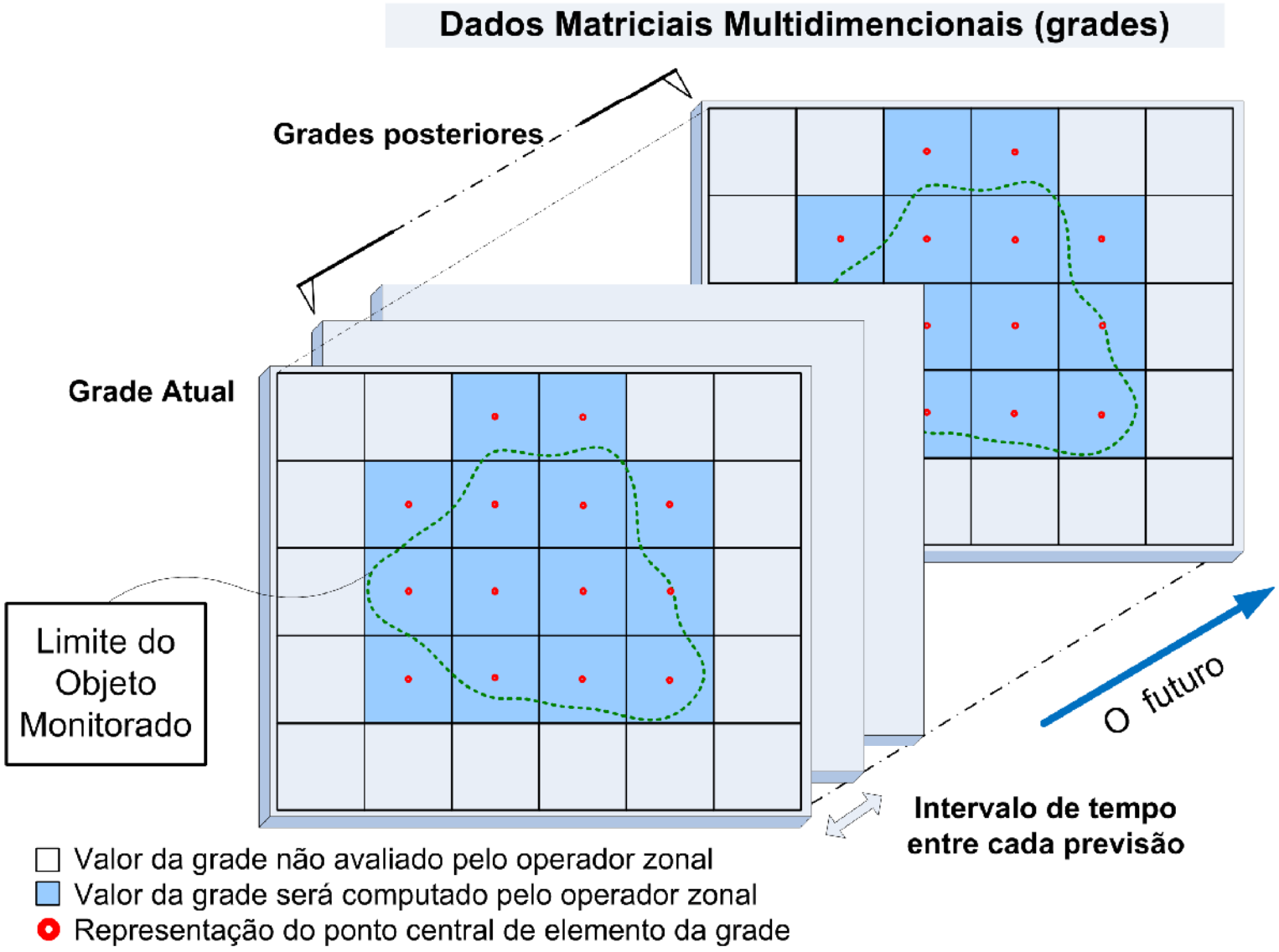
grid.zonal.history.interval.variance(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <band>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.In_diff, 800, "m")
x = grid.zonal.history.interval.variance("grade_chuva", “12h”, “2h”, 0, b1)
```

2.III.F - Operadores zonais de dados matriciais – Zonal de Previsão





2.III.F - Operadores zonais de dados matriciais – Zonal de Previsão

Operadores sobre dados matriciais multidimensionais (multicamadas).

Cada camada do arquivo representa um horário da previsão para uma quantidade da variável (não acumulada) prevista (precipitação, intensidade de vento, umidade relativa, temperatura, etc.).

Cada modelo de previsão tem seu intervalo de tempo definido entre cada camada.

As analises são feitas sobre as camadas a partir do horário atual para o frente (o futuro), ou intervalo de tempo no futuro.
O usuário define um número de horas para qual ele deseja calcular a previsão da variável escolhida e então é realizada uma operação zonal dentro desse número de horas.

Para todos os operadores, primeiro é realizada a operação de soma sobre os pontos da grade para depois realizar a operação zonal.

2.III.F - Operadores zonais de dados matriciais – Zonal de Previsão

Grupo de operadores que consideram as próximas camadas de dados dinâmicos matriciais de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.<operator>(“<dynamic_data_grid>”, “<time>”, <band>, <buffer>) onde:

- Operator : **num, list, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de previsão.
- time: String com o intervalo de tempo, a partir da hora atual para o futuro, para filtrar as camadas de previsão. Ver utilitário Unidades de tempo.
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório. Não utilizar com operador “num e list”.

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Número**

Retorna o número de camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.num(“<dynamic_data_grid>”, “<time>”)

Exemplo:

x = grid.zonal.forecast.num(“Eta15km", “12h”)

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Lista**

Retorna a lista com data/hora das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.list(“<dynamic_data_grid>”, “<time>)

Exemplo:
x = grid.zonal.forecast.list(“Eta15km", “12h”)

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Mínimo**

Retorna o menor valor dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.min(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
x = grid.zonal.forecast.min("Eta15km ", “12h”, b1)
```

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Máximo**

Retorna o maior valor dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.max(“<dynamic_data_grid>”, “<time>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 400, "m")
x = grid.zonal.forecast.max("Eta15km ", “12h”, b1)
```

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Média**

Retorna a média dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.mean(“<dynamic_data_grid>”, “<time>, <buffer>)

Exemplo:

b1 = Buffer(BufferType.Level, 10, “km“, 5, “km”)

x = grid.zonal.forecast.mean("Eta15km ", “12h”, b1)

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Soma**

Retorna a soma dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.sum(“<dynamic_data_grid>”, “<time>, <buffer>)

Exemplo:

x = grid.zonal.forecast.sum("Eta15km ", “12h”)

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Mediana**

Retorna a mediana dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.median(”<dynamic_data_grid>”, “<time>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.forecast.median("Eta15km ", “12h”, b1)
```

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Desvio Padrão**

Retorna o desvio padrão dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.standard_deviation(“<dynamic_data_grid>”, “<time>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.forecast.standard_deviation("Eta15km ", “12h”, b1)
```

2.III.F - Operadores zonais de dados matriciais
Zonal de Previsão : **Variância**

Retorna a variância dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.variance(“<dynamic_data_grid>”, “<time>, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.In_diff, 800, "m")
x = grid.zonal.forecast.variance("Eta15km ", “12h”, b1)
```

2.III.G - Operadores zonais de dados matriciais – Zonal de Previsão Acumulado

Grupo de operadores que consideram as próximas camadas de dados dinâmicos matriciais de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro. Primeiro é realizada a operação de soma dos pontos da matriz no intervalo de tempo para depois realizar a operação zonal

grid.zonal.forecast.accum<operator>(“<dynamic_data_grid>”, “<time>”, <buffer>) onde:

- Operator : **count, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de previsão.
- time: String com o intervalo de tempo, a partir da hora atual para o futuro, para filtrar as camadas de previsão. Ver utilitário Unidades de tempo.
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Contagem**

Retorna a quantidade de “pixels” das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.num(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

x = grid.zonal.forecast.accum.num(“Eta15km", “12h”)

TerraMA²

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Mínimo**

Retorna o menor valor dentre os pixels das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.min(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
x = grid.zonal.forecast.accum.min("Eta15km ", “12h”, b1)
```

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Máximo**

Retorna o maior valor dentre os pixels das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.max(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 400, "m")
x = grid.zonal.forecast.accum.max("Eta15km ", “12h”, b1)
```

TerraMA²

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Média**

Retorna a média dos valores dentre os pixels das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.mean(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Level,  10, “km“, 5, “km”)
x = grid.zonal.forecast.accum.mean("Eta15km ", “12h”, b1)
```

TerraMA²

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Soma**

Retorna a soma dos valores dentre os pixels das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.sum(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

x = grid.zonal.forecast.accum.sum("Eta15km ", “12h”)

TerraMA²

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Mediana**

Retorna a mediana dos valores dentre os pixels das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.median(”<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.forecast.accum.median("Eta15km ", “12h”, b1)
```

TerraMA²

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Desvio Padrão**

Retorna o desvio padrão dos valores dentre os pixels das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.standard_deviation(“<dynamic_data_grid>”, “<time>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.forecast.accum.standard_deviation("Eta15km ", “12h”, b1)
```

2.III.G - Operadores zonais de dados matriciais
Zonal de Previsão Acumulado : **Variância**

Retorna a variância dos valores dentre os pixels das camadas de previsão acumulado que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo definido entre a data/hora atual e o valor de tempo informado no futuro.

grid.zonal.forecast.accum.variance(“<dynamic_data_grid>”, “<time_begin>”, “<time>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.In_diff, 800, "m")
x = grid.zonal.forecast.accum.variance("Eta15km ", “12h”, b1)
```

TerraMA²

2.III.H - Operadores zonais de dados matriciais – Zonal de Previsão por Intervalo

Grupo de operadores que consideram intervalo de tempo das próximas camadas de dados dinâmicos matriciais de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.interval.<operator>(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>) onde:

- Operator : **num, list, min, max, mean, sum, median, standard_deviation, variance**
- dynamic_data_grid : String com o nome da série de dados matriciais de previsão.
- time_begin: String inicial (mais próximo da hora atual) do intervalo de tempo para filtrar as camadas de previsão.
- time_end: String final (mais distante da hora atual) do intervalo de tempo para filtrar as camadas de previsão.
- buffer : [Opcional] Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Número**

Retorna o número de camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.num(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”)

Exemplo:

x = grid.zonal.forecast.num(“Eta15km", “12h”)

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Lista**

Retorna a lista com data/hora das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.list(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”)

Exemplo:
x = grid.zonal.forecast.list(“Eta15km", “12h”)

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Mínimo**

Retorna o menor valor dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.accum.min(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out_union, 2, "km")
x = grid.zonal.forecast.accum.min("Eta15km ", “12h”, b1)
```

TerraMA²

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Máximo**

Retorna o maior valor dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.accum.max(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 400, "m")
x = grid.zonal.forecast.accum.max("Eta15km ", “12h”, b1)
```

TerraMA²

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Média**

Retorna a média dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.accum.mean(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

b1 = Buffer(BufferType.Level, 10, “km“, 5, “km”)

x = grid.zonal.forecast.accum.mean("Eta15km ", “12h”, b1)

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Soma**

Retorna a soma dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.accum.sum(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

x = grid.zonal.forecast.accum.sum("Eta15km ", “12h”)

TerraMA²

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Mediana**

Retorna a mediana dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.accum.median(”<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.forecast.accum.median("Eta15km ", “12h”, b1)
```

TerraMA²

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Desvio Padrão**

Retorna o desvio padrão dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.accum.standard_deviation(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, <buffer>)

Exemplo:

```
b1 = Buffer(BufferType.Out, 800, "m")
x = grid.zonal.forecast.accum.standard_deviation("Eta15km ", “12h”, b1)
```

2.III.H - Operadores zonais de dados matriciais
Zonal de Previsão por Intervalo : **Variância**

Retorna a variância dos valores dentre os pixels das camadas de previsão que interceptam o objeto monitorado ou sua área de influência (buffer) no intervalo de tempo inicial e final informado no futuro em função da data/hora atual.

grid.zonal.forecast.accum.variance(“<dynamic_data_grid>”, “<time_begin>”, “<time_end>”, “<time>”, <buffer>)

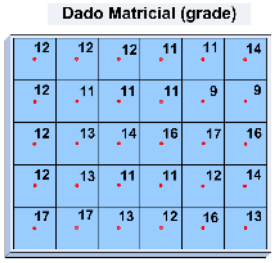
Exemplo:

```
b1 = Buffer(BufferType.In_diff, 800, "m")
x = grid.zonal.forecast.accum.variance("Eta15km ", “12h”, b1)
```

3 - Operadores com Grades

Operadores com Grades

I. Operadores da Grade
II. Operadores Históricos de Grades
III. Operadores Históricos de Grades com intervalo
IV. Operadores Históricos de Grades de Previsão
V. Operadores Históricos de Grades de Previsão com intervalo



Exemplo: local var1 = **amostras**(´hidro´)
Resultado: var1 = cada valor da grade

3.I- Operadores da Grade

Conjunto de operadores para dado do tipo **Grade**

grid.<operator>(“dataSeriesName”) onde:

- Operator : **sample, min, max, mean, sum, median, standard_deviation, variance**
- dataSeriesName : String com o nome da série de dados de grades de observação

3.I - Operadores da Grade – Amostra

Retorna o valor da célula da grade. Se resolução da grade diferente da grade de saída da análise, será utilizado o método de interpolação escolhido.

grid.sample(“dataSeriesName”)

3.I - Operadores da Grade

min, max, mean, sum, median, standard_deviation, variance

Retorna o valor **mínimo** (ou **máximo, média, soma, mediana, desvio padrão** ou **variância**) da grade atual.

```
grid.min(“dataSeriesName”)
grid.max(“dataSeriesName”)
grid.mean(“dataSeriesName”)
grid.sum(“dataSeriesName”)
grid.median(“dataSeriesName”)
grid.standard_desviation(“dataSeriesName”)
grid.variance(“dataSeriesName”)
```

3.II - Operadores Históricos de Grade

min, max, mean, sum, median, standard_deviation, variance

Retorna uma grade com valores **mínimo** (ou **máximo, média, soma, mediana, desvio padrão** ou **variância**) dos pontos da grade de observação em questão no tempo determinado a partir do atual

Sintaxe: grid.history.min(“dataSeriesName”, “dateFilter”)

grid.history.max(“dataSeriesName”, “dateFilter”)

grid.history.mean(“dataSeriesName”, “dateFilter”)

grid.history.sum(“dataSeriesName”, “dateFilter”)

grid.history.median(“dataSeriesName”, “dateFilter”)

grid.history.standard_deviation(“dataSeriesName”, “dateFilter”)

grid.history.variance(“dataSeriesName”, “dateFilter”)

3.III - Operadores Históricos de Grade com Intervalo

min, max, mean, sum, median, standard_deviation, variance

Retorna uma grade com valores **mínimo** (ou **máximo, média, soma, mediana, desvio padrão** ou **variância**) dos pontos da grade de observação em questão no intervalo de tempo determinado.

Sintaxe: grid.history.interval.min(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.history.interval.max(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.history.interval.mean(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.history.interval.sum(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.history.interval.median(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.history.interval.standard_deviation(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.history.variance(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

Op. Amostra Ponto Histórico da grade de previsão numérica

Retorna (MIN, MAX ,SOMA,MEDIA, etc) os pontos da grade de previsão numérica em questão no tempo determinado a partir do horário atual.

Sintaxe: grid.forecast.min(“dataSeriesName”, “dateFilter”)

grid.forecast.max(“dataSeriesName”, “dateFilter”)

grid.forecast.mean(“dataSeriesName”, “dateFilter”)

grid.forecast.sum(“dataSeriesName”, “dateFilter”)

grid.forecast.median(“dataSeriesName”, “dateFilter”)

grid.forecast.standard_deviation(“dataSeriesName”, “dateFilter”)

grid.forecast.variance(“dataSeriesName”, “dateFilter”)

Op. Amostra Ponto Histórico da grade de previsão numérica (INTERVALO)

Retorna (MIN,MAX ,SOMA,MEDIA) os pontos da grade de previsão numérica em questão de um intervalo de tempo determinado.

Sintaxe: grid.forecast.interval.min(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.forecast.interval.max(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.forecast.interval.mean(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.forecast.interval.sum(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.forecast.interval.median(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.forecast.interval.standard_deviation(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

grid.forecast.interval.variance(“dataSeriesName”, “dateFilterBegin”, “dateFilterEnd”)

4 - Operadores de PCD

Operadores com dados de PCD

I. Operadores entre PCD´s

II. Operadores Históricos de PCD´s

4.I Operadores entre PCD´s

Retorna (MIN, MAX ,SOMA,MEDIA, etc) dos pontos de PCD´s de um atributo no horário atual.

Sintaxe: dcp.min(“attribute”)

dcp.max(“attribute”)

dcp.mean(“attribute”)

dcp.sum(“attribute”)

dcp.median(“attribute”)

dcp.standard_deviation(“attribute”)

dcp.variance(“attribute”)

4.II Operadores Históricos de PCD´s

Retorna (COUNT, MIN, MAX ,SOMA,MEDIA, etc) de cada PCD num intervalo de tempo a patir do horário atual.

Sintaxe: dcp.history.count(“dateFilter”)

dcp.history.min(“attribute”, “dateFilter”)

dcp.history.max(“attribute”, “dateFilter”)

dcp.history.mean(“attribute”, “dateFilter”)

dcp.history.sum(“attribute”, “dateFilter”)

dcp.history.median(“attribute”, “dateFilter”)

dcp.history.standard_deviation(“attribute”, “dateFilter”)

dcp.history.variance(“attribute”, “dateFilter”)