

TerraMA²

Inovações na versão 4

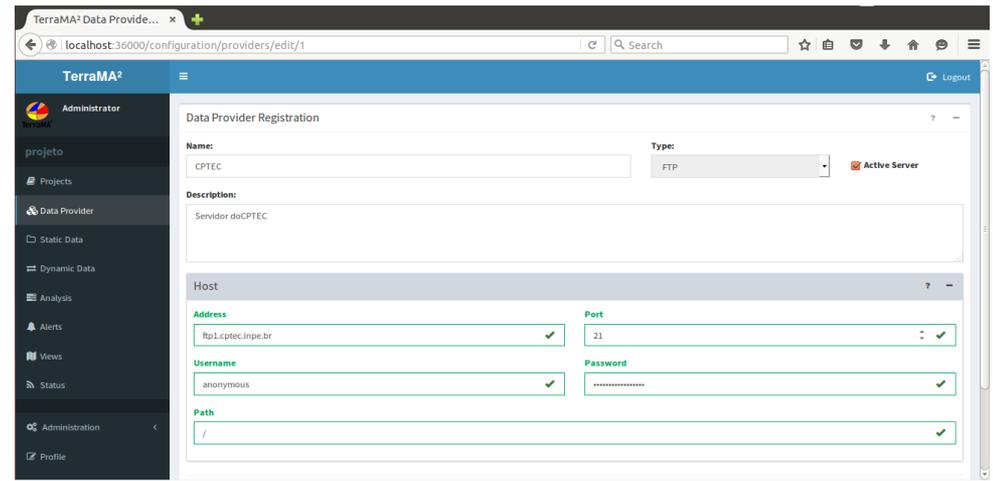
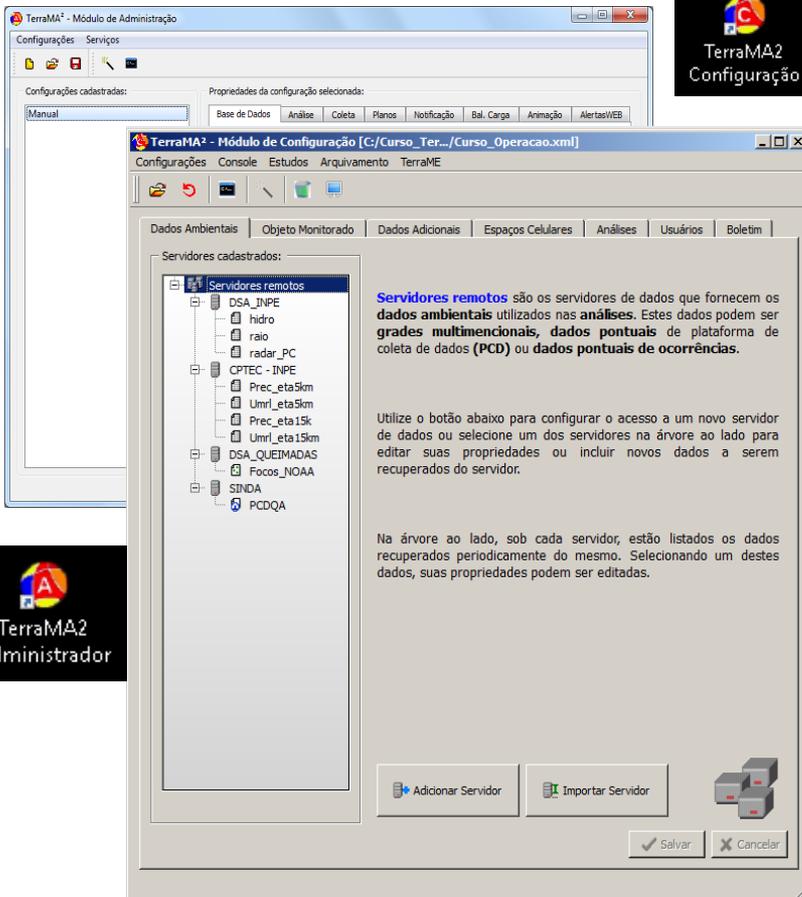
EYMAR LOPES
GILBERTO RIBEIRO



Diferenças

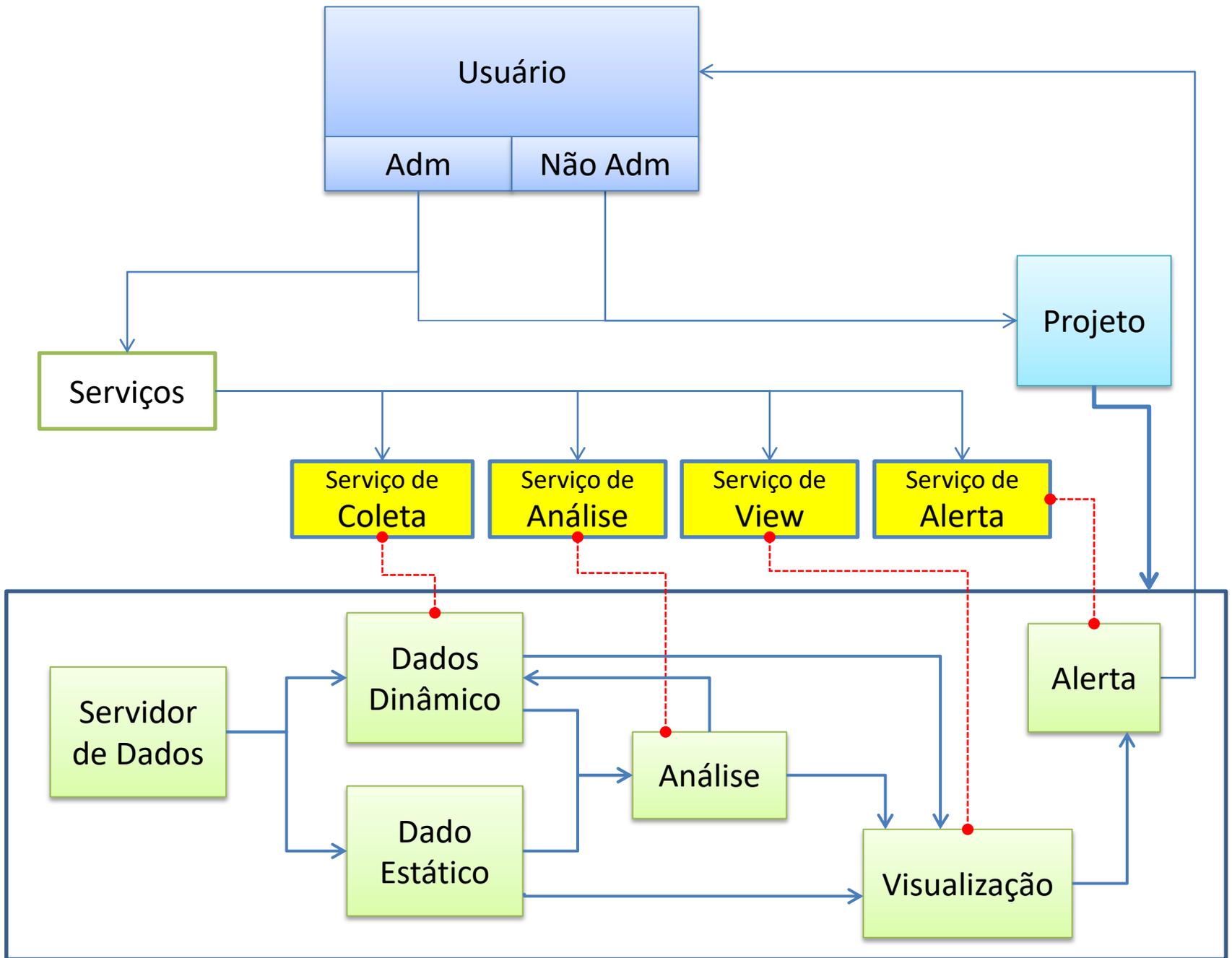
- Módulo de Administração e Configuração (2 executáveis)

- Módulo de Administração com diferentes perfis de usuário (1 aplicação web)



Diferenças TerraMA² v3 x v4

- Uso da TerraLib 4
- Necessidade dos módulos de Adm e Conf. serem executados na mesma máquina onde está o banco
- Serviços locais
- Somente cadastro de usuário para acesso a aplicação web de monitoramento.
- Ambiente de trabalho carregado pro um arquivo .
- Todos os dados e metadados num único banco
- Módulo de Monitoramento com TerraOGC
- Estilo do TerraView 4.2
- Versões para Win e Linux
- Uso da TerraLib 5
- Módulo de Administração Web com diferentes perfis de usuário (1 aplicação web)
- Serviços locais e remotos (Ssh)
- Administração de usuários com privilégio de administrador ou não.
- Conceito de projeto
- Dados Geo. distribuídos em arquivos ou tabelas
- Módulo de Monitoramento com Geoserver
- Estilos do Geoserver
- Versões para Win, Linux e Mac



TerraMA²: 2015-2017

Projeto

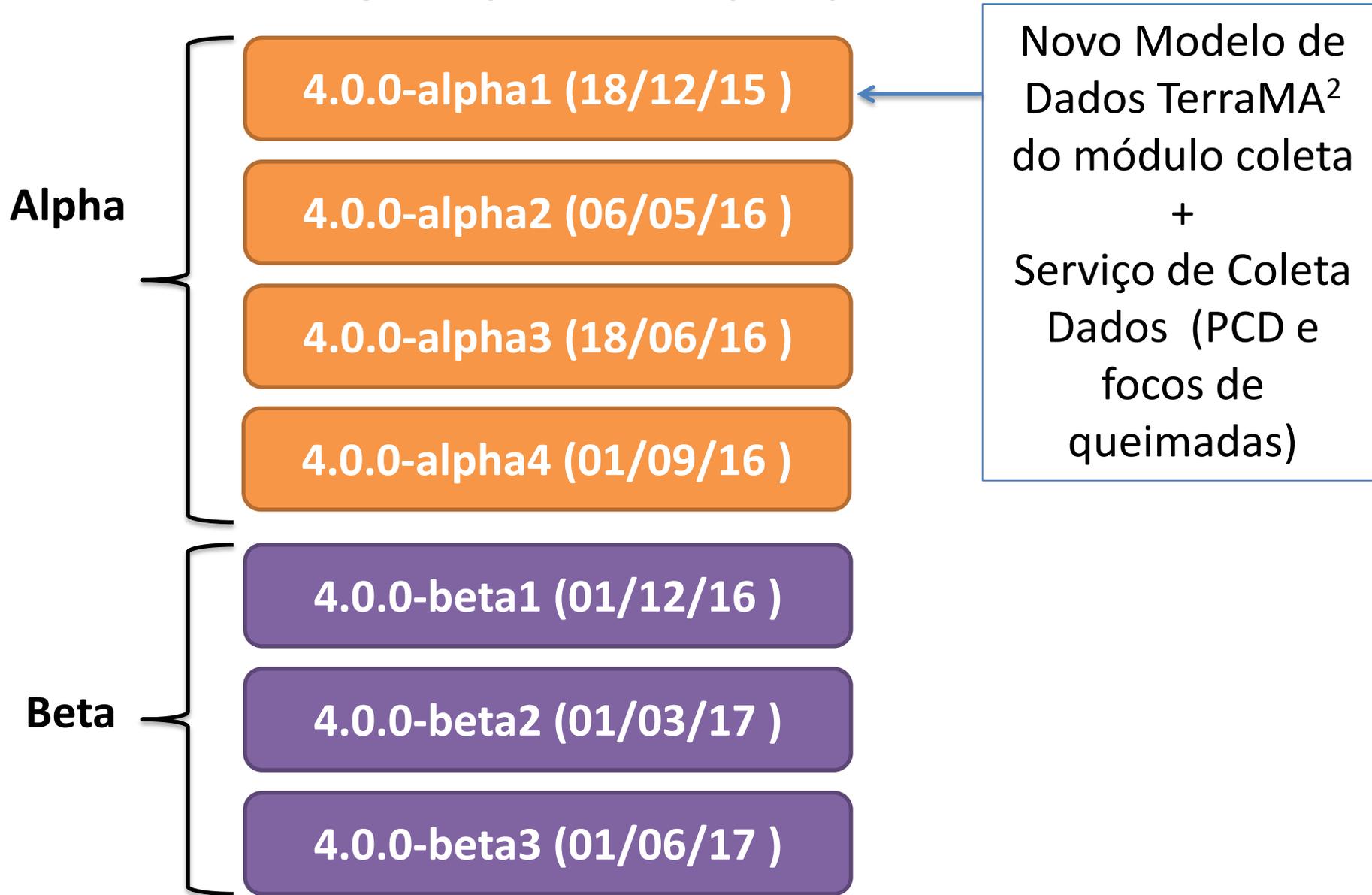
“BRAZIL CERRADO CLIMATE CHANGE MITIGATION PLATFORM OF MONITORING AND WARNING OF FOREST FIRES IN THE BRAZILIAN CERRADO PROJECT”

(Alberto Setzer)

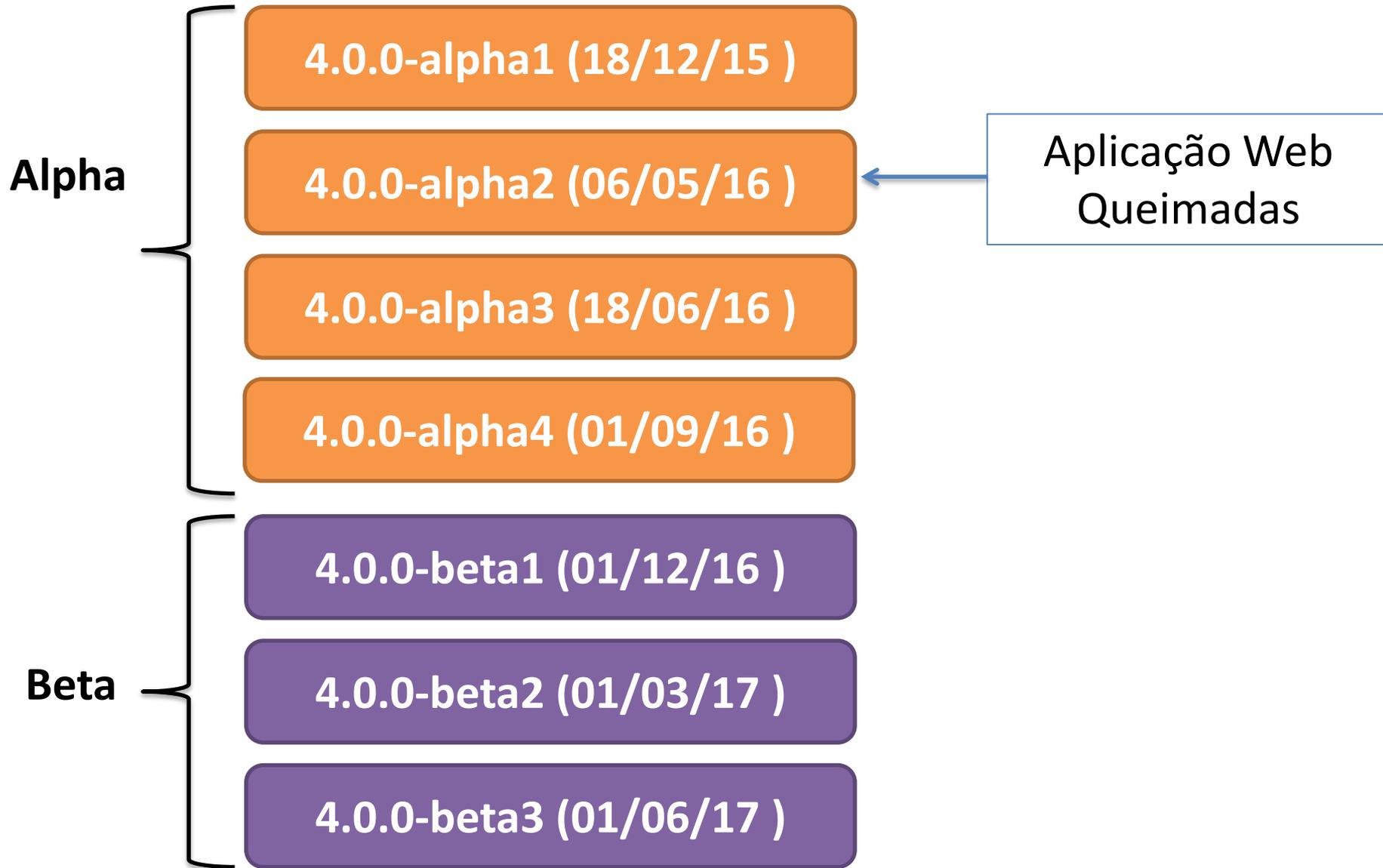
Financiamento: Banco Mundial

Prazo: 30 meses a partir de 01-Julho de 2015

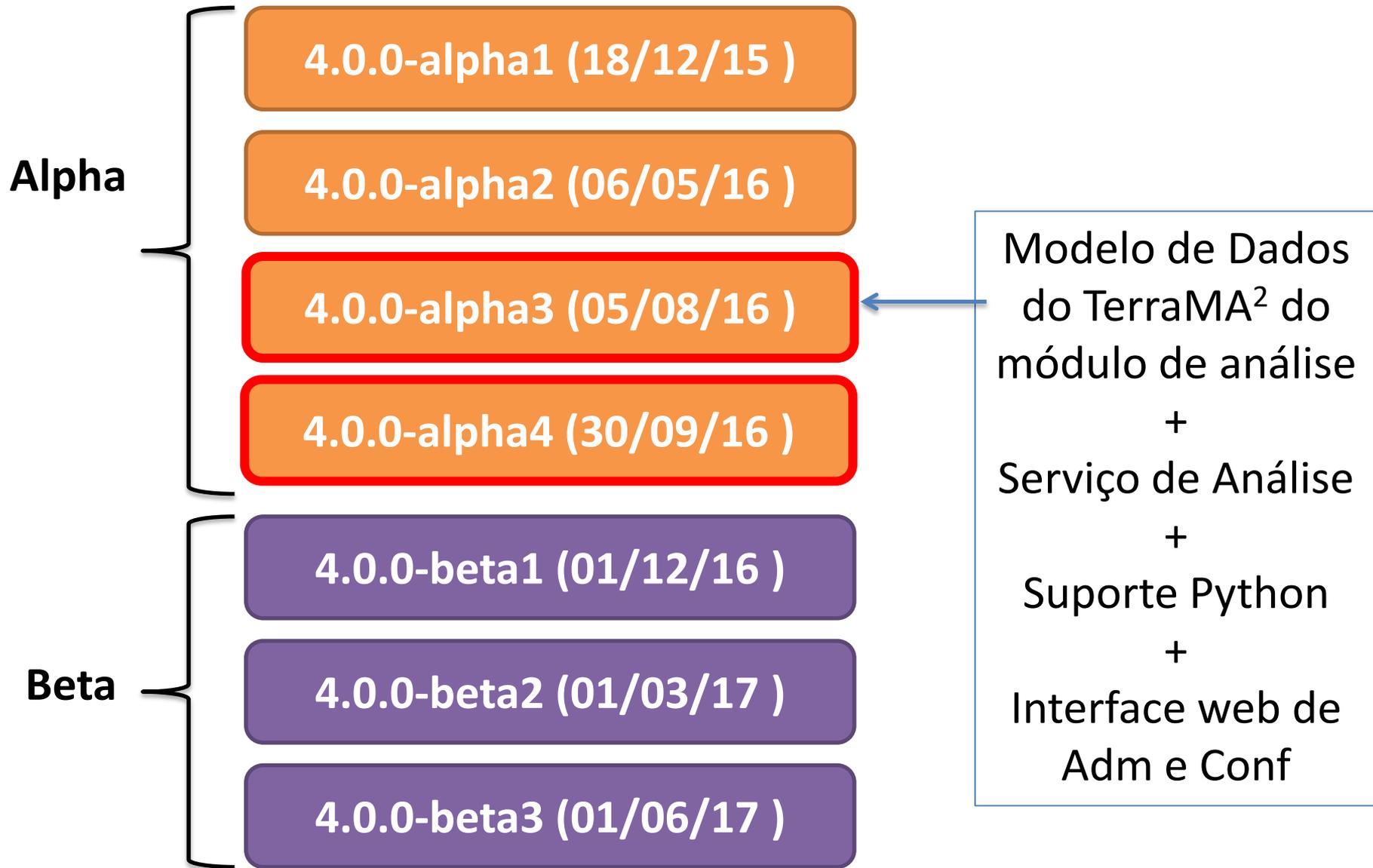
Roadmap: <https://trac.dpi.inpe.br/terraama2>



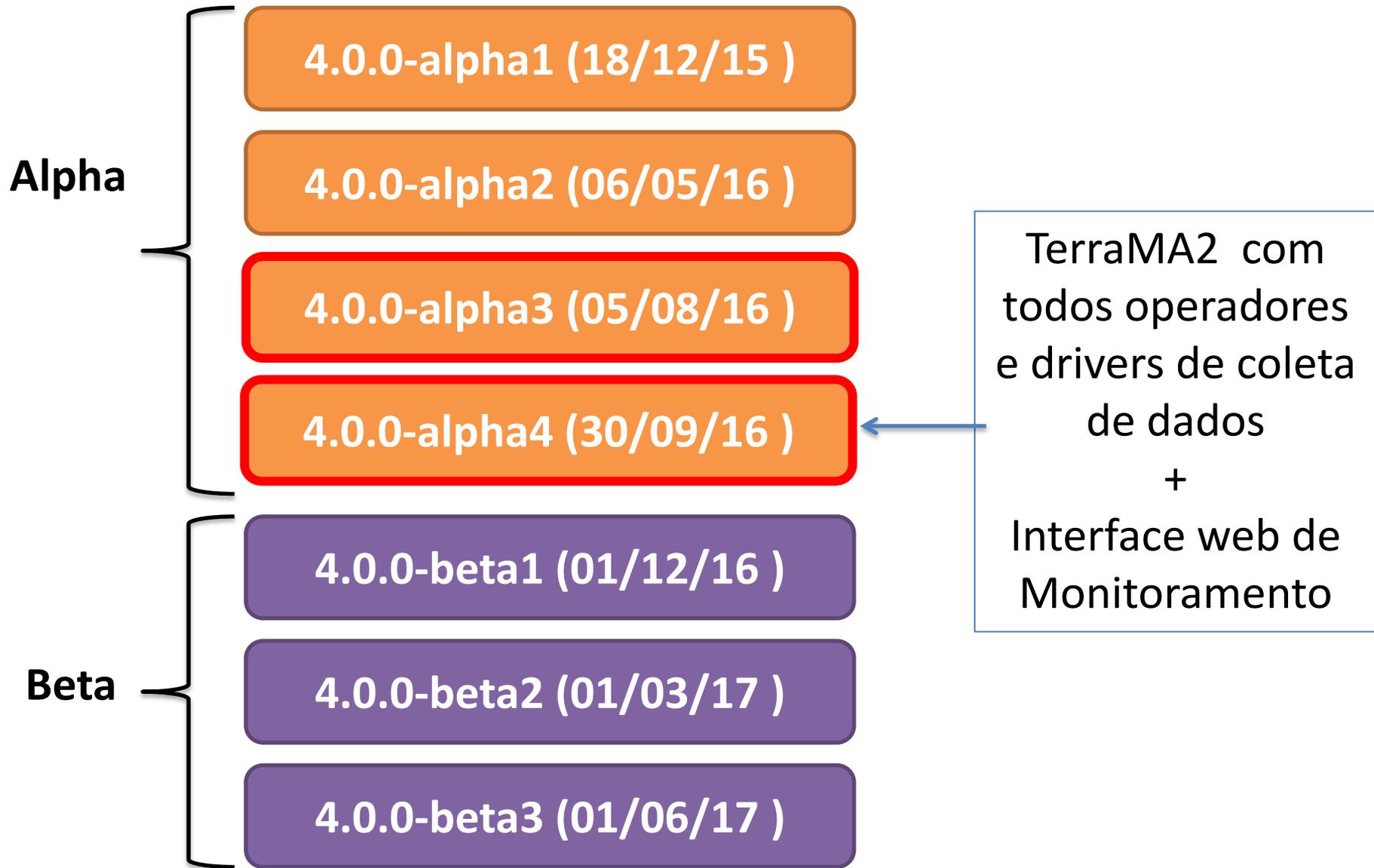
Roadmap: <https://trac.dpi.inpe.br/terrama2>



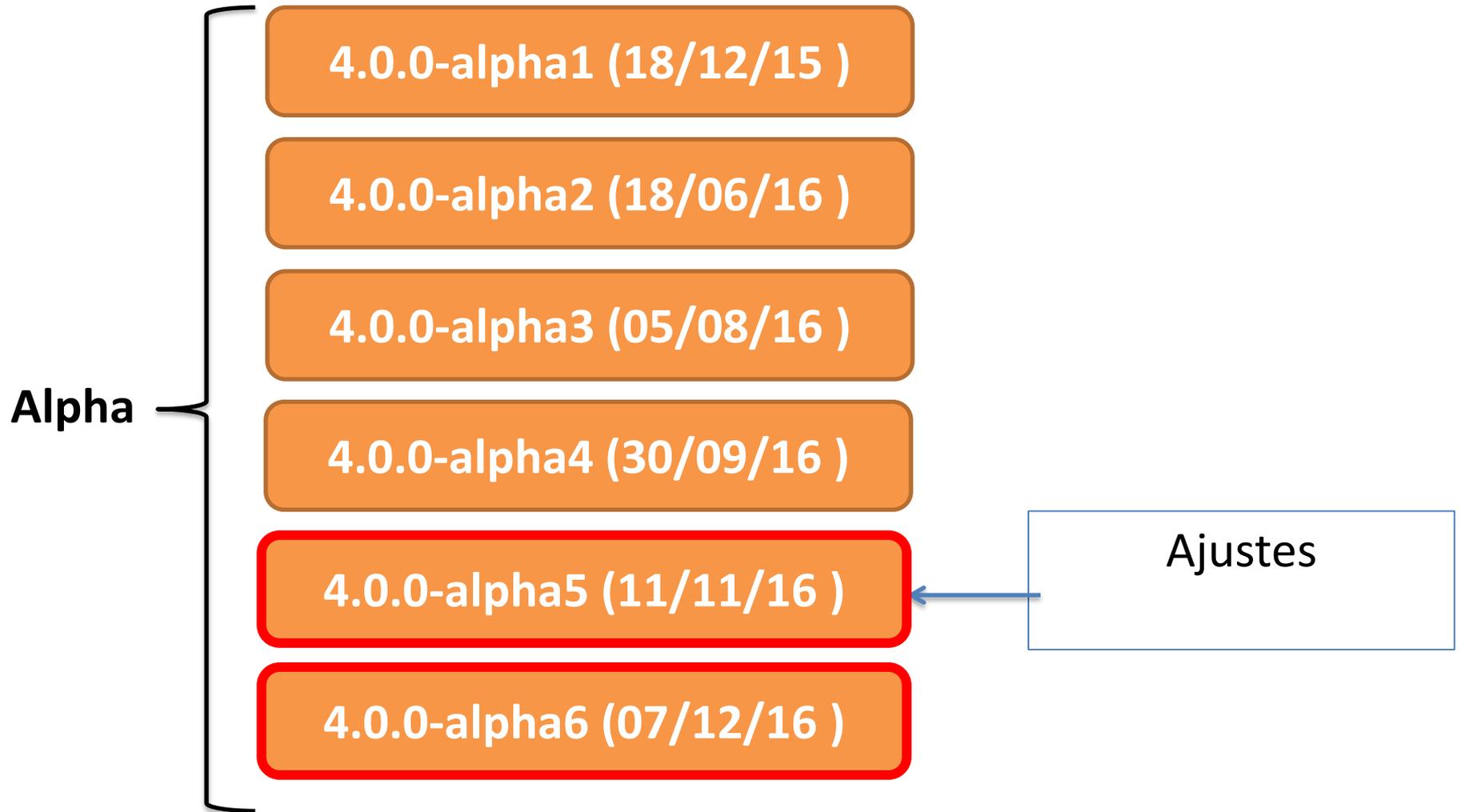
Roadmap: <https://trac.dpi.inpe.br/terrama2>



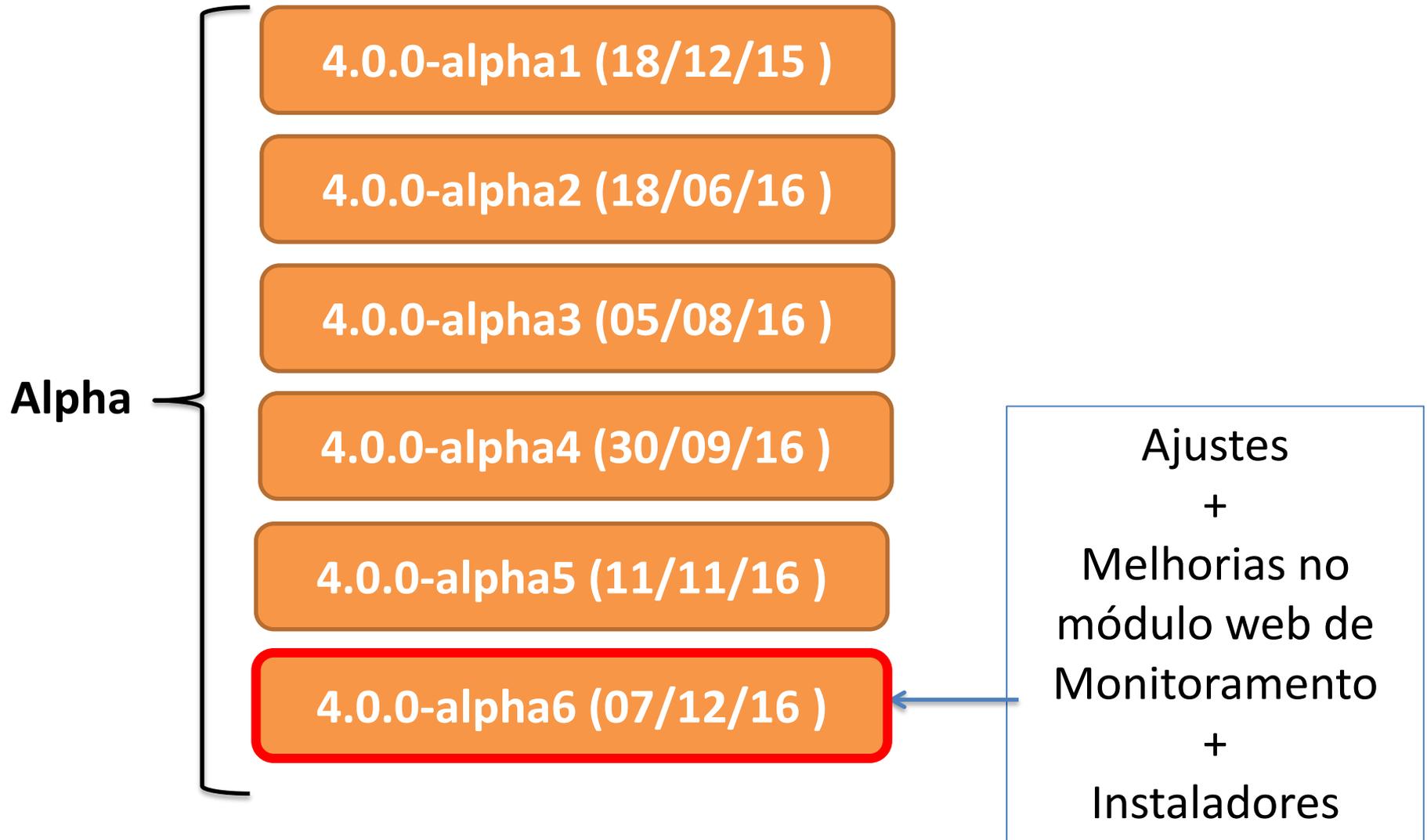
Roadmap: <https://trac.dpi.inpe.br/terrama2>



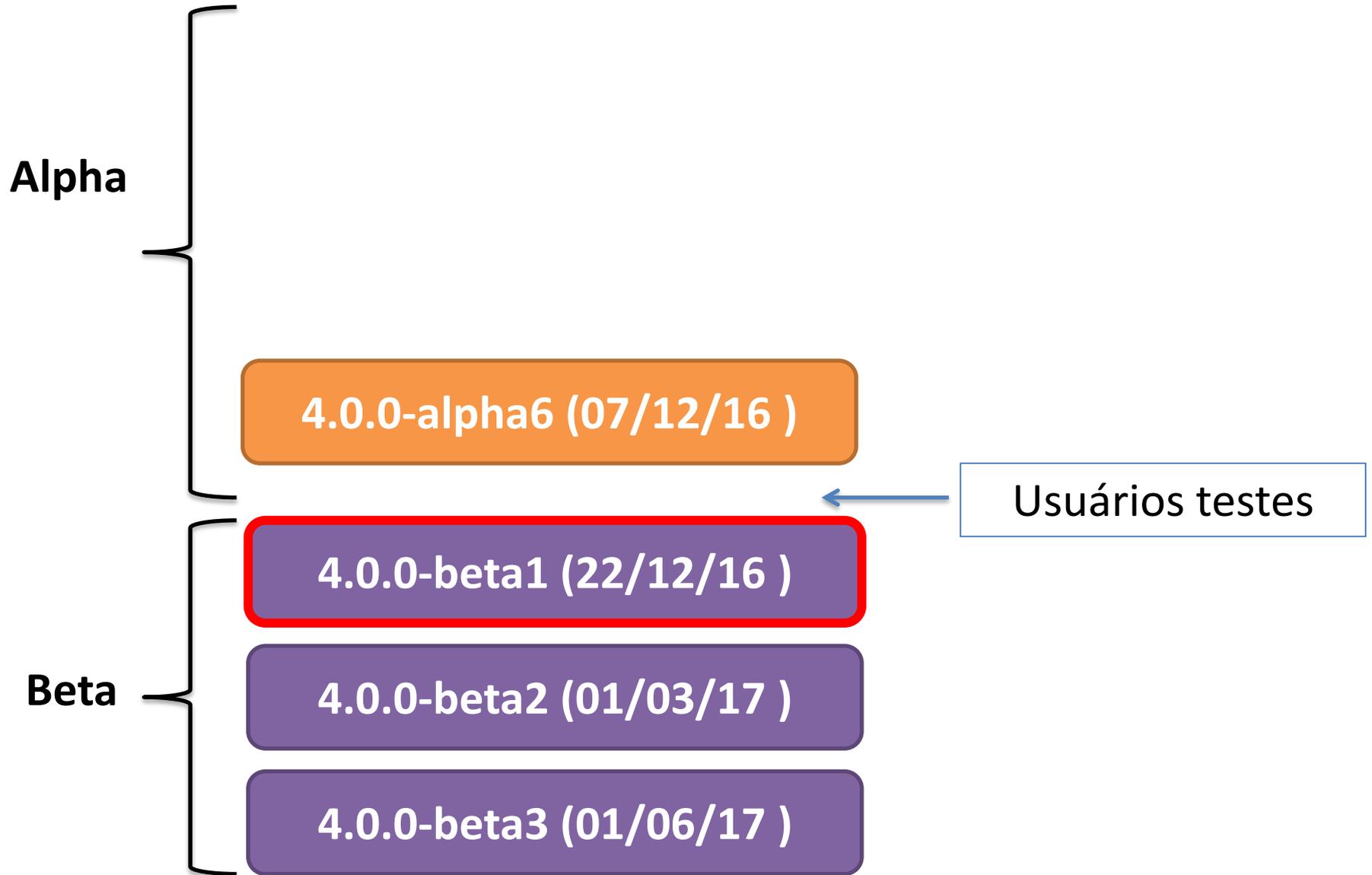
Roadmap: <https://trac.dpi.inpe.br/terrama2>



Roadmap: <https://trac.dpi.inpe.br/terrama2>



Roadmap: <https://trac.dpi.inpe.br/terrามา2>



Roadmap: <https://trac.dpi.inpe.br/terrama2>

**Release
Candidate**

4.0.0-rc1 (01/08/17)

4.0.0-rc2 (01/09/17)

4.0.0-rc3 (02/10/17)

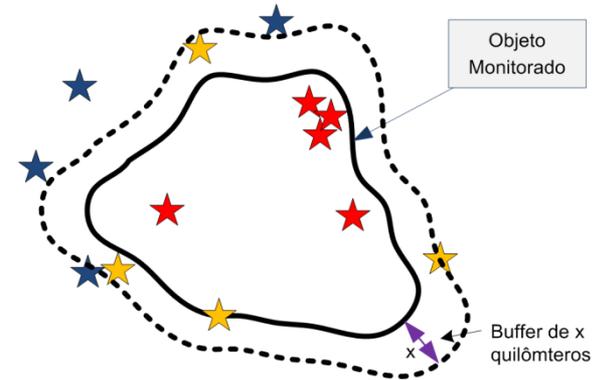
4.0.0-rc4 (01/11/17)

4.0.0 (01/12/17)

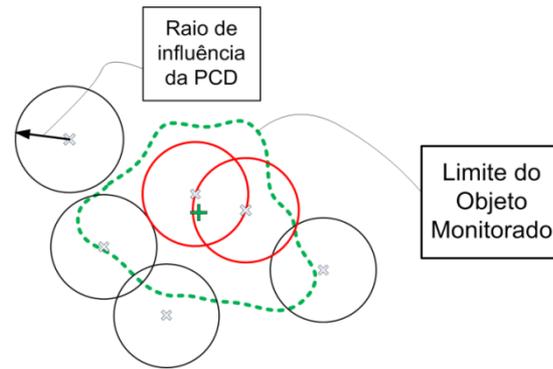
Dados Dinâmicos

Três tipos de Dados Dinâmicos podem ser coletados de servidores locais ou remotos

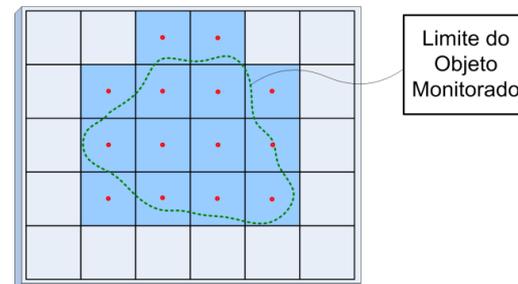
Dados de Ocorrências



Dados de PCD (pontos fixos)



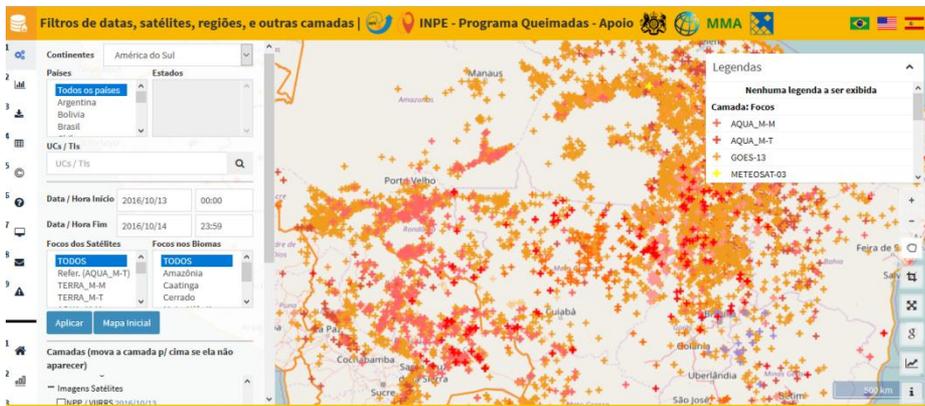
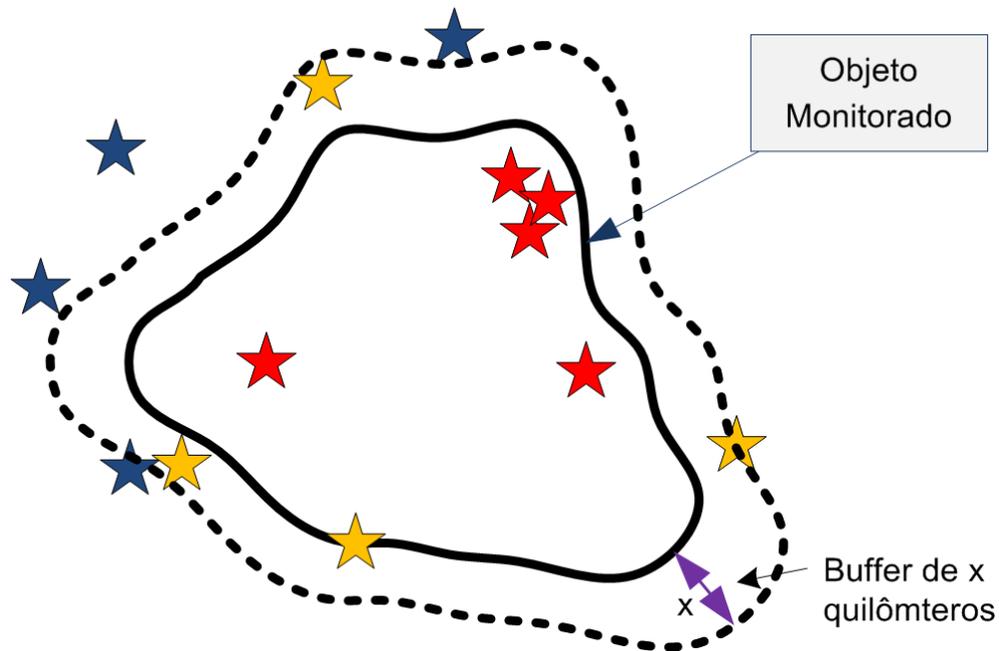
Grades numéricas multidimensional



Dados Dinâmicos

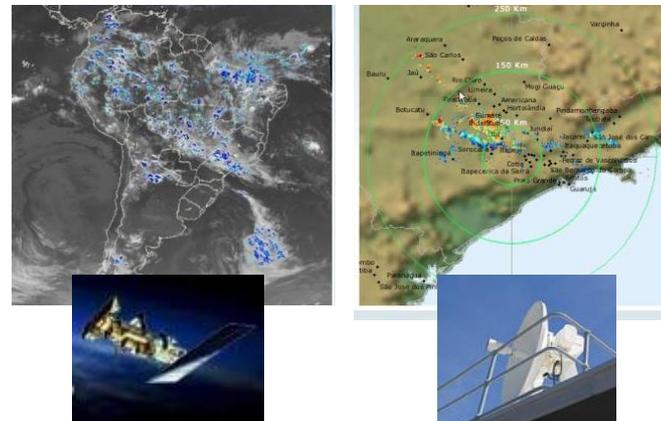
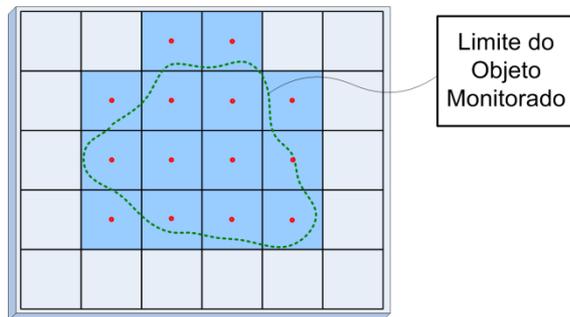
Dados de ocorrências

- Focos de incêndios
- Focos de doenças
- Sismos
- Descargas elétricas
- Ocorrências gerais



Dados Dinâmicos

Grades numéricas multidimensional

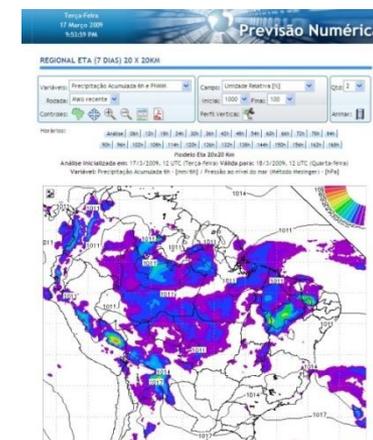
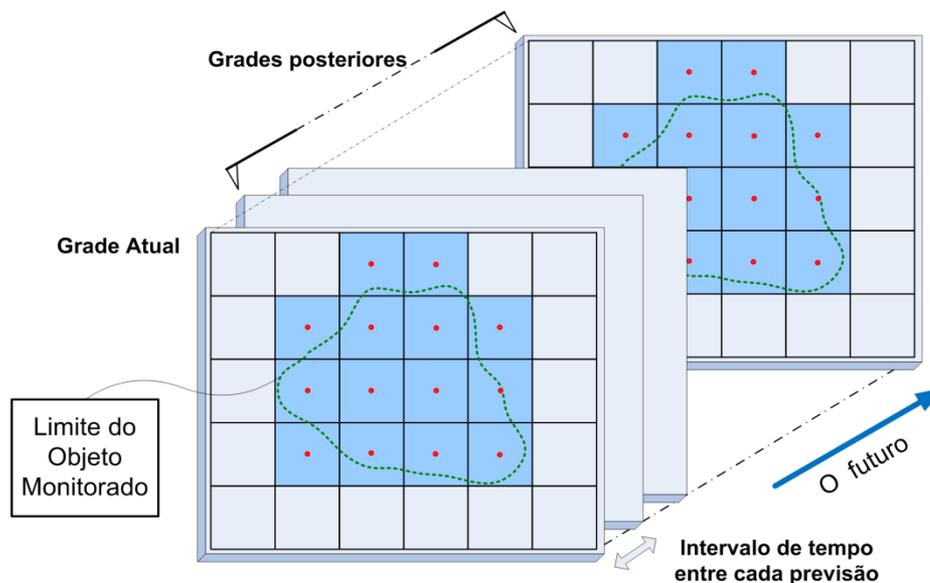


OBSERVAÇÃO

- Hidroestimador
- Raios
- Radar meteorológico

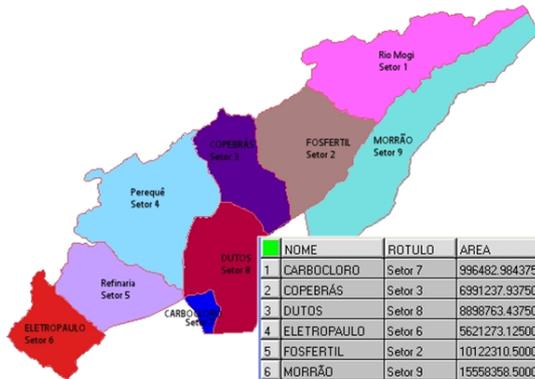
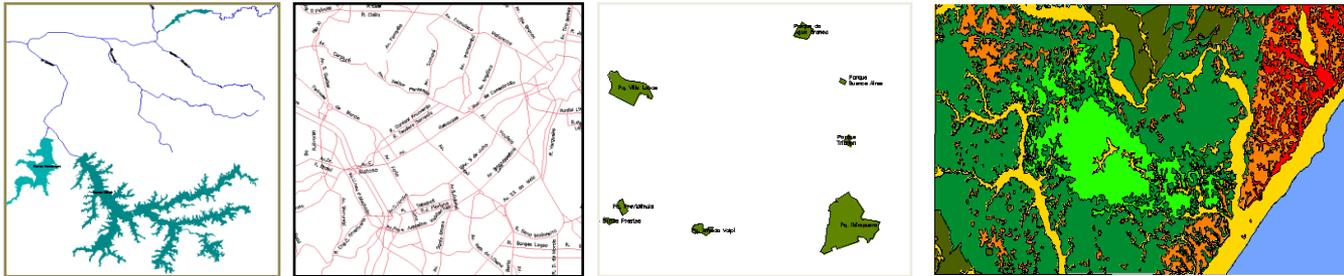
PREVISÃO

- Modelos de previsão



Dados Estáticos

- Mapas vetoriaiais – rios, estradas, dutos, áreas ocupadas, mapas de risco, etc.
 - Utilizados como objetos a serem monitorados nas análises, onde são cruzados com dados dinâmicos ou outros estáticos
 - Podem fazer interseção espacial com o objeto de monitoramento
 - Podem ser utilizados apenas para visualização no módulo de monitoramento WEB



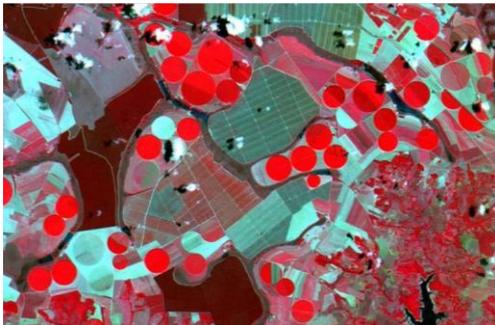
	■	NOME	ROTULO	AREA	K	POSTO1	POSTO2
1		CARBOCLORO	Sector 7	996482.984375	2500.00000		
2		COPEBRÁS	Sector 3	6991237.937500	2033.00000	29	26
3		DUTOS	Sector 8	8898763.437500	2700.00000		
4		ELETROP PAULO	Sector 6	5621273.125000	3467.00000	24	
5		FOSFERTIL	Sector 2	10122310.500000	2033.00000	29	26
6		MORRÃO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

Atributos disponíveis para serem utilizados nas regras de análise

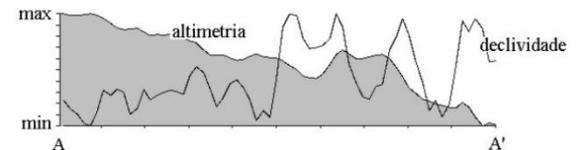
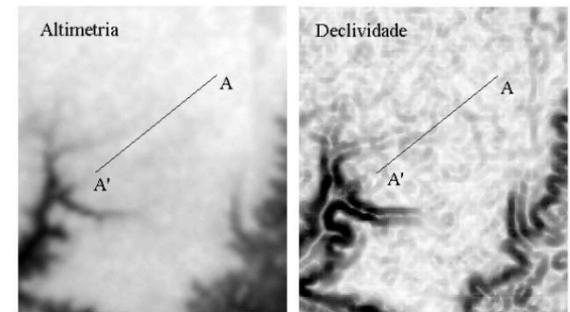
Dados Estáticos

- Mapas matriciais
 - Podem ser utilizados como imagens de fundo no aplicativo de monitoramento Web.
 - Podem ser utilizados em análises (ex: grade de declividade) juntamente com dados dinâmicos.

CBERS-2 CCD, Minas Gerais, Brazil

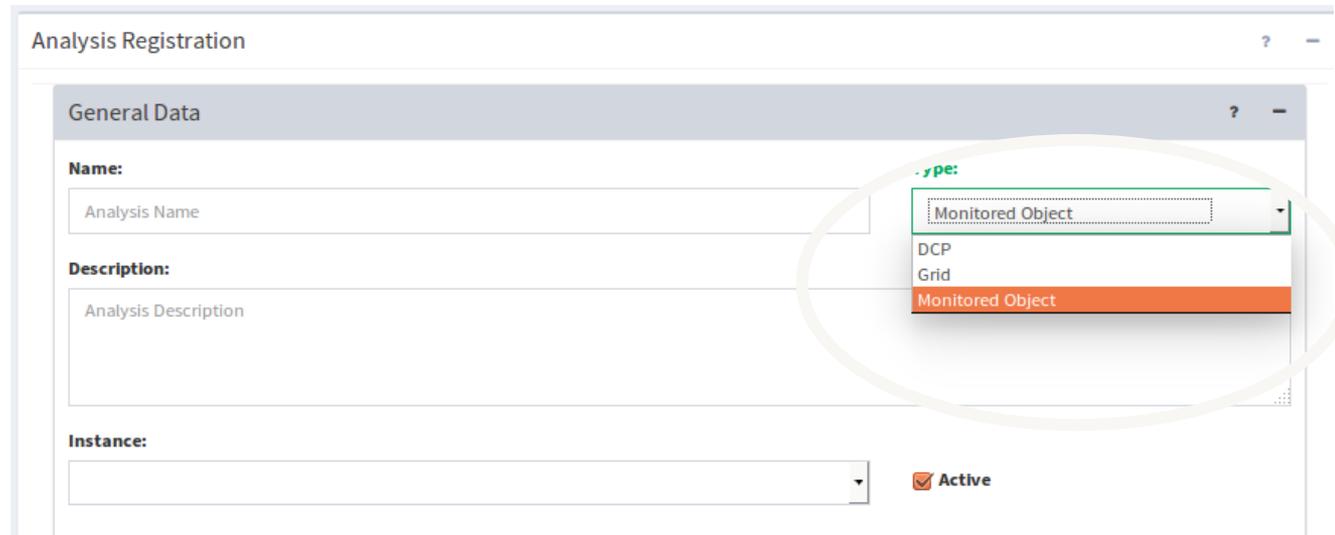


Brasilia: HRC + CCD



Tipos de Análises

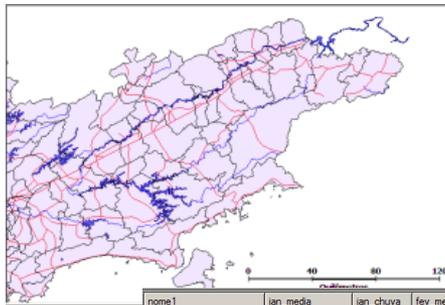
- 1) Baseada em Objetos Monitorados
- 2) Baseada em Grades
- 3) Baseada em PCD



The image shows a screenshot of a web application interface titled "Analysis Registration". The interface is divided into sections. The "General Data" section is highlighted with a grey header. It contains three main input areas: "Name:" with a text box containing "Analysis Name", "Description:" with a text box containing "Analysis Description", and "Instance:" with a dropdown menu. To the right of the "Description:" field, a dropdown menu is open, showing a list of options: "Monitored Object", "DCP", "Grid", and "Monitored Object". The "Monitored Object" option at the bottom of the list is highlighted in orange. A light yellow oval is drawn around the dropdown menu. At the bottom right of the form, there is a checkbox labeled "Active" which is checked.

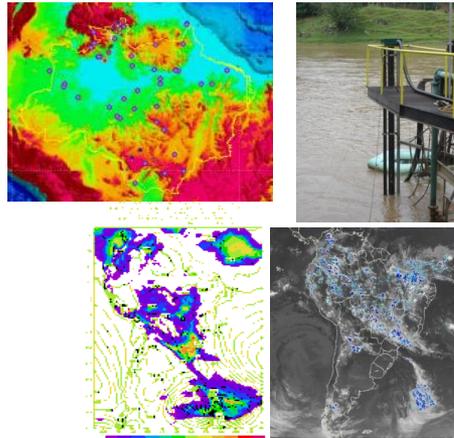
Análise com Objetos Monitorados

Utiliza um mapa estático para cruzar com dados dinâmicos



nome1	jan_media	jan_chuva	fev_media
Sebastianópolis do Sul	25.5	247	25.7
São João das Duas Pontes	25.8	243.7	25.9
Americana	24.5	238.7	24.6
Juquá	27.8	276.8	28.1
Picatu	25.6	195.8	25.8
Platina	25.4	213.3	25.6

+



=

nome1	jan_media	jan_chuva	fev_media
Sebastianópolis do Sul	25.5	247	25.7
São João das Duas Pontes	25.8	243.7	25.9
Americana	24.5	238.7	24.6
Juquá	27.8	276.8	28.1
Picatu	25.6	195.8	25.8
Platina	25.4	213.3	25.6

nome1	jan_media	jan_chuva
Sebastianópolis do Sul	25.5	247
São João das Duas Pontes	25.8	243.7
Americana	24.5	238.7
Juquá	27.8	276.8
Picatu	25.6	195.8
Platina	25.4	213.3

Mapa com áreas a serem monitoradas

Dados Ambientais dinâmicos

Novas colunas com resultados

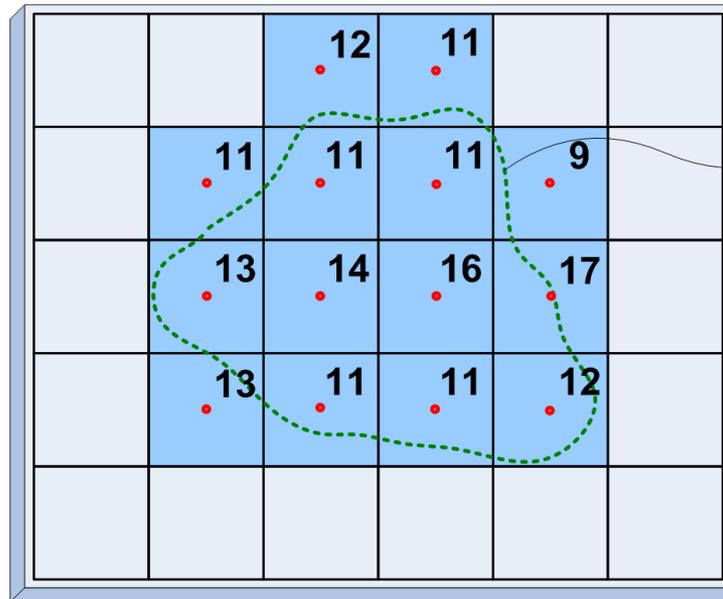
Análises com Objetos Monitorados

Operadores com Grades Numéricas simples

Sintaxe: `grid.zonal.mean("dataSeriesName", buffer)`

Dado Dinâmico Matricial (grade)

Grade atual de observação de umidade relativa de nome "Umin"



Limite do Objeto Monitorado

Exemplo:

```
buf1 = Buffer( )
```

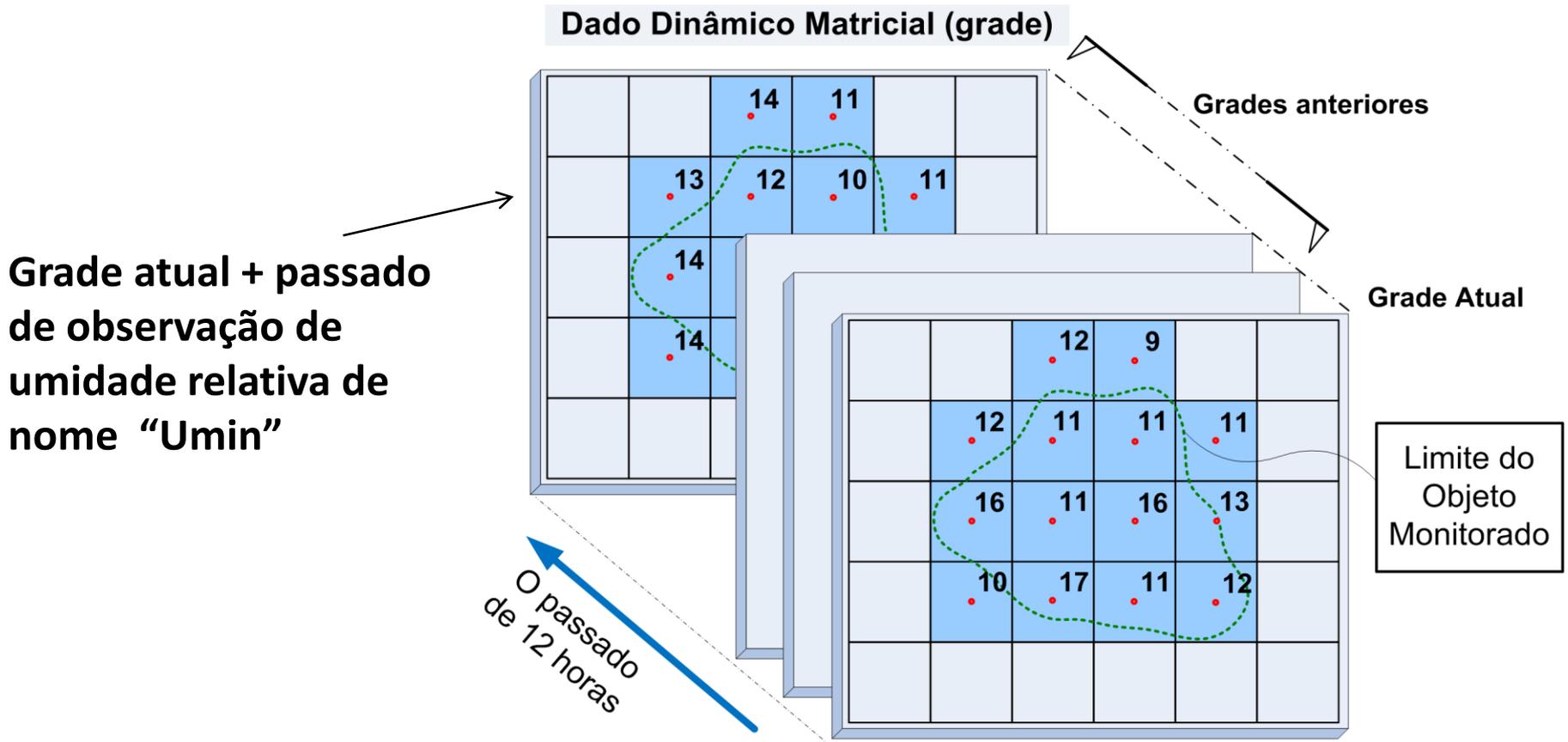
```
x = grid.zonal.mean("Umin", buf1)
```

Resultado: $x = 12,28$

Análises com Objetos Monitorados

Operadores com Grades Numéricas simples

Sintaxe: `grid.zonal.history.<operador>("dataSeriesName", "dateFilter", buffer)`



Exemplo:

```
buf1 = Buffer( )
```

```
x = grid.zonal.history.min("Umin", "12h", buf1)
```

Resultado: $x = \text{mínimo de } ((10 + \dots + 14) \dots (9 + \dots + 11))$

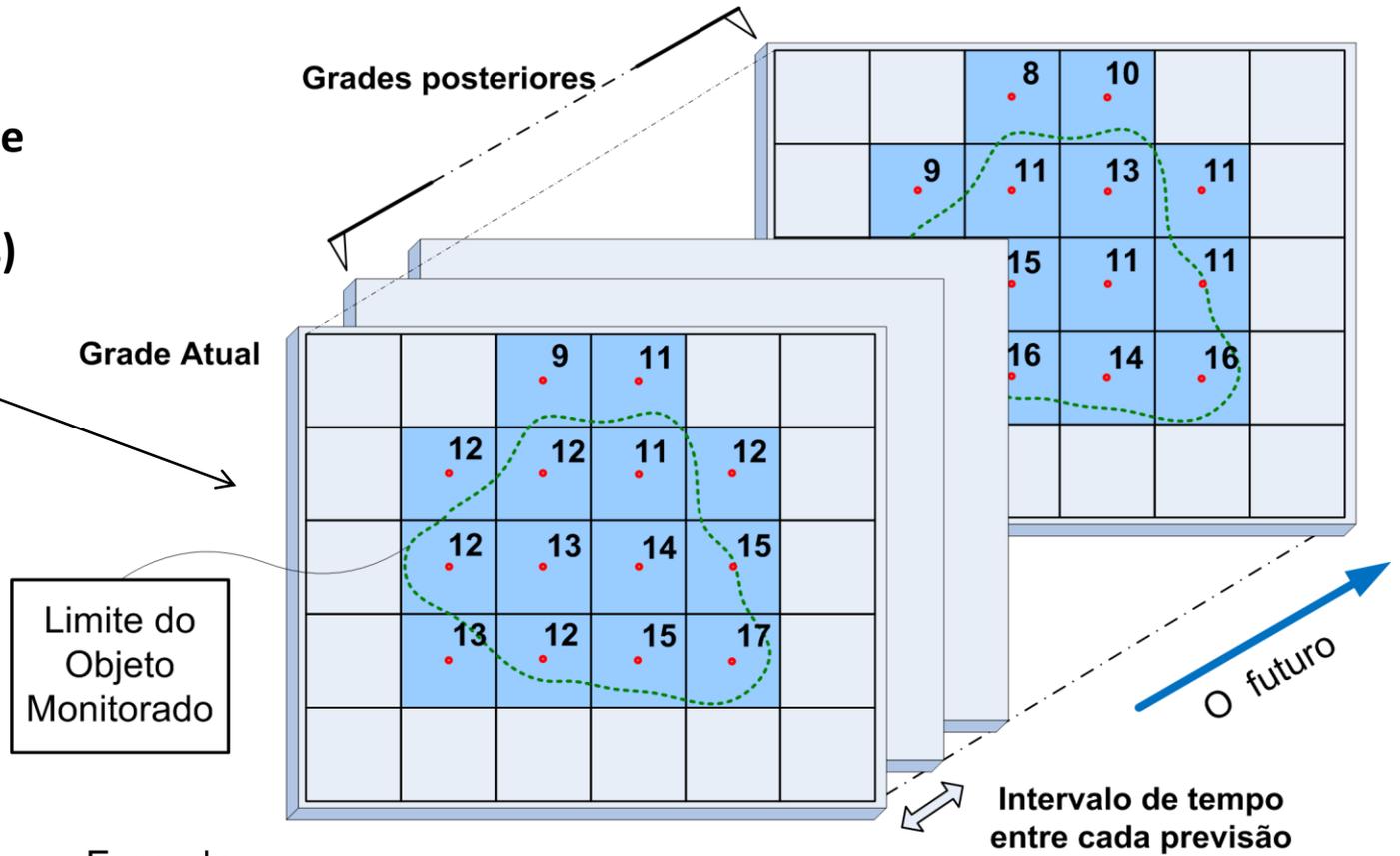
Análises com Objetos Monitorados

Operadores com Grades Numéricas Multidimensional

Sintaxe: `grid.zonal.forecast.<operator>("dataSeriesName", "dateFilter", buffer)`

Dados Dinâmicos Matriciais Multidimensionais (grades)

Grade de previsão de umidade (arquivo com várias camadas) de nome "Umin"



Exemplo:

```
buf1 = Buffer( )
```

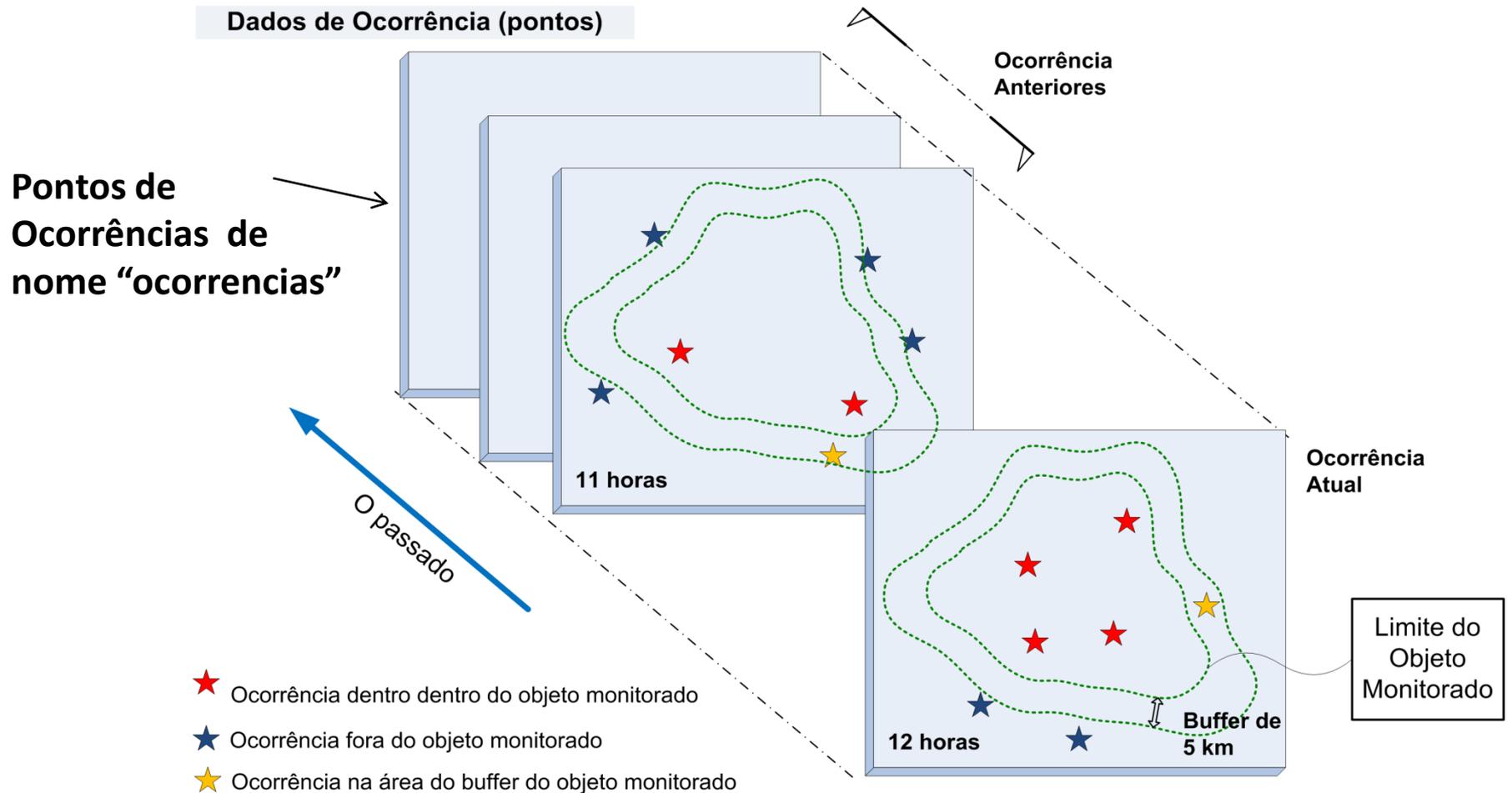
```
x = grid.zonal.forecast.max("Umin", buf1)
```

Resultado: $x = \text{máximo de } ((14 + \dots + 17), \dots, (12 + \dots + 14))$

Análises com Objetos Monitorados

Operadores com Ocorrências

Sintaxe: `occurrence.count("dataSeriesName", buffer, "dateFilter", "restriction")`

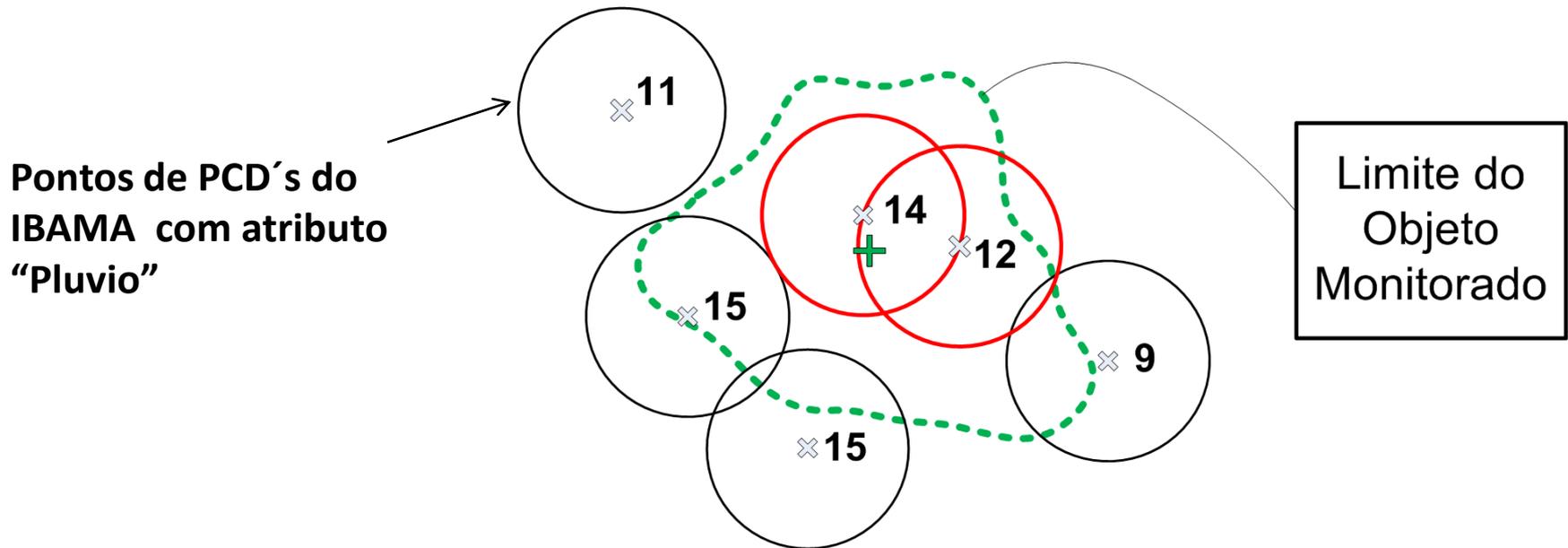


Exemplo: `buf1 = Buffer(Buffer(BufferType.Out, 5, "km"))`
`x = occurrence.count("ocorrencias", buf1, "2h", "UF = 'AM'")`
Resultado: `x = 2` (1 ponto as 11h e 1 ponto as 12h)

Análises com Objetos Monitorados

Operadores com PCD

Sintaxe: `dcp.zonal.<operator>("dataSeriesName", dcpid, "attribute")`



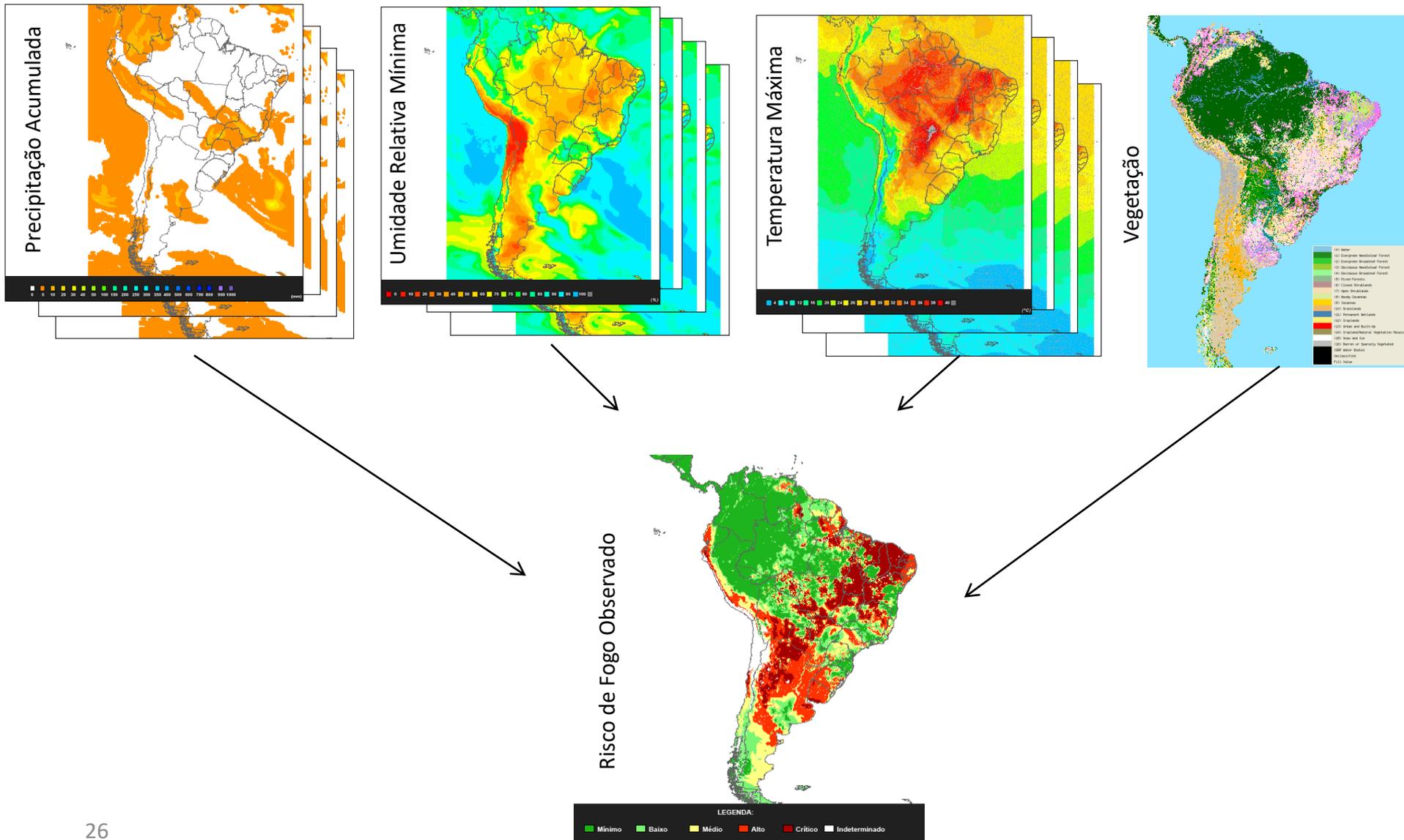
Exemplo:

```
buf1 = Buffer( )  
ids = dcp.influence.by_rule("Serra do Mar", buf1)  
x = dcp.zonal.max("PCD_IBAMA", "unid", ids)
```

Resultado: `var1 = 15` (se regra de influência é TOCA)

`var1 = 14` (se regra de influência é CENTRO)

Cálculo do Risco de Fogo Observado



TerraMA² - Análise baseado em Grades

Cálculo do Risco de Fogo Observado

$$\begin{aligned}
 fp1 &= e^{-0.14*prec}; & fp2 &= e^{-0.07*prec}; & fp3 &= e^{-0.04*prec}; & fp4 &= e^{-0.03*prec}; \\
 fp5 &= e^{-0.02*prec}; & fp6a10 &= e^{-0.01*prec}; & fp11a15 &= e^{-0.008*prec}; \\
 fp16a30 &= e^{-0.004*prec}; & fp31a60 &= e^{-0.002*prec}; & fp61a90 &= e^{-0.001*prec}; \\
 fp91a120 &= e^{-0.0007*prec}.
 \end{aligned}$$

$$PSE = 105 * fp1 * fp2 * fp3 * fp4 * fp5 * fp6a10 * fp11a15 * fp16a30 * fp31a60 * fp61a90 * fp91a120$$

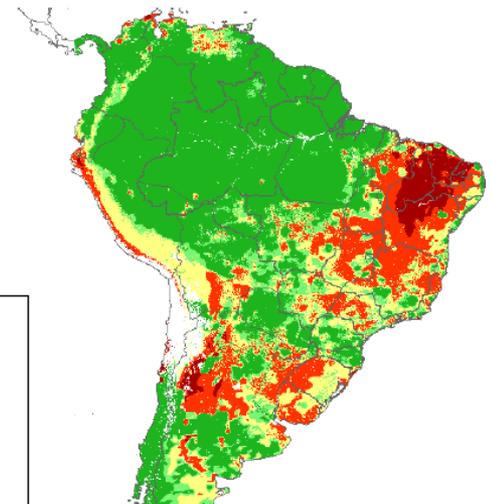
$$Rb_{n=1,7} = \frac{0.9 * [1 + \sin(A_{n=1,7} * PSE)]}{2},$$

$$\text{Fator Umidade} = FU = UR * -0.006 + 1.3$$

$$\text{Fator Temperatura} = FT = Tmax * 0.02 + 0.4$$

$$RF = Rb * FT * FU,$$

RISCO	Valores RF
Mínimo	> 0.15
Baixo	> 0.15 < 0.40
Médio	> 0.40 < 0.70
Alto	> 0.70 < 0.95
Critico	> 0.95



Análise de PCD

Utiliza a localização de PCD's para alertar de alguma anomalia no ponto



nome1	jan_media	jan_chuva	fev_media
Sebastianópolis do Sul	25.5	247	25.7
São João das Duas Pontes	25.8	243.7	25.9
Americana	24.5	238.7	24.6
Juquá	27.8	276.8	28.1
Piçatu	25.6	195.8	25.8
Platina	25.4	213.3	25.6

PCD's no campo

=

nome1	jan_media	jan_chuva	fev_media
Sebastianópolis do Sul	25.5	247	25.7
São João das Duas Pontes	25.8	243.7	25.9
Americana	24.5	238.7	24.6
Juquá	27.8	276.8	28.1
Piçatu	25.6	195.8	25.8
Platina	25.4	213.3	25.6

nome1	jan_media	jan_chuva
Sebastianópolis do Sul	25.5	247
São João das Duas Pontes	25.8	243.7
Americana	24.5	238.7
Juquá	27.8	276.8
Piçatu	25.6	195.8
Platina	25.4	213.3



Novas colunas com resultados

Operadores – TerraMA²

Tipos de Análise

1- Utilitários

2 - Operadores com objeto monitorado

3 - Operadores entre grades

4 – Operadores de PCD

1- Utilitários para as análises

Os seguintes utilitários estão disponíveis

Unidade de distância : unidade utilizada pelo operador “buffer”

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

Unidade de tempo : unidade utilizada pelo operadores históricos

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.object_plus_buffer, 50, "cm")`
- `buffer = Buffer(BufferType.object_plus_buffer, 100, "m")`
- `buffer = Buffer(BufferType.object_plus_buffer, 10, "km")`



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.

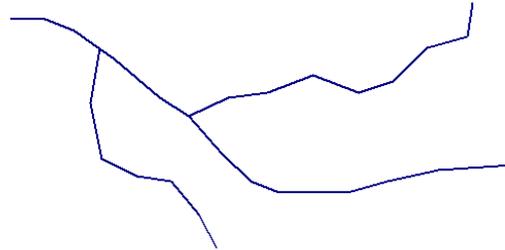
BufferType.Level: A diferença entre a geometria do buffer 1 e a geometria do buffer 2.

1 - Utilitários – Buffer

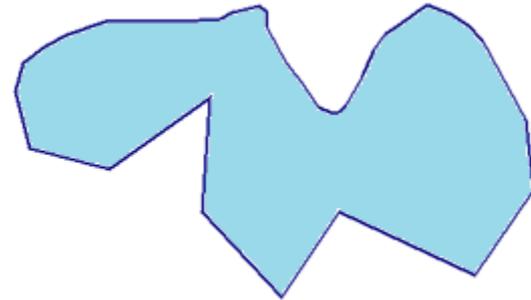
BufferType.None() : Sem buffer



Pont
o



Linha



Área do Polígono

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

ou

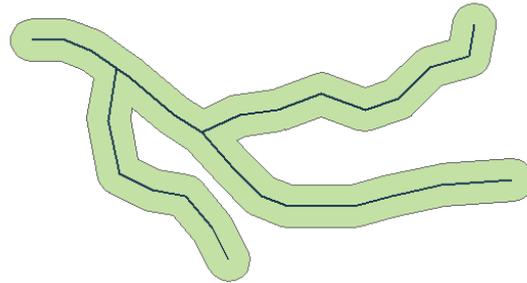
OBS: Considera somente a localização do ponto, linha e área do polígono que
fizer interseção com o dado dinâmico

1 - Utilitários – Buffer

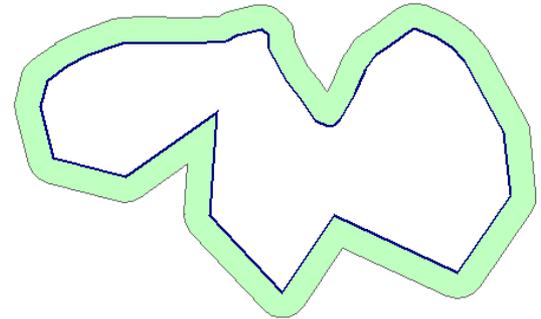
BufferType.Out : Somente a geometria do buffer externo.



Buffer em
volta do
ponto



Buffer em
volta da linha



Buffer externo ao
polígono

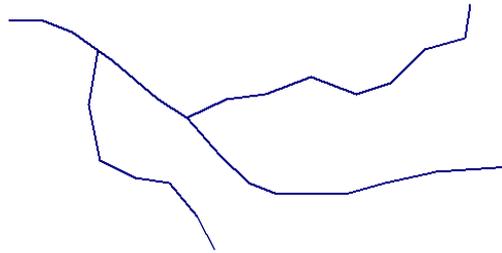
Exemplo: `b1 = Buffer(BufferType.Out, 200, "m")`

1 - Utilitários – Buffer

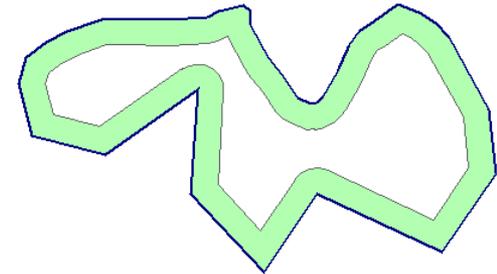
BufferType.In : Somente a geometria do buffer interno.



Ponto



Linha



Buffer interno ao
polígono

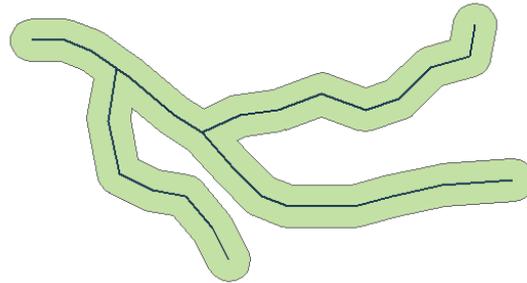
Exemplo: `b1 = Buffer(BufferType.In, 200, "m")`

1 - Utilitários – Buffer

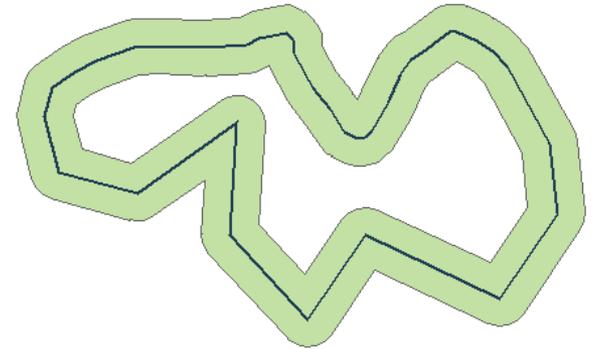
Buffer.Type.In_out : A união da geometria do buffer externo com a geometria do buffer interno.



Buffer em
volta do
ponto



Buffer em
volta da linha



Buffer interno e
externo ao
polígono

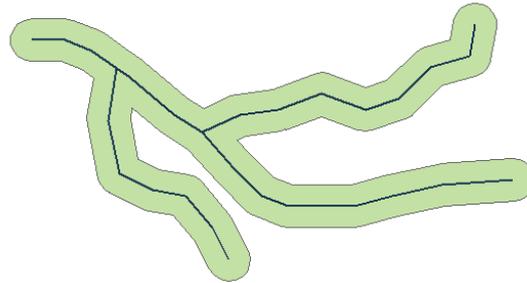
Exemplo: `b1 = Buffer(BufferType.In_out, 200, "m", 200, "m")`

1 - Utilitários – Buffer

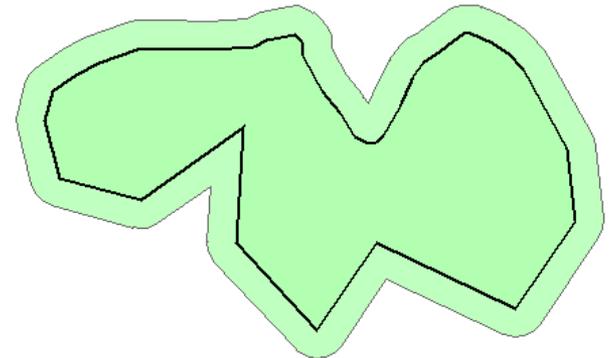
BufferType.Out_union : Interior da geometria mais a geometria do buffer externo



Buffer em
volta do
ponto



Buffer em
volta da linha



Buffer externo
mais área do
polígono

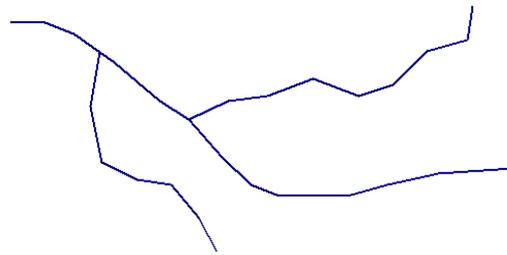
Exemplo: `b1 = Buffer(BufferType.Out_union, 200, "m")`

1 - Utilitários – Buffer

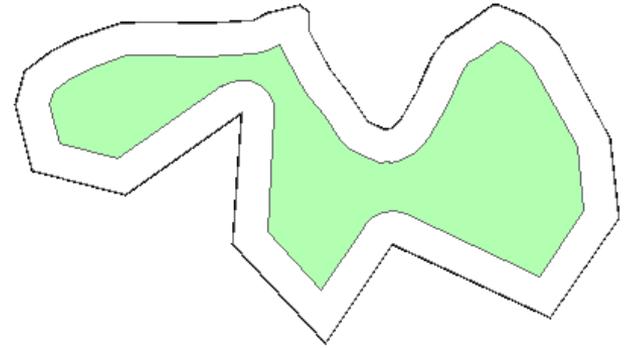
BufferType.In_diff : Interior da geometria menos a geometria do buffer interno.



Ponto



Linha

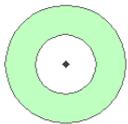


Área do polígono
menos buffer
interno

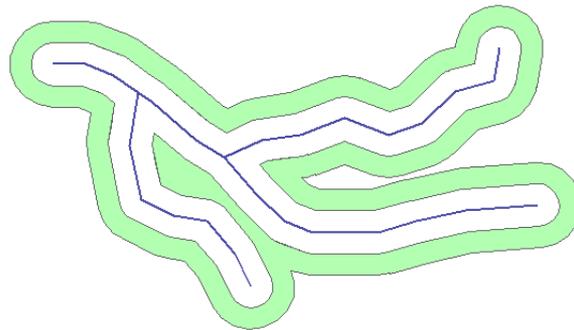
Exemplo: `b1 = Buffer(BufferType.In_diff, 200, "m")`

1 - Utilitários – Buffer

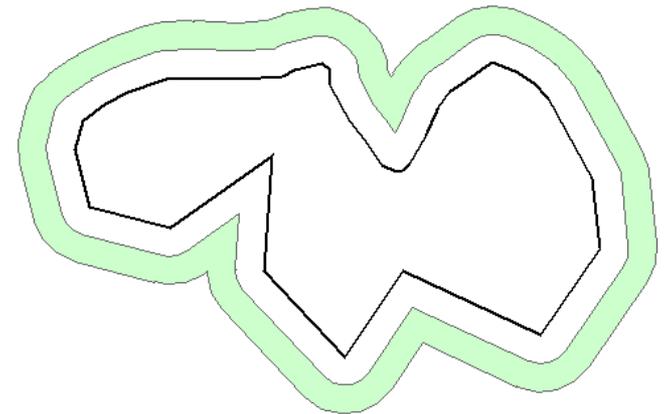
BufferType.Level: A diferença entre a geometria do buffer 1 e a geometria do buffer 2 (distancia de 1 maior que 2)



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 - 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

- `x = occurrence.count("ocorrencias", Buffer(), "30sec", "")`
- `x = occurrence.count("ocorrencias", Buffer(), "10min", "")`
- `x = occurrence.count("ocorrencias", Buffer(), "1h", "")`
- `x = occurrence.count("ocorrencias", Buffer(), "2d", "")`
- `x = occurrence.count("ocorrencias", Buffer(), "2w", "")`

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 : Operador estatísticos para agregação

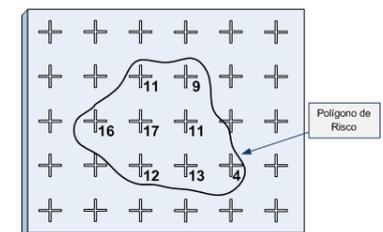
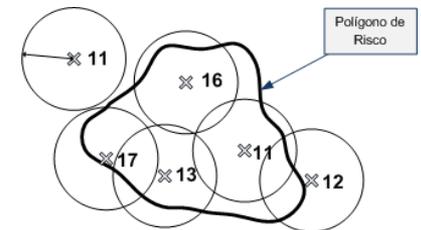
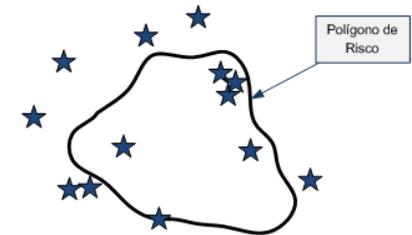
- `Statistic.min` : valor mínimo de uma lista de valores
- `Statistic.max` : valor máximo de uma lista de valores
- `Statistic.mean` : valor médio de uma lista de valores
- `Statistic.sum` : soma de uma lista de valores
- `Statistic.mean` : média de uma lista de valores
- `Statistic.standard_deviation`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

2- Análise baseada em Objetos Monitorados

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

Tipos de operadores:

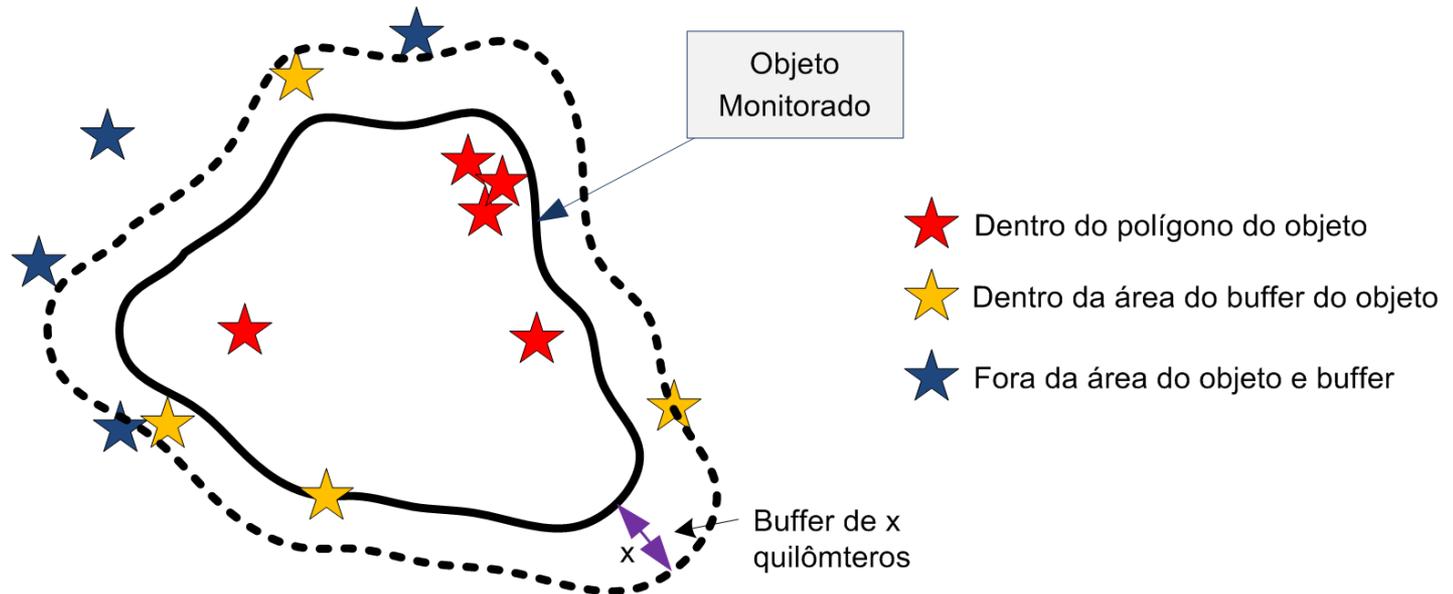
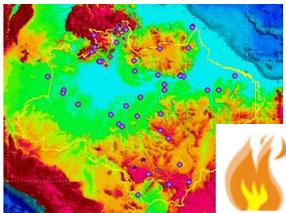
- I. [Operadores de ocorrência](#)
- II. [Operadores de agregação de ocorrências](#)
- III. [Operadores de PCDs](#)
- IV. [Operadores de histórico de PCDs](#)
- V. [Operadores zonais sobre grades](#)
- VI. [Operadores zonais histórico de grades](#)
- VII. [Operadores zonais sobre previsões](#)



2.1 - Operadores de ocorrência

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

Consideram os pontos num intervalo de tempo passado a partir da data/hora atual dentro ou a uma área de influência (buffer) do objeto monitorado.



2.1 - Operadores de ocorrência

Conjunto de operadores para série de dados do tipo **Ocorrência**

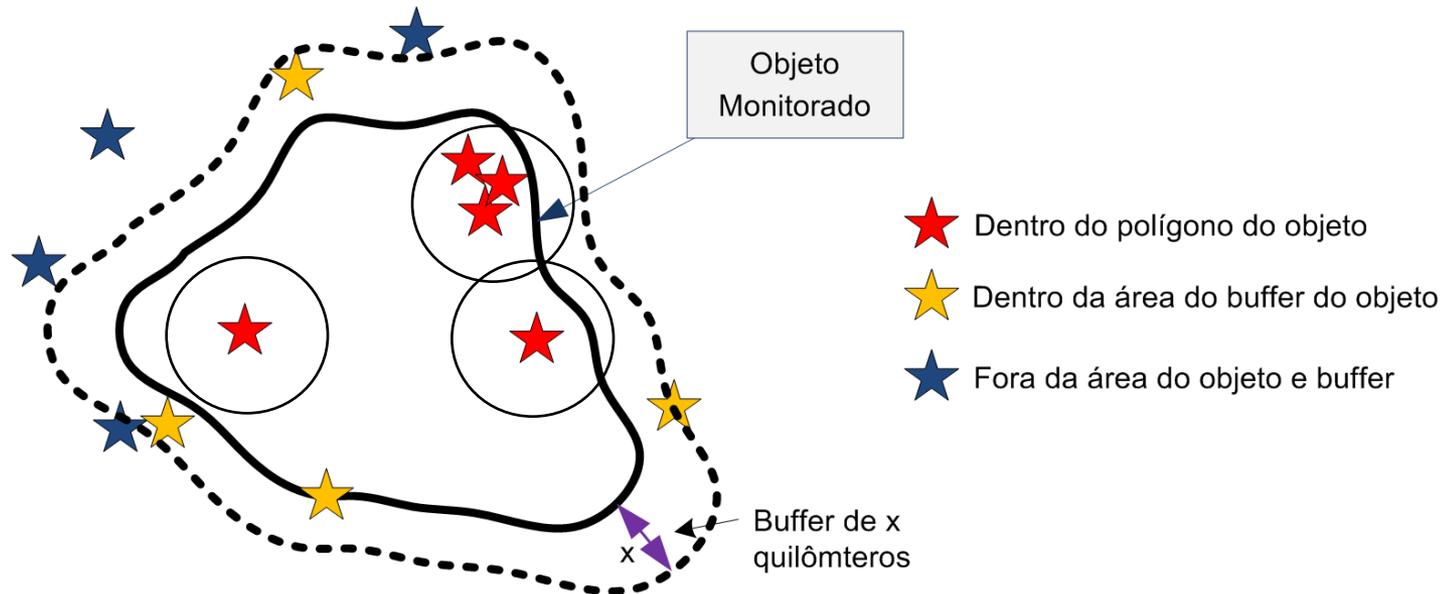
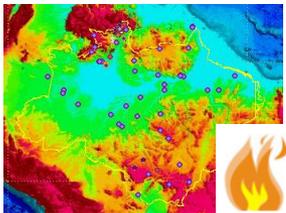
occurrence.<operator>("dataSeriesName", "attribute", "dateFilter", buffer, "restriction") onde:

- Operator : **count, min, max, mean, sum, median, standard_deviation, variância**
- dataSeriesName : String com o nome da série de dados de 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"
- dateFilter : String com o intervalo de tempo para filtrar as ocorrências. Ver utilitário Unidades de tempo
- buffer : Objeto Buffer para ser aplicado ao objeto monitorado. **Não obrigatório**. Ver utilitário Buffer.
- restriction : String com a restrição SQL a ser aplicada. **Não obrigatório**. Não utilizar se não houver restrição.

2.II - Operadores de agregação de ocorrência

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

Consideram os pontos num intervalo de tempo passado a partir da data/hora atual dentro ou a uma área de influência (buffer) do objeto monitorado, porém agrega pontos dentro da área de influência dos pontos



2.II - Operadores de agregação de ocorrência

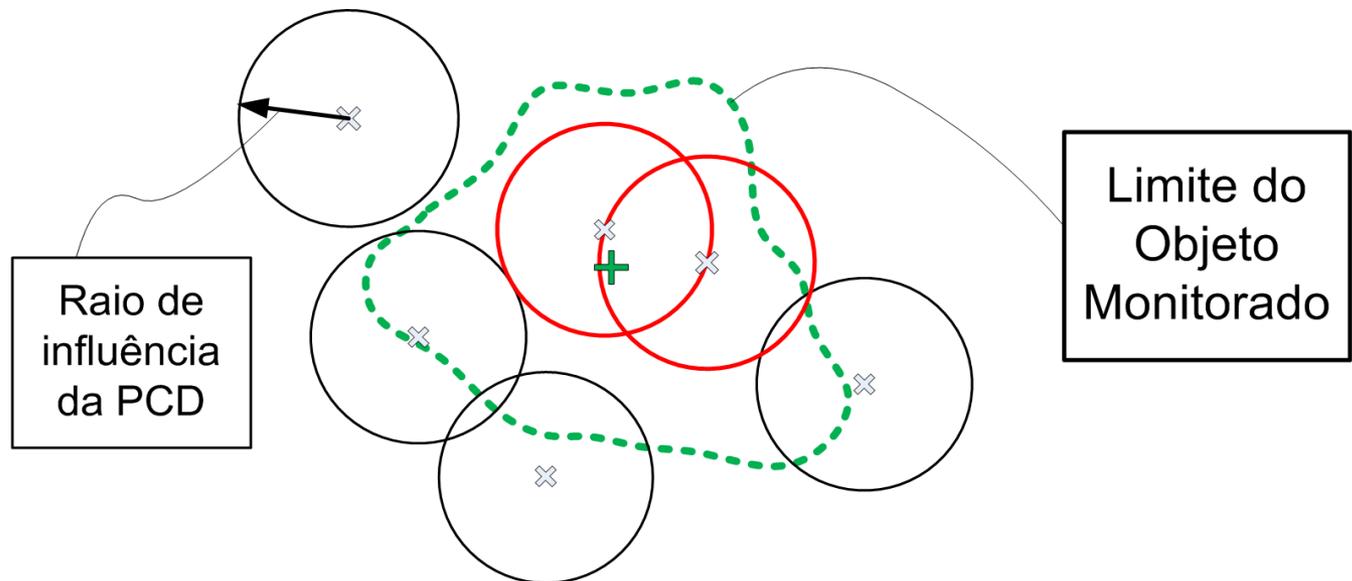
Conjunto de operadores para série de dados do tipo **Ocorrência**

occurrence.aggregation.<operator>("dataSeriesName", "dateFilter",
"attribute", aggregationStatistic, aggregationBuffer, buffer, "restriction")
onde:

- operator : **count, min, max, mean, sum, median, standard_deviation, variance**
- dataSeriesName : String com o nome da série de dados de 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"
- dateFilter : String com o intervalo de tempo para filtrar as ocorrências. Ver utilitário Unidades de tempo
- 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 : Objeto Buffer para ser aplicado ao objeto monitorado. **Não obrigatório**. Ver utilitário Buffer
- restriction : String com a restrição SQL a ser aplicada. **Não Obrigatório**. Não utilizar se não houver restrição.

2.III - Operadores de PCDs

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



- + - Centro de massa do objeto (polígono) monitorado
- - Área de influência da PCD **toca** área do objeto
- - Área de influência da PCD envolve o **centro** de massa do objeto além de **tocar** área do mesmo

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

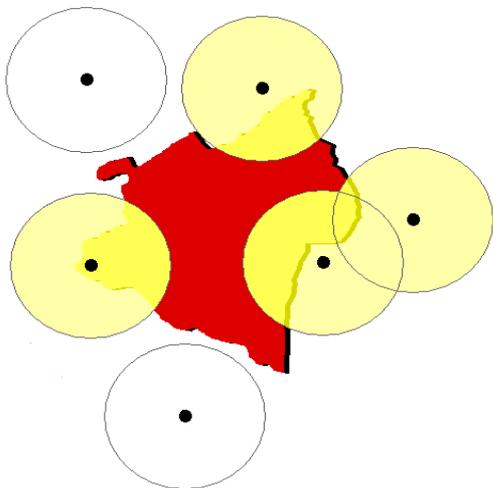
```
dcp.influence.by_attribute("dataSeriesName", attributeList)
```

```
dcp.influence.by_rule("dataSeriesName", buffer) onde:
```

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

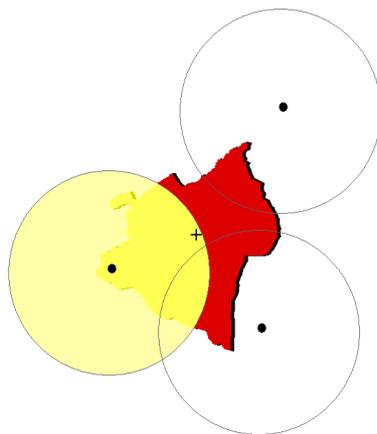
Influência PCD - (Regra de Influência)

Raio (toca)



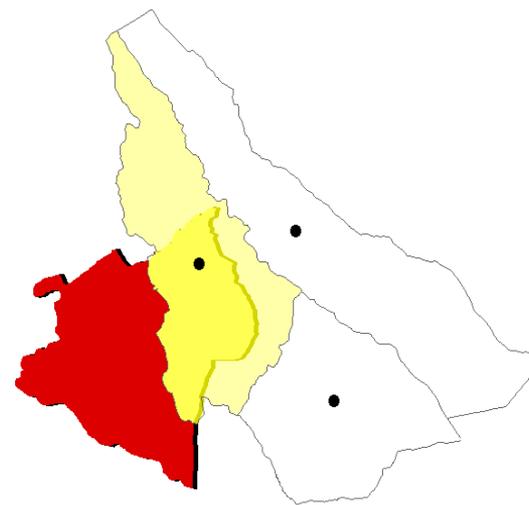
círculo de influência
intersecta o
polígono

Raio (centro)



círculo de influência
precisa conter o
centróide do
polígono

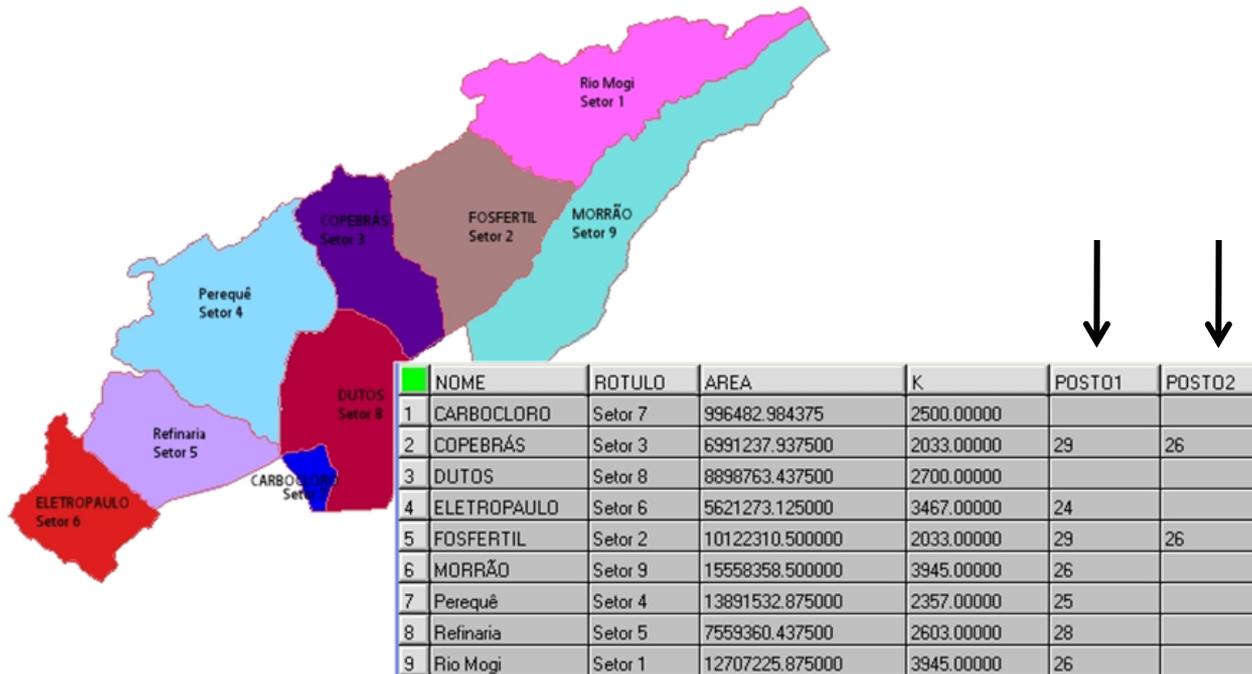
Região



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

Influência PCD - (Atributo do Objeto)

Definido pelo atributo do Objeto



um ou mais atributos do objeto monitorado (dado estático) que especifica quais PCD's devem ser consideradas.

2.III - Operadores de PCDs

Conjunto de operadores para série de dados do tipo PCD

dcp.**zonal**.<operator>("dataSeriesName", "attribute", datasetIds) onde:

- operator : **count, min, max, mean, sum, median, standard_deviation, variance**
- dataSeriesName : 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). Não usar para operador "count".
- datasetIds : lista contendo a identificação das PCD's que influenciam o objeto monitorado. Ver operador "Influência PCD". Não usar se operador zonal for "count", no caso usar operador Buffer se desejar.

2.IV - Operadores de histórico de PCDs

Conjunto de operadores de histórico para série de dados do tipo PCD

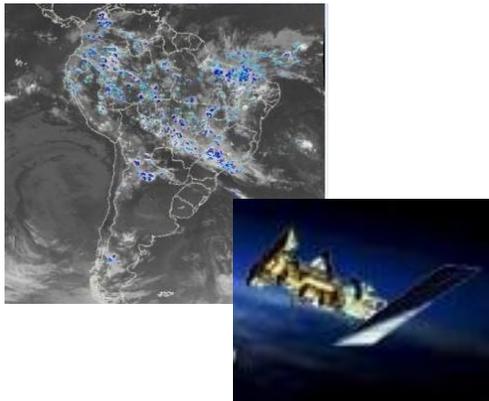
dcp.**zonal**.history.<operator>("dataSeriesName", "attribute",
"dateFilter", datasetIds) onde:

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

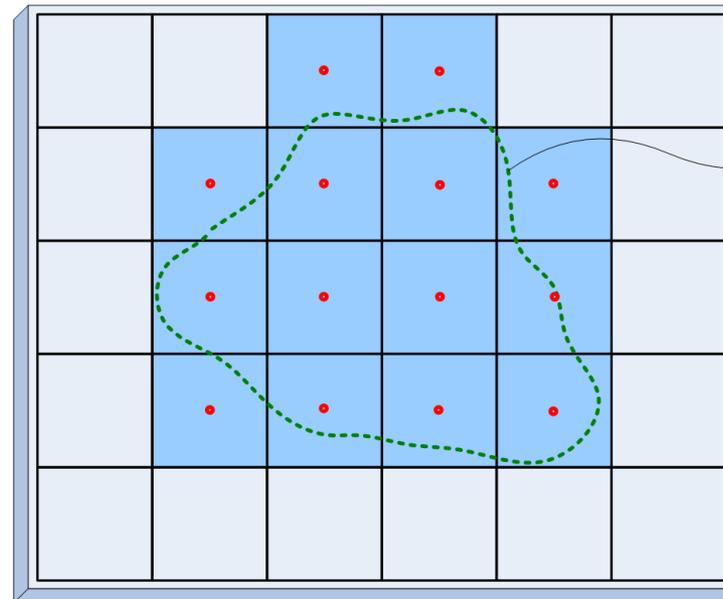
- dataSeriesName : String com o nome da série de dados de PCD
- attribute: String com o nome do atributo da PCD que deve ser utilizado para recuperar os valores. O atributo deve ser do tipo numérico (Ex. Integer, Float, Double, Long).
- dateFilter : String com o intervalo de tempo para filtrar as ocorrências. Ver utilitário Unidades de tempo
- datasetIds : lista contendo a identificação das PCD's que influenciam o objeto monitorado. Ver operador "Influência PCD".

2.V - Operadores zonais sobre grades

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. Pode ser por exemplo um arquivo com **uma única camada** (satélite meteorológico)



Dado Matricial (grade)



Limite do Objeto Monitorado

- Valor da grade não avaliado pelo operador zonal
- Valor da grade será computado pelo operador zonal
- Representação do ponto central de elemento da grade

2.V - Operadores zonais sobre grades

Conjunto de operadores zonais para série de dados do tipo **Grade Retangular**

`grid.zonal.<operator>("dataSeriesName", buffer)` onde:

- Operator : **count, min, max, mean, sum, median, standard_deviation, variance**
- `dataSeriesName` : String com o nome da série de dados de grades
- `buffer` : Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.VI - Operadores zonais histórico de grades

Operadores históricos para grades atuam somente sobre as fontes de dados matricial (grades, raster) para agrupar valores coletados em um determinado período de tempo.

Através desses operadores pode-se, por exemplo, obter a chuva acumulada em um período somando as informações de radar ou do hidroestimador.

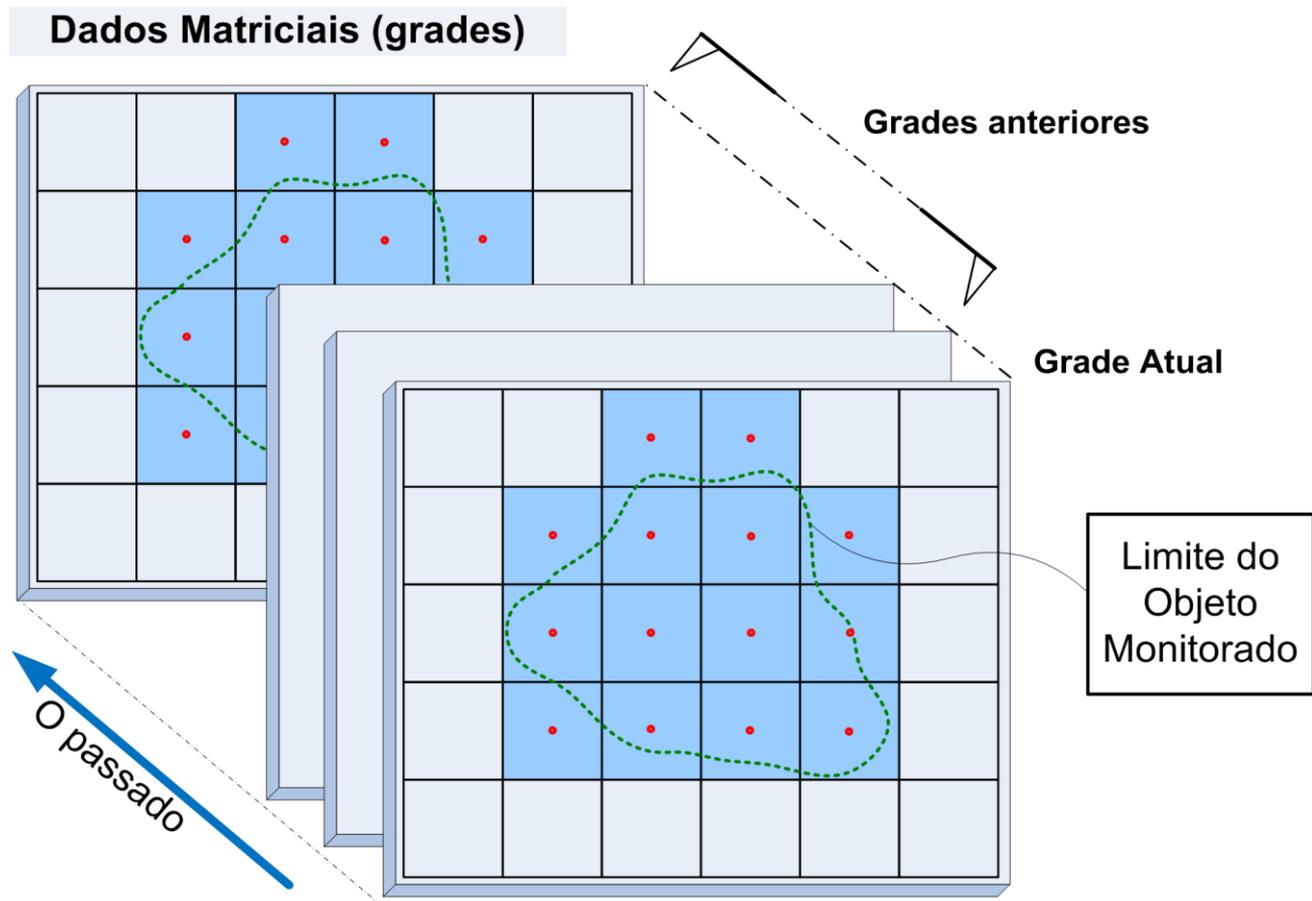
Retornam valores que fazem interseção dos pontos de um conjunto de grades com o objeto monitorado ou sua área de influência.

O cálculo é realizado sempre que o serviço de coleta obtém uma nova grade mais as grade anteriores.

Tipos: **Histórico apenas** e **Precipitação histórica (em mm/h)**

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

2.VI - Operadores zonais histórico de grades



2.VI - Op. zonal histórico p/ grades - Histórico

Conjunto de operadores para série de dados do tipo **Grade Retangular**

`grid.zonal.history.<operator>("dataSeriesName", "dateFilter", buffer)` onde:

- **Operator** : **num, list, min, max, mean, sum, median, standard_deviation, variance**
- **dataSeriesName** : String com o nome da série de dados de grades
- **dateFilter** : String com o intervalo de tempo para filtrar as ocorrências. Ver utilitário Unidades de tempo
- **buffer** : Objeto Buffer para ser aplicado ao objeto monitorado. Ver utilitário Buffer. Não obrigatório.

2.VI - Op. zonal histórico p/ grades - PRECIPITAÇÃO

Conjunto de operadores para série de dados do tipo **Grade Retangular** com umidade em mm/h.

terra2.grid.zonal.prec.<operator>("dataSeriesName", "dateFilter", buffer)
onde:

- Operator : **min, max, mean, sum, median, standard_deviation, variance**
- dataSeriesName : String com o nome da série de dados de grades
- dateFilter : String com o intervalo de tempo para filtrar as ocorrências.
Ver utilitário Unidades de tempo
- buffer : Objeto Buffer para ser aplicado ao objeto monitorado.
Ver utilitário Buffer

2.VII - Operadores zonais sobre previsões

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.).

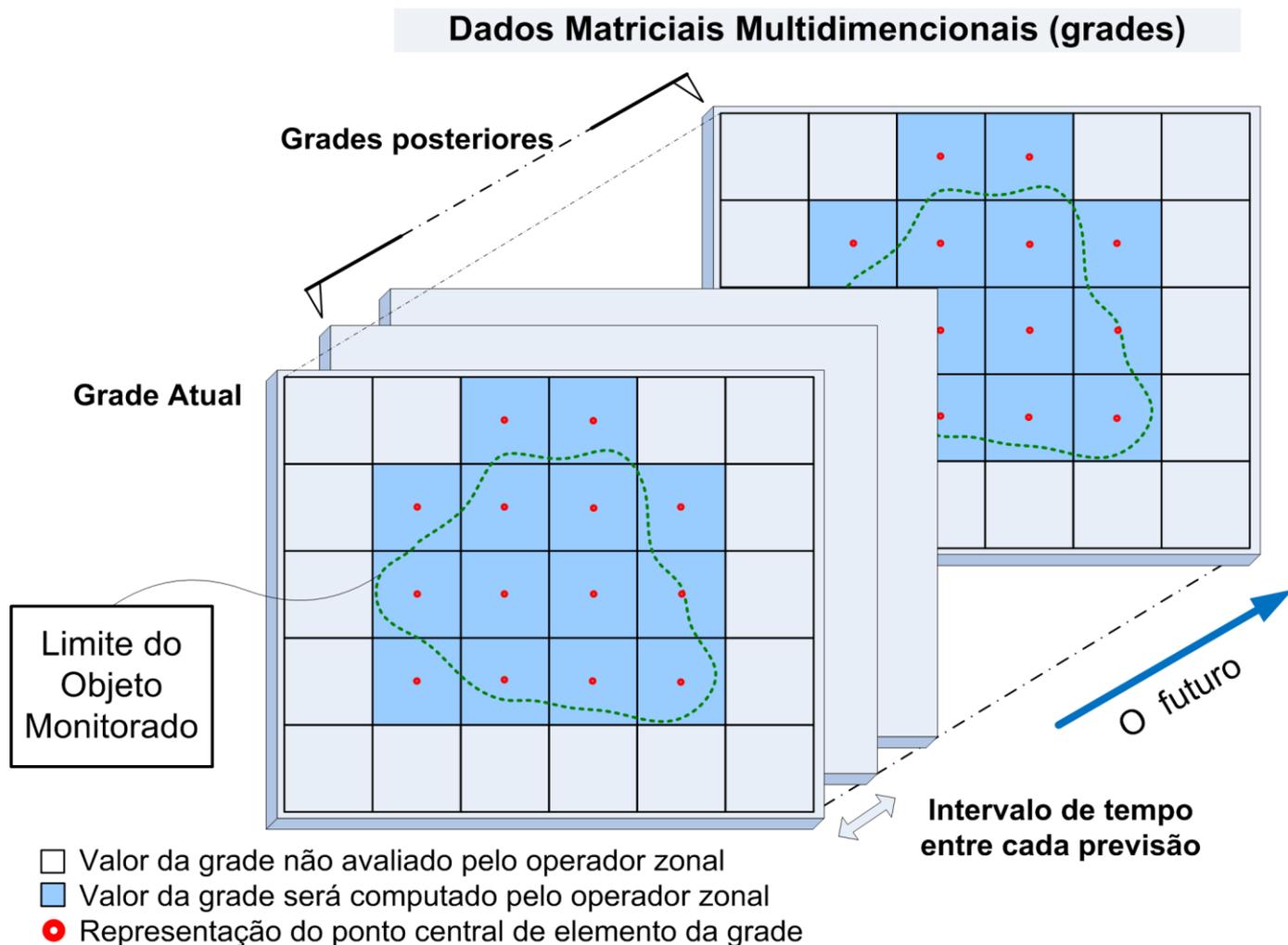
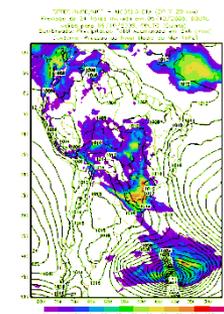
Cada modelo de previsão tem seu intervalo de tempo definido entre cada camada.

As análises são feitas sobre as camadas a partir do horário atual para o frente (o 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.VII - Operadores zonais sobre previsões



2.VII - Operadores zonais sobre previsões

Conjunto de operadores para série de dados do tipo **Grade Retangular Multidimensional (n camadas de previsão)**

grid.zonal.forecast.<operator>("dataSeriesName", "dateFilter", buffer) onde:

- Operator : **num, count, min, max, mean, sum, median, standard_deviation**
- dataSeriesName : String com o nome da série de dados de grades de previsão
- dateFilter : String com o intervalo de tempo para filtrar as ocorrências.
Ver utilitário Unidades de tempo
- buffer : Objeto Buffer para ser aplicado ao objeto monitorado.
Ver utilitário Buffer

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](#)

Dado Matricial (grade)

12	12	12	11	11	14
12	11	11	11	9	9
12	13	14	16	17	16
12	13	11	11	12	14
17	17	13	12	16	13

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

3.1- 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

3.1 - 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")
```

3.1 - 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_deviation("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")`

IMPLEMENTAR

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")`

IMPLEMENTAR

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](#)

IMPLEMENTAR

4.1 [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")

IMPLEMENTAR

4.11 [Operadores Históricos de PCD's](#)

Retorna (COUNT, MIN, MAX ,SOMA,MEDIA, etc) de cada PCD num intervalo de tempo a partir 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")`

