

# Documentação do módulo de scripts vetoriais do TerraAmazon

Responsável: Lucas Ramos

## Objetivo

Apresentar a documentação das operações disponíveis em Lua

### Operação Dissolve

```
dissolve(  
<NOME_LAYER_ENTRADA>,  
<RESTRIÇÃO>,  
<COLUNAS>,  
<MINIMIZAR_USO_MEMORIA>,  
<NOME_LAYER_SAÍDA>)
```

Parâmetros de entrada:

NOME\_LAYER\_ENTRADA: "Nome do layer de entrada"

RESTRIÇÃO : "Restrição a ser aplicada ao layer, mesmo formato da restrição dos temas (SQL)"

COLUNAS : "Lista com os nomes das colunas que serão consideradas no agrupamento  
Ex. { "nome1", "nome2" }"

MINIMIZAR\_USO\_MEMORIA: " Minimizar uso de memória: true OU false"

NOME\_LAYER\_SAÍDA : "Nome do layer de saída"

### Exemplo de uso:

```
layerIn = "EstadosBrasil"  
restriction = ""  
columns = {"regiao"}  
layerOut = "EstadosPorRegiao"  
dissolve(layerIn , restriction, columns, true, layerOut)
```

A Atribuição para variáveis é opcional, isto é uma boa prática na programação e

facilita a visualização.

O mesmo código poderia ser escrito assim:

```
dissolve ( "EstadosBrasil" , "", {"regiao"}, false, "EstadosPorRegiao")
```

## Operação Intersection

```
intersection(  
<NOME_LAYER_ENTRADA>,  
<RESTRIÇÃO_LAYER_ENTRADA>,  
<COLUNAS_LAYER_ENTRADA>,  
<COLUNA_CÉLULA_LAYER_ENTRADA>,  
<NOME_LAYER_INTERSECÇÃO>,  
<RESTRIÇÃO_LAYER_INTERSECÇÃO>,  
<COLUNAS_LAYER_INTERSECÇÃO>,  
<COLUNA_CÉLULA_LAYER_INTERSECÇÃO>,  
<NOME_LAYER_SAÍDA>))
```

Parâmetros de entrada:

NOME\_LAYER\_ENTRADA: " Nome do layer de entrada"

RESTRIÇÃO\_LAYER\_ENTRADA : "Restrição a ser aplicada ao layer de entrada, mesmo formato da restrição dos temas (SQL)"

COLUNAS\_LAYER\_ENTRADA : "Lista com os nomes das colunas que serão consideradas do layer de entrada Ex. { "coluna1", "coluna2" } "

COLUNA\_CÉLULA\_LAYER\_ENTRADA: "Coluna de célula do layer de entrada"

NOME\_LAYER\_INTERSECÇÃO: "Nome do layer de interseção"

RESTRIÇÃO\_LAYER\_INTERSECÇÃO : "Restrição a ser aplicada ao layer de interseção, mesmo formato da restrição dos temas (SQL)"

COLUNAS\_LAYER\_INTERSECÇÃO : "Lista com os nomes das colunas que serão consideradas do layer de interseção Ex. { "coluna1", "coluna2" } "

COLUNA\_CÉLULA\_LAYER\_INTERSECÇÃO: "Coluna de célula do layer de interseção"

NOME\_LAYER\_SAÍDA : "Nome do layer de saída"

## Exemplo de uso:

```
layerIn = "DADO_1"  
restrictionIn = ""  
columnsFromLayerIn = {"classe", "objeto"}
```

```

cellColumnIn = "cell_oid"
layerInter = "CELL"
restrictionInter = ""
columnsFromInter = {}
cellColumnInter = "cell_oid"
layerOut = "DADO_1_CELL_LUA"
intersection( layerIn, restrictionIn, columnsFromLayerIn, cellColumnIn, layer

```

## Operação Identity

```

identity(
<NOME_LAYER_ENTRADA>,
<COLUNAS_LAYER_ENTRADA>,
<COLUNA_CÉLULA_LAYER_ENTRADA>,
<NOME_LAYER_INTERSECÇÃO>,
<COLUNAS_LAYER_INTERSECÇÃO>,
<COLUNA_CÉLULA_LAYER_INTERSECÇÃO>,
<NOME_LAYER_SAÍDA>)

```

Parâmetros de entrada:

NOME\_LAYER\_ENTRADA: "Nome do layer de entrada"

COLUNAS\_LAYER\_ENTRADA : "Lista com os nomes das colunas que serão consideradas do layer de entrada Ex. { "coluna1", "coluna2" } "

COLUNA\_CÉLULA\_LAYER\_ENTRADA: "Coluna de célula do layer de entrada"

NOME\_LAYER\_INTERSECÇÃO: "Nome do layer de interseção"

COLUNAS\_LAYER\_INTERSECÇÃO : "Lista com os nomes das colunas que serão consideradas do layer de interseção Ex. { "coluna1", "coluna2" } "

COLUNA\_CÉLULA\_LAYER\_INTERSECÇÃO: "Coluna de célula do layer de interseção"

NOME\_LAYER\_SAÍDA : "Nome do layer de saída"

## Exemplo de uso:

```

layerIn = "DADO_1_CELL"
columnsFromLayerIn = {"classe", "objeto"}
cellColumnIn = "cell_oid"
layerInter = "DADO_2_CELL"
columnsFromInter = {}
cellColumnInter = "cell_oid"
layerOut = "IDENTITY_LUA"
identity( layerIn, columnsFromLayerIn, cellColumnIn, layerInter, columnsFromI

```

## Operação Union By Touch

```
unionByTouch(  
<NOME_LAYER_A>, ,  
<RESTRIÇÃO_LAYER_A>,  
<COLUNAS_LAYER_A>  
<COLUNA_CÉLULA_LAYER_A>,  
<NOME_LAYER_B>,  
<RESTRIÇÃO_LAYER_B>,  
<COLUNA_CÉLULA_LAYER_B>,  
<NOME_LAYER_SAÍDA>)
```

Parâmetros de entrada:

NOME\_LAYER\_A: "Nome do layer de entrada"

RESTRIÇÃO\_LAYER\_A: "Restrição a ser aplicada ao layer de entrada, mesmo formato da restrição dos temas (SQL)"

COLUNAS\_LAYER\_A: "Lista com os nomes das colunas que serão consideradas do layer de entrada Ex. { "coluna1", "coluna2" }"

COLUNA\_CÉLULA\_LAYER\_A: "Coluna de célula do layer de entrada"

NOME\_LAYER\_B: "Nome do layer B"

RESTRIÇÃO\_LAYER\_B: "Restrição a ser aplicada ao layer B, mesmo formato da restrição dos temas (SQL)"

COLUNA\_CÉLULA\_LAYER\_B: "Coluna de célula do layer B"

NOME\_LAYER\_SAÍDA: "Nome do layer de saída"

## Exemplo de uso:

```
layerA = "DADO_1_CELL"  
restricaoA = ""  
colunasA = {"classe", "objeto"}  
celulaA = "cell_oid"  
layerB = "DADO_2_CELL"  
restricaoB = ""  
celulaB = "cell_oid"  
layerSaida = "UNION_AB"  
unionByTouch ( layerA , restricaoA, colunasA, celulaA, layerB , restricaoB, c
```

## Operação Difference

```
difference(  
<NOME_LAYER_ENTRADA>,  
<COLUNAS_LAYER_ENTRADA>,  
<RESTRIÇÃO_LAYER_ENTRADA>,  
<COLUNA_CÉLULA_LAYER_ENTRADA>,  
<NOME_LAYER_DIFERENÇA>,  
<RESTRIÇÃO_LAYER_DIFERENÇA>,  
<COLUNA_CÉLULA_LAYER_DIFERENÇA>,  
<NOME_LAYER_SAÍDA>)
```

Parâmetros de entrada:

NOME\_LAYER\_ENTRADA: "Nome do layer de entrada"

RESTRIÇÃO\_LAYER\_ENTRADA: "Restrição a ser aplicada ao layer de entrada, mesmo formato da restrição dos temas (SQL)"

COLUNAS\_LAYER\_ENTRADA: "Lista com os nomes das colunas que serão consideradas do layer de entrada Ex. { "coluna1", "coluna2" } "

COLUNA\_CÉLULA\_LAYER\_ENTRADA: "Coluna de célula do layer de entrada"

NOME\_LAYER\_DIFERENÇA: "Nome do layer de diferença"

RESTRIÇÃO\_LAYER\_DIFERENÇA: "Restrição a ser aplicada ao layer de diferença, mesmo formato da restrição dos temas (SQL)"

COLUNA\_CÉLULA\_LAYER\_DIFERENÇA: "Coluna de célula do layer de diferença"

NOME\_LAYER\_SAÍDA: "Nome do layer de saída"

## Exemplo de uso:

```
layerIn = "DADO_1"  
restrictionIn = ""  
columnsFromLayerIn = {"classe", "objeto"}  
cellColumnIn = "cell_oid"  
layerDiff = "DADO_2"  
restrictionDiff = ""  
cellColumnDiff = "cell_oid"  
layerOut = "DIFFERENCE_LUA"  
difference( layerIn, restrictionIn, columnsFromLayerIn, cellColumnIn, layerDi
```

## Operação Merge

```
merge(  
<NOME_LAYER_ENTRADA>,  
<RESTRIÇÃO_LAYER_ENTRADA>,  
<MANTER_OBJECTID>,  
<MAP_COLUNAS>,  
<NOME_LAYER_SAÍDA>)
```

Parâmetros de entrada:

NOME\_LAYER\_ENTRADA: "Nome do layer de entrada"

RESTRIÇÃO\_LAYER\_ENTRADA: "Restrição a ser aplicada ao layer de entrada, mesmo formato da restrição dos temas (SQL)"

MANTER\_OBJECTID: "Manter o objectid do layer de entrada? true OU false"

MAP\_COLUNAS: "Mapa de chave/valor, no qual chave é o nome da coluna do layer de entrada e valor é o nome da coluna do layer de saída. "

NOME\_LAYER\_SAÍDA: "Nome do layer de saída"

### Exemplo de uso:

```
layerIn = "DADO_2_CELL"  
restrictionIn = "classe = 'CLASSE_B1'"  
keepObjectId = false  
tableColumns = { classe = "classe", objeto = "objeto" }  
layerOut = "DADO_1_CELL_LUA"  
merge(layerIn, restrictionIn, keepObjectId, tableColumns, layerOut)
```

### Operação Cálculo de área

```
calculateArea(<NOME_LAYER_ENTRADA>, <UNIDADE_MEDIDA_AREA>, <NOME_COLUNA>)
```

Parâmetros de entrada:

NOME\_LAYER\_ENTRADA: "Nome do layer de entrada"

UNIDADE\_MEDIDA\_AREA: "Unidade de medida de área (Em inglês, Ex. Meters Kilometers Hectares Millimeters Centimeters Miles Feet Inches)"

NOME\_COLUNA: "Nome da coluna para armazenar o resultado no banco de dados"

### Exemplo de uso:

```
calculateArea("DADO_2", "Meters", "area_meters_lua");
```

### Remover um determinado layer

```
bool Layer.remove(<NOME_LAYER>)
```

Parâmetros de entrada:

NOME\_LAYER: " Nome do layer"

Retorno

bool: "Retorna um true se o layer foi removido e false quando não conseguir remover"

### Exemplo de uso:

```
Layer.remove("DADO_2")
```

### Verificar a existência de um determinado layer

```
bool Layer.exit(<NOME_LAYER>)
```

Parâmetros de entrada:

NOME\_LAYER: " Nome do layer"

Retorno

bool: "Retorna um true se o layer existir e false quando não existe"

OBS.: Somente um usuário administrador tem permissão para remover um layer.

### Exemplo de uso:

```
layerExist = Layer.exist("DADO_2")
if(layerExist) then
    Layer.remove("DADO_2")
end
```

### Recuperar todas as colunas do layer

```
Table Layer.getColumns(<NOME_LAYER>)
```

Parâmetros de entrada:

NOME\_LAYER: " Nome do layer"

Retorno

bool: "Retorna uma lista com as colunas do layer informado"

### Exemplo de uso:

```
columns = Layer.getColumns("DADO_2_CELL")
```

### Executar um comando SQL

```
sql(<COMANDO>)
```

Parâmetros de entrada:

COMANDO: " Comando SQL a ser executado "

OBS.: A função pode ser utilizada para executar comandos de alteração de dados e não para fazer consultas, pois não há exibição de dados consultados.

**Exemplo de uso:**

```
sql("update output set x = 1")
```