Tutorial para Rodar o LUGA Model

# Introdução

O luga model foi desenvolvido apenas por meio de scripts PostgreSQL e scripts bash. Todo o script necessário para rodá-lo deve ser baixado diretamente do GitLAB em:

https://gitlab.com/geoimaflora/LTModel-pgsql.git

# Estrutura de arquivos SQL

* proc0\_sigef\_solve\_1.sql
* proc1\_01\_makevalid\_1.sql
* proc1\_01\_makevalid\_2.sql
* proc1\_01\_makevalid\_3.sql
* proc1\_02\_outsidebr\_1.sql
* proc1\_03\_cleanequalshape\_1.sql
* proc1\_04\_cleansamecar\_1.sql
* proc1\_07\_car\_autointersection\_1.sql
* proc1\_07\_car\_autointersection\_2.sql
* proc1\_08\_car\_premium\_1.sql
* proc1\_08\_car\_premium\_2.sql
* proc1\_08\_car\_premium\_3.sql
* proc1\_09\_car\_intersects\_1.sql
* proc1\_09\_car\_intersects\_2.sql
* proc1\_09\_car\_intersects\_3.sql
* proc1\_10\_car\_poor\_clean\_1.sql
* proc1\_10\_car\_poor\_clean\_2.sql
* proc1\_11\_car\_premium\_clean\_1.sql
* proc1\_11\_car\_premium\_clean\_2.sql
* proc1\_12\_car\_poor-premium\_1.sql
* proc1\_12\_car\_poor-premium\_2.sql
* proc1\_12\_car\_poor-premium\_3.sql
* proc1\_13\_car\_poor\_eliminate\_1.sql
* proc1\_14\_car-sigef\_1.sql
* proc1\_14\_car-sigef\_2.sql
* proc1\_14\_car-sigef\_3.sql
* proc1\_15\_car\_sigef\_remove\_1.sql

Temos os scripts ordenados na ordem em que serão processados, esses scripts realizam as operações de análise de autosobreposição de SIGEF/SNCI e do CAR.

Quando o script termina por \_1.sql ou \_3.sql significa que ele não roda em paralelo, porque são scripts que criam ou removem tabelas (no caso do \_1.sql) ou que apenas fazem o log das operações que foram realizadas (para \_3.sql).

## proc0\_sigef\_solve

Realiza toda a consistência das bases privadas do INCRA, não roda em paralelo porque as operações são simples e rodam em apenas alguns minutos.

## proc1\_01\_makevalid

Valida os polígonos do CAR

## proc1\_02\_outsidebr

Remove propriedades que estejam inteiramente fora do Brasil

## proc1\_03\_cleanequalshape

Remove propriedades que possuem o mesmo polígono mantendo o mais recente apenas.

## proc1\_04\_cleansamecar

Remove propriedades que possuem o mesmo número de CAR.

## proc1\_07\_car\_autointersection

Analisa toda a auto intersecção entre os CAR, para posteriormente separar em Premium e Poor pela porcentagem de sobreposição. Este é o script mais pesado da série CAR, por isso ele deve ser pulado em processamentos posteriores caso seja dispensável.

## proc1\_08\_car\_premium

Separa na tabela se é CAR poor ou preimium.

## proc1\_09\_car\_intersects

Cria todos os pares de CAR que possuem intersecção (tabela auxiliar para acelerar os passos seguintes)

## proc1\_10\_car\_poor\_clean

Analisa autointersecção de CAR poor com CAR poor, dando prioridade para pequeno, grande ou aleatório segundo o parâmetro em lt\_model.params.

## proc1\_11\_car\_premium\_clean

Idem ao anterior, mas referente a autointersecção de CAR premium.

## proc1\_12\_car\_poor-premium

Limpa o CAR poor que é sobreposto por CAR premium.

## proc1\_13\_car\_poor\_eliminate

Se o CAR poor for descaracterizado, mescla ele ao seu adjacente.

## proc1\_14\_car-sigef

Limpa o CAR que é sobreposto por uma propriedade privada da base do INCRA.

## proc1\_15\_car\_sigef\_remove

Remove o CAR que foi descaracterizado pelo SIGEF.

# Estrutura de scripts Bash

* bash\_00\_parseOptions: Script auxiliar que apenas avalia as opções passadas para os scripts.
* bash\_01\_helpers: funções auxiliares que são usadas em outros scripts (como displaytime para imprimir a hora atual) e runProc que chama o bash\_02\_execproc para executar o proc desejado.
* bash\_02\_execproc: Ao receber um proc, o script avalia quais arquivos possui terminando com \_1.sql, \_2.sql e \_3.sql e executa-os em paralelo quando for \_2.sql. Esse script também gera o log mostrando o proc que está sendo executado.
* bash\_03\_run\_all: Executa as análises de sobreposição de privados do INCRA e CAR.
* bash\_05\_run\_all\_overlaying.sh: Roda apenas as regras de sobreposição de camadas para os 3 tipos de prioridade pequeno aleatório e grande, alterando a tabela lt\_model.params da forma necessária e renomeando as tabelas de saída com \_random \_small e \_large. Esse script não roda novamente as análises de sobreposição de terras privadas INCRA e CAR.
* luga\_process: Roda todas as análises para os três tipos de prioridade.

## Opções para passar aos scripts

Todos os parâmetros importantes que afetam o modelo LUGA são configurados no BD nas tabelas lt\_model.inputs e lt\_model.params. As opções do script são puramente relacionadas aos dados de conexão ao banco de dados e ao funcionamento dos scripts em si.

As opções de conexão são exatamente iguais aos psql:

* -h: servidor para se conectar
* -U: nome de usuário do banco
* -d: nome do banco de dados a se conectar
* -p: número da porta de conexão

Outras opções que são aceitas são:

* -j (Jobs): número de processos a criar
* -w (wait): Após cada processo dos scripts do CAR e INCRA o script pausa e solicita confirmação para continuar
* specificProc: esse argumento é posicional e deve ser o último a ser passado caso necessite. Esse argumento refere-se a qual proc de processamento do CAR se deseja iniciar, nesse caso ele pula alguns processamentos caso eles já tenham sido efetuados. Deve-se omitir o sufixo \_1.sql por exemplo e escrever apenas proc1\_08\_car\_premium se quiser resumir a partir do proc1\_08\_car\_premium\_1.sql.

# Exemplos de uso:

## Rodar tudo para os três modelos

./luga\_process.sh -h 127.0.0.1 -U postgres -p 5432 -d atlas -j 40

Caso deseje a saída para um log, incluindo erros e avisos colocar no final

./luga\_process.sh -h 127.0.0.1 -U postgres -p 5432 -d atlas -j 40 > saída.log 2>&1

## Rodar apenas as regras de sobreposição entre as camadas (CAR já foi processado)

./bash\_05\_run\_all\_overlaying.sh -h 127.0.0.1 -U postgres -p 5432 -d atlas -j 40 > saída.log 2>&1

## Rodar apenas depois do proc do CAR proc1\_07\_car\_autointersection (o mais demorado e que já rodou anteriormente)

Esse proc não diferencia entre prioridade pequeno, grande ou aleatório, por isso, se quiser apenas separar os CAR pequeno, grande e aleatório não precisa rodar novamente esse proc, é apenas depois dele que se faz necessário rodar três vezes para gerar os três modelos diferentes.

./luga\_process.sh -h 127.0.0.1 -U postgres -p 5432 -d atlas -j 40 proc1\_08\_car\_premium > saída.log 2>&1

# Nota sobre as opções

Todos os argumentos são opcionais, os padrões para os argumentos de conexão são os mesmos padrões utilizados pelo psql, para os demais são os seguintes:

* -j (Jobs): 1
* -w (wait): não para depois de cada proc, se especificar -w, não precisa de valor qualquer, apenas colocar -w o script já vai parar depois de cada proc.
* specificProc: nenhum, vai rodar os procs desde o início.