

Instalação do OSM

Requisitos:

- PostgreSQL + pgRouting + PostGIS + HStore
- GeoServer
- Tomcat
- Java JDK
- osm2pgsql
- osmosis
- Arquivo 'default.style'
- Pasta de estilos SLD.
- Arquivos de estilos adicionais (admin0_countries, contornos, relevo e admin1_states)
- Arquivo 'create_views.sql'
- Arquivo 'import-shapes.sh'
- Arquivo 'import-osm-data.sh'
- Arquivo 'update-diffs.sh'
- Arquivo PBF de dados do OSM
- Arquivos SHP de geometrias adicionais (land, water, admin0, admin1)

Extratos de áreas:

<http://download.geofabrik.de/>

Planeta:

<http://ftp5.gwdg.de/pub/misc/openstreetmap/planet.openstreetmap.org/>

Arquivo state.txt para replicação:

<http://ftp5.gwdg.de/pub/misc/openstreetmap/planet.openstreetmap.org/replication/day/>

Arquivos Land, Water, Admin0 e Admin1:

<http://data.openstreetmapdata.com/land-polygons-split-4326.zip>

<http://data.openstreetmapdata.com/water-polygons-split-4326.zip>

<http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/>

- ne_10m_admin_1_states_provinces.zip
- ne_10m_admin_0_countries.zip
- ne_10m_time_zones.zip

Tunning do PostgreSQL:

Criar pasta para o tablespace.

Dar acesso ao usuário **postgres:postgres** na pasta do tablespace.

postgresql.conf para uma máquina com 32 GB de RAM

default_statistics_target	500
max_connections	200
shared_buffers	6GB
work_mem	157286kB
maintenance_work_mem	2GB
fsync	off (habilitar novamente depois da importação)
autovacuum	off (habilitar novamente depois da importação)
checkpoint_segments	60
min_wal_size	2GB
max_wal_size	4GB
checkpoint_completion_target	0.9
wal_buffers	16MB
random_page_cost	1.1
effective_io_concurrency	2
effective_cache_size	18GB
temp_tablespace	'<TABLESPACE_NOME>'
listen_address	'*'

Criar Banco de Dados para o OSM:

```
$psql> create tablespace <TABLESPACE_NOME> location '<CAMINHO_TABLESPACE>';
$psql> CREATE DATABASE osm WITH OWNER postgres tablespace <TABLESPACE_NOME>;
$psql> \connect osm;
$psql> CREATE EXTENSION postgis;
$psql> CREATE EXTENSION postgis_topology;
$psql> CREATE EXTENSION hstore;
$psql> \q
$psql> exit
```

Reiniciar o PostgreSQL.

Importar o OSM para o PostgreSQL (rodar com *nohup* pois demora):
(Exemplo no arquivo *import-osm-data.sh*)

```
/usr/local/bin/osm2pgsql --number-processes 8 --flat-nodes /opt/osm/data/osm_flat_nodes.db
--latlong --verbose --create --hstore --slim --cache 20000 --database osm --username postgres
--host 127.0.0.1 --style /opt/osm/data/default.style <ARQUIVO_PBF>
```

SE FOR NECESSÁRIO IMPORTAÇÕES ADICIONAIS DE OUTROS EXTRATOS PBF:

```
/usr/local/bin/osm2pgsql --number-processes 8 --flat-nodes /opt/osm/data/osm_flat_nodes.db
--latlong --verbose --append --hstore --slim --cache 20000 --database osm --username postgres
--host 127.0.0.1 --style /opt/osm/data/default.style <ARQUIVO_PBF>
```

Atenção ao parâmetro *--append* e *--create* para não apagar os dados anteriores!

Atenção:

Não apagar o arquivo *osm_flat_nodes.db* pois é necessário para posterior atualização dos dados.

Criar as visões das camadas no PostgreSQL:

Usar o arquivo *create_views.sql*

```
$ export PGPASSWORD=admin
$ psql -U postgres -d osm -a -h localhost -f create_views.sql
```

Verificar no banco de dados se as visões foram criadas.

Importar arquivos SHP para o banco:

Usar o arquivo *import-shapes.sh* após verificar o caminho correto dos arquivos.

Instalar estilos no Geoserver:

Criar Workspace 'osm' no GeoServer

Criar Datastore PostGIS no GeoServer com o nome 'openstreetmap' apontando para o banco 'osm'.

NÃO PUBLICAR AS CAMADAS. ISSO SERÁ FEITO AUTOMATICAMENTE PELO SCRIPT 'SLD_create.sh'

NÃO MUDAR OS NOMES. SÃO USADOS NOS SCRIPTS DE CRIAÇÃO DE CAMADAS.

Ir para a pasta de estilos SLD.

Dar permissão 0777 no arquivo *SLD_create.sh*

Editar o arquivo *SLD_create.sh* e ajustar o endereço do Geoserver, usuário e senha.

```
restapi=http://<IP>:8080/geoserver/rest
login=admin:geoserver
workspace=osm
store=openstreetmap
```

Executar o arquivo *SLD_create.sh*

Verificar no Geoserver se as camadas e estilos foram criados.

Criar os estilos para as camadas *admin0_countries* e *admin1_states* no Geoserver usando os arquivos SLD equivalentes (*admin0_countries.sld* e *admin1_states.sld*) e aplicá-los nas camadas.

Atualizando com o osmosis : Método 1 (preferencial)

Preparar para atualizações (supondo que a pasta atual é /opt/osm/data/osmupdates/):

Criar pasta /opt/osm/data/osmupdates/tmp

Executar:

```
$ export JAVACMD_OPTIONS="-Djava.io.tmpdir=/opt/osm/data/osmupdates/tmp"
$ /opt/osm/sources/osmosis/bin/osmosis --rrii workingDirectory=.
```

Editar o arquivo 'configuration.txt' e modificar 'baseUrl':

baseUrl=http://ftp5.gwdg.de/pub/misc/openstreetmap/planet.openstreetmap.org/replication/day/

Baixar o arquivo 'state.txt' que está neste mesmo endereço e substituir o atual.

Editar o arquivo 'update-diffs.sh' e verificar os diretórios e dados de conexão do banco de dados se estão corretos.

IMPORTANTE: Os passos a seguir deverão ser como *root*:

Criar um cron job

```
$ crontab -e
```

Acrescentar aos jobs existentes (verificar os diretórios a seguir se estão corretos):

(Executar as atualizações todos os dias às 23:00h)

```
00 23 * * * /opt/osm/data/osmupdates/update.sh 1> /opt/osm/data/osmupdates/update.log 2>
/opt/osm/data/osmupdates/error.log
```

Verificar os arquivos update.log e error.log após 48 horas para ver se está atualizando.

Atualizando com o osmupdate : Método 2

Baixar e compilar o osmupdate:

```
wget -O - http://m.m.i24.cc/osmupdate.c | cc -x c - -o osmupdate
```

```
./osmupdate 2017-05-28T23:30:00Z -v --day change_file.osc.gz
```

Exemplos de URL de atualização:

```
--base-url=mirror
--base-url=http://planet.fosm.org/planet/ --base-url-suffix=-replicate
--base-url=http://planet.openstreetmap.org/ --base-url-suffix=-replicate
--base-url=http://planet.openstreetmap.org/redaction-period/ --base-url-suffix=-replicate
--base-url=download.geofabrik.de/europe/germany/bremen-updates
```

Ajustes de desempenho:

Iniciar o Tomcat (catalina.sh) com as seguintes opções (\$CATALINA_OPTS):

```
-Xms128m -Xmx5G -XX:SoftRefLRUPolicyMSPerMB=36000 -XX:+UseParNewGC
-Dorg.geotools.coverage.jaiext.enabled=true
```

No Geoserver:

- Ativar GeoWebCache
- Usar sempre PNG8
- Criar novo BlobStore em disco
- Definir o novo blobstore como padrão.

Editar o web.xml do Geoserver (webapps/geoserver/WEB_INF/):

- Trocar a estratégia de PARTIAL-BUFFER para SPEED

Considerar:

http://docs.geoserver.org/stable/en/user/configuration/image_processing/index.html#jai

<http://docs.geoserver.org/stable/en/user/production/java.html#install-native-jai-and-jai-image-i-o-extensions>

Reiniciar o Tomcat.

Dados técnicos:

Sistemas em execução

: Apache Tomcat (8080), PostgreSQL (5432), Geoserver (8080), GeoNetwork (8080).

Tamanho do banco de dados OSM no PostgreSQL

: 270 GB

Tamanho do arquivo de atualização diária

: 60 MB

Tamanho do cache de atualização (/opt/osm/data/osm_flat_nodes.db)

: 36 GB

Ciclo de atualizações (acesso à internet para baixar o arquivo)

: diária (23:00h)

Porta usada para baixar o arquivo de atualização

: 80 (http)

Origem do arquivo de atualização

: <http://ftp5.gwdg.de/pub/misc/openstreetmap/planet.openstreetmap.org/replication/day/>

Destino do arquivo de atualização (temporário)

: /opt/osm/data/osmupdates/

Demais acessos do servidor para rede externa (internet)

: Nenhum

Acessos externos para serviços no servidor

: porta 8080 (Tomcat - Usuários - Acesso aos mapas e serviços)

: porta 5432 (PostgreSQL - Técnicos - Ferramenta de administração do banco)

: porta **** (SSH - Técnicos - Acesso para manutenção do servidor)

Caminho do cache de imagens do Geoserver

: /opt/osm/apache-tomcat-9.0.0.M17/webapps/geoserver/data/blobstores/

Duração do processo de atualização

: 4 horas para importar dados.

: 2 horas para atualizar as views no banco e reindexar.

Os arquivos de atualização são apagados após serem importados para o banco de dados.

Os dados no banco podem sofrer acréscimo, edição ou remoção, não significando que o banco aumente seu tamanho em 60MB após cada atualização.

Durante o processo de atualização a taxa de utilização do disco (I/O) chegará a 100% podendo deixar alguns serviços lentos ou indisponíveis.

Durante o acesso aos mapas, o PostgreSQL poderá momentaneamente consumir até 100% de utilização de disco (I/O) caso o Geoserver não tenha criado imagens de cache para a área acessada.

Durante o processo de atualização, o servidor Tomcat será desabilitado, interrompendo os sistemas para poupar memória e disco. A atualização dura aproximadamente 6 horas.

Instalar linhas de contornos de altitude

<http://katze.tfiu.de/projects/phyghtmap/phyghtmap.1.html>
<http://katze.tfiu.de/projects/phyghtmap/#Usage>

```
apt-get install python-matplotlib
```

```
wget http://katze.tfiu.de/projects/phyghtmap/phyghtmap_1.80-1_all.deb  
dpkg -i phyghtmap_1.80-1_all.deb
```

Baixar e converter em PBF os arquivos HGT da NASA (toda a América do Sul):

```
phyghtmap --pbf --no-zero-contour --output-prefix contour --line-cat=500,100 --step=10 --jobs=8  
--srtm=1 --a -44.978:-23.383:-40.902:-20.705 --earthdata-user=icemagno --earthdata-  
password=Antares2#2
```

```
-line-cat=500,100 ( major em intervalos de 500m e medium em intervalos de 100m )  
-step=10 ( minor em intervalos de 10m )
```

```
( Somente a área metropolitana do RJ : -43.7544:-23.2363:-42.0378:-22.3183 )  
( Somente o Brasil: -74.18:-33.87:-27.69:4.83 )  
( Estado do RJ : -44.978:-23.383:-40.902:-20.705 )
```

Ferramenta online para identificar os arquivos DEM (HGT) por região: <http://dwtkns.com/srtm30m/>

Consolidar todos os arquivos PBF em um só:

Instalar o osmium:

```
Debian Jessie, adicionar ao /etc/apt/sources.list:  
deb http://ftp.debian.org/debian jessie-backports main
```

```
apt-get install osmium-tool
```

Garantir que não haja um consolidado, senão haverá duplicidade de dados:

```
rm -f consolidado.osm.pbf
```

Consolidar todos os PBF em um só:

```
osmium merge --verbose *.osm.pbf -o consolidado.osm.pbf --overwrite
```

Criar o banco de dados “contour” e adicionar as extensões do PostGIS.

Importar para o PostgreSQL:

```
osm2pgsql --latlong --verbose --create --style ./srtm.style --database contour --username  
postgres -W --host 127.0.0.1 consolidado.osm.pbf
```

Criar uma tabela para ficar com um nome amigável:

```
drop view if exists "contours_line";  
create view "contours_line" AS (  
    SELECT osm_id, way, ele as elevation, contour_ext as cont_ext  
    FROM planet_osm_line  
);
```

srtm.style

#	OsmType	Tag	DataType	Flags
node,way	contour	text	linear	
node,way	contour_ext	text	linear	
node,way	ele	int4	linear	

Criar a camada no Geoserver apontando para a tabela "contours_line" e decorar com o estilo “contornos.sld”.

Criar imagens de relevo

Aproveitar os arquivos HGT baixados pelo phyghtmap na criação das linhas de contorno.

Instalar o UbuntuGIS no repositório e instalar as bibliotecas GDAL.

```
add-apt-repository ppa:ubuntugis/ppa && sudo apt-get update  
sudo apt-get install gdal-bin
```

Construir um arquivo VRT com os HGT:

Criar um arquivo com a lista de todos os HGT baixados:

```
ls hgt/SRTM1v3.0/*.hgt > list_of_files.txt
```

Criar o arquivo VRT:

```
gdalbuildvrt -input_file_list list_of_files.txt -overwrite -addalpha imagens.vrt
```

Consolidar todos os HGT em um único GeoTIFF, com a resolução apropriada:

```
gdal_translate -tr 0.000170 0.000170 -r cubicspline -of GTiff imagens.vrt imagem.tif
```

Criar o GeoTIFF final, para importar no Geoserver:

```
gdaldem hillshade -co TILED=YES -co compress=lzw -s 111120 -z 4 -az 315 -alt 60 -combined  
-compute_edges imagem.tif final.tif
```

Criar a camada no Geoserver e decorar com o estilo “relevo.sld”.

Ir na configuração para WMS do Geoserver e selecionar “bilinear” para “Opções de renderização para dados raster”.

NASA Jet Propulsion Laboratory (JPL), 2013, NASA Shuttle Radar Topography Mission United States 1 arc second. Version 3. 6oS, 69oW. NASA EOSDIS Land Processes DAAC, USGS Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota (<https://lpdaac.usgs.gov>)
