

MANUAL DE INSTALACIÓN

GENERACIÓN DEL FONDO DE MAPA DE OPENSTREETMAP CON POSTGRESQL-POSTGIS, IMPOSM Y MAPSERVER

Preparado por:

Reynaldo Condori, Sylvain Lesage

GEOBOLIVIA

Fecha: 01/04/2012



Índice de contenido

1. Introducción.....	3
2. Importación de los datos de OpenStreetMap.....	6
2.1. Descarga del archivo de datos de OpenStreetMap.....	6
2.2. Instalación de PostGreSQL.....	7
2.2.1. Instalación de PostgreSQL con PostGIS.....	7
2.2.2. Configuración de PostGreSQL.....	7
2.2.3. Acceso al servidor PostGreSQL.....	8
2.2.3.1. IP y puerto.....	8
2.2.3.2. Derechos de acceso.....	8
2.3. Instalación de Imposm.....	10
2.3.1. Apt-pinning para utilizar los repositorios de Debian Wheezy.....	10
2.4. Creación de Usuario y Base de Datos.....	15
2.5. Importación del archivo de datos OSM.....	16
2.5.1. Análisis del archivo de datos.....	16
2.5.2. Importación a la base de datos.....	17
2.5.3. Instalación de los datos para la producción.....	20
2.6. Actualización periódica de la base de datos.....	20
3. Generación del fondo de mapa.....	22
3.1. Instalación de MapServer 6.0.1.....	23
3.2. Instalación del archivo de estilo.....	27
3.2.1. Instalación de mapserver-utils.....	27
3.2.2. Configuración de la compilación.....	29
3.2.3. Compilación del archivo de estilo.....	31
3.3. Publicación del fondo de mapa vía Apache2.....	32
4. Acceso al fondo de mapa.....	33
5. Derechos de Uso	35
5.1. Usted es libre de:.....	35
5.2. Bajo las condiciones siguientes:.....	35
5.3. Entendiendo que:	35

1. Introducción

El presente manual describe el proceso de instalación de un servidor dedicado a la generación de un fondo de mapas utilizando los datos de OpenStreetMap (<http://www.openstreetmap.org>), y su publicación bajo forma de flujo web WMS.

Al fin del proceso de instalación, tendremos:

- un servidor de base de datos PostGreSQL / PostGIS, conteniendo una base de datos geográfica con los datos de OpenStreetMap
- un servidor de mapas MapServer, que provee, vía el servidor Apache2, un flujo WMS (<http://direccióndeIP:puerto/cgi-bin/mapserv>), es decir un webservice que permita acceder de manera remota al mapa de OpenStreetMap

El fondo de mapa generado puede ser utilizado de manera remota (vía Internet, o en una red local) utilizando cualquier tipo de cliente WMS

- desktop: ArcGis, gvSig, QuantumGis, etc.
- web: OpenLayers, GeoNetwork, MapFishApp, etc.

En caso de publicación en la web, es muy importante definir la región que será disponible en el flujo WMS. En efecto, hay pocos proveedores libres de flujos WMS en el mundo, y si se publica el fondo WMS para todo el planeta, existe un gran riesgo que el servidor sea saturado (a nivel del ancho de banda y a nivel del trabajo del procesador) por las numerosas conexiones. La solución, en el caso de Bolivia, es de proveer un fondo solo para Bolivia y sus alrededores.

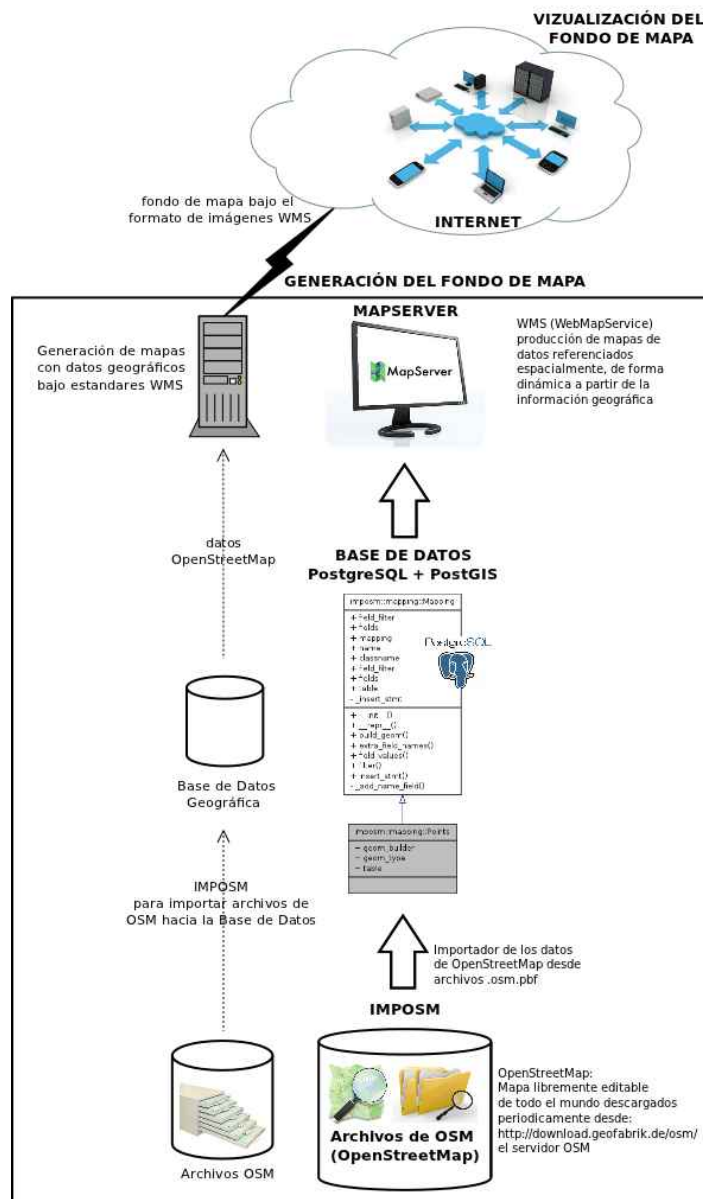
Para lograr este objetivo, los pasos son los siguientes

- **importación de los datos de OpenStreetMap**
 - **descarga del archivo de datos de OpenStreetMap** desde un repositorio en la Internet (los datos de OpenStreetMap son disponibles en libre acceso para la descarga)
 - **instalación de PostGreSQL**, el servidor de base de datos
 - **instalación de imposm, la utilidad de importación**
 - **creación de una base de datos**, que llamaremos imposm, con el soporte geográfico (PostGIS)
 - **importación** del archivo de datos de OpenStreetMap dentro de la base de datos
- **generación del fondo de mapa**

- **instalación de MapServer**, el servidor de mapas
- **instalación del archivo de estilo** osm-outlined,google.map
- **publicación del fondo de mapa vía Apache2**

El proceso de publicación del fondo de mapa esta descrito en el esquema siguiente:

GENERACIÓN DEL FONDO DE MAPA DE OPENSTREETMAP CON IMPOSM, POSTGRESQL - POSTGIS Y MAPSERVER



Todas las herramientas que instalaremos son software libre, y solo describimos la instalación para una

plataforma Linux. El manual de instalación está basado en la distribución Debian 6 Squeeze. Las instrucciones son similares para otras versiones de Debian o de Ubuntu. Para las otras distribuciones, la instalación de paquetes y las direcciones de los archivos pueden ser diferentes. No describimos la instalación en una plataforma Windows.

2. Importación de los datos de OpenStreetMap

2.1. Descarga del archivo de datos de OpenStreetMap

El sitio <http://geofabrik.de> publica un espejo de los datos de OpenStreetMap para su descarga. Se pueden bajar los datos a nivel mundial, regional o nacional. Por ejemplo, para Bolivia:

- bolivia.osm.pbf
- south-america.osm.pbf

Podríamos utilizar el archivo de datos a nivel de Bolivia, pero es más interesante utilizar el archivo de datos a nivel de América del Sur, lo que permite tener los datos alrededor de las fronteras, para que Bolivia no parezca flotando en el aire, en el mapa producido.

Descargamos `south-america.osm.pbf`, con los datos de toda América del Sur, en el directorio `/opt/OSM/`. El formato `.osm` corresponde a los datos OpenStreetMap, y el formato `.osm.pbf` es su versión comprimida.

```
cd /opt/OSM
wget -c http://download.geofabrik.de/osm/south-america.osm.pbf
```

La respuesta a la orden

```
Resolviendo download.geofabrik.de... 188.138.116.142
Connecting to download.geofabrik.de|188.138.116.142|:80... conectado.
Petición HTTP enviada, esperando respuesta... 206 Partial Content
Longitud: 243776331 (232M), 94462571 (90M) remaining [text/plain]
Saving to: `south-america.osm.pbf'
```

```
100%[+++++=====>] 243.776.331 21,7K/s in
  2h 4m
2011-12-12 21:27:35 (12,4 KB/s) - `south-america.osm.pbf' saved
[243776331/243776331]
```

2.2. Instalación de PostGreSQL

2.2.1. Instalación de PostgreSQL con PostGIS

Instalamos la versión 8.4 de PostgreSQL.

```
aptitude install postgresql-8.4 postgresql-8.4-postgis postgresql-doc
postgresql-contrib
```

2.2.2. Configuración de PostgreSQL

Creamos un usuario con derechos de administración de la siguiente manera.

```
su postgres -
createuser -P -s -e nombreusuario
```

la respuesta debería darnos algo parecido a:

```
Ingrese la contraseña para el nuevo rol: .....(colocamos
una contraseña)
Ingrésela nuevamente: .....(repetimos la contraseña)
----> nos debería dar lo siguiente:

CREATE ROLE nombreusuario PASSWORD
'md51512c7191c2fe77ffad84afcfad7ef2f' SUPERUSER CREATEDB
CREATEROLE INHERIT LOGIN;
```

2.2.3. Acceso al servidor PostgreSQL

2.2.3.1. IP y puerto

Asumiendo que la dirección IP del servidor es 192.168.0.1, la añadimos a la lista de direcciones sobre cuales el servidor PostgreSQL esta escuchando (además de localhost) para poder acceder desde afuera:

```
nano /etc/postgresql/8.4/main/postgresql.conf
```

La configuración debe ser parecido

```
listen_addresses = 'localhost, 192.168.0.1'
#listen_addresses = 'localhost'          # what IP address(es) to
#listen on;
#                                     # comma-separated list of
#                                     # addresses;
#                                     # defaults to 'localhost', '*'
#                                     # (change requires restart)
#                                     # (change requires restart)
port = 5432
```

2.2.3.2. Derechos de acceso

Cambiamos los derechos de acceso al servidor PostgreSQL para permitir el acceso del usuario **nombreusuario** desde una otra maquina, por ejemplo 192.168.0.2, y también desde el propio servidor, 127.0.0.1.

```
nano /etc/postgresql/8.4/main/pg_hba.conf
```

La configuración debe ser parecido

```
# Database administrative login by UNIX sockets
```



```
local      all        postgres          ident
# TYPE  DATABASE  USER            CIDR-ADDRESS      METHOD
# "local" is for Unix domain socket connections only
local      all        all              ident
# IPv4 local connections:
host       all        all              127.0.0.1/32      md5
host       all        nombreusuario    127.0.0.1/32      md5
hostssl    all        nombreusuario    127.0.0.1/32      md5
host       all        nombreusuario    192.168.0.2/32    md5
hostssl    all        nombreusuario    192.168.0.2/32    md5
# IPv6 local connections:
host       all        all              ::1/128           md5
```

Para que se tomen en cuenta los cambios, recargamos la configuración en el servidor postgresql:

```
service postgresql reload
```

Detalle: según el archivo de configuración, existen dos formas de tomar en cuenta las modificaciones:

- cuando se configura el archivo “postgresql.conf”, se tiene que reiniciar el servidor

```
service postgresql restart
```

- cuando se configura el archivo “pg_hba.conf”, solo hay que recargar la configuración

```
service postgresql reload
```

Para conectarse al servidor PostGreSQL desde el propio servidor (192.168.0.1):

```
psql -Unombreusuario -W -h 127.0.0.1 -p5432 postgres
```

Para conectarse al servidor PostGreSQL desde la maquina del usuario (192.168.0.2):

```
psql -Unombreusuario -W -h 192.168.0.1 -p5432 postgres
```

Para salir del postgresql

```
\q
```

OJO: en el caso de que el servidor no permita el ingreso a la base de datos con el usuario creado verificar las reglas de un eventual firewall.

2.3. *Instalación de Imposm*

Instalamos imposm, la utilidad que permite importar los archivos .osm, o .osm.pbf, dentro de una base de datos PostGreSQL.

2.3.1. **Apt-pinning para utilizar los repositorios de Debian Wheezy**

Para realizar la siguiente instalación de imposm dentro de Debian Squeeze, tenemos que hacer un apt-pinning, lo que quiere decir que, como el paquete imposm no se encuentran dentro de los repositorios de Debian Squeeze, tendremos que añadir los repositorios de la versión siguiente Debian Wheezy, cual sí tiene imposm dentro de sus repositorios. Procedemos

```
nano /etc/apt/sources.list
```

Añadimos las siguientes líneas

```
##### Wheezy #####  
deb http://ftp.us.debian.org/debian/ wheezy main  
deb http://security.debian.org/ wheezy/updates main
```

Si no existe todavía, creamos el archivo preferences

```
nano /etc/apt/preferences
```

con el contenido siguiente

```
Package: *  
Pin: release a=squeeze  
Pin-Priority: 900  
  
Package: *  
Pin: release a=wheezy  
Pin-Priority: 800
```

lo que indica que Squeeze tiene mayor prioridad (900) en la selección de los paquetes y Wheezy una menor prioridad (800) por lo que se utilizarán los paquetes de Wheezy solo si no existen en Squeeze, o si están requeridos como dependencias de otros paquetes (por ejemplo, en nuestro caso como lo veremos más tarde: python)

Ahora actualizamos el caché de los paquetes

```
aptitude update
```

Instalamos las dependencias de imposm

```
aptitude install build-essential python-dev protobuf-compiler  
libprotobuf-dev libtokyocabinet-dev python-psycopg2 libgeos-c1
```

Instalamos python-pip, el instalador para python

```
aptitude install python-pip
```

Instalamos pg_config, que se encuentra en el paquete debian libpq-dev

```
aptitude install libpq-dev
```

También cython

```
aptitude install cython
```

Una vez instalados las dependencias de paquetes, ahora instalamos imposm

```
aptitude install imposm
```

La respuesta debería ser algo parecido

```
The following actions will resolve these dependencies:
```

```
Install the following packages:
```

- 1) libexpat1-dev [2.0.1-7.2 (testing)]
- 2) libpython2.7 [2.7.2-7 (testing)]
- 3) libssl1.0.0 [1.0.0e-3 (testing)]
- 4) libtinfo5 [5.9-4 (testing)]
- 5) multiarch-support [2.13-21 (testing)]
- 6) python-egenix-mxdatetime [3.2.1-1 (testing)]
- 7) python-egenix-mxtools [3.2.1-1 (testing)]
- 8) python-psycopg2 [2.4.2-1 (testing)]
- 9) python2.7 [2.7.2-7 (testing)]
- 10) python2.7-dev [2.7.2-7 (testing)]
- 11) python2.7-minimal [2.7.2-7 (testing)]

Upgrade the following packages:

- 12) cython [0.12.1-1.1 (now, stable) -> 0.15.1-1 (testing)]
- 13) libexpat1 [2.0.1-7 (now, stable) -> 2.0.1-7.2 (testing)]
- 14) python [2.6.6-3+squeeze6 (now, stable) -> 2.7.2-9 (testing)]
- 15) python-dev [2.6.6-3+squeeze6 (now, stable) -> 2.7.2-9 (testing)]
- 16) python-minimal [2.6.6-3+squeeze6 (now, stable) -> 2.7.2-9 (testing)]
- 17) python-pkg-resources [0.6.14-4 (now, stable) -> 0.6.24-1 (testing)]
- 18) python-setuptools [0.6.14-4 (now, stable) -> 0.6.24-1 (testing)]
- 19) python-support [1.0.10 (now, stable) -> 1.0.14 (testing)]

Accept this solution? [Y/n/q/?] Y

The following NEW packages will be installed:

```
imposm libexpat1-dev{a} libprotobuf7{a} libpython2.7{a}
libssl1.0.0{a} libtinfo5{a} libtokyocabinet8{a} multiarch-
support{a}
```

```
python-egenix-mxdatetime{a} python-egenix-mxtools{a} python-
imposm{a} python-imposm-parser{a} python-psycpg2{a} python-
shapely{a} python2.7{a}
```

```
python2.7-dev{a} python2.7-minimal{a}
```

The following packages will be REMOVED:

```
libpython2.6{u} python2.6-dev{u}
```

The following packages will be upgraded:

```
cython libexpat1 python python-dev python-minimal python-pkg-  
resources python-setuptools python- support
```

8 packages upgraded, 17 newly installed, 2 to remove and 0 not upgraded.

Need to get 17.2 MB of archives. After unpacking 29.4 MB will be used.

OJO durante el proceso de instalación nos aparecerá algo parecido a esto:

Hay algunos servicios instalados en el sistema que requieren reiniciarse al actualizar

paquetes como libpam, libc, y libssl. Ya que reiniciar estos servicios puede provocar una interrupción de servicio del sistema, habitualmente se le solicitar?? en cada actualización una lista de los servicios que desea reiniciar. Puede seleccionar esta opción para impedir que se le solicite esta información; en su lugar, cada reinicio de servicio se hará de forma automática de forma que evitará que se le planteen preguntas cada vez que se actualice una biblioteca.

Quiere que los servicios se actualicen durante una actualización de paquete sin solicitar confirmación?

<Yes>

Probamos que imposm se instaló correctamente

```
imposm --version
```

debe devolver

```
imposm 2.3.2
```

Añadimos "Shapely Speedups" para acelerar los importes, primero instalando

```
aptitude install libgeos-dev
```

y luego, vía pip:

```
pip install https://github.com/olt/shapely_speedups/tarball/master
```

2.4. Creación de Usuario y Base de Datos

Creamos el usuario imposm, dedicado a la futura base de datos de los datos de OpenStreetMap. Este usuario será dueño de esta base, y no tendrá derechos para las otras bases. Desde el servidor:

```
createuser -h 127.0.0.1 -U nombreusuario --no-superuser --no-
  createrole --createdb -P imposm

Ingrese la contraseña para el nuevo rol: .....
  (agregamos una contraseña)

Ingrésela nuevamente: ..... (repetimos la contraseña)

Contraseña: ..... (ingresamos con la contraseña)
```

Modificamos el pg_hba.conf para dar a imposm los derechos de conexión en local

```
nano /etc/postgresql/8.4/main/pg_hba.conf
```

La configuración debería ser de la siguiente manera:

```
# IPv4 local connections:
host      all             imposm    127.0.0.1/32      md5
hostssl   all             imposm    127.0.0.1/32      md5
```

Recargamos la configuración

```
service postgresql reload
```

Creamos la base de datos imposm, con el dueño imposm

```
createdb -h 127.0.0.1 -U nombreusuario -E UTF8 -O imposm -T template0  
imposm
```

Importamos el soporte geográfico, Postgis, en la base imposm, conectandonos con el nuevo usuario imposm

```
createlang -h 127.0.0.1 -U imposm plpgsql imposm  
  
psql -h 127.0.0.1 -U imposm -d imposm -f  
/usr/share/postgresql/8.4/contrib/postgis-1.5/postgis.sql  
  
psql -h 127.0.0.1 -U imposm -d imposm -f  
/usr/share/postgresql/8.4/contrib/postgis-1.5/spatial_ref_sys.sql  
  
psql -h 127.0.0.1 -U imposm -d imposm -f  
/usr/share/postgresql/8.4/contrib/postgis_comments.sql
```

2.5. Importación del archivo de datos OSM

Imposm funciona en tres fases:

- análisis del archivo de datos, con exportación en una carpeta (opción --read)
- importación de los datos analizados en la base de datos, con un prefijo sobre las tablas (opción --write)
- instalación de los datos para la producción, suprimiendo el prefijo de tablas (opción --deploy-production-tables): por ejemplo la tabla osm_new_admin se vuelve osm_admin.

2.5.1. Análisis del archivo de datos

Creamos la carpeta donde se almacenará la análisis del archivo south-america.osm.pbf

```
mkdir -p /opt/OSM/imposm  
  
cd /opt/OSM/imposm
```



```
imposm --read /opt/OSM/south-america.osm.pbf
```

La respuesta debería ser parecida a

```
Patching Shapely.  
[15:37:49] ## reading /opt/OSM/south-america.osm.pbf  
[15:37:49] coords: 18283k nodes: 365k ways: 2611k relations: 18k  
          (estimated)  
[15:40:05] coords: 23575k nodes: 151k ways: 2222k relations: 21k  
[15:40:05] reading took 2 m 16s
```

Esta fase de análisis del archivo .osm.pbf crea archivos de caché en la carpeta corriente, /opt/OSM/imposm. En efecto, listando los archivos:

```
ls /opt/OSM/imposm/
```

nos debería mostrar los siguientes archivos

```
imposm_coords.cache  imposm_nodes.cache  imposm_relations.cache  
imposm_ways.cache
```

Ojo 1: tener en cuenta que la memoria RAM del servidor tiene que ser superior o igual a 1024MB, para que imposm pueda ejecutarse sin error.

2.5.2. Importación a la base de datos

Importamos la información del cache en la base de datos:

```
imposm --write --database imposm --host 127.0.0.1 --user imposm
```

La respuesta es la siguiente

```
Enabling Shapely speedups.  
password for imposm at 127.0.0.1:  
[15:40:05] ## dropping/creating tables  
[15:40:11] ## writing data  
[15:40:30] relations: 21k/21k  
[15:40:41] relations: total time 29s for 21375 (724/s)  
[2012-03-12 15:50:36,232] imposm.db.postgis - WARNING - error while  
importing "(91630020, u'administrative',  
<psycopg2._psycopg.Binary object at 0x4d5b5f0>, u'Ross  
D\xe9pendency (New Zealand Claim)', 2)": transform: couldn't  
project point (-4.14066e-08 -90 0): latitude or longitude  
exceeded limits (-14)  
[15:50:36] ways: total time 9 m 54s for 2221861 (3734/s)  
[15:51:07] nodes: 151k/151k  
[15:51:08] nodes: total time 32s for 151935 (4671/s)  
[15:51:08] ## creating generalized tables  
[15:51:36] generalizing tables took 27s  
[15:51:36] ## creating union views  
[15:51:36] creating views took 0 s  
[15:51:36] writing took 11m 30s
```

Imposm tiene también una opción para optimizar las tablas de la base de datos:

```
imposm --optimize --database imposm --host 127.0.0.1 --user imposm
```

lo que nos muestra:

```
password for imposm at 127.0.0.1:  
[15:51:36] ## optimizing tables  
Clustering table osm_new_transport_areas  
Clustering table osm_new_mainroads  
Clustering table osm_new_buildings  
Clustering table osm_new_mainroads_gen1  
Clustering table osm_new_mainroads_gen0  
Clustering table osm_new_amenities  
Clustering table osm_new_waterareas_gen1  
Clustering table osm_new_waterareas_gen0  
Clustering table osm_new_motorways_gen0  
Clustering table osm_new_aeroways  
Clustering table osm_new_motorways  
Clustering table osm_new_transport_points  
Clustering table osm_new_railways_gen0  
Clustering table osm_new_railways_gen1  
Clustering table osm_new_landusages  
Clustering table osm_new_waterways  
Clustering table osm_new_railways  
Clustering table osm_new_motorways_gen1  
Clustering table osm_new_waterareas  
Clustering table osm_new_places
```

```
Clustering table osm_new_admin  
Clustering table osm_new_minorroads  
Clustering table osm_new_landusages_gen1  
Clustering table osm_new_landusages_gen0  
Vacuum analyze  
[15:54:43] optimizing took 3 m 6 s
```

OjO: las tres operaciones (análisis, importación y optimización) se pueden hacer en el mismo paso con:

```
imposm --overwrite-cache --read --write --optimize --database imposm  
--host 127.0.0.1 --user imposm /opt/OSM/south-america.osm.pbf
```

2.5.3. Instalación de los datos para la producción

Cambiamos el prefijo de importación de las tablas, y permitir su uso en producción. En efecto, luego de la importación, las tablas se llaman `osm_new_admin`, `osm_new_aeroways`, etc. El comando siguiente saca la parte “new_” del nombre de las tablas, para que se llamen `osm_admin`, `osm_aeroways`, etc. Estos nombres (sin “new_” son los nombres utilizados luego por MapServer para acceder a los datos)

```
imposm --deploy-production-tables --database imposm --host 127.0.0.1  
--user imposm
```

La respuesta tiene que ser

```
Patching Shapely.  
password for imposm at 127.0.0.1:  
[15:55:01] imposm took 0 s
```

2.6. Actualización periódica de la base de datos

Los datos de OpenStreetMap están enriquecidos cada día por su comunidad, lo que implica que su

calidad mejora regularmente. Para sacar provecho de esta mejora constante de los datos, se puede configurar la descarga y importación de los nuevos datos en la base de datos, vía la utilidad cron del sistema.

Para automatizar la actualización de los datos, creamos un script que, cada determinado tiempo, descargue los datos de un espejo de OpenStreetMap y los suba de forma automática en la base de datos.

Para que los archivos sean más organizados, creamos unos directorios en la siguiente dirección **/opt/OSM/**.

```
mkdir -p /opt/OSM/datos /opt/OSM/scripts /opt/OSM/logs /opt/OSM/cache
```

- El directorio datos almacenará los archivos .osm.pbf descargados,
- el directorio scripts almacenará el script de actualización (extensión .sh),
- el directorio logs almacenará los archivos .log los cuales serán generados de forma automática al ejecutarse el script y contendrán información de todo el proceso del script
- el directorio cache será el directorio donde se almacenarán los archivos temporales generados cuando se lanza la subida de datos OSM a la base de datos PostgreSQL.

Creamos el script de actualización de los datos en el directorio scripts

```
nano import_dat_osm.sh
```

Dentro del archivo colocamos el siguiente código

```
#!/bin/sh

# Descargamos datos desde OSM

## hacemos una copia de backup del archivo anterior

mv /opt/OSM/datos/south-america.osm.pbf /opt/OSM/datos/south-america.osm.pbf.old

## descargamos el nuevo archivo

wget -O /opt/OSM/datos/south-america.osm.pbf -c
http://download.geofabrik.de/osm/south-america.osm.pbf
```

```
# Analizamos el archivo y lo importamos en la base de datos

cd /opt/OSM/cache/

imposm --overwrite-cache --read --write --optimize --connection
  postgis://imposm:supassword@localhost/imposm
  /opt/OSM/datos/south-america.osm.pbf

# Cambiamos el prefijo para uso en producción

imposm --deploy-production-tables --connection
  postgis://imposm:supassword@localhost/imposm
```

Le damos permisos de ejecución al archivo .sh

```
chmod +x import_dat_osm.sh
```

Ahora configuramos el **CRON** ya que será quien active de forma automática el script.

```
crontab -u root -e
```

Dentro del archivo en la última línea colocamos lo siguiente.

```
# activar cada 23 horas con 10 min. (23h10)

10 23 * * * /opt/OSM/scripts/import_dat_osm.sh >
  /opt/OSM/logs/import_dat_osm`date +%Y_%m_%d`.log
  -->dejar_un_espacio_vacío_al_final_de_la_línea
```

3. Generación del fondo de mapa

Para generar el fondo de mapa a partir de los datos de OpenStreetMap, instalamos MapServer (el servidor de mapas) y descargamos un archivo de estilo dedicado a los datos OSM. Finalmente configuramos el servidor web Apache2 para que llame al script de MapServer para generar el fondo de mapa según las consultas web.

3.1. *Instalación de MapServer 6.0.1*

Se necesita instalar la versión 6.0.1 de MapServer para poder generar el fondo de mapa con el estilo que descargaremos. Lamentablemente, en Squeeze y Wheezy, solo se tiene la versión 5.6.6. Necesitamos entonces compilar MapServer desde su código fuente.

Para preparar la compilación, instalamos las utilidades y librerías necesarias

```
aptitude install g++ libpng12-0 libfreetype6 libgd2-xpm zlib-bin
```

Instalamos las librerías recomendadas

```
aptitude install libproj0 libcurl3 libagg-dev libgdal-dev
```

Instalamos las librerías opcionales

```
aptitude install libtiff4-dev libgeotiff-dev libjpeg-dev libgeos-dev  
libxml2-dev libpq-dev libgd2-xpm-dev libpng12-dev libfreetype6-  
dev zlib1g-dev libproj-dev libcurl-dev
```

Descargamos el código fuente de MapServer

```
mkdir -p /opt/mapserver  
cd /opt/mapserver  
wget http://download.osgeo.org/mapserver/mapserver-6.0.1.tar.gz  
tar -xvzf mapserver-6.0.1.tar.gz
```

Antes de lanzar la compilación, lanzamos la configuración, que verifica, entre otros, si el sistema tiene todos los requisitos para la compilación

```
cd /opt/mapserver/mapserver-6.0.1  
./configure --prefix=/usr --with-wfs --with-wcs --with-sos --with-  
wmsclient --with-wfsclient --with-proj --with-gdal --with-ogr  
--with-geos --with-gd --with-tiff --with-jpeg --with-png=/usr
```

```
--with-agg --with-eppl --with-postgis --with-sde --with-freetype  
--with-threads --with-experimental-png
```

La respuesta a la orden

MapServer is now configured for

```
----- Compiler Info -----  
C compiler:          gcc -O2 -fPIC -Wall -DNDEBUG  
C++ compiler:       g++ -O2 -fPIC -Wall -DNDEBUG  
Debug:  
Generic NINT:  
  
----- Renderer Settings -----  
OpenGL support:  
zlib support:       -DUSE_ZLIB  
png support:        -DUSE_PNG  
gif support:        -DUSE_GIF  
jpeg support:       -DUSE_JPEG  
freetype support:   -DUSE_FREETYPE  
iconv support:      -DUSE_ICONV  
AGG support:        internal  
SVG Symbol support: -DUSE_AGG_SVG_SYMBOLS  
Cairo (SVG,PDF) support:  
KML support:  
  
----- Support Libraries -----  
Proj.4 support:     -DUSE_PROJ  
Libxml2 support:    -DUSE_LIBXML2
```


FriBidi support:

Curl support: -DUSE_CURL -DUSE_CURLOPT_PROXYAUTH

FastCGI support:

Threading support: -DUSE_THREAD

GEOS support: -DUSE_GEOS

XML Mapfile support:

XSLT support:

EXSLT support:

----- Data Format Drivers -----

PostGIS support: -DUSE_POSTGIS
-DPOSTGIS_HAS_SERVER_VERSION

ArcSDE support:

OGR support: -DUSE_OGR

GDAL support: -DUSE_GDAL

Oracle Spatial support:

----- OGC Services -----

WMS Server: -DUSE_WMS_SVR

WMS Client: -DUSE_WMS_LYR

WFS Server: -DUSE_WFS_SVR

WFS Client: -DUSE_WFS_LYR

WCS Server: -DUSE_WCS_SVR

```
SOS Server: -DUSE_S0S_SVR
```

```
----- MapScript -----
```

```
PHP MapScript: no
```

Ahora lanzamos la compilación

```
make
```

Verificamos que la compilación funcionó (mapserv es el único archivo creado en el directorio /opt/mapserver/mapserver-6.0.1) lanzando

```
./mapserv
```

La respuesta normal tiene que ser la siguiente si funcionó la compilación.

```
This script can only be used to decode form results and  
should be initiated as a CGI process via a httpd server.
```

Hacemos la instalación de MapServer en el servidor, copiando los binarios en el siguiente directorio /usr/local/bin:

```
cp shp2img /usr/local/bin  
cp legend /usr/local/bin  
cp shptree /usr/local/bin  
cp mapserv /usr/local/bin  
cp shptreevis /usr/local/bin  
cp shptreetst /usr/local/bin
```

```
cp scalebar /usr/local/bin
cp tile4ms /usr/local/bin
cp sortshp /usr/local/bin
cp mapscriptvars /usr/local/bin
cp msencrypt /usr/local/bin
cp mapserver-config /usr/local/bin
```

3.2. *Instalación del archivo de estilo*

Para instalar el archivo de estilo para la región que queremos (Bolivia y sus alrededores), seguimos los siguientes pasos:

- descarga de mapserver-utils, que contiene un estilo base y un entorno de compilación (Makefile) que permite generar un archivo de estilo adaptado
- configuración de la compilación según los parámetros de Bolivia
- generación del archivo de estilo por compilación

3.2.1. **Instalación de mapserver-utils**

La página <https://code.google.com/p/mapserver-utils/wiki/TweakingMapStylesImposm> explica como utilizar mapserver-utils. Recuperamos el código fuente de la rama imposm vía subversion. Primero instalamos la herramienta de manejo de versiones subversion

```
aptitude install subversion
```

y luego recuperamos el código

```
cd /opt/
svn checkout https://mapserver-
utils.googlecode.com/svn/branches/imposm-dm-branch/ mapserver-
utils
```

lo que devolverá la lista de los archivos descargados

```
A    mapserver-utils/landusage.map
A    mapserver-utils/places.map
A    mapserver-utils/symbols
A    mapserver-utils/symbols/ushwy-shield.png
A    mapserver-utils/symbols/circle-shield.png
A    mapserver-utils/symbols/interstate-shield.png
A    mapserver-utils/fonts
A    mapserver-utils/fonts/DejaVuSansCondensed-Bold.ttf
A    mapserver-utils/fonts/DejaVuSansCondensed.ttf
A    mapserver-utils/fonts.lst
A    mapserver-utils/land.map
A    mapserver-utils/buildings.map
A    mapserver-utils/README
A    mapserver-utils/boundaries.sql.in
A    mapserver-utils/epsg
A    mapserver-utils/highways.map
A    mapserver-utils/boundaries-gadm.sql
A    mapserver-utils/generate_style.py
A    mapserver-utils/osmbase.map
A    mapserver-utils/resetsizes.inc
A    mapserver-utils/data
```

```
A mapserver-utils/data/seed.shx
A mapserver-utils/data/boundaries.shp
A mapserver-utils/data/boundaries.prj
A mapserver-utils/data/boundaries.dbf
A mapserver-utils/data/seed.shp
A mapserver-utils/data/seed.dbf
A mapserver-utils/data/seed.prj
A mapserver-utils/data/boundaries.qix
A mapserver-utils/data/boundaries.shx
A mapserver-utils/data/Makefile
A mapserver-utils/borders.map
A mapserver-utils/Makefile
U mapserver-utils

Revisión obtenida: 111
```

3.2.2. Configuración de la compilación

En el archivo

```
nano /opt/mapserver-utils/generate_style.py
```

modificamos el acceso a la base de datos imposm

```
'db_connection': '"host=127.0.0.1 dbname=imposm user=imposm
password=***** port=5432"',
```

Modificamos también el archivo Makefile

```
nano /opt/mapserver-utils/Makefile
```

para configurar el prefijo de las tablas, el cuadro de la región a mapear, y el estilo a aplicar

```
OSM_PREFIX=osm_  
OSM_EXTENT=-7767064.05291532 -2619995.95998552 -6415134.98573511  
-1111649.76901235  
STYLE=outlined,google
```

Parece también que se tiene que modificar el Makefile de la carpeta data/, para añadir las cuatro líneas de cambio de nombre (mv), porque algunos archivos cambiaron de nombre

```
nano /opt/mapserver-utils/data/Makefile
```

Agregamos lo siguiente (con una tabulación, no espacios, al principio de las 6 últimas líneas)

```
10m-admin-0-boundary-lines-land.shp: 10m-admin-0-boundary-lines-  
land.zip  
  
unzip 10m-admin-0-boundary-lines-land.zip  
  
mv ne_10m_admin_0_boundary_lines_land.dbf  
10m_admin_0_boundary_lines_land.dbf  
  
mv ne_10m_admin_0_boundary_lines_land.prj  
10m_admin_0_boundary_lines_land.prj  
  
mv ne_10m_admin_0_boundary_lines_land.shp  
10m_admin_0_boundary_lines_land.shp  
  
mv ne_10m_admin_0_boundary_lines_land.shx  
10m_admin_0_boundary_lines_land.shx  
  
touch 10m_admin_0_boundary_lines_land.shp
```

3.2.3. Compilación del archivo de estilo

Lanzamos la creación del archivo de estilo .map (OjO: muy largo la primera vez, porque se tienen que descargar algunos archivos SHP)

```
cd /opt/mapserver-utils/  
make
```

La Respuesta será

```
python generate_style.py -s outlined,google -l 2 > level2.inc  
python generate_style.py -s outlined,google -l 4 > level4.inc  
python generate_style.py -s outlined,google -l 3 > level3.inc  
python generate_style.py -s outlined,google -l 5 > level5.inc  
python generate_style.py -s outlined,google -l 6 > level6.inc  
python generate_style.py -s outlined,google -l 7 > level7.inc  
python generate_style.py -s outlined,google -l 8 > level8.inc  
python generate_style.py -s outlined,google -l 9 > level9.inc  
python generate_style.py -s outlined,google -l 10 > level10.inc  
python generate_style.py -s outlined,google -l 11 > level11.inc  
python generate_style.py -s outlined,google -l 12 > level12.inc  
python generate_style.py -s outlined,google -l 13 > level13.inc  
python generate_style.py -s outlined,google -l 14 > level14.inc  
python generate_style.py -s outlined,google -l 15 > level15.inc  
python generate_style.py -s outlined,google -l 16 > level16.inc  
python generate_style.py -s outlined,google -l 17 > level17.inc
```

```
python generate_style.py -s outlined,google -l 18 > level18.inc
cd data; make
make[1]: Entering directory `/opt/mapserver-utils/data'
shptree 10m_admin_0_boundary_lines_land.shp 8
creating index of new LSB format
make[1]: Leaving directory `/opt/mapserver-utils/data'
cpp -D_debug=1 -D_layerdebug=1 -DOSM_PREFIX=osm_ -DOSM_SRID=900913 -P
-o osm- outlined,google.map osmbase.map -Dtheme=\"style-
outlined,google.inc\" -D_proj_lib=\"`pwd`\"
sed -i 's/##.*$/g' osm-outlined,google.map
sed -i '/^ *$/d' osm-outlined,google.map
sed -i -e 's/OSM_PREFIX_/osm_/g' osm-outlined,google.map
sed -i -e 's/OSM_SRID/900913/g' osm-outlined,google.map
sed -i -e 's/OSM_UNITS/meters/g' osm-outlined,google.map
sed -i -e 's/OSM_EXTENT/-7767064.05291532 -2619995.95998552
-6415134.98573511 -1111649.76901235/g' osm-outlined,google.map
sed -i -e 's/OSM_WMS_SRS/\"EPSG:900913 EPSG:4326 EPSG:3857 EPSG:2154
EPSG:310642901 EPSG:4171 EPSG:310024802 EPSG:310915814
EPSG:310486805 EPSG:310702807 EPSG:310700806 EPSG:310547809
EPSG:310706808 EPSG:310642810 EPSG:310642801 EPSG:310642812
EPSG:310032811 EPSG:310642813 EPSG:2986\"/g' osm-
outlined,google.map
```

El archivo de estilo ha sido generado y se encuentra en /opt/mapserver-utils/outlined,google.map.

3.3. Publicación del fondo de mapa vía Apache2

La última etapa corresponde a la publicación del fondo de mapa vía el servidor web Apache2, o sea que Apache2 llamará al script MapServer cuando le llegará una consulta, y el script MapServer se conectará con la base de datos y el archivo de estilo para generar el fondo de mapa pedido.

Instalamos el servidor web apache2

```
aptitude install apache2
```

Copiamos el script de mapserver en la carpeta de scripts de Apache2

```
cp /opt/mapserver/mapserver-6.0.1/mapserv /usr/lib/cgi-bin/
```

Creamos, con los derechos adecuados, la carpeta donde se generaran los archivos de salida del script de MapServer

```
mkdir /tmp/ms_tmp  
chmod 777 /tmp/ms_tmp
```

Para que está carpeta se cree a cada reinicio del servidor (porque se puede borrar la carpeta /tmp), añadimos está creación automática

```
nano /etc/rc.local
```

```
# Crear la carpeta /tmp/ms_tmp necesitada por MapServer  
mkdir /tmp/ms_tmp  
chmod 777 /tmp/ms_tmp
```

Para verificar que funciona el vínculo entre Apache2 y MapServer, probamos en un navegador de acceder a `http://direcciónIP/cgi-bin/mapserv`, en este caso sin parámetros. La respuesta debería ser

```
No query information to decode. QUERY_STRING is set, but empty.
```

4. Acceso al fondo de mapa

Para verificar que la generación del fondo de mapa es correcta, se puede utilizar el visualizador de

MapServer, llamando desde un navegador a la dirección siguiente

```
http://direcciónIP/cgi-bin/mapserv?map=/opt/mapserver-utils/osm-  
outlined,google.map&mode=browse&template=openlayers&layers=all
```

Para probar el acceso al fondo de mapa desde Qgis, por ejemplo, entrar la dirección siguiente en el campo de URL del flujo WMS:

```
http://direcciónIP/cgi-bin/mapserv?map=/opt/mapserver-utils/osm-  
outlined,google.map&mode=browse&template=openlayers&layers=all
```

Una nueva capa tiene que aparecer, con el fondo de mapa de OpenStreetMap.

5. Derechos de Uso

Este obra está bajo una [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported](#).



5.1. Usted es libre de:

- copiar, distribuir y comunicar públicamente la obra
- **Remezclar** — transformar la obra.

5.2. Bajo las condiciones siguientes:

- **Reconocimiento** — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
- **No comercial** — No puede utilizar esta obra para fines comerciales.
- **Compartir bajo la misma licencia** — Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

5.3. Entendiendo que:

- **Renuncia** — Alguna de estas condiciones puede [no aplicarse](#) si se obtiene el permiso del titular de los derechos de autor
- **Dominio Público** — Cuando la obra o alguno de sus elementos se halle en el [dominio público](#) según la ley vigente aplicable, esta situación no quedará afectada por la licencia.
- **Otros derechos** — Los derechos siguientes no quedan afectados por la licencia de ninguna manera:
 - Los derechos derivados de [usos legítimos](#) u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.
 - Los derechos [morales](#) del autor;
 - Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo [derechos de imagen](#) o de privacidad.
- **Aviso** — Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.