

Manual de configuración

Alvaro Garcia Sanabria

1 Creación de nuevos mapas

Uno de los cambios más probables es el de la creación de nuevos mapas (con independencia de que sean parciales para una visualización sobreimpresa a otros), proceso que requiere de los siguientes pasos:

- Obtención de la información y, en su caso, iconografía a mostrar.
- Creación del archivo de configuración de mapa.
- Incorporación del mapa generado a la memoria caché.
- Generación de la copia caché.
- Incorporación del mapa generado al visor web.

A continuación se explican tales pasos con más detalle.

1.1 Obtención de informaciones e iconografías a mostrar

El primer paso es el de recabar las fuentes y, en su caso, obtener los archivos que contengan las informaciones que se deseen usar en los mapas a generar, así como los elementos gráficos que se deseen incorporar a tales mapas.

El método de obtención de estos se deja en manos del usuario-lector ya que es imposible saber, a priori, los datos que en cada caso se emplearán.

En cualquier caso, los archivos han de depositarse en mapserver/bind-mounts/maps/ o alguna subcarpeta de esta, ya que de otro modo MapSever podría no acceder a los mismos.

1.2 Generación del archivo de configuración de mapa

Obtenidas las informaciones o los orígenes a incorporar al nuevo mapa, se deben indicar qué características se van a plasmar en el nuevo mapa, lo que se consigue mediante un nuevo archivo de configuración de mapa para MapServer, que deberá respetar el formato establecido para ellos, disponible en la documentación de MapServer¹.

Durante la creación de estos archivos de configuración deberá tenerse en cuenta que, salvo modificaciones, los archivos depositados en `mapserver/bind-mounts/maps/` estarán disponibles para MapServer en `/maps` desde la raíz del sistema de archivos del contenedor, por lo que se tendrá que tener en cuenta tal cambio al detallar cualesquiera rutas en los archivos de configuración nuevos.

En cualquier caso, tales archivos también han de depositarse en `/mapserver/bind-mounts/maps/` o alguna subcarpeta de esta, ya que de otro modo MapSever podría no acceder a los mismos.

Se recomienda encarecidamente comprobar que MapServer es capaz de procesar adecuadamente el archivo de configuración creado y que el resultado es el deseado antes de continuar.

1.3 Incorporación del mapa generado a la caché

Una vez que MapServer es capaz de generar el mapa deseado, debe incluirse este nuevo mapa a la configuración de MapCache para poder disponer de la copia en la caché de MapCache del mapa.

Para ello en el software deberán definirse, al menos, otro origen (*source*) y otro conjunto de teselas (*tileset*) que incluya al origen creado. Estos elementos deberán describirse en el archivo de configuración XML del servicio, disponible en `mapserver/bind-mounts/mapcache/config.xml`. Téngase en cuenta que, al editar este archivo, la ruta `mapserver/bind-mounts/mapcache/` se expone en el contenedor, desde la raíz de su sistema de archivos, como `/mapcache/`. En cuanto a la sintaxis específica a emplear en este archivo, si bien pueden usarse los elementos ya definidos como ejemplo, viene especificada en la documentación del programa ².

¹<https://mapserver.org/mapfile/index.html>

²<https://mapserver.org/mapcache/config.html>

1.4 Generación de copia caché (semillado) del nuevo mapa

Añadidos los elementos necesarios a la configuración de MapCache para que este pueda manejar la copia caché, es necesario generarla. Ello se puede hacer manualmente mientras se ejecuta el contenedor que aloja a MapServer y MapCache y usando la herramienta que el segundo aporta para ello, adjuntando los parámetros adecuados al final del siguiente comando:

```
$ docker exec -it mapserver mapcache_seed
```

Aunque tal opción permite ajustar manualmente los ajustes del proceso de semillado cada vez, también se puede recurrir a modificar el script creado para un semillado lo más rápido y completo posible, disponible en:

```
/mapserver/bind-mounts/mapcache/mapcache_seed_cache.sh
```

Posteriormente proceda a su uso mediante el siguiente comando:

```
$ docker exec -it mapserver /mapcache/mapcache_seed_cache.sh
```

1.5 Incorporación del mapa creado al cliente web

Generada la copia caché, queda incluir el mapa a mostrar en los ajustes del cliente web para que se muestre a los usuarios, para lo que se deben modificar dos archivos.

El primero de ellos, **webapp/.env**, es un archivo de texto plano que contiene la definición de diversas constantes, y para facilitar el mantenimiento incluye constantes que aportan las URL de acceso a las teselas que se deseen mostrar a los usuarios que accedan al cliente, mínimamente parametrizadas para que en tiempo de ejecución se acceda a la tesela correcta según sea necesario. Por ello se deberá crear una nueva constante en el archivo que indique la URL parametrizada con la que acceder a las nuevas teselas que se desee mostrar.

El otro archivo a modificar, en **webapp/src/Map/Map.jsx**, incluye el código JavaScript que se ejecuta en los equipos de los clientes que acceden para manejar y cargar los mapas y las capas a mostrar en sus equipos. En él debe añadirse:

- Una nueva constante que referencie el mapa que se incorpora, pudiéndose tomar el contenido de la constante **OSMTiles** como ejemplo.
- Añadir la nueva constante creada a la lista **layers** de la constante **mapParams**.

- Si se quiere que el mapa que se añade sea una base sobre la que se puedan superponer otros mapas creados aprovechando el uso de transparencias, incluir la constante creada en el punto primero a la constante `baseMaps` junto con un texto que permita a los usuarios identificarlo.

Realizados estos cambios, el mapa dispuesto podrá consultarse mediante el cliente web.

2 Cambio de URL de acceso a los servicios

Con la configuración por defecto de los servicios, aunque todos son accesibles desde cualquier lugar, están configurados para su uso local desde el anfitrión de los contenedores y no a través de red alguna. Por ello, deberán reconfigurarse una vez se conozca la dirección IP que vaya a tener la máquina en producción, ya sea disponiendo una pública para que funcione a través de Internet, o disponiendo una de alguno de los espacios privados de uso público para limitar su uso a una red privada.

Un similar ocurre si se cambiara el puerto de acceso a alguno de los servicios, ya que la URL de estos cambia.

2.1 Cliente web

Dado que el cliente web tiene por función principal conectar los distintos servicios desplegados facilitando su uso a los usuarios finales, es el primero que necesitará modificaciones cuando deba cambiar la URL de acceso a cualquiera de los servicios.

Estos cambios deben, primero, consignarse en el archivo de variables `webapp/.env`, modificándolo con arreglo a la siguiente tabla de asociaciones:

Aplicación	Variable a modificar
MapCache	<code>REACT_APP_MAPCACHE_URL</code>
Nominatim	<code>REACT_APP_NOMINATIM_URL</code>
Valhalla	<code>REACT_APP_VALHALLA_URL</code>

Table 1: Correspondencia entre variables del cliente web y servicios involucrados.

Una vez establecidas las URL correctas, es *necesario* recompilar el contenedor del cliente web para aplicar los cambios:

```
$ docker compose build webapp
$ docker compose down webapp ; docker compose up webapp
```

3 Traslado de PostgreSQL a otra máquina externa

Como los sistemas de gestión de bases de datos en toda organización ocupan un rol central, y suelen ser de los más regulados, no es extraño que la instancia de PostgreSQL a usar deba residir en otra máquina.

A continuación se indican los pasos a seguir para el uso de bases de datos PostgreSQL externas.

Este procedimiento sólo es aplicable en instalaciones ya funcionales que necesiten cambiar la ubicación de su servidor de PostgreSQL. Para una nueva instalación con un servidor de PostgreSQL separado, consulte el apartado 4

3.1 Traslado de información

Como en toda migración de bases de datos, lo primero a hacer, *si ya hubieran bases de datos en uso*, es volcar toda la información preexistente al nuevo servidor, para lo que es *indispensable* asegurar que no haya acciones de escritura en curso sobre la base de datos y que no las habrá hasta completada la migración, siendo la forma más sencilla de asegurar tal condición es deteniendo los otros servicios que la usan.

```
$ docker compose down nominatim mapserver
```

No obstante, dado que unos y otros servicios del sistema interactúan entre sí y se necesitan unos a otros para un correcto funcionamiento, *se recomienda*³, *aunque no es necesario, detener todos los servicios a excepción del gestor de bases de datos, liberando más recursos y expeditando la migración:*

```
$ docker compose down nominatim mapserver valhalla webapp
```

Llegado este punto, la forma más recomendable de proceder es mediante el volcado de usuarios y bases de datos con la utilidad `pg_dump`:

```
$ pg_dumpall -d "postgresql://root:password@127.0.0.1:5432/" --roles-only | \
  grep -v "ROLE root" > sql_dump_roles.sql
$ for db in mapserver_osm nominatim ; do \
  pg_dump -c -C -F p -f sql_dump_${db}.sql --if-exists \
  --quote-all-identifiers --serializable-deferrable \
  -d "postgresql://root:password@127.0.0.1:5432/${db}" ; done
```

³Considérese que, con toda probabilidad, se ha dispuesto una ventana de mantenimiento al detener servicios que aportan funcionalidades críticas del sistema.

Una vez obtenidos los volcados, se puede detener el contenedor de PostgreSQL:

```
$ docker compose down postgis
```

Aunque los volcados pueden importar desde la misma máquina a la nueva instancia de PostgreSQL, *es más recomendable transferir los volcados primero a la que la aloje*⁴, e importarlos desde esta otra máquina, lo que acelera el proceso ya que, por lo general, la red suele hacer de cuello de botella.

Si se han usado los comandos anteriores para el volcado, la restauración deberá hacerse usando el cliente de PostgreSQL, `psql`, *ajustando los parámetros de conexión en las URL de los siguientes comandos*:

```
$ psql -d "postgresql://root:password@127.0.0.1:5432/" < sql_dump_roles.sql
$ for db in mapserver_osm nominatim ; do \
    psql -d "postgresql://root:password@127.0.0.1:5432/" < sql_dump_${db}.sql ; \
done
```

Una vez completados estos últimos comandos, el nuevo servidor está listo para tomar el relevo al anterior.

3.2 Cambio de servidor de base de datos en Nominatim

Preparada la nueva base de datos, se ha de cambiar la conexión de Nominatim a esta, lo que se consigue modificando la variable `NOMINATIM_DATABASE_DSN` con los datos del nuevo servidor de PostgreSQL en el archivo `docker-compose.yml`.

Posteriormente deberá restablecerse el servicio, lo que se aborda más adelante en el apartado 3.4.

3.3 Cambio de servidor de base de datos en MapServer

Siendo otro servicio que usa las bases de datos migradas, se han de hacer ajustes a MapServer para que use la instancia de PostgreSQL a la que se han migrado los datos, lo que puede hacerse regenerando los archivos de configuración de mapa con los ajustes correctos (para los creados mediante basemaps) y/o modificando uno a uno los archivos de configuración de mapa que se hayan creado y que hagan uso de las bases de datos a migrar.

⁴El método de transferencia queda en manos del lector-administrador, usando la forma que estime más conveniente, si bien hay utilidades especialmente diseñadas al efecto como `rsync`

3.3.1 Mediante regeneración de archivos de configuración

Dado que el script de creación de configuraciones de mapa incluido en basemaps está altamente parametrizado, se puede ajustar el origen de datos cambiándolo en el archivo `basemaps/settings.mk` y recreando los archivos de configuración de mapa obtenidos a su través, conforme a las instrucciones del apartado .

3.3.2 Mediante modificaciones manuales

Otra forma de realizar las modificaciones planteadas es modificar uno a uno los archivos de configuración de mapa que se hayan creado y que hagan uso de bases de datos. Localizarlos es sencillo, ya que son aquellos en los que aparece la palabra clave `CONNECTION`. Conociendo esto, si se han ido guardando los diversos archivos de configuración de mapa en una misma carpeta, listar sólo los que necesitan modificaciones es posible con el siguiente comando, ejecutado dentro de la misma carpeta:

```
$ grep -lrF 'CONNECTION'
```

Una vez conocidos los ficheros a revisar y/o modificar, basta con modificar adecuadamente los ajustes de conexión, que se exponen en la misma línea que la mencionada palabra clave justo después de ella.

A modo de ejemplo, véase el siguiente espécimen de definición de conexión a una base de datos PostgreSQL:

```
CONNECTION "host=postgis dbname=mapserver_osm user=mapserver password=mapserver
port=5432"
```

Téngase en cuenta que en un mismo archivo puede definirse más de una vez conexiones a la misma base de datos, por lo que deben revisarse todas las ocurrencias de la palabra clave “CONNECTION” en cada archivo detectado.

3.4 Restablecimiento de los servicios

Llegado este punto, se pueden reiniciar todos los contenedores a excepción del de PostgreSQL⁵, lo que se logra con los siguientes comandos:

```
$ docker compose up nominatim mapserver valhalla webapp
```

⁵Obviamente, el (re)inicio del servidor PostgreSQL correspondiente queda a la mejor razón del lector-administrador.

3.5 (Opcional) Modificación del script de creación de bases de datos y usuarios

A fin de mantener la mera consistencia entre todos los elementos ante el cambio de servidor, se recomienda modificar la URL incluida en el script de creación de bases de datos y usuarios para que apunte al nuevo servidor, en caso de que en algún momento en el futuro se necesitare usar.

Concretamente, se trata de cambiar una línea del fichero `postgis/sql_skel_create.sh` para que refleje el servidor usado:

```
CONNECTION_STRING='postgresql://root:password@127.0.0.1:5432/'
```

4 Uso de una instancia de PostgreSQL en otra máquina

De desear cambiar el servidor de PostgreSQL usado en una nueva instalación antes de proceder a la misma, han de seguirse las instrucciones indicadas mas arriba para el cambio de base de datos de Nominatim y seguir también, según apliquen, las del cambio de servidor de base de datos en MapServer, indicadas antes. Después puede seguirse el conjunto de procedimientos del manual de despliegue, teniendo en cuenta las advertencias sobre los cambios de los parámetros de bases de datos.