MxSIG

Índice

[Software necesario para el funcionamiento del MxSIG 3](#_Toc57835658)

[Herramientas/librerías necesarias a instalar inicialmente 3](#_Toc57835659)

[Proceso para poner en marcha el MxSIG 4](#_Toc57835660)

[1. PostgreSQL 4](#_Toc57835661)

[2. PostGIS 7](#_Toc57835662)

[3. APACHE 9](#_Toc57835663)

[4. MapServer 11](#_Toc57835664)

[5. Java 15](#_Toc57835665)

[6. Tomcat 16](#_Toc57835666)

[7. MxSIG 20](#_Toc57835667)

[Procedimiento para subir archivos (SHP) 24](#_Toc57835668)

[Anexos 27](#_Toc57835669)

[Problemas actuales en el servidor 27](#_Toc57835670)

# Software necesario para el funcionamiento del MxSIG

Servidor Linux con las siguientes herramientas instaladas:

* POSTGRESQL
* POSTGIS
* MAPSERVER
* APACHE (Servidor Web)
* TOMCAT

# Herramientas/librerías necesarias a instalar inicialmente

* GCC (compilador de C)
* GCC-C++ (compilador de C++)
* CMake (cmake)
* SQLite (sqlite-devel)
* PROJ (proj-devel)
* LibJasper (libjasper-devel)
* SFCGAL (SFCGAL-devel)
* Libxml (libxml2-devel)
* Libpng (libpng-devel)
* FreeType (freetype-devel)
* cURL (libcurl-devel)
* Giflib (giflib-devel)
* Cairo (cairo-devel)
* FastFCGI (mod-fcgi y fcgi-devel)
* GeoTIFF (libgeotiff16)
* GEOS (geos-devel)
* Libpq (libpq-devel)
* PHP

# Proceso para poner en marcha el MxSIG

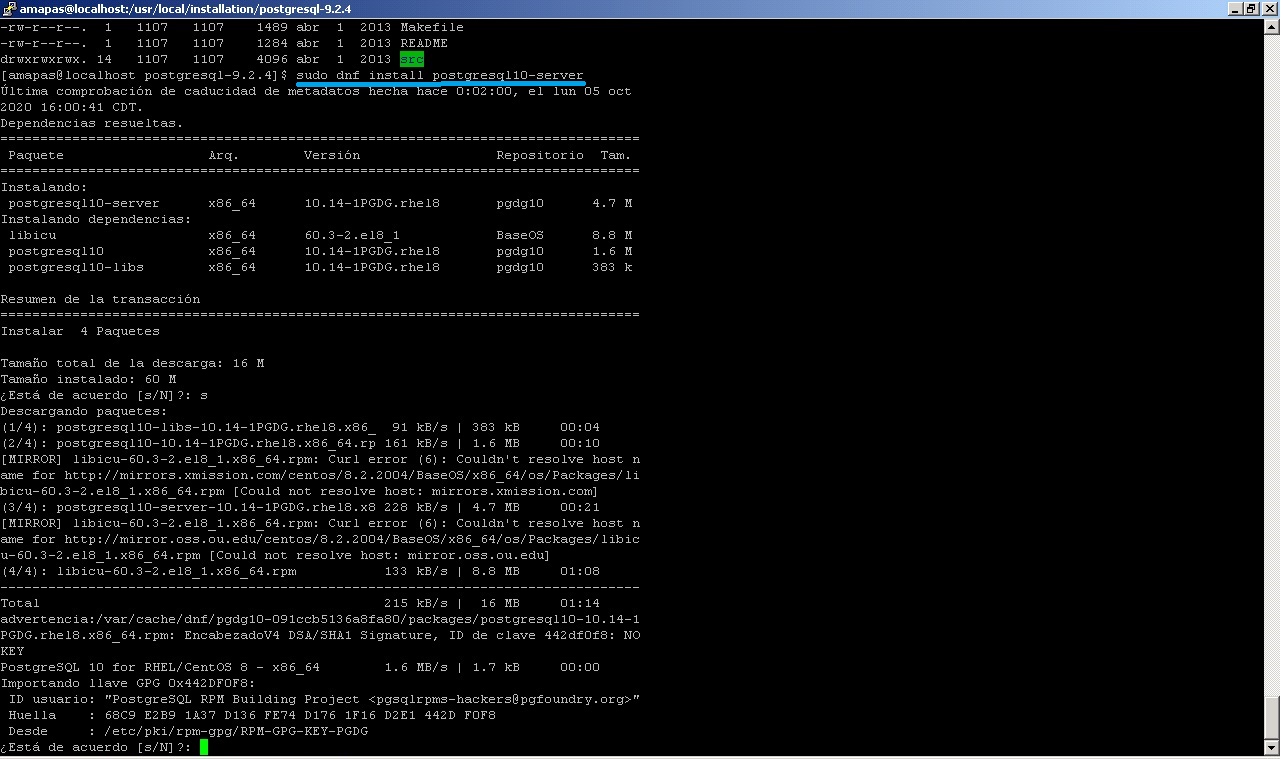
1. Instalar el sistema operativo **CentOS 8**, en una partición del sistema.
2. Para trabajar a distancia, configurar un puerto SSH para la configuración del servidor = 3022
3. A través del programa **Putty**, se realizará la configuración del servidor.
4. Crear un usuario diferente a root: *amapas,* para mayor seguridad.
   1. Configurar a ese usuario para que use comandos root.

## PostgreSQL

1. Instalar **PostgreSQL 10:**

*# sudo dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86\_64/pgdg-redhat-repo-latest.noarch.rpm*

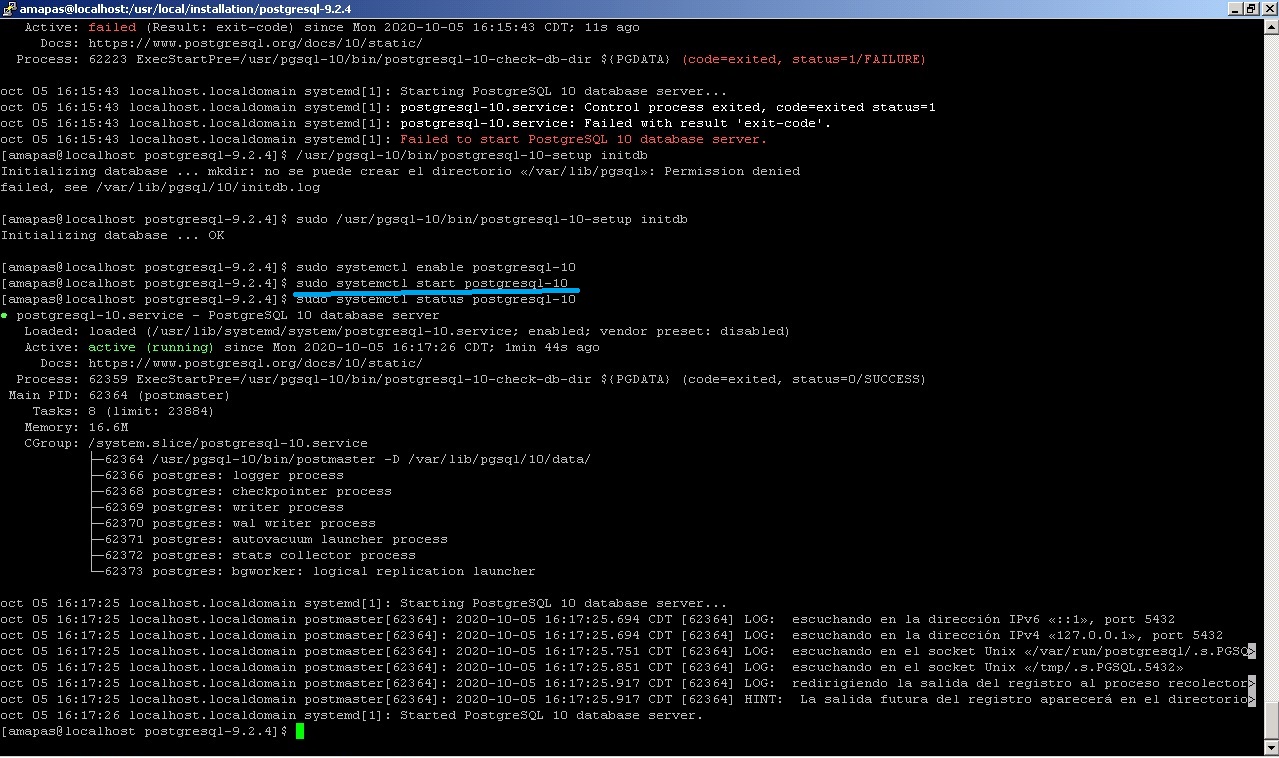
*# dnf -qy module disable postgresql*

*# dnf install postgresql10-server*

1. Habilitar y ejecutar PostgreSQL:

*# /usr/pgsql-10/bin/postgresql-10-setup initdb*

*# systemctl enable postgresql-10*

*# systemctl start postgresql-10*

1. Crear un **usuario** y una **contraseña** para manipular PostgreSQL.

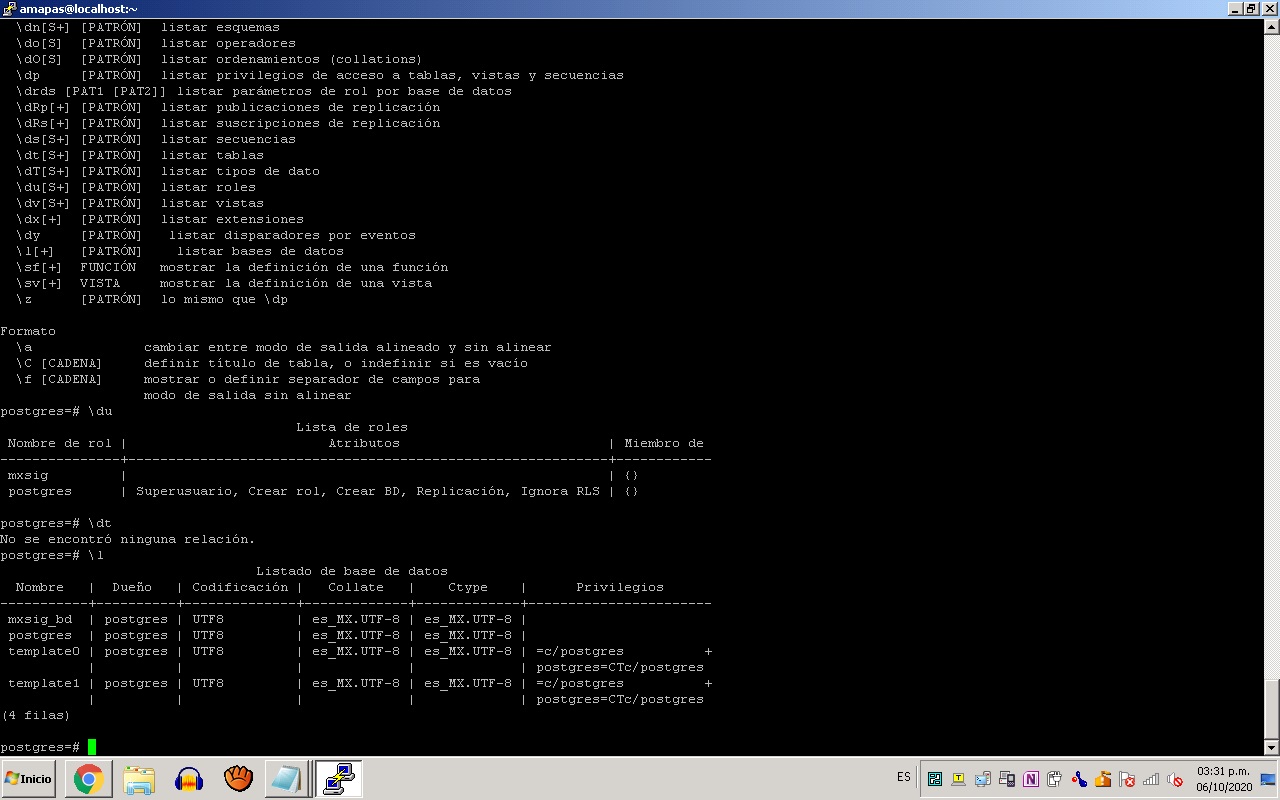
*# sudo passwd postgres*

Contraseña: Colhid20

1. Crear la base de datos que contendrá los datos de los mapas.

*# createdb mxsig\_bd*

1. Despues verificamos que se haya creado la base de datos:



1. Ahora, configurar archivos para la **conexión remota** de la base de datos:

*# nano /var/lib/pgsql/10/data/postgresql.conf*

Modificar las mismas líneas de código:

*listen\_addresses = '\*'*

*port 5444*

*# nano /var/lib/pgsql/10/data/pg\_hba.conf*

Agregar al final siguiendo el espaciado:

*host all all 0.0.0.0/0 md5*

1. Aplicar los cambios a los archivos:

*# sudo systemctl restart postgresql-10*

1. **Habilitar** el puerto en el **firewall** para evitar bloqueo de la conexión:

*# sudo firewall-cmd --permanent --add-port=5444/tcp*

*# sudo firewall-cmd –reload*

1. Podemos ver que se configuró correctamente con el siguiente comando:

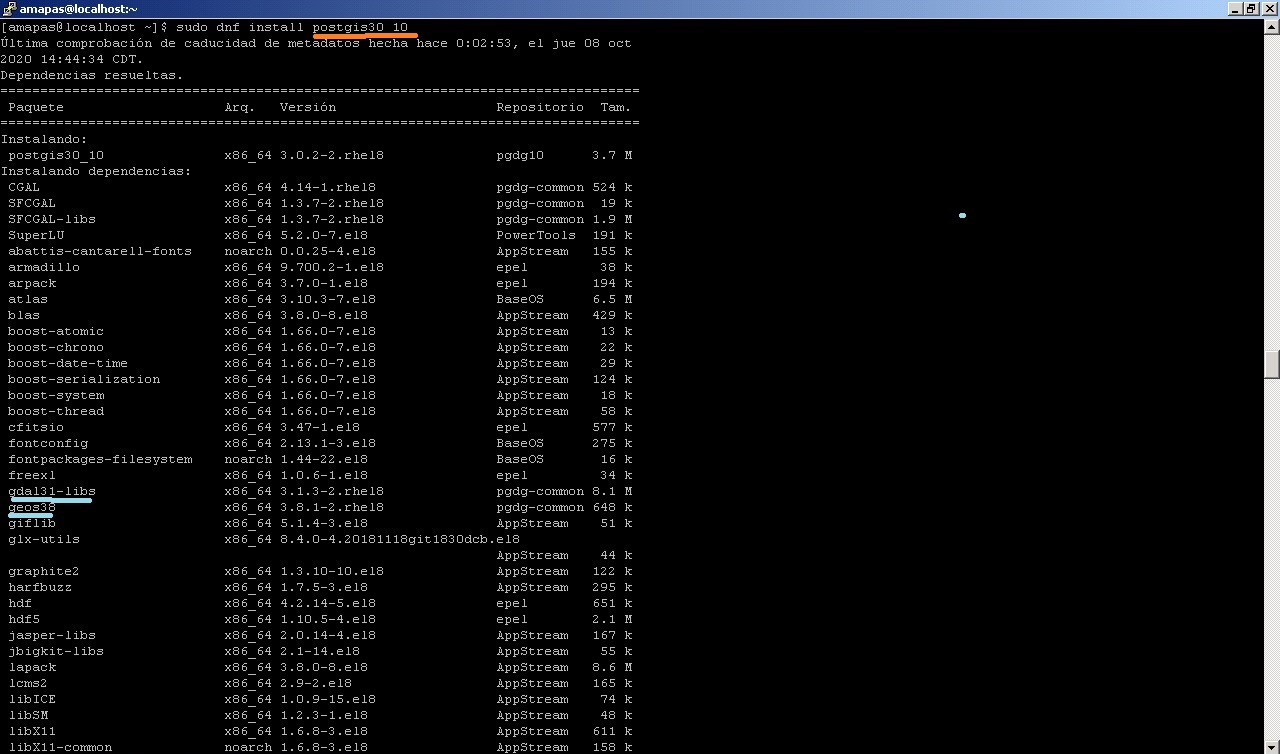
*# sudo firewall-cmd --list-all*

## PostGIS

1. Instalar **PostGIS:**

*# sudo dnf install epel-release*

*# sudo dnf config-manager --set-enabled PowerTools*

*# sudo dnf install postgis30\_10-devel*

1. **Habilitar PostGIS** en la base de datos ya hecha.

*# psql*

*# createlang plpgsql mxsig\_bd*

*# psql mxsig\_bd*

*# CREATE EXTENSION postgis;*

*# CREATE EXTENSION postgis\_raster;*

*# CREATE EXTENSION postgis\_topology;*

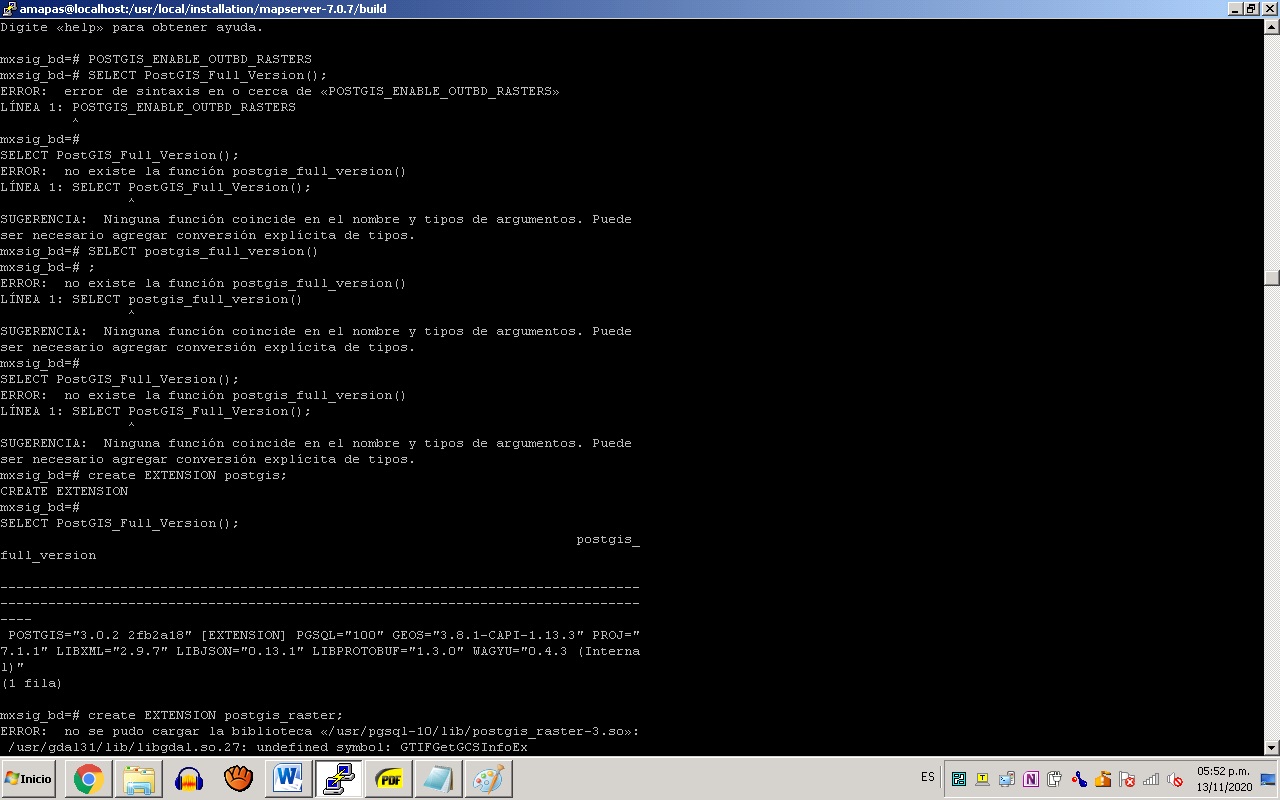
*# CREATE EXTENSION postgis\_sfcgal;*

*# \q*

*# POSTGIS\_ENABLE\_OUTDB\_RASTERS=1*

1. Luego podemos verificar la configuración de PostGIS con el siguiente comando:

*# SELECT PostGIS\_Full\_Version();*



1. Y creamos un **rol** para **manejar** la base de datos a distancia.

*# CREATE ROLE mxsig with LOGIN;*

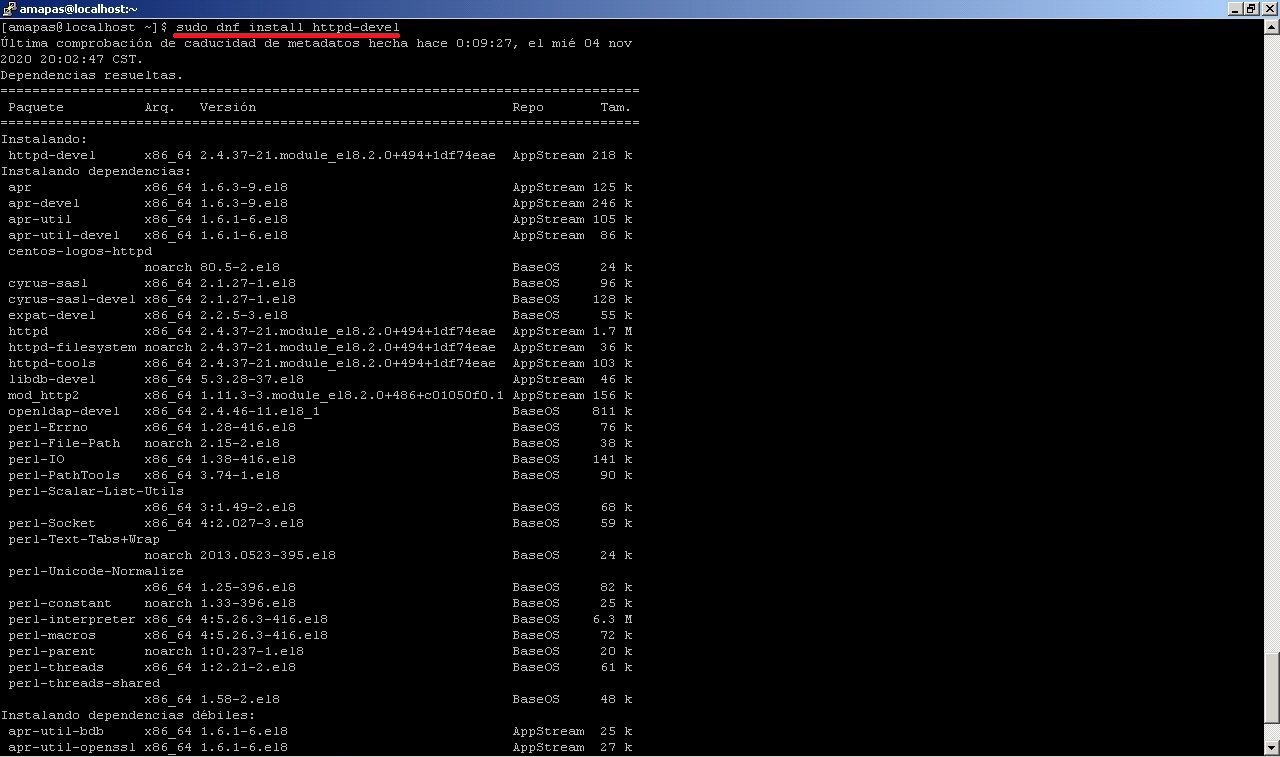
*# \password mxsig*

*# psidetuh*

## APACHE

1. Ahora a instalar el **servidor web Apache:**

*# sudo dnf install httpd-devel*



1. Habilitar el servicio en el **firewall** para **permitir** la entrada de **tráfico**:

*# sudo firewall-cmd --permanent --add-service=http*

*# sudo firewall-cmd --reload*

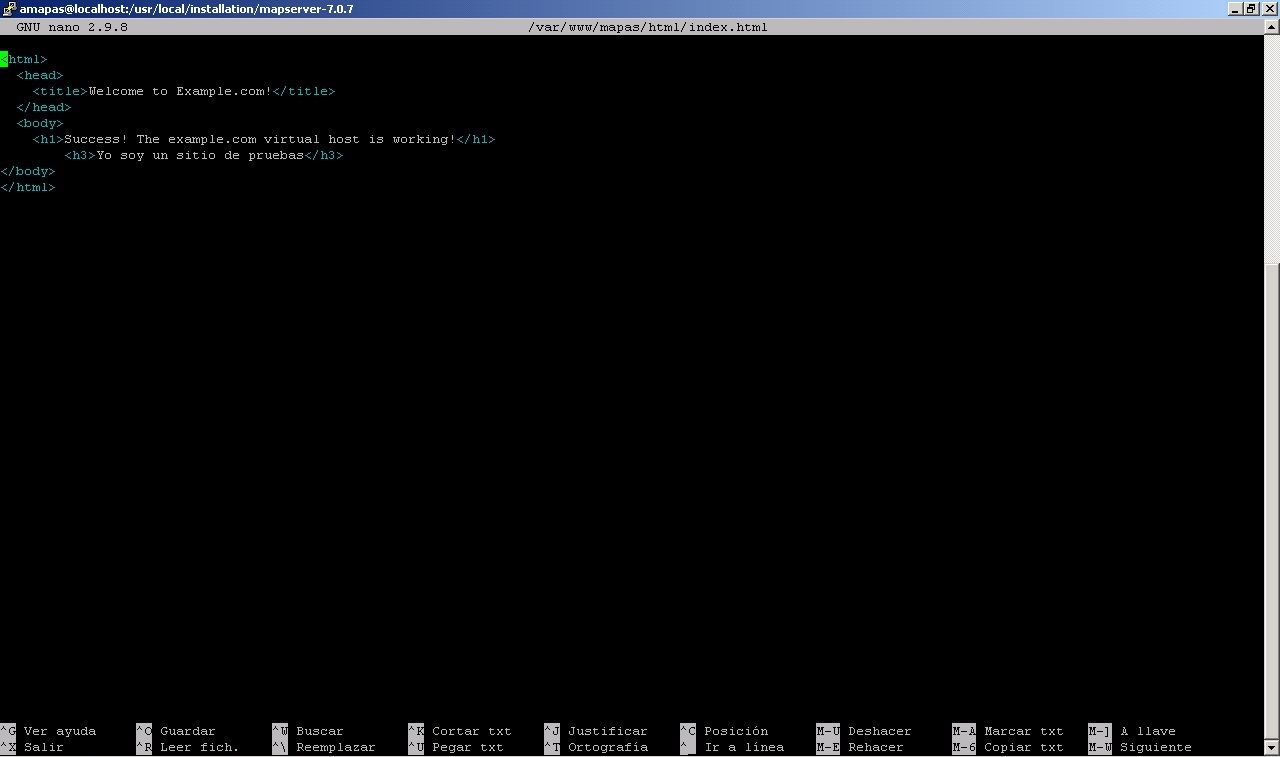
*# sudo firewall-cmd --list-all*

1. **Iniciar** y **habilitar** permanentemente **Apache.**

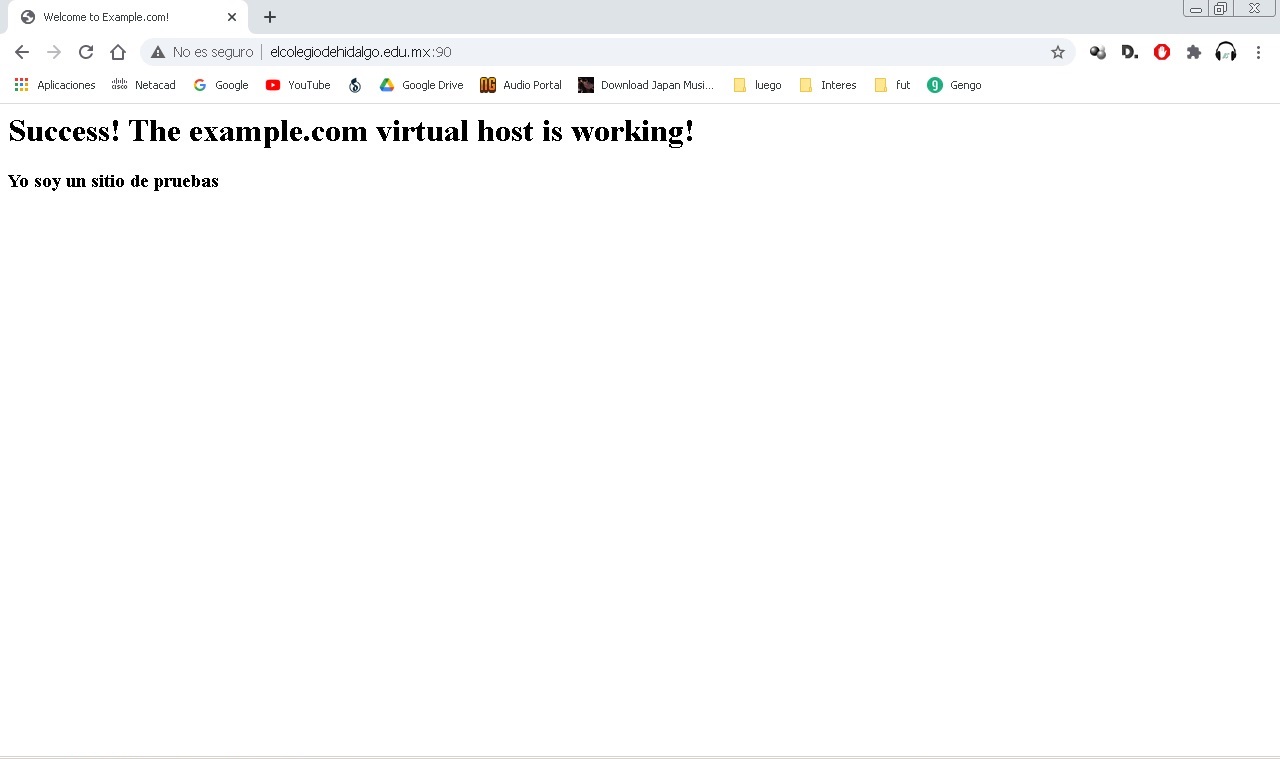
*# sudo systemctl start httpd*

*# sudo systemctl enable httpd*

1. Luego, creamos una página para **probar el funcionamiento** del servidor web.



1. La probamos en el navegador con esta dirección: ***elcolegiodehidalgo.edu.mx:90***



## MapServer

Antes de instalar MapServer, instalaremos la **librería GDAL** manualmente, ya que se necesita de la utilería **ECW/JP2**, que no viene por defecto en GDAL ni en algún repositorio de CentOS.

1. Primero instalaremos la utilería con algunos parches de mejora:

*# sudo wget https://github.com/bogind/libecwj2-3.3/raw/master/libecwj2-3.3-2006-09-06.zip*

*# sudo unzip libecwj2-3.3-2006-09-06.zip*

*# wget http://trac.osgeo.org/gdal/raw-attachment/ticket/3162/libecwj2-3.3-msvc90-fixes.patch*

*# sudo dnf install patch*

*# patch -p0< libecwj2-3.3-msvc90-fixes.patch*

*# wget http://osgeo-org.1560.x6.nabble.com/attachment/5001530/0/libecwj2-3.3-wcharfix.patch*

*# wget http://trac.osgeo.org/gdal/raw-attachment/ticket/3366/libecwj2-3.3-NCSPhysicalMemorySize-Linux.patch*

*# cd libecwj2-3.3*

*# sudo patch -p0< ../libecwj2-3.3-NCSPhysicalMemorySize-Linux.patch*

*# sudo patch -p1< ../libecwj2-3.3-wcharfix.patch*

*# sudo dnf install make*

*# sudo ./configure*

*# sudo make*

*# sudo make install*

1. Después **compilaremos GDAL:**

*# sudo wget http://download.osgeo.org/gdal/3.1.1/gdal-3.1.1.tar.gz*

*# sudo tar -xvzf gdal-3.1.1.tar.gz*

*# cd gdal-3.1.1*

*# sudo ./configure --with-ecw=/usr/local*

*# make*

*# make install*

1. Comprobar que la utilería fue instalada:

*# gdalinfo --formats | grep ECW*

1. Ahora es momento de **instalar MapServer:**

*# sudo wget http://download.osgeo.org/mapserver/mapserver-7.0.7.tar.gz*

*# sudo tar xzvf mapserver-7.0.7.tar.gz*

*# cd mapserver-7.0.7*

*# sudo mkdir build*

*# cd build*

*# chown amapas mapserver-7.0.7*

1. En el siguiente apartado es donde **configuramos** como queremos que se instale **MapServer.**

*# cmake -DCMAKE\_INSTALL\_PREFIX=/usr -DCMAKE\_PREFIX\_PATH=/usr/lib64 -DWITH\_CLIENT\_WFS=ON -DWITH\_CLIENT\_WMS=ON -DWITH\_CURL=ON -DWITH\_FRIBIDI=0 -DWITH\_HARFBUZZ=0 -DWITH\_ORACLESPATIAL=0 -DWITH\_SVGCAIRO=0 -DWITH\_GIF=0 ../ > ../configure.out.txt*

1. Nos da **problemas PROJ**, así que modificaremos el archivo para que podamos usarlo sin problemas:

*# sudo nano /usr/include/proj\_api.h*

1. Añadiremos al archivo la línea de código después de los comentarios: *define ACCEPT\_USE\_OF\_DEPRECATED\_PROJ\_API\_H*



*# sudo make*

*# sudo make install*

1. Para comprobar que funciona correctamente, ejecutar el siguiente comando:

*# ./mapserv*

El comando anterior debe de mostrar el siguiente mensaje: *this script can only be used to decode form results and should be initiated as a CGI process via a httpd server.*

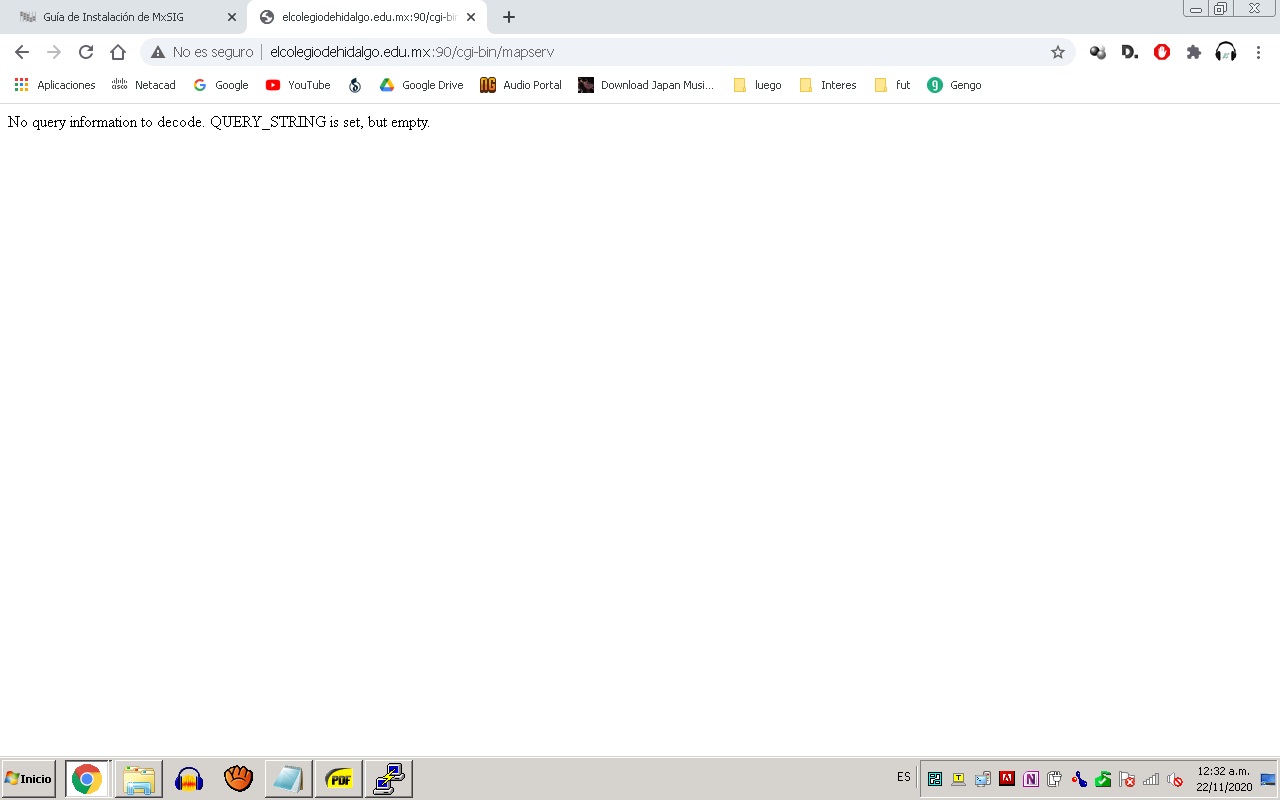
1. Para que se pueda **usar MapServer** en el servidor **web**, copiaremos el archivo *mapserv* a una carpeta especial llamada CGI-Bin:

*# sudo cp mapserv /var/www/cgi-bin*

1. Podemos ver la configuración de MapServer con el siguiente comando:

*# /var/www/cgi-bin/./mapserv -v*

1. Y ahora tecleando la URL ***elcolegiodehidalgo.edu.mx:90/cgi-bin/mapserv/*** nos mostrara que funciona MapServer adecuadamente.



1. Ahora se crea la siguiente estructura de carpetas:

*# sudo mkdir –p /opt/logs/mapserver*

*# sudo chmod -R 777 /opt/logs*

1. Y se crea un archivo:

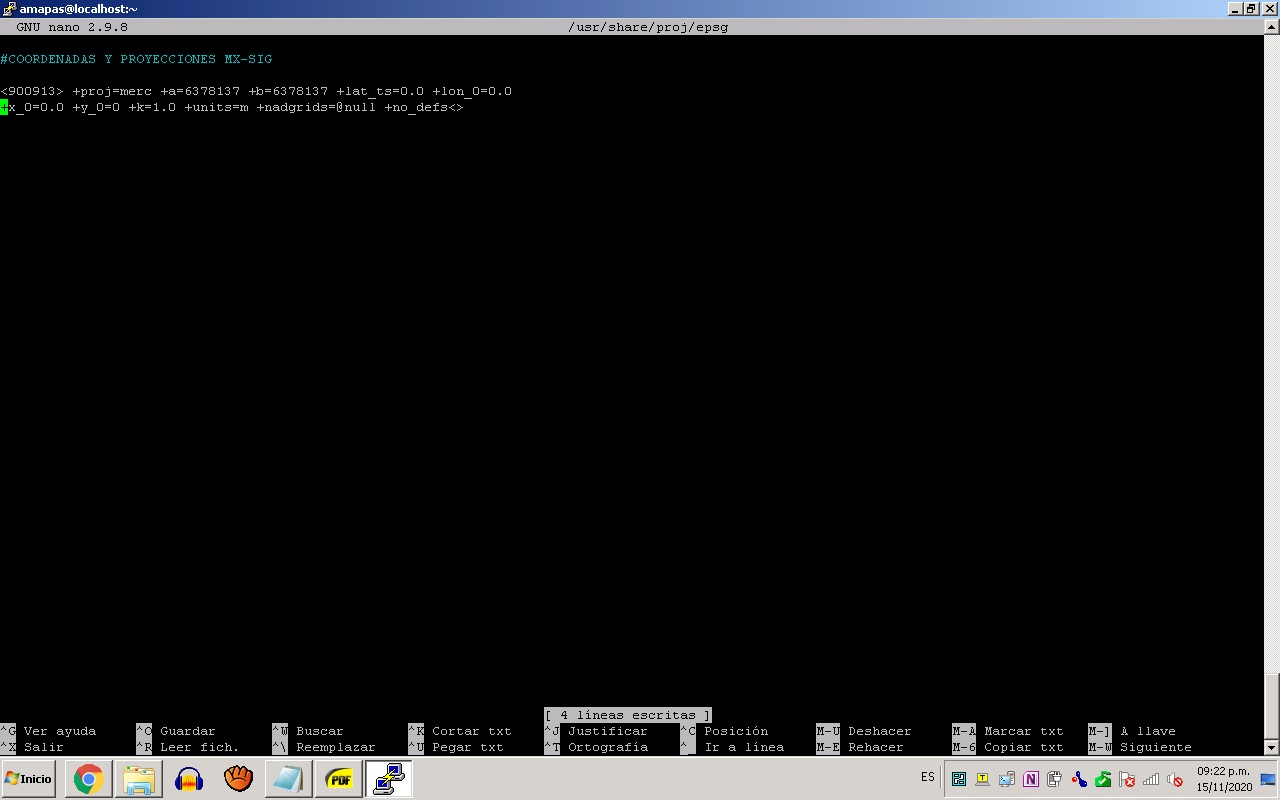
*# sudo nano /opt/logs/mapserver/mdm6.log*

*# sudo systemctl restart httpd*

1. Por último, se define el **sistema de coordenadas y proyecciones** para la visualización de mapas:

*# sudo nano /usr/share/proj/epsg*

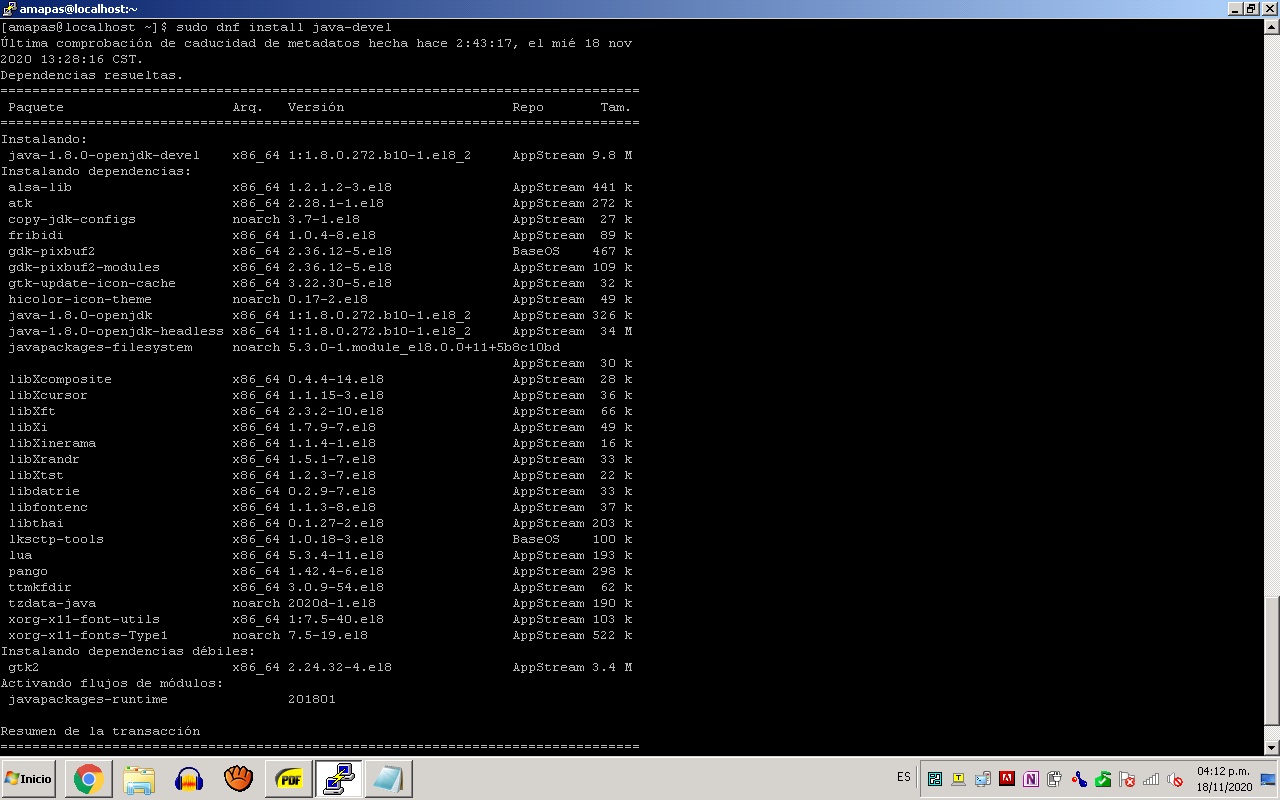
*<900913> +proj=merc +a=6378137 +b=6378137 +lat\_ts=0.0 +lon\_0=0.0 +x\_0=0.0 +y\_0=0 +k=1.0 +units=m +nadgrids=@null +*no\_defs<>\*



## Java

1. Instalar **Java:**

*# sudo dnf install java-devel*



1. Ver donde se encuentra instalado Java:

*# sudo alternatives --config java*

*# nano .bashrc*

1. Ahora configuramos unas **variables globales**, para que los programas detecten Java sin problemas.

Añadir al final la siguiente línea:

*JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.272.b10-1-el8\_2.x86\_64/jre/bin/java*

1. Y guardamos la configuración.

*# source .bashrc*

## Tomcat

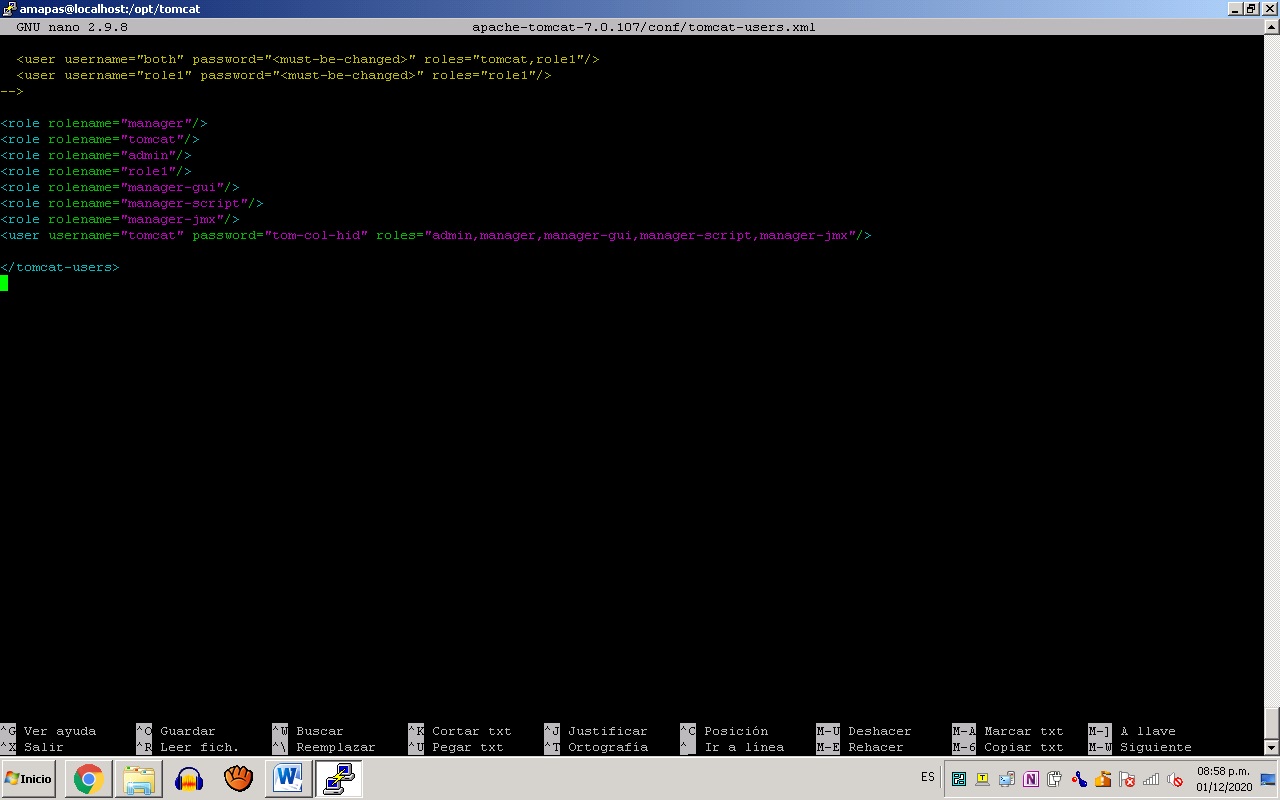
1. Instalar **Tomcat:**

*# sudo wget https://downloads.apache.org/tomcat/tomcat-7/v7.0.107/bin/apache-tomcat-7.0.107.tar.gz*

*#sudo tar -xvf apache\*tar.gz*

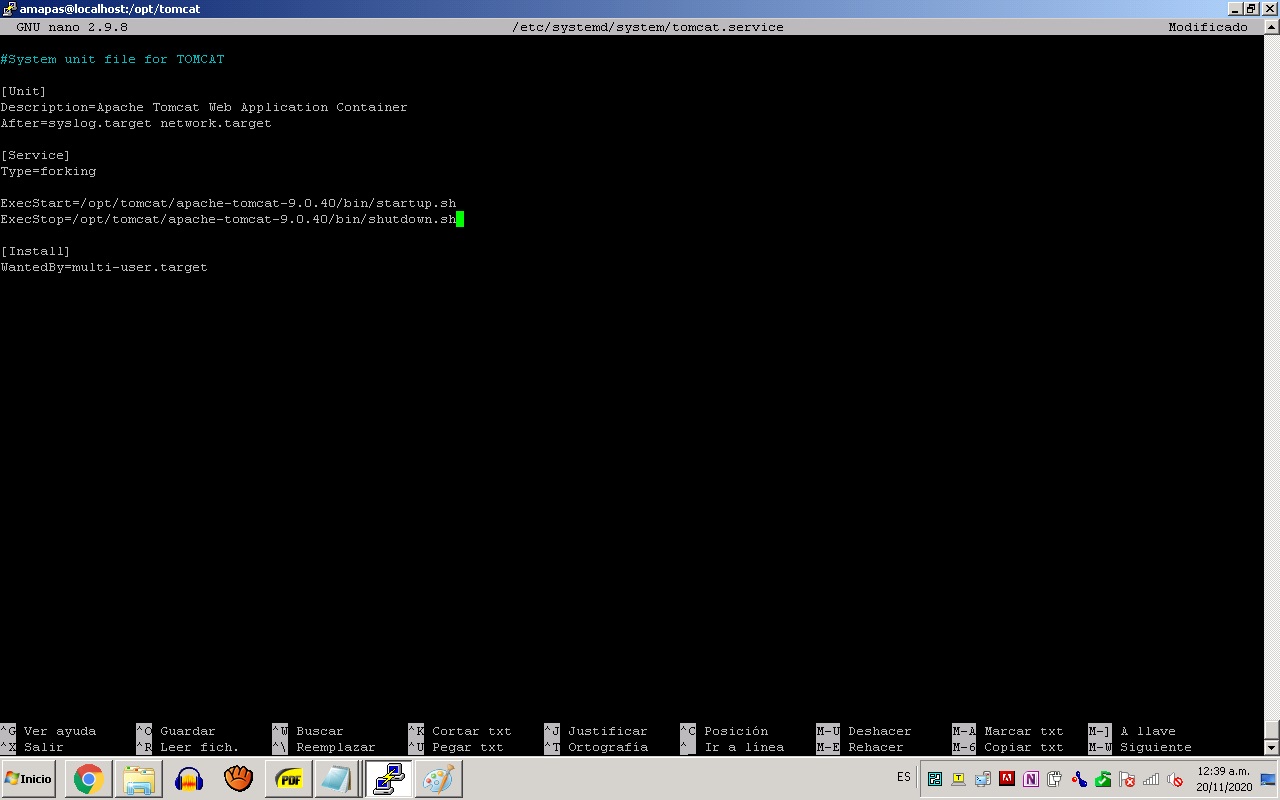
1. Configurar el archivo de los **roles de Tomcat:**

*#sudo nano apache-tomcat-9.0.40/conf/tomcat-users.xml*



*# sudo chown amapas -R /opt/tomcat*

1. Para que sea más fácil **manipular Tomcat,** se ejecutara como un **servicio:**

*# sudo nano /etc/systemd/system/tomcat.service*

*# sudo systemctl daemon-reload*

*# sudo systemctl start tomcat*

*# sudo systemctl enable tomcat*

1. Para la **conexión Apache-Tomcat,** descargaremos e instalaremos un conector para ese fin:

*# sudo wget https://downloads.apache.org/tomcat/tomcat-connectors/jk/tomcat-connectors-1.2.48-src.tar.gz*

*# sudo tar -xvf apache\**

*# sudo dnf install redhat-rpm-config*

*# cd tomcat-connectors-1.2.48-src/native*

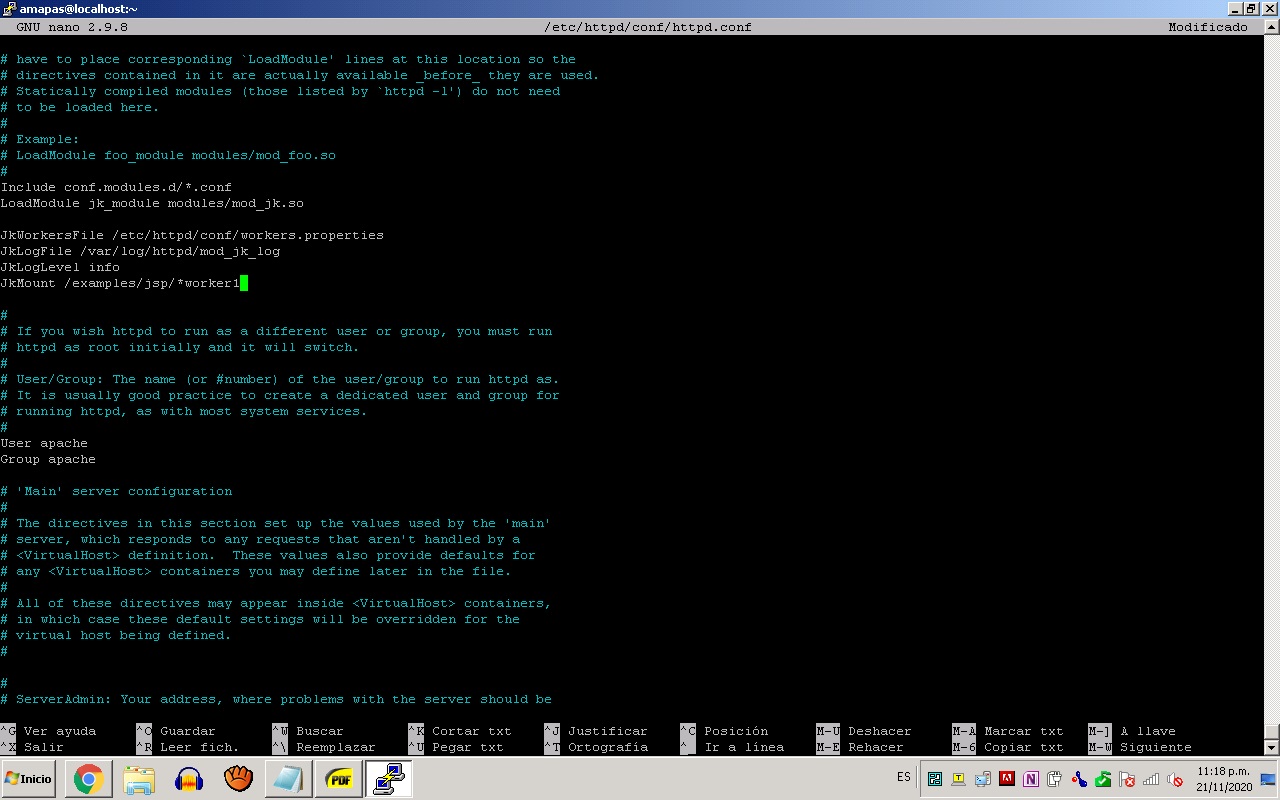
*# sudo ./configure --with-apxs=/usr/bin/apxs*

*# sudo make*

*# sudo cp ./apache-2.0/mod\_jk.so /usr/lib64/httpd/modules*

1. **Configurar** el **conector** para que funcione correctamente:

*# sudo nano /etc/httpd/conf/httpd.conf*

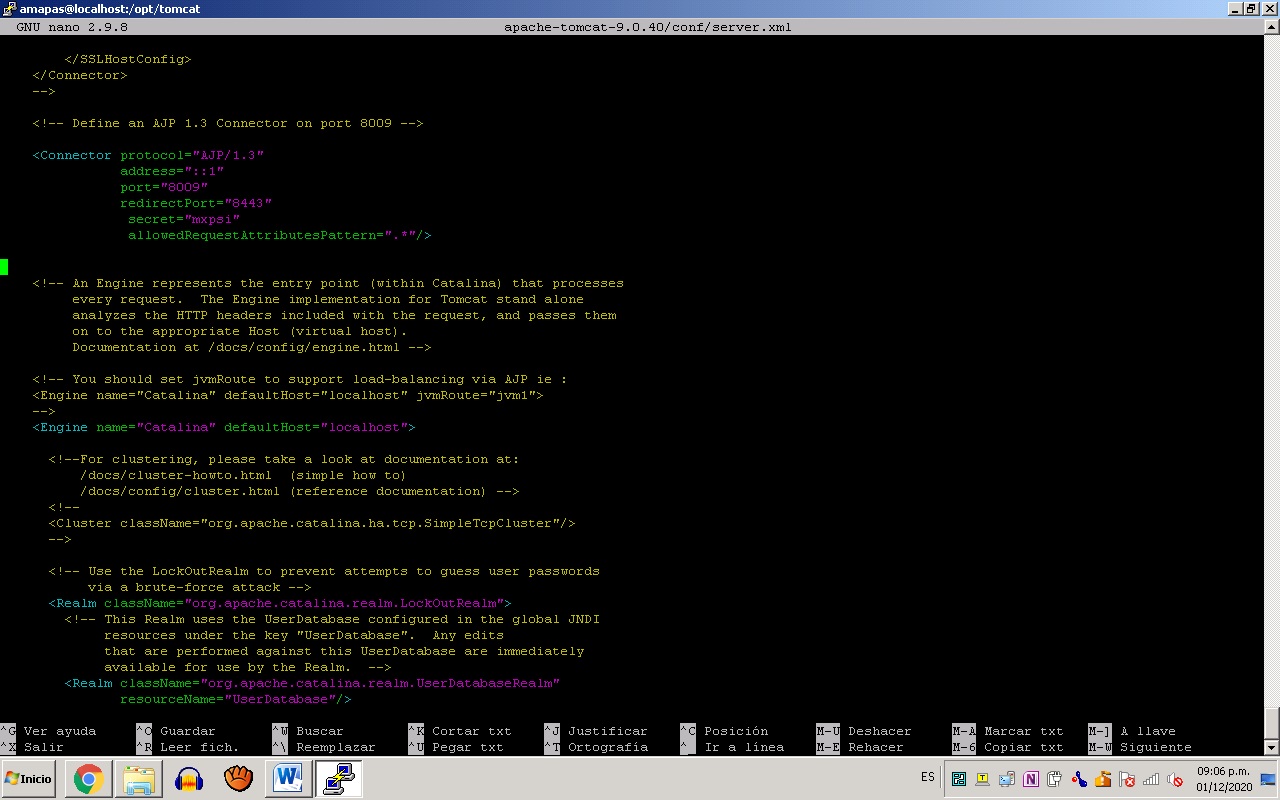


*# sudo nano /etc/httpd/conf/workers.properties*



1. Descomentaremos unas líneas para que se acepte el **protocolo** que conectara Apache y Tomcat: **AJP**

*# sudo nano /opt/tomcat/apache-tomcat-9.0.40/conf/server.xml*



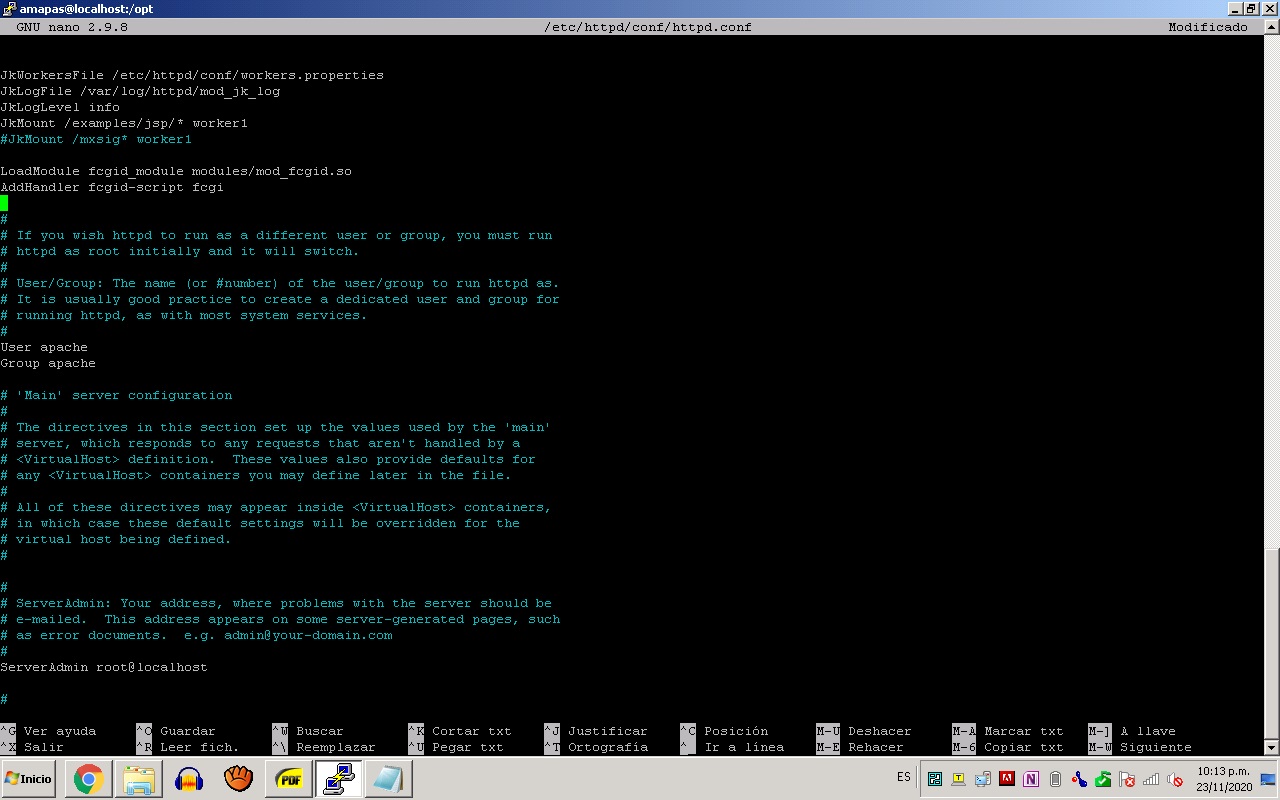
*# sudo systemctl restart tomcat*

*# sudo systemctl restart httpd*

1. Configurar el **módulo fcgi:**

*# sudo mkdir -p /tmp2/fcgi/dynamic*

*# sudo nano /etc/httpd/conf/httpd.conf*



*# sudo systemctl restart httpd*

## MxSIG

1. Ahora **descargamos** de la página del Inegi el software libre **MxSIG:**

*# sudo wget https://www.inegi.org.mx/servicios/doc/MxSIG.zip*

*# sudo unzip MxSIG.zip -d /var/www/html*

1. **Duplicar** algunas carpetas para evitar que no se carguen algunos archivos en la página web:

*# cd /var/www/mapas/html/mxsig/js/frameworks/*

*# sudo ln -s openlayers Openlayers*

*# cd ../core/ui/widgets*

*# sudo cp -R fancybox fancyBox*

1. Configurar el **archivo WAR** para su ejecución en Tomcat:

*# cp /var/www/html/Tabl\* /opt/tomcat/ap\*/webapps/*

*# sudo nano /etc/httpd/conf.d/conector\_jk.conf*

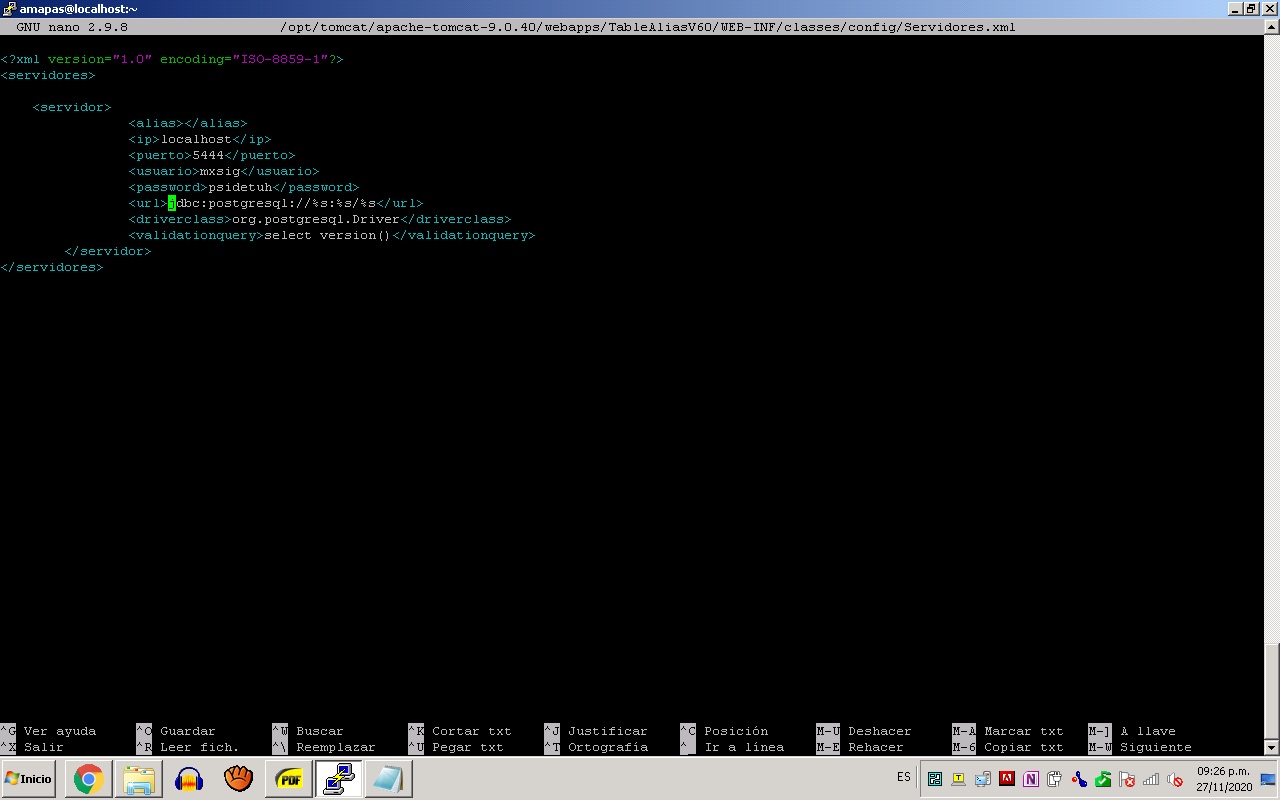


*# sudo systemctl restart tomcat*

*# sudo systemctl restart httpd*

1. Aquí se modificarán parámetros para nuestro servidor:

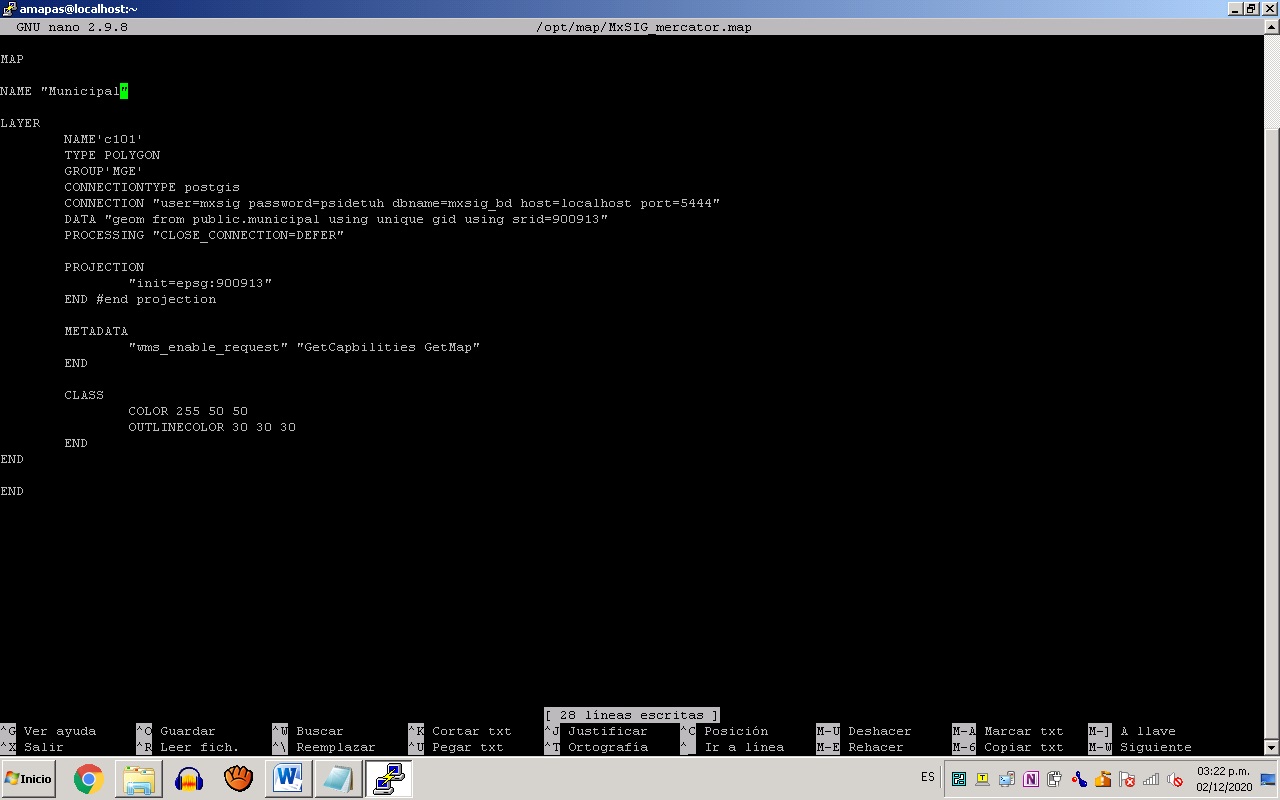
*# sudo nano opt/tomcat/ap\*/webapps/Tab\*V60/WEB-INF/classes/config/Servidores.xml*



*# chmod +r -R /opt/tomcat/ap\*/webapps/Ta\*60*

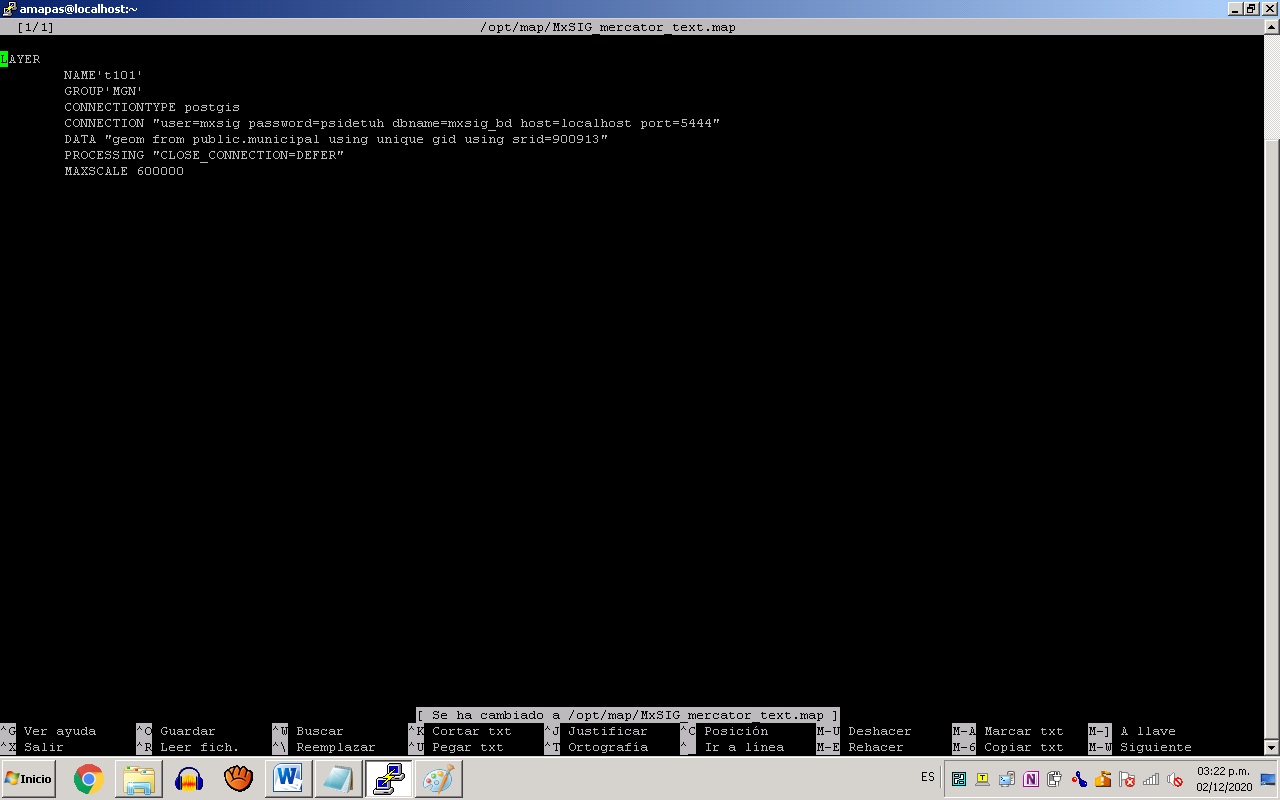
1. Configurar los **archivos MAP** para obtener la información de la base de datos.

*# sudo nano MxSIG\_mercator.map*



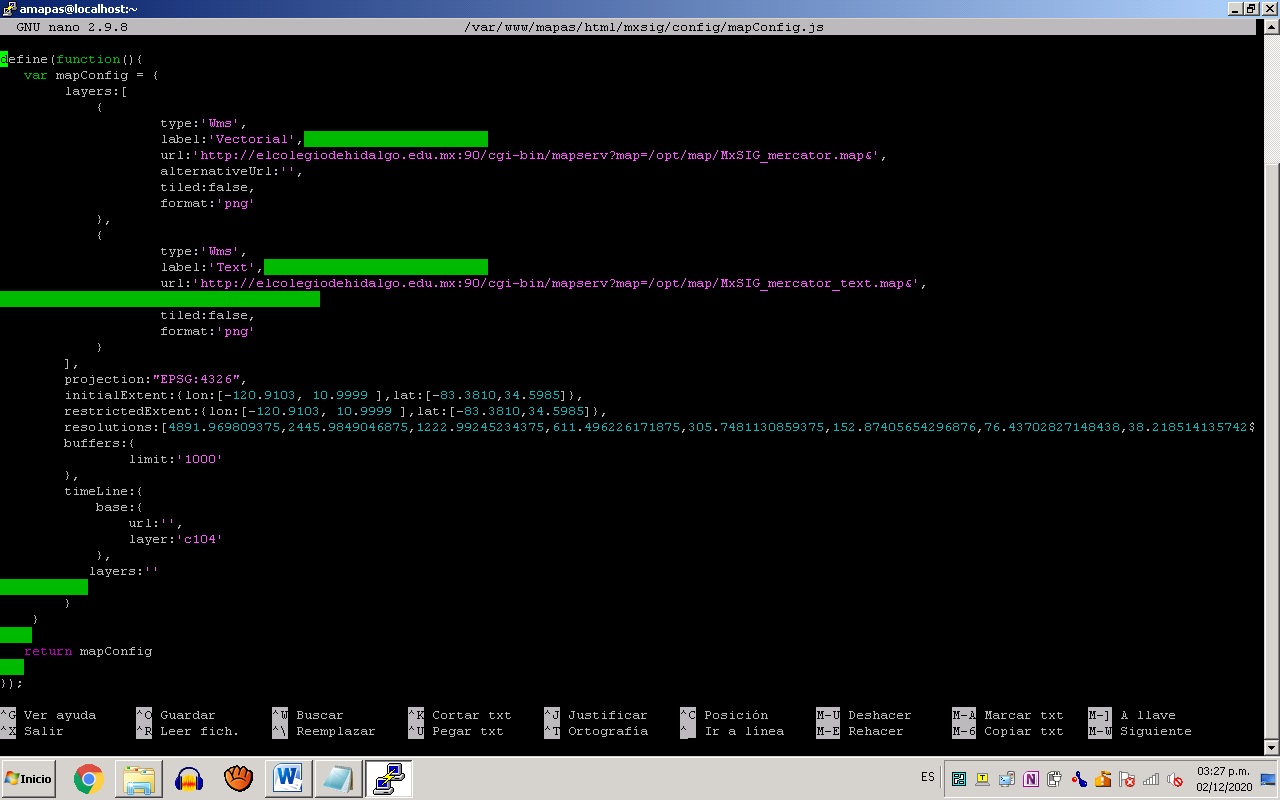
*# sudo cp MxSIG\_mercator.map MxSIG\_mercator\_text.map*

*# sudo nano MxSIG\_mercator\_text.map*



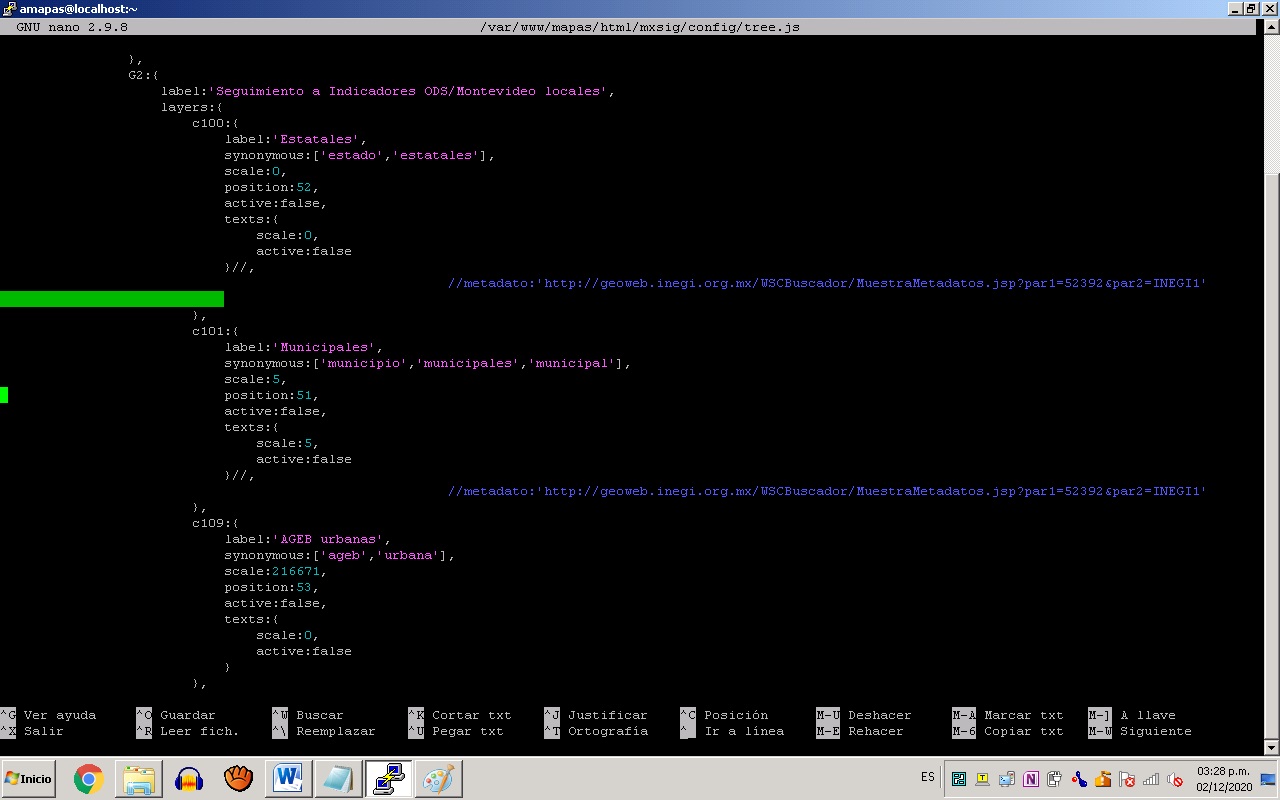
1. Configurar algunos archivos para **plasmar los mapas** en el MxSIG:
   1. En este primer archivo se configurará la ubicación de los archivos MAP.

*# sudo nano mxsig/config/mapConfig.js*

**

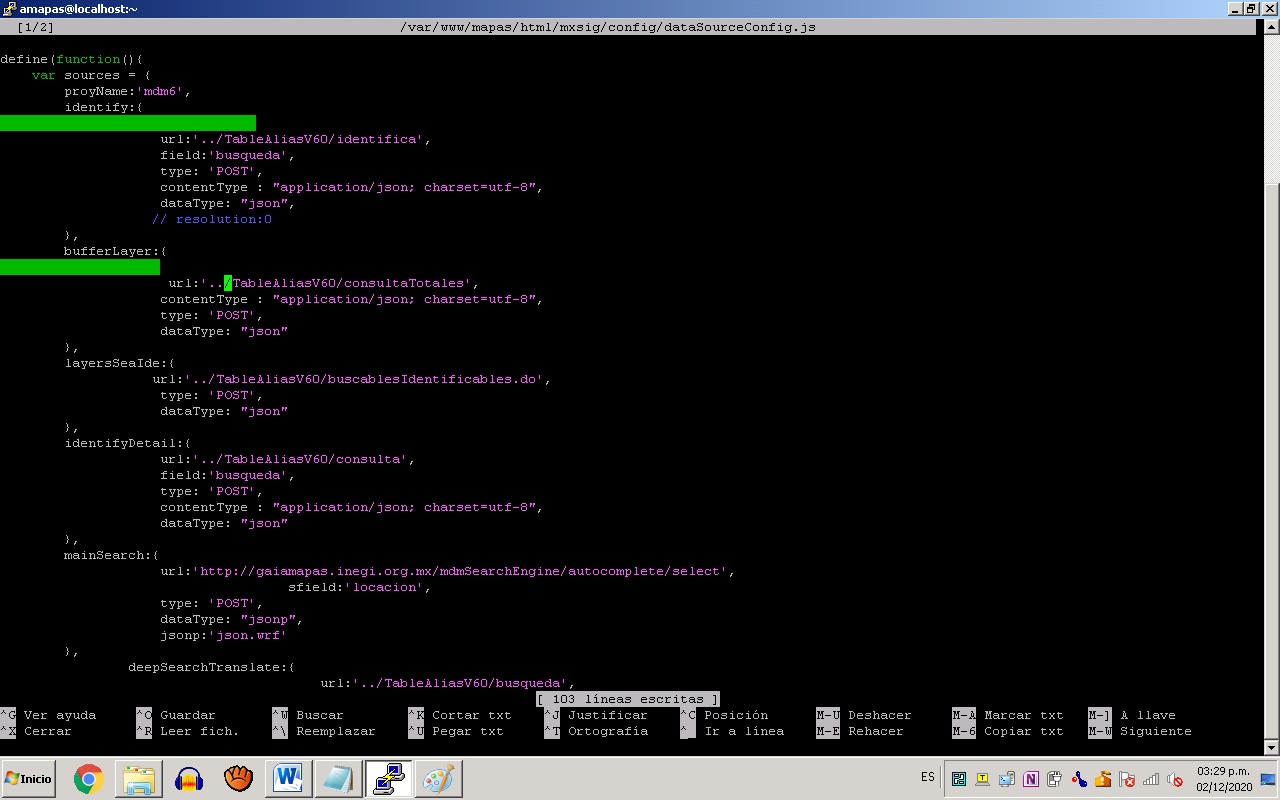
* 1. En este segundo archivo se modificará el **despliegue de las capas de búsqueda**.

*# sudo nano mxsig/config/tree.js*

**

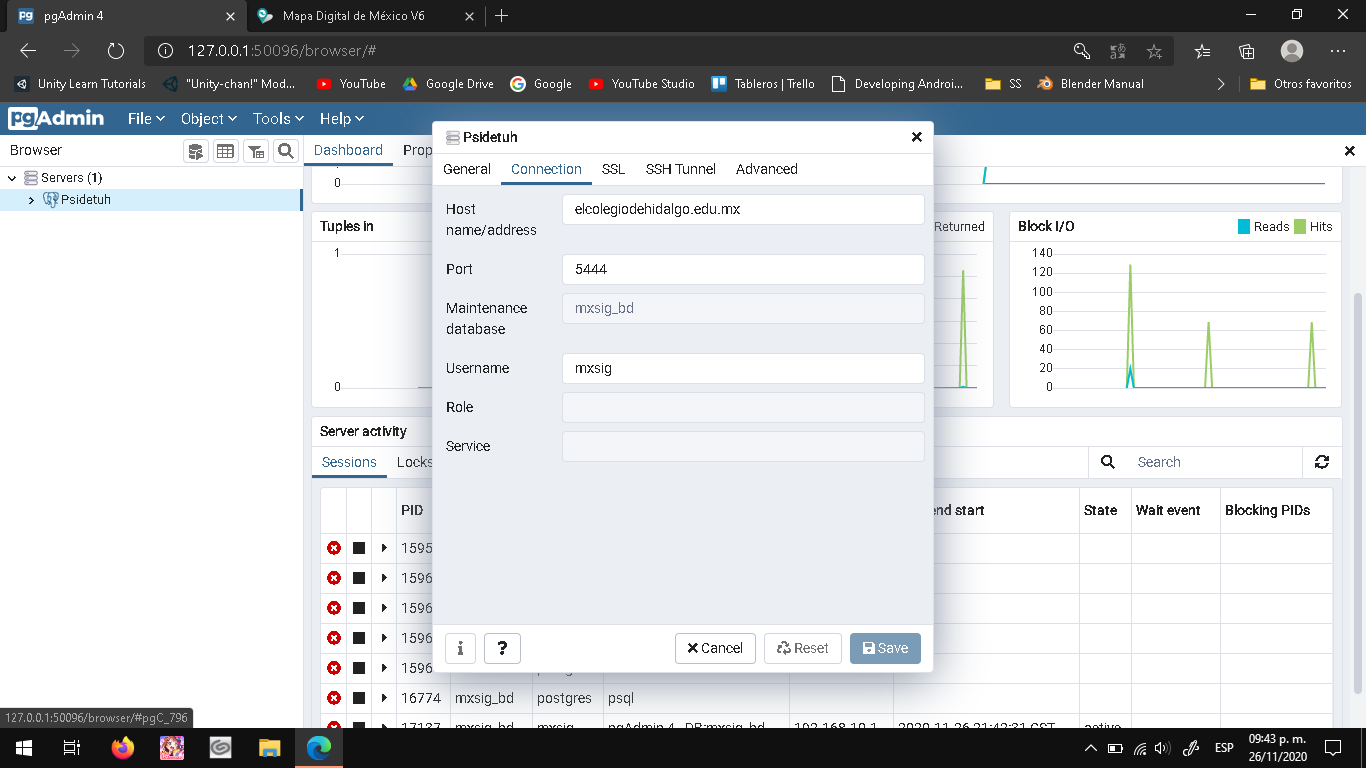
* 1. Y en este último archivo, se configura **la ubicación del WAR** para atender las peticiones que necesiten de él.

*# sudo nano mxsig/config/dataSour\*.js*

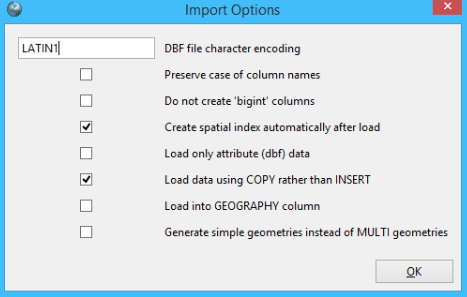


# Procedimiento para subir archivos (SHP)

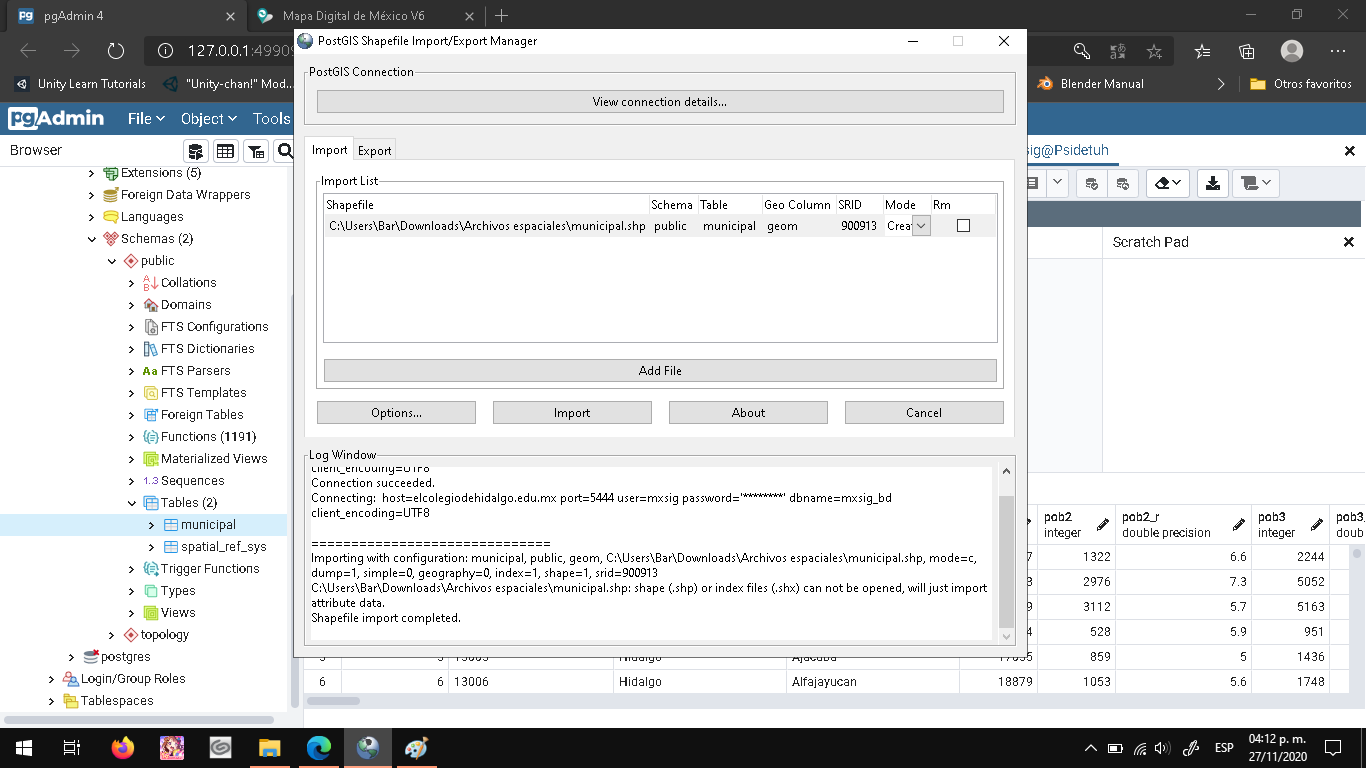
* Instalar PGAdmin y configurar la conexión a la Base de Datos:



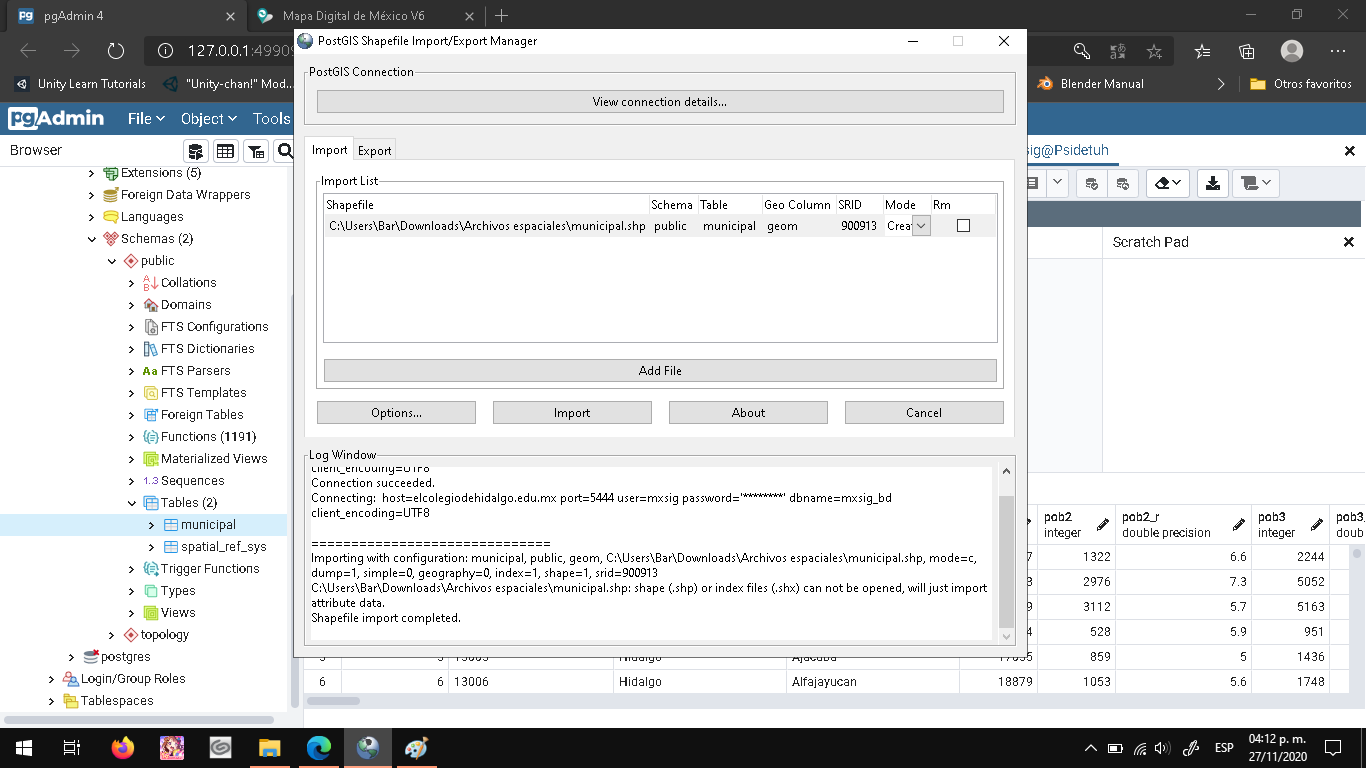
* Descargar el paquete *Postgis-bundle 3.0* en la computadora remota donde se subirán los archivos shapefile SHP.
* Juntar todos los archivos del mapa a subir para que se pueda reconocer toda su información al importarlos con una herramienta de PostGIS, esta herramienta llamada **shp2pgsql-gui**, se ubica dentro de la carpeta *postgis-bundle/bin/postgisgui/.*
* También tendrá que configurar la conexión a la base de datos, como se hizo con PGAdmin.



* Elige el archivo SHP a subir con *Add File,* en la opcion de importacion lo configura como se muestra arriba, despues modifica el SRID a 900913, y elige el esquema(shema) a donde sera exportado a la base de datos.



* Para saber si se importó correctamente, en el mensaje de importacion debe de reconocer el tipo de mapa que es.
* Y en la base de datos, debe de aparecer una nueva tabla con el nombre que se le dio en la importación.



(…)

# Anexos

## Problemas actuales en el servidor

Actualmente, hay un problema con el archivo TableAlias.WAR, ya que no se despliega adecuadamente en Tomcat (no detecta los archivos *.jar* o un *bean* no se detecta*)* por lo tanto, no es posible ejecutar archivos de este software/programa y no es posible ver si un mapa se puede desplegar en la página web.

Aunque el problema puede deberse a que el procedimiento habitual se hace gráficamente y en la configuración de este servidor se hizo desde consola, no hay garantías de que eso se arregle si se despliega como se hace habitualmente. Pero si ese fuese el caso, estaría feliz que se arreglara de esa manera.