

Practica MAPSERVER y POSTGRESQL sobre MAQUINA VIRTUAL

Curso Geoinformación en WEB

Repositorio de la práctica: <https://github.com/AndresHerrera/mapserver-geoinformacion>

Clonar el repositorio de la práctica:

```
user@osgeolive: ~/mv-geo/mapserver-geoinformacion
user@osgeolive:~/mv-geo$ sudo git clone https://github.com/AndresHerrera/mapserver-geoinformacion.git
[sudo] password for user:
Cloning into 'mapserver-geoinformacion'...
remote: Enumerating objects: 64, done.
remote: Counting objects: 100% (64/64), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 64 (delta 27), reused 64 (delta 27), pack-reused 0
Unpacking objects: 100% (64/64), done.
user@osgeolive:~/mv-geo$ ls
```

git clone <https://github.com/AndresHerrera/mapserver-geoinformacion.git>

o descargue el archivo **mapserver-geoinformacion.zip** disponible en el Drive de la clase.

```
user@osgeolive: ~/mv-geo/mapserver-geoinformacion
user@osgeolive:~/mv-geo/mapserver-geoinformacion$ mapserv -v
MapServer version 7.4.1 OUTPUT=PNG OUTPUT=JPEG OUTPUT=KML SUPPORTS=PROJ SUPPORTS=AGG SUPPORTS=FREE
TYPE SUPPORTS=CAIRO SUPPORTS=SVG SYMBOLS SUPPORTS=RSVG SUPPORTS=ICONV SUPPORTS=FRIBIDI SUPPORTS=WMS
SERVER SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER SUPPORTS=WFS_CLIENT SUPPORTS=WCS_SERVER SUPPORTS=
SOS_SERVER SUPPORTS=FASTCGI SUPPORTS=THREADS SUPPORTS=GEOS SUPPORTS=PBF INPUT=JPEG INPUT=POSTGIS I
NPUT=OGR INPUT=GDAL INPUT=SHAPEFILE
```

\$ mapserv -v

Uso de shp2img para generar un archivo de mapa a partir de la estructura de un map file.

Documentación sobre shp2img : <https://mapserver.org/utilities/shp2img.html>

Que es un mapfile : <https://www.mapserver.org/mapfile/>

EJEMPLO 1:

Los archivos geográficos que acompañan este ejemplo se encuentran en **geodata**

```
MAP
  NAME Ejemplo1
  SIZE 800 600
  STATUS ON
  #EXTENT [xmin] [ymin] [xmax] [ymax]
  EXTENT 1053855.50 860198.93 1068480.77 879011.06

  SHAPEPATH "geodata/" ←
  FONTSET "lib/fonts/fonts.txt"
  SYMBOLSET "lib/symbols/symbols.sym"

  IMAGECOLOR 255 255 255
  UNITS METERS

  WEB
    IMAGEPATH "/var/www/html/geoinformacion/mapserver-geoinformacion/tmp/" ←
    IMAGEURL "tmp/"
END
```

La ruta de salida de la imagen generada mediante shp2img deberá estar en:
/var/www/html/geoinformacion/mapserver-geoinformacion/tmp/

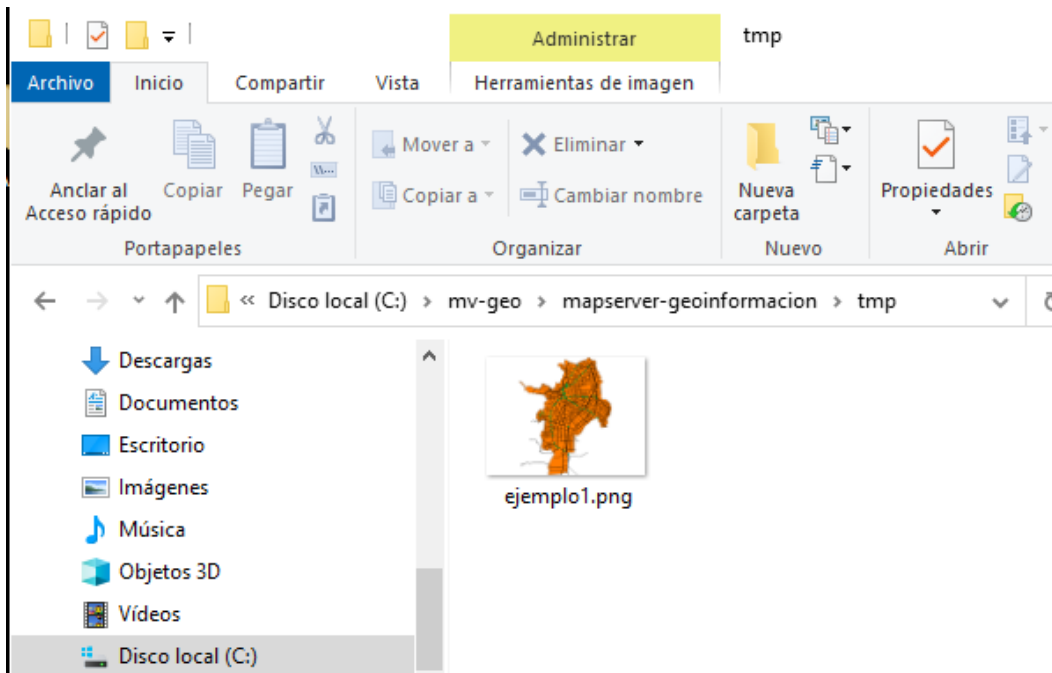
Cree una carpeta en esta ubicación:

mkdir /var/www/html/geoinformacion/mapserver-geoinformacion /tmp/

Genere el mapa mediante el comando shp2img

`shp2img -m ejemplo1.map -o tmp/ejemplo1.png -i PNG`

Visualice el resultado:



Nota: El Ejemplo 1 y 2 generan la imagen a partir de archivos shapefile

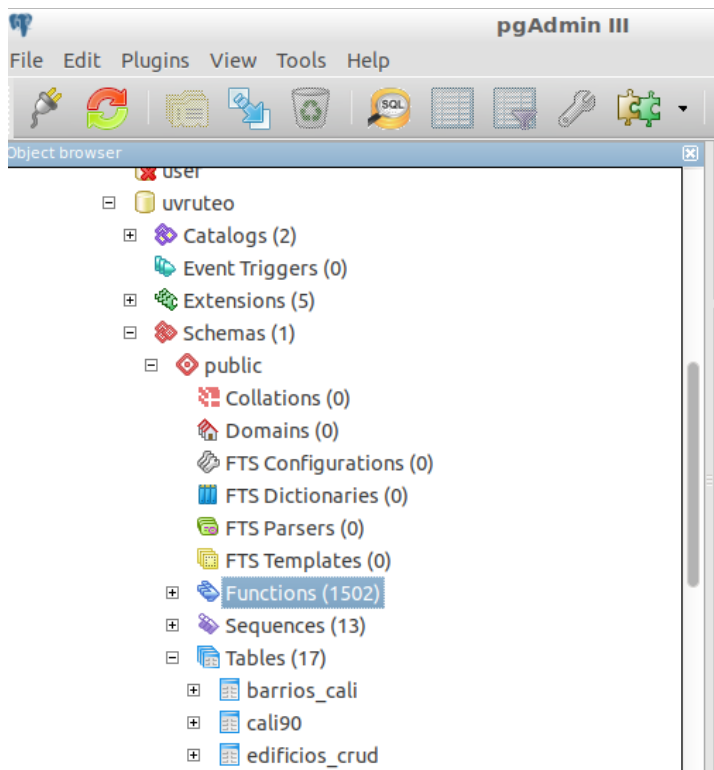
Ejemplo 3: Se deben subir los archivos `rutas_mio.sql` , `estaciones_mio.sql` y `barrios_cali.sql` a la base de datos sig3 (Creada previamente).

En este paso usamos el comando `psql` (visto en clase) en donde le indicamos los parámetros necesarios para acceder a la base de datos. (usuario, puerto, nombre base de datos y archivo que deseamos cargar)

```
$ psql -h localhost -p 5432 -U user -d uvruteo -f barrios_cali.sql
```

```
user@osgeolive: ~/mv-geo/mapserver-geoinformacion/geodata
user@osgeolive:~/mv-geo/mapserver-geoinformacion/geodata$ ls
barrios_cali.dbf*      estaciones_mio.shp*
barrios_cali.prj*      estaciones_mio.shx*
barrios_cali.qpj*      estaciones_mio.sql*
barrios_cali.shp*      rutas_mio.dbf*
barrios_cali.shx*      rutas_mio.prj*
barrios_cali.sql*      rutas_mio.qpj*
estaciones_mio.dbf*    rutas_mio.shp*
estaciones_mio.prj*    rutas_mio.shx*
estaciones_mio.qpj*    rutas_mio.sql*
user@osgeolive:~/mv-geo/mapserver-geoinformacion/geodata$ psql -h localhost -p 5432 -U user -d uvrouteo -f barrios_cali.sql
```

Una vez completada la carga de cada uno de los archivos. Verificamos:

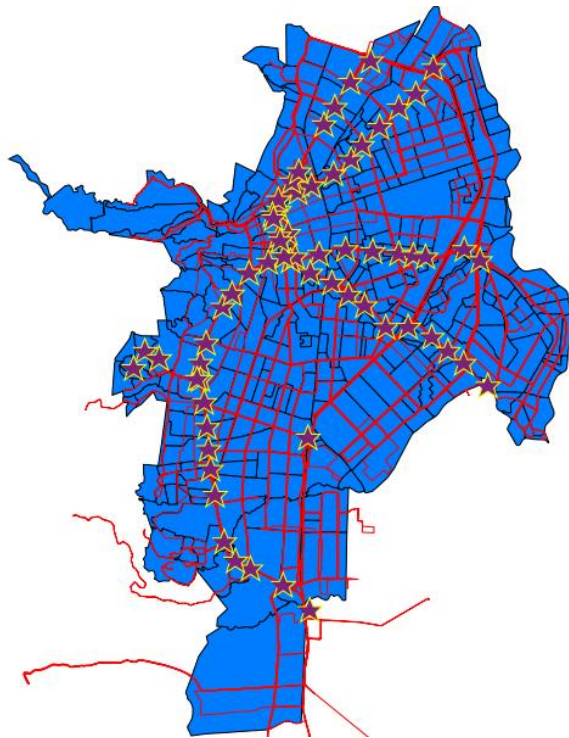


Generamos el mapa del ejemplo3.map

Nota: En este ejemplo los datos son obtenidos desde la base de datos

```
LAYER
  CONNECTIONTYPE postgis
  NAME "Poligonos"
  STATUS ON
  CONNECTION "user=user password=user dbname=uvruteo host=localhost port=5432"
  DATA "the_geom FROM barrios_cali as poligonos using unique gid using SRID=3115"
  TYPE POLYGON
  CLASS
    STYLE
      COLOR 0 125 255
      OUTLINECOLOR 0 0 0
    END
  END
END
```

shp2img -m ejemplo3.map -o tmp/ejemplo3.png -i PNG

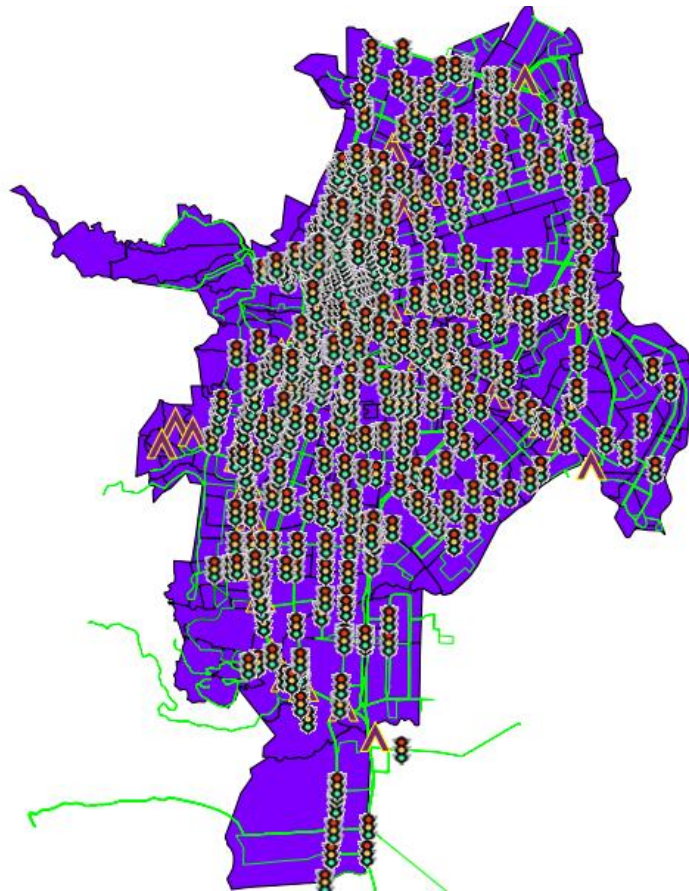


Ejemplo 4:

Obtenemos desde un servicio WMS (IDES Cali) la capa de semáforos

shp2img -m ejemplo4.map -o tmp/ejemplo4.png -i PNG

```
LAYER
  NAME "Semaforos"
  TYPE RASTER
  STATUS ON
  CONNECTION "http://ws-idesc.cali.gov.co:8081/geoserver/wms?"
  CONNECTIONTYPE WMS
  METADATA
    "wms_srs" "EPSG:3115"
    "wms_name" "movilidad:mt_sdv_intersecciones_semaforizadas"
    "wms_server_version" "1.1.0"
    "wms_format" "image/png"
  END
END
```



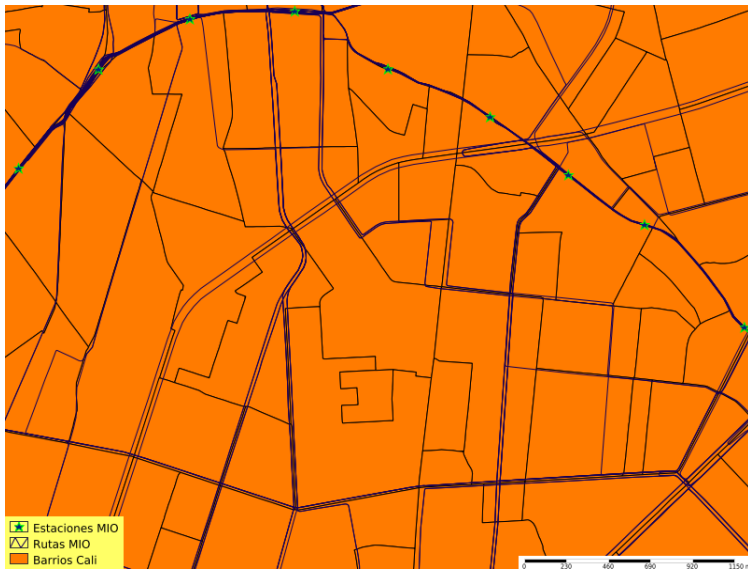
Ejemplo 5

Generación de leyenda y escala grafica

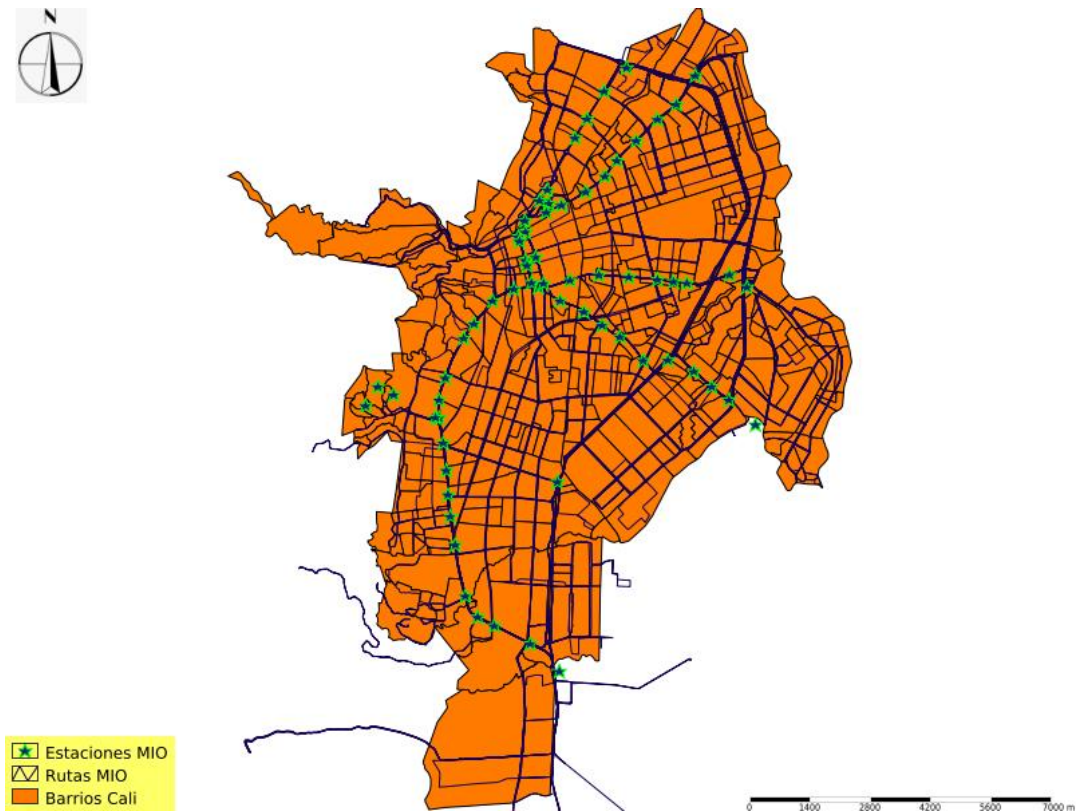
```
#Leyenda embebida
LEGEND
  STATUS EMBED
  IMAGECOLOR 255 255 102
  POSITION LL
  KEYSIZE 20 12
  KEYSPPACING 5 5
  LABEL
    TYPE TRUETYPE
    FONT "sans"
    SIZE 9
    COLOR 0 0 0
  END
  OUTLINECOLOR 51 51 51
  #TRANSPARENT on
END

SCALEBAR
  STYLE 0
  STATUS EMBED
  SIZE 250 4
  COLOR 0 0 0
  UNITS METERS
  INTERVALS 5
  TRANSPARENT off
  POSITION LR
  BACKGROUNDOLOR 255 255 255
  IMAGECOLOR 255 255 255
  OUTLINECOLOR 211 211 211
  LABEL
    COLOR 0 0 0
    SIZE TINY
  END
END
```

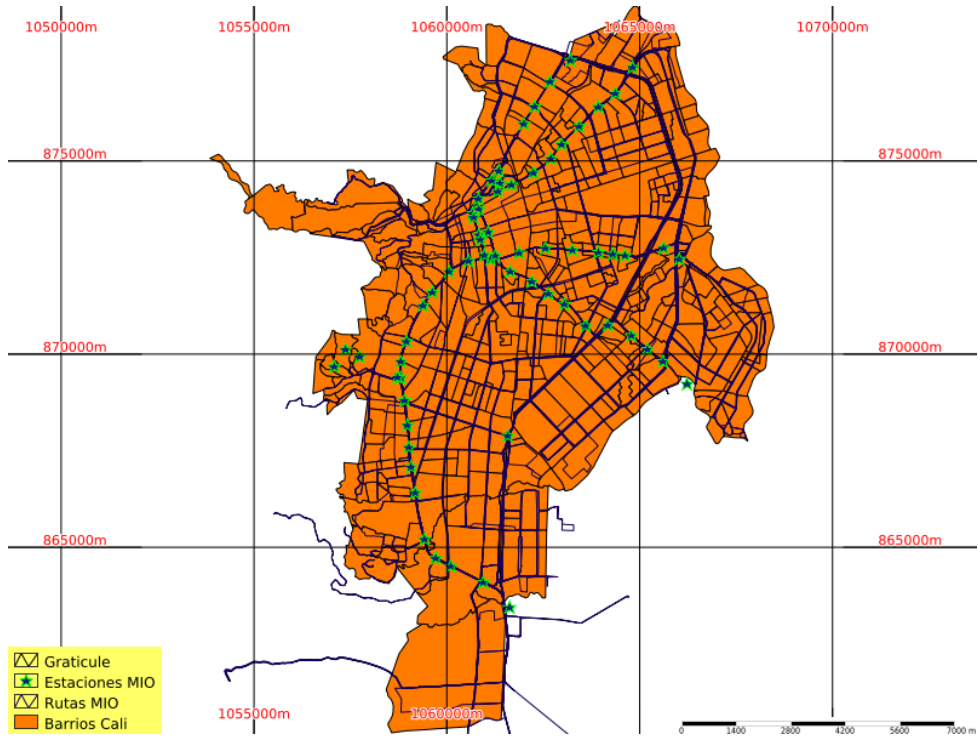
shp2img -m ejemplo5.map -o tmp/ejemplo5.png -i PNG



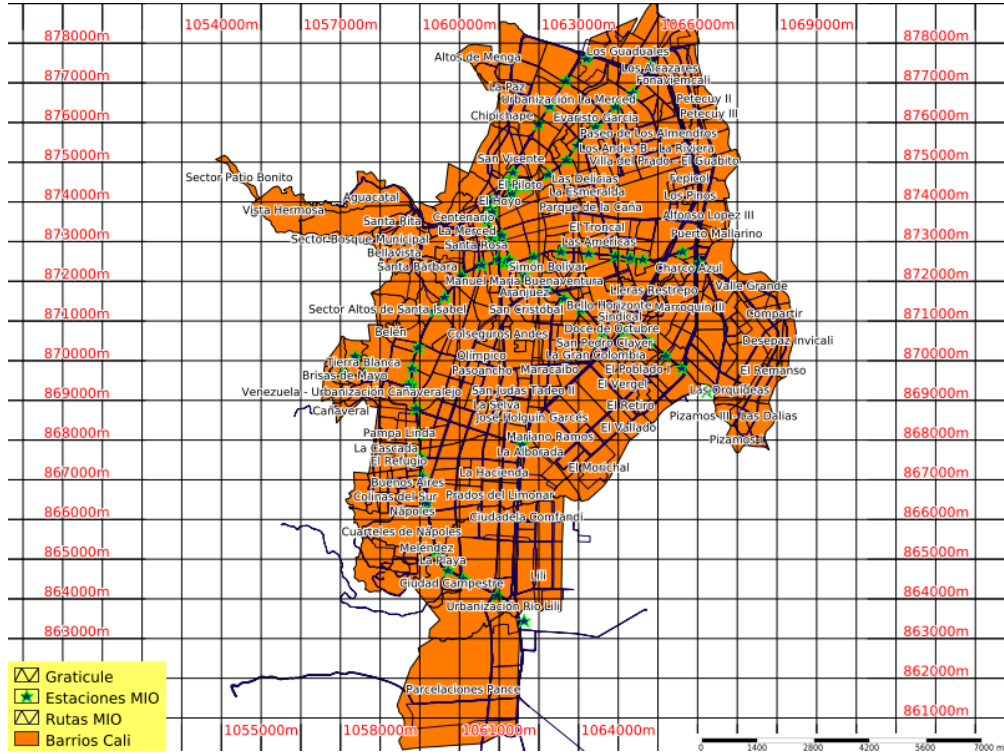
EJEMPLO 6



EJEMPLO 7



EJEMPLO 8



TALLER

1) Construir un Map file que permita generar un mapa con las siguientes características:

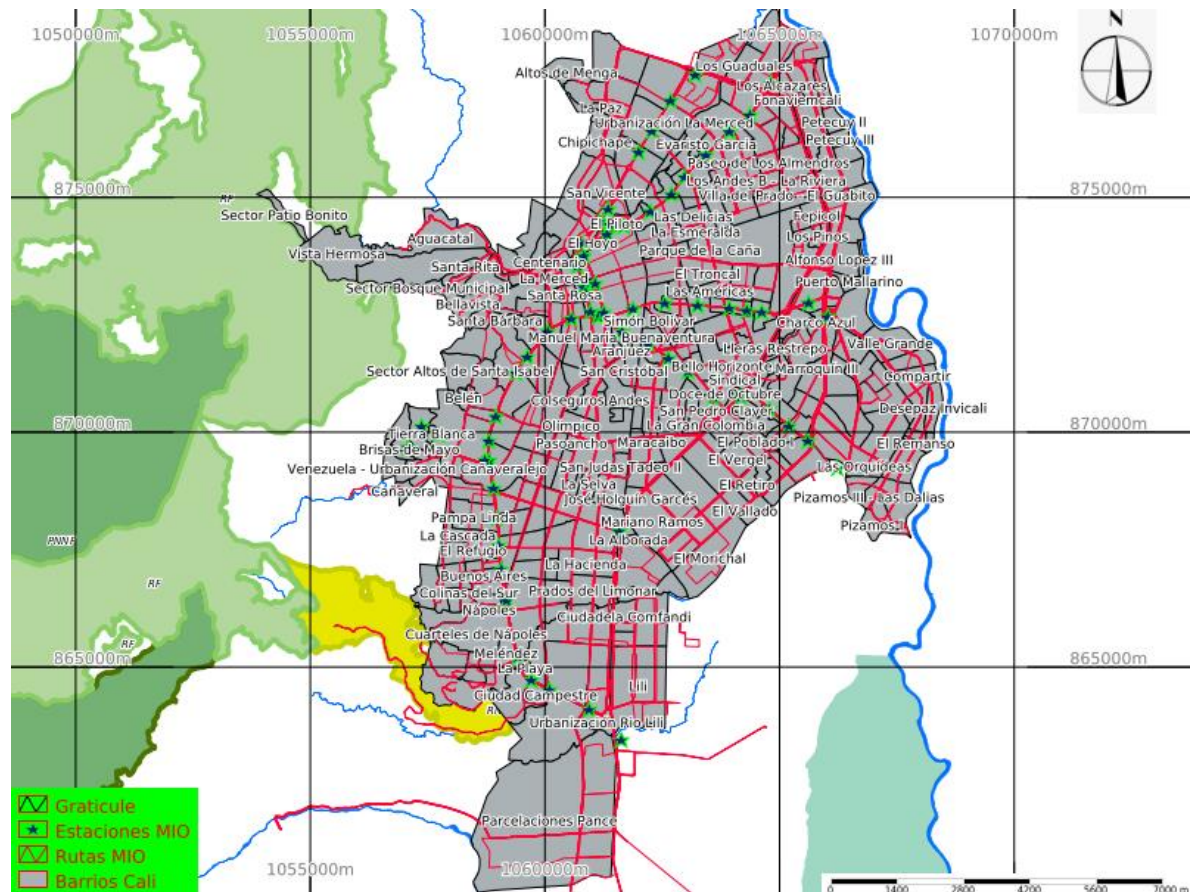
Colores Similares para Cuadro de leyenda

Rosa de los vientos

Colores para las capas de: rutas de MIO, estaciones MIO y barrios

Capas obtenidas de IDES CALI

- Ríos
- Áreas Protegidas
- Zona ambiental del Rio Cauca



Forma y fecha de entrega: 06 abril Informe individual con procedimiento realizado en formato .pdf, incluirlo junto al código fuente del desarrollo de la practica en un repositorio de github llamado **taller-mapserver-geoinformacion**