**Requerimientos**

pgAdmin v1.20.0 <http://www.pgadmin.org/>

PostgreSQL v9.3 <http://www.postgresql.org.es/>

PostGIS v2.2 <http://postgis.net/>

Geoserver v2.8.2 <http://geoserver.org/>

QGIS v2.14.0 <http://qgis.org/es/site/>

NetBeans IDE v7.1 <https://netbeans.org/>

JDK v1.6.0 JRE v6 <http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>

Apache Tomcat

**Importar un archivo shape en PostgreSQL**

Importar el archivo .shp mediante el comando shp2pgsql especificando:

* el sistema de coordenadas –s “4326”
* la creación de un índice espacial –I
* la ruta completa de nuestro archivo shp
* el esquema y nombre de la tabla
* nombre del archivo de salida sql



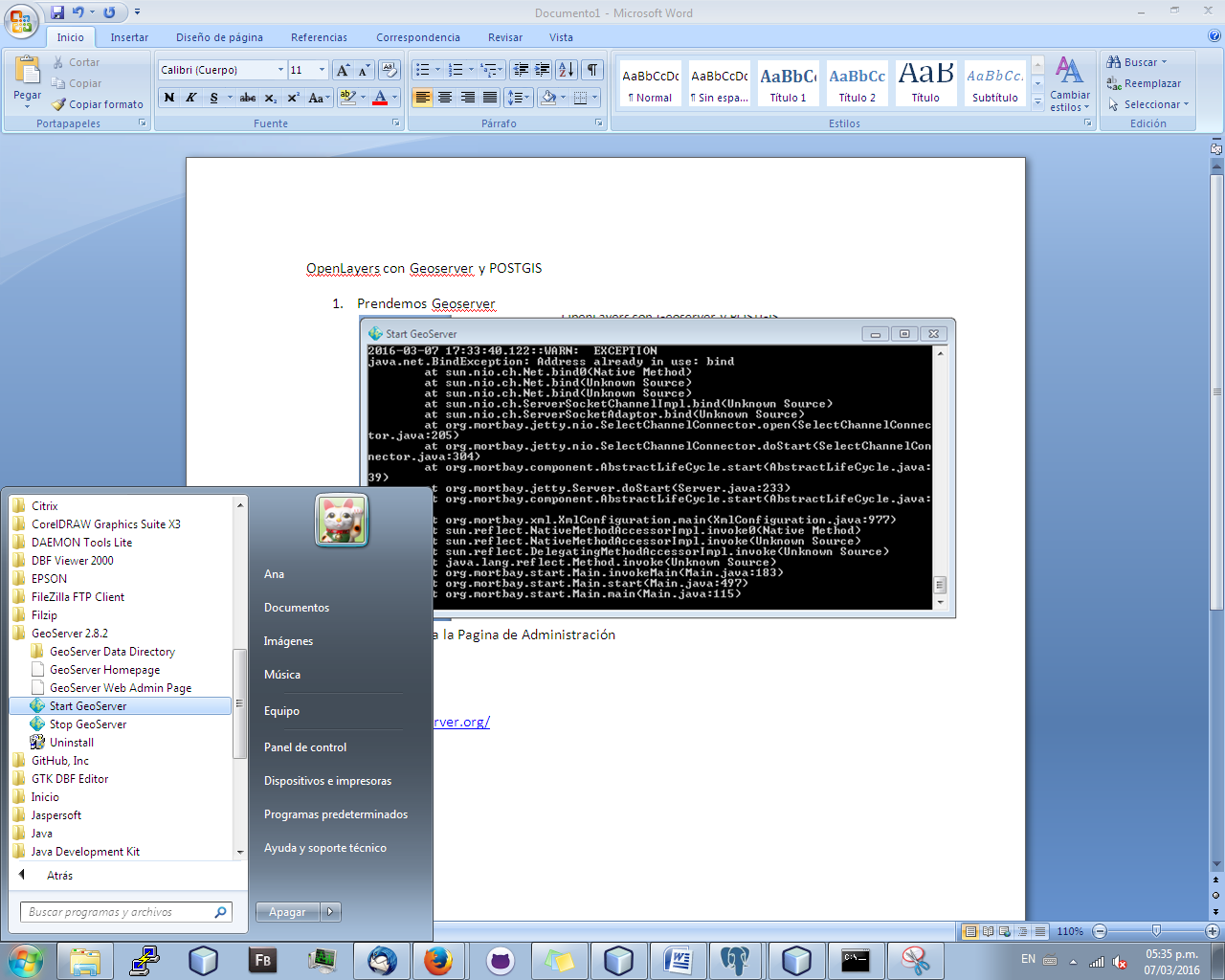
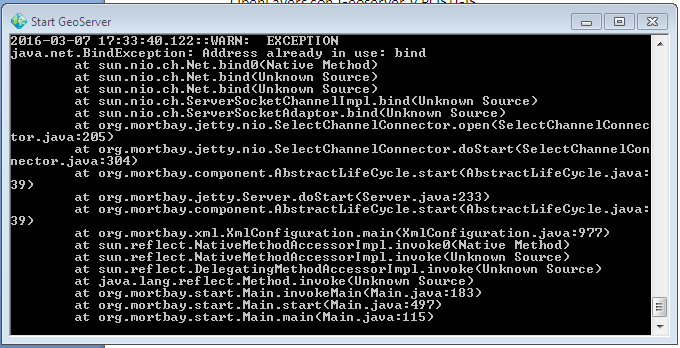
Ejecutamos el archivo sql para crear la tabla especificando:

* la base de datos en la que se creará la tabla –d mapas
* el host –h localhost
* el usuario –U postgres
* el puerto –p 5432
* el archivo que queremos cargar –f archivo.sql

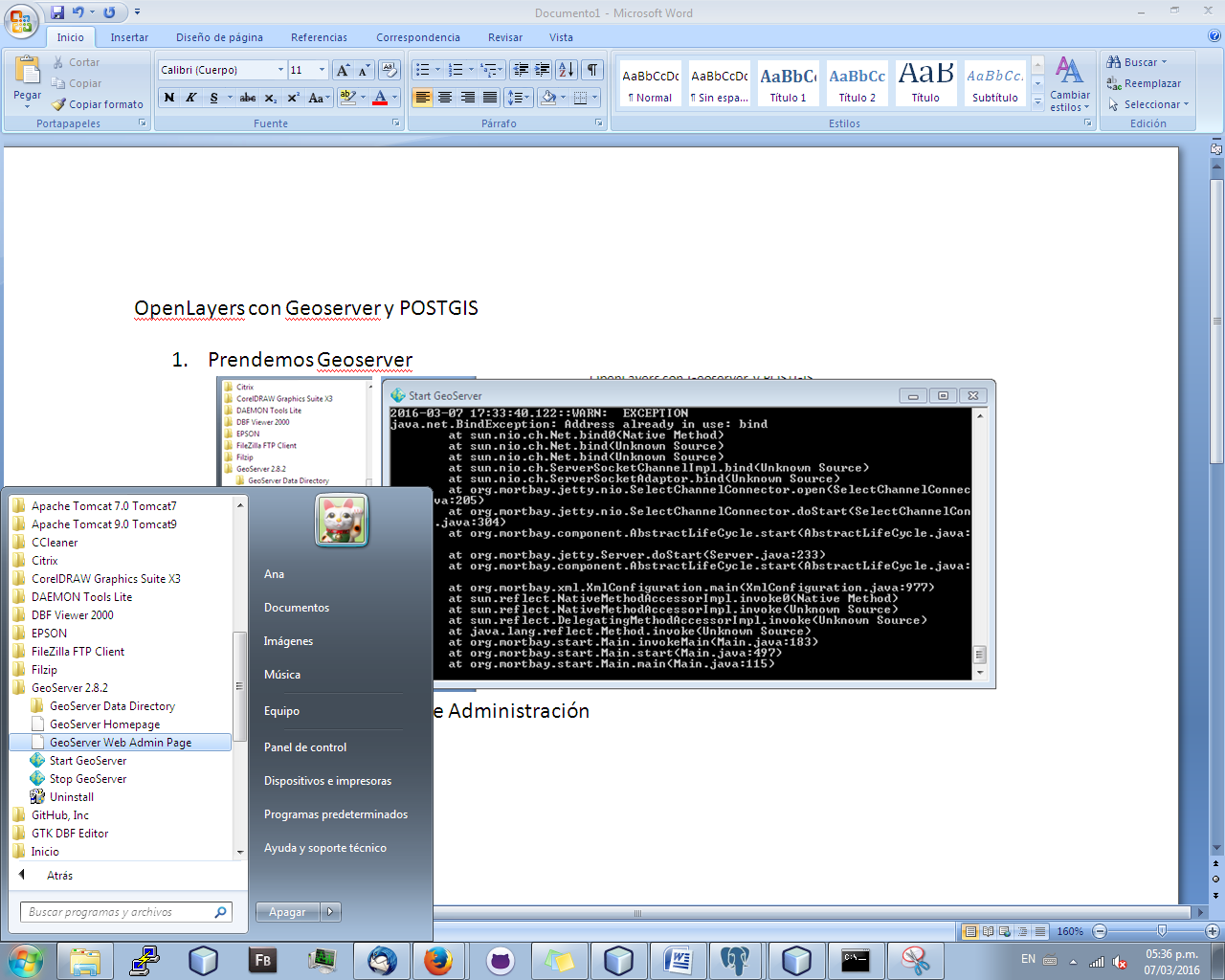


**Importar las Capas de PostGIS en GeoServer**

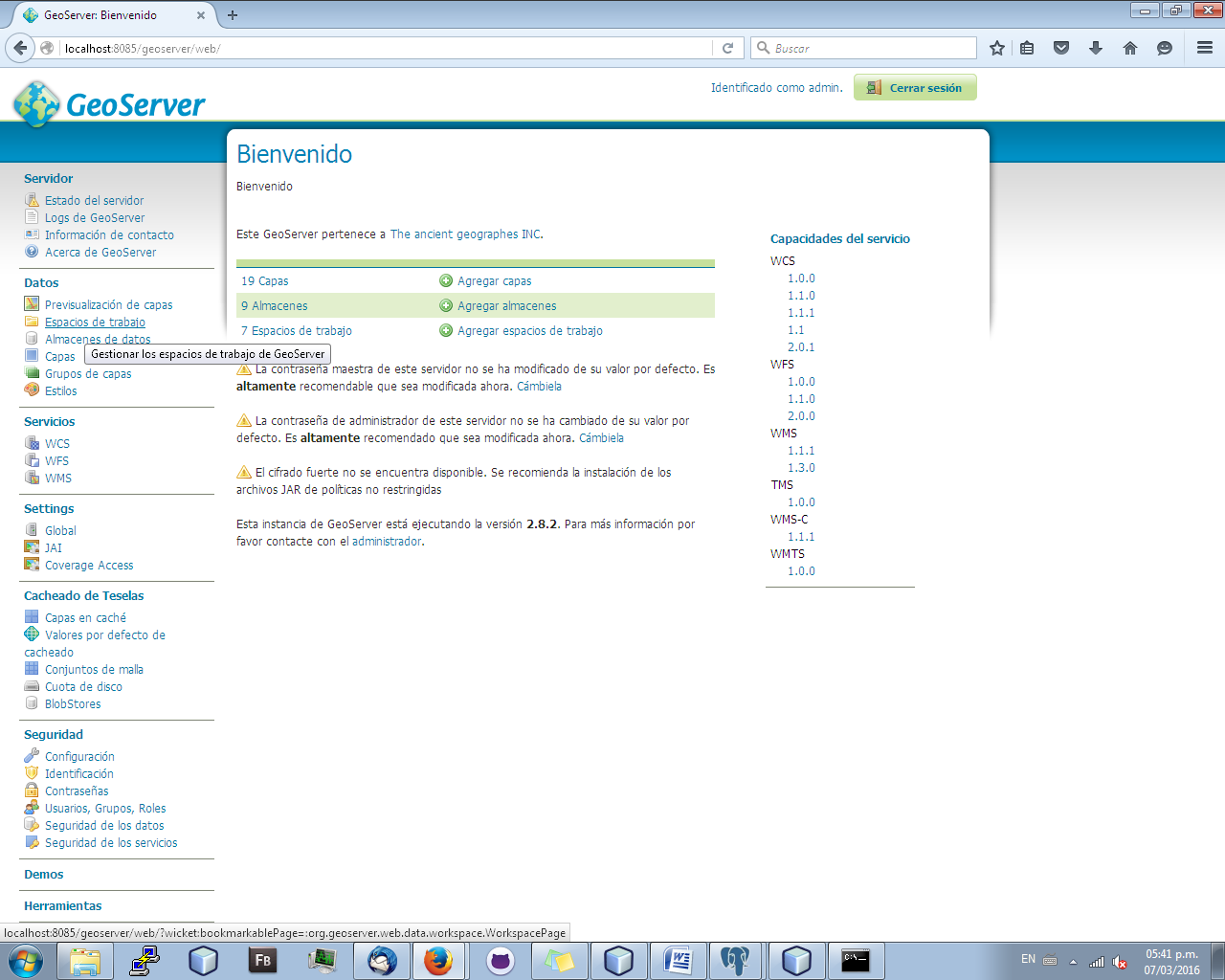
1. Prendemos Geoserver

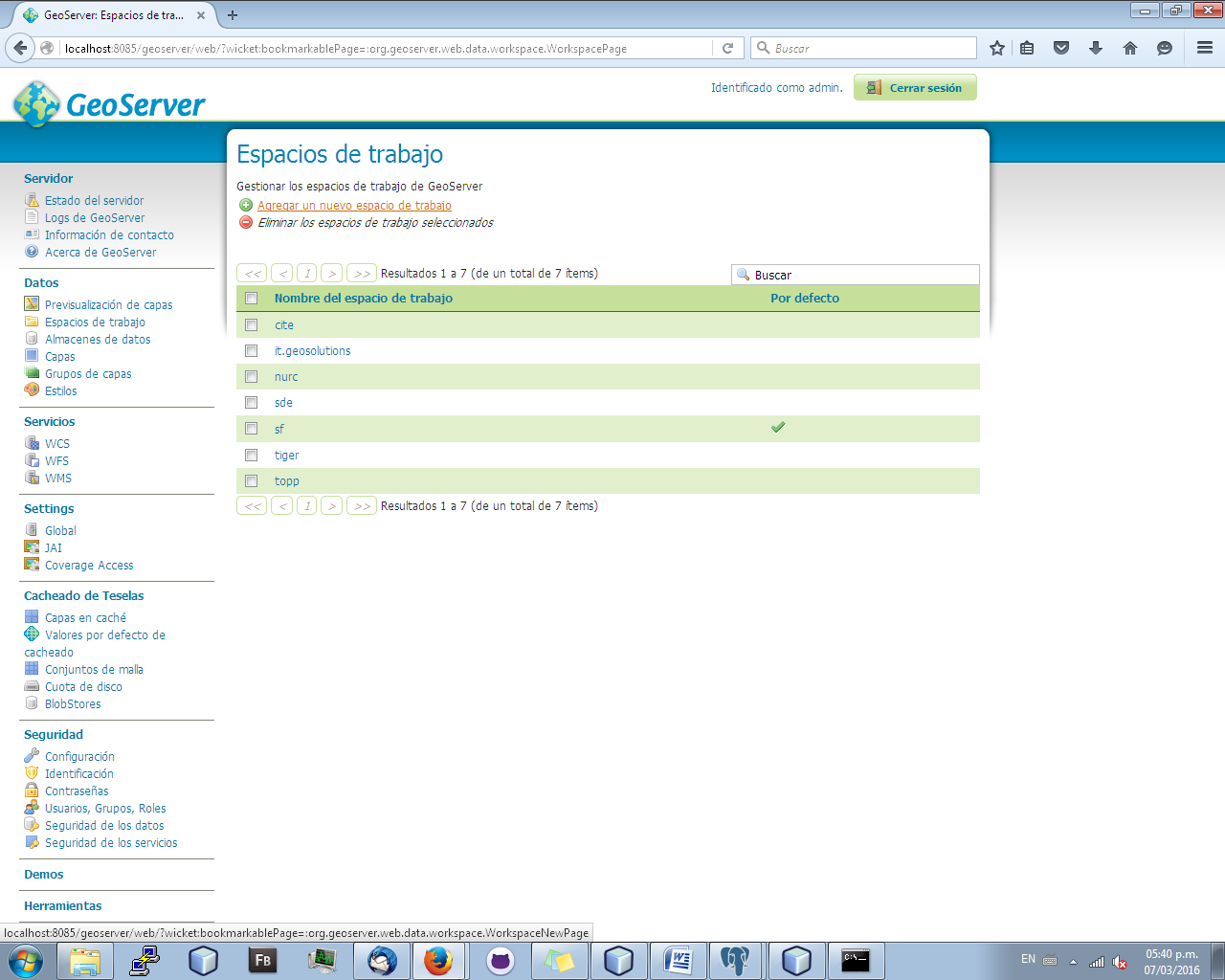
 

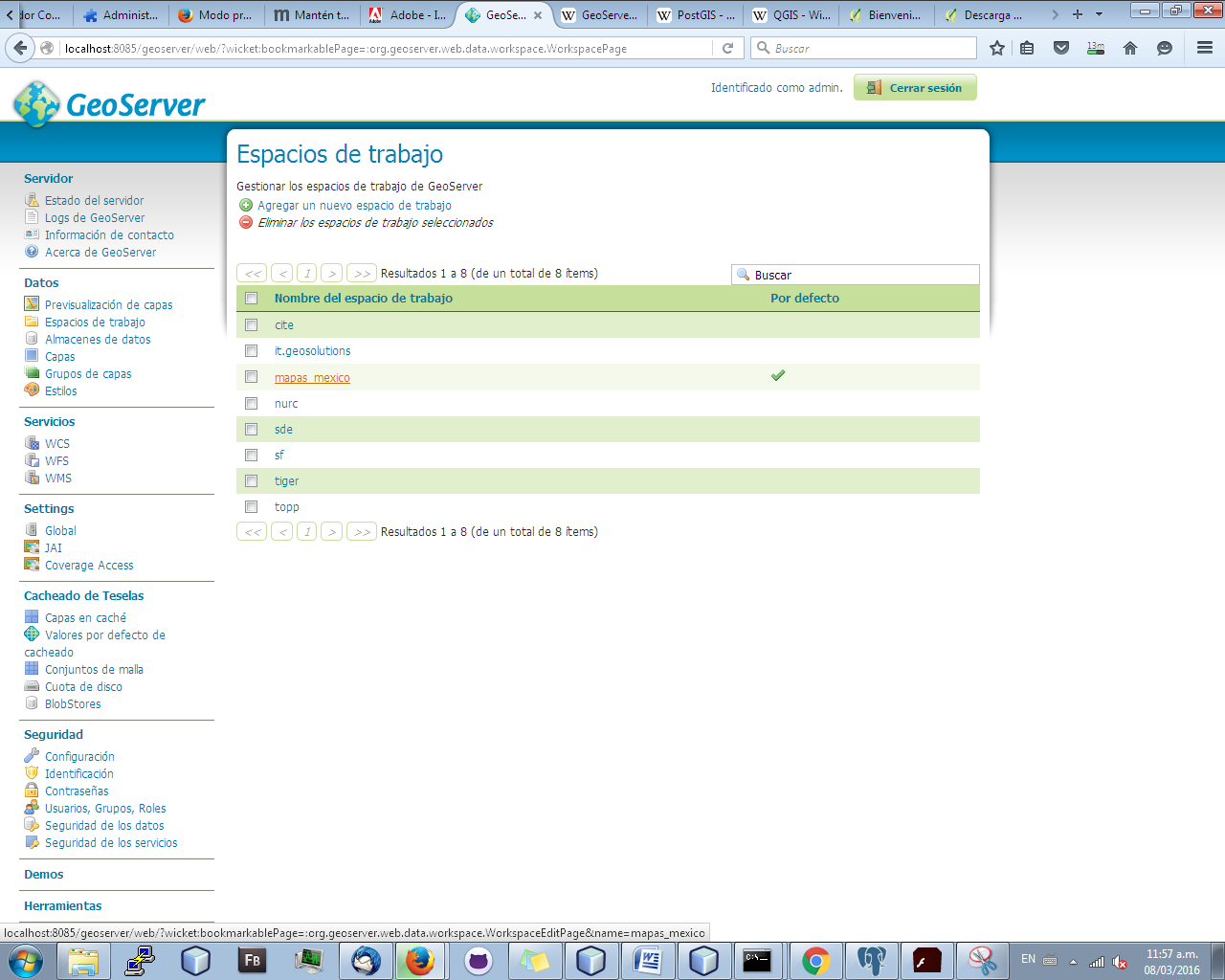
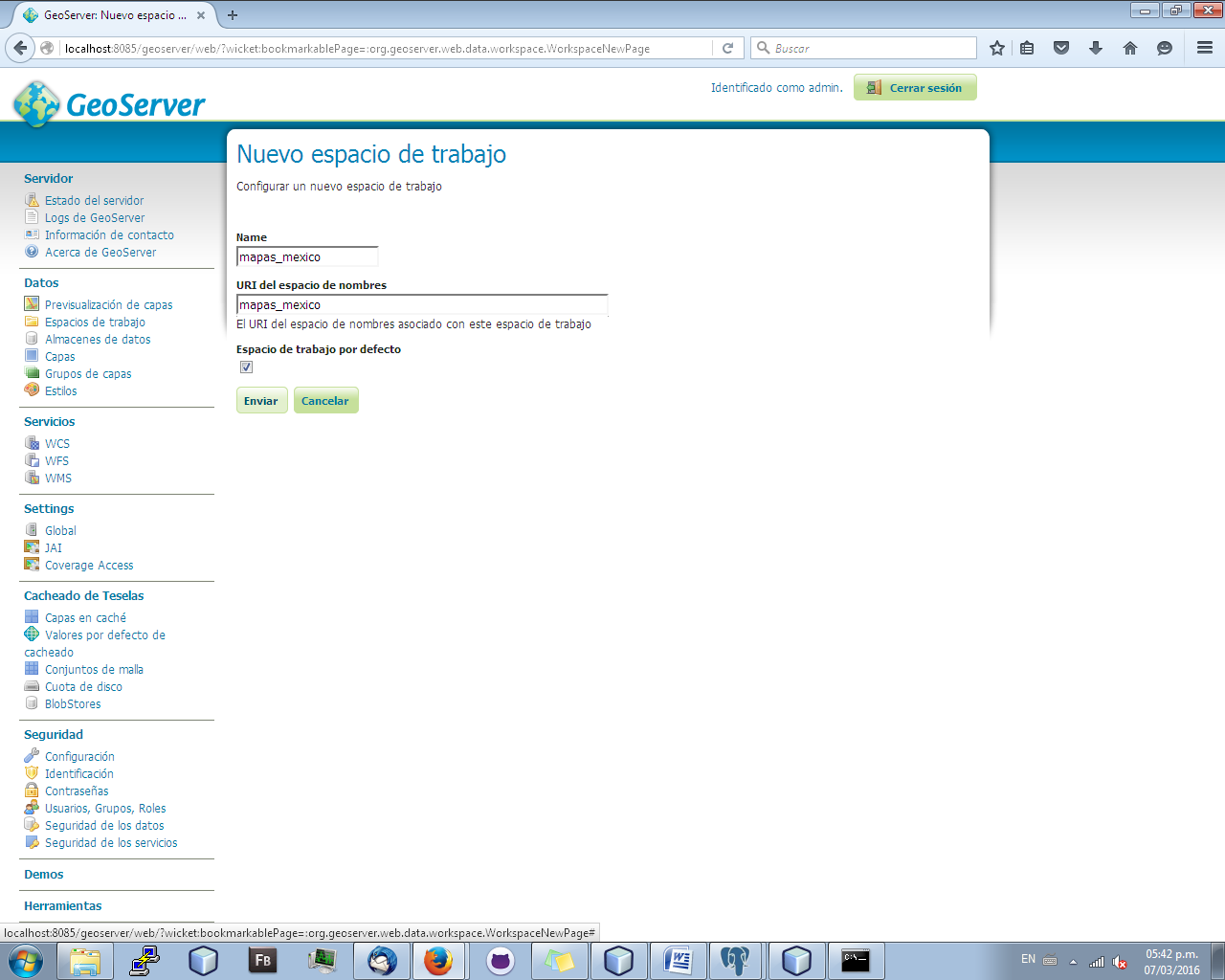
1. Accedemos a la Pagina de Administración

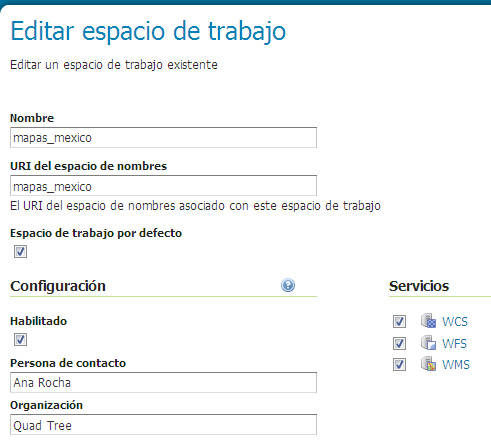


1. Agregamos un nuevo espacio de Trabajo y lo habilitamos

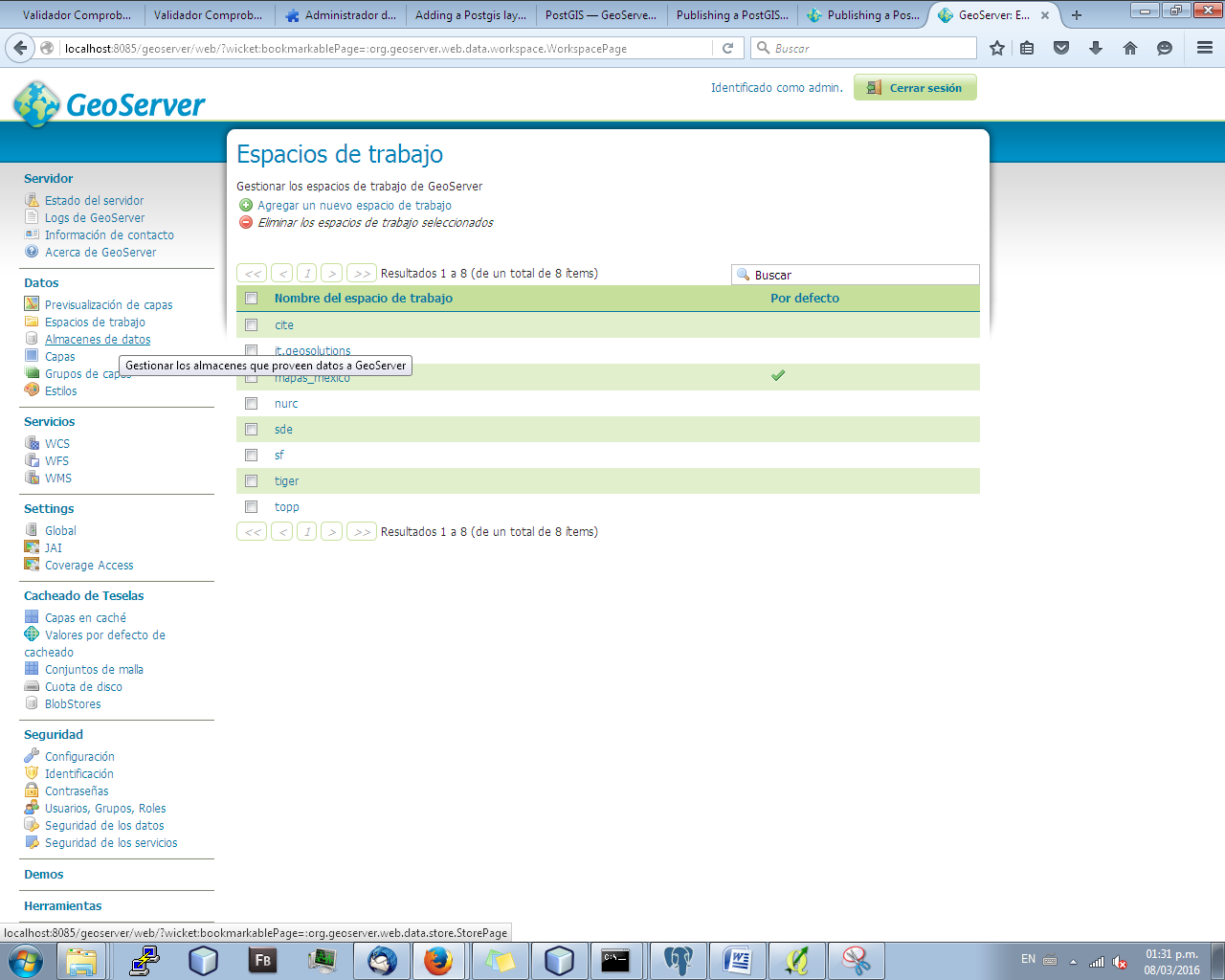


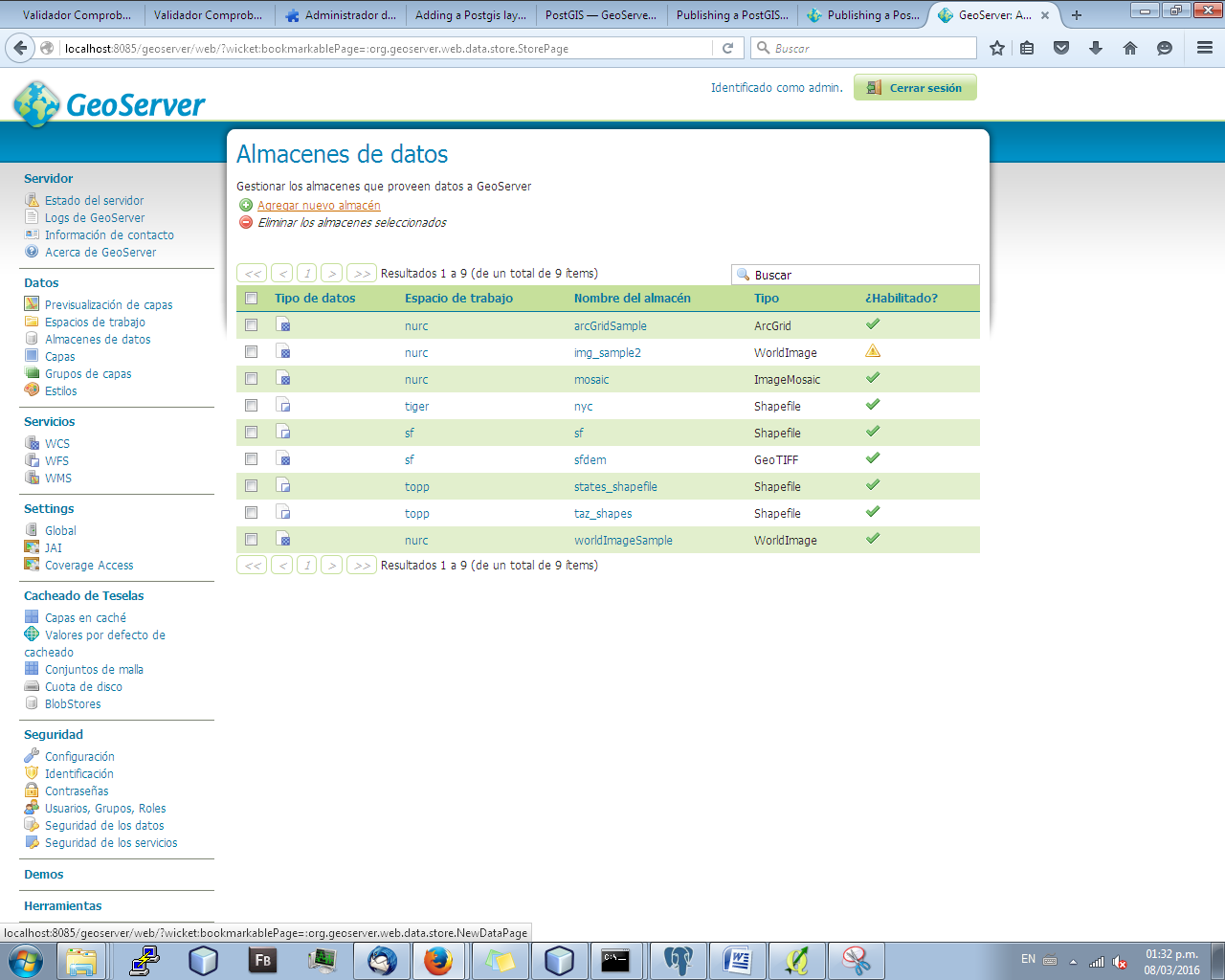


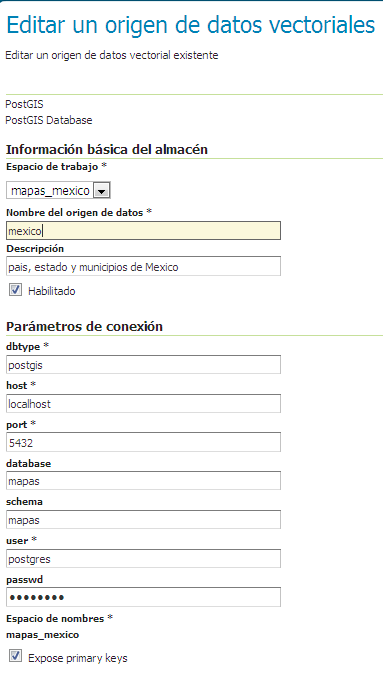


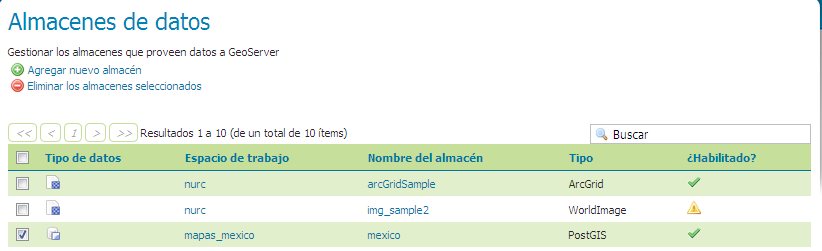


1. Creamos el Almacén de Datos

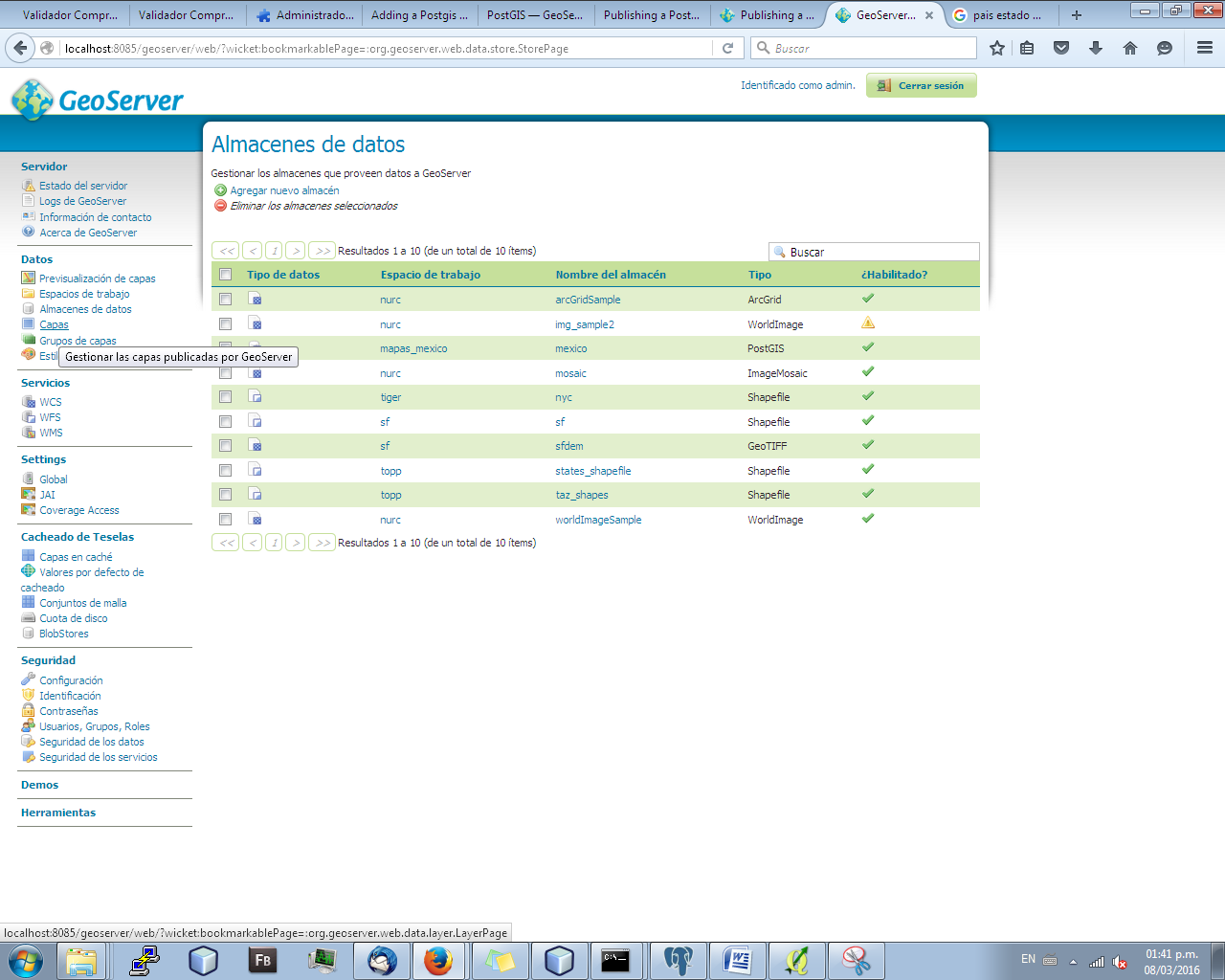


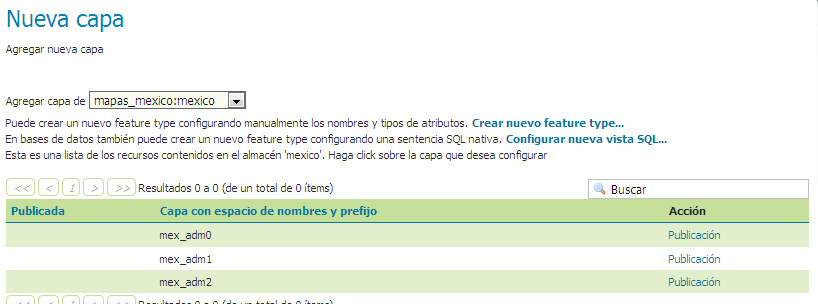


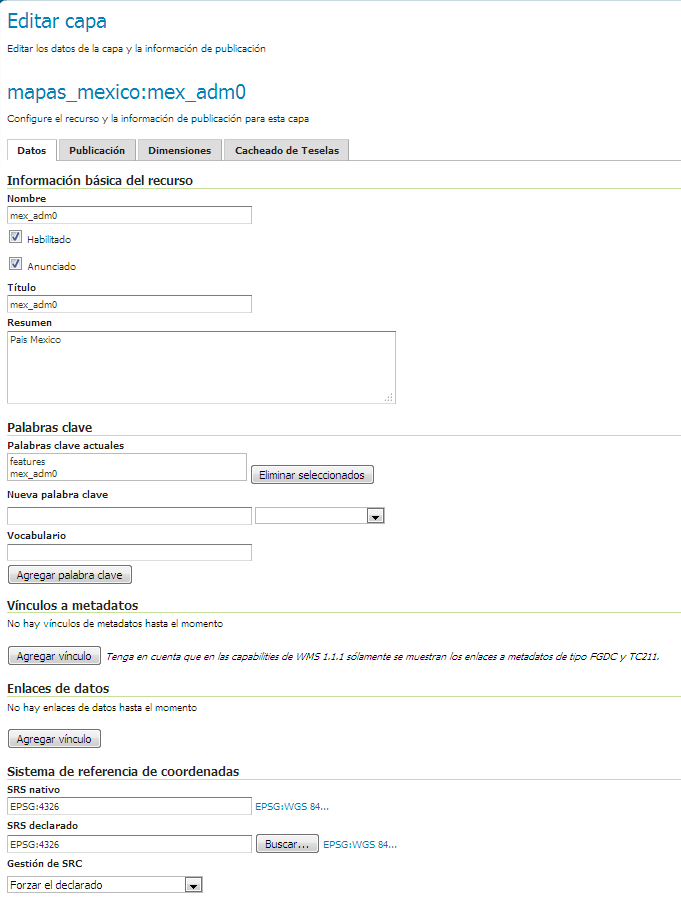




1. Agregamos cada una de las capas

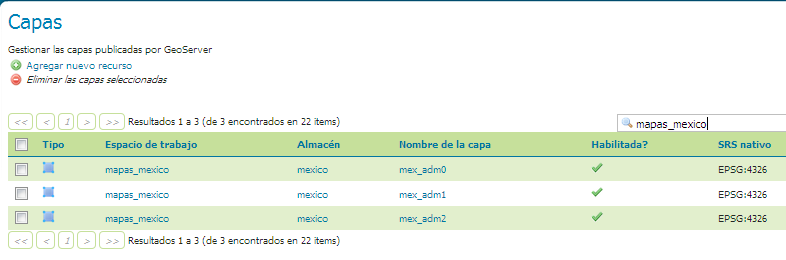




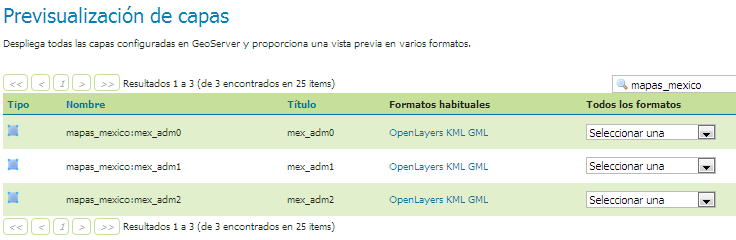


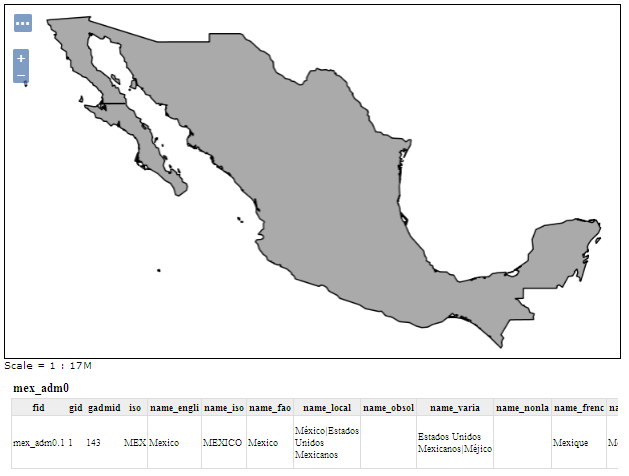
Calculamos los encuadres desde los datos, y la latitud y longitud desde el encuadre nativo

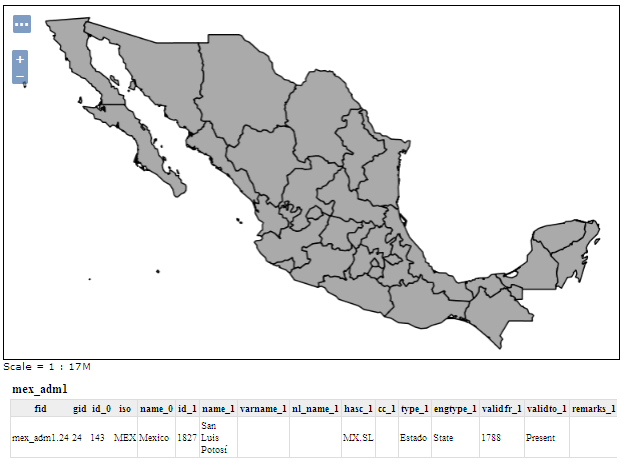


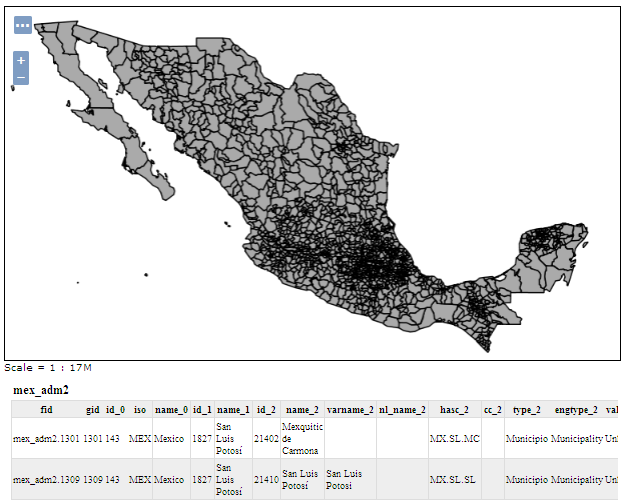


1. Ahora podemos visualizar nuestras capas con OpenLayers en la Pre visualización de Capas





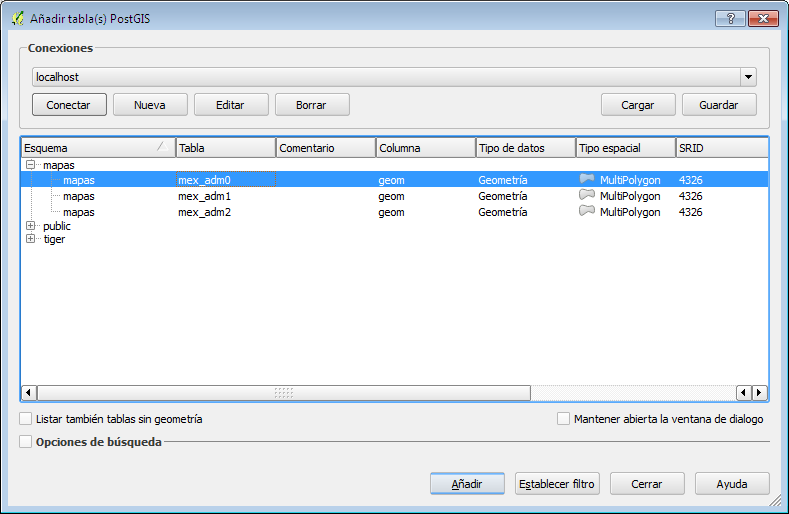
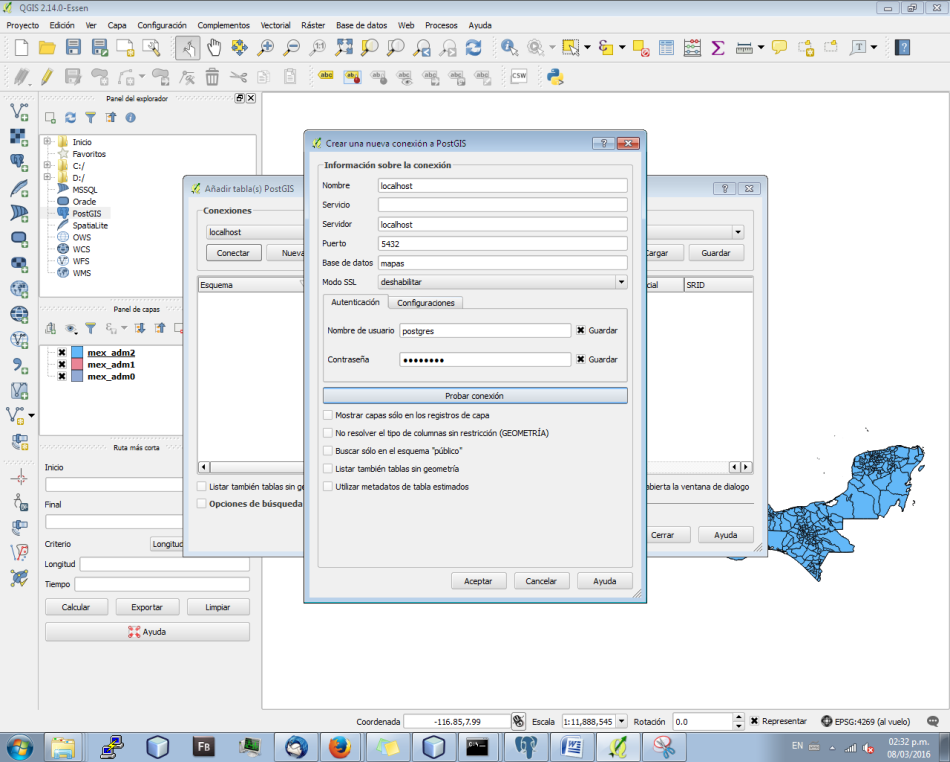




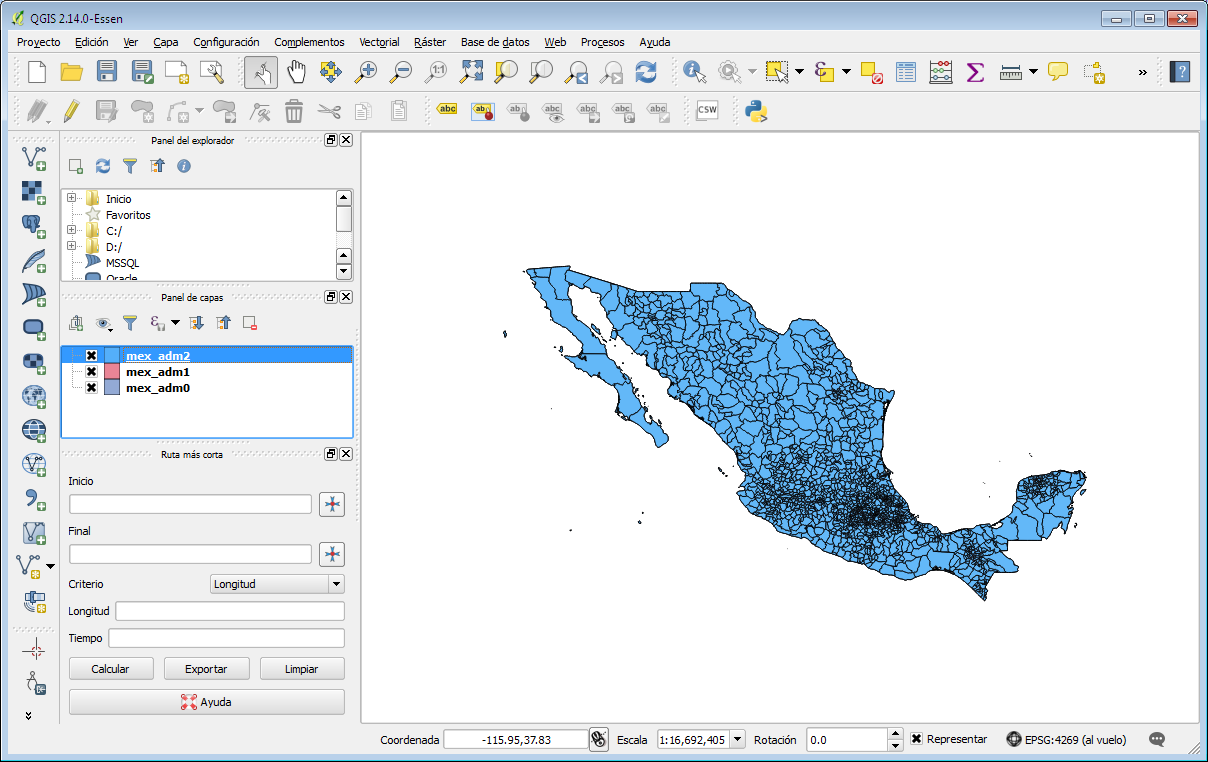
También lo podemos visualizar desde QGIS

Creamos la nueva conexión a la Base de Datos y la conectamos, identificamos nuestro esquema y seleccionamos las capas para añadirlas al espacio de trabajo actual

Añadir capas PostGIS



Una vez hecho esto se podrá visualizar las capas



**Visualizar en Java Web el mapa base con OpenLayers**

Para poder utilizar las clases proporcionadas por OpenLayers habrá que incorporar una refrencia al script de la librería en la cabecera de nuestro documento HTML:

****

La clase 'Map' encapsula el mapa que se mostrará en pantalla. Para ello habrá que crear una instancia de la clase.

****

Al constructor de la clase 'Map' hay que pasarle como primer argumento una referencia al elemento <div> de la página web donde queremos que se muestre el mapa. Lo que se le pasa es el valor del atributo 'id' del elemento <div>.

****

La clase Map tiene un atributo 'layers' que guarda las referencias a las capas que se añaden al mapa. Existen varios tipos de capas, en este caso añadiremos una capa tipo WMS (Web Map Services) a la cual le pasaremos el URL del servicio y los parametros de la petición 'GetMap' que queramos especificar, en el ejemplo accederemos al mapa mundi proporcionado por Metacarta.



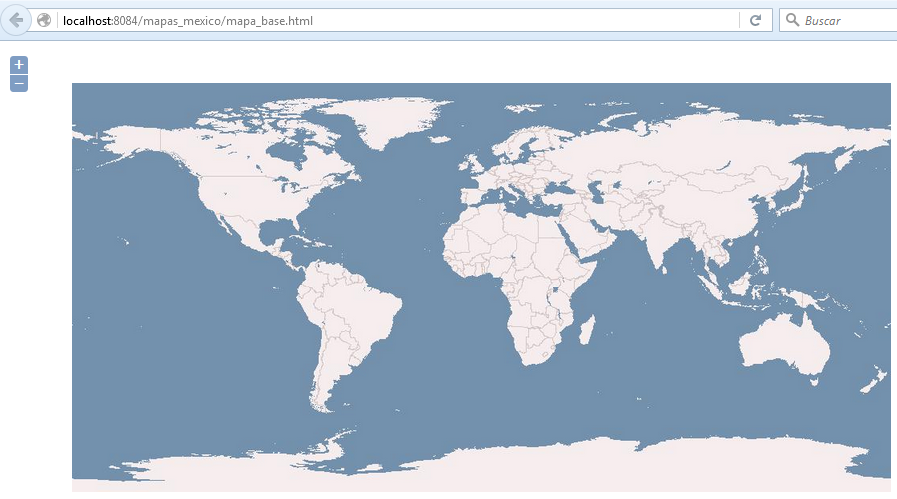
Agregaremos la capa al mapa



Por último fijaremos la posición del centro del mapa y el nivel de zoom inicial



El ejemplo corriendo se verá así:



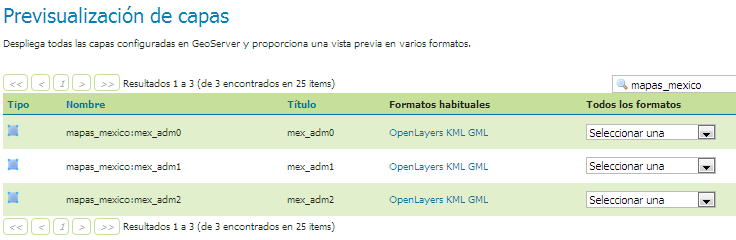
**Visualizar en Java Web las capas de Geoserver con OpenLayers**

Existen diferentes tipos de capas: [ArcGISCache.js](http://dev.openlayers.org/docs/files/OpenLayers/Layer/ArcGISCache-js.html), [ArcGIS93Rest](http://dev.openlayers.org/docs/files/OpenLayers/Layer/ArcGIS93Rest-js.html), [ArcIMS](http://dev.openlayers.org/docs/files/OpenLayers/Layer/ArcIMS-js.html), [Bing](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Bing-js.html), [Boxes](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Boxes-js.html), [EventPane](http://dev.openlayers.org/docs/files/OpenLayers/Layer/EventPane-js.html), [FixedZoomLevels](http://dev.openlayers.org/docs/files/OpenLayers/Layer/FixedZoomLevels-js.html), [GeoRSS](http://dev.openlayers.org/docs/files/OpenLayers/Layer/GeoRSS-js.html), [Google](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Google-js.html), [Google.v3](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Google/v3-js.html), [Grid](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Grid-js.html), [HTTPRequest](http://dev.openlayers.org/docs/files/OpenLayers/Layer/HTTPRequest-js.html), [Image](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Image-js.html), [KaMap](http://dev.openlayers.org/docs/files/OpenLayers/Layer/KaMap-js.html), [KaMapCache](http://dev.openlayers.org/docs/files/OpenLayers/Layer/KaMapCache-js.html), [MapGuide](http://dev.openlayers.org/docs/files/OpenLayers/Layer/MapGuide-js.html), [MapServer](http://dev.openlayers.org/docs/files/OpenLayers/Layer/MapServer-js.html), [Markers](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Markers-js.html), [PointGrid](http://dev.openlayers.org/docs/files/OpenLayers/Layer/PointGrid-js.html), [PointTrack](http://dev.openlayers.org/docs/files/OpenLayers/Layer/PointTrack-js.html), [SphericalMercator](http://dev.openlayers.org/docs/files/OpenLayers/Layer/SphericalMercator-js.html), [Text](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Text-js.html), [TileCache](http://dev.openlayers.org/docs/files/OpenLayers/Layer/TileCache-js.html), [TMS](http://dev.openlayers.org/docs/files/OpenLayers/Layer/TMS-js.html), [Vector](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Vector-js.html), [Vector.RootContainer](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Vector/RootContainer-js.html), [WMS](http://dev.openlayers.org/docs/files/OpenLayers/Layer/WMS-js.html), [WMTS](http://dev.openlayers.org/docs/files/OpenLayers/Layer/WMTS-js.html), [WorldWind](http://dev.openlayers.org/docs/files/OpenLayers/Layer/WorldWind-js.html), [XYZ](http://dev.openlayers.org/docs/files/OpenLayers/Layer/XYZ-js.html), [Zoomify](http://dev.openlayers.org/docs/files/OpenLayers/Layer/Zoomify-js.html), [OSM](http://dev.openlayers.org/docs/files/OpenLayers/Layer/OSM-js.html), [UTFGrid](http://dev.openlayers.org/docs/files/OpenLayers/Layer/UTFGrid-js.html)

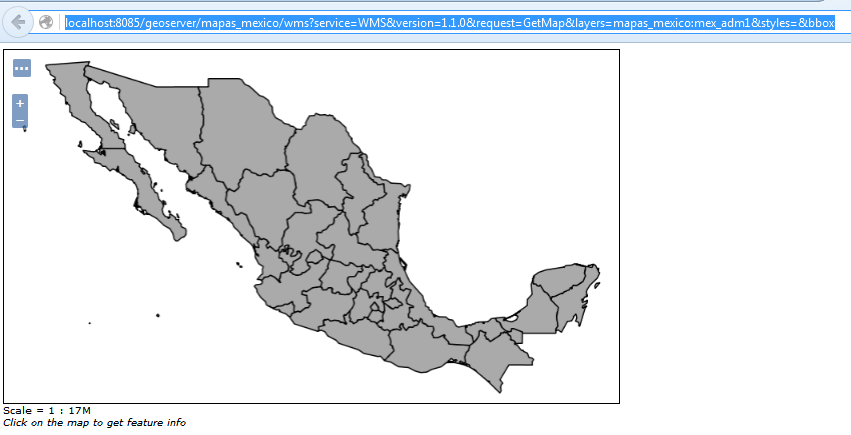
En este caso utilizaremos las capas de tipo WMS que son las que están pensadas para albergar los datos provenientes de consultas a los Web Map Services de acuerdo con las especificaciones del OGC ([Open Geospatial Consortium](http://www.opengeospatial.org/)).

Para acceder a un WMS lo primero que debemos conocer es la url del servicio de mapas es aquí donde hacemos uso de nuestros mapas definidos en Geoserver

Accedemos a Geoserver a Pre visualización de capas

****

Seleccionamos la visualización por Open Layers y nos fijamos en la URL



### A partir de esta crearemos un objeto tipo Layer en el que especificaremos:

### El nombre de la capa

### La url base para el WMS

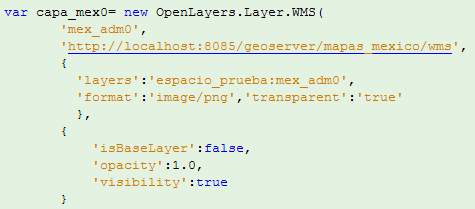
En nuestro ejemplo especificamos la url como:

“http://”+ host + puerto + ”/geoserver” + espacio de trabajo + “/wms”

### Los parámetros en formato de cadena con los pares clave/valor representando la llamada a GetMap (“layers”, “transparent”, “styles” y “format”)

### En nuestro ejemplo especificaremos layers como: espacio de trabajo + “:”+nombre de la capa, el formato como una imagen de tipo png, y levantaremos la bandera de transparent, por último el estilo es se especifica usando documentos de estilo XML SLD que se asocian con las capas de GeoServer en este ejemplo no lo contemplaremos, pero se puede obtener más información en el siguiente enlace: <http://docs.geoserver.org/stable/en/user/styling/sld-cookbook/index.html>

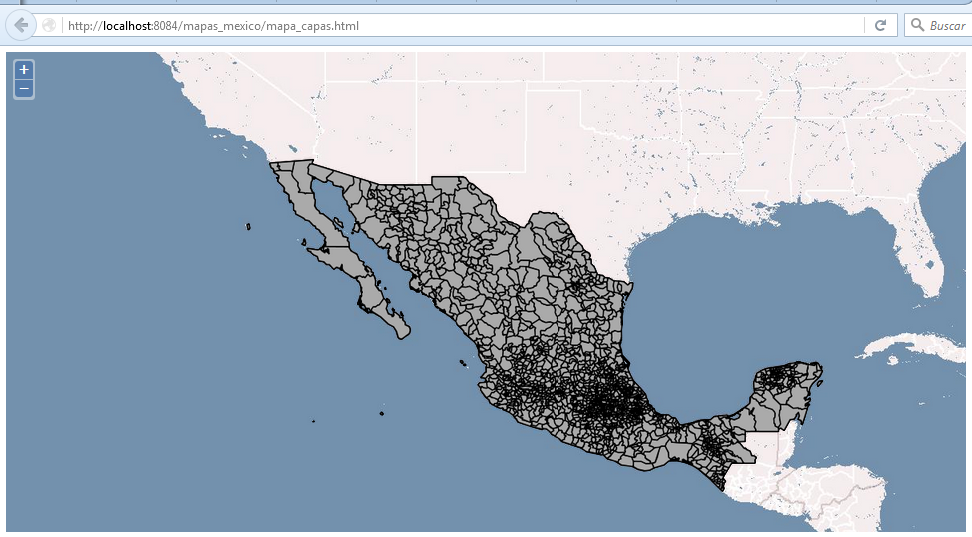
### Las opciones extra de cómo se visualizar a nuestra capa (“isBaseLayer”, “encodeBBOX”, “noMagic”, “yx”…) en nuestro ejemplo especificaremos que no es la capa base, la haremos visible con una opacidad de 1



Así vamos definiendo una a una todas las capas que queramos agregar, por ultimo hacemos uso del método addLayers para agregar las capas que definimos a nuestro mapa:



El ejemplo corriendo se verá así:



**Agregando Controles al Mapa**

Fuente:

<http://openlayers.bicimap.es/manualOpenLayers.html>