

Manual de instalación del servidor Sierra

| | |
|-------------------------------------------------------------------------------------------|----------|
| 1. Contenido y preparación de la instalación | 1 |
| 2. Instalación de PostgreSQL con PostGIS para el soporte de datos espaciales. | 2 |
| 3. Instalación de Apache Geronimo | 3 |
| 4. Creación de las bases de datos | 5 |
| 5. Instalación de las aplicaciones web | 8 |

Capítulo 1. Contenido y preparación de la instalación

En el disco de instalación encontramos lo siguiente:

- `proj-4.5.0.tar.gz`: paquete para la instalación de PROJ.
- `geos-3.0.0rc4.tar.bz2`: paquete para la instalación de GEOS.
- `postgis-1.2.1.tar.gz`: paquete para la instalación de PostGIS.
- `geronimo-tomcat6-jee5-2.0.1-bin.tar.gz`: paquete para la instalación de Apache Geronimo.
- `instalaPG`: script que instala el sistema de gestión de base de datos PostgreSQL con la extensión geográfica PostGIS.
- `instalaAG`: script que instala Java en el sistema y además instala y configura Apache Geronimo.
- `geronimo.sh`: script utilizado para la configuración del arranque en el inicio del sistema de Apache Geronimo.
- `creabd`: directorio con los ficheros necesarios para crear una base de datos espacial.
- `lib`: directorio con la librería de acceso a la base de datos con información espacial de Ruralcaja.
- `app`: directorio donde se encuentran las aplicaciones web.
- `doc`: directorio donde se encuentra documentación adicional.
- `Instalacion.pdf`: este manual.

Para realizar la instalación deberemos copiar el contenido del disco de instalación en el disco duro y seguir los pasos indicados en cada uno de los siguientes capítulos. Esta instalación se debe realizar en un **Ubuntu 6.06 Desktop LTS**.

Capítulo 2. Instalación de PostgreSQL con PostGIS para el soporte de datos espaciales.

Para la instalación de PostgreSQL con PostGIS deberemos situarnos en el directorio donde hayamos copiado el contenido del disco de instalación y ejecutar el script `instalaPG` como usuario `root`. Por ejemplo:

\$ sudo ./instalaPG # En este caso se pide la contraseña del usuario actual

El script se encarga inicialmente de instalar PostgreSQL 8.1 de los repositorios de `apt-get` y de configurar el acceso general a la base de datos. Posteriormente descomprime, compila e instala (por este orden) los paquetes PROJ, GEOS y PostGIS. Finalmente copia el directorio de PostGIS en `/usr/src/postgis`, necesario posteriormente para la creación de bases de datos. Sólomente le tenemos que indicar al poco tiempo de iniciar la instalación el password que queremos para el usuario `postgres`, que es el usuario administrador de PostgreSQL. El resto de tareas las realiza automáticamente.

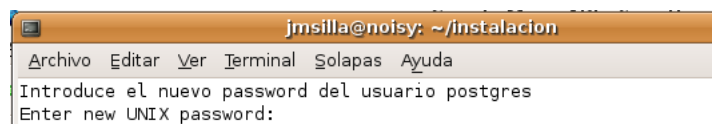


Figura 2.1. Momento de la instalación en el que debemos introducir el password del usuario `postgres`

Capítulo 3. Instalación de Apache Geronimo

En la instalación de Geronimo se dispone del script `instalaAG` que realiza la instalación de Java, necesario para que Geronimo funcione, e instala y configura Geronimo para que se inicie con el arranque del sistema. De la misma manera que con PostgreSQL, deberemos ejecutar el script como usuario `root`:

\$ sudo ./instalaAG

En este caso la instalación es también automática, salvo que debemos aceptar las licencias de Java que nos aparecerán en dos ventanas azules y rojas.

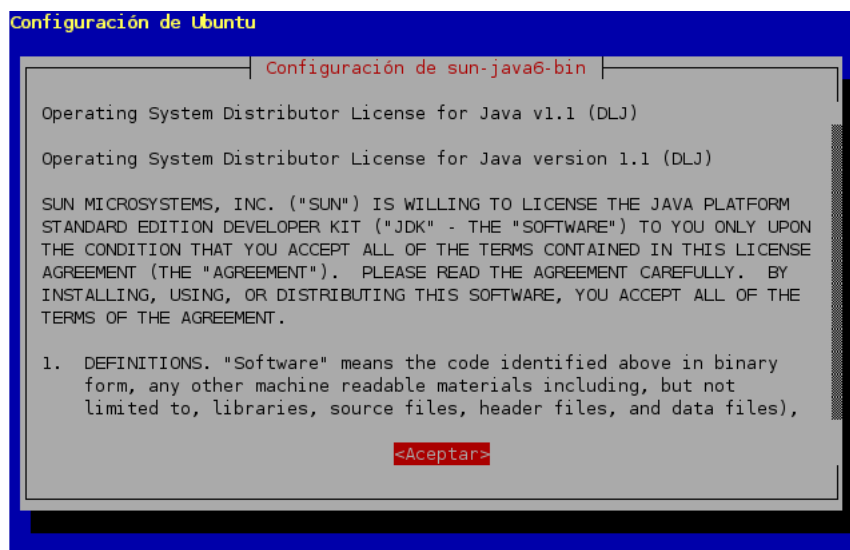


Figura 3.1. Licencias de Java

El script instala Geronimo en `/usr/local/geronimo`, configura el script de arranque en `/etc/init.d/geronimo.sh` y arranca el servidor. Para comprobar que la instalación ha funcionado correctamente deberemos abrir la dirección `http://localhost:8080` desde un navegador, y el navegador nos debe mostrar la pantalla de bienvenida de Geronimo.

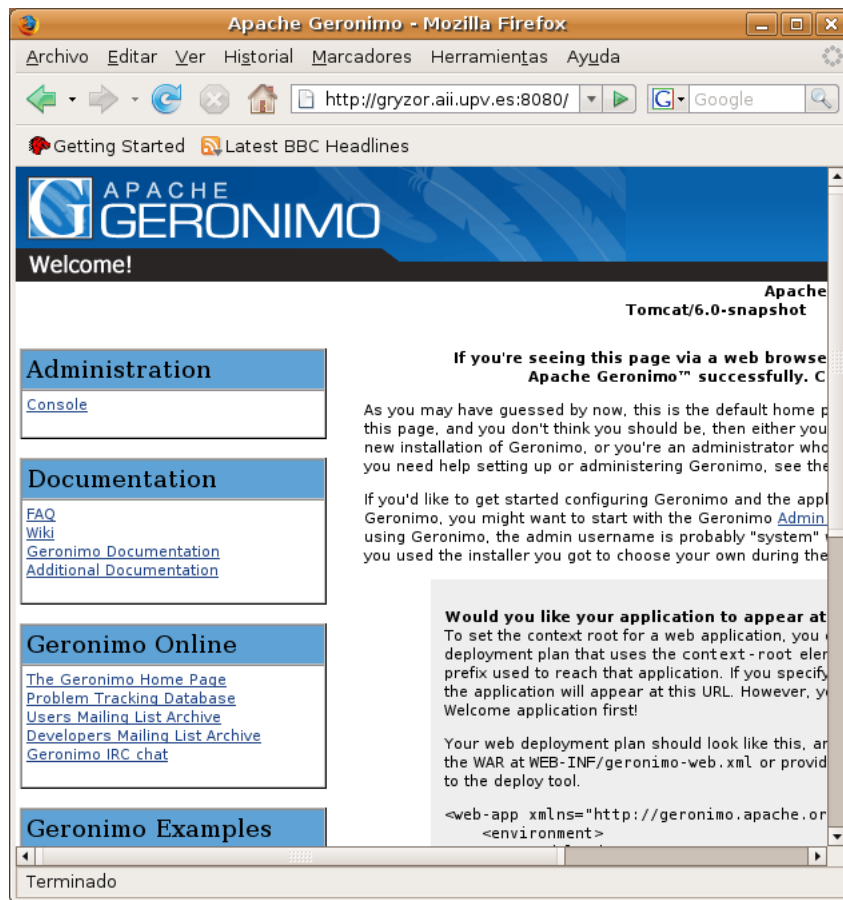


Figura 3.2. Pantalla de inicio

Capítulo 4. Creación de las bases de datos

Previamente a la ejecución del script de creación de base de datos, es conveniente leer el documento `contraseñas.txt` que se encuentra en el directorio `doc`, por si se desea cambiar las contraseñas que hay puestas por defecto para los usuarios que accederán a las bases de datos.

Para la creación de una base de datos debemos situarnos en el directorio `creabd` y cambiar al usuario `postgres` (administrador de la base de datos). Para crear una base de datos ejecutaremos el script `creabd`, de la siguiente manera:

`./creabd nombre_base_de_datos directorio_fuentes_postgis`

- **nombre_base_de_datos**: es el nombre que queremos que tenga nuestra base de datos.
- **directorio_fuentes_postgis**: es el directorio donde hemos descomprimido los fuentes de PostGIS, y posteriormente hemos compilado e instalado el paquete. La instalación de PostGIS ha colocado este directorio en `/usr/src/postgis`, así que podemos utilizar esta ruta como segundo argumento. ¡Ojo! La ruta no debe contener una barra al final (utilizar `/usr/src/postgis` en lugar de `/usr/src/postgis/`).

En nuestro caso deberemos crear la base de datos `sierra`, con lo cual ejecutaríamos la siguiente orden como usuario `postgres`:

`$./creabd sierra /usr/src/postgis`

Una vez haya creado la base de datos procederemos a dar acceso a los dos usuarios que acceden a ella: `reader` y `sradmin`. El acceso a las bases de datos PostgreSQL se define en el fichero `/etc/postgresql/8.1/main/pg_hba.conf`.

Las reglas para dar permisos se estructuran de la siguiente manera:

| # | TYPE | DATABASE | USER | CIDR-ADDRESS | METHOD |
|---|--------------------------|------------------------|----------------------|-------------------------------------|-------------------------|
| | <code>tipo_acceso</code> | <code>nombre_bd</code> | <code>usuario</code> | <code><lista_maquinas></code> | <code>metodo_aut</code> |

Donde:

- `tipo_acceso`: denota el lugar desde donde se accede a la base de datos. Los valores son: `local` (acceso a la base de datos desde el ordenador local donde se encuentra instalado PostgreSQL), `host` (acceso desde un host externo), `hostssl` y `hostnossl` (conexiones desde un host externo que utilicen/no utilicen ssl).
- `nombre_bd`: el nombre de la base de datos que es accedida. También se puede poner `all` para denotar acceso a cualquier base de datos.
- `usuario`: el usuario que está accediendo a la base de datos. Este valor también puede ser `all` para denotar acceso de cualquier usuario.

- `<lista_maquinas>`: rango de ip's de acceso para este registro. El formato es `IP/Número` para denotar un rango de IPs, en el que el número indican los bits a 1 por la izquierda de la máscara (por ejemplo, un 8 sería equivalente a una máscara `255.0.0.0`). Para especificar cualquier IP se usa el valor `0.0.0.0/0`. El formato de la IP puede ser también de IPv6. Este campo también se puede dividir en dos columnas: una dirección IP y una máscara en formato IP (por ejemplo `255.255.255.0`).
- `metodo_aut`: el método de autenticación utilizado para acceder. Existen varias posibilidades, pero las que más se utilizan son: `md5` (acceso mediante password cifrado con MD5) y `ident sameuser` (para los accesos desde local, utiliza la contraseña del usuario del sistema que está accediendo).

En nuestro caso concreto deberemos dejar acceso desde el exterior por `md5` tanto al usuario `reader` como al usuario `sradmin`. Por lo tanto deberemos editar el fichero `/etc/postgresql/8.1/main/pg_hba.conf` como usuario `root` (**`$ sudo gedit /etc/postgresql/8.1/main/pg_hba.conf`**) y añadir las siguientes líneas:

| # TYPE | DATABASE | USER | CIDR-ADDRESS | METHOD |
|--------|----------|---------|--------------|--------|
| host | sierra | reader | 0.0.0.0/0 | md5 |
| host | sierra | sradmin | 0.0.0.0/0 | md5 |
| local | sierra | reader | | md5 |
| local | sierra | sradmin | | md5 |

Con estas líneas estamos indicando que si hay un acceso desde cualquier host externo o desde el propio host a la base de datos `sierra` con el usuario `reader` o `sradmin`, y el usuario se autentifica con contraseña cifrada, permitir el acceso (siempre que la contraseña proporcionada sea correcta). Es conveniente que las líneas que añadamos se añadan al principio, ya que la primera línea que se encuentra que coincide con los parámetros de acceso es la única que considera. Si todo va bien se permite el acceso, pero si la línea coincide y se deniega el acceso ya no se miran las siguientes líneas del fichero y, por lo tanto, se deniega el acceso definitivamente (aunque en líneas inferiores se permita).

Una vez hayamos configurado correctamente el acceso de los dos usuarios deberemos reiniciar el servidor PostgreSQL. En el caso de que hayamos creado algún pool de conexiones desde Geronimo (ver capítulo siguiente), primero deberemos parar Geronimo y después PostgreSQL. Los comandos para parar y para reiniciar los servidores son:

`$ sudo /etc/init.d/geronimo.sh stop # Sólo si se ha creado algún pool de conexiones`

`# a alguna base de datos de nuestro servidor`

`$ sudo /etc/init.d/postgresql-8.1 restart # Reiniciando PostgreSQL`

`$ sudo /etc/init.d/geronimo.sh start # Rearrancando Geronimo sólo si es necesario`

Para comprobar que la base de datos se ha instalado correctamente y que se permite el acceso de los usuarios, se puede probar a entrar como usuario `reader` con la siguiente orden:

\$ psql -U reader -d sierra

Se introduce la contraseña de `reader` y si es correcta se accede a la consola de PostgreSQL. Una vez allí se puede probar a hacer una selección en la tabla `ubicaciones` con la siguiente orden:

sierra=> SELECT * FROM ubicaciones;

Si la consulta devuelve algún resultado es que la base de datos ha sido creada correctamente, y el acceso de los usuarios ha sido configurado adecuadamente.

Capítulo 5. Instalación de las aplicaciones web

Previamente al despliegue de las aplicaciones web deberemos crear los pools de conexiones a la base de datos en Geronimo que son utilizados por estas aplicaciones. Deberemos crear dos pools de conexiones: uno que acceda mediante el usuario `reader` para realizar operaciones de sólo lectura, y otro que acceda mediante el usuario `sradmin` para realizar lecturas, inserciones, borrados y actualizaciones.

Para crear un pool de conexiones, deberemos acceder a la consola de Geronimo entrando en la dirección `http://localhost:8080` y haciendo click en la opción "Console" del menú Administration. Accederemos con el usuario `system` y con la contraseña correspondiente. Si es la primera vez que accedemos la contraseña por defecto es 'manager'. Para cambiar la contraseña del usuario `system` haremos click en "Console Realms" en el menú de la izquierda. Nos aparecerá el usuarios `system` y el grupo `admin`. Pulsamos en "Edit" en el usuario `system`, y en los dos cuadros que aparecen escribimos y confirmamos la nueva contraseña. Luego pulsamos en Update y, si todo ha ido bien, el cambio de contraseña estará realizado.

Para instalar un Database Pool, en el menú de la izquierda hacemos click en "Database Pools". Después hacemos click sobre el enlace "Using the Geronimo database pool wizard" para iniciar el asistente de creación del Database Pool.



Figura 5.1. Pantalla de listado y creación de Database Pools

En la casilla "Name of Database Pool" escribimos el nombre que queramos para el Database Pool, y en la lista desplegable de abajo seleccionamos el tipo de base de datos, en nuestro caso PostgreSQL.

Como Geronimo no tiene instalado en su repositorio el driver de PostgreSQL, en la siguiente pantalla hacemos click en "Download a driver". En la lista "Select Driver" seleccionamos PostgreSQL 8.1, esperamos a que el driver esté descargado y volvemos a la pantalla anterior, pero ahora tenemos seleccionado el driver que acabamos de descargar. La descarga del driver sólo deberemos hacerla cuando creamos el primer datapool de PostgreSQL 8.1. Cuando creamos más datapools de PostgreSQL 8.1 simplemente seleccionaremos el driver de nombre `postgresql/postgresql-8.1/404.jdbc3/jar` de la lista.

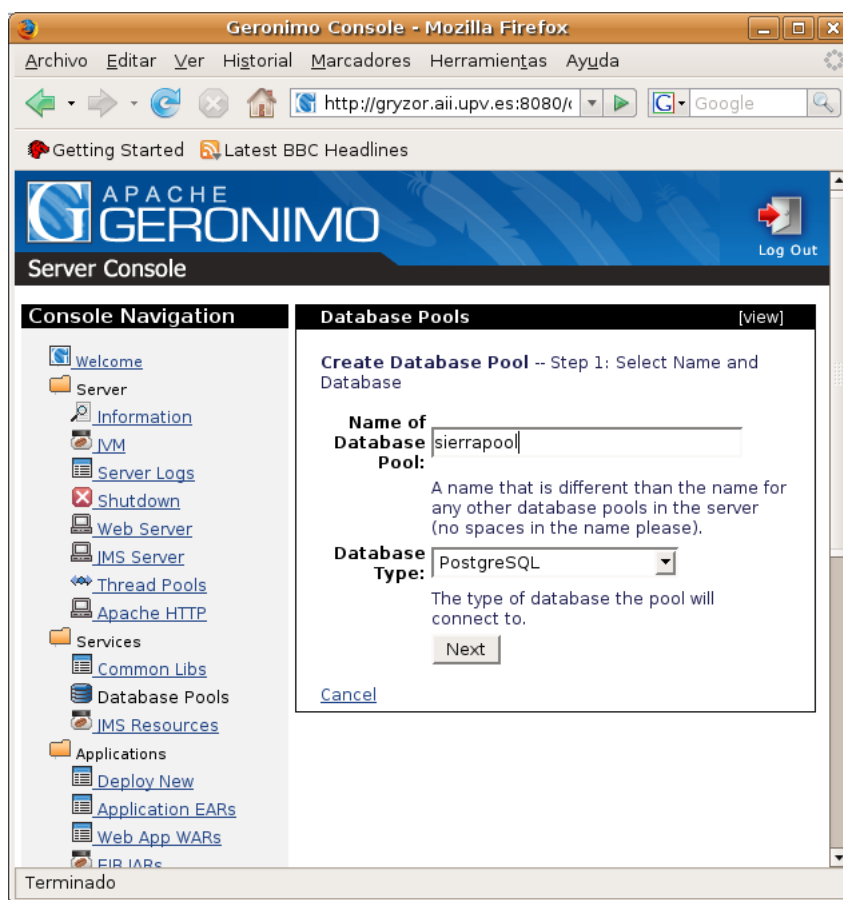


Figura 5.2. Pantalla de selección de driver de base de datos y de nombre del Database Pool

Rellenamos los campos de acceso a nuestra base de datos (nombre de usuario de acceso, contraseña, host, nombre de base de datos y puerto). Cuando hayamos rellenado los campos pulsamos "Next".

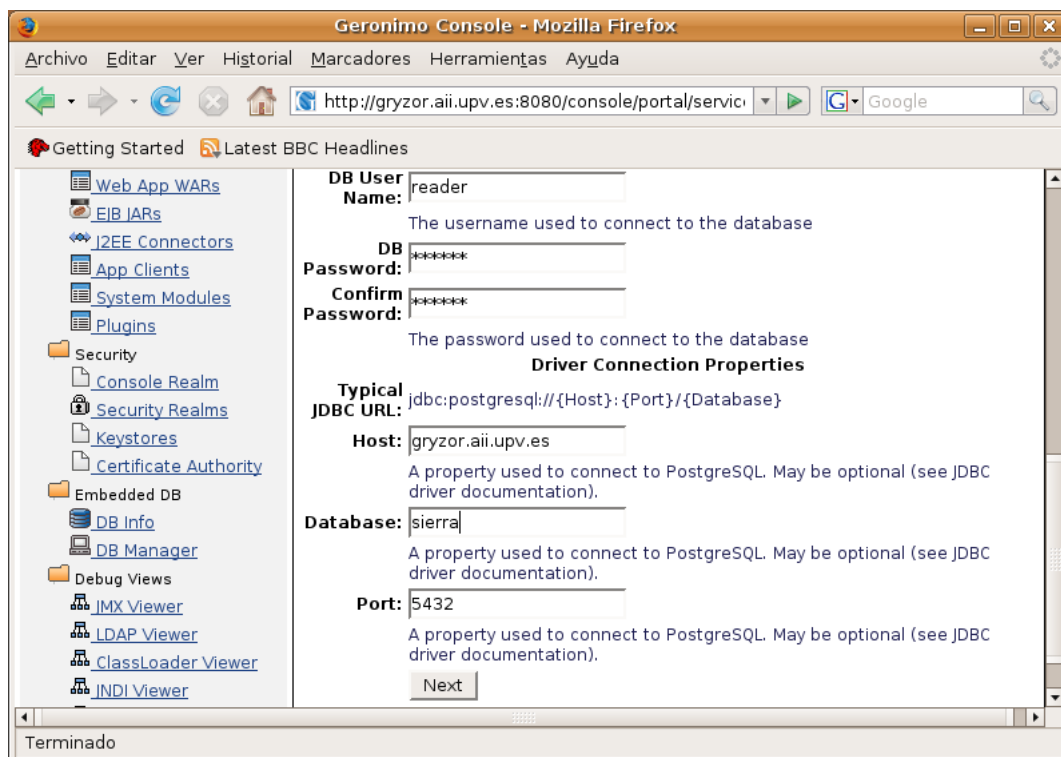


Figura 5.3. Pantalla de introducción de datos sobre la conexión a la base de datos

En la siguiente pantalla hacemos click en "Test connection" para probar que la conexión del Database Pool que está siendo creado funciona correctamente. Si todo ha funcionado correctamente nos aparecerá el mensaje "Test result: Connected to PostgreSQL 8.1.8". En este caso, para finalizar la creación del Database Pool hacemos click en "Deploy". Después nos aparece una lista con todos los Database Pools, debemos comprobar que el nuestro aparece en la lista.

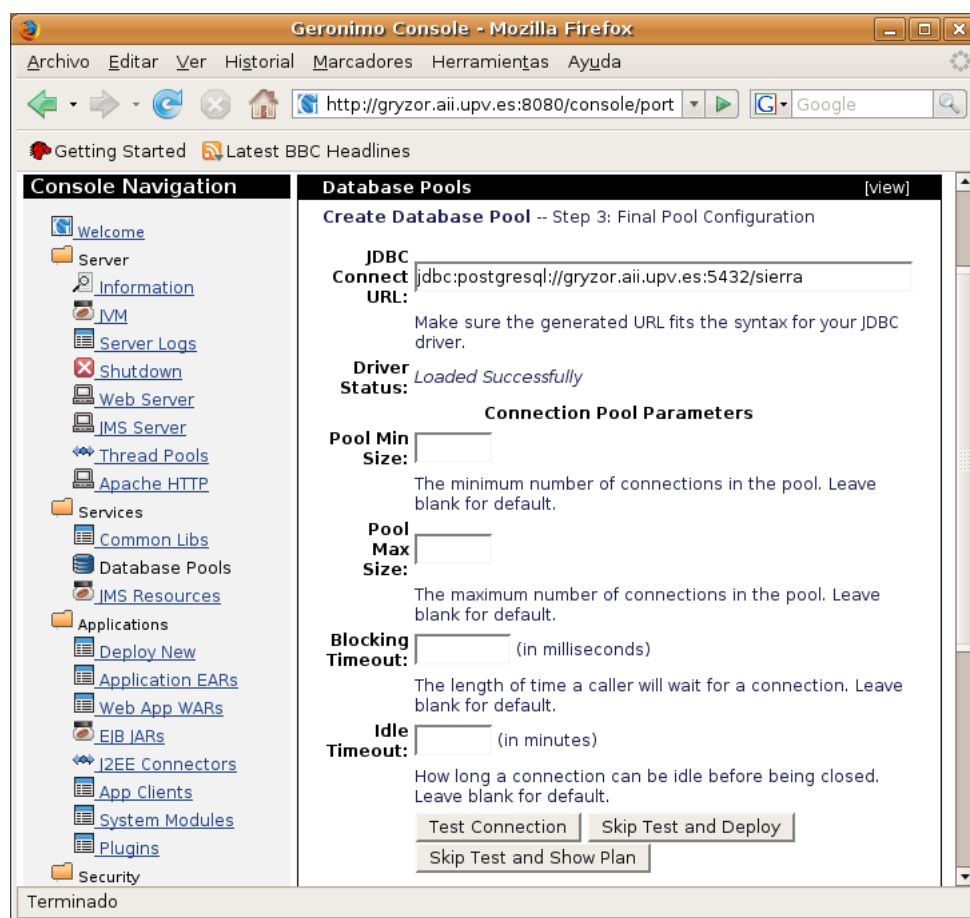


Figura 5.4. Pantalla de configuración del Database Pool

En nuestro caso particular, deberemos crear dos pools de conexiones con la siguiente información:

sierrapool:

- Nombre del pool: sierrapool
- Usuario de la base de datos: reader
- Contraseña: (ver directorio de documentación)
- Host: localhost
- Nombre de la base de datos: sierra
- Puerto: 5432 (el que viene por defecto en el asistente)

adminsiera:

- Nombre del pool: adminsiera
- Usuario de la base de datos: sradmin

- Contraseña: (ver directorio de documentación)
- Host: localhost
- Nombre de la base de datos: sierra
- Puerto: 5432 (el que viene por defecto en el asistente)

¡Atención! Es importante crear los pools de conexiones con los nombres indicados ya que si se cambia el nombre de los pools las aplicaciones que se desplegarán posteriormente no funcionarán correctamente.

Una vez se han creado los dos pools de conexiones, pasaremos a desplegar la librería de acceso a la base de datos espacial en Geronimo. Para ello haremos click en "Common Libs", en el menú de la izquierda de la consola de Geronimo. Nos aparecerá un formulario con varios campos. El primero es el campo que indica la ruta de la librería que queremos instalar. Pulsamos en "Examinar" y seleccionamos la librería que se encuentra en el directorio `lib`, de nombre `sierra-api-1.0.jar`. Los campos los debemos rellenar de la siguiente manera:

- Group: `es.upv.ai2.sierra`
- Artifact: `sierra-api`
- Version: `1.0`
- Type: `jar`

Posteriormente haremos click en "Install" para que la librería sea instalada.

¡Atención! Es importante que se introduzcan los datos anteriores sin cambiar los nombres, ya que cambiarlos impediría que las aplicaciones web funcionaran correctamente.

Para comprobar que la librería ha sido instalada correctamente, en la lista que aparece debajo de "Current Repository Entries" debe aparecer la línea `es.upv.ai2.sierra/sierra-api/1.0/jar`.

Una vez la librería de acceso a la base de datos con información espacial de Ruralcaja ha sido instalada, procederemos a subir las aplicaciones web. Estas aplicaciones se encuentran en el directorio `app`. Para subir las aplicaciones deberemos hacer click en la opción "Deploy new" del menú de la izquierda, submenú "Applications". Haremos click en el botón "Examinar" del campo "Archive" y seleccionaremos la aplicación web `demoCajero.war`. Una vez seleccionada haremos click en "Install", y si todo ha ido bien se nos mostrará en la parte superior el mensaje "The application was succesfully deployed" y también el mensaje "The application was succesfully started". Si algo no ha ido bien se nos mostrará una lista con los errores encontrados, con lo cual deberemos revisar los pasos anteriores. Para subir la aplicación web `sierra-admin.war` deberemos seguir los mismos pasos.



Figura 5.5. Pantalla de despliegue de aplicaciones web

Las aplicaciones web estarán disponibles en las rutas `http://direccion_del_servidor:8080/demoCajero` y `http://direccion_del_servidor:8080/sierra-admin`.

Finalmente deberemos cambiar la clave de Google Maps de las dos aplicaciones, ya que esta clave depende del nombre del servidor en el que instalamos las aplicaciones. Para ver qué clave de Google Maps necesitamos tener una cuenta en Google (se puede crear una cuenta en `https://www.google.com/accounts/NewAccount?service=local`) y habernos logueado con ella. Después de estar logueados vamos a la página `http://www.google.com/apis/maps/signup.html`. En la parte inferior de la página escribimos la URL de acceso a una de las aplicaciones web y marcamos la casilla de confirmación de que hemos leído la sección "Terms and Conditions". Después hacemos click en "Generate API Key".

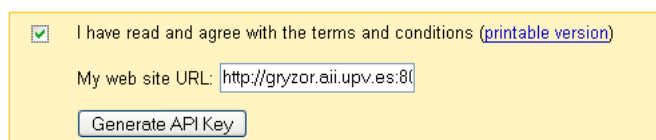


Figura 5.6. Introducción de los datos para la generación de la clave de Google Maps

En la página siguiente nos aparece la clave del API que debemos de utilizar para nuestra aplicación web. Deberemos repetir este proceso para las dos aplicaciones web que tenemos.

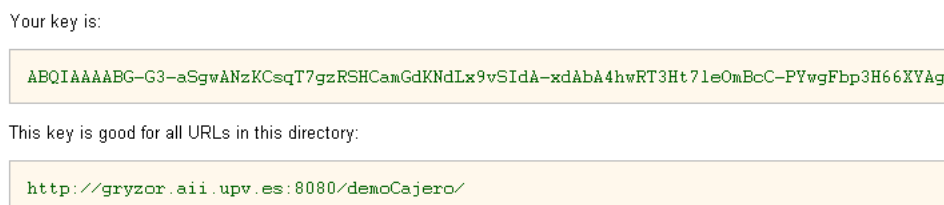


Figura 5.7. Clave de Google Maps

Una vez tengamos las claves de Google Maps deberemos aplicarlas a nuestras aplicaciones web. Para ello entraremos en la página `http://direccion_del_servidor:8080/nombre_aplicacion/cambiaClave.html`. Nos aparecerá un formulario en el que deberemos poner la clave correspondiente a la aplicación web que estamos actualizando y hacer click en el botón "Cambiar". Si todo ha ido correctamente se nos mostrará un mensaje indicando que todo ha ido bien.

Finalmente ya tenemos las aplicaciones web funcionando perfectamente y mostrando los mapas correctamente.