

Manual de instalación

Sistema integral de gestión de proyectos, GCBA





Historial de revisiones

|  |  |  |
| --- | --- | --- |
| VERSIÓN | FECHA | AUTOR – PRINCIPALES CAMBIOS |
| 0.1 | 08/08/2016 | Federico Bucich – Versión Inicial |

Índice

[1 Introducción 3](#_Toc458525461)

[1.1 Propósito de este documento 3](#_Toc458525462)

[1.2 Definición de términos y abreviaturas 3](#_Toc458525463)

[2 Consideraciones previas 3](#_Toc458525464)

[3 Instalación de Java 8 3](#_Toc458525465)

[4 Instalación de apache2 4](#_Toc458525466)

[5 Instalación de tomcat 5](#_Toc458525467)

[6 Base de Datos – MariaDB 6](#_Toc458525468)

[7 Correr la Aplicación 7](#_Toc458525469)

# Introducción

## Propósito de este documento

Este documento detalla los pasos a seguir para llevar a cabo la instalación de un ambiente para la plataforma Sistema de Gestión de Proyectos GCBA.

## Definición de términos y abreviaturas

| Término o Abreviatura | Definición |
| --- | --- |
|  |  |

# Consideraciones previas

El hardware está basado en una arquitectura x86\_64, con 40 Gb de disco y 3 Gb de memoria.

El sistema operativo base a utilizado es Ubuntu 15.10 (wily).

Datos de conexión (acceso por ssh):

* Servidor: 10.30.10.104
* User: proyectosba
* Pass: hexacta

# Instalación de Java 8

Ingresar por ssh al ambiente y ejecutar:

sudo su

apt-get update

apt-get install openjdk-8-jdk

# Instalación de apache2

Vamos a utilizar Apache2 como webserver para servir la aplicación frontend hecha en AngularJS.

Ingresar por ssh al ambiente y ejecutar:

sudo su

apt-get install apache2

Con el comando anterior se instala Apache2 y está configurado para tomar como document root /var/www/html y escuchando en el puerto 80.

La configuración del sitio se encuentra en /etc/apache2/sites-enabled/000-default.conf

<VirtualHost \*:80>

# The ServerName directive sets the request scheme, hostname and port that

# the server uses to identify itself. This is used when creating

# redirection URLs. In the context of virtual hosts, the ServerName

# specifies what hostname must appear in the request's Host: header to

# match this virtual host. For the default virtual host (this file) this

# value is not decisive as it is used as a last resort host regardless.

# However, you must set it for any further virtual host explicitly.

#ServerName www.example.com

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

<Directory /var/www/html>

RewriteEngine on

# Don't rewrite files or directories

RewriteCond %{REQUEST\_FILENAME} -f [OR]

RewriteCond %{REQUEST\_FILENAME} -d

RewriteRule ^ - [L]

# Rewrite everything else to index.html to allow html5 state links

RewriteRule ^ index.html [L]

</Directory>

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,

# error, crit, alert, emerg.

# It is also possible to configure the loglevel for particular

# modules, e.g.

#LogLevel info ssl:warn

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

# For most configuration files from conf-available/, which are

# enabled or disabled at a global level, it is possible to

# include a line for only one particular virtual host. For example the

# following line enables the CGI configuration for this host only

# after it has been globally disabled with "a2disconf".

#Include conf-available/serve-cgi-bin.conf

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

La url para acceder desde el browser es: http://hxv-solicitudesba/

Para iniciar y detener iniciar el servicio, utilizar:

/etc/init.d/apache2 start

/etc/init.d/apache2 stop

/etc/init.d/apache2 restart

# Instalación de tomcat

Vamos a utilizar Apache Tomcat para servir el contenido de la aplicación backend, escrita en Java y que expone WebServices REST.

Vamos a bajar el Tomcat, lo descomprimimos y lo copiamos a /opt:

sudo su

wget http://mirrors.dcarsat.com.ar/apache/tomcat/tomcat-8/v8.0.32/bin/apache-tomcat-8.0.32.zip

unzip apache-tomcat-8.0.32.zip

cp -r apache-tomcat-8.0.32 /opt

# Base de Datos – MariaDB

Para instalar MariaDB:

sudo apt-get install mariadb-server

Luego de instalarla mostrara una ventana y nos pedirá definir una contraseña de root

Validaremos que el servicio de la base de datos está corriendo correctamente:

sudo systemctl status mariadb.service

#sino lo arracamos

sudo systemctl start mariadb.service

* Nos conectamos con root a la base

mysql -u root -p

* crear la base de datos CREATE DATABASE proyectos\_ba\_generated;

CREATE DATABASE proyectos\_ba\_generated;

* crear un usuario y darle permisos para poder leer y escribir en toda la base anteriormente creada. Luego actualizar el archivo colocado en /etc/proyectoba/be/application.properties con el usuario y contraseña (spring.datasource.username=<usuario>, spring.datasource.password=<contraseña>)

* para crear usuario primero correr para generar el hash de la nueva contraseña. Copiar todo el numero incluyendo el asterisco (\*).

MariaDB [(none)]> SELECT PASSWORD('admin');

+-------------------------------------------+

| PASSWORD('admin') |

+-------------------------------------------+

| \*4ACFE3202A5FF5CF467898FC58AAB1D615029441 |

+-------------------------------------------+

* conectarse a la base:

mysql -u root -p

* creamos el usuario y ponemos el hash anteriormente generado(incluyendo el ‘\*’)

CREATE USER 'admin'@'localhost' IDENTIFIED BY PASSWORD 'poner el HASH con asterisco';

#Verificamos el usuario

MariaDB [(none)]> SELECT user, host FROM mysql.user;

+---------+--------------------+

| user | host |

+---------+--------------------+

| admin | localhost |

+---------+--------------------+

* le damos los permisos para poder usar nuestra base:

GRANT ALL PRIVILEGES ON proyectos\_ba\_generated.\* TO admin@localhost WITH GRANT OPTION;

#verificamos el usuario creado, salimos de la base e intentamos loguearnos con el user y password.

mysql -u admin –p proyectos\_ba\_generated

* correr el script de creación y carga de tablas:

mysql -u admin -p proyectos\_ba\_generated < proyectos\_ba\_completa.sql

# para verificar, nos logueamos a las base proyectos\_ba\_generated y hacemos un select de la tabla juridiccion y nos tiene que traer datos.

mysql -u admin –p proyectos\_ba\_generated

select nombre from jurisdiccion;

# Instalacion ActiveMQ

Creamos la carpeta en opt, descomprimir y copiar el ActiveMQ a la carpeta opt. Y le damos los permisos 755 a la carpeta

sudo mkdir /opt/apache-activemq

sudo tar -xvf /usr/local/src/apache-activemq-5.14.0-bin.tar.gz ´

sudo cp -r /usr/local/src/apache-activemq-5.14.0/\* /opt/apache-activemq/

sudo chmod 755 /opt/apache-activemq

#arracamos activeMQ

sudo /opt/apache-activemq/bin/activemq start

#verificamos que levanto de forma correcta

ps aux | grep activemq

# Correr la Aplicación

**Carpetas de configuración**:

Se deberán crear las siguientes carpetas:

- Carpetas de configuración: con permiso de lectura y ejecucion.

#carpetas de configuración.

sudo mkdir /etc/proyectosBA

sudo mkdir /etc/proyectosBA/ds

sudo mkdir /etc/proyectosBA/be

sudo chmod 755 -R /etc/proyectosBA

# carpetas de logs.

sudo mkdir /var/proyectosBA

sudo mkdir /var/proyectosBA/be

sudo mkdir /var/proyectosBA/be/log

sudo mkdir /var/proyectosBA/ds

sudo mkdir /var/proyectosBA/ds/log

sudo chmod 777 -R /var/proyectosBA

- En /etc/proyectosBA/ds colocar los archivos application.properties, template.xlsx correspondientes de dicha aplicación.

- /etc/proyectosBA/be colocar los archivos application.properties, proyectosBA-BE.conf, proyectosBA-BE.service correspondientes.

cp ds/application.properties /etc/proyectosBA/ds/

cp ds/template.xlsx /etc/proyectosBA/ds/

cp be/application.properties /etc/proyectosBA/be/

cp be/proyectosBA-BE.conf /etc/proyectosBA/be/

cp be/proyectosBA-BE.service /etc/proyectosBA/be/

# volvemos a correr los permisos para los archivos copiados.

sudo chmod 755 -R /etc/proyectosBA

**Aplicación Data-Services**:

* Copiar el archivo proyectosBA-DS.war a la carpeta webapp del Tomcat

cp proyectosBA-DS.war /opt/apache-tomcat-8.0.32/webapp

# o si el tomcat esta como servicio, correr para ver donde están las carpetas respectivas.

ll /usr/share/tomcat/

bin

conf -> /etc/tomcat

lib -> /usr/share/java/tomcat

logs -> /var/log/tomcat

temp -> /var/cache/tomcat/temp

webapps -> /var/lib/tomcat/webapps

work -> /var/cache/tomcat/work

# en este caso copiaremos el war a /var/lib/tomcat/webapps

cp ds/proyectosBA-DS.war /var/lib/tomcat/webapps

* Copiar el archive con las configuraciones para correr la aplicacion en el tomcat

cp setenv.sh /opt/apache-tomcat-8.0.32/bin

# o a /usr/share/tomcat/bin

cp ds/setenv.sh /usr/share/tomcat/bin

# darle los permisos al archivo copiado

sudo chmod 755 -R /usr/share/tomcat/bin/

# y al usuario tomcat

sudo chown root:tomcat -R /usr/share/tomcat/bin/

* Arrancar el tomcat (sin sudo):

/opt/apache-tomcat-8.0.32/bin/startup.sh

# o como servicio

systemctl start tomcat.service

**Aplicación Backend**:

* Copiar a la carpeta /etc/proyectosBA/be:

cp proyectosBA-BE.jar /etc/proyectosBA/be

* copiar el archivo con la descripción del servicio:

cp /etc/proyectosBA/be/proyectosBA-BE.service /etc/systemd/system

* Luego para habilitar la aplicación como servicio:

systemctl enable proyectosBA-BE.service

* para ver el status y arrancar la aplicacion:

systemctl status proyectosBA-BE.service

# y para arrancarlo "start"

systemctl start proyectosBA-BE.service

* luego de iniciar ambas aplicaciones ir a ver el contenido de los logs:

# logs de la aplicacion Data Services (web).

tail –f /var/proyectosBA/ds/log/proyectosBA-DS.log

# logs de la aplicación Back End.

Tail –f /var/proyectosBA/be/log/proyectosBA-BE.log

