

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 CentOS 7 x86\_64 que es la versión free basada en RedHat.

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:

cd /opt/

# Tiene que ser Java 8, pero no es necesario que sea la 91.

wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u91-b14/jdk-8u91-linux-x64.tar.gz"

# Descomprimimos

tar xzf jdk-8u91-linux-x64.tar.gz

cd /opt/jdk1.8.0\_91/

# Agregamos la version para que use Java 8

alternatives --install /usr/bin/java java /opt/jdk1.8.0\_91/bin/java 2

# Configuramos para que la use. Elegimos la opción recién instalada.

alternatives --config java

# corremos para confirmer que quedo la version correcta instalada

Java –version

java version "1.8.0\_91"

Java(TM) SE Runtime Environment (build 1.8.0\_91-b14)

Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)

# 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 yum install httpd

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

#Copiamos el build a la carpeta html del apache

cp –R ui/proyectosba /var/www/html

# Luego copiamos el archivo de configuración

cp ui/proyectosba.conf /etc/httpd/conf.d

Para iniciar y detener iniciar el servicio, utilizar:

sudo systemctl start httpd.service

La url para acceder desde el browser es: <http://10.9.4.86/login>, siempre la primer carga tarda un poco mas porque tiene que levantar la apliacion.

# 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

# Puede ser cualquier versión de Tomcat 8.

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:

yum 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/pba/be/application.properties con el usuario y contraseña (spring.datasource.username=<usuario>, spring.datasource.password=<contraseña>) del paso 8.

* 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/pba

sudo mkdir /etc/pba/ds

sudo mkdir /etc/pba/be

sudo chmod 755 -R /etc/pba

# carpetas de logs.

sudo mkdir /var/pba

sudo mkdir /var/pba/be

sudo mkdir /var/pba/be/log

sudo mkdir /var/pba/ds

sudo mkdir /var/pba/ds/log

# carpeta para guardar los archivos adjuntos de los Proyectos

Sudo mkdir /var/pba/adjuntos

sudo chmod 777 -R /var/pba

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

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

cp ds/application.properties /etc/pba/ds/

cp ds/template.xlsx /etc/pba/ds/

cp be/application.properties /etc/pba/be/

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

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

# volvemos a correr los permisos para los archivos copiados.

sudo chmod 755 -R /etc/pba

**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, SIEMPRE AL COPIAR EL ARCHIVO MANTENER EL NOMBRE SIN LA VERSION, sino nos modificara las rutas de donde queda instalada la app. (proyectosBA-DS.war)

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

sudo chmod -R 755 /var/lib/tomcat/webapps

#le damos permisos a la carpeta para que pueda desplegar la aplicacion

sudo chmod -R 777 /usr/share/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 asignar la carpeta de nuevo al usuario tomcat, si cambio

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/pba/be:

# SIEMPRE AL COPIAR EL ARCHIVO MANTENER EL NOMBRE SIN LA VERSION, sino nos modificara las rutas de donde queda instalada la app. (proyectosBA-BE.jar)

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

# actualizar permisos

sudo chmod –R 755 /etc/pba

* copiar el archivo con la descripción del servicio:

cp /etc/pba/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/pba/ds/log/proyectosBA-DS.log

# logs de la aplicación Back End.

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

