

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

Se utilizará Java 8, verificar que la jdk y la jre las 2 usen dicha versión.

La versión del Tomcat será la 7.0.54 que es la versión mas reciente y homologada.

# 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 la aplicación angular compilada a la carpeta html del apache

# por ahora se encuentra en el repo git de la ASI, en la carpeta “source/configuracion/ui/proyectosba”

**cp –R proyectosba /var/www/html**

# Luego abrimos el archivo de configuración y hacemos que el Proxypass este

# apuntando a la URL correcta. Tambien aca podremos agregar la configuración para HTTPS.

# **ProxyPass /api/ http://<ip de la VM>:8080/proyectosBA-DS/**

# Por ultimo copiamos el archivo de configuración a la carpeta del Apache

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

Para iniciar y detener iniciar el servicio, utilizar:

sudo systemctl start httpd

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.

# Base de Datos – MariaDB

* 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.

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

**mysql -u admin –p proyectos\_ba\_generated**

# correr los scripts de creación y carga de tablas, el script esta en source/configuración/database:

**mysql -u admin -p proyectos\_ba\_generated < 1-SIGP-creacion\_de\_tablas.sql**

**mysql -u admin -p proyectos\_ba\_generated < 2-SIGP-carga\_inicial\_de\_tablas.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.

**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

# instalamos

**sudo yum install activemq activemq-client**

# arracamos activeMQ

**sudo systemctl start activemq**

#verificamos que levanto de forma correcta

**ps aux | grep activemq**

# Configuracion de las Aplicaciones

**Carpetas de configuración y logs**:

Se deberán crear las siguientes carpetas, verificar los permisos de lectura y escritura:

#carpetas de configuración. Solo lectura

**sudo mkdir /etc/pba**

**sudo mkdir /etc/pba/ds**

**sudo mkdir /etc/pba/be**

# carpetas de logs. Lectura y escritura.

**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. Lectura y escritura.

**sudo mkdir /var/pba/adjuntos**

# carpeta para guardar los archivos temporales de importación. Lectura y escritura.

**sudo mkdir /var/pba/archivos**

**Aplicación Angular:**

1. Ir a <https://www.google.com/recaptcha>.
2. Hacer click en 'Get reCAPTCHA'
3. Loguearnos con la cuenta sigppaneldecontrol@gmail.com / Hexacta2954
4. Ahí vamos a ver las KEYs ya creadas
5. Si vamos mas abajo vamos a poder generar una nueva
6. Agregar un label y los dominios permitidos(ejemplo para QA usamos: pgi.qa.gcba.gob.ar).
7. Registrar
8. Copiar el API Key.
9. Pegar el API Key en /etc/pba/ds/application.properties:
10. Para que la aplicación tome el nuevo valor vamos a tener que reiniciar el servicio del Tomcat.

Ejemplo:

configs.key.value=captcha;6LcSLyoTAAAAAN4zBx2nuOKIUwdQ0G1ZINLiqOkG

**Aplicación Data-Services (DS)**:

Copiar los archivos de configuración

# Archivo de configuracion de la aplicacion web que corre en el Tomcat. Se deberá

# renombrar sin el “.dist” y comparar la propiedades para no pisar ninguna especifica.

**cp /source/configuracion/ds/application.properties.dist /etc/pba/ds/ application.properties**

# Excel que se va a descargar desda la aplicacion web.

**cp /source/configuracion/ds/template.xlsx /etc/pba/ds/**

# En el archivo /etc/tomcat/conf/tomcat.conf agregaremos la variable CATALINA\_OPTS con el pefil que queremos correr la aplicación web. Al final del archvo

**CATALINA\_OPTS="-Dspring.profiles.active=prod"**

# Copiar el archivo proyectosBA-DS.war a la carpeta webapps del Tomcat. Este debe ser tomado del que genero Jenkins. El repositorio donde Jenkins guarda los archivos es [**https://asi-nexus.dev.gcba.gob.ar/nexus/content/repositories/releases/ar/gob/buenosaires/backend/**](https://asi-nexus.dev.gcba.gob.ar/nexus/content/repositories/releases/ar/gob/buenosaires/backend/)

SIEMPRE AL COPIAR EL ARCHIVO MANTENER EL NOMBRE SIN LA VERSION, sino nos modificara las rutas de donde queda instalada la app

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

* Arrancar el tomcat

**systemctl start tomcat**

# para verificar que levanto correcto la aplicacion revisamos el log a ver si hay algún error en

**/var/pba/ds/log/proyectosBA-DS.log**

**Aplicación Backend (BE)**:

- Copiar los archivos de configuración

# Archivo de configuracion de la aplicación que va a correr como servicio.

# En este archivo también debemos actualizar el user y password de la base de datos.

**cp source/configuracion/be/application.properties /etc/pba/be/application.properties**

# Este archivo contendra el perfil con el que corre la aplicacion

**cp source/configuracion/be/proyectosBA-BE.conf /etc/pba/be/**

# este archive es el que contendra la descripcion del servicio con la ruta a donde se encuentra el jar de la aplicación.

**cp source/configuracion/be/proyectosBA-BE.service /etc/systemd/system/**

# Copiar el archivo proyectosBA-BE.jar generado por Jenkins. El repo esta en [**https://asi-nexus.dev.gcba.gob.ar/nexus/content/repositories/releases/ar/gob/buenosaires/backend/**](https://asi-nexus.dev.gcba.gob.ar/nexus/content/repositories/releases/ar/gob/buenosaires/backend/)

SIEMPRE AL COPIAR EL ARCHIVO MANTENER EL NOMBRE SIN LA VERSION, y verificar que la ruta en el archivo poryectosBA-BE.service apunta a donde se encuentra el jar.

# y darle los permisos necesarios para poder ejecutarlo como servicio (sudo chmod 755 proyectosBA-BE.jar)

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

Luego para habilitar la aplicación como servicio:

**systemctl enable proyectosBA-BE.service**

# y para arrancarlo "start"

**systemctl start proyectosBA-BE**

# por ultimo verificamos que haya ningún error en los logs de la aplicación Back End.

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

