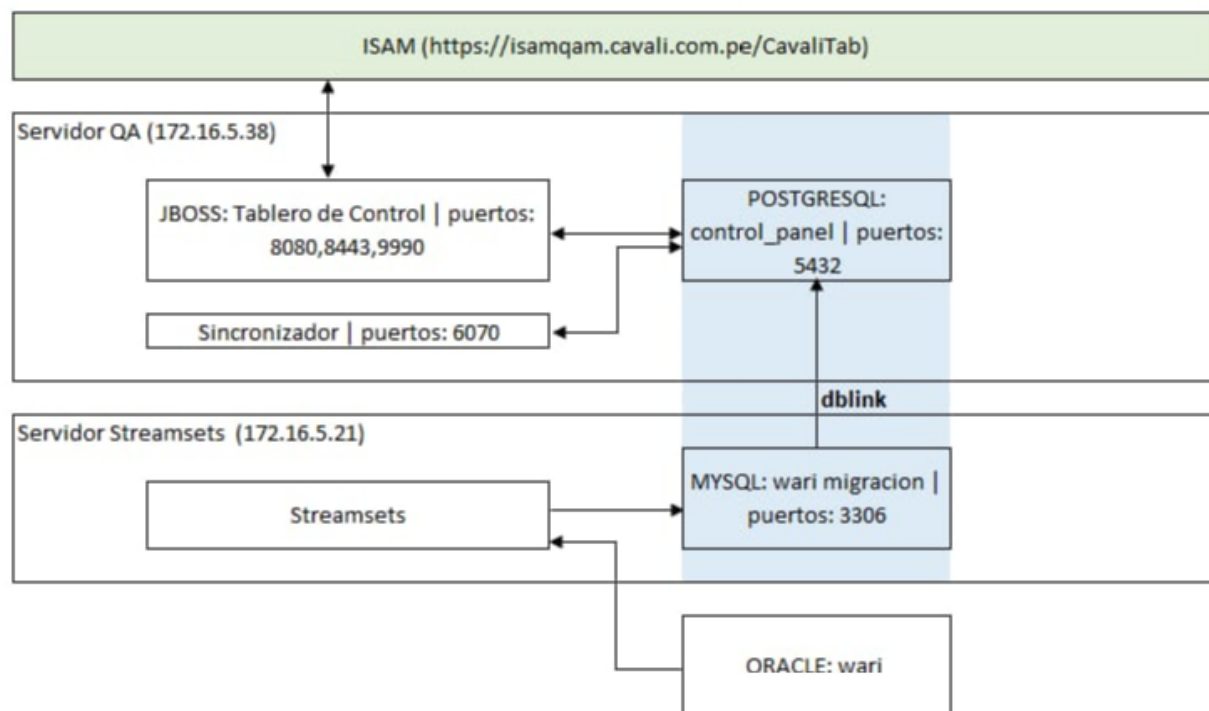


## 2.6. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA

### Requisitos de instalación

- Sistema operativo Red Hat Enterprise Linux con repositorios oficiales instalados en ruta /etc/yum.repos.d/redhat.repo
- Conexión a Internet para descarga de paquetes.
- Permisos de administrador (root) en el sistema operativo.

### REGLAS DE ACCESO



### Resumen de comandos

Comando
yum install
adduser <usr>
passwd <usr>
chmod +x <file-path>
chown <usr>:<group> <file-path>
chkconfig --add <executable-file-path>

service <executable> <action>
firewall-cmd <config>
iptables <config>
yum update

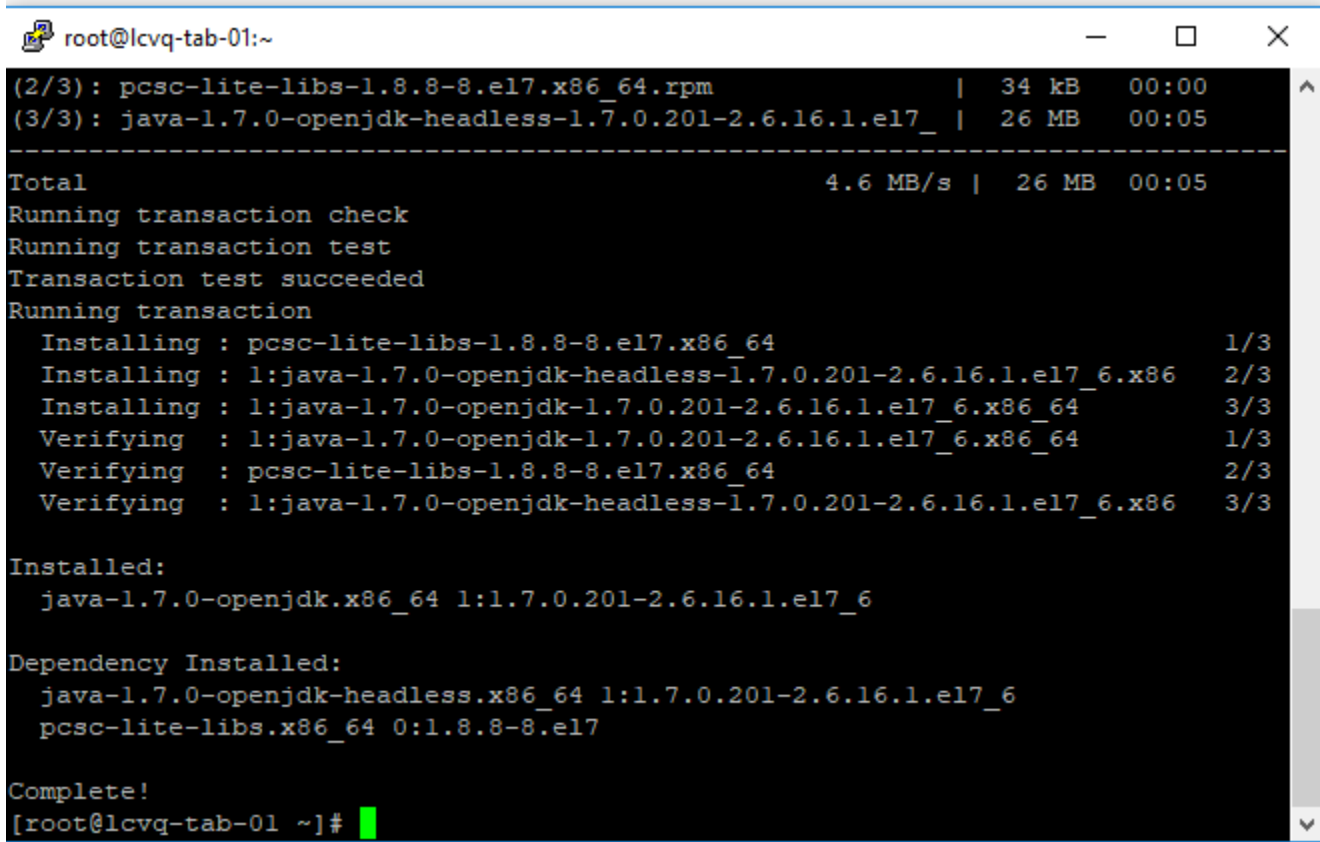
## Instalación Servidor 1

### Instalación de Java Runtime Environment 1.7

<b>Objetivo</b>	Instalar Java JRE para que el servicio JBoss pueda ser ejecutado
<b>Requisitos</b>	<ul style="list-style-type: none"><li>• Repositorios del sistema operativo configurados.</li><li>• Permisos de root</li></ul>

El sistema Tablero de Control hace uso de Java v. 1.7. Se realizan los siguientes pasos para la instalación de la Java Runtime Machine & Java Development Kit en Red Hat Enterprise Linux 7

```
[root@lcvq-tab ~]# yum install java-1.7.0-openjdk.x86_64
```



```
root@lcvq-tab-01:~  
(2/3): pcsc-lite-libs-1.8.8-8.el7.x86_64.rpm | 34 kB 00:00  
(3/3): java-1.7.0-openjdk-headless-1.7.0.201-2.6.16.1.el7_6.x86_64 | 26 MB 00:05  
-----  
Total 4.6 MB/s | 26 MB 00:05  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
Installing : pcsc-lite-libs-1.8.8-8.el7.x86_64 1/3  
Installing : 1:java-1.7.0-openjdk-headless-1.7.0.201-2.6.16.1.el7_6.x86_64 2/3  
Installing : 1:java-1.7.0-openjdk-1.7.0.201-2.6.16.1.el7_6.x86_64 3/3  
Verifying : 1:java-1.7.0-openjdk-1.7.0.201-2.6.16.1.el7_6.x86_64 1/3  
Verifying : pcsc-lite-libs-1.8.8-8.el7.x86_64 2/3  
Verifying : 1:java-1.7.0-openjdk-headless-1.7.0.201-2.6.16.1.el7_6.x86_64 3/3  
  
Installed:  
java-1.7.0-openjdk.x86_64 1:1.7.0.201-2.6.16.1.el7_6  
  
Dependency Installed:  
java-1.7.0-openjdk-headless.x86_64 1:1.7.0.201-2.6.16.1.el7_6  
pcsc-lite-libs.x86_64 0:1.8.8-8.el7  
  
Complete!  
[root@lcvq-tab-01 ~]#
```

En caso exista una instalación previa de java, se usa la herramienta 'alternatives' para configurar la versión correcta de Java:

```
[root@lcvq-tab ~]# alternatives --config java
```

There are 2 programs which provide 'java'.

Selection	Command
-----	

```
* 1 java-1.8.0-openjdk.x86_64
(/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.141-2.b16.el7_4.x86_64/jre/bin/java)

* 2 java-1.7.0-openjdk.x86_64
(/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.151-2.6.11.1.el7_4.x86_64/jre/bin/java)
```

Seleccionamos '2' y verificamos la versión de Java:

```
[root@lcvq-tab ~]# java -version
java version "1.7.0_151"
```

```
OpenJDK Runtime Environment (rhel-2.6.11.1.el7_4-x86_64 u151-b00)
```

```
OpenJDK 64-Bit Server VM (build 24.151-b00, mixed mode)
```

## Instalación de JBoss EAP 6.4

### Instalación del servidor de aplicaciones

<b>Objetivo</b>	Descargar, configurar e iniciar el servidor de aplicaciones
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>• Acceso a Internet</li> <li>• Java JRE 7 previamente instalado</li> <li>• Permisos de root</li> <li>• Registrar una cuenta en la página:  <a href="http://developers.redhat.com/download-manager/file/jboss-eap-6.4.0.GA.zip">http://developers.redhat.com/download-manager/file/jboss-eap-6.4.0.GA.zip</a> </li> </ul>

#### Instalación Standalone


Se realizará la instalación del servidor de aplicaciones en modo standalone-full, que contiene todos los servicios de Java EE tales como: Web Services, REST, Servicio de colas con JMS, entre otros. La instalación en modo Domain (Server groups, master nodes y slave nodes) es recomendada en entornos donde se deba administrar más de un nodo (servidor virtual). En el caso de Tablero de Control, se utilizará un único servidor virtual. Los modos standalone y domain no tienen diferencias en performance o cantidad de servicios.

Descargar el empaquetado del servidor de aplicaciones desde la siguiente pagina de Red Hat:

<http://developers.redhat.com/download-manager/file/jboss-eap-6.4.0.GA.zip>

El enlace redigirá a la página de login de Red Hat, para lo cual es necesario una cuenta registrada.

Es seguro | [https://developers.redhat.com/auth/realms/rhd/protocol/openid-connect/auth?response\\_type=code&client\\_id=download-manager-client&redirect\\_uri=https%3A%2F...](https://developers.redhat.com/auth/realms/rhd/protocol/openid-connect/auth?response_type=code&client_id=download-manager-client&redirect_uri=https%3A%2F...)


REGISTER

### Log In

You can use your existing Red Hat Developer Program, Red Hat Customer Portal, OpenShift Online, or other Red Hat account to log in.

**LOG IN USING YOUR RED HAT ACCOUNT**

Email address or other Red Hat Login ID

Case Sensitive

Password

SHOW

Forgot Password?

**LOG IN**

**LOG IN USING SOCIAL ACCOUNT**

GITHUB

f FACEBOOK

STACKOVERFLOW

g- GOOGLE

LINKEDIN

MICROSOFT

TWITTER

JBOSS DEVELOPER

Al ingresar, la descarga comenzará automáticamente.

Mover el archivo descargado jboss-eap-6.4.0.zip dentro a la carpeta /opt y descomprimir:

```
[root@lcvq-tab ~]# cd /opt
```

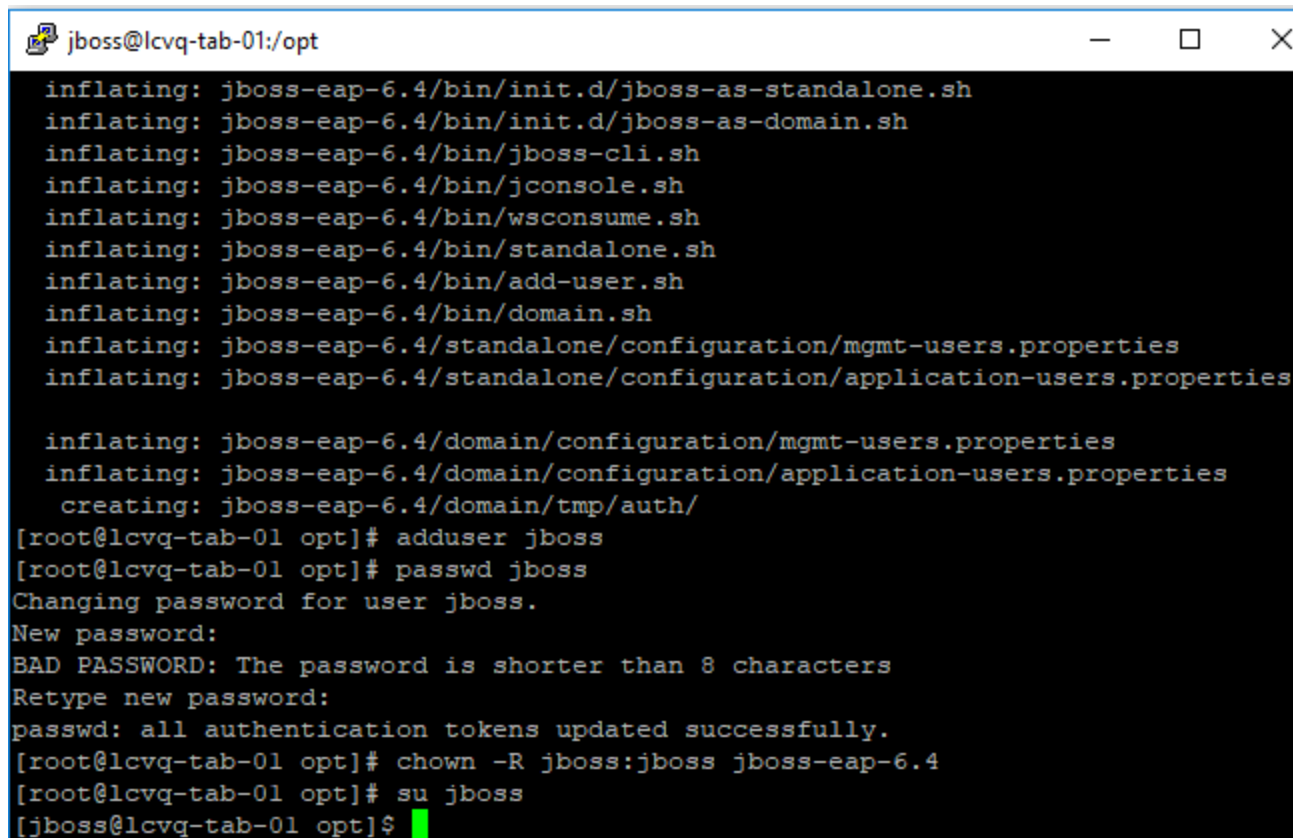
```
[root@lcvq-tab opt]# unzip jboss-eap-6.4.zip
```

Creamos el usuario 'jboss' que tendra control total del servidor de aplicaciones Jboss EAP 6.4 y asignamos permisos necesarios:

```
[root@lcvq-tab opt]# adduser jboss
[root@lcvq-tab opt]# passwd jboss
[root@lcvq-tab opt]# chown -R jboss:jboss jboss-eap-6.4
```

Iniciamos una sesión con el usuario recientemente creado:

```
[root@lcvq-tab opt]# su jboss
```

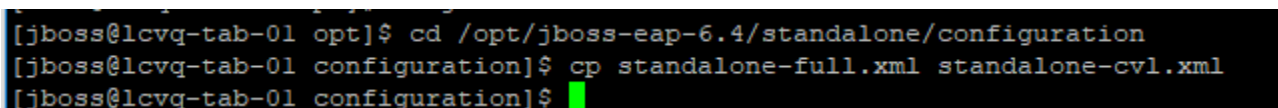


```
jboss@lcvq-tab-01:/opt
inflating: jboss-eap-6.4/bin/init.d/jboss-as-standalone.sh
inflating: jboss-eap-6.4/bin/init.d/jboss-as-domain.sh
inflating: jboss-eap-6.4/bin/jboss-cli.sh
inflating: jboss-eap-6.4/bin/jconsole.sh
inflating: jboss-eap-6.4/bin/wsconsume.sh
inflating: jboss-eap-6.4/bin/standalone.sh
inflating: jboss-eap-6.4/bin/add-user.sh
inflating: jboss-eap-6.4/bin/domain.sh
inflating: jboss-eap-6.4/standalone/configuration/mgmt-users.properties
inflating: jboss-eap-6.4/standalone/configuration/application-users.properties

inflating: jboss-eap-6.4/domain/configuration/mgmt-users.properties
inflating: jboss-eap-6.4/domain/configuration/application-users.properties
creating: jboss-eap-6.4/domain/tmp/auth/
[root@lcvq-tab-01 opt]# adduser jboss
[root@lcvq-tab-01 opt]# passwd jboss
Changing password for user jboss.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@lcvq-tab-01 opt]# chown -R jboss:jboss jboss-eap-6.4
[root@lcvq-tab-01 opt]# su jboss
[jboss@lcvq-tab-01 opt]$
```

Nos ubicamos en la carpeta de configuración del servidor de aplicaciones y realizamos una copia del archivo de configuración del servidor:

```
[jboss@lcvq-tab ~]# cd /opt/jboss-eap-6.4/standalone/configuration
[jboss@lcvq-tab configuration]# cp standalone-full.xml standalone-cv1.xml
```



```
[jboss@lcvq-tab-01 opt]$ cd /opt/jboss-eap-6.4/standalone/configuration
[jboss@lcvq-tab-01 configuration]$ cp standalone-full.xml standalone-cv1.xml
[jboss@lcvq-tab-01 configuration]$
```

## Configuración del servidor como servicio del sistema

Objetivo	Configurar JBoss como servicio del sistema, de tal manera que se inicie junto con el sistema operativo.
Requisitos	<ul style="list-style-type: none"><li>• Servidor JBoss previamente instalado</li><li>• Permisos de root</li></ul>

Iniciamos sesión con usuario 'root' y copiamos los archivos necesarios:

```
[root@lcvq-tab ~]# cd /opt/jboss-eap-6.4/bin
[root@lcvq-tab bin]# mkdir /etc/jboss-as
```

El archivo `jboss-as.conf` contiene las configuraciones por defecto que serán usadas en el archivo ejecutable que se instalará como servicio para administrar el inicio/parada/reinicio del servidor de aplicaciones JBoss. Debemos ubicar este archivo en la carpeta **`/etc/jboss-as`**. La lista de variables configurables se lista a continuación:

Variable de entorno	Descripción
JBOSS_USER=jboss	Usuario del sistema que cuenta con permisos para iniciar/detener/reiniciar el servidor de aplicaciones JBoss
STARTUP_WAIT=30	Tiempo de espera del sistema operativo para iniciar el servidor de aplicaciones JBoss. Después de este tiempo se detendrá el inicio del servicio
SHUTDOWN_WAIT=30	Tiempo de espera del sistema operativo para detener el servidor de aplicaciones JBoss. Después de este tiempo se abortará la detención del servicio
JBOSS_CONSOLE_LOG=/var/log/jboss-as/console.log	Ruta de logs del servidor de aplicaciones solamente.
JBOSS_CONFIG=standalone-cvl.xml	Ruta del archivo de configuración usado por el servidor de aplicaciones JBoss
JBOSS_HOME=/opt/jboss-eap-6.4	Ruta de instalación del servidor de aplicaciones JBoss
BIND_ADDR=172.16.7.122	Dirección de red a la cual se ligará el servidor de aplicaciones (puertos 8080 y 8443)
BIND_ADDR_MGMT=0.0.0.0	Dirección de red a la cual se ligará la consola de administración de JBoss (puerto 9990)

```

# General configuration for the init.d scripts,
# not necessarily for JBoss AS itself.

# The username who should own the process.
#
JBOSS_USER=jboss

# The amount of time to wait for startup
#
STARTUP_WAIT=30

# The amount of time to wait for shutdown
#
SHUTDOWN_WAIT=30

# Location to keep the console log
#
JBOSS_CONSOLE_LOG=/var/log/jboss-as/console.log

JBOSS_CONFIG=standalone-cv1.xml

JBOSS_HOME=/opt/jboss-eap-6.4

BIND_ADDR=172.16.7.122

BIND_ADDR_MGMT=0.0.0.0

```

El archivo `jboss-as-standalone` contiene el script ejecutable necesario para el inicio/detención/reinicio del servidor de aplicaciones JBoss. Hace uso de las variables de entorno configuradas en el archivo `/etc/jboss-as/jboss-as.conf` por lo que están altamente ligados. Debemos ubicar este archivo en la carpeta: **`/etc/init.d`**



jboss-as.conf



jboss-as-standalone.dat

/etc/rc.d/init.d/				
Nombre	Tamaño	Modificado	Permisos	Propietario
..		11/01/2019 16:49:56	rwxt-xt-x	root
jboss-as.conf	1 KB	19/02/2019 16:01:32	rw-r--r--	root
jboss-as-standalone.dat	4 KB	19/02/2019 16:01:29	rw-r--r--	root
README	2 KB	7/09/2018 03:00:47	rw-r--r--	root
network	8 KB	24/08/2018 05:23:27	rwxt-xt-x	root
netconsole	5 KB	24/08/2018 05:23:27	rwxt-xt-x	root
functions	18 KB	24/08/2018 05:23:27	rw-r--r--	root
rhnsd	3 KB	19/10/2017 02:51:42	rwxt-xt-x	root

Modificamos los permisos de ejecución del archivo y configuramos el servicio:

```
[root@lcvq-tab ~]# chmod +x /etc/rc.d/init.d/jboss-as-standalone.dat
[root@lcvq-tab ~]# chkconfig --add /etc/init.d/jboss-as-standalone.dat
[root@lcvq-tab ~]# service jboss-as-standalone.dat start

Starting jboss-as-standalone (via systemctl):          [ OK ]
```

Podemos verificar el estado del servidor de aplicaciones con:

```
[root@lcvq-tab jboss-as]# service jboss-as-standalone.dat status

jboss-as is running (pid 4693)
```

Verificamos que el comando 'stop' funcione propiamente:

```
[root@lcvq-tab jboss-as]# service jboss-as-standalone.dat stop
Stopping jboss-as-standalone (via systemctl): [ OK ]
```

Iniciamos nuevamente el servicio para continuar con la instalación:

```
[root@lcvq-tab jboss-as]# service jboss-as-standalone.dat start
Starting jboss-as-standalone (via systemctl): [ OK ]
```

La siguiente tabla muestra los comandos disponibles del servicio:

Comando	Descripción
service jboss-as-standalone start	Inicia el servicio. Si ya existe un servicio iniciado, no efectúa acción alguna.
service jboss-as-standalone stop	Detiene el servicio.
service jboss-as-standalone status	Muestra si el estado del servicio
service jboss-as-standalone restart	Ejecuta las acciones de detención e inicio.
service jboss-as-standalone reload	Detiene e inicia las aplicaciones desplegadas, sin reiniciar el servidor.

## Habilitación de la consola de administración

Objetivo	Crear un usuario de administración que nos permita ingresar a la consola de administración de JBoss
Requisitos	<ul style="list-style-type: none"><li>• Servidor JBoss previamente instalado e iniciado</li></ul>

Con el servidor de aplicaciones iniciado. Iniciamos sesión con el usuario 'jboss' y procedemos a agregar un usuario de administración para el servidor de aplicaciones:

```
[root@lcvq-tab ~]# su jboss
[jboss@lcvq-tab ~]# cd /opt/jboss-eap-6.4/bin
[jboss@lcvq-tab bin]$ sh add-user.sh --group admin --password D4shb0ard_clv --user dashboard
```

```
1. Home
2. 192.168.151.121 (root) - QA

> SSH session to root@192.168.151.121
• SSH compression : ✓
• SSH-browser      : ✓
• X11-forwarding   : ✓ (remote display is forwarded through SSH)
• DISPLAY          : ✓ (automatically set on remote server)

> For more info, ctrl+click on help or visit our website

Last login: Thu Sep 27 09:56:24 2018 from 172.16.17.84
[root@localhost ~]# service jboss-as-standalone status
Redirecting to /bin/systemctl status jboss-as-standalone.service
Unit jboss-as-standalone.service could not be found.
[root@localhost ~]# service jboss-as-standalone.dat status
boss-as is not running
[root@localhost ~]# chmod +x /etc/rc.d/init.d/jboss-as-standalone.dat
[root@localhost ~]# chkconfig --add /etc/init.d/jboss-as-standalone.dat
[root@localhost ~]# service jboss-as-standalone.dat start
Starting jboss-as-standalone.dat (via systemctl): [ OK ]
[root@localhost ~]# service jboss-as-standalone.dat status
boss-as is running (pid 7043)
[root@localhost ~]# service jboss-as-standalone.dat stop
Stopping jboss-as-standalone.dat (via systemctl): [ OK ]
[root@localhost ~]# service jboss-as-standalone.dat start
Starting jboss-as-standalone.dat (via systemctl): [ OK ]
[root@localhost ~]# su jboss
jboss@localhost root]# cd /opt/jboss-eap-6.4/bin
jboss@localhost bin]$ sh add-user.sh --group admin --password D4shb0ard_clv --u
ser dashboard
Added user 'dashboard' to file '/opt/jboss-eap-6.4/standalone/configuration/mgmt
-users.properties'
Added user 'dashboard' to file '/opt/jboss-eap-6.4/domain/configuration/mgmt-use
rs.properties'
Added user 'dashboard' with groups admin to file '/opt/jboss-eap-6.4/standalone/
configuration/mgmt-groups.properties'
Added user 'dashboard' with groups admin to file '/opt/jboss-eap-6.4/domain/conf
iguration/mgmt-groups.properties'
jboss@localhost bin]$
```

QA

```
jboss@lcvq-tab-01:/opt/jboss-eap-6.4/bin

[jboss@lcvq-tab-01 bin]$ cd /opt/jboss-eap-6.4/bin
[jboss@lcvq-tab-01 bin]$ sh add-user.sh --group admin --password D4shb0ard_clv --user dashboard
Added user 'dashboard' to file '/opt/jboss-eap-6.4/standalone/configuration/mgmt-users.properties'
Added user 'dashboard' to file '/opt/jboss-eap-6.4/domain/configuration/mgmt-users.properties'
Added user 'dashboard' with groups admin to file '/opt/jboss-eap-6.4/standalone/configuration/mgmt-groups.
properties'
Added user 'dashboard' with groups admin to file '/opt/jboss-eap-6.4/domain/configuration/mgmt-groups.prop
erties'
[jboss@lcvq-tab-01 bin]$
```



```
[jboss@lcvq-tab-01 bin]$ cd /opt/jboss-eap-6.4/bin
[jboss@lcvq-tab-01 bin]$ sh add-user.sh --group admin --password D4shb0ard_clv --user dashboard
Added user 'dashboard' to file '/opt/jboss-eap-6.4/standalone/configuration/mgmt-users.properties'
Added user 'dashboard' to file '/opt/jboss-eap-6.4/domain/configuration/mgmt-users.properties'
Added user 'dashboard' with groups admin to file '/opt/jboss-eap-6.4/standalone/configuration/mgmt-groups.properties'
Added user 'dashboard' with groups admin to file '/opt/jboss-eap-6.4/domain/configuration/mgmt-groups.properties'
[jboss@lcvq-tab-01 bin]$
```

Para eliminar un usuario, debemos removerlo del archivo mgmt-users.properties ubicado en la carpeta /opt/jboss-eap-6.4/standalone/configuration

```
[root@lcvq-tab configuration]# vim mgmt-users.properties
```

```
dashboard=8dc4d6a0b479264e9206ff127e3914d2
```

Para modificar un usuario, debemos efectuar una eliminación y volver a agregar el usuario de la forma previamente vista. Para mayor información sobre administración de usuarios, consultar: [https://access.redhat.com/documentation/en-US/JBoss\\_Enterprise\\_Application\\_Platform/6.4/html/Administration\\_and\\_Configuration\\_Guide/chap-User\\_Management.html](https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.4/html/Administration_and_Configuration_Guide/chap-User_Management.html)

## Configuración del Firewall

<b>Objetivo</b>	Habilitar puertos 8080 (HTTP), 9990(HTTP) y 8443(HTTPS) que permiten acceso a servidor JBoss
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>• Permisos de root</li> </ul>

Jboss utiliza el puerto 8080(HTTP) y 8443(HTTPS) para exponer aplicaciones desplegadas y 9990 para la consola de administración.

Si usamos firewallD para la gestión de firewall en Red Hat 7. Habilitamos los puertos de la siguiente manera:

```
[root@lcvq-tab ~]# firewall-cmd --zone=public --add-port=8080/tcp
```

```
[root@lcvq-tab ~]# firewall-cmd --zone=public --add-port=9990/tcp
```

```
[root@lcvq-tab ~]# firewall-cmd --zone=public --add-port=8443/tcp
```

```
[root@localhost ~]# firewall-cmd --zone=public --add-port=8080/tcp
success
[root@localhost ~]# firewall-cmd --zone=public --add-port=9990/tcp
success
[root@localhost ~]# firewall-cmd --zone=public --add-port=8443/tcp
success
[root@localhost ~]# █
```

Si usamos iptables para el control de firewall, lo habilitamos con los siguientes comandos:

```
[root@lcvq-tab ~]# iptables -A IN_public_allow -p tcp -m tcp --dport 8080 -j ACCEPT
```

```
[root@lcvq-tab ~]# iptables -A OUTPUT -p tcp -m tcp --dport 8080 -j ACCEPT
```

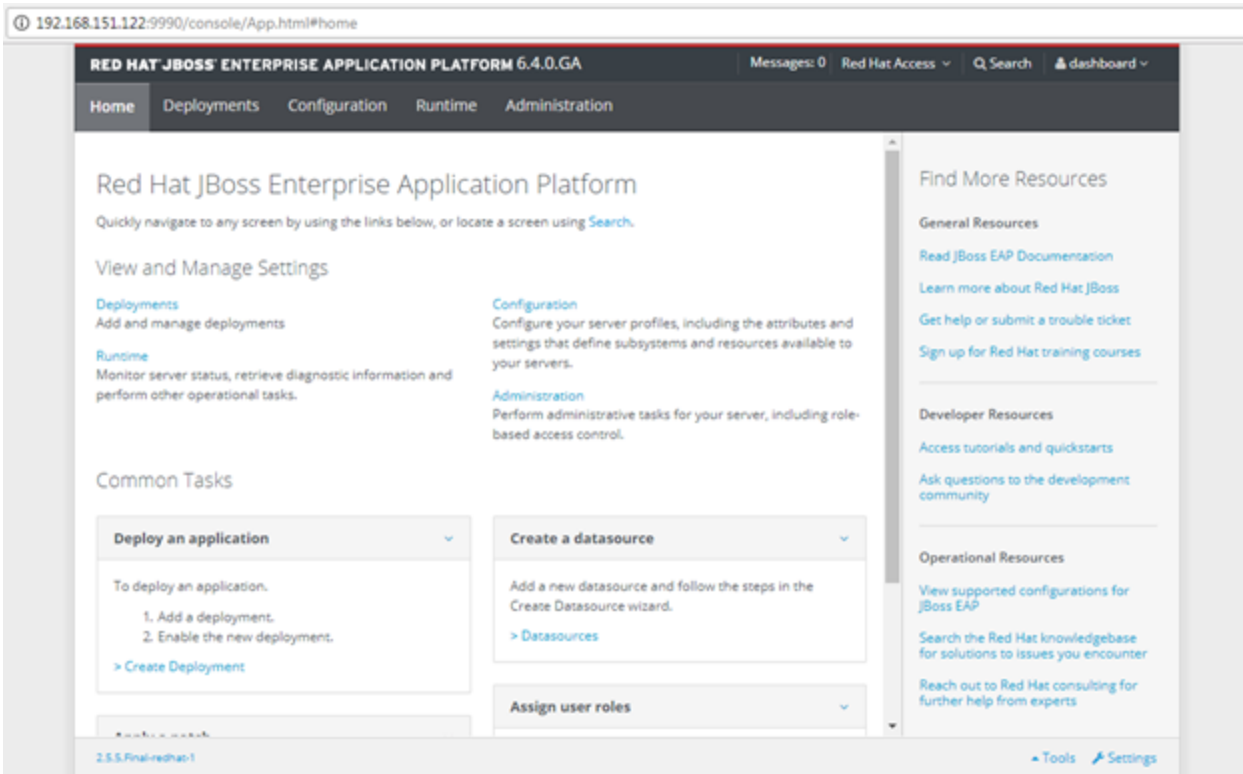
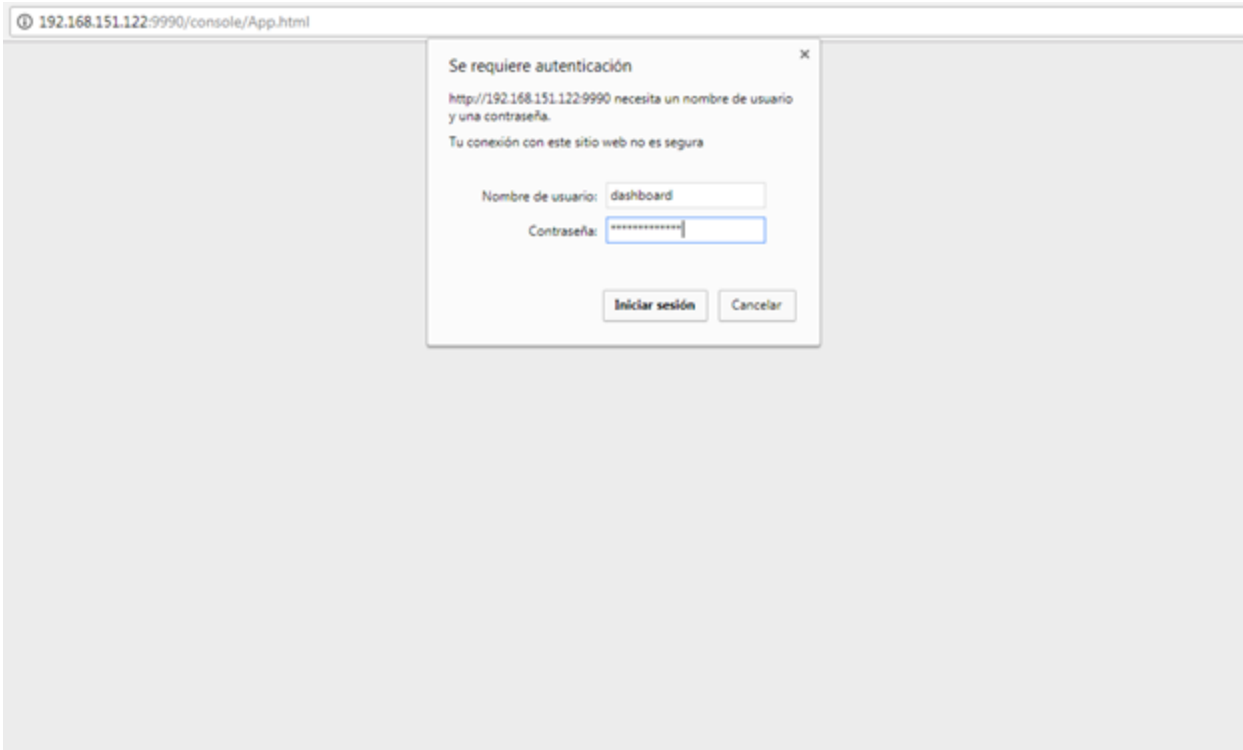
```
[root@lcvq-tab ~]# iptables -A IN_public_allow -p tcp -m tcp --dport 9990 -j ACCEPT
```

```
[root@lcvq-tab ~]# iptables -A OUTPUT -p tcp -m tcp --dport 9990 -j ACCEPT
```

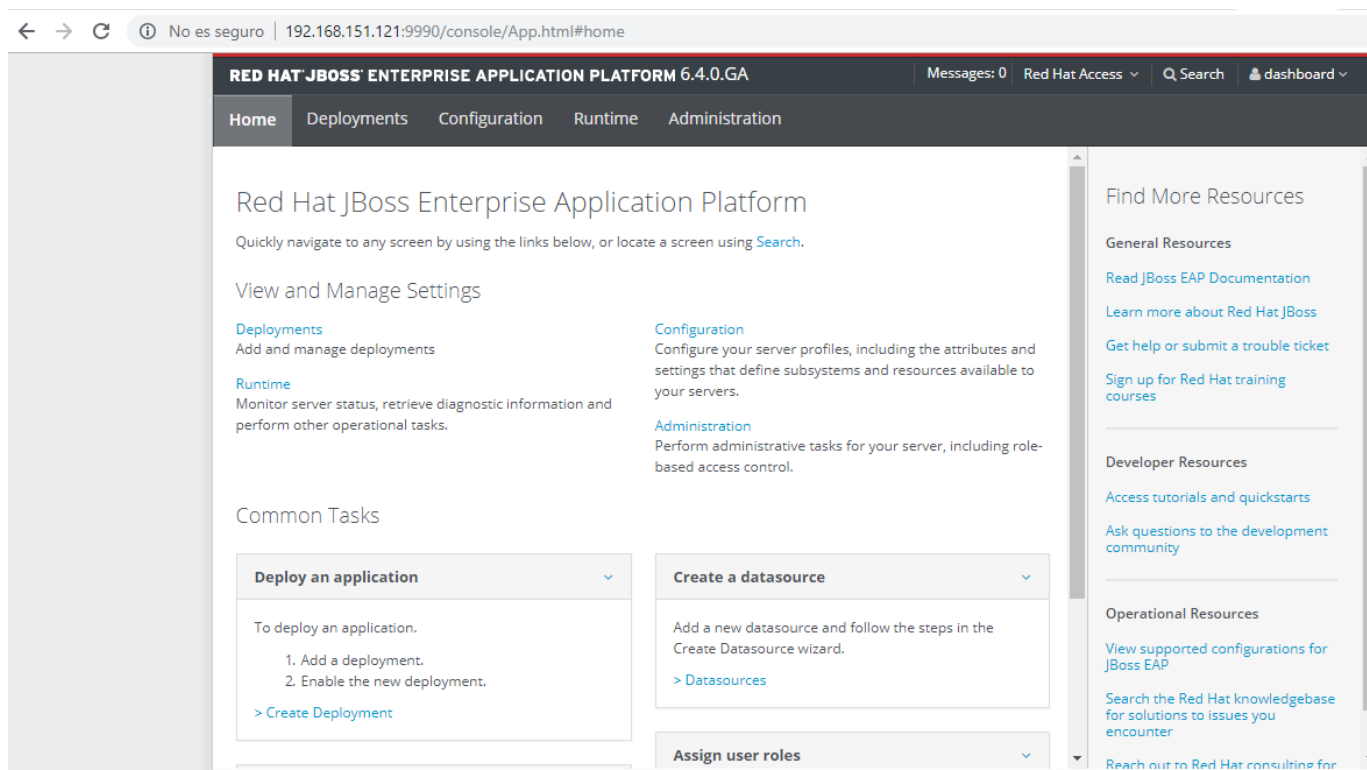
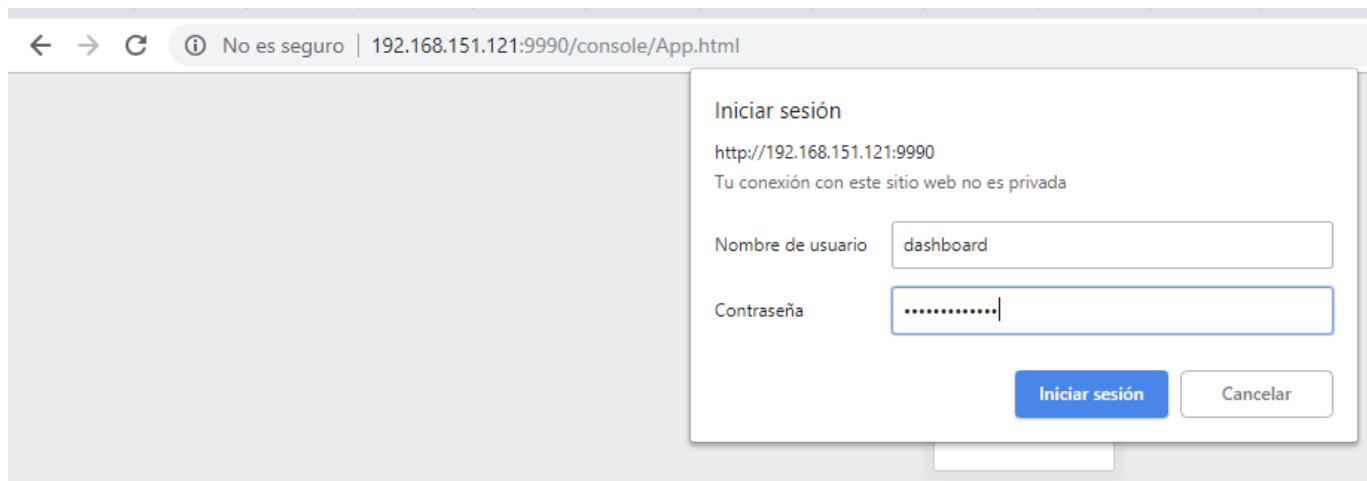
```
[root@lcvq-tab ~]# iptables -A IN_public_allow -p tcp -m tcp --dport 8443 -j ACCEPT
```

```
[root@lcvq-tab ~]# iptables -A OUTPUT -p tcp -m tcp --dport 8443 -j ACCEPT
```

Iniciamos el explorador y verificamos que la consola de administración se encuentra operativa, ingresando a <http://<servidor>:9990/>



QA:



## Módulos de la consola de administración JBoss

La consola tiene los siguientes componentes:

Pestañas	Descripción
Home	Muestra las acciones que se pueden realizar en la consola y accesos directos
Deployments	Muestra las aplicaciones desplegadas, aquí se pueden remover, iniciar y reiniciar las aplicaciones. Asi como el detalle de los componentes de la aplicación
Configuration	Muestra los diferentes servicios del servidor de aplicaciones
Runtime	Muestra detalles de monitoreo de la situación actual del servidor de aplicaciones, como por ejemplo memoria usada por la JVM, conexiones de datasource, transacciones, etc.
Administration	Muestra detalles de usuarios y grupos usados por la consola de administración

## Configuración de Driver Postgres

<b>Objetivo</b>	Instalar driver Postgres como módulo del servidor de aplicaciones JBoss
<b>Requisitos</b>	<ul style="list-style-type: none"><li>• Servidor JBoss previamente instalado e iniciado</li><li>• Acceso a Internet</li></ul>

Para que el sistema pueda conectarse a Postgres es necesaria la instalación del driver. Descargamos el driver y configuración adjunta. Ubicamos el archivo org.zip en la siguiente ruta: /opt/jboss-eap-6.4/modules



procedemos a descomprimirlo

```
[root@lcvq-tab ~]# unzip org.zip
```

```
[root@lcvq-tab ~]# chown -R jboss:jboss org
```

Verificamos que el driver se haya instalado correctamente verificando que existan los siguientes archivos:

```
[root@lcvq-tab ~]# su jboss
```

```
[jboss@lcvq-tab root]$
```

```
[jboss@lcvq-tab root]$ cd /opt/jboss-eap-6.4/modules/org/postgresql/main
```

```
[jboss@lcvq-tab main]$ ls -l
```

```
total 704
```

```
-rw-rw-r--. 1 jboss jboss    303 Aug 14 16:12 module.xml
```

```
-rw-rw-r--. 1 jboss jboss 713037 Aug 14 16:12 postgresql-42.1.4.jre7.jar
```

```
jboss@lcvq-tab-01:/opt/jboss-eap-6.4/modules/org/postgresql/main
[root@lcvq-tab-01 bin]# cd /opt/jboss-eap-6.4/modules
[root@lcvq-tab-01 modules]# unzip org.zip
Archive:  org.zip
  creating: org/
  creating: org/backup/
  inflating: org/backup/module.xml
  inflating: org/backup/postgresql-42.1.4.jre7.jar
  creating: org/postgresql/
  creating: org/postgresql/main/
  inflating: org/postgresql/main/module.xml
  inflating: org/postgresql/main/postgresql-42.1.4.jar
  inflating: org/postgresql/main/postgresql-42.2.5.jre7.jar
[root@lcvq-tab-01 modules]# chown -R jboss:jboss org
[root@lcvq-tab-01 modules]# su jboss
[jboss@lcvq-tab-01 modules]$ cd /opt/jboss-eap-6.4/modules/org/postgresql/main
[jboss@lcvq-tab-01 main]$ ls -l
total 1440
-rw-r--r--. 1 jboss jboss    324 Nov  5 15:00 module.xml
-rw-r--r--. 1 jboss jboss 713037 Aug 14 2017 postgresql-42.1.4.jar
-rw-r--r--. 1 jboss jboss 753264 Nov  5 14:47 postgresql-42.2.5.jre7.jar
[jboss@lcvq-tab-01 main]$
```

## Configuración del Datasource

Objetivo	Configurar un datasource de tipo XA usando un driver postgres apuntando a la base de datos de Tablero de Control
Requisitos	<ul style="list-style-type: none"><li>• Servidor JBoss previamente instalado e iniciado</li></ul>

Una vez instalado el driver, procedemos con la configuración del datasource. Nuevamente accedemos al Command Line Interface del servidor de aplicaciones: Jboss CLI:

```
[root@lcvq-tab ~]# su jboss
[jboss@lcvq-tab ~]# cd /opt/jboss-eap-6.4/bin
[jboss@lcvq-tab bin]# sh jboss-cli.sh --connect
[standalone@localhost:9999 /]
[standalone@localhost:9999 /]
/subsystem=datasources/jdbc-driver=postgresql:add(driver-name=postgresql,driver-module-name=org.postgresql,
driver-xa-datasource-class-name=org.postgresql.xa.PGXADatasource)
{"outcome" => "success"}
[standalone@localhost:9999 /]
[jboss@lcvq-tab-01 bin]$ sh jboss-cli.sh --connect
[standalone@localhost:9999 /] /subsystem=datasources/jdbc-driver=postgresql:add(driver-name=postgresql,dri
ver-module-name=org.postgresql,driver-xa-datasource-class-name=org.postgresql.xa.PGXADatasource)
{"outcome" => "success"}
[standalone@localhost:9999 /]
```

```
[standalone@localhost:9999 /] xa-data-source add --name=PostgresXADS --driver-name=postgresql
--jndi-name=java:jboss/PostgresXADS --user-name=control_panel --password=D4shb0ard_clv --max-pool-size=25
--set-tx-query-timeout=true --enabled=true --blocking-timeout-wait-millis=5000 --pool-use-strict-min=false
--transaction-isolation=TRANSACTION_READ_COMMITTED
--valid-connection-checker-class-name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker --no-recovery=false --background-validation-millis=30000 --use-fast-fail=true
--exception-sorter-class-name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter
--background-validation=false

[standalone@localhost:9999 /]
/subsystem=datasources/xa-data-source=PostgresXADS/xa-datasource-properties=ServerName:add(value=127.0.0.1)

{"outcome" => "success"}

[standalone@localhost:9999 /]
/subsystem=datasources/xa-data-source=PostgresXADS/xa-datasource-properties=PortNumber:add(value=5432)

{"outcome" => "success"}

[standalone@localhost:9999 /]
/subsystem=datasources/xa-data-source=PostgresXADS/xa-datasource-properties=DatabaseName:add(value=control_panel)

{"outcome" => "success"}

[standalone@localhost:9999 /] xa-data-source enable --name=PostgresXADS
```

```
jboss@lcvq-tab-01:/opt/jboss-eap-6.4/bin
--isolation=TRANSACTION_READ_COMMITTED --valid-connection-checker-class-name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker --no-recovery=false --background-validation-millis=30000 --use-fast-fail=true --exception-sorter-class-name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter --background-validation=false
{"JBAS014653: Composite operation failed and was rolled back. Steps that failed:" => {"Operation step-1" => "JBAS014688: Wrong type for password. Expected [EXPRESSION, STRING] but was OBJECT"}}
[standalone@localhost:9999 /] xa-data-source add --name=PostgresXADS --driver-name=postgresql --jndi-name=java:jboss/PostgresXADS --user-name=control_panel --password=D4shb0ard_clv --max-pool-size=25 --set-tx-query-timeout=true --enabled=true --blocking-timeout-wait-millis=5000 --pool-use-strict-min=false --transaction-isolation=TRANSACTION_READ_COMMITTED --valid-connection-checker-class-name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker --no-recovery=false --background-validation-millis=30000 --use-fast-fail=true --exception-sorter-class-name=org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter --background-validation=false
[standalone@localhost:9999 /] /subsystem=datasources/xa-data-source=PostgresXADS/xa-datasource-properties=ServerName:add(value=127.0.0.1)
{"outcome" => "success"}
[standalone@localhost:9999 /] /subsystem=datasources/xa-data-source=PostgresXADS/xa-datasource-properties=PortNumber:add(value=5432)
{"outcome" => "success"}
[standalone@localhost:9999 /] /subsystem=datasources/xa-data-source=PostgresXADS/xa-datasource-properties=DatabaseName:add(value=control_panel)
{"outcome" => "success"}
[standalone@localhost:9999 /] xa-data-source enable --name=PostgresXADS
[standalone@localhost:9999 /] █
```

## Habilitar protocolo para websockets

<b>Objetivo</b>	Configurar el conector correcto para soportar transacciones a través de web sockets en el servidor de aplicaciones JBoss
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>Servidor JBoss previamente instalado e iniciado</li> </ul>

Ingresamos a la carpeta /opt/jboss-eap-6.4/bin y nos conectamos a la línea de comandos de JBoss. Agregamos la sentencia especificada write-attribute y verificamos que se realizó el cambio con read-attribute:

```
[root@lcvq-tab bin]# ./jboss-cli.sh --connect
```

```
jboss@lcvq-tab-01:/opt/jboss-eap-6.4/bin
[standalone@localhost:9999 /] exit

[jboss@lcvq-tab-01 bin]$ ./jboss-cli.sh --connect
[standalone@localhost:9999 /] █
```

```
[standalone@localhost:9999 /]
/subsystem=web/connector=http/:write-attribute(name=protocol,value=org.apache.coyote.http11.Http11NioProtocol)
```

```
{ "outcome" => "success" }
```

```
[root@localhost ~]# cd /opt/jboss-eap-6.4/bin
[root@localhost bin]# ./jboss-cli.sh --connect
[standalone@localhost:9999 /] /subsystem=web/connector=http/:write-attribute(name=protocol,value=org.apache.coyote.http11.Http11NioProtocol)
{
  "outcome" => "success",
  "response-headers" => {
    "operation-requires-reload" => true,
    "process-state" => "reload-required"
  }
}
[standalone@localhost:9999 /] █
```

```
[standalone@localhost:9999 /] /subsystem=web/connector=http/:read-attribute(name=protocol)
```

```
{
  "outcome" => "success",
  "result" => "org.apache.coyote.http11.Http11NioProtocol"
}
```

```
[standalone@localhost:9999 /]
```

```
[standalone@localhost:9999 /] \q
```

```
[standalone@localhost:9999 /] /subsystem=web/connector=http/:read-attribute(name=protocol)
{
  "outcome" => "success",
  "result" => "org.apache.coyote.http11.Http11NioProtocol",
  "response-headers" => {"process-state" => "reload-required"}
}
[standalone@localhost:9999 /] \q

[jboss@lcvq-tab-01 bin]$ █
```

```
[root@lcvq-tab bin]#
```

Tipeamos \q para salir de la línea de comandos de JBoss.

## Configurar Interfaz del jboss

Editamos el standalone-cv1.xml con el usuario **root**

```
[root@lcvq-tab ~]# cd /opt/jboss-eap-6.4/standalone/configuration
```

```
[root@lcvq-tab ~]# vim standalone-cv1.xml
```

```

<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address:127.0.0.1}"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="{jboss.bind.address.unsecure:127.0.0.1}"/>
  </interface>
</interfaces>

```

editamos las interfases y en vez de 127.0.0.1 ponemos la ip del servidor (172.16.7.122)

```

root@lcvq-tab-01:/opt/jboss-eap-6.4/standalone/configuration
<subsystem xmlns="urn:jboss:domain:weld:1.0"/>
</profile>

<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:172.16.7.122}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address:172.16.7.122}"/>
  </interface>
  <interface name="unsecure">
    <inet-address value="{jboss.bind.address.unsecure:172.16.7.122}"/>
  </interface>
</interfaces>

<socket-binding-group name="standard-sockets" default-interface="public" port-offset="{jboss.socket.b
inding.port-offset:0}">
  <socket-binding name="management-native" interface="management" port="{jboss.management.native.por
t:9999}"/>
  <socket-binding name="management-http" interface="management" port="{jboss.management.http.port:9
990}"/>
  <socket-binding name="management-https" interface="management" port="{jboss.management.https.port
:9443}"/>
"standalone-cv1.xml" 485L, 24237C                                     462,13      96%

```

Salimos y Grabamos

:wq!

Procedemos a reiniciar el jboss

## Instalación de Postgres 9.6

<b>Objetivo</b>	Instalar el servidor de base de datos Postgres 9.6 en el sistema operativo
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>• Acceso a Internet</li> <li>• Permisos de root</li> </ul>

Iniciamos sesión con root y descargamos el repositorio de paquetes de Postgres para Red Hat Enterprise Linux 7:

```

[root@lcvq-tab ~]# yum install
https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm

```



root@lcvq-tab-01:/opt/jboss-eap-6.4/standalone/configuration

```
=====
Installing:
pgdg-redhat96          noarch          9.6-3          /pgdg-redhat96-9.6-3.noarch          2.7 k
=====
Transaction Summary
=====
Install 1 Package

Total size: 2.7 k
Installed size: 2.7 k
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : pgdg-redhat96-9.6-3.noarch          1/1
  Verifying  : pgdg-redhat96-9.6-3.noarch          1/1

Installed:
pgdg-redhat96.noarch 0:9.6-3

Complete!
[root@lcvq-tab-01 configuration]#
```

[root@lcvq-tab ~]# yum install postgresql96

root@lcvq-tab-01:/opt/jboss-eap-6.4/standalone/configuration

```
Installed size: 8.7 M
Is this ok [y/d/N]: y
Downloading packages:
(1/2): postgresql96-9.6.12-1PGDG.rhel7.x86_64.rpm | 1.4 MB 00:00:01
(2/2): postgresql96-libs-9.6.12-1PGDG.rhel7.x86_64.rpm | 319 kB 00:00:01
-----
Total                                          910 kB/s | 1.7 MB 00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql96-libs-9.6.12-1PGDG.rhel7.x86_64          1/2
  Installing : postgresql96-9.6.12-1PGDG.rhel7.x86_64              2/2
  Verifying  : postgresql96-9.6.12-1PGDG.rhel7.x86_64              1/2
  Verifying  : postgresql96-libs-9.6.12-1PGDG.rhel7.x86_64          2/2

Installed:
postgresql96.x86_64 0:9.6.12-1PGDG.rhel7

Dependency Installed:
postgresql96-libs.x86_64 0:9.6.12-1PGDG.rhel7

Complete!
[root@lcvq-tab-01 configuration]#
```

[root@lcvq-tab ~]# yum install postgresql96-server

```
root@lcvq-tab-01:/opt/jboss-eap-6.4/standalone/configuration

Installing:
postgresql96-server          x86_64          9.6.12-1PGDG.rhel7          pgdg96          4.5 M

Transaction Summary
=====
Install 1 Package

Total download size: 4.5 M
Installed size: 19 M
Is this ok [y/d/N]: y
Downloading packages:
postgresql96-server-9.6.12-1PGDG.rhel7.x86_64.rpm          | 4.5 MB  00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql96-server-9.6.12-1PGDG.rhel7.x86_64          1/1
  Verifying  : postgresql96-server-9.6.12-1PGDG.rhel7.x86_64          1/1

Installed:
  postgresql96-server.x86_64 0:9.6.12-1PGDG.rhel7

Complete!
[root@lcvq-tab-01 configuration]#
```

El sistema creará las carpetas necesarias de la instalación y además instalará un usuario de tipo bash llamado 'postgres'.

Se crearán los siguientes directorios:

Directorio de instalación	/usr/pgsql-9.6
Directorio de clúster de base de datos por defecto	/var/lib/pgsql/9.6/data

Agregamos la ruta de los archivos binarios de postgres al PATH del sistema:

```
[root@lcvq-tab ~]# vim /root/.bash_profile
```

Agregamos la siguiente línea:

```
PATH=$PATH:/usr/pgsql-9.6/bin
```

```
root@lcvq-tab-01:~  
# .bash_profile  
  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
    . ~/.bashrc  
fi  
  
# User specific environment and startup programs  
  
PATH=$PATH:$HOME/bin  
PATH=$PATH:/usr/pgsql-9.6/bin  
export PATH  
  
-- INSERT --
```

## Configuración de BD de Tablero de control

## Creación de Tablas y Roles Iniciales

<b>Objetivo</b>	Inicializar el esquema, tablespace y base de datos lógica en Postgres 9.6 de la aplicación Tablero de Control
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>• Permisos de root</li> </ul>

Se crearán 2 directorios para almacenar información de la base de datos del Tablero de Control:

<b>Clúster de Base de datos</b>	/opt/dbdata/
<b>Tablespace</b>	/opt/dbts/

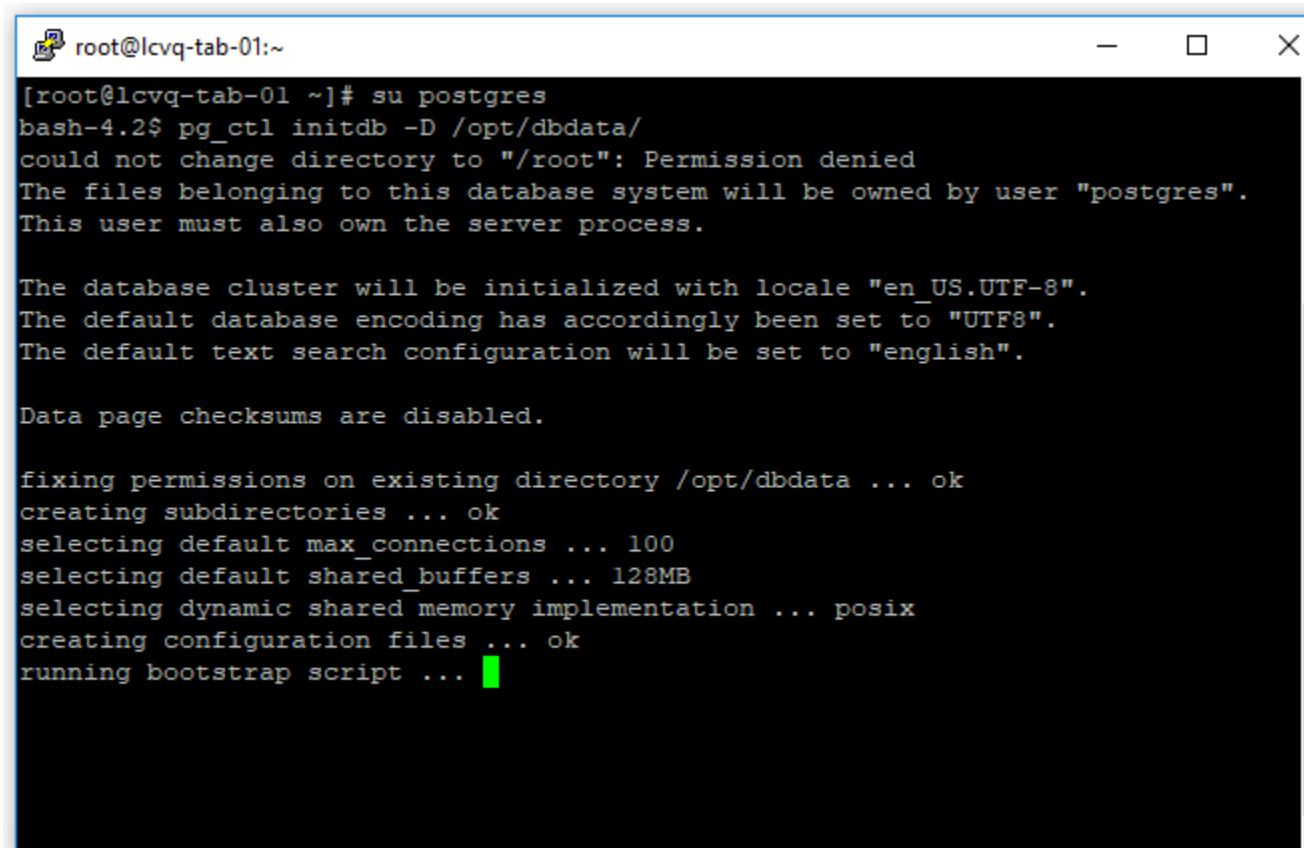
```
[root@lcvg-tab ~]# cd /opt
[root@lcvg-tab ~]# mkdir dbdata
[root@lcvg-tab ~]# mkdir dbts
[root@lcvg-tab ~]# chown postgres:postgres /opt/dbdata
[root@lcvg-tab ~]# chown postgres:postgres /opt/dbts
```

```
[root@localhost ~]# cd /opt
[root@localhost opt]# mkdir dbdata
[root@localhost opt]# mkdir dbts
[root@localhost opt]# chown postgres:postgres /opt/dbdata
[root@localhost opt]# chown postgres:postgres /opt/dbts
[root@localhost opt]#
```

Reiniciar el servidor: sudo reboot

```
[root@lcvq-tab ~]# su postgres
```

```
bash-4.2$ pg_ctl initdb -D /opt/dbdata/
```

A terminal window titled 'root@lcvq-tab-01:~' with standard window controls. The terminal shows the output of the 'pg\_ctl initdb' command. It starts with a prompt to switch to the 'postgres' user, followed by the command 'pg\_ctl initdb -D /opt/dbdata/'. The output includes a 'Permission denied' message for changing to '/root', followed by initialization details: locale 'en\_US.UTF-8', encoding 'UTF8', and search configuration 'english'. It also states that data page checksums are disabled. A series of configuration steps are listed: fixing permissions, creating subdirectories, selecting default max\_connections (100), shared\_buffers (128MB), dynamic shared memory implementation (posix), and creating configuration files. The process ends with 'running bootstrap script' followed by a green cursor.

```
root@lcvq-tab-01:~  
[root@lcvq-tab-01 ~]# su postgres  
bash-4.2$ pg_ctl initdb -D /opt/dbdata/  
could not change directory to "/root": Permission denied  
The files belonging to this database system will be owned by user "postgres".  
This user must also own the server process.  
  
The database cluster will be initialized with locale "en_US.UTF-8".  
The default database encoding has accordingly been set to "UTF8".  
The default text search configuration will be set to "english".  
  
Data page checksums are disabled.  
  
fixing permissions on existing directory /opt/dbdata ... ok  
creating subdirectories ... ok  
selecting default max_connections ... 100  
selecting default shared_buffers ... 128MB  
selecting dynamic shared memory implementation ... posix  
creating configuration files ... ok  
running bootstrap script ... █
```

```
[root@lcvq-tab ~]# cd /opt/dbdata
```

```
[root@lcvq-tab ~]# vim postgresql.conf
```

**Solo edito los siguientes parámetros, en caso alguno este comentado, descomentarlo.**

```
max_connections = 500  
shared_buffers = 512MB
```

```
work_mem = 64MB  
maintenance_work_mem = 512MB  
effective_cache_size = 8GB
```

**Una vez modificado los campos, grabar y salir**

**:wq!**

```
bash-4.2$ pg_ctl start -D /opt/dbdata/
```

```
server starting
```

```
[root@localhost ~]# cd /opt
[root@localhost opt]# su postgres
bash-4.2$ pg_ctl initdb -D /opt/dbdata/
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.
```

```
The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".
```

```
Data page checksums are disabled.
```

```
fixing permissions on existing directory /opt/dbdata ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok
```

```
WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.
```

```
Success. You can now start the database server using:
```

```
    /usr/pgsql-9.6/bin/pg_ctl -D /opt/dbdata -l logfile start
```

```
bash-4.2$ pg_ctl start -D /opt/dbdata/
```

```
server starting
```

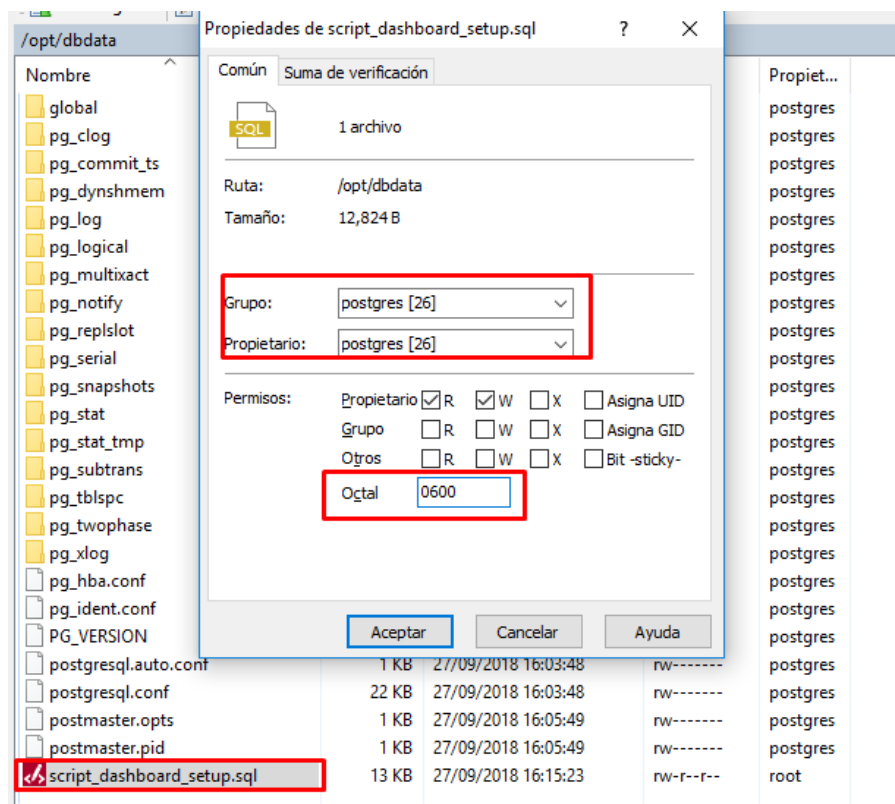
```
bash-4.2$ < 2018-09-27 16:05:49.371 PET > LOG:  redirecting log output to logging collector process;
< 2018-09-27 16:05:49.371 PET > HINT:  Future log output will appear in directory "pg_log".
```

Procedemos a crear la base de datos ejecutando los siguientes scripts:

copiar manualmente el archivo adjunto en la siguiente ruta: /opt/dbdata del servidor

```
[root@lcvq-tab ~]# cd /opt/dbdata
```

el archivo debe contar con los siguientes permisos



Ingresar a la ruta y descomprimir el archivo

```

root@lcvq-tab-01:~
bash-4.2$ cd /opt/dbdata
bash-4.2$ unzip scripts.zip
Archive:  scripts.zip
  inflating: script_dashboard_control_panel.sql
  inflating: script_dashboard_setup.sql
bash-4.2$

```

Creación de database, tablespace, schema y rol del tablero de control, adicional a esto también se adjunta los procedures que se encargara de la carga de informacion



scripts\_creacion.zip

```
[root@lcvq-tab ~]# su postgres
```

```
bash-4.2$ psql -U postgres -a -f script_dashboard_setup.sql
```

```
root@lcvq-tab-01:~
bash-4.2$ cd /opt/dbdata
bash-4.2$ unzip scripts.zip
Archive:  scripts.zip
  inflating: script_dashboard_control_panel.sql
  inflating: script_dashboard_setup.sql
bash-4.2$ psql -U postgres -a -f script_dashboard_setup.sql

CREATE TABLESPACE ts_dashboard location '/opt/dbts';
CREATE TABLESPACE
--Create database CONTROL_PANEL
CREATE DATABASE CONTROL_PANEL with tablespace = 'ts_dashboard';
CREATE DATABASE
--Crea nuevo rol CONTROL_PANEL
CREATE ROLE CONTROL_PANEL WITH LOGIN  PASSWORD 'D4shb0ard_clv';
CREATE ROLE
--asigna los privilegios de la base de datos CONTROL_PANEL al rol CONTROL_PANEL
GRANT ALL PRIVILEGES ON DATABASE CONTROL_PANEL TO CONTROL_PANEL;
GRANT
--Asigna rol de crear Base de datos
ALTER ROLE CONTROL_PANEL CREATEDBS;
ALTER ROLE
ALTER DATABASE CONTROL_PANEL OWNER TO CONTROL_PANEL;
ALTER DATABASE
--cambiar a la base de datos CONTROL_PANEL
\c control_panel
You are now connected to database "control_panel" as user "postgres".
--Create schema CONTROL_PANEL
CREATE SCHEMA CONTROL_PANEL;
CREATE SCHEMA
ALTER DEFAULT PRIVILEGES IN SCHEMA CONTROL_PANEL GRANT INSERT, UPDATE, DELETE ON TABLES TO CONTROL_PANEL;
ALTER DEFAULT PRIVILEGES
--Privilegios del schema CONTROL_PANEL a al role CONTROL_PANEL
GRANT ALL PRIVILEGES ON SCHEMA CONTROL_PANEL TO CONTROL_PANEL;
GRANT
--Privilegios de las tablas del esquema de CONTROL_PANEL al rol CONTROL_PANEL
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA CONTROL_PANEL TO CONTROL_PANEL;
GRANT

bash-4.2$
```

```
[root@lcvq-tab ~]# su postgres
```

```
bash-4.2$ psql -U control_panel -d control_panel -a -f script_dashboard_control_panel.sql
```

```
root@lcvq-tab-01:~
GRANT
-- Name: SEQUENCE sq_id_rule_notification_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_id_rule_notification_pk TO PUBLIC;
GRANT
-- Name: SEQUENCE sq_id_rule_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_id_rule_pk TO PUBLIC;
GRANT
-- Name: SEQUENCE sq_id_schedule_config_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_id_schedule_config_pk TO PUBLIC;
GRANT
-- Name: SEQUENCE sq_id_schedule_process_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_id_schedule_process_pk TO PUBLIC;
GRANT
-- Name: SEQUENCE sq_id_table_column_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_id_table_column_pk TO PUBLIC;
GRANT
-- Name: SEQUENCE sq_id_user_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_id_user_pk TO PUBLIC;
GRANT
-- Name: SEQUENCE sq_master_table_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_master_table_pk TO PUBLIC;
GRANT
-- Name: SEQUENCE sq_parameter_table_pk; Type: ACL; Schema: control_panel; Owner: control_panel
--
GRANT ALL ON SEQUENCE control_panel.sq_parameter_table_pk TO PUBLIC;
GRANT
-- Name: DEFAULT PRIVILEGES FOR TABLES; Type: DEFAULT ACL; Schema: control_panel; Owner: postgres
--
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA control_panel REVOKE ALL ON TABLES FROM postgres;
psql:script_dashboard_control_panel.sql:98561: ERROR:  must be member of role "postgres"
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA control_panel GRANT INSERT,DELETE,UPDATE ON TABLES TO control_panel;
psql:script_dashboard_control_panel.sql:98562: ERROR:  must be member of role "postgres"
--
-- PostgreSQL database dump complete
--

bash-4.2$
```

## Configuración de Firewall

<b>Objetivo</b>	Habilitar el puerto 5432 de Postgres 9.6 para acceso remoto
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>• Permisos de root</li> </ul>

Agregamos una regla de firewall para el servicio postgres:

```
[root@localhost ~]# firewall-cmd --add-service=postgresql --zone=public --permanent
```

```

root@lcvq-tab-01:~
[root@lcvq-tab-01 ~]# firewall-cmd --add-service=postgresql --zone=public --permanent
success
[root@lcvq-tab-01 ~]#

```

## Habilitación de acceso remoto

<b>Objetivo</b>	Habilitar el puerto 5432 de Postgres 9.6 para acceso remoto
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>• Permisos de root</li> </ul>

Paramos la instancia de base de datos antes de realizar cambios en la configuración:

```
[root@lcvq-tab ~]# su postgres
```

```
bash-4.2$ cd /opt/dbdata
```

```
bash-4.2$ pg_ctl stop -D /opt/dbdata/
```

```
waiting for server to shut down.... done
```

```
server stopped
```

Una vez creado el rol y esquema de Postgres necesario para el tablero de control, podemos habilitar acceso remoto al servidor, para la administración de la base de datos.

```

[root@lcvq-tab ~]# cd /opt/dbdata/
[root@lcvq-tab dbdata]# echo 'host      all          all          192.168.0.0/16          md5' >>
pg_hba.conf

```

```

[root@lcvq-tab dbdata]# echo 'host      all          all          172.16.0.0/16          md5' >>
pg_hba.conf

```

Esto permitirá a la subnet 192.168.0.0/16 conectarse al servidor de datos mediante password con 'encoding' MD5.

Adicionalmente, debemos especificar el puerto público al cual el servidor de base de datos escucha:

```
[root@lcvq-tab ~]# cd /opt/dbdata/
```

**Actualizar con la IP del servidor:**

```
[root@lcvq-tab dbdata]# echo "listen_addresses = '127.0.0.1,172.16.5.38' " >> postgresql.conf
```

Iniciamos nuevamente la base de datos y verificamos la conexión remota mediante un cliente Postgres tal como pgAdmin

```
[root@lcvq-tab ~]# su postgres
```

```
bash-4.2$ pg_ctl start -D /opt/dbdata/
```



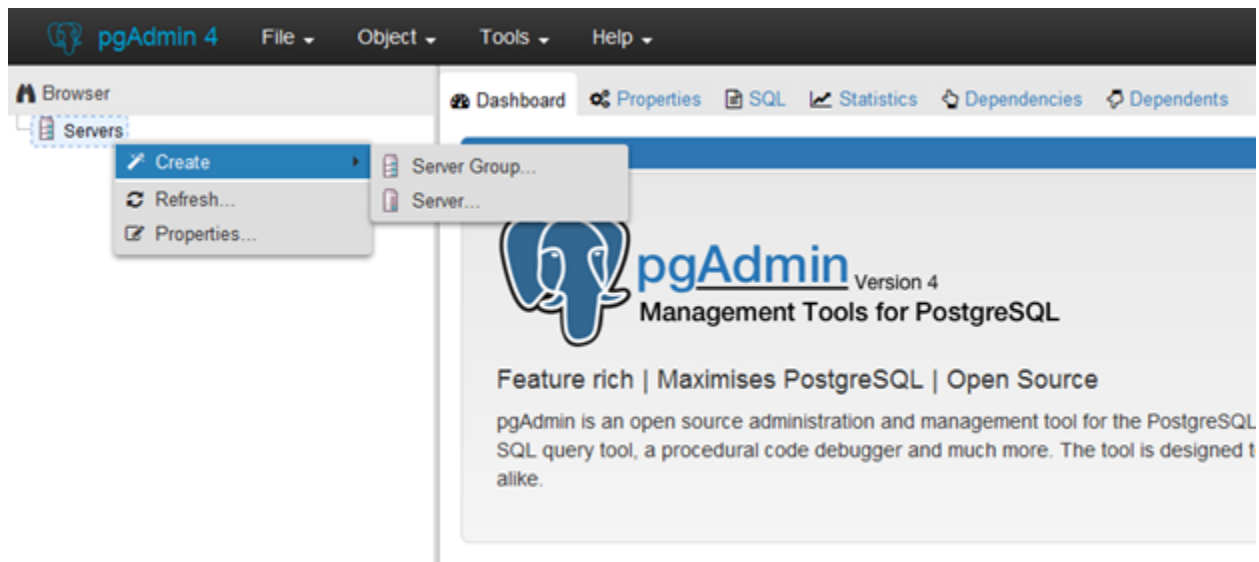
```
root@lcvq-tab-01:/opt
bash-4.2$ echo 'host      all          all          192.168.0.0/16      md5' >> pg_hba.conf
bash-4.2$ echo 'host      all          all          172.16.0.0/16      md5' >> pg_hba.conf
bash-4.2$ pwd
/opt/dbdata
bash-4.2$ echo "listen_addresses = '127.0.0.1,172.16.7.122' " >> postgresql.conf
bash-4.2$ pg_ctl start -D /opt/dbdata/
server starting
bash-4.2$ < 2019-02-19 17:12:58.880 EST > LOG:  redirecting log output to logging collector process
< 2019-02-19 17:12:58.880 EST > HINT:  Future log output will appear in directory "pg_log".
```

Para ver el estado:

```
bash-4.2$ pg_ctl status -D /opt/dbdata/
```

```
bash-4.2$ pg_ctl status -D /opt/dbdata/
pg_ctl: server is running (PID: 9896)
/usr/pgsql-9.6/bin/postgres "-D" "/opt/dbdata"
bash-4.2$
```

Seleccionamos Create > Server...



Create - Server

General

Connection

Advanced

Name

dbprod

Server group

Servers

Connect now?

☒

Comments

Either Host name or Host address must be specified.

i?

Save

Cancel

Reset



## Create - Server



General

Connection

Advanced

Host  
name/address

192.168.151.122

Port

5432

Maintenance  
database

postgres

Username

control\_panel

Password

.....

Save  
password?

☐

Role

control\_panel

SSL mode

Prefer



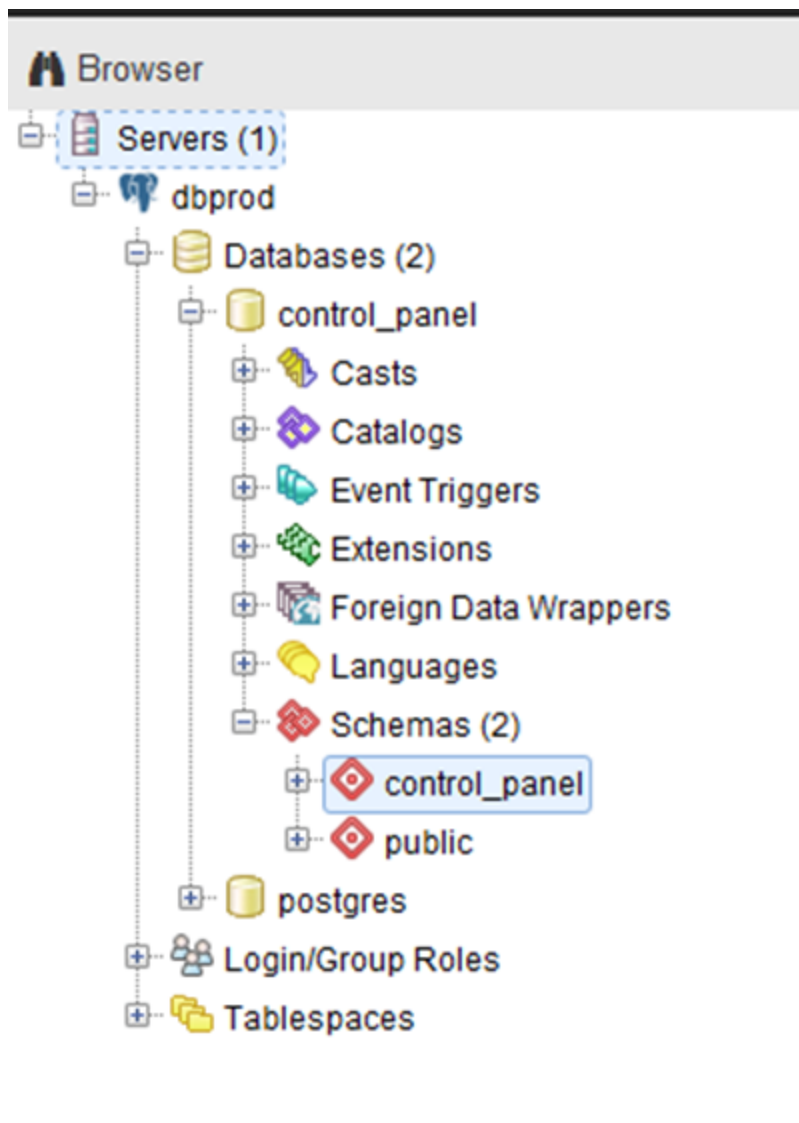
Save



Cancel



Reset



## Restricción de acceso remoto

Objetivo	Habilitar acceso remoto a nivel de base de datos Postgres 9.6
Requisitos	<ul style="list-style-type: none"><li>• Permisos de root</li></ul>

Para restringir acceso remoto al servidor, editamos el archivo `vim /opt/dbdata/pg_hba.conf` y removemos aquellas direcciones diferentes a 127.0.0.1. En nuestro caso, removemos la siguiente línea:

```
host    all             all             192.168.0.0/16      md5
host    all             all             172.16.0.0/16       md5
```

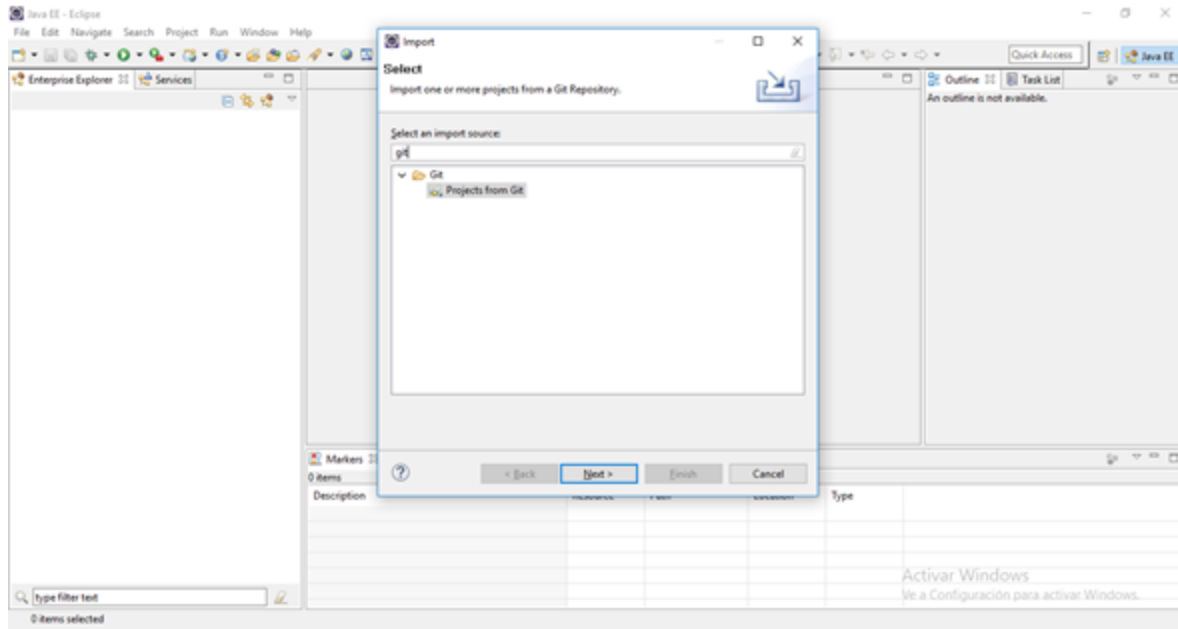
El archivo `pg_hba.conf` debe contener solo las siguientes líneas:

```
local   all             all                                     trust
host    all             all             127.0.0.1/32        trust
host    all             all             ::1/128              trust
```

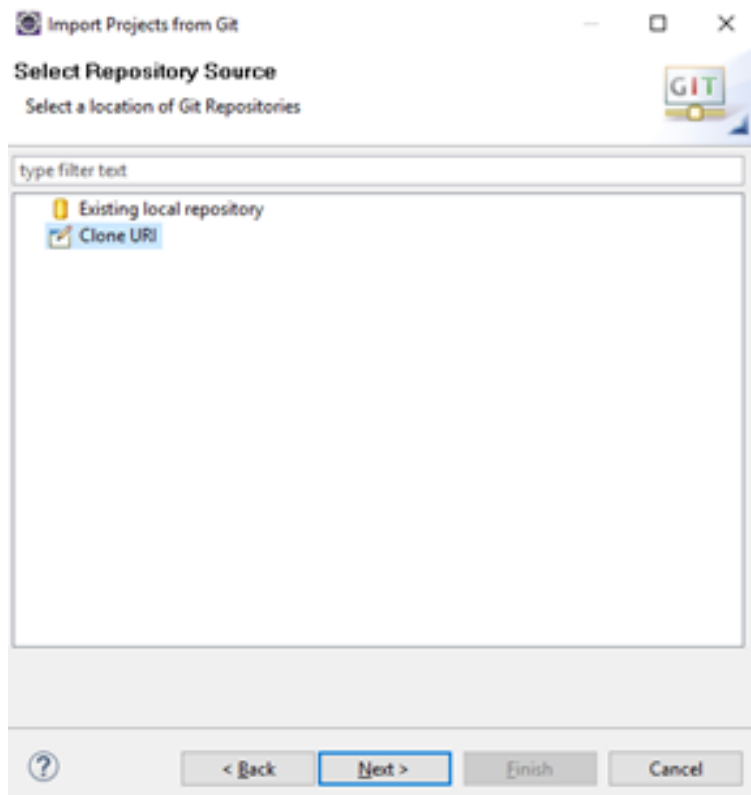
## Compilación de Aplicaciones desde Eclipse

<b>Objetivo</b>	Realizar la compilación de las fuentes de la aplicación a fin de obtener un WAR deployable
<b>Requisitos</b>	<ul style="list-style-type: none"> <li>• Acceso a Internet</li> <li>• Accesos de root</li> <li>• Servidor de aplicaciones levantado</li> </ul>

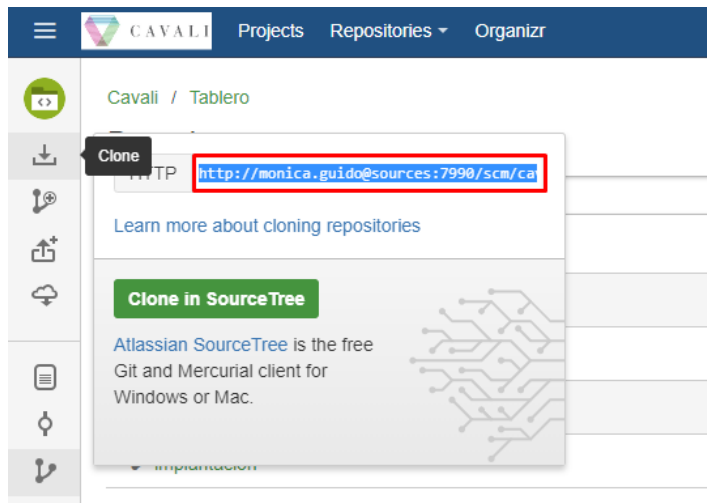
Debemos importar los proyectos necesarios en el espacio del proyecto, seleccionando. Import Projects from Git:



Seleccionamos Clone URI



En Location: URI: Pegamos la url del código fuente,



Import Projects from Git

**Source Git Repository**

Enter the location of the source repository.

**Location**

URI:

Host:

Repository path:

**Connection**

Protocol:

Port:

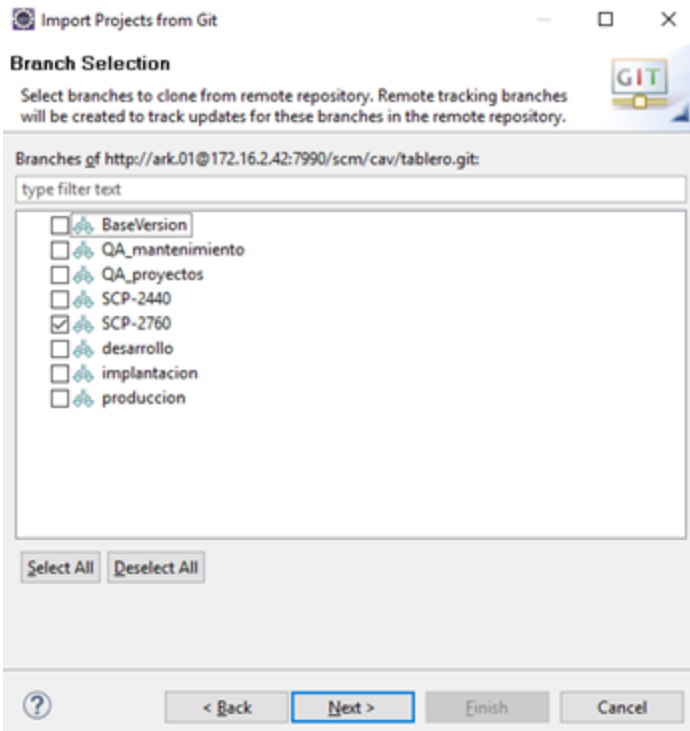
**Authentication**

User:

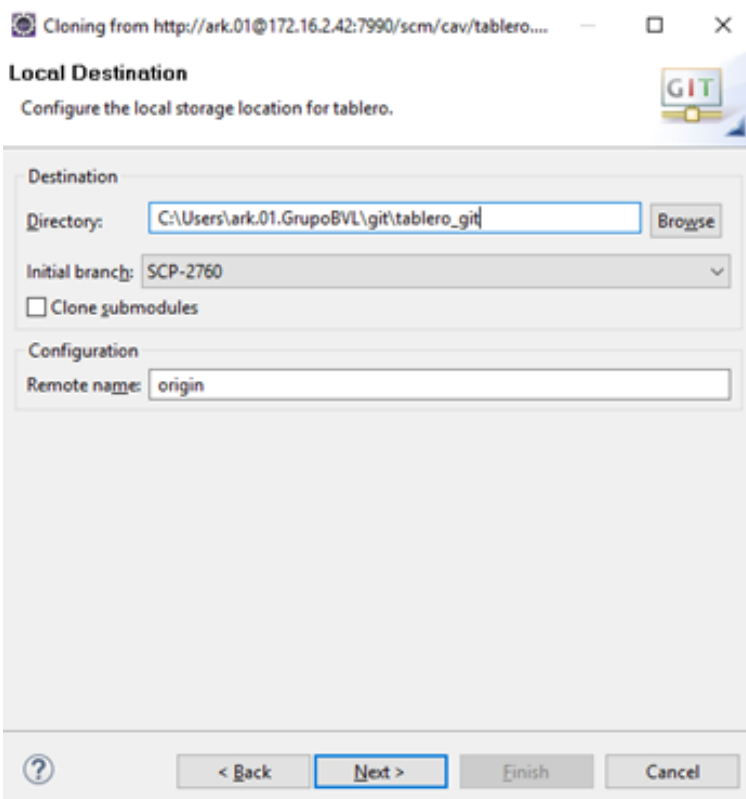
Password:

☐ Store in Secure Store

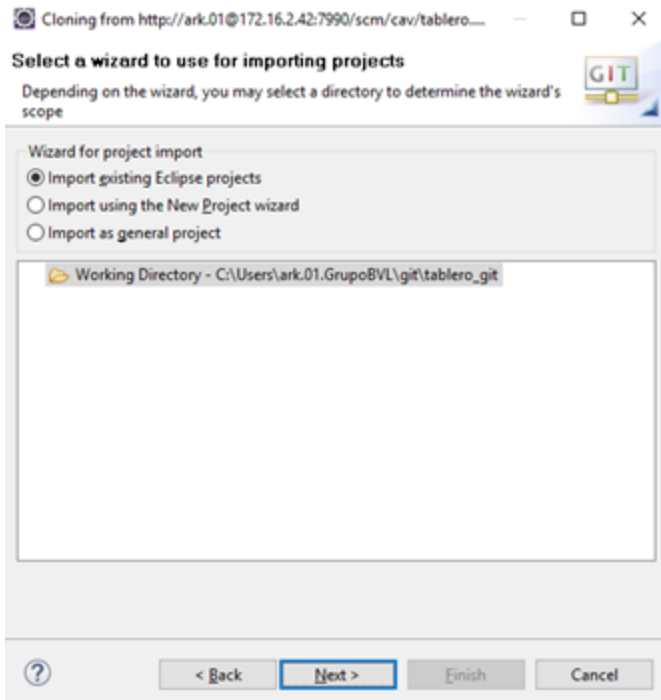
Seleccionamos el Branch actualizado:



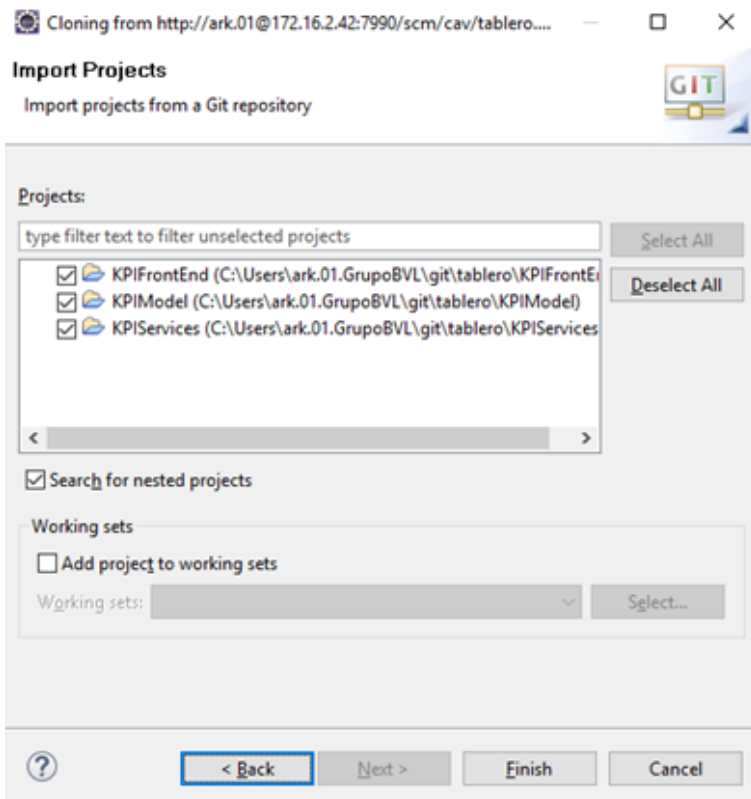
Seleccionamos la ruta local donde se almacenara el código:



Damos Siguiente:



Importamos los proyectos luego de haber sido descargados y completamos con “Finish”



Cambiamos la configuración de dos archivos, de acuerdo a la configuración del servidor:

Ruta	Valor
/opt/dbsource/ITE01/KPIServices/src/main/resources/configuration.properties	application.auth.method=dz



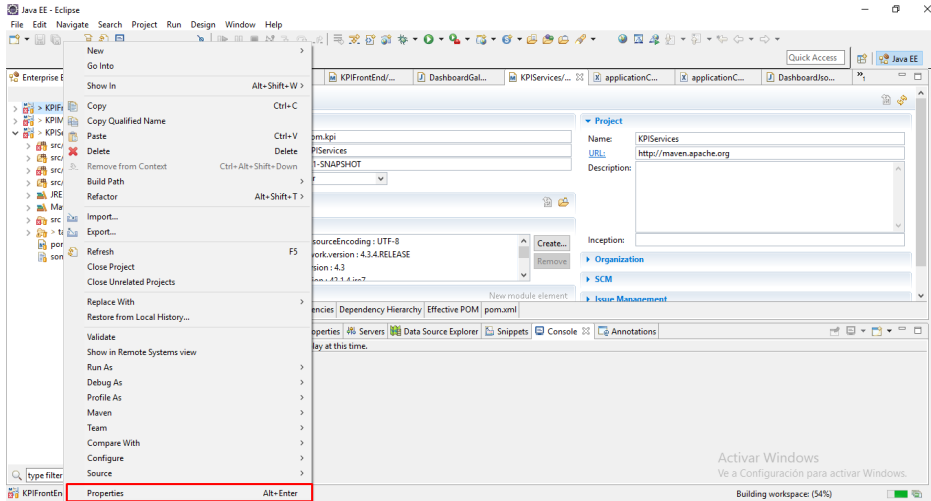
/opt/dbsource/ITE01/KPIFrontEnd/src/main/webapp/resources/app/services/serviceRequest/configurationService.js

LINEA 8: return  
'http://<IPSERVER:PUERT

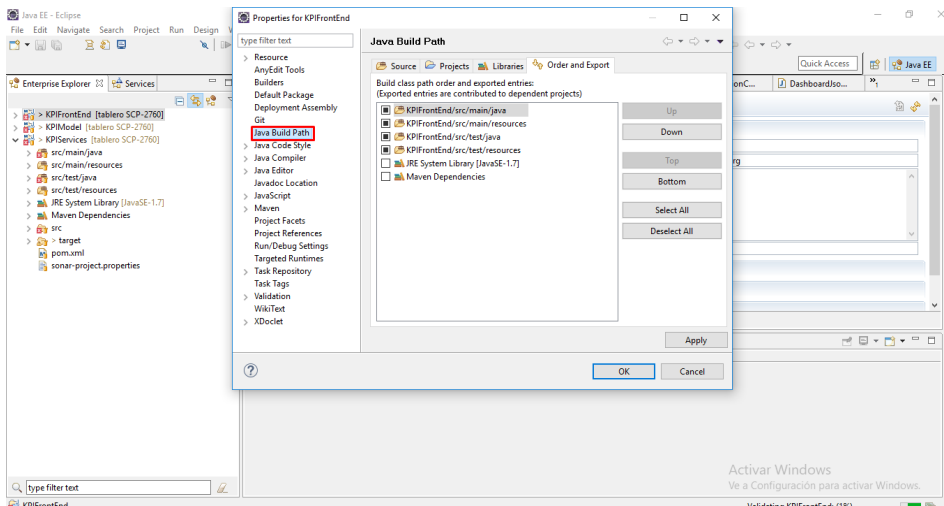
LINEA 15: return  
'ws://<IPSERVER:PUERTC

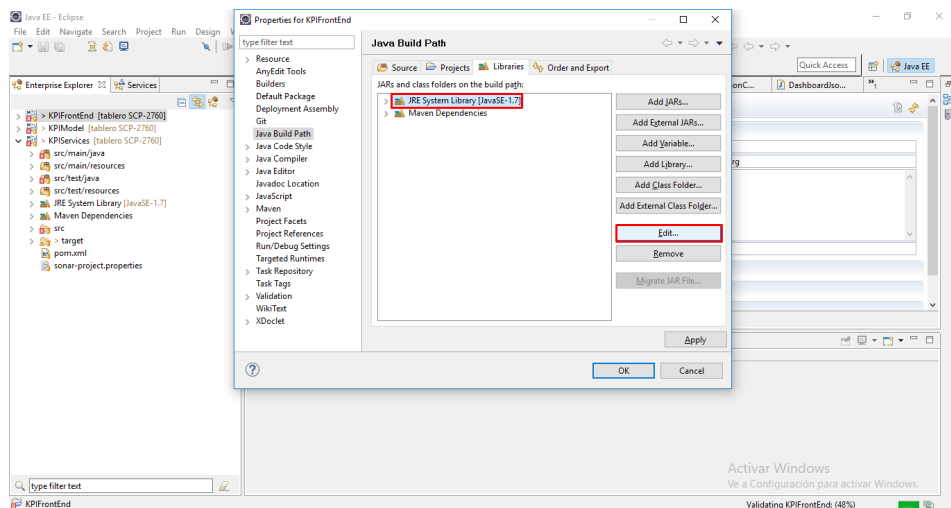
## Cambiar JRE por JDK

Dar click derecho sobre el proyecto descargado, e ingresar a la opción Properties

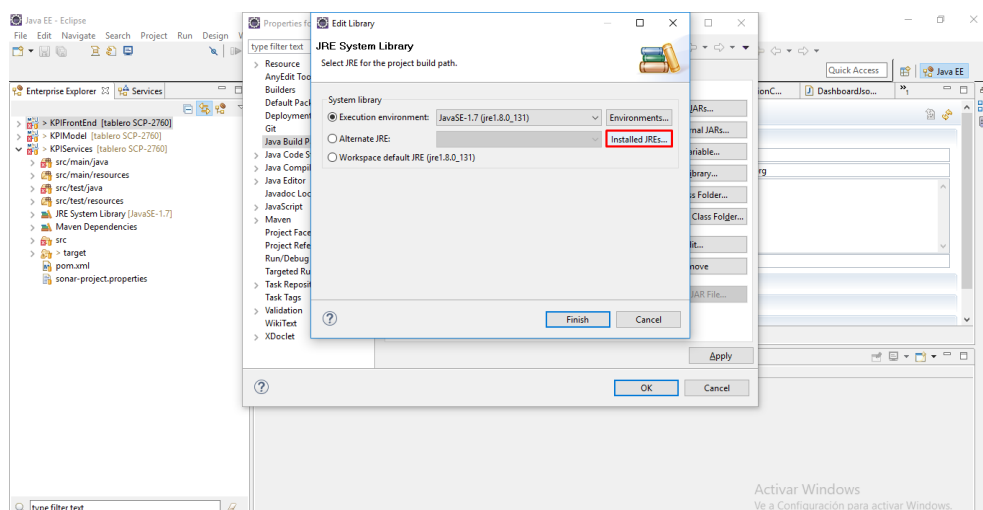


Ingresar a la opción Java Build Path, ingresar a la pestaña "Libraries" seleccionar el JRE del proyecto y pulsar el botón Edit.

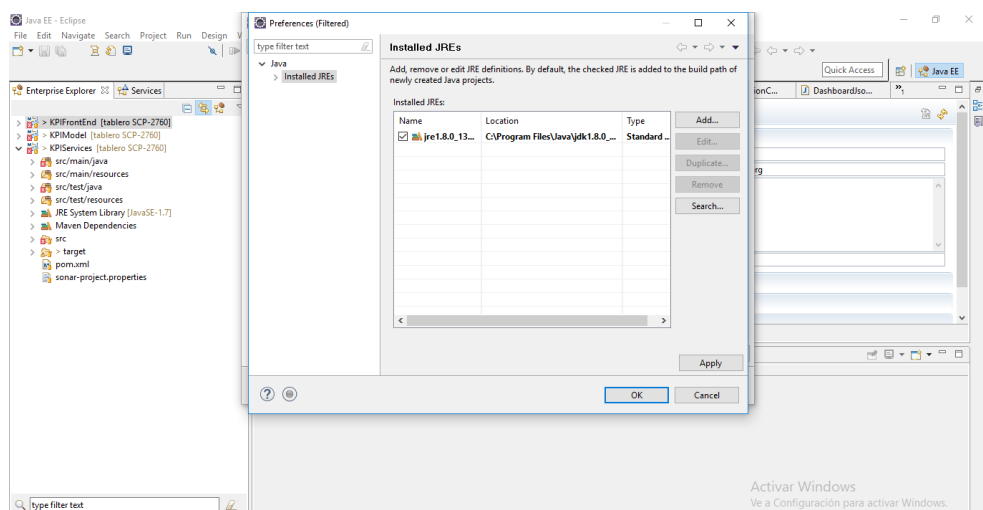


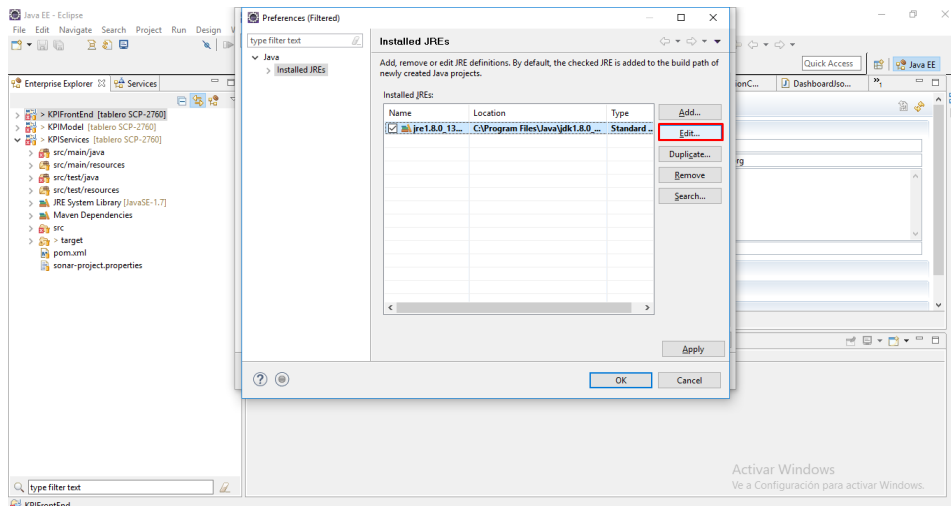


Se muestra la ventana "Edit Library", pulsar el botón "Installed JREs".

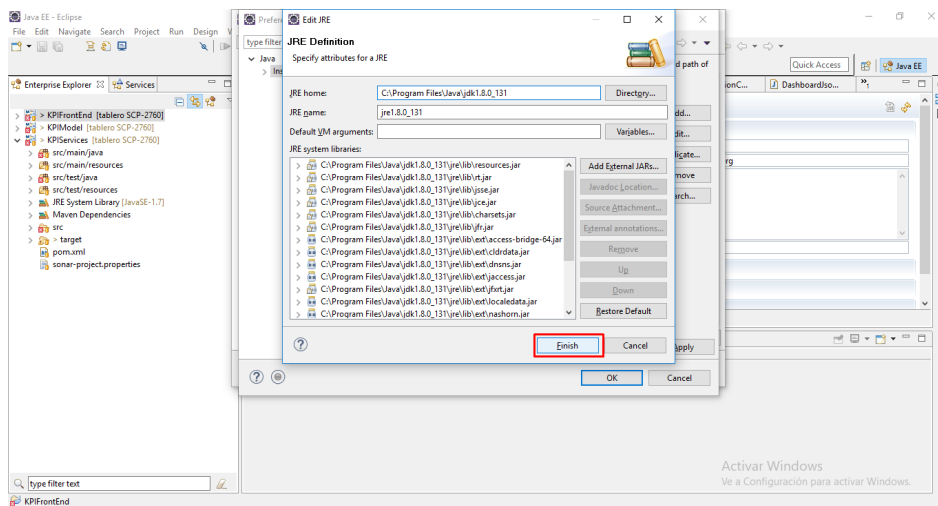
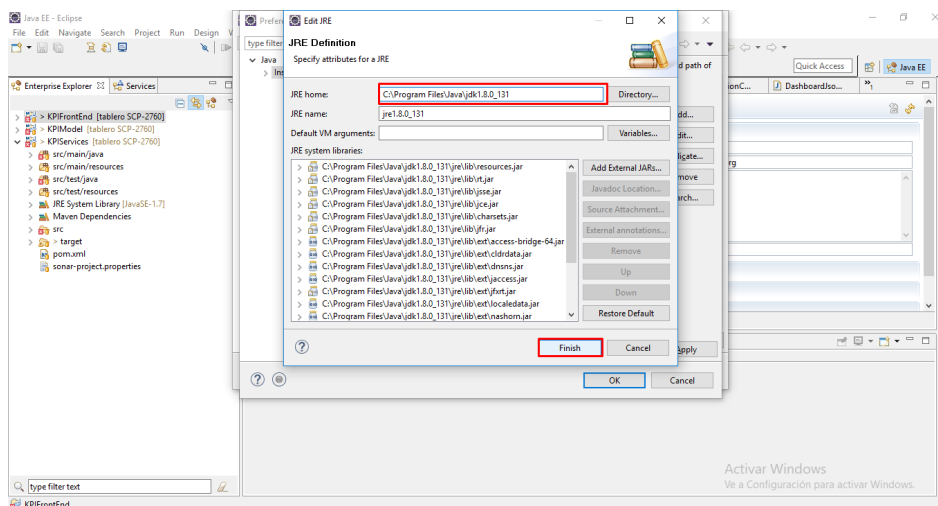


Se muestra la pantalla Installed JREs, seleccionar el JRE y pulsar el botón EDIT.

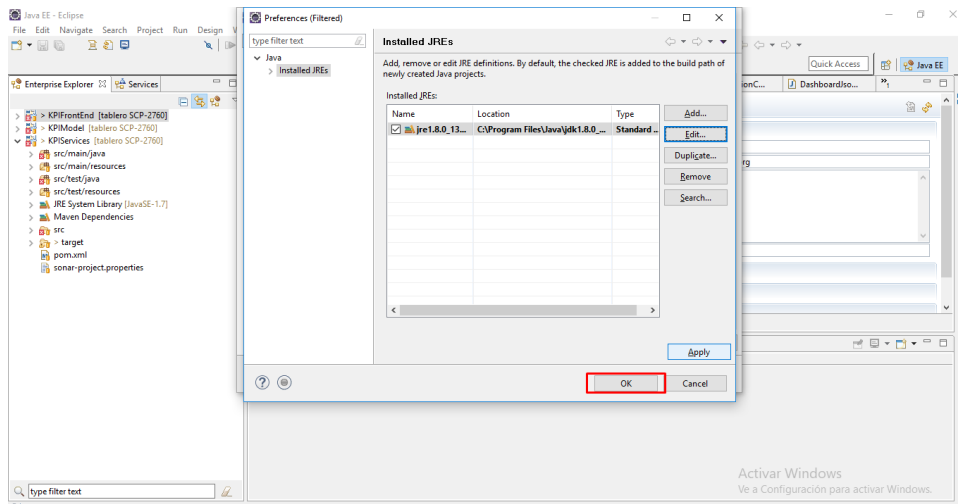




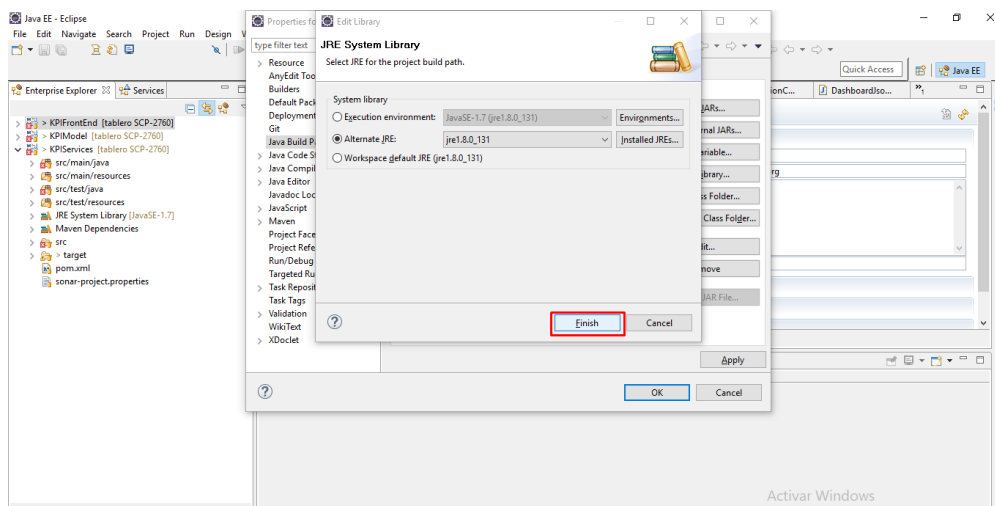
Verificar la ruta en la que se tenga el JDK descargado\*, ingresar la ruta y pulsar el botón Finish.



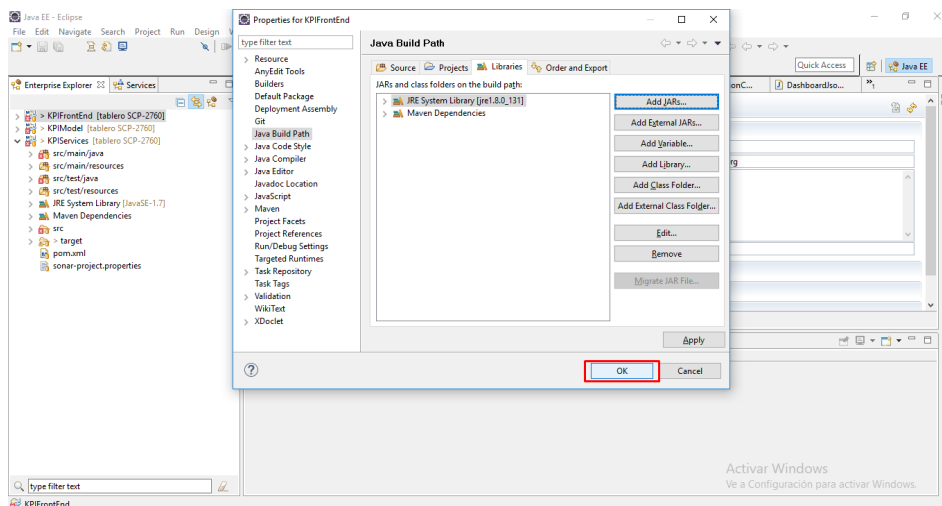
Pulsar el botón OK.



Pulsar el botón Finish.



Pulsar el botón OK.

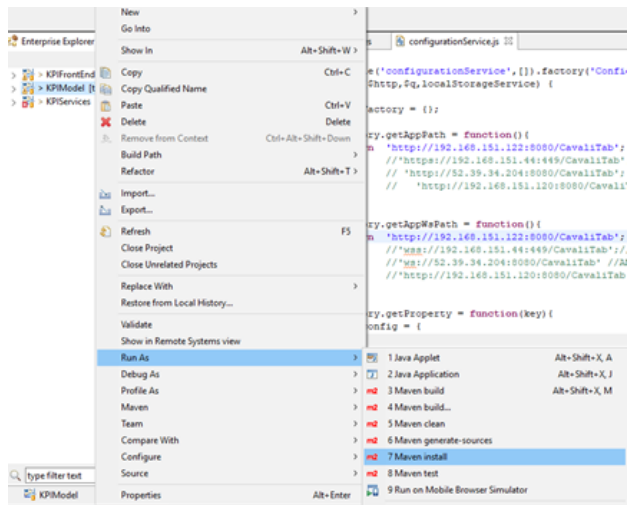


\*Verificar que la pc en la que se está realizando la migración, tenga el jdk descargado (Versión 1.8), en caso no está instalado descargarlo (<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>).

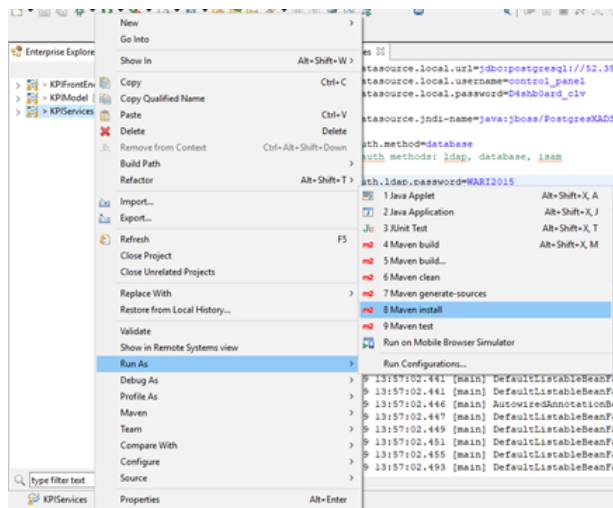
Realizar el cambio de JRE a JDK en cada proyecto.

Para compilar, debemos ejecutar "Maven Install" en todos los proyectos, en el siguiente orden:

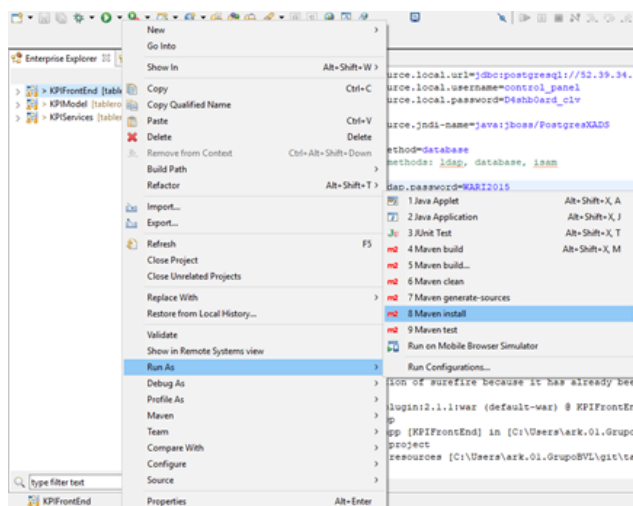
## KPIModel:



## KPIServices



## KPIFrontEnd



## Despliegue de Aplicaciones

<b>Objetivo</b>	Desplegar los módulos del Tablero de Control en el servidor de Aplicaciones JBoss
<b>Requisitos</b>	<ul style="list-style-type: none"><li>• Aplicación compilada con Maven (3.5)</li><li>• Servidor de aplicaciones JBoss iniciado</li></ul>

La aplicación consta de 3 módulos:

Módulo de Vista	KPIFrontEnd
Módulo de Servicios	KPIServices
Módulo de Datos	KPIModel

Una vez compilado el sistema, tomamos el ejecutable de la carpeta **KPIFrontEnd/target/KPIFrontEnd.war**

o Local (D:) > monica.guido > workspace_tableros > tablero > KPIFrontEnd > target					
Nombre	Fecha de modifica...	Tipo	Tamaño		
classes	3/10/2018 11:58 a. ...	Carpeta de archivos			
generated-sources	3/10/2018 12:33 p....	Carpeta de archivos			
generated-test-sources	3/10/2018 12:33 p....	Carpeta de archivos			
KPIFrontEnd	3/10/2018 12:33 p....	Carpeta de archivos			
m2e-wtp	3/10/2018 11:58 a. ...	Carpeta de archivos			
maven-archiver	3/10/2018 12:33 p....	Carpeta de archivos			
maven-status	3/10/2018 12:33 p....	Carpeta de archivos			
surefire-reports	3/10/2018 12:33 p....	Carpeta de archivos			
test-classes	3/10/2018 11:58 a. ...	Carpeta de archivos			
KPIFrontEnd.war	3/10/2018 12:33 p....	Archivo WAR	62,749 KB		

Ingresar a <http://<IPSERVER>:9990/console/App.html#deployments>

Seleccionamos el botón ADD y seleccionamos el WAR:

## Create Deployment

Managed Unmanaged

### Step1/2: Deployment Selection

Please choose a file that you want to deploy.

Seleccionar archivo

KPIFrontEnd.war



Damos "Next"

## Create Deployment

### Step 2/2: Verify Deployment Names

[Need Help?](#)

Name:

Runtime Name:

Enable: ☒



Finalmente, "Save" y podremos verificar que el archivo se desplegó:

## DEPLOYMENTS

### Deployments

Currently deployed application components.

Available Deployments

Filter:

[Add](#)[Remove](#)[En/Disable](#)[Replace](#)

▶ KPIFrontEnd.war ✓

Deployment

[Need Help?](#)

Name: KPIFrontEnd.war

Activar Window  
Ve a Configuración

← → ↻ ⓘ No es seguro 192.168.151.122:8080/KPIFrontEnd/pages-login.html ☆ ⓘ



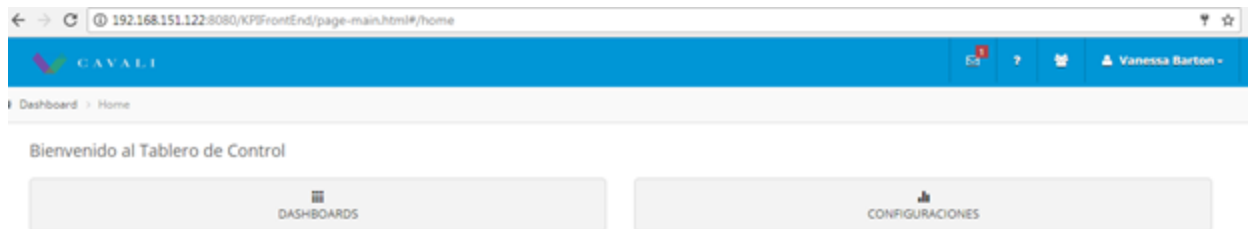
Ingrese al Sistema

👤 Usuario

🔒 Contraseña

Ingresar >





### Código fuente a Modificar:

- configurationService.js:

Cambias las urls por las correctas, en este caso como se usa la autenticacion por ISAM se usara

```
1 'use strict';
2 angular.module('configurationService', []).factory('ConfigurationService', ['$http', '$q', 'localStorageService',
3     function($http,$q,localStorageService) {
4
5         var dataFactory = {};
6
7         dataFactory.getAppPath = function(){
8             return 'https://192.168.151.41:446/CavaliTab';
9         }
10
11         dataFactory.getAppWsPath = function(){
12             return 'wss://192.168.151.41:446/CavaliTab';
13         }
14     }
15 ])
```

- configuration.properties:

Modificar:

application.auth.method por isam o database, esto es para saber que tipo de autenticación usara

application.auth.isam.url por la url del ISAM desde la cual se autenticara

```

1 application.datasource.local.url=jdbc:postgresql://52.39.34.204:5432/control_panel_kpi
2 application.datasource.local.username=control_panel
3 application.datasource.local.password=D4shb0ard_clv
4
5 application.datasource.jndi-name=java:jboss/PostgresXADS
6
7 application.auth.method=isam
8 ## supported auth methods: ldap, database, isam
9
10 application.auth.ldap.password=WARI2015
11 application.auth.ldap.url=ldap://172.16.7.147:389
12 application.auth.ldap.userDn=cn=CAVACCESS,cn=users,dc=cavali,dc=com,dc=pe
13 application.auth.ldap.base=cn=users,dc=cavali,dc=com,dc=pe
14 application.auth.ldap.search=uid={0},${application.auth.ldap.base}
15
16 #application.auth.isam.url = https://192.168.151.44:449
17 application.auth.isam.url = https://192.168.151.41:446
18 application.auth.isam.url.app = ${application.auth.isam.url}/CavaliTab
19 application.auth.isam.url.changepwd = ${application.auth.isam.url}/pkmspasswd
20 application.auth.isam.url.logout = ${application.auth.isam.url}/pkmslogout
21 application.auth.isam.header.user = IV-USER
22 application.auth.isam.header.address = IV-REMOTE-ADDRESS
23
24 application.auth.secret = KPId4shboard
25
26 application.users.autogenerate-usernames=false
27
28

```

## PASOS A SEGUIR UNA VEZ CULMINADA EL DESPLIEGUE DE LA APLICACION

Se ejecutara los procedures de carga en la bd



procedures\_carga.txt

ahora, procederemos a subir el proyecto backend.

Descargamos el siguiente jar:



Los unimos con el 7zip y Luego el archivo arkinmail-0.0.1-SNAPSHOT lo copiamos al servidor en la carpeta /opt

y ejecutamos el siguiente comando para iniciarlo

```
[root@lcvq-tab ~]# nohup java -jar arkinmail-0.0.1-SNAPSHOT.jar &
```

y para detenerlo solo buscaremos el proceso con el siguiente comando

```
[root@lcvq-tab ~]# ps aux | grep arkinmail
```

```
[root@lcvq-tab ~]# kill -9 #numero_del_proceso
```

## Inicio de aplicaciones:

### Postgres

```
[root@lcvq-tab ~]# su postgres
```

```
[root@lcvq-tab ~]# cd /opt/dbdata
```

```
[root@lcvq-tab ~]# pg_ctl status -D /opt/dbdata/
```

```
[root@lcvq-tab ~]# pg_ctl start -D /opt/dbdata/
```

```
[root@lcvq-tab ~]# pg_ctl stop -D /opt/dbdata/
```

### backend

```
[root@lcvq-tab ~]# nohup java -jar arkinmail-0.0.1-SNAPSHOT.jar &
```

### Jboss

```
[root@lcvq-tab ~]# service jboss-as-standalone.dat status
```

```
[root@lcvq-tab ~]# service jboss-as-standalone.dat stop
```

```
[root@lcvq-tab ~]# service jboss-as-standalone.dat start
```

## DBLINK:

