



iBit

Illes Balears
innovació
tecnològica

Instal·lació de SISTRA

Guia Ràpida d'Instal·lació de SISTRA





Informació general del document.

Descripció.

Títol: Guia Ràpida d'Instal·lació de SISTRA
Estat: Esborrany/Aprovat
Versió: 1.0
Autor/s: Antoni Nadal
Creat: 23/11/09
Modificat 21/12/11
Fitxer: Guia_Rapida_de_Instalacio_de_Sistra.odt

Històric de modificacions.

Comentari:	Autor/s:	Data:
------------	----------	-------

Font documental.



Index de contingut

1.-Introducció.....	3
2.-Configurar JBOSS.....	3
2.1.-Llibreries.....	3
2.2.-Fitxers de Configuració.....	4
2.3.-Copia de binaris.....	6
3.-Gestió de BBDD.....	7
4.-ANNEX I: Compilar SISTRA des de subversion de sourceforge.....	11
5.-ANNEX II: Script que automatitza la cridada als scripts sql.....	12

1.- Introducció

Aquest document explicar com a partir d'un binari SISTRA o una compilació del codi de SISTRA, es pot instal·lar sobre un servidor jboss i el sistema gestor de bbdd.

(A) COMPILAT: S'han de seguir les instruccions de l'annex I d'aquest document.

(B) DESCARREGAR BINARI: Accedint al projecte sistra de sourceforge en podem descarregar el binari corresponents. Per exemple accedir via web a la següent adreça <http://sourceforge.net/projects/sistra/files/sistra-1.1/sistra-1.1.1/bin/> i descarregar el fitxer sistra-1.1.1_postgresql_jaas.zip o sistra-1.1.1_oracle_jaas.zip, segons la BBDD que tinguem. Després descomprimir el zip en l'arrel del nostre home.

Aquí s'enumeren la llista d'accions per posar en marxa un sistema SISTRA:



2.- Configurar JBOSS

2.1.- Llibreries

- 1.- Copiar commons-codec-1.3.jar a /usr/local/jboss-3.2.8.SP1/server/default/lib.

```
$ sudo cp ~/sistra/lib/commons-codec-1.3.jar /usr/local/jboss-3.2.8.SP1/server/default/lib
```

- 2.- Actualitzar arxius de xerces:

```
$ sudo cp ~/sistra/doc/resources/xerces2.7.1/* /usr/local/jboss-3.2.8.SP1/lib/endorsed/
```

- 3.- Actualització libreria SAAJ

```
$ sudo rm /usr/local/jboss-3.2.8.SP1/server/default/lib/jboss-saaj.jar
$ sudo cp ~/sistra/lib/saaj-api.jar /usr/local/jboss-3.2.8.SP1/server/default/lib/
$ sudo cp ~/sistra/lib/saaj-impl.jar /usr/local/jboss-3.2.8.SP1/server/default/lib/
```

2.2.- Fitxers de Configuració

- 1.- Editar el fitxer /usr/local/jboss-3.2.8.SP1/server/default/conf/jboss-service.xml:

- 1.1.- Modificar la línia afegint "deploysistra/":

ACTUAL	MODIFICAT
<pre><attribute name="URLs"> deploy/ </attribute></pre>	<pre><attribute name="URLs"> deploy/, deploysistra/ </attribute></pre>

- I crear el nou directori deploysistra:

```
$ sudo mkdir -p /usr/local/jboss-3.2.8.SP1/server/default/deploysistra
```

- 1.2.- Indicar ubicació fitxers de propietats. Afegir el següent codi al final del fitxer (abans de </server>):

```
<mbean code="org.jboss.varia.property.SystemPropertiesService"
name="jboss:type=Service,name=BootProperties">
<attribute name="Properties">
```



```
<!-- En /config_sistra/sistra estaran los archivos de configuracion -->
ad.path.properties=/config-sistra/
</attribute>
</mbean>
```

Executar les següents comandes:

```
$ sudo mkdir /config-sistra
$ sudo mkdir /config-sistra/sistra
$ sudo cp ~/sistra/doc/resources/config/sistra/* /config-sistra/sistra
```

2.- Afegir DISPLAY en Sistemes unix. En /usr/local/jboss-3.2.8.SP1/bin/run.sh afegir les següent línies:

```
DISPLAY=:0.0
export DISPLAY
```

3.- Afegir la següent entrada al final del fitxer /usr/local/jboss-3.2.8.SP1/server/default/deploy/jms/jbossmq-destinations-service.xml

```
<mbean code="org.jboss.mq.server.jmx.Queue"
name="jboss.mq.destination:service=Queue,name=AvisadorBTE">
<depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
<!-- redelivery delay 30 [s] -->
<attribute name="RedeliveryDelay">30000</attribute>
<!-- unlimited redelivery -->
<attribute name="RedeliveryLimit">3</attribute>
</mbean>
```

4.- Configurar Correu

(a) Copiar fitxer de service:

```
$sudo cp ~/sistra/doc/resources/bbdd/datasources/mobtratel-mailTest-
service.xml /usr/local/jboss-3.2.8.SP1/server/default/deploy/sistra/
```

(b) Editar el fitxer /usr/local/jboss-3.2.8.SP1/server/default/deploy/sistra/mobtratel-mailTest-service.xml i configurar-ho:



```
<!DOCTYPE server>
<server>
  <mbean code="org.jboss.mail.MailService"
    name="jboss:service=MobtratelMailTest">
    <attribute name="JNDIName">java:/es.caib.mobtratel.mailTest</attribute>
    <attribute name="User">anadal@ibit.org</attribute>
    <attribute name="Password">contrasenya</attribute>
    <attribute name="Configuration">
      <configuration>
        <property name="mail.transport.protocol" value="smtp"/>
        <property name="mail.smtp.host" value="smtp.mailserver.es"/>
        <property name="mail.from" value="cuental@mailserver.es"/>
        <property name="mail.debug" value="false"/>
        <property name="mail.smtp.auth" value="true"/>
      </configuration>
    </attribute>
  </mbean>
</server>
```

5.- Descomentar entrada dins el fitxer /usr/local/jboss-3.2.8.SP1/server/default/conf/jboss-service.xml

ACTUAL	MODIFICAT
<pre><mbean code="org.jboss.tm.XidFactory" name="jboss:service=XidFactory"> <!--attribute name="Pad">true</attribute--> </mbean></pre>	<pre><mbean code="org.jboss.tm.XidFactory" name="jboss:service=XidFactory"> <attribute name="Pad">true</attribute> </mbean></pre>

6.- Configurar accés dels usuaris. Afegir al final del fitxer /usr/local/jboss-3.2.8.SP1/server/default/conf/login-config.xml abans de "</policy>" la següent entrada:

```
<application-policy name = "seycon">
<authentication>
<login-module code = "es.caib.mock.loginModule.MockCertificateLoginModule"
flag = "sufficient">
<module-option name="roleTothom">tothom</module-option>
</login-module>
<login-module code = "es.caib.mock.loginModule.MockDatabaseLoginModule"
flag = "sufficient">
<module-option name="unauthenticatedIdentity">nobody</module-option>
<module-option name = "dsJndiName">java:/es.caib.mock.loginModule.db</module-option>
<module-option name = "principalsQuery">SELECT USU_PASS,USU_NOM,USU_NIF FROM
SC_WL_USUARI WHERE USU_CODI = ?</module-option>
<module-option name = "rolesQuery">SELECT UGR_CODGRU, 'Roles' FROM SC_WL_USUGRU
WHERE UGR_CODUSU = ?</module-option>
</login-module>
</authentication>
</application-policy>
```

7.- Modificar el Tomcat per habilitar el single sign on entre les aplicacions. Per això hi ha que



descomentar el valve que l'implementa el fitxer /usr/local/jboss-3.2.8.SP1/server/default/deploy/jbossweb-tomcat50.sar/server.xml

```
<Valve className="org.apache.catalina.authenticator.SingleSignOn" debug="0"/>
```

2.3.- Copia de binaris

1.- Hi ha que col·locar el jar loginModuleMOCK.jar en /server/default/lib. A més en aquest directori també es requereixen les llibreries de bouncy castle pel maneig de certificats (bcmail.jar, bcprov.jar i bctsp.jar). Podem trobar aquestes llibreries en el directorio del projecte: ~/sistra/pluginsMOCK/lib. Són necessàries si s'empra el modul de login de test i el plugin de firma de test.

```
$ cp ~/sistra/pluginsMOCK/product/lib/loginModuleMOCK.jar /usr/local/jboss-3.2.8.SP1/server/default/lib
$ cp ~/sistra/pluginsMOCK/lib/bcmail.jar /usr/local/jboss-3.2.8.SP1/server/default/lib
$ cp ~/sistra/pluginsMOCK/lib/bcprov.jar /usr/local/jboss-3.2.8.SP1/server/default/lib
$ cp ~/sistra/pluginsMOCK/lib/bctsp.jar /usr/local/jboss-3.2.8.SP1/server/default/lib
```

2.- Copiar ears a deploysistra

```
$ sudo cp ~/sistra/product/ear/1-sistra.ear /usr/local/jboss-3.2.8.SP1/server/default/deploysistra/
$ sudo cp ~/sistra/pluginsMOCK/product/ear/2-modul-pluginsMOCK.ear /usr/local/jboss-3.2.8.SP1/server/default/deploysistra/
```

3.- Gestió de BBDD

1.- Els datasources definits a ~/sistra/doc/resources/bbdd/datasources/ s'han de copiar a /usr/local/jboss-3.2.8.SP1/server/default/deploysistra/:

- audita-ds.xml
- loginMOCK-ds.xml
- redose-ds.xml
- bantel-ds.xml
- mobtratel-ds.xml
- sistra-ds.xml
- form-ds.xml
- zonaper-ds.xml



```
$sudo cp ~/sistra/doc/resources/bbdd/datasources/[bbdd]/*.xml
/usr/local/jboss-3.2.8.SP1/server/default/deploy/sistra/
```

on [bbdd] pot ser oracle o postgresql.

2.- Crear usuaris i BBDD:

2.1.- Connectar-se a la BBDD

```
$ sudo bash
$ su postgres
$ psql -U postgres
```

2.2.- Per cada BBDD crearem un usuari:

```
CREATE USER "audita" WITH ENCRYPTED PASSWORD 'audita' NOCREATEUSER;
CREATE USER "loginMock" WITH ENCRYPTED PASSWORD 'loginMock' NOCREATEUSER;
CREATE USER "redose" WITH ENCRYPTED PASSWORD 'redose' NOCREATEUSER;
CREATE USER "bantel" WITH ENCRYPTED PASSWORD 'bantel' NOCREATEUSER;
CREATE USER "mobtratel" WITH ENCRYPTED PASSWORD 'mobtratel' NOCREATEUSER;
CREATE USER "sistra" WITH ENCRYPTED PASSWORD 'sistra' NOCREATEUSER;
CREATE USER "form" WITH ENCRYPTED PASSWORD 'form' NOCREATEUSER;
CREATE USER "zonaper" WITH ENCRYPTED PASSWORD 'zonaper' NOCREATEUSER;
```

2.3.- Crear la BBDD

```
CREATE DATABASE "audita" WITH OWNER=audita;
CREATE DATABASE "loginMock" WITH OWNER="loginMock";
CREATE DATABASE "redose" WITH OWNER=redose;
CREATE DATABASE "bantel" WITH OWNER=bantel;
CREATE DATABASE "mobtratel" WITH OWNER=mobtratel;
CREATE DATABASE "sistra" WITH OWNER=sistra;
CREATE DATABASE "form" WITH OWNER=form;
CREATE DATABASE "zonaper" WITH OWNER=zonaper;
```

2.4.- Sortir

```
\q
```

3.- Permisos d'accés a la BBDD per part de l'usuari i d'altres host

3.1.- S'ha d'editar el fitxer /etc/postgresql/8.4/main/pg_hba.conf i afegir les següents línies

```
## ===== audita =====
local audita          audita          password
host      audita      audita      www.xxx.yyy.zzz/24  trust
## ===== loginMock =====
local loginMock      loginMock          password
```




```

host      loginMock      loginMock      www.xxx.yyy.zzz/24  trust
## ===== redose =====
local redose      redose      password
host      redose      redose      www.xxx.yyy.zzz/24  trust
## ===== bantel =====
local bantel      bantel      password
host      bantel      bantel      www.xxx.yyy.zzz/24  trust
## ===== mobtratel =====
local mobtratel      mobtratel      password
host      mobtratel      mobtratel      www.xxx.yyy.zzz/24  trust
## ===== sistra =====
local sistra      sistra      password
host      sistra      sistra      www.xxx.yyy.zzz/24  trust
## ===== form =====
local form      form      password
host      form      form      www.xxx.yyy.zzz/24  trust
## ===== zonaper =====
local zonaper      zonaper      password
host      zonaper      zonaper      www.xxx.yyy.zzz/24  trust

```

On posa `www.xxx.yyy.zzz` l'hem de substituir per els tres primers números de la nostra IP. Per exemple si la nostra IP es `192.168.121.132` llavors on posa `www.xxx.yyy.zzz` hem d'escriure `192.168.121.132`. Podem obtenir la nostra IP escrivint `ipconfig` en sistemes Windows o `ifconfig` en sistemes linux.

3.2.- Reiniciem el postgres per a que els canvis s'apliquin:

```
$ sudo /etc/init.d/postgresql restart
```

3.3.- Si no teniu accés des d'ordinadors diferents a localhost, seguiu les recomanacions el que es diu en la següent pàgina:

<http://www.cyberciti.biz/tips/postgres-allow-remote-access-tcp-connection.html>

4.- Crear BBDD i configurar-la (Per cadascun dels usuaris i BBDD)

4.1.- Connectar-se al servidor de BBDD. Si la BBDD esta en el mateix servidor executar

```
$ psql -h localhost -p 5432 -U [USER] -W -d [BBDD]
```

i si està en un altra servidor llavors executar

```
$ psql -h www.xxx.yyy.zzz -p 5432 -U [USER] -W -d [BBDD]
```

4.2.- Donam permisos al usuari:

```

GRANT ALL PRIVILEGES ON DATABASE "[USER]" TO [BBDD];
GRANT ALL PRIVILEGES ON SCHEMA PUBLIC TO [USER];

```



4.3.- Importam l'estructura i dades de taules per a la BBDD en concret

```
\i ~/sistra/doc/resources/bbdd/scripts/[bbdd]/version1.x.y/[BBDD]-????
```

4.4.- Sortir:

```
\q
```

Per exemple per "redose" i per una bbdd postgresql, la llista de comandes a executar és::

```
$ psql -h localhost -p 5432 -U redose -W -d redose
GRANT ALL PRIVILEGES ON DATABASE "redose" TO redose;
GRANT ALL PRIVILEGES ON SCHEMA PUBLIC TO redose;
\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.2/redose-create.sql
\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.2/redose-initialData.sql
\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.3/redose-update.sql
\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.1.0/redose_update.sql
\q
```

Fer el mateix per la resta d'scripts.

En l'ANNEX II, podeu trobar un script que automatitza la cridada als scripts sql (Només per postgresql).

5.- Modificar postgresql.conf

Configurar arxiu postgresql.conf i modificar entrada "# max_prepared_transactions = 0" per "max_prepared_transactions = 20".

6.- Funcions d'Oracle per PostgreSQL

A causa que algunes parts de SISTRA estan implementades internament fent ús de funcions específiques d'Oracle en les consultes SQL, cal instal·lar dins del PostgreSQL un paquet anomenat orafce format per les funcions SQL més importants d'Oracle implementades per PostgreSQL.

Podeu descarregar el paquet de <http://orafce.projects.postgresql.org/>



4.- ANNEX I: Compilar SISTRA des de subversion de sourceforge

Aquest manual explica com compilar l'aplicació SISTRA a partir del repositori de subversion de sourceforge. Es requereix java i ant. La compilació es realitzarà en un directori sistra del home de l'usuari (~/sistra/).

- (1) Estant en el home de l'usuari fer check out executant la següent comanda:

```
$svn co https://sistra.svn.sourceforge.net/svnroot/sistra/sistra-core/trunk sistra
```

Podem baixar la branca principal (trunk) o una branca qualsevol (branch/INDRA-103-110)

- (2) Baixar el fitxer ojdbc14.jar de http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_10201.html i copiar-ho a ~/sistra/lib i dins /usr/local/jboss-3.2.8.SP1/server/default/lib/.
- (3) Editar fitxer ~/sistra/global.properties i seleccionar el SGBD que volem per hibernate: (Descomentar l'opció desitjada). Per postgresql quedaria com:

```
# Configuracion Hibernate
# ---- PostgreSQL
hibernate.dialect=net.sf.hibernate.dialect.PostgreSQLDialect
hibernate.query.substitutions=

# ----- Oracle
# hibernate.dialect=net.sf.hibernate.dialect.Oracle9Dialect
# hibernate.query.substitutions=true 1, false 0
```

- (4) Anar a ~/sistra/ i compilar executant la següent comanda:

```
$ ant
```

- (5) Anar a ~/sistra/pluginsMOCK i executar:

```
$ant
```



5.- ANNEX II: Script que automatitza la cridada als scripts sql

L'script següent només serveix per postgresql i presuposa que el Servidor de BBDD esta en la mateixa màquina (localhost).

IMPORTANT: Aquest script no processa els fitxers SQL de forms, per la qual cosa s'han d'executar a ma.

```
#!/bin/sh

export IP=localhost
echo _____ > sql.log
for BBDD in audita loginMock redose bantel mobtratel sistra form zonaper
do
    echo ===== $BBDD ===== >> sql.log
    echo > scriptdata.txt

    echo GRANT ALL PRIVILEGES ON DATABASE \"$BBDD\" TO \"$BBDD\"; >> scriptdata.txt
    echo GRANT ALL PRIVILEGES ON SCHEMA PUBLIC TO \"$BBDD\"; >> scriptdata.txt

    if [ -e ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.2/$BBDD-create.sql ]
    then
        echo \\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.2/$BBDD-create.sql >>
scriptdata.txt
    fi
    if [ -e ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.2/$BBDD-initialData.sql ]
    then
        echo \\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.2/$BBDD-
initialData.sql >> scriptdata.txt
    fi
    if [ -e ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.3/$BBDD-update.sql ]
    then
        echo \\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.3/$BBDD-update.sql >>
scriptdata.txt
    fi
    if [ -e ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.3/${BBDD}_update.sql ]
    then
        echo \\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.0.3/${BBDD}_update.sql
>> scriptdata.txt
    fi
    if [ -e ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.1.0/$BBDD-update.sql ]
    then
        echo \\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.1.0/$BBDD-update.sql >>
scriptdata.txt
    fi
    if [ -e ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.1.0/${BBDD}_update.sql ]
    then
        echo \\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.1.0/${BBDD}_update.sql
>> scriptdata.txt
    fi
    if [ -e ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.1.0/$BBDD-updateData.sql ]
    then
        echo \\i ~/sistra/doc/resources/bbdd/scripts/postgresql/version1.1.0/$BBDD-updateData.sql
>> scriptdata.txt
    fi
    echo \\q >> scriptdata.txt
    export PGPASSWORD=$BBDD

    echo ===== INICI CODI PER $BBDD ===== >> sql.log
```



```
cat scriptdata.txt >> sql.log
echo ----- FINAL CODI PER $BBDD ----- >> sql.log

echo ===== EXECUCIO PER $BBDD ===== >> sql.log
echo psql -h $IP -p 5432 -U $BBDD -d $BBDD >> sql.log
psql -h $IP -p 5432 -U $BBDD -d $BBDD < scriptdata.txt >> sql.log 2>> sql.log
echo ===== FINAL EXECUCIO PER $BBDD ===== >> sql.log

done

rm scriptdata.txt
```