Configuración Portal Bonita

Escriba el nombre del cliente

Escriba el nombre del Proyecto

Escriba el Nº del Proyecto

Versión 0.1 | Nivel de Seguridad N3 - Confidencial, contiene datos personales

Registro de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Revisado | Resumen de los cambios producidos |
| 1 | 20/10/2014 |  | borrador |
|  |  |  |  |
|  |  |  |  |

Versión de la Plantilla: 2.0

Índice

1 | Introducción 4

1.1 Objetivo 4

1.2 Abreviaturas y Acrónimos 4

2 | Definición de la Arquitectura del Sistema 5

2.1 Definición de la Arquitectura Software 5

2.1.1 Diagrama de componentes 5

2.2 Datos acceso a la maquina LINUX 5

3 | CONFIGURAR EL PORTAL. 6

3.1 IMPORTAR UNA ORGANIZACIÓN 6

3.1.1 Configurar usuarios y administradores del portal de bonita 7

3.1.2 Gestionar usuarios de la organización 8

3.2 Instalación de procesos en el portal de bonita 9

3.2.1 Deshabilitar la instanciación de procesos 10

3.3 Instalación de Procesos en Silcam 12

3.3.1 Instalación de diagramas BPMN en silcam 13

3.4 BUNDLE-BONITA 14

3.4.1 Configuración del servidor 14

3.4.1.1 Session 14

3.4.1.2 BBDD 14

3.4.1.3 HTTPS 14

3.4.1.4 CONFIGURACION ENTORNO 14

3.4.1.5 En windows 14

3.4.2 Modificaciones Bundle para SILCAM. 15

3.4.2.1 Actualizar el servicio de autentificación de bonita 15

3.4.2.2 Actualizar los servicios de acceso de BBDD de hibernate de bonita 16

3.4.3 Librerias servidor bundle portal de bonita 16

3.4.3.1 Actualizacion de librerias. 18

3.4.3.2 La conexión con la BBDD del motor de bonita 18

3.4.3.3 Las llamadas Rest: 19

3.4.4 Conexión con terceros: 20

3.5 Integrar Licenciamiento con Bonita. 21

3.6 CAS-SATEC 22

3.7 CORS 23

# Introducción

## Objetivo

El objetivo de este documento es explicar los pasos necesarios para montar el portal de bonita de modo que esté integrado con la aplicación SILCAM. Es decir desde el proyecto de licenciamiento (catastro-web) se pueda consultar información del motor de bonita y mostrar formularios de los procesos dados de alta en el portal y registrados por licenciamiento. Así mismo desde dichos procesos se pueda acceder a proyectos, BBDD de Silcam.

## Abreviaturas y Acrónimos

*Aquí, se definirán todas las abreviaturas y acrónimos utilizados en el documento.*

# Definición de la Arquitectura del Sistema

## Definición de la Arquitectura Software

*En esta actividad se define la arquitectura general del sistema de información, especificando la descomposición lógica en subsistemas de diseño y la ubicación de cada uno de los mismos.*

### Diagrama de componentes

*Un diagrama UML que representará cómo el sistema de* [*software*](http://es.wikipedia.org/wiki/Software) *es dividido en* [*componentes*](http://es.wikipedia.org/wiki/Componentes_de_software) *mostrando las* [*dependencias*](http://es.wikipedia.org/w/index.php?title=Dependencia_(UML)&action=edit&redlink=1) *entre éstos. Los componentes físicos incluyen* [*archivos*](http://es.wikipedia.org/wiki/Archivo_(computaci%C3%B3n))*, cabeceras,* [*librerías compartidas*](http://es.wikipedia.org/wiki/Biblioteca_(programaci%C3%B3n))*,* [*módulos*](http://es.wikipedia.org/wiki/M%C3%B3dulo_(programaci%C3%B3n))*,* [*ejecutables*](http://es.wikipedia.org/wiki/Ejecutable)*, o* [*paquetes*](http://es.wikipedia.org/wiki/Paquete_de_software)*. Los diagramas de Componentes prevalecen en el campo de la* [*arquitectura de software*](http://es.wikipedia.org/wiki/Arquitectura_de_software) *pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.*

## Datos acceso a la maquina LINUX

* **DESARROLLO:**

**DNS**: silcam.igeo.satec.es

**User/Pass:** root / satec2015

[http://silcam.silcam.igeo.satec.es:8090/bonita](http://demo.silcam.igeo.satec.es:8090/bonita)

* **DEMO (DESARROLLO)**

**DNS**: demo.silcam.igeo.satec.es

**User/Pass:** root/satec2015

[http://demo.silcam.silcam.igeo.satec.es:8090/bonita](http://demo.silcam.igeo.satec.es:8090/bonita)

* **PRE**

**DNS**: bonita-silcam.pre.mgmt.igeo.gov.ao

**User/Pass:** root/Satec2014

[http://bonita-silcam.pre.igeo.gov.ao:8090/bonita](http://demo.silcam.igeo.satec.es:8090/bonita)

* **PRO**

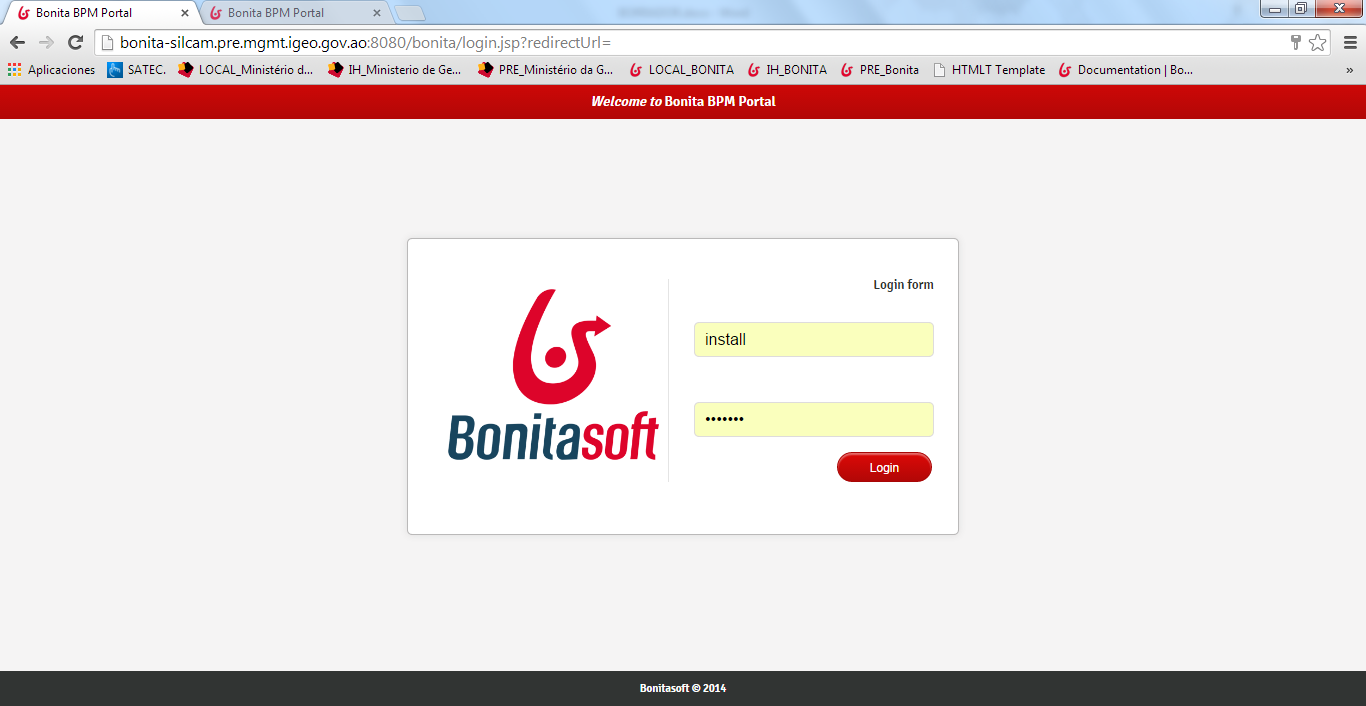
**DNS**: bonita-silcam.pro.mgmt.igeo.gov.ao

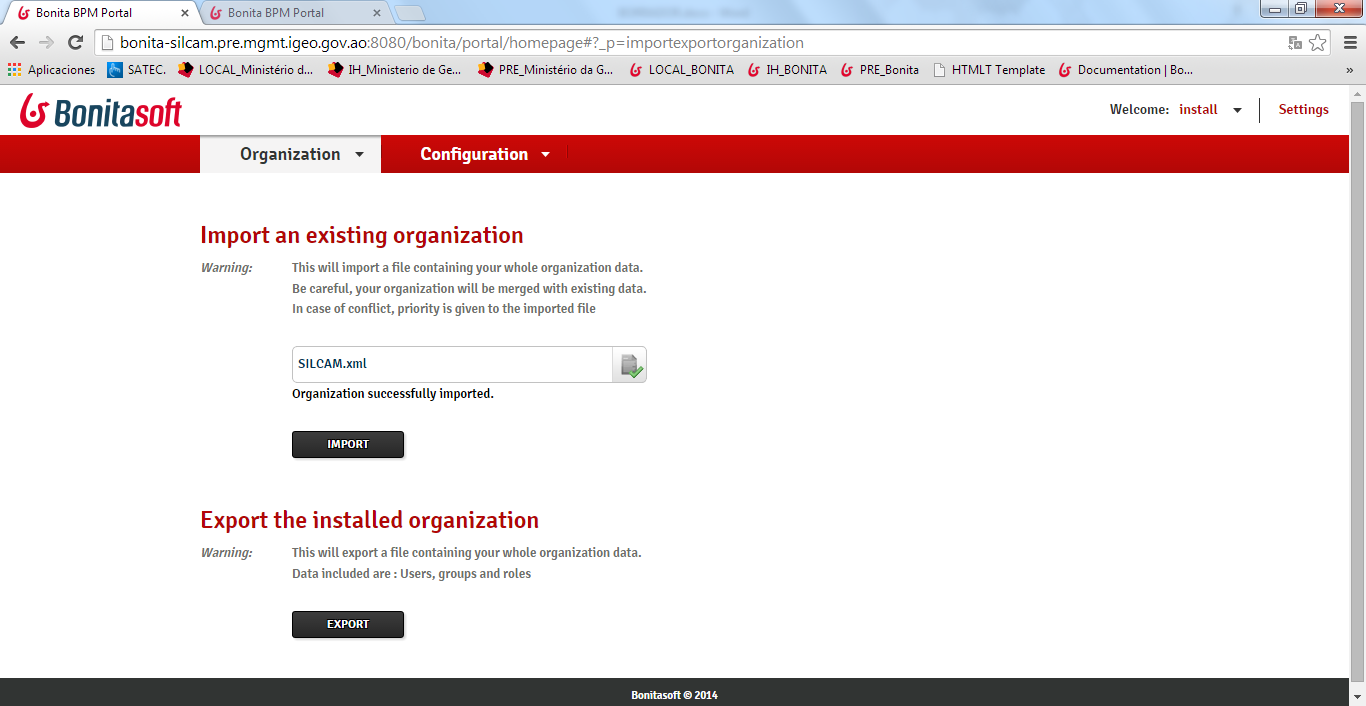
**User/Pass:** root/Satec2014

[http://bonita-silcam.pro.igeo.gov.ao:8090/bonita](http://demo.silcam.igeo.satec.es:8090/bonita)

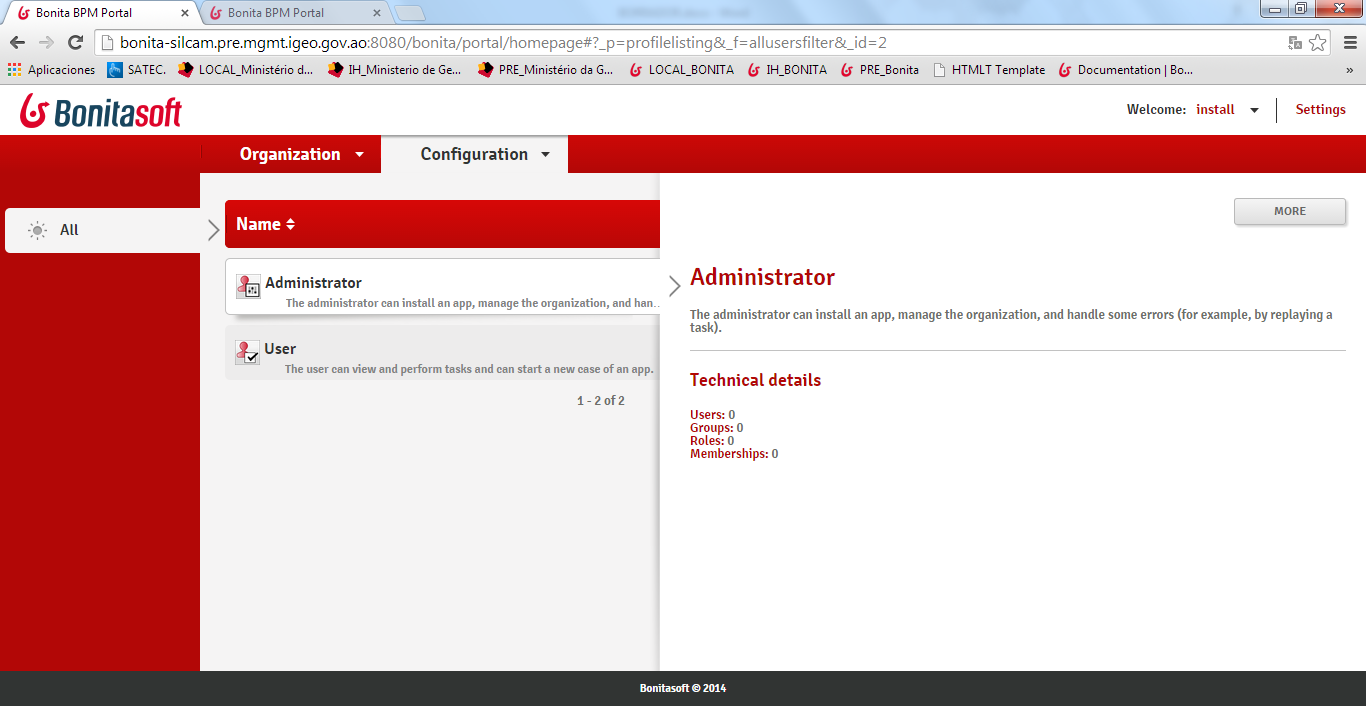
# CONFIGURAR EL PORTAL.

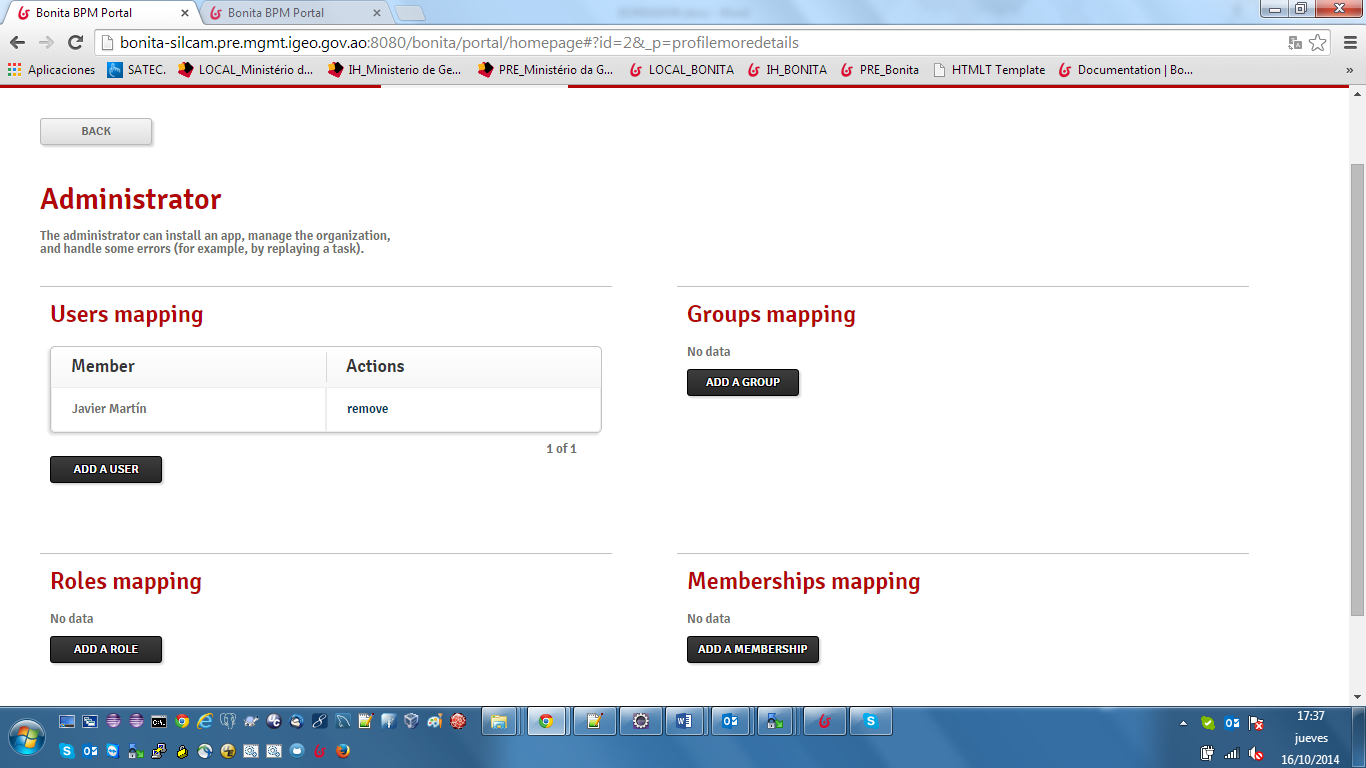
## IMPORTAR UNA ORGANIZACIÓN

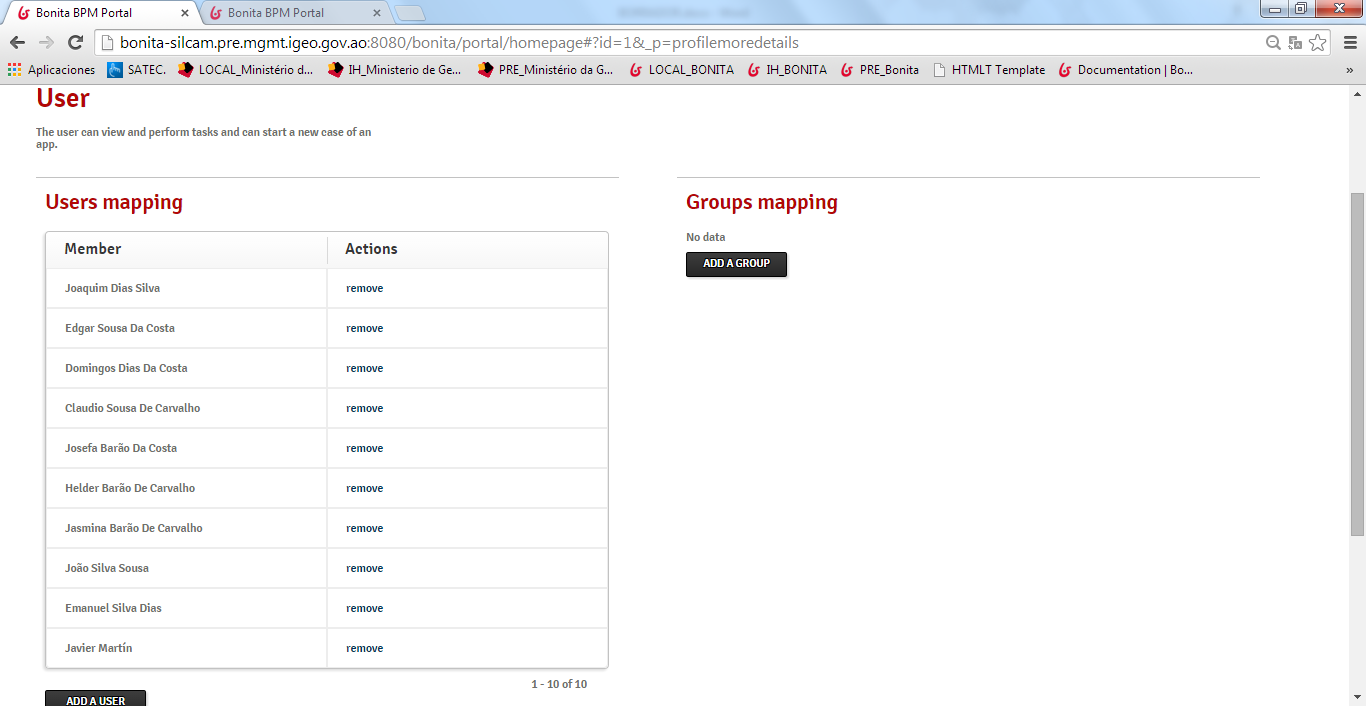




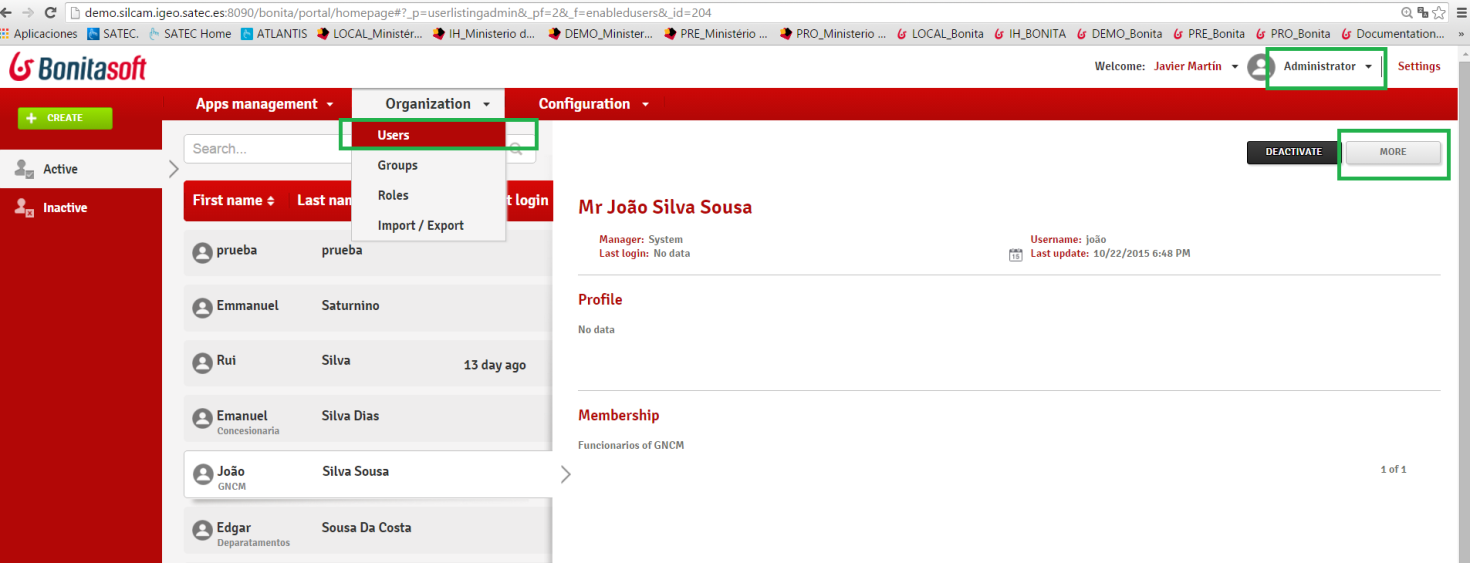
### Configurar usuarios y administradores del portal de bonita





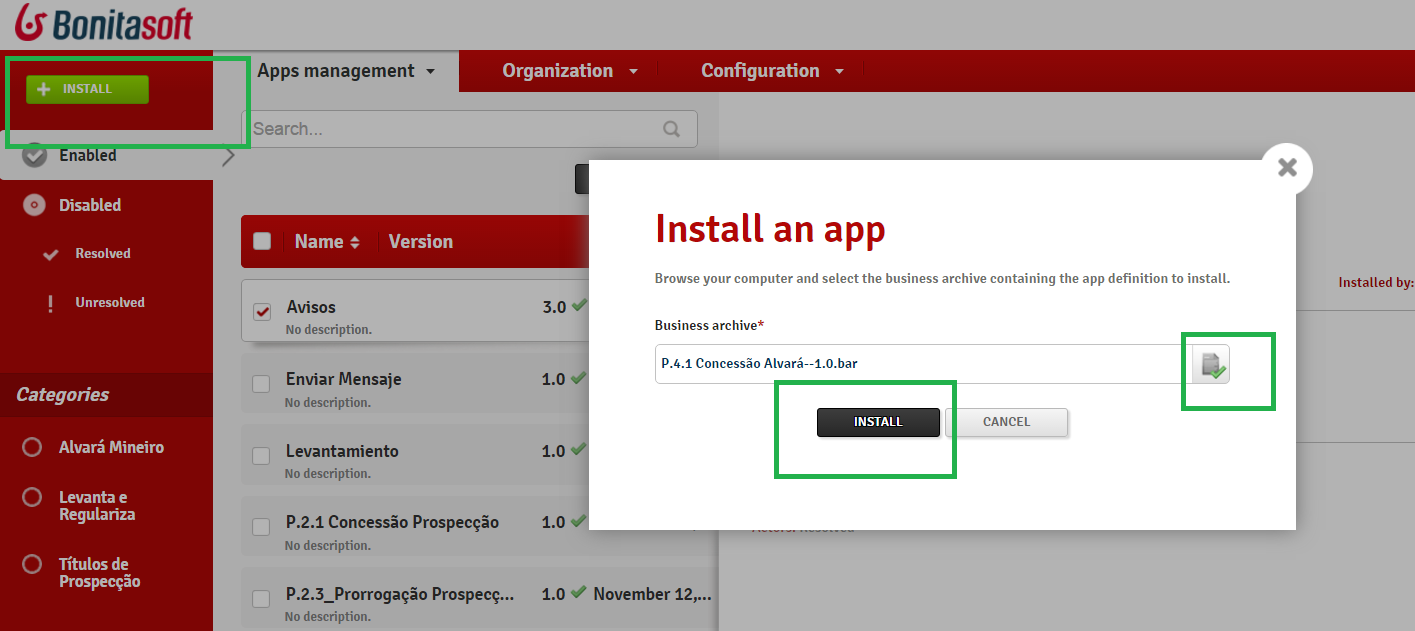


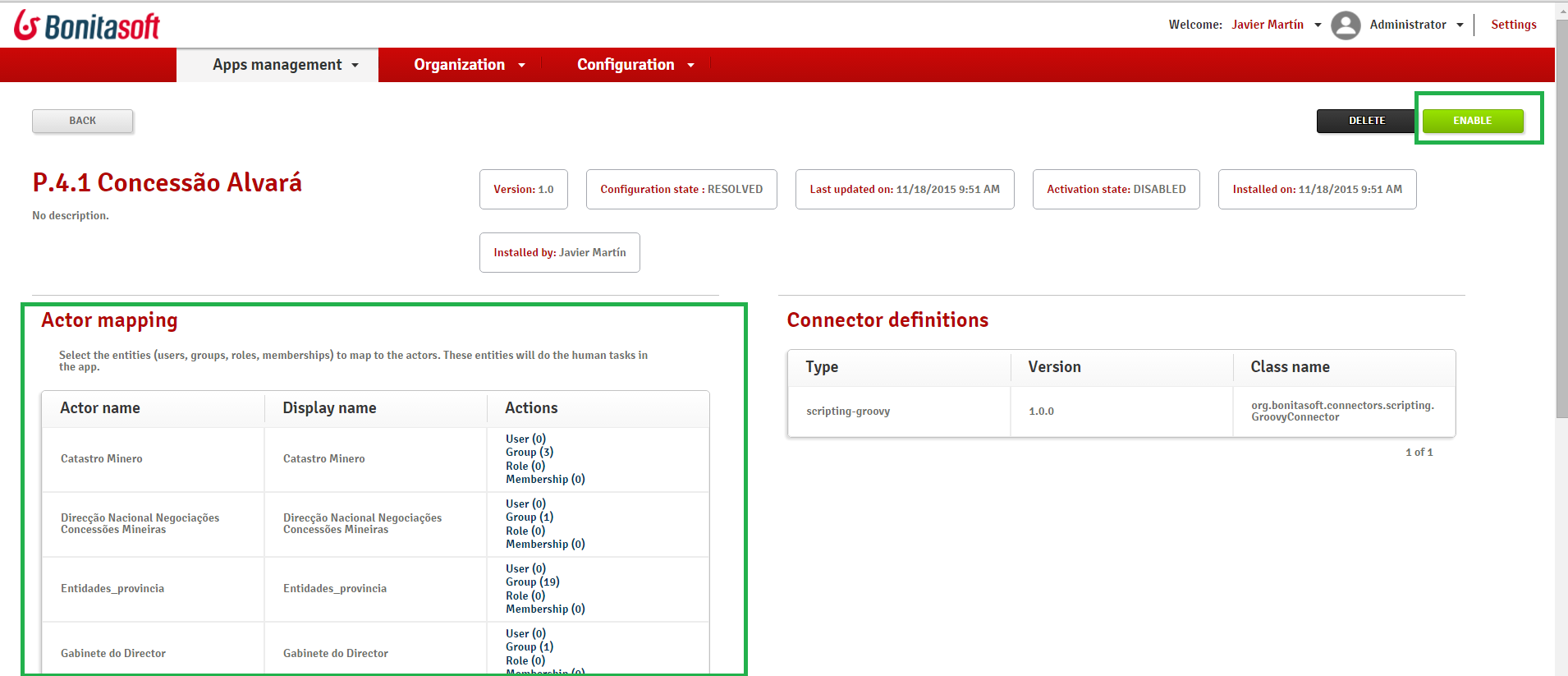
### Gestionar usuarios de la organización



### 

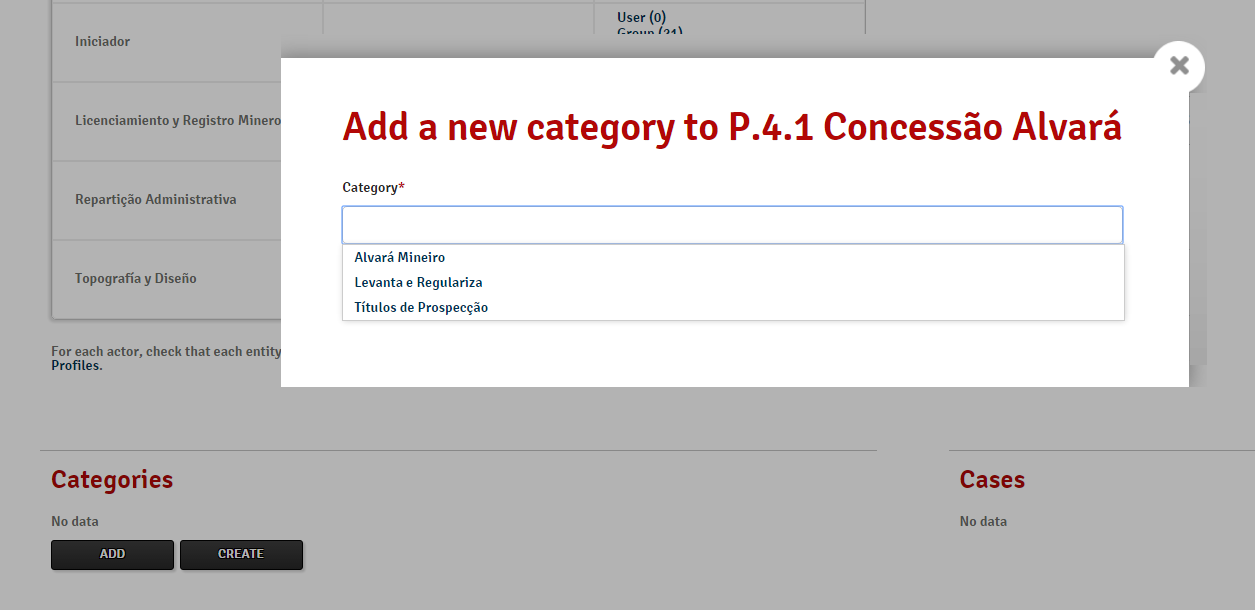
## Instalación de procesos en el portal de bonita





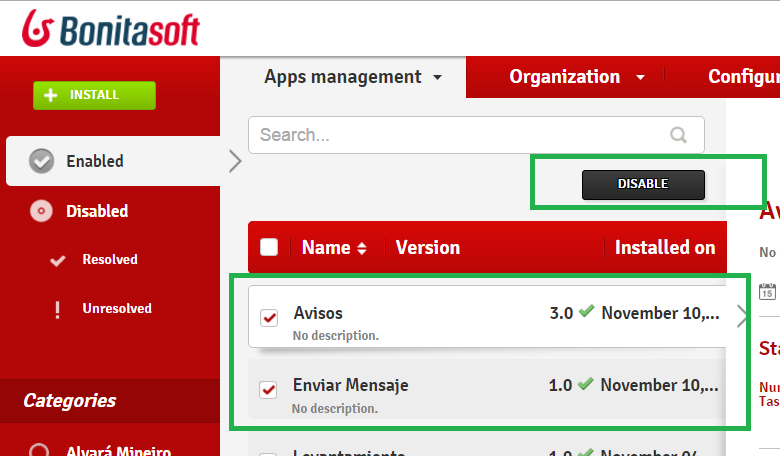
SeS

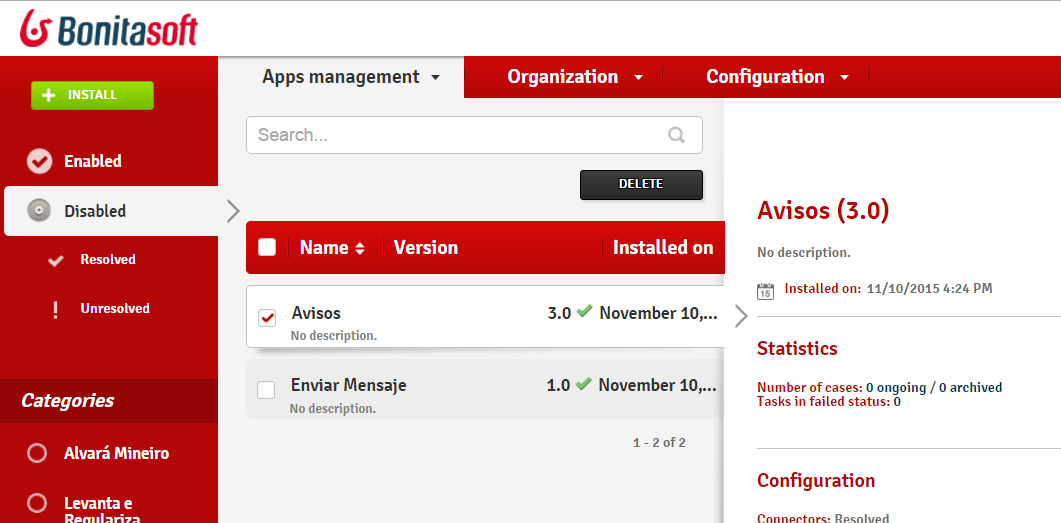
**Se deben mapear todos los actores para poder activar el proceso**



Añadir la categoría a la que pertenece el proceso.

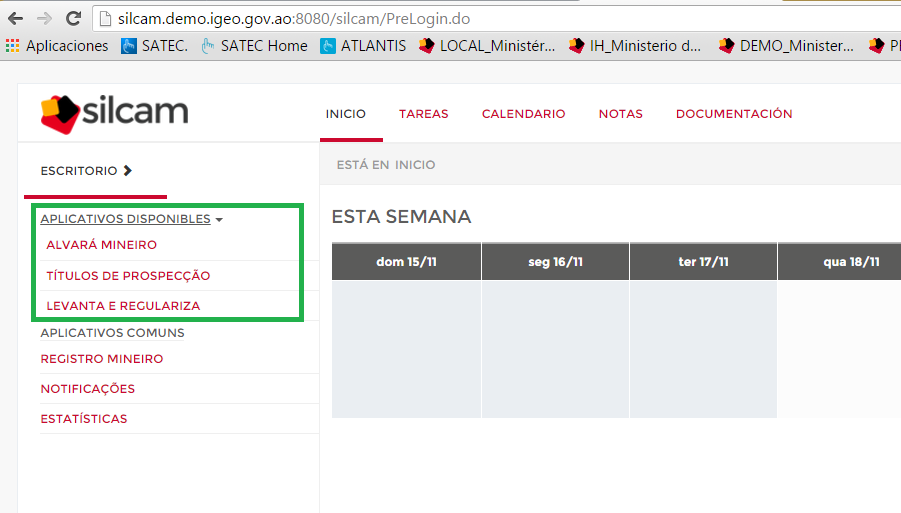
### Deshabilitar la instanciación de procesos

Con esto se consigue que se puedan seguir trabajando con los expedientes ya instanciados pero no se permite crear ninguna instancia nueva.



## Instalación de Procesos en Silcam

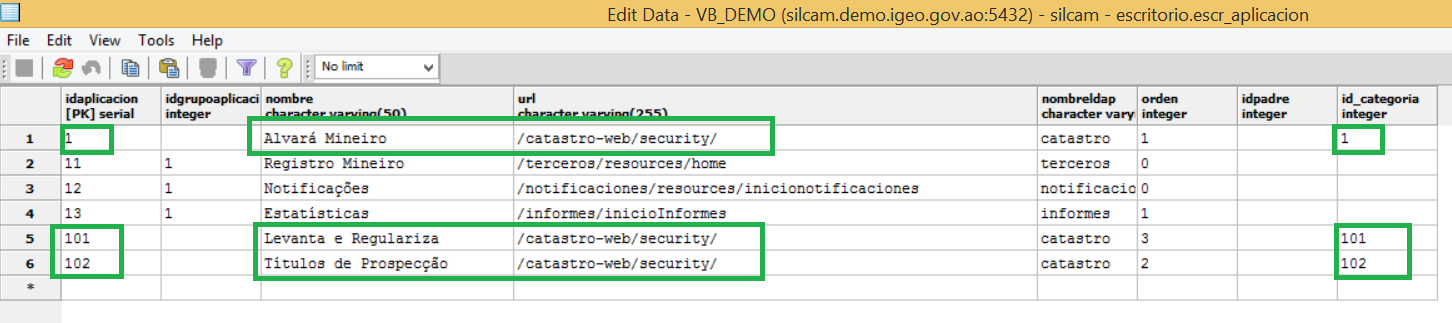
Para que los procesos estén accesibles desde SILCAM y se pueda trabajar con ellos correctamente, hay que tener añadir unos registros en bbdd



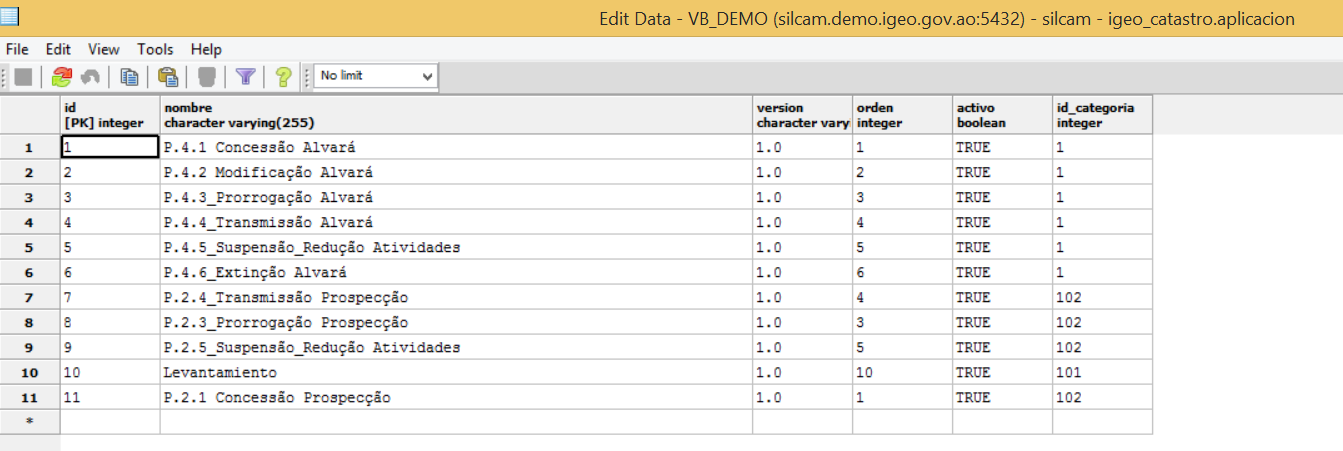


en bbdd:

- Para mostrar el menu de aplicaciones disponibles en el escritorio de SILCAM, debe haber entradas similares a estas en la tabla escritorio.escr\_aplicacion debe haber una entrada



- Para cada categoría debe aparecer una entrada como la siguiente en la tabla igeo\_catastro.aplicacion **(deben coincidir los id´s de categoría**):

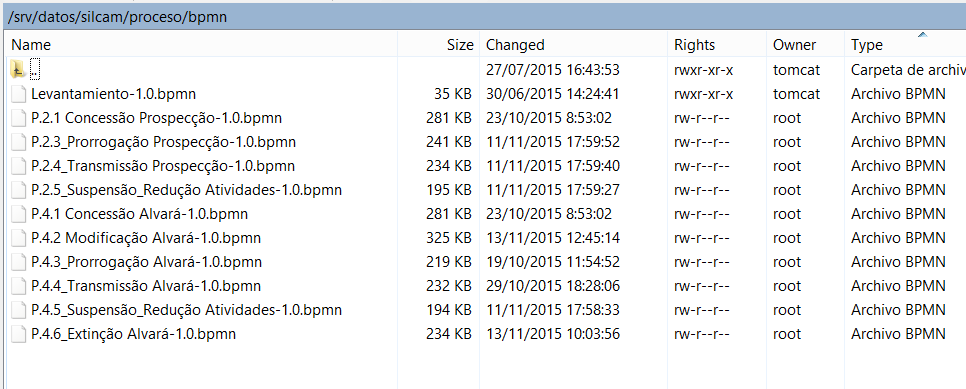


**también en esta tabla si queremos que el proceso deje de aparecer en SILCAM, basta con que en la columna activo pongamos el valor false.**

- La relación de los procesos con la documentación de alfresco se guarda en el schema igeo\_doc

### Instalación de diagramas BPMN en silcam

Esta acción por ahora se debe de hacer de modo manual, copiando directamente los bpmn´s generados por el bonita studio en la máquina linux que aloja a SILCAM, en la ruta: /srv/datos/silcam/proceso/bpmn



**El BPMN se debe llamar igual que el proceso y debe contener en su nombre la versión y release del proceso:**

**[nombre proceso]-[Version].[Release].bpmn**

## BUNDLE-BONITA

### Configuración del servidor

#### Session del servidor de bonita

- /BonitaBPMCommunity-6.3.0-Tomcat-6.0.37/conf/web.xml

<!-- created sessions by modifying the value below. -->

<session-config>

<session-timeout>60</session-timeout>

</session-config>

#### BBDD

- /BonitaBPMCommunity-6.3.0-Tomcat-6.0.37/conf/bitronix-resources.properties

- /BonitaBPMCommunity-6.3.0-Tomcat-6.0.37/conf/Catalina/localhost/Catalina/localhost/bonita.xml

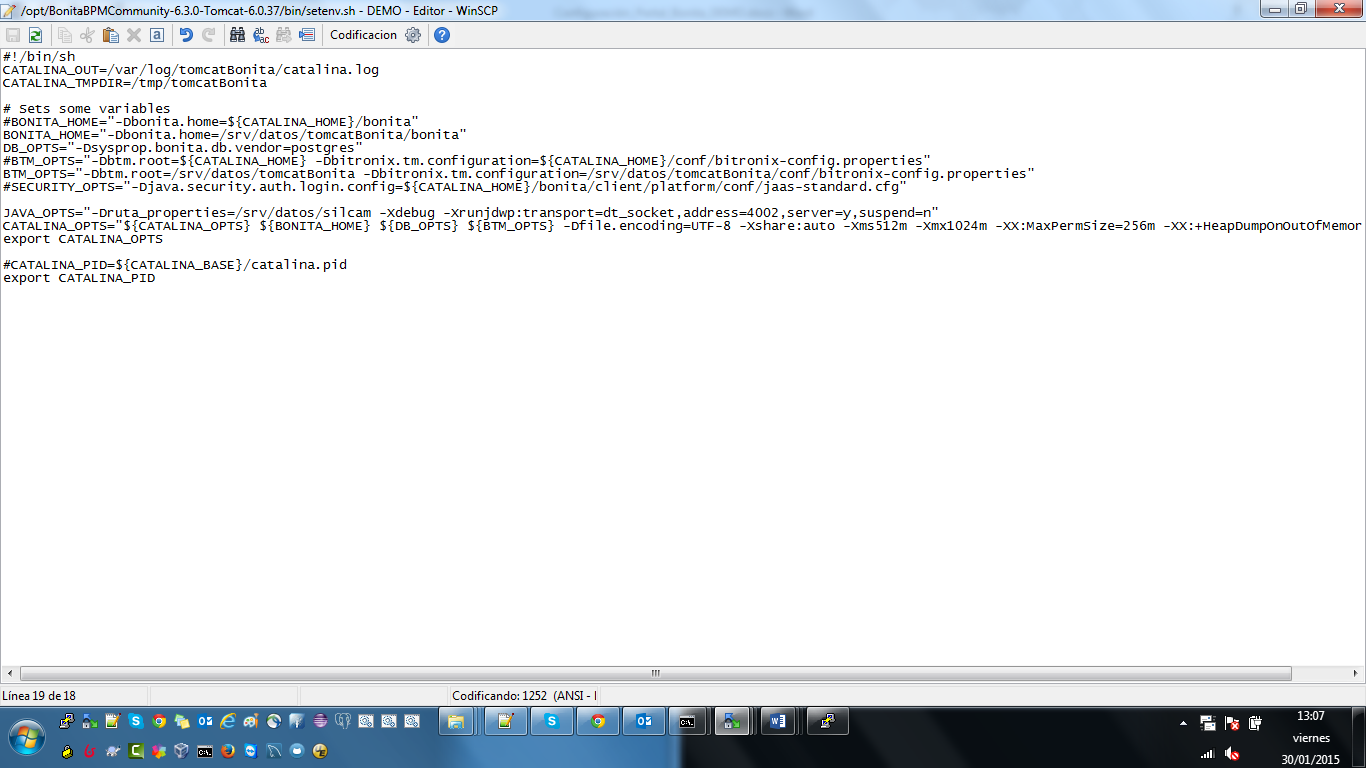
- /BonitaBPMCommunity-6.3.0-Tomcat-6.0.37\lib\bonita\postgresql-9.3-1101.jdbc41.jar

#### HTTPS

/BonitaBPMCommunity-6.3.0-Tomcat-6.0.37/conf/Server.xml --> configurar puertos de seguridad 8443

#### CONFIGURACION ENTORNO

En el setenv.sh , añadir a JAVA\_OPTS la variable ruta\_properties con la ruta al directorio donde está el igeo.properties.



JAVA\_OPTS="-Druta\_properties=/srv/datos/silcam -Xdebug -Xrunjdwp:transport=dt\_socket,address=4002,server=y,suspend=n"

#### En windows

BonitaBPMCommunity-6.3.0-Tomcat-6.0.37\bin\setenv.bat

BonitaBPMCommunity-6.3.0-Tomcat-6.0.37\bin\catalina.bat

### Modificaciones Bundle para SILCAM.

#### Actualizar el servicio de autentificación de bonita

* Modificar el cfg-bonita-authentication-impl.xml para que apunte a la implementación del plugin:

/srv/datos/tomcatBonita/bonita/server/tenants/1/conf/services

<bean id="authenticationService" class="es.satec.org.bonitasoft.engine.authentication.impl.SatecAuthenticationServiceImpl">

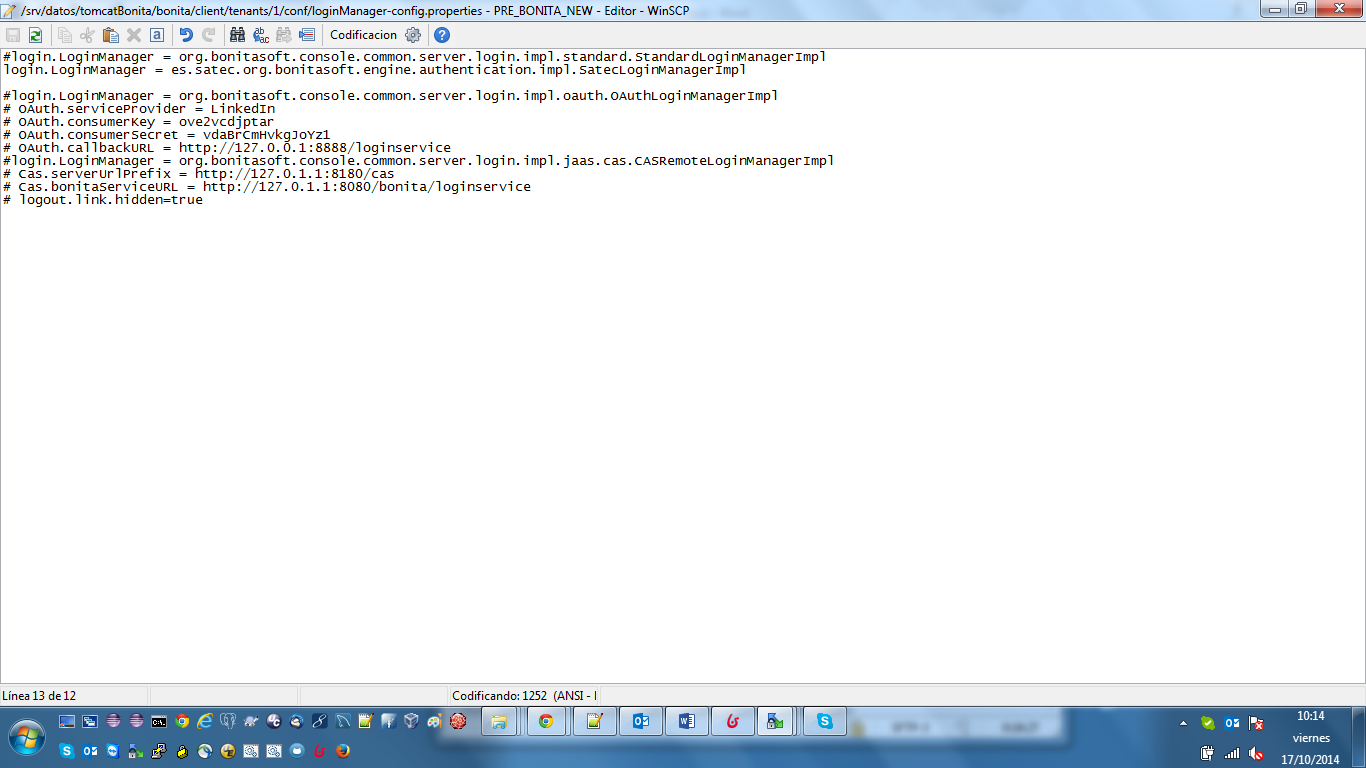
<constructor-arg name="logger" ref="tenantTechnicalLoggerService" />

</bean>

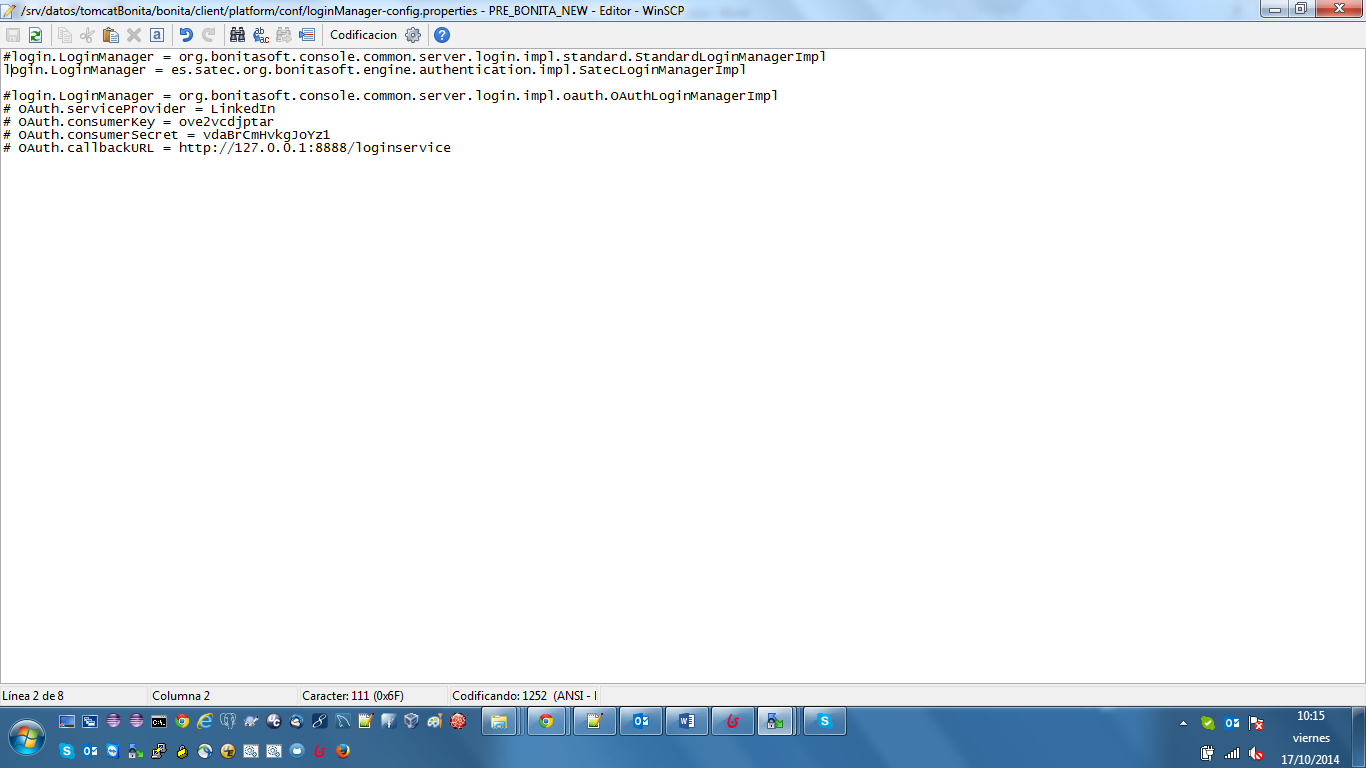


* Modificar los ficheros loginManager-config.properties para coger el loginManager del plugin, en tres lugares:

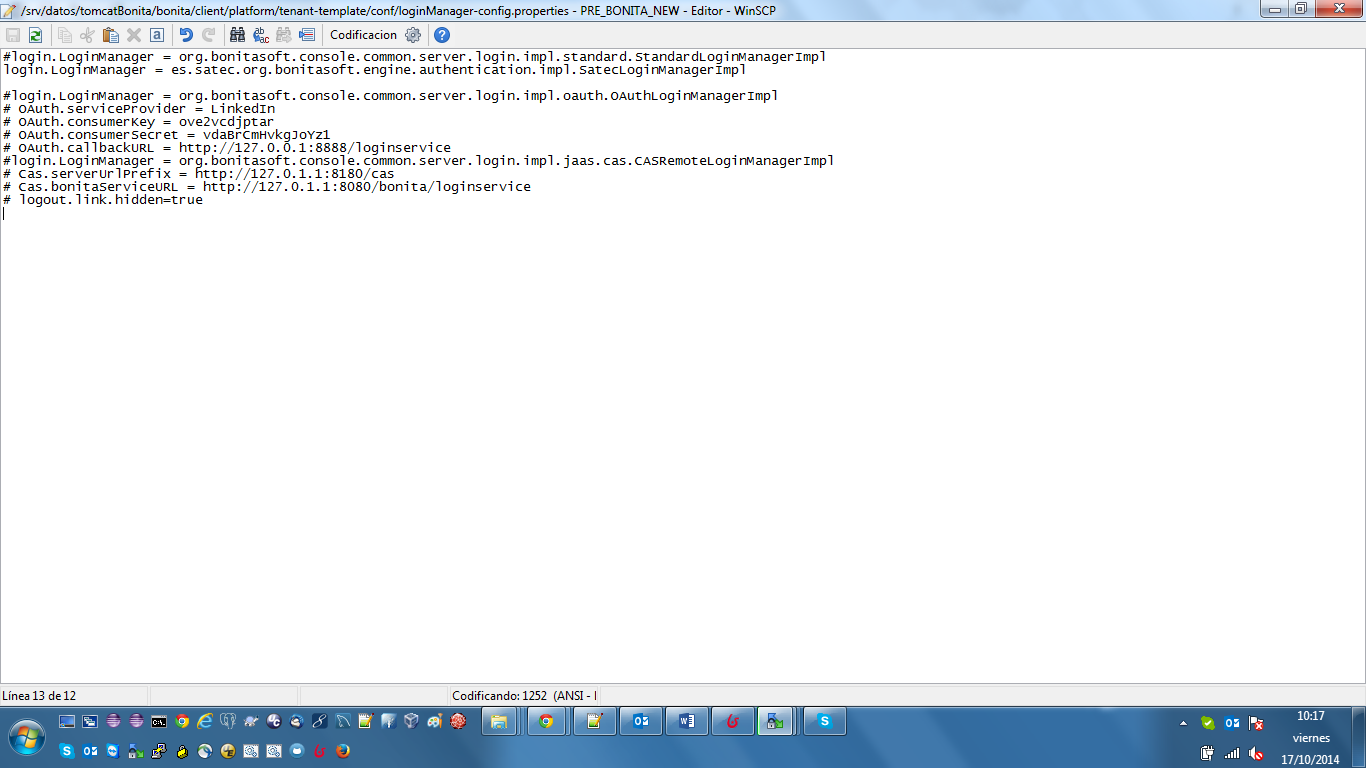
/srv/datos/tomcatBonita/bonita/client/tenants/1/conf



/srv/datos/tomcatBonita/bonita/client/platform/conf

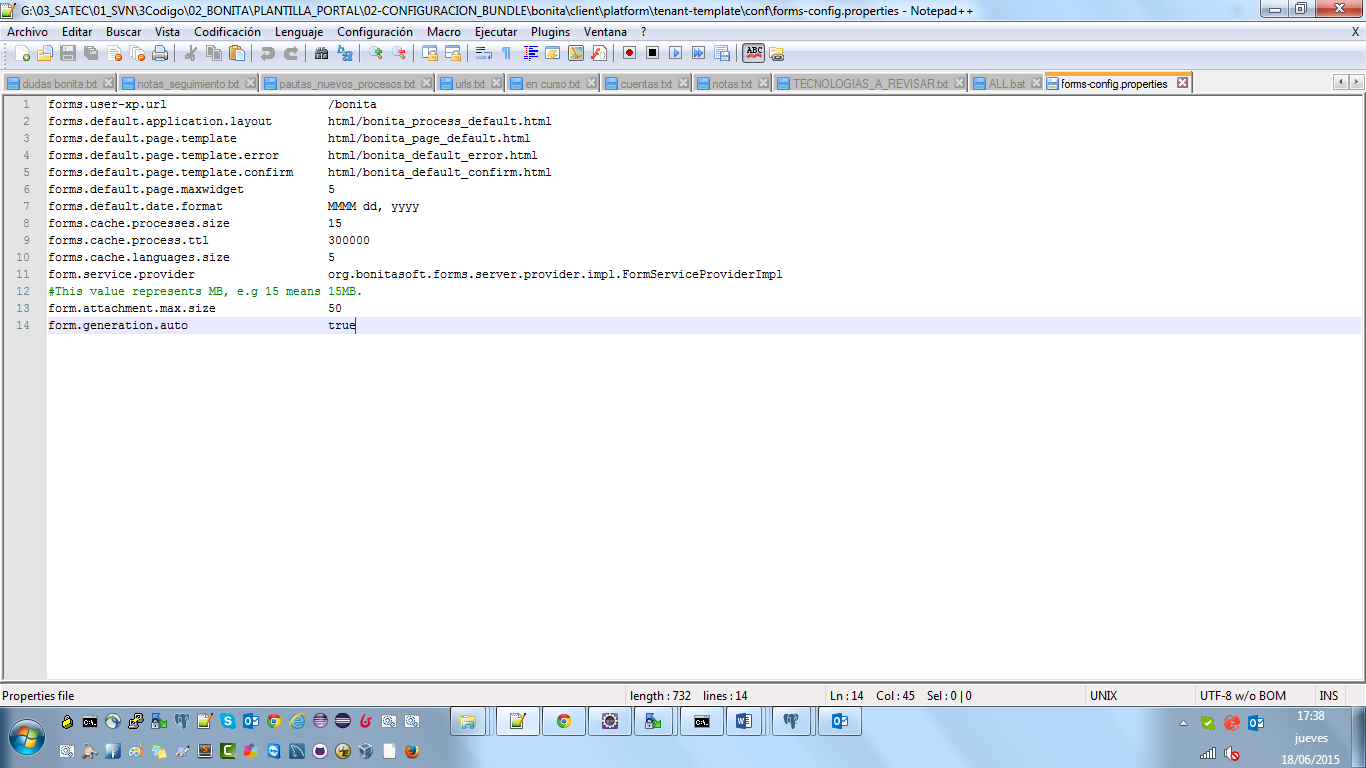


/srv/datos/tomcatBonita/bonita/client/platform/tenant-template/conf



#login.LoginManager = org.bonitasoft.console.common.server.login.impl.standard.StandardLoginManagerImpl

login.LoginManager = es.satec.org.bonitasoft.engine.authentication.impl.SatecLoginManagerImpl

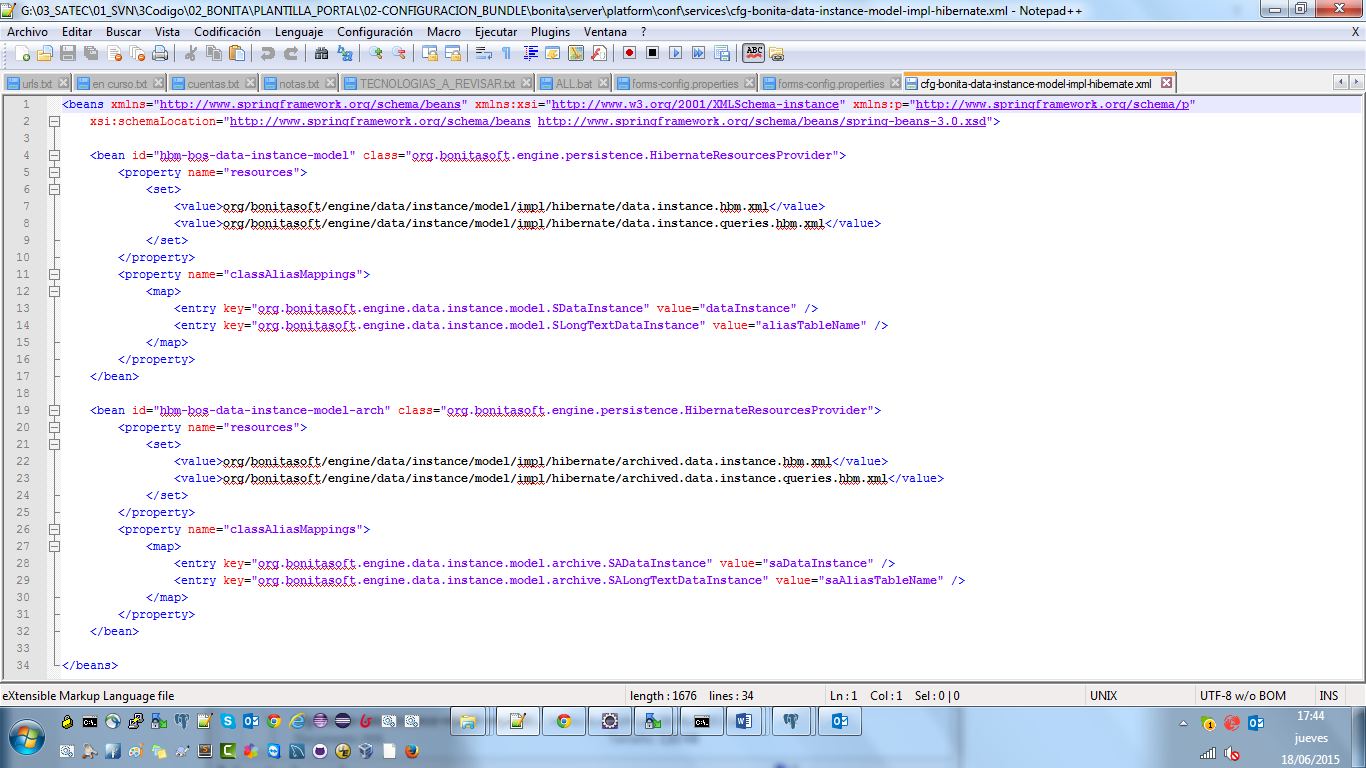


form.attachment.max.size 50

#### Actualizar los servicios de acceso de BBDD de hibernate de bonita

* Modificar cfg-bonita-data-instance-model-impl-hibernate.xml para añadir alias:

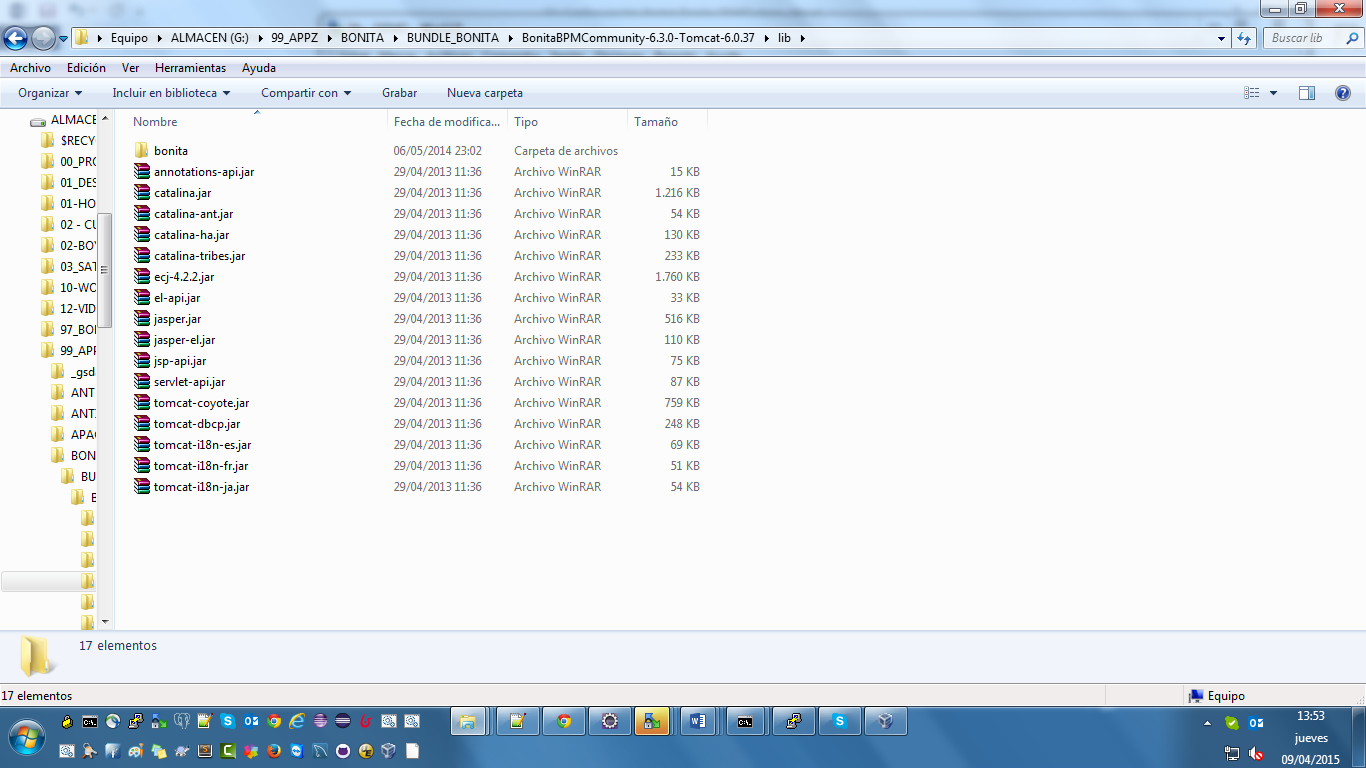
/srv/datos/tomcatBonita/bonita/server/platform/conf/services/cfg-bonita-data-instance-model-impl-hibernate.xml

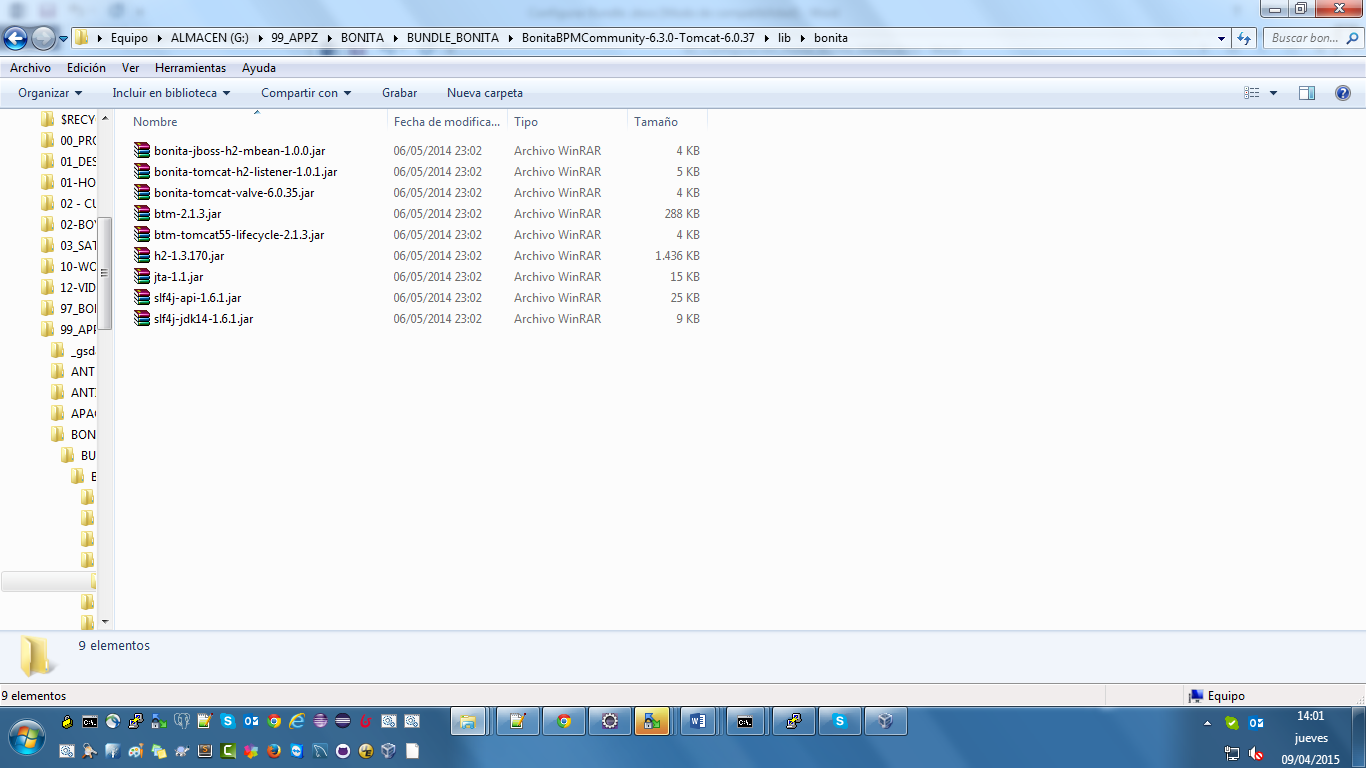


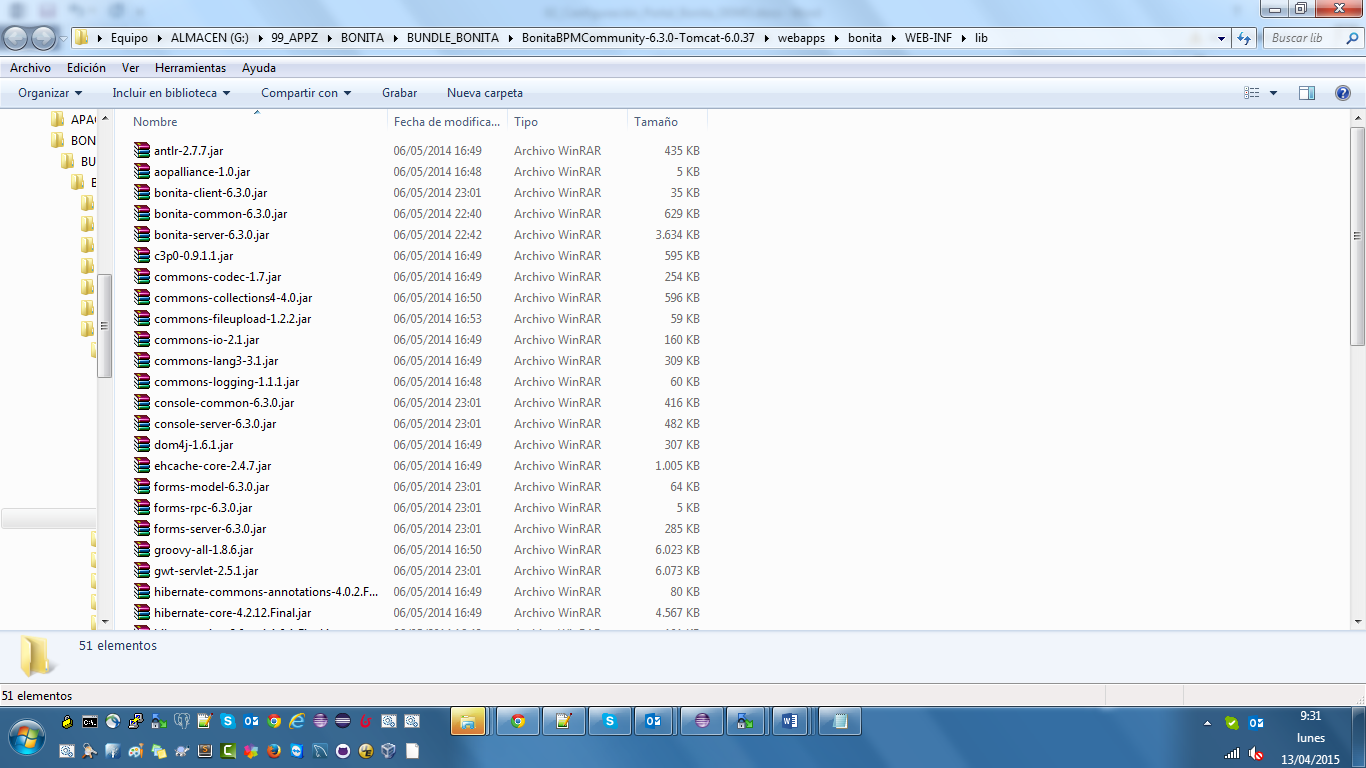
### Librerias servidor bundle portal de bonita

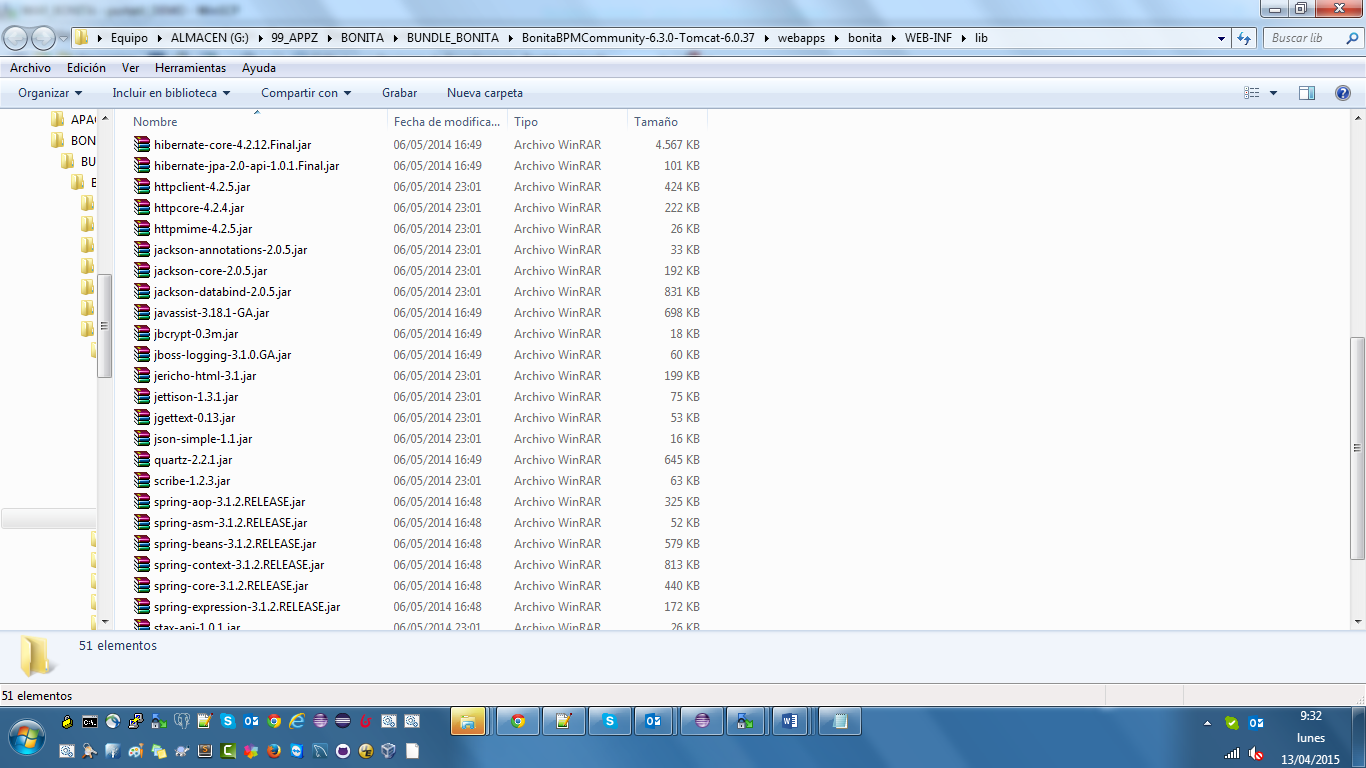
A continuación indico las actualizaciones de librerías que se deben llevar a cabo en el bundle del portal de bonita, para poder acceder a los servicios desarrollados por Satec.

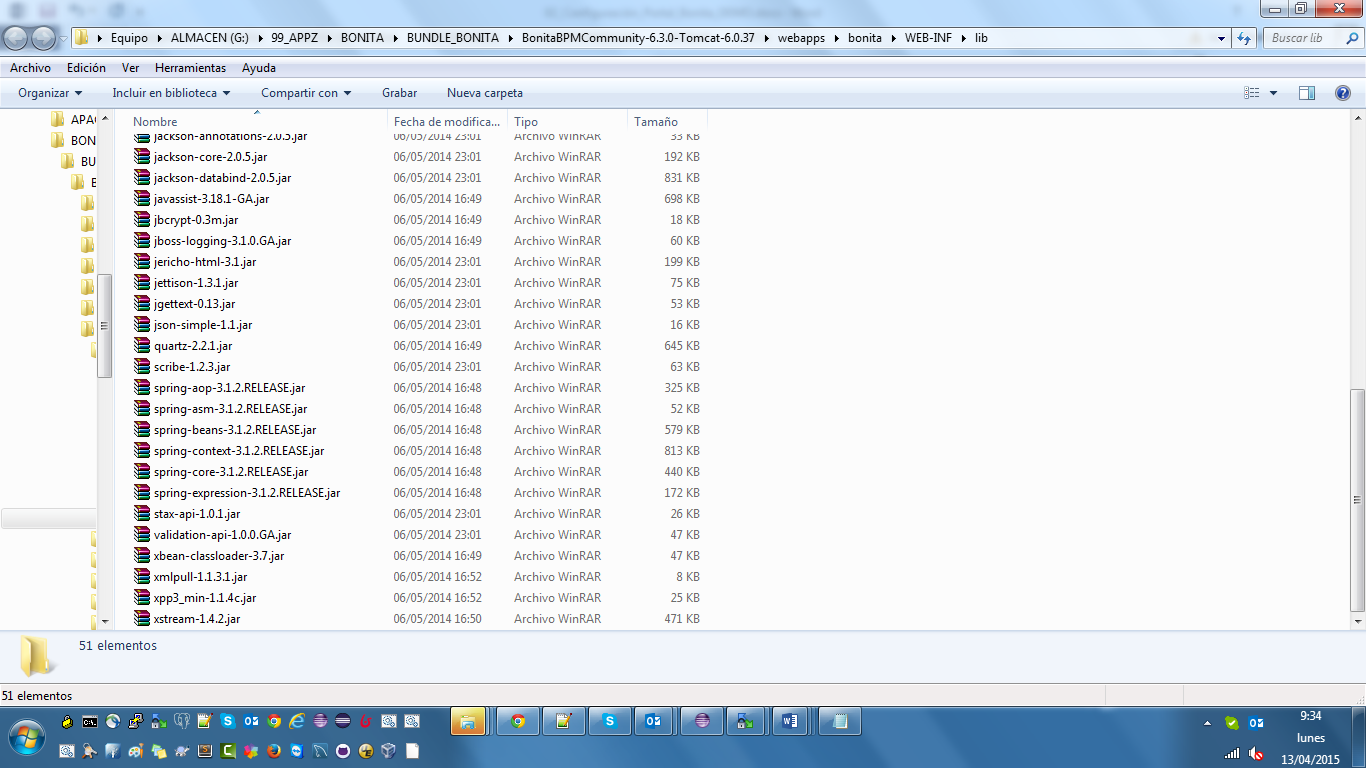
Partimos de una instalación limpia con las siguientes librerías:









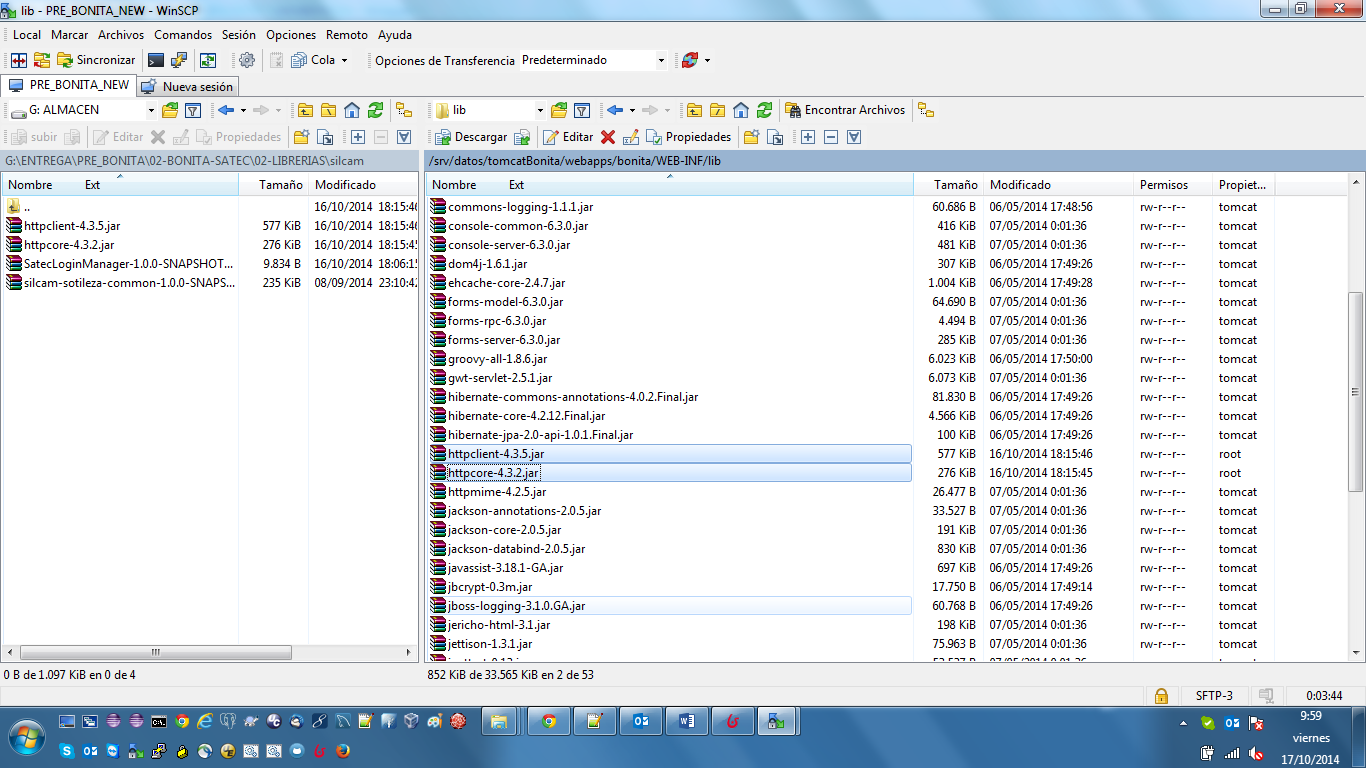
  
  
En primer lugar comentar que hay que realizar una serie de cambios necesarios en el servidor de bonita para su uso:

#### Actualizacion de librerias.

Se deben actualizar ciertas librerías que vienen por defecto en el bundle de bonita , en concreto:

* 1. en /srv/datos/tomcatBonita/webapps/bonita/WEB-INF/lib

1. Httpclient-4.2.5 por Httpclient-4.3.5
2. Httpcore-4.2.4 por Httpcore-4.3.2



**Estas librerías se usan para la conexión con CAS**

* 1. Hay que añadir las siguientes librerías

1. core-security-lib.jar
2. core-tipos-datos-lib.jar
3. spring-security-core-3.1.2.RELEASE.jar
4. bonita-server-6.3.0.jar
5. bonita-client-6.3.0.jar
6. bonita-common-6.3.0.jar

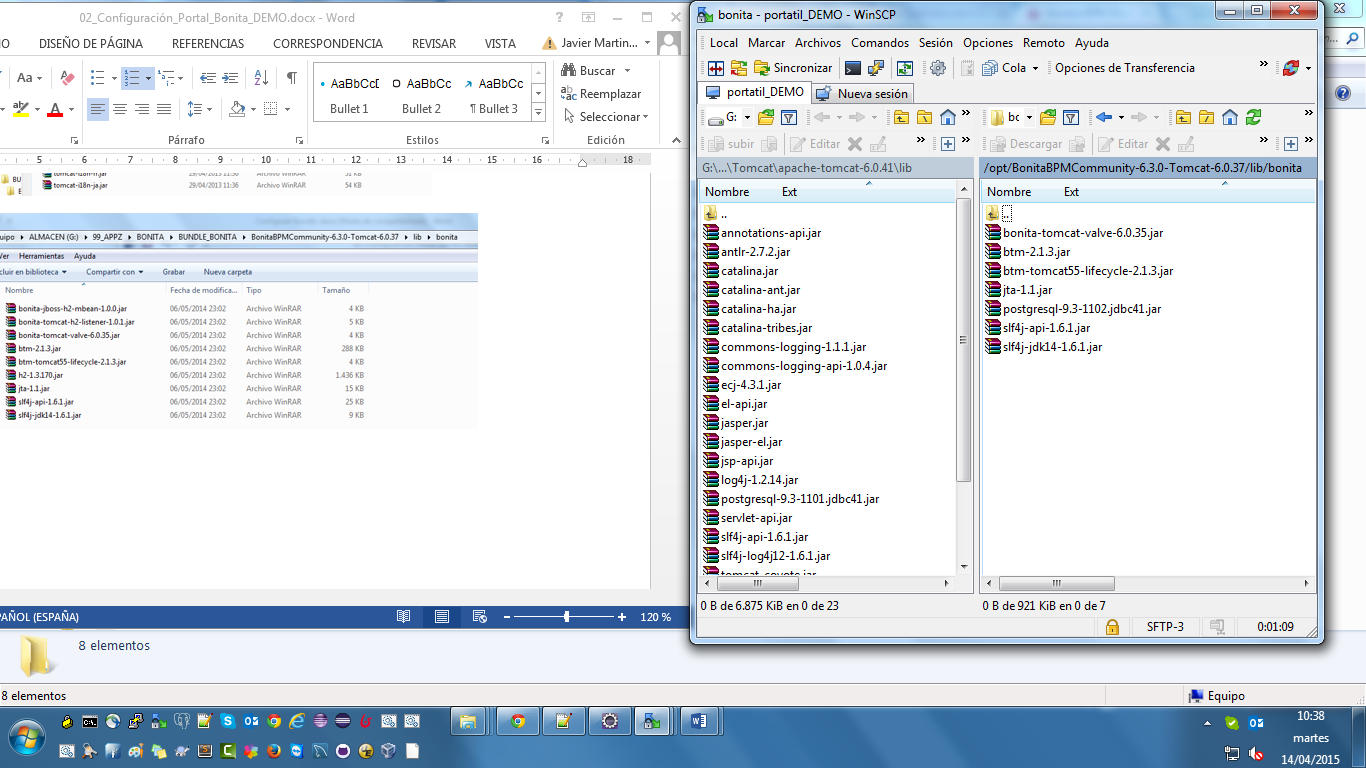
**Estas librerías han sido creadas para hacer compatible Bonita con CAS y permitir búsquedas de expedientes mas rápidas y especificas.**

#### La conexión con la BBDD del motor de bonita

[ruta\_instalacion]\BonitaBPMCommunity-6.3.0-Tomcat-6.0.37\lib\bonita

Eliminamos los jar de conexión a BBDD que no usamos y añadimos el de postgresql:

postgresql-9.3-1101.jdbc41.jar



#### Las llamadas Rest:

Para hacer uso de los métodos groovy en las llamadas Rest hay que añadir en “**BonitaBPMCommunity-6.3.0-Tomcat-6.0.37\webapps\bonita\WEB-INF\lib”** los siguientes (\*.jar):

http-builder-0.7.jar

y sus dependencias:

appengine-api-1.0-sdk-1.3.8.jar

commons-beanutils-1.8.0.jar

~~commons-codec-1.6.jar~~

commons-codec-1.7.jar 🡪 Es más alto el que viene con bonita

commons-collections-3.2.1.jar (puede convivir con los superiores, por ahora hemos dejado el de bonita)

commons-collections4-4.0.jar 🡪 Es más alto el que viene con bonita

commons-lang-2.4.jar 🡪 si no, no funciona

commons-lang3-3.1.jar

commons-logging-1.1.1.jar

ezmorph-1.0.6.jar

~~httpclient-4.2.1.jar~~

httpclient-4.3.5.jar

~~httpcore-4.2.1.jar~~

httpcore-4.3.2.jar

json-lib-2.3-jdk15.jar

nekohtml-1.9.16.jar

signpost-commonshttp4-1.2.1.1.jar

signpost-core-1.2.1.1.jar

xercesImpl-2.9.1.jar

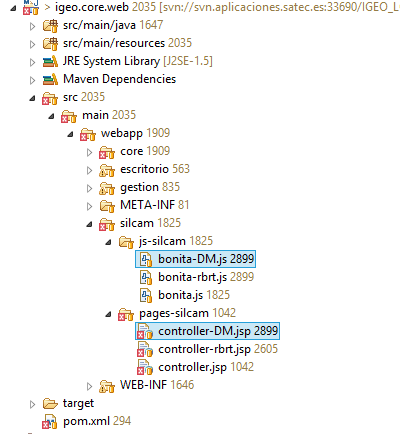
xml-apis-1.4.01.jar

xml-resolver-1.2.jar

### Conexión con terceros:

A continuación pongo un resumen de lo que hay hacer para hacer funcionar terceros en bonita con lo que se ha desarrollado. Ahora misma se consigue llamar a terceros de bonita y devolver el tercero seleccionado sin resetear el formulario de bonita y utilizando el ticket cas para que permita realizar las llamadas entre aplicaciones de forma segura.  
  
En primer lugar comentar que hay que realizar una serie de cambios necesarios en el servidor de bonita para su uso:

**1 -** Copiar dentro de pages-silcam el fichero **controller-DM.jsp** y copiar dentro de js-silcam el fichero **bonita-DM.js**. Están en el svn en el proyecto igeo.core.web aunque también podéis cogerlos de ih.



## Integrar Licenciamiento con Bonita.

Bonita tiene configurado como medida de seguridad que los iframes no se puedan ejecutar desde fuera de su servidor, lo cual se puede esquivar modificando en el web.xml de bonita el SecurityFilter. Este comportamiento se debe a implementaciones de seguridad de bonita y al modificarlo desactiva las cabeceras:

**X-Frame-Options** está configurado para SAMEORIGIN con el fin de evitar los ataques clickjaking.

**X-XSS-Protection** permite a una tercera parte inyectar en páginas web vistas por el usuario código JavaScript o en otro lenguaje script similar, evitando medidas de control como la Política del mismo origen.

**X-WebKit-CSP** evita cross-site scripting (XSS) y los ataques relacionados.

**X-Content-Type-Options** (Impedir la detección del tipo MIME. Ahora, las aplicaciones web pueden evitar la detección del tipo MIME. El envío del nuevo encabezado X-Content-Type-Options: nosniff impide que use la detección del tipo MIME para cambiar el contenido declarado por el servidor.).

Comentar en el /srv/datos/tomcatBonita/webapps/bonita/WEB-INF/web.xml

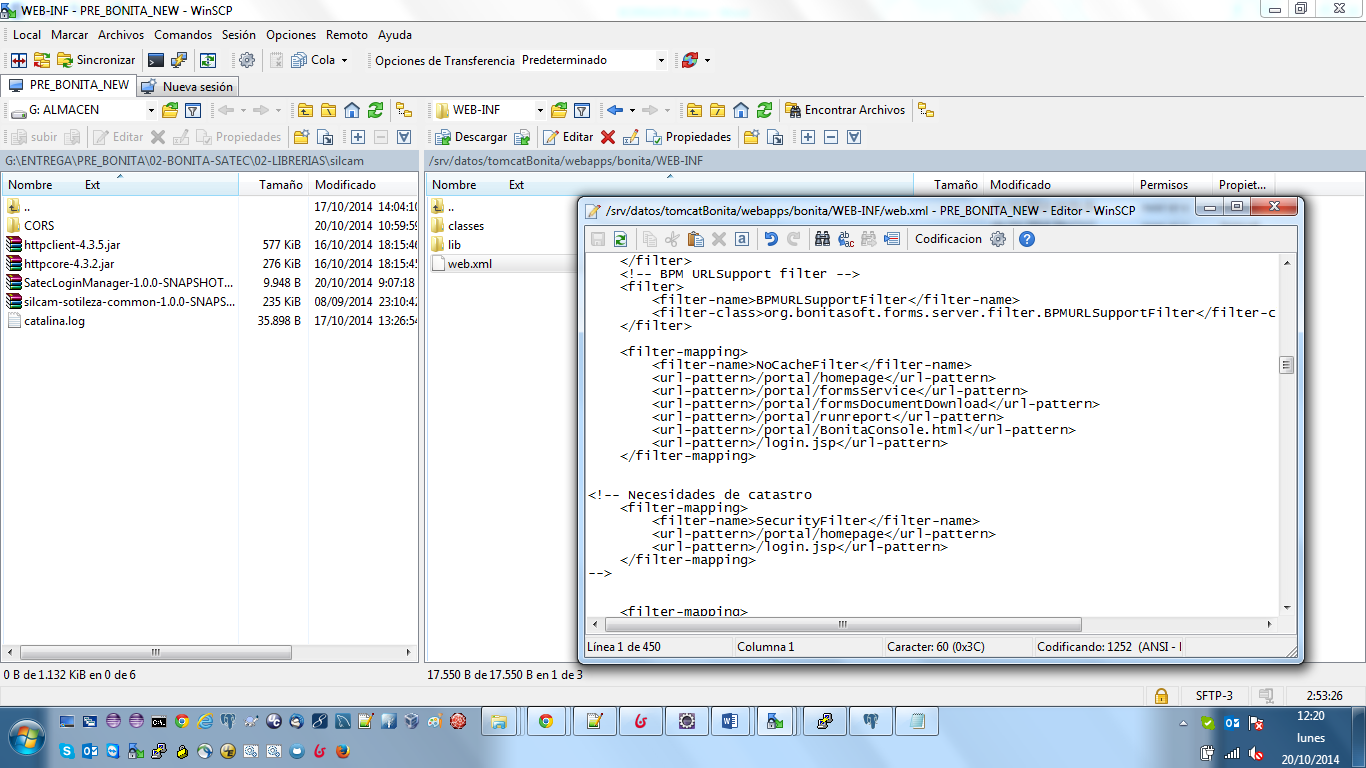
<filter-mapping>

<filter-name>SecurityFilter</filter-name>

<url-pattern>/portal/homepage</url-pattern>

<url-pattern>/login.jsp</url-pattern>

</filter-mapping>



## CAS-SATEC

1. Revisar en la ruta "/srv/datos/silcam". Actualizar el igeo-properties

Debe contener estos datos, revisarlo con sistemas. La idea es que estén dados de alta los servicios y los grupos de los servicios:

####

# Datos para conectar con CAS

####

ruta\_cas = https://cas.igeo.satec.es:8443/cas

ruta\_cas\_login = https://cas.igeo.satec.es:8443/cas/login

ruta\_dominio\_cas = .igeo.satec.es

ruta\_servicio\_silcam = http://demo.silcam.igeo.satec.es:8080/SILCAM/j\_spring\_cas\_security\_check

ruta\_servicio\_bonita = http://demo.silcam.igeo.satec.es:8090/bonita/j\_spring\_cas\_security\_check

grupo\_cas\_silcam = cn=silcam,ou=groups,dc=int,dc=igeo,dc=satec,dc=es

grupo\_cas\_bonita = cn=bonita-silcam,ou=groups,dc=int,dc=igeo,dc=satec,dc=es

## CORS

Para poder mostrar desde licenciamiento páginas de Bonita, hay que configurar CORS para poder permitir que distintos servidores confíen entre ellos para incluir páginas en un iframe uno, de otro. Sino daría un error como este:

Refused to display 'http://bonita-silcam.pre.igeo.gov.ao:8080/bonita/portal/homepage?ui=form&ui=form?locale=pt\_BR#form=Pedido de concessão de Direitos Mineiros--1.0$entry&process=6507193768897333579&mode=form&method=null&id=null&tipo=null' in a frame because it set 'X-Frame-Options' to 'SAMEORIGIN'.

Hay que configurar el filtro en:

/opt/BonitaBPMCommunity-6.3.0-Tomcat-6.0.37/conf/web.xml

<!-- Access-Control-Allow-Origin -->

<filter>

<filter-name>CORS</filter-name>

<filter-class>com.thetransactioncompany.cors.CORSFilter</filter-class>

<init-param>

<param-name>cors.allowOrigin</param-name>

<param-value>\*</param-value>

</init-param>

<init-param>

<param-name>cors.supportedMethods</param-name>

<param-value>GET, POST, HEAD, PUT, DELETE</param-value>

</init-param>

<init-param>

<param-name>cors.supportedHeaders</param-name>

<param-value>Accept, Origin, X-Requested-With, Content-Type, Last-Modified</param-value>

</init-param>

<init-param>

<param-name>cors.exposedHeaders</param-name>

<param-value>Set-Cookie</param-value>

</init-param>

<init-param>

<param-name>cors.supportsCredentials</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CORS</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!--hasta aqui Access-Control-Allow-Origin -->

Y añadir las librerías:

**cors-filter-2.1.1.jar**

**java-property-utils-1.9.1.jar**