# Alliberament SISTRA: Mòdul login

Codi: DSI-LIBSISTRA-LOGIN


# FULL DE CONTROL DE DOCUMENT

## DOCUMENT/ARXIU

| | |
|---|---|
| Títol: Alliberament SISTRA: Mòdul login | Nom arxiu/s: **SISTRA-PLUGINLOGIN** |
| Codi: **SISTRA-PLUGINLOGIN** | Suport lògic: **Word** |
| Data: setembre 2010 | |
| Versió: **2** | |

## REGISTRE DE CANVIS

| Versió | Pàgines | Motiu del canvi |
|---|---|---|
| 1 | 8 | Creació del document. |
| 2 | 11 | Integració login CAS. |
| | | |

## DISTRIBUCIÓ DEL DOCUMENT

| Nom | Personal |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

## CONTROL DEL DOCUMENT

| PREPARAT | REVISAT/APROVAT | ACCEPTAT | ACCEPTAT |
|---|---|---|---|
| Rafael Sanz | José Vicente Juan Pérez | **Jeroni Navarrete** | **Bernat Albertí** |

***Emplenau-ho amb el nom, la signatura i la data.***

***Només per a clients***

# ÍNDEX

# 1. Objecte

L'objecte d'aquest document és definir les accions que s'han de dur a terme per permetre l'ús de la plataforma SISTRA sense el mòdul de login personalitzat de la DGIDT.

# 2. Mòdul de login DGIDT

La DGIDT ha definit un mòdul de login personalitzat fortament lligat al seu repositori d'usuaris SEYCON, que té les característiques següents:

- Single sign on corporatiu dins de la CAIB (acoblat a la gestió d'usuaris de la DGIDT).
- Permet 3 formes d'accés: anònim, autenticat per usuari/contrasenya i autenticat per certificat.
- Utilitza el mòdul de signatura de la DGIDT.
- Desenvolupat per a ús exclusiu en el Jboss customitzat de la CAIB.

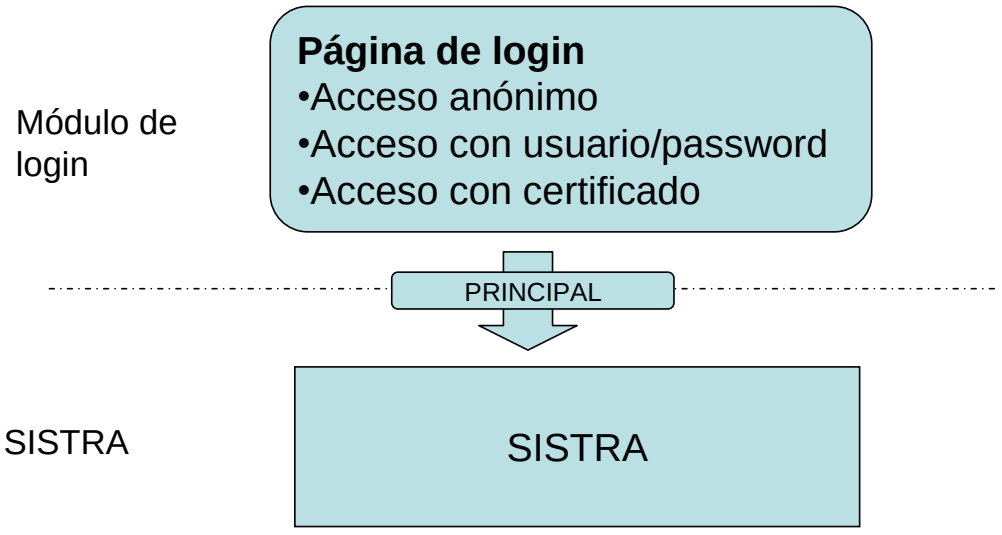
# 3. Funcionalitats requerides del mòdul de login

La plataforma de SISTRA està basada en els estàndards de seguretat de J2EE, segons els quals una aplicació espera rebre una identitat (principal) que tingui definits uns permisos d'accés (rols).

Des del punt de vista de la plataforma, l'únic requisit és que s'ha de rebre una identitat (principal) que ens indiqui les propietats següents:

- Mètode d'accés (anònim, autenticat per usuari/contrasenya o autenticat per certificat).
- NIF/CIF de l'usuari autenticat (en cas que no sigui anònim).
- Nom de l'usuari autenticat (en cas que no sigui anònim).

L'accés anònim serà un accés autenticat amb usuari especial (nobody).



# 4. Solució proposada

Perquè la plataforma SISTRA s'alliberi de manera que no comprometi l'ús del mòdul de login de la DGIDT (ni cap altra solució de single sign on), es proposa alliberar la plataforma amb una solució que faci la plataforma operativa però que no obligui a adoptar cap solució de single sign on.

És tasca de l'organisme decidir quina serà la seva solució més adequada per a la gestió corporativa del single sign on i estudiar com s'ha d'integrar amb la plataforma SISTRA.

La solució de single sign on amb la qual s'alliberarà SISTRA i que servirà com a exemple de la funcionalitat requerida es basarà en:

- Implementació d'un login module JAAS per Jboss. Aquest login module funcionarà contra unes taules d'usuaris de test. Es proveiran els fitxers font d'aquest login module per si l'organisme el vol modificar per fer-lo funcionar contra altres taules, ldap, etc.
- Pàgines de login situades en cada mòdul web que implementaran un accés similar al que ofereix la pàgina de login de la DGIDT. Per a l'accés amb certificats, s'utilitzarà el plugin de @firma, ja que se suposa que serà la DGIDT l'única que utilitzi la seva API de signatura.
- Per a la gestió del single sign on s'utilitzarà la gestió incorporada per JBoss. Jboss té incrustat un Apache Tomcat que proporciona un mecanisme de SSO usant una valve (http://www.jboss.org/community/docs/doc-12280).

# 5. Interfície principal

La interfície que ha d'implementar el principal autenticat és la següent:

| Mètode | Descripció | Paràmetres | Resultat |
|---|---|---|---|
| getMetodoAutenticacion | Obté mètode d'autenticació. | Principal: principal autenticat | Mètode autenticació |
| getNif | Retorna el NIF/CIF de l'usuari autenticat (en cas que no sigui anònim). | Principal: principal autenticat | NIF |
| getNombreCompleto | Obté el nom complet de l'usuari autenticat (en cas que no sigui anònim). | Principal: principal autenticat | Nom complet |

*Per a més informació consultau el javadoc.*

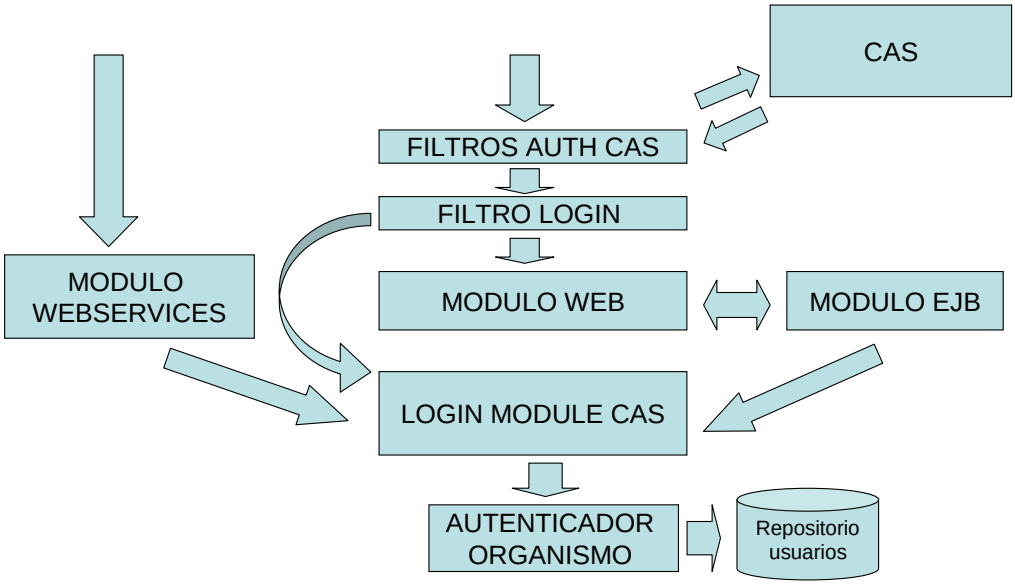
# 6. ANNEX I: Integració amb CAS

SISTRA ofereix un plugin de login que permet integrar-se amb el servei de single sign on CAS (Central Authentication Service) de Jasig.

CAS està basat en filtres web específics que implementen l'autenticació personalitzada de CAS. Aquesta autenticació no es trasllada al contenidor web (funcions getPrincipal i isUserInRole de la HttpServletRequest), sinó que són simulats per mitjà d'un wrapper de la petició que intercepta la petició original. Com que no es trasllada al contenidor web, no es propaga la informació de seguretat al contenidor d'EJB quan s'invoca la capa EJB des de la capa web.

Per realitzar aquesta integració s'ha dissenyat la solució següent:

- Inclusió dels filtres web d'autenticació de CAS en tots els mòduls web.
- Inclusió d'un filtre web personalitzat que realitza el login en el contenidor web, perquè la informació es traslladi al contenidor d'EJB.
- Login module que rep el principal validat per CAS en la capa web i ho autentica. Aquest login module pot rebre dos tipus de peticions:
    - Referència a usuaris validats per CAS, pels quals es fia de la seva autenticació
    - Usuari/contrasenya (per exemple, per a capa de web services) que autenticarà contra la font de dades d'usuaris. Per poder personalitzar la font de dades d'usuaris s'ha externalitzat aquesta autenticació en una classe que haurà de ser implementada segons la implantació en l'organisme.




D'altra banda, CAS haurà de ser particularitzat perquè SISTRA pugui rebre la informació següent en els atributs de l'usuari autenticat per CAS:

- NIF: NIF de l'usuari autenticat
- Nom: Nom complet de l'usuari autenticat
- Mètode d'autenticació: C (Certificat) / O (Usuari) / A (Anònim)

Per realitzar la integració de SISTRA amb CAS cal realitzar els passos següents:

- Compilar el projecte indicant que es treballa amb CAS:
  - En el fitxer config.properties, que estableix les opcions de compilació del projecte, cal establir les propietats següents:

```
# Configuracion login: JAAS o CAS
login-config=CAS


# Login CAS: info necessària per als filtres de CAS
#    - URL on es troba CAS
cas.urlCas=https://hostnameCAS/cas
#      - URL on es troba sistra
cas.urlSistra=https://hostnameSistra
#      - URL on es troba sistra (separada per fronts
(sistrafront,formfront,zonaperfront,redosefront) i backs
(resta)
cas.urlSistra.front=https://hostnameSistraFronts
cas.urlSistra.back=https://hostnameSistraBacks
```

- Compilar el mòdul d'integració amb SISTRA: /integracio/llibreria-casClient a partir del build.xml situat en aquest directori. Es generarà el jar: /integracio/llibreria-casClient/output/product/sistra-casClient.jar.

- Incloure les llibreries següents en el default/lib de Jboss:

  cas-client-core-3.1.10.jar

  saaj-api.jar

  saaj-impl.jar

  loginModuleCIM.jar

  opensaml-1.1b.jar

  sistra-casClient.jar

  xmlsec-1.3.0.jar

- Eliminar del default/lib de JBoss la implementació de saaj de JBoss, ja que és massa antiga per usar-se amb CAS:

  jboss-saaj.jar

- Assegurar-se que s'ha actualitzat el stack d'XML de JBoss. Per això, cal copiar a /lib/endorsed les llibreries:

  resoldre.jar

  xercesImpl.jar

  xml-apis.jar

- Modificar la configuració de login de JBoss per afegir el login module per CAS:

```
<application-policy name = "seycon">
```

```
 <authentication>
<login-moduli code = "és.caib.sistra.casClient.loginModule.CasInternalLoginModule"
        flag = "sufficient">
                <moduli-option name="unauthenticatedIdentity">nobody</moduli-option>
 </login-moduli>
 </authentication>
</application-policy>
```

- Finalment, caldrà establir el fitxer de propietats del plugin de CAS i indicar la classe que realitza l'autenticació d'usuaris per a usuaris no provinents de CAS. Aquest fitxer de propietats ha d'anar situat a <dir_config> \sistra\plugins\ plugin-login.properties.

```
# Atributs on es recullen les dades de l'usuari autenticat
cas.nifAttribute=nif
cas.nombreAttribute=nombreApellidos
cas.metodoAutenticacionAttribute=metodoAutenticacion
# Rols: en cas que es recuperin d'un atribut de la informació
# recollida per CAS s'indicarà el nom de la propietat. Si els rols
# els estableix la classe autenticadora pròpia de l'organisme s'establirà
# com INTERNAL
cas.rolesAttribute=INTERNAL
#cas.rolesAttribute=rols

# Classe autenticadora
cas.loginModule.autenticador=és.xxx.xxx.xxx
```

La classe autenticadora a implementar per l'organisme ha de complir la interfície següent és.caib.sistra.casClient.loginModule.AutenticadorInt:

```
/**
 * Autentica l'usuari contra la font d'usuaris
 * @param propsPlugin Propietats del plugin
 * @param user Username
 * @param pass Password
 * @return En cas correcte, retorna la informació de l'usuari. En cas incorrecte, retorna nul.
 */
public UserInfo autenticar(Properties propsPlugin, String user,String pass);

/**
 * Obtenir llista de rols d'un usuari. S'usarà quan la llista de rols no sigui proporcionada per CAS
 * @param propsPlugin Propietats del plugin
 * @param user Username
 * @return En cas correcte, retorna llista de rols de l'usuari. En cas incorrecte, retorna nul.
 */
public List<String> obtenerRoles(Properties propsPlugin, String user);
```