Projecte de Postgrau
Seguretat en Serveis i Aplicacions
Universitat Oberta de Catalunya

Servei de Central Authentication Server (CAS) & Single Sign On (SSO)

Cas pràctic

Carlos Marcos Rodríguez
Cristina Pérez Solà

6 de juny de 2016

1 Llicència

Apartat eliminat per motius de confidencialitat

2 Índex de continguts

3 Índex d'il·lustracions

4 Introducció

Central Authentication Service (CAS) es una aplicació web que ens permet implementar un servei de Single Sign On (SSO), que es un procediment d'autenticació que habilita a un usuari per accedir a diferents aplicacions web (en diferents dominis i servidors) fent login (autenticant-se) una única vegada.

El funcionament general és el següent:

1. L'usuari es connecta a una de les aplicacions client del sistema de SSO. Aquesta comprova si l'usuari està autenticat i, si no ho està, redirigeix a la pantalla del servidor de CAS per autenticar-se.

2. L'usuari introdueix el seu usuari i password al sistema, i el sistema d'autenticació CAS valida la correctesa (fa la autenticació).

3. Si l'autenticació es correcta, es crea el token de sessió i l'aplicació de CAS redirigeix al servei web inicial que va sol·licitar l'autenticació. Si l'autenticació és incorrecta, no valida l'autenticació i, per tant no deixa continuar.

4. L'aplicació client rep el token de sessió (inclou l'usuari que s'ha autenticat), valida la seva autenticitat i considera l'usuari autenticat. En aquest moment, l'aplicació client és la que decideix finalment l'accés al seu aplicatiu en funció del rol de l'usuari; és per tant, l'aplicació client la que fa l'autorització.

En aquest moment, l'usuari es troba autenticat en el servidor de CAS i, si s'escau, autoritzat en l'aplicació client. En aquest moment, si l'usuari accedeix a un altre sistema client del mateix CAS (l'anomenarem aplicació client 2) que utilitza la validació SSO, es produeix el mateix procediment. La diferència és que, en el punt 2, simplement es valida el token de sessió existent i el servidor de CAS ja considera l'usuari autenticat, així que directament passa al punt 4, donant el control de nou a "l'aplicació client 2" que procedirà a fer l'autorització per aquest nou aplicatiu; aquest es el que és coneix com Single Sign On (SSO).

És important remarcar la diferència entre els conceptes d'autenticació i autorització. El concepte d'autenticació simplement es tracta de validar si un usuari és vàlid (amb un parell usuari/password correcte, certificat digital vàlid, etc..) mentre que l'autorització és, donat un usuari vàlid, autoritzar o denegar l'accés d'aquest usuari a un servei en concret.

Tal com acabem de veure, la responsabilitat del procés d'autenticació recau sobre el servidor de CAS mentre que el procediment d'autorització recau sobre l'aplicació client.

Una de les solucions més reconegudes que implementen un sistema de SSO és Jasig Cas. Aquesta implementació es una de les més complertes i usades per tal d'implementar un servei de CAS, té abundant documentació i una comunitat molt activa;

a més a més, la seva publicació amb llicència opensource de tipus Apache License 2.0 és una altre dels seus avantatges.

5 Objectius del projecte

Com ja hem vist, una de les tasques principals que ha de fer el servidor de CAS és el procés d'autenticació de l'usuari, que bàsicament consisteix en validar que l'usuari és qui diu ser; l'exemple típic de validació seria el parell usuari-password.

Aquest procés de validació de l'usuari es pot fer de diferents maneres; en el cas d'usuari-password podríem tenir validacions en un arxiu de text, contra un servidor de LDAP o Active Directory o contra una Base de Dades.

Jasig CAS és una aplicació amb molts mòduls i opcions disponibles per autenticació, com per exemple els que cobreixen els mètodes esmentats abans (Active Directory, LDAP, Bases de dades) a més molts altres. Dintre d'aquests, tenim també disponible l'autenticació mitjançant certificats X509 (Certificats digitals).

Per aquest mòdul, afegirem a l'annex un exemple de funcionament i configuració per tal d'extendre la solució presentada.

Per tant, podem resumir els objectius del projecte en:

1. Presentar, implementar i configurar el servei de Jasig CAS en les seves variants més típiques com són Base de Dades i Arxius de text, que ens permetran comprendre el funcionament bàsic de la solució.

2. Implementar un parell d'aplicacions client que facin servir el servidor de CAS per tal de comprendre el funcionament del sistema de Single Sign On (SSO).

Amb aquest objectius, cobrim un extens anàlisi de l'aplicatiu Jasig CAS i veiem la seva implementació i configuració amb diferents sistemes. A més a més, presentarem la possibilitat d'autenticació mitjançant certificat digital amb el mòdul d'autenticació amb X509.

6 Metodologia

Tenint en compte els objectius del projecte, he optat per seguir la metodologia clàssica o en cascada, ja que els objectius estan ben definits i no seran canviants en el temps de desenvolupament.

Les etapes d'aquest cicle de vida són:

1. Anàlisi: Definició funcional del sistema a construir i Definició de requisits funcionals i no funcionals de l'aplicatiu
2. Disseny: Definició de l'arquitectura, Disseny intern de l'aplicació (aplicatius, mòduls, llibreries, estructures de dades)
3. Implementació: construcció del programari.
4. Proves i Validació: Pla de proves a executar per tal de validar l'aplicatiu en base al requisits funcionals
5. Documentació final: Agrupar la diferent documentació generada durant el projecte per tal de construir l'entregable de l'aplicatiu i la documentació final

7 Planificació temporal

En base a la metodologia explicada en el punt anterior i tenint en compte els terminis generals del projecte, es proposa la següent planificació del projecte (Il·lustració 1) en un esquema de Diagrama de Gantt.

S'han agrupat les tasques en tres grans fites, que en el cas del projecte són les diferents PAC's; tot i que tenen una data concreta d'entrega, no necessàriament ocupen el període complert que per data li corresponen, si no que s'aprofita el temps en funció de les necessitats concretes de cada fase.

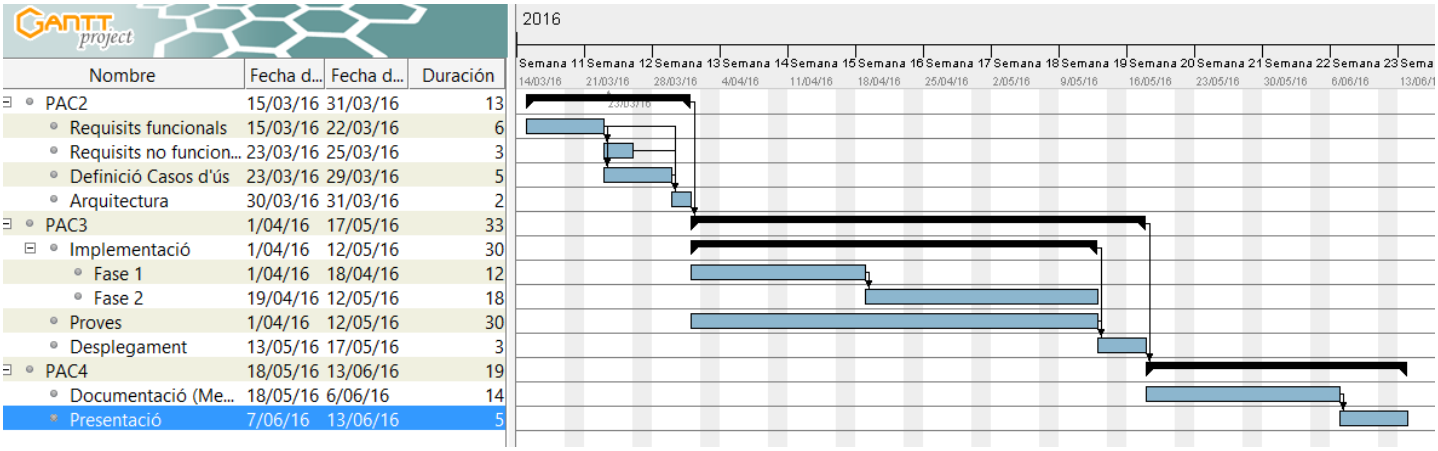
La mesura que s'ha utilitzat és per dies, tenint en compte la dedicació de 2,5h diàries.

GANTT project

Nombre	Fecha de inicio	Fecha de fin	Duración
PAC2	15/03/16	31/03/16	13
Requisits funcionals	15/03/16	22/03/16	6
Requisits no funcionals	23/03/16	25/03/16	3
Definició Casos d'ús	23/03/16	29/03/16	5
Arquitectura	30/03/16	31/03/16	2
PAC3	1/04/16	17/05/16	33
Implementació	1/04/16	12/05/16	30
Fase 1	1/04/16	18/04/16	12
Fase 2	19/04/16	12/05/16	18
Proves	1/04/16	12/05/16	30
Desplegament	13/05/16	17/05/16	3
PAC4	18/05/16	13/06/16	19
Documentació (Memòr...	18/05/16	6/06/16	14
Presentació	7/06/16	13/06/16	5

Il·lustració 1

La relació de dependències entre les tasques és la següent (Il·lustració 2):



Nombre	Fecha d...	Fecha d...	Duración
PAC2	15/03/16	31/03/16	13
Requisits funcionals	15/03/16	22/03/16	6
Requisits no funcion...	23/03/16	25/03/16	3
Definició Casos d'ús	23/03/16	29/03/16	5
Arquitectura	30/03/16	31/03/16	2
PAC3	1/04/16	17/05/16	33
Implementació	1/04/16	12/05/16	30
Fase 1	1/04/16	18/04/16	12
Fase 2	19/04/16	12/05/16	18
Proves	1/04/16	12/05/16	30
Desplegament	13/05/16	17/05/16	3
PAC4	18/05/16	13/06/16	19
Documentació (Me...	18/05/16	6/06/16	14
Presentació	7/06/16	13/06/16	5

Il·lustració 2

Aquesta definició ens dóna un total de 65 dies, el que tenint en compte la mesura estimada de 2,5 hores diàries ens dona un esforç total de 162,5 hores.

El contingut de les diferents tasques, en base a la metodologia definida anteriorment és la següent:

7.1 PAC2

Aquesta primera part inclou les fases d'anàlisi i disseny.

- Requisits funcionals: En la definició dels requisits funcionals del sistema, es defineixen les entrades, comportaments i sortides que són necessàries en l'aplicatiu software.

- Requisits no funcionals: a diferència dels requisits funcionals, el no funcionals són criteris que no especifiquen necessitats que es refereixen al comportament de l'aplicatiu. Típicament son requisits de funcionament com poden ser el rendiment, la disponibilitat, el hardware i el software o l'escalabilitat.

- Definició de casos d'ús: a partir de la definició dels requisits funcionals de l'aplicatiu, es desenvoluparan els diferents casos d'us, que descriuran els passos i activitats a portar a terme per tal de realitzar els diferents processos.

- Arquitectura: a partir de la definició de tot l'anterior, es passarà a definir l'arquitectura definitiva sobre la qual es construirà tot l'aplicatiu software.

7.2 PAC3

A la segona part s'inclou la fase d'implementació i proves que deriva de la fase d'anàlisi i disseny recentment portada a terme.

- Implementació fase 1: En aquesta primera fase d'implementació, es farà la construcció del software que avarca una part dels objectius definits: Configuració del servei CAS per fer servir autenticació mitjançant arxiu i Base de dades

- Implementació fase 2: en aquesta segona fase, es desenvoluparà la resta del programari que forma part dels objectius definits, que bàsicament és implementar i configurar el parell d'aplicacions clients pel servei de CAS.

- Proves: es definirà la bateria de proves que validarà l'aplicatiu software. Aquesta definició i fase de proves s'anirà executant durant tota la construcció, seguint principis de TDD (Test Driven Development).

- Desplegament: una vegada construït i validat el software, es procedirà al seu desplegament per tal de ser utilitzat per producció.

7.3 PAC4

Finalment a la tercera i última part s'inclou la fase de documentació

- Documentació (Memòria): la memòria i documentació del projecte s'anirà desenvolupant durant tot el procés. Aquesta fase final consistirà en la consolidació de tota aquesta documentació, maquetació i validació de la mateixa, així com la elaboració de les conclusions finals del projecte.

- Presentació: elaboració de la presentació final del projecte.

8 Disseny

A continuació, s'inclou tota la fase de d'anàlisi i disseny de l'aplicatiu a construir. Aquesta fase inclou, per tant, la definició dels Requisits funcionals i no funcionals, la definició dels casos d'ús i la presentació de l'arquitectura software escollida.

Dintre de tota aquesta fase de disseny, farem referència al sistema CAS a construir com 'el sistema'; d'altra banda, ens referirem com a 'aplicacions clients' a les aplicacions que es construiran i que fan servir el sistema com a servei d'autenticació.

8.1 Requisits funcionals

Els requisits funcionals ens han d'especificar d'una manera clara i concisa com ha de ser el nostre sistema. A continuació, es presenten els requisits funcionals que s'han detectat que ha de tenir el sistema que anem a construir; aquests requisits es defineixen numerats de la forma RFX

- RF2 - El sistema ha de ser permetre autenticar-se mitjançant usuari/password validat contra un arxiu de text que inclou parells usuari/password.

- RF3 - El sistema ha de ser permetre autenticar-se mitjançant usuari/password validat contra una taula en Base de Dades que inclou els parells usuari/password.

- RF4 - El sistema, una vegada validat l'usuari, ha de generar una sessió que ha de ser vàlida per totes les aplicacions que fan servir el sistema com a servei d'autenticació.

- RF5 - El sistema ha d'impedir l'accés a usuaris que no siguin vàlids conforme als mètodes d'autenticació definits als RF1, RF2 i RF3.

- RF5 - El sistema ha de permetre fer logout del sistema, de forma que s'invalidi la sessió en totes les aplicacions web que fan servir el sistema com a servei d'autenticació.

Tal com s'ha definit anteriorment a l'abast i objectiu del projecte, s'han d'implementar també un parell d'aplicacions client que facin servir el sistema de CAS construït per tal de validar la seva correctesa i mostrar el seu funcionament. Tot i que no seran aplicacions que formin part pròpiament del sistema construït i objectiu primer d'aquest projecte, sí que s'han de definir els Requisits que han de tenir aquestes aplicacions client. Aquest Requisits funcionals de les aplicacions client es numeren de la forma RF_C_X

- RF_C_1 - Si un usuari accedeix a l'aplicació client i no té una sessió vàlida, aquesta ha d'impedir l'accés i ha de redirigir al sistema per tal que sigui aquest qui faci l'autenticació

- RF_C_2 - Si un usuari accedeix a l'aplicació client i té una sessió vàlida, l'aplicatiu client ha d'autoritzar/denegar l'accés en base a la seva configuració pròpia d'accés d'usuaris.

- RF_C_3 - Si un usuari es desconnecta de l'aplicació client (fa logout), s'ha d'invalidar la sessió a tots els aplicatius client.

8.2 Requisits no funcionals

A diferència dels requisits funcionals, el no funcionals són criteris que no especifiquen necessitats que es refereixen al comportament de l'aplicatiu. A continuació, s'enumeren els requisits no funcionals que han d'incloure l'aplicatiu a construir.

Com que es tracta de requisits no funcionals, que no es refereixen per tant al comportament del sistema, no s'ha fet agrupació en forma de sistema/clients com s'ha fet anteriorment, sinó que son els propis requisits no funcionals els que especifiquen l'abast del propi requisit.

Es numeren mitjançant la nomenclatura R_NF_X.

- R_NF_1 - El sistema i els clients s'han de comunicar de forma segura mitjançant el protocol de comunicació SSL

- R_NF_2 - Els passwords que s'emmagatzemin al sistema es guardaran de forma segura mitjançant alguna funció de hash com per exemple MD5 o bé SHA256

- R_NF_3 - El sistema ha de ser compatible amb els navegadors web Google Chrome, Mozilla Firefox i Internet Explorer en les seves darreres versions.

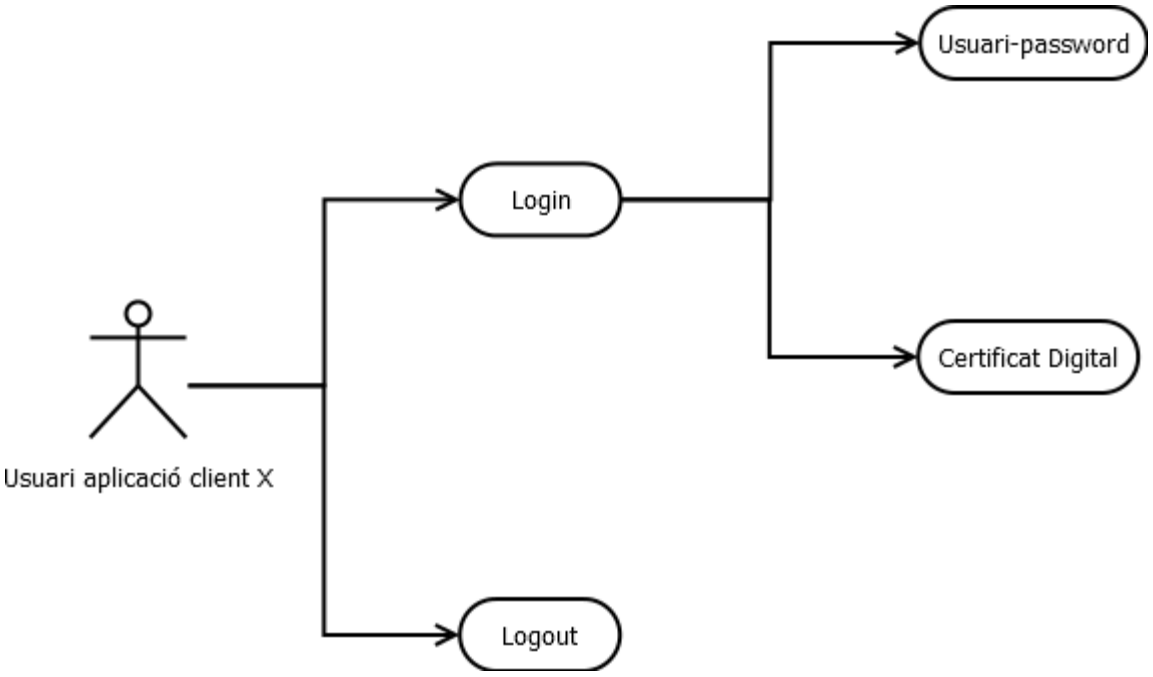
- R_NF_4 - Les llibreries i programari de tercers que faci servir el sistema, ha de ser amb llicència Opensource i gratuït.

- R_NF_5 - Les versions de les llibreries i programari de tercers que faci servir el sistema, seran les darreres versions estables (anomenades versions Release)

8.3 Definició de casos d'ús

A partir de la definició dels requisits funcionals que hem fet anteriorment, a continuació es presenten els diferents casos d'ús que formen part del sistema i que descriuen les diferents activitats i passos que realitzen els actors amb el sistema per tal de realitzar els diferents processos.

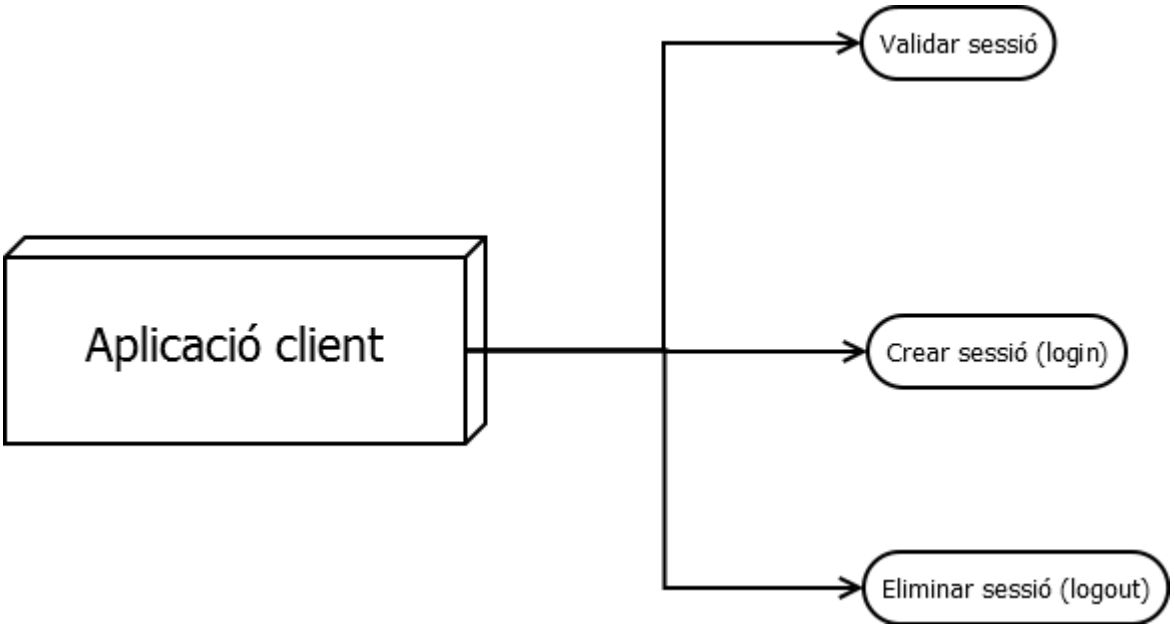
Es presenta, primerament, el cas d'ús que pot fer l'usuari (Il·lustració 3); en aquest cas, ens referim usuari com aquell que fa servir una de les aplicacions client:



Il·lustració 3

Com podem veure, l'actor és Usuari de l'aplicació client X. Això és així ja que els nostres usuaris treballaran directament amb les aplicacions client, no amb el sistema. La figura és auto explicativa pels diferents casos d'ús que pot fer l'usuari.

Com que els nostres usuaris treballen directament amb les aplicacions clients, són aquestes les que s'han de comunicar amb el sistema per tal d'executar les diferents accions que deriven dels casos d'ús de l'usuari; per tant, tenim un segon actor que es tracta de les pròpies aplicacions client que es comuniquen amb el nostre sistema (Il·lustració 4)



Il·lustració 4

Com podem veure a la Il·lustració 4, són les aplicacions clients les que es comuniquen amb el sistema per tal de portar a terme els diferents casos d'ús que tenen a veure amb la gestió de la sessió de l'usuari, que estan directament relacionats amb els casos d'ús que porta a terme el propi usuari.

8.4 Arquitectura

Apartat eliminat per motius de confidencialitat

9 Fase d'implementació

En aquesta tercera part s'inclou la fase d'implementació i proves que deriva de la fase d'anàlisi i disseny s'ha portat a terme en la fase anterior.

- **Implementació fase 1:** En aquesta primera fase d'implementació, es farà la construcció del software que avarca una part dels objectius definits: Configuració del servei CAS per fer servir autenticació mitjançant arxiu i Base de dades

- **Implementació fase 2:** en aquesta segona fase, es desenvoluparà la resta del programari que forma part dels objectius definits: configurar el parell d'aplicacions clients pel servei de CAS

- **Proves:** es definirà la bateria de proves que validarà l'aplicatiu software. Aquesta definició i fase de proves s'anirà executant durant tota la construcció, seguint principis de TDD (Test Driven Development).

- **Desplegament**: una vegada construït i validat el software, es procedirà al seu desplegament per tal de ser utilitzat per producció.

9.1.1 Aplicatius d'exemple

Per tal de validar el funcionament de l'aplicatiu de CAS, és necessari que disposem de un mínim de dos aplicacions clients per tal de validar el seu correcte funcionament. A continuació, es presenta la definició dels dos aplicatius clients que ens serviran per tal de validar el software construït.

9.1.1.1 Aplicatiu 1

Suposarem que una empresa que es dedica a la importació i distribució de productes té el personal següent (Il·lustració 5):

Personal	*Carrec*	*Login*	*Password*
Saul Berenson	*Director*	*berenson*	*berenson75*
Carrie Mathison	*Cap d'operacions*	*mathison*	*mathison75*
Fara Sherazi	*Analista*	*sherazi*	*sherazi75*
Peter Quinn	*Coordinador de missions*	*quinn*	*quinn75*

Il·lustració 5

Com podem veure, en aquest quadre s'especifica el password que tindrà cadascun dels usuaris. Cal remarcar que aquests passwords estaran gestionats per l'aplicatiu de CAS (encarregat de la validació dels usuaris); és l'aplicació client que s'encarregarà de l'autorització.

L'empresa disposa d'una aplicació que té els mòduls següents: Direcció, Operacions i Missions on es realitzen les diferents operacions.

Els rols necessaris per gestionar l'aplicació son els següents (Il·lustració 6):

Rol	*Descripció*
Director (DIR)	*Tasques de gestió de clients i pressupostos*
Oficial d'Intel·ligència (ODI)	*Decisions sobre les operacions*
Analista (ANA)	*Anàlisis d'operacions*
Agent black-ops (ABO)	*Coordinació treball de camp.*

Il·lustració 6

Dintre d'aquest aplicatiu, tenim la següent assignació dels rols als usuaris (Il·lustració 7):

Personal de l'empresa	*Rol/s*
Saul Berenson	*DIR, ODI, ANA, ABO*
Carrie Mathison	*ODI, ANA*
Fara Sherazi	*ANA*
Peter Quinn	*ABO*

Il·lustració 7

Els rols necessaris per executar les diferents operacions de cada mòdul son els es presenten a continuació (Il·lustració 8):

Mòdul	Operació	Rol
Direcció	Gestió clients	DIR
	Gestió pressupostos	DIR
Operacions	Gestió operacions	ODI
	Gestió fonts d'informació	ODI
	Adquisició dades	ANA
	Anàlisi dades	ANA
Missions	Gestió material	ABO
	Gestió viatges	ABO
	Informes	ABO
	Aprovació missió	ODI

Il·lustració 8

Les diferents pantalles que formen part d'aquest aplicatiu no tindran funcionalitat, simplement es presentarà la pàgina amb la opció que correspongui; amb això ja tenim suficient per validar el comportament de l'aplicatiu client, ja que no és objectiu del projecte implementar aquesta funcionalitat de l'aplicatiu client.

Per tal d'implementar aquesta solució, s'indiquen les principals configuracions portades a terme a l'aplicatiu client:

Organitzarem l'aplicatiu en forma de carpetes/subcarpetes per tal de restringir l'accés d'una manera còmoda i tenir les funcionalitats agrupades de forma clara.

- Directori /public → inclourà la pàgina principal d'accés
- Directori /secure → inclourà tota la part que requereix autenticació, organitzada com:
    - /direccio, /missions, /operacions → s'inclouen totes les pàgines per realitzar aquestes operacions, de forma que podem restringir l'accés per URL

Per configurar l'autorització a l'accés a cadascun de les opcions, fem servir el mòdul java Spring Security.

Donada aquesta organització de funcionalitats per URL-Carpetes, es realitza la següent configuració al mòdul Spring Security a l'arxiu de applicationContext-Security.xml de l'aplicatiu client (Il·lustració 9):

```
<security:http entry-point-ref="casAuthenticationEntryPoint" auto-config="true">

    <security:intercept-url pattern="/secure/direccio/*" access="ROLE_DIR"></security:intercept-url>

    <security:intercept-url pattern="/secure/operacions/adquisicioAnalisi/*" access="ROLE_ANA"></security:intercept-url
    <security:intercept-url pattern="/secure/operacions/gestio/*" access="ROLE_ODI"></security:intercept-url>

    <security:intercept-url pattern="/secure/missions/aprovacio/*" access="ROLE_ODI"></security:intercept-url>
    <security:intercept-url pattern="/secure/missions/gestioInformes/*" access="ROLE_ABO"></security:intercept-url>

    <security:intercept-url pattern="/secure/**" access="ROLE_DIR,ROLE_ODI,ROLE_ANA,ROLE_ABO"></security:intercept-url>

    <security:custom-filter position="CAS_FILTER" ref="casAuthenticationFilter"></security:custom-filter>

    <security:logout logout-success-url="/cas-logout.jsp"/>
    <security:custom-filter ref="requestSingleLogoutFilter" before="LOGOUT_FILTER"/>
    <security:custom-filter ref="singleLogoutFilter" before="CAS_FILTER"/>
</security:http>
```

Il·lustració 9

Com podem veure, restringim l'accés als diferents apartats per URL al rols requerits. D'aquesta forma, ens assegurem que l'accés a cadascuna de les seccions es fa segons els requisits de l'aplicació.

Altra configuració addicional que necessitem, és configurar cadascun dels rols que té cada usuari dintre de l'aplicació. Com ja hem comentat, és responsabilitat de l'aplicatiu client la gestió de l'autorització de l'accés (Il·lustració 10).

```
<security:user-service id="userService">

    <security:user name="berenson" authorities="ROLE_DIR,ROLE_ODI,ROLE_ANA,ROLE_ABO"></security:user>
    <security:user name="mathinson" authorities="ROLE_ODI,ROLE_ANA"></security:user>
    <security:user name="sherazi" authorities="ROLE_ANA"></security:user>
    <security:user name="quinn" authorities="ROLE_ABO"></security:user>
</security:user-service>
```

Il·lustració 10

El codi mostrat a la configuració és suficientment explicatiu: s'assignen els diferents rols que tindran els usuaris dintre de l'aplicació.

També hem d'indicar a la configuració de Spring Security que aquest aplicatiu funcionarà mitjançant autenticació en un servidor de CAS (Il·lustració 11).

Il·lustració 11

Les referències a server/cas, serà el lloc on desplegarem el nostre servidor de CAS. Addicionalment, s'afegeix la configuració per tal de fer el logout comú de totes les aplicacions (Single Sign Out), de forma que quan es faci logout d'una aplicació client, s'invalidin tots els accessos a la resta d'aplicatius client del servidor de CAS (**¡Error! No se encuentra el origen de la referencia.**).

Apartat eliminat per motius de confidencialitat

Apartat eliminat per motius de confidencialitat

9.1.1.2 Aplicatiu 2

Tenint en compte de que es tracta de dues aplicacions en permetin veure el correcte funcionament dels sistema construït, es proposa que aquesta segon aplicatiu sigui idèntic a l'anterior, amb la diferència que els usuaris disposaran de rols diferents. Es proposa, per tant, aquesta assignació de rols (Il·lustració 12):

Personal de l'empresa	*Rol/s*
Saul Berenson	*DIR, ODI, ANA, ABO*
Carrie Mathison	*DIR, ODI, ANA, ABO*
Fara Sherazi	*ABO*
Peter Quinn	*ABO*

Il·lustració 12

Com podem veure, ara tenim que Saul Berenson i Carrie Mathinson tenen els mateixos rols i, per altra banda, Fara Sherazi i Peter Quin també comparteixen rol dintre de l'aplicació.

Per tant, la configuració serà idèntica a l'aplicatiu 1 presentat anteriorment, amb la diferència en l'assignació de rols, que serà acord a la requerida anteriorment.

Per tal de diferencial ambdues aplicacions, també es canviarà l'aspecte estètic d'aquestes, de forma que sempre puguem diferenciar en quina de les dues estem navegant en aquest moment.

9.1.2 Configuració del sistema Jasig-CAS

A continuació, es presentes les dues implementacions realitzades al sistema CAS: configuració per arxiu pla i configuració per Base de Dades.

9.1.2.1 Configuració CAS per arxiu pla

La primera configuració que implementarem per tal de autenticar els usuaris en el servei de CAS serà la de arxiu pla. Segons els requisits definits a fases anteriors, el password s'emmagatzemarà codificat en format SHA256.

Per fer això, remarquem les principals configuracions que s'han portat a terme:

Arxiu deployerConfigContext.xml:

Modifiquem el servei d'autenticació (primaryAuthenticationHandler) per fer servir arxiu pla:

```
<alias name="acceptUsersAuthenticationHandler" alias="primaryAuthenticationHandler" />

<!-- USER FILE CONFIGURATION -->
<alias name="fileAuthenticationHandler" alias="primaryAuthenticationHandler" />
<alias name="defaultPasswordEncoder" alias="passwordEncoder" />
```

Amb això, ja tenim definit el comportament que necessitem al CAS. Els detalls d'aquesta implementació els trobem a l'arxiu cas.properties

cas.properties:

```
##
# File Authentication
#
file.authn.filename=file:C:\\dev\\casconfig\\users.txt
file.authn.separator=::


##
# General Authentication
#
cas.authn.password.encoding.alg=SHA-256
```

Ja tenim, per tant, definit el comportament concret de la nostra implementació del servei de CAS en arxiu pla. Com podem veure, a l'arxiu de cas.properties definim la ruta on tenim l'arxiu amb els usuaris i passwords i el format (separador), així com la format en que codificarem el password (SHA-256).

Amb aquesta configuració, l'arxiu users.txt amb els parells usuari::password queda com es mostra a continuació (Il·lustració 13):

```
berenson::d81dabe2dd619abba23718eea7db91be363c9598f2ac811dee4dc8e4006897dc
mathinson::36c047898b32c8eba04b649712c61e506f85657bfe9d31c3f21d92cd7b3140fe
sherazi::638e3505faa578f17db505f7b52c5e6a6dfb5ca1efa5aa353f707a70ecc95e98
quinn:239b2cf91673b6ee7dc55f0e2232fb2a8ebc3d7063decea6166341ec4d9df8cb
```

Il·lustració 13

Amb aquestes configuracions, ja tenim llest el sistema CAS per tal d'accedir mitjançant la configuració en arxiu pla.

9.1.2.2 Configuració CAS per Base de Dades

Apartat eliminat per motius de confidencialitat

9.2 Bateria de proves

Per tal de validar tot el conjunt de implementat, es defineixen les següents proves funcionals que s'executen per tal de validar el correcte funcionament del sistema

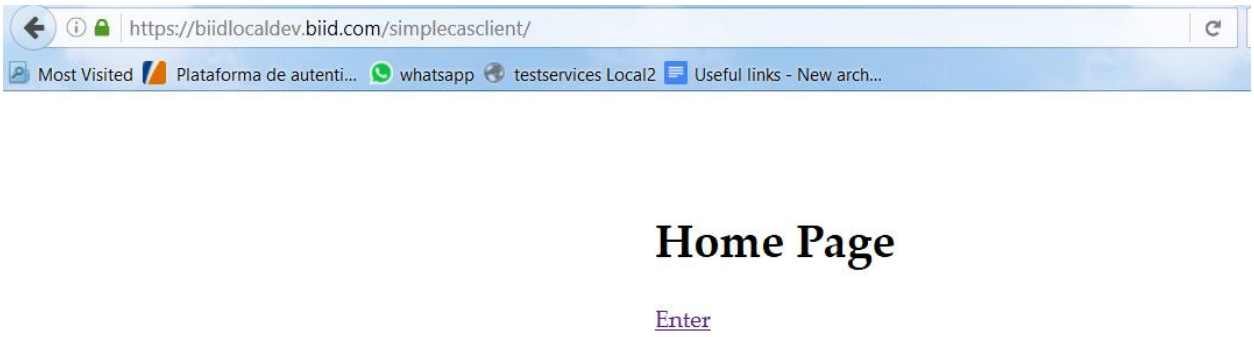
9.2.1 Desplegament

Desplegar els tres components .war (aplicatius d'exemple i CAS) en un servidor Apache Tomcat. El servidor de CAS es trobarà en un Apache Tomcat independent de un altre on es desplegaran els aplicatius client; per tant, s'ha de configurar aquest perquè facin servir ports diferents i no donin conflictes per tal de arrencar-los en una mateixa màquina. Aquests servidors Apache Tomcat ha de tenir configurat SSL sota un servidor Apache 2, tal i com s'especifica a la fase de disseny. Aquests components es desplegaran amb els noms:

/cas → cas.war, el servei d'autenticació CAS
/simplecasclient → simplecasclient.war, aplicatiu de prova 1
/othercasclient → othercasclient.war, aplicatiu de prova 2

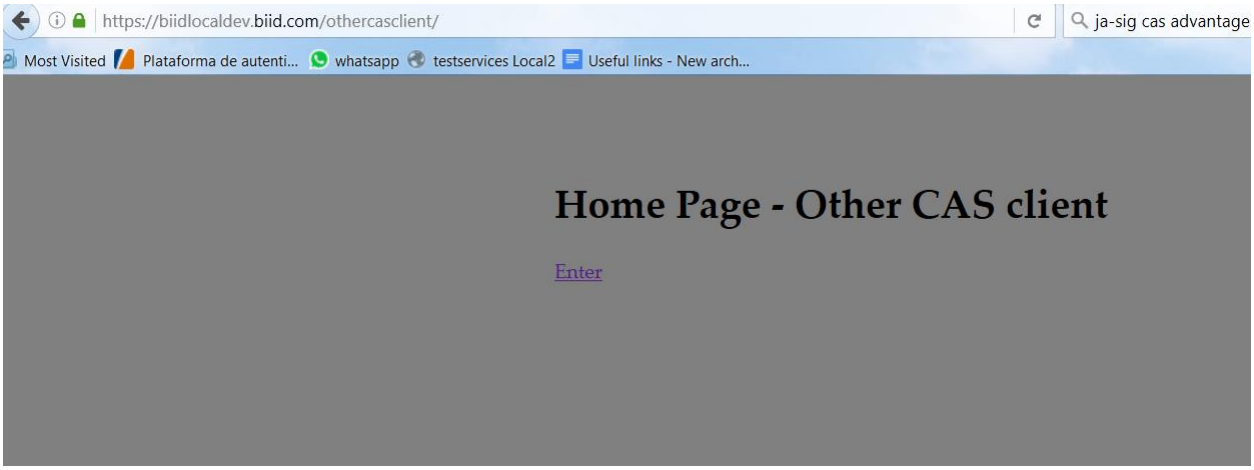
En aquest punt, ja tenim tots el artefactes desplegats. Farem la prova d'accés a cadascun d'ells:

- Simplecasclient (Il·lustració 14)



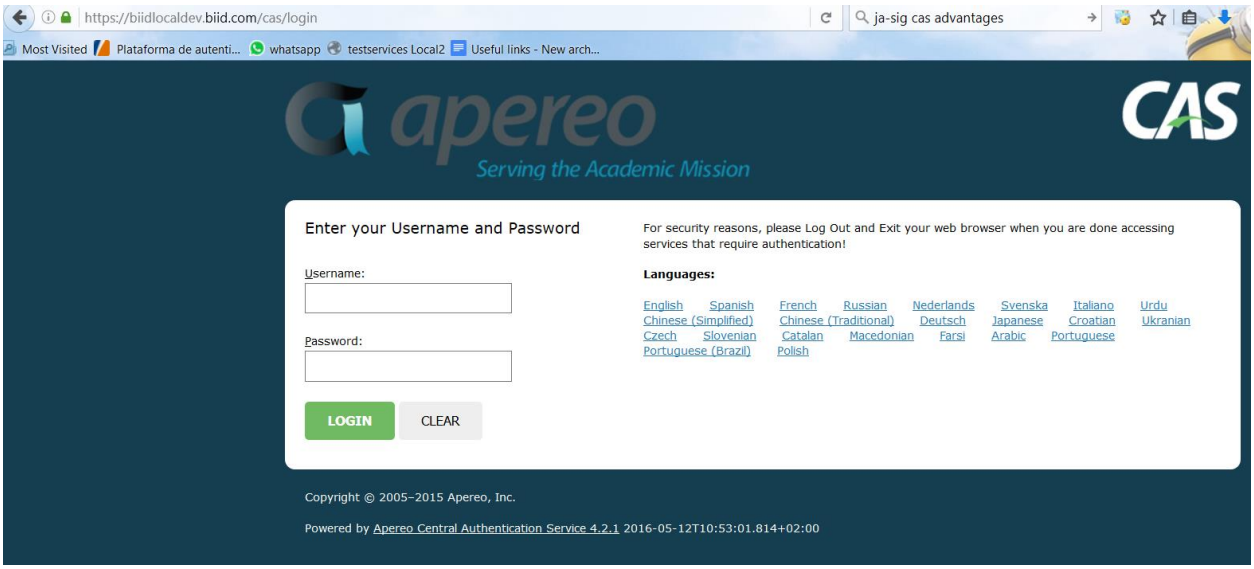
Il·lustració 14

- Othercasclient (Il·lustració 15)



Il·lustració 15

- Cas (Il·lustració 16)



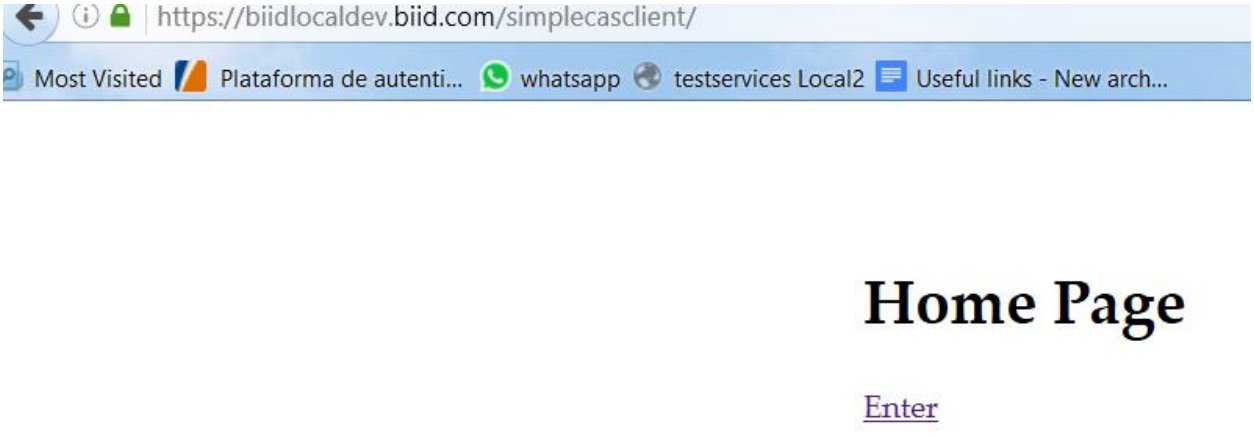
Il·lustració 16

Per a referències específiques de la instal·lació i detalls de la configuració, tenim disponible a l'annex 10.1, 10.2, 10.3 i 10.4

9.2.2 Proves funcionals

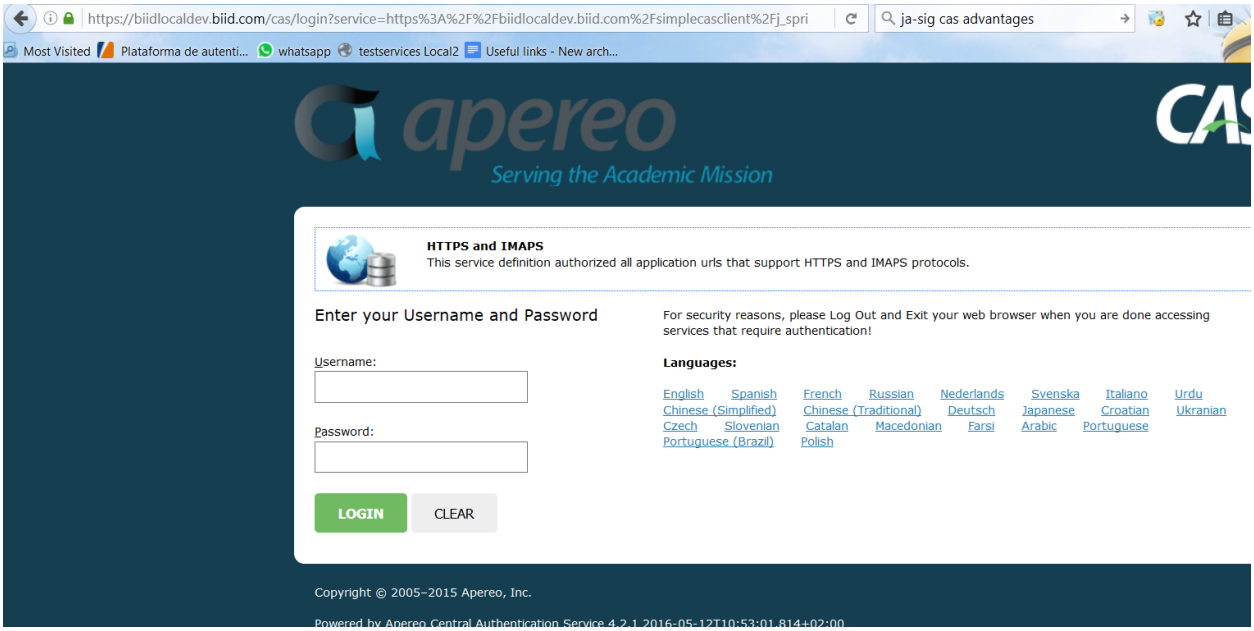
Una vegada desplegats els artefactes, la fase de proves consistirà:

1. Intentar entrar a l'aplicatiu de prova 1 (/casclient, botó "Enter")



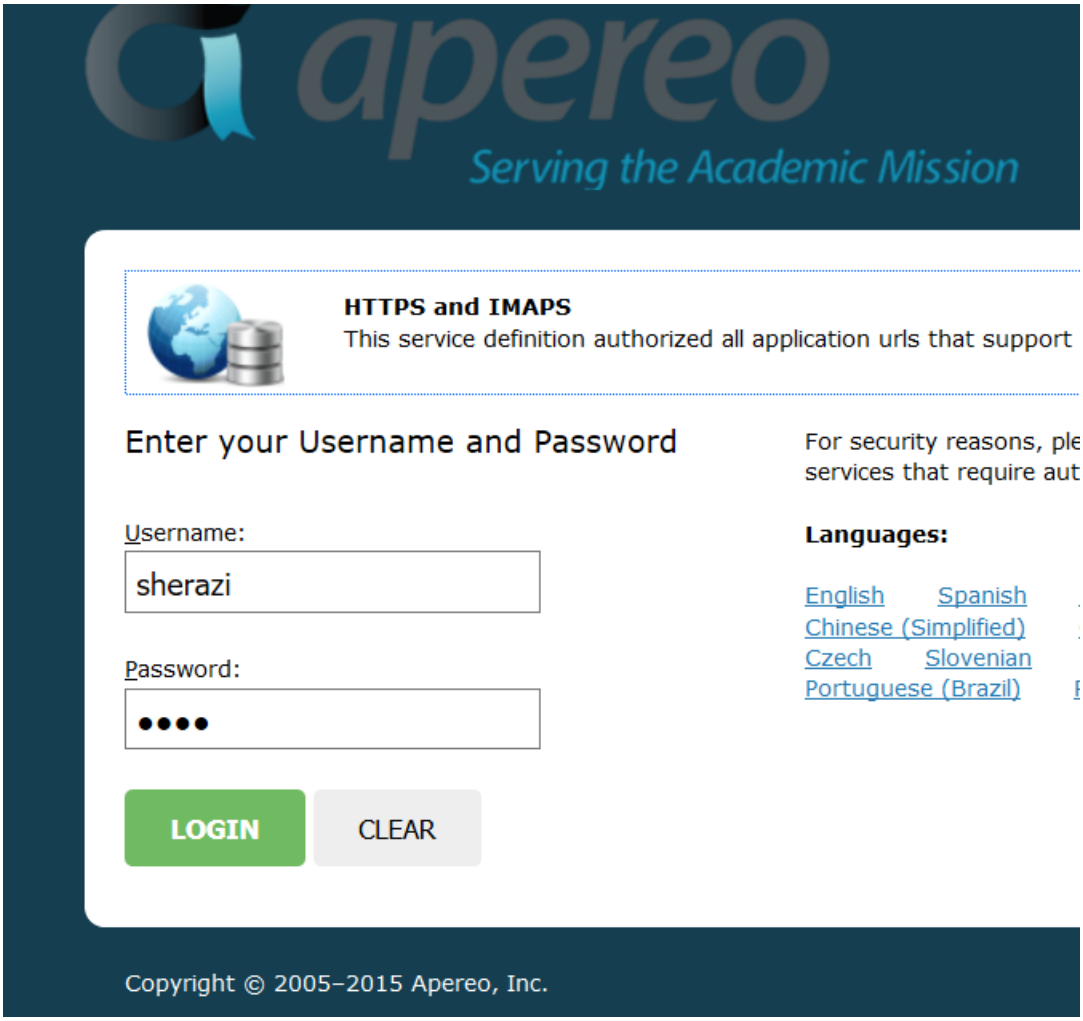
Il·lustració 17

2. El sistema detecta que no ens hem logejat, per tant, ens redirecciona al servidor de cas (ens redirigeix a /cas/login). Si ens fixem a la URL, veiem com indica al servidor de CAS quin es l'aplicatiu origen (simplecasclient)

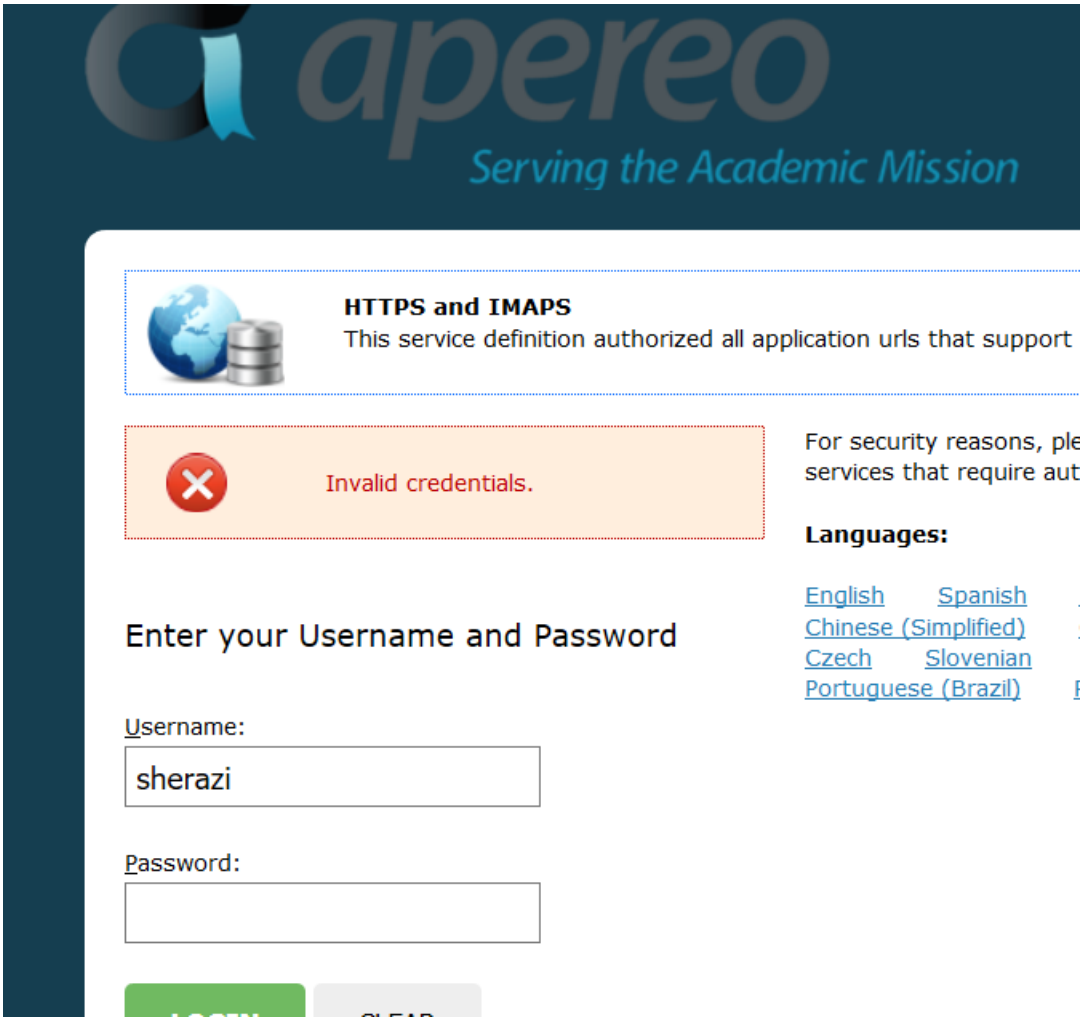


Il·lustració 18

3. Introduïm un usuari/contrasenya no vàlid. El sistema no ens deixa entrar.

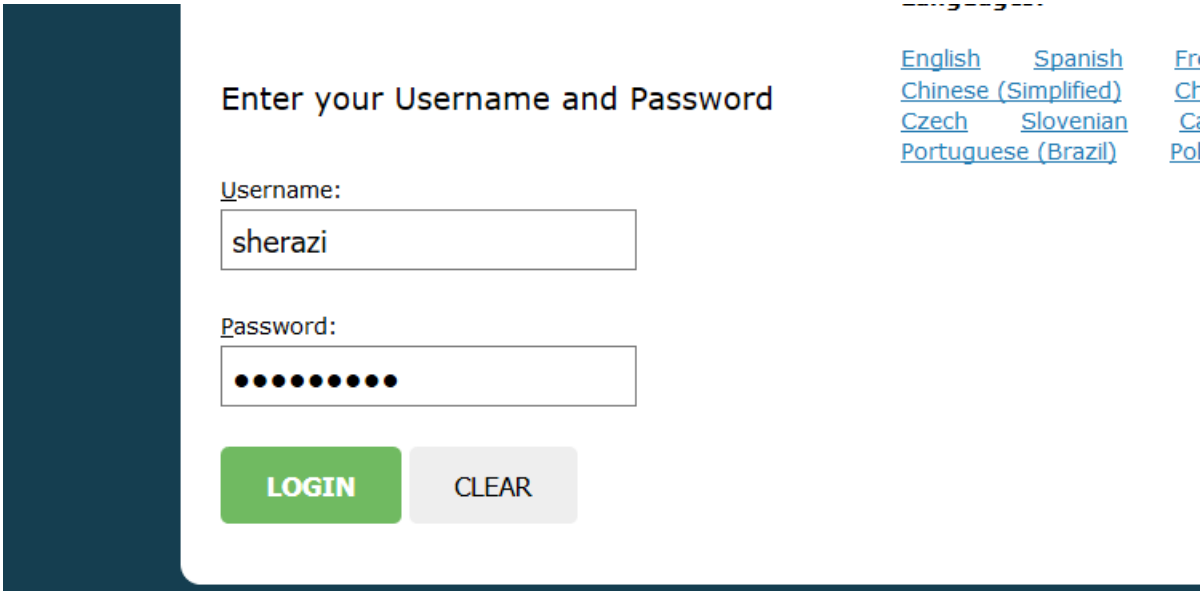


Il·lustració 19



Il·lustració 20

4. Introduïm un usuari/contrasenya vàlid. El sistema ens deixa entrar.



Il·lustració 21

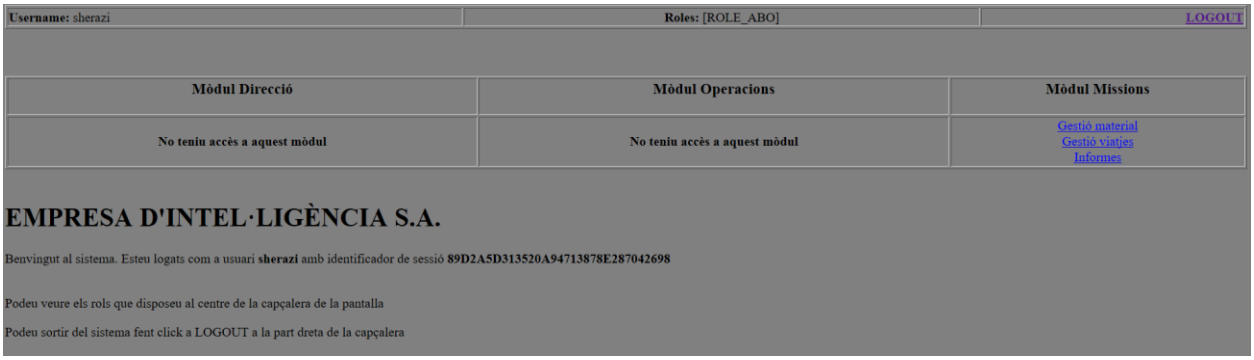
Il·lustració 22

5. Comprovem que podem visualitzar i accedir a les diferents àrees de l'aplicatiu de prova en concordança amb els rols que té l'usuari. Com podem veure, es corresponen les seccions (veure Il·lustració 7 i Il·lustració 8)

6. Provem d'entrar a una àrea a la qual no tenim permisos pel rol de l'usuari. El sistema no ens deixa entrar.

Il·lustració 23

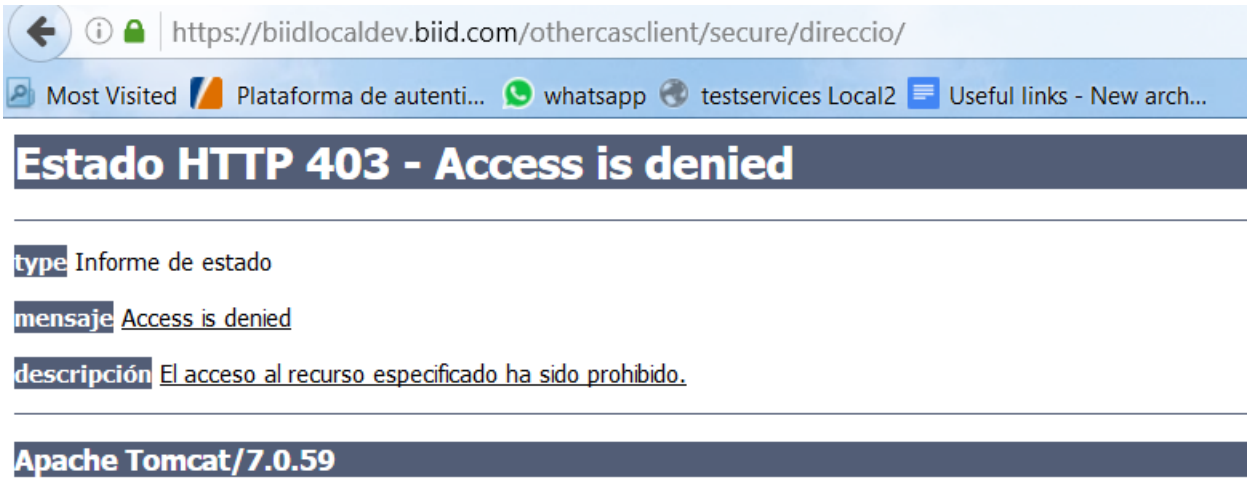
7. Provem d'entrar en l'aplicatiu 2. Com que ja tenim una sessió vàlida a CAS, el sistema ens deixa entrar



Il·lustració 24

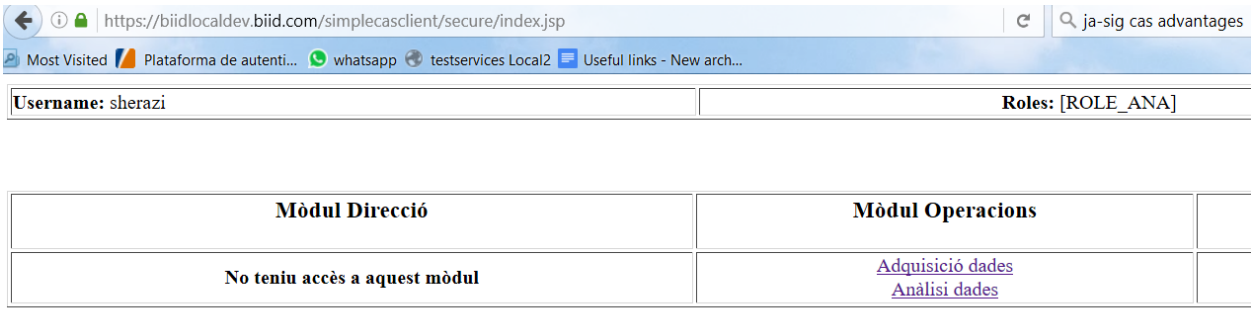
8. Comprovem que podem visualitzar i accedir a les diferents àrees de l'aplicatiu de prova 2 en concordança amb els rols que té l'usuari en aquest segon aplicatiu (veure Il·lustració 12)

9. Provem d'entrar en una àrea a la qual no tenim permisos pel rol de l'usuari. El sistema no ens deixa entrar.



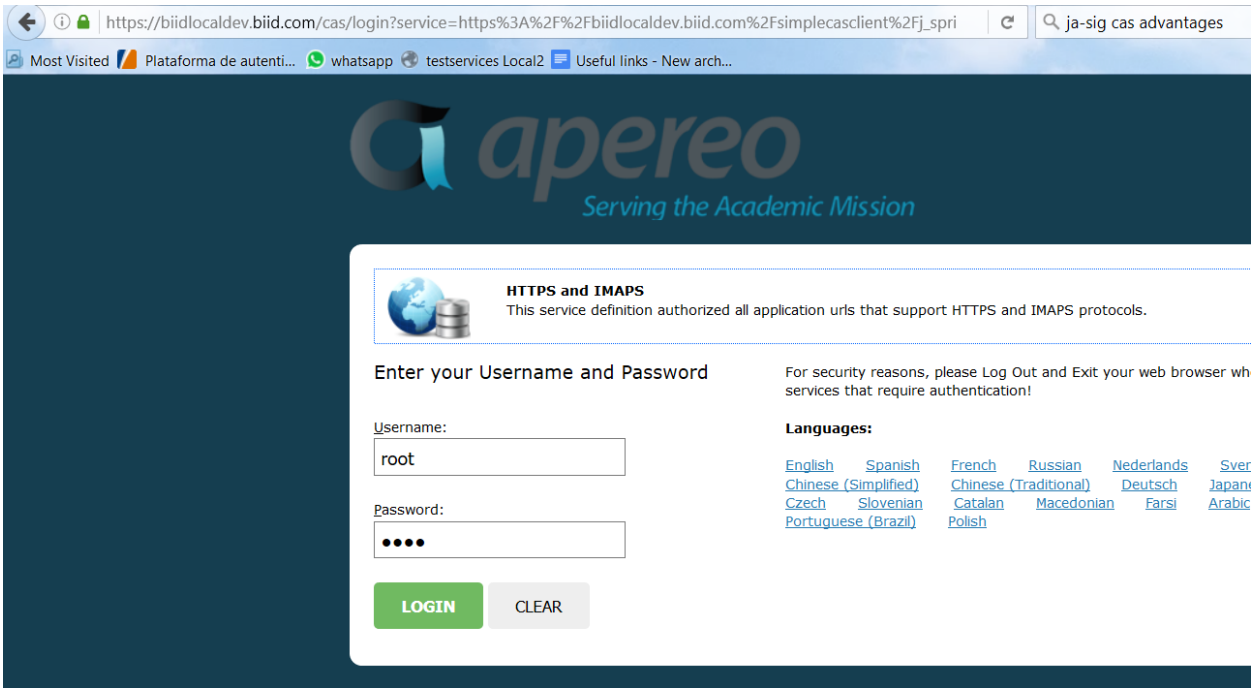
Il·lustració 25

10. Provem d'entrar de nou a l'aplicatiu 1. El sistema ens deixa entrar.



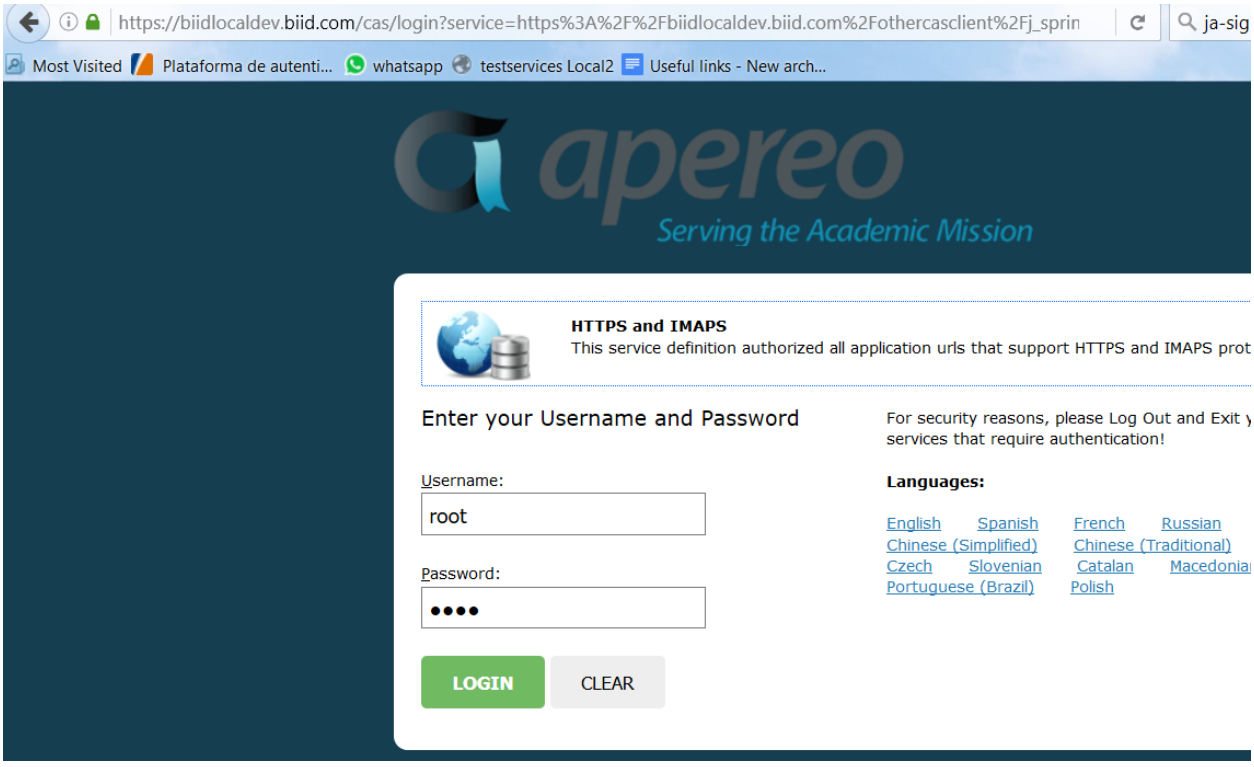
Il·lustració 26

11. Fem logout de l'aplicatiu 1 (botó logout). Provem d'entrar de nou i aquest ens redirigeix a la pantalla de login del CAS



Il·lustració 27

12. Provem d'entrar a l'aplicatiu 2. El sistema ens redirigeix a la pantalla de login del CAS.



Il·lustració 28

Aquest procediment es repeteix pels 4 diferents usuaris que tenim al sistema i amb les dues configuracions (base de dades i arxiu pla), validant el seu funcionament com a usuaris dels aplicatius de prova en funció dels rols que té cadascú en els dos sistemes. En aquesta memòria, ometem la resta de les captures dels següents usuaris, per tal de facilitar la lectura i considerant que no aporten res de nou a la mateixa.

Amb aquest test funcional validem:

- Les dues configuracions pel servidor de CAS (base de dades i arxiu pla) pel que fa a la validació.
- La configuració dels aplicatius clients pel que fa a l'autorització dels usuaris i les restriccions de seguretat
- La correcta configuració del servidor de CAS, ja que ens permet canviar d'aplicació client conservant la sessió i, per tant, no hem de fer login de nou.
- La correcta configuració del servidor de CAS pel que fa al logout, ja que en fer logout d'una aplicació client s'invalida la sessió per totes les aplicacions clients del servidor de CAS (el que s'anomena Single Sign Out)

10 Annex

10.1 Descripció dels arxius adjunts

Apartat eliminat per motius de confidencialitat

10.2 Requisits pels desplegament

Apartat eliminat per motius de confidencialitat

10.3 Desplegament per defecte de la solució:

Apartat eliminat per motius de confidencialitat

10.4 Detalls sobre la configuració per defecte

Apartat eliminat per motius de confidencialitat

10.5 Autenticació mitjançant Certificat digital X509

Apartat eliminat per motius de confidencialitat

11 Referències

http://jasig.github.io/cas/4.1.x/index.html
12 d'abril de 2016

https://en.wikipedia.org/wiki/Jasig
18 d'abril de 2016

http://www.adictosaltrabajo.com/tutoriales/introduccion-cas/
30 d'abril de 2016

http://jasig.github.io/cas/4.1.x/javadocs/license.html
15 de maig de 2016

https://wiki.jasig.org/display/CASUM/X.509+Certificates
13 de abril de 2016

https://es.wikipedia.org/wiki/Apache_License
14 d'abril de 2016

https://es.wikipedia.org/wiki/Requisito_funcional
13 de abril de 2016

https://es.wikipedia.org/wiki/Requisito_no_funcional
13 de abril de 2016

https://es.wikipedia.org/wiki/Caso_de_uso
13 de abril de 2016

https://www.linode.com/docs/websites/apache/apache-web-server-on-ubuntu-14-04
https://github.com/Jasig/cas/releases
14 de maig de 2016

http://dev.mysql.com/downloads/mysql/
18 de maig de 2016

http://tomcat.apache.org/download-70.cgi
30 de maig de 2016

http://www.oracle.com/technetwork/es/java/javase/downloads/index.html
1 de maig de 2016

http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html
1 de maig de 2016

http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf#.VxP-PkfKNj8
20 de maig de 2016

http://apereo.github.io/cas/4.2.x/installation/Database-Authentication.html
1 de juny de 2016

https://github.com/apereo/cas/issues/1691
10 d'abril de 2016

http://comments.gmane.org/gmane.comp.java.jasig.cas.user/20762
17 de maig de 2016

https://groups.google.com/forum/#!topic/jasig-cas-user/H5cosVbGQrg
https://wiki.jasig.org/display/CASUM/Best+Practice+-+Setting+Up+CAS+Locally+using+the+Maven+WAR+Overlay+Method
14 d'abril de 2016