



**PLEC DE PRESCRIPCIONS TÈCNIQUES QUE HA DE REGIR LA
CONTRACTACIÓ DEL SERVEI D'INTEGRACIÓ DE LES APLICACIONS
DE BASE A LA PLATAFORMA DE SIGNATURA ELECTRÒNICA
AUTOMATITZADA (1-2025/0057-GEC)**

ÍNDEX

PRIMERA. OBJECTE DEL CONTRACTE.....	2
SEGONA. DESCRIPCIÓ DEL SERVEI.....	2
2.1. Definicions.....	2
2.2. Objectius	3
2.3. Tipus de servei	3
2.4. Descripció funcional del servei	3
2.4.1. OBJ1. Anàlisi de les aplicacions a adaptar.....	4
2.4.2. OBJ2. Adaptar les aplicacions.....	5
2.4.3. OBJ3. Proves d'Integració.....	5
2.4.4. OBJ4. Documentació	5
2.5. Requeriment tècnics	6
2.6. Equip	7
2.6.1. Interlocutor de l'adjudicatari.....	8
2.6.2. Analista funcional/orgànic.....	8
2.6.3. Programador sènior.....	10

PRIMERA. OBJECTE DEL CONTRACTE

L'objecte del contracte és el servei d'integració de les aplicacions de BASE-Gestió d'Ingressos amb la nova plataforma de signatura electrònica automatitzada. Aquest servei neix de la substitució de la plataforma de signatura electrònica i el desplegament del nou servei signador automatitzat.

SEGONA. DESCRIPCIÓ DEL SERVEI

2.1. Definicions

- **Entorn:** les aplicacions de BASE es despleguen i proven en un entorn de test (Entorn Desenvolupament) amb l'objectiu de validar els desenvolupaments. Seguidament es promocionen a un entorn on l'usuari final pot fer les proves per validar si el desenvolupament es correspon amb les seves necessitats (Entorn Pre-producció). Finalment es promocionen a l'entorn productiu real (Entorn de Producció). Per tant quan parlem d'entorn, fem referència a aquesta conceptualització.
- **Granja:** és el grup de servidors que tenen el mateix objectiu, per exemple la granja empleat, són els servidors que donen servei als clients interns, la granja ciutadà, són els servidors que donen servei als clients externs...
- **Integració.** Acció d'integrar un component A amb un component B.
- **Redmine.** És una eina per a la gestió de projectes, que amb les seves diverses funcionalitats permet als usuaris de diferents projectes realitzar el seguiment i organització dels mateixos.
- **Git.** És una eina de control de versions del codi desenvolupat.
- **REST.** És una interfície per connectar sistemes basats en el protocol HTTP.

2.2. Objectius

Els principals objectius que BASE vol aconseguir amb aquest projecte són els següents:

- **OBJ1. Anàlisi de les aplicacions a adaptar**

Cal detectar les funcionalitats i mètodes de cada aplicació que s'integren amb la actual plataforma de signatura.

- **OBJ2. Adaptar les aplicacions**

Cal adaptar el codi font de cada aplicació per a que totes les funcionalitats i mètodes deixin d'integrar-se amb l'actual plataforma de signatura i ho facin amb el nou servei signador automatitzat.

- **OBJ3. Proves integració**

Cal provar totes les funcionalitats de cada aplicació per tal de comprovar que els documents es signen correctament amb la nova plataforma.

- **OBJ4. Documentació**

A mesura que es van assolint els objectius caldrà documentar tots els canvis aplicats.

2.3. Tipus de servei

Els tipus de servei demanats són els d'anàlisi, implementació i documentació dels objectius que s'han enumerat al punt anterior.

2.4. Descripció funcional del servei

En aquesta secció es presenta una descripció funcional de cadascun dels objectius per tal que el licitador pugui comprendre l'activitat que ha de desenvolupar, a la vegada que quantificar-ne l'esforç.

D'altra banda, cal tenir en compte que BASE – Gestió d'Ingressos té definit i aplica un model de gestió del cicle de vida de desenvolupament del software. Per tant, tots els objectius d'aquest contracte hauran de tenir en compte els requisits del model de gestió de BASE. En "l'Annex1_situacio_actual", es pot trobar aquesta informació.

2.4.1. OBJ1. Anàlisi de les aplicacions a adaptar

Cal analitzar les funcionalitats de cada aplicació i el corresponent codi font per detectar les integracions amb l'actual plataforma signatura.

Com a resultat de la realització d'aquest objectiu es demana un informe per cada aplicació on es vegin els mètodes, classes i serveis que integren amb la plataforma actual. Aquest informe s'utilitzarà com a guio per implementar els evolutius i per fer-ne les provatures.

A continuació es presenten les aplicacions que estan integrades amb el signador automatitzat i que s'han d'analitzar i adaptar:

- RED - Registre Electrònic de Decrets
- RES – Registre d'Entrada i Sortida
- GST - Gestió de Sol·licituds Telemàtiques
- PSE – Portasignatures electrònic
- PEC - Procediments d'Emissió de Certificats
- EBI - Embargament de Bens Immobles
- TIP - Tractament Integral de la Persona
- GDB - Gestió de Domiciliacions Bancàries
- GIM - Gestió Integral de les Multes de Trànsit
- GEA - Gestor d'Expedients d'Arxiu
- GCE - Gestor de Comunicacions Externes
- PST - Presentació de sol·licituds telemàtiques

2.4.2. OBJ2. Adaptar les aplicacions

S'han d'adaptar les classes i serveis de totes les aplicacions de BASE analitzades en l'objectiu anterior per tal que deixin d'invocar a l'actual plataforma de signatura i es faci conta el nou servei REST de signatura automatitzada.

Per fer-ho es detallen els casos d'ús per assolir l'objectiu en "l'Annex RESTPROXY".

L'objectiu es considerarà realitzat quan el codi aportat en el repositori GIT de BASE no doni cap error de compilació i les seves funcionalitats es mantinguin operatives.

2.4.3. OBJ3. Proves d'Integració

Les adaptacions aplicades s'han de desplegar i el licitador haurà de realitzar les proves de la correcta integració i aplicar els correctius necessaris per tal que conservi operatives totes les funcionalitats indicades en l'informe d'anàlisi presentat durant el primer objectiu d'anàlisi.

2.4.4. OBJ4. Documentació

Cal entregar tota la documentació relativa a les tasques realitzades per aconseguir els objectius anteriors. La documentació a entregar és la següent:

- Documentació a l'eina de projectes: cada aplicació té un projecte assignat a l'eina de gestió de projectes. Per fer un correcte seguiment es demana introduir en l'eina les tasques a realitzar, l'estat de les mateixes, les hores dedicades, el temps estimat... A més a més, la informació de caire rellevant i que es preveu que serà consultada de forma recurrent s'ha d'introduir en la wiki del projecte.
- Documentació en format informe: totes les característiques especials, particularitats o incidències que hagin pogut sorgir de la realització de les tasques dels objectius han de quedar explicades en un informe.

2.5. Requeriment tècnics

A continuació es detallen els requeriments tecnològics:

- Entorn de desenvolupament JEE (BASEFrame – Framework JEE de BASE-Gestió d'Ingressos).
- Gestor documental: Documentum.
- Servidors d'aplicacions: Red Hat – Jboss.
- Servidors de base de dades: Informix/Oracle.
- Gestió de projectes/ticketing: Redmine.

Per a la implementació de les històries d'usuari, s'ha de treballar amb:

- Java JDK 1.6.0_29 o superior.
- Maven 3.0.5 o superior.
- Encoding: ISO-8859-15.
- Utilitzar la versió d'Spring posada a les llibreries de BASE.
- Per generar les plantilles: Jasper Reports.
- Per generar la documentació UML: Papyrus.

L'estructura de l'aplicació ha de complir amb l'esquelet definit en l'archetype de BASE, que consta de:

- **Model de domini.** Artefacte on incloem el codi referent al domini. Aquí podem trobar les classes entitats i les interfícies dels mètodes que la capa de presentació utilitza per cridar als mètodes de la lògica de negoci. Aquest projecte ha de ser compartit per la capa de presentació i la lògica de negoci.
- **Lògica.** Aquí es troba la lògica de negoci que implementa la interfície que es troba al domini i els mètodes DAO que fan els accessos a la base de dades.
- **Recursos.** Conté els arxius de recursos per a la resolució de la connexió amb el servei (EJB, Enterprise JavaBean) i els arxius de missatges, si n'hi ha, dels errors específics de la lògica. Aquest mòdul és opcional i només és necessari en cas que el projecte publiqui un servei o hi hagi missatges d'error específics.
- **UI.** Conté la capa de presentació.
- **EAR.** Conté la resta d'artefactes per a la seva distribució unificada.
- **Test.** Subprojecte opcional i que depèn del tipus de projecte. Normalment, en els projectes de serveis_d'arquitectura, s'incorpora aquest subprojecte per fer



simulacions del servei en situacions en què no és pràctic definir-les com a un test unitari/integració. Aquest tipus de projectes de test estan orientats a poder invocar un servei també amb el propòsit d'iniciar una tasca de la qual no es té encara desenvolupat el front-end. El projecte normalment consta d'una única classe principal *main* i aconsegueix les referències als servei a fer les proves per el JNDI (Java Naming and Directory Interface).

2.6. Equip

L'equip requerit per BASE consisteix en tres perfils clau: el d'interlocutor de l'adjudicatari, el d'analista funcional/orgànic i el d'analista de sistemes.

La composició de l'equip i la distribució orientativa dels perfils bàsics del projecte ha de ser de la següent manera:

Perfil	Nombre
Interlocutor de l'adjudicatari	1
Analista funcional/orgànic	1
Programador	1

El cap de projecte de BASE és l'encarregat de vetllar que la documentació i la planificació compleixen amb els requisits establerts.

És responsabilitat de l'empresa adjudicatària l'assoliment de totes les fites requerides per executar cadascuna de les tasques comunicades i prioritzades pel cap de projecte de BASE, des de la presa de requeriments fins a la coordinació del desplegament a producció.

Tots els treballs s'han de realitzar de forma remota via VPN en les pròpies oficines de l'adjudicatari.

2.6.1. Interlocutor de l'adjudicatari

L'interlocutor de l'adjudicatari és la figura clau en la planificació, execució i control del projecte i és qui ha d'impulsar l'avenç d'aquest mitjançant la presa de decisions necessàries per assolir els objectius. De la mateixa manera és l'encarregat de coordinar-se i trobar les solucions adequades, adaptades a l'estructura actual de BASE i acceptades pel cap de Serveis Informàtics.

Tasques a realitzar per l'interlocutor de l'adjudicatari:

- Col·laborar amb el cap de projecte de BASE en la definició i concreció dels objectius del projecte.
- Planificar el projecte en tots els seus aspectes, identificant les activitats a realitzar, els recursos a destinar, els terminis i els costos previstos.
- Dirigir i coordinar tots els recursos destinats al projecte. En cas de mancances, elevar-les al cap de Serveis Informàtics per trobar una solució.
- Gestionar la demanda de peticions mitjançant l'eina Redmine.
- Ser el responsable d'obtenir l'aprovació de les definicions per part del cap de Serveis Informàtics.
- Prendre les decisions necessàries per conèixer en tot moment la situació en relació amb els objectius previstos.
- Adoptar les mesures correctores pertinents per redirigir les desviacions que s'hagin detectat.
- En cas que el cap de projecte de BASE hagi sol·licitat reunions de seguiment, el interlocutor de l'adjudicatari ha de realitzar-ne les actes.
- En cas que el cap de projecte de BASE requereixi un Informe per detallar els motius d'una desviació o d'alguna problemàtica associada al projecte, l'interlocutor de l'adjudicatari és qui l'ha de redactar.

2.6.2. Analista funcional/orgànic

L'analista funcional/orgànic ha d'identificar i donar resposta a les necessitats funcionals de BASE amb l'elaboració i execució del projecte, complint amb els estàndards i la

documentació requerida per la Metodologia de gestió de projectes de BASE (especificada a BASEFrame), seguint la normativa de control de la qualitat determinat pel cap de projecte de BASE.

En els requeriments funcionals i en els requeriments tècnics definits en aquest *Plec* i en els seus *Annexos*, BASE especifica què vol aconseguir, com ho vol aconseguir i amb quin objectiu ho vol aconseguir. A partir d'aquesta documentació, el perfil d'analista funcional i orgànic ha de ser capaç de complir amb les següents competències:

- Entendre els requeriments funcionals i requeriments tècnics sent-hi crític amb l'objectiu d'aportar qualsevol millora.
- Realitzar l'anàlisi tècnic des d'una perspectiva funcional, és a dir, realitzar diagrames d'arquitectura lògics, diagrames de seqüència de les operacions, diagrames conceptuals...
- Aplicar tot el que ha estat documentat i diagramat en la implantació de la solució, és a dir, que la documentació s'ha de correspondre 100x100 amb la realitat.

Més específicament i associades a les competències esmentades, aquest mateix perfil ha de realitzar les següents tasques:

- Ser l'encarregat d'agafar els requeriments del cap de projecte de BASE i plasmar-los en la solució *objectiu*. Per realitzar aquesta tasca, és important adequar el vocabulari i el nivell de detall del discurs (menys tecnicismes, menys nivell de detall...) amb l'objectiu de facilitar la comprensió.
- Gaudir d'autonomia de decisió, capaç de decidir si una petició es pot realitzar, si es disposen de les eines necessàries i apropiades i, finalment, si és la millor manera de donar-li cobertura.
- Ha de realitzar una completa descripció de cada unitat funcional i en conseqüència orgànica. També ha de definir els mètodes de control per evitar errades de la solució proposada.
- Detectar, en la mesura del possible, eventuais omissions en la petició del cap de projecte de BASE i comunicar-les per consignar-les en la petició identificada.
- Validar i obtenir l'aprovació per part del cap de projecte de BASE.

- Liderar les reunions amb BASE exposant i plantejant els punts a tractar i, un cop acabades les reunions, assignar les tasques a cadascuna de les parts implicades, si escau.
- Revisar les estimacions.
- Entendre els conceptes de la metodologia àgil de BASE i documentar els diferents aspectes del projecte segons els principis d'aquesta i seguint els requisits del cap de projecte de BASE.
- Definir els casos de prova.
- Guiar l'usuari en l'execució dels casos de prova.
- Proposar les necessitats de millores.
- Administrar els canvis.
- Analitzar alternatives de les possibles implementacions.
- Definir les parametritzacions.
- Implementar la solució tot buscant el consens amb l'equip de desenvolupament de BASE.
- Fer els canvis de codificació i de configuració necessaris per satisfer els objectius del contracte.
- Definir proves funcionals.
- Realitzar proves funcionals.
- Donar el suport post implantació.

2.6.3. Programador sènior

El programador sènior ha d'escriure, depurar i mantenir el codi font complint amb els estàndards de BASE, fent ús del framework de desenvolupament de BASE i seguint la normativa de qualitat determinat pel cap de projecte de BASE.

Tasques a realitzar pel programador sènior:

- Implementar les especificacions de l'analista funcional/orgànic en codi executable.
- Fer ús de les funcionalitats del framework de desenvolupament de BASE.
- Definir les proves unitàries del software.



- Implementar i executar les proves unitàries.
- Detectar eventuais omissions en les definicions de l'analista funcional/orgànic.
- Comunicar-se constantment amb l'analista funcional amb l'objectiu d'evitar desviacions innecessàries.

Carles Casas Vidal
El cap de Servei de Serveis Informàtics

BASE

GESTIÓ D'INGRESSOS



Diputació Tarragona

Servei de Serveis Informàtics

C. Pere Martell, 2

43001 Tarragona

Tel. 977 253 414

Fax 977 258 451

Annex Situació actual

Sistema d'Integració Continua (SIC)

BASE

Gestió d'Ingressos

*Sistema d'Integració Continua (SIC)***Informació general**

Control documental				
Projecte		Sistema d'Integració Continua.		
Entitat destí		BASE – Gestió d'Ingressos, Departament de Serveis Informàtics. Unitat de Desenvolupament		
Codi de referència		SIC		
Versió		v1.0.0		
Data de creació		28/11/23		
Fitxer		Annex_situació_actual-v1.odt		
Format publicació		Annex_situació_actual-v1.pdf		
Autor		Jordi Samarra Alegre		
Estat formal				
Preparat		Revisat		Aprovat
Nom: Jordi Samarra Alegre Data: 27/11/23		Nom: Data:		Nom: Data:
Control de versions				
Versió	Parts que canvien	Descripció del canvi	Autor del canvi	Data del canvi
v1.0	Original	Creació document	Jordi Samarra	27/11/23
Llicència				
Creative Commons		Reconeixement – No comercial – Sense obra derivada 3.0 Espanya		

*Sistema d'Integració Continua (SIC)***Índex de continguts**

Informació general.....	2
1 Introducció.....	4
2 Situació actual.....	4

Índex d'il·lustracions

Il·lustració 2 1: Diagrama conceptual situació actual.....	6
--	---

Sistema d'Integració Continua (SIC)

1 Introducció

Aquest document neix de la necessitat d'exposar el nou entorn de desenvolupament de BASE amb l'objectiu que el licitador la conegui així com la metodologia de treball del Departament de Serveis Informàtics.

2 Situació actual

Els tècnics de BASE accedeixen al codi font de les diferents aplicacions emmagatzemades al Git de BASE.

Els proveïdors en entregar el codi al repositori Git de BASE el SIC compila el projecte modificat.

D'aquesta manera el proveïdor sap si el canvi aplicat ha compilat correctament i ha passat els tests.

En el cas que la compilació sigui correcta el Jenkins activa el pipeline de l'aplicació modificada per tal de preparar el desplegament. A més a més, emmagatzema els artefactes en repositori Sonatype Nexus i a Sonarqube per a que analitzi la qualitat del codi.

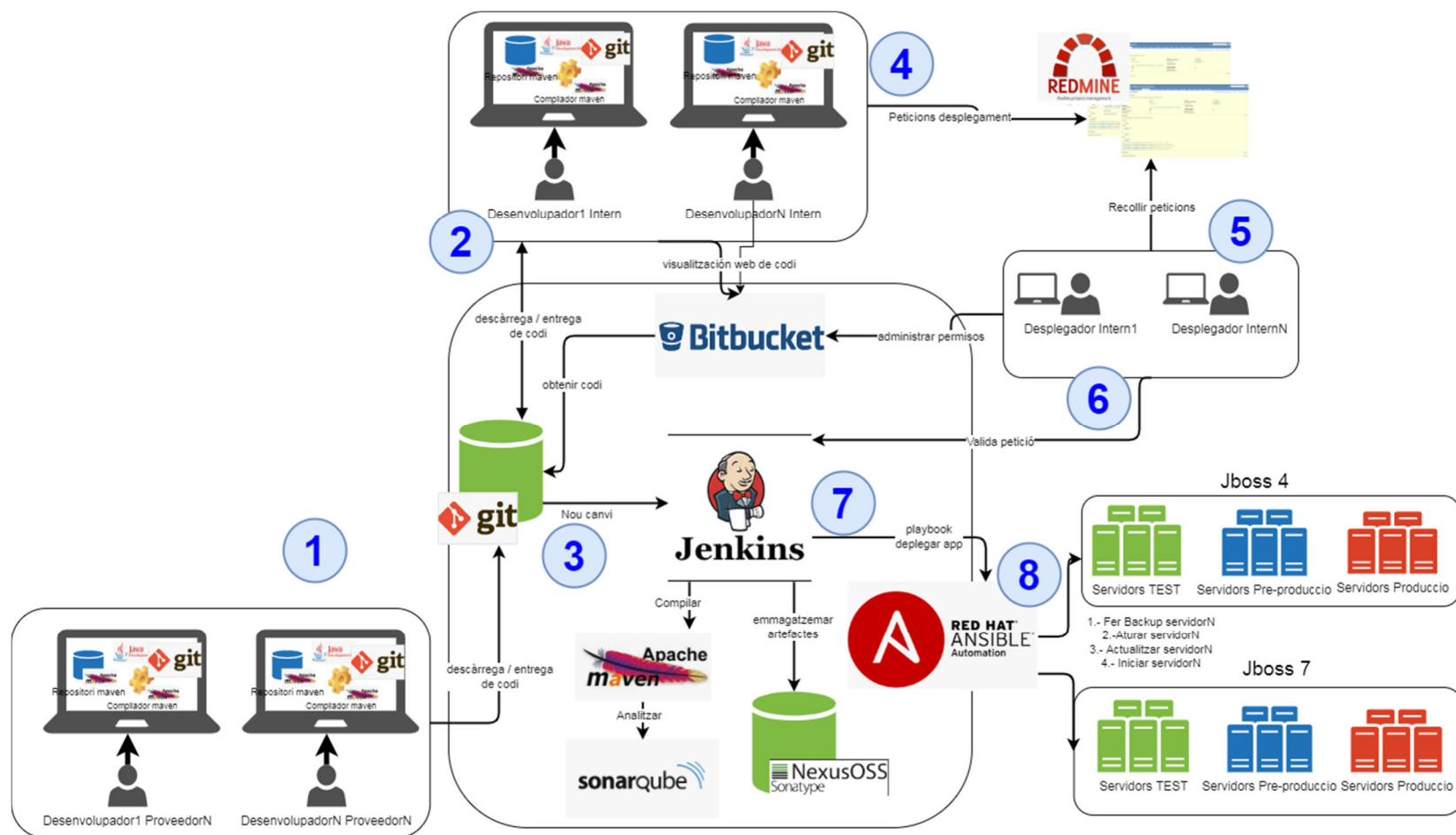
Els responsables dels desplegaments gestionen quan es despleguen segons criteris d'eficiència en el treball i prèvia autorització del responsable.

Els responsables dels projectes poden demanar el desplegament als responsables de desplegament a través de la creació d'un assumpte tipus desplegament en el gestor de projectes Redmine.

A continuació Jenkins a través d'Ansible desplega els artefactes reiniciant els servidors i aplicant variables d'entorn.

Finalment els tècnics de BASE comproven les noves funcionalitats i canvis aplicats, retornaran els resultats als proveïdors a través del Redmine.

A continuació es presenta un digrama de la situació actual on es pot veure tot el comentat anteriorment.



Il·lustració 2 1: Diagrama conceptual situació actual.

Annex Manual de casos d'ús de REST Proxy

Plataforma Signatura Automatitzada

BASE

Gestió d'Ingressos

INFORMACIÓ GENERAL

Control documental				
Projecte		Plataforma Signatura Automatitzada		
Entitat destí		BASE – Gestió d'Ingressos, Departament de Serveis Informàtics. Unitat de Desenvolupament		
Codi de referència		PSA		
Versió		v1.0.0		
Data de creació		1/2/24		
Fitxer		Annex RESTPROXY.odt		
Format publicació		Annex RESTPROXY.pdf		
Autor		Jordi Samarra Alegre		
Estat formal				
Preparat		Revisat		Aprovat
Nom: Jordi Samarra Alegre Data: 1/2/24		Nom: Data:		Nom: Data:
Control de versions				
Versió	Parts que canvien	Descripció del canvi	Autor del canvi	Data del canvi
v1.0	Original	Creació document	Jordi Samarra	1/2/24
Llicència				
Creative Commons		Reconeixement – No comercial – Sense obra derivada 3.0 Espanya		



TAULA DE CONTINGUTS

Informació general.....	2
2 Propòsit d'aquest document.....	4
3 Signatura de documents.....	5
3.1 Descripció.....	5
3.2 Casos de Prova.....	5
3.2.1 Paràmetres comuns.....	5
3.2.2 Signatura de Binaris (CAAdES).....	5
3.2.3 Signatura de XML (XAdES).....	6
3.2.4 Signatura de Binaris (CAAdES).....	8
4 Timestamp.....	10
4.1 Descripció.....	10
4.2 Cas de Prova (Parametrizació).....	10
5 Verificació de Certificats.....	11
5.1 Descripció.....	11
5.2 Cas de Prova (Parametrizació).....	11

1 PROPÒSIT D'AQUEST DOCUMENT

Aquest document es crea per proporcionar la parametrització bàsica necessària per fer crides als diferents serveis de la plataforma de signatura automatitzada Proxy REST de Base Tarragona.

S'ha creat un projecte amb les implementacions de referència necessàries i la parametrització bàsica en mètodes que proven un ampli repertori de firmes amb la verificació corresponent així com amb la verificació d'un Certificat i l'obtenció d'un Timestamp.



```
15
16 public SignarTest(String sUrl) {
17     super(sUrl);
18 }
19
20 public static void all() throws Exception {
21
22     SignarTest objSignarTest = new SignarTest(PATH_SIGN);
23
24     System.out.println("TESTS DE FIRMA");
25     System.out.println("-----");
26     objSignarTest.signar_CADES_EIDAS_B_LEVEL("Signar" - 0101, "signar_CADES_EIDAS_B_LEVEL");
27     System.out.println("-----");
28     objSignarTest.signar_CADES_EIDAS_B_LEVEL_POLITICA("Signar" - 0102, "signar_CADES_EIDAS_B_LEVEL_POLITICA");
29     System.out.println("-----");
30     objSignarTest.signar_CADES_EIDAS_T_LEVEL("Signar" - 0103, "signar_CADES_EIDAS_T_LEVEL");
31     System.out.println("-----");
32     objSignarTest.signar_CADES_EIDAS_LT_LEVEL("Signar" - 0104, "signar_CADES_EIDAS_LT_LEVEL");
33     System.out.println("-----");
34     objSignarTest.signar_CADES_EIDAS_LTA_LEVEL("Signar" - 0105, "signar_CADES_EIDAS_LTA_LEVEL");
35     System.out.println("-----");
36     objSignarTest.signar_CADES_BES("Signar" - 0106, "signar_CADES_BES");
37     System.out.println("-----");
38     objSignarTest.signar_CADES_XLONG("Signar" - 0107, "signar_CADES_XLONG");
39
40     //
41     System.out.println("-----");
42     objSignarTest.signar_XADES_EIDAS_B_LEVEL("Signar" - 0201, "signar_XADES_EIDAS_B_LEVEL");
43     System.out.println("-----");
44     objSignarTest.signar_XADES_EIDAS_B_LEVEL_POLITICA("Signar" - 0202, "signar_XADES_EIDAS_B_LEVEL_POLITICA");
45     System.out.println("-----");
46     objSignarTest.signar_XADES_EIDAS_T_LEVEL("Signar" - 0203, "signar_XADES_EIDAS_T_LEVEL");
47     System.out.println("-----");
48     objSignarTest.signar_XADES_EIDAS_LT_LEVEL("Signar" - 0204, "signar_XADES_EIDAS_LT_LEVEL");
49     System.out.println("-----");
50 }
```

cat.base.usignerproxy.test.testCases

- SignarTest
- PATH_SIGN: String
- SignarTest(String)
- all(): void
- signar_CADES_EIDAS_B_LEVEL(String, String): void
- signar_CADES_EIDAS_B_LEVEL_POLITICA(String, String): void
- signar_CADES_EIDAS_T_LEVEL(String, String): void
- signar_CADES_EIDAS_LT_LEVEL(String, String): void
- signar_CADES_EIDAS_LTA_LEVEL(String, String): void
- signar_CADES_BES(String, String): void
- signar_CADES_XLONG(String, String): void
- signar_XADES_EIDAS_B_LEVEL_POLITICA(String, String): void
- signar_XADES_EIDAS_T_LEVEL(String, String): void
- signar_XADES_EIDAS_LTA_LEVEL(String, String): void
- signar_XADES_141_BES(String, String): void
- signar_XADES_141_XLONG(String, String): void
- signar_PADES_EIDAS_B_LEVEL(String, String): void
- signar_PADES_EIDAS_B_LEVEL_POLITICA(String, String): void
- signar_PADES_EIDAS_T_LEVEL(String, String): void
- signar_PADES_EIDAS_LT_LEVEL(String, String): void
- signar_PADES_EIDAS_LTA_LEVEL(String, String): void
- signar_PADES_BES(String, String): void
- signar_PADES_LTI(String, String): void

En aquest document es detallen els paràmetres mínims esperats per plataforma signatura automatitzada Proxy REST per fer cadascuna de les operacions.

2 SIGNATURA DE DOCUMENTS

2.1 Descripció

El Mòdul Servidor de Signatures (d'ara endavant, SignatureServer) signa dades i documents en diferents formats i amb diferents algorismes, i verifica firmes realitzades en qualsevol d'aquests formats i amb qualsevol d'aquests algorismes.

El Servidor de Signatures és capaç de generar signatura múltiple seqüencial i paral·lela, tant directa com diferida.

Aquest mòdul fa ús dels mòduls PolicyManager per obtenir els certificats vàlids per signar, els prestadors vàlids per verificar, etc., X509Validator per verificar l'estat dels certificats implicats en les operacions de signatura i verificació i NonRepudiation per registrar totes les operacions de verificació de signatures realitzades.

2.2 Casos de Prova

2.2.1 Paràmetres comuns

```
1 invokingApp
2 setOperationTypeId
3 setDataEncodedInBase64
```

2.2.2 Signatura de Binaris (CADES)

Signatura CADES EIDAS B_LEVEL

```
1 objParamsSignCADESInfo.setDataType(com.tb solutions.asf.util.Constants.CADES_EIDAS_FORMAT);
2 objParamsSignCADESInfo.setFormatSubType(com.tb solutions.asf.util.Constants.B_LEVEL);
```

Signatura CADES EIDAS B_LEVEL amb política

```
1 objParamsSignCADESInfo.setDataType(com.tb solutions.asf.util.Constants.CADES_EIDAS_FORMAT);
2 objParamsSignCADESInfo.setFormatSubType(com.tb solutions.asf.util.Constants.B_LEVEL);
3
4 ParamsSignaturePolicyIdInfo poliyInfoCADES = new ParamsSignaturePolicyIdInfo();
5 poliyInfoCADES.setSigPolicyId(policyId);
6 objParamsSignCADESInfo.setSignaturePolicyIdInfo(poliyInfoCADES);
```

Signatura CADES EIDAS T_LEVEL

```
1 objParamsSignCADESInfo.setDataType(com.tb solutions.asf.util.Constants.CADES_EIDAS_FORMAT);
2 objParamsSignCADESInfo.setFormatSubType(com.tb solutions.asf.util.Constants.T_LEVEL);
```

Firma CADES EIDAS LT_LEVEL

```
1objParamsSignCADESInfo.setDataType(com.tbsolutions.asf.util.Constants.CADES_EIDAS_FORMAT);
2objParamsSignCADESInfo.setFormatSubType(com.tbsolutions.asf.util.Constants.LT_LEVEL);
```

Signatura CADES EIDAS LTA_LEVEL

```
1objParamsSignCADESInfo.setDataType(com.tbsolutions.asf.util.Constants.CADES_EIDAS_FORMAT);
2objParamsSignCADESInfo.setFormatSubType(com.tbsolutions.asf.util.Constants.LTA_LEVEL);
```

Signatura CADES BES

```
1objParamsSignCADESInfo.setDataType(com.tbsolutions.asf.util.Constants.CADES_FORMAT);
2objParamsSignCADESInfo.setFormatSubType(com.tbsolutions.asf.signatureserver.util.Constants.CADES_BES);
3objParamsSignCADESInfo.setAttachedCertPathLength(com.tbsolutions.asf.util.Constants.INCLUDE_SIGNING_CERT);
```

Signatura CADES XLONG

```
1objParamsSignCADESInfo.setDataType(com.tbsolutions.asf.util.Constants.CADES_FORMAT);
2objParamsSignCADESInfo.setFormatSubType(com.tbsolutions.asf.signatureserver.util.Constants.CADES_XLONG);
3objParamsSignCADESInfo.setTimeStamp(true);
4objParamsSignCADESInfo.setAttachedCertPathLength(com.tbsolutions.asf.util.Constants.INCLUDE_FULL_PATH);
5objParamsSignCADESInfo.setVerifySigningCerts(com.tbsolutions.asf.signatureserver.util.Constants.SIGNING_CERT);
```

2.2.3 Signatura de XML (XAdES)

Signatura XAdES EIDAS B_LEVEL

```
1objParamsSignXADESInfo.setDataType(com.tbsolutions.asf.util.Constants.XADES_EIDAS_FORMAT);
2objParamsSignXADESInfo.setSignSubType(com.tbsolutions.asf.util.Constants.B_LEVEL);
3
4ParamsXAdESDataObjectFormatInfo oParamsXAdESDataObjectFormatInfo = new ParamsXAdESDataObjectFormatInfo();
5oParamsXAdESDataObjectFormatInfo.setDataObjectFormatMimeType("application/octet-stream");
6oParamsXAdESDataObjectFormatInfo.setDataObjectFormatEncoding("http://www.w3.org/2000/09/xmldsig#base64");
7objParamsSignXADESInfo.setDataObjectFormatInfo(new ParamsXAdESDataObjectFormatInfo[] {oParamsXAdESDataObjectFormatInfo});
8
9// "ATTACHEDENVELOPED_PAR",
10objParamsSignXADESInfo.setSignType(com.tbsolutions.asf.signatureserver.util.Constants.ATTACHEDENVELOPED);
11objParamsSignXADESInfo.setSequential(false);
12objParamsSignXADESInfo.setXPathsSignatureTargets(new String[]{""});
13objParamsSignXADESInfo.setIdsToSign(null);
14objParamsSignXADESInfo.setExternalURLData(null);
```

Signatura XAdES EIDAS B_LEVEL amb política

```
1objParamsSignXADESInfo.setDataType(com.tbsolutions.asf.util.Constants.XADES_EIDAS_FORMAT);
2objParamsSignXADESInfo.setSignSubType(com.tbsolutions.asf.util.Constants.B_LEVEL);
3
4ParamsXAdESDataObjectFormatInfo oParamsXAdESDataObjectFormatInfo = new ParamsXAdESDataObjectFormatInfo();
5oParamsXAdESDataObjectFormatInfo.setDataObjectFormatMimeType("application/octet-stream");
6oParamsXAdESDataObjectFormatInfo.setDataObjectFormatEncoding("http://www.w3.org/2000/09/xmldsig#base64");
7objParamsSignXADESInfo.setDataObjectFormatInfo(new ParamsXAdESDataObjectFormatInfo[] {oParamsXAdESDataObjectFormatInfo});
```



```
8
9// "ATTACHEDENVELOPED_PAR",
10objParamsSignXADESInfo.setSignType(com.tbtsolutions.asf.signatureserver.util.Constants.ATTACHEDENVELOPED);
11objParamsSignXADESInfo.setSequential(false);
12objParamsSignXADESInfo.setXPathsSignatureTargets(new String[]{""});
13objParamsSignXADESInfo.setIdsToSign(null);
14objParamsSignXADESInfo.setExternalURLData(null);
15
16ParamsXADESPolicyInfo policyInfoXADES = new ParamsXADESPolicyInfo();
17policyInfoXADES.setPolicyURI(policyURI);
18objParamsSignXADESInfo.setPolicyInfo(policyInfoXADES);
```

Signatura XAdES EIDAS T_LEVEL

```
1objParamsSignXADESInfo.setDataType(com.tbtsolutions.asf.util.Constants.XADES_EIDAS_FORMAT);
2objParamsSignXADESInfo.setSignSubType(com.tbtsolutions.asf.util.Constants.T_LEVEL);
3
4ParamsXADESDataObjectFormatInfo oParamsXADESDataObjectFormatInfo = new ParamsXADESDataObjectFormatInfo();
5oParamsXADESDataObjectFormatInfo.setDataObjectFormatMimeType("application/octet-stream");
6oParamsXADESDataObjectFormatInfo.setDataObjectFormatEncoding("http://www.w3.org/2000/09/xmldsig#base64");
7objParamsSignXADESInfo.setDataObjectFormatInfo(new ParamsXADESDataObjectFormatInfo[] {oParamsXADESDataObjectFormatInfo});
8
9// "ATTACHEDENVELOPED_PAR",
10objParamsSignXADESInfo.setSignType(com.tbtsolutions.asf.signatureserver.util.Constants.ATTACHEDENVELOPED);
11objParamsSignXADESInfo.setSequential(false);
12objParamsSignXADESInfo.setXPathsSignatureTargets(new String[]{""});
13objParamsSignXADESInfo.setIdsToSign(null);
14objParamsSignXADESInfo.setExternalURLData(null);
```

Signatura XAdES EIDAS LT_LEVEL

```
1objParamsSignXADESInfo.setDataType(com.tbtsolutions.asf.util.Constants.XADES_EIDAS_FORMAT);
2objParamsSignXADESInfo.setSignSubType(com.tbtsolutions.asf.util.Constants.LT_LEVEL);
3
4ParamsXADESDataObjectFormatInfo oParamsXADESDataObjectFormatInfo = new ParamsXADESDataObjectFormatInfo();
5oParamsXADESDataObjectFormatInfo.setDataObjectFormatMimeType("application/octet-stream");
6oParamsXADESDataObjectFormatInfo.setDataObjectFormatEncoding("http://www.w3.org/2000/09/xmldsig#base64");
7objParamsSignXADESInfo.setDataObjectFormatInfo(new ParamsXADESDataObjectFormatInfo[] {oParamsXADESDataObjectFormatInfo});
8
9// "ATTACHEDENVELOPED_PAR",
10objParamsSignXADESInfo.setSignType(com.tbtsolutions.asf.signatureserver.util.Constants.ATTACHEDENVELOPED);
11objParamsSignXADESInfo.setSequential(false);
12objParamsSignXADESInfo.setXPathsSignatureTargets(new String[]{""});
13objParamsSignXADESInfo.setIdsToSign(null);
14objParamsSignXADESInfo.setExternalURLData(null);
```

Signatura XAdES EIDAS LTA_LEVEL

```
1objParamsSignXADESInfo.setDataType(com.tbtsolutions.asf.util.Constants.XADES_EIDAS_FORMAT);
2objParamsSignXADESInfo.setSignSubType(com.tbtsolutions.asf.util.Constants.LTA_LEVEL);
3
4ParamsXADESDataObjectFormatInfo oParamsXADESDataObjectFormatInfo = new ParamsXADESDataObjectFormatInfo();
5oParamsXADESDataObjectFormatInfo.setDataObjectFormatMimeType("application/octet-stream");
6oParamsXADESDataObjectFormatInfo.setDataObjectFormatEncoding("http://www.w3.org/2000/09/xmldsig#base64");
```



```
ParamsXAdESDataObjectFormatInfo[] {oParamsXAdESDataObjectFormatInfo});
```

```
// "ATTACHEDENVELOPED_PAR",
objParamsSignXAdESInfo.setSignType(com.tbtsolutions.asf.signatureserver.util.Constants.ATTACHEDENVELOPED);
objParamsSignXAdESInfo.setSequential(false);
objParamsSignXAdESInfo.setXPathsSignatureTargets(new String[]{""});
objParamsSignXAdESInfo.setIdsToSign(null);
objParamsSignXAdESInfo.setExternalURLData(null);
```

Signatura XAdES 141 BES

```
1objParamsSignXAdESInfo.setDataType(com.tbtsolutions.asf.util.Constants.XAdES_1_4_1_FORMAT);
2objParamsSignXAdESInfo.setSignSubType(com.tbtsolutions.asf.signatureserver.util.Constants.XAdES_BES);
3
4// "ATTACHEDENVELOPED_PAR",
5objParamsSignXAdESInfo.setSignType(com.tbtsolutions.asf.signatureserver.util.Constants.ATTACHEDENVELOPED);
6objParamsSignXAdESInfo.setSequential(false);
7objParamsSignXAdESInfo.setXPathsSignatureTargets(new String[]{""});
8objParamsSignXAdESInfo.setIdsToSign(null);
9objParamsSignXAdESInfo.setExternalURLData(null);
```

Signatura XAdES 141 XLONG

```
1objParamsSignXAdESInfo.setDataType(com.tbtsolutions.asf.util.Constants.XAdES_1_4_1_FORMAT);
2objParamsSignXAdESInfo.setSignSubType(com.tbtsolutions.asf.signatureserver.util.Constants.XAdES_XLONG);
3
4// "ATTACHEDENVELOPED_PAR",
5objParamsSignXAdESInfo.setSignType(com.tbtsolutions.asf.signatureserver.util.Constants.ATTACHEDENVELOPED);
6objParamsSignXAdESInfo.setSequential(false);
7objParamsSignXAdESInfo.setXPathsSignatureTargets(new String[]{""});
8objParamsSignXAdESInfo.setIdsToSign(null);
9objParamsSignXAdESInfo.setExternalURLData(null);
10objParamsSignXAdESInfo.setExternalURLData(null);
1110
1211objParamsSignXAdESInfo.setTimeStamp(true);
1312objParamsSignXAdESInfo.setVerifySigningCerts(com.tbtsolutions.asf.signatureserver.util.Constants.SIGNING_CERTS);
1413objParamsSignXAdESInfo.setAttachedCertPathLength(com.tbtsolutions.asf.util.Constants.INCLUDE_FULL_PATH);
```

2.2.4 Signatura de Binaris (CAdES)

Signatura PAdES EIDAS B_LEVEL

```
1objParamsSignPAdESInfo.setDataType(com.tbtsolutions.asf.util.Constants.PAdES_EIDAS_FORMAT);
2objParamsSignPAdESInfo.setFormatSubType(com.tbtsolutions.asf.util.Constants.B_LEVEL);
```

Signatura PAdES EIDAS B_LEVEL con política

```
1objParamsSignPAdESInfo.setDataType(com.tbtsolutions.asf.util.Constants.PAdES_EIDAS_FORMAT);
2objParamsSignPAdESInfo.setFormatSubType(com.tbtsolutions.asf.util.Constants.B_LEVEL);
3
4ParamsSignaturePolicyIdInfo poliyInfoPAdES = new ParamsSignaturePolicyIdInfo();
```



```
5polyInfoPADES.setSigPolicyId(policyId);
6objParamsSignPADESInfo.setSignaturePolicyIdInfo(polyInfoPADES);
```

Signatura PAdES EIDAS T_LEVEL

```
1objParamsSignPADESInfo.setDataType(com.tbsolutions.asf.util.Constants.PADES_EIDAS_FORMAT);
2objParamsSignPADESInfo.setFormatSubType(com.tbsolutions.asf.util.Constants.T_LEVEL);
```

Signatura PAdES EIDAS LT_LEVEL

```
1objParamsSignPADESInfo.setDataType(com.tbsolutions.asf.util.Constants.PADES_EIDAS_FORMAT);
2objParamsSignPADESInfo.setFormatSubType(com.tbsolutions.asf.util.Constants.LT_LEVEL);
```

Signatura PAdES EIDAS LTA_LEVEL

```
1objParamsSignPADESInfo.setDataType(com.tbsolutions.asf.util.Constants.PADES_EIDAS_FORMAT);
2objParamsSignPADESInfo.setFormatSubType(com.tbsolutions.asf.util.Constants.LTA_LEVEL);
```

Signatura PAdES BES

```
1objParamsSignPADESInfo.setDataType(com.tbsolutions.asf.util.Constants.PADES_FORMAT);
2objParamsSignPADESInfo.setFormatType(com.tbsolutions.asf.signatureserver.util.Constants.PADES_ENHANCED_BES);
```

Signatura PAdES LTV

```
1objParamsSignPADESInfo.setDataType(com.tbsolutions.asf.util.Constants.PADES_FORMAT);
2objParamsSignPADESInfo.setFormatType(com.tbsolutions.asf.signatureserver.util.Constants.PADES_ENHANCED_BES);
3objParamsSignPADESInfo.setAdvancedFormatSubType(com.tbsolutions.asf.signatureserver.util.Constants.PADES_LTV);
4objParamsSignPADESInfo.setTimeStamp(true);
5objParamsSignPADESInfo.setAttachedCertPathLength(com.tbsolutions.asf.util.Constants.INCLUDE_FULL_PATH);
6objParamsSignPADESInfo.setVerifySigningCerts(com.tbsolutions.asf.signatureserver.util.Constants.SIGNING_CERT);
```

3 TIMESTAMP

3.1 Descripció

El Mòdul de TimeStampClient ofereix una senzilla interfície basada en Web Services (SOAP) per obtenir segells de temps, amaga la complexitat de les autoritats de segellat concentrant en un únic punt tota la funcionalitat del segellat de temps.

Es permet una configuració completa a través de la consola d'administració d'U Signer, tant dels certificats amb què signen les respectives Autoritats de Sellado (TSA en endavant) com els certificats necessaris per a la signatura de peticions en cas que les TSAs exigeixin la signatura de les mateixes

3.2 Cas de Prova (Parametrizació)

```
HashMap<String, Object> objHashMapInput = new HashMap<String, Object>();

objHashMapInput.put("tsaClientPolicyAlias", TestConstants.ALIAS_TSA);
String document = FileUtils.readFileInB64(TestConstants.PATH_UNSIGNED_DOCUMENTS +
TestConstants.NAME_UNSIGNED_XML);
objHashMapInput.put("messageToHash", document);
objHashMapInput.put("idOperacio", TestConstants.CONSOLE_SIGNATURE_OPERATION);
objHashMapInput.put("idAplicacio", TestConstants.CONSOLE_SIGNATURE_APPLICATION);

objHashMapInput.put("protocol", "HTTP");
objHashMapInput.put("includeTimestampCertificateAndRevocationValuesIn", 2);

String input = JsonUtils.convertMapToJSON(objHashMapInput);
```

4 VERIFICACIÓ DE CERTIFICATS

4.1 Descripció

El mòdul de Validació de Certificats X.509 (d'ara endavant, X509Validator) és l'encarregat de validar tant el període de validesa com l'estat de revocació de certificats X.509.

El mòdul abasta diversos protocols de validació de l'estat de revocació, OCSP, CRL mitjançant LDAP, HTTP i explotació de tota la informació disponible als propis certificats (Punts de Distribució de CRLS i Informació d'Accés a l'Autoritat amb URLs de servidors OCSP).

Aquest mòdul fa ús del mòdul PolicyManager per obtenir els prestadors de confiança, els mètodes de comprovació de revocació associats a un prestador, etc. i del mòdul NonRepudiation per registrar a la base de dades de no repudi la informació utilitzada per comprovar la revocació dels certificats (CRLs, respostes OCSP, etc.).

A alt nivell, els serveis més importants oferts per aquest mòdul a les aplicacions són:

- Validació d'un certificat (no caducitat ni revocació)
- Validació d'un certificat i extracció de les dades incloses (mitjançant la utilització del mòdul de gestió de polítiques)

4.2 Cas de Prova (Parametrizació)

```
String sCertificateInBase64 = FileUtils.readFile(TestConstants.PATH_TEST_CERTS +  
TestConstants.CERT_VALID, "UTF-8");
```

```
HashMap<String, Object> objHashMapInput = new HashMap<String, Object>();  
String[] llistaDeCertificats = { sCertificateInBase64 };
```

```
objHashMapInput.put("idAplicacio", TestConstants.CONSOLE_SIGNATURE_APPLICATION);  
objHashMapInput.put("llistaDeCertificats", llistaDeCertificats);  
objHashMapInput.put("limitDeValidesa", 1);  
objHashMapInput.put("limitDeCaducitat", -1);  
objHashMapInput.put("limitDeRevocacio", 1);
```

```
String input = JsonUtils.convertMapToJSON(objHashMapInput);
```