Manual de Instalación, Despliegue y Administración de Integra-services

Integra 2.2.4_000

Documento nº:	@Firma-Integra-services-InstalacionDespliegueYAdministracion-MAN
Revisión:	011
Fecha:	29-01-2024
Período de retención:	Permanente durante su período de vigencia + 3 años después de su anulación

CONTROL DE MODIFICACIONES

Documento nº: @Firma-Integra-services-InstalacionDespliegueYAdministracion-MAN
Revisión: 011
Fecha: 29-01-2024

Rev. 001
Fecha 23-05-2016
Descripción Documentación inicial.

Rev. 002
Fecha 20-04-2017
Descripción Se actualiza el documento para hacer referencia a la versión 2.2.0_000 de Integr@.

Rev. 003
Fecha 16-10-2017
Descripción Se actualiza el documento para hacer referencia a la versión 2.2.1_000 de Integr@.

Rev. 004
Fecha 09-03-2020
Descripción Se añaden nuevas propiedades asociadas al fichero de configuración de TS@.

Rev. 005
Fecha 23-12-2020
Descripción Se actualiza el documento para hacer referencia a la versión 2.2.2_000 de Integr@.

Rev. 006
Fecha 22-02-2022
Descripción Se actualiza el documento para hacer referencia a la versión 2.2.2_001 de Integr@.

Rev. 007
Fecha 21-04-2022
Descripción Se actualiza el documento para hacer referencia a la versión 2.2.2_002 de Integr@.

Rev. 008
Fecha 22-09-2022
Descripción Se actualiza el documento con la nueva versión de log4j.

Rev.	009
Fecha	26-09-2022
Descripción	Se actualiza el documento para hacer referencia a la versión 2.2.3_000 de Integr@.

Rev.	010
Fecha	07-10-2022
Descripción	Se actualiza el documento para hacer referencia a la versión 2.2.3_001 de Integr@.

Rev.	011
Fecha	29-01-2024
Descripción	Se actualiza el documento para hacer referencia a la versión 2.2.4_000 de Integr@ y se corrige la versión de algunas librerías.

CONTROL DE DISTRIBUCIÓN

Documento nº:	@Firma-Integra-services-InstalacionDespliegueYAdministracion-MAN
Revisión:	011
Fecha:	29-01-2024

Copias Electrónicas:

La distribución de este documento ha sido controlada a través del sistema de información.

ÍNDICE

1 Objeto 6

2 Alcance 7

3 Siglas 8

4 Introducción 10

5 Información Técnica 11

5.1 Requisitos Obligatorios 11

5.2 Tecnologías 11

6 Guía de Instalación 12

6.1 Instalación del servidor 12

6.1.1 Proceso de instalación y despliegue 12

7 Acceso a los servicios 13

7.1 AfirmaServices: 13

7.2 IntegraServices: 13

7.3 TSAServices: 13

7.4 EvisorServices: 13

7.5 CipherServices: 14

8 Configuración 15

8.1 Configuración estática 15

8.1.1 Carpeta transformersTemplates 15

8.1.2 Archivo hsm.properties 19

8.1.3 Archivo Language.properties 20

8.1.4 Archivo integra-log4j2.xml 20

8.1.5 Archivo parserParameters.properties 20

8.1.6 Archivo transformers.properties 20

8.2 Configuración dinámica 21

8.2.1 Archivo mappingFiles.properties 23

8.2.2 Configuraciones generales Integr@ 23

8.2.3 Configuraciones de acceso a TS@ 42

8.2.4 Configuraciones de acceso a @firma 47

8.2.5 Configuraciones de acceso a eVisor 49

1 Objeto

Es objeto de este documento es describir el proceso de instalación y despliegue, así como la configuración del servicio web Integra-services versión 2.2.4_000.

2 Alcance

El objetivo global de este documento es describir el proceso completo de instalación y puesta en marcha de la plataforma, así como su configuración.

3 Siglas

AGE	Administración General del Estado
API	Application Programming Interface
ASiC	Associated Signature Containers
ASiC-S	Simple Associated Signature Containers
ASN.1	Abstract Syntax Notation One
CAdES	CMS Advanced Electronic Signatures
CAdES-BES	CAdES Basic Electronic Signature
CAdES-EPES	CAdES Explicit Policy Electronic Signature
CAdES-T	CAdES Timestamp
CAdES-C	CAdES Complete
CAdES-X	CAdES Extended
CAdES-XL	CAdES Extended Long-Term
CAdES-A	CAdES Archive
CAdES B-Level	CAdES Basic Level
CAdES T-Level	CAdES Trusted Time for Signature Existence Level
CAdES LT-Level	CAdES Long Term Level
CAdES LTA-Level	CAdES Long Term with Archive Time-stamps Level
CMS	Cryptographic Message Syntax
DSS	Digital Signature Services
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JCEKS	Java Cryptography Extension KeyStore
JKS	Java KeyStore
JRE	Java Runtime Environment
MINHAP	Ministerio de Hacienda y Administraciones Públicas de España
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PAdES	PDF Advanced Electronic Signatures
PAdES-Basic	PAdES Basic
PAdES-BES	PAdES Basic Electronic Signature
PAdES-EPES	PAdES Explicit Policy Electronic Signature
PAdES-LTV	PAdES Long Term Validation
PAdES B-Level	PAdES Basic Level

PAdES T-Level	PAdES Trusted Time for Signature Existence Level
PAdES LT-Level	PAdES Long Term Level
PAdES LTA-Level	PAdES Long Term with Archive Time-stamps Level
PDF	Portable Document Format
PKCS	Public-Key Cryptography Standards
RFC	Request For Comments
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
RSA	Rivest, Shamir y Adleman
SAML	Security Assertion Markup Language
SHA	Secure Hash Algorithm
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time
XAdES	XML Advanced Electronic Signatures
XAdES-BES	XAdES Basic Electronic Signature
XAdES-EPES	XAdES Explicit Policy Electronic Signature
XAdES-T	XAdES Timestamp
XAdES-C	XAdES Complete
XAdES-X	XAdES Extended
XAdES-XL	XAdES Extended Long-Term
XAdES-A	XAdES Archive
XAdES B-Level	XAdES Basic Level
XAdES T-Level	XAdES Trusted Time for Signature Existence Level
XAdES LT-Level	XAdES Long Term Level
XAdES LTA-Level	XAdES Long Term with Archive Time-stamps Level
XML	eXtensible Markup Language
XPath	XML Path Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformations
UUID	Identificador Único
WS	Web Services

4 Introducción

Integr@ es un conjunto de librerías compuestas por clases java, ficheros de configuración y plantillas XML que facilitan la integración de una aplicación con los servicios WS de @Firma, los servicios WS de TS@, el servicio RFC 3161 de TS@, el servicio OCSP de validación de certificados de @Firma, y servicios OCSP de validación de certificados ajenos a @Firma. También ofrece servicios propios de firma y cifrado.

Para facilitar el uso de los servicios de Integr@ en sistemas no Java o en los que simplemente se prefiere realizar la integración con los servicios de firma sin usar la API que provee Integr@, se dispone de un WS (Integra-services) en el que se dispondrá de los servicios disponibles en Integr@ en modo WS.

El presente documento describe el proceso de instalación, despliegue y administración de Integra-services v2.2.4_000.

5 Información Técnica

En este apartado vamos a ver los requisitos necesarios que hay que cumplir para el correcto funcionamiento del servicio, así como información sobre los componentes tecnológicos de los que hace uso.

5.1 Requisitos Obligatorios

Para instalar el Sistema Integra-services hay que cumplir una serie de requisitos previos para iniciar la instalación.

Componentes	Descripción	Comentarios
Java Virtual Machine	JDK 1.8.	
Servidor Aplicaciones	Servidor Aplicaciones Apache Tomcat 9.0	

5.2 Tecnologías

En la siguiente tabla se detalla la relación de tecnologías más importantes que integra la aplicación.

Componentes	Descripción
AXIS2	Apache Axis2 es un motor nuclear para servicios web. Es un rediseño total y una re-implementación completa de la ampliamente difundida pila SOAP "Apache Axis".
Trazabilidad	Log4j v.2.20.0
Apache Maven	Maven es una herramienta de gestión de proyectos software y dependencias. Basado en el concepto de Modelo de Objetos de Proyecto (POM), Maven puede gestionar la construcción de proyectos, generación de informes y documentación desde una pieza central de información. Apache Maven 2.3

6 Guía de Instalación

A continuación, en los siguientes apartados se describe la instalación del servidor.

6.1 Instalación del servidor

En este apartado se detallará el **proceso de instalación del servidor de aplicaciones con la solución Integra-services v2.2.4_000 desde cero**.

Para poder completar la instalación del **Integra-services v2.2.4_000** es necesario haber realizado los siguientes pasos previos:

1. Instalar JDK 1.8 en la máquina del Servidor. Versión recomendada 1.8.0_281.
2. Instalar Apache Tomcat 9.0. Versión recomendada 9.0.67.
3. Declarar la carpeta de configuración en los parámetros de arranque de la máquina virtual de Java de la siguiente forma -Dintegra.config=”ruta de la carpeta de configuración”. En esta carpeta se situarán los ficheros de configuración necesarios para el correcto funcionamiento de la solución y que se describirán más adelante.

Una vez cumplido estos requisitos iniciales podremos comenzar el proceso de instalación.

6.1.1 Proceso de instalación y despliegue

- Copiar el fichero war incluido en el entregable en la ruta CATALINA_HOME/webapps/ de Apache Tomcat.
- Reiniciar Apache Tomcat para que se realice el deploy de Integra-services.

7 Acceso a los servicios

Integra-services publica 5 interfaces que se describen a continuación.

7.1 AfirmaServices:

La interfaz engloba los servicios de firma, co-firma, contra-firma, actualización, validación de firma y certificados enviando peticiones al servidor de @firma y OCSP.

Puede consultarse el WSDL descriptor del servicio en la siguiente URL:

http://<SERVIDOR>:<PUERTO>/Integra-services/services/AfirmaServices?wsdl

7.2 IntegraServices:

La interfaz engloba los servicios de firma, co-firma, contra-firma, actualización, validación de firmas realizados por Integra.

Puede consultarse el WSDL descriptor del servicio en la siguiente URL:

http://<SERVIDOR>:<PUERTO>/Integra-services/services/IntegraServices?wsdl

7.3 TSAServices:

La interfaz engloba los servicios relativos a sellos de tiempo obtenidos por petición a una TSA.

Puede consultarse el WSDL descriptor del servicio en la siguiente URL:

http://<SERVIDOR>:<PUERTO>/Integra-services/services/TSAServices?wsdl

7.4 EvisorServices:

La interfaz engloba los servicios relativos a la generación y validación de reportes de firma enviando peticiones al servidor eVisor.

Puede consultarse el WSDL descriptor del servicio en la siguiente URL:

http://<SERVIDOR>:<PUERTO>/Integra-services/services/EvisorServices?wsdl

7.5 CipherServices:

La interfaz engloba los servicios de cifrado disponibles en Integr@.

Puede consultarse el WSDL descriptor del servicio en la siguiente URL:

http://<SERVIDOR>:<PUERTO>/Integra-services/services/CipherServices?wsdl

8 Configuración

La configuración de Integra-services consiste básicamente en el mantenimiento de una serie de ficheros de propiedades colocados en una carpeta definida en las propiedades de la máquina virtual de Java.

8.1 Configuración estática

Elementos de configuración con nombre estático.

- Carpeta **transformersTemplates** (Esta carpeta no tiene que estar necesariamente en la carpeta de configuración del WS, ya que su localización se define en el fichero transformers.properties).
- Archivo **hsm.properties**.
- Archivo **Language.properties**.
- Archivo integra-**log4j2.xml**.
- Archivo **parserParameters.properties**.
- Archivo **transformers.properties**.

8.1.1 Carpeta transformersTemplates

La función de estas plantillas es definir la estructura final de las interfaces XML a generar y los parámetros a extraer de las respuestas XML en los procesos de comunicación con los servicios web de @Firma, eVisor y TS@. La carpeta contiene a su vez 2 carpetas:

8.1.1.1 Carpeta parserTemplates

Esta carpeta incluye las plantillas asociadas a las respuestas XML obtenidas de los diferentes servicios de @Firma, TS@ y eVisor. Las plantillas que lo componen son:

- **DSSArchiveRetrievalResponse_V1.xml**: Plantilla para el procesado de respuestas asociadas al servicio de obtención de firmas registradas.
- **DSSAsyncResponseStatus_V1.xml**: Plantilla para el procesado de las respuestas asociadas a los servicios de consulta de peticiones asíncronas, de @Firma.

- **DSSBatchResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas a los servicios de validaciones en lote, de @Firma.

- **DSSCounterSignResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio firma servidor CounterSign, de @Firma.

- **DSSSignResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas a los servicios de firma delegada, de @Firma.

- **DSSTSAReTimestampResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio de renovación de sello de tiempo, de TS@.

- **DSSTSATimestampResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio de creación de sello de tiempo, de TS@.

- **DSSTSATimestampValidationResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio de validación de sello de tiempo, de TS@.

- **DSSVerifyCertificateResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio de validación de certificados, de @Firma.

- **DSSVerifyResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio de validación y actualización de firmas, de @Firma.

- **EVisor_GenerateReportResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio de generación de informes, de eVisor.

- **EVisor_ValidationReportResponse_V1.xml**: Plantilla para el procesado de las respuestas asociadas al servicio de validación de informes firmados, de eVisor.

8.1.1.2 Carpeta xmlTemplates

Esta carpeta incluye las plantillas asociadas a la construcción de las peticiones XML hacia los diferentes servicios de @Firma, TS@ y eVisor. Las plantillas que la componen son:

- **AlmacenarDocumento_V1.xml**: Plantilla para la construcción de las peticiones al servicio para almacenar documentos, de @Firma.

- **DSSAfirmaSignRequest_V1.xml**: Plantilla para la construcción de las peticiones a los servicios de firma delegada, de @Firma.

- **DSSArchiveRetrievalRequest_V1.xml**: Plantilla para la construcción de peticiones al servicio de obtención de firma registrada, de @Firma.

- **DSSAsyncRequestStatus_V1.xml**: Plantilla para la construcción de las peticiones al servicio de consulta de peticiones asíncronas, de @Firma.

- **DSSBatchRequest_V1.xml**: Plantilla para la construcción de las peticiones a los servicios de validaciones por lote, de @Firma.

- **DSSTSAReTimestampRequest_V1.xml**: Plantilla para la construcción de las peticiones al servicio de renovación de sello de tiempo, de TS@.

- **DSSTSATimestampRequest_V1.xml**: Plantilla para la construcción de las peticiones al servicio de generación de sello de tiempo, de TS@.

- **DSSTSATimestampValidationRequest_V1.xml**: Plantilla para la construcción de las peticiones al servicio de validación de sello de tiempo, de TS@.

- **DSSVerifyRequest_V1.xml**: Plantilla para la construcción de las peticiones a los servicios de validación y actualización de firmas, de @Firma.

- **EliminarContenidoDocumento_V1.xml:** Plantilla para la construcción de peticiones al servicio para eliminar el contenido de documentos, de @Firma.

- **EVisor_GenerateReportRequest_V1.xml**: Plantilla para la construcción de las peticiones al servicio de generación de informes, de eVisor.

- **EVisor_ValidationReportRequest_V1.xml**: Plantilla para la construcción de las peticiones al servicio de validación de informes firmados, de eVisor.

- **FirmaServidor_V1.xml**: Plantilla para la construcción de las peticiones al servicio de firma servidor, de @Firma.

- **FirmaServidorCoSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio de firma servidor CoSign, de @Firma.

- **FirmaServidorCounterSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio de firma servidor CounterSign, de @Firma.

- **FirmaUsuario2FasesF2_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 2 de firma usuario 2 fases, de @Firma.

- **FirmaUsuario3FasesF1_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 1 de firma usuario 3 fases, de @Firma.

- **FirmaUsuario3FasesF1CoSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 1 de firma usuario 3 fases CoSign, de @Firma.

- **FirmaUsuario3FasesF1CounterSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 1 de firma usuario 3 fases CounterSign, de @Firma.

- **FirmaUsuario3FasesF3_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 3 de firma usuario 3 fases, de @Firma.

- **GetInfoCertificate_V1.xml**: Plantilla para la construcción de las peticiones al servicio de obtención de información de certificado, en inglés, de @Firma.

- **GetTransactionSignature_V10.xml**: Plantilla para la construcción de las peticiones al servicio de obtener una firma a partir de su identificador de transacción, en inglés, de @Firma.

- **ObtenerContenidoDocumento_V1.xml**: Plantilla para la construcción de las peticiones al servicio para obtener el contenido de un documento, de @Firma.

- **ObtenerContenidoDocumentoId_V1.xml**: Plantilla para la construcción de las peticiones al servicio para obtener el contenido de un documento a partir de su identificador, de @Firma.

- **ObtenerFirmaTransaccion_V1.xml**: Plantilla para la construcción de las peticiones al servicio de obtener una firma a partir de su identificador de transacción, de @Firma.

- **ObtenerIdDocumento_V1.xml**: Plantilla para la construcción de las peticiones al servicio para obtener el identificador de un documento a partir de su identificador de transacción, de @Firma.

- **ObtenerInfoCertificado_V1.xml**: Plantilla para la construcción de las peticiones al servicio de obtención de información de certificado, de @Firma.

- **ServerSignature_V1.xml**: Plantilla para la construcción de las peticiones al servicio de firma servidor, en inglés, de @Firma.

- **ServerSignatureCoSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio de firma servidor CoSign, en inglés, de @Firma.

- **ServerSignatureCounterSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio de firma servidor CounterSign, en inglés, de @Firma.

- **Signature_Validation_V1.xml**: Plantilla para la construcción de las peticiones al servicio de validación de firmas, en inglés, de @Firma.

- **StoreDocument_V1.xml**: Plantilla para la construcción de las peticiones al servicio para almacenar documentos, en inglés, de @Firma.

- **ThreePhaseUserSignatureF1_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 1 de firma usuario 3 fases, en inglés, de @Firma.

- **ThreePhaseUserSignatureF1CoSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 1 de firma usuario 3 fases CoSign, en inglés, de @Firma.

- **ThreePhaseUserSignatureF1CounterSign_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 1 de firma usuario 3 fases CounterSign, en inglés, de @Firma.

- **ThreePhaseUserSignatureF3_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 3 de firma usuario 3 fases, en inglés, de @Firma.

- **TwoPhaseUserSignatureF2_V1.xml**: Plantilla para la construcción de las peticiones al servicio fase 2 de firma usuario 2 fases, en inglés, de @Firma.

- **Validacion_Firma_V1.xml**: Plantilla para la construcción de las peticiones al servicio de validación de firmas, de @Firma.

- **ValidarCertificado_V1.xml**: Plantilla para la construcción de las peticiones al servicio de validación de certificados, de @Firma.

- **ValidateCertificate_V1.xml**: Plantilla para la construcción de las peticiones al servicio de validación de certificados, en inglés, de @Firma.

8.1.2 Archivo hsm.properties

Este archivo define las propiedades asociadas al uso de almacenes de claves de tipo HSM. Es necesario si se hace uso de dispositivos HSM.

- **HSM_CONFIG_PATH:** Ruta absoluta al fichero de configuración PKCS11 donde se establece la ruta absoluta a la librería nativa del HSM entre otras configuraciones adicionales. Por ejemplo "/opt/users/HSM/config/pkcs11.cfg" indica la ruta al fichero "pkcs11.cfg" que contiene la configuración PKCS11 del HSM, cuyo contenido podría ser:

```
name = HSM_RealSec
library = /opt/users/HSM/lib/libcryptosec.so
description = Dispositivo HSM RealSec
disabledMechanism = {
CKM_SHA512
}
```

- **HSM_PASSWORD:** Contraseña de acceso al HSM. Esta propiedad podría estar vacía en caso de no ser necesaria dicha contraseña.

8.1.3 Archivo Language.properties

Este archivo permite configurar el idioma asociado a los mensajes de la API Integr@, así como para los mensajes emitidos por el WS. Admite seleccionar los mensajes en inglés o en español.

- **LANGUAGE:** Idioma asociado a los mensajes de la API Integr@. Los valores posibles son:
  - **es_ES** → Idioma español.
  - **en_US** → Idioma inglés.

8.1.4 Archivo integra-log4j2.xml

Archivo de configuración de la API Log4j2.

8.1.5 Archivo parserParameters.properties

Fichero de propiedades que contiene los identificadores empleados como atajo o acceso directo a las rutas de los nodos extraídos en el procesado de las respuestas XML de los servicios de @Firma, TS@ y eVisor.

Cada clave se corresponderá con el nombre del atajo o acceso directo al elemento, y el valor será la ruta xPath del nodo que obtener de la respuesta XML. Por ejemplo:

```
RFC3161Timestamp = dss:SignatureObject/dss:Timestamp/dss:RFC3161TimeStampToken
```

En este caso, el atajo con nombre **RFC3161Timestamp** se utilizaría para obtener, de una respuesta del *Servicio de Generación de Sello de Tiempo* de TS@ o del *Servicio de Renovación de Sello de Tiempo* de TS@, el valor del elemento **dss:RFC3161TimeStampToken** contenido dentro del elemento **dss:Timestamp**, que a su vez está contenido dentro del elemento **dss:SignatureObject**.

8.1.6 Archivo transformers.properties

Fichero de propiedades donde se definen las propiedades necesarias para la conversión de parámetros a XML en las peticiones a los WS de @Firma, TS@ y eVisor, así como las propiedades necesarias para el procesamiento de las respuestas de dichos WS.

8.1.6.1 Parámetros de Uso Común a los Servicios de @Firma, eVisor y TS@

- **TransformersTemplatesPath:** Ruta al directorio donde se encuentran las plantillas que contienen la información para generar las interfaces XML a generar y las plantillas que contienen la información para extraer los nodos de las respuestas XML y convertirlos a parámetros.

8.1.6.2 Parámetros Específicos a Cada Servicio de @Firma, eVisor y TS@

Cada propiedad tendrá el formato **NOMBRE_PETICION.NOMBRE_SERVICIO.VERSION_MENSAJE.NOMBRE_PARAMETRO** donde

- **NOMBRE_PETICION** es el nombre asociado a la petición y respuesta del servicio
- **NOMBRE_SERVICIO** es el nombre del servicio
- **VERSION_MENSAJE** es la versión del mensaje
- **NOMBRE_PARAMETRO** es el nombre del parámetro específico

Por ejemplo:

```
ValidarCertificado.ValidarCertificado.1_0.request.transformerClass=es.gob.afirma.transformers.xmlTransformers.CommonXmlTransformer
```

Los parámetros específicos son:

- **request.transformerClass:** Nombre completo de la clase java que genera la interfaz XML para el servicio.
- **request.template:** Nombre de la plantilla XML empleada en la generación de la petición XML. Debe corresponderse con el nombre de alguno de los ficheros XML indicados en 8.1.1.2.
- **parser.transformerClass:** Nombre completo de la clase java que realizará el procesado de la respuesta XML.
- **parser.rootElement:** Ruta XPATH del elemento padre de la información relevante de la respuesta a partir del tag *<mensajeSalida>.*
- **parser.template:** Nombre de la plantilla XML empleada en el procesamiento de la respuesta XML. Debe corresponderse con el nombre de alguno de los ficheros XML indicados en 8.1.1.1.

8.2 Configuración dinámica

En estos ficheros de propiedades se encuentran las configuraciones necesarias para el acceso a @firma, TS@, eVisor, así como los parámetros necesarios para la configuración de ciertos servicios de Integr@.

Dado que puede ser necesario disponer de distintas configuraciones de un mismo servicio en función de las necesidades o clientes que ataquen al servicio, se ha diseñado un sistema mediante el cual Integra-services resuelve el fichero de configuración necesario en cada caso de la siguiente forma.

Cada petición que se realice debe incluir un identificador del cliente que ataca.

Se dispondrá de un fichero único y obligatorio llamado mappingFiles.properties. El fichero contendrá entradas clave=valor donde el valor es el fichero de propiedades de turno situado en la misma carpeta y la clave será un literal que resuelve Integra-services de la siguiente forma:

- Acceso a @firma:

  Los ficheros de acceso a @firma se localizarán con una clave formada por el literal "afirma" concatenándole el identificador del cliente (que será enviado en la petición) y el identificador de la aplicación de @firma dada de alta para las peticiones.

  Por ejemplo, si se realiza una petición con identificador de cliente "cliente1" que se comunicará con @firma mediante la aplicación "appAfirma", en el fichero mappingFiles.properties debe encontrarse una entrada

  afirmacliente1appAfirma = configuracionAfirma.properties

  siendo configuracionAfirma.properties un fichero existente bajo la misma carpeta y que contenga las configuraciones de acceso a @firma para la aplicación "appAfirma" y el cliente "cliente1".

- Acceso a TS@:

  Los ficheros de acceso a TS@ se localizarán con una clave formada por el literal "tsa" concatenándole el identificador del cliente (que será enviado en la petición) y el identificador de la aplicación de TS@ dada de alta para las peticiones.

  Por ejemplo, si se realiza una petición con identificador de cliente "cliente1" que se comunicará con TS@ mediante la aplicación "appTsa", en el fichero mappingFiles.properties debe encontrarse una entrada

  tsacliente1appTsa = configuracionTSA.properties

  siendo configuracionTSA.properties un fichero existente bajo la misma carpeta y que contenga las configuraciones de acceso a TS@ para la aplicación "appTsa" y el cliente "cliente1".

- Acceso a eVisor:

  Los ficheros de acceso a eVisor se localizarán con una clave formada por el literal "evisor" concatenándole el identificador del cliente (que será enviado en la petición) y el identificador de la aplicación de evisor dada de alta para las peticiones.

Por ejemplo, si se realiza una petición con identificador de cliente “cliente1” que se comunicará con eVisor mediante la aplicación “appEvisor”, en el fichero mappingFiles.properties debe encontrarse una entrada

evisorcliente1appEvisor = configuracionEvisor.properties

siendo configuracionEvisor.properties un fichero existente bajo la misma carpeta y que contenga las configuraciones de acceso a eVisor para la aplicación “appEvisor” y el cliente “cliente1”.

- Configuraciones generales Integr@:

Los ficheros de configuración general de integr@ se localizarán con una clave formada por el literal integra concatenándole el identificador del cliente (que será enviado en la petición).

Por ejemplo, si se realiza una petición con identificador de cliente “cliente1”, en el fichero mappingFiles.properties debe encontrarse una entrada

integracliente1 = configuracionIntegra.properties

siendo configuracionIntegra.properties un fichero existente bajo la misma carpeta y que contenga las configuraciones generales de Integr@ para el cliente “cliente1”.

A continuación, se define el contenido de los posibles ficheros de propiedades a configurar.

8.2.1 Archivo mappingFiles.properties

En este archivo se mapean los nombre de los ficheros de propiedades con nombre dinámico a elección del integrador.

Pueden mapearse N ficheros de propiedades atendiendo a una serie de reglas básicas comentadas anteriormente.

8.2.2 Configuraciones generales Integr@

Archivo con configuraciones generales. El nombre del fichero queda a elección del integrador y se definirá en el fichero mappingFiles.properties. Se propone como nombre integraidCliente.properties siendo idCliente el identificador del cliente que envía la petición.

8.2.2.1 Parámetros Comunes a Todas las Aplicaciones de @Firma, eVisor y TS@:

- **com.trustedstorePath:** Ruta al almacén de certificados de confianza usado en comunicaciones seguras.

- **com.trustedstorepassword:** Contraseña del almacén de certificados de confianza usado en comunicaciones seguras.

8.2.2.2 Parámetros Comunes a Todas las Aplicaciones de TS@:

- **com.serviceWSDLPath:** Ruta al fichero descriptor con los servicios web.

8.2.2.3 Parámetros Comunes a Todas las Aplicaciones de @Firma:

- **com.certificatesCache.use:** Indicador para cachear las respuestas de validación de certificados por los servicios DSS simple, nativo en inglés y nativo en español para cada certificado. Los posibles valores son:

    - **true** → Se cachean las respuestas de validación de certificados.

    - **false** → No se cachean las respuestas de validación de certificados.

- **com.certificatesCache.entries:** Número de entradas que formarán la caché de respuestas de validación de certificados. Cada vez que se alcance esta cifra de respuestas de validación de certificados cacheadas, la próxima vez que se vaya a almacenar una entrada en dicha caché se eliminará previamente la entrada más antigua.

- **com.certificatesCache.lifeTime:** Tiempo de validez de cada entrada que formará la caché de respuestas de validación de certificados, en segundos. Cuando una entrada supere ese tiempo de validez será eliminada de la caché.

8.2.2.4 Propiedades para la Validación de Firmantes

Estas propiedades son necesarias para completar correctamente los procesos de firma, co-firma, contra-firma, actualización y validación de firmas.

- **CERTIFICATE_VALIDATION_LEVEL:** Nivel de validación de los firmantes. Esta propiedad está asociada a la validación de firmas CAdES (Baseline o no), XAdES (Baseline o no), PAdES (Baseline o no), y ASiC-S. Los valores posibles son:

    - **0** → Nivel de validación nulo. No se realiza verificación alguna sobre el certificado. Este modo no es aplicable a los procesos de actualización, es decir,

si para un proceso de actualización de firma se ha indicado este nivel de verificación de certificado se realizará la validación indicada en el nivel con valor **1**.

- **1** → Nivel de validación simple. Se verifica el periodo de validez y caducidad del certificado.
- **2** → Nivel de validación completo. Se verifica el periodo de validez y caducidad del certificado, y el estado de revocación del certificado mediante servicio OCSP.

8.2.2.5 Propiedades para la Comunicación con TS@

Estas propiedades son necesarias para completar correctamente los procesos de firma, co-firma, contra-firma y actualización de firmas.

- **TSA_APP_ID:** Identificador de la aplicación cliente para la comunicación con TS@ en caso de necesitarse un sello de tiempo para formatos T.
- **TSA_COMMUNICATION_TYPE:** Tipo de comunicación a usar para obtener el sello de tiempo de TS@. Los valores posibles son:
  - **DSS** → Obtención del sello de tiempo mediante servicio web DSS.
  - **RFC3161-TCP** → Obtención del sello de tiempo mediante servicio RFC 3161 - TCP.
  - **RFC3161-HTTPS** → Obtención del sello de tiempo mediante servicio RFC 3161 - HTTPS.
  - **RFC3161-SSL** → Obtención del sello de tiempo mediante servicio RFC 3161 - SSL.
- **TSA_TIMESTAMP_TYPE:** Tipo de sello de tiempo a solicitar a TS@. Los valores posibles son:
  - **ASN1**→ Sello de tiempo ASN.1.
  - **XML**→ Sello de tiempo XML.

8.2.2.6 Propiedades para la comunicación con @firma

- **AFIRMA_APP_ID:** Identificador de la aplicación cliente para la comunicación con @firma en caso de necesitarse para actualizar a formatos superiores no soportados por la firma propia de Integr@.

8.2.2.7 Propiedades para la Fachada de Generación de Firmas, Co-Firmas y Contra-Firmas

Estas propiedades son necesarias para completar correctamente los procesos de firma, co-firma y contra-firma desde la fachada de invocación para los servicios propios de generación de firmas, co-firmas y contra-firmas de Integr@.

- **FACADE_SIGNATURE_ALGORITHM:** Algoritmo de firma a utilizar en la generación de firmas, co-firmas y contra-firmas. Los valores posibles son:
  - **SHA1withRSA**
  - **SHA256withRSA**
  - **SHA384withRSA**
  - **SHA512withRSA**
- **AFIRMA_APP_ID:** Identificador de la aplicación cliente para la comunicación con @firma en caso de necesitarse para actualizar a formatos superiores no soportados por la firma propia de Integr@.

8.2.2.8 Propiedades del servicio web

- **WS_KEYSTORE:** Ruta del keystore usado por el servicio web para obtener los certificados usados para firmar en las peticiones a los servicios IntegraServices.
- **WS_KEYSTORE_PASS:** Password del keystore usado por el servicio web para obtener los certificados usados para firmar en las peticiones a los servicios IntegraServices.
- **WS_KEYSTORE_TYPE:** Tipo del keystore usado por el servicio web para obtener los certificados usados para firmar en las peticiones a los servicios IntegraServices. Los valores posibles son:
  - **PKCS12**
  - **JKS**
  - **JCEKS**
- **UPPER_FORMAT_UPGRADE_AFIRMA:** Parámetro que indica si se recurre a upgrade de firmas desde @firma en caso de haberse solicitado un formato de firma no soportado por Integr@. Valores posibles true y false.

8.2.2.9 Propiedades de política

Propiedades necesarias para la validación de firmas con política de firma asociada. Admite múltiples implementaciones de políticas de firma, diferenciándolas en función del tipo de firma sobre la que aplique, esto es, ASN.1, XML y PDF.

Para facilitar la comprensión de las propiedades, éstas se dividirán por bloques. Cada bloque permite la inclusión de nuevas propiedades.

8.2.2.9.1 Listado de OIDs para Elementos ASN.1

Este bloque define los elementos ASN.1 a los que se hará referencia en las distintas políticas de firma, de manera que, por cada uno de ellos, se le asociará su respectivo OID. Así, la clave de cada elemento ASN.1 será un texto descriptivo con su nombre, y el valor será su OID. Por defecto, se definen los siguientes elementos ASN.1:

```
ContentType = 1.2.840.113549.1.9.3
MessageDigest = 1.2.840.113549.1.9.4
SigningCertificate = 1.2.840.113549.1.9.16.2.12
SigningCertificateV2 = 1.2.840.113549.1.9.16.2.47
SigningTime = 1.2.840.113549.1.9.5
SignaturePolicyIdentifier = 1.2.840.113549.1.9.16.2.15
ContentHints = 1.2.840.113549.1.9.16.2.4
ContentReference = 1.2.840.113549.1.9.16.2.10
ContentIdentifier = 1.2.840.113549.1.9.16.2.7
SignerLocation = 1.2.840.113549.1.9.16.2.17
SignerAttributes = 1.2.840.113549.1.9.16.2.18
ContentTimeStamp = 1.2.840.113549.1.9.16.2.20
CounterSignature = 1.2.840.113549.1.9.6
CommitmentTypeIndication = 1.2.840.113549.1.9.16.2.16
ArchiveTimeStamp = 1.2.840.113549.1.9.16.2.48
CompleteCertificateRefs = 1.2.840.113549.1.9.16.2.21
CompleteRevocationRefs = 1.2.840.113549.1.9.16.2.22
TimestampedCertsCRLs = 1.2.840.113549.1.9.16.2.26
Data = 1.2.840.113549.1.7.1
```

Todo elemento ASN.1 que se indique en la configuración de alguna política de firma deberá estar definido siguiendo el modelo antes indicado.

8.2.2.9.2 Listado de URIs para Algoritmos de Hash en Firmas XML

Este bloque define los algoritmos de hash para firmas XML a los que se hará referencia en las distintas políticas de firma, de manera que, por cada uno de ellos, se le asociará su respectiva URI. Así, la clave de cada algoritmo de hash tendrá el formato **XML_HASH-HASH_ALGORITHM** donde:

- **XML_HASH** es una cadena de texto fijo
- **HASH_ALGORITHM** es una cadena de texto con el nombre algoritmo de hash

Por defecto, se definen los siguientes algoritmos de hash para firmas XML:

```
XML_HASH-SHA1 = http://www.w3.org/2000/09/xmldsig#sha1
XML_HASH-SHA256 = http://www.w3.org/2001/04/xmlenc#sha256
XML_HASH-SHA512 = http://www.w3.org/2001/04/xmlenc#sha512
```

Todo algoritmo de hash para firmas XML que se indique en la configuración de alguna política de firma deberá estar definido siguiendo el modelo antes indicado.

8.2.2.9.3 Listado de URIs para Algoritmos de Firma en Firmas XML

Este bloque define los algoritmos de firma para firmas XML a los que se hará referencia en las distintas políticas de firma, de manera que, por cada uno de ellos, se le asociará su respectiva URI. Así, la clave de cada algoritmo de firma tendrá el formato **XML_SIGN_HASH-HASH_ALGORITHM** donde:

- **XML_SIGN_HASH** es una cadena de texto fijo
- **HASH_ALGORITHM** es una cadena de texto con el nombre del algoritmo de firma

Por defecto, se definen los siguientes algoritmos de firma para firmas XML:

```
XML_SIGN_HASH-SHA1WithRSA = http://www.w3.org/2000/09/xmldsig#rsa-sha1
XML_SIGN_HASH-SHA256WithRSA = http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
XML_SIGN_HASH-SHA512WithRSA = http://www.w3.org/2001/04/xmldsig-more#rsa-sha512
```

Todo algoritmo de firma para firmas XML que se indique en la configuración de alguna política de firma deberá estar definido siguiendo el modelo antes indicado.

8.2.2.9.4 Listado de OIDs para Algoritmos de Hash en Firmas ASN.1 y PDF

Este bloque define los algoritmos de hash para firmas ASN.1 y PDF a los que se hará referencia en las distintas políticas de firma, de manera que, por cada uno de ellos, se le asociará su respectivo OID. Así, la clave de cada algoritmo de hash tendrá el formato **ASN1_HASH-HASH_ALGORITHM** donde:

- **ASN1_HASH** es una cadena de texto fijo
- **HASH_ALGORITHM** es una cadena de texto con el nombre algoritmo de hash

Por defecto, se definen los siguientes algoritmos de hash para firmas ASN.1 y PDF:

```
ASN1_HASH-SHA1 = 1.3.14.3.2.26
ASN1_HASH-SHA256 = 2.16.840.1.101.3.4.2.1
```

```
ASN1_HASH-SHA512 = 2.16.840.1.101.3.4.2.3
```

Todo algoritmo de hash para firmas ASN.1 y PDF que se indique en la configuración de alguna política de firma deberá estar definido siguiendo el modelo antes indicado.

8.2.2.9.5 Listado de OIDs para Algoritmos de Firma en Firmas ASN.1 y PDF

Este bloque define los algoritmos de firma para firmas ASN.1 y PDF a los que se hará referencia en las distintas políticas de firma, de manera que, por cada uno de ellos, se le asociará su respectivo OID. Así, la clave de cada algoritmo de firma tendrá el formato **ASN1_SIGN_HASH-HASH_ALGORITHM** donde:

- **ASN1_SIGN_HASH** es una cadena de texto fijo
- **HASH_ALGORITHM** es una cadena de texto con el nombre del algoritmo de firma

Por defecto, se definen los siguientes algoritmos de firma para firmas ASN.1 y PDF:

```
ASN1_SIGN_HASH-SHA1WithRSA = 1.2.840.113549.1.1.5
ASN1_SIGN_HASH-SHA256WithRSA = 1.2.840.113549.1.1.11
ASN1_SIGN_HASH-SHA512WithRSA = 1.2.840.113549.1.1.13
```

Todo algoritmo de firma para firmas ASN.1 y PDF que se indique en la configuración de alguna política de firma deberá estar definido siguiendo el modelo antes indicado.

8.2.2.9.6 Identificadores de Política de Firma por Tipo

Este bloque define 3 identificadores de política de firma, uno por cada tipo de firma:

- **XML_POLICY_ID:** Identificador de la política de firma a usar en la generación de firmas XML. El valor no puede contener los caracteres dos puntos (:), igual (=), o espacios en blanco. Por ejemplo:

```
XML_POLICY_ID = XML_AGE_1.9_URL
```

- **ASN1_POLICY_ID:** Identificador de la política de firma a usar en la generación de generar firmas ASN.1. El valor no puede contener los caracteres dos puntos (:), igual (=), o espacios en blanco. Por ejemplo:

```
ASN1_POLICY_ID = ASN1_AGE_1.9
```

- **PDF_POLICY_ID:** Identificador de la política de firma a usar en la generación de generar firmas PDF. El valor no puede contener los caracteres dos puntos (:), igual (=), o espacios en blanco. Por ejemplo:

```
PDF_POLICY_ID = PDF_AGE_1.9
```

8.2.2.9.7 Normas Asociadas a una Política de Firma

Este bloque define el conjunto de normas y restricciones para una política de firma concreta. Toda clave asociada a una política de firma se compone de 2 partes separadas por el carácter guion (-), de manera que la parte situada a la izquierda del guion se corresponderá con el identificador de la política de firma, y la parte situada a la derecha del guion se corresponderá con el identificador de la clave de norma. Por ejemplo, para indicar que el algoritmo de resumen a usar en el cálculo de la huella digital del documento legible de la política de firma con identificador *XML_AGE_1.8_XML* debe ser SHA-1 se identificaría de la siguiente manera:

```
XML_AGE_1.9_URL-HASH_ALGORITHM = SHA-1
```

A continuación, se describirá cada una de las normas:

8.2.2.9.7.1 Identificador de la Política de Firma

Este identificador puede servir para firmas ASN.1, XML y PDF.

Si la política de firma se refiere a firmas ASN.1 tendrá el formato **ID_POLICY-IDENTIFIER_ASN1** donde:

- **ID_POLICY** es el identificador de política de firma en este archivo
- **IDENTIFIER_ASN1** es una cadena de texto fijo

Por ejemplo:

```
ASN1_AGE_1.9-IDENTIFIER_ASN1 = 2.16.724.1.3.1.1.2.1.9
```

Si la política de firma se refiere a firmas XML tendrá el formato **ID_POLICY-IDENTIFIER_ XML** donde:

- **ID_POLICY** es el identificador de política de firma en este archivo
- **IDENTIFIER_ XML** es una cadena de texto fijo

Por ejemplo:

```
XML_AGE_1.9_URL-IDENTIFIER_XML =
http://administracionelectronica.gob.es/es/ctt/politicafirma/politica_firma_AGE_v1
_9.pdf
```

Si la política de firma se refiere a firmas PDF tendrá el formato **ID_POLICY-IDENTIFIER_ASN1** donde:

- **ID_POLICY** es el identificador de política de firma en este archivo
- **IDENTIFIER_PDF** es una cadena de texto fijo

Por ejemplo:

```
PDF_AGE_1.9-IDENTIFIER_PDF = 2.16.724.1.3.1.1.2.1.9
```

8.2.2.9.7.2 Algoritmo de Resumen que usar para Calcular la Huella Digital del Documento Legible de Política de Firma

Tendrá el formato **ID_POLICY-HASH_ALGORITHM** donde:

- **ID_POLICY** es el identificador de política de firma en este archivo
- **HASH_ALGORITHM** es una cadena de texto fijo

Valores permitidos:

- SHA-1
- SHA-256
- SHA-512

Por ejemplo:

```
XML_AGE_1.9_URL-HASH_ALGORITHM = SHA-1
```

8.2.2.9.7.3 Valor del Resumen del Documento Legible de Política de Firma codificado en Base64

Tendrá el formato **ID_POLICY-HASH_VALUE** donde:

- **ID_POLICY** es el identificador de política de firma en este archivo
- **HASH_VALUE** es una cadena de texto fijo

Por ejemplo:

```
XML_AGE_1.9_URL-HASH_VALUE = 7SxX3erFuH31TvAw9LZ70N7p1vA=
```

8.2.2.9.7.4 Algoritmos de Resumen Válidos

Cada elemento estará separado por coma (,). Tendrá el formato **ID_POLICY-ALLOWED_HASH_ALGORITHM** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **ALLOWED_HASH_ALGORITHM** es una cadena de texto fijo

Si el algoritmo de resumen es para una firma ASN.1 o PDF su nombre se cogerá de la lista de OIDs para algoritmos de hash en firmas ASN.1 y PDF descrita en 8.2.2.9.4. Por ejemplo:

```
ASN1_AGE_1.9-ALLOWED_HASH_ALGORITHM = ASN1_HASH-SHA1,ASN1_HASH-SHA256,ASN1_HASH-SHA512
```

Si el algoritmo de resumen es para una firma XML su nombre se cogerá de la lista de URIs para algoritmos de hash en firmas XML descrita en 8.2.2.9.2. Por ejemplo:

```
XML_AGE_1.9_URL-ALLOWED_HASH_ALGORITHM = XML_HASH-SHA1,XML_HASH-SHA256,XML_HASH-SHA512
```

8.2.2.9.7.5 Algoritmos de Firma Válidos

Cada elemento estará separado por coma (,). Tendrá el formato **ID_POLICY-ALLOWED_SIGN_ALGORITHM** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **ALLOWED_SIGN_ALGORITHM** es una cadena de texto fijo

Si el algoritmo de firma es para una firma ASN.1 o PDF su nombre se cogerá de la lista de OIDs para algoritmos de firma en firmas ASN.1 y PDF descrita en 8.2.2.9.5. Por ejemplo:

```
ASN1_AGE_1.9-ALLOWED_SIGN_ALGORITHM = ASN1_SIGN_HASH-SHA1WithRSA,ASN1_SIGN_HASH-SHA256WithRSA,ASN1_SIGN_HASH-SHA512WithRSA
```

Si el algoritmo de firma es para una firma XML su nombre se cogerá de la lista de OIDs para algoritmos de firma en firmas XML descrita en 8.2.2.9.3. Por ejemplo:

```
XML_AGE_1.9_URL-ALLOWED_SIGN_ALGORITHM = XML_SIGN_HASH-SHA1WithRSA,XML_SIGN_HASH-SHA256WithRSA,XML_SIGN_HASH-SHA512WithRSA
```

8.2.2.9.7.6 Descripción de la Política de Firma

Tendrá el formato **ID_POLICY-DESCRIPTION** donde:

- **ID_POLICY** es el identificador de política de firma en este archivo
- **DESCRIPTION** es una cadena de texto fijo

Por ejemplo:

```
PDF_AGE_1.9-DESCRIPTION = Política de Firma Electrónica y de Certificados de la
Administración General del Estado 1.9 para firmas PAdES
```

8.2.2.9.7.7 Elementos Firmados Obligatorios

Los elementos pueden ser XML o ASN.1. Cada elemento estará separado por coma (,). Para especificar que se debe elegir entre varios se usará el operador lógico OR (|). Tendrá el formato **ID_POLICY-MANDATORY_SIGNED_ELEMENTS** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **MANDATORY_SIGNED_ELEMENTS** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-MANDATORY_SIGNED_ELEMENTS =
ContentType,MessageDigest,SigningCertificate|SigningCertificateV2,SigningTime,Sig
naturePolicyIdentifier,ContentHints
```

Si el elemento es XML se incluirá sin el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo prefijo para el espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL-MANDATORY_SIGNED_ELEMENTS =
SigningTime,SigningCertificate,SignaturePolicyIdentifier,DataObjectFormat
```

8.2.2.9.7.8 Elementos Firmados Opcionales

Los elementos pueden ser XML o ASN.1. Cada elemento estará separado por coma (,). Para especificar que se debe elegir entre varios se usará el operador lógico OR (|). Tendrá el formato **ID_POLICY-OPTIONAL_SIGNED_ELEMENTS** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **OPTIONAL_SIGNED_ELEMENTS** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-OPTIONAL_SIGNED_ELEMENTS =
ContentReference,ContentIdentifier,CommitmentTypeIndication,SignerLocation,Signer
Attributes,ContentTimeStamp
```

Si el elemento es XML se incluirá sin el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo prefijo para el espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL-OPTIONAL_SIGNED_ELEMENTS =
SignatureProductionPlace,SignerRole,CommitmentTypeIndication,AllDataObjectsTimeSt
amp,IndividualDataObjectsTimeStamp
```

8.2.2.9.7.9 Elementos no Firmados Obligatorios

Los elementos pueden ser XML o ASN.1. Cada elemento estará separado por coma (,). Para especificar que se debe elegir entre varios se usará el operador lógico OR (|).Tendrá el formato **ID_POLICY-MANDATORY_UNSIGNED_ELEMENTS** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **MANDATORY_UNSIGNED_ELEMENTS** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-MANDATORY_UNSIGNED_ELEMENTS =
CompleteCertificateRefs,CompleteRevocationRefs,ArchiveTimeStamp|TimestampedCertsC
RLs
```

Si el elemento es XML se incluirá sin el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo prefijo para el espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL-MANDATORY_UNSIGNED_ELEMENTS =
SignatureTimeStamp,SigAndRefsTimeStamp|RefsOnlyTimeStamp,ArchiveTimeStamp
```

8.2.2.9.7.10 Elementos no Firmados Opcionales

Los elementos pueden ser XML o ASN.1. Cada elemento estará separado por coma (,). Para especificar que se debe elegir entre varios se usará el operador lógico OR (|). Tendrá el formato **ID_POLICY-OPTIONAL_UNSIGNED_ELEMENTS** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **OPTIONAL_UNSIGNED_ELEMENTS** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-OPTIONAL_UNSIGNED_ELEMENTS = CounterSignature
```

Si el elemento es XML se incluirá sin el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo prefijo para el espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL-OPTIONAL_UNSIGNED_ELEMENTS = CounterSignature
```

8.2.2.9.7.11 Elementos Firmados no Permitidos

Los elementos pueden ser XML o ASN.1. Cada elemento estará separado por coma (,). Tendrá el formato **ID_POLICY-NOT_ALLOWED_SIGNED_ELEMENTS** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **NOT_ALLOWED_SIGNED_ELEMENTS** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-NOT_ALLOWED_SIGNED_ELEMENTS =
ContentType,MessageDigest,SigningCertificateV2
```

Si el elemento es XML se incluirá sin el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo prefijo para el espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL-NOT_ALLOWED_SIGNED_ELEMENTS =
SigningTime,SigningCertificate,DataObjectFormat
```

8.2.2.9.7.12 Elementos no Firmados no Permitidos

Los elementos pueden ser XML o ASN.1. Cada elemento estará separado por coma (,). Tendrá el formato **ID_POLICY-NOT_ALLOWED_UNSIGNED_ELEMENTS** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **NOT_ALLOWED_UNSIGNED_ELEMENTS** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-NOT_ALLOWED_UNSIGNED_ELEMENTS =
CompleteCertificateRefs,CompleteRevocationRefs
```

Si el elemento es XML se incluirá sin el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo prefijo para el espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL- NOT_ALLOWED_UNSIGNED_ELEMENTS =
SignatureTimeStamp,SigAndRefsTimeStamp,ArchiveTimeStamp
```

8.2.2.9.7.13 Entradas Obligatorias

Incluye una lista con los nombres de las entradas de carácter obligatorio que debe contener la firma. Cada entrada se incluirá con el prefijo barra inclinada (/). Esta propiedad sólo se usará en el caso de firmas PAdES. Cada entrada estará separada por coma (,). Para especificar que se debe elegir entre varias se usará el operador lógico OR (|). Tendrá el formato **ID_POLICY-REQUIRED_ENTRIES** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **REQUIRED_ENTRIES** es una cadena de texto fijo

Por ejemplo:

```
PDF_AGE_1.9-NOT_ALLOWED_ENTRIES = /M,/Location,/Reason|/Cert
```

8.2.2.9.7.14 Entradas Opcionales

Incluye una lista con los nombres de las entradas de carácter opcional que puede contener la firma. Cada entrada se incluirá con el prefijo barra inclinada (/). Esta propiedad sólo se usará en el caso de firmas PAdES. Cada entrada estará separada por coma (,). Para especificar que se debe elegir entre varios se usará el operador lógico OR (|). Tendrá el formato **ID_POLICY-OPTIONAL_ENTRIES** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **OPTIONAL_ENTRIES** es una cadena de texto fijo

Por ejemplo:

```
PDF_AGE_1.9-OPTIONAL_ENTRIES = /M,/Location
```

8.2.2.9.7.15 Entradas no Permitidas

Incluye una lista con los nombres de las entradas que no puede contener la firma. Cada entrada se incluirá con el prefijo barra inclinada (/). Esta propiedad sólo se usará en el caso de firmas PAdES. Cada entrada estará separada por coma (,). Tendrá el formato **ID_POLICY- NOT_ALLOWED_ENTRIES** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **NOT_ALLOWED_ENTRIES** es una cadena de texto fijo

Por ejemplo:

```
PDF_AGE_1.9-NOT_ALLOWED_ENTRIES = /Cert,/Reason
```

8.2.2.9.7.16 Elementos Hijos Obligatorios

Para un elemento se puede especificar su lista de hijos requeridos. Los elementos sólo pueden ser XML. Si el elemento es XML se incluirá SIN el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Cada elemento estará separado por coma (,). Para especificar que se debe elegir entre varios se usará el operador lógico OR (|). Tendrá el formato **ID_POLICY-[ELEMENT_NAME]-REQUIRED_CHILD** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **ELEMENT_NAME** es el nombre del elemento XML
- **REQUIRED_CHILD** es una cadena de texto fijo

Por ejemplo:

```
XML_AGE_1.9_URL-[SignerRole]-REQUIRED_CHILD = ClaimedRoles|CertifiedRoles
```

8.2.2.9.7.17 Elementos Hijos Opcionales

Para un elemento se puede especificar su lista de hijos opcionales. Los elementos sólo pueden ser XML. Si el elemento es XML se incluirá SIN el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Cada elemento estará separado por coma (,). Para especificar que se debe elegir entre varios se usará el operador lógico OR (|). Tendrá el formato **ID_POLICY-[ELEMENT_NAME]-OPTIONAL_CHILD** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **ELEMENT_NAME** es el nombre del elemento XML

- **OPTIONAL_CHILD** es una cadena de texto fijo

Por ejemplo:

```
XML_AGE_1.9_URL-[SignerRole]-OPTIONAL_CHILD = ClaimedRoles|CertifiedRoles
```

8.2.2.9.7.18 Elementos Hijos no Permitidos

Para un elemento se puede especificar su lista de hijos opcionales. Los elementos sólo pueden ser XML. Si el elemento es XML se incluirá SIN el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Cada elemento estará separado por coma (,). Tendrá el formato **ID_POLICY-[ELEMENT_NAME]-NOT_ALLOWED_CHILD** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **ELEMENT_NAME** es el nombre del elemento XML
- **NOT_ALLOWED_CHILD** es una cadena de texto fijo

Por ejemplo:

```
XML_AGE_1.9_URL-[SignerRole]-NOT_ALLOWED_CHILD = ClaimedRoles,CertifiedRoles
```

8.2.2.9.7.19 Valor Obligatorio de Elemento

Para un elemento se puede especificar su valor requerido. Los elementos pueden ser XML, ASN.1 o PDF. Si el elemento admite más de un valor posible se debe elegir entre varios mediante el operador lógico OR (|). Tendrá el formato **ID_POLICY-[ELEMENT_NAME]- REQUIRED_VALUE** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **ELEMENT_NAME** es el nombre del elemento
- **REQUIRED_VALUE** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-[ContentType]-REQUIRED_VALUE = Data
```

Si el elemento es XML se incluirá SIN el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL-[ClaimedRoles]-REQUIRED_VALUE =
supplier|emisor|customer|receptor|third party|tercero
```

Si el elemento es PDF hará referencia a una entrada. La entrada se incluirá con el prefijo barra inclinada (/). Por ejemplo:

```
PDF_AGE_1.9-[/SubFilter]-REQUIRED_VALUE = ETSI.CAdES.detached
```

8.2.2.9.7.20 Valores no Permitidos para un Elemento

Para un elemento se puede especificar los valores no permitidos. Los elementos pueden ser XML, ASN.1 o PDF. Si el elemento no admite ningún valor de una lista de posibles, cada valor estará separado por coma (,). Tendrá el formato **ID_POLICY-[ELEMENT_NAME]-NOT_ALLOWED_VALUE** donde

- **ID_POLICY** es el identificador de política de firma en este archivo
- **ELEMENT_NAME** es el nombre del elemento
- **NOT_ALLOWED_VALUE** es una cadena de texto fijo

Si el elemento es ASN.1 su nombre se cogerá de la lista de OIDs descrita en 8.2.2.9.1. Por ejemplo:

```
ASN1_AGE_1.9-[ContentType]-NOT_ALLOWED_VALUE = Data
```

Si el elemento es XML se incluirá SIN el prefijo asociado a su espacio de nombres. Sólo se admiten elementos cuyo espacio de nombres pueda ser *ds*, *xades* o *xadesv141*. Por ejemplo:

```
XML_AGE_1.9_URL-[ClaimedRoles]-NOT_ALLOWED_VALUE = emisor,receptor,tercero
```

Si el elemento es PDF hará referencia a una entrada. La entrada se incluirá con el prefijo barra inclinada (/). Por ejemplo:

```
PDF_AGE_1.9-[/SubFilter]-NOT_ALLOWED_VALUE = adbe.pkcs7.s5
```

8.2.2.9.7.21 Modos de Firma Permitidos

Sólo aplicable a firmas XML y ASN.1. Establece los modos de firma permitidos. Si el tipo de firma admite más de un modo de firma, cada valor estará separado por coma (,). Para el caso de firmas ASN.1 los modos de firma posibles son el implícito, (la firma incluye los datos originales), y el explícito, (la firma no incluye los datos originales), identificados respectivamente por:

- Implicit
- Explicit

Por ejemplo, para indicar que los modos de firma permitidos para firmas ASN.1 son ambos, (implícito y explícito), se definiría:

```
ASN1_AGE_1.9-ALLOWED_SIGNING_MODES = Implicit,Explicit
```

Para el caso de firmas XML los modos de firma posibles son "detached", (la firma incluye una referencia a los datos firmados), "enveloped", (la firma incluye los datos firmados), y "enveloping", (la firma posee una estructura XML que contiene internamente el contenido firmado en un nodo propio), identificados respectivamente por:

- Detached
- Enveloped
- Enveloping

Por ejemplo, para indicar que el modo de firma permitido para firmas XML es únicamente "enveloped" se definiría:

```
XML_AGE_1.9_URL-ALLOWED_SIGNING_MODES = Enveloped
```

8.2.2.10 Propiedades para acceso OCSP

Propiedades asociadas a la comunicación con un servidor OCSP para la validación del estado de revocación de certificados.

- **OCSP_URL:** URL de acceso al servicio OCSP.
- **OCSP_ISSUER_KEYSTORE_PATH:** Ruta del almacén de claves donde se almacenan los certificados raíz emisores del certificado a validar.
- **OCSP_ISSUER_KEYSTORE_TYPE:** Tipo del almacén de claves donde se almacenan los certificados raíz emisores del certificado a validar. Los valores posibles son:
  - **PKCS12**
  - **JKS**
  - **JCEKS**
- **OCSP_ISSUER_KEYSTORE_PASSWORD:** Contraseña del almacén de claves donde se almacenan los certificados raíz emisores del certificado a validar.
- **OCSP_RESPONSE_CERTIFICATE_PATH:** Ruta del certificado con el que firma la respuesta el servidor OCSP.

- **OCSP_TIMEOUT:** Tiempo de vida definido para la comunicación con el servidor OCSP, en milisegundos.

- **OCSP_SIGN_REQUEST:** Indicador para firmar la petición OCSP. Los valores posibles son:

    - **true** → La petición OCSP debe ir firmada.

    - **false** → La petición OCSP no debe ir firmada.

- **OCSP_REQUEST_KEYSTORE_PATH:** Ruta al almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.

- **OCSP_REQUEST_KEYSTORE_TYPE:** Tipo de almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado. Los valores posibles son:

    - **PKCS12**

    - **JKS**

    - **JCEKS**

- **OCSP_REQUEST_KEYSTORE_PASSWORD:** Contraseña del almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.

- **OCSP_REQUEST_PRIVATE_KEY_ALIAS:** Alias de la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.

- **OCSP_REQUEST_PRIVATE_KEY_PASSWORD:** Contraseña de la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.

- **OCSP_APP_ID:** Identificador de la aplicación cliente para indicar en la petición OCSP.

- **OCSP_USE_GET:** Indicador para realizar la comunicación con el servidor OCSP mediante una petición GET o POST. Los valores posibles son:

    - **true** → La petición es de tipo GET.

    - **false** → La petición es de tipo POST.

- **OCSP_TRUSTEDSTORE_PATH:** Ruta al almacén de confianza (JKS) para conexiones seguras.

- **OCSP_TRUSTEDSTORE_PASSWORD:** Contraseña del almacén de confianza (JKS) para conexiones seguras.

- **OCSP_HTTPS_USE_AUTH_CLIENT:** Indicador para utilizar autenticación HTTPS mediante certificado cliente. Los valores posibles son:
  - **true** → El cliente se autentica mediante certificado.
  - **false** → El cliente no requiere de autenticación mediante certificado.
- **OCSP_HTTPS_KEYSTORE_PATH:** Ruta al almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación HTTPS del cliente por certificado.
- **OCSP_HTTPS_KEYSTORE_TYPE:** Tipo de almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación HTTPS del cliente por certificado. Los valores posibles son:
  - **PKCS12**
  - **JKS**
  - **JCEKS**
- **OCSP_HTTPS_KEYSTORE_PASSWORD:** Contraseña del almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación HTTPS del cliente por certificado.

8.2.3 Configuraciones de acceso a TS@

El nombre del fichero queda a elección del integrador y se definirá en el fichero mappingFiles.properties. Se propone como nombre tsaidClienteidAplicacion.properties siendo idCliente el identificador del cliente que envía la petición e idAplicacion el identificador de la aplicación TS@ utilizada para la comunicación con TS@.

8.2.3.1 Parámetros Específicos a Cada Una de las Aplicaciones Configuradas en la Plataforma de TS@

- **callTimeout:** Tiempo de vida para las peticiones SOAP, en milisegundos.
- **renewTimeStampWS.validationLevel:** Modo de validación para los sellos de tiempo que van a ser renovados. Los valores posibles son:
  - 0 → No se llevará a cabo ninguna validación. **Este modo va a en contra del estándar definido por OASIS que establece que en una operación de renovación de sello de tiempo el cliente debe validar el sello de tiempo previamente**.

  - 1 → Validación de la integridad. Se comprobará si el documento de entrada (*InputDocument*) asociado al sello de tiempo es correcto.
  - 2 → Validación completa. El sello de tiempo a renovar será previamente validado contra TS@.

- **authorizationMethod:** Tipo de autenticación a utilizar para las peticiones SOAP. Los valores posibles son:
  - **UserNameToken** → Autorización por usuario y contraseña.
  - **X509CertificateToken** → Autorización por certificado.
  - **SAMLToken** → Autorización por SAML.
- **UserNameToken.userName:** Nombre de usuario para el caso de autenticación por usuario y contraseña para la petición SOAP.
- **UserNameToken.userPassword:** Contraseña de usuario para el caso de autenticación por usuario y contraseña para la petición SOAP.
- **X509CertificateToken.inclusionMethod:** Mecanismo de inclusión del certificado para el caso de autorización por certificado para la petición SOAP. Los valores posibles son:
  - **Direct** → Binary Security Token.
  - **Identifier** → Key Identifier.
  - **IssuerSerialNumber** → Issuer and Serial Number.
- **X509CertificateToken.keystorePath:** Ruta al almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.
- **X509CertificateToken.keystoreType:** Tipo de almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado. Los valores posibles son:
  - **PKCS12**
  - **JKS**
  - **JCEKS**
- **X509CertificateToken.keystorePassword:** Contraseña del almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.

- **X509CertificateToken.privateKeyAlias:** Alias de la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.

- **X509CertificateToken.privateKeyPassword:** Contraseña de la clave privada a usar para firmar la petición SOAP en el caso de autenticación por certificado.

- **SAMLToken.method:** Método de confirmación del sujeto para el caso de autorización por SAML. Los valores posibles son:

    - **HOK** → Holder-of-Key.

    - **SV** → Sender-Vouches.

- **SAMLToken.keystorePath:** Ruta al almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por SAML.

- **SAMLToken.keystoreType:** Tipo de almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por SAML. Los valores posibles son:

    - **PKCS12**

    - **JKS**

    - **JCEKS**

- **SAMLToken.keystorePassword:** Contraseña del almacén de claves donde se encuentra almacenada la clave privada a usar para firmar la petición SOAP en el caso de autenticación por SAML.

- **SAMLToken.privateKeyAlias:** Alias de la clave privada a usar para firmar la petición SOAP en el caso de autenticación por SAML.

- **SAMLToken.privateKeyPassword:** Contraseña de la clave privada a usar para firmar la petición SOAP en el caso de autenticación por SAML.

- **request.symmetricKey.use:** Indicador para cifrar las peticiones SOAP con clave simétrica o no. Los valores posibles son:

    - **true** → Las peticiones SOAP irán cifradas.

    - **false** → Las peticiones SOAP no irán cifradas.

- **request.symmetricKey.alias:** Alias de la clave simétrica a usar para cifrar las peticiones SOAP.

- **request.symmetricKey.value**: Valor de la clave simétrica, en hexadecimal, a usar para cifrar las peticiones SOAP.

- **request.symmetricKey.algorithm:** Algoritmo a usar para el cifrado de la comunicación simétrica.

- **response.validate:** Indicador para validar las respuestas SOAP firmadas o no. Los valores posibles son:

    - **true** → Las peticiones SOAP irán cifradas.

    - **false** → Las peticiones SOAP no irán cifradas.

-

- **response.keystorePath:** Ruta al almacén de claves donde se encuentra almacenado el certificado a usar para validar las respuestas SOAP que se encuentren firmadas.

- **response.keystoreType:** Tipo de almacén de claves donde se encuentra almacenado el certificado a usar para validar las respuestas SOAP que se encuentren firmadas. Los valores posibles son:

    - **PKCS12**

    - **JKS**

    - **JCEKS**

- **response.keystorePassword:** Contraseña del almacén de claves donde se encuentra almacenado el certificado a usar para validar las respuestas SOAP que se encuentren firmadas.

- **response.certificateAlias:** Alias del certificado a usar para validar las respuestas SOAP que se encuentren firmadas.

- **response.SAML.keystorePath:** Ruta al almacén de claves donde se encuentra almacenado el certificado a usar para validar las respuestas SOAP aseguradas con SAML.

- **response.SAML.keystoreType:** Tipo de almacén de claves donde se encuentra almacenado el certificado a usar para validar las respuestas SOAP aseguradas con SAML. Los valores posibles son:

    - **PKCS12**

    - **JKS**

    - **JCEKS**

- **response.SAML.keystorePassword:** Contraseña del almacén de claves donde se encuentra almacenado el certificado a usar para validar las respuestas SOAP aseguradas con SAML.

- **response.SAML.certificateAlias:** Alias del certificado a usar para validar las respuestas SOAP aseguradas con SAML.

- **response.symmetricKey.alias:** Alias de la clave simétrica a usar para descifrar las respuestas SOAP cifradas con clave simétrica.

- **response.symmetricKey.value:** Valor de la clave simétrica, en hexadecimal, a usar para descifrar las respuestas SOAP cifradas con clave simétrica.

- **rfc3161.host:** Dirección host donde se encuentra desplegado el servicio RFC 3161.

- **rfc3161.timestampPolicyOID:** OID de la política de sello de tiempo a indicar en la petición.

- **rfc3161.applicationOID:** OID de la aplicación a indicar en la petición. No es necesario en caso de atacar a una TSA que no requiera aplicación cliente en la invocación.

- **rfc3161.Timeout:** Tiempo de vida para las peticiones al servicio RFC 3161, en milisegundos.

- **rfc3161.shaAlgorithm:** Algoritmo de resumen que aplicar sobre los datos a sellar. Los valores posibles son:

    - **SHA**
    - **SHA-256**
    - **SHA-512**
    - **RIPEMD-160**

- **rfc3161.portNumber:** Número del puerto donde se encuentra desplegado el servicio RFC 3161.

- **rfc3161HTTPS.portNumber:** Número del puerto donde se encuentra desplegado el servicio RFC 3161 - HTTPS.

- **rfc3161HTTPS.context:** Contexto para la conexión con el servicio RFC 3161 por HTTPS.

- **rfc3161HTTPS.useAuthClient:** Indicador para utilizar autenticación HTTPS mediante certificado cliente o no. Los valores posibles son:

    - **true** → El cliente se autentica mediante certificado.

  - **false** → El cliente no requiere de autenticación mediante certificado.

- **rfc3161HTTPS.keystorePath:** Ruta al almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación HTTPS del cliente por certificado.

- **rfc3161HTTPS.keystoreType:** Tipo de almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación HTTPS del cliente por certificado. Los valores posibles son:

  - **PKCS12**

  - **JKS**

  - **JCEKS**

- **rfc3161HTTPS.keystorePassword:** Contraseña del almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación HTTPS del cliente por certificado.

- **rfc3161SSL.portNumber:** Número del puerto donde se encuentre desplegado el servicio RFC 3161 que permite autenticación por SSL.

- **rfc3161SSL.keystorePath:** Ruta al almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación por SSL.

- **rfc3161SSL.keystoreType:** Tipo de almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación por SSL. Los valores posibles son:

  - **PKCS12**

  - **JKS**

  - **JCEKS**

- **rfc3161SSL.keystorePassword:** Contraseña del almacén de claves donde se encuentra almacenada la clave privada a usar para la autenticación por SSL.

8.2.4 Configuraciones de acceso a @firma

Archivo con configuraciones de acceso a @firma. El nombre del fichero queda a elección del integrador y se definirá en el fichero mappingFiles.properties. Se propone como nombre afirmaidClienteidAplicacion.properties siendo idCliente el identificador del cliente que envía la petición e idAplicacion el identificador de la aplicación @firma utilizada para la comunicación con @firma.

8.2.4.1 Parámetros Específicos a Cada Una de las Aplicaciones Configuradas en la Plataforma de @Firma:

- **secureMode:** Indicador de invocación del servicio web mediante un canal seguro. Los valores posibles son:
  - **true** → La invocación al servicio web se hará mediante un canal seguro.
  - **false** → La invocación al servicio web se hará mediante un canal no seguro.
- **endPoint:** IP y puerto donde se encuentran publicados los distintos WS. Sigue el formato **IP:Puerto** donde:
  - **IP** es la IP donde se encuentran publicados los distintos WS
  - **Puerto** es el número del puerto donde se encuentran publicados los distintos WS
- **servicePath:** Ruta donde se encuentran publicados los distintos servicios web de @Firma.
- **callTimeout:** Tiempo de espera máximo, en milisegundos, para una solicitud de servicio.
- **authorizationMethod:** Método de autorización de ejecución de servicios Web empleado. Los valores posibles son:
  - **none** → No se utilizará ningún método de autorización.
  - **UsernameToken** → Se utilizará autorización mediante usuario y contraseña.
  - **BinarySecurity** → Se utilizará autorización mediante certificado digital.
- **authorizationMethod.user:** Nombre de usuario para el método de autorización **UsernameToken**, o alias del certificado (*.p12 o *.pfx) empleado para el método de autorización **BinarySecurity**.
- **authorizationMethod.password:** Contraseña del usuario para el método de autorización **UsernameToken**, o contraseña del certificado (*.p12 o *.pfx) empleado para el método de autorización **BinarySecurity**.
- **authorizationMethod.passwordType:** Modo en que se transmitirá la contraseña del usuario indicado en el método de autorización **UsernameToken**. Los valores posibles son:
  - **clear** → Se envía la contraseña en formato de texto plano.
  - **digest** → Se envía el resumen de la contraseña.
- **authorizationMethod.userKeystore:** Ruta al almacén de certificados donde se encuentra almacenado el certificado empleado para el método de autorización **BinarySecurity**.

- **authorizationMethod.userKeystorePassword:** Contraseña del almacén de certificados donde se encuentra almacenado el certificado empleado para el método de autorización **BinarySecurity**.

- **authorizationMethod.userKeystoreType:** Tipo del almacén de certificados donde se encuentra almacenado el certificado empleado para el método de autorización **BinarySecurity**. Los valores posibles son:

    - **JKS**
    - **PKCS12**
    - **JCEKS**

- **response.validate:** Indicador para validar la firma de la respuesta de los servicios de @Firma, en caso de encontrarse firmada. Los valores posibles son:

    - **true** → Se validará la firma de las respuestas de los servicios de @Firma que se encuentren firmadas.
    - **false** → No se validará la firma de las respuestas de los servicios de @Firma que se encuentren firmadas.

- **response.certificateAlias:** Alias empleado para identificar la clave pública del certificado usado en la firma de la respuesta de los servicios de @Firma. La clave pública estará almacenada en el almacén de claves indicado en la propiedad **authorizationMethod.userKeystore**.

8.2.5 Configuraciones de acceso a eVisor

Archivo con configuraciones de acceso a eVisor. El nombre del fichero queda a elección del integrador y se definirá en el fichor mappingFiles.properties. Se propone como nombre evisoridClienteidAplicacion.properties siendo idCliente el identificador del cliente que envía la petición e idAplicacion el identificador de la aplicación eVisor utilizada para la comunicación con eVisor.

8.2.5.1 Parámetros Específicos a Cada Una de las Aplicaciones Configuradas en la Plataforma de eVisor:

- **secureMode:** Indicador de invocación del servicio web mediante un canal seguro. Los valores posibles son:

    - **true** → La invocación al servicio web se hará mediante un canal seguro.
    - **false** → La invocación al servicio web se hará mediante un canal no seguro.

- **endPoint:** IP y puerto donde se encuentran publicados los distintos WS. Sigue el formato **IP:Puerto** donde:
    - **IP** es la IP donde se encuentran publicados los distintos WS
    - **Puerto** es el número del puerto donde se encuentran publicados los distintos WS
- **servicePath:** Ruta donde se encuentran publicados los distintos servicios web de @Firma.
- **callTimeout:** Tiempo de espera máximo, en milisegundos, para una solicitud de servicio.
- **authorizationMethod:** Método de autorización de ejecución de servicios Web empleado. Los valores posibles son:
    - **none** → No se utilizará ningún método de autorización.
    - **UsernameToken** → Se utilizará autorización mediante usuario y contraseña.
    - **BinarySecurity** → Se utilizará autorización mediante certificado digital.
- **authorizationMethod.user:** Nombre de usuario para el método de autorización **UsernameToken**, o alias del certificado (*.p12 o *.pfx) empleado para el método de autorización **BinarySecurity**.
- **authorizationMethod.password:** Contraseña del usuario para el método de autorización **UsernameToken**, o contraseña del certificado (*.p12 o *.pfx) empleado para el método de autorización **BinarySecurity**.
- **authorizationMethod.passwordType:** Modo en que se transmitirá la contraseña del usuario indicado en el método de autorización **UsernameToken**. Los valores posibles son:
    - **clear** → Se envía la contraseña en formato de texto plano.
    - **digest** → Se envía el resumen de la contraseña.
- **authorizationMethod.userKeystore:** Ruta al almacén de certificados donde se encuentra almacenado el certificado empleado para el método de autorización **BinarySecurity**.
- **authorizationMethod.userKeystorePassword:** Contraseña del almacén de certificados donde se encuentra almacenado el certificado empleado para el método de autorización **BinarySecurity**.
- **authorizationMethod.userKeystoreType:** Tipo del almacén de certificados donde se encuentra almacenado el certificado empleado para el método de autorización **BinarySecurity**. Los valores posibles son:

- **JKS**
- **PKCS12**
- **JCEKS**

- **response.validate:** Indicador para validar la firma de la respuesta de los servicios de @Firma, en caso de encontrarse firmada. Los valores posibles son:
  - **true** → Se validará la firma de las respuestas de los servicios de @Firma que se encuentren firmadas.
  - **false** → No se validará la firma de las respuestas de los servicios de @Firma que se encuentren firmadas.
- **response.certificateAlias:** Alias empleado para identificar la clave pública del certificado usado en la firma de la respuesta de los servicios de @Firma. La clave pública estará almacenada en el almacén de claves indicado en la propiedad **authorizationMethod.userKeystore**.