



MANUAL DE USUARIO

PARA LA DEFINICIÓN DE PLANTILLAS

DE FIRMA PDF

© Izenpe s.a. 2016

Este documento es propiedad de Izenpe, s.a. y su contenido es confidencial. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de Izenpe, s.a.. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. Izenpe, s.a. no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

■ Beato Tomás de Zumárraga
71 - 1^a Planta
01008 Vitoria - Gasteiz

www.izenpe.com
info@izenpe.com
Tel.: 945 06 77 23

1. INTRODUCCIÓN	3
2. ESTRUCTURA DE PLANTILLAS	4
2.1. ELEMENTO <CSS:PDFSIGNATUREINFO>	4
2.1.1. ELEMENTO <CSS.PDFATTRIBUTES>	4
2.1.2. ELEMENTO <CSS:APPEARANCE>.....	7
2.1.3. ELEMENTO <CSS:RECT>	8
2.1.4. ELEMENTO <CSS:BACKGROUND>.....	9
2.1.5. ELEMENTO <CSS:FOREGROUND>.....	9
2.1.6. ELEMENTO <CSS:IMAGE>	10
2.1.7. ELEMENTO <CSS:TEXT>	11
3. ESTRUCTURA PETICIÓN ZAIN	14
3.1. ELEMENTO <DSS:SIGNREQUEST>.....	14
3.1.1. ELEMENTO <DSS:OPTIONALINPUTS>	14
3.1.2. ELEMENTO <DSS:INPUTDOCUMENTS>.....	17
3.2. ELEMENTO <DSS:SIGNRESPONSE>	18
3.2.1. ELEMENTO <DSS:OPTIONALOUTPUTS>.....	18
4. EJEMPLOS	19
4.1. PLANTILLA EN XML	19
4.2. JAVA	20
4.2.1. SMARTWRAPPER.....	20
4.2.2. AXIS.....	21
4.3. .NET	21

INTRODUCCIÓN

En el presente documento se describen un conjunto de estructuras XML que se utilizan tanto en los protocolos de generación, verificación y actualización de firmas de documentos PDF como en la definición de plantillas de firma PDF.

Importante

Se desaconseja acceder el servicio de actualización de firmas bajo el perfil PDF con la finalidad de extender la longevidad de la firma de un PDF. Si esa fuera su intención deberá firmar el documento PDF según el estándar CAdES (i.e utilizando el perfil CMS/PKCS #7) y posteriormente realizar las actualizaciones pertinentes sobre la firma resultante. Por otro lado, el servicio de actualización de firmas sólo permite actualizar una firma de PDF (i.e. bajo el perfil PDF) cuando el tamaño del PKCS #7 resultante (i.e. de la firma actualizada) no sobrepase el espacio reservado en el PDF para la firma. Además, si el documento PDF contiene más de una firma, el servicio de actualización de firmas sólo actualizará la última (porque la actualización de una firma invalida todas las firmas posteriores).

El tamaño que reserva ZAIN en los PDFs para las firmas es de 80kb.

ESTRUCTURA DE PLANTILLAS

A continuación se detallan los elementos que componen las plantillas de firma PDF.

ELEMENTO <CSS:PDFSIGNATUREINFO>

El elemento `<css:PdfSignatureInfo>` contiene información sobre los atributos de firma propios de Adobe. La estructura de este elemento es la siguiente:

```
<xs:element name="PdfSignatureInfo" type="css:PdfSignatureInfoType"
maxOccurs="unbounded"/>
<xs:complexType name="PdfSignatureInfoType">
    <xs:all>
        <xs:element name="PdfAttributes" type="css:PdfAttributesType"/>
        <xs:element name="Appearance" type="css:AppearanceType"
minOccurs="0"/>
    </xs:all>
    <xs:attribute name="name" type="xs:string" use="optional"/>
    <xs:attribute name="order" type="xs:integer" use="optional"/>
</xs:complexType>
```

Donde puede observarse que el elemento `<css:PdfSignatureInfo>` incluye los siguientes elementos:

Elemento <css.PdfAttributes>

Configuración del modo de firma. El elemento `<css:PdfAttributes>` se puede usar en dos contextos básicos:

Configuración de los parámetros relacionados con el modo de firma.

Obtención de información sobre el modo de firma en operaciones de verificación.

Y su definición es la siguiente:

```
<xs:element name="PdfAttributes" type="css:PdfAttributesType"/>
<xs:complexType name="PdfAttributesType">
    <xs:sequence>
        <xs:element name="certified" minOccurs="0">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
```

```
<xs:enumeration value="0"/>
<xs:enumeration value="1"/>
<xs:enumeration value="2"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="validationMethod">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="PPKMS"/>
<xs:enumeration value="PPKLITE"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="signaturePosition">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="FIRST"/>
<xs:enumeration value="LAST"/>
<xs:enumeration value="ADDLAST"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="params" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element name="location" type="xs:string"
minOccurs="0"/>
<xs:element name="reason" type="xs:string"
minOccurs="0"/>
<xs:element name="contactInfo" type="xs:string"
minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

Donde puede observarse que el elemento `<css:PdfAttributes>` contiene una secuencia de los siguientes elementos:

Elemento `<css:signatureAlg>`

Indica el tipo de firma PDF. La siguiente tabla describe los valores soportados por este elemento.

Valor	Tipo de firma
DETACHED	Firma PKCS #7 detached sobre los datos del PDF (adbe.pkcs7.detached). Adobe recomienda emplear el algoritmo DETACHED para un mejor cumplimiento de los estándares.

Elemento <css:validationMethod>

Selecciona el plug-in con el que validar la firma PDF en AdobeReader. La siguiente tabla describe los valores soportados por este elemento.

Valor	Plug-in de validación
PPKMS	Validador de Microsoft
PPKLITE	Validador de Adobe.

Elemento <css:signaturePosition>

Indica el emplazamiento de la apariencia de la firma. La siguiente tabla describe los valores soportados por este elemento.

Valor	Emplazamiento
FIRST	Primera página
LAST	Última página
ADDLAST	Página añadida expresamente al final del documento

Elemento <css:params>

Define las propiedades de una firma PDF (que corresponden a entradas de su diccionario) mediante los siguientes elementos :

- <css:location>: Lugar de firma (indicado mediante una cadena de texto).

Corresponde a la entrada de clave Location del diccionario de la firma.

- <css:reason>: Compromiso de firma (indicado mediante una cadena de texto).

Corresponde a la entrada de clave Reason del diccionario de la firma.

- <css:contactInfo>: Información de contacto del firmante (e.g. número de teléfono)

indicado mediante una cadena de texto. Corresponde a la entrada de clave

ContactInfo del diccionario de la firma.

Nota

Debe tenerse en cuenta que el elemento <css:certified> no se utiliza en las definición de plantillas PDF.

Ejemplo de elemento <css:PdfAttributes>

El siguiente código XML es un ejemplo de elemento <css:PdfAttributes>.

```
<css:PdfAttributes>
  <css:validationMethod>PPKMS</css:validationMethod>
  <css:signaturePosition>LAST</css:signaturePosition>
  <css:params>
    <css:location>Vitoria</css:location>
    <css:reason>Razon de la firma</css:reason>
    <css:contactInfo>+34 555 666 777</css:contactInfo>
  </css:params>
</css:PdfAttributes>
```

Elemento <css:Appearance>

Parámetros de la apariencia de firma. Si este elemento no está presente, se genera una firma de tipo invisible. Define el formato de la apariencia de la firma. La estructura de este elemento es la siguiente:

```
<xs:element name="Appearance" type="css:AppearanceType" minOccurs="0"/>
<xs:complexType name="AppearanceType">
  <xs:sequence>
```

```
<xs:element name="Rect" type="css:RectType" minOccurs="0"/>
<xs:element name="Background" type="css:BackgroundType"
minOccurs="0"/>
<xs:element name="Foreground" type="css:ForegroundType"
minOccurs="0"/>
</xs:sequence>
</xs:complexType>
```

Donde puede observarse que el elemento `<css:Appearance>` contiene los siguientes elementos:

Los parámetros de este elemento son los siguientes:

<css:Rect>

Espacio donde se incrustará la apariencia si es necesario crear un nuevo campo de firma.
Si se trabaja con un campo de firma vacío, este elemento no tiene efecto alguno.

<css:Background>

Imagen de fondo de la apariencia de la firma.

<css:Foreground>

Imagen en primer plano de la apariencia de la firma.

Elemento <css:Rect>

El elemento `<css:Rect>` indica dónde situar una firma que deba mostrarse (i.e una firma visible, con apariencia) . Su definición es la siguiente:

```
<xs:complexType name="RectType">
<xs:attribute name="x0" type="xs:positiveInteger"/>
<xs:attribute name="y0" type="xs:positiveInteger"/>
<xs:attribute name="x1" type="xs:positiveInteger"/>
<xs:attribute name="y1" type="xs:positiveInteger"/>
</xs:complexType>
```

Donde se observa que los atributos x0, y0, x1, y1 definen las coordenadas del rectángulo en el que se muestra la firma. Las coordenadas origen (0,0) se sitúan en la esquina inferior izquierda de la página.

- x0, y0: Punto inferior izquierdo (expresado en píxeles).
- x1, y1: Punto superior derecho (expresado en píxeles).

Elemento <css:Background>

El elemento <css:Background> define la imagen de fondo (en segundo plano) de una firma con apariencia. Su definición es la siguiente:

```
<xs:element name="Background" type="css:BackgroundType"/>
<xs:complexType name="BackgroundType">
    <xs:all>
        <xs:element name="image" type="css:ImageType"/>
    </xs:all>
</xs:complexType>
```

Donde se observa que el elemento <css:Background> sólo contiene el elemento <css:image>, el cual se describe a continuación.

Elemento <css:Foreground>

El elemento <css:Foreground> define la imagen (en primer plano) y el texto de la firma. Su definición es la siguiente:

```
<xs:element name="Foreground" type="css:ForegroundType" minOccurs="0"/>
<xs:complexType name="ForegroundType">
    <xs:all>
        <xs:element name="image" type="css:ImageType"/>
        <xs:element name="text" type="css:TextType"/>
    </xs:all>
</xs:complexType>
```

Donde se observa que el elemento <css:Foreground> contiene los dos siguientes elementos:

- <css:image>: Imagen de la firma.
- <css:text>: Texto con información sobre la firma.

Elemento <css:image>

El elemento <css:image> define la imagen de una firma. Su definición es la siguiente:

```
<xss:element name="image" type="css:ImageType"/>
<xss:complexType name="ImageType">
  <xss:all>
    <xss:element name="data" type="xs:string"/>
    <xss:element name="imageSize" minOccurs="0">
      <xss:complexType>
        <xss:attribute name="width" type="xs:positiveInteger"/>
        <xss:attribute name="height"
          type="xs:positiveInteger"/>
      </xss:complexType>
    </xss:element>
    <xss:element name="position" type="css:PositionType"
      minOccurs="0"/>
  </xss:all>
  <xss:attribute name="encodeType">
    <xss:simpleType>
      <xss:restriction base="xs:string">
        <xss:enumeration value="uri"/>
        <xss:enumeration value="base64"/>
      </xss:restriction>
    </xss:simpleType>
  </xss:attribute>
</xss:complexType>
```

Donde puede observarse que el elemento <css:image> contiene los siguientes atributos:

- //image/@encodeType: Indica la codificación utilizada por el elemento <css:data> para definir la imagen de firma. La siguiente tabla describe los valores soportados por este atributo:

Valor de encodeType	Contenido de <css:data>
uri	Referencia al fichero donde está contenida la imagen.
base64	Imagen codificada en base64

y los siguientes elementos:

- <css:data>: Imagen de la firma codificada del modo que indica el atributo //image/@encodeType. El formato de la imagen debe ser JPEG.
- <css:imageSize>: indica, mediante los siguientes atributos, el tamaño de visualización de la imagen:
 - //imageSize/@width: Ancho de la imagen (en píxeles).
 - //imageSize/@height: Altura de la imagen (en píxeles).

Nota: Si este elemento se omite, el tamaño de visualización será el real.

- <css:position>: Indica la posición relativa de la imagen dentro del espacio que define el elemento <css:Rect>. En concreto, este elemento establece la coordenada inferior izquierda de la imagen mediante los siguientes atributos:
 - //position/@x: Desplazamiento horizontal (en píxeles).
 - //position/@y: Desplazamiento vertical (en píxeles).

Elemento <css:text>

El elemento <css:text> contiene un texto con información sobre la firma. Su definición es la siguiente:

```
<xs:element name="text" type="css:TextType"/>
<xs:complexType name="TextType">
    <xs:all>
        <xs:element name="properties" minOccurs="0">
            <xs:complexType>
                <xs:attribute name="color" type="xs:string"/>
                <xs:attribute name="fontSize"
type="xs:positiveInteger"/>
            </xs:complexType>
        </xs:element>
    </xs:all>
</xs:complexType>
```

```
<xs:element name="position" type="css:PositionType"
minOccurs="0"/>
<xs:element name="SignatureInfos" type="css:SignatureInfosType"
minOccurs="0"/>
</xs:all>
</xs:complexType>
<xs:element name="SignatureInfos" type="css:SignatureInfosType" minOccurs="0"/>
<xs:complexType name="SignatureInfosType">
<xs:sequence>
<xs:element name="signatureInfo" maxOccurs="unbounded">
<xs:complexType>
<xs:attribute name="title" type="xs:string"/>
<xs:attribute name="id">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration
value="Subject"/>
<xs:enumeration
value="Issuer"/>
<xs:enumeration
value="SerialNumber"/>
<xs:enumeration
value="Reason"/>
<xs:enumeration
value="Location"/>
<xs:enumeration
value="ContactInfo"/>
<xs:enumeration
value="Date"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

Donde se observa que el elemento <css:text> contiene los siguientes elementos:

- <css:properties> [Opcional]. Define, mediante los siguientes atributos, el color y el tamaño del texto:
 - //properties/@color: Color del texto (en notación RGB, con valores entre 0 y 1).
 - //properties/@fontSize: Tamaño de la fuente.

- <css:position> [Opcional]. Indica la posición relativa del texto dentro del espacio que define el elemento <css:Rect>. En concreto, este elemento indica la coordenada inferior izquierda del texto mediante los siguientes atributos.

- //position/@x: Desplazamiento horizontal (en píxeles). //position/@y: Desplazamiento vertical (en píxeles).

- <css:SignatureInfos> [Opcional]: Incluye una secuencia de elementos <css:SignatureInfo>, cada uno de los cuales corresponde a un campo textual que se integra en el texto de la firma y tiene los siguientes atributos:

- //SignatureInfo/@title: Etiqueta que se muestra junto al campo textual al representar la firma gráficamente.
 - //SignatureInfo/@id: Identificador del campo textual. Éste puede corresponder tanto a campos del certificado (Subject, Issuer, SerialNumber), como a propiedades de la firma del PDF (Reason, Location, ContactInfo, Date) que estén contenidas en su diccionario. Los valores que puede tener son los siguientes:

Campo	Descripción
Subject	Titular del certificado de firma.
Issuer	Autoridad de Certificación (CA) que ha emitido el certificado de firma.
SerialNumber	Número de serie del certificado de firma.
Reason	Razón de la generación de la firma.
Location	Lugar de firma.
ContactInfo	Información de contacto de la persona/entidad que ha realizado la firma.
Date	Fecha de la firma.

ESTRUCTURA PETICIÓN ZAIN

Este capítulo detalla los elementos y atributos necesarios para la construcción de una petición de generación de firma según el perfil de firma PDF.

ELEMENTO <DSS:SIGNREQUEST>

El elemento `<dss:SignRequest>` permite gestionar el contenido de una petición de firma de documento PDF. Este elemento incluye los siguientes atributos:

- *Profile* [Obligatorio]. El valor de este atributo debe ser el siguiente:
urn:safelayer:tws:dss:1.0:profiles:pdf:1.0:sign
- *RequestID* [Opcional]. Identificador de la petición. Este atributo sirve para enlazar peticiones y respuestas en un entorno asíncrono (porque el atributo RequestID de la correspondiente respuesta tendrá el mismo valor).

El elemento `<dss:SignRequest>` incluye los siguientes elementos:

- `<dss:OptionalInputs>` [Obligatorio].
- `<dss:InputDocuments>` [Opcional].

Elemento <DSS:OPTIONALINPUTS>

La petición de firma `<dss:SignRequest>` debe incluir un elemento `<dss:OptionalInputs>`.

Los elementos de entrada opcionales para el perfil de firma PDF son los siguientes:

- `<dss:ServicePolicy>` [Opcional]. Política de generación de firma deseada (entre las políticas equivalentes seleccionadas por Zain para dicha operación). Si el usuario no incluye este elemento, Zain utiliza la primera política de firma que cumpla las políticas de seguridad del entorno de ejecución.
- `<dss:Language>` [Opcional].
- `<dss:KeySelector>` [Obligatorio]. Identificador de la clave para realizar la firma.

- <*dss:Properties*> [Opcional]. Propiedades de la firma.
- <*css:StoreSignatureField*> [Opcional]. Campo de firma donde adjuntar la firma.
- <*css:SignatureForm*> [Opcional]. Formato de firma deseado.

Importante: Cuando este elemento se omite, la firma no incluye el atributo *signing-certificate*. Esto es, el formato de la firma no puede considerarse equivalente al formato PKCS #7 de una firma CAdES-BES (y por tanto es una firma NoAdES).

- <*css:SignatureDigestAlgorithm*> [Opcional]. Algoritmo de hash utilizado para generar la firma PKCS #7/CMS contenida en el documento PDF.
- <*css:ReturnDocumentHash*> [Opcional]. Sigue la respuesta incluya el resumen (hash) de los datos firmados.
- <*css:ExcludeChainCertificates*> [Opcional]. Sigue que no se incluyan los certificados de la cadena de certificación en la firma. Cuando este elemento se omite, se incluyen todos los certificados de la cadena certificación en las firmas del perfil PDF (aunque el certificado raíz no se incluye como certificado de la cadena de certificación).
- <*css:ExcludeSignerCertificate*> [Opcional].

Elemento <*dss:Language*>

El elemento opcional <*dss:Language*> indica el idioma en el que deben devolverse los mensajes de error del servicio Zain. Este elemento soporta los siguientes valores: es (Español) en (Inglés). El idioma por defecto es el inglés.

Elemento <*dss:Properties*>

El elemento <*dss:Properties*> puede contener un elemento <*dss:SignedProperties*> que, a su vez, incluya un conjunto de elementos <*dss:Property*>. Los identificadores de propiedad soportados por el perfil de PDF son los siguientes:

Nota: La inclusión de las propiedades location, policies, roles y commitments en la firma depende de la política de generación de firma aplicada.

- Lugar de realización de la firma: urn:safelayer:tws:dss:1.0:property:location
- Políticas de firma: urn:safelayer:tws:dss:1.0:property:policies
- Roles de firma: urn:safelayer:tws:dss:1.0:property:roles.
- Compromisos: urn:safelayer:tws:dss:1.0:property:commitments
- Propiedades de firma PDF: urn:safelayer:tws:dss:1.0:property:pdfattributes

El valor de esta propiedad es un elemento <css:PdfSignatureInfo> con información sobre los atributos de firma propios de Adobe.

Nota: Al generar la firma PDF, las características de firma definidas en <css:PdfSignatureInfo> sólo se aplican si la política de generación de firma utilizada permite al cliente seleccionar la plantilla PDF.

Elemento <css:SignatureForm>

El elemento <css:SignatureForm> permite indicar el formato deseado para la firma. Los posibles valores de este elemento son los siguientes:

- Firma electrónica básica (según ETSI):
urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:BES
- Firma con sello de tiempo (según ETSI):
urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-T
- Firma electrónica con política explícita (según ETSI)
urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:EPES
- Firma electrónica en un formato no ETSI (i.e. firma no avanzada)
urn:safelayer:tws:dss:1.0:profiles>NoAdES

Nota: El formato de firma definido en <css:SignatureForm> sólo se aplica a la generación de firma PDF si la política utilizada permite al cliente seleccionar el formato de firma

Elemento <css:ExcludeSignerCertificate>

El elemento opcional <css:ExcludeSignerCertificate> solicita que la firma no incluya el certificado del firmante ni los certificados de su cadena de certificación. Cuando este elemento se omite, las firmas siempre incluyen el certificado del firmante y todos los certificados de la cadena certificación (aunque el certificado raíz no se incluye como certificado de la cadena de certificación).

Nota: Si no se incluye en la firma, el certificado del firmante debe añadirse mediante el “optional input” <css:SignersCertificates> para poder verificar la firma.

Elemento <dss:InputDocuments>

El elemento <dss:InputDocuments> contiene los datos que se desean firmar. Según el tipo de firma esperado, este elemento contendrá alguno de los siguientes datos de entrada:

- Firmas simples.
- Firmas múltiples en serie.

Generación de firmas simples

Para generar firmas simples, el perfil PDF soporta el documento de entrada <dss:Document>. Este elemento contiene, en un elemento <dss:Base64Data>, un documento PDF codificado en base64.

Nota: El elemento <dss:Base64Data> contiene un atributo *MimeType* con el valor *application/pdf*.

Generación de firmas múltiples en serie

Para generar firmas múltiples en serie, el perfil PDF soporta el documento de entrada `<dss:Document>`. Este elemento contiene, en un elemento `<dss:Base64Data>`, la codificación base64 de un documento PDF con varias firmas.

Nota: El elemento `<dss:Base64Data>` contiene un atributo `MimeType` con el valor `application/pdf`.

ELEMENTO <DSS:SIGNRESPONSE>

El elemento `<dss:SignResponse>` incluye los siguientes atributos:

- *Profile* [Obligatorio]. El valor de este atributo debe ser el siguiente:
`urn:safelayer:tws:dss:1.0:profiles:pdf:1.0:sign`
- *RequestID* [Opcional]. Este atributo debe tener el mismo valor que el atributo *RequestID* del elemento `<dss:SignRequest>` de la correspondiente petición (si está presente).

El elemento `<dss:SignResponse>` incluye los siguientes elementos:

- `<dss:Result>`. Resultado de la operación.
- `<dss:OptionalOutputs>`.

Elemento <DSS:OptionalOutputs>

Según el resultado de la operación y las opciones solicitadas en la petición, el elemento `<dss:OptionalOutputs>` contendrá los siguientes elementos:

- `<dss:DocumentWithSignature>`
- `<dss:DocumentHash>` Resumen (hash) de los datos firmados. Presente si se solicita en la petición con el elemento `<css:ReturnDocumentHash>`.

Elemento <DSS:DocumentWithSignature>

Si TWS-DS ha podido generarla, la firma digital se devuelve en un elemento `<dss:DocumentWithSignature>`. Dicho elemento contiene un elemento `<dss:XMLData>`

que, a su vez, incluye un elemento `<dss:Base64Data>`. El elemento `<dss:Base64Data>` contiene la codificación base64 del documento PDF con una firma embebida.

Nota: El elemento `<dss:Base64Data>` contiene un atributo `MimeType` con el valor `application/pdf`.

EJEMPLOS

A continuación se muestran una serie de ejemplos, tanto de plantillas como de llamadas a ZAIN, solicitando la generación de una firma PDF según una plantilla determinada en la propia petición SOAP.

Para los ejemplos se ha utilizado la misma plantilla que por defecto aplica ZAIN, en caso de no especificar ninguna en la petición.

PLANTILLA EN XML

```
<PdfSignatureInfo xmlns="http://www.safelayer.com/TWS">
  <PdfAttributes>
    <validationMethod>PPKMS</validationMethod>
    <signaturePosition>LAST</signaturePosition>
  </PdfAttributes>
  <Appearance>
    <Rect x0="15" x1="580" y0="5" y1="25"/>
    <Foreground>
      <image encodeType="base64">
        <data>/9j/...]==</data>
        <imageSize height="13" width="20"/>
        <position x="0" y="1"/>
      </image>
      <text>
        <properties color="0 0 0" fontSize="3"/>
        <position x="30" y="1"/>
      </text>
      <SignatureInfos>
        <signatureInfo id="Subject" title="Firmante:"/>
        <signatureInfo id="Issuer" title="Emisor del certificado:"/>
        <signatureInfo id="SerialNumber" title="Número de serie:"/>
        <signatureInfo id="Date" title="Fecha de firma:"/>
      </SignatureInfos>
    </text>
  </Foreground>
```

```
</Appearance>
</PdfSignatureInfo>
```

JAVA

SMARTWRAPPER

En el siguiente ejemplo, se muestra como realizar la generación de firma basáandonos en una plantilla que tenemos creada en un fichero xml.

Suponemos que la plantilla está ya creada, por lo que lo único que hay que hacer es configurar la petición con los parámetros correspondientes, entre los que se incluye la definición de dicha plantilla.

```
/*
 * Se crea la petición a través de la URL definida en ZAIN_ENDPOINT.
 */
SmartSignRequest smartSignReq = new
SmartSignRequest(Propiedades.ZAIN_ENDPOINT);

// Se establece el perfil de firma PDF en la petición de Generación
// de firma.
smartSignReq.setProfile(Constants.Profile.PDF);

/*
 * Se establece el idioma de la petición en español. 2
 * posibilidades: 1. Español: 'es' 2. Inglés: 'en'
 */
smartSignReq.setLanguage("es");

/*
 * Cuando este elemento se omite, la firma no incluye el atributo
 * signingcertificate. Esto es, el formato de la firma no puede
 * considerarse equivalente al formato PKCS #7 de una firma
 * CAdES-BES (y por tanto es una firma NoAdES).
 */
smartSignReq.setSignatureFormat(Constants.SignatureFormat.ES_T);

// Certificado con el que realizar la firma.
smartSignReq.setKeySubjectName(Propiedades.ZAIN_SIGNER);

// Datos a firmar.

smartSignReq.setInputPdfBase64Data(Base64.encode(ZainUtil.readBinaryFi
le("data/input/" + fileName)));
```

```
// Asignación de una plantilla
    SignRequest signReq = (SignRequest)
smartSignReq.getInternalObject();
    OptionalInputs optionalInputs = signReq.getOptionalInputs();

    Properties props = new Properties();
    PropertiesType signedProp = new PropertiesType();
    Property prop = new Property();
    prop.setIdentifier(new
URI("urn:safelayer:tws:dss:1.0:property:pdffattributes"));

    InputStream plantillaPdf = new
FileInputStream("resources/plantilla-pdf.xml");
    org.w3c.dom.Document document =
XMLUtils.newDocument(plantillaPdf);
    MessageElement messageElement = new
MessageElement(document.getDocumentElement());
    AnyType XMLData = new AnyType(new MessageElement[] {
messageElement });
    prop.setValue(XMLData);

    signedProp.setProperty(new Property[] { prop });
    props.setSignedProperties(signedProp);

    optionalInputs.setProperties(props);
// Fin asignación plantilla

// Envío de la petición.
SmartSignResponse smartSignResp = smartSignReq.send();
```

AXIS

La generación de una firma con perfil PDF, especificando en la petición la plantilla a utilizar es equivalente a como se hace con SmartWrapper.

.NET

A continuación se describe la forma en la que se realiza la generación de la plantilla para la posterior generación de firma en .NET.

```
/*
 * Se inicializa la conexión del servicio web. Para ello, se
 * localiza al servicio de la variable ZAIN_ENDPOINT:
 */
TWSDS ds = new TWSDS();
```

```

ds.Url = Propiedades.ZAIN_ENDPOINT;

//Se crea la petición de Generación de firma PDF.
SignRequest sr = new SignRequest();

/*
 * Se establece el perfil de firma PDF en la petición de
Generación
 * de firma:'urn:safelayer:tws:dss:1.0:profiles:pdf:1.0:sign'
*/
sr.Profile = Constantes.PROF_SIGN_PDF;

/*
 * Se establece el idioma de la petición en español. 2 posibilidades:
*   1. Español: 'es'
*   2. Inglés: 'en'
*/
OptionalInputs optionalInputs = new OptionalInputs();
optionalInputs.Language = "es";

//Se establece el tipo de firma a PKCS#7.
optionalInputs.SignatureType = Constantes.TYPE_PKCS_7;

/*
 * Se escoge la clave con la que se firmará el documento. Para ello,
se añade:
*   1. El firmante, especificado en la variable BALMASEDA_SIGNER
*   2. El formato que se va adoptar, especificado en la variable
URI_DN:
*           'urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName'
*/
KeySelector ks = new KeySelector();
KeySelector1 ksl = new KeySelector1();
NameIdentifierType nit = new NameIdentifierType();
nit.Format = Constantes.URI_DN;
nit.Value = Propiedades.BALMASEDA_SIGNER;
ksl.Name = nit;
ks.Item = ksl;
optionalInputs.KeySelector = ks;

//Creamos la plantilla que queremos que ZAIN aplique al PDF
zain.ds.Properties props = new zain.ds.Properties();
Property signedProp = new Property();
signedProp.Identifier =
"urn:safelayer:tws:dss:1.0:property:pdfattributes";

PdfSignatureInfoType plantilla = new PdfSignatureInfoType();
PdfAttributesType attr = new PdfAttributesType();
attr.signatureAlg = PdfAttributesTypeSignatureAlg.DETACHED;
attr.validationMethod = PdfAttributesTypeValidationMethod.PPKMS;
attr.signaturePosition =
PdfAttributesTypeSignaturePosition.FIRST;

```

```
PdfAttributesTypeParams pdfParams = new  
PdfAttributesTypeParams();  
pdfParams.location = "Irun";  
pdfParams.reason = "Ayuntamiento de Irun - Irungo Udala";  
pdfParams.contactInfo = "+34 943 50 51 52";  
attr.@params = pdfParams;  
plantilla.PdfAttributes = attr;  
  
AppearanceType appearance = new AppearanceType();  
RectType rect = new RectType();  
rect.x0 = "7";  
rect.x1 = "550";  
rect.y0 = "753";  
rect.y1 = "870";  
appearance.Rect = rect;  
  
BackgroundType background = new BackgroundType();  
ImageType backImage = new ImageType();  
backImage.encodeType = ImageTypeEncodeType.base64;  
backImage.encodeTypeSpecified = true;  
ImageTypeImageSize backImageSize = new ImageTypeImageSize();  
backImageSize.height = "43";  
backImageSize.width = "30";  
backImage.imageSize = backImageSize;  
PositionType backImagePosition = new PositionType();  
backImagePosition.x = "10";  
backImagePosition.y = "10";  
backImage.position = backImagePosition;  
backImage.data = "/9j/4AA...";  
background.image = backImage;  
appearance.Background = background;  
  
ForegroundType foreground = new ForegroundType();  
ImageType foreImage = new ImageType();  
foreImage.encodeType = ImageTypeEncodeType.base64;  
foreImage.encodeTypeSpecified = true;  
ImageTypeImageSize foreImageSize = new ImageTypeImageSize();  
foreImageSize.height = "1";  
foreImageSize.width = "1";  
foreImage.imageSize = foreImageSize;  
PositionType foreImagePosition = new PositionType();  
foreImagePosition.x = "350";  
foreImagePosition.y = "120";  
foreImage.position = foreImagePosition;  
foreImage.data = "/9j/4AA... ";  
foreground.image = foreImage;  
  
TextType text = new TextType();  
TextTypeProperties textProps = new TextTypeProperties();  
textProps.color = "0 0 0";  
textProps.fontSize = "7";  
text.properties = textProps;
```

```
PositionType textPos = new PositionType();
textPos.x = "60";
textPos.y = "5";
text.position = textPos;
SignatureInfosTypeSignatureInfo subject = new
SignatureInfosTypeSignatureInfo();
subject.id = SignatureInfosTypeSignatureInfoID.Subject;
subject.idSpecified = true;
subject.title = "Autor de la firma";

SignatureInfosTypeSignatureInfo issuer = new
SignatureInfosTypeSignatureInfo();
issuer.id = SignatureInfosTypeSignatureInfoID.Issuer;
issuer.idSpecified = true;
issuer.title = "Emisor certificado";

SignatureInfosTypeSignatureInfo serialNumber = new
SignatureInfosTypeSignatureInfo();
serialNumber.id = SignatureInfosTypeSignatureInfoID.SerialNumber;
serialNumber.idSpecified = true;
serialNumber.title = "Numero de serie";

SignatureInfosTypeSignatureInfo reason = new
SignatureInfosTypeSignatureInfo();
reason.id = SignatureInfosTypeSignatureInfoID.Reason;
reason.idSpecified = true;
reason.title = "Razon";

SignatureInfosTypeSignatureInfo location = new
SignatureInfosTypeSignatureInfo();
location.id = SignatureInfosTypeSignatureInfoID.Location;
location.idSpecified = true;
location.title = "Localizacion";

SignatureInfosTypeSignatureInfo contactInfo = new
SignatureInfosTypeSignatureInfo();
contactInfo.id = SignatureInfosTypeSignatureInfoID.ContactInfo;
contactInfo.idSpecified = true;
contactInfo.title = "Informacion de contacto";

SignatureInfosTypeSignatureInfo date = new
SignatureInfosTypeSignatureInfo();
date.id = SignatureInfosTypeSignatureInfoID.Date;
date.idSpecified = true;
date.title = "Fecha firma";

text.SignatureInfos = new SignatureInfosTypeSignatureInfo[] {
subject, issuer, serialNumber, reason, location, contactInfo, date };
foreground.text = text;
appearance.Foreground = foreground;

plantilla.Appearance = appearance;
// Fin creación plantilla
```

```
//Serializacion del objeto plantilla. Lo convertimos en un objeto AnyType.
AnyType any = new AnyType();
TextWriter writer = new StringWriter();
XmlSerializer serializer = new XmlSerializer(plantilla.GetType(),
new XmlAttributeOverrides(), new Type[0], new
XmlAttributeOverrides("Manifest"), "http://www.w3.org/2000/09/xmldsig#");
serializer.Serialize(writer, plantilla);
string manisfestStr = writer.ToString();
 XmlDocument xmlDoc = new XmlDocument();
xmlDocument.LoadXml(manisfestStr);
XmlElement elem = xmlDoc.DocumentElement;
any.Any = new XmlElement[] { elem };
//Fin serialiacion

signedProp.Value = any;

props.SignedProperties = new Property[]{signedProp};
optionalInputs.Properties = props;
//

/*
 * Se añaden los parámetros opcionales a la petición de Generación
 * de firma.
 */
sr.OptionalInputs = optionalInputs;

/*
 * Contiene los datos que se van a firmar. El elemento `Document`
 * va dentro del elemento 'InputDocument'.
 */
InputDocuments inputsDocuments = new InputDocuments();
Document doc = new Document();

//Se añaden al documento los datos en base 64.
Base64Data base64 = new Base64Data();
base64.Value = System.IO.File.ReadAllBytes("../data/input/" +
fileName);
doc.Item = base64;
inputsDocuments.Items = new object[] { doc };
inputsDocuments.ItemsElementName = new ItemsChoiceType3[] {
ItemsChoiceType3.Document };
sr.InputDocuments = inputsDocuments;

//Se añade el certificado de cliente.
ds.ClientCertificates.Add(Propiedades.BALMASEDA_AUTH_CERT);

try
{
    System.Console.WriteLine(TxUtil.SerializeToString(sr));
    //Se envía la petición de Generación de firma PDF.
```



```
SignResponse response = ds.Sign(sr);
```

■ Beato Tomás de Zumárraga
71 - 1^a Planta
01008 Vitoria - Gasteiz

www.izenpe.com
info@izenpe.com
Tel.: 945 06 77 23