

Manual de Programación de Plantillas para la Generación de Informes

eVisor 2.1.1

Documento n°:	@Firma-eVisor-Progs-Templates-MAN
Revisión:	004
Fecha:	10-07-2015
Período de retención:	Permanente durante su período de vigencia + 3 años después de su anulación

CONTROL DE MODIFICACIONES

Documento n°: @Firma-eVisor-Progs-Templates-MAN
Revisión: 004
Fecha: 10-07-2015

Rev. 001
Fecha 22/11/2011
Descripción Documentación inicial

Rev. 002
Fecha 17/04/2012
Descripción Incluido formatos de los documentos soportados

Rev. 003
Fecha 15/05/2013
Descripción Modificado a la versión 2.1

Rev. 004
Fecha 10/07/2015
Descripción Modificado a la versión 2.1.1

CONTROL DE DISTRIBUCIÓN

Documento: @Firma-eVisor-Progs-Templates-MAN
Revisión: 004
Fecha: 10-07-2015

Propiedad del documento:

Este documento pertenece al Gobierno de España y posee un carácter de público para uso y distribución en ámbitos autorizados por este mismo, según se recoge en la declaración de privacidad.

Declaración de privacidad:

El contenido de este documento está sujeto al protocolo de libre distribución dentro del entorno definido para el contexto.

Copias Electrónicas:

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

Copias en Papel:

La vigencia de las copias impresas en papel está condicionada a la coincidencia de su estado de revisión con el que aparece en el sistema electrónico de distribución de documentos.

El control de distribución de copias en papel para su uso en proyectos u otras aplicaciones es responsabilidad de los usuarios del sistema electrónico de información.

Fecha de impresión 13 de julio de 2015

Índice

1	Objeto.....	6
2	Alcance.....	6
3	Siglas	7
4	Documentos de Referencia	7
5	Proceso de Generación de Informe	8
	5.1 Proceso de Generación de Informe PDF.....	8
	5.2 Proceso de Generación de Informes ODF	9
6	XML de entrada	11
	6.1 ValidationResult.....	13
	6.2 GenerationTime.....	14
	6.3 ExternalParameters	14
	6.4 DocumentInfo	15
	6.5 Barcodes.....	15
7	Diseño de plantillas PDF.....	16
	7.1 Diseño del fichero XSL-FO	16
	7.2 Diseño de la plantilla.....	20
	7.2.1 Imágenes	24
	7.2.2 Documento firmado	25
	7.2.3 Códigos de barras.....	28
8	Diseño de plantillas ODF.....	29
	8.1 Diseño del ODF base	29
	8.2 Diseño de la plantilla.....	31
	8.2.1 Implementación de los ficheros XSLT	33
	8.2.2 Empaquetado de la plantilla.....	35
A.1	Esquema del mensajes XML de entrada.....	37

A.2	Formatos de documentos soportados	40
-----	---	----

1 Objeto

El objeto de este manual es servir de guía para el diseño de plantillas de generación de informe de firma para la aplicación eVisor.

2 Alcance

Este documento cubre los siguientes aspectos:

- Descripción de los procesos de generación de informes de firma en la aplicación eVisor.
- Detallar los componentes que forman una plantilla de informe para facilitar la implementación de las mismas.
- Exposición del proceso de diseño de plantilla PDF y ODF con el desarrollo de ejemplos prácticos.

3 Siglas

TGS	Telvent Global Services
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
XSL-FO	eXtensible Stylesheet Language Formatting Objects
XSLT	XSL Transformations
W3C	World Wide Web Consortium

4 Documentos de Referencia

AFIRMA-DSS-MAN	@Firma-Global-XMLSOAP_PROFILE_DSS - Perfiles para la adaptación de los Web Services de @Firma 5 al protocolo OASIS-DSS.
EVISOR-PROG-MAN	@Firma-eVisor-Progs-XMLSOAP-MAN - Manual de Programación de Web Services
EVISOR-ADMIN-MAN	@Firma-eVisor-ADM-MAN - Manual de la Herramienta de Administración

5 Proceso de Generación de Informe

La aplicación eVisor permite, a partir de firmas electrónicas, generar informes de firma PDF u ODF en los que se incluya información asociada al proceso de firma del usuario, así este informe de firma podrá contener información del firmante, documento firmado, sello de tiempo, etc.

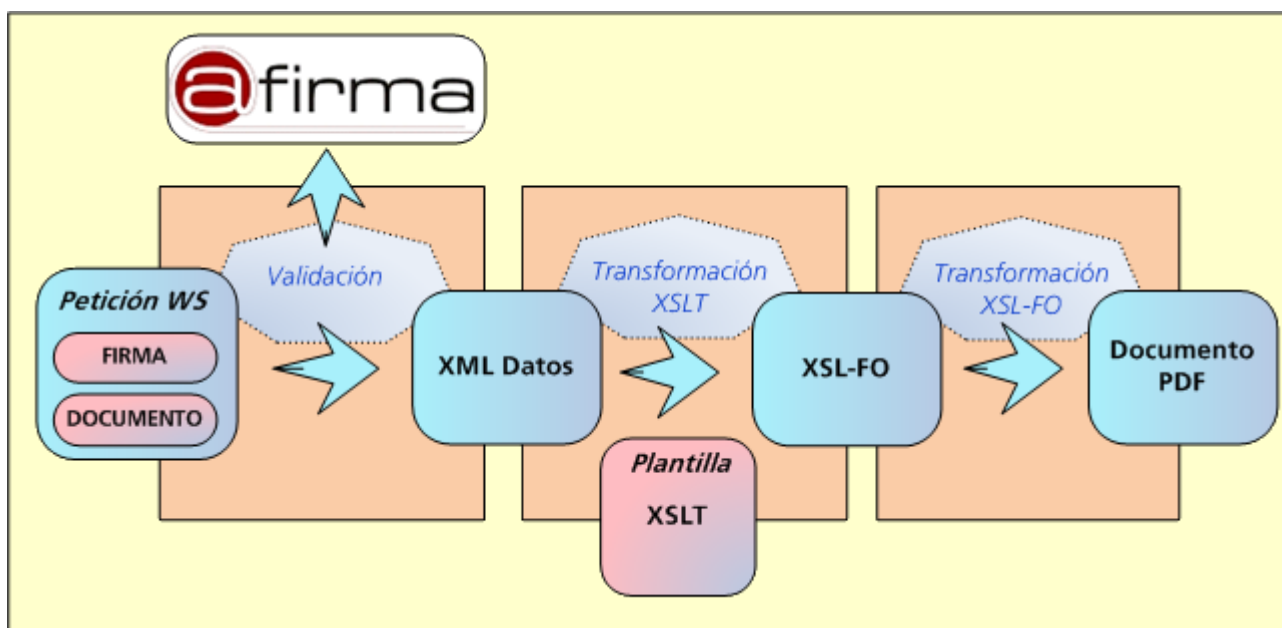
La generación de estos informes se realiza mediante la aplicación de plantillas registradas en el sistema en las cuales las aplicaciones configuran el formato en el que se presenta la información del reporte.

A continuación vamos a detallar el proceso de generación de informe en los formatos soportados ya que para llevar a cabo el diseño de una plantilla es necesario comprender los elementos que intervienen en dicho proceso.

5.1 Proceso de Generación de Informe PDF

La generación de informe PDF está basado en la utilización de XSL-FO (eXtensible Stylesheet Language Formatting Objects), XSL-FO es una recomendación W3C que mediante XML especifica cómo se van a formatear unos datos para presentarlos en pantalla, papel u otros medios (puede consultarse información sobre las especificaciones XSL-FO en <http://www.w3.org/TR/xsl11/>).

En la siguiente figura se muestra la secuencia de pasos del proceso de generación del PDF.



El proceso de generación es el siguiente:

1. Una aplicación externa solicita mediante petición WS la generación de un informe de firma, la petición incluye la firma electrónica, los datos firmados e información adicional para la generación del informe de firma.
2. El sistema solicita la validación de la firma electrónica a la plataforma de validación y firma @firma.

3. A partir del resultado de validación de @firma y los parámetros adicionales incluidos en la petición el sistema construye un XML previamente definido.
4. La aplicación previamente habrá registrado en el sistema una plantilla XSLT que permita obtener un documento XSL-FO a partir del XML creado en el paso anterior.
5. Una vez obtenido el documento XSL-FO podemos obtener el documento PDF mediante el procesador XSL-FO “Apache-FOP” (<http://xmlgraphics.apache.org/fop/>)

Si el formato del documento original está soportado por el sistema (ver anexo A.2), podemos incluirlo en el informe final mediante alguno de los siguientes métodos:

- **Incrustando el documento original en el informe.** Dentro del fichero XSLT que representa la plantilla PDF se puede especificar que paginas del documento firmado serán incluidas en el informe tras la transformación XSL-FO.
- **Incluyendo imágenes de las páginas del documento original.** El XML de datos utilizado para la transformación XSLT incluirá las imágenes de las paginas del documento original de forma que el diseñador de la plantilla pueda configurar el informe final incluyendo estas imágenes en el fichero XSL-FO.
- **Concatenando al informe el documento original.** Una vez realizado la transformación XSL-FO se puede concatenar el PDF resultante con el documento original.

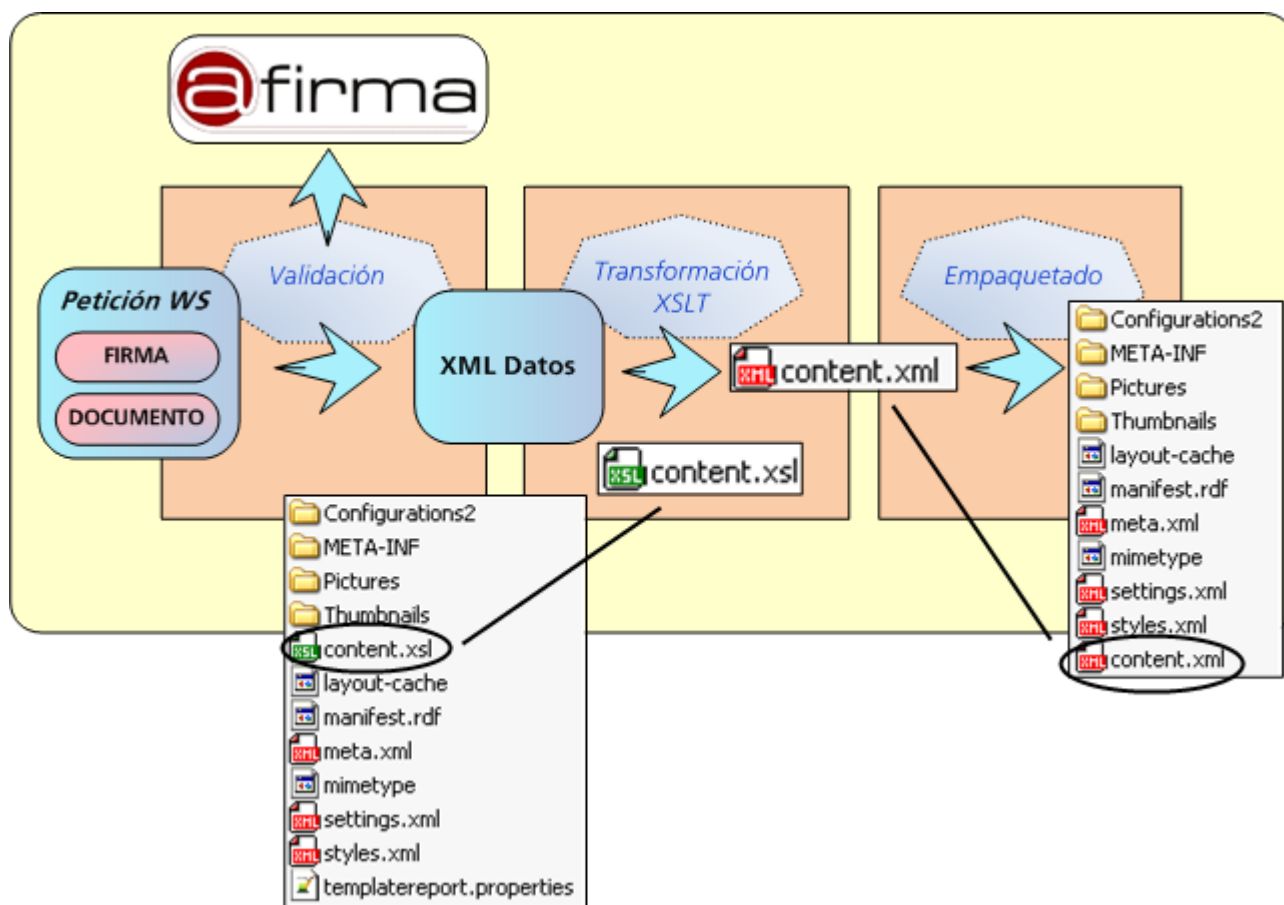
Además, los informes PDF permiten anexar datos del proceso de firma electrónica, como puede ser el documento original, certificados firmantes, firma electrónica, etc.

5.2 Proceso de Generación de Informes ODF

Para la generación de informes ODF se ha optado por la transformación XSLT en el que el resultado del procesamiento de los datos de validación sea alguno de los ficheros que forman el documento ODF final.

Así una plantilla ODF será un documento ODF (fichero ZIP) en el que sea incluido una serie de ficheros XSLT que tendrán como resultado alguno de los XML que componen el ODF.

A modo de ejemplo, supongamos que queremos establecer la información de validación en el fichero “content.xml” propio del ODF, en la siguiente figura se representa el proceso de elaboración del informe ODF.



En este caso el proceso de generación del informe ODF consta de los siguientes pasos.

1. Una aplicación externa solicita mediante petición WS la generación de un informe de firma, la petición incluye la firma electrónica, los datos firmados e información adicional para la generación del informe de firma.
2. El sistema solicita la validación de la firma electrónica a la plataforma de validación y firma @firma.
3. Una vez validada la firma electrónica se genera el XML de datos con la información de validación y el resto de parámetros adicionales.
4. Previamente la aplicación habrá registrado en el sistema una plantilla ODF. Esta plantilla es un fichero ODF en el que se han incluido uno o más ficheros XSLT para obtener los ficheros XML propios del ODF.
5. El sistema extrae los ficheros XSLT de la plantilla y aplica cada uno al XML generado debiéndose obtener ficheros XML propios del ODF final. En el ejemplo mostrado el sistema aplica la transformación indicada en el fichero "content.xml" al XML generado en el paso 3 obteniendo un fichero "content.xml" en el que sea podido incluir información del firmante, documento firmado, etc.

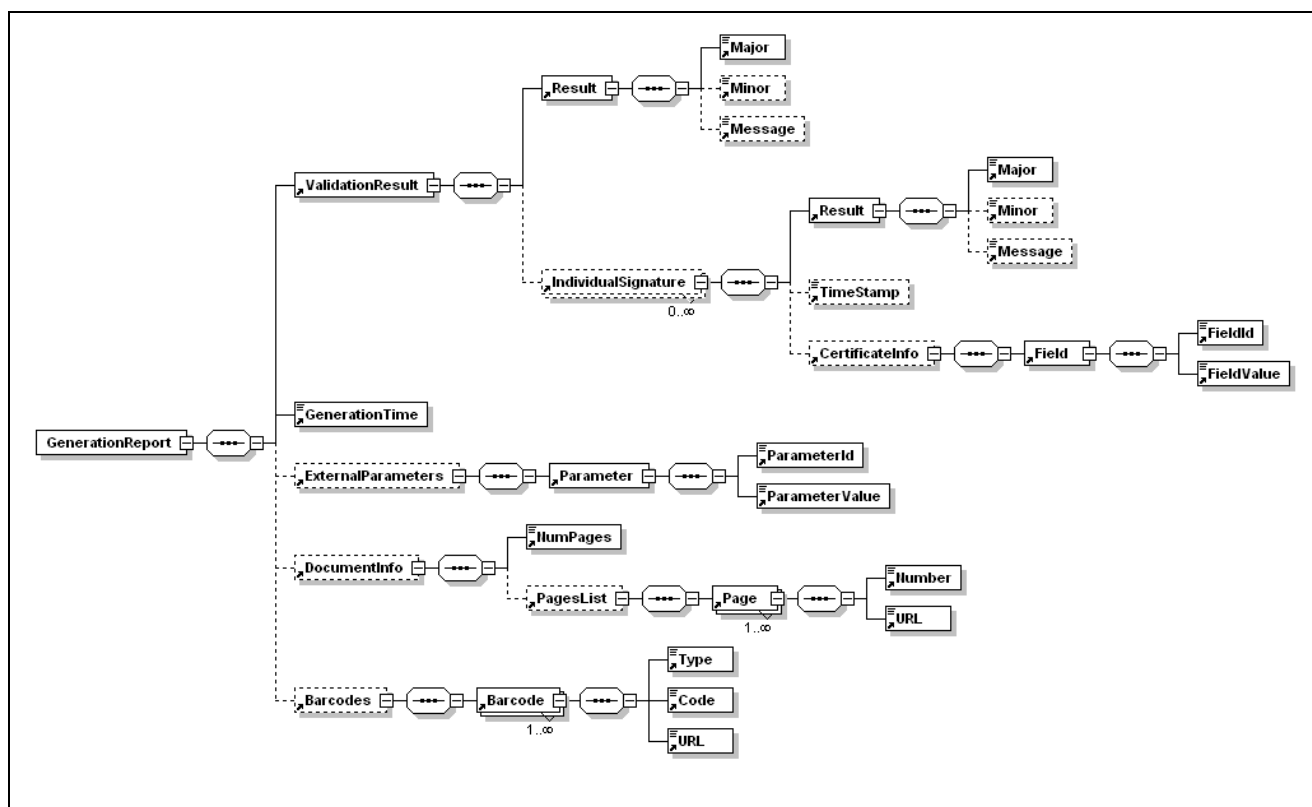
6. Se incluye en el ODF los ficheros obtenido mediante tranformación y se extraen los asociados a la configuración de la plantilla (ficheros XSLT y properties). El resultado será un ODF con la información de validación.

Si el formato del documento original está soportado por el sistema (ver anexo A.2), el sistema permite incluir imágenes de las páginas del documento origina. El sistema incluirá las rutas relativas de las imágenes generadas en el XML de entrada para que pueda incluirse en el reporte final.

6 XML de entrada

Como se ha descrito en el apartado anterior todos los informes se generan a partir de aplicar transformación XSLT a un XML de definido en el que se recoge la información de validación de la firma electrónica y los parámetros adicionales incluidos en la petición.

En el siguiente diagrama se representa los componentes que puede tener este XML.



En el anexo A.1 se puede consultar la definición formal de cada uno de estos componentes.

Así un XML que contenga los elementos representados tendría la siguiente estructura.

```
<GenerationReport xmlns="urn:es:gob:signaturereport:generation:inputparameters">
  <ValidationResult>
    <Result>
      <Major>${/dss:VerifyResponse/dss:Result/dss:ResultMajor}</Major>
      <Minor>${/dss:VerifyResponse/dss:Result/dss:ResultMinor}</Minor>
      <Message>${/dss:VerifyResponse/dss:Result/dss:ResultMessage}</Message>
    </Result>
    <IndividualSignature>
      <TimeStamp>
      <CertificateInfo>
        <Field>
          <FieldId>
          <FieldValue>
        </Field>
      </CertificateInfo>
    </IndividualSignature>
  </ValidationResult>
  <GenerationTime>
  <ExternalParameters>
    <Parameter>
      <ParameterId>
      <ParameterValue>
    </Parameter>
  </ExternalParameters>
  <DocumentInfo>
    <NumPages>
    <PagesList>
      <Page>
        <Number>
        <URL>
      </Page>
    </PagesList>
  </DocumentInfo>
  <Barcodes>
    <Barcode>
      <Type>
      <Code>
      <URL>
    </Barcode>
  </Barcodes>
</GenerationReport>
```

```
</Result>
<IndividualSignature>
  <Result>
    <Major>${//vr:IndividualSignatureReport/dss:Result/dss:ResultMajor}</Major>
    <Minor>${//vr:IndividualSignatureReport/dss:Result/dss:ResultMinor}</Minor>
    <Message>${//vr:IndividualSignatureReport/dss:Result/dss:ResultMessage}</Message>
  </Result>
  <TimeStamp>${//vr:SignatureTimeStamp/vr:TimeStampContent/vr:CreationTime}</TimeStamp>
  <CertificateInfo>
    <Field>
      <FieldId>${//afxp:ReadableCertificateInfo/afxp:ReadableField/afxp:FieldIdentity}</FieldId>
      <FieldValue>${//afxp:ReadableCertificateInfo/afxp:ReadableField/afxp:FieldValue}</FieldValue>
    </Field>
  </CertificateInfo>
</IndividualSignature>
</ValidationResult>
<GenerationTime>${FECHA_GENERACION}</GenerationTime>
<ExternalParameters>
  <Parameter>
    <ParameterId>${/GenerationRequest/ExternalParameters/Parameter/ParameterId}</ParameterId>
    <ParameterValue>${/GenerationRequest/ExternalParameters/Parameter/ParameterId}</ParameterValue>
  </Parameter>
</ExternalParameters>
<DocumentInfo>
  <NumPages>${NÚMERO_DE_PAGINAS}</NumPages>
  <PagesList>
    <Page>
      <Number>${NÚMERO_PAGINA}</Number>
      <URL>${LOCALIZACIÓN_PAGINA}</URL>
    </Page>
  </PagesList>
</DocumentInfo>
<Barcodes>
  <Barcode>
    <Type>${/GenerationRequest/Barcodes/Barcode/Type}</Type>
    <Code>${/GenerationRequest/Barcodes/Barcode/Message}</Code>
    <URL>${LOCALIZACIÓN_CÓDIGO}</URL>
  </Barcode>
</Barcodes>
</GenerationReport>
```

Como observamos en la figura anterior el XML de entrada (GenerationReport) utilizado para la generación de informe puede contener los siguientes cinco elementos XML:

- **ValidationResult.** Este componente incluye la información resultante del proceso de verificación de firma mediante el servicio DSSAfirmaVerify de @firma. Puede consultarse una información detallada del servicio en [AFIRMA-DSS-MAN].
- **GenerationTime.** Este componente incluye la fecha de generación del informe en formato “dd/MM/yyyy HH:mm:ss”.

- ExternalParameters. En este elemento se incluyen los parámetros adicionales incluidos en la petición de generación de informe. Se puede consultar los esquemas de la mensajería WS del eVisor en [EVISOR-PROG-MAN].
- DocumentInfo. Este componente XML incluye información sobre el documento originalmente firmado.
- Barcodes. Este elemento incluye información sobre códigos de barra generados por el sistema y que fueron solicitados en la petición.

A continuación vamos a desarrollar cada uno de estos elementos.

6.1 ValidationResult

```
<ValidationResult>
  <Result>
    <Major>${/dss:VerifyResponse/dss:Result/dss:ResultMajor}</Major>
    <Minor>${/dss:VerifyResponse/dss:Result/dss:ResultMinor}</Minor>
    <Message>${/dss:VerifyResponse/dss:Result/dss:ResultMessage}</Message>
  </Result>
  <IndividualSignature>
    <Result>
      <Major>${//vr:IndividualSignatureReport/dss:Result/dss:ResultMajor}</Major>
      <Minor>${//vr:IndividualSignatureReport/dss:Result/dss:ResultMinor}</Minor>
      <Message>${//vr:IndividualSignatureReport/dss:Result/dss:ResultMessage}</Message>
    </Result>
    <TimeStamp>${//vr:SignatureTimeStamp/vr:TimeStampContent/vr:CreationTime}</TimeStamp>
    <CertificateInfo>
      <Field>
        <FieldId>${//afxp:ReadableCertificateInfo/afxp:ReadableField/afxp:FieldIdentity}</FieldId>
        <FieldValue>${//afxp:ReadableCertificateInfo/afxp:ReadableField/afxp:FieldValue}</FieldValue>
      </Field>
    </CertificateInfo>
  </IndividualSignature>
</ValidationResult>
```

Este componente está destinado a recoger la información resultante del proceso de validación de la firma electrónica mediante el servicio DSSAfirmaVerify de @firma. Se compone de los siguientes elementos:

- Result. En él se recoge el resultado global de la validación de firma.
 - Major. Incluye el valor del campo “dss:ResultMajor” global que se devuelve en la respuesta de @firma.
 - Minor. Incluye el valor del campo “dss:ResultMinor” global que se devuelve en la respuesta de @firma.
 - Message. Incluye el valor del campo “dss:ResultMessage” global que se devuelve en la respuesta de @firma.

- IndividualSignature. Se incluirá por cada elemento “vr:IndividualSignatureReport” que contenga la respuesta de @firma un componente de este tipo. En este elemento se incluirá:
 - Result. Resultado parcial de la firma del firmante.
 - Major. Incluye el valor del campo “dss:ResultMajor” parcial que se devuelve en la respuesta de @firma.
 - Minor. Incluye el valor del campo “dss:ResultMinor” parcial que se devuelve en la respuesta de @firma.
 - Message. Incluye el valor del campo “dss:ResultMessage” parcial que se devuelve en la respuesta de @firma.
 - Timestamp. Incluye la fecha de sellado de tiempo devuelta en la respuesta dentro del elemento “vr:TimeStampContent/vr:CreationTime”
 - CertificateInfo. En este componente se incluye la información del certificado firmante devuelta por @firma en el elemento “afxp:ReadableCertificateInfo”.
 - Fied. Se incluye un elemento de este tipo por cada elemento “afxp:ReadableField” incluida en el componente “afxp:ReadableCertificateInfo” de la respuesta.
 - FieldId. Incluye el valor del campo “afxp:FieldIdentity”.
 - FieldValue. Incluye el valor del campo “afxp:FieldValue”.

6.2 GenerationTime

```
<GenerationTime>${FECHA_GENERACION}</GenerationTime>
```

Este elemento incluye la fecha de generación del XML de entrada en formato “dd/MM/yyyy HH:mm:ss”.

6.3 ExternalParameters

```
<ExternalParameters>
  <Parameter>
    <ParameterId>${/GenerationRequest/ExternalParameters/Parameter/ParameterId}</ParameterId>
    <ParameterValue>${/GenerationRequest/ExternalParameters/Parameter/ParameterId}</ParameterValue>
  </Parameter>
</ExternalParameters>
```

El servicio de generación de informe del sistema eVisor permite suministrar a la plantillas información adicional, para ello se define en el mensaje de petición WS el componente “ExternalParameters” cuyo contenido se incluye en este elemento. Puede consultarse una información detallada del servicio en [EVISOR-PROG-MAN].

6.4 DocumentInfo

```
<DocumentInfo>
  <NumPages>${NÚMERO_DE_PAGINAS}</NumPages>
  <PagesList>
    <Page>
      <Number>${NÚMERO_PAGINA}</Number>
      <URL>${LOCALIZACIÓN_PAGINA}</URL>
    </Page>
  </PagesList>
</DocumentInfo>
```

En el caso de que el documento firmado sea en formato PDF (para plantillas PDF) u ODF (para plantillas ODF) este elemento incluirá información sobre el documento. Los elementos que lo componen son:

- NumPages. Número de páginas del documento.
- PagesList. En el caso que la plantilla esté configurada para incluir el documento como imagen, este elemento incluirá la lista de páginas exportadas a imagen. El proceso de paso de documento a imagen es un proceso costoso en memoria y tiempo de ejecución por lo que **no se recomienda incluir el documento firmado como imagen**, el sistema incluye métodos alternativos más eficientes para incluir la información firmada en el informe generado. Puede consultarse una información detallada sobre el establecimiento en la configuración de las plantillas en el manual [EVISOR-ADMIN-MAN].
 - Page. Por cada página exportada se incluirá un componente de este tipo.
 - Number. Contiene el número de la página.
 - URL. En este elemento contiene la localización de la imagen generada.

6.5 Barcodes

```
<Barcodes>
  <Barcode>
    <Type>${/GenerationRequest/Barcodes/Barcode/Type}</Type>
    <Code>${/GenerationRequest/Barcodes/Barcode/Message}</Code>
    <URL>${LOCALIZACIÓN_CÓDIGO}</URL>
  </Barcode>
</Barcodes>
```

El sistema permite la inclusión de códigos de barras en los informes generados, una de las vías para añadir un código de barra al informe es mediante la solicitud del mismo en la petición WS (puede consultarse una información detallada del servicio en [EVISOR-PROG-MAN]). En este caso el sistema genera los códigos solicitados e incluye en este componente XML la información necesaria para inclusión en el informe.

Este componente está formado por los siguientes elementos:

- Barcode. Componente que representa a un código de barras generado. Se incluirá tantos elementos de este tipo como códigos de barras se haya solicitado en la petición. Este elemento lo forman los siguientes elementos:
 - Type. Especifica el tipo de código de barras generado, corresponde con el indicado en la petición y debe ser alguno de los admitidos por el sistema (Code39, Code128, EAN128, Codabar, PDF417, DataMatrix o QRCode).
 - Code. Mensaje utilizado para la generación del código.
 - URL. Localización de la imagen del código de barras.

7 Diseño de plantillas PDF

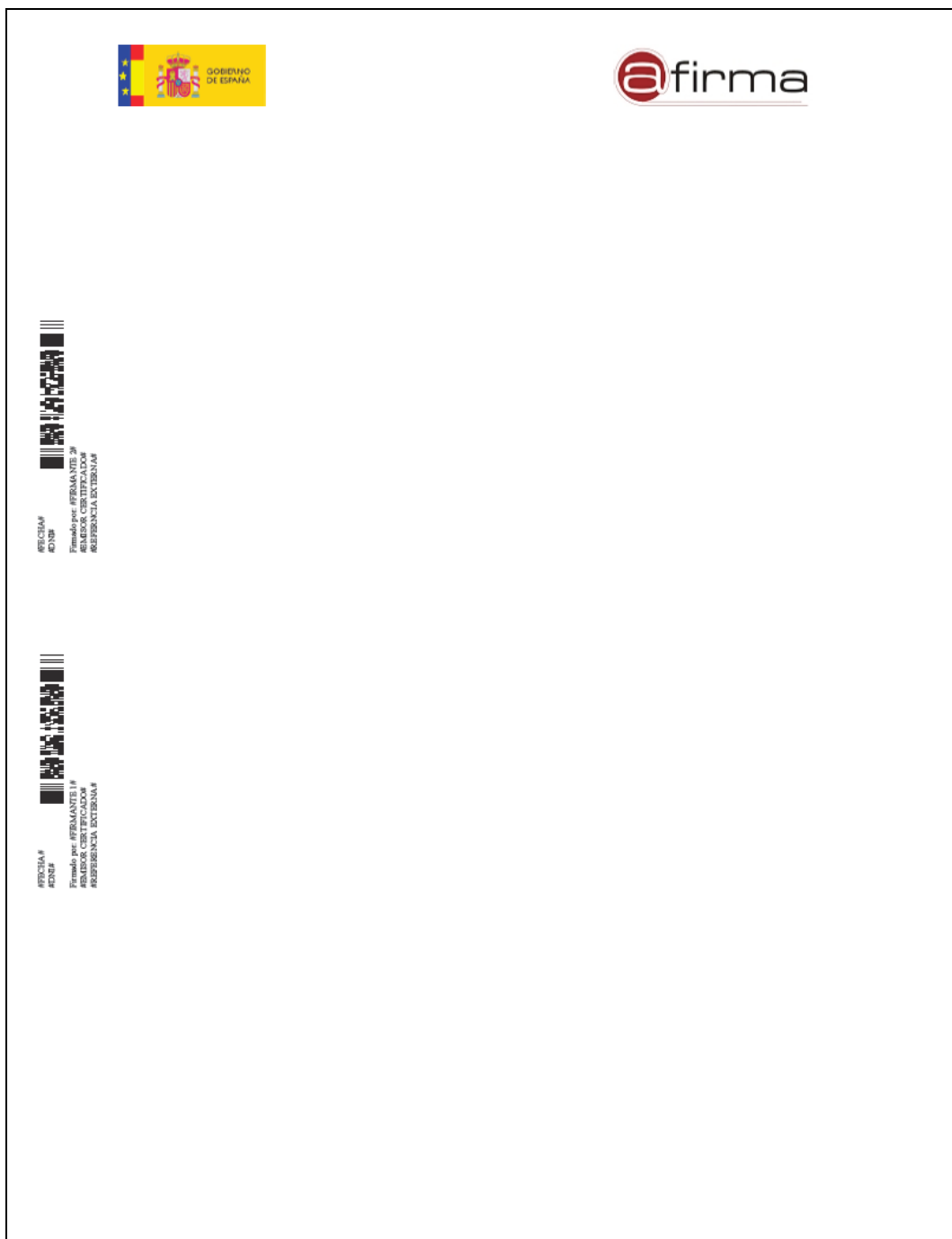
Como vimos en el apartado 5.1 la generación de informes PDF se realiza mediante la aplicación de transformada XSL al XML que contiene la información de validación (descrito en el apartado 6).

El resultado de esta transformación debe ser un fichero XSL-FO válido que será procesado para obtener el informe final.

A continuación vamos a describir el proceso de diseño de una plantilla PDF.

7.1 Diseño del fichero XSL-FO


El primer paso sería definir el fichero XSL-FO que daría como resultado el informe esperado, por ejemplo supongamos que deseamos un informe como el siguiente.



En este caso diseñamos el informe dividiéndolo en tres zonas:

- Cabecera con los logos de la aplicación y el organismo en cuestión.
- Izquierda donde incluiremos la información de los firmates. Esta será la parte que completaremos mediante transformación XSLT con la información de validación, concretamente queremos incluir por firmante un objeto como el siguiente:

Donde:

#FECHA#	
#DNI#	
Firmado por: #FIRMANTE#	
#EMISOR CERTIFICADO#	
#REFERENCIA EXTERNA#	

- #FECHA# Incluiremos el sello de tiempo de la firma.
 - #DNI# DNI del firmante.
 - Un código PDF417 generado mediante el DNI del firmante.
 - #FIRMANTE# Incluiremos el nombre y apellidos del firmante.
 - #EMISOR# Emisor del certificado.
 - #REFERENCIA EXTERNA#. En este campo incluiremos un parámetro adicional añadido en la petición.
- Reservamos la zona central para incluir el documento firmado en caso de ser PDF.

Un documento de este tipo tendría el siguiente fichero XSL-FO:

```
<?xml version="1.0" encoding="utf-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="simple" page-height="29.7cm" page-width="21cm" margin-top="0.5cm"
margin-bottom="0.5cm" margin-left="0.5cm" margin-right="0.5cm">
      <fo:region-body margin-top="2cm" margin-bottom="2cm" margin-right="2cm" margin-left="2cm"/>
      <fo:region-before extent="2cm"/>
      <fo:region-after extent="1cm"/>
      <fo:region-start extent="2cm"/>
      <fo:region-end extent="1cm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="simple">
    <fo:static-content flow-name="xsl-region-start">
      <!-- Región izquierda-->
      <fo:block-container reference-orientation="90">
        <fo:block font-family="Arial" font-size="6pt">
          <fo:table>
            <fo:table-body>
              <fo:table-row>
                <fo:table-cell width="8cm">
                  <fo:block></fo:block>
                </fo:table-cell>
                <!-- Información del firmante -->
                <fo:table-cell width="8cm">
                  <fo:block>
                </fo:table>
              </fo:table-body>
            </fo:table>
          </fo:block>
        </fo:block-container>
      </fo:static-content>
    </fo:page-sequence>
  </fo:root>
```

```

<fo:table-body>
  <fo:table-row>
    <fo:table-cell>
      <fo:block>
        <fo:table>
          <fo:table-body>
            <fo:table-row>
              <fo:table-cell text-align="left" width="2cm">
                <fo:block>#FECHA#</fo:block>
                <fo:block>#DNI#</fo:block>
              </fo:table-cell>
              <fo:table-cell text-align="left">
                <fo:block text-align="left">
                  <fo:instream-foreign-object>
                    <barcode:barcode xmlns:barcode="http://barcode4j.krysalis.org/ns"
message="08033029A">
                      <barcode:pdf417/>
                    </barcode:barcode>
                  </fo:instream-foreign-object>
                </fo:block>
              </fo:table-cell>
            </fo:table-row>
          </fo:table-body>
        </fo:table>
      </fo:block>
    </fo:table-cell>
  </fo:table-row>
  <fo:table-row>
    <fo:table-cell>
      <fo:block text-align="left">Firmado por: #FIRMANTE 1#</fo:block>
    </fo:table-cell>
  </fo:table-row>
  <fo:table-row>
    <fo:table-cell>
      <fo:block text-align="left">#EMISOR CERTIFICADO#</fo:block>
    </fo:table-cell>
  </fo:table-row>
  <fo:table-row>
    <fo:table-cell>
      <fo:block text-align="left">#REFERENCIA EXTERNA#</fo:block>
    </fo:table-cell>
  </fo:table-row>
</fo:table-body>
</fo:table>
</fo:block>
</fo:table-cell>
</fo:table-row>
</fo:table-body>
</fo:table>
</fo:block>
</fo:block-container>
</fo:static-content>
<fo:static-content flow-name="xsl-region-after">
  <!-- Pie de página-->
</fo:block>
</fo:static-content>
<fo:static-content flow-name="xsl-region-before">

```

```

<fo:block>
  <fo:table>
    <fo:table-body>
      <fo:table-row>
        <fo:table-cell>
          <!-- Bloque destinado al logo del Gobierno-->
          <fo:block>
            <fo:external-graphic src="C:\Documentos\imagenes\logo_gobierno.png" content-width="5cm" content-
height="1.5cm"/>
          </fo:block>
        </fo:table-cell>
        <fo:table-cell>
          <!-- Bloque destinado al logo de la aplicación -->
          <fo:block text-align="right">
            <fo:external-graphic src="C:\Documentos\imagenes\afirma.png" content-width="5cm" content-
height="1.5cm"/>
          </fo:block>
        </fo:table-cell>
      </fo:table-row>
    </fo:table-body>
  </fo:table>
</fo:block>
</fo:static-content>
<fo:static-content flow-name="xsl-region-end">
  <fo:block>
    <!-- Region de la derecha -->
    <fo:block>
      </fo:static-content>
      <fo:flow flow-name="xsl-region-body">
        <fo:block>
          </fo:block>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>

```

7.2 Diseño de la plantilla

Una vez tenemos claro el informe que deseamos, vamos a implementar el fichero XSLT que generará el fichero XSL-FO anterior.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:ri="urn:es:gob:signaturereport:generation:inputparameters"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output method="xml" version="1.0" omit-xml-declaration="no" indent="yes" encoding="UTF-8"/>
  <xsl:template match="ri:GenerationReport">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master master-name="simple" page-height="29.7cm" page-
width="21cm" margin-top="0.5cm" margin-bottom="0.5cm" margin-left="0.5cm" margin-right="0.5cm">
          <fo:region-body margin-top="2cm" margin-bottom="2cm" margin-
right="2cm" margin-left="2cm"/>
          <fo:region-before extent="2cm"/>
          <fo:region-after extent="1cm"/>
          <fo:region-start extent="2cm"/>
          <fo:region-end extent="1cm"/>
        </fo:simple-page-master>

```

```

</fo:layout-master-set>
<xsl:variable name="externalReference"
select="ri:ExternalParameters/ri:Parameter[ri:ParameterId='externalReference']/ri:ParameterValue"/>
<xsl:variable name="includeDocInfo" select="count(ri:DocumentInfo)"/>
<xsl:variable name="numTotalPages" select="ri:DocumentInfo/ri:NumPages"/>
<xsl:variable name="iniPage" select="0"/>
<xsl:variable name="signersNumber" select="count(//ri:CertificateInfo)"/>

```

```

<xsl:variable name="srcImagenGobierno">C:\Documentos\imagenes\logo_gobierno.png</xsl:variable>
<xsl:variable
name="srcImagenAplicacion">C:\Documentos\imagenes\afirma.png</xsl:variable>

```

```

<xsl:if test="$includeDocInfo!=0">
  <xsl:call-template name="pageSequenceWithDocument">
    <xsl:with-param name="externalReference">
      <xsl:value-of select="$externalReference"/>
    </xsl:with-param>
    <xsl:with-param name="srcImagenGobierno">
      <xsl:value-of select="$srcImagenGobierno"/>
    </xsl:with-param>
    <xsl:with-param name="srcImagenAplicacion">
      <xsl:value-of select="$srcImagenAplicacion"/>
    </xsl:with-param>
    <xsl:with-param name="currentPage">
      <xsl:value-of select="$iniPage"/>
    </xsl:with-param>
    <xsl:with-param name="numPages">
      <xsl:value-of select="$numTotalPages"/>
    </xsl:with-param>
    <xsl:with-param name="signersNumber">
      <xsl:value-of select="$signersNumber"/>
    </xsl:with-param>
  </xsl:call-template>
</xsl:if>
</fo:root>
</xsl:template>
<!-- Se tiene información del documento firmado-->
<xsl:template name="pageSequenceWithDocument">
  <xsl:param name="externalReference"/>
  <xsl:param name="srcImagenGobierno"/>
  <xsl:param name="srcImagenAplicacion"/>
  <xsl:param name="currentPage"/>
  <xsl:param name="numPages"/>
  <xsl:param name="signersNumber"/>
  <xsl:if test="$currentPage < $numPages">

```

```

<ri:IncludePage Ypos="30" Xpos="30" Width="150" Height="230">
  <ri:DocumentPage>
    <xsl:value-of select="$currentPage + 1"/>
  </ri:DocumentPage>
  <ri:ReportPage>
    <xsl:value-of select="$currentPage + 1"/>
  </ri:ReportPage>
</ri:IncludePage>

```

```

<fo:page-sequence master-reference="simple">
  <fo:static-content flow-name="xsl-region-start">

```

```

<!-- Región izquierda-->
<fo:block-container reference-orientation="90">
  <fo:block font-size="6pt">
    <fo:table>
      <fo:table-body>
        <fo:table-row>
          <!-- Información del firmante -->
          <xsl:for-each select="//ri:IndividualSignature">
            <xsl:variable name="fecha" select="substring(ri:TimeStamp,0,11)"/>
            <xsl:variable name="nombre"
select="ri:CertificateInfo/ri:Field[ri:FieldId='nombreResponsable']/ri:FieldValue"/>
            <xsl:variable name="primerApellido"
select="ri:CertificateInfo/ri:Field[ri:FieldId='primerApellidoResponsable']/ri:FieldValue"/>
            <xsl:variable name="segundoApellido"
select="ri:CertificateInfo/ri:Field[ri:FieldId='segundoApellidoResponsable']/ri:FieldValue"/>
            <xsl:variable name="dni" select="ri:CertificateInfo/ri:Field[ri:FieldId='NIFResponsable']/ri:FieldValue"/>
            <xsl:variable name="emisor" select="ri:CertificateInfo/ri:Field[ri:FieldId='emisor']/ri:FieldValue"/>
            <xsl:variable name="espacio" select="'' ''"/>
            <fo:table-cell width="8cm">
              <fo:block>
                <fo:table>
                  <fo:table-body>
                    <fo:table-row>
                      <fo:table-cell>
                        <fo:block>
                          <fo:table>
                            <fo:table-body>
                              <fo:table-row>
                                <fo:table-cell text-align="left" width="2cm">
                                  <fo:block>
                                    <xsl:value-of select="$fecha"/>
                                  </fo:block>
                                  <fo:block>
                                    <xsl:value-of select="$dni"/>
                                  </fo:block>
                                </fo:table-cell>
                                <fo:table-cell text-align="left">
                                  <fo:block text-align="left">
                                    <fo:instream-foreign-object>
                                      <barcode:barcode xmlns:barcode="http://barcode4j.krysalis.org/ns">
                                        <xsl:attribute name="message"><xsl:value-of select="$dni"/></xsl:attribute>
                                        <barcode:pdf417></barcode:pdf417>
                                      </barcode:barcode>
                                    </fo:instream-foreign-object>
                                  </fo:block>
                                </fo:table-cell>
                              </fo:table-row>
                            </fo:table-body>
                          </fo:table>
                        </fo:block>
                      </fo:table-cell>
                    </fo:table-row>
                  </fo:table-body>
                </fo:table>
              </fo:block>
            </fo:table-cell>
          </fo:table-row>
        </fo:table-body>
      </fo:table>
    </fo:block>
    <fo:table-row>
      <fo:table-cell>
        <fo:block text-align="left">Firmado por: <xsl:value-of select="$primerApellido"/>
          <xsl:value-of select="$espacio"/>
          <xsl:value-of select="$segundoApellido"/>, <xsl:value-of select="$espacio"/>
        </fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:block>

```

```

        <xsl:value-of select="$nombre"/>
      </fo:block>
    </fo:table-cell>
  </fo:table-row>
  <fo:table-row>
    <fo:table-cell>
      <fo:block text-align="left">
        <xsl:value-of select="$emisor"/>
      </fo:block>
    </fo:table-cell>
  </fo:table-row>
  <fo:table-row>
    <fo:table-cell>
      <fo:block text-align="left">
        <xsl:value-of select="$externalReference"/>
      </fo:block>
    </fo:table-cell>
  </fo:table-row>
</fo:table-body>
</fo:table>
</fo:block>
</fo:table-cell>
</xsl:for-each>
</fo:table-row>
</fo:table-body>
</fo:table>
</fo:block>
</fo:block-container>
</fo:static-content>

<fo:static-content flow-name="xsl-region-after">
  <!-- Pie de página-->
  <fo:block/>
</fo:static-content>
<fo:static-content flow-name="xsl-region-before">
  <fo:block>
    <fo:table>
      <fo:table-body>
        <fo:table-row>
          <fo:table-cell>
            <!-- Bloque destinado al logo del Gobierno-->
            <fo:block>
              <fo:external-graphic content-width="5cm" content-height="1.5cm">
                <xsl:attribute name="src"><xsl:value-of select="$srcImagenGobierno"/></xsl:attribute>
              </fo:external-graphic>
            </fo:block>
          </fo:table-cell>
          <fo:table-cell>
            <!-- Bloque destinado al logo de la aplicación -->
            <fo:block text-align="right">
              <fo:external-graphic content-width="5cm" content-height="1.5cm">
                <xsl:attribute name="src"><xsl:value-of select="$srcImagenAplicacion"/></xsl:attribute>
              </fo:external-graphic>
            </fo:block>
          </fo:table-cell>
        </fo:table-row>
      </fo:table-body>
    </fo:table>
  </fo:block>

```

```
</fo:table>
</fo:block>
</fo:static-content>
<fo:static-content flow-name="xsl-region-end">
  <fo:block>
    <!-- Region de la derecha -->
  </fo:block>
</fo:static-content>
<fo:flow flow-name="xsl-region-body">
  <fo:block/>
</fo:flow>
</fo:page-sequence>

  <xsl:call-template name="pageSequenceWithDocument">
    <xsl:with-param name="externalReference">
      <xsl:value-of select="$externalReference"/>
    </xsl:with-param>
    <xsl:with-param name="srcImagenGobierno">
      <xsl:value-of select="$srcImagenGobierno"/>
    </xsl:with-param>
    <xsl:with-param name="srcImagenAplicacion">
      <xsl:value-of select="$srcImagenAplicacion"/>
    </xsl:with-param>
    <xsl:with-param name="currentPage">
      <xsl:value-of select="$currentPage + 1"/>
    </xsl:with-param>
    <xsl:with-param name="numPages">
      <xsl:value-of select="$numPages"/>
    </xsl:with-param>
    <xsl:with-param name="signersNumber">
      <xsl:value-of select="$signersNumber"/>
    </xsl:with-param>
  </xsl:call-template>
</xsl:if>
</xsl:template>
</xsl:stylesheet>
```

En este caso pretendemos generar una página como la anterior para cada página del documento firmado.

Este ejemplo nos va a servir para describir como se pueden incluir los elementos más comunes en un informe PDF.

7.2.1 Imágenes

```
....
<xsl:variable name="srcImagenGobierno">C:\Documentos\imagenes\logo_gobierno.png</xsl:variable>
...
<fo:external-graphic content-width="5cm" content-height="1.5cm">
  <xsl:attribute name="src"><xsl:value-of select="$srcImagenGobierno"/></xsl:attribute>
</fo:external-graphic>
....
```

Para incluir las imágenes en el informe podemos utilizar el componente “fo:external-graphic” indicando la localización de la imagen en el servidor eVisor.

En el caso que no queramos tener la imagen en una carpeta del servidor podemos incluirla en la plantilla indicando la localización de la siguiente forma:

```
....
<xsl:variable name="srcImagenGobierno">
url(&quot;data:image/png;base64,#IMAGEN_CODIFICADA_EN_BASE64#&quot;)&
</xsl:variable>
....
<fo:external-graphic content-width="5cm" content-height="1.5cm">
  <xsl:attribute name="src"><xsl:value-of select="$srcImagenGobierno"/></xsl:attribute>
</fo:external-graphic>
....
```

7.2.2 Documento firmado

Para incluir el documento en el informe de firma PDF eVisor facilita tres métodos de inclusión:

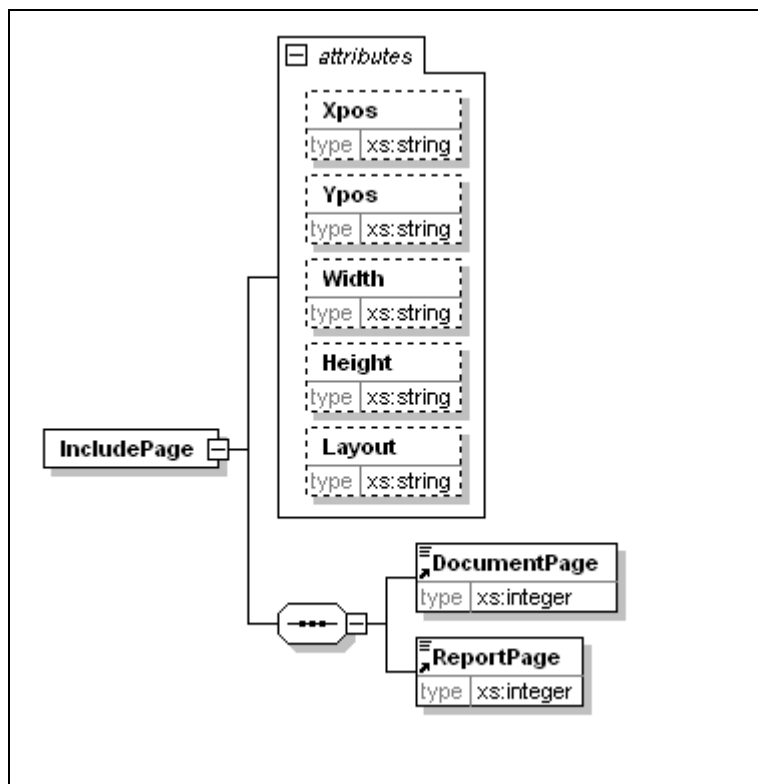
- Embebido o incrustado. Dentro del fichero XSLT que representa la plantilla PDF se puede especificar que paginas del documento firmado serán incluidas en el informe tras la transformación XSL-FO.
- Mediante imagen. El XML de datos utilizado para la transformación XSLT incluirá la ruta a las imágenes de las páginas del documento original de forma que el diseñador de la plantilla pueda configurar el informe final incluyendo estas imágenes en el fichero XSL-FO.
- Concatenación. Una vez realizado la transformación XSL-FO se puede concatenar el PDF resultante con el documento original.

A continuación se describe los diferentes modos enumerados.

7.2.2.1 Incrustando el documento original en el informe

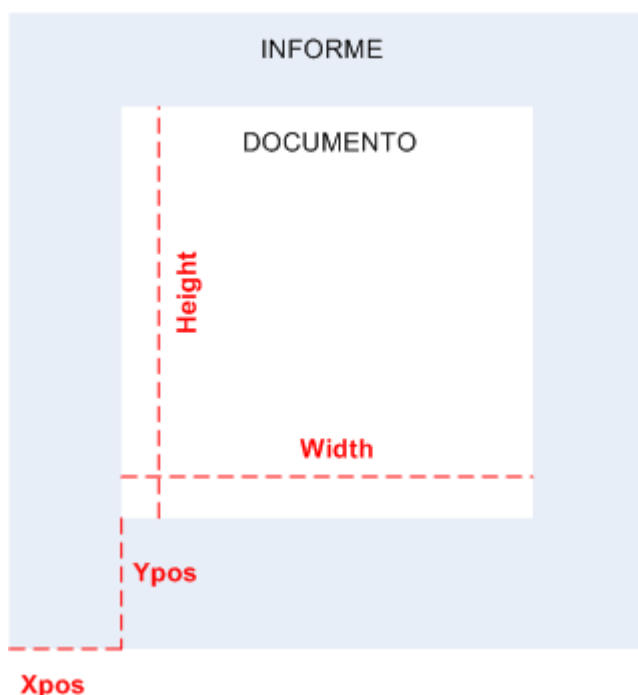
Para permitir la incrustación del documento en el informe es necesario tener establecido en la configuración de la plantilla el modo de inclusión de documento a “Embebido”, esto lo podemos hacer desde la Herramienta de Administración del eVisor cuyo uso se describe en el manual [EVISOR-ADMIN-MAN].

La aplicación permite indicar en la plantilla que se incluyan determinadas páginas del documento firmado para ello se ha implementado el componente “IncludePage”.



Donde:

- **Xpos**. Es la distancia (en el eje horizontal) del vértice inferior izquierdo del informe a donde se situará el vértice inferior izquierdo del documento firmado. La distancia se debe indicar en milímetros.
- **Ypos**. Es la distancia (en el eje vertical) del vértice inferior izquierdo del informe a donde se situará el vértice inferior izquierdo del documento firmado. La distancia se debe indicar en milímetros.
- **Width**. Ancho (en milímetros) que tendrá la página insertada.
- **Height**. Alto (en milímetros) que tendrá la página insertada.



- Layout. Atributo que si la página a insertar se coloca sobre o tras la página del informe. Los valores que puede tomar son:
 - front (opción por defecto). La página insertada se colocará encima de la página generada tras la transformación XSL-FO
 - back. El documento se incluirá debajo de la página del informe.
- DocumentPage. Elemento que indica el número de la página del documento firmado que deseamos incluir en el informe de firma.
- ReportPage. Componente que indica el número de página del informe donde se incluirá la página del documento.

La descripción formal de este componente puede consultarse en el esquema incluido en el anexo A.1

En el ejemplo mostrado anteriormente se utiliza este método para incluir en el informe final el documento firmado.

```
<ri:IncludePage Ypos="30" Xpos="30" Width="150" Height="230">
  <ri:DocumentPage>
    <xsl:value-of select="$currentPage + 1"/>
  </ri:DocumentPage>
  <ri:ReportPage>
    <xsl:value-of select="$currentPage + 1"/>
  </ri:ReportPage>
</ri:IncludePage>
```

7.2.2.2 Inclusión del documento firmado mediante imagen

Podemos incluir el documento firmado mediante la inclusión de las imágenes de las páginas firmadas.

Para ello al registrar la plantilla se debe especificar que el modo de inclusión de documento a “Imagen”, durante el proceso de alta de la plantilla podemos especificar si deseamos exportar todas las páginas del documento o solamente un rango determinado. Para más información sobre la configuración de las plantillas puede consultarse el manual de Administración [EVISOR-ADMIN-MAN].

En el proceso de generación de informe, el sistema exportará a imagen las páginas indicadas en la configuración incluyendo las rutas a las imágenes en el elemento “DocumentInfo” (apartado 6.4) del XML de entrada previo a la transformación.

Para incluir estas imágenes en el informe final podremos utilizar el componente “fo:external-graphic” como se muestra en el siguiente ejemplo:

```
....  
<xsl:variable name="primeraPagina"  
select="/GenerationReport/DocumentInfo/PagesList/Page[Number='1']/URL"/>  
....  
<fo:external-graphic>  
  <xsl:attribute name="src"><xsl:value-of select="$primeraPagina"/></xsl:attribute>  
</fo:external-graphic>  
....
```

La exportación de las páginas de un documento PDF a imagen es un proceso costoso por lo que se recomienda utilizar la incrustación del documento (detallado en el apartado 7.2.2.1) para incluir la información firmada en el reporte de firma.

7.2.2.3 Concatenación de informe y documento

Una alternativa para incluir el documento firmado en el informe de firma es mediante la concatenación del resultado tras la transformación XSL-FO y el documento firmado.

En este caso al registrar la plantilla indicaremos que el modo de inclusión del documento firmado es “Concatenado” y especificaremos la regla de concatenación del mismo, es decir estableceremos el orden y las páginas en el que se unirán informe y documento. Para más información sobre la configuración de las plantillas puede consultarse el manual de Administración [EVISOR-ADMIN-MAN].

7.2.3 Códigos de barras

En el caso de generación de informe PDF podemos incluir un código de barras mediante dos vías:

- Solicitándolo en la Petición WS. En este caso el sistema genera los códigos solicitado como imagen indicando la localización de las imágenes en el XML de entrada.

```
....
<xsl:variable name="barCode" select="/GenerationReport/Barcodes/Barcode/URL"/>
...
<fo:external-graphic>
  <xsl:attribute name="src"><xsl:value-of select="$barCode"/></xsl:attribute>
</fo:external-graphic>
....
```

- Utilizando “Barcode4j” como FOP Extensions. Apache FOP permite añadir “extensiones” para generar aquellos elementos no contemplados en el estándar XSL-FO para ello suministra guías de desarrollo e instalación de estas extensiones (puede consultarse más información en <http://xmlgraphics.apache.org/fop/trunk/extensions.html>). Una implementación de extensión FOP nos la ofrece la librería de generación de códigos de barras “Barcode4j” que nos permite crear códigos de barras mediante transformación XSL-FO (podemos consultar mas información sobre la utilización de la librería Barcode4j en <http://barcode4j.sourceforge.net/2.1/fop-ext.html>).

```
.....
<fo:instream-foreign-object>
  <barcode:barcode xmlns:barcode="http://barcode4j.krysalis.org/ns">
    <xsl:attribute name="message"><xsl:value-of select="$dni"/></xsl:attribute>
    <barcode:pdf417></barcode:pdf417>
  </barcode:barcode>
</fo:instream-foreign-object>
.....
```

8 Diseño de plantillas ODF

Como vimos en el apartado 5.2 el proceso de generación de un informe ODF se realiza mediante transformación XSLT del XML de validación (descrito en el apartado 6) en uno o varios ficheros XML que forman el ODF final.

En este apartado, utilizando un ejemplo, vamos a describir el proceso de diseño de una de estas plantillas.

8.1 Diseño del ODF base

Con nuestro editor de documento ODF podemos crear el fichero ODF que utilizaremos para generar la plantilla. Como ejemplo, supongamos que nuestro informe tiene la siguiente estructura:

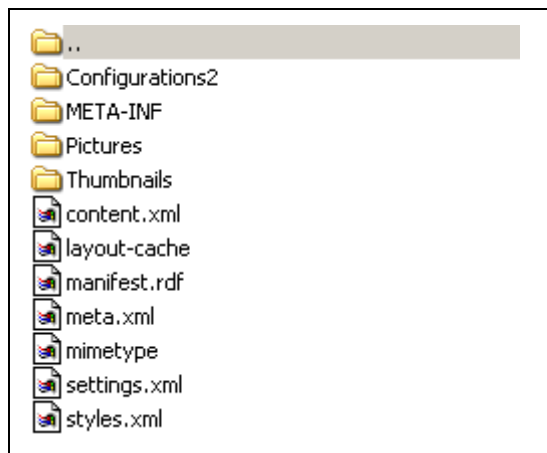
	
	
Datos firmante	

En este caso el informe de firma se estructurará en tres partes:

- Cabecera. En la cabecera se incluirá el logo de la administración y la aplicación.
- Parte central. La parte central del documento se reserva para el documento originalmente firmado.
- Pie de página. En el pie de página colocaremos una tabla que incluirá un código de barras y la información de cada uno de los firmantes del documento.

8.2 Diseño de la plantilla

Si abrimos nuestro ODF base con una herramienta de comprensión (WinZip, WinRAR, etc) podemos ver que nuestro ODF tiene el siguiente contenido:



Para crear el informe deseado deberíamos modificar el contenido de los ficheros:

- styles.xml. Este fichero contiene los estilos utilizados en el documento ODF, en nuestro caso queremos modificar el contenido del pie de página con la información del firmante y el código de barras. En la siguiente figura se representa el elemento que describe el pie de página del documento ODF.

```

...
<style:footer>
  <table:table table:name=" Tabla1" table:style-name=" Tabla1" >
    <table:table-column table:style-name=" Tabla1.A" />
    <table:table-row>
      <table:table-cell table:style-name=" Tabla1.A1" office:value-
type=" string" >
        <text:p text:style-name=" Table_20_Contents" >

          <draw:frame draw:style-name=" Mfr2"
draw:name=" gráficos1" text:anchor-type=" paragraph" svg:width=" 3.358cm"
svg:height=" 0.533cm" draw:z-index=" 5" >
            <draw:image xlink:href=" Pictures/
1000000000000009E0000001B63F7BCBB.png" xlink:type=" simple"
xlink:show=" embed" xlink:actuate=" onLoad" />
          </draw:frame>

        </text:p>
      </table:table-cell>
    </table:table-row>

    <table:table-row>
      <table:table-cell table:style-name=" Tabla1.A2" office:value-
type=" string" >
        <text:p text:style-name=" Table_20_Contents" >Datos
firmante </text:p>
      </table:table-cell>
    </table:table-row>

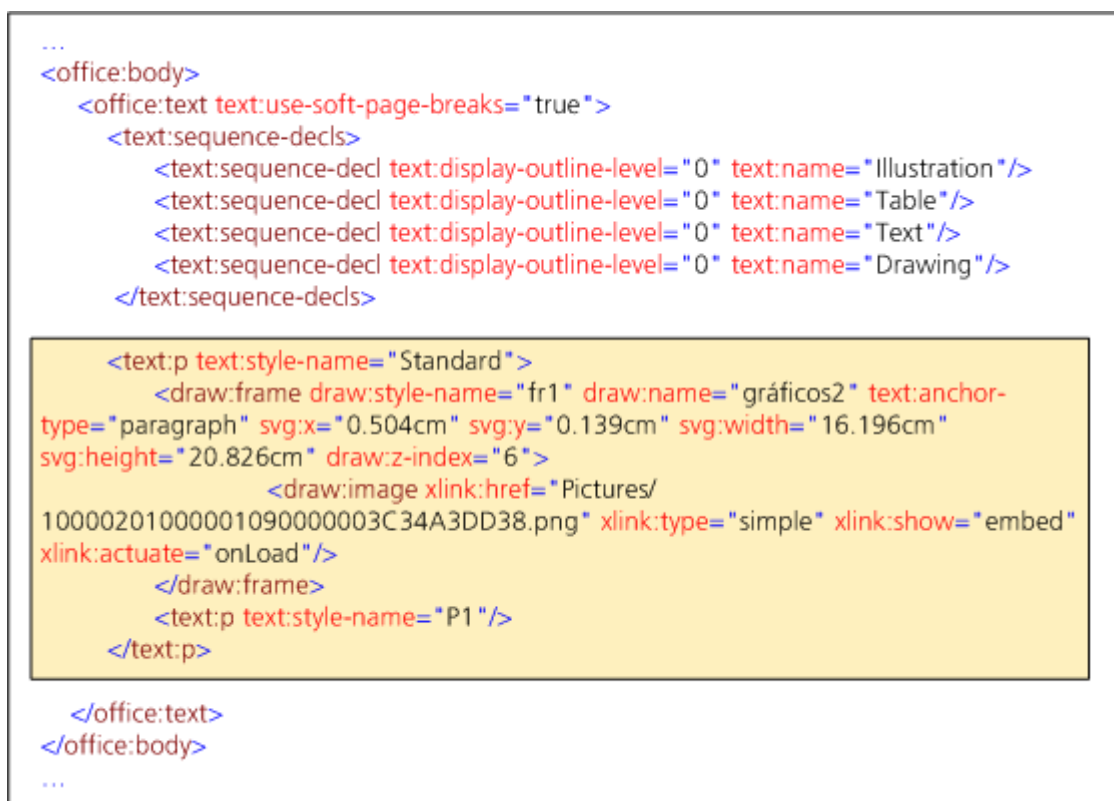
  </table:table>
  <text:p text:style-name=" Footer" />
</style:footer>
.....

```

En el primer recuadro se indica la localización de la imagen del código de barras que sustituiremos por el solicitado en la petición WS y generado por el sistema.

En el segundo cuadro se indica la información del firmante que debe ser completada con el resultado de validación de la firma electrónica.

- content.xml. En este XML se incluye la información presentada en el cuerpo del documento. En este fichero queremos incluir las imágenes de las páginas del ODF originalmente firmado.



La parte remarcada en la figura corresponde a una imagen situada en el centro del documento y destinada a contener la imagen de la página del documento originalmente firmado.

8.2.1 Implementación de los ficheros XSLT

Una vez conocemos los ficheros XML que debemos modificar (style.xml y content.xml) para incluir la información asociada a la firma, procedemos a implementar los ficheros XSLT que nos permitirán obtener los ficheros XML.

8.2.1.1 style.xsl

Partiendo de la información esperada en el XML de entrada (apartado 6) en el que se incluye la información de validación y los códigos de barra solicitados en la petición WS el fichero XSLT necesario para obtener el pie de página deseado sería el siguiente:

<pre> ... <style:footer> <table:table table:name="Tabla1" table:style-name="Tabla1"> <table:table-column table:style-name="Tabla1.A"/> <xsl:for-each select="//ri:Barcode"> <table:table-row> <table:table-cell table:style-name="Tabla1.A1" office:value-type="string"> <text:p text:style-name="Table_20_Contents"> <draw:frame draw:style-name="Mfr2" draw:name="gráficos1" text:anchor- type="paragraph" svg:width="3.358cm" svg:height="0.533cm" draw:z-index="10"> <draw:image xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad"> <xsl:attribute name="href" namespace="http://www.w3.org/1999/xlink"><xsl:value-of select="ri:URL"/></xsl:attribute> </draw:image> </draw:frame> </text:p> </table:table-cell> </table:table-row> </xsl:for-each> </pre>
<pre> <xsl:for-each select="//ri:CertificateInfo"> <xsl:variable name="nombre" select="ri:Field[ri:FieldId='nombreResponsable']/ri:FieldValue"/> <xsl:variable name="primerApellido" select="ri:Field[ri:FieldId='primerApellidoResponsable']/ ri:FieldValue"/> <xsl:variable name="segundoApellido" select="ri:Field[ri:FieldId='segundoApellidoResponsable']/ ri:FieldValue"/> <xsl:variable name="dni" select="ri:Field[ri:FieldId='NIFResponsable']/ri:FieldValue"/> <xsl:variable name="espacio" select="''"/> <table:table-row> <table:table-cell table:style-name="Tabla1.A2" office:value-type="string"> <text:p text:style-name="Table_20_Contents">Firmado por: <xsl:value-of select="\$nombre"/> <xsl:value-of select="\$espacio"/> <xsl:value-of select="\$primerApellido"/> <xsl:value-of select="\$espacio"/> <xsl:value-of select="\$segundoApellido"/>,<xsl:value-of select="\$espacio"/> <xsl:value-of select="\$dni"/> </text:p> </table:table-cell> </table:table-row> </xsl:for-each> </table:table> <text:p text:style-name="Footer"/> </style:footer> ... </pre>

En el primer cuadro se enmarca el establecimiento de la URL del código de barras generado por el sistema y que se incluye en el XML de entrada. En el segundo cuadro se establece la información del firmante.

8.2.1.2 content.xml

Durante el registro de la plantilla ODF podemos establecer el modo de inclusión de documento a “Imagen” (puede consultarse más información sobre la configuración de plantillas en el manual [EVISOR-ADMIN-MAN]), en este caso el sistema imágenes de las páginas del documento firmado e indicará su localización en el XML de entrada a la transformación. A continuación vamos a ver como sería la transformación XSLT para obtener el fichero “content.xml” con las imágenes de las páginas del documento ya incluidas.

```
...
<office:body>
  <office:text text:use-soft-page-breaks="true">
    <text:sequence-decls>
      <text:sequence-decl text:display-outline-level="0" text:name="Illustration"/>
      <text:sequence-decl text:display-outline-level="0" text:name="Table"/>
      <text:sequence-decl text:display-outline-level="0" text:name="Text"/>
      <text:sequence-decl text:display-outline-level="0" text:name="Drawing"/>
    </text:sequence-decls>

    <xsl:for-each select="//ri:Page">
      <text:p text:style-name="Standard">
        <draw:frame draw:style-name="fr1" draw:name="gráficos2" text:anchor-
type="paragraph" svg:x="0.504cm" svg:y="0.139cm" svg:width="16.196cm"
svg:height="20.826cm" draw:z-index="6">
          <draw:image xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad">
            <xsl:attribute name="href" namespace="http://www.w3.org/1999/
xlink"><xsl:value-of select="ri:URL"/></xsl:attribute>
          </draw:image>
        </draw:frame>
        <text:p text:style-name="P1"/>
      </text:p>
    </xsl:for-each>

  </office:text>
</office:body>
...
```

Como observamos en el cuadro remarcado se incluye la localización (URL) de la imagen de la página incluida en el XML de entrada.

8.2.2 Empaquetado de la plantilla

Una vez que tenemos los ficheros XSLT necesarios los incluimos dentro del fichero comprimido que representa el ODF base.

Además debemos generar un fichero con el nombre “**templatereport.properties**” en el que se informe al sistema cuales son los ficheros XSLT que debe aplicar, el contenido de este fichero debe ser:

#Nombre de las plantillas incluidas en el ZIP

template.xml=content.xml,styles.xml

#Nombre con el que se incluirán en el ZIP los resultados de la transformación XSLT

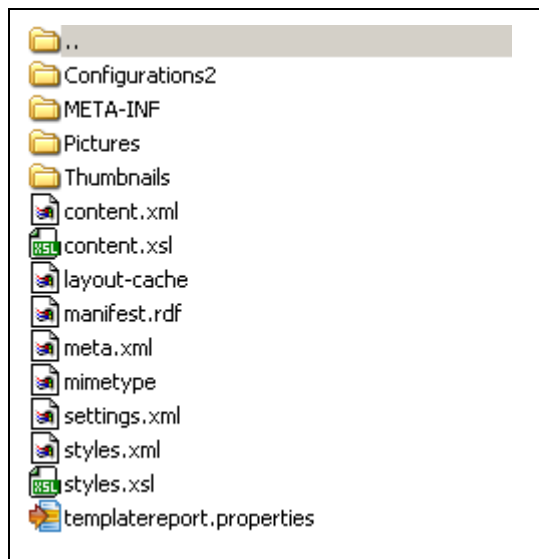
template.results=content.xml,styles.xml

Se debe establecer dos propiedades:

- **template.xml**. Propiedad que indica el nombre de los ficheros XSLT a aplicar. Se puede poner mas de uno separados por el carácter “,”.

- `template.result`. Propiedad que establece el nombre que se tendrán los ficheros resultantes de la transformación. Debe seguir el mismo orden que los establecidos en la propiedad anterior.

Una vez incluido los ficheros XSLT y el fichero de propiedades la plantilla ODF tendrá la siguiente estructura:



ANEXO

A.1 Esquema del mensajes XML de entrada

En la siguiente figura se representa la definición de los componentes que forman el XML de entrada utilizado para la generación de un informe de firma.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:es:gob:signaturereport:generation:inputparameters"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:es:gob:signaturereport:generation:inputparameters">
  <xs:element name="ValidationResult">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Result"/>
        <xs:element ref="IndividualSignature" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="TimeStamp" type="xs:string"/>
  <xs:element name="Result">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Major"/>
        <xs:element ref="Minor" minOccurs="0"/>
        <xs:element ref="Message" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ParameterValue" type="xs:string"/>
  <xs:element name="ParameterId" type="xs:string"/>
  <xs:element name="Parameter">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ParameterId"/>
        <xs:element ref="ParameterValue"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Minor" type="xs:anyURI"/>
  <xs:element name="Message" type="xs:string"/>
  <xs:element name="Major" type="xs:anyURI"/>
  <xs:element name="IndividualSignature">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Result"/>
        <xs:element ref="TimeStamp" minOccurs="0"/>
        <xs:element ref="CertificateInfo" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GenerationReport">
    <xs:complexType>
      <xs:sequence>
```

```

        <xs:element ref="ValidationResult"/>
        <xs:element ref="GenerationTime"/>
        <xs:element ref="ExternalParameters" minOccurs="0"/>
        <xs:element ref="DocumentInfo" minOccurs="0"/>
        <xs:element ref="Barcodes" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="GenerationTime" type="xs:string"/>
<xs:element name="FieldValue" type="xs:string"/>
<xs:element name="FieldId" type="xs:string"/>
<xs:element name="Field">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="FieldId"/>
            <xs:element ref="FieldValue"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ExternalParameters">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Parameter"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CertificateInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Field"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="DocumentInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="NumPages"/>
            <xs:element ref="PagesList" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="NumPages" type="xs:integer"/>
<xs:element name="PagesList">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Page" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Page">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Number"/>
            <xs:element ref="URL"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```
<xs:element name="Number" type="xs:integer"/>
<xs:element name="URL" type="xs:string"/>
<xs:element name="Barcodes">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Barcode" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Barcode">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Type"/>
      <xs:element ref="Code"/>
      <xs:element ref="URL"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Type" type="xs:string"/>
<xs:element name="Code" type="xs:string"/>
<xs:element name="IncludePage">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="DocumentPage"/>
      <xs:element ref="ReportPage"/>
    </xs:sequence>
    <xs:attribute name="Xpos" type="xs:string"/>
    <xs:attribute name="Ypos" type="xs:string"/>
    <xs:attribute name="Width" type="xs:string"/>
    <xs:attribute name="Height" type="xs:string"/>
    <xs:attribute name="Layout" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="DocumentPage" type="xs:integer"/>
<xs:element name="ReportPage" type="xs:integer"/>
</xs:schema>
```

A.2 Formatos de documentos soportados

El sistema permite incluir en el informe información sobre los documentos originalmente firmado, en la siguiente tabla se muestran los formatos soportados por las plantillas PDF y ODF para la generación de informes

FORMATO	MIME TYPE	PDF	ODF
PDF	application/pdf	✓	✓
ODT	application/vnd.oasis.opendocument.text	✓	✓
ODP	application/vnd.oasis.opendocument.presentation	✓	✓
ODS	application/vnd.oasis.opendocument.spreadsheet	✓	✓
ODG	application/vnd.oasis.opendocument.graphics	✓	✓
RTF	application/rtf	✓	✓
DOC	application/msword	✓	✓
DOCX	application/vnd.openxmlformats-officedocument.wordprocessingml.document	✓	✓
PPT	application/vnd.ms-powerpoint	✓	✓
PPTX	application/vnd.openxmlformats-officedocument.presentationml.presentation	✓	✓
XLS	application/vnd.ms-excel	✓	✓
XLSX	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	✓	✓
JPEG	image/jpeg	✓	✓
GIF	image/gif	✓	✓
PNG	image/png	✓	✓
BMP	image/x-ms-bmp	✓	✓
XML	application/xml	✓	✓
TXT	text/plain	✓	✓
XSLT	application/xslt+xml	✓	✓
HTML	text/html	✓	✓