

Requisitos para la generación de Xml válido

Referencia: AST_iBOA_RequisitosGeneracionXml.doc

Categoría: Público

Compañía: Eurohelp Consulting

Autor: D. Sancho

Versión: 1.0.0

Fecha de creación: 30 de julio de 2012

Última actualización: 31 de julio de 2012

Datos del Documento

Registro de Cambios

Versión	Fecha	Autor	Descripción
1.0.0	30/07/2012	D. Sancho	Creación del documento

Lista de distribución

Nombre	Empresa/Departamento
Todo el personal	Eurohelp Consulting S.L.
Todo el personal	Alcañiz Fresnos
Todo el personal	AST

Lista de revisión

Nombre	Fecha	Cargo	Empresa/Departamento
M. A. Pina	31/07/2012	Jefe de proyecto	Eurohelp Consulting S.L.

Contenido

1. INTRODUCCIÓN.....	6
2. ESQUEMA DEL XML	7
3. CUESTIONES GENERALES DE LA GENERACIÓN DEL XML	12
3.1. FUNCIONAMIENTO GENERAL	12
3.2. DIFERENCIAS ENTRE XML DE ENTRADA Y DE SALIDA	12
3.3. PROCESADO DEL XML EN LA FINALIZACIÓN DEL BOLETÍN	12
4. ELEMENTOS DEL XML.....	14
4.1. ROOT.....	14
4.1.1. Tipo	14
4.1.2. Descripción	14
4.2. NUM_PAG.....	14
4.2.1. Tipo	14
4.2.2. Descripción	14
4.3. DISPOSICION_INI	15
4.3.1. Tipo	15
4.3.2. Descripción	15
4.4. NUM_DISPOSICIONES.....	15
4.4.1. Tipo	15
4.4.2. Descripción	15
4.5. ANYO.....	15
4.5.1. Tipo	15
4.5.2. Descripción	15
4.6. FECHA	15
4.6.1. Tipo	15
4.6.2. Descripción	15
4.6.3. Atributos.....	16
4.7. NUMERO	16
4.7.1. Tipo	16
4.7.2. Descripción	16
4.7.3. Atributos.....	16
4.8. INDICE	16
4.8.1. Tipo	16
4.8.2. Descripción	16
4.9. BOLETIN	17
4.9.1. Tipo	17
4.9.2. Descripción	17
4.10. SUMARIO	17
4.10.1. Tipo	17
4.10.2. Descripción	17
4.11. NEWPAGE.....	17
4.11.1. Tipo	17

4.11.2. Descripción	17
4.12. DETALLE_TEXTO	18
4.12.1. Tipo	18
4.12.2. Descripción	18
4.13. TITULO	18
4.13.1. Tipo	18
4.13.2. Descripción	18
4.14. SECCION.....	18
4.14.1. Tipo	18
4.14.2. Descripción	18
4.14.3. Atributos.....	19
4.15. SUBSECCION	19
4.15.1. Tipo	19
4.15.2. Descripción	19
4.15.3. Atributos.....	19
4.16. TITULO_SEC	19
4.16.1. Tipo	19
4.16.2. Descripción	19
4.17. TITULO_SUBSEC	20
4.17.1. Tipo	20
4.17.2. Descripción	20
4.18. CONTENIDOEMISOR	20
4.18.1. Tipo	20
4.18.2. Descripción	20
4.19. EMISOR	20
4.19.1. Tipo	20
4.19.2. Descripción	20
4.20. CONTENIDOTITULO	21
4.20.1. Tipo	21
4.20.2. Descripción	21
4.21. NUM_PAGINA.....	21
4.21.1. Tipo	21
4.21.2. Descripción	21
4.22. EMISOR_TEXT.....	21
4.22.1. Tipo	21
4.22.2. Descripción	21
4.22.3. Atributos.....	21
4.23. DISPOSICION	22
4.23.1. Tipo	22
4.23.2. Descripción	22
4.23.3. Atributos.....	22
4.24. TITULO_TEXT.....	22
4.24.1. Tipo	22
4.24.2. Descripción	22
4.25. TEXTO	23
4.25.1. Tipo	23
4.25.2. Descripción	23
4.26. P (PÁRRAFO)	23

4.26.1. Tipo	23
4.26.2. Descripción	23
4.27. CENTER (PÁRRAFO CENTRADO)	24
4.27.1. Tipo	24
4.27.2. Descripción	24
4.28. UL (LISTA DESORDENADA)	24
4.28.1. Tipo	24
4.28.2. Descripción	24
4.29. OL (LISTA ORDENADA)	24
4.29.1. Tipo	24
4.29.2. Descripción	24
4.30. LI (ELEMENTO DE LISTA)	25
4.30.1. Tipo	25
4.30.2. Descripción	25
4.31. EM (CURSIVA)	25
4.31.1. Tipo	25
4.31.2. Descripción	25
4.32. U (SUBRAYADO)	25
4.32.1. Tipo	25
4.32.2. Descripción	26
4.33. ANEXO	26
4.33.1. Tipo	26
4.33.2. Descripción	26
4.33.3. Atributos	26
4.34. TABLE, TGROUP, COLSPEC, TBODY, ROW, ENTRY	26
5. CODIFICACIÓN DE LOS NUEVOS TEXTOS Y EMISORES	28
5.1. CODIFICACIÓN DE LOS NUEVOS EMISORES	28
5.2. CODIFICACIÓN DE LOS NUEVOS TEXTOS	29

1. Introducción

En el presente documento se detallan los requisitos que debe cumplir el xml generado a partir de Indesign para que la subida del mismo a la aplicación iBOA funcione correctamente.

En primer lugar se muestra el esquema del xml, esquema que debe respetarse en todo momento para el correcto procesado del mismo. El segundo apartado menciona cuestiones generales de obligado cumplimiento durante la manipulación del xml. A continuación se detalla cada uno de los elementos que forman la estructura del xml y en el último apartado se da información complementaria sobre cómo debe introducirse un nuevo texto o emisor en caso de que sea requerido.

2. Esquema del xml

El archivo xml generado por Indesign se valida contra un esquema xsd al subirse a la aplicación iBOA, este esquema es el que se muestra a continuación.

Por otra parte el xml de entrada a la aplicación cumple el mismo esquema, únicamente no contiene la etiqueta <indice> ya que esta se genera mediante los scripts.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" xmlns:Q1="xsd">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="num pag"/>
        <xs:element ref="disposicion ini"/>
        <xs:element ref="num disposiciones"/>
        <xs:element ref="boletin"/>
        <xs:element ref="Anyo"/>
        <xs:element ref="Fecha"/>
        <xs:element ref="Numero"/>
        <xs:element ref="indice"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="num pag" type="xs:integer"/>
  <xs:element name="disposicion ini" type="xs:string"/>
  <xs:element name="num disposiciones" type="xs:integer"/>
  <xs:element name="indice" type="xs:string"/>
  <xs:element name="boletin">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="sumario"/>
        <xs:element ref="newpage"/>
        <xs:element ref="detalle texto"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="sumario">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="titulo"/>
        <xs:element maxOccurs="unbounded" ref="seccion"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="newpage" type="xs:string"/>
  <xs:element name="detalle_texto">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="seccion"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Anyo" type="xs:string"/>
  <xs:element name="Fecha">
    <xs:complexType mixed="true">
      <xs:attribute name="fecha" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="Numero">
    <xs:complexType mixed="true">
      <xs:attribute name="nRegBol" type="xs:integer" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="anyo">
    <xs:complexType/>
  </xs:element>
  <xs:element name="fecha">
    <xs:complexType/>
  </xs:element>
  <xs:element name="numero">
    <xs:complexType/>
  </xs:element>
</xs:schema>
```

```
</xs:element>
<xs:element name="titulo" type="xs:string"/>
<xs:element name="seccion">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="titulo_sec"/>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="contenidoEmisor"/>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="subseccion"/>
    </xs:sequence>
    <xs:attribute name="codSeccion" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="titulo_sec" type="xs:string"/>
<xs:element name="subseccion">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="titulo_subsec"/>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="contenidoEmisor"/>
    </xs:sequence>
    <xs:attribute name="codSubSeccion" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="titulo_subsec" type="xs:string"/>
<xs:element name="contenidoEmisor">
  <xs:complexType>
    <xs:choice>
      <xs:sequence>
        <xs:element ref="emisor"/>
        <xs:element maxOccurs="unbounded" ref="contenidoTitulo"/>
      </xs:sequence>
      <xs:sequence>
        <xs:element ref="emisor_text"/>
        <xs:element maxOccurs="unbounded" ref="disposicion"/>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="emisor" type="xs:string"/>
<xs:element name="contenidoTitulo">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="titulo"/>
      <xs:element ref="num_pagina"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="num_pagina" type="xs:string"/>
<xs:element name="emisor_text">
  <xs:complexType mixed="true">
    <xs:attribute name="tipoEmisor" type="xs:integer"/>
    <xs:attribute name="codEmisor" type="xs:integer"/>
    <xs:attribute name="fechaEmisor" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="disposicion">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element ref="numero_disposicion" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="titulo_text"/>
      <xs:element ref="texto"/>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="anexo"/>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="enexo"/>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="nuevaPage"/>
    </xs:sequence>
    <xs:attribute name="fechaDis" type="xs:string" use="required"/>
    <xs:attribute name="rango" type="xs:string" use="required"/>
    <xs:attribute name="clave" type="xs:string" use="required"/>
    <xs:attribute name="aley" type="xs:string" use="required"/>
    <xs:attribute name="oley" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="numero_disposicion" type="xs:string" />
<xs:element name="titulo_text" type="xs:string"/>
<xs:element name="texto">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="br"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```



```
<xs:element ref="em"/>
<xs:element ref="strong"/>
<xs:element ref="u"/>
<xs:element ref="p"/>
<xs:element ref="ul"/>
<xs:element ref="ol"/>
<xs:element ref="span"/>
<xs:element ref="anexo int"/>
<xs:element ref="anexo"/>
<xs:element ref="center"/>
<xs:element ref="sup"/>
<xs:element ref="sub"/>
<xs:element ref="firma_izq"/>
<xs:element ref="firma_der"/>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="p">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="br" />
      <xs:element ref="em" />
      <xs:element ref="strong" />
      <xs:element ref="u" />
      <xs:element ref="ul" />
      <xs:element ref="ol" />
      <xs:element ref="span" />
      <xs:element ref="anexo int" />
      <xs:element ref="anexo" />
      <xs:element ref="firma_izq" />
      <xs:element ref="firma_der" />
      <xs:element ref="enexo" />
      <xs:element ref="sup" />
      <xs:element ref="sub" />
      <xs:element ref="table" />
      <xs:element ref="center" />
    </xs:choice>
    <xs:attribute name="id" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<!-- <xs:element name="p" type="xs:string"/> -->
<xs:element name="center">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="strong"/>
      <xs:element ref="em"/>
      <xs:element ref="u"/>
      <xs:element ref="anexo_int" />
      <xs:element ref="anexo" />
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="ul">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="ul"/>
      <xs:element ref="ol"/>
      <xs:element ref="li"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="ol">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="ul"/>
      <xs:element ref="ol"/>
      <xs:element ref="li"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="span">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element minOccurs="0" ref="strong"/>
      <xs:element ref="br"/>
      <xs:element ref="em"/>
      <xs:element ref="u"/>
      <xs:element ref="p"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

```
<xs:element ref="span"/>
<xs:element ref="ul"/>
<xs:element ref="ol"/>
<xs:element ref="anexo int"/>
<xs:element ref="anexo"/>
<xs:element ref="firma_izq"/>
<xs:element ref="firma_der"/>
<xs:element ref="enexo"/>
<xs:element ref="sup"/>
<xs:element ref="sub"/>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="li">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element minOccurs="0" ref="strong"/>
      <xs:element ref="br"/>
      <xs:element ref="em"/>
      <xs:element ref="u"/>
      <xs:element ref="p"/>
      <xs:element ref="span"/>
      <xs:element ref="anexo int"/>
      <xs:element ref="anexo"/>
      <xs:element ref="firma izq"/>
      <xs:element ref="firma der"/>
      <xs:element ref="enexo"/>
      <xs:element ref="sup"/>
      <xs:element ref="sub"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="strong">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="br"/>
      <xs:element ref="em"/>
      <xs:element ref="u"/>
      <xs:element ref="p"/>
      <xs:element ref="span"/>
      <xs:element ref="anexo int"/>
      <xs:element ref="anexo"/>
      <xs:element ref="firma izq"/>
      <xs:element ref="firma der"/>
      <xs:element ref="enexo"/>
      <xs:element ref="sup"/>
      <xs:element ref="sub"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="u">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="br"/>
      <xs:element ref="em"/>
      <xs:element ref="strong"/>
      <xs:element ref="p"/>
      <xs:element ref="span"/>
      <xs:element ref="anexo int"/>
      <xs:element ref="anexo"/>
      <xs:element ref="firma izq"/>
      <xs:element ref="firma der"/>
      <xs:element ref="enexo"/>
      <xs:element ref="sup"/>
      <xs:element ref="sub"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="br">
  <xs:complexType/>
</xs:element>
<xs:element name="em">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="br"/>
      <xs:element ref="u"/>
      <xs:element ref="strong"/>
      <xs:element ref="p"/>
```

```
<xs:element ref="span"/>
<xs:element ref="anexo int"/>
<xs:element ref="anexo"/>
<xs:element ref="firma izq"/>
<xs:element ref="firma der"/>
<xs:element ref="enexo"/>
<xs:element ref="sup"/>
<xs:element ref="sub"/>
</xs:choice>
</xs:complexType>
</xs:element>
<xs:element name="anexo" >
  <xs:complexType mixed="true">
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="tipoAnexo" type="xs:integer" use="required"/>
    <xs:attribute name="type" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="table">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tgroup"/>
    </xs:sequence>
    <xs:attribute name="frame" type="xs:string" />
  </xs:complexType>
</xs:element>
<xs:element name="tgroup">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" ref="colspec"/>
      <xs:element ref="tbody"/>
    </xs:sequence>
    <xs:attribute name="cols" type="xs:integer" />
  </xs:complexType>
</xs:element>
<xs:element name="colspec">
  <xs:complexType mixed="true">
    <xs:attribute name="colname" type="xs:string" />
    <xs:attribute name="colwidth" type="xs:string" />
  </xs:complexType>
</xs:element>
<xs:element name="tbody">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="row"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="row">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="entry"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="entry">
  <xs:complexType mixed="true">
    <xs:attribute name="morerows" type="xs:string" use="optional"/>
    <xs:attribute name="namest" type="xs:string" use="optional"/>
    <xs:attribute name="nameend" type="xs:string" use="optional"/>
    <xs:attribute name="colsep" type="xs:integer" use="optional"/>
    <xs:attribute name="rowsep" type="xs:integer" use="optional"/>
    <xs:attribute name="align" type="xs:string" use="required"/>
    <xs:attribute name="valign" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="enexo" type="xs:string"/>
<xs:element name="anexo int" type="xs:string"/>
<xs:element name="firma izq" type="xs:string"/>
<xs:element name="firma_der" type="xs:string"/>
<xs:element name="sup" type="xs:string"/>
<xs:element name="sub" type="xs:string"/>
</xs:schema>
```

3. Cuestiones generales de la generación del xml

3.1. Funcionamiento general

El xml de entrada es utilizado para importar el boletín en Indesign, siendo la estructura de Indesign una copia del mismo (con los cambios necesarios para su correcto funcionamiento). Los cambios en Indesign (reflejados en su estructura) deben ser coherentes con el xml para el funcionamiento correcto de los scripts.

De forma general, el contenido del sumario debe ser coherente con el contenido del texto y el número de disposiciones (entendiendo ahora el número de disposiciones y anuncios) debe coincidir con el indicado en la etiqueta `<num_disposiciones>` del xml y la estructura de Indesign.

Por otra parte no deben existir secciones, subsecciones o emisores que no tengan contenido, es decir, no puede haber etiquetas `<seccion>`, `<subseccion>` o `<contenidoEmisor>` si no contienen textos (etiquetas `<contenidoTitulo>` en el sumario y `<disposición>` en el contenido).




Todos los códigos existentes en el xml de entrada (codificados en muchos casos en los atributos) deben permanecer inalterados a lo largo de todo proceso y mostrarse en el xml final. De la misma forma los textos asociados a un emisor deben permanecer asociados a este para que no falle la carga del xml. Es posible borrar un texto del boletín o modificar su contenido, pero no se pueden modificar sus códigos ni cambiar su emisor en ningún caso.

Los elementos del xml deben estar en el mismo orden en el que aparecen en el esquema xsd, en caso de ser obligatorios deben aparecer aunque no tengan función aparente.

En caso de que se deba añadir un nuevo emisor o texto al boletín, se debe respetar una codificación especial que se explica en el apartado 5 de esta guía.


3.2. Diferencias entre xml de entrada y de salida






Existen algunas diferencias importantes entre el xml que se utiliza para la importación del boletín y el xml que se obtiene tras la finalización del mismo. Estas diferencias son:

-  El xml de entrada no tiene el elemento `<indice>` creado al final del mismo, por lo que la validación con el esquema del punto 2 fallará al no encontrarlo.
-  El xml de entrada contiene las cadenas **`**ini_disposicion#`** y **`**fin_disposicion#`**. Estas marcas de inicio y fin de disposición son usadas por el script *BoalImport.jsxbin* y eliminadas tras su ejecución.
-  El xml de entrada contiene además caracteres de saltos de línea, que son eliminados al generar el xml de salida.

3.3. Procesado del xml en la finalización del boletín

Durante el proceso de maquetación se pueden crear etiquetas que no son compatibles con el esquema del boletín, por lo que antes de generar el xml final se hacen los siguientes cambios:

-  Se sustituyen las etiquetas `<ul1></ul1>`, `<ul2></ul2>`, `<ul3></ul3>`, `<ul4></ul4>`, `<ul5></ul5>`, `<ul6></ul6>`, `<ul7></ul7>`, `<ul8></ul8>` y `<ul9></ul9>` por las etiquetas ``.

-  Se sustituyen las etiquetas `<ol1></ol1>`, `<ol2></ol2>`, `<ol3></ol3>`, `<ol4></ol4>`, `<ol5></ol5>`, `<ol6></ol6>`, `<ol7></ol7>`, `<ol8></ol8>` y `<ol9></ol9>` por las etiquetas ``.
-  Se elimina el atributo "lista" de la etiqueta `<p>` en caso de contener un anexo, es decir, si aparece `<p id="lista"><anexo>` debe quedar como `<p><anexo>`.
-  Se sustituyen los caracteres `"` (ASCII 93 y 94) por comillas simples `'` (ASCII 22).
-  Se sustituye el carácter `—` (ASCII 97) por `-` (ASCII 20).
-  Se eliminan los caracteres de salto de línea.

4. Elementos del xml

A continuación se detallan los elementos del xml que son necesarios para la correcta formación del boletín.

4.1. root

4.1.1. Tipo

Tipo complejo que contiene la secuencia de elementos:

- num_pag (ver 4.2).
- disposicion_ini (ver 4.3).
- num_disposiciones (ver 4.4).
- boletin (ver 4.9).
- Anyo (ver 4.5).
- Fecha (ver 4.6).
- Numero (ver 4.7).
- indice (ver 4.8).

4.1.2. Descripción

Elemento principal del documento xml que contiene a todos los demás, estando estructurados a partir de él.

4.2. num_pag

4.2.1. Tipo

Xs:integer.

4.2.2. Descripción

Número de página con la que empieza el boletín, corresponde a la página de la portada (primera página del sumario).

4.3. disposicion_ini

4.3.1. Tipo

Xs:string.

4.3.2. Descripción

Número de disposición inicial, actualmente no se utiliza (se pasará una etiqueta vacía).

4.4. num_disposiciones

4.4.1. Tipo

Xs:string.

4.4.2. Descripción

Número de textos (disposiciones y anuncios) que contiene el boletín.

4.5. Anyo

4.5.1. Tipo

Xs:string.

4.5.2. Descripción

Cadena con el año de edición del boletín en números romanos (por ejemplo: "XXXI"). Se cuenta de forma correlativa a partir del primer año que se editó el B.O.A. y aparece en la portada del boletín.

4.6. Fecha

4.6.1. Tipo

Xs:string.

4.6.2. Descripción

Fecha del boletín que aparece en la portada (por ejemplo: "30 de julio de 2012").

4.6.3. Atributos

- **fecha** (xs:string): contiene la fecha del boletín en el formato dd/mm/yyyy (por ejemplo: "30/07/2012").

4.7. Numero

4.7.1. Tipo

Xs:integer.

4.7.2. Descripción

Número de publicación del boletín.

4.7.3. Atributos

- **nRegBol** (xs:integer): número del registro en base de datos del boletín. Este número no se debe modificar al cambiar el número de publicación del boletín ya que hace referencia a la clave en la base de datos.

4.8. indice

4.8.1. Tipo

Xs:string.

4.8.2. Descripción

Contiene una cadena con los índices de cada uno de los textos (disposiciones y anuncios) y del documento completo de la forma:

"PaginaInicioDisposicion1-PaginaFinDisposicion1;PaginaInicioDisposicion2-PaginaFinDisposicion2; ...
;PaginaInicioBoletin-PaginaFinBoletin"

Por ejemplo una cadena "34-34;35-35;36-36;33-36" hace referencia a un boletín con un texto de una página en las páginas 34, 35 y 36, siendo el boletín completo desde la página 33 a la 36 (al incluir el sumario).

4.9. boletin

4.9.1. Tipo

Tipo complejo que contiene la secuencia de elementos:

- `sumario` (ver 4.10).
- `newpage` (ver 4.11).
- `detalle_texto` (ver 4.12).

4.9.2. Descripción

Elemento que almacena los contenidos del boletín, tanto el sumario como el detalle de cada una de las disposiciones y anuncios del mismo.

4.10. sumario

4.10.1. Tipo

Tipo complejo que contiene la secuencia de elementos:

- `titulo` (ver 4.13).
- `seccion` (ver 4.14).

4.10.2. Descripción

Elemento que almacena los contenidos del boletín referidos al sumario.

4.11. newpage

4.11.1. Tipo

Xs:string.

4.11.2. Descripción

Etiqueta que se utiliza para la correcta maquetación del boletín, no tiene contenido asociado.

4.12. detalle_texto

4.12.1. Tipo

Tipo complejo que contiene la secuencia de elementos:

- seccion (ver 4.14).

4.12.2. Descripción

Elemento que almacena los contenidos del boletín referidos al detalle de cada una de las disposiciones y anuncios.

4.13. titulo

4.13.1. Tipo

Xs:string.

4.13.2. Descripción

Etiqueta que se utiliza en dos lugares:

1. Dentro de <sumario> especificando el título del mismo "Sumario" en la portada.
2. Dentro de <contenidoTitulo> especifica el título de cada uno de los textos en el sumario.

4.14. seccion

4.14.1. Tipo

Tipo complejo que contiene la secuencia de elementos:

- titulo_sec (ver 4.16).
- contenidoEmisor (ver 4.18).
- subseccion (ver 4.15).

4.14.2. Descripción

Este elemento representa cada una de las secciones en las que se divide el boletín ("I. Disposiciones Generales", "II. Autoridades y Personal", etc). Se utiliza dentro de las etiquetas <sumario> y <detalle_texto>, usándose una por cada una de las secciones que están presentes en el boletín. En caso de tener subsecciones la información se encuentra bajo la etiqueta <subseccion>, en caso de no tener subsecciones la información se encuentra en cada una de las etiquetas <contenidoEmisor> que dependen de esta.

Si la etiqueta <seccion> depende de <sumario> el atributo "codSeccion" puede dejarse vacío, en caso contrario debe estar informado con el código correcto.

4.14.3. Atributos

- **codSeccion** (xs:string): código de la sección, la lista de códigos de sección válidos puede obtenerse a partir del archivo xmlAux.xml.

4.15. subseccion

4.15.1. Tipo

Tipo complejo que contiene la secuencia de elementos:

- titulo_subsec (ver 4.17).
- contenidoEmisor (ver 4.18).

4.15.2. Descripción

Este elemento representa cada una de las subsecciones en las que se divide una sección ("a) Nombramientos, situaciones e incidencias", "b) Oposiciones y concursos", etc). Se utiliza en la etiqueta <seccion> bajo las ramas de las etiquetas <sumario> y <detalle_texto>, usándose una por cada una de las subsecciones que están presentes en la sección dada. En cualquier caso la información se encuentra en cada una de las etiquetas <contenidoEmisor> que dependen de esta.

Si la etiqueta <subseccion> depende de <sumario> el atributo "codSubSeccion" puede dejarse vacío, en caso contrario debe estar informado con el código correcto.

4.15.3. Atributos

- **codSubSeccion** (xs:string): código de la subsección, la lista de códigos de sección válidos puede obtenerse a partir del archivo xmlAux.xml.

4.16. titulo_sec

4.16.1. Tipo

Xs:string.

4.16.2. Descripción

Contiene la descripción de la sección (por ejemplo, "I. Disposiciones Generales", "II. Autoridades y Personal", etc).

4.17. titulo_subsec

4.17.1. Tipo

Xs:string.

4.17.2. Descripción

Contiene la descripción de la subsección (por ejemplo, "a) Nombramientos, situaciones e incidencias", "b) Oposiciones y concursos", etc).

4.18. contenidoEmisor

4.18.1. Tipo

Tipo complejo que contiene la siguiente secuencia de elementos si se encuentra en una rama dependiente de la etiqueta <sumario>:

- emisor (ver 4.19).
- contenidoTitulo (ver 4.20).

En caso de encontrarse en una rama dependiente de la etiqueta <detalle_texto> contiene la siguiente secuencia de elementos:

- emisor_text (ver 4.22).
- disposicion (ver 4.23).

4.18.2. Descripción

Este elemento representa cada uno de los emisores y departamentos que contiene cada sección o subsección. Contiene la descripción del emisor o departamento así como la relación de cada uno de los textos que publica.

4.19. emisor

4.19.1. Tipo

Xs:string.

4.19.2. Descripción

Descripción del emisor o departamento.

4.20. contenidoTitulo

4.20.1. Tipo

Tipo complejo que contiene la siguiente secuencia de elementos:

- titulo (ver 4.13).
- num_pagina (ver 4.21).

4.20.2. Descripción

Este elemento contiene la descripción del título de cada uno de los textos publicados por el mismo emisor o departamento.

4.21. num_pagina

4.21.1. Tipo

Xs:string.

4.21.2. Descripción

Elemento auxiliar que contiene un salto de línea.

4.22. emisor_text

4.22.1. Tipo

Xs:string.

4.22.2. Descripción

Descripción del emisor o departamento.

Los atributos de este elemento codifican el emisor del anuncio o disposición, no pudiendo ser nulos y teniendo que ser coherentes con los datos de la aplicación iBOA. Los elementos <emisor> del archivo xmlAux.xml contienen los códigos válidos ya que este elemento contiene los atributos "codEmi" (código de emisor), "codTipEmi" (tipo de emisor) y "fechaEmi" (fecha de emisor) de los emisores válidos para la aplicación iBOA. En caso de observarse varios emisores con los mismos atributos y que sólo difieran en la fecha de emisor, debe usarse la fecha más reciente.

4.22.3. Atributos

- **tipoEmisor** (xs:integer): código interno de la aplicación que especifica el tipo de emisor usado.

- **codEmisor** (xs:integer): código interno de la aplicación que especifica el código del emisor usado.
- **fechaEmisor** (xs:string): código interno de la aplicación que especifica la fecha del emisor referenciado por su código y tipo de emisor.

4.23. disposicion

4.23.1. Tipo

Tipo complejo que contiene la siguiente secuencia de elementos:

- **titulo_text** (ver 4.24).
- **texto** (ver 4.25).
- **newpage** (ver 4.11).

Los elementos "numero_disposicion", "anexo" y "enexo" actualmente no se utilizan en la generación del xml, por lo que pueden ignorarse.

4.23.2. Descripción

Este elemento representa cada uno de los textos (ya sean anuncios o disposiciones) existentes en el boletín. Los atributos del mismo contienen la codificación del texto en la aplicación iBOA (estos atributos no deben modificarse durante la edición en Indesign).

4.23.3. Atributos

- **fechaDis** (xs:string): fecha de la disposición en la aplicación iBOA.
- **rango** (xs:string): codificación del tipo de texto en la aplicación iBOA.
- **clave** (xs:string): clave que contiene los códigos del texto en la aplicación iBOA.
- **aley** (xs:string): codificación del año (si se ha especificado uno) en la aplicación iBOA.
- **oley** (xs:string): codificación del número de texto (si se ha especificado uno) en la aplicación iBOA.

4.24. titulo_text

4.24.1. Tipo

Xs:string.

4.24.2. Descripción

Título del anuncio o disposición.

4.25. texto

4.25.1. Tipo

Tipo complejo que contiene uno o varios de los siguientes elementos:

- p (ver 4.26).
- center (ver 4.27).
- ul (ver 4.28).
- ol (ver 4.29).
- anexo (ver 4.33).

El resto de elementos definidos en el xsd actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.25.2. Descripción

Este elemento contiene cada una de las etiquetas que forman el contenido de una disposición o anuncio (párrafos, listados o anexos).

4.26. p (párrafo)

4.26.1. Tipo

Tipo complejo que puede contener uno o varios de los siguientes elementos:

- em (ver 4.31).
- u (ver 4.32).
- anexo (ver 4.33).

El resto de elementos definidos en el xsd actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.26.2. Descripción

Este elemento representa cada uno de los párrafos del texto y su valor corresponde con el texto de los mismos.

4.27. center (párrafo centrado)

4.27.1. Tipo

Tipo complejo que puede contener uno o varios de los siguientes elementos:

- em (ver 4.31).
- u (ver 4.32).
- anexo (ver 4.33).

Los elementos “strong” y “anexo_int” actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.27.2. Descripción

Este elemento representa cada uno de los párrafos **centrados** del texto y su valor corresponde con el texto de los mismos.

4.28. ul (lista desordenada)

4.28.1. Tipo

Tipo complejo que contiene uno o varios de los siguientes elementos:

- li (ver 4.30).

Los elementos “ul” y “ol” actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.28.2. Descripción

Este elemento representa una lista desordenada y esta compuesto por elementos .

4.29. ol (lista ordenada)

4.29.1. Tipo

Tipo complejo que contiene uno o varios de los siguientes elementos:

- li (ver 4.30).

Los elementos “ul” y “ol” actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.29.2. Descripción

Este elemento representa una lista ordenada y esta compuesto por elementos .

4.30. li (elemento de lista)

4.30.1. Tipo

Tipo complejo que puede contener uno o varios de los siguientes elementos:

- em (ver 4.31).
- u (ver 4.32).
- anexo (ver 4.33).

El resto de elementos definidos en el xsd actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.30.2. Descripción

Este elemento representa cada uno de los elementos del listado y su valor corresponde con el texto de los mismos.

4.31. em (cursiva)

4.31.1. Tipo

Tipo complejo que puede contener uno o varios de los siguientes elementos:

- u (ver 4.32).
- anexo (ver 4.33).

El resto de elementos definidos en el xsd actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.31.2. Descripción

Este elemento representa un texto en cursiva.

4.32. u (subrayado)

4.32.1. Tipo

Tipo complejo que puede contener uno o varios de los siguientes elementos:

- em (ver 4.31).
- anexo (ver 4.33).

El resto de elementos definidos en el xsd actualmente no se utilizan en la generación del xml bajo este elemento, por lo que pueden ignorarse.

4.32.2. Descripción

Este elemento representa un texto subrayado.

4.33. anexo

4.33.1. Tipo

Xs:string.

4.33.2. Descripción

El elemento representa cada uno de los anexos del boletín, ya sean anexos en formato pdf o tiff. En el xml inicial el valor del elemento es la descripción de la marca, pero esta descripción puede borrarse durante la maquetación (no pudiéndose borrar ni el elemento ni sus atributos asociados).

4.33.3. Atributos

- **name** (xs:string): nombre del fichero adjunto que contiene el anexo al que se hace referencia.
- **tipoAnexo** (xs:integer): contiene el código del tipo de anexo (0- pdf, 1- tiff).
- **type** (xs:string): tipo, normalmente tendrá el valor "anexo".

4.34. table, tgroup, colspec, tbody, row, entry

Estos elementos permiten que durante el proceso de importación de disposiciones word (usando el script *Inserción Disposición Word.jsxbin*) se añadan las tablas existentes en los archivos word (que en caso de ser añadidas en la aplicación iBOA aparecen como anexos Aspose).

El uso de los mismos es similar al estándar html: se define un elemento <table> que contiene un elemento <tgroup> especificando el número de columnas, este contiene varios <colspec> especificando los atributos de cada columna y un <tbody>. La etiqueta <tbody> tiene un elemento <row> por cada fila existente, y cada elemento <row> contiene a su vez los elementos <entry> correspondientes a las celdas.

Un ejemplo de tabla añadida a través de este método sería:

```
<table frame="all">
  <tgroup cols="3">
    <colspec colname="c1" colwidth="139pt">
    </colspec>
    <colspec colname="c2" colwidth="139pt">
    </colspec>
    <colspec colname="c3" colwidth="139pt">
    </colspec>
```

```
<tbody>
  <row>
    <entry align="left" valign="bottom">Fila 1 Columna 1</entry>
    <entry align="left" valign="bottom">Fila 1 Columna 2</entry>
    <entry align="left" valign="bottom">Fila 1 Columna 3</entry>
  </row>
  <row>
    <entry align="left" valign="bottom">Fila 2 Columna 1</entry>
    <entry align="left" valign="bottom">Fila 2 Columna 2</entry>
    <entry align="left" valign="bottom">Fila 2 Columna 3</entry>
  </row>
</tbody>
</tgroup>
</table>
```

5. Codificación de los nuevos textos y emisores

En algunos casos es necesario insertar en el boletín textos que no se encontraban en el mismo, siendo en ocasiones necesario crear un nuevo emisor que los contenga ya que los existentes actualmente en la aplicación no sirven.

La creación de un nuevo emisor siempre estará asociada a la inserción de un nuevo texto (no se puede crear un nuevo emisor y mover un texto que pertenecía a otro emisor, siempre se debe respetar la asociación de textos a los emisores que proporciona el xml de entrada).

El archivo xmlAux.xml proporciona todos los códigos de la aplicación iBOA que pueden ser necesarios para la manipulación del xml en Indesign, por lo que siempre se deberá usar una versión actualizada de este archivo para asegurarnos de que utilizamos la información más reciente.

5.1. Codificación de los nuevos emisores





Cuando revisamos la etiqueta <emisor_text> en el xml podemos ver algo similar a esto:

```
<emisor_text tipoEmisor="5" codEmisor="1281" fechaEmisor="03/01/2012">DEPARTAMENTO DE OBRAS PÚBLICAS, URBANISMO, VIVIENDA Y TRANSPORTES</emisor_text>
```

La lista completa de emisores se encuentra en el archivo xmlAux.xml bajo la etiqueta <comboEmisores>, en concreto el emisor equivalente al código anterior es:





```
<emisor codEmi="1281" codTipEmi="5" fechaEmi="03/01/2012">DEPARTAMENTO DE OBRAS PÚBLICAS, URBANISMO, VIVIENDA Y TRANSPORTES</emisor>
```

Puede verse la relación entre los atributos de uno y otro:

-  **tipoEmisor** corresponde con **codTipEmi**.
-  **codEmisor** corresponde con **codEmi**.
-  **fechaEmisor** corresponde con **fechaEmi**.
-  La descripción en ambos casos es la misma.

Cuando queramos insertar un nuevo texto en el boletín y no esté el emisor correspondiente presente en la sección/subsección requerida se debe insertar un nuevo emisor respetando la codificación que muestra el listado del fichero xmlAux.xml. **IMPORTANTE: en caso de existir varios emisores iguales (mismos códigos y descripción) que únicamente se diferencien en la fecha de emisor (fechaEmi) se debe utilizar siempre la fecha más reciente.**

En el caso de que se quiera insertar un emisor nuevo (que aparecerá en la aplicación iBOA como un nuevo emisor independiente posteriormente) se deben insertar los siguientes atributos en la etiqueta <emisor_text>:

-  **tipoEmisor** debe corresponder con el tipo del nuevo emisor, el listado de códigos de los tipos de emisores puede verse en el listado bajo la etiqueta <comboTipoEmisores> del fichero xmlAux.xml.
-  **codEmisor** debe tener valor "-100".
-  **fechaEmi** debe corresponder con el valor de fecha de emisor máxima del listado de emisores bajo la etiqueta <comboEmisores> del archivo xmlAux.xml.
-  La descripción debe corresponder con la descripción en mayúsculas que tendrá el nuevo emisor.

Un ejemplo de un nuevo emisor sería:





```
<emisor text tipoEmisor="4" codEmisor="-100" fechaEmisor="03/01/2012">NUEVO  
EMISOR</emisor_text>
```

5.2. Codificación de los nuevos textos

La codificación de los nuevos textos (etiqueta <disposición>) tiene el siguiente aspecto:

```
<disposicion fechaDis="02/07/2012" rango="15" clave="dsp" aley="" oley="">
```

Los atributos del nuevo texto deben ser:

-  **fechaDis** corresponde con la fecha de la disposición, que puede ser la fecha actual.
-  **rango** corresponde con el tipo de texto, se explicará a continuación cómo calcularlo.
-  **clave** corresponde con las cadenas "asp" o "dsp" en función de si el texto es un anuncio ("asp", los anuncios son los textos pertenecientes a los apartados IV o V del boletín) o si es una disposición ("dsp", las disposiciones son los textos pertenecientes a los apartados I, II o III del boletín).
-  **aley** y **oley** deben quedarse como cadenas vacías.

El rango (le llamaremos tipo de texto en adelante) se calculará para el nuevo texto de la siguiente forma:

1. Se revisará qué tipo de texto queremos insertar verificando la primera palabra en mayúsculas del mismo ("ANUNCIO", "ORDEN", "LEY", etc) y si este es una disposición (pertenece a los apartados I, II o III) o un anuncio (pertenece a los apartados IV o V). También deberemos revisar a qué tipo de emisor pertenece (puede consultarse el código del tipo de emisor que se obtenía en el listado bajo la etiqueta <comboTipoEmisores> del archivo xmlAux.xml).
2. En el fichero xmlAux.xml se buscará el listado <comboRango> y bajo el mismo el apartado correspondiente a las disposiciones (etiqueta <dispo>) o a los anuncios (etiqueta <anun>).
3. Se buscará el elemento <rangoDispo> o <rangoAnun> cuya descripción coincida con el tipo de texto que queremos insertar y cuyo código de tipo de emisor se encuentre en el listado separado por comas del atributo "codTipEmi".
4. Una vez localizado el elemento <rangoDispo> o <rangoAnun> que cumpla lo anterior usamos el código del atributo "codRango" para insertarlo en el atributo "rango" de la nueva disposición o anuncio.