

“Guía de Desarrollo y Uso de Esquemas de Gobierno”

2009

Recomendaciones Generales

Modelar datos y no formatos de documentos

Recomendación

Los esquemas XML DEBERÍAN modelar la estructura de los datos relevantes de un documento, en forma independiente de sus posibles representaciones.

Modelar datos y no formatos de documentos

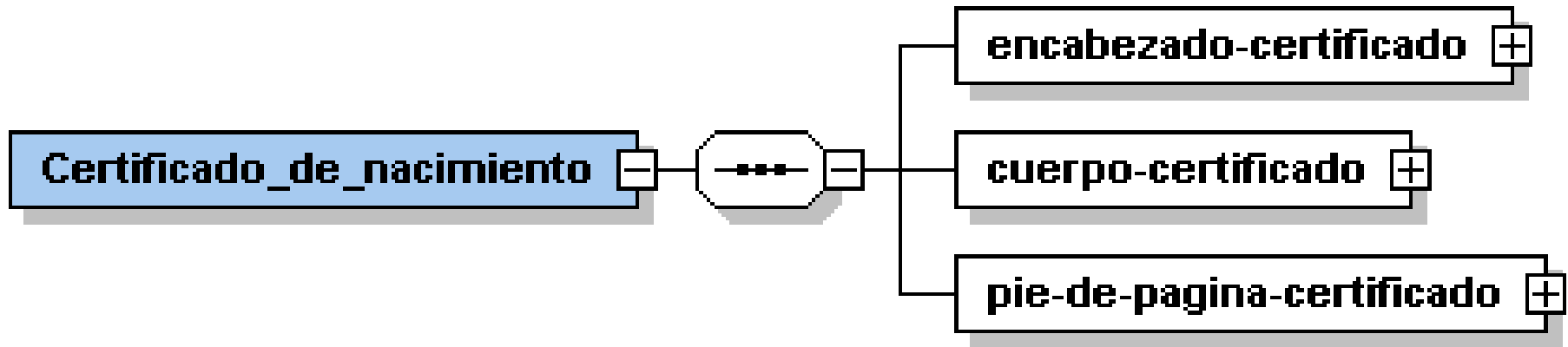
Explicación

Los esquemas XML deberían reflejar la estructura abstracta de los datos, en forma independiente de sus posibles representaciones (formularios, reportes, etc.).

Un esquema XML es la representación específica de la estructura de los datos con la cual el sistema opera. La tarea de modelar los datos implica identificar los tipos de datos, su estructura y sus relaciones.

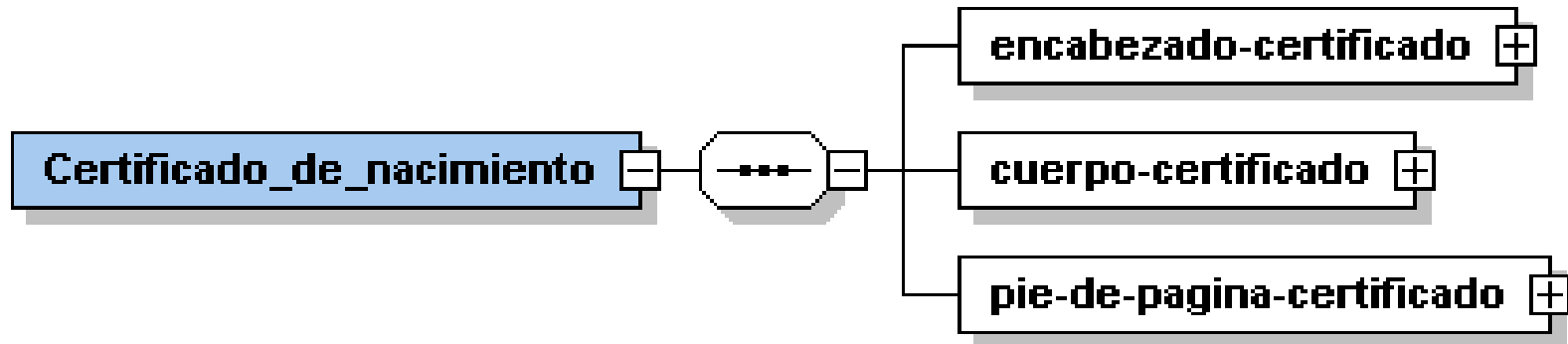
Modelamiento de datos

Esquema inicial

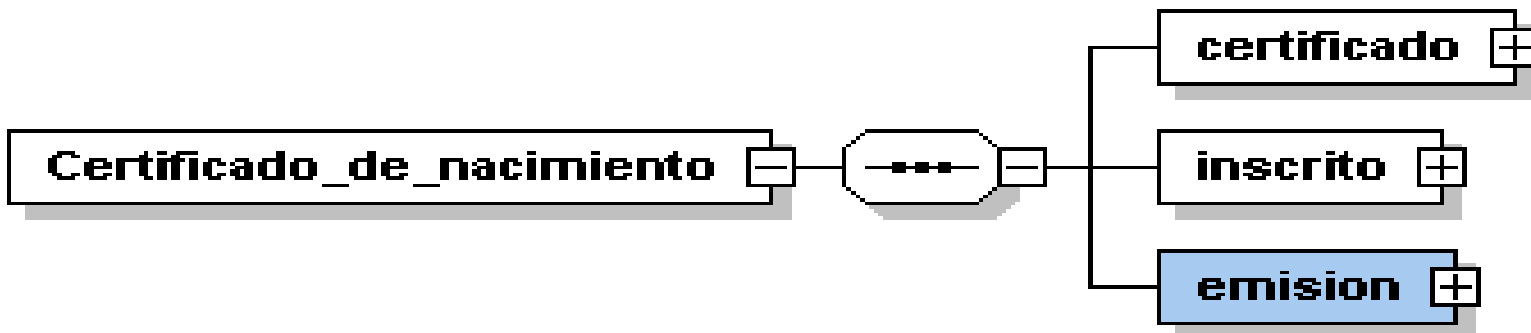


Modelamiento de datos

Esquema inicial

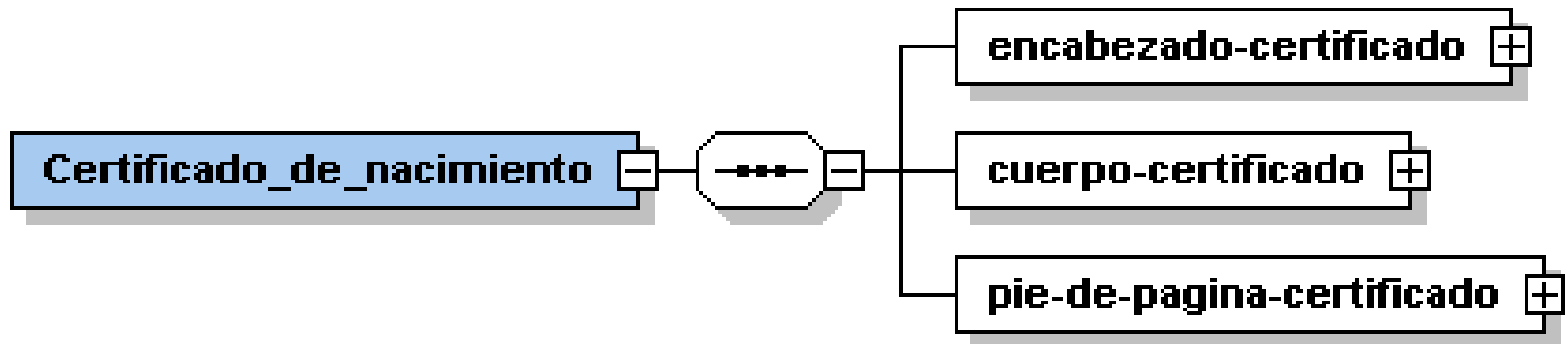


Esquema orientado a datos

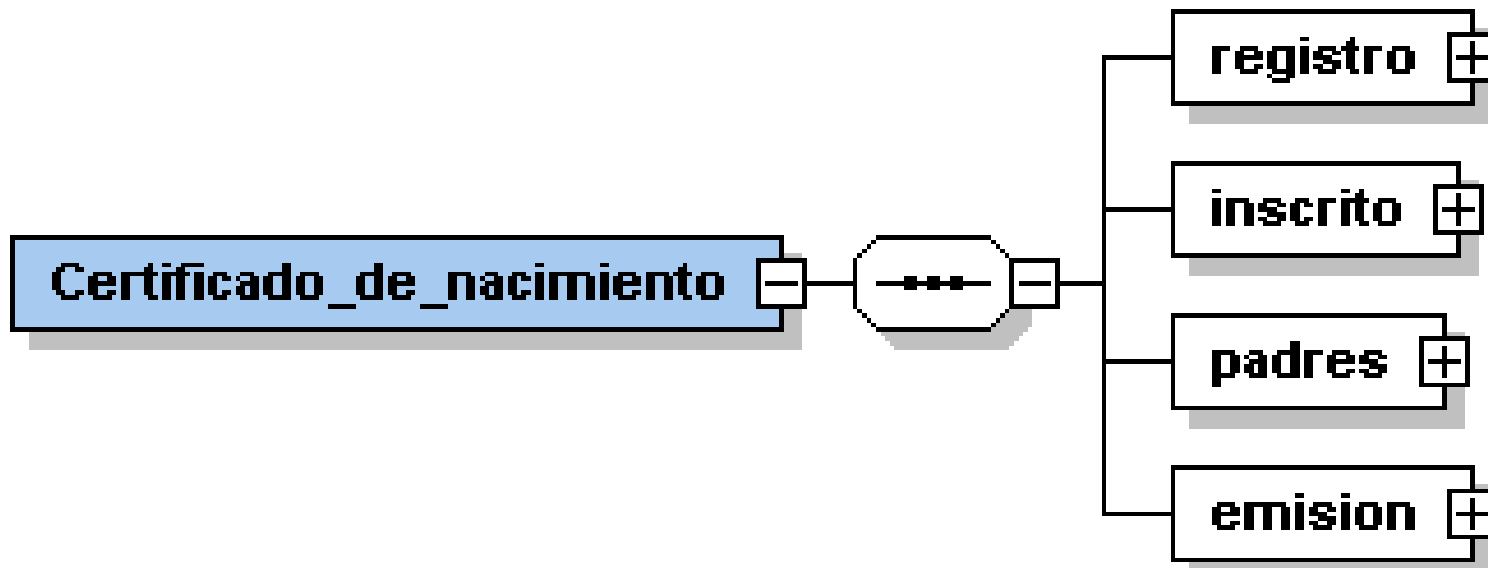


Modelamiento de datos

Esquema inicial

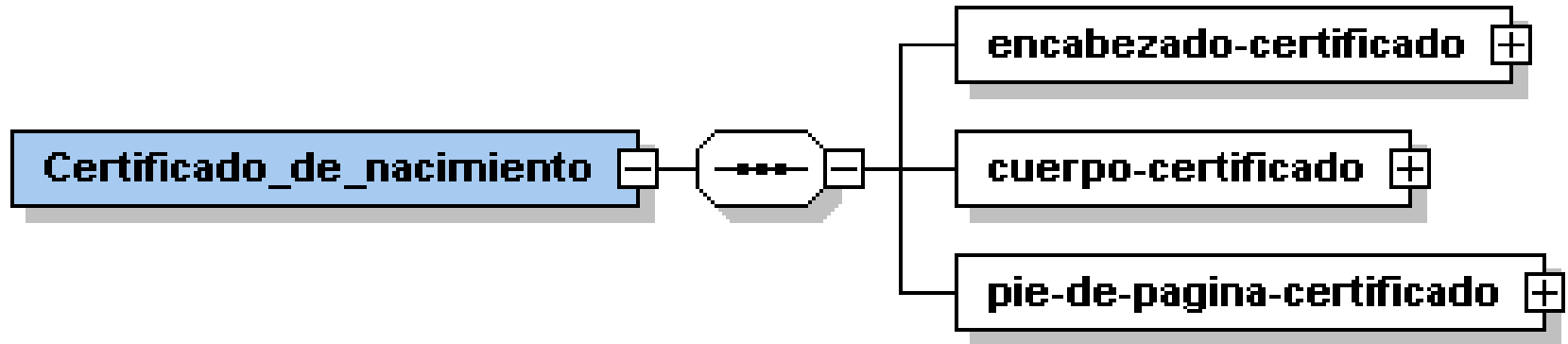


Esquema orientado a datos

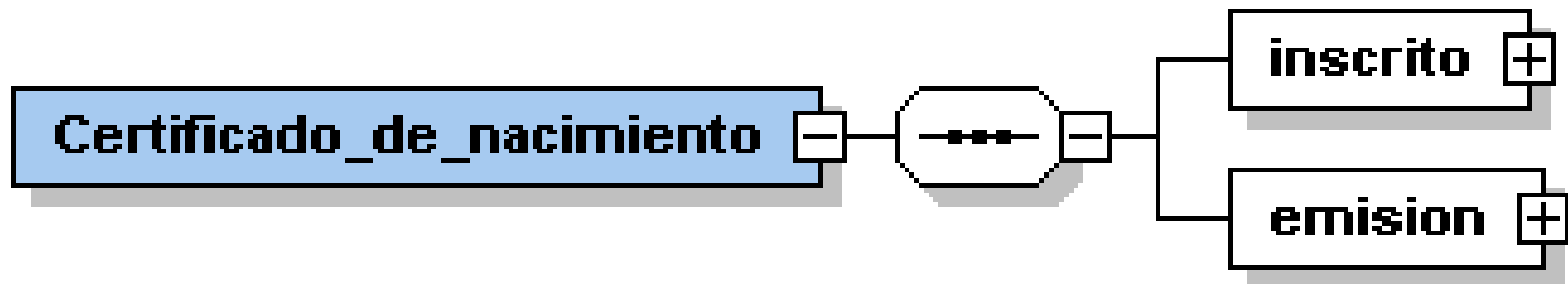


Modelamiento de datos

Esquema inicial

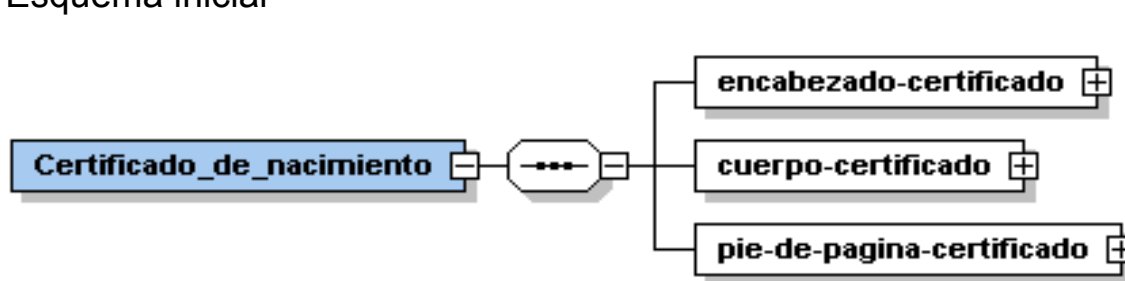


Esquema orientado a datos

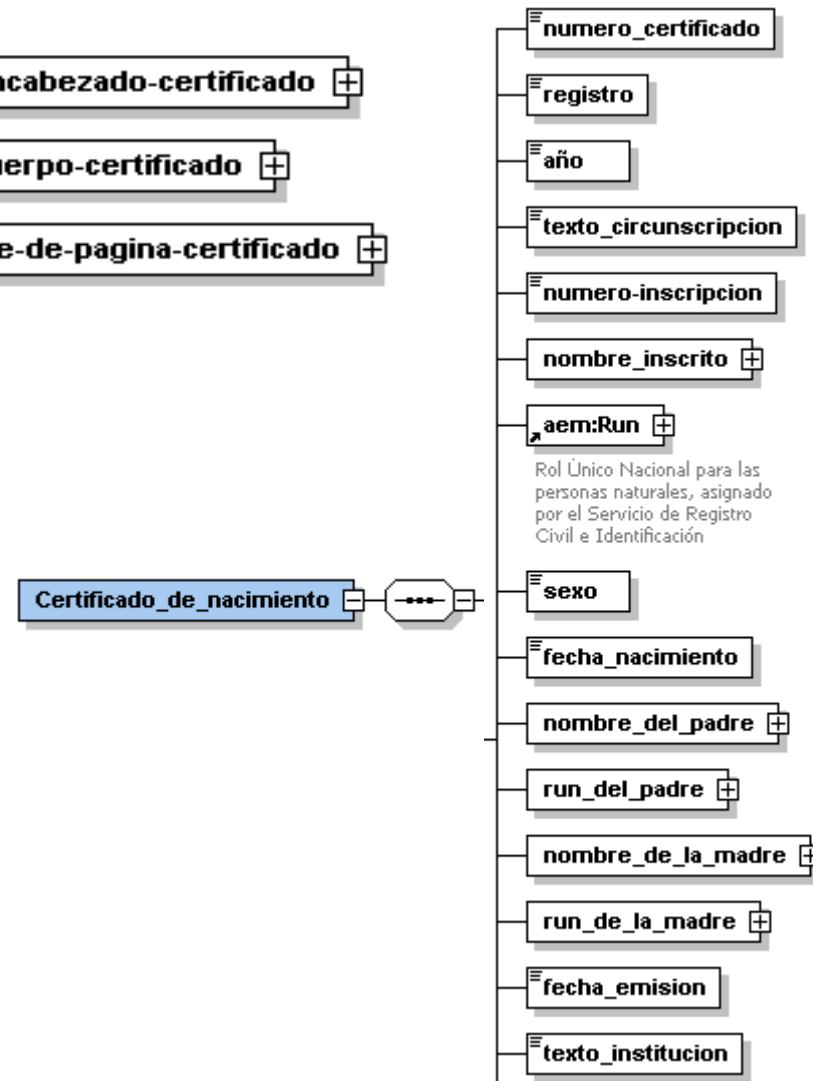


Modelamiento de datos

Esquema inicial



Esquema orientado a datos



Comprensibilidad de los esquemas XML

Recomendación

Los esquemas XML DEBERÍAN ser diseñados en forma clara, coherente e inequívoca.

Comprensibilidad de los esquemas XML

Explicación

Los esquemas XML deberían ser diseñados de forma tal que faciliten su identificación y comprensión.

Estos esquemas deberían contener documentación entendible para el usuario, y donde sea apropiado, deberían contener enlaces a documentos de requerimientos o de diseño.

Simplicidad en el diseño de los esquemas XML

Recomendación

Los esquemas XML DEBERÍAN ser diseñados de manera simple, es decir evitando el uso de funcionalidades poco comunes y esquemas de gran tamaño, y usando las representaciones más intuitivas posibles para los elementos y atributos.

Simplicidad en el diseño de los esquemas XML

Explicación

El lenguaje XML Schema permite variedad y flexibilidad en la definición de esquemas para documentos XML. En la mayoría de los casos, la estructura de los datos puede ser modelada por diversos esquemas XML.

Debido a que todo esquema XML será de dominio público, varios usuarios con diferentes experiencias podrían utilizar el esquema XML propuesto.

Por consiguiente, dicho esquema debería ser definido de la manera más simple posible, con la finalidad de potenciar su reutilización por parte de diferentes usuarios.

Completitud semántica de los esquemas XML

Recomendación

Los esquemas XML DEBEN definir el significado de cada elemento y atributo que será usado en los documentos electrónicos de una organización.

Completitud semántica de los esquemas XML

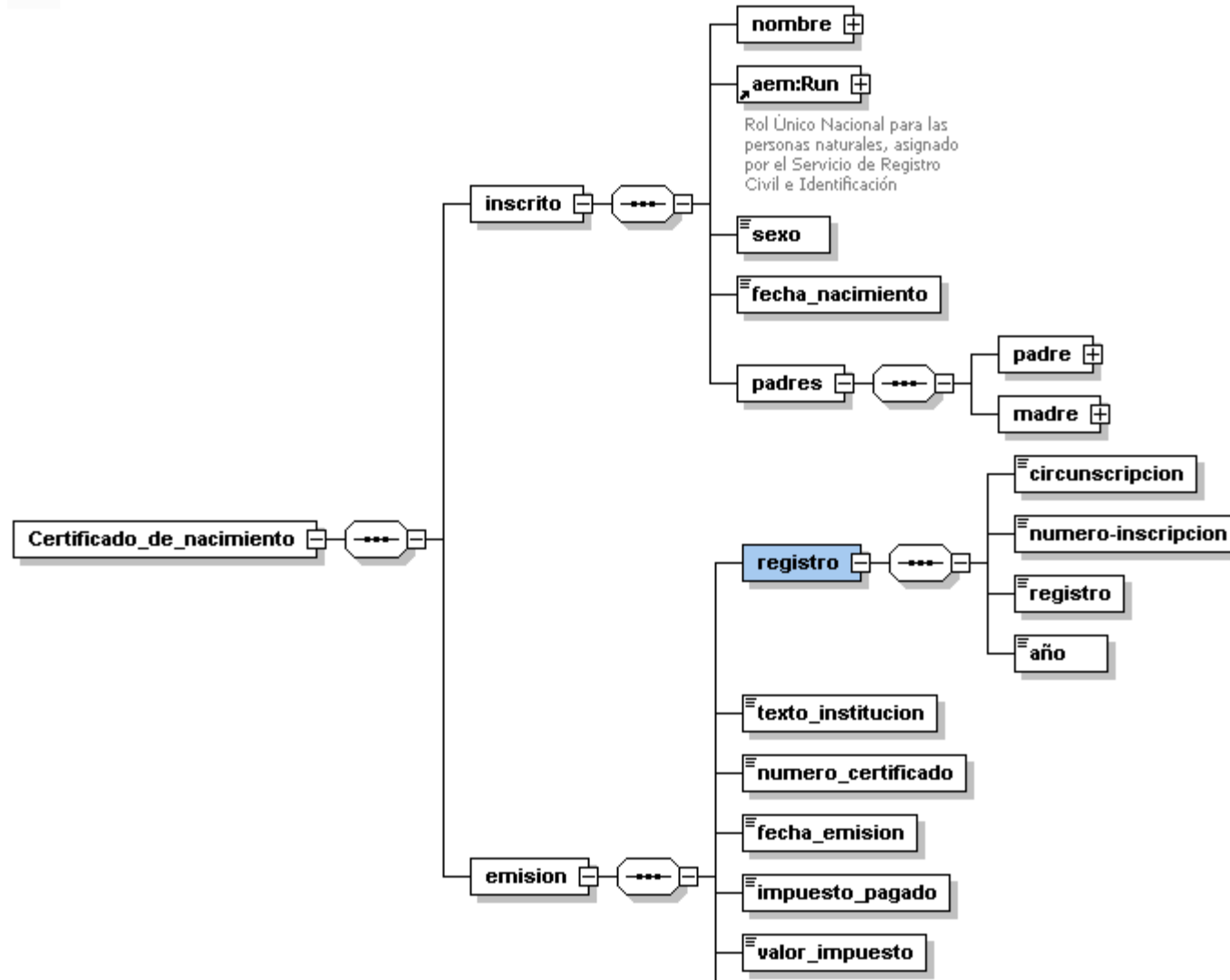
Explicación

Los esquemas XML deben definir el significado de cada uno de los elementos y atributos que serán usados en la validación de los documentos electrónicos.

Por ejemplo, si un documento electrónico utiliza un atributo o elemento, entonces el significado de ese ítem debe estar presente en el esquema XML asociado.

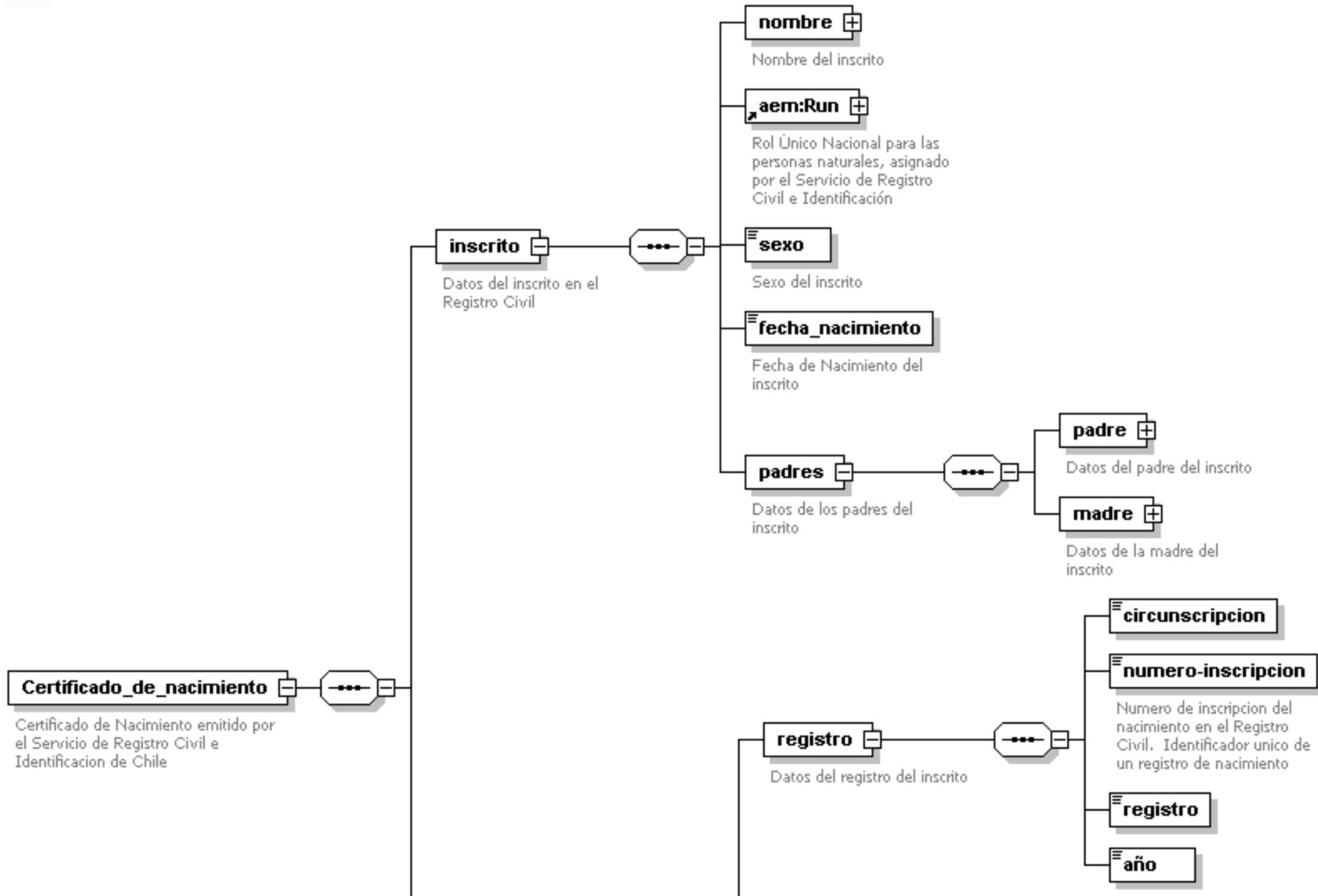
Complejidad semántica

Esquema inicial



Complejidad semántica

Esquema cambiado



Restricciones de los esquemas XML

Recomendación

Los esquemas XML DEBERÍAN restringir los valores de los elementos y atributos que serán usados en los documentos electrónicos.

Restricciones de los esquemas XML

Explicación

Cuando se diseña un esquema XML se deberían delimitar los valores de los elementos y atributos, a un conjunto de valores válidos.

Esto debe hacerse con la finalidad de que, tanto los elementos como los atributos, puedan ser manejados adecuadamente por una aplicación computacional.

Los esquemas XML establecen un contrato que permite, al creador y al receptor de un documento electrónico, verificar que la instancia del documento electrónico cumple con el contrato.

Restricciones

```
<xs:element name="año" type="xs:gYear"/>
```

```
<xs:element name="registro">
```

```
  <xs:simpleType>
```

```
    <xs:restriction base="xs:string">
```

```
      <xs:maxLength value="3"/>
```

```
      <xs:minLength value="1"/>
```

```
    </xs:restriction>
```

```
  </xs:simpleType>
```

```
</xs:element>
```

```
<xs:element name="numero_certificado">
```

```
  <xs:simpleType>
```

```
    <xs:restriction base="xs:integer">
```

```
      <xs:maxInclusive value="99999999"/>
```

```
    </xs:restriction>
```

```
  </xs:simpleType>
```

```
<xs:element name="fecha_emision" type="xs:date"/>
```

```
<xs:element name="nombre_funcionario_autorizado" type="aem:nameType"/>
```

Reusabilidad y redundancia de los esquemas XML

Recomendación

Los documentos de esquemas XML DEBERÍAN ser diseñados de manera tal que sus esquemas XML puedan ser reutilizados por otros documentos de esquemas XML.

Reusabilidad y redundancia de los esquemas XML

Explicación

Los documentos de esquemas XML deberían ser diseñados de tal forma que potencien la reutilización y eliminen la redundancia de esquemas XML.

Por ejemplo, un documento de esquemas XML podría importar o incluir otros documentos de esquemas XML, con la finalidad de reutilizar los esquemas allí contenidos.

Extensibilidad de los esquemas XML

Recomendación

Los esquemas XML DEBERÍAN ser diseñados para ser extensibles en el tiempo.

Extensibilidad de los esquemas XML

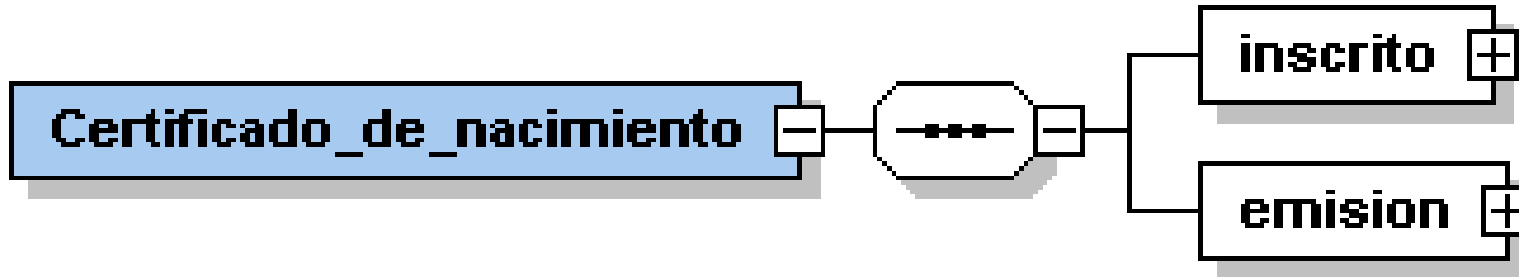
Explicación

Los esquemas XML deberían ser fácilmente extensibles en el tiempo, a fin de permitir la incorporación de nuevos componentes XML.

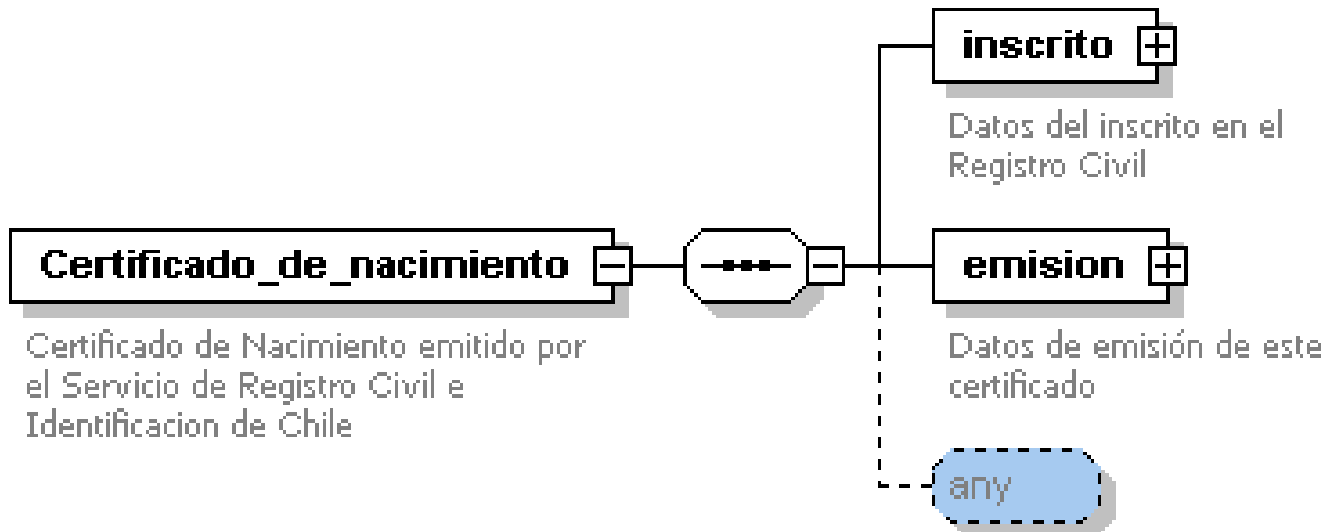
Los puntos de extensión pueden ser explícitos, si el esquema XML está diseñado para evolucionar.

Extensibilidad

Esquema inicial

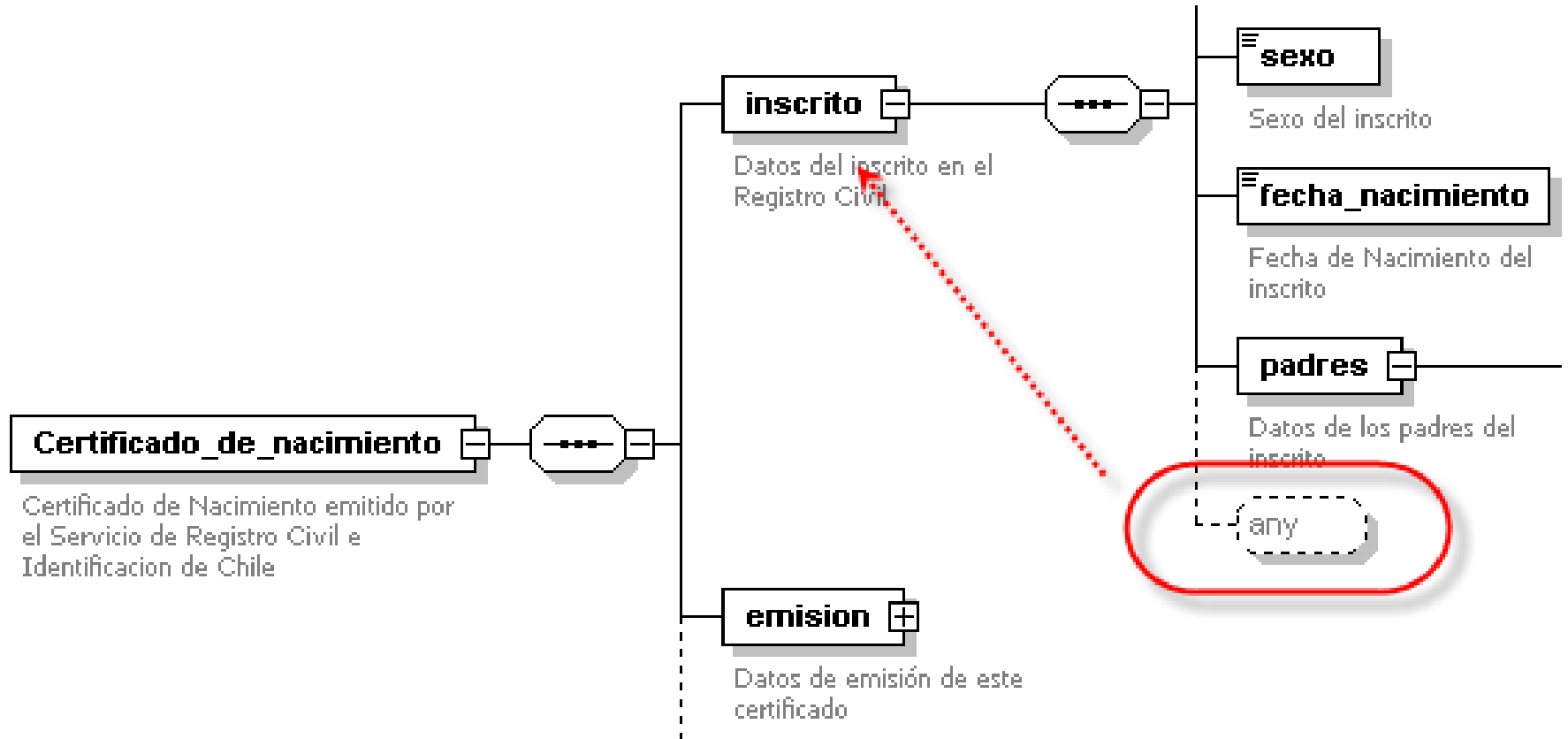


Esquema cambiado



Extensibilidad explícita

Esquema cambiado



Definiciones Generales

Idioma oficial

Recomendación

Los esquemas XML DEBEN ser diseñados con nombres de elementos, atributos y tipos de datos que deben estar compuestos de palabras en el idioma Español.

Idioma oficial

Explicación

Los nombres de los elementos, atributos y tipos de datos usados en los esquemas XML, deben utilizar palabras del idioma Español, a fin de facilitar su identificación y comprensión por parte del usuario.

Los caracteres permitidos son:

- a-z y A-Z.
- vocales acentuadas,
- la letra ' n ' con virgulilla o tilde (ñ y Ñ)
- la vocal ' u ' con diéresis (ü y Ü).

Sin embargo, se recomienda no utilizar estos últimos, dentro de lo posible.

Codificación de caracteres

Recomendación

La codificación de caracteres a usar en los esquemas XML
DEBERÍA ser UTF-8.

Codificación de caracteres

Explicación

El artículo 9 del Decreto Supremo 81 establece que para la codificación de caracteres se utilizará preferentemente UTF-8.

En caso de utilizar otra codificación, se recomienda el uso de un servicio de conversión a UNICODE.

Nombramiento de elementos y atributos

Recomendación

Los nombres de los elementos en los esquemas XML DEBERÍAN ser escritos de acuerdo a la convención Upper Camel Case.

Los nombres de los atributos DEBERÍAN ser escritos siguiendo la convención Lower Camel Case.

Nombramiento de elementos y atributos

Explicación

Camel Case es un estilo de escritura que se aplica a frases o palabras compuestas. Esta convención es utilizada por varios lenguajes de programación y guías de especificaciones en la industria y el gobierno. Los nombres de los elementos en los esquemas XML deberían ser escritos de acuerdo a la convención Upper Camel Case, esto es, palabras compuestas sin espacios y poniendo en mayúscula la primera letra de cada palabra (por ejemplo: UpperCamelCase).

En cuanto a los nombres de los atributos, estos deberían ser escritos siguiendo la convención Lower Camel Case, esto es, palabras compuestas sin espacios, donde la primera letra de la primera palabra debe ir con minúscula, y poniendo en mayúscula la primera letra a partir de la segunda palabra (por ejemplo: lowerCamelCase).

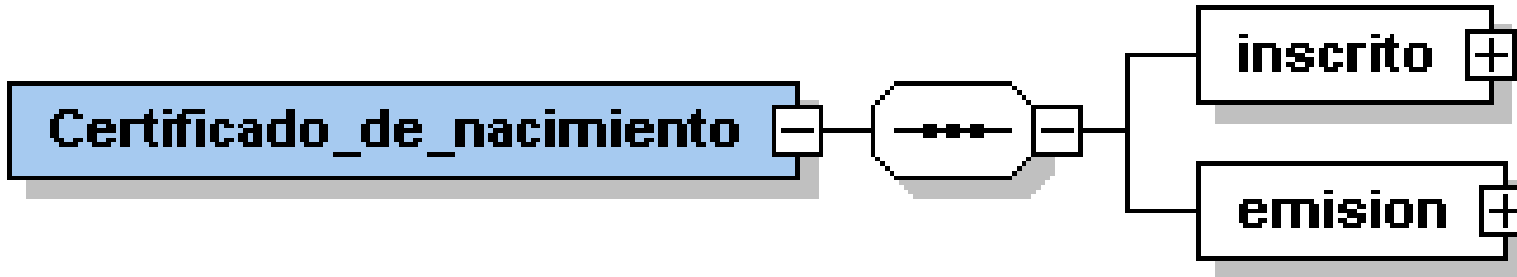
Tanto los elementos como los atributos deberían ser nombrados en forma singular, a no ser que el concepto sea en sí mismo plural.

Símbolos tales como: punto, guiones (alto y bajo) u otra clase de separadores (por ejemplo, espacios) no deben usarse en los elementos y atributos. Esto mismo vale para aquellos caracteres no permitidos en la especificación XML del consorcio W3C.

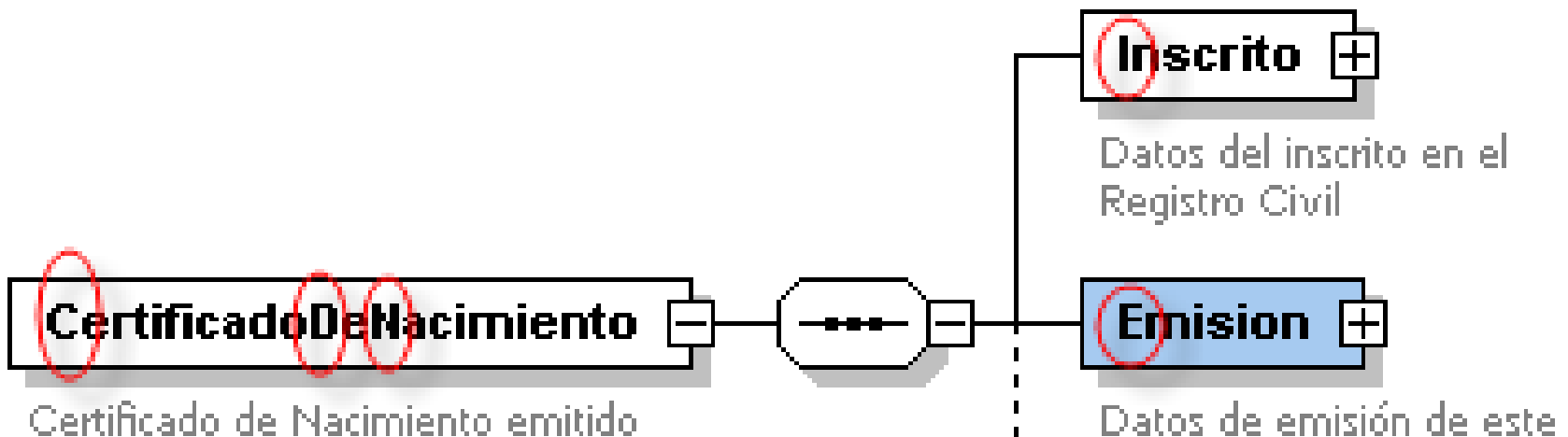
Los acrónimos, abreviaturas u otro tipo de truncado de palabras no deberían ser usados, a menos que estos sean parte de un registro central. En ese caso, si se utilizan al inicio de la declaración de un atributo, entonces deben aparecer en minúsculas; en el resto de los casos siempre deben ser escritos en mayúsculas. Aquellas abreviaturas o acrónimos que se usen en elementos o tipos de datos, deben permanecer en mayúsculas.

Nombramiento en modelo

Esquema inicial

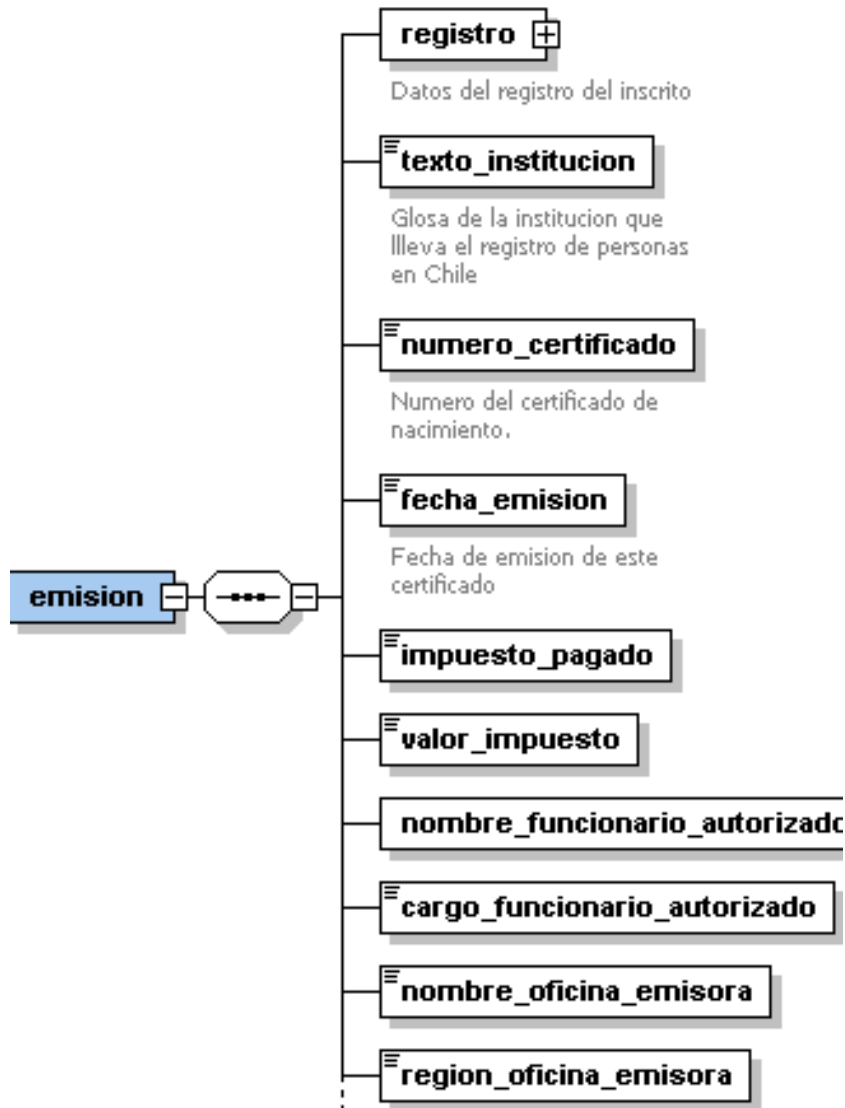


Esquema cambiado

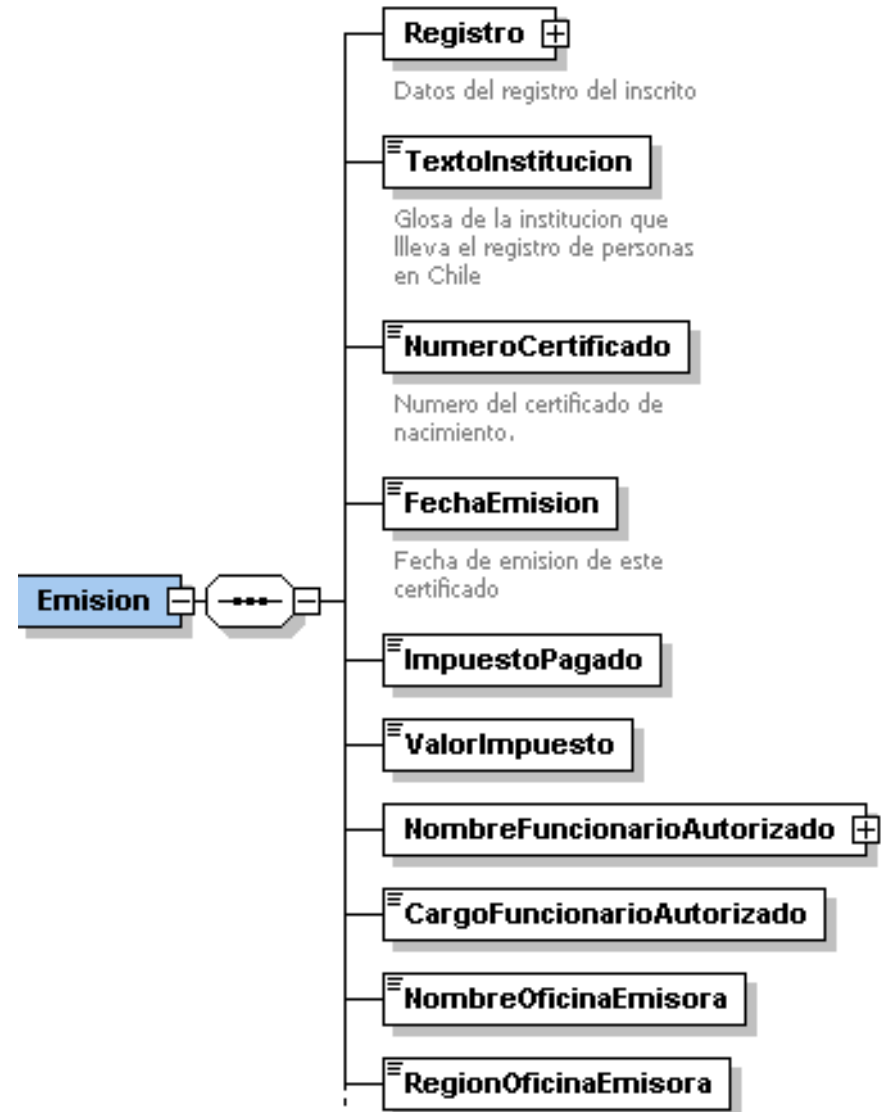


Nombramiento en modelo

Esquema inicial



Esquema cambiado



Nombramiento de tipos

Recomendación

Los nombres de los tipos de datos (tanto simples como complejos) en los esquemas XML, DEBERÍAN ser escritos de acuerdo a la convención Upper Camel Case, anexando al final del nombre la palabra "Type".

Espacio de Nombres (namespaces)

Definición de namespaces

Recomendación

Es RECOMENDADO que cada organización defina su propio espacio de nombres, según la especificación de namespace de la W3C.

Definición de namespaces

Explicación

El uso de múltiples vocabularios trae problemas de reconocimiento y colisión de nombres de elementos y atributos.

Por lo tanto, el tener diferentes espacios de nombres permite desarrollar en forma modular y distribuida los esquemas. Esto da flexibilidad a la definición de esquemas.

Namespace

Esquema inicial

```
<?xml version="1.0" encoding="UTF-8"?>|  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:aem="http://valida.aem.gob.cl" elementFormDefault="qualified" attributeFormDefault="unqualified">  
  <xs:import namespace="http://valida.aem.gob.cl" schemaLocation="run.xsd"/>
```

Esquema cambiado

```
<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns="http://registrocivil.cl/tramites/2009/certificados" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:aem="http://valida.aem.gob.cl" targetNamespace="http://registrocivil.cl/tramites/2009/certificados" elementFormDefault="qualified" attributeFormDefault="unqualified">  
  <xs:import namespace="http://valida.aem.gob.cl" schemaLocation="run.xsd"/>
```

Namespace de XML Schema

Recomendación

El namespace de XML Schema de la W3C DEBE ser calificado con un prefijo de xsd o xs.

Uso del default namespace y el target namespace

Recomendación

Si un documento de esquemas XML define un target namespace, el default namespace para el documento DEBE tener el mismo valor que el target namespace.

Uso de atributos `elementFormDefault` y `attributeFormDefault`

Recomendación

El atributo `elementFormDefault` DEBE estar definido como `qualified`

y el atributo `attributeFormDefault` DEBE estar definido como `unqualified`.

Uso de atributos `elementFormDefault` y `attributeFormDefault`

Explicación

Esto asegura que un diseñador que desea leer o reutilizar un documento de esquemas XML, pueda confiar en los namespaces y en los prefijos visibles, en lugar de tener que hacer un seguimiento de la estructura interna del documento de esquemas XML.

Definición de componentes genéricos

Recomendación

Los componentes genéricos PODRÍAN no tener asignado un namespace.

Definición de componentes particulares

Recomendación

Los componentes particulares de una organización DEBEN tener asignado un namespace definido por dicha organización.

Estructura del Esquema

Tipos de datos y declaración de elementos

Recomendación

Un componente XML DEBE ser definido como un tipo de dato si:

- el componente XML tendrá diferentes nombres en diferentes contextos; o
- el componente XML podría ser usado para definir otros componentes XML (derivación de tipos de datos).

Un componente DEBERIA ser definido como un elemento si:

- el nombre del componente XML será siempre el mismo; o
- el componente XML no será usado para definir nuevos componentes XML.

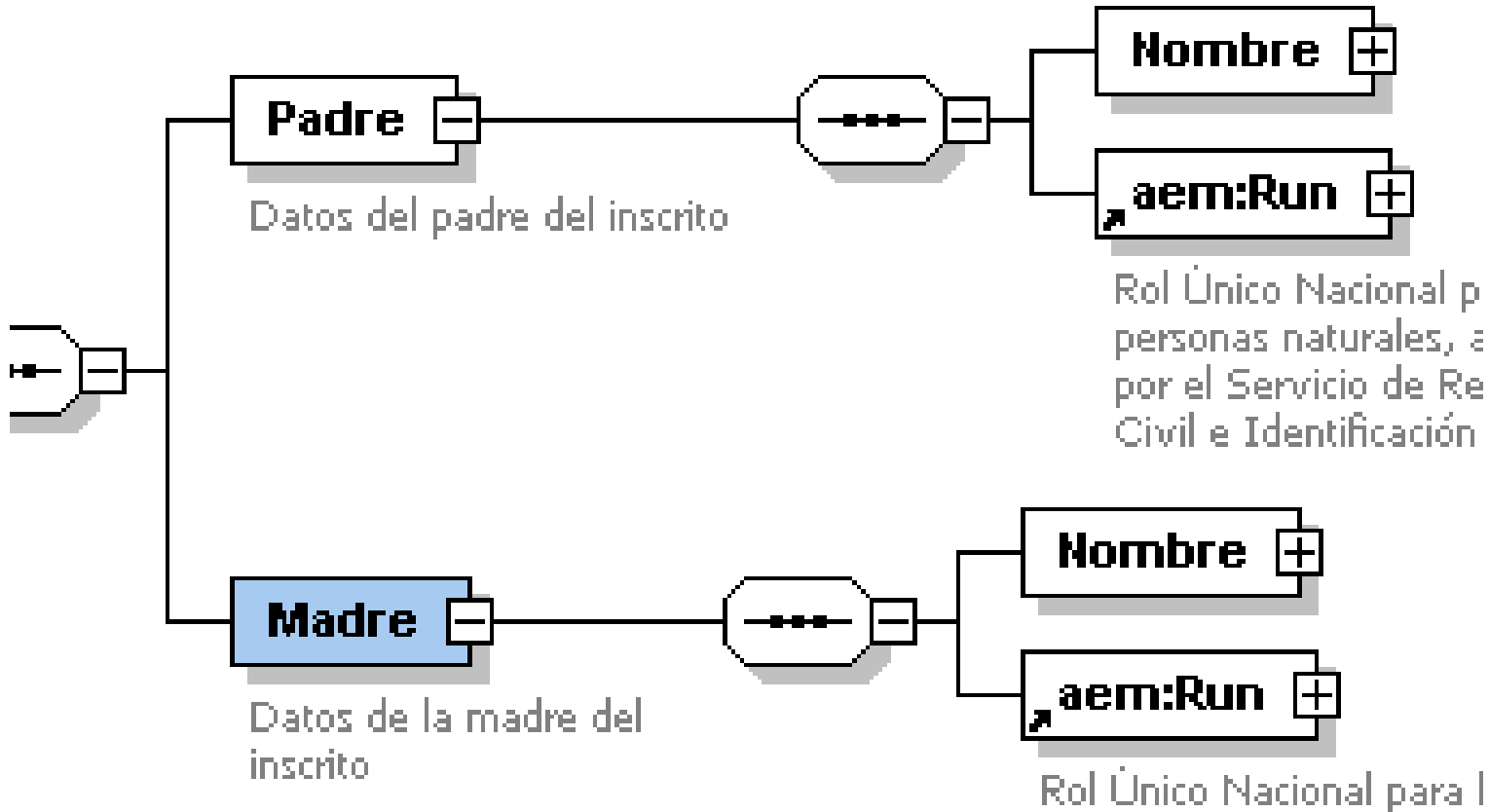
Es RECOMENDABLE elaborar un diccionario de datos con nombres de componentes que tienen una semántica general conocida (por ejemplo RUN).

Otro motivo para definir un componente como un tipo de dato, radica en que éste podría ser usado para definir nuevos componentes (por ejemplo mediante derivación).

En el caso de los esquemas XML arquitecturales, existen situaciones en las cuales sería apropiado definir un componente como un elemento y a la vez como un tipo de dato. De esta forma el elemento está disponible para ser usado con una semántica conocida establecida, y el tipo de dato estaría disponible para ser heredado y modificado apropiadamente.

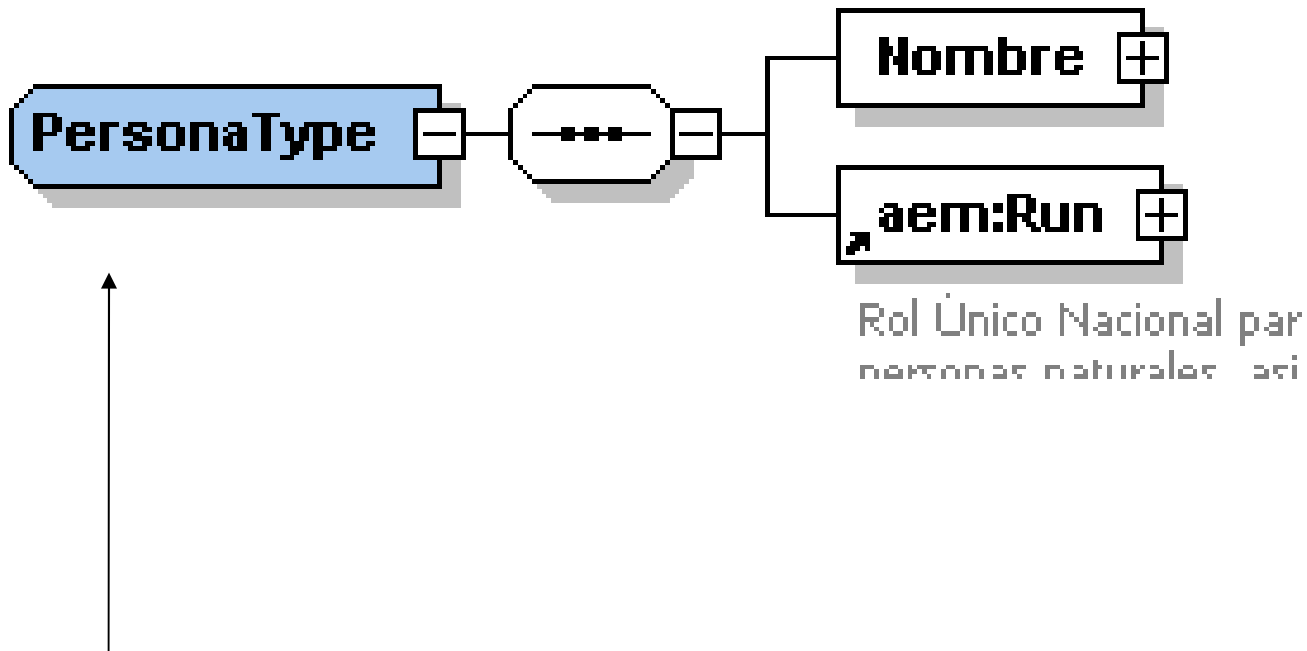
Tipos de datos

Esquema inicial



Tipos de datos

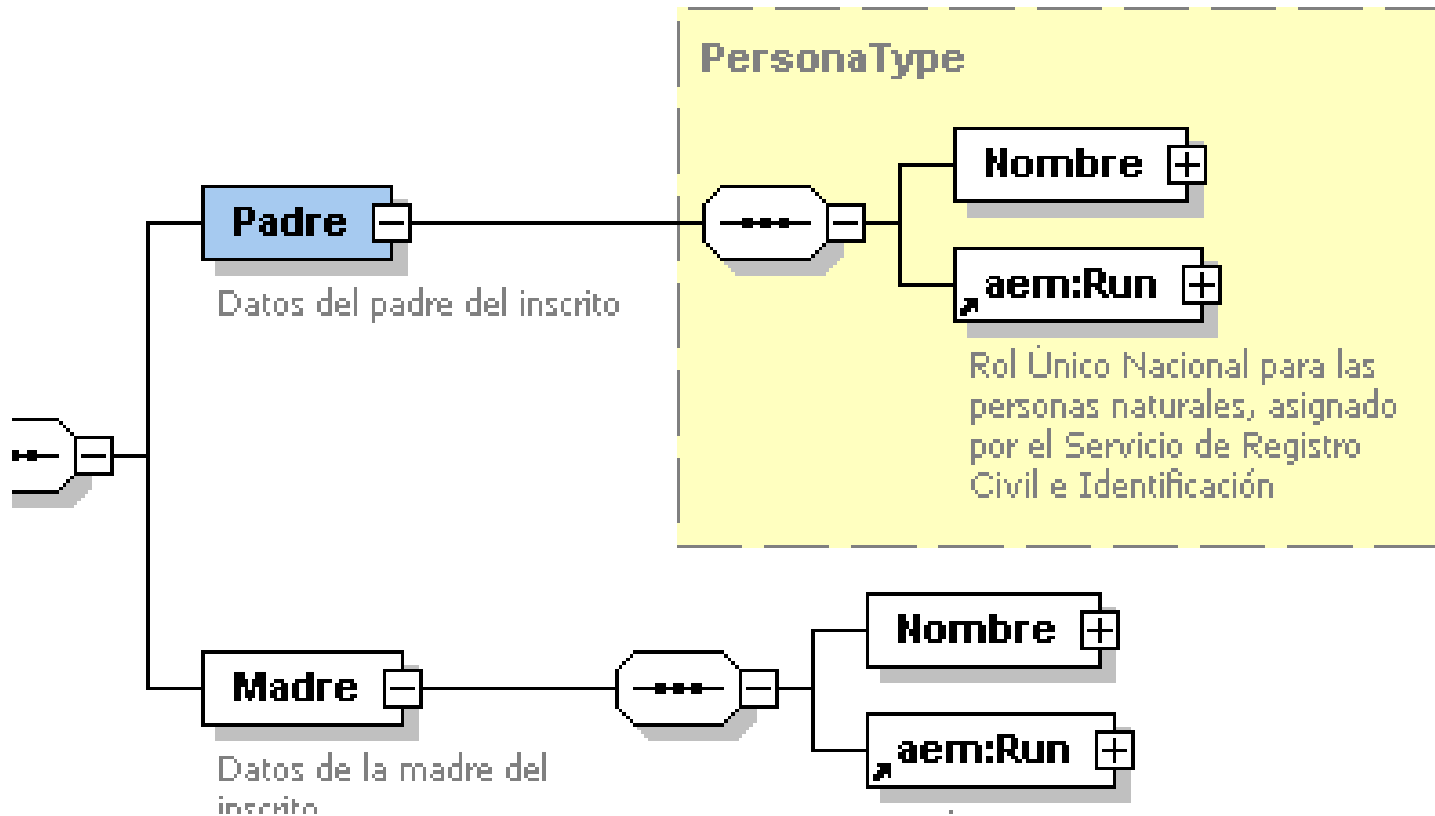
Definición del tipo de datos



Nombramiento Upper Camel Case + "Type"

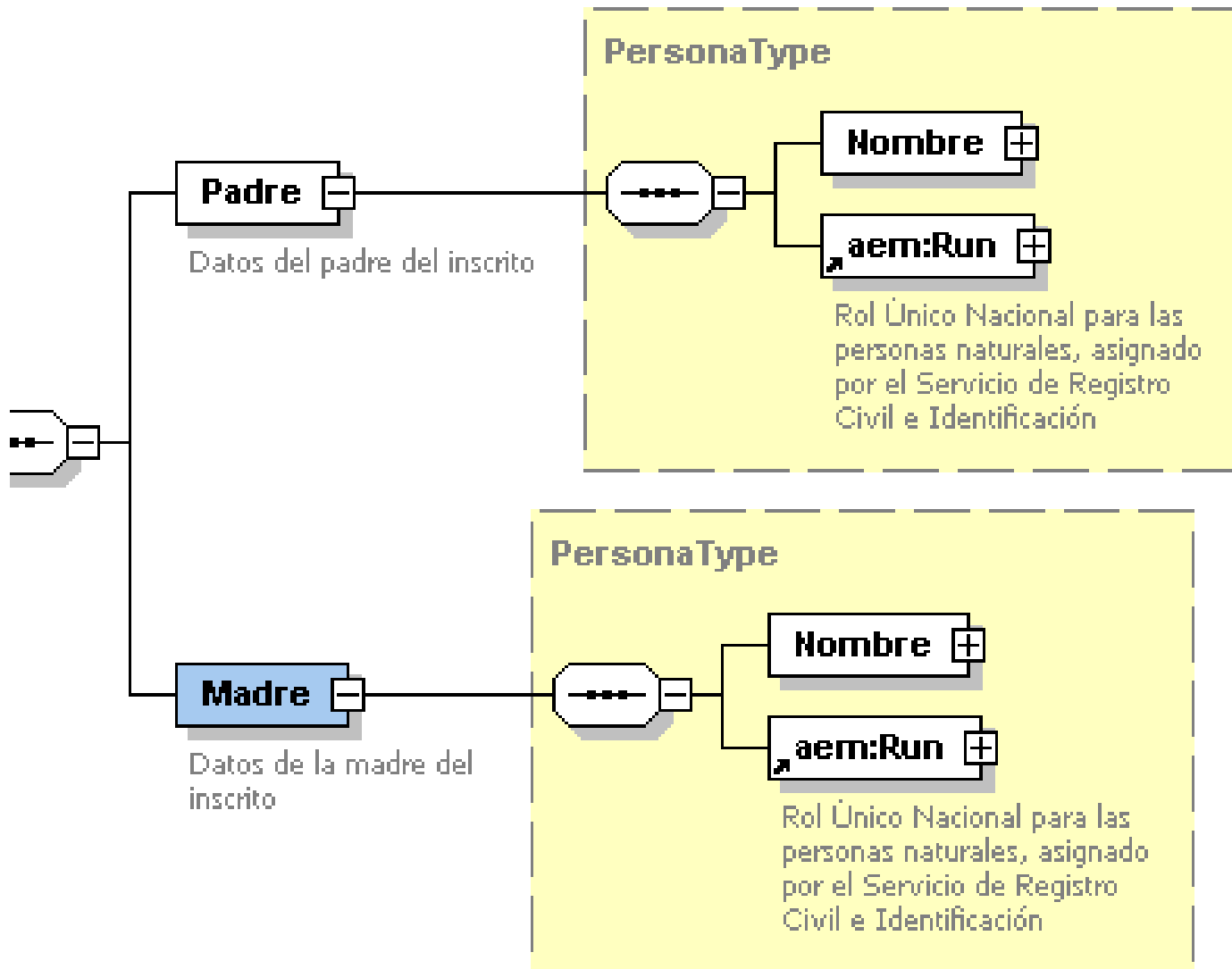
Tipos de datos

Asignación del tipo de datos



Tipos de datos

Reutilización del tipo de datos



Definición de atributos

Recomendación

Los atributos DEBEN ser usados sólo para agregar meta-datos que ayuden a clarificar el contenido de un elemento.

- Los atributos DEBERÍAN ser usados para contener unidades de información que no serán extendidas o divididas.
- Un atributo NO DEBE ser usado para calificar otros atributos.
- El valor de un atributo DEBE ser corto.

Definiciones globales

Recomendación

Un componente XML DEBERÍA ser definido globalmente en un documento de esquemas XML si el componente XML:

- será reutilizado dentro del documento de esquemas XML;
- será reutilizado en otros documentos de esquemas XML (al incluir / importar el documento de esquemas XML que contiene la definición del componente); o
- será usado como un elemento raíz en un documento instancia XML.

Atributos globales versus atributos locales

Recomendación

Los atributos DEBERÍAN ser definidos localmente.

Uso de los atributos default y fixed

Recomendación

El atributo default NO DEBERÍA ser usado para agregar información relevante a elementos o atributos en los documentos instancia XML.

El atributo fixed NO DEBERÍA ser usado para agregar información relevante a elementos o atributos en los documentos instancia XML, con la posible excepción de atributos obligatorios.

Definición de componentes obligatorios

Recomendación

Los componentes obligatorios DEBERÍAN aparecer en todo documento instancia XML y no deben permitir contenido vacío.

Definición de elementos opcionales

Recomendación

Los componentes XML definidos como opcionales NO DEBERÍAN permitir contenido vacío.

Definición de elementos opcionales

Explicación

Si un esquema XML define un elemento como opcional, su definición debería asegurar que:

- (a) si el elemento tiene contenido, deberá ocurrir en el documento instancia XML;
- (b) si el elemento no presenta contenido, no deberá ocurrir en el documento instancia XML.

Representación de componentes alternativos

Recomendación

Los componentes alternativos DEBEN representarse a través de valores, en lugar de usar la presencia o ausencia de los componentes.

Representación de componentes alternativos

Explicación

La ausencia de un elemento o atributo que forma parte de un conjunto de alternativas debe indicarse a través de un valor especial en su contenido. Por ejemplo, se podría usar el valor de “0” cuando el contenido del elemento debiera ser un número entero positivo. Otra opción es el uso de un atributo especial (por ejemplo, un atributo “presente” con valores “si” o “no”).

Esta indicación explícita permite una mejor comprensión de los documentos instancia XML.

Componentes Alternativos

```
<xs:element name="ImpuestoPagado">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:enumeration value="si"/>  
      <xs:enumeration value="no"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

Texto y códigos

Recomendación

Un código NO DEBERÍA ser usado en lugar de texto sin alguna indicación de su significado.

Uso de elementos con contenido mixto

Recomendación

Si un componente es definido orientado a modelar sus datos, su esquema XML NO DEBERÍA permitir un contenido mixto.

Redefinición de componentes de esquemas XML

Recomendación

La redefinición de componentes de esquemas XML (a través del elemento `xsd:redefine`) DEBERÍA evitarse.

Redefinición de componentes de esquemas XML

Explicación

Esta recomendación está orientada tanto a evitar efectos no deseados debido a la reutilización de los componentes, como a incrementar la claridad y legibilidad de los documentos de esquemas XML.

Importación de documentos de esquemas XML

Recomendación

El elemento `xsd:import` DEBERÁ ser usado con el atributo `namespace`.

1. Uso de derivación de tipos

Recomendación

La herencia de tipos DEBERÍA mantener el estándar de datos subyacente.

Uso de derivación de tipos

Explicación

La herencia de tipos (a través de los elementos `xsd:extension` o `xsd:restriction`) debería usarse cuidadosamente, para mantener la consistencia entre las definiciones originales y aquellas creadas producto de la herencia.

Por ejemplo, si el tipo base define un conjunto estándar de valores permitidos, entonces un tipo derivado debería permitir un subconjunto de los valores definidos por el tipo base.

Referencias absolutas y relativas

Recomendación

Si dos o más documentos de esquemas XML están estrechamente relacionados (respecto a la URI donde están declarados), las declaraciones `xsd:include` y `xsd:import` DEBERÍAN usar referencias relativas.

Cuando los documentos de esquemas XML no tienen una relación cercana, se DEBERÍAN usar referencias absolutas.

Sólo DEBEN ser referenciados esquemas que están en el Administrador de Esquemas y Metadatos (AEM) en estado de “Uso Preliminar” o “Uso Consolidado”.

Metadatos y Documentación de Esquemas

Metadatos estándar para documentos de esquemas XML

Recomendación

Un documento de esquemas XML DEBERÍA incluir metadatos estándar Dublin Core.

Metadatos estándar para documentos de esquemas XML

Explicación

Los metadatos para un documento de esquemas XML deberían estar basados en el vocabulario definido por Dublin Core.

Dublin Core es un modelo de metadatos elaborado y auspiciado por la DCMI (Dublin Core Metadata Initiative), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de metadatos. Dublin Core se define en la norma ISO 15836 del año 2003.

Metadatos estándar

```

<xs:element name="CertificadoDeNacimiento">
  <xs:annotation>
    <xs:documentation>Certificado de Nacimiento emitido por el Servicio de Registro Civil e Identificación de Chile</
xs:documentation>
    <xs:appinfo>
      <dc:title>W3C Schema de un certificado de nacimiento</dc:title>
      <dc:subject>Certificado de nacimiento</dc:subject>
      <dc:contributor Href="mailto:jperez@registrocivil.cl">Juan Perez, División Informatica, Servicio de Registro Civil e
Identificación, Ministerio de Justicia</dc:contributor>
      <dc:contributor>Servicio de Registro Civil e Identificación, Ministerio de Justicia</dc:contributor>
      <dc:creator>John Doe, Servicio de Registro Civil e Identificación, Ministerio de Justicia</dc:creator>
      <dc:date>2009-08-01</dc:date>
      <dc:description>W3C Schema de un certificado de nacimiento para uso en esquemas del Gobierno de Chile</
dc:description>
      <dc:identifier>http://serviciocivil.cl/xsd/CertificadoNacimiento.xsd</dc:identifier>
      <dc:format>text/xml</dc:format>
      <dcterms:modified>2009-08-04</dcterms:modified>
      <dcterms:created>2009-08-01</dcterms:created>
      <dc:type>XML Schema</dc:type>
      <dc:language>es</dc:language>
    </xs:appinfo>
  </xs:annotation>

```

Versionamiento de documentos de esquemas XML

Recomendación

Los documentos de esquemas XML DEBEN indicar su versión utilizando el atributo versión del elemento `xsd:schema`.

Versionamiento de documentos de esquemas XML

Explicación

Indicar la versión de un documento de esquemas XML es una buena práctica y ayuda a prevenir el uso de versiones incorrectas.

Versionamiento de documentos de esquemas XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns="http://registrocivil.cl/tramites/2009/certificados" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:aem="http://valida.aem.gob.cl" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/" targetNamespace="http://registrocivil.cl/tramites/2009/certificados" elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">
```

Indicación de la versión de los documentos de esquemas XML en los documentos instancia XML

Recomendación

Los esquemas XML DEBEN requerir que su versión se indique en sus instancias. Esta versión DEBERÍA ser indicada por medio de uno de los siguientes métodos:

- Usando namespace versionado, definido como el namespace en el esquema.
- Obligando al elemento (usualmente el elemento raíz) a incluir la versión usando el atributo SchemaVersion. Este atributo PUEDE usar un valor fixed junto al atributo required.

Indicación de la versión de los documentos de esquemas XML en los documentos instancia XML

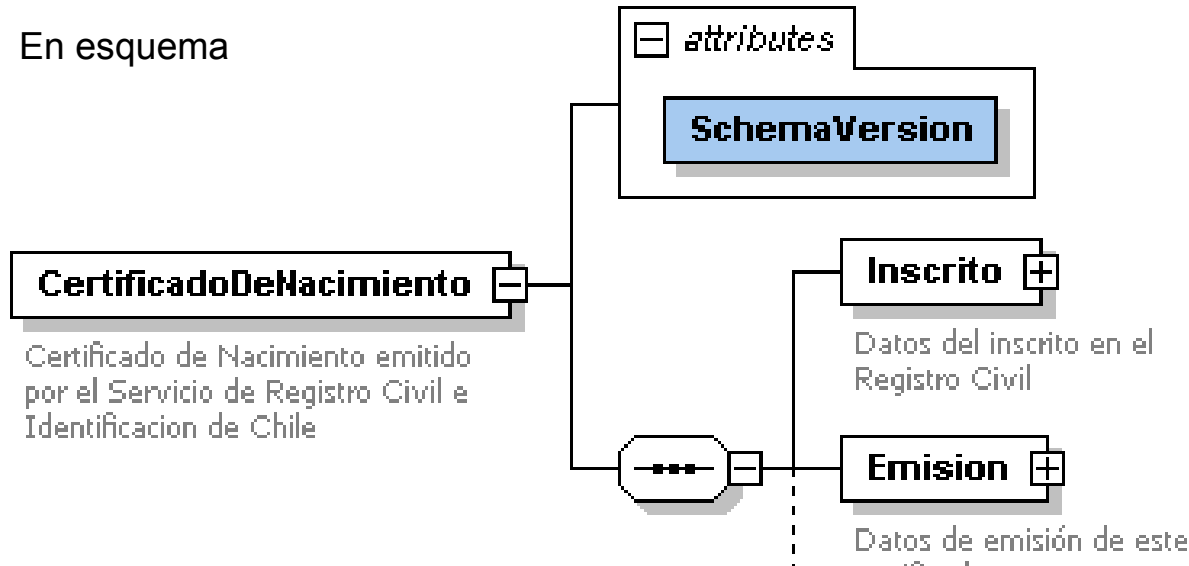
Explicación

Varios documentos instancia XML pudieran estar almacenados sin tener la versión del documento de esquemas XML que los valide.

Indicar la versión del documento de esquemas XML usado en un documento instancia XML permite identificar qué versión se está utilizando dentro del conjunto de documentos de esquemas XML.

Versionamiento del esquema en la instancias XML- SchemaVersion

En esquema



```
<xs:attribute name="SchemaVersion" type="xs:NMTOKEN" use="required" fixed="1.0"/>
```

En instancia XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<CertificadoDeNacimiento SchemaVersion="1.0"
```

Versionamiento del esquema en la instancias XML- Namespace

En esquema

```
<xs:schema xmlns="http://registrocivil.cl/tramites/certificados/cn01"
```

En instancia XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<CertificadoDeNacimiento SchemaVersion="1.0" xmlns="http://registrocivil.cl/tramites/certificados/cn01"
```

Versionamiento del esquema en la instancias XML- version 2.0

En esquema

```
<xs:schema xmlns="http://registrocivil.cl/tramites/certificados/cn02"
```

```
<xs:attribute name="SchemaVersion" type="xs:NMTOKEN" use="required" fixed="2.0"/>
```

En instancia XML

```
<CertificadoDeNacimiento SchemaVersion="2.0" xmlns="http://registrocivil.cl/tramites/certificados/cn02"
```

Uso del atributo ID en el elemento xsd:schema

Recomendación

Los documentos de esquemas XML DEBERÍAN ser identificados con el atributo id.

El id DEBERÍA ser único en el targetNamespace.

Uso del atributo ID en el elemento xsd:schema

Explicación

Es una buena práctica proporcionar un identificador de un documento de esquemas XML. Para ello, el atributo id es más genérico y por lo tanto debería utilizarse con tal fin.

Por ser el atributo id un identificador, éste debe ser único; en caso contrario pierde gran parte de su potencial utilidad.

Uso del atributo id en esquema

En esquema

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema id="MJ-RC-CN01" xmlns="http://regis
```

Uso de referencia para namespaces

Recomendación

La información sobre un namespace DEBERÍA ser accesible públicamente en un lugar que tenga permanencia en el tiempo.

Si la URI (Uniform Resource Identifier) del namespace es una URL (Uniform Resource Locator), entonces la documentación DEBERÍA estar en la misma localización (mismo URL).

Uso de referencia para namespaces

Explicación

Es útil a menudo poder hacer referencia a la información sobre un namespace. Dicha información podría incluir una descripción de la finalidad del namespace y referencias a recursos relacionados, tales como especificaciones y esquemas.

El consorcio de la W3C hace esto para sus propios namespaces. Ejemplos de ello se pueden encontrar accediendo al namespace de la W3C a través de un navegador.

1. Asignación de nombres a archivos de documentos de esquemas XML

Recomendación

Los documentos de esquemas XML DEBEN ser guardados en un archivo cuyo nombre identifique su versión.

Asignación de nombres a archivos de documentos de esquemas XML

Explicación

Los documentos de esquemas XML deben indicar en su nombre de archivo, la versión del documento en forma completa. Para ello, se debe seguir el siguiente formato para nombrar a dichos archivos:

NombreDelArchivo-vm-n.xsd.

Primero se debe colocar el nombre que identifica al documento de esquemas XML y añadir al final un guión

“-“. Segundo, se debe añadir el carácter “v”, el número mayor de versión y añadir al final un guión “- “. Finalmente, se debe incorporar el número menor de versión y la extensión del archivo .xsd.

Si el documento de esquemas XML va a ser cambiado a lo largo del tiempo, el nombre del archivo debe incorporar los cuatro dígitos del año. Para ello, se debe seguir el siguiente formato:

NombreDelArchivo-Año-vm-n.xsd

Incorporación de comentarios en documentos de esquemas XML

Recomendación

Los documentos de esquemas XML DEBEN documentarse utilizando el elemento documentation.

```
<xs:element name="CertificadoDeNacimiento">
  <xs:annotation>
    <xs:documentation>Certificado de Nacimiento emitido por el Servicio de Registro Civil e Identificación de Chile</
xs:documentation>
    <xs:appinfo>
      <dc:title>W3C Schema de un certificado de nacimiento</dc:title>
      <dc:subject>Certificado de nacimiento</dc:subject>
      <dc:contributor Href="mailto:jperez@registrocivil.cl">Juan Perez, División Informatica, Servicio de Registro Civil e
Identificación, Ministerio de Justicia</dc:contributor>
      <dc:contributor>Servicio de Registro Civil e Identificación, Ministerio de Justicia</dc:contributor>
      <dc:creator>John Doe, Servicio de Registro Civil e Identificación, Ministerio de Justicia</dc:creator>
      <dc:date>2009-08-01</dc:date>
      <dc:description>W3C Schema de un certificado de nacimiento para uso en esquemas del Gobierno de Chile</
dc:description>
      <dc:identifier>http://serviciocivil.cl/xsd/CertificadoNacimiento.xsd</dc:identifier>
      <dc:format>text/xml</dc:format>
      <dcterms:modified>2009-08-04</dcterms:modified>
      <dcterms:created>2009-08-01</dcterms:created>
      <dc:type>XML Schema</dc:type>
      <dc:language>es</dc:language>
    </xs:appinfo>
  </xs:annotation>
```

Incorporación de comentarios en documentos de esquemas XML

Explicación

El elemento `documentation` es un sub-elemento de elemento `annotation`, y existe para ayudar a documentar los esquemas XML. La ventaja de utilizar este elemento, en lugar de poner texto en comentarios XML, es que el contenido puede ser procesado de manera más fácil con una hoja de estilo (stylesheet). Por ejemplo, para preparar la documentación de usuario. La información en comentarios XML no es tenida en cuenta por un procesador de XML, por ejemplo XSLT.