El XSD empieza con esto SIEMPRE!!

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:nombre\_primer\_elemento\_del\_documento\_linkado>

<xs:complexType>

El xml enlazado al XSD tiene que tener esto!!

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

<nombre.xsd **xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**

**xsi:noNamespaceSchemaLocation="./nombre.xsd">**

Tipos de elemento

Simple: solo puede tener datos de tipo carácter

Ej:

< xs: element name=”nombre” type=”tipodeDato”(entero,long,double,string,char)/>

Complejo: puede contener atributos, elementos hijo y contenido mixto,.

Los complejos tienen se definen

<xs:element name=Biblioteca>

<xs:complexType> --- al tener mas cosas que solo datos tipo carácter pues es complejo

<xs:sequence>--aquí se listan los elementos uqe

lo componen equivale a ( elemento , elemento, elemento del DTD)

… permite definir minOccurs,maxOccurs

</xs:sequence>

<xs:choice> --equivale a la barra OR | |en DTD, permite elegir un minimo y un maximo

<xs:all> define un grupo no ordenado de elementos, en este tipo, los elementos que lo componen pueden aparecer 0 o 1 vez. == ? de DTD

</xs:complexType>

<xs:element name=Libro type=”xs:string”> (si no se pone minOccurs o maxOccurs, se utiliza el valor por defecto para los dos, que es 1.

**Tipos de datos**

ATRIBUTOS

Los atributos de un elemento se definen como

<xs:attribute name=”nombre\_atributo” type=”tipo\_atributo” >

Han de ser siempre de tipo simple, han de declararse justo despues de la declaracion de los subelementos

No tienen cardinalidad , esto es… un atributo SOLO puede aparecer una vez dentro del elemento

Es opcional por defecto

Tres atributos para restringir el valor del atributo:USE ,DEFAULT, FIXED

-use: puede tomar tres valores

-required: atributo obligatorio

-optional: puede o no aparecer (por defecto)

-prohibited: no puede aparecer en el elemento

-default:

-por defecto. Puede aparecer o no

-fixed:

-fijo. Puede aparecer o no, pero en caso de aparecer, solo puede tomar el valor indicado

Exiten 44 tipos simples divididos en dos grupos

\_primitivos

-Derivados

Además un usuario puede definir un nuevo tipo simple mediante la expresion

<xs:simpleType name=” nombre” > … </xs:simpleType> -- habria que ver un ejemplo

Los tipos de datos se pueden clasificar como

Atomico/lista/union

Built-in ó derivado por el usuario

Tambien se pueden derivar

-por restriccion <xs:restriction base=” tipodato\_delque\_se\_deriva“>..</xs:restriction>

-por valor: restringe el valor del tipo de dato. Mediante patrones <xs:pattern value =” patron” /> (es la de perl la sintaxis de los patrones)

Tambien se puede usar length,maxLength , minLength, totalDigits,fractionDigits (maximo numero de digitos decimales), minInclusive, maxExclusive

**Ej de un tipo derivado a nivel global:**

**Tiene que estar fuera de todos los elementos, dentro del schema!!**

<xs:simpleType name=”nombreUser”>

<xs:restriction base=”xs:string”>

<xs:maxLength value=”12”>

</xs:restriction>

</xs:simpleType>

**Ej de un tipo derivado a nivel local:**

<xs:element name=”nombre”>

<xs:simpleType>

<xs:restriction base=”xs:string”>

<xs:maxLength value=”32”>

</xs:restriction>

</xs:simpleType>

</xs:element>

-por enmueracion: Se disponen los valores que puede tomar el atributo

<xs:restriction base=”xs:token>

<xs:enumeration value=”small”/>

<xs:enumeration value=”medium”/>

-por lista:

-por extension <xs:extension base=” tipodato\_delque\_se\_deriva“>..</xs:extension> =====🡺solo permite crear tipos de datos complejos

Se puede utilizar el atributo final

<xs:complexType name=”persona” final=”(extension|restriction|#all)”> …<xs:complexType>

TIPOS DE DATOS COMPLEJOS

Mixto:texto elementos y atributo 🡪>se indica con el atributo mixed=”true”

Vacio : solo tiene atributos, no tiene texto dentro.

Elementos y atributos: mixe

Texto y atributos (sin elementos) : se indica como xs:complextype xs:simpleContent.

Cualquier cosa ANY

Derivacion por extension:

No

Si quiero un elemento con atributos, pero que el elemento este vacío…o sea que solo tenga atributos…

Hay que definirlo

<xs:element name="trabajador">

<xs:complexType>

<xs:attribute name="nombre" type="palabra" use="required"/>

</xs:complexType>

</xs:element>

CUANDO UN ELEMENTO TIENE SUBELEMENTOS Y ATRIBUTOS (ES COMPLEJO),

PRIMERO SE DEFINEN LOS ELEMENTOS, DESPUES LOS ATRIBUTOS (FUERA DE LA SECUENCIA DE ELEMENTOS)

<xs:element name="articulo">

                <xs:complexType>

                           <xs:sequence>

                                        <xs:element name="n\_unidades" type="xs:positiveInteger"/>

                                        <xs:element name="precio" type="xs:positiveInteger"/>

                                        <xs:element name="observaciones" type="palabra"/>

                             </xs:sequence>

                             <xs:attribute name="cod\_articulo" type="codigo3cifras"/>

                   </xs:complexType>

           </xs:element>