[Elementos simples 2](#_Toc64029508)

[Definición de elementos simples 2](#_Toc64029509)

[Puerta cerrada y ventana abierta 2](#_Toc64029510)

[Elementos simples 3](#_Toc64029511)

[Fichas de personas 3](#_Toc64029512)

[Restricciones 4](#_Toc64029513)

[Edad entre 0 y 130 años 4](#_Toc64029514)

[Precios de tres dígitos 5](#_Toc64029515)

[Tipo de vehículo 6](#_Toc64029516)

[Iniciales de personas famosas 7](#_Toc64029517)

[Iniciales al revés 9](#_Toc64029518)

[Respuestas admitidas 11](#_Toc64029519)

[Números y letras 12](#_Toc64029520)

[Escribir expresiones regulares 13](#_Toc64029521)

[Letras admitidas 15](#_Toc64029522)

[Longitud fija de una clave 16](#_Toc64029523)

[Longitud mínima y máxima de una clave 17](#_Toc64029524)

[Extensiones 18](#_Toc64029525)

[Información de persona ampliada 18](#_Toc64029526)

[Colores de muebles 21](#_Toc64029527)

[Información de ubicaciones 22](#_Toc64029528)

[Precios de artículos 24](#_Toc64029529)

[Indicadores 25](#_Toc64029530)

[Panel de vuelos 25](#_Toc64029531)

[Factura 28](#_Toc64029532)

[Registro de conexiones 32](#_Toc64029533)

## Elementos simples

### Definición de elementos simples

Para los siguientes elementos:

Escribir sus definiciones de elementos simples correspondientes:

|  |
| --- |
| <ciudad>Roma</ciudad>  <fecha-de-nacimiento>1996-12-18</fecha-de-nacimiento>  <hora>18:29:45</hora>  <nota>7.5</nota>  <apto>true</apto> |

Solución

|  |
| --- |
| <xs:element name="**ciudad**" type="**xs:string**"/>  <xs:element name="**fecha-de-nacimiento**" type="**xs:date**"/>  <xs:element name="**hora**" type="**xs:time**"/>  <xs:element name="**nota**" type="**xs:decimal**"/>  <xs:element name="**apto**" type="**xs:boolean**"/> |

### Puerta cerrada y ventana abierta

Definir un elemento llamado puertaCerrada de tipo lógico, que por defecto tenga el valor "falso", y otro elemento llamado ventanaAbierta también de tipo lógico, que tenga asignado el valor fijo "verdadero".

|  |
| --- |
| <xs:element name="**puertaCerrada**" type="**xs:boolean**" **default="false"**/>  <xs:element name="**ventanaAbierta**" type="**xs:boolean**" **fixed="true"**/> |

## Elementos simples

### Fichas de personas

Escribir el contenido del archivo "fichas.xsd" que permita validarlo.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha numero="1">  <nombre>Ana Sanz Tin</nombre>  <edad>22</edad>  </ficha>  <ficha numero="2">  <nombre>Iker Rubio Mol</nombre>  <edad>23</edad>  </ficha>  </fichas> |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  <xs:element name="edad" type="xs:integer"/>  </xs:sequence>  **<xs:attribute name="numero" type="xs:integer"/>**  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

El atributo hay que declararlo, obligatoriamente, justo antes de la etiqueta de cierre </xs:complexType>.

## Restricciones

### Edad entre 0 y 130 años

Dado el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha numero="1">  <nombre>Ana Sanz Tin</nombre>  <edad>22</edad>  </ficha>  <ficha numero="2">  <nombre>Iker Rubio Mol</nombre>  <edad>23</edad>  </ficha>  </fichas> |

Escribir el contenido del archivo "fichas.xsd" que permita validarlo, teniendo en cuenta que se debe definir la "edad" con la restricción de que el valor que tome no pueda ser menor que 0 ni mayor que 130. Además, en vez de xs:minInclusive y xs:maxInclusive, se debe utilizar:

* xs:minExclusive que sirve para especificar que el valor debe ser mayor que el indicado.
* xs:maxExclusive que sirve para especificar que el valor debe ser menor que el indicado.

**Solución 1: la restricción solamente debe poder ser utilizada por el elemento "edad"**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  **<xs:element name="edad">**  **<xs:simpleType>**  **<xs:restriction base="xs:integer">**  **<xs:minExclusive value="-1"/>**  **<xs:maxExclusive value="131"/>**  **</xs:restriction>**  **</xs:simpleType>**  **</xs:element>**  </xs:sequence>  <xs:attribute name="numero" type="xs:integer"/>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

**Solución 2: la restricción debe poder ser utilizada por "edad" y otros elementos**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  **<xs:element name="edad" type="edadPersona"/>**  </xs:sequence>  <xs:attribute name="numero" type="xs:integer"/>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  **<xs:simpleType name="edadPersona">**  **<xs:restriction base="xs:integer">**  **<xs:minExclusive value="-1"/>**  **<xs:maxExclusive value="131"/>**  **</xs:restriction>**  **</xs:simpleType>**  </xs:schema> |

### Precios de tres dígitos

Dado el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <precios xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="precios.xsd">  <precio>8</precio>  <precio>2.6</precio>  <precio>4.95</precio>  <precio>187</precio>  </precios> |

Escribir el contenido del archivo "precios.xsd" que permita validarlo, teniendo en cuenta que el elemento "precio" puede tomar por valor un número que contenga tres dígitos como máximo y, de ellos, solamente dos pueden ser decimales. Para ello, escribir una restricción que no podrá ser utilizada por otros elementos y, por otra parte, haga uso de:

* xs:totalDigits que sirve para especificar el número máximo de dígitos que puede tener un número, incluyendo a los decimales.
* xs:fractionDigits que sirve para especificar el número máximo de decimales que puede tener un número.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="precios">  <xs:complexType>  <xs:sequence>  <xs:element name="precio" maxOccurs="unbounded">  <xs:simpleType>  <xs:restriction base="xs:decimal">  <xs:totalDigits value="3"/>  <xs:fractionDigits value="2"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

### Tipo de vehículo

Dada la siguiente definición:

|  |
| --- |
| <xs:element name="vehiculo">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:enumeration value="barco"/>  <xs:enumeration value="bicicleta"/>  <xs:enumeration value="coche"/>  <xs:enumeration value="tren"/>  </xs:restriction>  </xs:simpleType>  </xs:element> |

¿De qué otro modo se puede definir el elemento "vehículo" y un xs:simleType llamado tipoDeVehiculo que restringiese a "barco", "bicicleta", "coche" y "tren" como los únicos valores aceptables para el vehículo, de forma que dicho tipo pudiera ser también utilizado por otros elementos?

|  |
| --- |
| <xs:element name="vehiculo" type="**tipoDeVehiculo**"/>  <xs:simpleType name="**tipoDeVehiculo**">  <xs:restriction base="xs:string">  <xs:enumeration value="barco"/>  <xs:enumeration value="bicicleta"/>  <xs:enumeration value="coche"/>  <xs:enumeration value="tren"/>  </xs:restriction>  </xs:simpleType> |

### Iniciales de personas famosas

Dado el archivo "fichas.xsd" cuyo contenido es:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  <xs:element name="iniciales">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="[A-Z][A-Z][A-Z]"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="edad" type="xs:integer"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

Corregir los errores cometidos en el siguiente documento XML, para que sea válido:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha>  <nombre>Antonio Machado Ruiz</nombre>  <iniciales>AMR</iniciales>  <edad>22</edad>  </ficha>  <ficha>  <nombre>Mario Moreno</nombre>  <iniciales>MM</iniciales>  <edad>23</edad>  </ficha>  <ficha>  <iniciales>ALO</iniciales>  <nombre>Ada Lovelace</nombre>  <edad>24</edad>  </ficha>  <ficha>  <nombre>pablo ruiz picasso</nombre>  <iniciales>prp</iniciales>  <edad>24</edad>  </ficha>  </fichas> |

SOLUCION

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha>  <nombre>Antonio Machado Ruiz</nombre>  <iniciales>AMR</iniciales>  <edad>22</edad>  </ficha>  <ficha>  <nombre>Mario Moreno</nombre>  <iniciales>MM**O**</iniciales>  <edad>23</edad>  </ficha>  <ficha>  **<nombre>Ada Lovelace</nombre>**  **<iniciales>ALO</iniciales>**  <edad>24</edad>  </ficha>  <ficha>  <nombre>pablo ruiz picasso</nombre>  <iniciales>**PRP**</iniciales>  <edad>24</edad>  </ficha>  </fichas> |

**Errores detectados y corregidos**

* <xs:pattern value="[A-Z][A-Z][A-Z]"/> indica que el valor del elemento "iniciales" tiene ser una cadena de caracteres de tres letras mayúsculas de la "A" a la "Z". Por tanto, el valores "MM" y "prp" no son admitidos.
* Con <xs:sequence> se ha indicado que los elementos "nombre", "iniciales" y "edad" tienen que escribirse en este orden.

### Iniciales al revés

Realizar los cambios necesarios en "fichas.xsd", para que el siguiente documento XML sea válido:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  <xs:element name="iniciales">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="[A-Z][A-Z][A-Z]"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="edad" type="xs:integer"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha>  <nombre>Ana Sanz Tin</nombre>  <iniciales>AST</iniciales>  <edad>22</edad>  <iniciales-al-reves>TSA</iniciales-al-reves>  </ficha>  <ficha>  <nombre>Iker Rubio Mol</nombre>  <iniciales>IRM</iniciales>  <edad>23</edad>  <iniciales-al-reves>MRI</iniciales-al-reves>  </ficha>  </fichas> |

Nota: hay que tener en cuenta que el elemento iniciales-al-reves debe ser del mismo tipo que iniciales, y dicho tipo solamente deberá definirse una vez.

**SOLUCION**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  <xs:element name="iniciales" **type="tresLetras"**/>  <xs:element name="edad" type="xs:integer"/>  **<xs:element name="iniciales-al-reves" type="tresLetras"/>**  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  **<xs:simpleType name="tresLetras">**  **<xs:restriction base="xs:string">**  **<xs:pattern value="[A-Z][A-Z][A-Z]"/>**  **</xs:restriction>**  **</xs:simpleType>**  </xs:schema> |

Ampliación del ejercicio

Si se quisiera permitir que las tres letras de iniciales e iniciales-al-reves, sean minúsculas o mayúsculas indistintamente; en lugar de [A-Z][A-Z][A-Z] ¿qué se podría escribir?

|  |
| --- |
| [a-zA-Z][a-zA-Z][a-zA-Z] |

### Respuestas admitidas

En el siguiente ejemplo se define un elemento llamado "respuesta" con la restricción de que el único valor aceptable es una de las siguientes letras: "A", "B", "C", "D" o "E":

|  |
| --- |
| <xs:element name="**respuesta**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="[ABCDE]"/>**  </xs:restriction>  </xs:simpleType>  </xs:element> |

En vez de escribiendo "[ABCDE]", ¿de qué otras formas se podría especificar la misma restricción?

|  |
| --- |
| "[A-E]"  "[A-DE]"  "[AB-DE]"  ...  "[A|B|C|D|E]"  "[A-D|E]"  ... |

**Ampliación del ejercicio**

Sin hacer uso de xs:pattern, ¿de qué otro modo podríamos especificar lo mismo que lo expresado con <xs:pattern value="[ABCDE]"/>?

|  |
| --- |
| <xs:element name="respuesta">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:enumeration value="A"/>**  **<xs:enumeration value="B"/>**  **<xs:enumeration value="C"/>**  **<xs:enumeration value="D"/>**  **<xs:enumeration value="E"/>**  </xs:restriction>  </xs:simpleType>  </xs:element> |

### Números y letras

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha numero="01" letra="Z">  <codigo>11</codigo>  <nombre>Ana Sanz Tin</nombre>  </ficha>  <ficha numero="02">  <codigo>12</codigo>  <nombre>Iker Rubio Mol</nombre>  </ficha>  </fichas> |

Escribir el contenido del archivo "fichas.xsd" que permita validarlo, teniendo en cuenta que:

* Tanto el atributo numero como el elemento "código" utilizan la misma restricción que solamente les permite tomar un valor entero expresado con dos dígitos comprendidos entre "00" y "19".
* El atributo letra puede tomar por valor una de las siguientes letras: "X", "Y" o "Z". La restricción debe definirse de forma que solamente pueda ser utilizada por dicho atributo.
* Para cada ficha se tiene que indicar un número, obligatoriamente. Sin embargo, la letra es opcional.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="codigo" **type="dosDigitos"**/>  <xs:element name="nombre" type="xs:string"/>  </xs:sequence>  **<xs:attribute name="numero" type="dosDigitos" use="required"/>**  **<xs:attribute name="letra">**  **<xs:simpleType>**  **<xs:restriction base="xs:string">**  **<xs:pattern value="[XYZ]"/>**  **</xs:restriction>**  **</xs:simpleType>**  **</xs:attribute>**  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  **<xs:simpleType name="dosDigitos">**  **<xs:restriction base="xs:integer">**  **<xs:pattern value="[01][0-9]"/>**  **</xs:restriction>**  **</xs:simpleType>**  </xs:schema> |

### Escribir expresiones regulares

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <expresiones-regulares xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="expresiones-regulares.xsd">  <expresiones>  <expresion1>Capítulo 0</expresion1>  <expresion2>Capítulo 0</expresion2>  <expresion3>Capítulo 1</expresion3>  <expresion4>Capítulo 0</expresion4>  <expresion5>aZ</expresion5>  <expresion6>ABBC</expresion6>  <expresion7>RSS</expresion7>  <expresion8>COD645pera</expresion8>  </expresiones>  <expresiones>  <expresion1>Capítulo 9</expresion1>  <expresion2>Capítulo 99</expresion2>  <expresion3>Capítulo 99</expresion3>  <expresion4>Capítulo 99999</expresion4>  <expresion5>?Z</expresion5>  <expresion6>ABBBBBC</expresion6>  <expresion7>RSSSSSSSSSSS7</expresion7>  <expresion8>COD646manzana</expresion8>  </expresiones>  </expresiones-regulares> |

Escribir el contenido del archivo "expresiones-regulares.xsd" que permita validarlo utilizando las expresiones regulares escritas en este ejercicio.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="expresiones-regulares">  <xs:complexType>  <xs:sequence>  <xs:element name="expresiones" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="**expresion1**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="Capítulo \d"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="**expresion2**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="Capítulo \d\d?"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="**expresion3**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="Capítulo [1-9]\d?"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="**expresion4**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="Capítulo \d+"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="**expresion5**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="\DZ"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="**expresion6**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="AB{2,5}C"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="**expresion7**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="RSS+[3-8]?"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="**expresion8**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="COD\d{3}.+"/>**  </xs:restriction>  </xs:simpleType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

### Letras admitidas

En el siguiente ejemplo se define un elemento llamado "letras" con la restricción de que puede tomar por valor cero o más (\*) letras minúsculas de la "a" a la "z":

|  |
| --- |
| <xs:element name="**letras**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="([a-z])\*"/>**  </xs:restriction>  </xs:simpleType>  </xs:element> |

Nota: los paréntesis de la expresión regular se pueden omitir, escribiendo simplemente: [a-z]\*

Realizar los cambios necesarios en el código del ejemplo anterior para que "letras" pueda tomar por valor uno o más pares (+) de letras, y cada par de letras deberá estar formado por una letra mayúscula seguida de otra minúscula. Por ejemplo, "HoLa" sería admitido, pero no lo sería "Hola", "HOLa", "hola", etc.

Solución:

|  |
| --- |
| <xs:element name="letras">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="**([A-Z][a-z])+**"/>  </xs:restriction>  </xs:simpleType>  </xs:element> |

Fíjese que, ahora, los paréntesis de la expresión regular ([A-Z][a-z])+ no se pueden omitir.

### Longitud fija de una clave

Definir un elemento "clave" que pueda tomar por valor exactamente diez caracteres, los cuales podrán ser letras mayúsculas o minúsculas de la "a" a la "z", o dígitos del "0" al "9". Por ejemplo, serán válidos los valores siguientes: "abcde12345", "Clave55ABC", "1A2b3c4D5f", etc.

Solución 1: utilizando las facetas xs:pattern y xs:length

|  |
| --- |
| <xs:element name="**clave**">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="[a-zA-Z0-9]\*"/>**  **<xs:length value="10"/>**  </xs:restriction>  </xs:simpleType>  </xs:element> |

Solución 2: utilizando solamente xs:pattern

|  |
| --- |
| <xs:element name="clave">  <xs:simpleType>  <xs:restriction base="xs:string">  **<xs:pattern value="[a-zA-Z0-9]{10}"/>**  </xs:restriction>  </xs:simpleType>  </xs:element> |

### Longitud mínima y máxima de una clave

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha>  <nombre>Ana Sanz Tin</nombre>  <clave>G8w2</clave>  </ficha>  <ficha>  <nombre>Iker Rubio Mol</nombre>  <clave>ag32Ue7AFF</clave>  </ficha>  </fichas> |

Escribir el contenido del archivo "fichas.xsd" que permita validarlo, teniendo en cuenta que el elemento "clave" debe poder tomar por valor un mínimo de cuatro caracteres y un máximo de diez. Dichos caracteres pueden ser indistintamente letras mayúsculas o minúsculas de la "a" a la "z", o dígitos del "0" al "9". La restricción solamente podrá aplicarse al elemento "clave".

Para ello, se debe utilizar xs:pattern y también:

* xs:minLength que permite especificar la longitud mínima.
* xs:maxLength que permite especificar la longitud máxima.

Solución

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  **<xs:element name="clave">**  **<xs:simpleType>**  **<xs:restriction base="xs:string">**  **<xs:pattern value="[a-zA-Z0-9]\*"/>**  **<xs:minLength value="4"/>**  **<xs:maxLength value="10"/>**  **</xs:restriction>**  **</xs:simpleType>**  **</xs:element>**  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

## Extensiones

### Información de persona ampliada

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" type="infoPersonaAmpliada" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:complexType name="infoPersonaAmpliada">  <xs:complexContent>  <xs:extension base="infoPersona">  <xs:sequence>  <xs:element name="ciudad" type="xs:string"/>  <xs:element name="pais" type="xs:string"/>  </xs:sequence>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="infoPersona">  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  <xs:element name="edad" type="edadPersona"/>  </xs:sequence>  <xs:attribute name="numero" type="xs:integer"/>  </xs:complexType>  <xs:simpleType name="edadPersona">  <xs:restriction base="xs:integer">  <xs:minExclusive value="-1"/>  <xs:maxExclusive value="131"/>  </xs:restriction>  </xs:simpleType>  </xs:schema> |

Añadir, al archivo "fichas.xsd", la definición de un nuevo elemento complexType llamado "infoPersonaAmpliada2" que amplíe la definición de "infoPersonaAmpliada", permitiendo validar el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="fichas.xsd">  <ficha numero="1">  <nombre>Eva</nombre>  <edad>25</edad>  <ciudad>París</ciudad>  <pais>Francia</pais>  **<telefono>999888777</telefono>**  </ficha>  <ficha numero="2">  <nombre>Giovanni</nombre>  <edad>26</edad>  <ciudad>Florencia</ciudad>  <pais>Italia</pais>  **<telefono>111222333</telefono>**  </ficha>  </fichas> |

Solución

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="fichas">  <xs:complexType>  <xs:sequence>  <xs:element name="ficha" type="**infoPersonaAmpliada2**" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  **<xs:complexType name="infoPersonaAmpliada2">**  **<xs:complexContent>**  **<xs:extension base="infoPersonaAmpliada">**  **<xs:sequence>**  **<xs:element name="telefono" type="xs:string"/>**  **</xs:sequence>**  **</xs:extension>**  **</xs:complexContent>**  **</xs:complexType>**  <xs:complexType name="infoPersonaAmpliada">  <xs:complexContent>  <xs:extension base="infoPersona">  <xs:sequence>  <xs:element name="ciudad" type="xs:string"/>  <xs:element name="pais" type="xs:string"/>  </xs:sequence>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="infoPersona">  <xs:sequence>  <xs:element name="nombre" type="xs:string"/>  <xs:element name="edad" type="edadPersona"/>  </xs:sequence>  <xs:attribute name="numero" type="xs:integer"/>  </xs:complexType>  <xs:simpleType name="edadPersona">  <xs:restriction base="xs:integer">  <xs:minExclusive value="-1"/>  <xs:maxExclusive value="131"/>  </xs:restriction>  </xs:simpleType>  </xs:schema> |

### Colores de muebles

|  |
| --- |
| <xs:complexType name="tipoColorMueble">  <xs:simpleContent>  <xs:extension base="tipoMueble">  <xs:attribute name="color">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:enumeration value="blanco"/>  <xs:enumeration value="gris"/>  <xs:enumeration value="negro"/>  <xs:enumeration value="wengue"/>  </xs:restriction>  </xs:simpleType>  </xs:attribute>  </xs:extension>  </xs:simpleContent>  </xs:complexType>  <xs:simpleType name="tipoMueble">  <xs:restriction base="xs:string">  <xs:enumeration value="armario"/>  <xs:enumeration value="mesa"/>  <xs:enumeration value="silla"/>  </xs:restriction>  </xs:simpleType> |

Escribir el contenido del archivo "muebles.xsd" que permita validar el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <muebles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="muebles.xsd">  <mueble color="blanco">mesa</mueble>  <mueble color="gris">silla</mueble>  </muebles> |

Solución:

|  |
| --- |
| **<?xml version="1.0" encoding="UTF-8"?>**  **<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">**  **<xs:element name="muebles">**  **<xs:complexType>**  **<xs:sequence>**  **<xs:element name="mueble" type="tipoColorMueble" maxOccurs="unbounded"/>**  **</xs:sequence>**  **</xs:complexType>**  **</xs:element>**  <xs:complexType name="tipoColorMueble">  <xs:simpleContent>  <xs:extension base="tipoMueble">  <xs:attribute name="color">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:enumeration value="blanco"/>  <xs:enumeration value="gris"/>  <xs:enumeration value="negro"/>  <xs:enumeration value="wengue"/>  </xs:restriction>  </xs:simpleType>  </xs:attribute>  </xs:extension>  </xs:simpleContent>  </xs:complexType>  <xs:simpleType name="tipoMueble">  <xs:restriction base="xs:string">  <xs:enumeration value="armario"/>  <xs:enumeration value="mesa"/>  <xs:enumeration value="silla"/>  </xs:restriction>  </xs:simpleType>  **</xs:schema>** |

### Información de ubicaciones

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="ubicaciones">  <xs:complexType>  <xs:sequence>  <xs:element name="ubicacion" type="direccion" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:simpleType name="direccion">  <xs:restriction base="xs:string">  <xs:enumeration value="norte"/>  <xs:enumeration value="sur"/>  <xs:enumeration value="este"/>  <xs:enumeration value="oeste"/>  </xs:restriction>  </xs:simpleType>  </xs:schema> |

Añadir, al archivo "ubicaciones.xsd", la definición de un nuevo elemento complexType llamado "infoUbicacion" que amplíe la definición de "direccion", permitiendo validar el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <ubicaciones xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="ubicaciones.xsd">  <ubicacion metros="32">norte</ubicacion>  <ubicacion metros="25">este</ubicacion>  <ubicacion metros="64">este</ubicacion>  </ubicaciones> |

SOLUCION

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="ubicaciones">  <xs:complexType>  <xs:sequence>  <xs:element name="ubicacion" type="**infoUbicacion**" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  **<xs:complexType name="infoUbicacion">**  **<xs:simpleContent>**  **<xs:extension base="direccion">**  **<xs:attribute name="metros" type="xs:integer"/>**  **</xs:extension>**  **</xs:simpleContent>**  **</xs:complexType>**  <xs:simpleType name="direccion">  <xs:restriction base="xs:string">  <xs:enumeration value="norte"/>  <xs:enumeration value="sur"/>  <xs:enumeration value="este"/>  <xs:enumeration value="oeste"/>  </xs:restriction>  </xs:simpleType>  </xs:schema> |

### Precios de artículos

Dado el archivo "precios.xsd":

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="precios">  <xs:complexType>  <xs:sequence>  <xs:element name="precio" maxOccurs="unbounded">  <xs:complexType>  <xs:simpleContent>  <xs:extension base="xs:decimal">  <xs:attribute name="moneda" type="xs:string" />  </xs:extension>  </xs:simpleContent>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

Realizar los cambios necesarios en el archivo "precios.xsd" para que permita validar el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <articulos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="articulos.xsd">  <articulo>  <nombre>Mesa</nombre>  <precio moneda="Euro">50</precio>  </articulo>  <articulo>  <nombre>Silla</nombre>  <precio moneda="Dólar">78.99</precio>  </articulo>  </articulos> |

solución

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="**articulos**">  <xs:complexType>  <xs:sequence>  <xs:element name="**articulo**" maxOccurs="unbounded">  <xs:complexType>  **<xs:sequence>**  **<xs:element name="nombre" type="xs:string"/>**  **<xs:element name="precio">**  **<xs:complexType>**  <xs:simpleContent>  <xs:extension base="xs:decimal">  <xs:attribute name="moneda" type="xs:string"/>  </xs:extension>  </xs:simpleContent>  **</xs:complexType>**  **</xs:element>**  **</xs:sequence>**  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema> |

El archivo "precios.xsd" tiene que renombrarse a "articulos.xsd".

## Indicadores

### Panel de vuelos

Si para representar la siguiente información ficticia:

| **PANEL DE VUELOS DE SALIDA DEL AEROPUERTO JFK – 20/12/2013** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| ***Código*** | ***Diario*** | ***Origen*** | ***Destino*** | ***Hora salida*** | ***Hora llegada*** | ***Estado*** |
| V22 | SI | New York | Chicago | 9:30 | 11:30 | R |
| V23 | NO | New York | Miami | 10:15 | 11:15 | C |

se ha escrito el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <aeropuerto xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="aeropuerto.xsd">  <nombre>JFK</nombre>  <vuelos>  <vuelo código="V22" estado="R">  <diario />  <origen>New York</origen>  <destino>Chicago</destino>  <hora-salida>09:30:00</hora-salida>  <hora-llegada>11:30:00</hora-llegada>  </vuelo>  <vuelo código="V23" estado="C">  <origen>New York</origen>  <destino>Miami</destino>  <hora-salida>10:15:00</hora-salida>  <hora-llegada>11:15:00</hora-llegada>  </vuelo>  </vuelos>  <fecha>2013-12-20</fecha>  </aeropuerto> |

Escribir el código del archivo "aeropuerto.xsd" que permita validarlo, teniendo en cuenta que:

No debe utilizarse ni group ni attributeGroup.

* El nombre del aeropuerto, los vuelos y la fecha pueden aparecer en distinto orden.
* Se tiene que indicar que el código ha de ser único (esto se puede hacer definiéndolo de tipo xs:ID) y obligatorio para cada vuelo.
* Haciendo uso pattern indicar que los posibles estados de un vuelo son C (Cancelado), E (En hora), R (Retrasado). Dicha restricción sólo debe poder ser utilizada por el atributo estado. El valor por defecto debe ser E.
* Debe permitirse aparecer desde cero hasta ilimitados elementos vuelo y, para cada uno de ellos, se tiene que guardar la información en el mismo orden en el que aparece en el panel.
* Para indicar si un vuelo es diario, se debe utilizar un elemento vacío que, respecto a cada vuelo, podrá aparecer (en el caso de sí ser diario) o no aparecer (en el caso contrario).
* Respecto a los elementos nombre, origen, destino, hora-llegada, hora-salida y fecha, cada uno de ellos debe definirse del tipo que se considere más apropiado, de entre los proporcionados por XML Schema.

SOLUCIÓN: aeropuerto.xsd

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="aeropuerto">  <xs:complexType>  <xs:all>  <xs:element name="nombre" type="xs:string"/>  <xs:element name="vuelos">  <xs:complexType>  <xs:sequence>  <xs:element name="vuelo" minOccurs="0" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="diario" minOccurs="0"/>  <xs:element name="origen" type="xs:string"/>  <xs:element name="destino" type="xs:string"/>  <xs:element name="hora-salida" type="xs:time"/>  <xs:element name="hora-llegada" type="xs:time"/>  </xs:sequence>  <xs:attribute name="código" type="xs:ID" use="required"/>  <xs:attribute name="estado" default="E">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="[CER]"/>  </xs:restriction>  </xs:simpleType>  </xs:attribute>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="fecha" type="xs:date"/>  </xs:all>  </xs:complexType>  </xs:element>  </xs:schema> |

### Factura

Si para representar la información contenida en la siguiente factura ficticia:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FACTURA NÚMERO 27 – FECHA: 18/12/2013** | | | | | |
| **DATOS EMISOR:** | | | **DATOS CLIENTE:** | | |
| Librería Pérez | | | Biblioteca Txantrea | | |
| *CIF:* 44555666B | | | *CIF:* 33111222A | | |
| *Teléfono:* 777888999 | | | *Teléfono:* 333999444 | | |
| **DETALLE FACTURA:** | | | | | |
| CÓDIGO-ARTÍCULO | TIPO | DESCRIPCIÓN | CANTIDAD | OFERTA | PVP |
| AW7 | Libro | Analítica Web 2.0 | 1 | SI | 25.12€ |
| CP5 | DVD | Curso de HTML | 2 | NO | 30.5€ |
| *IMPORTE:* | | | | | 86.12€ |

se ha escrito el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <factura xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="factura.xsd"  número="27" fecha="2013-12-18" moneda="Euro">  <datos-emisor>  <nombre>Librería Pérez</nombre>  <cif>44555666-B</cif>  <teléfono>777888999</teléfono>  </datos-emisor>  <datos-cliente>  <nombre>Biblioteca Txantrea</nombre>  <cif>33111222-A</cif>  <teléfono>333999444</teléfono>  </datos-cliente>  <detalle-factura importe="86.12">  <línea código-artículo="AW7" tipo="Libro">  <descripción>Analítica Web 2.0</descripción>  <cantidad>1</cantidad>  <oferta />  <pvp>25.12</pvp>  </línea>  <línea código-artículo="CP5" tipo="DVD">  <descripción>Curso de HTML</descripción>  <cantidad>2</cantidad>  <pvp>30.5</pvp>  </línea>  </detalle-factura>  </factura> |

Escribir el código del archivo "factura.xsd" que permita validarlo, teniendo en cuenta que:

* Exceptuando los elementos datos-emisor, datos-cliente y detalle-factura, que no tienen porqué aparecer en este orden, el resto de elementos representados en el documento XML, sí deben escribirse obligatoriamente en el orden en el que aparecen.
* Excepto para los hijos directos del elemento factura, siempre que sea posible agrupar al menos dos elementos o dos atributos, se debe usar group o attributeGroup, respectivamente.
* Respecto al número de la factura (que debe ser un valor entero mayor que 0) y su fecha de emisión (de tipo xs:date), hay que indicar que son atributos obligatorios.
* El atributo moneda debe indicarse que es un valor fijo.
* Los nombres del emisor y cliente, así como, la descripción de cada artículo, deben ser del mismo tipo, al que llamaremos tipoTexto, y donde debe indicarse que los valores admitidos para dichos elementos pueden ser cadenas de un mínimo de 5 caracteres y un máximo de 20.
* Haciendo uso pattern hay que indicar que el valor del cif debe estar formado por una cadena de ocho dígitos del 0 al 9, seguidos de un guión "-" y una letra mayúscula de la "A" a la "Z". Dicha restricción sólo debe poder ser utilizada por el elemento cif.
* Haciendo uso pattern hay que indicar que el valor del teléfono debe estar formado por una cadena de nueve dígitos del 0 al 9. Dicha restricción sólo debe poder ser utilizada por el teléfono.
* Al menos tiene que aparecer una línea de detalle y como máximo 15.
* El importe debe indicarse que es obligatorio.
* El importe y el pvp deben ser del mismo tipo, al que llamaremos tipoPrecio, y donde debe indicarse, sin hacer uso de pattern, que los valores admitidos por este tipo pueden ser números decimales mayores que 0, pero no mayores que 999. Además, dichos valores podrán contener cinco dígitos como máximo y, de ellos, sólo dos podrán ser decimales.
* El código del artículo ha de ser único y obligatorio para cada artículo.
* Sin hacer uso pattern indicar que los posibles tipos de un artículo son Libro, DVD o Varios, no permitiéndose otro valor. Para ello, se debe definir un tipo de dato llamado tipoArtículo, que debe poder ser utilizado por otros atributos o elementos. Ahora bien, hay que tener en cuenta que este atributo es opcional.
* La cantidad de artículos indicada en cada línea, debe ser un valor entero mayor que 0.
* Para indicar si un artículo está de oferta, se debe utilizar un elemento vacío que, respecto a cada artículo, podrá aparecer (en el caso de sí estar de oferta) o no aparecer (en el caso contrario).
* No hay que definir más tipos de datos que los especificados en el ejercicio: tipoTexto, tipoArtículo y tipoPrecio.

SOLUCION: factura.xsd

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="factura">  <xs:complexType>  <xs:all>  <xs:element name="datos-emisor">  <xs:complexType>  <xs:sequence>  <xs:group ref="datosEmpresa"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="datos-cliente">  <xs:complexType>  <xs:sequence>  <xs:group ref="datosEmpresa"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="detalle-factura">  <xs:complexType>  <xs:sequence>  <xs:element name="línea" maxOccurs="15">  <xs:complexType>  <xs:sequence>  <xs:group ref="datosLínea"/>  </xs:sequence>  <xs:attributeGroup ref="atributosLínea"/>  </xs:complexType>  </xs:element>  </xs:sequence>  <xs:attribute name="importe" type="tipoPrecio"  use="required"/>  </xs:complexType>  </xs:element>  </xs:all>  <xs:attributeGroup ref="atributosFactura"/>  </xs:complexType>  </xs:element>  <xs:group name="datosEmpresa">  <xs:sequence>  <xs:element name="nombre" type="tipoTexto"/>  <xs:element name="cif">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="\d{8}-[A-Z]"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="teléfono">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="\d{9}"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  </xs:sequence>  </xs:group>  <xs:group name="datosLínea">  <xs:sequence>  <xs:element name="descripción" type="tipoTexto"/>  <xs:element name="cantidad" type="xs:positiveInteger"/>  <xs:element name="oferta" minOccurs="0"/>  <xs:element name="pvp" type="tipoPrecio"/>  </xs:sequence>  </xs:group>    <xs:attributeGroup name="atributosFactura">  <xs:attribute name="número" type="xs:positiveInteger"  use="required"/>  <xs:attribute name="fecha" type="xs:date" use="required"/>  <xs:attribute name="moneda" fixed="Euro"/>  </xs:attributeGroup>  <xs:attributeGroup name="atributosLínea">  <xs:attribute name="código-artículo" type="xs:ID"  use="required"/>  <xs:attribute name="tipo" type="tipoArtículo"/>  </xs:attributeGroup>  <xs:simpleType name="tipoTexto">  <xs:restriction base="xs:string">  <xs:minLength value="5"/>  <xs:maxLength value="20"/>  </xs:restriction>  </xs:simpleType>  <xs:simpleType name="tipoArtículo">  <xs:restriction base="xs:string">  <xs:enumeration value="Libro"/>  <xs:enumeration value="DVD"/>  <xs:enumeration value="Varios"/>  </xs:restriction>  </xs:simpleType>  <xs:simpleType name="tipoPrecio">  <xs:restriction base="xs:decimal">  <xs:minExclusive value="0"/>  <xs:maxInclusive value="999"/>  <xs:totalDigits value="5"/>  <xs:fractionDigits value="2"/>  </xs:restriction>  </xs:simpleType>    </xs:schema> |

### Registro de conexiones

Si para representar la siguiente información ficticia:

|  |  |
| --- | --- |
| **REGISTRO DE CONEXIONES DE USUARIOS Y EMPLEADOS DE UNA EMPRESA** | |
| **USUARIOS:** | |
| IDENTIFICADOR NOMBRE APELLIDOS EMAIL | CONEXIONES (FECHA HORA TIEMPO) |
| U123&Ana&Sanz Tapia&asanz@jmail.com | 2014-02-23&19:15:40&122  2014-02-23&20:30:22&617  2014-02-24&11:18:31&25 |
| U96&Pedro&Ruiz Hierro&pruiz@jotmail.com | 2014-02-25&20:33:55&390 |
| **EMPLEADOS:** | |
| IDENTIFICADOR NOMBRE APELLIDOS EMAIL DEPARTAMENTO | CONEXIONES (FECHA HORA TIEMPO) |
| E4&Marta&Vera Gil&mvera@yajoo.es&Marketing | *(Ninguna)* |

se ha escrito el siguiente documento XML:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <registro xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="registro.xsd">  <usuario identificador="U123">  <apellidos-y-nombre>Sanz Tapia,Ana</apellidos-y-nombre>  <email>asanz@jmail.com</email>  <conexiones>  <conexión fecha="2014-02-23" hora="19:15:40">122</conexión>  <conexión fecha="2014-02-23" hora="20:30:22">617</conexión>  <conexión fecha="2014-02-24" hora="11:18:31">25</conexión>  </conexiones>  </usuario>  <usuario identificador="U96">  <apellidos-y-nombre>Ruiz Hierro,Pedro</apellidos-y-nombre>  <email>pruiz@jotmail.com</email>  <conexiones>  <conexión fecha="2014-02-25" hora="20:33:55">390</conexión>  </conexiones>  </usuario>  <empleado identificador="E4">  <apellidos-y-nombre>Vera Gil,Marta</apellidos-y-nombre>  <email>mvera@yajoo.es</email>  <conexiones/>  <departamento>  <marketing />  </departamento>  </empleado>  </registro> |

Escribir el código del archivo "registro.xsd" que permita validarlo, teniendo en cuenta que:

* Todos los elementos y atributos son obligatorios, a menos que se indique lo contrario.
* Siempre que sea posible agrupar al menos dos elementos o dos atributos, se debe utilizar group o attributeGroup, respectivamente.
* Pueden aparecer de cero a ilimitados usuarios y, a continuación, de cero a ilimitados empleados.
* Usuario debe ser de un tipo definido por nosotros al que llamaremos tipoPersona, donde hay que definir los elementos apellidos-y-nombre, email y conexiones. Por otro lado, empleado ha de ser de otro tipo llamado tipoPersonaAmpliado, definido como una extensión de tipoPersona añadiéndosele el elemento departamento. El orden en que tienen que aparecer los elementos hijo de usuario y empleado, debe ser obligatoriamente el escrito en el documento XML.
* El valor del identificador debe ser una cadena formada por una letra "U" o "E" mayúscula, seguida de uno a cinco dígitos del 0 al 9.
* El valor del elemento apellidos-y-nombre debe ser una cadena de entre 1 a 30 caracteres (de la "a" a la "z", mayúsculas o minúsculas, o el carácter espacio en blanco) para los apellidos, seguida del carácter coma "," y de entre 1 a otras 20 letras (de la "a" a la "z", también mayúsculas o minúsculas, o el carácter espacio en blanco) para el nombre.
* El valor del email puede ser una cadena formada por 1 a 15 caracteres de la "a" a la "z", seguida del carácter "@", seguido de entre 1 a otras 25 letras de la "a" a la "z", seguidas del carácter punto "." y de entre otras 2 a 4 letras de la "a" a la "z".
* De cada usuario y empleado se reflejan sus conexiones, indicando para cada conexión la cantidad de segundos que duró, que debe ser un número entero mayor que cero. Hay que tener en cuenta que, como se puede ver en el documento XML, pueden aparecer desde cero hasta ilimitados elementos conexión.
* Respecto a los atributos fecha y hora, cada uno de ellos debe definirse del tipo que se considere más apropiado, de entre los proporcionados por XML Schema.
* Los posibles departamentos de la empresa a los que puede pertenecer un empleado son administración, informática o marketing. De tal forma que, para cada empleado, sólo uno de ellos debe escribirse en el documento XML mediante un elemento vacío, como en este caso se ha escrito <marketing />.
* No hay que definir más tipos de datos que los indicados en el ejercicio: tipoPersona y tipoPersonaAmpliado.

**Solucion registro.xsd**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="registro">  <xs:complexType>  <xs:sequence>  <xs:group ref="personas"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:group name="personas">  <xs:sequence>  <xs:element name="usuario" minOccurs="0" maxOccurs="unbounded" type="tipoPersona"/>  <xs:element name="empleado" minOccurs="0" maxOccurs="unbounded" type="tipoPersonaAmpliado"/>  </xs:sequence>  </xs:group>  <xs:complexType name="tipoPersonaAmpliado">  <xs:complexContent>  <xs:extension base="tipoPersona">  <xs:sequence>  <xs:element name="departamento">  <xs:complexType>  <xs:sequence>  <xs:group ref="departamentosEmpresa"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="tipoPersona">  <xs:sequence>  <xs:group ref="datosPersona"/>  </xs:sequence>  <xs:attribute name="identificador" use="required">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="[UE]\d{1,5}"/>  </xs:restriction>  </xs:simpleType>  </xs:attribute>  </xs:complexType>  <xs:group name="datosPersona">  <xs:sequence>  <xs:element name="apellidos-y-nombre">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="[a-zA-Z ]{1,30},[a-zA-Z ]{1,20}"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="email">  <xs:simpleType>  <xs:restriction base="xs:string">  <xs:pattern value="[a-z]{1,15}@[a-z]{1,25}[.][a-z]{2,4}"/>  </xs:restriction>  </xs:simpleType>  </xs:element>  <xs:element name="conexiones">  <xs:complexType>  <xs:sequence>  <xs:element name="conexión" minOccurs="0"  maxOccurs="unbounded">  <xs:complexType>  <xs:simpleContent>  <xs:extension base="xs:positiveInteger">  <xs:attributeGroup ref="atributosConexion"/>  </xs:extension>  </xs:simpleContent>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:group>  <xs:attributeGroup name="atributosConexion">  <xs:attribute name="fecha" type="xs:date" use="required"/>  <xs:attribute name="hora" type="xs:time" use="required"/>  </xs:attributeGroup>  <xs:group name="departamentosEmpresa">  <xs:choice>  <xs:element name="administración"/>  <xs:element name="informática"/>  <xs:element name="marketing"/>  </xs:choice>  </xs:group>    </xs:schema> |