

```
-----  
<comunidad xsi:noNamespaceSchemaLocation="vecinos.xsd"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
-----
```

```
dtd  
<familia codigo="P01-07" xsi:noNamespaceSchemaLocation="familia.xsd"  
    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
```

```
xsd  
<xs:schema version="1.0"  
    xmlns:xs="http://www.w3.org/2001/XMLSchema"  
    elementFormDefault="qualified">
```

```
element  
    complextype  
        sequency  
            element
```

```
minOccurs  
maxOccurs  
attribute
```

```
=====
```

```
(#PCDATA)  
CDATA #REQUIRED  
ID  
IDREFS  
=====
```

```
<!DOCTYPE biblioteca SYSTEM "biblioteca.dtd">  
<!ELEMENT (#PCDATA)>  
<!-- ATTLIST CDATA -->
```

```
=====
```

```
<!DOCTYPE cine[  
<!-- ELEMENT cine(peliculas,directores) -->
```

=====LIBRO=====

xml: libreria

tener cuidado con los espacios

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE biblioteca SYSTEM "biblioteca.dtd" >
<!--declarar el nodo raiz de nuestro xml-->

] <biblioteca><!--raiz-->
]   <libro numero="1"><!--atributo-->
      <titulo>Física y animación</titulo><!--hijo-->
      <autor>Kenny erliben</autor>
      <editorial>river media</editorial>
      <isbn>978-158450403904</isbn>
      <precio>12,00</precio>
-   </libro>
]
      <libro numero="100">
      <titulo>Seguridad informática</titulo>
      <autor>Carlos Guerra</autor>
      <editorial>DYKINSON</editorial>
      <isbn>978-84-9849-998-8</isbn>
      <precio>20,00</precio>
-   </libro>
- </biblioteca>
```

dtd: biblioteca

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

<!ELEMENT biblioteca (libro+) >

<!ELEMENT libro (titulo,autor+,editorial,isbn,precio) >
<!--ATTLIST libro numero CDATA #REQUIRED-->
:
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT editorial (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
```

```

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

<!--DTD-->
<!DOCTYPE cine[
<!ELEMENT cine (peliculas, directores)>

<!--Peliculas-->
<!ELEMENT peliculas (pelicula)+>
<!ELEMENT pelicula (#PCDATA)>
|      <!--ATTLIST pelicula codepel ID #REQUIRED-->

<!--Director-->
<!ELEMENT directores (director)+>
<!ELEMENT director (#PCDATA)>
|      <!--ATTLIST director filmografia IDREFS #REQUIRED-->
]>

<cine>
|  <peliculas>
|    <pelicula codepel="p1">Avatar</pelicula>
|    <pelicula codepel="p2">MysticRiver</pelicula>
|    <pelicula codepel="p3">TheTerminator</pelicula>
|    <pelicula codepel="p4">Titanic</pelicula>
|  </peliculas>
|  <directores>
|    <director filmografia = "p2">Clint Estwood</director>
|    <director filmografia = "p1 p2 p3 p4">Clint Estwood</director>
|  </directores>
</cine>

```

=====ATRIBUTOS LISTAS=====

xml deporte

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

<!DOCTYPE componentes[
<!ELEMENT componentes (resistencias | condensadores)+>

<!ELEMENT resistencia (nombre+, tipo+, precio, medida)>
<!ELEMENT condensadores (nombre+, tipo+, precio, medida)>

<!ELEMENT nombre (#PCDATA)>
<!ELEMENT tipo (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT medida (#PCDATA)>

]>
```

```
] <componentes>
|
|   <resistencias>
|       <nombre>Elias Machine</nombre>
|       <tipo>Peruano</tipo>
|       <precio>12,00€</precio>
|       <medida>3cm</medida>
-   </resistencias>
|
|   <condensasdores>
|       <nombre>Alpine</nombre>
|       <tipo>Carromato</tipo>
|       <precio>10000€</precio>
|       <medida>3m</medida>
-   </condensasdores>
- </componentes>
```

=====COMPONENTES 2 (Alberto)=====

xml componente

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

<!DOCTYPE componentes[

<!ELEMENT componentes (componente)+>

```
<!ELEMENT componente (nombre,precio,tipo+,medida)>
```

<!--ATTLIST componente id CDATA #REQUIRED-->

<!ELEMENT nombre (#PCDATA)>

```
<!ELEMENT precio (#PCDATA)>
```

<!ELEMENT tipo (#PCDATA)>

```
<!ELEMENT medida (#PCDATA)>
```

<componentes>

```
<componente id="1">
```

<nombre>Resistencia</nombre>

```
<precio>30,00</precio>
```

<tipo>01</tipo>

<medida>30cm</medida>

</componente>

```
<componente id="10">
```

<nombre>Condensador</nombre>

```
<precio>45,00</precio>
```

```
<tipo>01 v 02</tipo>
```

<medida>24cm</medida>

</componente>

</componentes>

=====Recursos Humanos=====

xml recursos

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE recursos SYSTEM "recursos.dtd">

<recursos>

    <!--creo un primer empleado que tiene un único apellido,
    y como es el primero tendrá un número de 1-->
    <empleado numero="1">
        <nombre>Antonio</nombre>
        <apellido>Riesgo</apellido>
        <edad>23</edad>
        <departamento>Finanzas</departamento>
        <salario>2500,00</salario>
    </empleado>

    <!--creo un segundo empleado, esta vez con dos apellidos y un número de 10,
    para que se entienda que puede haber todos los empleados deseados-->
    <empleado numero="10">
        <nombre>Marta</nombre>
        <apellido>Naranjo Medina</apellido>
        <edad>40</edad>
        <departamento>Contabilidad</departamento>
        <salario>5000,00</salario>
    </empleado>

</recursos>

```

dtd: recursos

```

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

<!ELEMENT recursos (empleado+)>
<!ELEMENT empleado (nombre,apellido+,edad,departamento,salario)>
    <!--ATTLIST empleado numero CDATA #REQUIRED-->

<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellido (#PCDATA)>
<!ELEMENT edad (#PCDATA)>
<!ELEMENT departamento (#PCDATA)>
<!ELEMENT salario (#PCDATA)>

```

dtd: en línea (es lo mismo)

```

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

<!DOCTYPE recursos [
<!ELEMENT recursos (empleado+)>

<!ELEMENT empleado (nombre,apellido+,edad,departamento,salario)>
    <!ATTLIST empleado NUMERO CDATA #REQUIRED>

<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellido (#PCDATA)>
<!ELEMENT edad (#PCDATA)>
<!ELEMENT departamento (#PCDATA)>
<!ELEMENT salario (#PCDATA)>
]>

| <recursos>
|
|     <empleado numero="1">
|         <nombre>Antonio</nombre>
|         <apellido>Riesgo</apellido>
|         <edad>23</edad>
|         <departamento>Finanzas</departamento>
|         <salario>2500,00</salario>
|     </empleado>
|
|     <!--creo un segundo empleado, esta vez con dos apellidos y un número de 10,
|     para que se entienda que puede haber todos los empleados deseados-->
|     <empleado numero="10">
|         <nombre>Marta</nombre>
|         <apellido>Naranjo Medina</apellido>
|         <edad>40</edad>
|         <departamento>Contabilidad</departamento>
|         <salario>5000,00</salario>
|     </empleado>
|
| </recursos>

```

=====Comunidad vecinos=====

xml comunidad primera parte


```

<?xml version="1.0"?>

<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xs:element name="comunidad">
    <xs:complexType><!--define un tipo complejo(contiene otros elementos y/o atributos)-->
      <xs:sequence><!--especifica que los elementos secundarios deben aparecer en una secuencia.-->
        <xs:element ref="vecino" minOccurs="1" maxOccurs="unbounded"/><!--/unbounded: ilimitado-->
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="vecino">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nombre" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="apellido" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="portal" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="PisoLetra" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="CCC" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="cargo" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="ref" type="xs:integer"/><!--atributo-->
    </xs:complexType>
  </xs:element>

  <!--definir tipo de campo-->
  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="apellido" type="xs:string"/>
  <xs:element name="portal" type="xs:string"/>
  <xs:element name="PisoLetra" type="xs:string"/>
  <xs:element name="CCC" type="xs:string"/>
  <xs:element name="cargo" type="xs:string"/>

</xs:schema>

```

=====Familia=====

xml: familia perez

```
<?xml version="1.0" encoding="UTF-8"?>
<!--<!DOCTYPE familia SYSTEM "familia.dtd"-->
<familia codigo="P01-07" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="Familia.xsd">

  <padre>
    <nombre>Antonio</nombre>
    <apellido>Pérez</apellido>
    <apellido>Rodríguez</apellido>
    <dni>12.345.678-A</dni>
  </padre>

  <madre>
    <nombre>Elena</nombre>
    <apellido>Roldán</apellido>
    <apellido>Benítez</apellido>
    <dni>90.123.456-B</dni>
  </madre>

  <hijos>
    <hijo pos="2">
      <nombre>Alejandro</nombre>
      <apellido>Pérez</apellido>
      <apellido>Roldán</apellido>
      <edad>12</edad>
    </hijo>
    <hijo pos="3">
      <nombre>Manuel</nombre>
      <apellido>Pérez</apellido>
      <apellido>Roldán</apellido>
      <edad>6</edad>
    </hijo>
    <hija pos="1">
      <nombre>Elena</nombre>
      <apellido>Pérez</apellido>
      <apellido>Roldán</apellido>
      <dni>78.901.234-C</dni>
    </hija>
  </hijos>
</familia>
```

dtd: familia.dtd

<!--

1. En una familia puede haber un padre y una madre, o bien un solo padre, o una sola madre

o ninguno de ellos. También hay posibilidad de que sean dos padres o dos madres.

2. El elemento padre y el elemento madre puede aparecer en cualquier orden

3. Posteriormente a los padres aparecerán los hijos.

4. El orden de aparición de hijo o hija es indiferente.

5. Una familia tiene al menos un hijo/a

6. Tanto el elemento padre, como madre, hijo o hija contienen los elementos: nombre, apellido (siempre dos), dni (opcional) y edad (opcional).

7. Limitar el tamaño de nombre y apellidos a una longitud de 30

8. El atributo posición toma valores del 1 al 30-->

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

<!--

1. En una familia puede haber un padre y una madre, o bien un solo padre, o una sola madre

o ninguno de ellos. También hay posibilidad de que sean dos padres o dos madres.

2. El elemento padre y el elemento madre puede aparecer en cualquier orden

3. Posteriormente a los padres aparecerán los hijos.

4. El orden de aparición de hijo o hija es indiferente.

5. Una familia tiene al menos un hijo/a

6. Tanto el elemento padre, como madre, hijo o hija contienen los elementos: nombre, apellido (siempre dos), dni (opcional) y edad (opcional).

7. Limitar el tamaño de nombre y apellidos a una longitud de 30

8. El atributo posición toma valores del 1 al 30-->

```
<!ELEMENT familia (((padre,madre?) | (madre?, padre?) | (padre, padre) | (madre, madre)),hijos)>
```

```
<!--DECLARAR UNA ENTITY PARA NO REPETIR DATOS-->
```

```
<!ENTITY % info_familia "(nombre, apellido, apellido, dni?, edad?)">
```

```
<!ELEMENT hijos (hijo | hija)+>
```

```
<!ELEMENT padre %info_familia;>
```

```
<!ELEMENT madre %info_familia;>
```

```
<!ELEMENT hijo %info_familia;>
```

```
<!ELEMENT hija %info_familia;>
```

```
<!ELEMENT nombre (#PCDATA)>
```

```
<!ELEMENT apellido (#PCDATA)>
```

```
<!ELEMENT dni (#PCDATA)>
```

```
<!ELEMENT edad (#PCDATA)>
```

```
<!ATTLIST familia codigo ID #REQUIRED>
```

```
<!ATTLIST hijo pos NMTOKEN #REQUIRED>
```

```
<!ATTLIST hija pos NMTOKEN #REQUIRED>
```

xsd: familia

```
<?xml version="1.0"?>

<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <!--declarar raiz-->
  <xs:element name="familia">
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:element ref="padre"/></xs:element>
          <xs:element ref="madre"/></xs:element>
        </xs:choice>
        <xs:element ref="hijos"/></xs:element>
      </xs:sequence>
      <xs:attribute name="codigo" use="required"/></xs:attribute>
    </xs:complexType>
  </xs:element>

  <!--atributo de raiz-->
  <xs:attribute name="codigo">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="[A-Z]\d{2}\-\d{2}"></xs:pattern>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>

  <!--declarar hijos-->
  <xs:element name="hijos">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:sequence>
          <xs:element ref="hijo"/></xs:element>
          <xs:element ref="hija"/></xs:element>
        </xs:sequence>
      </xs:choice>
    </xs:complexType>
  </xs:element>
```

```

<!--relacion padre e hijo / max y min-->
<xs:complexType name="datos_padres">
  <xs:sequence>
    <xs:element ref="nombre"></xs:element>
    <xs:element ref="apellido" minOccurs="2" maxOccurs="2"></xs:element>
    <xs:element ref="dni" minOccurs="0"></xs:element>
    <xs:element ref="edad" minOccurs="0"></xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="datos_hijos">
  <xs:complexContent>
    <xs:extension base="datos_padres">
      <xs:attribute ref="pos" use="required"></xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!--relaciona los datos con la familia-->
<xs:element name="padre" type="datos_padres"/>
<xs:element name="madre" type="datos_padres"/>
<xs:element name="hijo" type="datos_hijos"/>
<xs:element name="hija" type="datos_hijos"/>

<!--atributo hijos-->
<xs:attribute name="pos">
  <xs:simpleType>
    <xs:restriction base="xs:positiveInteger">
      <xs:maxInclusive value="30"></xs:maxInclusive>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>

<!--declarar la longitud-->
<xs:simpleType name="texto30">
  <xs:restriction base="xs:string">
    <xs:maxInclusive value="texto30"/>
  </xs:restriction>
</xs:simpleType>

```

```

<!--valores finales-->
<xs:element name="nombre" type="texto30"></xs:element>
<xs:element name="apellido" type="texto30"></xs:element>
<xs:element name="dni">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{2}\.\d{3}\.\d{3}\-[A-Z]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="edad" type="xs:positiveInteger"></xs:element>
s:schema>

```

=====

xml almacén

```

<?xml version="1.0" encoding="UTF-8"?>
<!--DOCTYPE bdProductos SYSTEM "almaceninformatica.dtd"-->
<!--Hay 10 tipos de descuentos
-El precio es un decimal con dos decimales
-Categoría puede ser A,B, C, D,E -->

<bdProductos codigo="1" xsi:noNamespaceSchemaLocation="bdProductos.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <producto>
    <nombre>Teclado</nombre>
    <codigo>C001</codigo>
    <categoria>A</categoria>
    <iva>21</iva>
    <precio>15.00</precio>
    <descuento>7</descuento>
  </producto>

  <producto>
    <nombre>Monitor</nombre>
    <codigo>C002</codigo>
    <categoria>B</categoria>
    <iva>21</iva>
    <precio>150.00</precio>
    <descuento>10</descuento>
  </producto>

</bdProductos>

```


dtd

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

<!ELEMENT bdProductos (producto)+>
<!--ATTLIST bdProductos codigo NMTOKEN #REQUIRED-->
<!ELEMENT producto (nombre, codigo, categoria,iva,precio,descuento)>
<!ELEMENT nombre (#PCDATA)>
<!--ELEMENT codigo (#PCDATA)-->
<!--ELEMENT categoria (#PCDATA)-->
<!--ELEMENT iva (#PCDATA)-->
<!--ELEMENT precio (#PCDATA)-->
<!--ELEMENT descuento (#PCDATA)-->
```

xsd

```
<?xml version="1.0"?>

<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xs:element name="bdProductos">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="producto" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="codigo" type="xs:integer"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="producto">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nombre" type="xs:string"/>
        <xs:element name="codigo" type="Tlongitudentre"/>
        <xs:element name="categoria" type="Tcat1"/>
        <xs:element name="iva" type="xs:integer"/>
        <xs:element name="precio" type="Tdecimal"/>
        <xs:element name="descuento" type="Tdescuento"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```

<xs:simpleType name="Tdescuento">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="10"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="Tdecimal">
  <xs:restriction base="xs:decimal">
    <xs:totalDigits value="4"/>
    <xs:fractionDigits value="2"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="Tcat1">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-E]"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="Tlongitudentre">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>

/xs:schema>

```