TP Nº 5: Normalización

Ejercicio 1

Una inmobiliaria almacena los datos de los inmuebles en un documento XML. Los datos están compuestos por un identificador del inmueble, el nombre del barrio, el código postal, un identificador para el barrio, la cantidad de metros cuadrados y el valor del metro cuadrado para el barrio en que se encuentra.

Restricciones:

- El valor del metro cuadrado es único para un determinado barrio.
- Los identificadores, tanto del inmueble como del barrio son únicos en todo el documento XML.
- Dado un código postal, el barrio es único, pero dado un barrio, los códigos postales válidos son múltiples.

Un documento XML esperado que responda a la estructura deseada sería:

```
<?xml version="1.0"?>
<inmuebles>
    <inmueble identificador="inmueble1">
        <barrio identificador="barrio1">
            <nombre></nombre>
            <codigoPostal></codigoPostal>
            <precioM2></precioM2>
        </barrio>
        <area></area>
    </inmueble>
    .
    .
    .
    <inmueble identificador="inmuebleN">
        <barrio identificador="barrio1">
            <nombre></nombre>
            <codigoPostal></codigoPostal>
            <precioM2></precioM2>
        </barrio>
        <area></area>
    </inmueble>
</inmuebles>
```

El cual es validado por el siguiente DTD:

```
<!ELEMENT inmuebles (inmueble)*>
<!ELEMENT inmueble (barrio, area)>
<!ELEMENT barrio (nombre, codigoPostal, precioM2)>
<!ELEMENT area (#PCDATA)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT codigoPostal (#PCDATA)>
<!ELEMENT precioM2 (#PCDATA)>
<!ATTLIST inmueble identificador ID #REQUIRED>
<!ATTLIST barrio identificador CDATA #REQUIRED>
```

1.1) ¿Se validan todas las restricciones? Mostrar un contraejemplo en los casos que no se validen.

Veamos cada una de las restricciones:

- El valor del metro cuadrado es único para un determinado barrio.
  Esta restricción no se valida.
  Veamos un contraejemplo:

```
<?xml version="1.0"?>
<inmuebles>
	<inmueble identificador="N2">
		<barrio identificador="B1">
			<nombre>Recoleta</nombre>
			<codigoPostal>1061</codigoPostal>
			<precioM2>2100</precioM2>
		</barrio>
		<area>120</area>
	</inmueble>
	<inmueble identificador="N1">
		<barrio identificador="B2">
			<nombre>San Telmo</nombre>
			<codigoPostal>1800</codigoPostal>
			<precioM2>1500</precioM2>
		</barrio>
		<area>50</area>
	</inmueble>
	<inmueble identificador="N3">
		<barrio identificador="B3">
			<nombre>San Telmo</nombre>
			<codigoPostal>1800</codigoPostal>
			<precioM2>1400</precioM2>
		</barrio>
		<area>50</area>
	</inmueble>
</inmuebles>
```

  En este xml, el barrio de San Telmo está presente dos veces, y cada una de las veces con valor diferente para el atributo precio de metro cuadrado.

- Los identificadores del inmueble como del barrio son únicos en todo el documento XML, pero no necesariamente ocurre lo mismo con los identificadores

de barrio dado que no son IDs, con lo cual puede existir un mismo barrio con distintos valores para el atributo 'identificador'.
Veamos un contraejemplo:

```
<?xml version="1.0"?>
<inmuebles>
	<inmueble identificador="N1">
		<barrio identificador="B1">
			<nombre>Recoleta</nombre>
			<codigoPostal>1061</codigoPostal>
			<precioM2>2100</precioM2>
		</barrio>
		<area>120</area>
	</inmueble>
	<inmueble identificador="N2">
		<barrio identificador="B2">
			<nombre>San Telmo</nombre>
			<codigoPostal>1800</codigoPostal>
			<precioM2>1500</precioM2>
		</barrio>
		<area>50</area>
	</inmueble>
	<inmueble identificador="N3">
		<barrio identificador="B3">
			<nombre>San Telmo</nombre>
			<codigoPostal>1800</codigoPostal>
			<precioM2>1400</precioM2>
		</barrio>
		<area>50</area>
	</inmueble>
</inmuebles>
```

- Dado un código postal, el barrio es único, pero dado un barrio, los códigos postales válidos son múltiples.
  Esta restricción no se valida.
  Veamos un contraejemplo:

```
<?xml version="1.0"?>
<inmuebles>
	<inmueble identificador="N1">
		<barrio identificador="B1">
			<nombre>Recoleta</nombre>
			<codigoPostal>1600</codigoPostal>
			<precioM2>2100</precioM2>
		</barrio>
		<area>120</area>
	</inmueble>
	<inmueble identificador="N2">
		<barrio identificador="B2">
			<nombre>San Telmo</nombre>
			<codigoPostal>1600</codigoPostal>
			<precioM2>1500</precioM2>
		</barrio>
		<area>50</area>
	</inmueble>
	<inmueble identificador="N3">
		<barrio identificador="B3">
			<nombre>San Telmo</nombre>
```

```
                    <codigoPostal>1800</codigoPostal>
                    <precioM2>1400</precioM2>
                </barrio>
                <area>50</area>
            </inmueble>
</inmuebles>
```

El código postal de San Telmo aparece con dos valores diferentes.

1.2) ¿Qué sucede si se borra el único inmueble para un determinado barrio?

Si se borrara el único inmueble para determinado barrio, perderíamos los datos del barrio.

1.3) Sugerir una nueva estructura para el documento XML y un nuevo DTD para que se validen todas las restricciones.

Para solucionar las anomalías de redundancia y actualización, proponemos separar el contenido del xml en tres listas: una de inmuebles, otra de barrios y una de códigos postales, y luego referenciar desde el inmueble el código postal al que pertenece y desde el código postal al barrio.

```
<?xml version="1.0"?>
<!DOCTYPE inmuebles SYSTEM "inmuebles.dtd">
<inmuebles>
        <inmueble identificador="N1" codigoPostalId="C1">
                <area>120</area>
        </inmueble>
        <inmueble identificador="N2" codigoPostalId="C2">
                <area>50</area>
        </inmueble>
        <inmueble identificador="N3" codigoPostalId="C1">
                <area>50</area>
        </inmueble>
        <codigoPostal identificador="C1" barrioId="B1">
                <numero>1200</ numero >
        </ codigoPostal >
        <codigoPostal identificador="C2" barrioId="B1">
                <numero>1400</ numero >
        </ codigoPostal >
        <barrio identificador="B1">
                <nombre>Recoleta</nombre>
                <precioM2>2100</precioM2>
        </barrio>
        <barrio identificador="B2">
                <nombre>San Telmo</nombre>
                <precioM2>1500</precioM2>
        </barrio>
</inmuebles>

<!ELEMENT inmuebles (inmueble*, codigoPostal*, barrio*)>
<!ELEMENT inmueble (area)>
<!ELEMENT codigoPostal (numero)>
<!ELEMENT barrio (nombre, precioM2)>
<!ELEMENT area (#PCDATA)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT numero (#PCDATA)>
<!ELEMENT precioM2 (#PCDATA)>
<!ATTLIST inmueble identificador ID #REQUIRED>
<!ATTLIST inmueble codigoPostalId IDREF #REQUIRED>
<!ATTLIST barrio identificador ID #REQUIRED>
<!ATTLIST codigoPostal identificador ID #REQUIRED>
<!ATTLIST codigoPostal barrioId IDREF #REQUIRED>
```

Ejercicio 2

2.1) Primero vamos a construir el xml schema que tenga el mismo funcionamiento que el DTD del ejercicio 1.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="inmuebles">
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="inmueble" type="imType" maxOccurs="unbounded"/>
                        <xs:element name="codigoPostal" type="codType" maxOccurs="unbounded"/>
                        <xs:element name="barrio" type="barrioType" maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
 </xs:element>
 <xs:complexType name="imType">
        <xs:sequence>
                <xs:element name="area" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="identificador" type="xs:ID" use="required"/>
        <xs:attribute name="codigoPostalId" type="xs:IDREF" use="required"/>
 </xs:complexType>
 <xs:complexType name="codType">
        <xs:sequence>
                <xs:element name="numero" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="identificador" type="xs:ID" use="required"/>
        <xs:attribute name="barrioId" type="xs:IDREF" use="required"/>
 </xs:complexType>
 <xs:complexType name="barrioType">
        <xs:sequence>
                <xs:element name="nombre" type="xs:string"/>
                <xs:element name="precioM2" type="xs:decimal"/>
        </xs:sequence>
        <xs:attribute name="identificador" type="xs:ID" use="required"/>
 </xs:complexType>
</xs:schema>
```

En este esquema, al igual que en DTD, podría ocurrir que por ejemplo, en el atributo que debe referenciar a un código postal, referenciemos otra cosa como un inmueble

```
<?xml version="1.0"?>
<!DOCTYPE inmuebles SYSTEM "inmuebles.dtd">
<inmuebles>
        <inmueble identificador="N1" codigoPostalId="C1">
                <area>120</area>
        </inmueble>
        <inmueble identificador="N2" codigoPostalId="N3">
                <area>50</area>
        </inmueble>
        <inmueble identificador="N3" codigoPostalId="C1">
                <area>50</area>
        </inmueble>
        <codigoPostal identificador="C1" barrioId="B1">
                <numero>1200</ numero >
        </ codigoPostal >
        <codigoPostal identificador="C2" barrioId="B1">
                <numero>1400</ numero >
        </ codigoPostal >
        <barrio identificador="B1">
                <nombre>Recoleta</nombre>
                <precioM2>2100</precioM2>
        </barrio>
        <barrio identificador="B2">
                <nombre>San Telmo</nombre>
                <precioM2>1500</precioM2>
        </barrio>
</inmuebles>
```

Para que esto no suceda, debemos declarar las key y keyref como se indica a continuación:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="inmuebles">
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="inmueble" type="imType" maxOccurs="unbounded"/>
                        <xs:element name="codigoPostal" type="codType" maxOccurs="unbounded"/>
                        <xs:element name="barrio" type="barrioType" maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <xs:key name="PK_barrio">
                <xs:selector xpath=".//barrio"/>
                <xs:field xpath="@identificador"/>
        </xs:key>
        <xs:key name="PK_cp">
                <xs:selector xpath=".//codigoPostal"/>
                <xs:field xpath="@identificador"/>
        </xs:key>
        <xs:keyref name="FK_cp" refer="PK_barrio">
                <xs:selector xpath=".//codigoPostal"/>
                <xs:field xpath="@barrioId"/>
        </xs:keyref>
        <xs:keyref name="FK_im" refer="PK_cp">
                <xs:selector xpath=".//inmueble"/>
                <xs:field xpath="@codigoPostalId"/>
        </xs:keyref>
 </xs:element>
 <xs:complexType name="imType">
        <xs:sequence>
                <xs:element name="area" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="identificador" type="xs:ID" use="required"/>
        <xs:attribute name="codigoPostalId" type="xs:IDREF" use="required"/>
 </xs:complexType>
 <xs:complexType name="codType">
        <xs:sequence>
                <xs:element name="numero" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="identificador" type="xs:ID" use="required"/>
        <xs:attribute name="barrioId" type="xs:IDREF" use="required"/>
 </xs:complexType>
 <xs:complexType name="barrioType">
        <xs:sequence>
                <xs:element name="nombre" type="xs:string"/>
                <xs:element name="precioM2" type="xs:decimal"/>
        </xs:sequence>
        <xs:attribute name="identificador" type="xs:ID" use="required"/>
 </xs:complexType>
</xs:schema>
```

Observar que tanto **key** como **keyref** se encuentran dentro del **element** que declara

Ejercicio 3

Un servicio de fletes almacena los datos de sus mudanzas en un documento XML llamado **flete.xml**, que tiene una raíz única (el elemento *flete*), con varios elementos *mudanza* (que indican el servicio de flete para una fecha y barrio determinados), seguidos de elementos *item* (que representan los objetos que pueden ser transportados).

flete.xml

```
<flete>
	<mudanza>
		<barrio>Palermo</barrio>
		<fecha>2022-03-01</fecha>
		<elemento item="c01">10</elemento>
		<elemento item="c03">4</elemento>
		<elemento item="m01">2</elemento>
	</mudanza>
	<mudanza>
		<barrio>Villa Devoto</barrio>
		<fecha>2022-02-26</fecha>
		<elemento item="c02">5</elemento>
		<elemento item="c03">6</elemento>
		<elemento item="m02">8</elemento>
		<elemento item="c01">1</elemento>
	</mudanza>
	<mudanza>
		<barrio>Palermo</barrio>
		<fecha>2022-02-10</fecha>
		<elemento item="c01">10</elemento>
		<elemento item="c01">10</elemento>
		<elemento item="m01">5</elemento>
	</mudanza>
	<item codigo="c01">
		<nombre>caja chica</nombre>
		<precio>1000</precio>
	</item>
	<item codigo="c02">
		<nombre>caja mediana</nombre>
		<precio>2000</precio>
	</item>
	<item codigo="c03">
		<nombre>caja grande</nombre>
		<precio>3000</precio>
	</item>
	<item codigo="m01">
		<nombre>mueble chico</nombre>
	</item>
	<item codigo="m02">
		<precio>2500</precio>
	</item>
</flete>
```

Este documento XML, a su vez, debe cumplir con las siguientes restricciones:

A. Cada ítem tiene un código que lo identifica (y que no puede repetirse para otro ítem).
B. Dados una fecha y un barrio, la mudanza debe ser única (es decir, que no puede haber dos tags mudanza con el mismo barrio y la misma fecha a la vez).
C. Cada tag *elemento* que se use en una mudanza debe referirse a un *item* existente (es decir, que el código del tag *elemento* debe ser el código de un *item* válido).

Se pide realizar el XML Schema **flete.xsd** que valide el XML **flete.xml** que cumpla con las restricciones y estructura dadas, declarando las claves y referencias necesarias

flete.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="flete">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="mudanza" type="mudanzaType" maxOccurs="unbounded"/>
        <xs:element name="item" type="itemType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
    <xs:key name="PK_mudanza">
      <xs:selector xpath=".//mudanza"/>
      <xs:field xpath="barrio"/>
      <xs:field xpath="fecha"/>
    </xs:key>
    <xs:key name="PK_item">
      <xs:selector xpath=".//item"/>
      <xs:field xpath="@codigo"/>
    </xs:key>
    <xs:keyref name="FK_itemElemento" refer="PK_item">
        <xs:selector xpath=".//mudanza/elemento"/>
        <xs:field xpath="@item"/>
    </xs:keyref>
  </xs:element>
  <xs:complexType name="mudanzaType">
    <xs:sequence>
      <xs:element name="barrio" type="xs:string"/>
      <xs:element name="fecha" type="xs:date"/>
      <xs:element name="elemento" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:int">
              <xs:attribute name="item" type="xs:int"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>

      </xs:element>
    </xs:sequence>

  </xs:complexType>

  <xs:complexType name="itemType">
    <xs:choice>
      <xs:sequence>
        <xs:element name="nombre" type="xs:string"/>
```

```
            <xs:element name="precio" type="xs:decimal" minOccurs="0"/>
          </xs:sequence>
          <xs:sequence>
            <xs:element name="precio" type="xs:decimal"/>
          </xs:sequence>
    </xs:choice>
    <xs:attribute name="codigo" type="xs:int" use="required"/>
  </xs:complexType>
</xs:schema>
```