

## Ejercicio 1

---

Construir el vocabulario para el documento XML que utiliza esta empresa para gestionar los pedidos utilizando un DTD externo:

### pedido.dtd

```
<!ELEMENT pedidos (sucursal+)>
<!ELEMENT sucursal (direccion+,trabajador,fecha,observacion?,articulo+)>
<!ATTLIST sucursal codigo CDATA #REQUIRED>
<!ELEMENT direccion (#PCDATA)>
<!ELEMENT trabajador (#PCDATA)>
<!ELEMENT fecha (#PCDATA)>
<!ELEMENT observacion (#PCDATA)>
<!ELEMENT articulo (unidades,precio,observaciones?,plazo)>
<!ATTLIST articulo codigo CDATA #REQUIRED>
<!ELEMENT unidades (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT observaciones (#PCDATA)>
<!ELEMENT plazo (#PCDATA)>
```

## Ejercicio 2

---

Construir ese mismo vocabulario utilizando el lenguaje XMLSchema debidamente documentado con fecha de creación, nombre del autor y utilidad del esquema.

### pedido.xsd

```
<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema" >
  <xsd:annotation>
    <xsd:documentation xml:lang="Es-es">
      Esquema para la realización de pedidos de la empresa Distanza S.A.
    </xsd:documentation>
    <fecha>2017-03-20</fecha>
    <autor>Adrian Alvarez Lois</autor>
  </xsd:annotation>
  <xsd:element name="pedidos" type="pedidoDatos"/>
  <xsd:complexType name="pedidoDatos">
    <xsd:sequence>
      <xsd:element name="sucursal" type="sucursalDatos" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="sucursalDatos">
    <xsd:sequence>
      <xsd:element name="direccion" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="trabajador" type="xsd:string"/>
      <xsd:element name="fecha" type="xsd:date" />
      <xsd:element name="observacion" type="xsd:string" minOccurs="0" />
      <xsd:element name="articulo" type="articuloDatos" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="codigo" type="codigoTipo" use="required" />
  </xsd:complexType>
  <xsd:simpleType name="codigoTipo">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[A-Za-z]([A-Za-z0-9]){7}" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="articuloDatos">
    <xsd:sequence>
      <xsd:element name="unidades">
        <xsd:simpleType>
          <xsd:restriction base="xsd:positiveInteger">
            <xsd:minInclusive value="1" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="precio">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:minInclusive value="1" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="observaciones" type="xsd:string" />
<xsd:element name="plazo">
  <xsd:simpleType>
    <xsd:restriction base="xsd:positiveInteger">
      <xsd:minInclusive value="1" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="codigo" type="codigoArticulo" use="required" />
</xsd:complexType>
<xsd:simpleType name="codigoArticulo">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[A-Z]{3}-[0-9]{3}" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

## Ejercicio 3

Realizar un fichero XML que se corresponda con una instancia del vocabulario diseñado y asociarle al DTD.

### pedido.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pedidos SYSTEM "pedido.dtd" >
<pedidos>
  <sucursal codigo="A0000001">
    <direccion>C/Castilla, 1</direccion>
    <direccion>C/Andalucia, 2</direccion>
    <trabajador>Adrian Alvarez Lois</trabajador>
    <fecha>20-03-2017</fecha>
    <observacion>Urgente</observacion>
    <articulo codigo="AAA-111">
      <unidades>12</unidades>
      <precio>100</precio>
      <observaciones>Producto muy frágil</observaciones>
      <plazo>10</plazo>
    </articulo>
    <articulo codigo="AAA-112">
      <unidades>10</unidades>
      <precio>10</precio>
      <observaciones>No apilar más de cinco paquetes</observaciones>
      <plazo>8</plazo>
    </articulo>
  </sucursal>
  <sucursal codigo="B00000002">
    <direccion>Avda. Gran Vía s/n</direccion>
    <trabajador>Andrea Lopez-Doriga</trabajador>
    <fecha>10-02-2012</fecha>
    <articulo codigo="ABC-001">
      <unidades>5</unidades>
      <precio>50</precio>
      <observaciones />
      <plazo>3</plazo>
    </articulo>
  </sucursal>
</pedidos>
```

## Ejercicio 4

---

Modificar ese fichero XML para asociarle al esquema diseñado.

```
<?xml version="1.0" encoding="UTF-8"?>
<pedidos
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation = "pedido.xsd" >
  <sucursal codigo="A0b00R01">
    <direccion>C/Castilla, 1</direccion>
    <direccion>C/Andalucia, 2</direccion>
    <trabajador>Adrian Alvarez Lois</trabajador>
    <fecha>2017-03-20</fecha>
    <observacion>Urgente</observacion>
    <articulo codigo="AAA-111">
      <unidades>12</unidades>
      <precio>100</precio>
      <observaciones>Producto muy frágil</observaciones>
      <plazo>10</plazo>
    </articulo>
    <articulo codigo="AAA-112">
      <unidades>10</unidades>
      <precio>10</precio>
      <observaciones>No apilar más de cinco paquetes</observaciones>
      <plazo>8</plazo>
    </articulo>
  </sucursal>
  <sucursal codigo="B0000002">
    <direccion>Avda. Gran Vía s/n</direccion>
    <trabajador>Andrea Lopez-Doriga</trabajador>
    <fecha>2017-03-21</fecha>
    <articulo codigo="ABC-001">
      <unidades>5</unidades>
      <precio>50</precio>
      <observaciones />
      <plazo>3</plazo>
    </articulo>
  </sucursal>
</pedidos>
```

## Anotaciones

---

Todos los ejercicios están desarrollados utilizando el IDE NetBeans en el que descargué e instalé un plugin para la validación de documentos XML, el cual ofrece validaciones XML, DTD y XSD, y que me han sido de mucha utilidad para la realización de los presentes ejercicios.

Todos los códigos están completamente validados y conforme a las especificaciones dadas.