

Validación por Esquema

A. Crea un fichero XML para almacenar los datos de un cliente

- Nombre : 5-30 caracteres.
- saldo: numérico flotante
- Numero compras: nº entero. ≥ 0
- email. Validación de email
- teléfono. 9 dígitos
- IBAN

```
<cliente>
  <nombre>Binta</nombre>
  <saldo>1992.56</saldo>
  <compras>0</compras>
  <email>bd90962@gmail.com</email>
  <telefono>643467898</telefono>
  <iban>ES91 2100 1234 56 0078901234</iban>
</cliente>

</usuario>
```

B. Describe un esquema para validar el fichero.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="usuario">
    <xs:complexType>
      <xs:sequence>

        <xs:element name="cliente">
          <xs:complexType>
            <xs:sequence>

              <!-- ELEMENTO NOMBRE -->
              <xs:element name="nombre">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <!-- LONGITUD MÁXIMA Y MÍNIMA -->
                    <xs:minLength value="5"/>
                    <xs:maxLength value="30"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>

              <!-- ELEMENTO SALDO -->
              <xs:element name="saldo" type="xs:float"/>

              <!-- ELEMENTO COMPRAS -->
              <xs:element name="compras">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minInclusive value="0"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Validación por Esquema

```
<!-- VALIDACIÓN DE EMAIL -->
<xs:element name="email">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<!-- ELEMENTO TELEFONO -->
<xs:element name="telefono">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="9" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<!-- ELEMENTO IBAN -->
<xs:element name="iban">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="ES[0-9]{2}[ ]*[0-9]{4}[ ]*[0-9]{4}[ ]*[0-9]{2}[ ]*[0-9]{10}" /></xs:pattern>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

</xs:sequence>
</xs:complexType>
</xs:element>
```

C. Validar el fichero XML frente a su esquema.

Referenciando el esquema externo para su validación:

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="clientes.xsd">

  <cliente>
    <nombre>Binta</nombre>
    <saldo>1992.56</saldo>
    <compras>0</compras>
    <email>bd90962@gmail.com</email>
    <telefono>643467898</telefono>
    <iban>ES91 2100 1234 56 0078901234</iban>
  </cliente>

</usuario>
```

Validación por Esquema

D. Modifica valores para provocar el incumplimiento de los requisitos

Prueba 1: nombre

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="clientes.xsd">

  <cliente>
    <nombre>María de los Ángeles García de la Vega</nombre>
    <!-- El nombre el límite de 30 caracteres -->
    <saldo>1992.56</saldo>
    <compras>0</compras>
    <email>bd90962@gmail.com</email>
    <telefono>643467898</telefono>
    <iban>ES91 2100 1234 56 0078901234</iban>
  </cliente>

</usuario>
```

Prueba 2: saldo

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="clientes.xsd">

  <cliente>
    <nombre>Binta</nombre>
    <saldo>mil novecientos noventa y dos</saldo>
    <!-- Solo permite valores tipo float -->
    <compras>0</compras>
    <email>bd90962@gmail.com</email>
    <telefono>643467898</telefono>
    <iban>ES91 2100 1234 56 0078901234</iban>
  </cliente>

</usuario>
```

Prueba 3: compras

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="clientes.xsd">

  <cliente>
    <nombre>Binta</nombre>
    <saldo>1992.56</saldo>
    <compras>-3</compras>
    <!-- La cantidad de compras realizadas no puede ser menor
    a cero -->
    <email>bd90962@gmail.com</email>
    <telefono>643467898</telefono>
    <iban>ES91 2100 1234 56 0078901234</iban>
  </cliente>

</usuario>
```

Validación por Esquema

Prueba 4: email

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="clientes.xsd">
  <cliente>
    <nombre>Binta</nombre>
    <saldo>1992.56</saldo>
    <compras>0</compras>
    <email>bd90962@gmail.com</email>
    <!-- Es obligatoria la aparición del punto -->
    <telefono>643467898</telefono>
    <iban>ES91 2100 1234 56 0078901234</iban>
  </cliente>
</usuario>
```

Prueba 5: teléfono

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="clientes.xsd">
  <cliente>
    <nombre>Binta</nombre>
    <saldo>1992.56</saldo>
    <compras>0</compras>
    <email>bd90962@gmail.com</email>
    <telefono>643467</telefono>
    <!-- La longitud ha de ser, estrictamente, de nueve caracteres -->
    <iban>ES91 2100 1234 56 0078901234</iban>
  </cliente>
</usuario>
```

Prueba 6: IBAN

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="clientes.xsd">
  <cliente>
    <nombre>Binta</nombre>
    <saldo>1992.56</saldo>
    <compras>0</compras>
    <email>bd90962@gmail.com</email>
    <telefono>643467898</telefono>
    <iban>ES91 2100 56 0078901234</iban>
    <!-- Formato inválido. Debe seguir el patrón:
    ESXX XXXX XXXX XX XXXXXXXXXX (la aparición de
    espacios no es obligatoria) -->
  </cliente>
</usuario>
```