

# XML Schema (XSD) Cheatsheet: Casos Representativos y Patrones Indispensables

## 1. Esquema Básico

- **Ejemplo de esquema simple con elementos secuenciales:**

```
xml
Copiar código
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://example.com"
            elementFormDefault="qualified">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

---

## 2. Tipos Simples

- **Definición básica de un elemento:**

```
xml
Copiar código
<xs:element name="start_date" type="xs:date"/>
```

- **Restricciones con Facets:**

```
xml
Copiar código
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

---

## 3. Enumeración y Patrones

- **Enumeración:**

```
xml
Copiar código
<xs:simpleType name="carType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="BMW"/>
    <xs:enumeration value="Tesla"/>
  </xs:restriction>
</xs:simpleType>
```

```
</xs:restriction>
</xs:simpleType>
```

- **Patrones con expresiones regulares:**

```
xml
Copiar código
<xs:simpleType name="phoneNumber">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d{3}-\d{3}-\d{4}"/>
  </xs:restriction>
</xs:simpleType>
```

---

## 4. WhiteSpace

- **Control de espacios en blanco:**

```
xml
Copiar código
<xs:element name="text">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="collapse"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

---

## 5. Tipos Complejos

- **Elementos con subelementos:**

```
xml
Copiar código
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

- **Elementos con texto y atributos:**

```
xml
Copiar código
<xs:complexType name="product">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="id" type="xs:positiveInteger"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

---

## 6. Creación de Tipos Personalizados

- **Tipos simples personalizados:**

```
xml
Copiar código
<xs:simpleType name="nombreCorto">
  <xs:restriction base="xs:string">
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="nombre" type="nombreCorto"/>
```

- **Tipos complejos personalizados:**

```
xml
Copiar código
<xs:complexType name="direccionType">
  <xs:sequence>
    <xs:element name="calle" type="xs:string"/>
    <xs:element name="ciudad" type="xs:string"/>
    <xs:element name="codigoPostal" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="direccion" type="direccionType"/>
```

---

## 7. Uso de Referencias (**ref**)

- **Definición global del elemento referenciado:**

```
xml
Copiar código
<xs:element name="direccion">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="calle" type="xs:string"/>
      <xs:element name="ciudad" type="xs:string"/>
      <xs:element name="codigoPostal" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

- **Uso de **ref** en otro tipo complejo:**

```
xml
Copiar código
<xs:complexType name="personaType">
  <xs:sequence>
    <xs:element ref="direccion"/>
    <xs:element name="nombre" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="persona" type="personaType"/>
```

---

## 8. Definición de Elementos Después de `ref`

- **Referencia y elementos adicionales:**

```
xml
Copiar código
<xs:complexType name="personaExtendidaType">
  <xs:sequence>
    <xs:element ref="direccion"/>
    <xs:element name="email" type="xs:string"/>
    <xs:element name="telefono" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:positiveInteger" use="required"/>
</xs:complexType>

<xs:element name="personaExtendida" type="personaExtendidaType"/>
```

---

## 9. Tipos Simples con Atributos

- **Definición de un tipo simple con atributos:**

```
xml
Copiar código
<xs:complexType name="simpleTypeWithAttribute">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="lang" type="xs:string" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:element name="greeting" type="simpleTypeWithAttribute"/>
```

### Uso en XML:

```
xml
Copiar código
<greeting lang="en">Hello</greeting>
```

- **Restricciones en atributos:**

```
xml
Copiar código
<xs:attribute name="lang">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="en"/>
      <xs:enumeration value="es"/>
      <xs:enumeration value="fr"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

---

## 10. Indicadores de Grupo

- **Opcional (choice):**

```
xml
Copiar código
<xs:complexType name="contactInfo">
  <xs:choice>
    <xs:element name="email" type="xs:string"/>
    <xs:element name="phone" type="xs:string"/>
  </xs:choice>
</xs:complexType>
```

- **Cualquier orden, máximo una vez (all):**

```
xml
Copiar código
<xs:complexType>
  <xs:all>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:all>
</xs:complexType>
```

---

## 11. Multiplicidad

- **Opcional (minOccurs y maxOccurs):**

```
xml
Copiar código
<xs:element name="nickname" type="xs:string" minOccurs="0"
maxOccurs="1"/>
```

- **Ilimitado (unbounded):**

```
xml
Copiar código
<xs:element name="hobby" type="xs:string" maxOccurs="unbounded"/>
```

---

## 12. Claves y Relaciones

- **Unicidad (unique):**

```
xml
Copiar código
<xs:unique name="uniqueID">
  <xs:selector xpath="person"/>
  <xs:field xpath="@id"/>
</xs:unique>
```

- **Claves y referencias (key y keyref):**

xml

Copiar código

```
<xs:key name="personID">
  <xs:selector xpath="person"/>
  <xs:field xpath="@id"/>
</xs:key>

<xs:keyref name="refPersonID" refer="personID">
  <xs:selector xpath="employee"/>
  <xs:field xpath="@personID"/>
</xs:keyref>
```