

# XML eskemen definizioa (XML Schema Definition)

Dokumentu motaren definizioa (DTD) formalizatutako lehen estandarra izan zen, baina kasu gehienetan, XML eskemen definizioak (XSD) ordezkatu du.

XML eskema batek zehaztasun-maila handiagoa ahalbidetzen du baliozkotze-arauak ezartzerakoan, DTD-ekin lortu ezin den zehetasun-maila lortzen, izan ere, azken honek ez bait du aukerarik ematen elementu edo atributu konplexuagoak zehazteko, ezta beste murrizketa konplexuago batzuk gehitzeko ere.

DTDen gaineko abantailak:

- Eskema bat XML dokumentu bat da eta, beraz, ondo osatuta dagoela balioztatu daiteke.
- Aurrez definitutako datu eta atributu mota asko ditu, eta hauetatik hedatuz mota berriak sortu egin daitezke.
- Zehaztasunez zehaztu daiteke elementu baten kardinaltasuna, hau da, dokumentu batean zenbat aldiz agertu ahal den.
- Hainbat izen espazio erabiltzeko aukera ematen du, horri esker, hainbat hiztegi edo etiketa-joko nahastu daitezke.

## XML esquema baten adierazpena

XML eskemak XML dokumentuak dira, baina kasu honetan *.xsd* fitxategi-luzapena duten dokumentuak dira, hau dela eta XSD dokumentuak deitzen zaie ere.

Horrela, eskema guztiek XML adierazpen batekin hasi behar du.

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

XML dokumentu guztiek bezala, XML Schema dokumentu batek erro elementu bat izan behar du. XSD dokumentu baten erro elementua `<schema>` da.

```
<xs:schema></xs:schema>
```

XML eskema bat sortzerakoan, XML Schema-ren estandarrean zehaztutako elementuak eta atributuak erabiliko dira. Elementu eta atributu hauek W3Ck definitutako izenen espazioan definituta daude: <http://www.w3.org/2001/XMLSchema>.

Definizio horiek XML dokumentuetan erabili ahal izateko, horren erreferentzia aipatu behar da, beraz, `<schema>` elementuak eskemarako izenen espazioa zehazten duen `xmlns` atributua barne hartzen du.

`xmlns` atributuaren sintaxi generikoa honako hau da:

```
xmlns:aurrizkia ="IZEN_ESPAZIOAREN_URIA"
```

Normalean `xs` aurritzkiarekin lotzen da, nahiz eta `xsd` aurritzia aurkitzea ere ohikoa izaten da.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Aurreko sententzian `xmlns:xs` formatuak `xs` aurritzia daramaten elementu edo atributu guztiak URIan zehaztutako izenen espaziokoak direla adierazten du.

## XML eskema baten erreferentzia

XML dokumentuetan, XSD baten erreferentzia XML dokumentuaren erroko elementuan dauden atributuen bidez egiten da.

Atributu horiek W3Ck definitutako izenen espazioan definituta daude:

<http://www.w3.org/2001/XMLSchema-instance>.

XML definizio horiek dokumentuetan erabili ahal izateko izen espazioari erreferentzia egin behar zaio; beraz, XML dokumentuaren erro-elementuak eskemarako izenen espazioa zehazten duen `xmlns` atributua barne hartzen du.

Normalean, izenen espazio horri `xsi` aurrizkia esleitzen zaio (nahiz eta edozein aurrizki erabil daitezkeen).

Izenen espazioaren erreferentzia honela geratuko litzateke:

```
<rootElement xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

Aurreko adierazpenarekin `xsi:schemaLocation` eta `xsi:noNamespaceSchemaLocation` atributuak erabil daitezke, XML bat ebaluatzeko erabil daitezkeen eskema-dokumentuen kokapen fisikoari buruzko iradokizunak ematen dituztenak.

Jarraian, eskemaren kokapena zehaztu behar da aurreko atributuak erabiliz: URL bat edo tokiko sarbideko ibilbide bat izan daiteke,.

- Eskema gune publiko batean badago:

```
<rootElement
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.com/namespace path/to/schema.xsd"
>
  <!-- rest of the document goes here -->
</rootElement>
```

- En caso de que el esquema esté en local (nuestro caso):

```
<rootElement
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="path/to/schema.xsd"
>
  <!-- rest of the document goes here -->
</rootElement>
```

## XSD datu motak

### Aurrez zehaztutako datu motak

XML Schemak elementu edo atributuatarako aurrez zehaztutako datu-mota ugari ditu, ohikoenak hurrengoak dira:

| Datu mota | Deskribapena  |
|-----------|---|
| string    | UNICODE karaktere kateak adierazten du.   |
| boolean   | Balio logikoak adierazten ditu, <code>true</code> edo <code>false</code> izan ahal direnak. |

|                                     |  |
|-------------------------------------|--|
| integer                             | Zenbaki oso positiboa edo negatiboa                                      |
| positiveInteger                     | Zenbaki oso positibo bat.  |
| negativeInteger                     | Zenbaki oso negatibo bat.  |
| decimal                             | Zenbaki hamartar bat   |
| dateTime                            | Data eta ordu absolutu bat adierazten du AAAA-MM-DD T HH:MM:SS formatuan |
| time                                | Ordu bat adierazten du hh:mm:ss formatuan.                               |
| date                                | Data bat adierazten du aaaa-mm-dd formatuan.                             |
| gYearMonth                          | Urte jakin bateko hilabete bat adierazten du CCYY-MM formatuan.          |
| gMonth                              | Hilabete bat adierazten du –MM formatuan.                                |
| anyURI                              | RFC 2396-an definitutako URI bat adierazten du.                          |
| language                            | RFC 1766-an definitutako hizkuntza-identifikatzaile bat adierazten ditu. |
| ID, IDREF, ENTITY, NOTATION, MTOKEN | DTD dokumentu batean adierazten duten berdina.                           |

Kontsultatu XSD estandarra aurrez definitutako tipo sinpleen [zerrenda osoa](#) lortzeko.

## Datu mota pertsonalizatuak

Sistemak emandako datuez gain, interesgarria izaten ohi da zehaztasun eta eraginkortasun handiagoa duten datu berri pertsonalizatuak sortzea.

Horretarako, XSDk bi talde handitan banatzen ditu elementuak, hauek dituzten datuetan oinarrituta:

- Eduki **sinpleko** elementuak: atributurik gabeko eta datuak baino ez dituzten elementuak dira.
- Eduki **konplexua** duten elementuak: atributuak izan ditzaketen elementuak dira, edukirik ez dutenak edo habiatutako elementuak dituztenak.

## Elementuen deklarazioa

Elementuak XML dokumentu guztien eraikuntza-bloke nagusia dira, datuak dituzte eta hauek dokumentuaren egitura zehaztuko dute.

XML alean erabiliko diren elementu guztiak eskeman adierazi behar dira.

XML Schema-ko elementuen adierazpenek hurrengo sintaxia dute, `element` elementua dela zehazten duen etiketa eta `name` XML etiketaren izena adierazten duena izanez:

```
<xsd:element name="elementuIzena"
    type="edukiSimplea/edukiKonplexua"
    minOccurs="[0..unbounded]"
    maxOccurs="[0..unbounded]"
    fixed="balioa"
    default="balioa"/>
```

Atributu hauek izan ditzake:

- **type:** elementu mota. XML Schemak bi elementu mota adierazten ditu:
  - Mota sinpleak: karaktere-datuak soilik eduki ditzaketen elementuak dira; ezin dute beste elementu edo atributurik izan.
  - Mota konplexuak: elementu hauek beste elementu eta/edo atributu batzuk izan ditzakete. Elementu horien edukia `<xs:complexType>` etiketen artean definitzen da.
- **minOccurs:** Elementuaren gutxieneko agerraldi kopurua.
  - Balio lehenetsia 1 da. Agerraldi kopuru mugagabea nahi bada, hartu beharko lukeen balioa `unbounded` da.
- **maxOccurs:** Elementuaren gehienezko agerraldi kopurua.
  - Balio lehenetsia 1 da. Agerraldi kopuru mugagabea nahi bada, hartu beharko lukeen balioa `unbounded` da.
- **default:** Elementuaren balio lehenetsia zehazten du.
- **fixed:** Atributuaren balioa zehazten du, baldin badago.

## Eduki sinpleko eredua

Mota sinpleko elementuak datuak baino ezin dituzte izan, eta onartzen duten datu motak karaktereak, datak, balio logikoak eta abar izan daitezke.

Ezin izango du beste elementurik eduki, ezta atributurik ere.

| XSD  | XML  |
|--|--|
| <code>&lt;xs:element name="date" type="xs:date"/&gt;</code>      | <code>&lt;date&gt;2023-01-01T11:00:00Z&lt;/date&gt;</code>   |
| <code>&lt;xs:element name="orderId" type="xs:int"/&gt;</code>    | <code>&lt;orderId&gt;3563&lt;/orderId&gt;</code>   |
| <code>&lt;xs:element name="address" type="xs:string"/&gt;</code> | <code>&lt;address&gt;99 London Road&lt;/address&gt;</code>   |
| <code>&lt;xs:element name="body" type="xs:string"/&gt;</code>    | <code>&lt;body&gt;&lt;/body&gt;</code>   |
|  | Mota bat kate gisa definitu daiteke eta edukirik ez izan, joera hau ez da datu mota guztien kasua. |

## Eduki konplexuko eredua

Mota konplexuko elementu bat beste elementu eta/edo atributu batzuk dituen XML elementu bat da.

`complexType` etiketa erabiliz adierazten dira, eta oinarrizko sintaxia honako hau da:

```
<xs:complexType name="datuKonplexuarenIzena">
  <xs:sequence> <!-- sequence/all/choice -->
    ... azpi-elementuak ...
  </xs:sequence>
  ... atributuak ...
</xs:complexType>
```

Atributu hauek izan ditzake:

- **name:** Datu konplexuaren izena.

- `mixed: true` edo `false` baloreak izan ditzake.
- `type`: Identifikatzen duen datu mota.

Bi egitura desberdinen bitartez adierazi daiteke:

- Hurrengo ereduan datu mota `element` egitura barruan adierazten da:

```
<xs:element name="contact">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string" />
      <xs:element name="from" type="xs:string" />
      <xs:element name="subject" type="xs:string" />
      <xs:element name="message" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="date" type="xs:date"/>
  </xs:complexType>
</xs:element>
```

- Bigarren honetan, lehenbiziz elementua datu motarekin adierazten da, eta ondoren datu mota hori era **globalean** adierazten da.

```
<xs:element name="contact" type="type_contact"/>

<xs:complexType name="type_contact">
  <xs:sequence>
    <xs:element name="to" type="xs:string" />
    <xs:element name="from" type="xs:string" />
    <xs:element name="subject" type="xs:string" />
    <xs:element name="message" type="xs:string" />
  </xs:sequence>
  <xs:attribute name="date" type="xs:date"/>
</xs:complexType>
```

Bigarren aukera hau egokiagoa da, izen bat duten `complexType` elementuak eskema osoan zehar berrerabiltzeko aukera ematen bait dute, erreferentzia zuzen bat erabiliz edo beste `complexType` desberdinak zehazteko base izanda.

Honek posible egiten du objetuei bideratutako egiturak eraikitzea, lan egiteko baita dokumentuak kudeatzeko erosoago eta errazagoak direnak.

Sequence, all eta choice inoiz ez dira batera erabiltzen elementuetan, honetako bat bakarrik agertzen da `complexType` elementu batean.

`complexType` batek dituen azpi-elementuak deskribatzeko eta zein ordenenean agertu behar diren zehazten dute.

## Sequence

Honen barruan zehazten diren elementu guztiek eta zehaztutako orden berean agertu behar direla adierazten du elementu honek. AND baten bakiokidea da.

Hurrengo ereduari liburu elementu bat adierazten da, hiru azpi-elementu dituenak eta orden honetan agertu behar direnak: 1<sup>o</sup> titulua, 2<sup>a</sup> autorea eta 3<sup>a</sup> editoriala. Orden horretan agertzen ez badira edo baten bat agertzen ez bada, *parser*-rak errore bat sorraziko du.

### XSD

```
<xs:element name="book" type="type_book"/>

<xs:complexType name="type_book">
  <xs:sequence>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="author" type="xs:string"/>
    <xs:element name="editorial" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

### XML

```
<book>
  <title>El señor de los
  anillos</title>
  <author>John Ronald
  Ruelen Tolkien</author>
  <editorial>Tirant Lo
  Blanch</editorial>
</book>
```

## All

Elementu honek zehaztutako elementu guztiak agertu behar direla adierazten du baina EZ daukate zertan zehaztutako ordenenagatik agertu behar.

Ereduari liburu elementua adierazten da, edozein ordenean agertu ahal diren derrigorrezko hiru azpi-elementuekin.

### XSD

```
<xs:element name="book" type="type_book"/>

<xs:complexType name="type_book">
  <xs:all>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="author" type="xs:string"/>
    <xs:element name="editorial" type="xs:string"/>
  </xs:all>
</xs:complexType>
```

### XML

```
<book>
  <author>John Ronald
  Ruelen Tolkien</author>
  <editorial>Tirant Lo
  Blanch</editorial>
  <title>El señor de los
  anillos</title>
</book>
```

## Choice

Zehaztutako elementu guztien artean bakar bat agertu behar dela adierazten du elementu honek. OR baten baliokidea da.

Ereduari liburu elementu bat adierazten da posible diren hiru azpi-elementuekin, hauetatik bat bakarrik agertu daiteke.

### XSD

```
<xs:element name="book" type="type_book"/>

<xs:complexType name="type_book">
  <xs:choice>
```

### XML

```
<book>
  <title>El señor de los
  anillos</title>
```

```

    <xs:element name="title" type="xs:string"/>        </book>
    <xs:element name="author" type="xs:string"/>
    <xs:element name="editorial" type="xs:string"/>
</xs:choice>
</xs:complexType>

```

## Kardinaltasuna

XML dokumentu batean agertzen den elementu baten instantzia kopurua murriztu daiteke.

Kardinaltasuna `minOccurs` eta `maxOccurs` atributuen bidez zehazten da, eta elementu bat nahitaezkoa, aukerakoa edo aldi kopuru jakin batera arte ager daitekeela zehazteko aukera ematen du.

Bi atributuetarako balio lehenetsia 1 da, beraz, atributuak falta badira (aurreko adibide guztietan bezala), elementua behin bakarrik agertu ahalko da.

Edozein balio ez-negatibo (0, 1, 2, 3, etab.) esleitu ahal zaie, edo unbounded konstante kate berezia. Azken honek agertu ahal den aldi kopuruan ez dagoela mugarik adierazten du, beraz, elementua behin baino gehiagotan agertu daitekeela.

| XSD  | Descripción  |
|--|--|
| <pre> &lt;xs:element name="username"             type="xs:string"/&gt; </pre>  | <p><code>minOccurs</code> edo <code>maxOccurs</code> zehazten ez bada, 1 balio lehenetsiak erabiltzen dira.</p> <p>Horrek esan nahi du <code>username</code> elementuak behin bakarrik agertu behar duela.</p>           |
| <pre> &lt;xs:element name="username"             type="xs:string"             minOccurs="0"         /&gt; </pre>                                   | <p><code>MinOccurs</code> 0 balioan ezartzen bada, <code>username</code> elementua aukerakoa izango da edo behin bakarrik agertu daiteke.</p>  |
| <pre> &lt;xs:element name="username"             type="xs:string"             minOccurs="0"             maxOccurs="unbounded"         /&gt; </pre> | <p><code>username</code> elementua aukerakoa da oraindik.</p> <p>Baina unbounded balioa ezartzerakoan <code>maxOccurs</code> atributuan elementua 0-tik zenbaki mugagabe batera arte agertu daitekela adierazten da.</p> |
| <pre> &lt;xs:element name="username"             type="xs:string"             minOccurs="2"             maxOccurs="10"         /&gt; </pre>        | <p><code>minOccurs</code> eta <code>maxOccurs</code> ezartzeak esan nahi du <code>username</code> elementua gutxienez bi aldiz agertu ahal dela, baina ez 10 aldiz baino gehiagotan.</p>                                 |

## Balio finkoak eta aurrez zehaztuak

XML dokumentuan elementuentzat baliozkoak diren datu-balioak are gehiago murritu daitezke propietate finko eta aurrezehaztuen bidez.

## default

XML dokumentuan baliorik zehazten ez bada, dokumentuak irakurtzen duen aplikazioak XSD dokumentuan zehaztutako balio lehenetsia erabili behar duela adierazten du.

## fixed

XML dokumentuko balioak XSD dokumentuan zehaztutako balioa baino ezin duela izan. Horregatik, ez du zentzurik default zein fixed elementuaren definizio berean erabiltzeak.

| XSD  | XML  |
|--|--|
| <pre>&lt;xs:element name="location"   type="xs:string"   default="unknown" /&gt;</pre> | <pre>&lt;location/&gt;</pre> <p>Dokumentua prozesatzen duen interpreteak "unknown" balioa esleitu behar dio elementuaren balioari.</p> |
| <pre>&lt;xs:element name="location"   type="xs:string"   fixed="UK" /&gt;</pre>        | <pre>&lt;location&gt;UK&lt;/location&gt;</pre> <p>Beste edozein balioa errorea emango luke XML dokumentua baliozkotzean.</p>           |

## Atributuak

Elementu baten edo elementu mota baten atributuak definitzeko, honako egitura hau erabiliko dugu:

```
<xs:attribute name="atributu_izena" type="atributu_mota" use="aldatzailea"/>
```

Atributu hauek izan ditzake:

- **name:** Atributuaren izena.
- **type:** Identifikatzen duen datu mota.
- **use:** Nahitaezko edo aukerako atributua den zehazteko. Atributu bat nahitaezko gisa definitzeko, **required** balioa esleituko diogu. Besterik ezean, aukerakoa da.

Orduan, hurrengo biak baliokideak dira:

```
<xs:attribute name="ID" type="xs:int" />
```

```
<xs:attribute name="ID" type="xs:int" use="optional" />
```

Atributuaren kokapena ezin da bere kabuz joan, egitura horrekin ez baikenuke jakingo zein elementuri buruzkoa den. Horretarako, **complexType** egitura baten barruan jartzen da beti.

| XSD  | XML                                 |
|--|-------------------------------------|
| <pre>&lt;xs:element name="book"&gt;   &lt;xs:complexType&gt;     &lt;xs:attribute name="price" type="xs:double"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre> | <pre>&lt;book price="20"/&gt;</pre> |

Atributuak ere globalki zehaztu daitezke, eta dokumentuan zehar erreferentziatu.



## Eredua

*katalogoa.xml* XML fitxategitik abiatuta:

```
<?xml version="1.0" encoding="UTF-8"?>
<katalogoa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="schema.xsd"
>
  <diska formatua="kasetea" hizkuntza="euskara">
    <izena>Dut</izena>
    <taldea>Dut</taldea>
    <urtea>1995</urtea>
    <herrialdea>Euskal Herria</herrialdea>
  </diska>
  <diska formatua="kasetea" hizkuntza="euskara">
    <izena>Jaiotze Basatia</izena>
    <taldea>Su Ta Gar</taldea>
    <urtea>1991</urtea>
    <herrialdea>Euskal Herria</herrialdea>
  </diska>
  <diska formatua="biniloa" hizkuntza="ingelesa">
    <izena>The Ionic Spell</izena>
    <taldea>Standstill</taldea>
    <urtea>2001</urtea>
    <herrialdea>Catalunya</herrialdea>
  </diska>
</katalogoa>
```

Aurreko XML dokumentua baliozkotuko duen *schema.xsd* XML eskema:

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

  <!-- Katalogo elementuaren egitura -->
  <xs:element name="katalogoa">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="diska"
          type="TDiska"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Diska elementuaren data mota -->
  <xs:complexType name="TDiska">
    <xs:sequence>
      <xs:element name="izena" type="xs:string"/>
      <xs:element name="taldea" type="xs:string"/>
      <xs:element name="urtea" type="xs:int"/>
      <xs:element name="herrialdea" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="formatua" type="xs:string" use="required"/>
    <xs:attribute name="hizkuntza" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

## Murrizketak eta hedadura

### Mota sinpleak (simpleType)

Datu mota sinple batek elementu sinpleei edo atributuei zenbait murrizketa definitzeko balio du. Oso erabilgarria da rangoak, zerrendatutako motak eta abar definitzeko.

simpleType etiketa erabiliz adierazten dira, eta oinarrizko sintaxia honako hau da:

```
<xs:simpleType name="motaSinplearenIzena">
  <xs:restriction> <!-- restriction/list/union -->
    ... azpi-elementuak ...
  </xs:restriction>
</xs:simpleType>
```

Hurrengo azpielementuak izan ditzake:

- **restriction:** Rangoak, patroiak, balio posibleak eta abar jartzeko erabiltzen da.
- **list:** Zerrenda-mota bat definitzeko erabiltzen da.
- **union:** Aurretik definitutako hainbat mota batean batzeko.

Atributu hauek izan ditzake:

- **name:** Mota sinpleari izen bat ezartzeko.

### Restriction

Restriction elementua rangoak, baliozko balioak zerrendatzeko ereduak eta abar jartzeko erabiltzen da.

```
<xs:restriction base="aurrezDefinitutakoMotarenIzena">
  ... azpi-elementuak ...
</xs:restriction>
```

Hurrengo atributua du:

- **base:** Aurrez definitutako datu-mota horren gainean eraikitzen da murrizketa.

Honako murrizketa hauek izan ditzake:

- **enumeration:** Elementuak har ditzakeen balioak zerrendatzen ditu.

```
<xs:simpleType name="generoa">
  <xs:restriction base="xs:string">
    <xs:enumeration value="emakumezkoa"/>
    <xs:enumeration value="gizonezkoa"/>
    <xs:enumeration value="beste bat"/>
  </xs:restriction>
</xs:simpleType>
```

- **maxExclusive, minExclusive:** Elementuak har ditzakeen balio minimoak edo maximoak, azken balioa sartu gabe.

- **maxInclusive, minInclusive:** Elementuak har ditzakeen balio minimoak edo maximoak, azken balioa barne.

```
<xs:element name="notak">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="100"/>
      <xs:maxInclusive value="200"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **pattern:** Murrizketa adierazten duen esamolde erregularra.

| <b>Patroia</b> | <b>Esanahia</b>                                     | <b>Patroia</b> | <b>Esanahia</b>                                |
|----------------|---|----------------|--|
| [A-Z a-z]      | Letra bat   | (A B)          | A edo B katearen berdina den katea             |
| [A-Z]          | Letra larria  | A?             | A katea zero edo behin.                        |
| [a-z]          | Letra xehea   | A+             | A katea behin edo gehiagotan.                  |
| [0-9]          | Digitu hamartarra                                   | A*             | A katea zero edo gehiagotan.                   |
| \t             | Tabuladorea   | A{2}           | A katea birritan.                              |
| \d             | Edozein karaktere, 0 eta 9 bitarteko bat izan ezik. | [abcd]         | Kortxete artean dauden karaktereetako bat.     |
| AB             | A eta B kateen kateaketa den katea.                 | [^abcd]        | Kortxeteen artean ez dagoen edozein karaktere. |

```
<xs:simpleType name="matrikula-zenbakia">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{4} [a-z]{3}"/>
  </xs:restriction>
</xs:simpleType>
```

- **length, maxLength, minLength:** Karaktereen luzera testu motako elementu batentzat.

```
<xs:simpleType name="password">
  <xs:restriction base="xs:string">
    <xs:length value="8"/>
  </xs:restriction>
</xs:simpleType>
```

- **totalDigits:** Baimendutako digituen kopuru zehatza.

```
<xs:simpleType name="nan">
  <xs:restriction base="xs:integer">
    <xs:length value="8"/>
  </xs:restriction>
</xs:simpleType>
```

- **fractionDigits:** Baimendutako hamartarren gehieneko kopurua.

## List

Mota sinple bat definitzeko aukera ematen du, espazioen artean bereiztutako beste mota sinple batzuen zerrenda bat osatuz:

## Union

Mota sinple bat baino gehiago konbinatzeko aukera ematen du.

## Hedadura

`xs:extension` erabilita `simpleType` edo `complexType` bat handitu dezakegu, aurretik definitutako oinarri-mota bati elementuak edo atributuak gehituz.

`simpleContent` edo `complexContent` etiketen bidez zehazten da elementua eduki sinple edo konplexu baten bidez zabalduko den.

Testua eta atributuak baino eduki ezin dituen (ez azpielementuak) elementu baten adierazpena:

```
<xs:complexType name="dokumentuMota">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="txantxiloia"
        type="xs:string"
        use="required"/>
      <xs:attribute name="berrikusia"
        type="xs:boolean"
        default="false"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Aurrez definitutako motaren gainean elementu gehiago gehitzen zaizkion elementu motaren adierazpena:

```
<xs:complexType name="pertsonaMota">
  <xs:sequence>
    <xs:element name="izena" type="xs:string"/>
    <xs:element name="adina" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:integer"/>
</xs:complexType>

<xs:complexType name="kontaktuMota">
  <xs:complexContent>
    <xs:extension base="pertsonaMota">
      <xs:sequence>
```

```
        <xs:element name="email" type="xs:string"/>
        <xs:element name="tlfm" type="xs:string"/>
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```