

# **Practica01 - RA05: XML i XSD: Validació i Conversió**

**Rosi Di Nino  
Àlex Gascon**

# Índex

<b>Practica01 - RA05: XML i XSD: Validació i Conversió.....</b>	<b>1</b>
Objectius.....	3
Fase 1: Cerca i anàlisi.....	4
• Cercar una web pública, aquesta web ens proporciona descarregar fitxers XML.....	4
• Analitzar l'estructura de l'XML (arbre, etiquetes, etc.).....	4
• Intentar localitzar si està vinculat a un XSD.....	6
Fase 2: Validació.....	6
• Fitxer XSD.....	6
Fase 3: Conversió.....	8
• Convertir l'XML a JSON.....	8
Fase 4: Lliurament i reflexió.....	10
• Quina web s'ha testejat.....	10
• Com és l'XML?.....	10
• Resultat de la validació.....	10
• Format i resultat de la conversió.....	10
• Aplicacions possibles d'aquest XML en intercanvi d'informació.....	11
Com detectar documents XML al navegador?.....	11

# Objectius

- Detectar i analitzar l'ús d'XML en entorns reals (webs públiques).
- Descarregar un fitxer XML públic i identificar si està associat a un esquema XSD.
- Validar el fitxer XML amb un validador online o d'escriptori.
- Convertir el document XML a JSON o altres formats (CSV, HTML...).
- Reflexionar sobre l'intercanvi d'informació en entorns web mitjançant XML.

## Fase 1: Cerca i anàlisi

- Cercar una web pública, aquesta web ens proporciona descarregar fitxers XML

<https://learn.microsoft.com/es-es/dotnet/standard/ling/sample-xml-file-customers-orders>

- Analitzar l'estructura de l'XML (arbre, etiquetes, etc.).

L'arrel del document és <root> i conté el següent:

```
<Root>

  <Customers>

    <Customer CustomerID="GREAL">

      <CompanyName>Great Lakes Food Market</CompanyName>

      <ContactName>Howard Snyder</ContactName>

      <ContactTitle>Marketing Manager</ContactTitle>

      <Phone>(503) 555-7555</Phone>

      <FullAddress>

        <Address>2732 Baker Blvd.</Address>

        <City>Eugene</City>

        <Region>OR</Region>

        <PostalCode>97403</PostalCode>

        <Country>USA</Country>

      </FullAddress>

    </Customer>

    .

  </root>
```

**Dintre de root hi ha 2 etiquetes order i costumer, l'arbre del xml:**

Root



- Intentar localitzar si està vinculat a un XSD

(sovint al principi de l'XML hi ha una referència amb `xsi:schemaLocation`).

Aquest fitxer xml no te una vinculacio amb XSD.

## Fase 2: Validació

- Fitxer XSD

Amb aquesta url creem el fitxer xsd del xml

<https://www.freeformatter.com/xsd-generator.html>

Has de ficar el xml

The screenshot shows the 'XSD/XML Schema Generator' website. At the top, it says 'Generates a XSD (XML Schema) from a XML file. Simply copy-paste OR upload your XML document and let the generator figure out the rest. The generator will try to use a 'smart' approach to figure out the data type (you can always refine it after).' Below this, it lists three design options: 'Russian Doll', 'Salami slice', and 'Venetian Blind'. The 'Russian Doll' option is selected in the 'XSD design' dropdown menu. The 'Generate XSD' button is highlighted with a red box. The XML input field contains the following code: 

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <Customers>
    <Customer CustomerID="GREAL">
      <CompanyName>Great Lakes Food Markets</CompanyName>
```

```

1 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
2   <xs:element name="registro">
3     <xs:complexType>
4       <xs:sequence>
5         <xs:element name="usuario" maxOccurs="unbounded" minOccurs="0">
6           <xs:complexType>
7             <xs:sequence>
8               <xs:element type="xs:string" name="apellidos-y-nombre"/>
9               <xs:element type="xs:string" name="email"/>
10              <xs:element name="conexiones">
11                <xs:complexType>
12                  <xs:sequence>
13                    <xs:element name="conexión" maxOccurs="unbounded" minOccurs="0">
14                      <xs:complexType>
15                        <xs:simpleContent>
16                          <xs:extension base="xs:string"/>

```

## -Generated XSD-

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Customers">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Customer" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="CompanyName"/>
                    <xs:element type="xs:string" name="ContactName"/>
                    <xs:element type="xs:string" name="ContactTitle"/>
                    <xs:element type="xs:string" name="Phone"/>
                    <xs:element type="xs:string" name="Fax" minOccurs="0"/>
                    <xs:element name="FullAddress">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="Address"/>

```

Descarrega el xsd i el xml i fes la validació

<https://www.freeformatter.com/xml-validator-xsd.html>

XML i XSD validat

JSON Formatter

HTML Formatter

SQL Formatter

Validators

XML Validator

JSON Validator

HTML Validator

XPath Tester

Credit Card Number Generator & ...

Regular Expression Tester

Java Regular Expression Tester

Cron Expression Generator (Quartz)

Converters

XSD Generator

XSLT (XSL Transformer)

XML to JSON Converter

JSON to XML Converter

CSV to XML Converter

CSV to JSON Converter

YAML to JSON Converter

JSON to YAML Converter

Epoch Timestamp To Date

Encoders / Cryptography

Url Encoder & Decoder

Base 64 Encoder & Decoder

Convert File Encoding

Message Digester (MD5, SHA-256, ...)

HMAC Generator

QR Code Generator

XML Validator - XSD (XML Schema)

Credit card validator

Validators / XML Validator - XSD (XML Schema)

Validates the XML string/file against the specified XSD string/file. XSD files are "XML Schemas" that describe the structure of a XML document. The validator checks for well formedness first, meaning that your XML file must be parsable using a DOM/SAX parser, and only then does it validate your XML against the XML Schema. The validator will report fatal errors, non-fatal errors and warnings.

The XML document is valid.

Option 1: Copy-paste your XML document here

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <Customers>
    <Customer CustomerID="GREAL">
      <CompanyName>Great Lakes Food Market</CompanyName>
```

Option 2: Or upload your XML file

Examinar...

No se ha seleccionado ningún archivo.

File encoding

UTF-8

Option 1: Copy-paste your XSD here (Optional if XSD referred in XML using schemaLocation)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified" elementFormDefault="qualified">
  <xs:element name="Root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Customers">
```

Option 2: Or upload your XSD document

Examinar...

No se ha seleccionado ningún archivo.

File encoding

UTF-8

Validate XML

## Fase 3: Conversió

- **Convertir l'XML a JSON**

Des de la mateixa pàgina d'abans podem convertir de XML a json

### XML to JSON Converter

[Converters](#) / XML to JSON Converter

This online tool allows you to convert an XML file into a JSON file. This process is not 100% accurate in that XML uses different item types that do not have an equivalent JSON representation.

- Attributes are treated as JSON properties and can be prefixed to differentiate them (@ is used by default)
- Sequences of two or more 'similar' elements will be converted to JSON arrays
- You can add an attribute \_type to your elements to infer the json type (boolean, float, integer, number, string)
- Terminal #text item types will be converted into a JSON property with the name #text. This can be changed in the options.

#### Option 1: Copy-paste your XML here

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <Customers>
    <Customer CustomerID="GREAL">
      <CompanyName>Great Lakes Food Market</CompanyName>
```

#### Option 2: Or upload your XML file

#### File encoding

Examinar...

No se ha seleccionado ningún archivo.

UTF-8

#### Prefix attributes with

@

#### #text property name

#text

#### Indentation level

2 spaces per indent level

#### Bracket style

Collapsed (braces on same line)

Convert XML to JSON

Convert XML to JSON in new window

### -Converted XML-

```
{
  "customers": [
    {
      "@CustomerID": "GREAL",
      "CompanyName": "Great Lakes Food Market",
      "ContactName": "Howard Snyder",
      "ContactTitle": "Marketing Manager",
      "Phone": "(503) 555-7555",
      "FullAddress": {
        "Address": "2732 Baker Blvd.",
        "City": "Eugene",
        "Region": "OR",
        "PostalCode": "97403",
        "Country": "USA"
      }
    },
    {
      "@CustomerID": "HUNGC",
      "CompanyName": "Hungry Coyote Import Store",
      "ContactName": "Yoshi Latimer",
```

Copy

Save



**CODI:**

{

"Customers": [

{

"@CustomerID": "GREAL",

"CompanyName": "Great Lakes Food Market",

"ContactName": "Howard Snyder",

"ContactTitle": "Marketing Manager",

"Phone": "(503) 555-7555",

"FullAddress": {

"Address": "2732 Baker Blvd.",

"City": "Eugene",

"Region": "OR",

"PostalCode": "97403",

"Country": "USA"

}

## Fase 4: Lliurament i reflexió

Informe final explicant:

- **Quina web s'ha testejat**

La web testejada ha sigut Microsoft Learn - XML de mostra per a LINQ. Aquesta pàgina proporciona un fitxer XML de mostra amb dades fictícies de clients i comandes.

<https://learn.microsoft.com/es-es/dotnet/standard/ling/sample-xml-file-customers-orders>

- **Com és l'XML?**

**L'estructura del XML té:**

Conté múltiples elements **<Order>**, cadascun amb:

- CustomerID (referència al client)
- EmployeeID, OrderDate, RequiredDate
- Subestructura **<ShipInfo>** amb:
  - Atribut: ShippedDate
  - Subetiquetas: ShipVia, Freight, ShipName, ShipAddress, etc.

Conté múltiples elements **<Customer>**, cadascun amb:

- Atribut CustomerID
- Subetiquetas: CompanyName, ContactName, ContactTitle, Phone, Fax (opcional)
- Subestructura **<FullAddress>** amb camps d'adreça.

- **Resultat de la validació**

El XSD l'hem generat de forma manual amb la pàgina web que ens proporciona la pràctica. La validació del XML s'ha realitzat de forma correcta amb el XSD que hem generat.

- **Format i resultat de la conversió**

El JSON s'ha generat correctament amb la mateixa pàgina web que hem utilitzat per a la validació.

- **Aplicacions possibles d'aquest XML en intercanvi d'informació**

## Intercanvi de dades entre empreses

- Les empreses poden utilitzar fitxers XML per enviar comandes, factures, o llistats de productes entre sistemes ERP.

## Integració de serveis web

- En serveis REST o SOAP, sovint s'utilitza XML com a format d'intercanvi. Aquest fitxer podria formar part d'una API on es consulten les comandes o les dades de clients.

## Validació i control de qualitat

- L'ús d'XSD permet assegurar que les dades compleixen un format establert. Això és important per evitar errors en el processament.

# Com detectar documents XML al navegador?

**Si el navegador accedeix a una API que retorna XML:**

Pots obrir les *Dev Tools* (F12), anar a la pestanya **Network**, i mirar els requests.

Si algun retorna **Content-Type: application/xml** o **text/xml**, ho tens.

