#### Validar documento XML.

XML significa "Lenguaje de Marcado Extensible" (eXtensible Markup Language, en inglés).

Es un lenguaje diseñado para almacenar y transportar datos de manera estructurada.

No es un lenguaje de programación, sino un lenguaje de marcado que define reglas para la estructura de los documentos.

##### Sintaxis de XML:

* XML utiliza etiquetas para marcar los elementos y estructuras de los datos.
* Las etiquetas están encerradas entre "<" y ">".
* Cada elemento tiene una etiqueta de apertura y una etiqueta de cierre.
* Ejemplo: "<nombre>Contenido del elemento</nombre>"

##### Elementos y atributos:

* Los datos en XML se organizan en elementos.
* Cada elemento tiene un nombre y puede contener texto, otros elementos o ambos.
* Los elementos pueden tener atributos que proporcionan información adicional.
* Los atributos se definen dentro de la etiqueta de apertura del elemento.
* Ejemplo: "<persona nombre="Juan" edad="30">Contenido de la persona</persona>"

##### Estructura jerárquica:

* XML sigue una estructura jerárquica, donde los elementos pueden anidarse dentro de otros elementos.
* Un elemento que contiene otros elementos se llama "elemento padre" y los elementos contenidos se llaman "elementos hijos".
* Los elementos pueden tener múltiples niveles de anidamiento.

*Ejemplo:*

<libro>

<titulo> El Gran Gatsby </titulo>

<autor> F. Scott Fitzgerald </autor>

</libro>

En este ejemplo, "libro" es el elemento padre que contiene los elementos hijos "titulo" y "autor".

#### DTD.

##### ¿Qué es DTD?

* DTD es un acrónimo de "Definición de Tipo de Documento" (Document Type Definition, en inglés).
* Es una forma de definir la estructura y las reglas de validación de un documento XML.
* Un DTD especifica los elementos permitidos, su orden, los atributos que pueden contener y otras restricciones.

##### Sintaxis básica de un DTD:

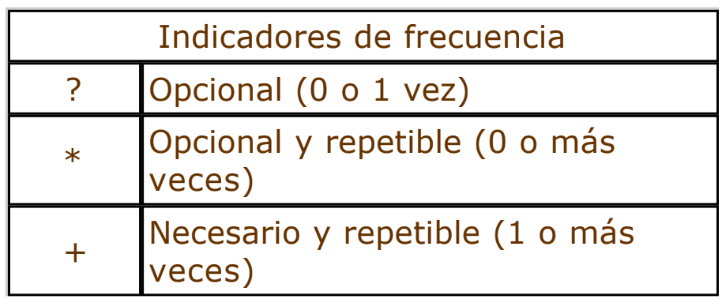
* Un DTD se define en un archivo de texto separado y se enlaza al documento XML correspondiente.
* La declaración de DTD se coloca entre corchetes angulares: "<!DOCTYPE>"
* Ejemplo de declaración de DTD:

<!DOCTYPE nombre\_del\_documento SYSTEM "ruta\_al\_archivo\_dtd.dtd">

##### Definición de elementos:

* Los elementos se definen en el DTD utilizando la palabra clave "ELEMENT".
* Sintaxis: "<!*ELEMENT* nombre\_del\_elemento tipo\_de\_contenido>"
* Los tipos de contenido más comunes son:
  + #PCDATA: Datos de texto.
  + EMPTY: Sin contenido.
  + (#PCDATA|elemento): Alternancia entre datos de texto y elementos.
  + (elemento1, elemento2, ...): Secuencia de elementos.
  + (elemento1 | elemento2 | ...): Elección entre elementos.
* Ejemplo de definición de elemento:

<!ELEMENT libro (#PCDATA)>



##### Definición de atributos:

* Los atributos se definen en el DTD utilizando la palabra clave "ATTLIST".
* Sintaxis: "<!ATTLIST nombre\_del\_elemento nombre\_del\_atributo tipo\_del\_atributo valor\_predeterminado>"
* Los tipos de atributo más comunes son:
  + CDATA: Datos de caracteres.
  + ID: Valor único dentro del documento.
  + IDREF: Referencia a un ID existente.
  + NMTOKEN: Token de nombre.
* Ejemplo de definición de atributo:

<!ATTLIST persona nombre CDATA #REQUIRED>

##### Uso de entidades:

* Las entidades se utilizan para definir y referenciar valores reutilizables en el documento XML.
* Las entidades se definen en el DTD utilizando la palabra clave "ENTITY".
* Sintaxis: "<!ENTITY nombre\_de\_entidad valor\_de\_entidad>"
* Las entidades se referencian utilizando la sintaxis "&nombre\_de\_entidad;".
* Ejemplo de definición y referencia de entidad:

<!ENTITY nombre "Juan">

<persona>&nombre;</persona>

*Ejemplo de un XML y DTD.*

<!DOCTYPE etiqueta[

<!ELEMENT etiqueta (nombre, calle, ciudad, pais, codigo)>

<!ELEMENT nombre (#PCDATA)>

<!ELEMENT calle (#PCDATA)>

<!ELEMENT ciudad (#PCDATA)>

<!ELEMENT pais (#PCDATA)>

<!ELEMENT codigo (#PCDATA)>

]>

<etiqueta>

<nombre>Fulano Mengánez</nombre>

<calle>c/ Mayor, 27</calle>

<ciudad>Valderredible</ciudad>

<pais>España</pais>

<codigo>39343</codigo>

</etiqueta>

#### XML SCHEMAS.

##### ¿Qué es XML Schema (XSD)?

* XML Schema es un lenguaje de esquema utilizado para definir la estructura, los tipos de datos y las restricciones de un documento XML.
* Proporciona una mayor capacidad de definición y validación en comparación con DTD.
* Los esquemas XSD se definen en archivos XML separados y se enlazan al documento XML correspondiente.

##### Definición de elementos:

* Los elementos se definen en un esquema XSD utilizando el elemento <xs:element>.
* Puedes especificar el nombre del elemento, su tipo de datos y otras restricciones.
* Ejemplo de definición de elemento:

<xs:element name="nombre\_del\_elemento" type="tipo\_de\_datos" />

##### Definición de tipos de datos:

* XSD ofrece una variedad de tipos de datos predefinidos, como cadena de caracteres (string), número entero (integer), booleano (boolean), etc.
* También puedes definir tus propios tipos de datos utilizando la construcción <xs:simpleType>.
* Ejemplo de definición de tipo de datos:

<xs:element name="edad" type="xs:integer" />

##### Validación de restricciones:

* Puedes aplicar restricciones a los elementos y atributos en un esquema XSD.
* Algunas restricciones comunes incluyen valores mínimos y máximos, longitud, patrones de caracteres, enumeraciones, etc.
* Ejemplo de restricción en un elemento:

<xs:element name="edad">

<xs:simpleType>

<xs:restriction base="xs:integer">

<xs:minInclusive value="0" />

<xs:maxInclusive value="150" />

</xs:restriction>

</xs:simpleType>

</xs:element>

##### Uso de referencias y anidamiento:

* Puedes referenciar y reutilizar elementos y tipos de datos definidos en el esquema XSD utilizando la construcción <xs:element ref="nombre\_del\_elemento" /> o <xs:complexType>.
* Esto permite una mayor modularidad y reutilización en la definición de esquemas.
* Ejemplo de referencia a un elemento:

<xs:element name="direccion">

<xs:complexType>

<xs:sequence>

<xs:element ref="calle" />

<xs:element ref="ciudad" />

</xs:sequence>

</xs:complexType>

</xs:element>

#### XPATH.

##### ¿Qué es XPath?

* XPath es un lenguaje de consulta utilizado para buscar y seleccionar elementos o nodos en un documento XML.
* XPath proporciona una sintaxis concisa y poderosa para navegar y filtrar la estructura jerárquica de un documento XML.

##### Expresiones XPath:

* Las expresiones XPath se utilizan para especificar los criterios de búsqueda y selección de elementos en un documento XML.
* Puedes utilizar las siguientes expresiones XPath:
  + Selección de nodos:
    - /: Selecciona el nodo raíz del documento.
    - //: Selecciona todos los nodos descendientes sin importar su ubicación.
    - nombre\_del\_elemento: Selecciona todos los nodos con el nombre especificado.
    - nombre\_del\_elemento1/nombre\_del\_elemento2: Selecciona los nodos hijos que están anidados dentro de otro nodo.
  + Filtros:
    - [condición]: Aplica una condición para filtrar los nodos.
    - @nombre\_del\_atributo: Selecciona el valor de un atributo.
  + Operadores:
    - and, or: Operadores lógicos.
    - =, !=, <, >, <=, >=: Operadores de comparación.

##### Ejemplos de expresiones XPath:

* /libros: Selecciona todos los nodos "libros" en el documento.
* //libro: Selecciona todos los nodos "libro" en cualquier nivel de anidamiento.
* /libros/libro[autor='F. Scott Fitzgerald']: Selecciona los nodos "libro" que tienen un autor igual a "F. Scott Fitzgerald".
* //libro[@categoria='novela']: Selecciona todos los nodos "libro" que tienen un atributo "categoria" con el valor "novela".

##### Uso de funciones XPath:

* XPath también proporciona una serie de funciones incorporadas que puedes utilizar en tus expresiones XPath.
* Algunas funciones comunes incluyen text(), contains(), starts-with(), substring(), count(), sum(), entre otras.