

Que son DTD y XSD


```
<!ELEMENT Libros (Libro+)>
<!ELEMENT Libro (Titulo, Autor)>
<!ELEMENT Titulo (#PCDATA)>
<!ELEMENT Autor (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Libros>
  <Libro>
    <Titulo>Don Quijote de la Mancha</Titulo>
    <Autor>Miguel de Cervantes</Autor>
  </Libro>
  <Libro>
    <Titulo>La sombra del viento</Titulo>
    <Autor>Carlos Ruiz Zafón</Autor>
  </Libro>
  <Libro>
    <Titulo>Cien años de soledad</Titulo>
    <Autor>Gabriel García Márquez</Autor>
  </Libro>
</Libros>
```

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

  <!-- Definición del elemento raíz, que contiene uno o más elementos 'Libro' -->
  <xs:element name="Libros">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Libro" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Titulo" type="xs:string"/>
              <xs:element name="Autor" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```



XSD. Introducción

¿Qué es un esquema XML (XSD)?

- Un esquema XML, también conocido como XSD (XML Schema Definition), es un lenguaje de descripción de documentos XML.
- Permite definir la estructura, los tipos de datos y las restricciones de un documento XML.
- Es una especificación formal que describe la estructura y el contenido válido de un documento XML.

XSD. Introducción

Ventajas de utilizar esquemas XSD

- Validación de documentos XML
- Definición de tipos de datos
- Documentación y mantenimiento
- Interoperabilidad

XSD. Introducción

Diferencias entre DTD y XSD

- ¿busca y entiende las diferencias entre dtd y XSD?

XSD. Introducción

Diferencias entre DTD y XSD

- Sintaxis
- Tipos de datos
- Modularidad
- Compatibilidad con espacios de nombres

XSD. Estructura básica de un esquema XSD

Elemento raíz <xs:schema>

- El elemento raíz de un esquema XSD es <xs:schema>.
- Define los parámetros principales del esquema, como la versión de XML Schema, los espacios de nombres y otras configuraciones globales.

Declaración de espacios de nombres

- Los esquemas XSD utilizan espacios de nombres para organizar y diferenciar los elementos y tipos de datos.
- La declaración de espacios de nombres se realiza dentro del elemento <xs:schema>, utilizando el atributo xmlns:xs.

```
xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- Definición de elementos y tipos de datos -->
</xs:schema>
```

XSD. Estructura básica de un esquema XSD

Uso de xmlns y espacios de nombres

- **Namespace:** Para definir un espacio de nombres, se usa el atributo xmlns. Se puede aplicar a nivel de documento, elemento o atributo.
- **Uso de prefijos:** Es común asignar un prefijo a un namespace para facilitar su uso. Ejemplo: xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance".

```
<root xmlns="http://www.example.com/main" xmlns:ext="http://www.example.com/extension">  
  <name>Documento principal</name>  
  <ext:name>Elemento extendido</ext:name>  
</root>
```


XSD. Estructura básica de un esquema XSD

Definición de elementos y atributos

- Dentro de <xs:schema>, se definen los elementos y atributos del documento XML.
- Elementos: <xs:element>
- Atributos: <xs:attribute>

```
<xs:element name="libro">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="titulo" type="xs:string"/>
      <xs:element name="autor" type="xs:string"/>
      <xs:element name="publicacion" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

XSD. Tipos de datos en XSD

Tipos de datos simples

Estos tipos de datos simples pueden ser utilizados para definir los elementos y atributos de un documento XML.

- xs:string: Cadena de texto
- xs:integer: Número entero
- xs:boolean: Valor booleano
- xs:date: Fecha
- xs:time: Hora
- xs:decimal: Número decimal
- xs:anyURI: URI o URL

Tipos de datos complejos

se definen utilizando los elementos `<xs:complexType>` y `<xs:element>`

- Elementos anidados
- Atributos
- Secuencias de elementos
- Elecciones entre elementos
- Grupos de elementos

```
<xs:complexType name="libroType">
  <xs:sequence>
    <xs:element name="titulo" type="xs:string"/>
    <xs:element name="autor" type="xs:string"/>
    <xs:element name="publicacion" type="xs:date"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>
```

XSD. Tipos de datos en XSD

Secuencias de elementos

```
<order>  
  <orderId>123</orderId>  
  <orderDate>2024-04-25</orderDate>  
  <customerName>Alice Johnson</customerName>  
</order>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:element name="order">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="orderId" type="xs:int"/>  
        <xs:element name="orderDate" type="xs:date"/>  
        <xs:element name="customerName" type="xs:string"/>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>  
</xs:schema>
```

XSD. Tipos de datos en XSD

Ejemplo elementos anidados:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="address">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="street" type="xs:string"/>
              <xs:element name="city" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<person>
  <name>John Doe</name>
  <address>
    <street>Main St 123</street>
    <city>Springfield</city>
  </address>
</person>
```

XSD. Tipos de datos en XSD

Atributos

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  <xs:element name="product">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="price" type="xs:decimal"/>
      </xs:sequence>
      <xs:attribute name="currency" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<product currency="USD">
  <name>Laptop</name>
  <price>999.99</price>
</product>
```

XSD. Tipos de datos en XSD

Elecciones entre elementos

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="vehicle">
    <xs:complexType>
      <xs:choice>
        <xs:element name="car" type="xs:string"/>
        <xs:element name="motorcycle" type="xs:string"/>
        <xs:element name="bicycle" type="xs:string"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<vehicle>
  <car>Sedan</car>
</vehicle>
```

```
<vehicle>
  <motorcycle>Sport Bike</motorcycle>
</vehicle>
```


XSD. Tipos de datos en XSD

Grupos de elementos

```
<person>
  <name>Bob Smith</name>
  <street>Park Ave 45</street>
  <city>Los Angeles</city>
  <state>California</state>
</person>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:group name="addressGroup">
    <xs:sequence>
      <xs:element name="street" type="xs:string"/>
      <xs:element name="city" type="xs:string"/>
      <xs:element name="state" type="xs:string"/>
    </xs:sequence>
  </xs:group>

  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:group ref="addressGroup"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XSD. Tipos de datos en XSD

Restricciones de tipos de datos

- Longitud mínima y máxima de cadenas
- Valores mínimos y máximos para números
- Patrones de expresiones regulares
- Enumeraciones de valores permitidos

- Se definen utilizando elementos como
- `<xs:minLength>`, `<xs:maxLength>`,
- `<xs:minInclusive>`, `<xs:maxInclusive>`,
- `<xs:pattern>` y `<xs:enumeration>`.

```
<xs:element name="edad">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="age">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="18"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```


XSD. Elementos y atributos XSD

Declaración de elementos

En los esquemas XSD, los elementos se definen utilizando el elemento `<xs:element>`. Cada elemento tiene un nombre único y puede tener atributos, tipos de datos y restricciones asociados.

```
<xs:element name="libro">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="titulo" type="xs:string"/>
      <xs:element name="autor" type="xs:string"/>
      <xs:element name="publicacion" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD. Elementos y atributos XSD

Declaración de atributos

Los atributos se definen utilizando el elemento `<xs:attribute>` dentro de la definición de un elemento o tipo de dato complejo. Los atributos tienen un nombre único y un tipo de dato asociado.

```
<xs:element name="libros">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="libro" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="titulo" type="xs:string"/>
            <xs:element name="autor" type="xs:string"/>
            <xs:element name="publicacion" type="xs:date"/>
          </xs:sequence>
          <xs:attribute name="id" type="xs:ID"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD. Estructuras de datos en XSD

Secuencias (<xs:sequence>)

Los esquemas XSD permiten definir secuencias de elementos, donde los elementos deben aparecer en un orden específico dentro del documento XML. Esto se logra utilizando el elemento <xs:sequence>.

```
<xs:element name="libro">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="titulo" type="xs:string"/>
      <xs:element name="autor" type="xs:string"/>
      <xs:element name="publicacion" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD. Estructuras de datos en XSD

Elecciones (<xs:choice>)

Los esquemas XSD también permiten definir estructuras de elección, donde solo uno de los elementos definidos dentro del <xs:choice> puede aparecer en el documento XML.

```
<xs:element name="documento">
  <xs:complexType>
    <xs:choice>
      <xs:element name="libro" type="libroType"/>
      <xs:element name="articulo" type="articuloType"/>
      <xs:element name="tesis" type="tesisType"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

XSD. Estructuras de datos en XSD

Grupos (<xs:group>)

Los grupos permiten agrupar elementos y reutilizarlos en diferentes partes del esquema XSD. Esto facilita la modularidad y la reutilización de estructuras de datos.

```
<xs:group name="datosPersona">
  <xs:sequence>
    <xs:element name="nombre" type="xs:string"/>
    <xs:element name="apellido" type="xs:string"/>
    <xs:element name="edad" type="xs:integer"/>
  </xs:sequence>
</xs:group>

<xs:element name="persona">
  <xs:complexType>
    <xs:group ref="datosPersona"/>
    <xs:attribute name="id" type="xs:ID"/>
  </xs:complexType>
</xs:element>
```

XSD. Herencia y extensión en XSD

Tipos de datos derivados

Los esquemas XSD permiten definir tipos de datos derivados a partir de otros tipos de datos existentes. Esto se logra utilizando el elemento `<xs:extension>` o `<xs:restriction>` dentro de la definición de un tipo de dato complejo.

En este ejemplo, `libroDigitalType` hereda los elementos y atributos de `libroType` y agrega nuevos elementos específicos para libros digitales.

```
<xs:complexType name="libroType">
  <xs:sequence>
    <xs:element name="titulo" type="xs:string"/>
    <xs:element name="autor" type="xs:string"/>
    <xs:element name="publicacion" type="xs:date"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID"/>
</xs:complexType>

<xs:complexType name="libroDigitalType">
  <xs:complexContent>
    <xs:extension base="libroType">
      <xs:sequence>
        <xs:element name="formato" type="xs:string"/>
        <xs:element name="tamano" type="xs:integer"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

XSD. Herencia y extensión en XSD

Restricción de tipos de datos

Además de la extensión, los esquemas XSD permiten restringir los tipos de datos existentes para crear nuevos tipos más específicos.

```
<xs:simpleType name="edadType">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="120"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="persona">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="nombre" type="xs:string"/>
      <xs:element name="edad" type="edadType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD y DTD. Validacion de documentos

Ejercicios

Ejercicio 7

Crea un esquema XSD para representar una lista de libros con los siguientes elementos:

- Título (obligatorio)
- Autor (obligatorio)
- Año de publicación (opcional)
- Género (obligatorio, con valores predefinidos: "Ficción", "No Ficción", "Poesía")

Ejercicio8

Crea un esquema XSD para representar una lista de empleados de una empresa, con los siguientes elementos: Nombre (obligatorio)

- Apellido (obligatorio)
- Edad (obligatorio, entre 18 y 65 años)
- Departamento (obligatorio, con valores predefinidos: "Ventas", "Recursos Humanos", "Finanzas", "Tecnología")
- Salario (obligatorio, con un valor mínimo de 1000 y máximo de 100000)

Ejercicios

Ejercicio 9

- Crea un esquema XSD para representar una lista de recetas de cocina, con los siguientes elementos:
- Nombre de la receta (obligatorio)
- Ingredientes (al menos uno) Nombre (obligatorio)
- Cantidad (obligatorio, decimal positivo)
- Unidad de medida (obligatorio, con valores predefinidos: "gramos", "mililitros", "cucharadas", "tazas")
- Instrucciones (obligatorio)

Ejercicios

Ejercicio 10 Crea un esquema XSD para representar una lista de productos de una tienda en línea, con los siguientes elementos:

Código de producto (obligatorio, único)

- Nombre (obligatorio)
- Descripción (obligatorio)
- Categoría (obligatorio, con valores predefinidos: "Electrónica", "Hogar", "Moda", "Deportes")
- Precio (obligatorio, decimal positivo)
- Existencias (obligatorio, entero positivo)
- Envío gratuito (opcional, booleano)