

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>
```



DTD. Introducción

- **Qué es DTD:** Definición y propósito de un DTD.
- **Tipos de DTD:** Integrados (in-line) y externos.
- **Elementos y Atributos en DTD:** Explicación de cómo se definen elementos y atributos.
- **Declaraciones Básicas:**
 - Elementos: `<!ELEMENT>`
 - Atributos: `<!ATTLIST>`

Ejercicio Práctico: Crear un DTD simple para el documento XML del ejercicio anterior

DTD. Introducción

- **Definición de DTD:**

Un DTD es un conjunto de reglas que define la estructura y los elementos de un documento XML. Se utiliza para validar la conformidad de un documento XML con una estructura esperada.

- **Por qué usar DTD:**

Asegura la consistencia y estructura de documentos XML, permitiendo la validación automática.

DTD. Introducción

- **DTD Interno (In-line):** El DTD se define dentro del propio documento XML, en la parte superior.
- **DTD Externo:** El DTD se encuentra en un archivo separado y se vincula al documento XML mediante una declaración.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE Libros [
    <!ELEMENT Libros (Libro+)>
    <!ELEMENT Libro (Titulo, Autor)>
    <!ELEMENT Titulo (#PCDATA)>
    <!ELEMENT Autor (#PCDATA)>
]>
<Libros>
    <Libro>
        <Titulo>Don Quijote de la Mancha</Titulo>
        <Autor>Miguel de Cervantes</Autor>
    </Libro>
    <Libro>
        <Titulo>La vida es suenno</Titulo>
        <Autor>Calderon de la Barca</Autor>
    </Libro>
</Libros>
```

Libros1.dtd

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

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE Libros SYSTEM "Libros1.dtd">
<Libros>
    <Libro>
        <Titulo>Don Quijote de la Mancha</Titulo>
        <Autor>Miguel de Cervantes</Autor>
    </Libro>
    <Libro>
        <Titulo>La vida es suenno</Titulo>
        <Autor>Calderon de la Barca</Autor>
    </Libro>
</Libros>
```

DTD. Introducción

Definición de Elementos (<!ELEMENT>):

- Sintaxis:

<!ELEMENT nombre_elemento contenido>

- Contenido puede ser:

- EMPTY: Elemento vacío.
- #PCDATA: Texto.
- Otros elementos, (por ejemplo, (elemento1, elemento2*)).

Ejemplo: <!ELEMENT Titulo (#PCDATA)>

DTD. Introducción

Definición de Atributos (<!ATTLIST>):

Sintaxis:

<!ATTLIST nombre_elemento nombre_atributo tipo valor_predeterminado>

- Tipos comunes:
 - CDATA: Texto sin restricciones.
 - ID: Identificador único.
 - IDREF: Referencia a un ID.
- Ejemplo: <!ATTLIST Libro id ID #REQUIRED>

DTD. Introducción

Elemento raíz:

Todo DTD debe tener uno y sólo un element raíz (también conocido como elemento documento).

Este elemento raíz debe coincidir con el nombre que aparece a continuación del DOCTYPE.

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

```
<!DOCTYPE Libros [
```


EJERCICIO

Definir un DTD para un documento XML que contenga información de un libro, con elementos como Título y Autor.

Ejercicio0

```
<Libros>
  <Libro>
    <Titulo>El Principito</Titulo>
    <Autor>Antoine de Saint-Exupéry</Autor>
  </Libro>
</Libros>
```

Ejercicio1

```
<biblioteca>
  <libro id="L001">
    <titulo>El Gran Gatsby</titulo>
    <autor>F. Scott Fitzgerald</autor>
    <año>1925</año>
  </libro>
  <libro id="L002">
    <titulo>Cien Años de Soledad</titulo>
    <autor>Gabriel García Márquez</autor>
    <año>1967</año>
  </libro>
  <!-- Agrega más libros aquí -->
</biblioteca>
```

DTD. Definición de Secuencias y Elecciones

•Secuencias:

- Uso de la coma , para definir una secuencia de elementos.
- `<!ELEMENT Persona (Nombre, Apellido)>`
 - define que Nombre debe ir antes de Apellido.

•Elecciones:

- Uso de la barra vertical | para definir opciones entre elementos.
- `:<!ELEMENT Contacto (Email | Teléfono)>`
 - significa que se puede tener un Email o un Teléfono, pero no ambos simultáneamente.

•**Ejercicio Práctico:** Definir un DTD que use secuencias y elecciones para un documento XML que represente una persona con un nombre, apellido y opcionalmente un contacto por email o teléfono.

DTD. Definición de Secuencias y Elecciones


•Ejercicio2

Definir un DTD que use secuencias y elecciones para un documento XML que represente una persona con un nombre, apellido y opcionalmente un contacto por email o teléfono.

```
<Persona>
  <Nombre>Juan</Nombre>
  <Apellido>Pérez</Apellido>
  <Contacto>
    <Email>juan.perez@example.com</Email>
  </Contacto>
</Persona>

<Persona>
  <Nombre>María</Nombre>
  <Apellido>González</Apellido>
  <Contacto>
    <Telefono>123-456-7890</Telefono>
  </Contacto>
</Persona>

<Persona>
  <Nombre>Carlos</Nombre>
  <Apellido>Rodríguez</Apellido>
</Persona>
```



DTD. Definición de Secuencias y Elecciones

•Ejercicio2

•DTD:

- <!ELEMENT Persona (Nombre, Apellido, Contacto?)>: Define que Persona debe tener un Nombre y un Apellido, con un Contacto opcional.
- <!ELEMENT Contacto (Email | Telefono)>: Define que Contacto puede ser Email o Telefono, pero no ambos simultáneamente.
- <!ELEMENT Nombre (#PCDATA)>, <!ELEMENT Apellido (#PCDATA)>, <!ELEMENT Email (#PCDATA)>, <!ELEMENT Telefono (#PCDATA)>: Define que estos elementos deben contener datos de texto.

•XML:

- El primer Persona tiene un Nombre, un Apellido y un Contacto con un Email.
- El segundo Persona tiene un Contacto con un Telefono.
- El tercer Persona solo tiene Nombre y Apellido, sin contacto.

DTD. Indicadores de repetición

- *: Cero o más ocurrencias.
- +: Una o más ocurrencias.
- ?: Cero o una ocurrencia.

<!ELEMENT Nombres (Nombre+)>

significa que debe haber al menos un Nombre.

DTD. Indicadores de repetición

•Ejercicio3

Definir un DTD que use indicadores de ocupación para un documento XML que represente una lista de tareas, cada una con una o más subtarea

```
<!DOCTYPE ListaDeTareas SYSTEM "lista_de_tareas.dtd">
<ListaDeTareas>
  <Tarea>
    <Titulo>Preparar la presentación</Titulo>
    <Subtareas>
      <Subtarea>Crear diapositivas</Subtarea>
      <Subtarea>Ensayar presentación</Subtarea>
    </Subtareas>
  </Tarea>
  <Tarea>
    <Titulo>Organizar el evento</Titulo>
    <Subtareas>
      <Subtarea>Reservar lugar</Subtarea>
      <Subtarea>Contratar servicios de catering</Subtarea>
    </Subtareas>
  </Tarea>
  <Tarea>
    <Titulo>Comprar regalos</Titulo>
    <Subtareas>
      <Subtarea>Seleccionar tiendas</Subtarea>
    </Subtareas>
  </Tarea>
</ListaDeTareas>
```



DTD. Indicadores de repetición

•Ejercicio3

•DTD:

- <!ELEMENT ListaDeTareas (Tarea+)>: Define que el elemento raíz debe contener una o más Tarea.
- <!ELEMENT Tarea (Titulo, Subtareas)>: Cada Tarea tiene un Titulo y un bloque de Subtareas.
- <!ELEMENT Subtareas (Subtarea+)>: El elemento Subtareas debe contener una o más Subtarea.
- <!ELEMENT Titulo (#PCDATA)>, <!ELEMENT Subtarea (#PCDATA)>: Estos elementos contienen solo texto (#PCDATA).

•XML:

- El XML tiene una ListaDeTareas con varias Tarea, cada una con un Titulo y una lista de Subtareas.
- Algunas tareas tienen más de una Subtarea, y otras solo una.

DTD. Elementos Vacíos y #PCDATA

- **Elementos Vacíos:**

- Usar EMPTY para elementos sin contenido.
- Ejemplo: <!ELEMENT EspacioVacio EMPTY>.

- **Contenido de Texto (#PCDATA):**

- Uso de #PCDATA para elementos que contienen texto simple.
- Ejemplo: <!ELEMENT Descripción (#PCDATA)>.

DTD. Elementos Vacíos y #PCDATA


•Ejercicio4

Crear un DTD que use elementos vacíos y elementos de texto para representar un documento XML que contenga información de un producto, donde algunos elementos pueden estar vacíos y otros contener texto

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Producto SYSTEM "producto.dtd">
<Producto>
  <Nombre>Laptop XYZ</Nombre>
  <Descripcion>Una laptop de alto rendimiento</Descripcion>
  <Precio>999.99</Precio>
  <Stock/>
</Producto>

<Producto>
  <Nombre>Teléfono ABC</Nombre>
  <!-- Sin descripción -->
  <Precio>499.99</Precio>
  <!-- Sin stock -->
</Producto>

<Producto>
  <Nombre>Tablet 123</Nombre>
  <Descripcion>Una tablet ligera y versátil</Descripcion>
  <Precio>299.99</Precio>
  <Stock/>
</Producto>
```



DTD. Elementos Vacíos y #PCDATA

•Ejercicio4

•DTD:

- <!ELEMENT Producto (Nombre, Descripcion?, Precio, Stock?)>: Define que un Producto debe tener un Nombre, un Precio, y opcionalmente una Descripcion y Stock.
- <!ELEMENT Nombre (#PCDATA)>, <!ELEMENT Descripcion (#PCDATA)>, <!ELEMENT Precio (#PCDATA)>: Estos elementos deben contener texto.
- <!ELEMENT Stock EMPTY>: El elemento Stock puede ser un elemento vacío.

•XML:

- El XML muestra varios productos. Algunos tienen todos los elementos, mientras que otros tienen elementos opcionalmente vacíos.
- En el primer producto, Stock está vacío para indicar que no contiene datos, mientras que los demás elementos tienen texto.
- El segundo producto carece de Descripcion y Stock, mostrando que pueden estar ausentes o vacíos.
- El tercer producto tiene Descripcion con texto y Stock vacío

DTD. Opciones con Atributos

Identificadores Únicos: ID

- ID garantiza que el valor del atributo es único dentro del documento XML.
- Sintaxis: <!ATTLIST nombre_elemento nombre_atributo ID #REQUIRED>.
- Uso Práctico: Identificar elementos por un identificador único.

•Ejemplo

<!ELEMENT Persona (Nombre, Apellido)>

<!ATTLIST Persona id ID #REQUIRED>

DTD. Opciones con Atributos

Referencias: IDREF e IDREFS

•IDREF:

- Se utiliza para referenciar un ID existente en otro elemento.
- Sintaxis:
 - <!ATTLIST nombre_elemento nombre_atributo IDREF #REQUIRED>.
- Ejemplo

```
<!ELEMENT Relacion (Persona1, Persona2)>
```

```
<!ATTLIST Relacion Persona1 IDREF #REQUIRED>
```

```
<!ATTLIST Relacion Persona2 IDREF #REQUIRED>
```

IDREFS:

- Permite referenciar múltiples ID.
 - Sintaxis:
 - <!ATTLIST nombre_elemento nombre_atributo IDREFS #REQUIRED>.
 - Ejemplo
- ```
<!ELEMENT Grupo (Miembros)>
```
- ```
<!ATTLIST Grupo Miembros IDREFS #REQUIRED>
```

DTD. Opciones con Atributos

Entidades: ENTITY y ENTITIES

•ENTITY:

- Se utiliza para definir una entidad que representa datos externos.
- Sintaxis:
 - <!ENTITY nombre_entidad "valor">.
- Ejemplo
`<!ENTITY logo "logo.png">`

ENTITIES:

- Permite definir múltiples entidades.
- Sintaxis:
`<!ATTLIST nombre_elemento nombre_atributo ENTITIES #REQUIRED>.`
- Ejemplo
`<!ELEMENT Documento (Contenido)>`
`<!ATTLIST Documento imagenes ENTITIES #REQUIRED>`

DTD. Opciones con Atributos

Tokenización: NMTOKEN y NMTOKENS

•NMTOKEN:

- Se utiliza para restringir un atributo a un valor tokenizado (sin espacios).
 - <!ATTLIST nombre_elemento nombre_atributo NMTOKEN #REQUIRED>.
 - Ejemplo
- <!ELEMENT Usuario (Nombre)>
- <!ATTLIST Usuario rol NMTOKEN #REQUIRED>

NMTOKENS:

- Permite definir múltiples valores tokenizados.
 - <!ATTLIST nombre_elemento nombre_atributo NMTOKENS #REQUIRED>.
 - Ejemplo
- <!ELEMENT Grupo (Usuarios)>
- <!ATTLIST Grupo roles NMTOKENS #REQUIRED>

DTD. Opciones con Atributos

Atributos con Valores Predeterminados, #REQUIRED, #IMPLIED, y #FIXED

•#REQUIRED:

- Atributo obligatorio.
- <!ATTLIST nombre_elemento nombre_atributo tipo #REQUIRED>.

•#IMPLIED:

- Atributo opcional.
- <!ATTLIST nombre_elemento nombre_atributo tipo #IMPLIED>.

•#FIXED:

- Atributo con un valor fijo.
- <!ATTLIST nombre_elemento nombre_atributo tipo #FIXED "valor">.

•Valores Predeterminados:

- Atributo con valor por defecto.
- Sintaxis: <!ATTLIST nombre_elemento nombre_atributo tipo "valor">.

DTD. Ejercicio 5

•Ejercicio5

•Creación de un DTD Complejo:

•sistema de gestión de proyectos. En este ejemplo, tendremos proyectos, tareas, usuarios y relaciones entre ellos, con referencias y atributos de varios tipos.

```
<Proyecto>
  <Nombre>Proyecto A</Nombre>
  <Tareas>
    <Tarea id="T1">
      <Titulo>Desarrollar aplicación</Titulo>
      <Responsable>
        <IDPersona>U1</IDPersona>
      </Responsable>
    </Tarea>
    <Tarea id="T2">
      <Titulo>Probar aplicación</Titulo>
      <Responsable>
        <IDPersona>U2</IDPersona>
      </Responsable>
    </Tarea>
  </Tareas>
  <Usuarios>
    <Usuario id="U1" rol="desarrollador">
      <NombreUsuario>Juan Pérez</NombreUsuario>
    </Usuario>
    <Usuario id="U2" rol="tester">
      <NombreUsuario>María González</NombreUsuario>
    </Usuario>
  </Usuarios>
```

