

# DTD Document type descriptor

Juan Ignacio Raggio

May 3, 2025

## Contents

<b>1</b>	<b>Que es DTD?</b>	<b>1</b>
<b>2</b>	<b>Reglas para declarar elementos</b>	<b>2</b>
2.1	Palabras reservadas -> Reescribirlo sin tabla . . . . .	2
2.1.1	Ejemplos . . . . .	2
<b>3</b>	<b>Reglas para declarar Atributos</b>	<b>3</b>
<b>4</b>	<b>Como usar DTD?</b>	<b>3</b>
4.1	Inline . . . . .	3
4.2	Documento externo . . . . .	3
<b>5</b>	<b>Limitaciones de DTD</b>	<b>3</b>

## 1 Que es DTD?

- Es una extension de gramaticas libres de contexto (extended context-free grammar)
- Se define un vocabulario con elementos posibles
- Se crean reglas que indican en que derivan ( NombreElemento -> Expresion regular )
- Hay una regla por cada ELEMENTO DISTINTO del XML
- A la derecha una expresion regular

## 2 Reglas para declarar elementos

`<!ELEMENT nombreElemento palabrareservada>`

### 2.1 Palabras reservadas -> Reescribirlo sin tabla

Modelo del contenido	Significado
EMPTY	Sin contenido ni subelementos, es una hoja
()	Dentro de los parentesis se colocan los elementos separados por: “,” Secuencia de elementos, orden estricto; “ ” Disyuncion
(#PCDATA)	El contenido es texto y no tiene subelementos. Este nodo tendra una hoja hijo

#### 2.1.1 Ejemplos

- `<!ELEMENT cliente EMPTY>`
  - ☐ `<cliente></cliente>`
  - ☐ `<cliente/>`
  - ☒ `<cliente>ITBA</cliente>` -> Pues tiene texto
- `<!ELEMENT cliente (#PCDATA)>`
  - ☐ `<cliente>ITBA</cliente>`
  - ☐ `<cliente/>`
  - ☒ `<cliente><nombre>ITBA</nombre></cliente>` -> Pues solo acepta texto
- Un ejemplo mas completo seria:
  - Se quiere construir un documento XML que tenga como raíz al elemento libro. Este debe contener 3 subelementos que deben aparecer exactamente una vez: titulo, listaautores y editorial, en ese orden. Listaautores no puede contener más que 5 sub-elementos autor pero al menos tiene que haber uno En base a esto, para escribir las reglas con DTD haríamos lo siguiente: `<!ELEMENT libro (titulo, listaautores, editorial)>` `<!ELEMENT titulo (#PCDATA)>` `<!ELEMENT listaautores (autor, autor?, autor?, autor?, autor?)>` `<!ELEMENT autor (#PCDATA)>` `<!ELEMENT editorial (#PCDATA)>`

### 3 Reglas para declarar Atributos

`<!ATTLIST elementoNombre atributoNombre1 atributoTipo1 valorDefault1  
atributoNombre2 atributoTipo2 valorDefault2 ... >`

- **elementoNombre:** Nombre del elemento al que perteneciera el atributo
- **atributoNombre:** Nombre del atributo, no puede repetirse para un mismo elemento
- **atributoTipo:**
  - **CDATA:** Cualquier texto, es el default
  - **():** Seleccion de entre algun valor de la lista alli mencionada. Los valores se separan con simbolo pipe
  - **ID:** Si el valor es unico dentro del documento. No puede haber mas de un atributo ID para un mismo elemento. **NO PUEDE USARSE JUNTO CON FIXED**
  - **IDREF:** El valor es un ID de otro elemento
  - **IDREFS:** El valor es una lista de otros IDs
  - **ENTITY, ENTITIES, NOTATION:** Es una asociacion entre cierto texto y un rotulo ( es como una macro en C )
- **valorDefault:**
  - **#REQUIRED:** El atributo es obligatorio
  - **#IMPLIED:** Opcional y no hay default asignado
  - **“valor”:** El valor del atributo es asignado automaticamente si no se especifica
  - **#FIXED“valor”:** Solo ese valor indicado es el valido

### 4 Como usar DTD?

#### 4.1 Inline

#### 4.2 Documento externo

### 5 Limitaciones de DTD

- Los contenidos son texto sin tipo ( int? date?)

- El DTD no es en sí un documento XML (doble parser)
- La frecuencia repetitiva sólo permite + o \* (no exacta)
- Para poder expresar un elemento mixto se pierden las validaciones de orden de los elementos
- El contenido del tipo ID no puede empezar con un número