

# **DESARROLLO DE APLICACIONES WEB / MULTIPLATAFORMA**

## **UD 1 Introducción a los lenguajes de marcas**

### **LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE INFORMACIÓN**

**JOSÉ HIPÓLITO BOGAS PERONA**

## Indice

0.- Objetivos.....	3
1.- Introducción.....	3
1.1.- Introducción a los lenguajes de marcas.....	3
1.2.- Contexto de los lenguajes de marcas.....	5
1.3.- Lenguajes de marcas: estructura, formatos y características.....	9
1.4.- Lenguajes de marcas y dialectos más comunes.....	11
1.4.1.- HTML.....	11
1.4.2.- XML.....	12
1.4.3.- XHTML.....	13
1.4.4.- JSON.....	14
1.4.5.- SVG.....	15
1.4.6.- MathML.....	15
1.4.7.- NewsML.....	16
1.4.8.- FXML.....	16
1.4.9.- XAML.....	16
1.4.10.- Android.....	16
1.4.11.- Servicios web.....	17
1.4.12.- RSS y Atom.....	18
1.4.13.- XSL.....	18
1.4.14.- Markdown.....	18
1.5.- Espacio de nombres.....	19
1.6.- Bibliografía y webgrafía.....	20

## **0.- Objetivos**

1. Descubrir el significado del término “lenguaje de marcas”.
2. Conocer el origen y la motivación que llevó a crear los lenguajes de marcas actuales.
3. Concretar los lenguajes de marcas más utilizados en la actualidad y cuál es el uso principal de cada uno de ellos.
4. Exponer la estructura básica de los lenguajes de marcas más comunes.
5. Enumerar algunas de las herramientas disponibles para consultar o editar ficheros escritos con los lenguajes de marcas utilizados habitualmente.
6. Conocer los espacios de nombres.

## **1.- Introducción**

Los lenguajes de marcas existen prácticamente desde el comienzo de la informática. Cubren la necesidad de disponer de lenguajes sencillos para el intercambio de información entre distintos sistemas. Con la llegada de *Internet* algunos de los lenguajes de marca se convirtieron en piedras angulares sobre las que construir la World Wide Web.

Existe un gran número de lenguajes de marcas distintos y obviamente no todos son igual de importantes. No obstante, comparten algunos aspectos comunes: fundamentos, orígenes, características básicas y elementos relacionados con su estructura. Es importante conocer qué son, cómo han evolucionado y, de aquellos que han tenido más éxito que pueden proporcionar.

### **1.1.- Introducción a los lenguajes de marcas**

Un **lenguaje de marcas** o **lenguaje de marcado** es aquel que permite representar información que contiene, además de los datos propiamente dichos, marcas o etiquetas que indican cómo se estructuran estos datos, su significado o cómo debe representarse desde un puesto de vista gráfico o visual.

Los lenguajes de marcas no son lenguajes de programación. No contienen instrucciones ni implican acciones directas. Son reglas que organizan la información con el objeto de darle una estructura uniforme y facilitar su procesamiento automático.

Una clasificación simplificada de los lenguajes de marcas podría ser la siguiente:

- Lenguajes orientados a la presentación de información.
- Lenguajes orientados al almacenamiento y procesado de información.

El ejemplo más representativo de un lenguaje de marcas orientado a la presentación de información es el lenguaje de marcas de hipertexto (Hypertext Markup Language) conocido como HTML.

El etiquetado de HTML indica el programa que va a presentar en la pantalla la información (generalmente un navegador web o *web browser*) cómo debe hacer dicha presentación. Es decir, proporciona junto a los datos que va a mostrar, la información referente a cómo han de mostrarse. Aunque ya existían previamente, a partir de la versión 5 de HTML se incluye un importante número de etiquetas con lo que se denomina **información semántica**. Estas etiquetas indican, además de cómo ha de mostrarse la información, qué significado tiene desde el punto de vista estructural. En la unidad 2 se profundizará con el conocimiento de este tipo de etiquetado y las diferencias alternativas que proporciona.

Un documento que contenga un texto es perfectamente legible por nosotros aunque inapropiado para publicar en una página web. Si le aplicamos formato mediante un lenguaje de marcas como por ejemplo HTML obtenemos un archivo también legible pero más difícil de interpretar. De esta manera se encargará el navegador, o mejor dicho el agente de usuario, que es quien **interpreta** las marcas de formato y las aplica convenientemente al texto para dar lugar a las páginas web que será mucho más agradable que leer el texto original.

Una **marca** es una señal colocada dentro de un texto con el fin de delimitar una parte del mismo y en muchos casos aplicarle un determinado formato (aunque existen marcas con otros propósitos ).

Las marcas más comunes están formadas por una palabra que describe su función encerrada entre símbolos menor < y mayor >, como <html>

Es muy habitual que aparezcan por parejas una al comienzo y otra al final. Por ejemplo:

<h1>Este texto aparece más grande </h1>

<h2> Y este otro texto aparece más pequeño </h2>

### ACTIVIDAD PROPUESTA 1.1 Saludo.

Guarda del texto anterior en el archivo saludo.txt, ábrelo con el navegador y comprueba que el texto permanece igual. Renombra el archivo como saludo.html y comprueba cómo interpreta las marcas el navegador.

Se puede observar, por lo tanto, que la definición de un lenguaje de marca está al alcance de cualquiera y no tiene por qué suponer mucho esfuerzo. El esfuerzo sería necesario en la creación de los programas que leyesen e interpretasen las etiquetas correctamente: habría que hacer tantos programas como lenguajes de marca existentes. Por fortuna, como se ha dicho con anterioridad ya existe una serie de lenguajes de marcas capaces de cubrir prácticamente todas las necesidades posibles gracias a su flexibilidad. Además, no requieren la elaboración de sofisticados programas ni aplicaciones de procesamiento porque ya están creadas. Lenguajes como HTML y XML, junto con sus dialectos, están estandarizados y son capaces de satisfacer cualquier necesidad referente a cómo se debe estructurar los datos o cómo deben representarse.

## 1.2.- Contexto de los lenguajes de marcas

El origen de estos lenguajes se remonta prácticamente al origen de la computación moderna. En el mismo momento en que se comenzó a querer almacenar información para procesarla por un programa informático, se detectó la necesidad de dar formato a dicha información. Un dato etiquetado con una determinada marca facilita su proceso automático.

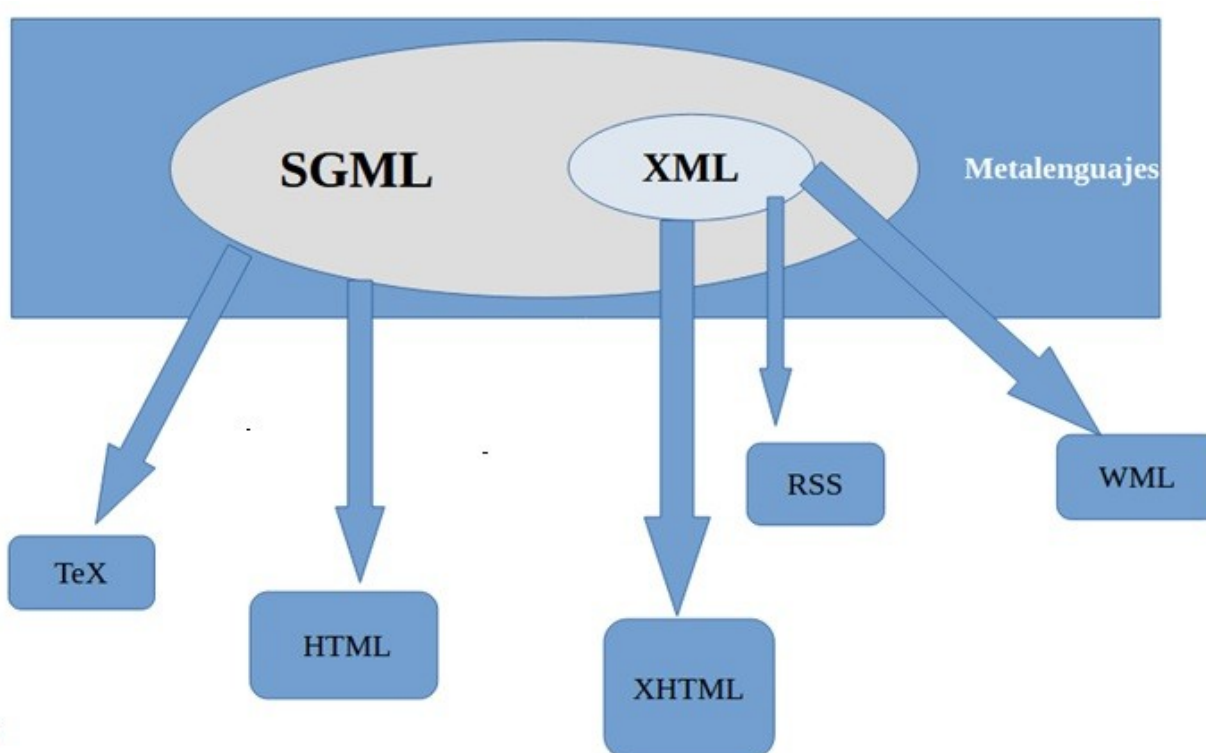
El problema surge cuando cada programa utiliza su propio lenguaje, ya que obliga, a la creación de un conversor de cada par de programas con los que se desea compartir información. Por tanto, esto impide compartir la información de forma ágil y sencilla, algo que en el mundo académico y empresarial suele ser un objetivo fundamental desde desde que se comenzaron a utilizar los ordenadores y los sistemas informáticos.

En los años 60 las empresas de publicación y manejo de documentos electrónicos tenían el problema de **falta de compatibilidad entre aplicaciones**. El problema existente era que cada aplicación utilizaba sus propias marcas para describir los diferentes elementos, esto impedía el intercambio de documentos entre plataformas. Otra carencia importante era la **separación entre estructura y aspecto del documento**.

IBM empresa pionera en investigación en informática y electrónica intentó resolver estos problemas a través de un lenguaje de marcas denominado **GML** (Generalized markup language)

GML independiza el documento del dispositivo que lo va a utilizar usando marcas genéricas. GML incorpora marcas descriptivas para estructura de documento que permite distinguir entre texto, listas, tablas, etcétera. El mismo documento puede entonces ser utilizado por varios dispositivos simplemente especificando un perfil para cada uno.

En 1986 GML pasó a manos de ISO Se convirtió en **SGML** (ISO 8879) Standard Generalized Markup Language, software libre y de código abierto.



SGML no es estrictamente un lenguaje sino un **metalenguaje**, es decir, un conjunto de normas que permiten crear otros lenguajes de marcas. Para ello, se define un vocabulario o conjunto de elementos a utilizar y una gramática o conjunto de reglas que rigen el uso de los elementos y sus atributos.

### Ventajas :

- Reutilización de datos
- integridad y control sobre los datos,
- portabilidad,
- adaptabilidad

### Inconvenientes:

- alta complejidad

Un documento SGML tiene dos partes:

- El **prólogo**: Contiene la estructura.
  - La declaración: indica que el documento es SGML y algunos parámetros.
  - La definición del tipo de documento (DTD): Indica la sintaxis particular del lenguaje creado.
- La **instancia del documento**: contiene los datos.

En los siguientes ejemplos se ha omitido la declaración SGML, la DTD sería el vocabulario y las reglas de uso, y la instancia de documento serían los datos.

Ejemplo 1: Los módulos del ciclo Asir

Vocabulario: asir, Módulo, título, contenido, unidad.

Reglas: asir contiene varios módulos, un módulo tiene un elemento simple de título y un elemento contenido, contenido tiene varias unidades y toda unidad debe estar en un contenido, las unidades son texto simple, detrás de cada unidad solo puede ir otra unidad o fin de contenido, detrás de un módulo sólo puede ir otro módulo o fin de asir .

<asir>

<modulo><titulo> Lenguajes de Marcas </titulo>

<contenido>

<unidad> 1. Introducción </unidad>

<unidad> 2. HTML</unidad>

<unidad> 3. CSS </unidad>

<unidad> 4. XML</unidad>

<unidad> 5. XLST</unidad>

<unidad> 6. Sindicación </unidad>

<unidad> 7. ERP </unidad>

</contenido>

</modulo>

...

</asir>

### Ejemplo 2 : Mini HTML

Vocabulario: html, head, title, body, p

Reglas: html contiene el elemento head y un elemento body, head contiene un elemento simple title, body contiene varios elementos p y todo elemento p debe estar en body, p es texto simple, detrás de p solo puede ir otro p o fin de body, detrás de body solo puede ir fin de html.

<html>

    <head> <title> Mi página</title></head>

    <body>

        <p> Hola mundo</p>

        <p> Esta es mi página web </p>

    </body>

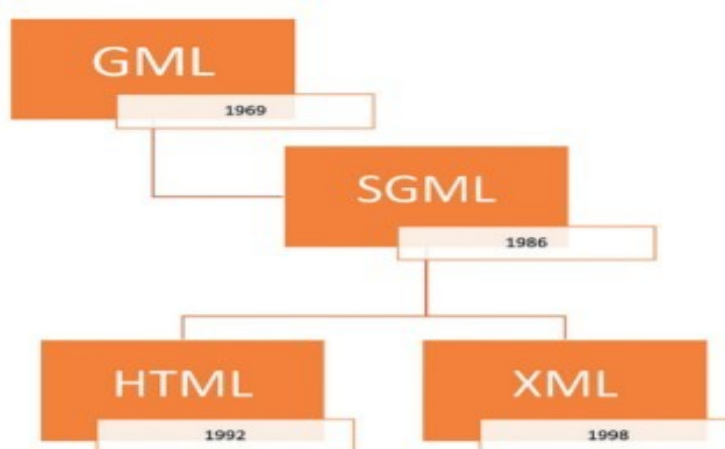
</html>

---

### ACTIVIDAD PROPUESTA 1.2 Mi primer documento SGML.

Crea tu propio documento SGML indicando el vocabulario y las reglas usadas.

---





### 1.3.- Lenguajes de marcas: estructura, formatos y características

Existen multitud de lenguajes de marcas y esto provoca que las estructuras y los formatos sean también numerosos. No existe, por lo tanto, una estructura común o un único formato. Por definición, cuando se habla de lenguajes de marcas, se está haciendo referencia a la información añadida mediante marcas a los datos propiamente dichos del documento.

Cada lenguaje tiene su propia estructura y formato. No obstante, con el objetivo de ilustrar, se presenta a continuación parte de la estructura de uno de los lenguajes de marcas más importantes que no es otro que XML. En la unidad cuatro se mostrará en detalle la estructura completa.

La estructura parcial y simplificada un documento XML es la siguiente:

- Cada elemento de un documento XML está precedido por una etiqueta de apertura compuesta por un nombre y rodeada por los símbolos < y >.
- Cada elemento de un documento XML está seguido por una etiqueta de cierre compuesta por el mismo nombre de la etiqueta de apertura y rodeada por los símbolos </ y >.
- El contenido de un elemento de un documento XML puede ser un texto, o 1 o más elementos.
- Sólo existe un elemento raíz.
- Un atributo está compuesto de un nombre del atributo, el símbolo igual y el valor del atributo entrecomillado.
- Un elemento puede tener entre 0 y n atributos.

Por su parte, el formato en el que está escrito un documento XML, es texto plano. Un ejemplo de un documento XML de acuerdo con esta estructura podría estar almacenado en un fichero denominado *misjuegos.xml* y tener el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<videojuegos>
  <videojuego multijugador="si">
    <titulo>Call Of Duty</titulo>
    <plataforma>Play Station 5</plataforma>
  </videojuego>
  <videojuego multijugador="si">
    <titulo>Fortnite</titulo>
    <plataforma>XBox One</plataforma>
  </videojuego>
  <videojuego multijugador="no">
    <titulo>Manic Miner</titulo>
    <plataforma>Amstrad CPC</plataforma>
  </videojuego>
</videojuegos>
```

Como se puede deducir los nombres de las etiquetas y de las propiedades no están definidas en ningún estándar. Es el autor del documento quien decide cómo se han de llamar dichas etiquetas, cuál es su orden y cómo se estructuran entre sí. Esta flexibilidad es debida a que como se ha comentado anteriormente, XML es un metalenguaje. En cualquier caso, la estructura ha sido respetada y eso convierte a este documento en un documento XML válido.

Este fichero está sujeto a una estructura y se almacena en un formato susceptible de ser leída y procesado por un programa escrito en cualquier lenguaje de programación y ejecutada en cualquier sistema operativo. Por su parte, HTML siendo también un lenguaje surgido a raíz de ese SGML, no es un metalenguaje como XML. Tiene también una estructura bien definida (similar a la de XML, pero no idéntica), aunque, al tratarse de un lenguaje concreto, tiene definidos con precisión los nombres de etiquetas y de los atributos.

En el siguiente ejemplo se muestra una sección de un documento HTML. Las etiquetas `<table>`, `<tr>` y `<td>` están definidas en la especificación del lenguaje HTML, así como el atributo `class`.

```
<table class="fondoGris">
  <tr>
    <td>Elemento 1</td>
    <td>Elemento 2</td>
  </tr>
  <tr>
    <td>Elemento 3</td>
    <td>Elemento 4</td>
  </tr>
</table>
```

En las unidades 2 y 4 se profundizará en las especificaciones formales de estos dos lenguajes así como de los dialectos derivados y de otros lenguajes relacionados.

Las características fundamentales de los lenguajes de marca son las siguientes:

- **Independencia.** Su uso no debe depender de ninguna plataforma hardware ni sistema de software concreto.
- **Almacenados en texto plano.** El documento está compuesto únicamente por caracteres de texto.
- **Flexibilidad.** Los lenguajes de marcas deben ser lo suficientemente flexibles para poder utilizarse en diferentes contextos.
- **Compactibilidad.** Las marcas y el contenido se encuentran en el mismo documento.
- **Facilidad de procesamiento.** El formato debe ser diseñado de tal manera que el procesamiento automático sea sencillo.

### 1.4.- Lenguajes de marcas y dialectos más comunes

La web y las aplicaciones desarrolladas sobre la tecnología de Internet han supuesto un importante incremento en el uso de los lenguajes de marcas tanto para representar información como para intercambiar datos entre sistemas. Además, existen diversos campos en los que los lenguajes de marcas tienen una utilidad importante como pueden ser la configuración de aplicaciones o la definición de interfaces gráficas de usuario.

Actualmente, los lenguajes de marcas más habituales tienen como objetivos principales la representación de información (HTML y XHTML) o el almacenamiento e intercambio de información (XML y sus dialectos) y el formato de intercambio (JSON).

En este apartado se van a presentar algunos de los lenguajes de marcas y dialectos de uso más común en la actualidad, pero no son los únicos. En XML, en concreto, existen multitud de dialectos para prácticamente cualquier campo del conocimiento .

#### 1.4.1.- HTML

Es el lenguaje de marcas más conocido y utilizado y el **World Wide Web Consorcium (W3C)** se encarga de su mantenimiento y estandarización. HTML es un lenguaje destinado a la creación de páginas web que nos permite mezclar en un mismo documento texto y contenido multimedia. Además dispone de una potente

herramienta que son los **enlaces o hipervínculos**, para conectar con otras páginas cuyo contenido está relacionado con la página actual.

Actualmente HTML, se encuentra en la versión 5.2 con fecha de 14 de diciembre de 2017. La versión 5.3 con fecha 18 de octubre de 2018 se halla en borrador y sujeta a revisión y discusión. Independientemente de la versión concreta, coloquialmente se hace referencia a todas las versiones 5.x y las que puedan aparecer de HTML como **HTML 5**.

Su objetivo principal sigue siendo el concebido por Tim Berners Lee: proporcionar un mecanismo para diseñar páginas web que van a ser representadas en un navegador, ya sea este ejecutado en un ordenador, en una videoconsola o en un dispositivo móvil.

El éxito de HTML 5 es indudable. Los diseñadores gráficos cuenta con una capacidad ilimitada de elementos de maquetación gracias al uso conjunto con las hojas de estilo en cascada CSS.

También los programadores de aplicación web han podido aprovechar las mejoras que se han ido incorporando al lenguaje de marcado HTML para programar aplicaciones más funcionales y consistentes

La extensión de los ficheros HTML suele ser **.html** o **.htm**.

Para ver el código HTML de una página *web* hay que pulsar el botón derecho sobre esta y seleccionar la opción «ver código fuente».

Además del navegador necesario para ver los resultados de nuestro trabajo necesitamos otra herramienta capaz de crear la página. Un archivo HTML es texto plano, por lo que para escribir en HTML necesitamos un simple editor de texto. Existen aplicaciones específicas para la creación de páginas web, quizás la más conocida es Dreamweaver, que nos ofrece muchas funcionalidades para aumentar nuestra productividad.

Es aconsejable en un principio utilizar una herramienta lo más sencilla posible para poder prestar la máxima atención a nuestro código en lugar de a la aplicación. Ejemplo: Notepad++, Software libre bajo licencia pública general de GNU para Windows, Brackets, etc.

### 1.4.2.- XML

XML (lenguaje de marcado extensible o Extensible Markup Language) es el lenguaje de intercambio de información entre sistemas por excelencia. Se puede afirmar que XML es para los sistemas el equivalente a HTML para los humanos. Un documento XML puede servir para almacenar la configuración de una aplicación, los mejores

marcadores de un videojuego, el diseño de una interfaz de usuario o la información de un gráfico vectorial.

La primera versión de XML salió a la luz en 1998. Es, al igual que HTML, una especificación del W3C y se define como «de propósito general». Esto significa que es un metalenguaje al venir definido por reglas y no por etiquetas concretas. Tomando la especificación de XML se pueden definir dialectos específicos para una gran variedad de aplicaciones de todo tipo.

La última versión vigente de XML actualmente es La 1.1. Debido a su naturaleza flexible y no específica, XML no requiere de revisiones ni mejoras frecuentes, ya que alberga en sí mismo la capacidad suficiente para adaptarse a las necesidades presentes y futuras.

La extensión de los ficheros XML suele ser **.xml**.

Las extensiones son un añadido que se agrega al nombre del fichero para facilitar su identificación pero no obliga a que el contenido del fichero sea acorde a dicha extensión. Por lo tanto, cualquier extensión puede contener todo tipo de formato.

### 1.4.3.- XHTML

El lenguaje de marcas XHTML (*eXtensible Hypertext Markup Language*) se puede definir de manera simplificada como la versión en formato XML del lenguaje de marcas HTML. Se presentó en el año 2000 como una recomendación del W3C.

XHTML tiene la misma relación con XML que HTML con SGML. Tanto HTML como XHTML surgen a partir de un metalenguaje y ambos son lenguajes de marcado para el desarrollo de páginas web. La diferencia fundamental es que XHTML tiene una sintaxis más estricta y es más formal que HTML.

HTML tiene una serie de características que impiden que sea lo que se denomina **documento XML bien formado** como por ejemplo la etiqueta <br>. Esta etiqueta indica al navegador web que tiene que mostrar un salto de línea y no tiene etiqueta de cierre, ya que no debe contener elementos en su interior. En XML todas las etiquetas de apertura deben cerrarse por lo cual el uso de esta etiqueta no sería aceptado. En la versión XML de HTML (XHTML) la etiqueta <br> se sustituye por **<br />**.

Además de la obligatoriedad de cerrar todas las etiquetas, el documento XHTML debe cumplir con todas las características de un documento XML bien formado y esto implica ciertos cambios. Estos cambios no restan ni un ápice de capacidad de diseño y maquetación, ya que todas las posibilidades que proporciona HTML se mantienen con XHTML.

El hecho de que XHTML sea un HTML «adaptado a XML» aparentemente no hace ninguna aportación a este último. Entonces ¿por qué emplear esfuerzo en crear un

nuevo lenguaje de marcas para hacer lo mismo que uno que ya existe? Como cabe esperar, existen razones y estas tienen que ver con el éxito de HTML como lenguaje para la creación de páginas web.

HTML tiene una sintaxis definida por un estándar, tal y como se ha dicho previamente, pero son los navegadores los programas que procesan los documentos y los representan en pantalla. Los navegadores, cuando encuentran un documento HTML mal construido (nombres de etiquetas incorrectas, atributos inexistentes, errores en la estructura del documento, etcétera), interpretan (“se inventan”) lo que quería decir el autor de la página. La mayoría de las veces acierta y este funcionamiento ha sido una gran contribución al éxito de la web: a más sencillez, más uso. Por otra parte, si la interpretación es errónea, el autor podrá ver en su navegador que la página está mal construida o, como mal mayor, la persona que accede a la página verá una página mal maquetada.

Como consecuencia negativa, una página HTML mal construida es difícil de procesar por un programa informático, ya que éstos no son tan flexibles como las personas. Si un programa (“se inventa”) qué es la cantidad de un producto en un listado y qué es el precio las consecuencias pueden ser imprevisibles.

A continuación se muestra un ejemplo de documento XHTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Título del documento</title>

</head>
<body>
    <h1>Lenguajes de Marcas y Sistemas de Gestión de la Información</h1>
    <br />
    <h2>Unidad 1. Introducción</h2>
    <p>Texto de la unidad</p>
</body>
</html>
```

### **1.4.4.- JSON**

JSON es el acrónimo de *Javascript Object Notation* («notación de objetos de Javascript»). En realidad, es el formato de texto en el que Javascript representa sus objetos incluidos los *arrays*. Pese a esa naturaleza original tan específica, JSON ha abandonado su vínculo con Javascript y se ha universalizado como alternativa a XML para realizar almacenamiento e intercambio de datos entre sistemas

independientemente de la plataforma y del lenguaje de programación. Casi todos los lenguajes que se usan en la actualidad disponen de analizadores JSON.

En el siguiente código se puede observar un ejemplo de fichero en formato JSON:

```
{
  "asunto": "Lenguaje de Marcas",
  "publicacion": "2021",
  "editorial": "Paraninfo",
  "lenguajes": [
    {
      "nombre": "HTML",
      "uso": "Web"
    },
    {
      "nombre": "XML",
      "uso": "Intercambio de datos"
    },
    {
      "nombre": "JSON",
      "uso": "Intercambio de datos"
    }
  ]
}
```

JSON es más ligero y cuenta con una sintaxis más sencilla que XML, pero por el contrario no dispone de las herramientas y riquezas que sí tiene XML. El debate entre JSON y XML no debe existir: Hay contextos en los que JSON es más apropiado que XML y viceversa. De hecho, es frecuente encontrar ambas alternativas como viviendo en un mismo sistema.

La extensión de los ficheros JSON suele ser **.json**.

### 1.4.5.- SVG

SVG (Scalable Vector Graphics o gráficos vectoriales escalables) es un dialecto de XML. Se utiliza para representar gráficos vectoriales de dos dimensiones.

La extensión de los ficheros SVG suele ser **.svg** o **.svgz**.

### 1.4.6.- MathML

MathML (Mathematical Markup Language o lenguaje de marcado matemático) es un dialecto de XML. Se utiliza para representar notaciones matemáticas.



### 1.4.7.- NewsML

NewsML (News Markup Language o lenguaje de marcado noticias) es un dialecto de XML. Se utiliza para representar noticias electrónicas.

### 1.4.8.- FXML

FXML es un dialecto de XML. Se utiliza para representar las interfaces de usuario en la tecnología JavaFx del lenguaje de programación Java.

En el siguiente código se muestra un segmento de la definición de una interfaz gráfica utilizando este dialecto :

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

<?import javafx.scene.text.*?>
<?import javafx.scene.control.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane fx:id="paneSecurity" prefHeight="600.0" prefWidth="800.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/
fxml/1" fx:controller="com.lafabricadeandroides.websecfx.controller.
SecurityController">
```

### 1.4.9.- XAML

XAML (eXtensible Application Markup Language o lenguaje de marcado extensible de aplicación) es un dialecto de XML. Es el lenguaje en el que se definen las interfaces de usuario en las aplicaciones para el sistema operativo Windows desarrolladas con la tecnología de Microsoft.

El siguiente código se muestra la definición de parte de una interfaz gráfica desarrollada con XAML:

```
<?xml version="1.0" encoding="UTF-8"?>
<Button Content="Click me">
  <Button.Margin>
    <Thickness Left="10" Top="20" Right="10" Bottom="30"/>
  </Button.Margin>
</Button>
```



### 1.4.10.- Android

En las aplicaciones Android, la forma habitual de diseñar las interfaces de usuario es mediante ficheros escritos en un dialecto de XML. En estos ficheros se definen los esquemas de distribución, así como los componentes y sus características que van a permitir al usuario interactuar con la aplicación.

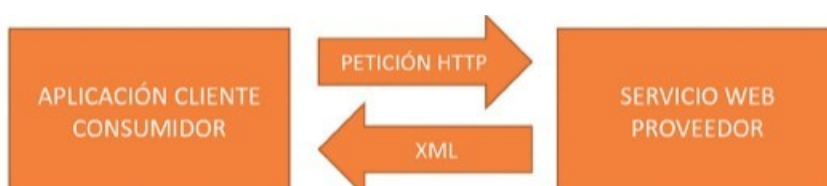
El siguiente código muestra un diseño simple de la interfaz gráfica de una aplicación Android:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

### 1.4.11.- Servicios web

Un servicio web (web service) es un componente software alojado en un servidor al que se accede utilizando protocolos estándares y tecnologías de la web como HTTP, HTTPS, XML o JSON.

Los denominados **servicios RESTful** (también llamados REST) utilizan con frecuencia dialectos de XML (en otras muchas ocasiones usan JSON) para el intercambio de información entre los puntos de la comunicación: el programa que proporcionan los datos (el servicio web) y el que los consume (la aplicación que hace de cliente en la comunicación). Estos dialectos pueden ser alguno de los dialectos existentes (los citados esta unidad y muchos más o creados a medida por un para un caso concreto.



Por otra parte, los servicios web denominados **SOAP** (Simple Object Access Protocol) utilizan dialectos concretos de XML, ya que la especificación de ese tipo de componente es mucho más precisa. Los formatos **WDSL** (Web Service Description Language o lenguaje de descripción de servicios web) o el utilizado para la creación de mensajes SOAP son algunos ejemplos.

### 1.4.12.- RSS y Atom

Tanto RSS (Really simple syndication o sindicación realmente simple) como Atom son dialectos de XML. Se utilizan para la distribución de contenidos de fuentes a los que los usuarios se han suscrito previamente, siendo Atom una alternativa a RSS creado con posterioridad.

El siguiente bloque de código muestra un segmento del documento de avisos de fenómenos meteorológicos de la Agencia Estatal de Meteorología AEMET en formato RSS:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>AEMET - Avisos de fenómenos meteorológicos adversos -
Extremadura</title>
    <link>http://www.aemet.es/es/eltiempo/prediccion/avisos?k=ext</link>
    <description>Avisos de fenómenos meteorológicos adversos según el Plan
Meteoalerta de la Agencia Estatal de Meteorología (AEMET) para Extremadura.
</description>
    <language>es</language>
```

### 1.4.13.- XSL

XSL (eXtensible stylesheet Language o lenguaje de hojas de estilo extensible) es una especificación del W3C utilizada para transformar y manipular documentos XML. Permite indicar cómo ha de presentarse la información contenida en un documento XML.

XSL consta de varias recomendaciones y varias de ellas se definen mediante documentos escritos en formato XML.

### 1.4.14.- Markdown

Markdown es un lenguaje de marcas ligero y de sintaxis sencilla, ideal para dar formato a textos de manera legible y atractiva. Se utiliza en diversos entornos desde documentación en repositorios hasta plataformas de trabajo colaborativo, permitiendo la inclusión de

listas, vínculos e imágenes. Aunque cuenta con pocos elementos, es una alternativa práctica cuando no se requiere la complejidad de otros lenguajes como HTML.

En conclusión, los lenguajes de marcas forman parte de los de las herramientas de los sistemas informáticos y de información. Presentan una gran diversidad tanto en su estructura como en su uso. Existen lenguajes muy completos, como HTML, que se utiliza para la creación de páginas web y otros más sencillos, como JSON, empleado principalmente para el almacenamiento y el intercambio de datos.

Cada lenguaje tiene sus propias características, ventajas e inconvenientes, por lo que no hay un lenguaje menor que otro, sino más apropiado para una necesidad concreta. Es necesario conocer lo más importante para poder determinar cuál utilizar en cada situación.

## 1.5.- Espacio de nombres

En el contexto de los lenguajes de marcas (como XML, HTML, SVG, MathML, etc.), un espacio de nombres (namespace) se puede definir así:

Un espacio de nombres es un **mecanismo** que **permite distinguir y organizar elementos y atributos** que pueden tener el mismo nombre pero pertenecer a vocabularios diferentes, evitando conflictos de ambigüedad.

- En XML, por ejemplo, distintos estándares o aplicaciones pueden definir elementos llamados igual (por ejemplo, <title> puede referirse al título de un libro en un vocabulario, o al título de una canción en otro).

- El espacio de nombres sirve como identificador único, normalmente expresado mediante una **URI** (*Uniform Resource Identifier*), que diferencia esos elementos aunque tengan el mismo nombre.

- Los espacios de nombres se declaran en los documentos mediante el **atributo xmlns** y un prefijo opcional.

Ejemplo en XML:

```
<libro xmlns:lit="http://ejemplo.org/literatura"
      xmlns:mus="http://ejemplo.org/musica">

  <lit:title>La Odisea</lit:title>
  <mus:title>Bohemian Rhapsody</mus:title>

</libro>
```

Aquí ambos elementos se llaman <title>, pero gracias al espacio de nombres lit y mus, sabemos a qué vocabulario pertenece cada uno.

En resumen:

Un **espacio de nombres** en los lenguajes de marcas es un **identificador único** (normalmente un **URI: Uniform Resources Identifier**) que **permite diferenciar elementos y atributos con el mismo nombre dentro de un documento**, pero pertenecientes a distintos vocabularios, evitando conflictos de ambigüedad entre los mismos y asegurando su interoperabilidad.

## 1.6.- Bibliografía y webgrafía

### **Bibliografía:**

Editorial Paraninfo. Lenguajes de marcas y sistemas de gestión de la información. Fernando Paniagua Martín. 2025

Editorial Garceta. Lenguajes de marcas y sistemas de gestión de la información 2ª edición. Mª Isabel Jiménez Cumbreiras. 2025

### **Webgrafía:**

HTML Tutorial: <https://www.w3schools.com/html/default.asp>

CSS Tutorial: <https://www.w3schools.com/css/default.asp>

W3.CSS Tutorial: <https://www.w3schools.com/w3css/default.asp>