

Tema 3. Principales lenguajes de programación para IA

La forma de representarlo en PROLOG es: tiene(coche,ruedas).

- ▶ Los nombres de objetos y relaciones deben comenzar con una letra minúscula.
- ▶ Primero se escribe la relación, y luego los objetos separados por comas y encerrados entre paréntesis.
- ▶ Al final de un hecho debe ir un punto (".").

El orden de los objetos dentro de la relación es arbitrario, pero es recomendable seguir una misma nomenclatura y mantener la coherencia a lo largo de la base de hechos.

Variables

No es variable con el concepto que se tiene de ella en la programación habitual. En Prolog, una variable representa el valor de un **Átomo**. Representan objetos que el mismo PROLOG determina.

Una variable puede estar instanciada o no instanciada. Estar instanciada cuando existe un objeto determinado representado por la variable. Los nombres de variables comienzan siempre por una letra mayúscula.

Un caso particular es la variable anónima, representada por el carácter subrayado ("_"). Es una especie de comodín que se utiliza en aquellos lugares que debería aparecer una variable, pero no interesa darle un nombre concreto ya que no va a ser utilizada posteriormente.

Tema 3. Principales lenguajes de programación para IA

Reglas

Las reglas se utilizan en PROLOG para significar que un hecho depende de uno o más hechos. Son la representación de las implicaciones lógicas del tipo $p \rightarrow q$ (p implica q).

- ▶ Una regla consiste en una cabeza y un cuerpo, unidos por el signo ":-".
- ▶ La cabeza está formada por un único hecho.
- ▶ El cuerpo puede ser uno o más hechos (conjunción de hechos), separados por una coma (", "), que actúa como el "y" lógico.
- ▶ Las reglas finalizan con un punto (".").

Tema 3. Principales lenguajes de programación para IA

3.9. Otros lenguajes y tecnologías

Los lenguajes anteriormente mencionados son solo algunos de los lenguajes de programación más utilizados para el desarrollo de programas de inteligencia artificial.

También existen otros lenguajes de programación y plataformas relevantes en esta área:

- ▶ C
- ▶ C++
- ▶ MATLAB
- ▶ Julia

Tema 4. Lenguajes de marcado. Información de sus etiquetas

4.1. Introducción y objetivos

Para estudiar este tema lee las Ideas Clave que se presentan a continuación y revisa los contenidos adicionales para mejorar la comprensión y aumentar los conocimientos sobre las materias tratadas.

Este tema supone un recorrido por los diferentes lenguajes de marcado más populares en la actualidad. Objetivos más concretos son:

- ▶ Comprender en qué consisten los **lenguajes de marcado** y sus diferencias con respecto a los lenguajes de programación.
- ▶ Conocer los diferentes **tipos de lenguaje de marcado**.
- ▶ Saber en qué situaciones se puede **aplicar un lenguaje de marcado** u otro.
- ▶ Conocer las **tecnologías relacionadas** con los lenguajes de marcado.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

4.2. Los lenguajes de marcado

Un **lenguaje de marcado** es un lenguaje legible por humanos utilizado para anotar un documento electrónico para que las máquinas puedan comprender mejor el estilo y la estructura del documento.

Un lenguaje de marcado electrónico utiliza **etiquetas** para definir diferentes elementos en un documento. Todos los humanos deben ser capaces de leer lenguajes de marcado, por lo que se basan principalmente en palabras reales inteligibles en lugar de la sintaxis ininteligible utilizada en los lenguajes máquina.

Muchos lenguajes de marcado utilizan conjuntos específicos de instrucciones que explican qué son las diferentes partes de un documento y cómo deben formatearse en la página. En el lenguaje de marcado más común, HTML, esto se hace usando herramientas conocidas como **etiquetas** y **elementos**.

Todos los lenguajes de marcado tienen una cosa en común: utilizan **etiquetas** (es decir, <>) para entrelazar el contenido de un documento con las instrucciones de marcado. Los corchetes angulares son los que marcan las etiquetas y lo que se encuentra entre las etiquetas está representado por el contenido formateado. El marcado puede ser semántico o de presentación, según su posición o uso en un documento.

Una **etiqueta** es responsable de mostrar dónde comienza y termina un elemento. Se utilizan en pares y constan de símbolos menor que (<) y mayor que (>) y divididos por caracteres que especifican las instrucciones. Estos incluyen una barra inclinada hacia atrás cuando las etiquetas se cierran. Una etiqueta de encabezado en HTML, que se usa para mostrar un título importante, sería así:

```
<title> </title>
```

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Las etiquetas son responsables de mostrar dónde comienza y termina un elemento.

Un **elemento** incluye etiquetas y lo que está escrito entre ellas. El elemento del encabezado anterior podría ser:

```
<title>Ejemplo de título</title>
```

Algunos de los **lenguajes de marcado más utilizados** son: XML, HTML, XHTML, MathML, SGML, KML. Muchos tienen en común las siglas ML en su nombre, que significa *Markup Language*.

Los lenguajes de marcado funcionan en un segundo plano. Cuando se abre un documento en línea no se ven las etiquetas ni elementos. En su lugar, se ve directamente el título y los textos de los párrafos con el formato que deberían tener, es decir, se visualiza un contenido renderizado.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

4.3. Tipos de lenguajes de marcado

Marcado semántico

Al marcado semántico también se le conoce como **marcado descriptivo**. El marcado semántico explica el **propósito** de un texto en una página, si se trata de un título, un párrafo, etc.

La **semántica** se refiere a la interpretación correcta del significado de una palabra u oración.

Usar una palabra semánticamente es usarla de una manera que esté alineada correctamente con el significado de la palabra. Cuando se usa mal una palabra, no se está usando semánticamente.

Muchas etiquetas HTML tienen un significado semántico. Es decir, el nombre de la etiqueta en sí transmite cierta información sobre el tipo de contenido entre las etiquetas de apertura y cierre.

Por ejemplo, cuando un navegador encuentra un encabezado h1, interpreta que esa etiqueta significa que el contenido del elemento h1 constituye el encabezado más importante de la sección que contiene el elemento. El significado semántico de una etiqueta h1 es que se utiliza para identificar el encabezado más importante de una página web o sección específica.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Marcado de presentación

El marcado de presentación se encarga del aspecto del contenido, indica el formato que tendrán los textos.

El marcado de presentación se encarga de explicar las características visuales específicas del texto. Por ejemplo, *< i >* es la etiqueta que se usa para indicar que un texto debe escribirse en cursiva.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

4.4. HTML



Ilustración 1: lenguaje de marcado HTML. Fuente: dominio público.

HTML son las siglas de **HyperText Markup Language** y se traduce como lenguaje de marcado de hipertexto.

Es el componente más básico en los entornos web. Este lenguaje permite definir el significado y la estructura del contenido web.

'Hipertexto' hace referencia a enlaces que conectan páginas entre sí, es decir, una web que contiene enlaces a otras páginas de esa misma web o webs diferentes.

HTML no es un lenguaje de programación, es un **lenguaje de marcado**, la diferencia reside en que simplemente define la estructura de los contenidos, no permite realizar lógica de control ni condicional.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

HTML consiste en una serie de etiquetas especiales que el navegador web interpreta y que permiten indicar qué es cada contenido: párrafos, una lista de elementos, una tabla, una imagen, etc.

Las etiquetas HTML se distinguen por:

- ▶ abrir con el símbolo ‘<’
- ▶ cerrar con el símbolo ‘>’

Por un lado, están las etiquetas de apertura, por ejemplo, un párrafo: <p>

Por otro lado, están las etiquetas de cierre, por ejemplo: </p>

Todo lo que esté contenido entre ambas adoptará la estructura que indique el tipo de etiqueta.



The screenshot shows a code editor window titled 'definicion.html' and a preview window titled 'Preview definicion.html'. The code editor displays the following HTML structure:

```
<p> Esto es un párrafo </p>
<p> Esto es otro párrafo </p>
<a href="#">enlace</a>
```

The preview window shows the rendered content:

Esto es un párrafo
Esto es otro párrafo
[enlace](#)

Ilustración 2: etiquetas HTML.

Gracias al lenguaje de marcado HTML se puede definir la estructura de un documento web de manera que los navegadores puedan interpretarla y mostrarla visualmente en forma de una web.

El principal objetivo es que los usuarios de internet puedan navegar fácilmente de una manera visual a través de los contenidos.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

En 1989 Tim Berners Lee inventó la WWW en el CERN (Ginebra, Suiza). La www o World Wide Web es un sistema de distribución de documentos de hipertexto interconectados y accesibles desde internet.

Para definir la estructura de estos documentos Tim Berners Lee crea el lenguaje HTML cuya primera versión oficial HTML 1.0 aparece en 1993.

Desde entonces HTML ha ido evolucionando en diferentes versiones:

HTML 1.0	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.0	1998
XHTML	2000
HTML 5.0	2014
HTML 5.1	2016
HTML 5.2	2017

Ilustración 3: versiones de HTML.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Estándar web

HTML es un estándar de la WWW permite que cualquier página web esté escrita en una determinada versión y los navegadores web sean capaces de interpretarlas. El estándar es formulado por:

- ▶ Consorcio WWW o W3C desde 1994 dirigido por Tim Berners-Lee.



Ilustración 4: Logo consorcio WWW. Fuente: dominio público.

- ▶ Web Hypertext Application Technology Working Group o WHATWG desde 2004.



Ilustración 5: logo WHATWG. Fuente: dominio público.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Funcionamiento de la web

Cuando se utiliza un navegador web para acceder a un sitio web a través de una URL ocurre lo siguiente:

- ▶ Se traduce la URL a una dirección IP utilizando la base de datos distribuida de internet conocida como DNS.
- ▶ Se envía una petición HTTP a la dirección IP solicitando el recurso web.
- ▶ El servidor procesa la petición y devuelve un resultado, un documento HTML.
- ▶ El navegador renderiza ese documento HTML de forma visual interpretando el código HTML, CSS y JavaScript para producir el resultado final, una web.

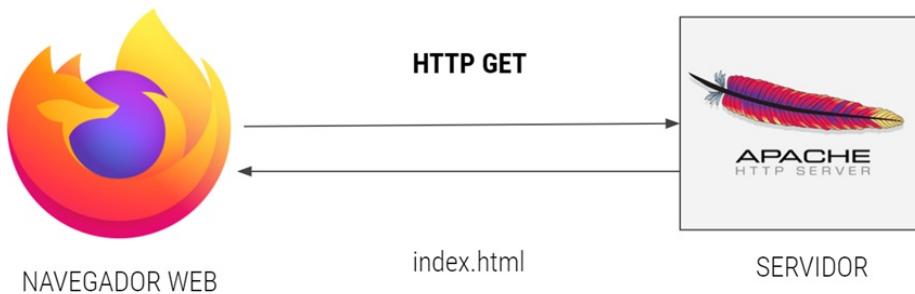


Ilustración 6: Ejemplo de petición HTTP GET.

HTML 5

HTML 5 incorpora nuevas funcionalidades y cambios con respecto a sus predecesores, algunas de las más importantes son:

- ▶ Simplificación en las declaraciones como:
 - Declaración Doctype html.
 - Declaración de codificación de caracteres.
 - Declaración de links a hojas css.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

- Declaración de scripts de javascript.
- ▶ Etiquetas y atributos eliminados.
- ▶ Nuevas etiquetas semánticas: <main>, <header>, <footer>, <nav>, <section>, <aside>, <article>.

Tecnologías web

HTML junto a las tecnologías CSS y JavaScript constituyen los pilares básicos en la creación de páginas web.

CSS o **Cascading Style Sheets** hace referencia a hojas de estilo en cascada y es un lenguaje de diseño gráfico que permite definir la presentación de contenido estructurado mediante un lenguaje de marcado, como por ejemplo HTML.

JavaScript es un lenguaje de programación multiparadigma que se emplea principalmente junto al código HTML y CSS para dotar a una web de comportamiento, es decir, para hacerla más dinámica al poder ejecutar lógica condicional, realizar operaciones, cargar datos desde el servidor, crear código HTML programáticamente, mostrar u ocultar elementos en base al comportamiento del usuario, etc.



Ilustración 7: tecnologías web HTML, JS y CSS. Fuente: dominio público.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Estructura HTML

Para crear un documento HTML utilizando la última versión HTML 5 se crea un archivo con extensión .html y con un entorno de desarrollo (IDE) se escribe el código HTML.

Existen multitud de entornos de desarrollo (IDEs) para programar en las tecnologías HTML, JavaScript y CSS:

- ▶ [IDEs de JetBrains](#)
- ▶ [Visual Studio Code](#)
- ▶ [Sublime Text](#)
- ▶ [Atom](#)
- ▶ [Brackets](#)

Para que un archivo HTML sea considerado un archivo HTML 5 correcto precisa una mínima estructura, que se compone de doctype, un head, y body.

Tema 4. Lenguajes de marcado. Información de sus etiquetas



```
ejemplo.html <html> <head> <title>Ejemplo título pestaña navegador</title> </head> <body> <h1>Ejemplo título header h1</h1> <p>Ejemplo de párrafo.</p> </body> </html>
```

Ejemplo título header h1

Ejemplo de párrafo.

Ilustración 8: ejemplo de documento HTML 5.

Existe multitud de etiquetas HTML, cada una para representar un tipo de contenido distinto.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Search 113 HTML elements					
	<input checked="" type="checkbox"/> experimental	<input checked="" type="checkbox"/> meta	<input checked="" type="checkbox"/> self-closing	<input checked="" type="checkbox"/> inline	<input checked="" type="checkbox"/> block
a			meta self-closing	inline	block
abbr			meta self-closing	inline	block
address			meta self-closing	inline	block
area			meta self-closing	inline	block
article			meta self-closing	inline	block
aside			meta self-closing	inline	block
audio			meta self-closing	inline	block
b			meta self-closing	inline	block

Ilustración 9: etiquetas HTML 5. Fuente: <https://htmlreference.io/>

Codificación

En informática toda la información se almacena en formato numérico. Para guardar caracteres (letras, números, signos de puntuación, diferentes lenguajes, símbolos, emojis, etc.) es necesario establecer una correspondencia entre los códigos numéricos y los caracteres. Al conjunto de correspondencias se le llama **juego de caracteres**.

Para que todos los ordenadores puedan entender los caracteres de la misma forma y las páginas web puedan ser correctamente interpretadas por igual se emplea un **juego de caracteres estándar**. Unicode es un conjunto de juegos de caracteres estándar utilizado en la actualidad y existen diferentes codificaciones para transformar los caracteres a binario: UTF-8, UTF-16, UTF-32...

Desde 2014 la recomendación HTML 5 especifica que **UTF-8** (Unicode Transformation Format) es la codificación predeterminada.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Los juegos de caracteres traducen caracteres a números. Las codificaciones traducen números a binario.

Para representar la codificación UTF-8 en las páginas HTML simplemente utilizamos la etiqueta **meta charset**.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
```

Ilustración 10: Codificación UTF-8.

Entidades carácter

Para poder mostrar caracteres especiales en HTML utilizamos las entidades de carácter, las cuales son referencias a caracteres.

```
ntidades-simbolos.html > html > body > p

<p>Mostramos el símbolo ampersand: &amp;</p>
<p>Mostramos el símbolo euro: &euro;</p>
<p>Mostramos el símbolo copyright: &copy;</p>
<p>Mostramos el símbolo marca registrada: &reg;</p>

<!-- Ejemplo simbolos matemáticos: -->

<p>Mostramos el símbolo matemático Para todos: &forall;</p>
<p>Mostramos el símbolo matemático Es un: &isin;</p>
<p>Mostramos el símbolo matemático No es un: &notin;</p>
<p>Mostramos el símbolo matemático Sumatorio: &sum;</p>

<!-- Acentos: en función de la configuración del alfabeto de los escritos directamente sobre html. Para asegurarnos de que se muestran correspondientes a las letras con los acentos que queremos mostramos -->

<p>Mostramos la letra a con acento: &aacute; &Aacute;;</p>
<p>Mostramos la letra e con acento: &eacute; &Eacute;;</p>
<p>Mostramos la letra i con acento: &iacute; &Iacute;;</p>
<p>Mostramos la letra o con acento: &oacute; &Oacute;;</p>
<p>Mostramos la letra u con acento: &uacute; &Uacute;;</p>
```

Ilustración 11: Entidades carácter.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Entidades numéricas

Para poder mostrar caracteres especiales en HTML también es posible utilizar entidades numéricas las cuales son referencias a caracteres equivalentes al número Unicode que referencia el carácter a mostrar.



Ilustración 12: Entidades numéricas.

Accesibilidad

La **accesibilidad** de una web o interfaz indica la capacidad de acceso a la misma y a sus contenidos por todas las personas, independientemente de sus conocimientos y de si presentan algún tipo de discapacidad (física, intelectual o técnica).

La accesibilidad web hace referencia al conjunto de prácticas destinadas a lograr que las páginas web sean utilizables por el máximo número de personas.

Algunas **prácticas** utilizadas para lograr la accesibilidad web son:

- ▶ Subtítulos en los videos.
- ▶ Imágenes o archivos multimedia que sustituyan o complementen textos.
- ▶ Tamaño fuente variable, para ajustarse a problemas de visión.
- ▶ Tamaño de los elementos en pantallas táctiles.
- ▶ Posibilidad de interacción por voz
- ▶ Posibilidad de reproducción del contenido textual mediante voz
- ▶ Ajustes en los colores

Tema 4. Lenguajes de marcado. Información de sus etiquetas



Ilustración 13: Imagen accesibilidad web. Fuente: dominio público.

Algunos recursos y proyectos dedicados a la mejora de la accesibilidad de las páginas web son:

- ▶ ARIA
- ▶ Web Content Accessibility Guidelines (WCAG)
- ▶ Web Accessibility in Mind (WebAIM)
- ▶ A11Y Style Guide

Usabilidad

La **usabilidad** web hace referencia al conjunto de técnicas que ayudan a los seres humanos a realizar tareas en entornos gráficos por ordenador.

Para que una interfaz gráfica se usable debe permitir realizar de forma sencilla e intuitiva cualquier acción a los usuarios de esta.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

La disposición de los elementos, la presentación de estos (aspecto visual), las secuencias de pasos a seguir para realizar cualquier acción son elementos cruciales para proporcionar una buena usabilidad web.

Una mayor usabilidad se traduce en una mejor experiencia de usuario y en consecuencia en navegaciones web satisfactorias, lo que fomenta que más gente utilice la web. Algunos principios que deben seguir las webs son:

- ▶ **Anticipación:** el sitio web debe anticiparse a las necesidades del usuario.
- ▶ **Autonomía:** la interfaz debe ser lo suficientemente intuitiva como para qué los usuarios tengan el control del sitio web sin necesidad de ayuda o indicaciones constantes.
- ▶ **Colores:** la paleta de colores utilizada debe estar en armonía y facilitar la visualización en lugar de complicarla o dispersar la atención.
- ▶ **Reversibilidad:** permitir deshacer acciones realizadas
- ▶ **Ley de Fitts:** el tiempo para alcanzar un objetivo con el ratón está en función de la distancia y el tamaño del objetivo, estos deben estar lo más cerca de la posición previa.

Validación HTML

Existen diversas herramientas online con las que verificar que el marcado HTML que hemos realizado es correcto y no tiene errores como, por ejemplo: etiquetas mal cerradas, etiquetas mal escritas, orden de etiquetas incorrecto, etc.

[W3C Markup Validation Service](#) es una de ellas.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

HTML semántico

HTML semántico hace referencia a utilizar los elementos HTML correctos para cada propósito específico.

Por ejemplo, a la hora de definir un botón, en vez de utilizar un elemento genérico como puede ser `<div>`, mejor utilizar una etiqueta semántica que ya esté creada para botones, como es el caso de `<button>`.

Elementos **genéricos**:

`<div>, `

Elementos **semánticos**:

`<button>, <form>, <table>, <h1>, <article>, <footer>, <nav>, <header>`

Los motores de búsqueda indexan mejor la información cuando utilizamos elementos semánticos, favoreciendo el SEO.

Otro aspecto semántico importante son los **atributos**. El atributo **alt** permite ofrecer un texto descriptivo para una imagen en caso de que la imagen no pueda ser cargada en el navegador por problemas de conexión lenta.

``

Tema 4. Lenguajes de marcado. Información de sus etiquetas

El atributo **title** sirve para añadir información entra a un elemento, de manera que cuando pasamos por encima con el ratón se muestra en forma de tooltip. Se emplea especialmente en enlaces:

Accede pulsando [aquí](/enlace "Enlace a Intranet")

Tema 4. Lenguajes de marcado. Información de sus etiquetas

4.5. XML



Ilustración 14: Logo lenguaje de marcado XML. Fuente: dominio público.

XML es el **lenguaje de marcado extensible**. Es un estándar aprobado por W3C para el marcado de documentos. Define una sintaxis genérica utilizada para marcar datos con etiquetas legibles por humanos. Proporciona un formato estándar para documentos informáticos que es lo suficientemente flexible como para personalizarse para dominios tan diversos como sitios web, intercambio de datos, gráficos vectoriales, genealogía, serialización de objetos, llamadas a procedimientos remotos, sistemas de correo de voz y más.

Es posible crear programas informáticos que lean y manipulen datos en documentos XML. Existe una amplia gama de librerías en una variedad de lenguajes de programación que pueden leer y escribir XML.

El software estándar, como navegadores web y editores de texto, tienen la capacidad de trabajar con documentos XML. Algunas de herramientas pueden trabajar con cualquier documento XML. Otras están personalizadas para admitir una aplicación XML particular en un dominio particular, como gráficos vectoriales, pero la misma sintaxis subyacente es utilizada en todos los casos.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

XML es un lenguaje de meta marcado para documentos de texto. Los datos están incluidos en XML en documentos como cadenas de texto. Los datos están rodeados por marcas de texto que describen los datos. La unidad básica de datos y marcado de XML se denomina **elemento**.

La especificación XML define la sintaxis exacta que debe seguir este marcado:

- ▶ Cómo los elementos están delimitados por etiquetas.
- ▶ Qué aspecto tiene una etiqueta.
- ▶ Qué nombres son aceptables para los elementos.
- ▶ Dónde se colocan los atributos.

Superficialmente, el marcado en un documento XML se parece mucho al marcado de un documento HTML, pero existen diferencias importantes.

Lo más importante es que XML es un lenguaje de meta marcado. Eso significa que no tiene conjunto fijo de etiquetas y elementos que se supone que funcionan para todos en todas las áreas de interés para todos los tiempos como ocurre en HTML.

XML permite inventar nuevos elementos según sean necesarios:

- ▶ Los químicos pueden usar elementos que describen moléculas, átomos, enlaces, reacciones y otros elementos encontrados en química.
- ▶ En el sector inmobiliario se pueden utilizar elementos que describen apartamentos, alquileres, comisiones, ubicaciones, y otros artículos necesarios para bienes raíces.
- ▶ Los músicos pueden usar elementos que describen negritas, medias notas, claves de sol, letras y otros objetos comunes en la música.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

La X en XML significa **Extensible**. Extensible significa que el idioma puede ser ampliado y adaptado para satisfacer muchas necesidades diferentes.

Aunque XML es bastante flexible en los elementos que permite, es bastante estricto en muchos otros aspectos. La especificación XML define una gramática para documentos XML que dice dónde se pueden colocar las etiquetas, cómo deben verse, qué elementos y nombres son legales, cómo se adjuntan los atributos a los elementos, etc.

Esta gramática es lo suficientemente específica como para permitir el desarrollo de analizadores XML que pueden leer cualquier documento XML. Los documentos que satisfacen esta gramática son documentos XML bien formados.

No se permiten documentos que no estén bien formados, como tampoco se permite un programa en Java que contiene un error de sintaxis. Los procesadores XML rechazan documentos que contienen errores de forma correcta.

Por razones de interoperabilidad, las personas u organizaciones pueden acordar utilizar solo ciertas etiquetas. Estos conjuntos de etiquetas se denominan aplicaciones XML. Una aplicación XML es no es una aplicación de software que utilice XML, como Mozilla o Microsoft Word. Es una aplicación de XML en un dominio particular, como gráficos vectoriales (SVG).

Marcado

El **marcado** en un documento XML describe la estructura del documento.

Permite ver qué elementos están asociados con qué otros elementos. En un documento XML bien diseñado, el marcado también describe la semántica del documento. Por ejemplo, el marcado puede indicar que un elemento es una fecha o una persona o un código de barras.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

En aplicaciones XML bien diseñadas, el marcado no dice nada sobre cómo debe mostrarse el documento. Es decir, no dice que un elemento esté en negrita o en cursiva o es un elemento de una lista. XML es un lenguaje de marcado estructural y semántico, no un lenguaje de marcado de tipo presentación.

El marcado permitido en una aplicación XML en particular se puede documentar en un **esquema**. Las instancias de documentos particulares se pueden comparar con el esquema. Se dice que los documentos que coinciden con el esquema son **válidos**.

Los documentos que no coinciden no son válidos. La validez depende del esquema. Es decir, si un documento es correcto o incorrecto depende del esquema con el que lo compare. No todos los documentos necesitan ser validados. Para muchos propósitos es suficiente que el documento esté bien formado.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

4.6. XHTML



Ilustración 15: Logo lenguaje de marcado XHTML. Fuente: dominio público.

XHTML (Extensible HyperText Markup Language), desarrollado por World Wide Consortium (W3C) en 2000, es la versión más extendida de HTML.

Es el mismo HTML, pero definido como una aplicación XML, que combina las ventajas de HTML y XML.

XHTML conserva todas las características de HTML, pero introduce reglas más estrictas para la creación de páginas, lo que le permite hacer que los sitios sean independientes del dispositivo de visualización y del navegador. Significa que el sitio se mostrará correctamente en todos los navegadores y plataformas modernos como ordenadores, tabletas, smartphones, etc.

Diferencias entre HTML y XHTML

- ▶ Siempre debe usar `<!DOCTYPE ...>`. Un documento XHTML debe incluir una declaración de tipo de documento XHTML. Otros elementos obligatorios son las etiquetas `<html>`, `<head>`, `<title>` y `<body>`.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

- ▶ Se deben insertar elementos XHTML correctamente. Un documento XHTML, no permite anidar elementos incorrectamente.
- ▶ A diferencia de HTML, donde a veces es posible omitir la etiqueta de cierre, en XHTML la etiqueta de cierre se requiere siempre y en todas partes.
- ▶ Las etiquetas en minúsculas y mayúsculas se distinguen en XHTML. Para evitar confusiones, todas las etiquetas, así como sus atributos, deben escribirse en minúsculas.
- ▶ Los valores de los atributos deben ir entre comillas.
- ▶ No se puede minimizar los atributos. Al atributo abreviado no se le asigna ningún valor. A un atributo se le puede asignar un valor de atributo que coincida con el nombre. Ejemplo:

Mal:

```
<input type="checkbox" name="flower" value="rose" checked />
```

Bien:

```
<input type="checkbox" name="flower" value="rose" checked="checked"  
      />
```

- ▶ En lugar del atributo de nombre, es mejor que use id. En XHTML, el nombre está parcialmente obsoleto, por eso se recomienda usar el atributo id.

Tema 4. Lenguajes de marcado. Información de sus etiquetas

Beneficios de XHTML

El hecho de que en XHTML todas las etiquetas deban cerrarse y anidarse correctamente hace que el código se vea mucho más limpio.

XHTML usa menos ancho de banda, lo que optimiza la carga del documento XHTML en los navegadores.

Gracias a su buen formato, los documentos XHTML se transportan fácilmente a dispositivos inalámbricos, lectores de Braille y otros entornos web especializados.

Al igual que con HTML normal, XHTML funciona con CSS para crear páginas web fácilmente.

XHTML vs. HTML

HTML es el lenguaje de marcado principal para crear páginas web, mientras que XHTML, es la versión extendida del mismo.

HTML es una aplicación del lenguaje de marcado estándar generalizado (SGML), mientras que XHTML es una aplicación de XML.

HTML se amplía desde SGML y XHTML desde XML y HTML. HTML fue creado por Tim Berners-Lee entre 1980 y 1989 y XHTML fue propuesto por World Wide Consortium en 2000.

La última diferencia entre estos dos es el marco flexible que requiere un analizador específico de HTML permisible en el caso de HTML y un subconjunto restrictivo de XML que debe analizarse con un analizador XML estándar en el caso de XHTML.