

## U.T.1 INTRODUCCIÓN A LOS LENGUAJES DE MARCAS

Pretende dar una visión global de los lenguajes de marcas, que el alumno conozca la historia y necesidad de los estándares web, conozca la labor del diseñador web, y valore la utilización de métodos estructurados.

### Contenido

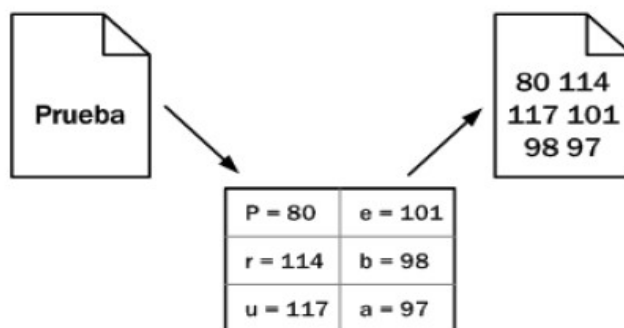
1	Introducción al lenguaje de marcas.....	2
2	Ventajas que proporcionan los lenguajes de marcas en el tratamiento de la información.....	4
3	Distintas clasificaciones de los lenguajes de marcado.....	5
4	Haciendo un poco de historia.....	5
4.1	SGML.....	6
4.2	Origen de HTML. Ventajas e inconvenientes. Ejemplo de HTML.....	6
	Origen del HTML.....	7
	Ventajas.....	9
	Inconvenientes.....	10
	Ejemplo de HTML.....	10
4.3	Se requiere algo nuevo...XML.....	10
	Se requiere algo nuevo.....	10
	El XML no es.....	11
	El XML sirve para.....	11
	Un documento XML representa una jerarquía de datos. Ejemplo:.....	12
	Ejemplo de XML.....	12
4.4	La X en XHTML.....	12
	Ejemplo de XHTML.....	13

4.5	CSS.....	13
5	Estándares web. ¿Por qué utilizar estándares web?.....	15
	¿Por qué utilizar estándares web?.....	15
6	No somos WYSIWYG. ¿Cuál es tu editor de texto sin formato favorito?.....	16
7	¿Windows o Linux?.....	16
	Bibliografía.....	17

## 1 Introducción al lenguaje de marcas

Uno de los retos iniciales a los que se tuvo que enfrentar la informática fue el de cómo almacenar la información en los archivos digitales. Como los primeros archivos sólo contenían texto sin formato, la solución utilizada era muy sencilla: se codificaban las letras del alfabeto y se transformaban en números.

De esta forma, para almacenar un contenido de texto en un archivo electrónico, se utiliza una tabla de conversión que transforma cada carácter en un número. Una vez almacenada la secuencia de números, el contenido del archivo se puede recuperar realizando el proceso inverso.



El proceso de transformación de caracteres en secuencias de números se denomina **codificación de caracteres** y cada una de las tablas que se han definido para realizar la transformación se conoce con el nombre de **páginas de código**. Una de las codificaciones más conocidas (y una de las primeras que se publicaron) es la codificación **ASCII**.

Una vez resuelto el problema de almacenar el texto simple, se presenta el reto de almacenar los contenidos de texto con formato. En otras palabras, ¿cómo se almacena un texto en negrita? ¿Y un texto de color rojo? ¿Y otro texto azul, en negrita y subrayado?

La solución que se emplea para guardar la información con formato consiste en que el archivo electrónico, almacena tanto los contenidos como la información sobre el formato de esos contenidos.

Si por ejemplo se quiere dividir el texto en párrafos y se desea dar especial importancia a algunas palabras, se podría indicar de la siguiente manera:

**<párrafo>**

Contenido de texto con **<importante>**algunas palabras**</importante>** resaltadas de forma especial.

**</párrafo>**

El principio de un párrafo se indica mediante la palabra **<párrafo>** y el final de un párrafo se indica mediante la palabra **</párrafo>**. De la misma manera, para asignar más importancia a ciertas palabras del texto, se encierran entre **<importante>** y **</importante>**.

El proceso de indicar las diferentes partes que componen la información se denomina **marcar** (*markup* en inglés). Cada una de las palabras que se emplean para marcar el inicio y el final de una sección se denominan **etiquetas**.

Según la Wikipedia ...

"Un lenguaje de marcado o lenguaje de marcas es una **forma de codificar un documento** que, junto con el texto, **incorpora etiquetas o marcas** que contienen **información adicional** acerca de la estructura del texto o su presentación"

Hay que tener en cuenta que los lenguajes de marcas...

No tienen **nada que ver** con los lenguajes de **programación**.

Tienen una **función** principal meramente **descriptiva**.

**No son nuevos**, ya que los primeros datan de finales de los 60.

En la última década han comenzado a tener **aplicación en múltiples facetas** de la Informática.

- Las **marcas** o etiquetas más empleadas suelen consistir en textos descriptivos entre signos de "menor" (<) y "mayor" (>)
- Lo habitual es usar **pares de marcas**: una de **principio** y otra de **fin**.
- El **almacenamiento** suele realizarse en **ficheros de texto plano**.

Los **archivos de texto plano** son aquellos que están compuestos únicamente por texto sin formato, sólo caracteres. Ejemplos: Vi de UNIX, bloc de notas (notepad) de Windows, VIM de linux.

Ejemplo de un documento generado con un lenguaje de marcas:

```
<ficha>
<nombre>Pepe </nombre>
<apellido>Pérez</apellido>
<apellido>Rodriguez</apellido>
<datos_profesionales>
<cuerpo>grupo A</cuerpo>
<especialidad>informática</especialidad>
<destino>Ayuntamiento de Burgos</destino>
</datos_profesionales>
</ficha>
```

## 2 Ventajas que proporcionan los lenguajes de marcas en el tratamiento de la información

Los lenguajes de marcas almacenan sus datos y programas en formato de texto, lo que trae consigo ventajas tales como:

- Una mayor portabilidad de los programas para ser implementados en cualquier arquitectura de hardware o software.
- El uso y reuso amplio de los programas ya que son de código abierto.

- Facilidad de mantenimiento debido a que los archivos son más entendibles por humanos, y además porque los datos y programas pueden ser creados y modificados por cualquier editor de textos.

### 3 Distintas clasificaciones de los lenguajes de marcado

Una posible [clasificación](#) de los lenguajes de marcas puede ser la siguiente:

- [Procedimentales](#): describen operaciones tipográficas. Ej.: HTML

LENGUAJES PROCEDIMENTALES expresan detalladamente las instrucciones que es necesario seguir para realizar un trabajo.
---

- [Estructurales](#): describen la estructura lógica de un documento, pero no su tipografía. Ej.: MusicXML
- [Híbridos](#): que forman una combinación de los dos tipos anteriores. Ej.: VRML (Virtual Reality Modelling Language)

También existen las [hojas de estilo](#) o los [lenguajes de transformación](#), que permiten la “[traducción](#)” de anotaciones [de tipo estructural](#) a anotaciones de carácter [tipográfico](#).

Otra posible [clasificación](#) de los lenguajes de marcas puede ser la siguiente:

- [De presentación](#): indica el formato del texto (información para el maquetado). Ej.: RTF (Rich Text Format)
- [De procedimientos](#): orientado también a la presentación, pero, en este caso, se indican los procedimientos que deberá realizar el software de representación. Ej.: TeX / LaTeX
- [Descriptivo o semántico](#): describen las diferentes partes en las que se estructura el documento pero sin especificar cómo deben representarse. Ej.: XML (eXtensible Markup Language)

### 4 Haciendo un poco de historia.

- En la década de los 60 se empieza a desarrollar la idea de separar presentación y estructura.

- Desde IBM se impulsa la creación del lenguaje **GML** (**G**eneralized **M**arkup **L**anguage - Goldfarb, Mosher & Lorie, 1969)
- En 1978, ANSI creó un comité de Lenguajes de Ordenador para Procesador de Texto (Computer Languages for the Processing of Text)
- En 1980, a partir de GML, se genera la primera versión de **SGML** (**S**tandard **G**eneralized **M**arkup **L**anguage)
- En 1986 se convierte en la norma **ISO 8879**
- **SGML** pasa a ser la base del diseño de nuevos lenguajes
- En 1992 se desarrolla **HTML** (**H**yper**T**ext **M**arkup **L**anguage) como solución para publicar las investigaciones de diversas fuentes y autores que trabajaban en el **CERN**
- En 1987, **Microsoft** desarrolla **RTF** (**R**ich **T**ext **F**ormat) como formato de intercambio de documentos multiplataforma
- En 1998, la **W3C** desarrolla **XML** (**eX**tensible **M**arkup **L**anguage), que será el lenguaje de base de una gran cantidad de lenguajes de marcas. Eclosión de los lenguajes de marcado.
- **RDF** (**R**esource **D**escription **F**ramework) y **OWL** (**O**ntology **W**eb **L**anguage) marcan el desarrollo de la **Web Semántica**.

### 4.1 SGML.

Es el estándar internacional para la definición de la estructura y el contenido

### 4.2 Origen de HTML. Ventajas e inconvenientes. Ejemplo de HTML.

Definiéndolo de forma sencilla, "HTML es lo que se utiliza para crear todas las páginas web de Internet". Más concretamente, HTML es el lenguaje con el que se "escriben" la mayoría de páginas web.

Los diseñadores utilizan el lenguaje HTML para crear sus páginas web, los programas que utilizan los diseñadores generan páginas escritas en HTML y los navegadores que utilizamos los usuarios muestran las páginas web después de leer su contenido HTML.

Aunque HTML es un lenguaje que utilizan los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas. En realidad, HTML son las siglas de HyperText Markup Language y más adelante se verá el significado de cada una de estas palabras.

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium (<http://www.w3.org/>) , más conocido como **W3C**. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

El propio **W3C** define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica.

### *Origen del HTML*

La historia completa de HTML es tan interesante como larga, por lo que a continuación se muestra su historia resumida a partir de la información que se puede encontrar en la Wikipedia.

El origen de HTML se remonta a 1980, cuando el físico **Tim Berners-Lee**, trabajador del CERN (<http://www.cern.ch/>) (*Organización Europea para la Investigación Nuclear*) propuso un nuevo sistema de "hipertexto" para compartir documentos.

Los sistemas de "hipertexto" habían sido desarrollados años antes. En el ámbito de la informática, el "hipertexto" permitía que los usuarios accedieran a la información relacionada con los documentos electrónicos que estaban visualizando. De cierta manera, los primitivos sistemas de "hipertexto" podrían asimilarse a los enlaces de las páginas web actuales.

Tras finalizar el desarrollo de su sistema de "*hipertexto*", Tim Berners-Lee lo presentó a una convocatoria organizada para desarrollar un sistema de "*hipertexto*" para Internet. Después de unir sus fuerzas con el ingeniero de sistemas **Robert Cailliau**, presentaron la propuesta ganadora llamada *WorldWideWeb (W3)*.

El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre "*HTML Tags*"

([http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/](http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html)

[Tags.html](http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html)) (*Etiquetas HTML*) y todavía hoy puede ser consultado online a modo de *reliquia informática*.

La primera propuesta oficial para convertir HTML en un estándar se realizó en 1993 por parte del organismo IETF (<http://www.ietf.org/>) (*Internet Engineering Task Force*). Aunque se consiguieron avances significativos (en esta época se definieron las etiquetas para imágenes, tablas y formularios) ninguna de las dos propuestas de estándar, llamadas HTML y HTML+ consiguieron convertirse en estándar oficial.

En 1995, el organismo IETF organiza un grupo de trabajo de HTML y consigue publicar, el 22 de septiembre de ese mismo año, el estándar HTML 2.0. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML.

A partir de 1996, los estándares de HTML los publica otro organismo de estandarización llamado W3C (<http://www.w3.org/>) (*World Wide Web Consortium*). La versión HTML 3.2 se

publicó el 14 de Enero de 1997 y es la primera recomendación de HTML publicada por el W3C.

Esta revisión incorpora los últimos avances de las páginas web desarrolladas hasta 1996, como *applets* de Java y texto que fluye alrededor de las imágenes.

HTML 4.0 se publicó el 24 de Abril de 1998 (siendo una versión corregida de la publicación original del 18 de Diciembre de 1997) y supone un gran salto desde las versiones anteriores.



Entre sus novedades más destacadas se encuentran las hojas de estilos CSS, la posibilidad de incluir pequeños programas o *scripts* en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas y mejoras en los formularios.

La última especificación oficial de HTML se publicó el 24 de diciembre de 1999 y se denomina

HTML 4.01. Se trata de una revisión y actualización de la versión HTML 4.0, por lo que no incluye novedades significativas.

Desde la publicación de HTML 4.01, la actividad de estandarización de HTML se detuvo y el W3C se centró en el desarrollo del estándar XHTML. Por este motivo, en el año 2004, las empresas Apple, Mozilla y Opera mostraron su preocupación por la falta de interés del W3C en HTML y decidieron organizarse en una nueva asociación llamada WHATWG (<http://www.whatwg.org/>) (*Web Hypertext Application Technology Working Group*).

La actividad actual del WHATWG se centra en el futuro estándar HTML 5, cuyo primer borrador oficial (<http://www.w3.org/TR/html5/>) se publicó el 22 de enero de 2008. Debido a la fuerza de las empresas que forman el grupo WHATWG y a la publicación de los borradores de HTML 5.0, en marzo de 2007 el W3C decidió retomar la actividad estandarizadora de HTML (<http://www.w3.org/2007/03/html-pressrelease>).

De forma paralela a su actividad con HTML, W3C ha continuado con la estandarización de XHTML, una versión *avanzada* de HTML y basada en XML. La primera versión de XHTML se denomina XHTML 1.0 y se publicó el 26 de Enero de 2000 (y posteriormente se revisó el 1 de Agosto de 2002).

### **Ventajas**

El hipertexto funciona (es navegable)

Es multiplataforma

Tiene una curva de aprendizaje muy plana

Barato (muchos editores, visores, verificadores, etc., gratuitos y comerciales).

Base de información grande

Los navegadores son baratos, sencillos de construir y de usar y poderosos

### *Inconvenientes*

- Pobre herramienta de presentación.
  - » Poco control de espaciado.
  - » Problemas con el control de guiones, justificación y otras manipulaciones de texto.
  - » EL uso de columnas es problemático.
- Pobre herramienta de marcaje (*markup*).
  - » No se pueden agregar etiquetas nuevas.
  - » No es modular, poca oportunidad de reciclar.
  - » Hay demasiado código inválido publicado actualmente.
- Es una representación de un documento, no de información.
- Es campo de batalla comercial (Mozilla vs IE)

### *Ejemplo de HTML*

```
<html>
  <head>
    <title>
      Brazil
    </title>
  </head>
  <body>
    <h1>Brazil</h1>
    <b>Director:</b> terry Gilliam <br>
    <b>Protagonista:</b> Jonathan Pryce <br>

  </body>
</html>
```

### **4.3 Se requiere algo nuevo...XML.**

XML.-Lenguaje extensible de marcas (eXtensible Markup Language)

*Se requiere algo nuevo...*

- Barato, veloz y sencillo:
  - » Para crear documentos.
  - » Para procesar documentos.
  - » Para presentar documentos.
- Extensible:
  - » Un conjunto de reglas, no un conjunto de etiquetas.
- Compatible con el HTML:
  - » Debe tener una manera sencilla de convertir en y desde HTML.
- Capaz de representar información en cualquier forma, no solo documentos.

En febrero de 1998, nace el XML como un borrador para una recomendación del W3C

*El XML no es...*

- Un lenguaje de marcas.
  - » **No.** Es un estándar que especifica una sintaxis para crear lenguajes de marcaje.
- Solo para Web.
  - » **No.** Puede ser usado para describir y comunicar cualquier información estructurada.
- Un superconjunto del HTML.
  - » **No.** Aunque el HTML puede ser definido con sintaxis de XML.
- Un invento de [x compañía].
  - » **No.** XML es un estándar creado por el W3C y soportado por compañías e instituciones de todo el mundo.

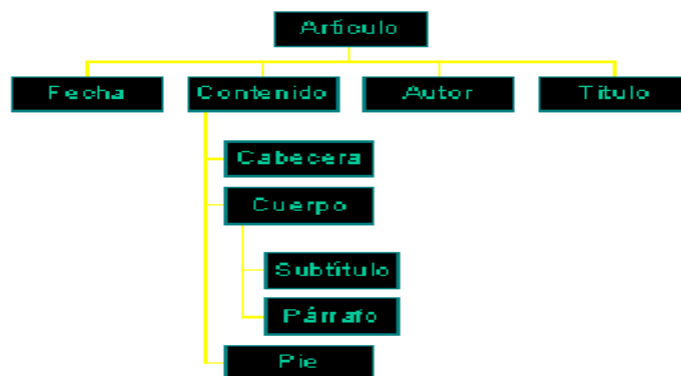
*El XML sirve para...*

- Hacer publicación electrónica independiente del medio.

- Crear protocolos para el intercambio de datos entre miembros de una industria.
- Facilitar el procesamiento de datos usando software barato.
- Permite a las personas visualizar la información de la manera que quieran.

Proporcionar metadatos que mejoran la calidad de la búsqueda de información

*Un documento XML representa una jerarquía de datos.  
Ejemplo:*



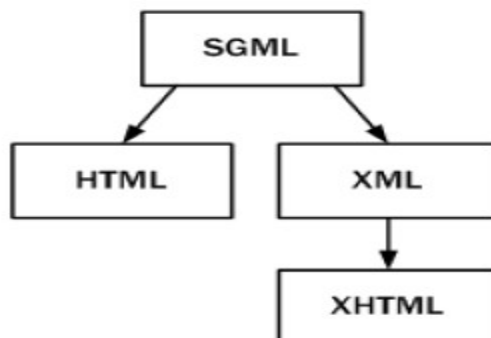
### *Ejemplo de XML*

```
<?xml version="1.0"?>
<pelicula>
  Brazil
<director> Terry Gilliam </director>
<actores>
  <actor>Jonathan Pryce</actor>
</actores>
</pelicula>
```

### **4.4 La X en XHTML.**

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Se han incluido todas las etiquetas HTML, pero siguiendo las directrices de XML. Técnicamente, HTML es

descendiente directo del lenguaje SGML, mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML).



Las páginas y documentos creados con XHTML son muy similares a las páginas y documentos HTML. Las discusiones sobre si HTML es mejor que XHTML o viceversa son recurrentes en el ámbito de la creación de contenidos web, aunque no existe una conclusión ampliamente aceptada.

Actualmente, entre HTML 4.01 y XHTML 1.0, la mayoría de diseñadores escogen XHTML. En un futuro cercano, quizás la elección sea diferente.

### *Ejemplo de XHTML*

```
<html>
  <head>
    <title>
      Brazil
    </title>
  </head>
  <body>
    <h1>Brazil</h1>
    <b>Director:</b> terry Gilliam <br/>
    <b>Protagonista:</b> Jonathan Pryce <br/>
  </body>
</html>
```

## 4.5 CSS.

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

## Lenguajes de Marcas y Sistemas de Gestión de la Información

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.



<pre> &lt;?xml version="1.0"?&gt; &lt;?xml-stylesheet type="text/css" href="peli.css"?&gt; &lt;pelicula&gt;     Brazil &lt;director&gt; Terry Gilliam &lt;/director&gt;  &lt;/pelicula&gt;         </pre>	<pre> pelicula {     display: block;     font-size:xx-large;     font-weight:bold;     margin-top:8px;     margin-left:8px; }  director {     display: block;     font-size:medium;     font-weight:normal;     top:21px;     position:relative; }  director before {     content:"Director ";     font-size:large;     font-weight:bold;}         </pre>
<b>XML</b>	<b>CSS</b>

# Brazil

Terry Gilliam

## 5 Estándares web. ¿Por qué utilizar estándares web?

El organismo W3C (<http://www.w3.org/>) (*World Wide Web Consortium*) elabora las normas que deben seguir los diseñadores de páginas web para crear las páginas HTML. Las normas oficiales están escritas en inglés y se pueden consultar de forma gratuita en las siguientes direcciones:

- Especificación oficial de HTML 4.01 (<http://www.w3.org/TR/html401/>)
- Especificación oficial de XHTML 1.0 (<http://www.w3.org/TR/xhtml1/>)

El estándar XHTML 1.0 incluye el 95% del estándar HTML 4.01, ya que sólo añade pequeñas mejoras y modificaciones menores. Afortunadamente, no es necesario leer las especificaciones y recomendaciones oficiales de HTML para aprender a diseñar páginas con HTML o XHTML. Las normas oficiales están escritas con un lenguaje bastante formal y algunas secciones son difíciles de comprender.

### *¿Por qué utilizar estándares web?*

Con la utilización de los estándares Web se acorta el tiempo de desarrollo, los sitios creados son fáciles de actualizar y mantener, se mejoran los rankings de los motores de búsqueda y, además, se mejora la accesibilidad, haciendo las páginas más accesibles a más gente, especialmente a los discapacitados, algunos de los cuales no pueden utilizar navegadores estándar.

El enfoque de los estandaristas Web se basa en separar contenido, estructura y significado de la información de su presentación visual y su diseño. Utilizar la herramienta adecuada para cada trabajo.

Separando contenido (las palabras e imágenes) y presentación (como estilar esas palabras e imágenes), reducimos considerablemente los tiempos de descarga de la página.

Separar contenido de presentación hace también el mantenimiento más sencillo. ¿Está cansado del diseño que creó en 2009? Cambie el diseño de todo el sitio web por otro nuevo y mejorado, actualizando simplemente la hoja de estilo.

Separar contenido de representación, facilita la creación de hojas de estilo personalizadas para PDA,s , teléfonos móviles y otros dispositivos distintos de los ordenadores.

## 6 No somos WYSIWYG. ¿Cuál es tu editor de texto sin formato favorito?

WYSIWYG es el [acrónimo](#) de *What You See Is What You Get* (en inglés, "lo que ves es lo que obtienes") Ejemplos de editores HTML tipo WYSIWYG son: [Dreamweaver](#), [NVU](#)/Kompozer, las versiones de [Composer](#) de [Netscape](#) y [Mozilla](#), [Amaya](#), [Writer](#) (de [OpenOffice.org](#)), Adobe Golive, Frontpage y Microsoft Word. También existen editores que se pueden integrar en formularios de páginas web como [FCKeditor](#), [TinyMCE](#), [FreeRichTextEditor](#).

En este curso no se va incentivar la utilización de software WYSIWYG. Nosotros vamos a escribir etiquetado de texto. ¿Por qué no utilizar el software WYSIWYG? Suena ideal. Arrastrar y soltar en un entorno visual, hacer clic en un botón y el software se ocupa del resto, escribiendo por ti el etiquetado. Suena perfecto. Pero no lo es. El problema de utilizar este tipo de software es que no tenemos un total conocimiento de lo que ocurre por detrás, porque el software lo hace todo por nosotros.

Este curso va dirigido a todo aquel que tenga acceso a un ordenador (que ni siquiera tiene que ser muy potente). Vamos a utilizar herramientas gratuitas o de bajo coste siempre que sea posible, dando prioridad al software libre. Es suficiente con un editor de textos sin formato.



## 7 ¿Windows o Linux?

### Actividades de enseñanza-aprendizaje

Establecer el entorno de trabajo.

Buscar, seleccionar, descargar e instalar un editor de texto.

Exploración del editor elegido.

### Criterios de evaluación

Se han Identificado las características generales de los lenguajes de marcas.

Se han identificado las ventajas que proporcionan en el tratamiento de la información.

Se han clasificado los lenguajes de marcas e identificado los más relevantes.

Se han diferenciado sus ámbitos de aplicación.

Diferenciación de la necesidad y los ámbitos específicos de aplicación de un lenguaje de marcas de propósito general.

## Bibliografía

Xml.- Juan Diego Gutiérrez Gallardo.-Anaya

HTML y CSS- Christopher Murphy y Nicklas persson