

## UT1.- LENGUAJES DE MARCAS. CARACTERÍSTICAS

RESULTADOS DE APRENDIZAJE ASOCIADOS
1. Reconoce las características de lenguajes de marcas analizando e interpretando fragmentos de código.
CRITERIOS DE EVALUACIÓN
<ul style="list-style-type: none"><li>a) Se han identificado las características generales de los lenguajes de marcas.</li><li>b) Se han reconocido las ventajas que proporcionan en el tratamiento de la información.</li><li>c) Se han clasificado los lenguajes de marcas e identificado los más relevantes.</li><li>d) Se han diferenciado sus ámbitos de aplicación.</li><li>e) Se ha reconocido la necesidad y los ámbitos específicos de aplicación de un lenguaje de marcas de propósito general.</li><li>f) Se han analizado las características propias del lenguaje XML.</li><li>g) Se ha identificado la estructura de un documento XML y sus reglas sintácticas.</li><li>h) Se ha contrastado la necesidad de crear documentos XML bien formados y la influencia en su procesamiento.</li><li>i) Se han identificado las ventajas que aportan los espacios de nombres.</li><li>j) Se conocen los mecanismos de codificación XML propios de cada idioma.</li><li>k) Se conocen los fundamentos básicos de programación.</li></ul>

## UT1.- LENGUAJES DE MARCAS. CARACTERÍSTICAS

### Índice de contenido

<b>1.- Introducción .....</b>	<b>3</b>
<b>1.1.- Historia y orígenes de los lenguajes de marcas.....</b>	<b>4</b>
<b>1.2.- Organizaciones desarrolladoras.....</b>	<b>5</b>
<b>2.- ¿Qué es entonces un lenguaje de marcas? .....</b>	<b>6</b>
<b>3.- Clasificación y características comunes de los lenguajes de marcas .....</b>	<b>6</b>
<b>3.1.1.- Clasificación de los lenguajes de marcas.....</b>	<b>7</b>
<b>4.- XML: estructura y sintaxis .....</b>	<b>8</b>
<b>4.1.- ¿Para qué se emplea XML? .....</b>	<b>8</b>
<b>4.2.- Estructura y sintaxis de XML.....</b>	<b>9</b>
<b>4.3.- Elaboración de documentos XML bien formados .....</b>	<b>10</b>
<b>4.5.- Procesador XML (XML processor) y aplicación (application).....</b>	<b>17</b>
<b>4.6.- Documentos bien formados y válidos.....</b>	<b>17</b>
<b>4.7.- Entidades predefinidas.....</b>	<b>18</b>
<b>4.8.- DTD y DOCTYPE .....</b>	<b>19</b>
<b>4.9.- Herramientas de edición .....</b>	<b>20</b>
<b>4.9.1.- Bloc de notas de Windows.....</b>	<b>20</b>
<b>4.9.2.- Gedit.....</b>	<b>20</b>
<b>4.9.3.- Notepad++.....</b>	<b>20</b>
<b>5.- Glosario de términos .....</b>	<b>22</b>

## 1.- Introducción

El uso de internet por parte de los usuarios está creciendo en los últimos años. La gran mayoría de los internautas navegan e intercambian información a través de la red, manejando las páginas web que internamente **hacen uso de los lenguajes de marcas** → proceso que es transparente para los usuarios.

Algunos de los programas instalados en el ordenador personal usan **lenguajes de marcas**, ya sea para almacenar información, configuración de la aplicación, para dar algún tipo de formato, etc.

Las marcas pueden representarse con distintos caracteres. Las más usadas son las que se expresan entre el signo menor que (<) y mayor que (>).

Ejemplos:

- ❖ Etiqueta de marca XML <nombre> con el valor “Litchi”:

```
<nombre>Litchi</nombre>
```

- ❖ Etiqueta de marca HTML de cabecera <h1> con el valor “Hola Mundo”:

```
<h1>Hola Mundo</h1>
```



No puede compararse un lenguaje de marcas con un lenguaje de programación como C#, Java o Pascal, pero sí permite combinar dentro de un documento lenguajes como PHP o Javascript junto con HTML, aportando funcionalidad.

## 1.1.- Historia y orígenes de los lenguajes de marcas

El origen de los lenguajes radica en el lenguaje de marcado generalizado estándar → **SGML** (*Standard Generalized Markup Language*).

Es un estándar que permite definir lenguajes de marcado, considerándose un metalenguaje.

Es descendiente de GML (*Generalized Markup Language*) de IBM.

En el año 1986, el GML se convirtió en SGML, lo que dio lugar a la ISO 8879.

A principios de los años noventa, Tim Berners-Lee desarrolla HTML, que permite visualizar la información de las páginas web en los distintos navegadores.

En 1998. El consorcio **W3C** (*World Wide Web Consortium*) encargado de crear recomendaciones y estándares que aseguren el crecimiento de WWW, crea **XML** (*eXtended Markup Language*). Más sencillo que SGML y potente que HTML

Una herramienta fundamental que se usa para visualizar documentos XML, HTML, XHTML, etc., es el **navegador web** o **browser** (en inglés) → Son aplicaciones instaladas en distintos dispositivos cuya funcionalidad es dar acceso a la web, mostrando a los usuarios la interpretación de los lenguajes de marcas.



**RECUERDA** que... un navegador NO es lo mismo que un buscador.

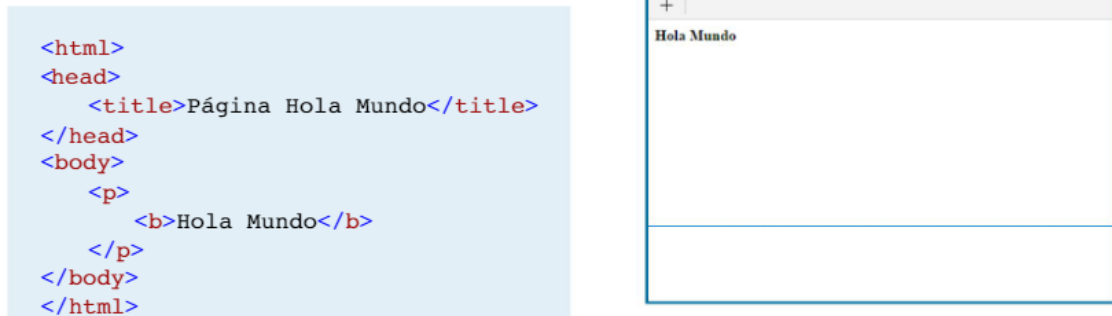
**Investiga:**



- ¿Cuántos navegadores tienes instalados en tu ordenador?
- ¿Cuántos navegadores conoces?
- Busca al menos 5 navegadores que no tengas en el ordenador.

## → EJEMPLO de estructura de un documento HTML

Dicho documento puede crearse con el Bloc de notas y ha de tener la extensión **.html**:



El resultado de la ejecución es la imagen de la derecha, donde se aprecia que **ninguna de las marcas** que se han creado en el documento **se muestra en el navegador**.

## 1.2.- Organizaciones desarrolladoras

### ❖ ISO (Organización Internacional para la Estandarización)

Se crea en 1947 con el objetivo de estandarizar normas de productos y seguridad en empresas en las ramas industriales excepto la eléctrica y electrónica.

- 163 países. Secretaría Central en Ginebra (Suiza)
- 1986 → publica el *Standard Generalized Markup Language (SGML)*, con código ISO 8879

### ❖ W3C (World Wide Web Consortium)

- Tutela el crecimiento y organización de la WEB. Creado en 1994 por Tim Bernes-Lee. Su sede está en MIT (*Massachusetts Institute of Tecnology*)
- Normaliza el HTML (lenguaje de marcas para crear páginas web)

- Crea XML como reglas para poder crear nuevos lenguajes de marcas con estructura y sintaxis común (metalenguaje) y usar las mismas herramientas

## 2.- ¿Qué es entonces un lenguaje de marcas?

Las páginas web ofrecen la posibilidad de tener, además de texto, columnas, colores, subrayados, negritas, imágenes, enlaces, tablas, listados, etc.

Todas las páginas web están constituidas por **texto plano**, o sea, texto sin más.



**Un Texto plano (*plain text*), son aquellos archivos formados exclusivamente por texto (sólo caracteres), sin ningún formato.**

En otras palabras, no hay información sobre el tipo de letra, ni formas (negrita, subrayados...), ni tamaños, ni columnas, ni enlaces, ni imágenes, etc.

¿Cómo incluimos todos estos elementos en una página sabiendo que el resultado debe seguir siendo texto plano? → Mediante el **lenguaje de marcas**

## 3.- Clasificación y características comunes de los lenguajes de marcas

En este apartado, se verá una clasificación de los **lenguajes de marcas** atendiendo a tres tipos distintos y mostrando ejemplos de los más usados, así como describiendo las características comunes que poseen.

### 3.1.1.- Clasificación de los lenguajes de marcas

Los lenguajes de marcas suelen clasificarse en tres tipos:

- **Tipo 1. De procedimiento** → También enfocado hacia la presentación del texto, sin embargo, las marcas que formatean el texto son visibles para el usuario y permiten procesamiento (realizar un conjunto de acciones) según el tipo de etiqueta. Ej: Latex, Postscript, HTML
- **Tipo 2. De presentación** → Indican como ha de presentarse el formato del texto, las etiquetas de marcado están ocultas. Ej: Microsoft Word
- **Tipo 3. Descriptivos o semánticos** → Es un marcado flexible que usa etiquetas sin especificar la manera de representarlas ni el orden. Las marcas dan información sobre su estructura y una descripción del contenido. Ejemplos: SGML, XHTML y XML.

Mira este ejemplo:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<exámenes>
  <materia>
    <nombre>Lenguaje de marcas</nombre>
    <aprobados>15</aprobados>
    <suspensos>10</suspensos>
    <unidades>4</unidades>
  </materia>
  <materia>
    <nombre>sistemas operativos</nombre>
    <aprobados>5</aprobados>
    <suspensos>20</suspensos>
    <unidades>5</unidades>
  </materia>
</exámenes>
```

## ACTIVIDAD 1.1

**Viendo el ejemplo anterior y analizando su estructura, ¿cómo añadirías otra materia al documento XML, teniendo en cuenta que las dos materias existentes son lenguaje de marcas y sistemas operativos?**

### 4.- XML: estructura y sintaxis

**XML**, siglas de *extensible markup language* (lenguaje de marcas extensibles).

- Se considera un metalenguaje, descendiente de SGML.
- Surge para resolver los problemas que plantea HTML → Permite describir la **semántica de los datos**, pero no la presenta.

¿Qué NO es XML?

- No es una versión mejorada de HTML
- No es un lenguaje sustituto de HTML
- No es un lenguaje para hacer mejores páginas web

#### 4.1.- ¿Para qué se emplea XML?

- En general, se puede decir que el XML ha tenido un gran éxito y se ha convertido en una herramienta básica para **el intercambio y almacenamiento de información** en muchos sectores de la informática
- **Convertor.** Actualmente son muchos los formatos que ofrecen servicios de conversión a XML, como PDF, HTML, .text, .docx o XHTML
- **Base de datos.** XML permite guardar datos de forma estandarizada para luego poder ser tratados por multitud de lenguajes diferentes. Su manejo es mucho más sencillo que bases de datos como MySQL y mucho más rico que utilizar ficheros de texto planos.



Ejemplos:

Área	Lenguaje	Descripción/Ejemplo
Matemática	MathML y OpenMath	Su objetivo es expresar notación matemática
Aeronáutica	Spacecraft ML	
Multimedia	Synchronized Multimedia Integration Language	El lenguaje SMIL permite integrar audio, video, imágenes, texto o cualquier otro contenido multimedia
Voz	VoiceXML	
Mensajería instantánea	XMPP	Queda establecida una plataforma para el intercambio de datos XML que puede ser usada en aplicaciones de mensajería instantánea
Gráficos 3D	VRML/X3D, STEP	Tiene como objetivo la representación de escenas u objetos interactivos tridimensionales

## 4.2.- Estructura y sintaxis de XML

Todo documento XML tiene estructura jerárquica arborescente y está compuesto por dos partes fundamentales:

- **Prólogo** → primera parte del documento. Compuesto de una o varias líneas
- **Cuerpo**

<pre>&lt;?xml version="1.0" encoding="iso-8859-1"?&gt;</pre>	<b>PRÓLOGO</b>
<pre>&lt;cine&gt;   &lt;película&gt;     &lt;director&gt;Señor X&lt;/director&gt;     &lt;duración&gt;60&lt;/duración&gt;     &lt;título&gt; Las aventuras de los SGML&lt;/título&gt;   &lt;/película&gt; &lt;/cine&gt;</pre>	<b>CUERPO</b>

- Las etiquetas van encerradas entre el signo menor que (<) y mayor que (>) y suelen ir emparejadas, una de apertura y otra de cierre.
- También existen las etiquetas vacías.



Los atributos añaden propiedades a los elementos y van dentro de la etiqueta de apertura.

Ejemplo:

```
<email>666777888@prueba.com</email>
<casado v="no" />
```

#### 4.3.- Elaboración de documentos XML bien formados

Para poder elaborar un documento XML bien formado, es necesario seguir una serie de reglas, entre las que destacan:

- El W3C recomienda empezar por la instrucción donde se indique la versión (instrucción de procesamiento), por lo que esta ha de ser siempre la primera instrucción.

Ejemplo:

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



La declaración XML indica el juego de caracteres del documento.  
El juego de caracteres que se utiliza. En este caso es UTF-8

**\*\*\*NOTA\*\*\*** → El juego de caracteres se debe escribir en mayúsculas (UTF-8 en vez de utf-8).

**¡ATENCIÓN!**



Es importante que el juego de caracteres que aparece en la declaración sea el juego de caracteres en que realmente está guardado el documento, porque si no el procesador XML puede tener problemas leyendo el documento.

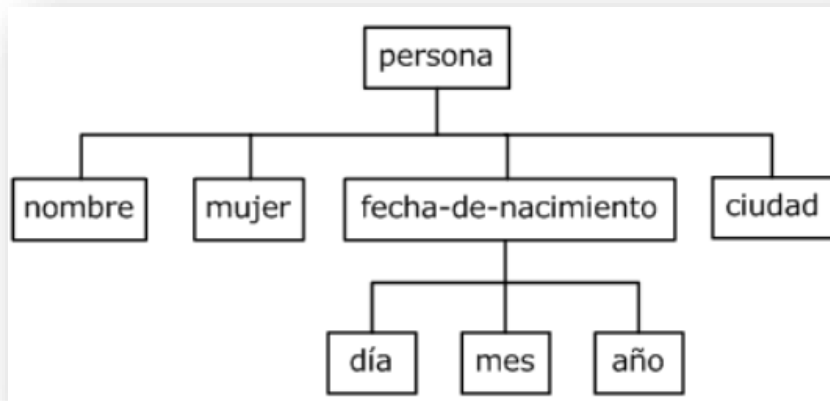
```
1 <?xml version="1.0" encoding="ISO859-1"?>
2
3 <prueba>
4   <texto>á é í ó ú </texto>
5 </prueba>
6
```

## This page contains the following errors:

error on line 1 at column 40: Unsupported encoding ISO859-1

Below is a rendering of the page up to the first error.

- Todo documento ha de tener una estructura jerárquica, en la que exista un único elemento raíz o principal, que será el primero que se abra y el último que se cierre.



Ejemplo: Elemento “fecha-de-nacimiento”:

```

<fecha-de-nacimiento>
  <día>18</día>
  <mes>6</mes>
  <año>1996</año>
</fecha-de-nacimiento>
  
```

```
<mujer v= "sí"/>
```

- Las etiquetas deben estar anidadas correctamente unas dentro de otras, cerrándose en orden inverso al que se abren. Los elementos no vacíos tendrán etiquetas de apertura y cierre.
  - Un elemento es un componente lógico de un documento que o bien comienza por una etiqueta de apertura y termina por la etiqueta de cierre correspondiente o que consiste en una única etiqueta vacía.
  - El contenido de un elemento es todo lo que se encuentra entre las etiquetas de apertura y cierre, incluso si estos son también elementos en cuyo caso se llaman elementos hijos.



Una etiqueta es un identificador que empieza por el carácter `<` y termina por `>`.

Existen tres tipos de etiquetas:

- las etiquetas de apertura (start-tag), que empiezan por el carácter `<` y terminan por `>`.

Por ejemplo: **`<apartado>`**

- las etiquetas de cierre (end-tag), que empiezan por los caracteres `</` y terminan por `>`.

Por ejemplo: **</apartado>**

- las etiquetas vacías (empty tag):
  - Las que no contienen nada: **<elemento></elemento>**
  - Y las que tienen auto cierre: **<elemento />**. Pueden tener atributos.

Ambas producen el mismo resultado

- Dentro de los atributos, los valores tienen que estar encerrados entre comillas dobles o simples y no podrá haber dos atributos que se llamen igual en un mismo elemento.
- Las etiquetas de cierre no contienen atributos.

```
<fecha-de-nacimiento>  
  <día>18</día>  
  <mes>6</mes>  
  <año>1996</año>  
</fecha-de-nacimiento>
```



```
<fecha-de-nacimiento dia="18" mes="6" año="1996" />
```

- A la hora de escribir las etiquetas, elementos o atributos, hay que tener en cuenta que es sensible a las mayúsculas y minúsculas.
- Los nombres de etiquetas, elementos y atributos tienen que seguir una nomenclatura específica como:
  - No empezar por número
  - No usar caracteres especiales reservados
  - No pueden empezar por la palabra "XML".
  - No puede empezar con número
  - Etc.

Tienen que empezar por:

- Letras
  - Seguidos de guiones, puntos, números u otras letras.
- Los comentarios no pueden ir dentro de las etiquetas.



Un comentario es una etiqueta que comienza por **<!--** y acaba por **-->**.

Los comentarios no pueden estar dentro de otras marcas y no pueden contener los caracteres "--".

Por ejemplo:

**<!--Esto es un comentario-->**

Ejemplo de documento XML bien formado:

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

<agenda>
  <!--vamos a crear un contacto-->
  <contactos>
    <nombre>Alumno</nombre>
    <ciudad cp="15500">Ciudad Real</ciudad>
    <telefono n="612345678"/>
    <email>alumno@gmail.com</email>
    <casado v="no"/>
  </contactos>
</agenda>
```


**EJEMPLO** En un documento XML, no se pueden escribir comentarios dentro de las etiquetas. Por ejemplo, no es correcto escribir:

**<alumno <!-- elemento vacío --> />**

**EJEMPLO** Por otro lado, hay que tener en cuenta que en los comentarios de un documento XML no está permitido usar dos guiones seguidos:

<!-- Dos guiones seguidos -- en un comentario da error -->

De forma que, **no es posible anidar comentarios en un documento XML.**



# ¡ATENCIÓN!

```
<?xml version="1.0" encoding="UTF-8"?>
<países>
  <país>
    <nombre>España</nombre>
    <capital>Madrid</capital>
  </país>
  <país>
    <nombre>Francia</nombre>
    <capital>París</capital>
  </país>
</países>
```

Dos formas de decir lo mismo

```
<?xml version="1.0" encoding="UTF-8"?>
<países>
  <país nombre="España" capital="Madrid" />
  <país nombre="Francia" capital="París" />
</países>
```



## RESUMEN

1. El documento debe tener un solo elemento raíz.
2. Todas las etiquetas (tags) abiertas deben tener sus respectivas etiquetas de cierre.
3. XML distingue mayúsculas de minúsculas por lo que todos los elementos y atributos deben seguir la definición.
4. Todos los elementos deben estar correctamente anidados.



5. Los valores de los atributos deben ir entre comillas simples o dobles.
6. Cualquier elemento Vacío debe terminar con /> o debe hacerlo no vacío añadiendo una etiqueta de cierre.
7. No se pueden repetir atributos en un mismo elemento. Por ejemplo, si se quiere representar múltiples autores para un libro, se debe definir el autor como un elemento y no como un atributo. Como norma general, se definirán elementos para los datos que van a ser leídos por las personas y atributos para aquellos que van a ser procesados por las máquinas

#### 4.5.- Procesador XML (XML processor) y aplicación (application)

- Cuando una aplicación necesita leer un documento XML, la aplicación recurre a un procesador XML.
  - 1) El procesador XML (o analizador XML, en inglés XML parser) es el que lee el documento
  - 2) analiza el contenido
  - 3) le transmite la información en un formato estructurado a la aplicación.

La recomendación XML especifica lo que debe hacer el procesador, pero no entra en lo que hace después la aplicación con esa información.

- Un procesador XML es pues, aquel que se encarga de **validar el documento**.

#### 4.6.- Documentos bien formados y válidos

Los documentos XML pueden ser correctos a dos niveles:

- **Bien formados** → Un documento XML se dice "bien formado" cuando cumple una serie de reglas descritas en la especificación oficial de XML v1.0.

- **Válidos** → Es el que está bien formado, y además cumple con la definición de un lenguaje de marcado particular especificado para el documento.

#### 4.7.- Entidades predefinidas

En XML se utilizan varias entidades de carácter de HTML, para poder escribir en cadenas de texto los caracteres que delimitan las marcas o las cadenas de texto.

**Esto generará un error de XML:**

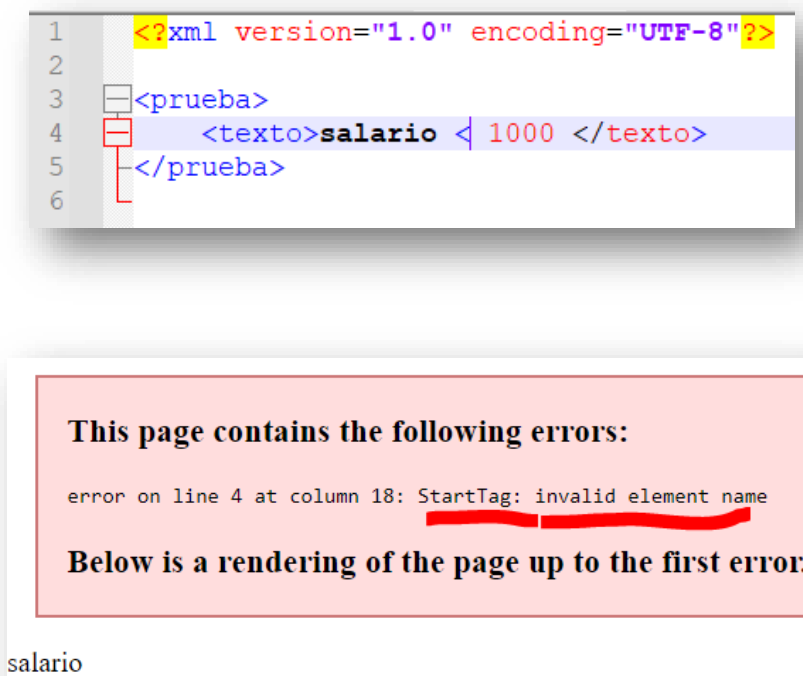


FIGURA 1. SALIDA DE ERROR DEL NAVEGADOR

Para evitar este error, hay que reemplazar el carácter "<" con una referencia de entidad:

Referencia a entidad	Carácter
&lt;	<
&gt;	>
&amp;	&
&apos;	'
&quot;	"

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <prueba>
4      <texto>salario &lt; 1000 </texto>
5  </prueba>
6

```



Échale un ojo:

Manual de UTF-8, Referencia HTML →

<http://www.w3big.com/es/charsets/ref-html-utf8.html#gsc.tab=0>

#### 4.8.- DTD y DOCTYPE

##### Definición de tipo de documento (DTD, Document Type Definition):

- Una DTD es un documento que define la estructura de un documento XML: los elementos, atributos, entidades, notaciones, etc., que pueden aparecer, el orden y el número de veces que pueden aparecer, cuáles pueden ser hijos de cuáles, etc.
- El procesador XML la utiliza para verificar si un documento es válido, es decir, si el documento cumple las reglas del DTD.

##### Declaración de tipo de documento (DOCTYPE, Document type declaration)

- Una declaración de tipo de documento es una etiqueta que comienza por **<!DOCTYPE** y acaba por **>** y que indica la(s) DTD(s) que debe utilizar el procesador XML para validar el documento. La DTD puede estar incluida en el propio documento o ser un documento externo.

Por ejemplo:

**<!DOCTYPE html>**

## 4.9.- Herramientas de edición

### 4.9.1.- Bloc de notas de Windows

El Bloc de notas de Windows (Notepad) es un editor de texto que viene con sistemas operativos Windows que permite crear documentos de texto plano.

### 4.9.2.- Gedit

Gedit es un editor de texto de fácil uso para el entorno de escritorio GNOME, aunque también está disponible para Windows.

### 4.9.3.- Notepad++

Es un potente editor de texto gratuito de licencia GPL para sistemas operativos Windows. Está escrito en C++ usando la API Win32.

## ACTIVIDAD 1.2

**Comprueba si el siguiente documento XML está bien formado. Para ello, ábrelo en un navegador que tengas instalado.**

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

<agenda>
  <!--vamos a crear un contacto-->
  <contactos>
    <nombre>Alumno</nombre>
    <ciudad cp="15500">Ciudad Real</ciudad>
    <telefono n="612345678"/>
    <email>alumno@gmail.com</email>
    <casado v="no"/>
  </contactos>
</agenda>
```

### ACTIVIDAD 1.3

1.- Comprueba si el siguiente documento XML está bien formado. Sino no lo estuviera, arréglalo para que cumpla con los estándares y sea validado por el navegador.

2.- Cuando hayas arreglado el documento XML, compruébalo en algún validador online que encuentres por internet.

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

<agenda>
  <!--Vamos a crear un contacto-->
  <contactos>
    <nombre>Alumno</nombre>
    <Ciudad cp="15500">Ciudad Real</ciudad>
    <teléfono n="612345678"/>
    <email>alumno@gmail.com</email>
    <casado v="no">
    <@AsignaturaPreferida>Matemáticas</@>
```

```
</contactos>  
</agenda>
```

## 5.- Glosario de términos

- **C# o C Sharp.** Lenguaje de programación de Microsoft para la plataforma .NET, de sintaxis similar a C++ y Java.
- **JavaScript.** Lenguaje de programación interpretado y poco tipado, usado mayormente en el lado del cliente junto con documentos HTML, haciéndolos dinámicos y mejorando su interfaz.
- **Unicode.** Estándar de codificación de caracteres mantenido por el UCT, que asocia a cada carácter un identificador numérico
- **Visual Studio.** Entorno de desarrollo que soporta los lenguajes C++, Java, C#, ASP.NET, etc.
- **W3C (World Wide Web Consortium).** Consorcio internacional creado en 1994 para generar estándares relacionados con la red informática mundial (www).
- **XMPP.** Protocolo extensible de mensajería y comunicación de presencia que está basado en XML y se emplea en mensajería instantánea.