



**UNIVERSIDAD DE PANAMÁ**  
**CENTRO REGIONAL UNIVERSITARIO DE LOS SANTOS**  
**FACULTAD DE INFORMÁTICA, ELECTRÓNICA Y COMUNICACIÓN**  
**ASIGNATURA: TECNOLOGÍA WEB 1**

## **ALTERNATIVAS PARA CREAR SITIOS WEB**



**PROFESOR: CÉSAR A. DELGADO B.**



@elprofecesard



@elprofecesard



@elprofecesard



@elprofecesard



@elprofecesard



elprofecesard.com

## Alternativas para crear de Sitios Web

El desarrollo de Sitios Web es amplio y diverso, por lo que se debe identificar de antemano cuál será su propósito en Internet. De ello depende que tipo de tecnología utilizar, ya que podemos encontrar en Internet sitios personales que no requieren niveles avanzados de seguridad y otros que requieren de asegurar la información.



El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en Linux o el Bloc de notas de Windows. Existen además otros editores para la realización de sitios web con características WYSIWYG (What You See Is What You Get, o en español: "lo que ves es lo que obtienes"). Estos editores permiten ver el resultado de lo que se está editando en tiempo real, a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino que una forma un tanto más simple ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML, la cual va generando todo el código a medida que se va trabajando. Algunos ejemplos de editores son Adobe Dreamweaver, KompoZer o Microsoft FrontPage.

Estos editores aceleran o facilitan la creación de código HTML, pero en algunas ocasiones también generan mucho más código del necesario (como es el caso de Microsoft FrontPage). Lo ideal es tener un control total sobre el código que se escribe y utilizar estos editores sólo como una pequeña ayuda. También podemos utilizar otro tipo de editores

que simplemente comprueben que el código HTML escrito es correcto (que las etiquetas y atributos son correctos, las etiquetas se cierran correctamente, etc.)

## Editores

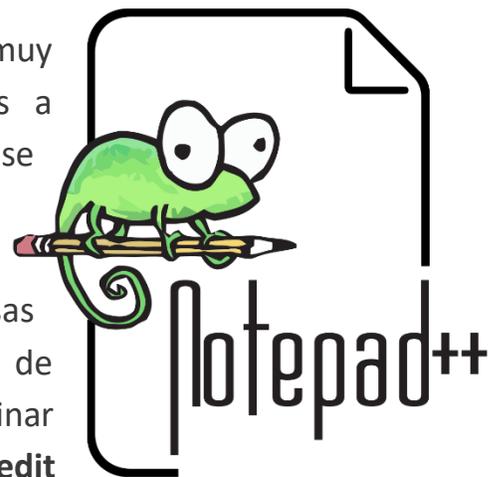
Un **editor de páginas web** o *editor HTML*, es una aplicación diseñada con el fin de facilitar la creación de documentos HTML. Aunque el código HTML de una página web puede ser escrito con un simple editor de texto plano, los editores de páginas web específicos ofrecen



numerosas funciones adicionales, como editor **WYSIWYG** o soporte para **CSS** y **JavaScript**. Existen varios tipos de editores y de diferente complejidad.

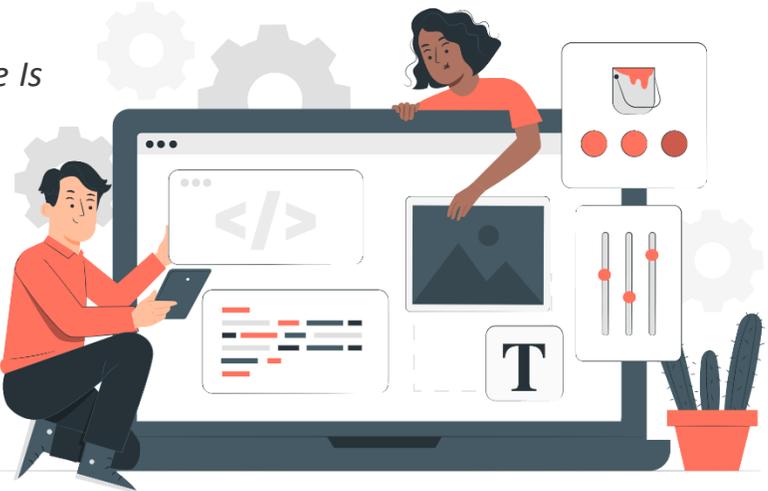
## Editores de texto sin formato

El **editor de texto plano** o **texto sin formato** es de uso muy sencillo. Los programas más sencillos son similares a **Notepad** o **Bloc de Notas**, y con cualquiera de ellos se puede escribir las líneas de código necesarias para crear un sitio web. Sin embargo, existen editores de texto plano avanzados que disponen de numerosas características avanzadas, como resaltado y coloreado de sintaxis, así como botones para insertar y combinar etiquetas **HTML**, como es el caso de **Kate** o **Gedit** (**GNU/Linux**).

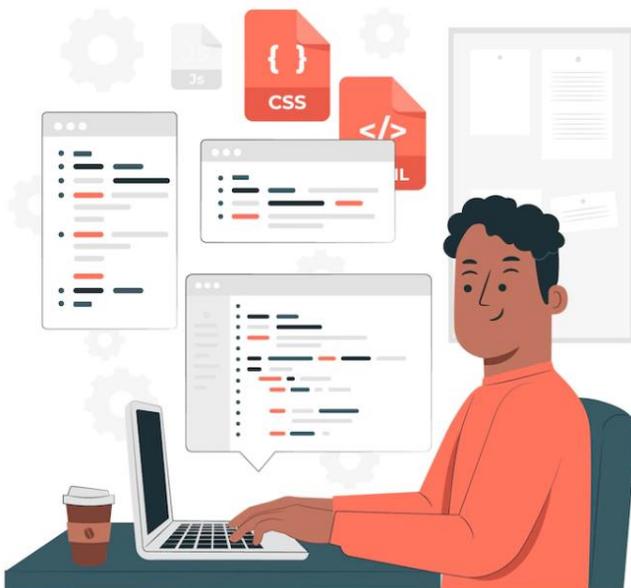


## Editor WYSIWYG

Un editor **WYSIWYG** (*What You See Is What You Get, Lo que ves es lo que obtienes*), es un editor en el que la vista de trabajo es a la vez la vista previa, de tal manera que permite colocar elementos en la página de manera visual, siendo el propio editor el que se encarga de generar el **código HTML**. De esta



manera, el usuario teclea el contenido, da formato e inserta imágenes y el editor se encarga de transformar la vista a **código HTML**, debidamente configurado. La ventaja de este tipo de editores es que brindan numerosas funcionalidades, mientras que el inconveniente es que suelen generar un código HTML de escasa calidad, con mucho código basura.



Los editores más avanzados permiten además maquetar la página, y no sólo trabajan con **HTML**, sino que también permiten editar **CSS**, **XML** o **JavaScript**, e incluso algunos permiten conectar con servidores remotos vía **FTP** o **WebDAV**, y con sistemas de control de versiones como **CVS** o **Subversion**.

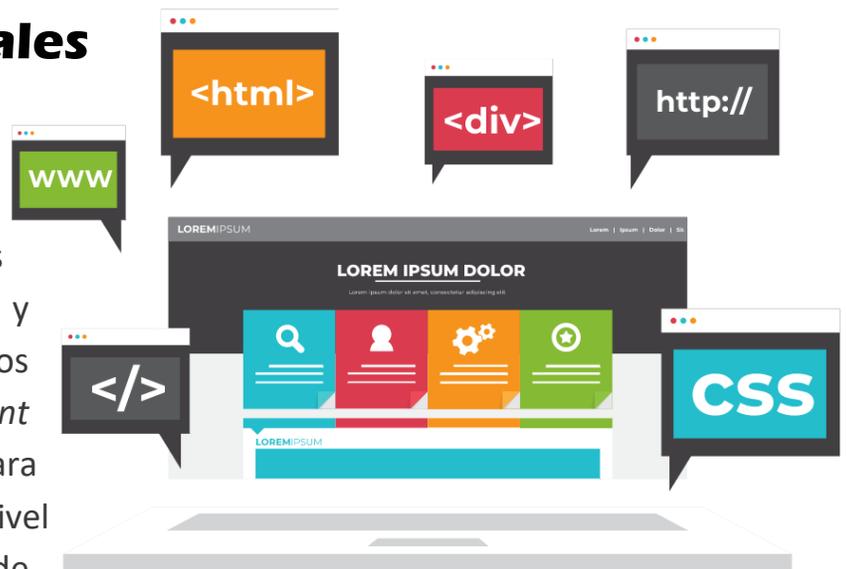
Editores **WYSIWYG** sencillos son **Word** o **Writer**, mientras que otros más sofisticados son **KompoZer**, **Amaya**, **Microsoft Frontpage**, **Mozilla Composer** o **Adobe DreamWeaver**.

## Editor de texto con ventanas desdobladas

Se trata de un editor que combina un **editor de texto sin formato** y un **editor WYSIWYG**, de tal manera que cuenta con dos ventanas, una de trabajo en d onde se ingresa el **código HTML** en formato de texto plano y otra en la que se visualiza el resultado final en tiempo real, sin necesidad de guardar el documento para luego abrirlo y visualizarlo, lo que resulta muy eficiente a la hora de previsualizar el resultado.

## Editores profesionales

Existen una multitud de editores de código para páginas o aplicaciones web, y en los últimos años ha crecido el mercado y también las alternativas. Aun así, los IDE (de *Integrated Development Environment* en inglés) líderes para editar código o programar a nivel profesional y en todo tipo de lenguaje de programación (*HTML, CSS, Javascript, Ruby, Go...*) son:



- VSCode, propiedad de Microsoft pero totalmente gratuito.
- WebStorm, de la empresa JetBrains, especializado en Javascript.
- Aptana, un IDE de software libre.
- PyCharm, también de JetBrains, pero especializado en Python.
- SublimeText

## Aplicaciones No Code y Low Code

No code o low code es una filosofía digital que permite a cualquier persona crear aplicaciones y programas sin necesidad de saber programación. Es lo que se conoce como “programación sin código” o “programación visual” que permite a muchas personas y empresas sin conocimientos tecnológicos crear Sitios Web, aplicaciones o negocios en línea sin necesidad de programar una sola línea de código. Los más conocidos son: **Shopify, Wix, WordPress, Joomla, Magento**, entre otros.



**No code:** son plataformas pensadas para usuarios que no tienen ningún conocimiento de programación, de manera que no requieren nada de código. Son útiles para crear aplicaciones con un tiempo de vida corto, en casos de negocio simples o procesos manuales que no requieran conexiones a sistemas de terceros, entre otros.

**Low code:** son plataformas con un contenido bajo en código, que con unas nociones muy básicas de programación manual permiten cierta personalización de las funcionalidades, los modelos y las plantillas preconfiguradas que ofrece la herramienta. Se utilizan para generar aplicaciones con un ciclo de vida mayor, soluciones que se tengan que integrar con sistemas corporativos más complejos o la creación Productos Mínimos Viables (MVP), por ejemplo: **Wordpress, Honeycode, Appsheet, PowerApps, Figma**.

## Convertidores

Las herramientas de conversión permiten transformar un documento escrito en otro formato, a un documento HTML. El código generado es casi imposible de leer, debido a que hay muchas distinciones que existen en lenguajes de formateo, se usan cuando no hay otra alternativa. Para usarlos, la idea básica es simplificar al máximo el documento, (eliminar sangrías, uniformar los tamaños y formatos de texto; Para luego pasar al conversor y convertirlo en HTML.



## Lenguajes de Mercado

Los lenguajes de marcado genéricos son aquellos que sirven para especificar la estructura de cualquier documento sin tener en cuenta los aspectos relativos a la presentación. Esto supone que el mismo documento se puede presentar de muchas maneras, de acuerdo con las normas de estilo que se le apliquen. A continuación, se presentan los lenguajes más conocidos y que se utilizan para la publicación en Web.

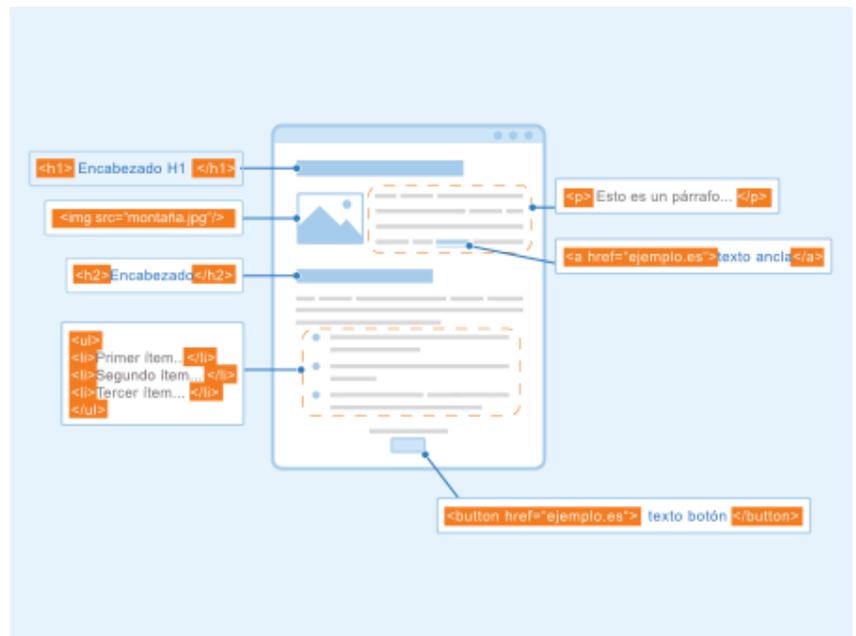


Figura 1 Lenguaje de marcado - Autor: Seobility - Licencia: CC BY-SA 4.0

El lenguaje de marcado o de marcas se refiere a un sistema de anotación organizado (es decir, “lenguaje”) que marca, de ahí su nombre, ciertas partes o elementos de un documento o aplicación web para, principalmente, dar al texto tanto un formato como una estructura específica.

Otra característica es que, para dar formato o estructura a una página, el lenguaje de marcas utiliza una serie de etiquetas (a menudo entre paréntesis angulares) cuyo símbolo es `<>` y que funcionan como instrucciones que, al estar contenidas entre `<>`, se diferencian sintácticamente del texto sin formato.

Las etiquetas son leídas e interpretadas por parsers o analizadores que luego generan una salida, generalmente como una página o un documento que tiene el formato y el estilo correspondiente y manteniendo la marca real oculta a simple vista.

Por ejemplo, en lenguaje HTML, la etiqueta `<h1>` indica el elemento de encabezado principal de una página web, advirtiendo a la máquina que presente el encabezado en negrita o con una fuente más grande según las convenciones estándar de HTML. No obstante, quien navegue por esta página únicamente verá el encabezado con formato, en lugar de las etiquetas.

## SGML

SGML (Standard Generalized Markup Language) es un ejemplo de lenguaje genérico que apareció con el identificador 8879 como norma ISO (International Organization for Standardization) en 1986. La comunidad editorial fue la que dio origen a esta norma,

al considerar que la flexibilidad en el diseño de los documentos era de máxima importancia. El objetivo que perseguía era proporcionar una manera normalizada de



transmitir los documentos en un formato adecuado para los procesos de edición e impresión. SGML es apropiado para describir texto altamente estructurado, aunque también se pueden incluir en los documentos otros elementos, como por ejemplo diagramas y gráficos, independientemente de su formato de codificación. SGML contiene las reglas para crear una infinita variedad de lenguajes de marcado, pero no describe el formato de los documentos marcados. Una definición similar clasifica a SGML como un sistema para especificar lenguajes de marcado, es decir, un metalenguaje. Esto hace posible que, mediante la utilización de una definición de tipo de documento (denominada DTD Document Type Definition), se pueda especificar la estructura lógica de una clase de escrito.

### **XML**

XML es una aplicación de SGML, lo que significa que en su especificación se indican como se deben describir los elementos que participan en el hiperdocumento, pero no los elementos en sí. Por tanto, cuando se quiere describir un documento mediante XML hay que describir en primer lugar el tipo de documento en que se basa, es decir la DTD, y a continuación los contenidos concretos asociados a cada elemento.

Un ejemplo de construcción de aplicaciones con XML es el desarrollo de normas para tipos concreto de documentos, como puede ser un libro electrónico. Una iniciativa en este campo, el desarrollo de una DTD para libros electrónicos a partir de XML es OpenEBook, un estándar en el que numerosas empresas del sector bibliográfico están trabajando de manera que con una misma plataforma se puedan leer, comprar, distribuir, etc. libros electrónicos que tengan un formato común.



## HTML

**HTML** es una aplicación de SGML que incluye tipos de documentos predefinidos. Por ello, contienen los mismos elementos y atributos, es decir todos los documentos de este tipo tienen la misma estructura, pero no los mismos contenidos. La última versión, **HTML5** (Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*) es el componente más básico de la Web, ya que define el significado y la estructura del contenido web.



## XHTML

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje 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 los diseñadores escogen XHTML. En un futuro cercano, si los diseñadores deben elegir entre HTML 5 y XHTML 1.1 o XHTML 2.0, quizás la elección sea diferente.



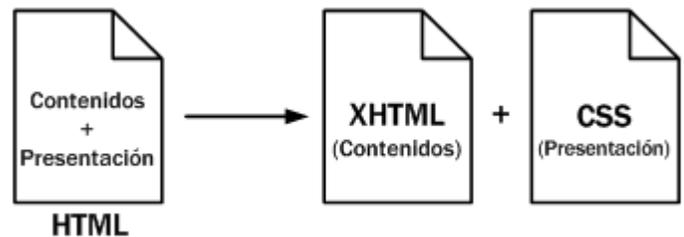
## CSS

Originalmente, las páginas HTML sólo incluían información sobre sus contenidos de texto e imágenes. Con el desarrollo del estándar HTML, las páginas empezaron a incluir también información sobre el aspecto de sus contenidos: tipos de letra, colores y márgenes.



La posterior aparición de tecnologías como JavaScript provocaron que las páginas HTML también incluyeran el código de las aplicaciones (llamadas *scripts*) que se utilizan para crear páginas web dinámicas.

Incluir en una misma página HTML los contenidos, el diseño y la programación complica en exceso su mantenimiento. Normalmente, los contenidos y el diseño de la página web son responsabilidad de diferentes personas, por lo que es conveniente separarlos.



CSS separa los contenidos definidos mediante XHTML y el aspecto que deben presentar esos contenidos:

Una ventaja añadida de la separación de los contenidos y su presentación es que los documentos XHTML creados son más flexibles, ya que se adaptan mejor a las diferentes plataformas: pantallas de ordenador, pantallas de dispositivos móviles, impresoras y dispositivos utilizados por personas discapacitadas.

De esta forma, utilizando exclusivamente XHTML se crean páginas web "*feas*" pero correctas. Aplicando CSS, se pueden crear páginas "*bonitas*" a partir de las páginas XHTML correctas.

### Otros lenguajes de programación

## **SMIL (Synchronized Multimedia Integration Language)**

SMIL es un lenguaje basado en XML para la definición de aplicaciones multimedia interactivas, de manera que un autor puede describir el comportamiento temporal de presentaciones multimedia, asociar enlaces a contenidos de cualquier tipo (por ejemplo, vídeos, sonidos, programas, etc.) y describir la presentación en la pantalla. SMIL (se pronuncia smile –sonrisa–) no es una solución que intente competir con tecnologías existentes de representación multimedia (por ejemplo, Quicktime o Flash) sino que lo que pretende es integrar esas tecnologías de manera estándar para que puedan combinarse. Existen numerosos visualizadores de documentos SMIL que permiten mostrar todas sus características. Algunos de los más conocidos son: RealPlayer, Quicktime Player (la versión gratuita sólo permite ver documentos muy sencillos) y GRINS Player. Además, algunos navegadores Web, como Internet Explorer, permiten visualizar también estos documentos. Un documento SMIL se compone de una cabecera (HEAD) y un cuerpo (BODY). En la cabecera se definen tanto la metainformación del documento como información relativa a cómo deben aparecer en la pantalla los elementos de información. En el cuerpo del documento se incluirán los contenidos del documento, así como las relaciones entre ellos. En el siguiente ejemplo, se muestra un documento SMIL que muestra las dos partes que se han mencionado anteriormente. En la primera se establece el nombre del autor del documento (“pepe”) y se divide la pantalla en dos zonas, una denominada “video” y la otra “imagen”, definiendo el punto superior izquierda de la zona (“top” y “left”) y el ancho y alto de la misma (width y height). En el cuerpo del documento se indica que se muestren de forma simultánea (“par”) tres contenidos, un vídeo, una imagen y un sonido. Además, se indica que el video se muestre en la región que se ha denominado “video” y la imagen en la que ha denominado “imagen”.

### **WML (Wireless Markup Language)**

El denominado protocolo WAP (Wireless Application Protocol) permite el desarrollo de aplicaciones sobre dispositivos móviles a través de redes inalámbricas. Se verá cómo se desarrollan aplicaciones WAP sencillas, sin entrar en detalles sobre la arquitectura de sus protocolos subyacentes. Para ello, se verá exclusivamente en los lenguajes WML y WMLScript, que son los equivalentes dentro del mundo “inalámbrico” al HTML y al JavaScript (o lenguajes similares, como VBScript o ECMAScript) dentro de las redes que usan el protocolo TCP/IP como hace Internet. Los documentos WML pueden mostrarse en teléfonos móviles, pero también en cualquier otro dispositivo que contenga un micronavegador (es decir, un dispositivo que sepa interpretar WML y WMLScript), como puede ser una agenda personal (PDA) o una aplicación en nuestro ordenador personal. Por ello, en ocasiones se hará referencia a cualquiera de estos dispositivos con el término general “agente de usuario” en lugar de hablar exclusivamente de los teléfonos móviles. Los documentos escritos en WML pueden accederse mediante URLs, de manera similar a cómo se hace con las páginas HTML. Para ello, se necesita un navegador WML, que puede estar empotrado en un teléfono móvil o ser un emulador que se utiliza desde un ordenador personal. En la figura 2 se puede ver uno de estos emuladores de teléfono móvil, en concreto del teléfono “Nokia 6210”, el cual se puede utilizar para navegar por esas páginas.

El lenguaje WML es un lenguaje de descripción de páginas que permite definir la presentación de la información en el teléfono móvil, solicitar entradas del usuario y responder a ciertas interacciones del usuario con el móvil, como puede ser la pulsación de una tecla. Este lenguaje se basa en una DTD de XML por lo que todo documento WML es a su vez un documento XML tabla 14.9. Los documentos WML están formados una definición de tipo de documento, y un mazo de cartas que se corresponden con las pantallas que se visualizan en el visor del dispositivo. A continuación, se presenta un ejemplo de documento WML que incluye una única carta en el mazo y su visualización (dependiendo de las dimensiones y tipos de letra del mismo). Todos estos ejemplos de lenguajes de marcado se basan en la definición de la estructura de los elementos, indicando cuáles existen en el dominio del documento y como participan en él, es decir lo que anteriormente se ha denominado lenguajes de marcado genéricos. Pero estos

lenguajes de marcado se complementan con otros lenguajes que se utilizan para describir cual es la apariencia final en el soporte físico de visualización.

## Referencias Bibliográficas

- Corporativa, I. (s/f). *Qué es el low code o programación sin código*. Iberdrola. Recuperado el 10 de septiembre de 2022, de <https://www.iberdrola.com/innovacion/low-code>
- *Editores y conversores*. (s/f). GUIA DE HTML. Recuperado el 1 de diciembre de 2022, de <http://guiahtml.weebly.com/editores-y-conversores.html>
- *Máster en Ingeniería de la Web Medios digitales. Edición, codificación e integración 2 de 8*. (s/f). Uned.es. Recuperado el 30 de noviembre de 2022, de [http://ocw.innova.uned.es/mm2/tm/contenidos/pdf/tema2/lenguajes\\_marcado.pdf](http://ocw.innova.uned.es/mm2/tm/contenidos/pdf/tema2/lenguajes_marcado.pdf)
- Prácticas, G. (2013, junio 11). *Editor de páginas web*. Guías Prácticas. <https://www.guiaspracticas.com/desarrollo-web/disenio-web/editor-de-paginas-web>
- *¿Qué es el Lenguaje de Marcado? - Seobility Wiki*. (s/f). Seobility.net. Recuperado el 1 de diciembre de 2022, de [https://www.seobility.net/es/wiki/Lenguaje\\_de\\_marcado](https://www.seobility.net/es/wiki/Lenguaje_de_marcado)
- Uniwebsidad. (s/f). *Introducción a XHTML. Capítulo 2. Características básicas*. Uniwebsidad.com. Recuperado el 1 de diciembre de 2022, de <https://uniwebsidad.com/libros/xhtml/capitulo-2>



### Importante

Ni la totalidad ni parte de este trabajo pueden reproducirse, registrarse o transmitirse, por un sistema de recuperación de información, en ninguna forma ni por ningún medio, sea electrónico, mecánico, fotoquímico, magnético o electroóptico, por fotocopia, grabación o cualquier otro, sin permiso previo. Este documento se distribuye con fines exclusivamente didácticos y es de uso exclusivo para estudiantes de la asignatura Multimedia 1 y 2 que se imparte en el CRU-Los Santos de la Universidad de Panamá y de Undécimo (11°) grado del Centro Educativo Federico Zúñiga Feliú por el profesor César Delgado.