

Material didáctico para exposición teórica

Familia profesional	IFC	Informática e comunicacións
Ciclo formativo	CSIFC03 CSIFC02 CSIFC01	Desenvolvemento de aplicacións web Desenvolvemento de aplicacións multiplataforma Administración de sistemas informáticos en rede
Grao		Superior
Módulo profesional	MP0373	Linguaxes de marcas e sistemas de xestión de información
Unidade didáctica	UD01	Utilización de linguaxes de marcas en contornos web
Actividade	A01	Introdución ás linguaxes de marcas
Profesora		María del Carmen Fernández Lameiro

1.	A01. Introducción ás linguaxes de marcas relacionadas coa web	3
1.1	Obxectivos	3
1.2	Definición e orixes	3
1.3	Características xerais	4
1.4	Vantaxes para o tratamento da información.....	4
1.5	Clasificación	5
1.6	Ámbitos de aplicación	5
1.7	Linguaxes empregadas nas aplicacións web	7
1.7.1	Linguaxes do lado do cliente	7
1.7.2	Linguaxes do lado do servidor	8
1.7.2.1	Funcionamento	8
1.8	As linguaxes HTML e XHTML.....	9
1.8.1	Características xerais dunha páxina HTML ou XHTML	9
1.8.2	Diferenzas entre HTML e XHTML	10
1.8.3	Estrutura dunha páxina HTML ou XHTML	11
1.8.4	Sistema de codificación.....	12
1.8.5	Referencias e entidades.....	13
1.9	A Linguaxe CSS	14
1.10	Ferramentas Web	15
2.	Documentos de apoio ou referencia	19

1. A01. Introducción ás linguaxes de marcas relacionadas coa web

1.1 Obxectivos

A presente actividade está orientada a conseguir os seguintes obxectivos:

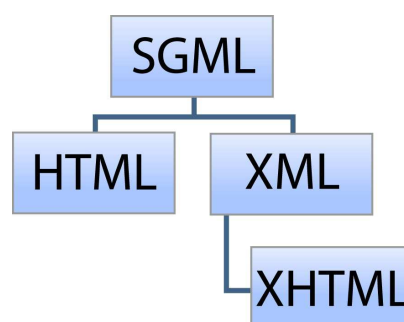
- Identificar e clasificar as linguaxes de marcas relacionadas coa web e as súas versións.
- Analizar a estrutura dun documento HTML e identificar as seccións que o compoñen.
- Recoñecer as semellanzas e diferenzas entre HTML e XHTML.
- Identificar ferramentas Web e utilizar basicamente as ferramentas elixidas polo profesor.

1.2 Definición e orixes

Unha linguaxe de marcado ou de marcas é unha linguaxe informática que permite incorporar etiquetas, marcas ou sinais nun texto para dar información adicional acerca da súa estrutura, formato, presentación ou calquera outro aspecto que se queira marcar. Teñen a súa orixe nas marcas que se colocaban nos textos para que o impresor soubera como tiña que facer a impresión dos mesmos e que se foron modificando e adaptando cando apareceu o tratamento automático da información mediante ordenadores.

As linguaxes de marcas non son linguaxes de programación xa que non dispoñen de mecanismos para manexar variables na memoria, nin estruturas de control.

A xeneralización no uso das linguaxes de marcas empeza na década de 1970 cando Charles Goldfarb xunto con Edward Mosher e Raymond Lorie, investigadores para IBM, deseñan a linguaxe GML (Generalized Markup Language e tamén as siglas dos apelidos dos seus creadores).



En 1980, Charles Goldfarb dirixe o comité destinado a estandarizar GML, nacente en 1986 o primeiro estándar SGML (Standard Generalized Markup Language) que cumpre a norma ISO 8879; esta é unha metalinguaxe (permite definir outras linguaxes de marcado) moi potente, que tivo gran aceptación pero era unha linguaxe complexa e difícil de aprender e utilizar.

Tim Berners-Lee creou en 1989, a linguaxe HTML (HyperText Markup Language) baseada en SGML, moito máis sinxela, orientada á representación de información estática a través dun navegador pero que non é unha metalinguaxe. Esta linguaxe tivo un papel fun-

damental no crecemento de internet xa que con poucos coñecementos de informática, podía escribir unha páxina web estática; isto que parece unha vantaxe, tamén foi un problema, xa que se xeraron moitos documentos mal estruturados sen respecto polos estándares.

XML (Extensible Markup Language) foi desenvolvido polo W3C¹, en 1996 para simplificar SGML e adaptalo para o seu uso en internet. É unha metalinguaxe estándar, estruturada, extensible e que pode ser validada. Estase convertendo nun estándar para a transmisión de información en internet.

Unha linguaxe derivada de XML e HTML é XHTML que une as características das dúas linguaxes.

1.3 Características xerais

As linguaxes de marcas teñen as seguintes características:

- Texto plano.

Os documentos escritos coas linguaxes de marcas son arquivos de texto plano e polo tanto poden ser interpretados directamente e son independentes da plataforma na que foi creado. Por exemplo, pode ser lidos mediante un sinxelo editor de texto nun sistema operativo Windows ou Linux.

- Compacto.

As instrucións de marcado mestúranse co contido nun único arquivo.

- Extensible.

Baseándose nunha linguaxe de marcas poden desenvolverse novas linguaxes especializadas para obxectivos concretos. Por exemplo unha linguaxe que permita a elaboración de manuais pode especializarse para a elaboración de manuais para a industria de software, ou hardware, ou telecomunicacións.

- Flexibilidade.

Permiten marcar documentos de texto pero tamén utilizarse con gráficos vectoriais, servizos web, sindicación de contidos, ou interfaces de usuario e incluso poden combinarse diferentes linguaxes de marcas no mesmo arquivo como por exemplo XHTML, marcado para fórmulas matemáticas (MathML) e gráficos vectoriais (SVG).

1.4 Vantaxes para o tratamento da información

As vantaxes que teñen as linguaxes de marcas para o tratamento da información están baseadas nas características das mesmas e son as seguintes:

- A información marcada para un fin específico pode perdurar no tempo e pode volver a utilizarse. Por exemplo: sobre un grupo de datos marcado con XML, poden facerse transformacións utilizando XSLT para crear unha páxina XHTML con eses datos ou para crear unha nova estrutura de datos.

¹ Siglas polas que se coñece ao World Wide Web Consortium, que é un organismo sen ánimo de lucro que recomenda estándares para a Web. A páxina oficial é <http://www.w3.org/>

- Poden establecerse controis sobre a información marcada e validar de forma automática que a estrutura verifique certas condicións.
- A información marcada non depende da plataforma sobre a que foi creada, se non que ao estar gardada nun arquivo de texto, poderá ser portada a outra plataforma.

1.5 Clasificación

Poden distinguirse tres clases de linguaxe de marcado:

- **Marcado de presentación**
 Serve para indicar como se vai a presentar o documento. Un exemplo pode ser HTML que permite modificar o aspecto que terá a información marcada no navegador. Outro exemplo son os editores de texto plano que permiten poñer retornos de carro e saltos de liña aínda que ao usuario se lle oculten esas marcas.
- **Marcado de procedemento**
 Permiten marcado de presentación e ademais teñen instrucións para procesar o texto do documento. Algúns autores inclúen este tipo de marcado no marcado de presentación. Un exemplo pode ser TeX e LaTeX que se utilizan para escribir documentos que conteñen fórmulas matemáticas (ver exemplo en <http://es.wikipedia.org/wiki/LaTeX>). Outro exemplo é Postscript que se utiliza para escribir gráficas que poden ser impresas en calquera impresora (ver exemplo en <http://es.wikipedia.org/wiki/PostScript>).
- **Marcado descritivo ou semántico**
 Permite describir as diferentes partes en que se estrutura o documento pero sen especificar como deben representarse ou en que orde deben aparecer. Por exemplo *SGML* e os seus derivados como XHTML (ver exemplo de epígrafe 3.1.1 de <http://www.w3.org/TR/xhtml1/>) ou XML (ver exemplo en [http://es.wikipedia.org/wiki/Extensible Markup Language](http://es.wikipedia.org/wiki/Extensible_Markup_Language)).

1.6 Ámbitos de aplicación

Hoxe en día a linguaxe de marcas por excelencia é XML, que é unha linguaxe de marcas multipropósito que permite definir outras linguaxes de marcas, estandarizado polo W3C; é descendente directo de SGML, do que tamén descende HTML.

A continuación expóñense algunhas das linguaxes de marcas máis salientables (a maior parte delas definidas mediante XML), clasificadas segundo o seu ámbito de aplicación:

- **Contidos web.** As linguaxes HTML e XHTML permiten definir as estruturas dos documentos web.
- **Procesadores de textos.** Empregan linguaxes de marcado para definir o formato e aparencia dos documentos, como por exemplo, os formatos de arquivo ODF ou RTF.
- **Sindicación de contidos.** Esta permite difundir información actualizada frecuentemente a usuarios subscritos á fonte de contidos, como por exemplo a linguaxe RSS.

- **Servizos web.** As linguaxes de marcas facilitan enormemente a transferencia de información, como por exemplo, a linguaxe SOAP que se emprega para definir a información que intercambian distintos servizos web.
- **Gráficos vectoriais.** Por exemplo, o formato de arquivo SVG (Scalable Vector Graphics) permite describir completamente complexas imaxes de dúas dimensións empregando unicamente texto.
- **Listas de reprodución.** Na actualidade, moitos dos reprodutores multimedia empregan linguaxes de marcas para definir as súas listas de reprodución, como por exemplo, os formatos ASX, PLA ou WPL.
- **Arquivos de configuración.** Moitos programas e servizos (por exemplo o servidor de aplicacións Tomcat) empregan linguaxes de marcas para os seus arquivos de configuración.
- **Linguaxe de marcas de propósito xeral.** As linguaxes anteriormente citadas están baseadas en XML. Esta é unha linguaxe de marcas de propósito xeral que permite definir linguaxes concretas de representación de documentos.

Exemplos de ámbitos de aplicación das linguaxes de marcas segundo Wikipedia (http://es.wikipedia.org/wiki/Lenguajes_de_marcas):

- Principais: GML->SGML->XML->Derivados de XML
- Documentos en xeral:
 - Linguaxes descritivas: ASN.1, EBML, YAML
 - Linguaxes de presentación: Rich Text Format, S1000D, TeX, troff, HTML
 - Para manuais: DocBook, HelpML, LinuxDoc, POD, Microsoft Assistance ML
 - Outras linguaxes: BBCode, Markdown, ReStructuredText, setext, Textile, Wikitexto
- Tecnoloxías de internet:
 - WWW: HTML, XHTML, Wireless ML, Handhelp ML, RDF, MetaContent Framework
 - Interfaz de usuario: GladeXML, MXML, User Interface ML, XAML and MyXaml, Xforms, XUL/XBL
 - Sindicación: Atom, RSS, ICE, OPML e OML, SyncML
 - Servizos web: WSDL, XINS, WSCL, WSFL, XML-RPC, Webml
- Linguaxes especializados:
 - Gráficos 2D: SVG, CGM, VML, InkML
 - Gráficos 3D: VRML/X3D, STEP
 - Matemáticas: MathML e OpenMath
 - Música: LilyPond e MusicXML
 - Finanzas: eXtensible Business Reporting Language, Financial products ML
 - Información xeográfica: Geography ML
 - Multimedia: Synchronized Multimedia Integration Language
 - Voz: VoiceXML
 - Videoxogos: BulletML, COLLADA

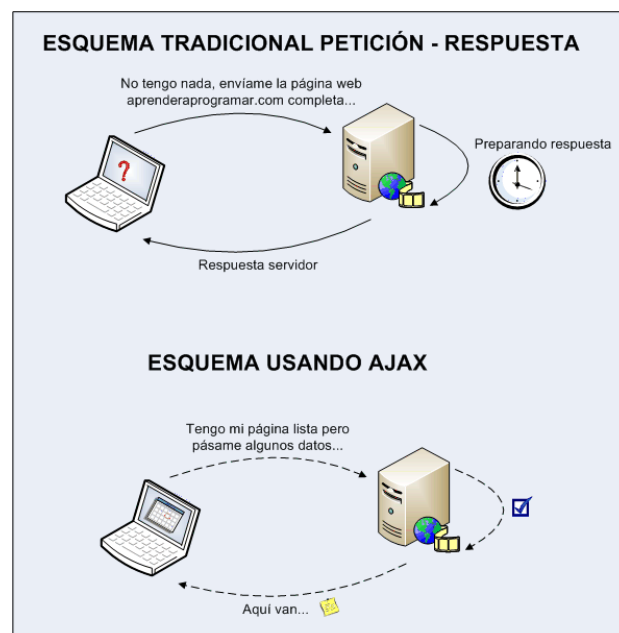
1.7 Linguaxes empregadas nas aplicacións web

As linguaxes empregadas para a creación de aplicacións web poden clasificarse en dous tipos: as linguaxes que serán interpretadas do lado do cliente e/ou do lado do servidor.

1.7.1 Linguaxes do lado do cliente

As linguaxes do lado do cliente son aquelas que poden ser interpretadas directamente polo navegador web. Son completamente independentes do servidor, o que permite que a páxina poida ser aloxada en calquera servidor. O único que fai falta é que o navegador teña instalados os plugins adecuados. O código é visible polo cliente, o que pode afectar á seguridade. Estas son as máis importantes:

- **HTML** (Hypertext Markup Language). Trátase dunha linguaxe de marcas que permite crear contido estático nos documentos web, definindo a súa estrutura.
- **CSS** (Cascade Style Sheet). É unha linguaxe empregada para definir a presentación dun documento web.
- **JavaScript**. É unha linguaxe de programación que permite crear scripts do lado do cliente para, por exemplo, controlar o navegador ou alterar o contido que se amosa nos documentos web.
- **AJAX** (Asynchronous JavaScript and XML). Trátase un conxunto de tecnoloxías web que permite que as páxinas web se actualicen de forma asíncrona intercambiando pequenas cantidades de datos co servidor de xeito que sexa posible actualizar partes da páxina sen recargar a páxina completa. As páxinas web que non utilizan AJAX deben recargar a páxina completa cando o contido cambia².



² Imaxe extraída do curso "Tutorial básico del programador web: JavaScript desde cero" de César Krall. http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=882:ique-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-javascript-asincrono-xml-y-json-cu01193e&catid=78:tutorial-basico-programador-web-javascript-desde-&Itemid=206

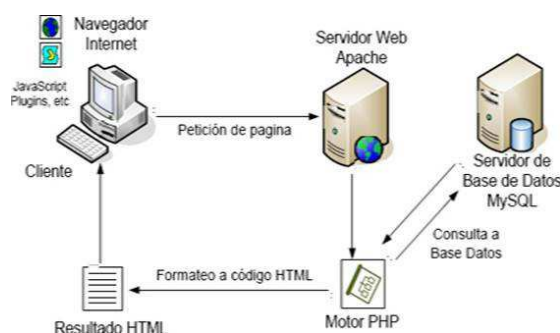
1.7.2 Linguaxes do lado do servidor

Son aquelas linguaxes que son recoñecidas, executadas e interpretadas polo propio servidor e se envían ao cliente nun formato comprensible para el. Son independentes do navegador utilizado. Ademais, o código pode ocultarse ao cliente, que só verá o código HTML finalizado. Algunhas das máis populares son:

- **PHP.** Linguaxe de script multiplataforma creado especialmente para a xeración de páxinas web dinámicas, posto que pode ser incrustado dentro do código HTML.
- **Java: páxinas JSP e Servlets.** Orientadas a desenvolver páxinas web en Java. Nas páxinas JSP, o código Java inclúese no documento HTML e execútase no servidor para incluír contido nestas de forma dinámica. Mentres que os Servlets son aplicacións Java completas no servidor que teñen como uso máis común a xeración de forma dinámica de páxinas web a partir dos parámetros que envíe os navegadores web.
- **ASP.NET.** É a sucesora de Active Server Pages (ASP) desenvolvido e comercializado por Microsoft. Consiste nun conxunto de clases .NET utilizadas para crear páxinas web. Utilízase en servidores Windows.
- **Python.** É unha linguaxe multiplataforma e multiparadigma, xa que permite o desenvolvemento de estilos diferentes de programación. É unha linguaxe simple, versátil e rápido de desenvolver. Incide nunha sintaxe que favoreza un código lexible.
- **Perl.** É unha linguaxe de programación moi práctica para extraer información de arquivos de texto e xerar informes a partir do contido dos ficheiros.

1.7.2.1 Funcionamento

No seguinte gráfico móstrase o proceso iniciado dende que un navegador cliente que utilizando protocolo http ou https, realiza unha petición dunha páxina web (con código PHP e información dunha base de datos MySQL) a un servidor Web Apache, ata que recibe a resposta³. O ordenador cliente ten acceso a internet e instalado un navegador cos complementos ou plugins necesarios para realizar a petición http ou https e recibir o código html. O servidor Web Apache que recibe a petición localizará a páxina solicitada; se esta ten código php, o servidor recorrerá ao motor php para executar os scripts php; se eses scripts necesitan datos almacenados nunha base de datos dun servidor MySQL, terá que realizar a petición dos datos ao servidor correspondente. O resultado final será unha páxina html que o servidor Apache enviará ao ordenador cliente.



³ Imaxe extraída do blog de Valentín Gómez onde fai unha introdución sobre as Webs dinámicas.
<https://valentingom.wordpress.com/2016/03/18/webs-dinamicas-introduccion/>

1.8 As linguaxes HTML e XHTML

Unha páxina web HTML ou XHTML é un documento de texto marcado para ser visto cun navegador web. As marcas permiten estruturar a presentación do texto, incluír imaxes ou contido multimedia, crear enlaces, engadir significado ao texto, etc.

Ao longo do tempo foron xurdindo versións de HTML e XHTML, algunhas relacionadas entre si; na seguinte táboa pódense ver as datas nas que foron publicadas e a relación entre elas. O desenvolvemento de XHTML está abandonado actualmente pero está estendido o uso de HTML5 cumprindo as normas sintácticas de XML (XHTML5) para conseguir código máis lexible.

Ano	Versión HTML	Versión XHTML	Ano
1991	HTML		
1995	HTML 2.0		
1997	HTML 3.2		
1997	HTML 4.0		
1999	HTML 4.01	XHTML 1.0	2000
		XHTML 1.1	2001
		XHTML 2.0	Abandonado
2014	HTML5		2014

Os navegadores non soportan todas as características de HTML5; pódese consultar <http://caniuse.com/> na pestana *Compare browsers*, para ver as funcionalidades soportadas por unha versión concreta dun navegador.

1.8.1 Características xerais dunha páxina HTML ou XHTML

As características xerais dunha páxina HTML ou XHTML son as seguintes:

- As marcas, elementos, tags ou etiquetas escríbense entre os signos menor e maior e soen ir en parellas (apertura e peche) rodeando unha porción de texto.
- A etiqueta de apertura ten como mínimo o nome da etiqueta e a de peche empeza polo carácter barra (/) seguido do nome da etiqueta. Algunhas etiquetas non encerran ningún texto, polo que poden aparecer só coa etiqueta de apertura e o carácter barra (/) despois do nome.
- As etiquetas de apertura poden conter un ou máis atributos que especifican algunha característica da etiqueta; existen atributos que poden utilizarse con todas as etiquetas e outros que son específicos. Os atributos poden ter un ou varios valores separados por espazos dentro de comiñas simples ou dobres.

```
<nome_etiqueta  nome_atributo = "valor_atributo"> contido </nome_etiqueta>
```

Atributo	
Etiqueta de apertura	Etiqueta de peche
Elemento	

- Non todas as etiquetas se poden aniñar, é dicir, colocar unha etiqueta dentro doutra.

- Poden conter comentarios que só se ven no código e non se ven no navegador. Estes comentarios aparecerán sempre entre as marcas de inicio e peche de comentario, <!-- e --> respectivamente. Dentro dun comentario pódese engadir calquera carácter excepto dous guións consecutivos (--) e non pode haber comentarios nas etiquetas.
- O contido das etiquetas, dos atributos e dos comentarios non poden levar caracteres que poidan provocar unha interpretación errónea do código por parte do navegador. Por exemplo: o contido dunha etiqueta non pode levar o símbolo menor ou maior e o contido dun atributo non pode levar o símbolo das comiñas dobres ou simples. Para solucionar estes casos empréganse entidades referenciadas que se verán máis adiante.
- Existen unhas marcas especiais como as de procesamento que empezan por ? ou as que empezan por !, como por exemplo, a que define o tipo de documento web (!DOCTYPE).

Exemplo de documento HTML básico:

```
<!DOCTYPE html>
<!--
Primeiro exemplo de html5
-->
<html>
  <head>
    <title>Exemplo de html5</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <meta name="viewport" content="width=device-width"/>
  </head>
  <body>
    <h2>Primeiro exemplo de html 5</h2>
    <hr/>
    <p>Estrutura xerada por NetBeans sen utilizar plantillas e modificada lixeiramente para que cumpra as normas sintácticas de XHTML</p>
  </body>
</html>
```

O aspecto do anterior documento dende o navegador Chrome é o seguinte:



1.8.2 Diferenzas entre HTML e XHTML

A linguaxe XHTML = HTML + XML, isto significa:

- Ter que cumprir as normas sintácticas de XML, polo que o código será máis fácil de manter ao ser máis limpo e claro que o de HTML, pero máis estrito. Un documento XHTML ten que estar ben formado e un HTML pode non estarlo. O tipo MIME⁴ utilizado no envío do documento obriga a que un navegador non mostre o documento XHTML (text/xml ou application/xhtml+xml) se ten erros sintácticos, pero mostrará un HTML (text/html) aínda que os teña. Se o documento non ten declaración do tipo MIME, dedúcese pola extensión do arquivo.

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
```

- Podería ser un documento XML e aplicarse as tecnoloxías baseadas en XML, como por exemplo, transformacións XSLT.

En concreto, un documento XHTML ten que cumprir as seguintes normas:

- Todas as etiquetas aníñanse en forma de árbore, existindo un único elemento raíz denominado html, do que descenden todos os demais.
- Os nomes de etiquetas e atributos escríbense en minúscula.
- Toda etiqueta de apertura ten a correspondente etiqueta de peche agás no caso das etiquetas baleiras que poden representarse coa forma abreviada.

```
<meta charset="utf-8" />
<meta charset="utf-8"></meta>
```

- As etiquetas aniñadas deben pecharse de forma estrutural, é dicir, en orden inversa a súa apertura. Dito doutra maneira, non poden cruzarse, é dicir, a etiqueta de peche da máis interior ten que aparecer antes da etiqueta de peche da etiqueta que a contén.
- Un atributo, se existe, sempre ten que ter un valor asignado que debe ir entre comiñas simples ou dobres e non pode aparecer máis dunha vez nunha etiqueta, aínda que se poden especificar en calquera orde.

```
<link rel="stylesheet" type="text/css" href="estilo.css" />
<link type="text/css" href="estilo.css" rel="stylesheet" />
```

1.8.3 Estrutura dunha páxina HTML ou XHTML

A estrutura dunha páxina HTML ou XHTML, consta das seguintes partes:

- Declaración de tipo de documento mediante a etiqueta de <!DOCTYPE html>. En XHTML esta etiqueta leva tamén a declaración do DTD que contén as definicións do vocabulario e sintaxe que será utilizado polas ferramentas que validen o código, como por exemplo: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- Elemento raíz <html>. En XHTML tamén se debe de especificar o espazo de nomes asociado, como por exemplo: <html xmlns="http://www.w3.org/1999/xhtml">

⁴ MIME (Multipurpose Internet Mail Extensions) é un conxunto de especificacións utilizadas para permitir a inserción de documentos nunha mensaxe a través de internet ou para clasificar documentos transferidos utilizando o protocolo http. O servidor envía o tipo MIME do arquivo na resposta a unha petición http dun explorador, para que o explorador saiba como mostrar o documento. Algúns tipos MIME son: text/plain, text/xml, application/xml, text/html, text/css. Ver máis en <http://es.ccm.net/contents/118-mime-extensiones-multiproposito-de-correo-internet>

- Cabeceira <head> que contén información que non se mostra pero importante para as aplicacións que utilizan ou manexan a páxina, como por exemplo:
 - Etiqueta <meta> que contén información dos encabezados de resposta http e que son utilizados, por exemplo, polos robots de busca ou programas que analicen a páxina. Máis información en <https://www.w3.org/TR/html5/document-metadata.html>. Por exemplo:

O tipo MIME e o sistema de codificación:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
```

Que a ventá de visualización se adapte ao ancho de calquera dispositivo:

```
<meta name="viewport" content="width=device-width"/>
```

Palabras clave, nome de autor, ou linguaxe utilizada.

```
<meta name="keywords" lang="es-es" content="vacaciones, Galicia, playa, sol"/>
```

```
<meta name="Author" content="Profesora" lang="es-es"/>
```

- Etiqueta <title>, obrigatoria, que contén o texto que se mostrará na pestana do navegador.
- Etiqueta <link> que permite enlazar con outro recurso externo á páxina, como por exemplo stylesheet para facer referencia a unha folla de estilos CSS externa, indicando mediante o atributo href a ruta para localizar o arquivo e mediante o atributo type o tipo MIME do documento enlazado.
- Corpo <body> que contén a información que se verá no navegador marcada coas etiquetas propias de HTML.

Na presente unidade didáctica, utilizarase HTML 5 pero velando polo cumprimento das normas sintácticas de XHTML, co obxectivo de facer máis lexible e comprensible o código.

1.8.4 Sistema de codificación

O sistema de codificación empregado nun documento é o xogo de caracteres empregado para representar internamente os símbolos que contén. Pódense utilizar os seguintes sistemas de codificación:

- ASCII (*American Standard Code for Information Interchange*). Emprega os primeiros 7 bits dun byte (entre 0 e 127). Definía como se debían representar 128 caracteres entre números, letras (fala inglesa soamente), signos de puntuación e 32 caracteres de control. O oitavo bit usábase para controlar erros na transmisión (control de paridade).
- ASCII estendido que utiliza os 8 bits para representar caracteres, é dicir, 128 máis có ASCII. Existen moitas páxinas de código para representalos entre as que se destacan:
 - CP437. É o xogo de caracteres orixinal do IBM PC (o que viña na ROM dos adaptadores de vídeo). Inclúe vocais minúsculas con til, eñes e apertura de exclamacións e interrogacións. Non inclúe, por exemplo, vocais maiúsculas con til, salvo a É.
 - CP850. É unha adaptación do CP437 que se empregou en MS-DOS en países de Europa occidental. Inclúe, entre outros, todas as vocais maiúsculas con til. Unha pequena modificación desta páxina de código incluíndo o símbolo do euro é a CP858.
 - Windows Code Pages. Son xogos de caracteres empregados polos sistemas operativos Windows na década dos 80. Existen varios, cada un pensado para ser empregado

nun alfabeto determinado. En Europa occidental e países de lingua inglesa empregouse o xogo de caracteres Windows-1252.

- ISO-8859 moi empregados en sistemas Linux. O máis usado para o español é o ISO-8859-1 ou latin-1 e o ISO-8859-15 que é unha pequena variante que inclúe o símbolo do euro (€). Segue a usarse amplamente hoxe en día.
- Unicode que naceu para evitar a complicación anterior, consiste nun repertorio de máis de 100.000 caracteres que abranguen a maior parte de alfabetos empregados na actualidade. Existen diversas formas de especificar a codificación de cada carácter. As máis utilizadas son:
 - UTF-8. Codificación orientada a byte con símbolos representados con lonxitude variable de un a catro bytes. Os 128 caracteres do ASCII orixinal codifícanse nun byte que sempre comeza por cero. Para representar outro carácter de ampla utilización (vocais con til, ñe, alfabeto grego, árabe, etc.), empréganse 2 bytes. Os demais caracteres usarán 3 ou 4 bytes. É moi utilizado pois posibilita a codificación de calquera carácter Unicode reducindo considerablemente o espazo ocupado polos documentos se o comparamos coas súas alternativas.
 - UTF-16. Emprega tamén codificación de lonxitude variable. Neste caso, divide os caracteres en dous grupos: os de máis ampla utilización, que codifica en 2 bytes, e o resto, que ocupan 4 bytes cada un.
 - UTF-32. é una codificación de lonxitude fixa que utiliza 4 bytes por carácter polo que en xeral, os documentos con este sistema de codificación ocuparán máis espazo que se utilizaran UTF-8.

Precaucións a ter en conta cos documentos con relación aos sistema de codificación:

- É moi importante especificar no documento o xogo de caracteres que se utilizou ao gravalo. De non ser así, a aplicación que o vaia a utilizar traducirá o código segundo o sistemas de codificación que teña por defecto e o resultado pode ser ilexible. Os navegadores web, por exemplo, permiten consultar e cambiar o xogo de caracteres utilizado para representar cada documento.
- Débese gardar o documento co mesmo sistema de codificación que o indicado no documento.
- Aínda cando se estea representando un documento no mesmo xogo de caracteres no que foi creado, isto non garante que a representación en pantalla sexa correcta. Necesítase ademais que o tipo de letra que esteamos usando conteña tódolos caracteres que utiliza o documento. A gran maioría de tipos de letra conteñen soamente os caracteres máis habituais que utilizamos ao escribir documentos.

1.8.5 Referencias e entidades

Os caracteres especiais como &, <, dobres comiñas (“) e apóstrofe ('), poden forma parte do contido dun elemento ou como valor dun atributo, utilizando unha das seguintes formas:

- Empregando unha referencia a un carácter Unicode. As referencias sempre empezan por &# (ou por &#x se empregamos hexadecimal) e rematan en punto e coma (;). Por

exemplo, os caracteres & e < poderían substituírse respectivamente polas referencias & e < (en hexadecimal serían & e <).

- Empregando entidades. As entidades empezan por &, van seguidas do nome da entidade e terminan en punto e coma (;). Poden crearse novas entidades como se explicará en unidades didácticas posteriores e existen entidades predefinidas como por exemplo:

Carácter	Entidade HTML	Carácter	Entidade HTML
á	á	Á	Á
é	é	É	É
í	í	Í	Í
ó	ó	Ó	Ó
ú	ú	Ú	Ú
ü	ü	Ü	Ü
ñ	ñ	Ñ	Ñ
<	<	&	&
>	>	"	"e;
Espacio	 	'	'

Por exemplo, o seguinte código provocaría 2 erros de sintaxe, se é que se tomaron as precaucións citadas anteriormente (páxina 13) con relación aos sistema de codificación:

```
<p>O operador de concatenación é &</p>
<p>Os operadores de comparación son < e >.</p>
```

A versión sen erros sería:

```
<p>O operador de concatenación é &amp;</p>
<p>Os operadores de comparación son &lt; e &gt;.</p>
```

Sería aconsellable substituír o símbolo > pola correspondente entidade para non confundila coa escritura dunha etiqueta.

```
<p>O operador de concatenación é &amp;</p>
<p>Os operadores de comparación son &lt; e &gt;>.</p>
```

Este anaco de código veríase no navegador Chrome da seguinte maneira:

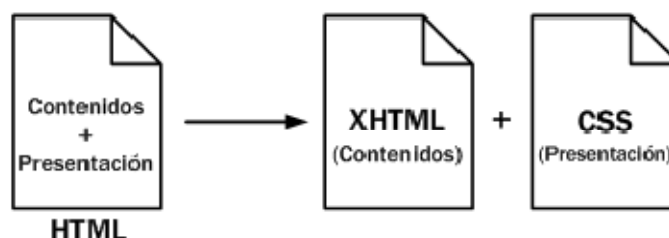


1.9 A Linguaxe CSS

CSS (Cascade Style Sheets) é unha linguaxe de follas de estilos creada para controlar o aspecto ou presentación dos documentos web escritos HTML e XHTML. Foi proposto por vez primeira por HåkonWiumLie en outubro de 1994, cando traballaba con TimBerners-Lee (o creador de HTML) no CERN. Naquel tempo, tamén se propuxeron outras linguaxes de follas de estilos, pero o W3C (World Wide Web Consortium), tras arduas discu-

sións a través das súas listas de correo, publicou a primeira recomendación de CSS (CSS nivel 1) en 1996.

A utilización da linguaxe CSS permite separar o contido do aspecto. A información contida na páxina, márcase mediante a linguaxe HTML/XHTML, designando a función de cada elemento dentro da páxina. O aspecto que vai ter cada elemento no navegador, defínese mediante a linguaxe CSS⁵.



CSS non ten versións no sentido tradicional, pero ten niveis. Cada nivel se constrúe sobre o anterior, cambiando definicións e agregando características. CSS nivel 1 considérase obsoleto. CSS nivel 2 pasou a ser unha recomendación antes de finalizar a etapa de candidatura, polo que apareceron moitos erros que en lugar de arranxar no nivel 2, publicáronse revisións deste nivel. CSS nivel 3 desenvólvese por módulos, agregando funcionalidades e refinando definicións. Na seguinte táboa podemos ver as distintas versións de CSS e os anos datas de publicación das recomendacións.

Ano	Niveis CSS
1996	CSS1
1998	CSS 2
2011	CSS 2.1
2011	CSS3 As recomendacións vanse publicando por módulos. Ver máis en https://www.w3.org/TR/css-2017/ .

1.10 Ferramentas Web

Existe gran variedade de ferramentas que axudan no desenvolvemento Web. Algunhas delas son:

- Para editar documentos html ou css, pódese utilizar calquera procesador ou editor de textos que permita crear e modificar arquivos de texto plano, pero é máis práctico utilizar contornos de desenvolvemento que ademais de permitir editar, axuden ó programador na escritura das linguaxes. Recoméndase utilizar contornos de desenvolvemento como NetBeans ou Eclipse que permitan realizar operacións globais de programación, como por exemplo, programación en varias linguaxes, validación de código, axuda en liña, execución ou visualización do resultado, formateo de código, busca e substitución, refactorización,...
- Un framework para aplicacións Web é un conxunto estandarizado de conceptos, prácticas, criterios e software que asiste ao desenvolvemento de sitios Web dinámicos, aplicacións Web e servizos Web. Este tipo de ferramentas alivia a carga de traballo asocia-

⁵ Imaxe extraída do material "Introducción a XHTML" de <http://www.librosweb.es>

da a oporacións comúns e repetitivas no desenvolvemento Web. Exemplos de framework: Skeleton, Concise, Bootstrap, Gumby.

- Existen sitios web que subministran recursos e plantillas CSS, creadas e probadas, que poden ser modificadas. Existen sitios Web e aplicacións que dispoñen dunha galería Web e elementos de deseño que facilitan o proceso de creación, como por exemplo, permiten elixir plantillas e xeran automaticamente o código HTML e CSS, permiten ter unha vista gráfica e modificar sobre ela e a edición do código e modificar sobre el. Exemplos: wix.com, Open source Web Design, Magix Web Designer ou Dreamweaver.
- Os clientes web ou navegadores, coma por exemplo Chrome, Mozilla ou IE, son capaces de interpretar as marcas HTML ou XHTML e presentar o documento tal e como o deseñou o autor, pero non mostran as marcas agás que se indique explicitamente. Os navegadores poden mostrar as marcas nunha ventá ou pestana aparte, normalmente con Ctrl+u ou premendo clic dereito sobre a páxina e elixindo a opción de Visualizar código fonte.
- O servizo de validación de W3C <https://validator.w3.org/> permite comprobar se un documento está ben, elixindo unha das seguintes opcións:
 - Tecleando o URI (*Validate by URI*).
 - Subindo o arquivo ó servidor (*Validate by File Upload*).
 - Tecleando directamente o código (*Validate by direct input*).

Pinchando en *More Options*, pódese seleccionar o tipo de documento que se quere validar ou deixar a opción por defecto que o detecte automaticamente.

W3 The W3C Markup Validator

Es seguro | https://validator.w3.org/#validate_by_input+with_options

Validate by URI Validate by File Upload **Validate by Direct Input**

Validate by direct input

Enter the Markup to validate:

```
<!DOCTYPE html>
<!--
Primeiro exemplo de html5
-->
<html>
<head>
<title>Exemplo de html5</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<meta name="keywords" lang="es-es" content="vacaciones, Galicia, playa, sol">
<meta name="Author" content="Profesora" lang="es"/>
<meta name="viewport" content="width=device-width"/>
</head>
```

▼ More Options

☒ **Validate Full Document**

Use Doctype: HTML5 (experimental) ☐ Only if Doctype is missing

☐ **Validate HTML fragment**


Use Doctype: ☒ HTML 4.01 ☐ XHTML 1.0


☒ List Messages Sequentially ☐ Group Error Messages by Type

☐ Show Source ☐ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☐ Verbose Output

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).

 The W3C validators rely on community support for hosting and development. [Donate](#) and help us build better tools for a better web.

0 

Para validar, débese premer no botón *Check*. Marcaranse e explicaranse os erros, se é que existen, ou sairá a mensaxe de que non hai erros graves nin leves.

Showing results for conti X

Es seguro | <https://validator.w3.org/nu/#textarea>

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by **text input**

```
<title>Exemplo de html5</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<meta name="keywords" lang="es-es" content="vacaciones, Galicia, playa, sol">
<meta name="Author" content="Profesora" lang="es"/>
<meta name="viewport" content="width=device-width"/>
</head>
<body>
<h2>Primeiro exemplo de html 5</h2>
<hr/>
<p1>Estrutura xerada por NetBeans sen utilizar plantillas e modificada lixeiramente
para que cumpra as normas sintácticas de XHTML</p>
<p>O operador de concatenación é &amp;</p>
<p>Os operadores de comparación son &lt; e &gt;</p>
</body>
</html>
```

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

- Error** Element `p1` not allowed as child of element `body` in this context. (Suppressing further errors from this subtree.)
[From line 16, column 9, to line 16, column 12](#)

```
> <p1>Estrut
```


Content model for element `body`:
[Flow content](#)

2. Documentos de apoio ou referencia

- FALCÓN DOCAMPO, Amalia; FERNÁNDEZ NOCELO, Laura; REY LÓPEZ, Marta. *Unidade didáctica 1(Aplicacións web e linguaxes de marcas: HTML e CSS), actividade 1(Introdución ás linguaxes de marcas A linguaxe XML) do módulo Linguaxes de marcas e sistemas de xestión da información*. Xunta de Galicia, Consellería de Cultura, Educación e Ordenación Universitaria. 2014.
- FALCÓN DOCAMPO, Amalia; FERNÁNDEZ NOCELO, Laura; REY LÓPEZ, Marta. *Unidade didáctica 1(Aplicacións web e linguaxes de marcas: HTML e CSS), actividade 2(Documentos web con linguaxes de marcas: HTML) do módulo Linguaxes de marcas e sistemas de xestión da información*. Xunta de Galicia, Consellería de Cultura, Educación e Ordenación Universitaria. 2014.
- FALCÓN DOCAMPO, Amalia; FERNÁNDEZ NOCELO, Laura; REY LÓPEZ, Marta. *Unidade didáctica 1(Aplicacións web e linguaxes de marcas: HTML e CSS), actividade 3(Follas de estilo en documentos web: CSS) do módulo Linguaxes de marcas e sistemas de xestión da información*. Xunta de Galicia, Consellería de Cultura, Educación e Ordenación Universitaria. 2014.
- FALCÓN DOCAMPO, Amalia; FERNÁNDEZ NOCELO, Laura; REY LÓPEZ, Marta. *Unidade didáctica 2(Outras linguaxes da web: XHTML, HTML5 e CSS3), actividade 1(Outras linguaxes da web) do módulo Linguaxes de marcas e sistemas de xestión da información*. Xunta de Galicia, Consellería de Cultura, Educación e Ordenación Universitaria. 2014.
- SINTES MARCOS, BARTOLOMÉ. *XML: Páginas web HTML y hojas de estilo CSS. Lenguaje de marcas extensible*. 2017.