

Tema 1: Introducción y reconocimiento de las características de los lenguajes de marcas.



Bloque 1: Desarrollo web en cliente: HTML y CSS.

Módulo: Lenguajes de marcas y sistemas de gestión de la información.

IES San Vicente
Curso 2021-2022

Índice de contenidos

1	Información digital.....	3
1.1	Ordenadores y almacenamiento de la información.....	3
1.2	Sistemas de codificación.....	3
1.3	Incompatibilidades.....	5
2	Lenguajes de marcas.....	5
2.1	Orígenes de los lenguajes de marcas.....	6
2.1.1	TeX y LaTeX.....	6
2.1.2	RTF.....	6
2.1.3	SGML.....	6
2.1.4	HTML.....	7
2.1.5	XML.....	7
2.1.6	JSON.....	7
2.2	Características de un lenguaje de marcas.....	8
2.3	Tipos de lenguajes de marcas.....	8
2.4	Organizaciones desarrolladoras de lenguajes de marcas.....	8
3	Entorno de desarrollo o IDE.....	9
3.1	Atom.....	9
3.2	Visual Studio Code.....	9
3.2.1	Entorno de trabajo.....	9
3.2.2	Atajos de teclado.....	10
3.2.3	Secciones del IDE.....	11
3.2.4	Extensiones.....	12
3.2.5	Cambiar el tema del editor.....	13
3.3	Otras alternativas.....	13
4	¿Qué es una aplicación web?.....	13
4.1	¿Qué es “la web”?.....	13
4.2	Elementos de una aplicación web.....	13
4.3	Funcionamiento de una aplicación web.....	13
4.4	Dominios.....	14
4.5	Protocolo HTTP.....	14
4.6	Lenguajes.....	14

1 Información digital

1.1 Ordenadores y almacenamiento de la información

Un ordenador es toda máquina capaz de representar información mediante 1's y 0's, es decir, utiliza el sistema binario para interpretar y almacenar la información, por tanto, se requiere que esa información con la que trabajamos sea codificada para que sea comprensible por el ordenador.

La información con la que trabaja el ser humano es sistema decimal para representar números, y otras formas de representación mucho más complejas para representar el texto, las imágenes, la música, etc.

1.2 Sistemas de codificación

Todo dato que no sea texto es considerado como dato binario, por ejemplo un vídeo, un programa, etc. La forma que hay de codificar a binario estos datos es muy variable. Por ejemplo, en las imágenes cada uno de los puntos (píxeles) es codificado de forma independiente.

El texto es la forma más “humana” de representar la información. Antes de la llegada del ordenador, la información se transmitía mediante documentos o libros. Pero, ¿Cómo representamos el texto en forma digital? Para ello, lo que se hace es codificar cada uno de los caracteres por los que se compone y se codifican. A lo largo de la historia han surgido numerosos sistemas de codificación, siendo los más conocidos el código ASCII y actualmente Unicode.

El código **ASCII** (American Standard Code for Information Interchange) surgió a finales de los 70, con el objetivo de tener una codificación común para codificar el texto en binario. Para ello se utilizaban 7 bits, permitiendo obtener 128 símbolos o caracteres distintos. En esta primera versión se podía codificar las letras del alfabeto (inglés) tanto mayúsculas como minúsculas, y los números del 0 al 9, así como algunos símbolos adicionales como signos de puntuación, etc. Sin embargo, esto no era suficiente puesto que no era posible codificar todos los caracteres utilizando sólo 7 bits (no había ñ, ni letras acentuadas, etc.). Por ello, se pasó a utilizar 8 bits dando lugar a los llamados códigos ASCII extendidos.

Caracteres ASCII de control				Caracteres ASCII imprimibles				ASCII extendido (Página de código 437)									
00	NULL	(carácter nulo)		32	espacio	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(inicio encabezado)		33	!	65	A	97	a	129	ú	161	í	193	ł	225	ß
02	STX	(inicio texto)		34	"	66	B	98	b	130	ê	162	ó	194	Ł	226	Ô
03	ETX	(fin de texto)		35	#	67	C	99	c	131	â	163	ú	195	ł	227	Õ
04	EOT	(fin transmisión)		36	\$	68	D	100	d	132	ä	164	ñ	196	Ł	228	ö
05	ENQ	(consulta)		37	%	69	E	101	e	133	å	165	Ñ	197	ł	229	Ö
06	ACK	(reconocimiento)		38	&	70	F	102	f	134	ä	166	ª	198	Ł	230	µ
07	BEL	(timbre)		39	'	71	G	103	g	135	ç	167	º	199	Ł	231	þ
08	BS	(retroceso)		40	(72	H	104	h	136	ê	168	¿	200	Ł	232	ÿ
09	HT	(tab horizontal)		41)	73	I	105	i	137	ë	169	®	201	Ł	233	Ù
10	LF	(nueva línea)		42	*	74	J	106	j	138	è	170	¬	202	Ł	234	Ú
11	VT	(tab vertical)		43	+	75	K	107	k	139	ï	171	½	203	Ł	235	Û
12	FF	(nueva página)		44	,	76	L	108	l	140	î	172	¼	204	Ł	236	Ü
13	CR	(retorno de carro)		45	-	77	M	109	m	141	ï	173	⅓	205	Ł	237	Ý
14	SO	(desplaza afuera)		46	.	78	N	110	n	142	Ä	174	«	206	Ł	238	–
15	SI	(desplaza adentro)		47	/	79	O	111	o	143	Å	175	»	207	Ł	239	—
16	DLE	(esc.vínculo datos)		48	0	80	P	112	p	144	É	176	•	208	Ł	240	≡
17	DC1	(control disp. 1)		49	1	81	Q	113	q	145	æ	177	◊	209	Ł	241	±
18	DC2	(control disp. 2)		50	2	82	R	114	r	146	Æ	178	◊	210	Ł	242	±
19	DC3	(control disp. 3)		51	3	83	S	115	s	147	ø	179	◊	211	Ł	243	¾
20	DC4	(control disp. 4)		52	4	84	T	116	t	148	ö	180	◊	212	Ł	244	¶
21	NAK	(conf. negativa)		53	5	85	U	117	u	149	õ	181	◊	213	Ł	245	§
22	SYN	(inactividad sinc)		54	6	86	V	118	v	150	ü	182	◊	214	Ł	246	÷
23	ETB	(fin bloque trans)		55	7	87	W	119	w	151	û	183	◊	215	Ł	247	÷
24	CAN	(cancelar)		56	8	88	X	120	x	152	ÿ	184	◊	216	Ł	248	°
25	EM	(fin del medio)		57	9	89	Y	121	y	153	Ö	185	◊	217	Ł	249	°
26	SUB	(sustitución)		58	:	90	Z	122	z	154	Ü	186	◊	218	Ł	250	°
27	ESC	(escape)		59	;	91	[123	{	155	ø	187	◊	219	Ł	251	°
28	FS	(sep. archivos)		60	<	92	\	124		156	£	188	◊	220	Ł	252	°
29	GS	(sep. grupos)		61	=	93]	125	}	157	Ø	189	◊	221	Ł	253	°
30	RS	(sep. registros)		62	>	94	^	126	~	158	×	190	¥	222	Ł	254	°
31	US	(sep. unidades)		63	?	95	_			159	f	191	γ	223	Ł	255	nbsp
127	DEL	(suprimir)															

de uso frecuente (idioma español)		vocales con acento (acento agudo español)		vocales con diéresis		símbolos matemáticos		símbolos comerciales		comillas, llaves paréntesis	
ñ	alt + 164	á	alt + 160	ä	alt + 132	½	alt + 171	\$	alt + 36	"	alt + 34
Ñ	alt + 165	é	alt + 130	ë	alt + 137	¼	alt + 172	£	alt + 156	'	alt + 39
@	alt + 64	í	alt + 161	ï	alt + 139	⅓	alt + 243	¥	alt + 190	(alt + 40
¿	alt + 168	ó	alt + 162	ö	alt + 148	²	alt + 251	¢	alt + 189)	alt + 41
?	alt + 63	ú	alt + 163	ü	alt + 129	³	alt + 252	¤	alt + 207	[alt + 91
¡	alt + 173	Á	alt + 181	Ä	alt + 142	⁴	alt + 253	©	alt + 169]	alt + 93
!	alt + 33	É	alt + 144	Ê	alt + 211	f	alt + 159	®	alt + 184	{	alt + 123
:	alt + 58	Í	alt + 214	Ë	alt + 216	±	alt + 241	™	alt + 166	}	alt + 125
/	alt + 47	Ó	alt + 224	Ö	alt + 153	×	alt + 158	°	alt + 167	«	alt + 174
\	alt + 92	Ú	alt + 233	Ü	alt + 154	÷	alt + 246	°	alt + 248	»	alt + 175

Sin embargo, utilizando 8 bits para representar todos los caracteres alfabetos del planeta sigue siendo insuficiente, por lo que se decidió optar a que cada zona usara su propia tabla ASCII extendida teniendo en común los 128 caracteres. Por ejemplo, ISO-8859-1 es el sistema ASCII extendido para Europa Occidental donde vienen incluidos símbolos como la letra ñ.

El problema que tenemos con esta codificación, es que a la hora de editar un documento se debe especificar el sistema de codificación que estamos utilizando para escribirlo y así que el programa que lo vaya a utilizar sepa cómo interpretarlo. El sistema **Unicode** ha conseguido resolver esto, incluyendo los caracteres de prácticamente todas las lenguas del planeta a cambio de que cada carácter ocupe más de 1byte (8 bits). Los 128 primeros elementos siguen correspondiendo al sistema ASCII original para mantener la compatibilidad y los 128 siguientes corresponden al sistema europeo **ISO-8859-1**. A partir de ahí, se le añade un segundo byte para lenguas cirílicas, griegas, etc. O un tercer byte para chino o japoneses, o un cuarto byte para símbolos adicionales (matemáticos, lenguas muertas, etc.). Por lo tanto, la codificación de Unicode como vemos puede llegar a ocupar hasta 4bytes dependiendo del alfabeto que queramos utilizar. Las distintas versiones del sistema **Unicode UTF** (Unicode Transformation Format) son:

- **UTF-8:** Es la más utilizada desde hace ya unos años. Utiliza para cada carácter de 1 a 4 caracteres, de forma que:
 - Utilizan 1byte los que pertenecen al código ASCII original
 - Utilizan 2bytes los pertenecientes a lenguas latinas, cirílicas, griegas, etc.
 - Utilizan 3bytes para chino o japonés
 - Utilizan 4bytes para símbolos especiales
- **UTF-16:** Utilizan 2 bytes para los dos primeros grupos anteriores, 3 para chino o japonés y 4 para el resto
- **UTF-32:** Utiliza 4bytes para todos los sistemas. Es el más sencillo de gestionar aunque ocupa más espacio, por lo que a día de hoy apenas se utiliza.

1.3 Incompatibilidades

En un archivo binario, la información se almacena como una secuencia de bits (unos y ceros) agrupados en grupos de 8, llamados bytes. Esta es la forma en la que el ordenador puede entender la información que almacena para a posteriori poderla procesar, ya sea visualizando una imagen, etc. Sin embargo, si el programa que utilizamos para abrir el archivo es diferente al que utilizamos para generar ese archivo puede producirse un error de incompatibilidad si se trata de un archivo binario, es decir, los archivos binarios son dependientes del software que lo genera y del sistema operativo utilizado.

Por otro lado, tenemos los archivos de texto. Un archivo de texto puede ser un .txt que hayamos generado en nuestro ordenador, y se va a poder abrir independientemente del sistema operativo que utilicemos (MacOS, Linux o Windows). El problema con estos archivos se encuentra si queremos darle formato al texto, añadirle color, utilizar un tipo de letra, etc. ya que a la par del archivo de texto se tenía que guardar en formato binario el formato de esta información. Esto traía los llamados problemas de incompatibilidad, y es ahí donde cobran importancia los lenguajes de marcas.

2 Lenguajes de marcas

Cuando queremos almacenar algún tipo de información como un documento de trabajo, un manual podemos hacerlo de distintas formas. Podemos emplear un editor de texto avanzado como Microsoft Office Word o LibreOffice Writer, aunque en este caso tendríamos el inconveniente de que no podríamos consultar el documento sin tener el software instalado puesto que al intentar abrirlo con un software diferente puede que ese documento no lo veamos correctamente por tema de incompatibilidad.

Los lenguajes de marcas nos permiten poder añadir anotaciones en un documento de texto que hagan referencia al formato del documento (tipo y tamaño de letra, color, etc.) y por otra parte, tener la estructura de la información del documento añadiendo unas marcas o etiquetas que indican qué tipo de información estamos almacenando.

2.1 Orígenes de los lenguajes de marcas

Charles Goldfarb es considerado el padre de los lenguajes de marcas, propuso la idea de que los documentos de texto indicasen el formato del mismo.

2.1.1 TeX y LaTeX

En los años 70, Donald Knuth creó TeX permitiendo que documentos científicos se vieran igual en cualquier ordenador utilizando la misma tipografía y permitiendo representar fórmulas matemáticas, expresiones, etc. Para ello, a la información se le añadía unas ciertas marcas que las complementaba e informaba al programa encargado de abrirlo de cómo se tenía que mostrar. Un ejemplo de código LaTeX:

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\Ejemplo}
\begin{document}
Este es el texto ejemplo de \LaTeX{}
Con datos en \emph{cursiva} o \textbf{negrita}.
Ejemplo de f'ormula
\begin{align}
E &= mc^2
\end{align}
\end{document}
```

Como podemos ver en el ejemplo anterior, se especifican cosas como el tamaño de la letra, si es cursiva o negrita, acentos, etc. Y su resultado sería:

Este es el texto ejemplo de L^AT_EX
Con datos en *cursiva* o **negrita**. Ejemplo de fórmula

2.1.2 RTF

$$E = mc^2 \quad (1)$$

Es el acrónimo de Rich Text Format, un lenguaje creado por Microsoft en 1987 para producir documentos de texto con formato. Actualmente, estos documentos son muy utilizados en Windows como formato de intercambio entre distintos procesadores por su potencia. El procesador de texto Word Pad incorporado por Windows lo utiliza como formato nativo.

2.1.3 SGML

Fue un lenguaje de marcado definido por la ISO como el estándar mundial de documentos de texto con etiquetas. Este lenguaje es considerado como el padre del lenguaje XML y la base sobre la que se sostiene el lenguaje HTML.

En SGML las etiquetas que contienen indicaciones para el texto se colocan entre los símbolos < y >. Las etiquetas se cierran con el signo /. Es decir, las reglas fundamentales de los lenguajes de etiquetado actuales ya habían sido definidas por SGML. En realidad, como pasa con XML, no es un lenguaje con unas etiquetas concretas, sino que sirve para definir lenguajes de etiquetas, o mejor dicho, sirve para definir formatos de documentos de texto con marcas. De entre todos los lenguajes de marcas definidos mediante SGML, HTML es el más popular.

2.1.4 HTML

Tim Bernes Lee utilizó SGML para definir un nuevo lenguaje de etiquetas que llamó Hypertext Markup Language para intercambiar información en entornos académicos a través de internet. El consorcio World Wide Web (W3C) regula las recomendaciones y versiones normalizadas de este lenguaje, aceptándolo como norma ISO desde el 2000.

Las etiquetas de este lenguaje, están pensadas para la organización lógica del contenido (título, párrafos, tablas, etc.) y no tanto para su presentación, por eso, el W3C tuvo que idear las denominadas hojas de estilo (Style Sheets).

2.1.5 XML

Se trata de un lenguaje creado para mejorar el propio SGML y con definir lenguajes de marcado con sintaxis más estricta pero más entendibles. XML ha sido uno de los lenguajes de marcado más utilizados para importar y exportar información.

2.1.6 JSON

JavaScript Object Notation (JSON), se trata de una notación de datos procedente del lenguaje JavaScript. En el año 2002, se le daba soporte desde muchos de los navegadores de modo que su fama ha hecho que se haya convertido en una notación independiente de JavaScript compitiendo claramente con XML hasta el punto de llegar a ser el más utilizado hoy en día para intercambiar información.

JSON no es considerado un lenguaje de marcas al no haber diferencia en el texto a través de etiquetas, sino que se basa en que el texto se divide en dato y metadato, por tanto, es conocido como una notación. Por ejemplo:

```
[
  {
    "nombre": "Carlos Calatrava",
    "dirección": "C/ Lentejas, 14",
    "teléfono": "666777888"
  },
  {
    "nombre": "Raúl Gutiérrez",
    "dirección": "C/ Bocadillo, 12",
    "teléfono": "616525434"
  },
  ...
]
```

2.2 Características de un lenguaje de marcas

Las características generales que debe cumplir un lenguaje de marcas que definamos o utilicemos son:

- Deben definir la información en archivos de texto plano, de forma que sean fáciles de exportar o utilizar entre las diferentes aplicaciones y sistemas operativos
- Deben mostrar información de forma compacta, intercalando las etiquetas o palabras del vocabulario con la información a almacenar.
- Deben coexistir con otros lenguajes en un documento. Por ejemplo, dentro de un archivo con HTML podemos encontrar código JavaScript o PHP entre otros.
- No son lenguajes de programación ya que carecen de la estructura básica de estos lenguajes (variables, bucles, funciones, etc.) Son lenguajes para etiquetar y estructurar la información de documentos sin más.

2.3 Tipos de lenguajes de marcas

- **Lenguajes de presentación:** definen el formato con el que se ve la información, sin detallar su estructura. Por ejemplo RTF.
- **Lenguajes descriptivos:** definen las partes del documento sin especificar cómo se va a mostrar, ni en qué orden. Por ejemplo XML o JSON
- **Lenguajes híbridos:** definen la estructura de un documento y su presentación. Por ejemplo HTML

En su origen, HTML era un lenguaje descriptivo, pero a lo largo de su evolución se fueron introduciendo etiquetas que indicaban cómo debía visualizarse cada elemento pasando a ser un lenguaje híbrido.

2.4 Organizaciones desarrolladoras de lenguajes de marcas

- **ISO** (Organización Internacional para la Estandarización) cuya sede se encuentra en Ginebra (Suiza). Se creó en 1947, se caracteriza porque sus normas son voluntarias y no obligatorias al no ser gubernamental. Después del éxito del GLM y después de un largo proceso, en 1986 publicó el Standard Generalized Markup Language (SGML) con rango de estándar internacional con el código ISO 8879
- **World Wide Web Consortium (W3C):** Creada en 1994 por Tim Berners-Lee. Su función principal es tutelar el crecimiento y organización de la web. Normalizó el lenguaje HTML, y creó el XML en 1998.

3 Entorno de desarrollo o IDE

A la hora de trabajar con uno o varios lenguajes de marcas, podemos utilizar de ciertos editores o herramientas software que nos faciliten el proceso. Vamos a ver algunos de ellos, centrándonos en el que utilizaremos a lo largo del curso.

3.1 Atom

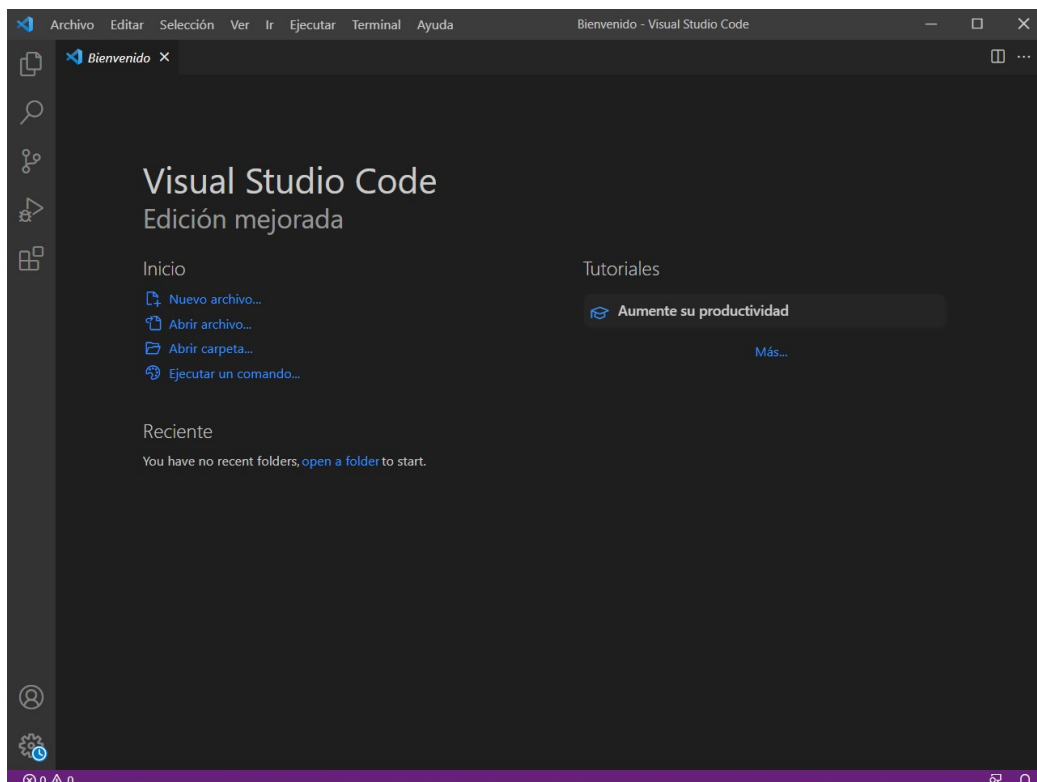
Editor de código abierto para macOS, Linux y Windows. Permite editar archivos de texto en diferentes lenguajes como C#, PHP, HTML, Sass, SQL, XML, JavaScript entre muchos otros. Podemos descargarlo desde su [página oficial](#).

3.2 Visual Studio Code

Es un editor de texto desarrollado por Microsoft para Windows, Linux y macOS. Es compatible con varios lenguajes de programación y un conjunto de características (plugins o extensiones) que facilitan mucho la programación. A lo largo de este curso, estaremos programando con este editor.

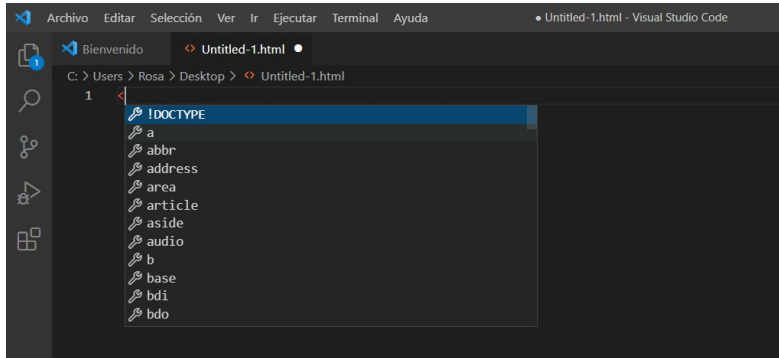
Podemos descargarlo de su [página oficial](#), donde para Windows o macOS utilizaremos el asistente de instalación que viene y para el caso de Linux lo instalaremos desde consola una vez descargado con el comando: `sudo dpkg -i <nombre_paquete.deb>`

3.2.1 Entorno de trabajo



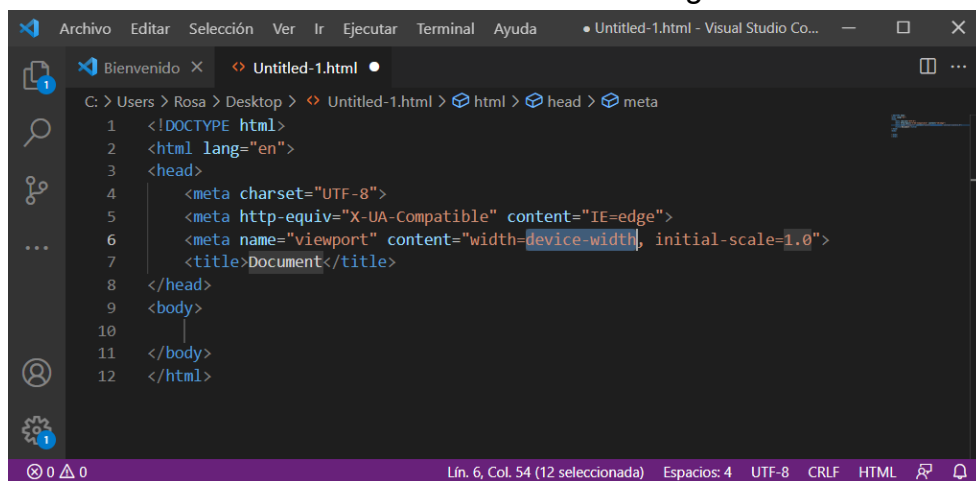
Al pulsar sobre nuevo archivo, nos crea una pestañita nueva donde nos permite seleccionar el idioma para empezar a trabajar. En nuestro caso, seleccionaremos HTML, y con esto lo que haremos es que Visual Studio nos ayude con las etiquetas.

3.2.2 Atajos de teclado



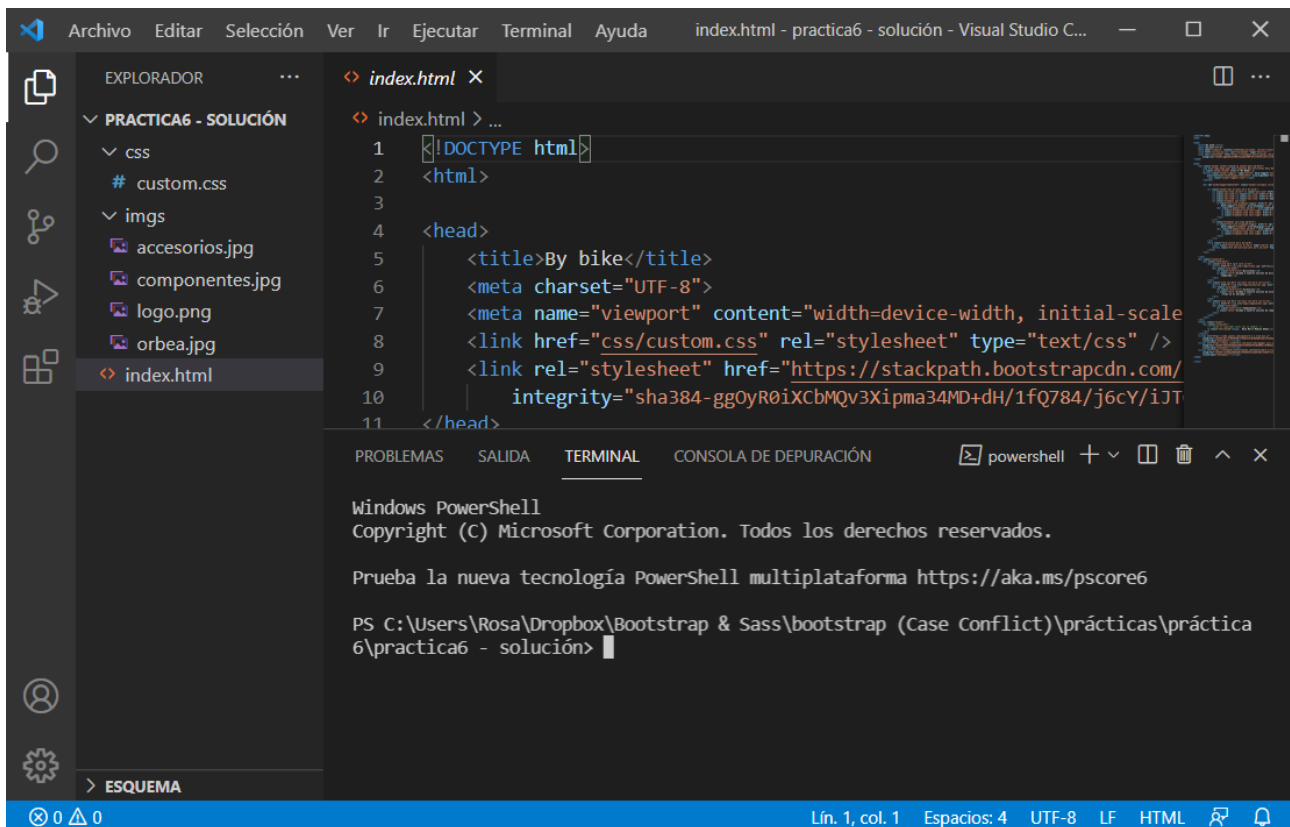
Hemos seleccionado la herramienta de Visual Studio Code porque nos ofrece múltiples atajos de teclado que nos van a facilitar el desarrollo de nuestras páginas web.

Por ejemplo, si en un archivo vacío como es en el caso de la imagen anterior, en lugar de haber teclado <, hubiéramos puesto html:5 y pusamos enter nos crea la estructura básica de un archivo HTML. El resultado es el siguiente:



Otro de los atajos más utilizados es el de dar formato al documento (indentarlo) con el comando Shift + Alt + F. Podemos encontrar multitud de páginas con atajos, y otras funcionalidades que presenta esta herramienta.

3.2.3 Secciones del IDE



Las partes en las que está compuesto nuestro IDE son:

1. **Editor:** Es la parte central y más importante, es donde estarán nuestros archivos para editarlos. El entorno nos permite tener varios archivos abiertos en diferentes pestañas, así como dividir la ventana del editor en tres para visualizar dos archivos a la vez.
2. **Panel lateral:** En este panel encontramos diferentes vistas como el explorador de archivos, en el caso de tener control de versiones también podremos ver los archivos que han cambiado con respecto al último commit, o la vista de búsqueda, depuración y extensiones.
3. **Panel de actividades:** barra que se encuentra a la izquierda donde encontraremos 5 grandes botones que nos permitirá cambiar entre las diferentes vistas: Explorador, Búsqueda, Control de versiones, Depuración y Extensiones. Estas son las que aparecerán en el panel lateral.
4. **Barra de estado:** Esta barra situada en la parte inferior de la aplicación, muestra diferente información como es el número de líneas del archivo y caracteres, o información relativa con el control de versiones.
5. **Paneles:** En la parte inferior del editor se muestran los diferentes paneles donde se recoge información acerca de la depuración, errores, avisos o un terminal de texto integrado.

3.2.4 Extensiones

Una de las características más importantes que tiene VSCode es la posibilidad que nos ofrece de enriquecerlo instalando extensiones desde su [Marketplace](#) y que nos permite agregar temas y muchos servicios adicionales que lo que hacen es facilitar su uso atucompletando, verificando sintaxis, etc.

Algunas de las extensiones más características y que nos facilitaran el trabajo en este módulo son: [Auto Close Tag](#) es un plugin que nos añade automáticamente el cierre de etiquetas en lenguajes como XML, HTML, PHP, etc. [ESLint](#) para cuando empecemos la unidad de JavaScript, [XML Tools](#) para cuando trabajemos con XML, [Spanish Language Pack for Visual Studio Code](#) para aquellos que todavía no están familiarizados a trabajar con herramientas en inglés.

Por ejemplo, vamos a instalar la extensión [Open in Browser](#), que nos permite abrir documentos HTML directamente en el navegador. Para instalarlo tenemos dos opciones, la primera es accediendo al [Marketplace](#) y buscando el paquete (o pulsando sobre el enlace que os he facilitado directamente) y la otra opción es pulsando del panel izquierdo el botón:



Una vez que hemos pulsado el botón pasamos a buscar la extensión que es desarrollada por TechER y pulsaremos sobre install.

Para poderla utilizar cuando estemos editando un archivo HTML, debemos pulsar con el botón derecho sobre el código y seleccionar Open in Default Browser.

Ejercicio 1:

Abre *Visual Studio Code*, selecciona el idioma HTML y usa el atajo de etiquetas que te genera una página web básica. Una vez generado dentro de las etiquetas body inserta lo siguiente: `<h1> Hola esto es una prueba</h1>`

El resultado del código debería ser algo así:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Hola esto es una prueba</h1>
</body>
</html>
```

Una vez hayas insertado el texto, guárdalo y muéstralo en el navegador utilizando la extensión anterior (Open in Browser)

3.2.5 Cambiar el tema del editor

Si queremos modificar el tema del editor que viene por defecto debemos ir a File → Preferences y elegir la opción Color Theme para pasar a elegir el estilo con el que más cómodo estemos.

3.3 Otras alternativas

Además de Visual Studio Code, hay otras alternativas también multiplataforma como son [Kate](#) o [Sublime Text](#). Todas ellas como podéis ver son muy parecidas y con la misma filosofía de trabajo. Se ha elegido VSCode por ser la más completa y más utilizada hoy en día para el desarrollo web.

4 ¿Qué es una aplicación web?

Según wikipedia, se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

En general, este término se utiliza para aquellos programas que son ejecutados en el entorno del navegador o en un motor de navegador embebido en otro sistema. Por tanto, además de las típicas páginas web, dentro de este término incluimos las llamadas aplicaciones híbridas y las aplicaciones de escritorio que utilizan tecnologías web.

4.1 ¿Qué es “la web”?

La web es el diminutivo de World Wide Web o www cuyas tecnologías para su funcionamiento (XHTML/HTML, URL, HTTP) fueron desarrolladas en el año 1990 por Tim Berners Lee.

4.2 Elementos de una aplicación web

Cuando hablamos de una aplicación web podemos distinguir dos grandes puntos. El primero de ellos es el cliente, que es donde está el usuario final que consumirá los recursos y que para ello utiliza el navegador web para acceder a la aplicación. El otro es el servidor que es el encargado de atender las peticiones de los clientes y facilitarles la información que solicitan.

4.3 Funcionamiento de una aplicación web

Para acceder a un servidor web es imprescindible saber su dirección IP. La dirección IP no es más que el número que nos permite identificar a un ordenador en una red TCP/IP como es internet.

Una dirección IP es un número de 32 bits que normalmente se expresa como 4 octetos de ocho bits en notación decimal punteada, y cada octeto va de 0 a 255.

Por ejemplo, la dirección IP de Google es: 142.250.184.3

Ejercicio 2:

- Averigua tu dirección IP (Pista: comando `ipconfig` ó `ifconfig`)
- Intenta acceder a la página web de Facebook a través de su dirección IP. Para ello, usa el comando: `nslookup` y averigua su dirección IP

4.4 Dominios

Como podemos observar, acordarse de la dirección IP de cada web a la que queremos acceder es algo francamente difícil, por ello, se ideó un método que simplificara todo esto, es el denominado servicio de nombres de dominio o DNS.

Un dominio de internet no es más que el nombre identificador de un equipo, y que por tanto lo hace más fácil de recordar que la dirección IP.

4.5 Protocolo HTTP

El protocolo HTTP es el encargado de gestionar el intercambio de información entre el servidor y el cliente. Cuando queremos acceder a una página web debemos indicar el protocolo que vamos a utilizar junto con el nombre del dominio del servidor en donde se encuentra.

Se denomina URL (Uniform Resource Location) al conjunto de protocolo + dominio + página web que queremos visualizar.

Por ejemplo: <https://iessanvicente.com/alumnos.php>

Como vemos, una URL comienza con el nombre de su esquema seguido de dos puntos (http, https, ftp, mailto, etc.). Las URLs empleados por HTTP, el protocolo para transmitir páginas web, es el tipo más popular de URL.

4.6 Lenguajes

Cuando hablamos de una aplicación web, es importante determinar el lenguaje o lenguajes en la que se desarrollan:

1. **Lenguajes servidor:** son los encargados de informar al servidor de realizar ciertas tareas tras recibir una petición por parte del cliente. Dentro de este conjuntos de lenguajes tenemos PHP, JSP, ASP.Net, y hoy en día JavaScript con frameworks como Node.js
2. **Lenguajes cliente:** permiten que el cliente vea e interactúe con la aplicación web. En este lado nos encontramos el lenguaje HTML y CSS para el diseño de páginas y JavaScript o TypeScript para permitir que el usuario pueda interactuar con la web.