

Tecnología y Uso de Internet

Transparencias (Año 2024)



©2024 Cristian Gómez Macías, Fco. Javier Pérez Blanco

Algunos derechos reservados. Este documento se distribuye bajo la licencia “Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en <https://creativecommons.org/licenses/by-sa/4.0/deed.es>

- Tema 1: Sociedad de la información..... 4
- Tema 2: Evolución y tecnologías de Internet y la web..... 88
- Tema 3: Maquetación web – HTML..... 134
- Tema 4: Diseño web – CSS..... 208
- Tema 5: Web dinámica: JavaScript..... 234
- Tema 6: Sistemas de gestión de contenidos..... 264

Tecnología y Uso de Internet

Tema I: Sociedad de la información

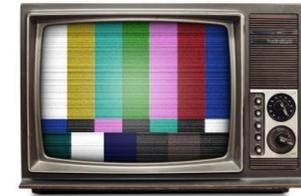
- 1. Introducción.**
- 2. Información e informática.**
- 3. Hardware y software.**
- 4. Sistemas operativos.**
- 5. Material adicional.**

I. Introducción

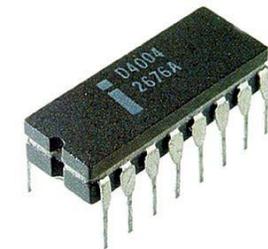
- Disponemos de mucha más **información** que en cualquier otra época de la historia, que además es posible transmitir de manera **instantánea** a prácticamente **cualquier lugar** del planeta.
- El avance de las TIC (Tecnologías de la Información y las Comunicaciones) ha permitido que la información **influya** en acontecimientos relevantes, en los mercados económicos, en nuestro día a día, etc.
- El término Sociedad de la Información trataba originariamente de reflejar esta realidad, en la que la información, puesta a disposición de la sociedad por medio de las nuevas tecnologías, juega un **papel preponderante**.

- La **tecnología** es capaz de abrir **nuevos mercados** y/o sacudir brutalmente los que ya existen.
- El nivel de adopción de la tecnología digital (Internet, las aplicaciones que corren sobre ella y los dispositivos que permiten a las personas mantenerse en contacto entre ellas y con la red, cuando, como y donde quieran) proporciona un **contexto histórico** que no se parece en nada a cualquier otra época de la historia.
- Determinar si este contexto es **una oportunidad o una amenaza** para las empresas, individuos, etc., depende de si somos capaces de:
 - Entender y asimilar los principios que lo rigen.
 - Entender la forma de abordar los retos que plantea.

- Drásticos aumentos en la cantidad y flujo de información:
 - Aparición de la imprenta.
 - El telégrafo.
 - El teléfono.
 - La radio.
 - La televisión.

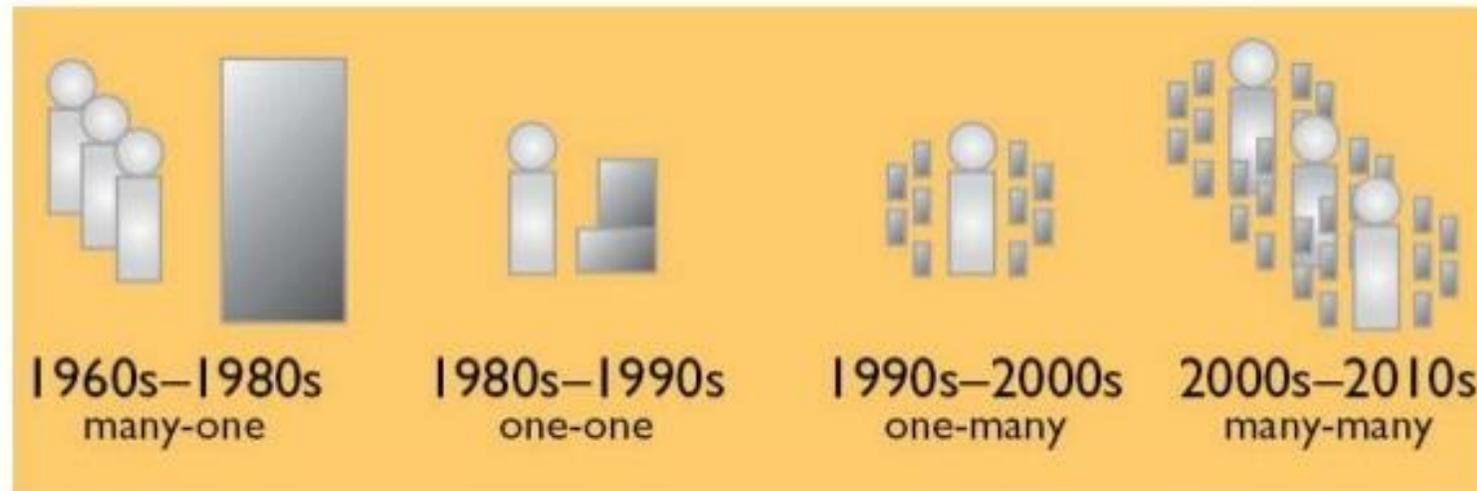


- Ninguno de ellos consiguió convertirse en el factor decisivo en la actividad económica.
- **Base** sobre la que se ha construido esta sociedad:
 - Aparición del PC.
 - Aparición de los microprocesadores.



Principales corrientes de la computación

- **Computación centralizada (60's – 80's)**
- **Computación de escritorio (80's – 90's)**
- **Computación ubicua (a partir de los 90's)**



- **Computación centralizada (60's – 80's):**
 - Múltiples usuarios.
 - Un ordenador.



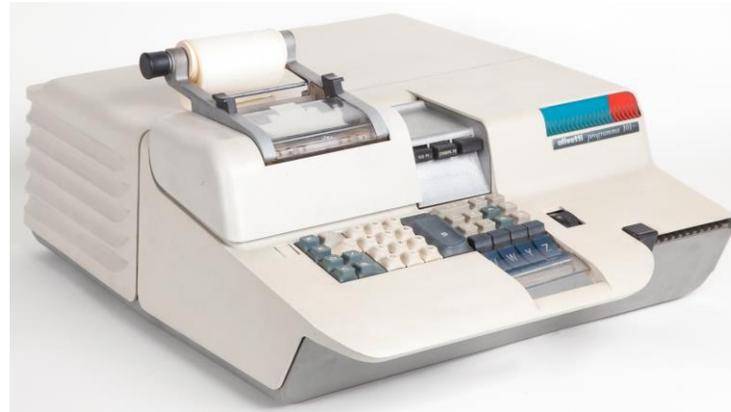
IBM 704 (1954)



IBM 360 (1964)



- **Computación de escritorio (80's – 90's):**
 - Un usuario.
 - Un ordenador.



Programma 101, Olivetti (1964)



MacBook Pro, Apple (2021)

- **Computación ubicua (a partir de los 90's):**
 - Múltiples usuarios.
 - Múltiples ordenadores.



2. Información e informática

- **Informática:**
 - El término nace en Francia en 1962, INFORMATIQUE, INFORmation autoMATIQUE.
 - Adoptado por España en 1968, INFORMÁTICA, INFORmación autoMÁTICA.
 - En Inglés COMPUTER SCIENCE.
- **Definición (RAE):** conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento racional y automático de la información.
- **Objetivo:** tratamiento racional y automático de la información.
- **Medio:** ordenadores.

- **Información:** conjunto de hechos o condiciones que pueden ser objeto de una comunicación, de una interpretación y/o de un tratamiento.
- **Tipos de Información:**
 - Elemental (no procesada): datos.
 - Elaborada (procesada): información que aporta valor. Resultados.
- **Valor de la información:** está directamente relacionado con la utilidad que presente para los responsables de decisiones en el cumplimiento de las metas de la organización.

- **Dato vs Información:**
 - La información es un conjunto de datos organizados de tal modo que adquieren un **valor adicional** más allá del propio.
 - La información, de forma contraria a los datos, implica que los datos están **procesados** de tal manera que resulten útiles o significativos para el receptor de los mismos.
 - Los datos se pueden considerar la **materia prima** para obtener información.
- Ejemplos:
 - 1** Datos: piezas de madera, clavos, etc.
Información: silla, mesa, armario, etc.
 - 2** Datos: nombres de calles, distancia entre dos puntos, horarios de apertura de tiendas.
Información: rutas desde un origen hasta un destino.

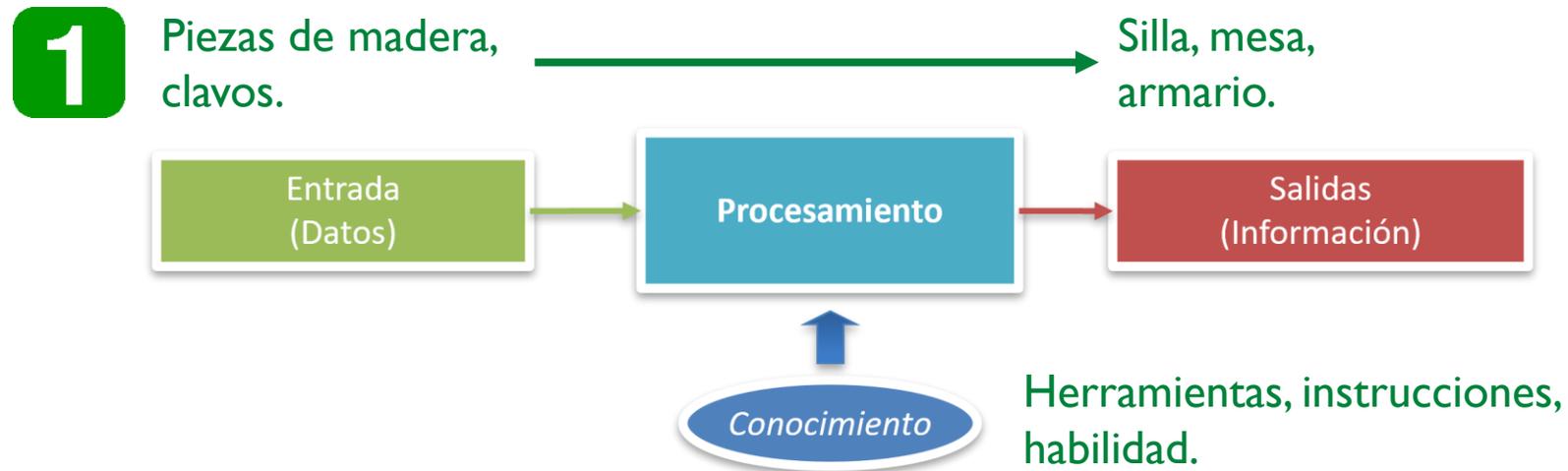
¿Cómo se transforman los datos en información?

- La conversión de datos en información implica un **proceso** o una serie de tareas lógicamente relacionadas entre sí y ejecutadas con el fin de producir un objetivo.
- El proceso para definir relaciones entre datos requiere de conocimiento.



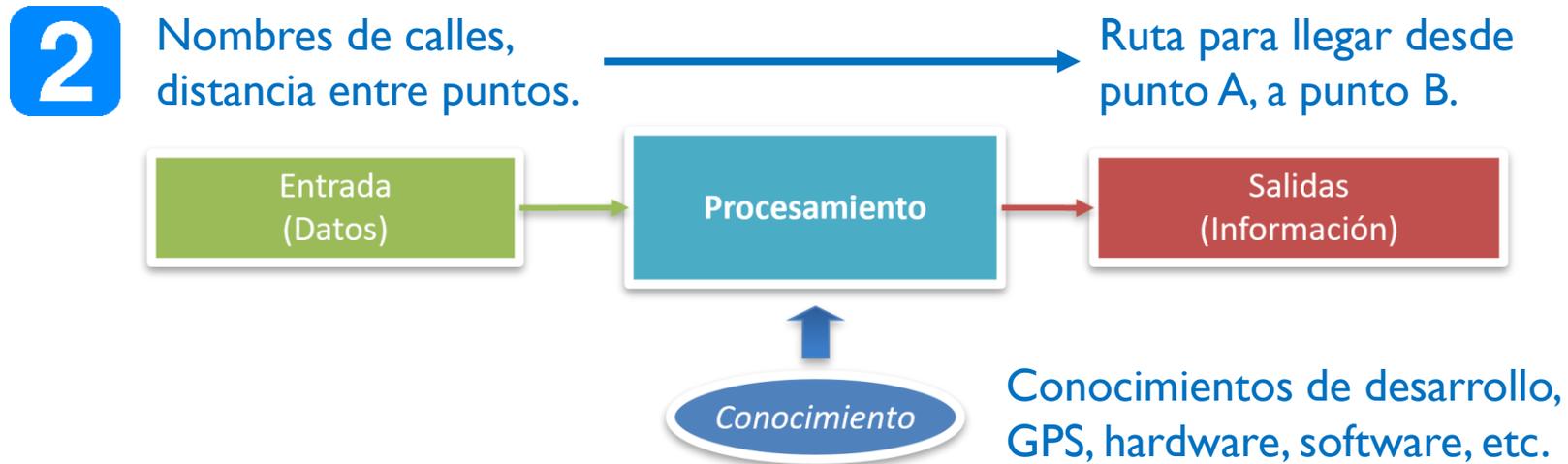
¿Cómo se transforman los datos en información?

- La conversión de datos en información implica un **proceso** o una serie de tareas lógicamente relacionadas entre sí y ejecutadas con el fin de producir un objetivo.
- El proceso para definir relaciones entre datos requiere de conocimiento.



¿Cómo se transforman los datos en información?

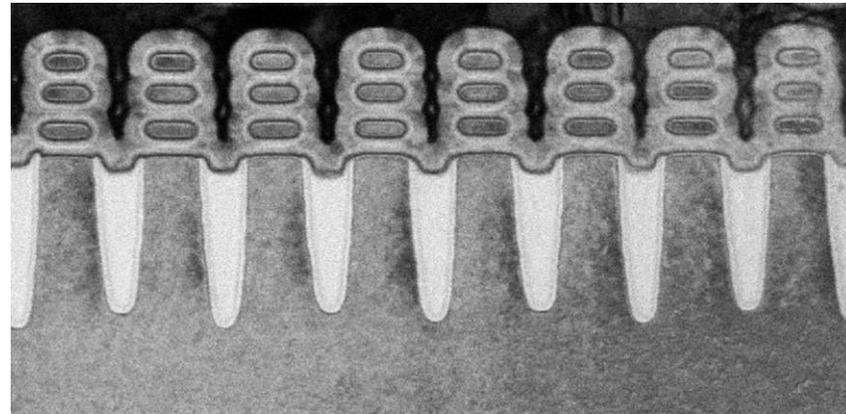
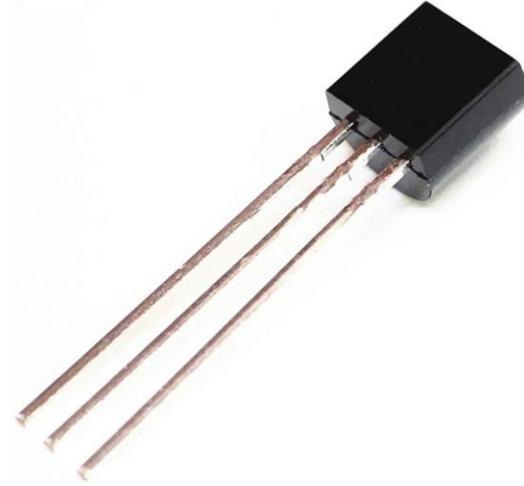
- La conversión de datos en información implica un **proceso** o una serie de tareas lógicamente relacionadas entre sí y ejecutadas con el fin de producir un objetivo.
- El proceso para definir relaciones entre datos requiere de conocimiento.



¿Sabrías decir por qué cuando vemos imágenes/vídeos de informática o información digital nos muestran imágenes con 0 y 1?



Representación de la información



Sistema decimal:

- Conjunto de símbolos y reglas que permiten representar datos numéricos.
- Diez símbolos distintos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- El valor de cada dígito está asociado al de una potencia de base 10.
- Sistema posicional: un símbolo tiene distinto valor dependiendo de la posición que ocupen en la cifra (unidades, decenas, centenas, etc.).
- Ejemplo:
 - 528
 - $5 * 10^2 + 2 * 10^1 + 8 * 10^0 = 500 + 20 + 8 = 528$

Sistema binario:

- Conjunto de símbolos y reglas que permiten representar datos numéricos.
- Dos símbolos distintos: 0, 1.
- El valor de cada dígito está asociado al de una potencia de base 2.
- Sistema posicional: un símbolo tiene distinto valor dependiendo de la posición que ocupen en la cifra.
- Los ordenadores representan los datos con interruptores eléctricos. Un interruptor encendido (paso de corriente) representa un 1. Un interruptor apagado (sin paso de corriente) representa un 0.
- Ejemplo:
 - $528 = 1000010000$
 - $1 * 2^9 + 0 * 2^8 + 0 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 0 * 2^1 + 0 * 2^0 = 512 + 16 = 528$

Más ejemplos:

- $1001 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
- $8 + 0 + 0 + 1 = 9$
- $1001_2 = 9_{10}$

- $101011 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$
- $32 + 0 + 8 + 0 + 2 + 1 = 43$
- $101011_2 = 43_{10}$

Realizar la conversión a decimal de los siguientes números binarios:

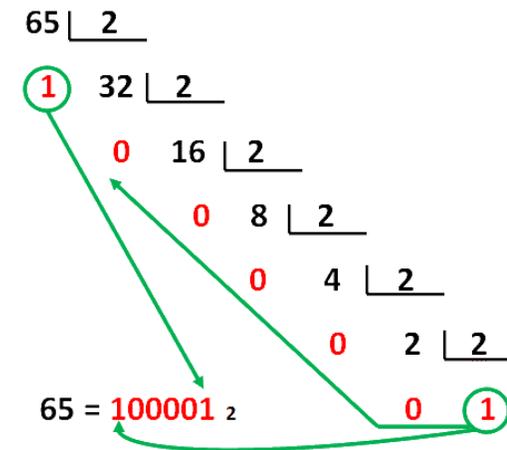
- a) 1011
- b) 11010
- c) 101101

Soluciones:

- a) 11
- b) 26
- c) 45

Conversión entre número decimal y binario:

- Dividimos sucesivamente entre 2 y escribimos los restos obtenidos en el orden inverso al que han sido obtenidos.
- Ejemplo: convertir al sistema binario el número 77.
 - $77 / 2 = 38$, resto: 1
 - $38 / 2 = 19$, resto: 0
 - $19 / 2 = 9$, resto: 1
 - $9 / 2 = 4$, resto: 1
 - $4 / 2 = 2$, resto: 0
 - $2 / 2 = 1$, resto: 0
 - $1 / 2 = 0$, resto: 1
 - Por tanto $77_{10} = 1001101_2$



Realizar la conversión a binario de los siguientes números en decimal:

- a) 9
- b) 23
- c) 38

Soluciones:

- a) 1001
- b) 10111
- c) 100110

Reglas generales:

- La cantidad de dígitos necesarios para representar un número en el sistema binario es mayor que en el sistema decimal.
- Con n dígitos binarios pueden representarse un máximo de 2^n números diferentes.
- El número más alto que puede escribirse con n dígitos binarios es 2^n-1 .

¿Y cómo representamos símbolos no numéricos?

- Códigos **ASCII** (American Standard Code for Information Interchange):
 - ASCII de 7 bit. $2^7 = 128$ Caracteres codificables.
 - ASCII Extendido de 8 bit. $2^8 = 256$ Caracteres codificables.
- Actualmente utilizamos la tabla **Unicode**:
 - UTF-8 (8 bits), UTF-16 (16 bits), UTF-32 (32 bits).

Código	Tipo de Caracteres	Código	Tipo de Caracteres
0-31	Caracteres de control	32	Espacio en Blanco
33-47	Signos y Símbolos	48-57	Dígitos (0 . . . 9)
58-64	Símbolos	65-90	Mayúsculas(A-Z)
91-96	Símbolos	97-122	Minúsculas(a-z)
123-127	Símbolos	128-255	Símbolos de Control

Ejemplo tabla ASCII:

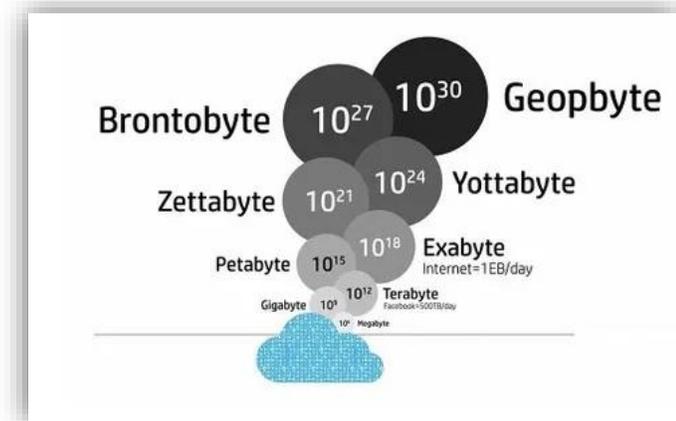
Caracteres ASCII imprimibles									ASCII extendido								
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł
33	21h	!	65	41h	A	97	61h	a	129	81h	ü	161	A1h	í	193	C1h	ł
34	22h	"	66	42h	B	98	62h	b	130	82h	é	162	A2h	ó	194	C2h	Ṁ
35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	û	195	C3h	Ṁ
36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ñ	196	C4h	Ṁ
37	25h	%	69	45h	E	101	65h	e	133	85h	à	165	A5h	Ñ	197	C5h	Ṁ
38	26h	&	70	46h	F	102	66h	f	134	86h	á	166	A6h	ª	198	C6h	Ṁ
39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	º	199	C7h	Ṁ
40	28h	(72	48h	H	104	68h	h	136	88h	è	168	A8h	¿	200	C8h	Ṁ
41	29h)	73	49h	I	105	69h	i	137	89h	ë	169	A9h	®	201	C9h	Ṁ
42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	è	170	AAh	¬	202	CAh	Ṁ
43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ï	171	ABh	½	203	CBh	Ṁ
44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	ì	172	ACH	¼	204	CCh	Ṁ
45	2Dh	.	77	4Dh	M	109	6Dh	m	141	8Dh	ì	173	ADh	»	205	CDh	Ṁ
46	2Eh	:	78	4Eh	N	110	6Eh	n	142	8Eh	Ë	174	Aeh	«	206	CEh	Ṁ
47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	À	175	Afh	»	207	CFh	Ṁ
48	30h	0	80	50h	P	112	70h	p	144	90h	É	176	B0h	⋮	208	D0h	Ṁ
49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h	⋮	209	D1h	Ṁ
50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h	⋮	210	D2h	Ṁ
51	33h	3	83	53h	S	115	73h	s	147	93h	ø	179	B3h	⋮	211	D3h	Ṁ
52	34h	4	84	54h	T	116	74h	t	148	94h	ò	180	B4h	⋮	212	D4h	Ṁ
53	35h	5	85	55h	U	117	75h	u	149	95h	ò	181	B5h	⋮	213	D5h	Ṁ
54	36h	6	86	56h	V	118	76h	v	150	96h	ù	182	B6h	⋮	214	D6h	Ṁ
55	37h	7	87	57h	W	119	77h	w	151	97h	ù	183	B7h	⋮	215	D7h	Ṁ
56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h	⋮	216	D8h	Ṁ
57	39h	9	89	59h	Y	121	79h	y	153	99h	Û	185	B9h	⋮	217	D9h	Ṁ
58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	Ü	186	BAh	⋮	218	DAh	Ṁ
59	3Bh	;	91	5Bh	[123	7Bh	{	155	9Bh	ø	187	BBh	⋮	219	DBh	Ṁ
60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch	£	188	BCh	⋮	220	DCh	Ṁ
61	3Dh	=	93	5Dh]	125	7Dh	}	157	9Dh	£	189	BDh	⋮	221	DDh	Ṁ
62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	x	190	BEh	⋮	222	DEh	Ṁ
63	3Fh	?	95	5Fh	_				159	9Fh	f	191	BFh	⋮	223	DFh	Ṁ
															224	E0h	Ó
															225	E1h	Ø
															226	E2h	Ò
															227	E3h	Ö
															228	E4h	ö
															229	E5h	Õ
															230	E6h	µ
															231	E7h	þ
															232	E8h	þ
															233	E9h	Û
															234	EAh	Ü
															235	EBh	Û
															236	ECh	ý
															237	EDh	ÿ
															238	EEh	-
															239	EFh	.
															240	F0h	
															241	F1h	±
															242	F2h	
															243	F3h	¼
															244	F4h	¶
															245	F5h	§
															246	F6h	÷
															247	F7h	
															248	F8h	°
															249	F9h	·
															250	FAh	·
															251	FBh	¹
															252	FCh	³
															253	FDh	²
															254	FEh	■
															255	FFh	

Unidades de información:

- Bit/Byte: unidades de información manipuladas en informática. Se usan para describir la capacidad de los componentes de un ordenador, tamaño de los programas y ficheros.
 - **BIT**, Binary digiT: unidad mínima de información del sistema binario. Cada bit, o dígito binario puede tomar el valor 0 o 1. Con un bit puedo hacer referencia a dos datos, dos elementos, verdadero o falso, encendido o apagado, etc.
 - **BYTE**, Agrupación de 8 bits. Con 1 Byte (8 bits) se puede hacer referencia a $2^8 = 256$ elementos distintos. Código ASCII 256 caracteres codificables.

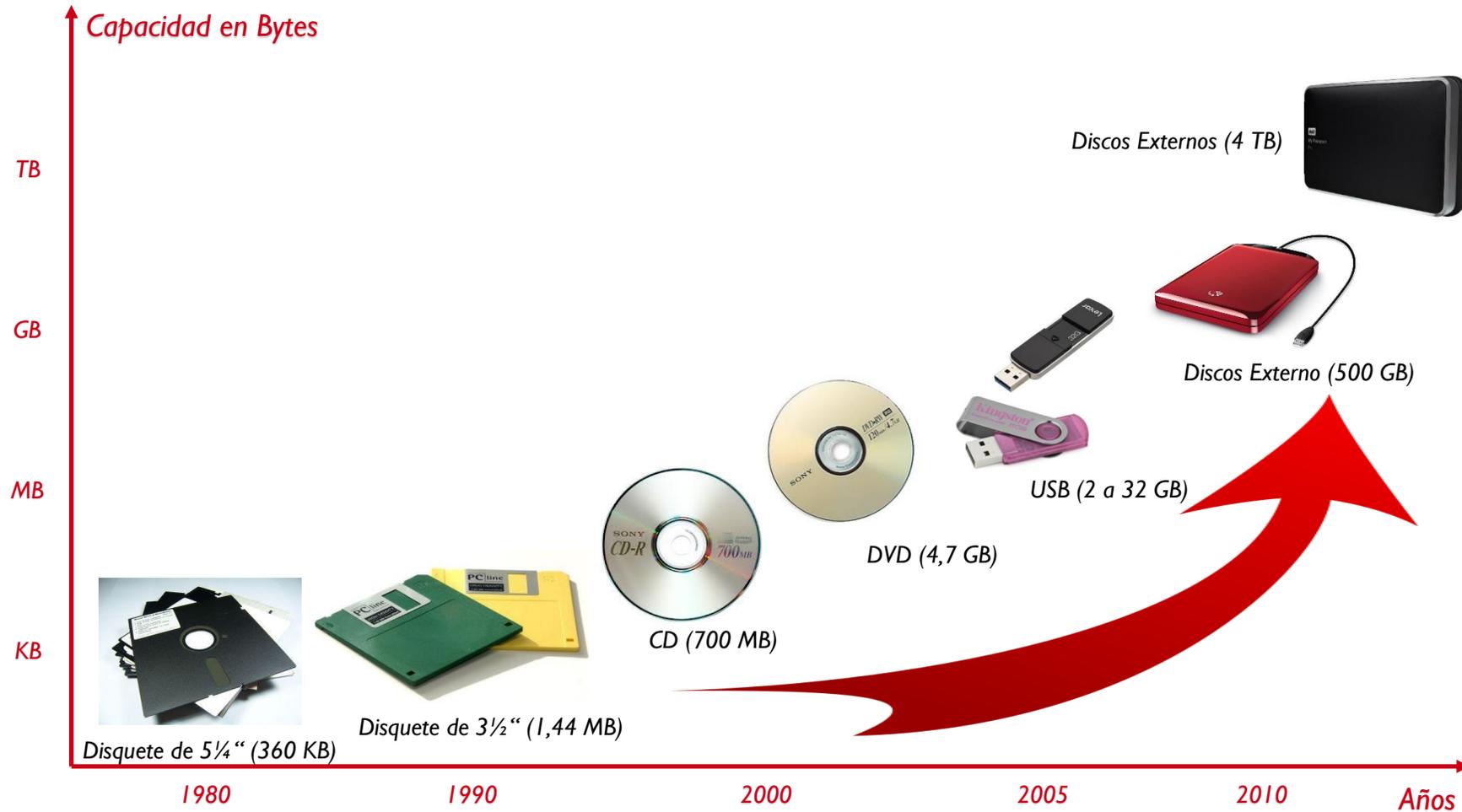
Múltiplos de bytes:

Múltiplo	Nº de bytes (potencia de 2)	Nº de bytes (decimal)
Kilobyte (KB)	2^{10} bytes	1024 bytes
Megabyte (MB)	2^{20} bytes	1024 KB
Gigabyte (GB)	2^{30} bytes	1024 MB
Terabyte (TB)	2^{40} bytes	1024 GB
Petabyte (PB)	2^{50} bytes	1024 TB
Exabyte	2^{60} bytes	1024 PB
Zetabyte	2^{70} bytes	1024 ZB
Yotabyte



Representación de la información

Evolución de las capacidades de los dispositivos de almacenamiento de información:



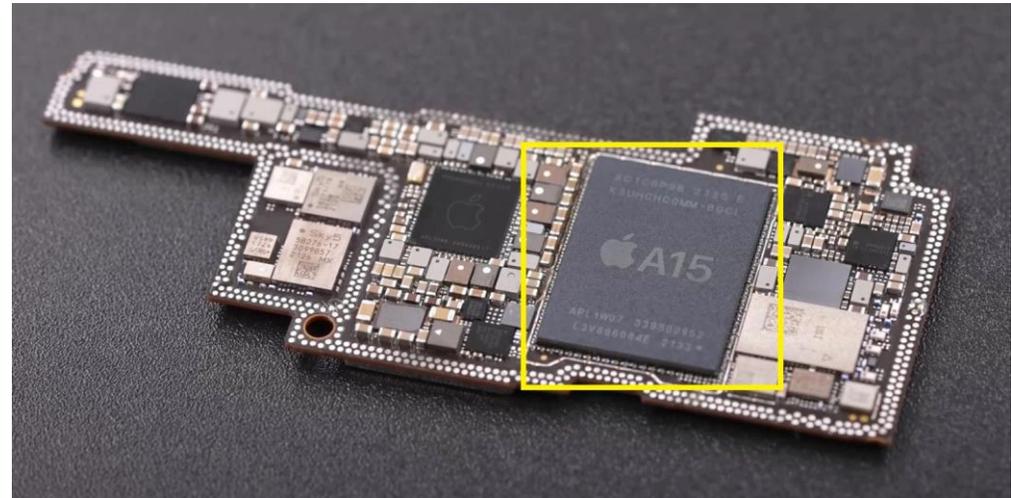
3. Hardware y software

- **Hardware:** componentes físicos de un ordenador. Ej.: procesador, memorias, pantalla, teclado, etc.
- **Software:** componentes lógicos (programas) de un ordenador. Ej.: Microsoft Word, Google Chrome, Windows, etc.
- Los componentes hardware no hacen nada si no hay un software que les diga **qué** tienen que hacer, y **cómo** lo tienen que hacer.



Componentes internos: componentes físicos imprescindibles para el funcionamiento del ordenador, y que son inseparables de este. Ejemplos:

- **Procesador (*Central Processing Unit* - CPU):** interpreta las instrucciones de un software realizando operaciones simples aritméticas y lógicas.



Componentes internos: componentes físicos imprescindibles para el funcionamiento del ordenador, y que son inseparables de este. Ejemplos:

- **Memoria ROM (*Read Only Memory*):** pequeña memoria no volátil de solo lectura que contiene el firmware estrictamente necesario para el funcionamiento de la máquina, por ejemplo, para el arranque. Este firmware estará diseñado para un hardware específico. Un ejemplo de firmware es la BIOS del sistema.



Componentes internos: componentes físicos imprescindibles para el funcionamiento del ordenador, y que son inseparables de este. Ejemplos:

- **Memoria RAM (*Random Access Memory*):** memoria principal de almacenamiento a corto plazo (volátil) en la que se guardan las instrucciones y los datos de los programas en ejecución.



Componentes internos: componentes físicos imprescindibles para el funcionamiento del ordenador, y que son inseparables de este. Ejemplos:

- **Disco duro (interno):** memoria secundaria de almacenamiento a largo plazo (no volátil).



HDD



SSD

Componentes internos: componentes físicos imprescindibles para el funcionamiento del ordenador, y que son inseparables de este. Ejemplos:

- **Tarjeta gráfica (*Graphics Processing Unit - GPU*):** coprocesador especializado en el procesamiento de gráficos y de operaciones en coma flotante que reduce la carga de trabajo del procesador principal.



Componentes internos: componentes físicos imprescindibles para el funcionamiento del ordenador, y que son inseparables de este. Ejemplos:

- **Placa base:** circuito impreso al que se conectan el resto de los componentes del ordenador.



Componentes internos: componentes físicos imprescindibles para el funcionamiento del ordenador, y que son inseparables de este. Ejemplos:

- **Otros:** tarjeta de sonido, tarjeta de red ethernet, tarjeta de red wifi, fuente de alimentación, etc.

Componentes externos o periféricos: componentes físicos auxiliares. Pueden ser de entrada, de salida, o de entrada-salida, en función de si permiten introducir datos o información desde el exterior hacia el interior del sistema, o proyectar datos o información desde el interior hacia el exterior del sistema. Ejemplos:

- **Periféricos de entrada:** ratón, teclado, micrófono, cámara, escáner, etc.
- **Periféricos de salida:** monitor, altavoces, impresora, proyector, etc.
- **Periféricos de entrada-salida:** pantalla táctil, impresora multifunción, disco duro externo, router, lector-grabador de CD/DVD etc.

Existen diferentes **clasificaciones del software**:

- **En función de su utilidad:**
 - **Software de aplicación:** programas especializados en realizar una tarea determinada. Ejemplos: Word, PowerPoint, Google Chrome, Adobe Photoshop, etc.
 - **Software de sistema:** programas que permiten el correcto funcionamiento del sistema. Ejemplos: sistemas operativos, firmwares, etc.

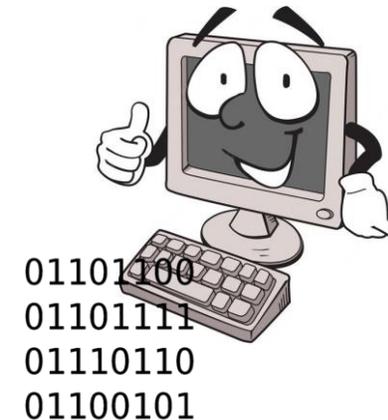
Existen diferentes **clasificaciones del software**:

- **En función de sus derechos de uso:**
 - **Software propietario o privativo:** programas que pertenecen a un individuo u organización, para el cual no existe acceso a su código fuente y normalmente se comercializan.
Ejemplos: Microsoft Windows, Office, Adobe Photoshop, etc.
 - **Software libre:** programas que se distribuyen libremente, con acceso a su código fuente para estudiarlo, modificarlo y/o mejorarlo. En función de la licencia de uso, permiten la redistribución del código modificado. Pueden pertenecer a una organización, o a la comunidad.
Ejemplos: Linux, OpenOffice, Mozilla Firefox, etc.

- **Algoritmo:** conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
- Normalmente pueden diseñarse diferentes algoritmos para resolver un mismo problema. Unos pueden ser más eficientes que otros.
- Ejemplos:
 - Ejemplo 1: llegar a clase por la mañana.
 - Ejemplo 2: resolver un puzle.
 - Ejemplo 3: suma lenta.
 - Ejemplo 4: resolver el cubo de Rubik (<https://www.rubikaz.com/resolucion.php>).

Una aplicación informática, programa o software es un **algoritmo expresado mediante algún lenguaje de programación.**

- Los componentes hardware no hacen nada si no hay un software que les diga **qué** tienen que hacer, y **cómo** lo tienen que hacer.
- Pero el hardware solo entiende de voltajes altos y voltajes bajos (binario).



```
01101100
01101111
01110110
01100101
```

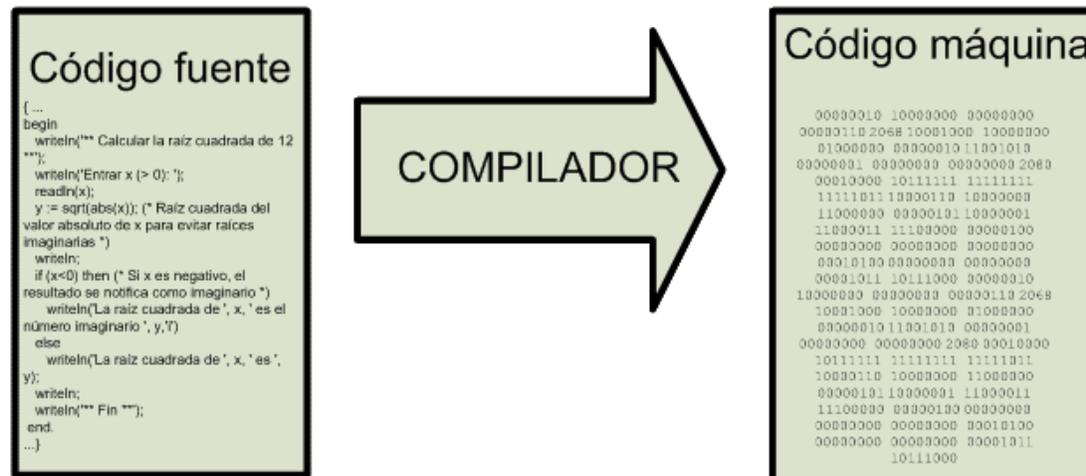
- ¿Debemos escribir los programas en **código binario**?

```

00101110 11110011 00101110 00101110 00101110 00101110 00111100
01010111 01100011 00101110 00101110 00101110 00101110 00101110
00101110 00101110 00101110 00101110 00101110 00101110 00101110
01000000 00101110 00101110 00101110 01110011 00101110 00101110
00101110 00101110 01100100 00101110 00101110 01000111 01001000
01100100 00101110 00101110 01010011 00101000 00101110 00101110
00101110 00101110 01110011 00101110 00101110 00101110 00101110
01001000 01100101 01101100 01101100 01101111 00100000 01010111
01101111 01110010 01101100 01100100 01001110 00101000 00101110
00101110 00101110 00101110 00101000 00101110 00101110 00101110
00101110 00101000 00101110 00101110 00101110 00101110 00101000
00101110 00101110 00101110 00101110 01110011 00101110 00101110
00101110 00101110 01101000 01100101 01101100 01101100 01101111
01110111 01101111 01110010 01101100 01100100 00101110 01110000
01111001 01110100 00101110 00101110 00101110 00101110 00111100
01101101 01101111 01100100 01110101 01101100 01100101 00111110
00101110 00101110 00101110 00101110 01110011 00101110 00101110
00101110 00101110
    
```

- **Inviabile:** a la hora de escribir en binario, deberíamos hacerlo para un **hardware específico**. Además, sería difícil entenderlo y no **cometer errores**.

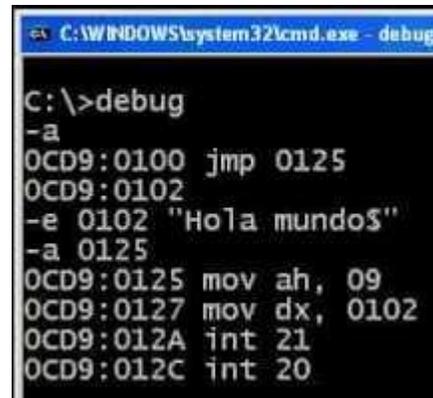
- **Solución:** escribir los programas utilizando un **lenguaje más natural** y emplear un *traductor* que lo transforme a binario.
- Ese *traductor* es un programa llamado **compilador**.



Existen diferentes **clasificaciones de los lenguajes de programación**:

- **En función del nivel de abstracción:**

- Lenguajes de **bajo nivel**: más cercanos al hardware. Permiten controlar el hardware directamente. Más alejados del lenguaje natural. Ejemplos: código máquina (binario), ensamblador, etc.
- Lenguajes de **alto nivel**: más alejados del hardware (mayor independencia). Más cercanos al lenguaje natural. Ejemplos: COBOL, C, Java, Python, JavaScript, etc.



```

C:\>debug
-a
OCD9:0100 jmp 0125
OCD9:0102
-e 0102 "Hola mundo$"
-a 0125
OCD9:0125 mov ah, 09
OCD9:0127 mov dx, 0102
OCD9:012A int 21
OCD9:012C int 20
    
```

Programa "Hello World" en Ensamblador.



```

1  /**
2   * HelloWorld
3   */
4  public class HelloWorld {
5
6     public static void main(String[] args) {
7         System.out.println("Hello World!");
8     }
9  }
    
```

Programa "Hello World" en Java.

Existen diferentes **clasificaciones de los lenguajes de programación**:

- **En función del método de ejecución:**
 - Lenguajes **compilados**: antes de poder ejecutar el programa se ha de generar el código máquina mediante el compilador. Este código máquina se ejecutará directamente en el ordenador de destino. Ejecución más rápida.
Ejemplos: C, C++, etc.
 - Lenguajes **interpretados**: un intérprete genera el código máquina a la vez que lo ejecuta. Ejecución más lenta, pero los programas son independientes del ordenador de destino.
Ejemplos: JavaScript (¡el intérprete forma parte del navegador web!).

Dos ejemplos: Java y JavaScript.

```

1 public class Main {
2
3     public static void main(String[] args) {
4
5         int i = 1;
6
7         if (i == 1) {
8             System.out.println("HOLA");
9         } else {
10            System.out.println("ADIOS");
11        }
12    }
13 }
14 }

```

Ejemplo de aplicación en Java.

```

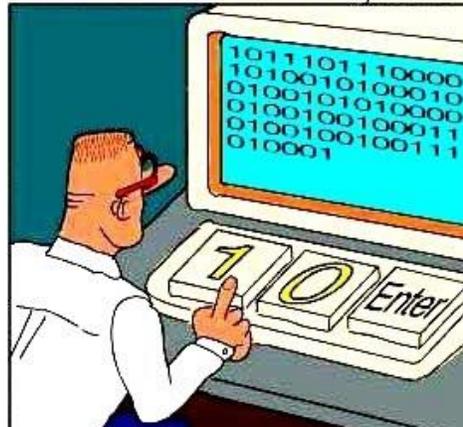
1 function miPrimeraFuncion() {
2
3     var i = 1;
4
5     if (i == 1) {
6         alert("HOLA");
7     } else {
8         alert("ADIOS");
9     }

```

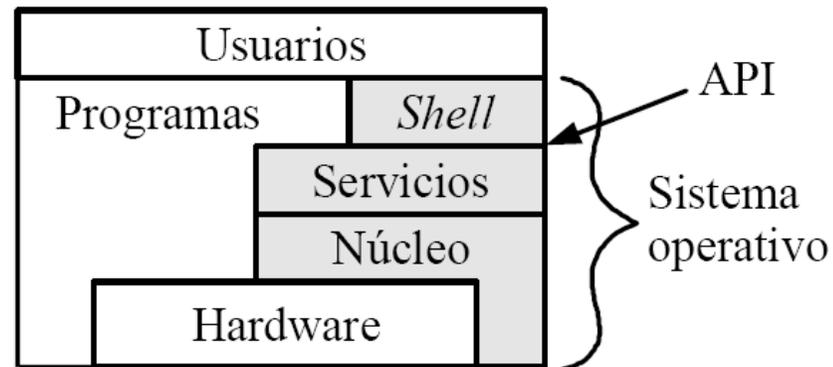
Ejemplo de aplicación en JavaScript.

4. Sistemas operativos

- Un ordenador moderno consta de componentes internos (procesadores, memorias, discos, etc.) y periféricos (impresora, teclado, ratón, monitor, etc.).
- Si los programadores tuviesen que comprender el funcionamiento y la interacción de todo esto, no escribirían código probablemente.

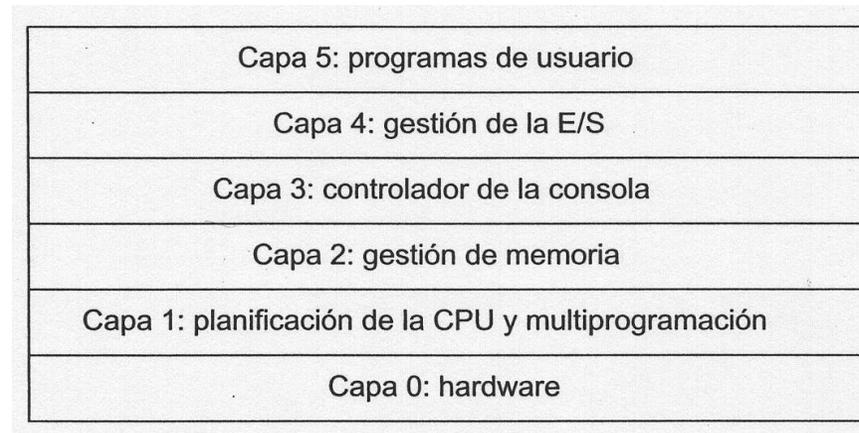


- Por el motivo anterior, los ordenadores cuentan con una capa de software denominada Sistema Operativo (S.O.).



- El S.O. proporciona a los programas de usuario un modelo de computador más simple y limpio.
- Además, se encarga de gestionar todos los recursos (software y hardware) del computador.

- Podemos entender el S.O. como un programa que hace de intermediario entre el usuario y el hardware de un ordenador.

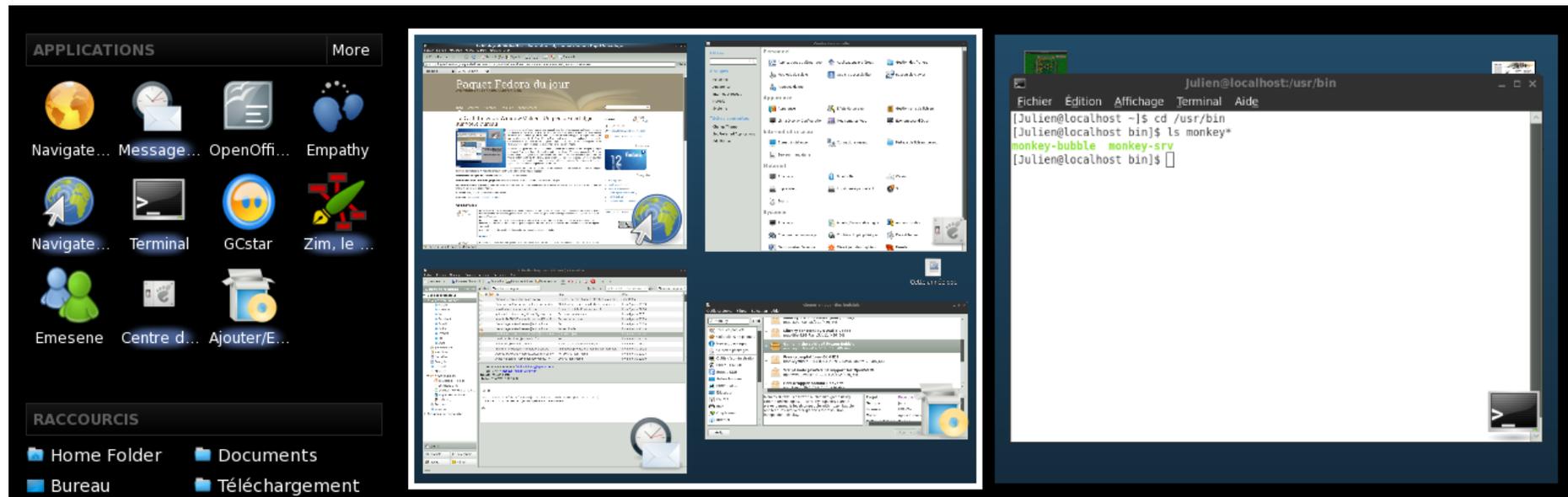


- Algunos de los principales **objetivos** del S.O. son:
 - Ejecutar los programas del usuario.
 - Hacer que el ordenador sea cómodo de usar.
 - Utilizar eficientemente los recursos del ordenador.

- Algunas de las principales **tareas** del S.O. son:
 - Comunicación con los periféricos.
 - Administración de recursos.
 - Coordinación de procesos simultáneos (multitasking).
 - Gestión de memoria.
 - Gestión de programas y datos.
 - Gestión de comunicaciones y redes.

Interacción de los usuarios con los S.O.:

- **Shell, consola o terminal:** basado en texto.
- **Interfaz gráfica de usuario (*Graphic User Interface* - **GUI**):** basado en elementos gráficos o iconos.



Principales S.O.:

- **Microsoft:**
 - MS-DOS (1981).
 - Microsoft Windows (1990).
- **Unix:**
 - macOS (1984)
 - GNU Linux (1991)



Microsoft Windows



Mac OS



GNU Linux

MS-DOS: Microsoft Disk Operating System:

- En 1980 IBM diseñó la IBM PC y contactó a Bill Gates para obtener una licencia de su intérprete de BASIC (un lenguaje de programación) y le preguntaron si sabía de un S.O. que se ejecutara en la PC.
- Gates sugirió a IBM contactar con Digital Research (D.R.).
- El jefe de D.R. no se reunió con IBM y mandó un subordinado, lo que se ha considerado como una de las peores decisiones de negocios de la historia.
- IBM volvió a contactar a Gates para ver si podía proveerles de un S.O.

MS-DOS: Microsoft Disk Operating System:

- Gates compró QDOS (Quick and Dirty Operating System) a un fabricante local de Seattle (Tim Paterson) por una cifra supuestamente entre 50 y 75 mil dólares.
- Posteriormente Gates ofreció a IBM un paquete DOS/BASIC y contrató a Tim Paterson para hacer algunas modificaciones.
- El resultado cambió su nombre a MS-DOS. Microsoft otorgó licencias a decenas de fabricantes de equipos y, a día de hoy, domina el mercado mundial.

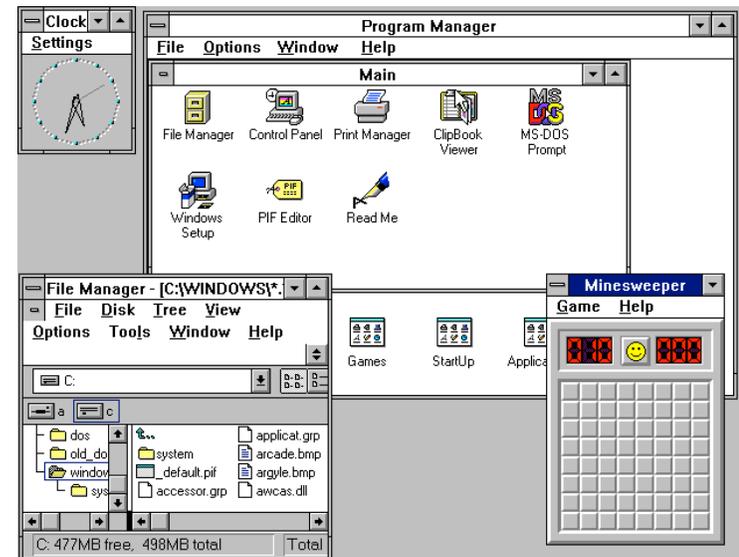
```
Starting MS-DOS...
HIMEM is testing extended memory...done.
C:\>C:\DOS\SMARTDRV.EXE /X
MODE prepare code page function completed
MODE select code page function completed
C:\>dir
Volume in drive C is MS-DOS_6
Volume Serial Number is 40B4-7F23
Directory of C:\

DOS             <DIR>           12.05.20   15:57
COMMAND.COM    54 645  94.05.31   6:22
TIMEZ0        386           9 349  94.05.31   6:22
CONFIG.SYS    144  12.05.20   15:57
AUTOEXEC.BAT  188  12.05.20   15:57
5 file(s)
64 326 bytes
24 760 320 bytes free
C:\>
```



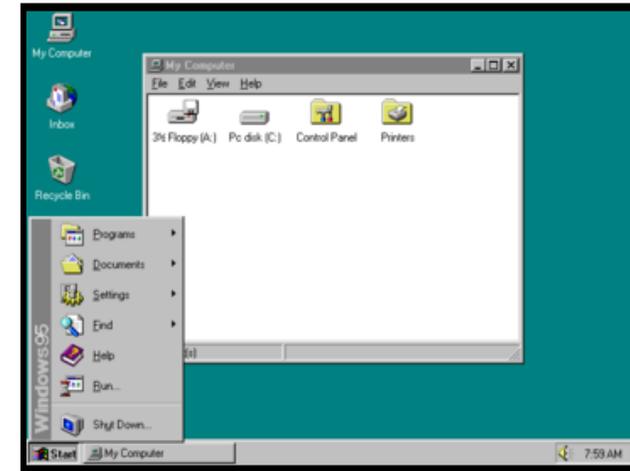
Microsoft Windows 3.0, 3.1, 3.11 (1990-1992):

- No es un S.O., es un ambiente operativo que se ejecuta sobre MS-DOS, el cual es el verdadero S.O.
- Trajo la interfaz gráfica al mundo de los ordenadores personales que utilizaban DOS.
- Aplicaciones específicas (Microsoft Mail).
- Soporte de red, archivos, impresoras.
- Concepto multimedia.
- +10 millones de copias vendidas.



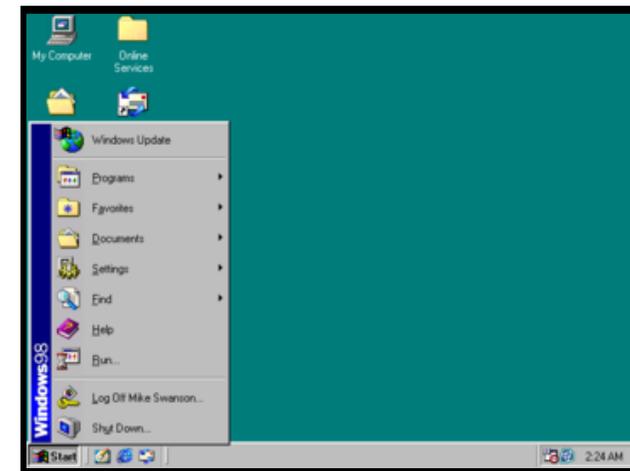
Microsoft Windows 95:

- Multitarea.
- Se incluye un navegador junto al S.O.
- Mezcla aplicaciones de 16 bits y de 32 bits.
- MS-DOS v.7 sigue estando presente en el sistema.



Microsoft Windows 98:

- Plugins HTML para el escritorio.
- Última versión atada al MS-DOS.
- Dispositivos plug and play.



Microsoft Windows XP:

- Abreviatura de eXPerience.
- Controlar a todos los usuarios del sistema.
- Multitarea multiprocesador.



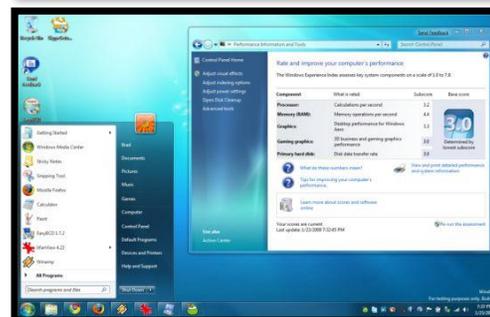
Microsoft Windows Vista:

- Requiere alta capacidad de recursos hw.
- Multiusuario.
- Baja compatibilidad con aplicaciones.
- Bajo rendimiento respecto a otros S.O.



Microsoft Windows 7:

- Actualización de Vista.
- Capacidades multitáctiles.



Microsoft Windows 8:

- Estilo Metro.
- Concebido para pantallas táctiles.
- Tienda de Apps.



Microsoft Windows 10:

- Combina virtudes de W7 y W8.
- Cortana: asistente digital.
- Nuevo menú Inicio.
- Nuevo navegador Edge.
- Modo Tablet.



UNIX:

- Primer S.O. completo.
- Multiusuario, multitarea ampliamente utilizado para controlar programas en estaciones de trabajo y servidores.
- Origen final de los '60. Proyecto de General Electric, AT&T Bell y el MIT. Ken Thompson y Dennis Ritchie retoman el proyecto.
- Escrito en C.
- Distribución gratuita a las Universidades. Distribuciones comerciales en los 70.
- En la mayoría de sus versiones se usa una interfaz de líneas de comando. Actualmente se usan interfaces gráficas.

Mac OS:

- Relacionado con la familia de ordenadores de Apple, la primera del mercado no compatible con IBM.
- Solo para hw específico.
- Basado en UNIX: estabilidad.
- Primera interfaz verdaderamente gráfica de usuario WIMP (Windows, Icons, Menus, Pointer). Famoso por la facilidad de uso.
- Todas las aplicaciones tienen la misma apariencia (look and feel).
- Muy adecuado para aplicaciones basadas en gráficos e imágenes.
- Versiones: System 1 a System 7, Mac OS 8 a 13.
- Multitarea, multiprocesador, multiusuario, en tiempo real.



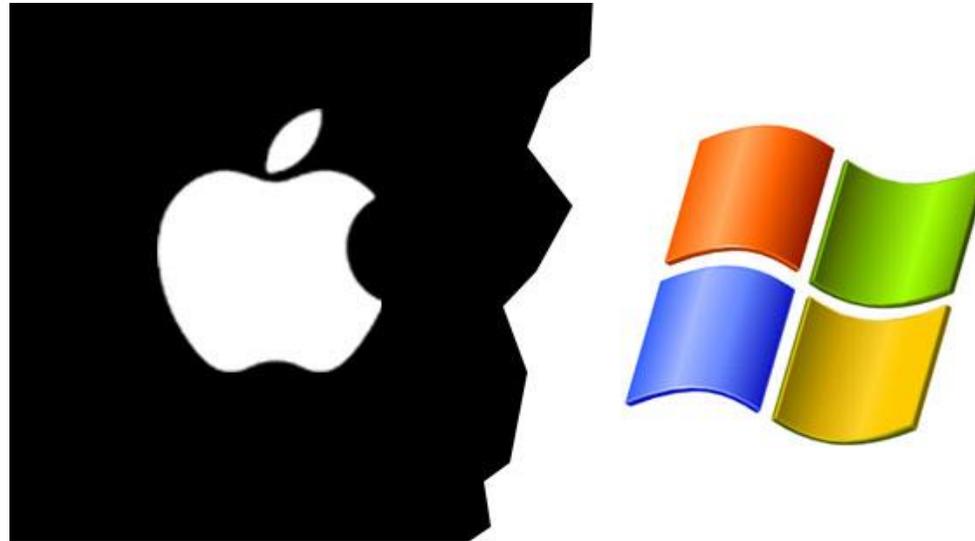
GNU Linux:

- 1985, Richard Stallman crea la Free Software Foundation y el proyecto GNU (GNU's Not Unix), como forma de recuperar el espíritu cooperativo de los primeros días de la computación.
- El proyecto GNU consiste en el desarrollo de un S.O. y un conjunto de aplicaciones completamente libre y compatible con Unix. Desarrollar versión libre de cualquier aplicación.
- Linus Torvalds en 1991 basándose en una versión de Unix llamada MINIX, generó paso a paso el núcleo (kernel) de este S.O.
- Demanda pocos recursos. Usable en cualquier ordenador.
- Multitarea, multiprocesador, multiusuario, multiplataforma.



Breve historia de la guerra de la publicidad entre Apple y Microsoft:

- <http://blog.hubspot.com/marketing/mac-pc-advertising-history>



Android:

- Android es un sistema operativo basado en el núcleo Linux.
- Diseñado para pantallas táctiles.
- Evolución:
 - 2003: Rich Miner, Nick Sears, Andy Rubin y Chris White empezaban a darle cuerpo al proyecto Android Inc.
 - 2005: Google adquiere Android Inc.
 - Octubre 2008: Android 1.0 (Alpha) - HTC Dream.
 - Agosto 2022: Android 13.0.



Versiones de Android:

- Android 1.0.
- Android 1.1 Petit Four.
- Android 1.5 Cupcake.
- Android 1.6 Donut.
- Android 2.0/2.1 Eclair.
- Android 2.2 Froyo.
- Android 2.3 Gingerbread.
- Android 3.0 Honeycomb.
- Android 4.0 Ice Cream Sandwich.
- Android 4.1 Jelly Bean.
- Android 4.4 KitKat.
- Android 5.0 Lollipop.
- Android 6.0 Marshmallow.
- Android 7.0 Nougat.
- Android 8.0: O ?



iOS:

- Sistema operativo para dispositivos móviles Apple.
- 2007: iOS 1.0. iPhone.
- 2023: iOS 17. iPhone 15.



Marzo, 2023 (datos globales):

- Android: 70,88%.
- iOS: 28,42%.
- Otros (Kai OS, Harmony OS, EMUI, MIUI, etc.): 0,7%.



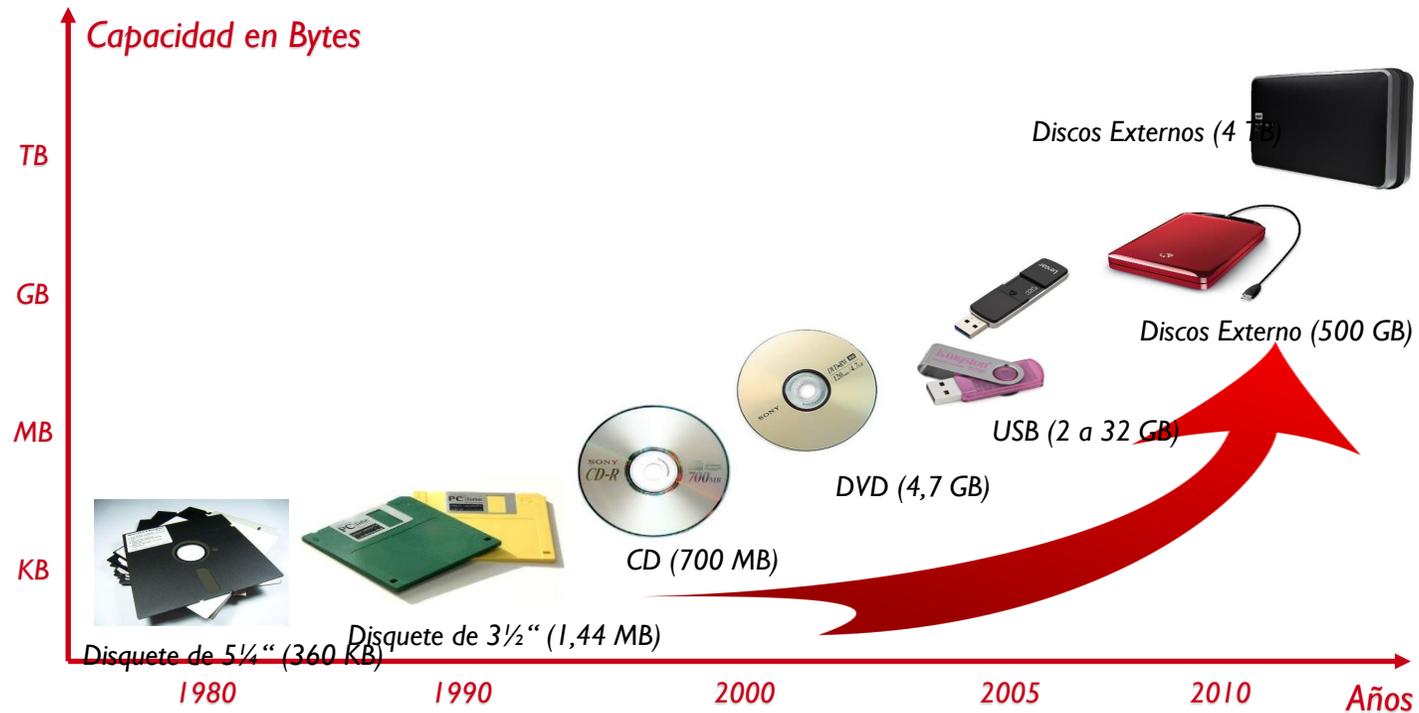
5. Material adicional

Algunos fenómenos derivados o que contribuyen a la sociedad de la información:

- Revolución de los datos masivos.
- Datificación.
- Digitalización.
- Transformación digital.

Revolución de los datos masivos

Como vimos al final de la sección 2 (Información e informática), las capacidades de los dispositivos de almacenamiento de información han crecido de manera prácticamente exponencial en los últimos años.



Este fenómeno se ha producido por **dos motivos relacionados entre sí**:

- **Avances tecnológicos** (reducción del tamaño de los transistores que forman parte de los dispositivos de almacenamiento - Ley de Moore) y abaratamiento de costes.
- Necesidades a nivel de individuo, organizaciones, etc. (cada vez queremos **archivos de mayor calidad** – Ej.: fotos con mayor resolución).

Dado que cada vez existe una mayor cantidad de datos digitales, hablamos de que se ha producido una **revolución de los datos masivos**. Podemos situar el **origen** de esta revolución en dos proyectos de investigación que generaron cantidades inmensas de datos digitales:

- Proyecto Sloan Digital Sky Survey.
- Proyecto Genoma Humano.

Sin embargo, a día de hoy **continuamos generando grandes cantidades de datos digitales**. Ejemplos:

- Internet, redes sociales, etc.
- Internet de las cosas (IoT): redes de dispositivos que pueden digitalizar datos.

Esta revolución de los datos masivos provocó **problemas para las grandes empresas de Internet** (Google, Yahoo, Amazon, etc.):

- La **gran cantidad de datos** acumulados hacía inviable su procesamiento en un único ordenador.
- **Heterogeneidad de datos**. Necesidad de insertar, consultar y procesar datos en diferentes estructuras.
- Necesidad de **procesar rápidamente los datos**.

Como respuesta a estos problemas, surgieron técnicas de análisis de datos relacionadas con el **Big Data**:

- El Big Data tiene como objetivo resolver problemas antiguos y nuevos de manera más eficiente, pero aplicado a extensos volúmenes de datos no usados hasta el momento.
- Podemos definirlo como el conjunto de técnicas y tecnologías para el tratamiento de datos, en entornos de **gran volumen, variedad de orígenes y en los que la velocidad de respuesta es crítica**.
- Técnicas basadas en la **correlación de variables**: como tenemos cantidades inmensas de datos, podemos cruzar dichos datos para encontrar relaciones que en principio no serían visibles a simple vista. Ej.: En EEUU, cuando se aproximan huracanes, se venden más tartas de fresa.

La **datificación** es el proceso de monitorizar una actividad que era "invisible" y convertirla en datos digitales. Dichos datos eran ignorados porque no se disponía de tecnología para medirlos. Actualmente, sin embargo, toda acción, dato, preferencia, etc. está sujeta a ser medida y almacenada. Ejemplos:

- Localización de una persona a través de dispositivos móviles (smartphone, reloj, etc.).
- Vibraciones del motor de un coche.
- Tiempo que un lector pasa en cada página de un libro mediante ebooks.

Todo esto contribuye a que la cantidad de datos digitales continúe aumentando.

Podemos encontrar dos definiciones de **digitalización**:

- Transcripción de señales analógicas en señales digitales con el propósito de facilitar su procesamiento. Ej: pasar un vídeo de una cinta VHS a DVD, escanear un libro, etc.
- Sinónimo de informatización: implantación o aplicación de medios informáticos para el desarrollo de una actividad o trabajo. Ej.: una gestoría que antes llevaba las cuentas de una empresa en un libro de contabilidad, y actualmente lo hace en un libro de Excel.

La **transformación digital** es la reinención de una organización a través de la utilización de la tecnología digital para mejorar la forma en que la organización se desempeña y sirve a quienes la constituyen:

- Digital se refiere al uso de la tecnología que genera, almacena y procesa los datos.
- El término transformación se refiere a un cambio fundamental en la actividad del día a día de una organización, desde los tipos de productos y servicios que produce hasta la forma en que los entrega.

Ejemplos de transformación digital:

- Industria musical: pasamos de vender discos físicos a vender suscripciones a aplicaciones como Spotify.
- Pasamos de vender enciclopedias en papel, a enciclopedias colaborativas como la Wikipedia.

La **diferencia** fundamental entre digitalización y transformación digital es que en el segundo caso **cambia el modelo de negocio**, mientras que en el caso de la digitalización las tareas son exactamente las mismas, pero utilizando medios digitales.



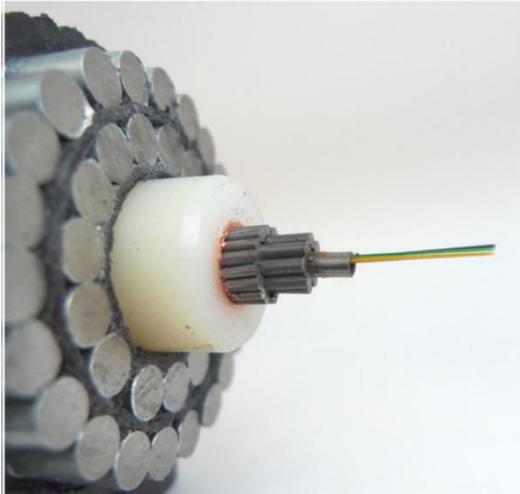
Tecnología y Uso de Internet

Tema 2: Evolución y tecnologías de Internet y la web

- 1. Introducción.**
- 2. Del telégrafo a ARPANET.**
- 3. TCP/IP y la aparición de Internet.**
- 4. El nacimiento de la web.**
- 5. La consolidación de la web.**

I. Introducción

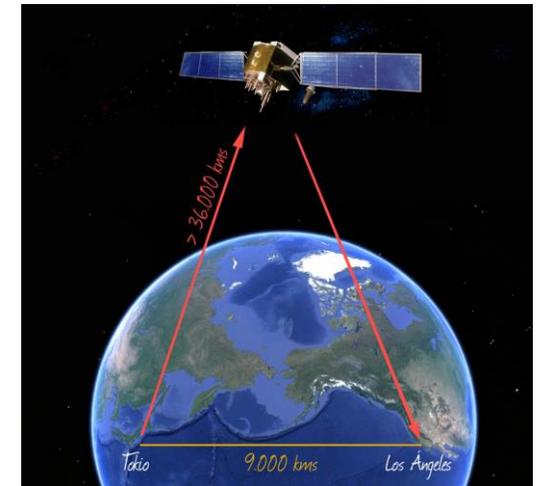
¿Cuál es el principal medio de transmisión de la red de Internet?



<https://www.xataka.com/historia-tecnologica/1-000-millones-de-metros-de-cable-submarino-son-los-responsables-de-que-tengas-internet-en-casa>

¿Por qué una red cableada de fibra óptica? Algunas ventajas:

- Aunque la luz y las ondas de radio pueden transmitirse a velocidades similares, utilizando cable las **distancias son más cortas**: una red satelital económica y sostenible debería emplear satélites geoestacionarios, que se encuentran a aproximadamente 36.000km de altura.
- La transmisión de información por cable es **más resistente** ante inclemencias meteorológicas.
- La transmisión por cable es **más segura** respecto a capturas de información intermedias.

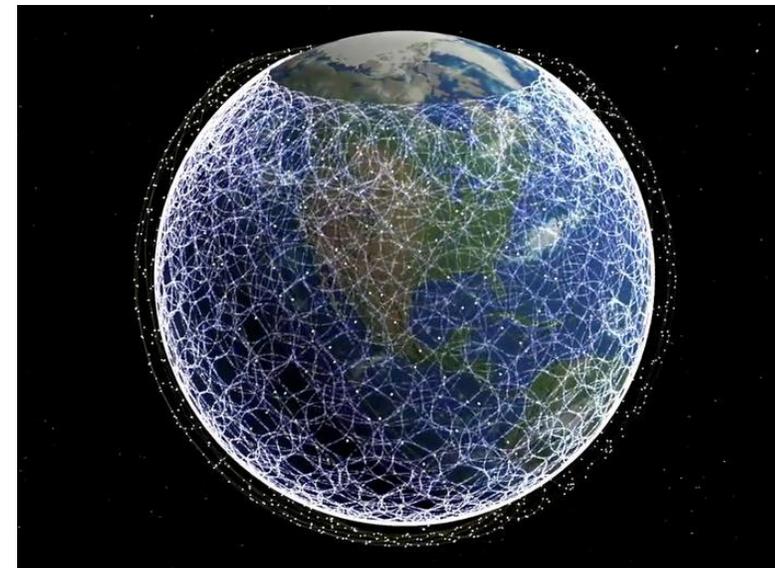


Pero también tiene desventajas. Algunas de ellas son:

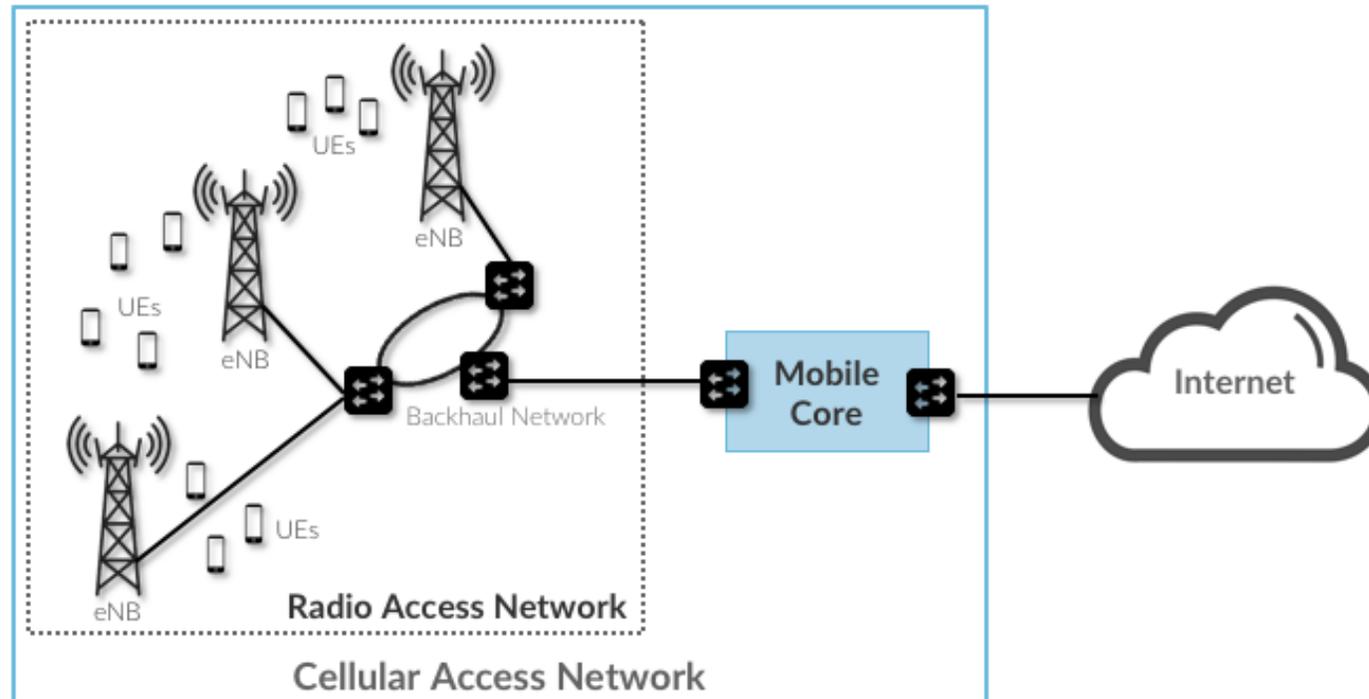
- Resulta **complicado abarcar todos los puntos del planeta** mediante una red cableada.
- Los **cables situados en el exterior y/o a poca profundidad**, aunque protegidos, quedan **demasiado expuestos**.
- Las **reparaciones son muy costosas y/o laboriosas** en determinadas zonas (ej.: océanos), en las que además son más **susceptibles de sufrir daños por catástrofes geológicas** como los terremotos.



Actualmente existe un proyecto para crear una red de Internet satelital utilizando satélites de órbita baja: *Starlink (SpaceX)*.

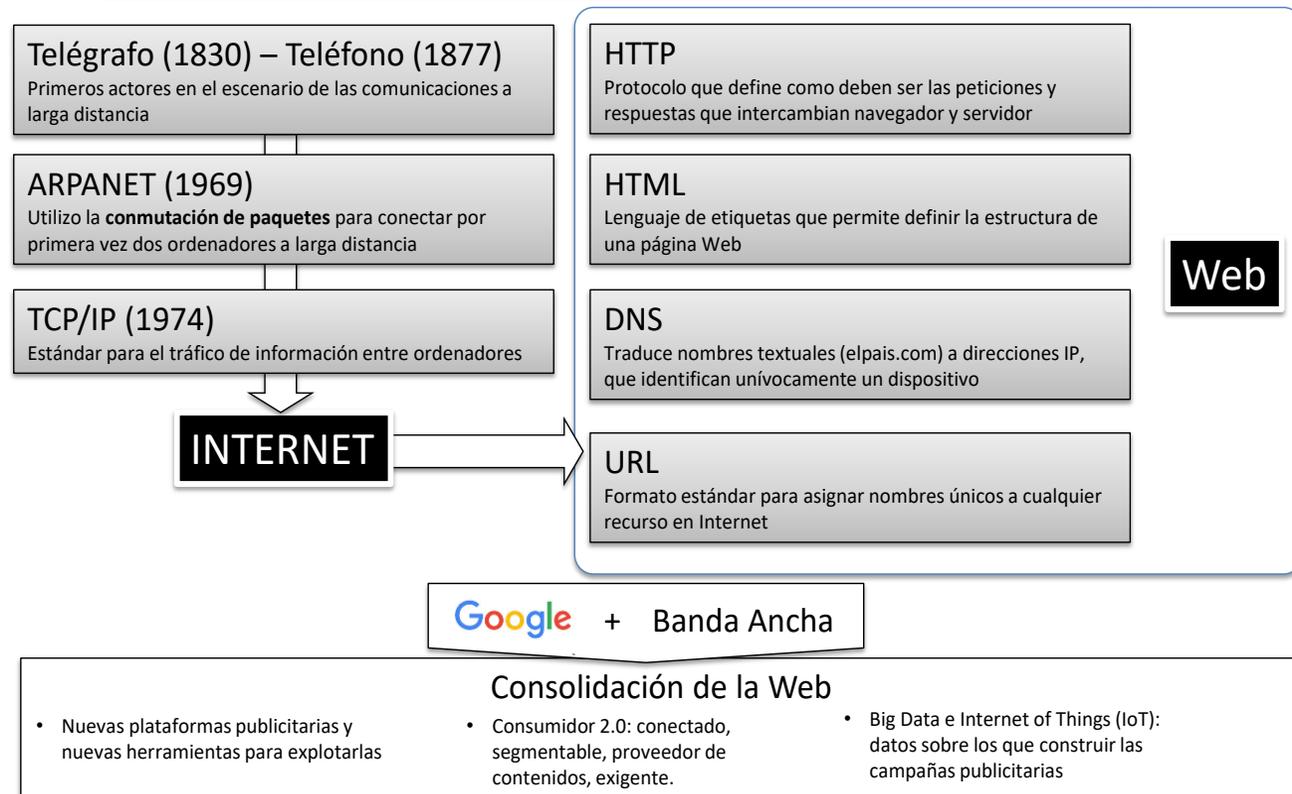


No confundir la red de Internet global con las redes móviles (3G, 4G, 5G, etc.).



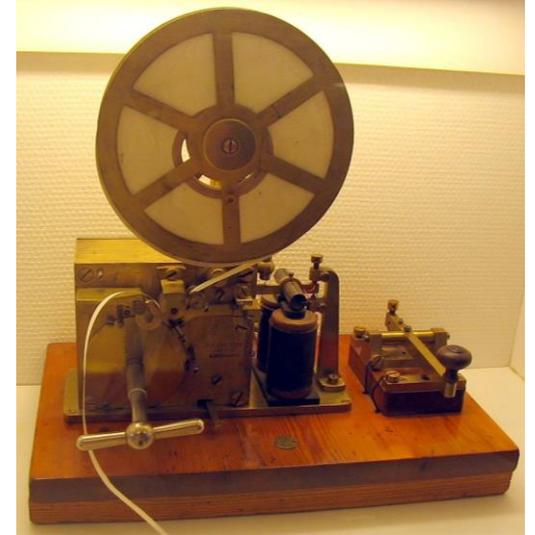
Pero... ¿Cómo hemos llegado hasta aquí?

- Del telégrafo a ARPANET.
- El correo electrónico.
- TCP/IP y la aparición de Internet.
- El nacimiento de la Web.
- La consolidación de la Web.



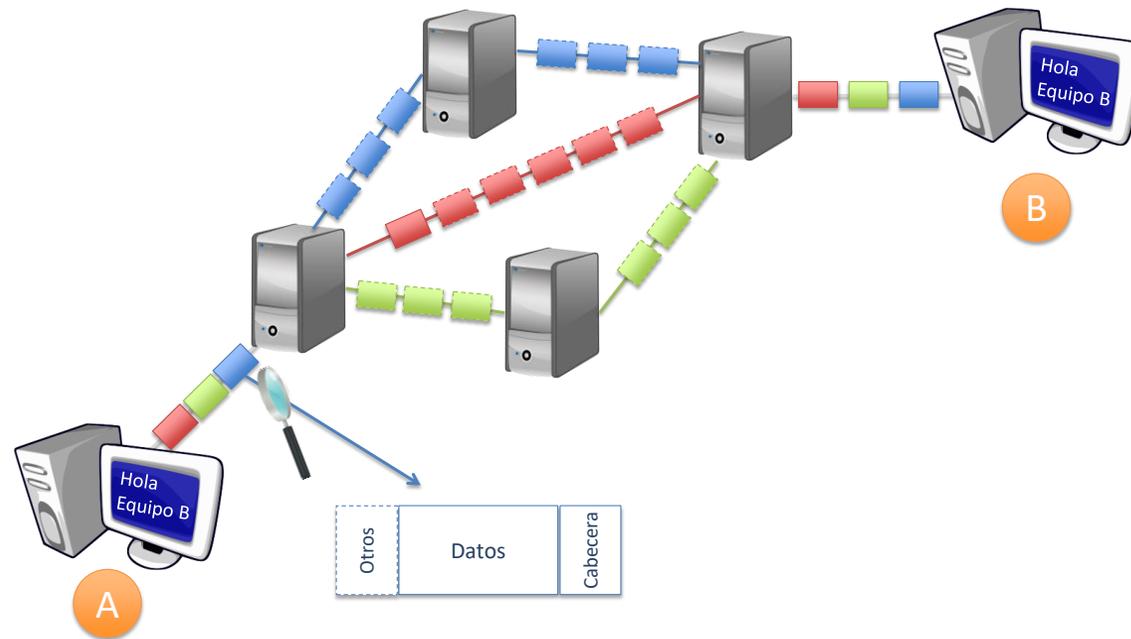
2. Del telégrafo a ARPANET

- Samuel Morse (inventor del código Morse) consiguió en 1843 que USA invirtiera \$30.000 en un proyecto para conectar Washington y Baltimore -> **Telégrafo Morse**.
- Hasta que Antonio Meucci inventó el teléfono y Graham Bell lo patentase en 1877, el telégrafo fue la única herramienta para las comunicaciones a larga distancia.
- Avances posteriores en las comunicaciones:
 - URSS lanza el Sputnik (1957 - 1961).
 - USA crea **ARPA** (Advanced Research Projects Agency, 1958).

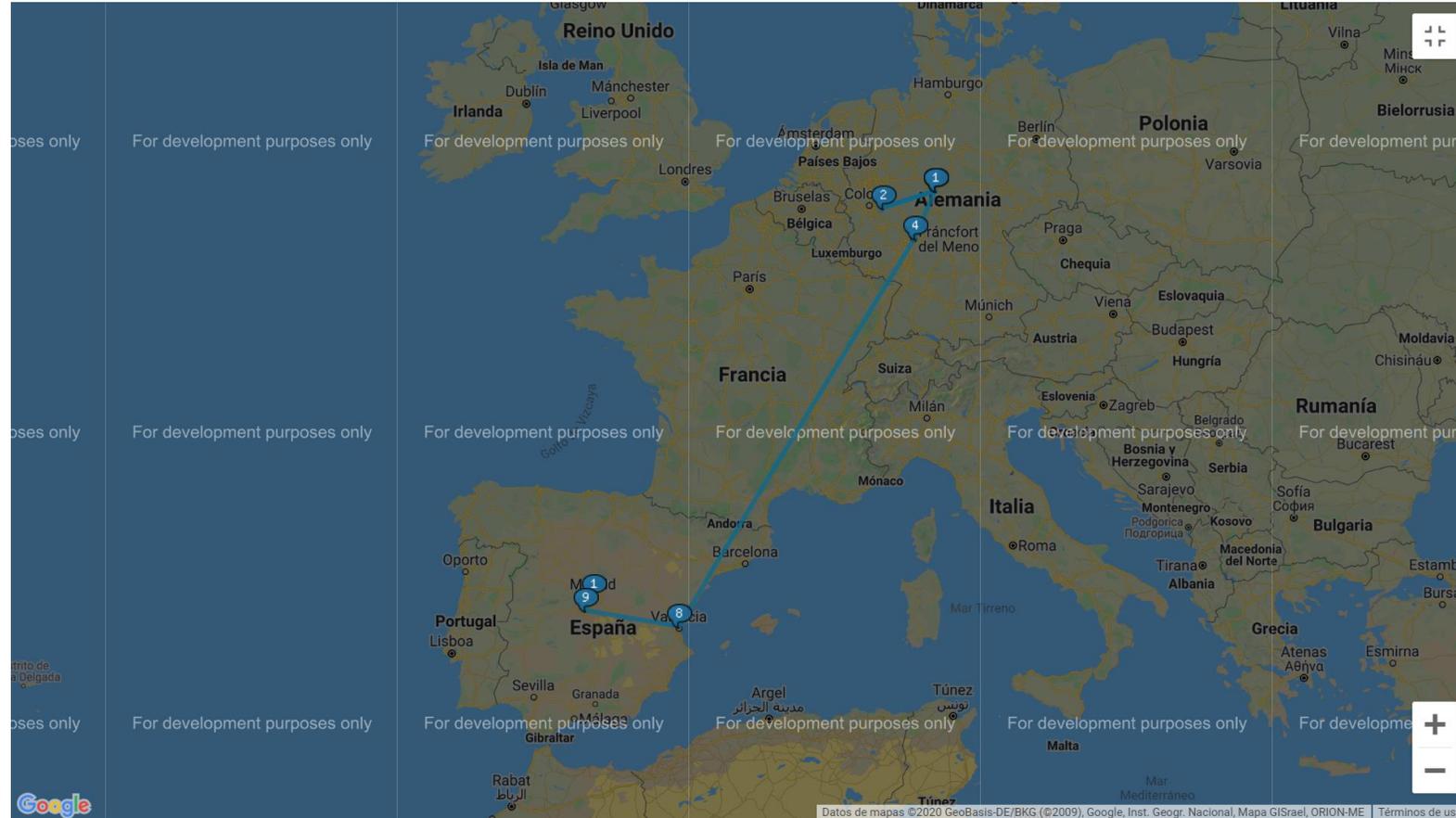


Del telégrafo a ARPANET

En ARPA (posteriormente DARPA) consiguieron **comunicar dos ordenadores** a través de una red utilizando la **conmutación de paquetes** y se continuó en un proyecto para poner en contacto todos los superordenadores de algunas Universidades e instituciones gubernamentales: **ARPANET** (1969).



Del telégrafo a ARPANET



- Tras desarrollar ARPANET, los usuarios de los ordenadores conectados a dicha red idearon un sistema para comunicarse entre ellos:
 - Dejaban un mensaje de texto en el buzón de otros usuarios.
 - Cuando se utilizaba un superordenador se verificaba si había mensajes en el buzón.
 - Había nacido el **correo electrónico o e-mail** (1971).
- **Ray Tomlinson**, ingeniero de ARPANET, escribió un programa en el que usaba la @ para unir el nombre de usuario al nombre de la máquina desde la que se enviaba el correo (raytomlinson@mit).
- No era de interés en ARPANET crear algo parecido al e-mail. Fueron las inquietudes e imaginación de un programador con algo de tiempo libre, lo que nos proporcionó una de las herramientas más importantes de la sociedad actual.



3. TCP/IP y la aparición de Internet

- El término “**Internet**” fue utilizado en **1974** por **Vint Cerf**, quien trabajaba en estandarizar el intercambio de información entre ordenadores.
- El resultado fue el **protocolo TCP/IP** (Transmission Control Protocol / Internet Protocol), el cual sigue siendo la base del intercambio de información entre ordenadores.
- ARPANET comenzó a utilizar TCP/IP en 1983, lo que se considera el **verdadero nacimiento de Internet**.
- 1989 = más de 100.000 nodos conectados a la Red.



TCP (Transmission Control Protocol):

A grandes rasgos, TCP es un protocolo de la capa de transporte que permite que dos ordenadores intercambien información garantizando la entrega de los paquetes de datos en el mismo orden en el que fueron enviados.

IP (Internet Protocol):

- Definir nombres (direcciones) que identifican unívocamente cualquier dispositivo conectado a una red.
- Se asigna una dirección de 32 bits.
- La dirección se presenta en notación decimal como 4 grupos de números separados por puntos.
- Todos los dispositivos conectados a una red requieren de una IP.

128.135.130.201

10000000.10000111.10000010.11001001

Tipos de IP:

- Públicas: visibles a través de la red global de internet. No puede haber dos IPs públicas iguales.
- Privadas: solo existen en una red local (ej.: la red wifi de casa). Dentro de una misma red local no pueden repetirse las IPs, pero sí puede haber IPs privadas iguales en diferentes redes locales.
- Estáticas: nunca cambian. Son útiles en el caso de servidores para poder acceder a ellos siempre.
- Dinámicas: cambian cada cierto tiempo. Útiles en conexiones domésticas por motivos de seguridad.

Conocer la IP de un ordenador:

- Windows: *ipconfig*
- Mac / Linux: *ifconfig*

```
Símbolo del sistema
Sufijo DNS específico para la conexión. . . :
Adaptador de LAN inalámbrica Wi-Fi:
Sufijo DNS específico para la conexión. . . : Home
Vínculo: dirección IPv6 local. . . . . : fe80::e9d7:e488:542:91ce%11
Dirección IPv4. . . . . : 192.168.1.131
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : fe80::1%11
                                           192.168.1.1
Adaptador de Ethernet Conexión de red Bluetooth:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . :
```

Configurar el router de una red doméstica:

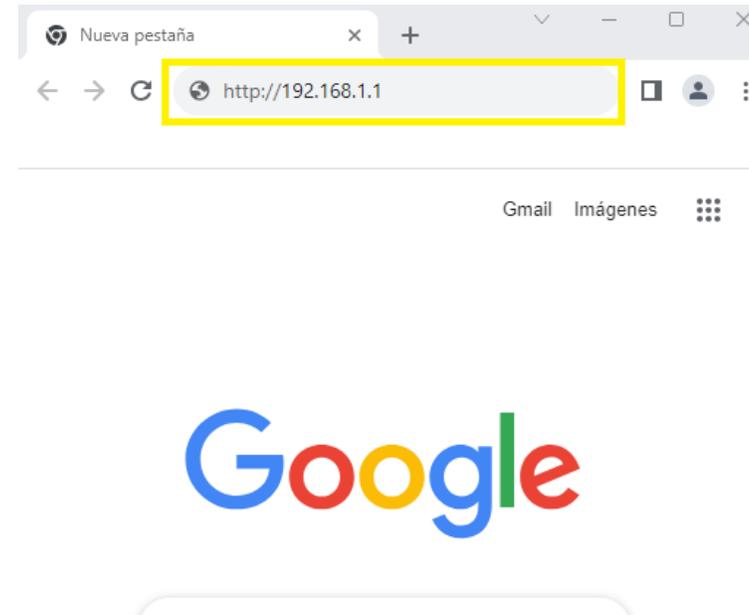
```

Símbolo del sistema
Sufijo DNS específico para la conexión. . . :
Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . . : Home
Vínculo: dirección IPv6 local. . . . . : fe80::e9d7:e488:542:91ce%11
Dirección IPv4. . . . . : 192.168.1.131
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada. . . . . : fe80::1%11
                                           192.168.1.1

Adaptador de Ethernet Conexión de red Bluetooth:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . :
    
```



- El protocolo IP original (IPv4) ya no es suficiente.
- IPv4 permitía algo más de 4.000 millones de equipos, un número a priori enorme, pero que se ha visto ampliamente superado:
 - PCs.
 - Smartphones.
 - Tablets.
 - Neveras.
 - Lavadoras.
 - Automóviles.
 - Televisiones.
 - Etc.
- Nueva versión: **IPv6** (dirección de 128 bits).

IPv4	IPv6
Deployed 1981	Deployed 1999
Address Size: 32-bit number	Address Size: 128-bit number
Address Format: Dotted Decimal Notation: 192.149.252.76	Address Format: Hexadecimal Notation: 3FFE:F200:0234:AB00:0123:4567:8901:ABCD
Prefix Notation: 192.149.0.0/24	Prefix Notation: 3FFE:F200:0234::/48
Number of Addresses: $2^{32} = \sim 4,294,967,296$	Number of Addresses: $2^{128} =$ $\sim 340,282,366,920,938,463,463,374,$ $607,431,768,211,456$

4. El nacimiento de la web

- El último paso de lo que hoy conocemos como la Web surgió gracias a Tim Berners-Lee (científico del CERN).
- Combinó algunas ideas que ya existían:
 - Uso de enlaces o hipertexto para navegar a través de la información en Internet.
 - Posibilidad de acceder a cualquier documento (una copia del mismo) almacenado en cualquier otra máquina conectada a la Red.
- Nacimiento de la **WWW (World Wide Web) - 1991**.

- **Pilares de la WWW (World Wide Web):**
 - HTTP (HyperText Transfer Protocol).
 - HTML (HyperText Markup Language).
 - DNS (Domain System Name).
 - URL (Uniform Resource Locator).



Ejemplo HTML:

```
hola.html x
1 <h1 style="color: blue">Esto es un título de primer nivel, en azul.</h1>
2 </br> <!-- Comentario: br es un salto de línea. -->
3 <p> Esto es un párrafo...</p>
4
```



Esto es un título de primer nivel, en azul.

Esto es un párrafo...

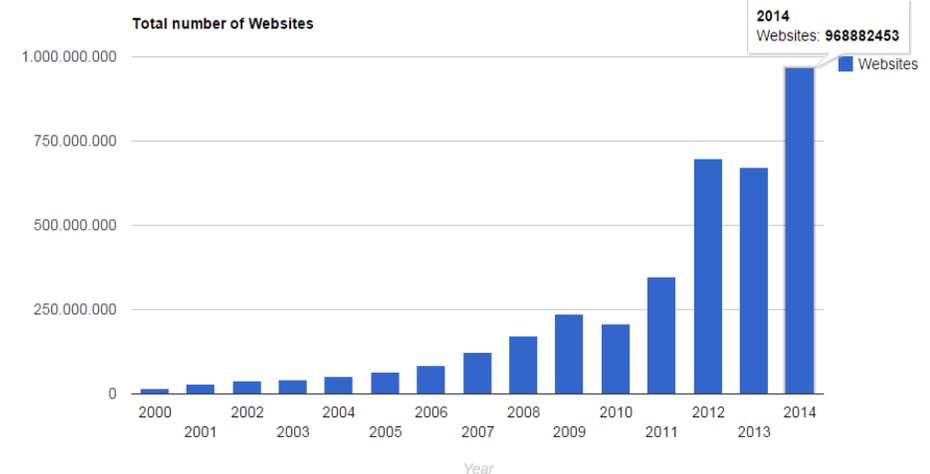
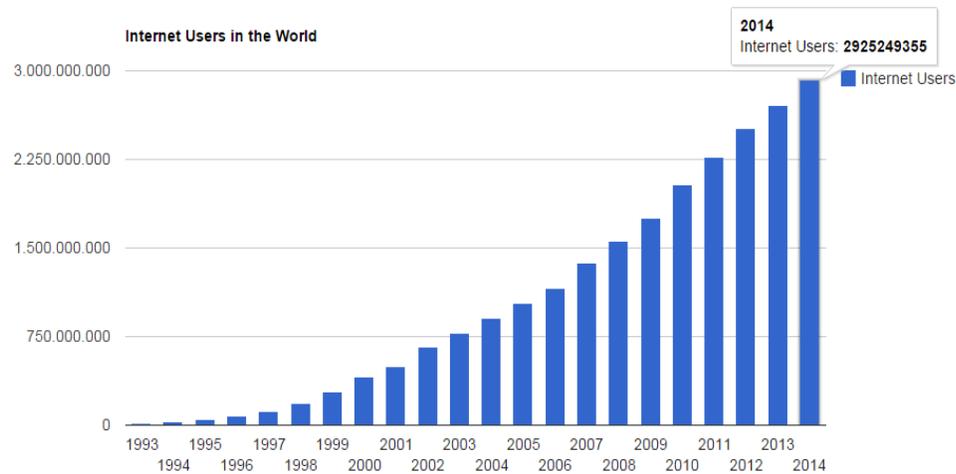
Otros conceptos:

- Web 1.0.
- Web 2.0.
- Deep web.
- Dark web.

5. La consolidación de la web

- Hasta el nacimiento de la Web, Internet era un recurso exclusivo de científicos e investigadores.
- Entre el 91 y el 97 la Web creció a un ritmo del 850% (tiempos de la burbuja de las .com). Amazon, Yahoo!, eBay, ... nacieron en esa época.
- Crecimiento continuo del NASDAQ (*National Association of Securities Dealers Automated Quotation*), bolsa de valores electrónica de USA. Alcanzó su máximo histórico en marzo de 2000.
- ¡La burbuja estalló! -> entre marzo de 2000 y octubre de 2002, más de 5 TRILLONES de dólares desaparecieron de la bolsa.
- Reajuste del mercado: lento, pero seguro.

La aparición de Google y la banda ancha en los hogares, contribuyó a impulsar esta nueva fase de crecimiento controlado.



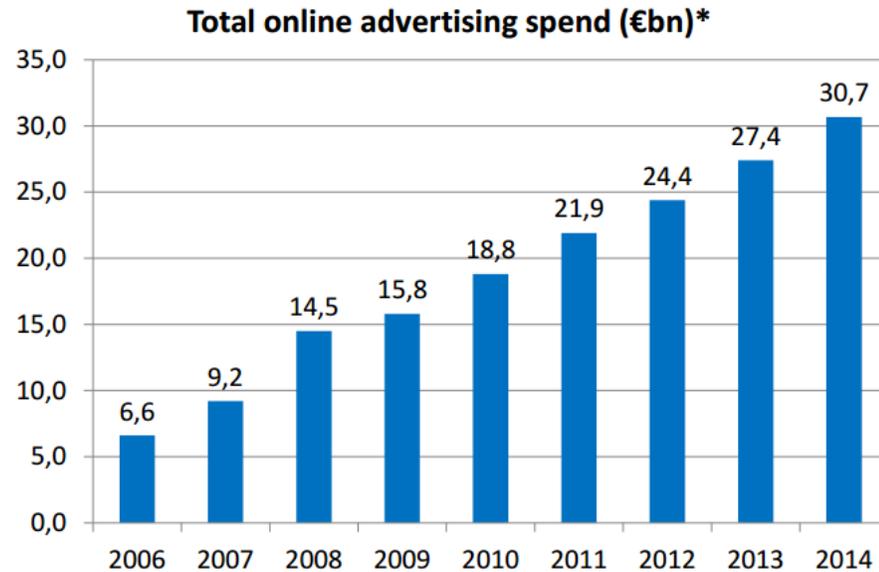
Para conseguir 50 millones de usuarios:

- La radio tardó:
 - 38 años
- La televisión tardó:
 - 13 años
- Los ordenadores personales tardaron:
 - 16 años
- Internet tardó:
 - 13 años
- Facebook tardó:
 - 1 año
- Twitter tardó:
 - 7 meses



La inversión en publicidad digital no ha parado de crecer:

La inversión en publicidad digital en Europa se ha doblado en los últimos 5 años



Datos en millones de euros



Los cambios más importantes que Internet y la web han traído pueden agruparse en 3 grandes áreas o factores:

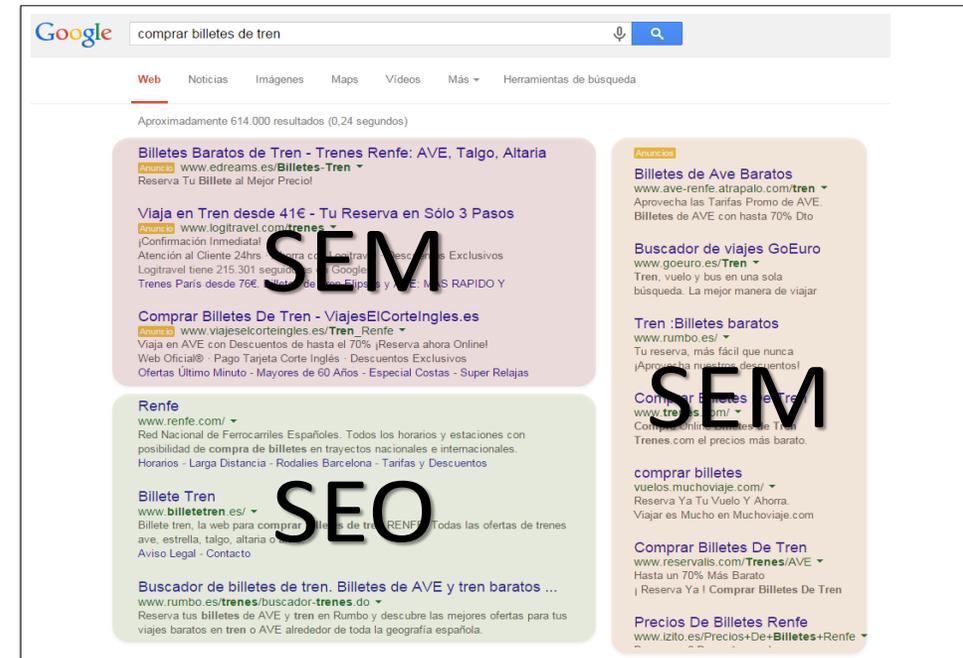
- **Nuevas plataformas:**
 - Sitios Web.
 - Microsites.
 - Analítica Web.
 - Publicidad en buscadores (SEO y SEM).
 - Publicidad en sitios Web.
 - Blogs.
 - Redes sociales.
- **El consumidor 2.0:**
 - Interconectividad.
 - Acceso a la información.
 - Filtrado.
 - Nichos de mercado.
 - On-demand.
- **Explosión de los datos:**
 - Big Data.

Nuevas plataformas:

- **Sitios Web:** inicialmente pensados como fuente de información adicional. Hoy es el elemento más importante de la imagen de cualquier empresa.
- **Microsites:** sitios Web creados con un fin concreto. Ejemplo: campañas de publicidad.
- **Analítica Web:** al navegar dejamos rastro de datos, cifras, clics, etc. La analítica Web trata de comprender el significado de esa información y traducirla a información relevante.

Nuevas plataformas:

- **Publicidad en buscadores:** publicidad enfocada al público de interés. El mejor lugar es en un motor de búsqueda, cuando el usuario está más predispuesto a escuchar un mensaje, siempre que esté relacionado con lo que busca.
 - **SEO** (Search Engine Optimization).
 - **SEM** (Search Engine Marketing).



Nuevas plataformas:

- **Publicidad en sitios Web:** otro tipo de publicidad contextual, relacionada con la temática del contenido del sitio. Herramienta de Google: AdSense.
- **Blogs:** blogosfera -> espacio virtual constituido por todos los blogs existentes en la Web. Es un altavoz de la opinión de la gente sobre cualquier tema.



El consumidor 2.0:

- **Interconectividad:** la tecnología ha habilitado la comunicación entre personas situadas en cualquier lugar del mundo.
- **Acceso a la información:** la tecnología ha universalizado el nivel de acceso a la información en cualquier parte del mundo.
- **Filtrado:** el consumidor digital necesita nuevas formas de agregar, categorizar y filtrar semejante cantidad de información -> software que permite excluir mensajes comerciales a voluntad.
- **Nichos de mercado:** la cantidad de contenidos disponibles permite a los consumidores satisfacer sus intereses y gustos, formando comunidades online cada vez más centradas en temas más concretos.
- **On-demand:** la tecnología ha provocado que nos encontremos en una sociedad ubicua, donde los consumidores pueden satisfacer sus necesidades en cualquier momento y lugar. Y cuanto más rápido consiguen lo que quieren, más rápido quieren conseguirlo.

Explosión de los datos:

- Big Data: muchas definiciones. No hay un consenso: "aquello que no puedes almacenar en una hoja de Excel".
- Auge del Big Data en la sociedad actual:
 - El Internet de las Cosas ha dotado a coches, neveras, lavadoras, etc. De la tecnología necesaria para interconectarse con otros dispositivos digitales.
 - Llegamos a producir 2.5 quintillones de bytes en forma de imágenes digitales, comentarios en RRSS, vídeos, sensores, GPS, etc.
 - Se necesitan nuevas herramientas y métodos para procesar estos datos: algoritmos, procesadores y aplicaciones software.
 - Se recaudaron 17 billones de dólares en el 2015 con los productos relacionados con el Big Data.
 - La empleabilidad relacionada creció un 123%.
 - Nueva industria relacionada con el almacenamiento y procesamiento de grandes volúmenes de datos.

Nuevos perfiles profesionales:

- Community Manager.
- Digital Media Planner/Strategist.
- Brand Reputation Specialist.
- Social Media Planner/Analyst.
- SEO Expert.
- SEM Expert.
- Etc.

Community Manager:

- Creación de contenidos para blogs y medios sociales en general, boletines y materiales de comunicación.
- Creación, gestión y desarrollo de la presencia online de la compañía a través de blogs, Twitter, Facebook, y otras herramientas sociales de importancia estratégica.
- Gestión de las solicitudes de los medios y desarrollo de las relaciones con los medios de comunicación del sector.
- Diseño y desarrollo de campañas de comunicación con su consecuente medición de los resultados.
- Relaciones con los clientes a la hora de responder a las preguntas hechas a través de correo electrónico, Twitter, Facebook, etc.
- Creación de estrategias de marketing y comunicación online.
- Medición, mediante analíticas web, de las relaciones y búsqueda de elementos de mejora a través de pruebas y de nuevas iniciativas.
- Desarrollo del modelo de negocio y de ventas, dependiendo de la experiencia.

Digital Media Planner/Strategist:

- Examinar los objetivos de las actividades online de la compañía y sugerir otros posibles objetivos.
- Examinar el tráfico de la web de la empresa y formular estrategias y acciones para incrementarla.
- Estudiar cuidadosamente las páginas de destino y dar consejos para su mejora.
- Evaluar el diseño general del sitio web y, de ser necesario, proponer revisiones.
- Evaluar el contenido y la redacción del sitio web.
- Examinar las funciones de búsqueda y navegación del sitio web.
- Analizar y monitorizar los instrumentos de la interacción directa con los usuarios (compra, generación de leads, etc.) a través de los cuales los visitantes llegan a tomar decisión de acción dentro del sitio (carrito de la compra, formulario de contacto, etc.).

Brand Reputation Specialist:

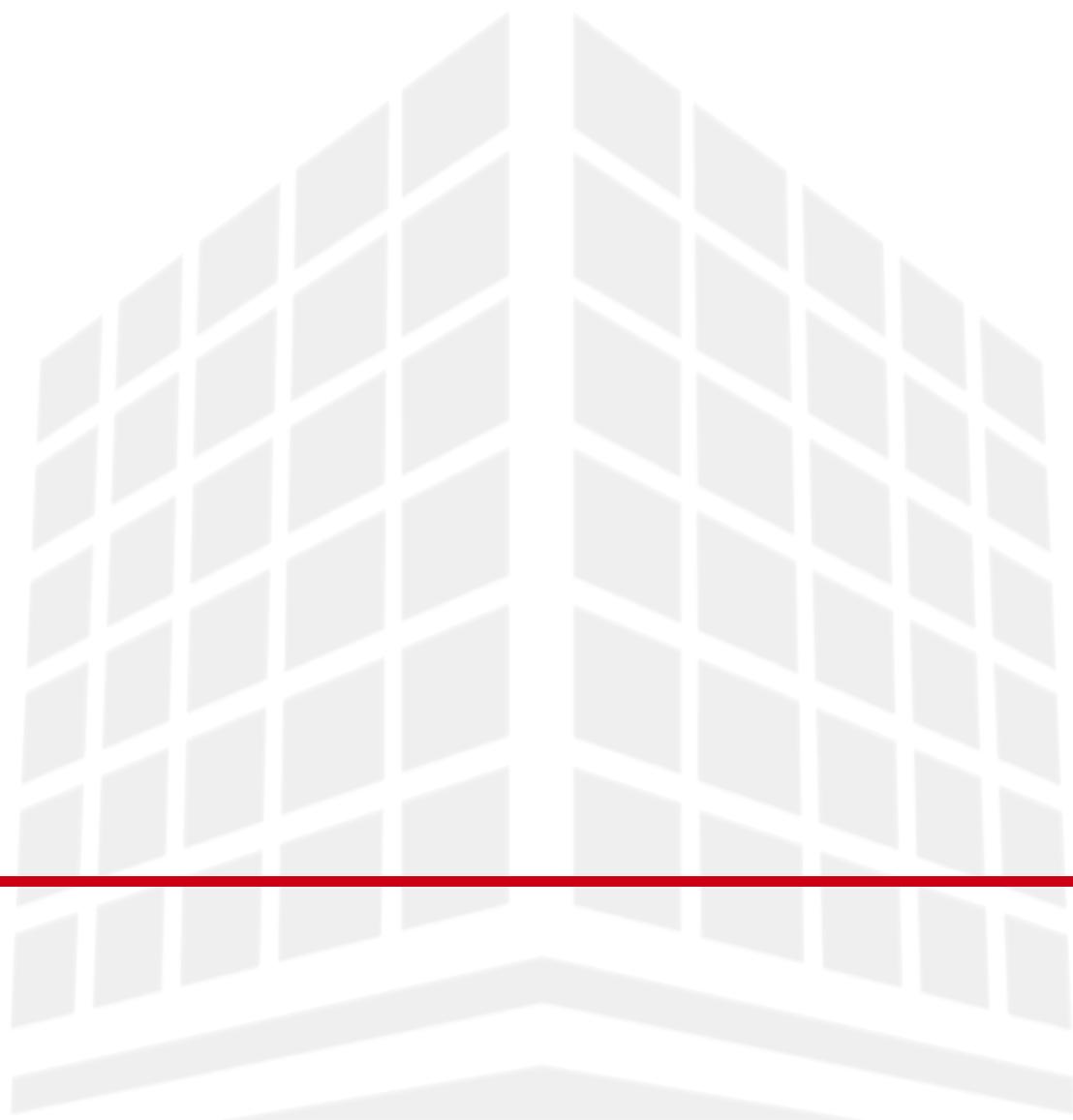
- Estudiar el ambiente (Internet, redes sociales, etc.) y sus reglas.
- Compartir con los usuarios la información (empresarial, de productos, etc.) para que puedan utilizarla.
- Poner a disposición la información empresarial y participar activamente en las conversaciones y debates acerca de la compañía y de la marca.

Social Media Planner/Analyst:

- Controlar, en las redes sociales, las conversaciones relacionadas con las marcas y productos de la competencia.
- Dar soporte a la implementación y planificación de todas las iniciativas de promoción de la empresa en los medios de comunicación sociales.
- Llevar a cabo un análisis mensual, trimestral y anual de los medios sociales con el fin de optimizar las campañas y la planificación.
- Crear y gestionar un calendario de contenidos para todos los perfiles de los medios sociales de la marca de la empresa, coordinándose con el director del marketing digital.
- Identificar continuamente nuevas oportunidades estratégicas de marketing en las redes sociales.
- Supervisar las actividades de marketing digital y monitorizar el grado de influencia de los medios de comunicación sociales.
- Coordinarse con otros departamentos para garantizar una óptima comunicación online y que distinga a la marca.
- Controlar todos los sitios para garantizar la coherencia de los mensajes en las diferentes redes.
- Preparar informes para actualizar a los demás responsables del marketing sobre las métricas y las estadísticas de uso de las redes sociales por parte de la empresa.
- Supervisar la presencia de la competencia en las redes sociales.
- Buscar nuevos blogs y canales para identificar nuevas tendencias.
- Contribuir en el desarrollo del blog empresarial y en el de los contenidos para las redes sociales.
- Monitorizar los canales sociales incluso fuera del horario tradicional de oficina.

SEO/SEM Expert:

- Poner en práctica las mejores estrategias y metodologías de marketing online, coordinando los múltiples canales de marketing para maximizar la visibilidad del sitio web de la empresa, garantizar mensajes coherentes y proporcionar una excelente experiencia al cliente.
- Llevar a cabo acciones cotidianas relacionadas con el marketing en los motores de búsqueda.
- Planificar, analizar e incrementar apropiadamente la lista de palabras clave.
- Crear anuncios de alta calidad para direccionar el tráfico hacia el sitio web de la empresa.
- Actualizar los informes de rendimiento de las campañas de marketing online y de sus bases de datos.
- Proponer estrategias que den soporte a las campañas de SEM de pago.
- Gestionar el presupuesto y analizar el rendimiento de las campañas con el fin de identificar nuevas oportunidades que permitan optimizar la eficacia del mismo presupuesto.
- Realizar análisis detallados para identificar los datos relativos a los puntos de vista y al rendimiento de la campaña.
- Realizar una investigación de la competencia.
- Gestionar las relaciones con los gestores de los motores de búsqueda.



Tecnología y Uso de Internet

Tema 3: Maquetación web – HTML

1. ¿Qué es HTML?

2. Elementos HTML:

- Etiquetas básicas.
- Encabezados.
- Párrafos.
- Imágenes.
- Enlaces.
- Listas.
- Tablas.
- Contenedores.
- Atributos.

3. HTML 5:

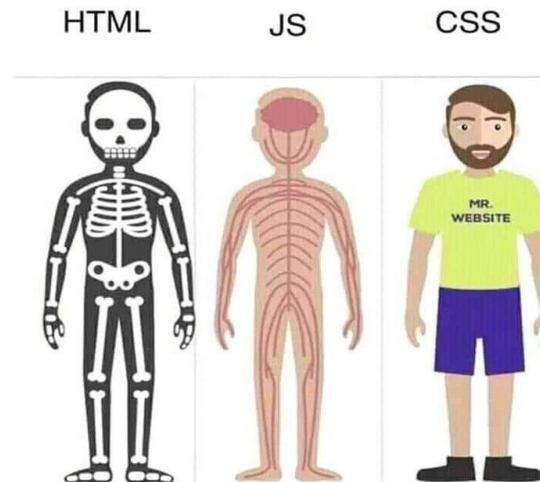
- Estructura.
- Etiquetas.

4. Formularios HTML:

- ¿Qué es un formulario?
- Construcción de formularios.

I. ¿Qué es HTML?

- HTML (HyperText MarkupLanguage).
- Es un lenguaje de marcas para la representación de contenido en la web.
- Se nutre de otros lenguajes para enriquecer cómo mostrar la información o mejorar la funcionalidad de la página web.



2. Elementos HTML

- Todo documento con extensión .html debe contener las etiquetas `<html>` y `</html>`.
- En algunas ocasiones, se especifica antes de la etiqueta `<html>` el tipo de documento para facilitar al navegador su visualización.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Título del documento</title>  
  </head>  
  <body>  
  
  </body>  
</html>
```

Doc -> Documento
Type -> Tipo html

- No imprescindible, pero sí recomendable es incluir en el html una cabecera, la cual brindará información adicional sobre la página al navegador.
- La cabecera se declara con las etiquetas `<head>` `</head>`.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título del documento</title>
  </head>
  <body>

  </body>
</html>
```

Aquí se puede alojar estilos, scripts, el título / codificación de la página.

Dentro de las etiquetas <head> nos podemos encontrar con (I):

- Las etiquetas **<meta>** que definen cómo tratar los metadatos (información acerca de los datos) del documento html por parte del navegador. Algunos atributos para dicha etiqueta son:
 - lang: Idioma del contenido de la página.
 - charset: Conjunto de caracteres con el que se codifica la página.
 - name = "keywords" content = "palabraClave1,..." Permite a los administradores la posibilidad de definir palabras clave para el buscador. Las keywords son aquellos criterios a los que responde un buscador para ofrecerle al usuario páginas HTML como respuesta.
 - name = "robots". Indica al buscador si debe indexar o no la página. Puede contener como valor del atributo content = " " los siguientes valores para especificar:
 - index: Si se quiere indexar (Por defecto, aparece seleccionada esta opción).
 - noindex: Si no se quiere indexar (No mostrar en los resultados de búsqueda).

SEO

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Una de las principales referencias de consulta: <https://www.w3schools.com/html/default.asp>

- Dentro de las etiquetas `<head>` nos podemos encontrar con (II):
 - Las etiquetas `<title>` y `</title>` que dan un título a la página (Aparece en la pestaña del navegador).
 - Las etiquetas `<style>` y `</style>` donde se definen estilos (“Código CSS”) para los elementos html del archivo.
 - Las etiquetas `<script>` y `</script>` las cuales contienen funciones Javascript que realizan alguna función sobre los elementos html / eventos del usuario con la página / información recogida en el html.
 - Las etiquetas `<link>` y `</link>` para importar archivos CSS, JS, añadir un favicon, etc.

Ejemplo de lo que nos podemos encontrar dentro de la cabecera:

Metadatos (Info para el navegador)
Título del documento

Estilos (CSS)

Script JS (Javascript)

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
    <script>
      alert("Hola chicos de Lenguaje de Marcas");
    </script>
  </head>
  <body>
    <h1>Esto es un título</h1>
    <h2>Esto es un título con - énfasis que el anterior</h2>
    <p>Esto es un párrafo</p>
  </body>
</html>

```

- En la sección body se mostrará el contenido de la información del documento .html (párrafos, imágenes, vídeos, enlaces, ...).
- Se declara con las etiquetas **<body>** **</body>**.
- Sólo puede haber una sección body en el html.
- Esta sección está enfocada al usuario.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título del documento</title>
  </head>
  <body>
  </body>
</html>
```

Aquí se puede alojar contenedores, títulos, imágenes, tablas, etc..

Elementos HTML: encabezados

- Los encabezados o títulos son declarados con la etiqueta `<hX>` donde X es un número comprendido entre 1 y 6.
- Cuanto mayor sea el número del título (número X) + pequeña será la fuente del texto.
- Ejercicio: ampliar el ejercicio anterior incluyendo un título de cada nivel.

Declaración: `<hX> Texto del título </hX>`

X es el mismo número

Títulos del documento (Máx. 6)

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Titulo del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
    <script>
      alert("Hola chicos de Lenguaje de Marcas");
    </script>
  </head>
  <body>
    <h1>Esto es un título</h1>
    <h2>Esto es un título con - énfasis que el anterior</h2>
    <h3>Esto es un título con - énfasis que el anterior</h3>
    <h4>Esto es un título con - énfasis que el anterior</h4>
    <h5>Esto es un título con - énfasis que el anterior</h5>
    <h6>Esto es un título con - énfasis que el anterior</h6>
    <p>Esto es un párrafo</p>
    <p>Y este otro párrafo</p>
  </body>
</html>

```

- Los párrafos se declaran utilizando las etiquetas `<p>` y `</p>`.
- No hay límite de párrafos.

Declaración: `<p>` Texto del párrafo `</p>`

Párrafos del documento

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
    <script>
      alert("Hola chicos de Lenguaje de Marcas");
    </script>
  </head>
  <body>
    <h1>Esto es un título</h1>
    <h2>Esto es un título con - énfasis que el anterior</h2>
    <h3>Esto es un título con - énfasis que el anterior</h3>
    <h4>Esto es un título con - énfasis que el anterior</h4>
    <h5>Esto es un título con - énfasis que el anterior</h5>
    <h6>Esto es un título con - énfasis que el anterior</h6>
    <p>Esto es un párrafo</p>
    <p>Y este otro párrafo</p>
  </body>
</html>

```

- Dentro de las etiquetas `<p>` y `</p>` puede haber otras etiquetas (I):
 - Las etiquetas `` texto `` o `` texto ``, ponen en **negrita** el contenido del párrafo entre dichas etiquetas.
 - Las etiquetas `<i>` texto `</i>` o `` texto ``, *enfatan* el contenido del párrafo entre dichas etiquetas y lo ponen en cursiva.
 - Las etiquetas `<small>` texto `</small>`, disminuyen el tamaño del contenido del párrafo entre dichas etiquetas.
 - Las etiquetas `` texto ``, ~~tachan~~ el contenido del párrafo entre dichas etiquetas.

- Ejemplo de las anteriores etiquetas que aparecen en párrafos:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
    <script>
      alert("Hola chicos de Lenguaje de Marcas");
    </script>
  </head>
  <body>
    <p>Esto es un párrafo con la palabra <strong> informática </strong> en negrita</p>
    <p>Esto es un párrafo con la palabra <em> informática </em> enfatizada</p>
    <p>Esto es un párrafo con la palabra <small> informática</small> más pequeño de lo normal</p>
    <p>Esto es un párrafo con la palabra <del> informática</del> tachada</p>
  </body>
</html>

```

Negrita ← ** informática **
Enfatizado ← * informática *
Cursiva ← *<i> informática</i>*
Pequeño ← <small> informática</small>
Tachado ← ~~ informática~~

- Dentro de las etiquetas `<p>` y `</p>` puede haber otras etiquetas (II):
 - Las etiquetas `_{` texto `}`, bajan el contenido del párrafo entre dichas etiquetas por debajo del nivel normal del texto
 - Las etiquetas `^{` texto `}`, sube el contenido del párrafo entre dichas etiquetas por encima del nivel normal del texto.
 - Las etiquetas `<u>` texto `</u>`, subrayan el contenido del párrafo entre dichas etiquetas.
 - Las etiquetas `<mark>` texto `</mark>`, marcan el contenido del párrafo entre dichas etiquetas.
 - Las etiquetas `` texto ``, destacan una sección del párrafo. Como valor para este atributo, van declaraciones CSS.

- Para poder insertar una imagen en nuestro HTML debemos hacer uso de la etiqueta ****
- Las imágenes en un documento HTML van dentro de la sección **<body>**
- La etiqueta **** tiene los siguientes atributos (todos los valores de estos atributos entre " "):
 - **src** = viene de source, aquí se indica el enlace/ruta de la foto.
 - **alt** = texto alternativo por si no se puede visualizar la imagen.
 - **width** = para especificar la anchura de la foto.
 - **height** = para especificar la altura de la foto.

- Ejemplo de uso de la etiqueta :

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Aquí va la imagen</h1>
    
  </body>
</html>

```

Link donde se aloja la foto

Texto alternativo

- Para poder insertar enlaces a otros sitios (webs o rutas dentro de nuestro S.O) debemos de utilizar la etiqueta `<a> `.
- Dentro de esta etiqueta podemos poner párrafos, imágenes, etc.
- Dicha etiqueta tiene un atributo importante denominado "href" que será el lugar donde indiquemos el hipervínculo a donde deseamos que se nos redirija cuando pulsemos sobre él.

`Texto sobre el que se pincha `

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Aquí va el link</h1>
    <a href="https://www.ifp.es/">Visita la página de IFP</a>
  </body>
</html>

```

- Las listas se crean para reflejar un conjunto de elementos en un documento HTML en formato de lista.
- Existen diferentes tipos de listas.

Elementos HTML: listas

- Las listas ordenadas se crean con las etiquetas `` y ``.
- Entre estas, colocaremos mediante las etiquetas `` Nombre del elemento de la lista `` para generar una lista ordenada.

Declaración: `` `` Elemento X `` ... ``

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista ordenada</h1>
    <ol>
      <li>Mates</li>
      <li>Lengua</li>
      <li>CC.Sociales</li>
      <li>Recreo</li>
    </ol>
  </body>
</html>

```

Elementos lista

- Si deseamos cambiar el índice de ordenación debemos añadir el atributo “type” dentro de la etiqueta .
- Valores del atributo type para los índices:
 - A -> letras del abecedario mayúsculas.
 - a -> letras del abecedario minúsculas.
 - 1 -> números naturales.
 - i -> números romanos.
 - I -> números romanos en mayúsculas.

Declaración: <ol type = ? > Elemento X ...

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista ordenada</h1>
    <ol type = A>
      <li>Mates</li>
      <li>Lengua</li>
      <li>CC.Sociales</li>
      <li>Recreo</li>
    </ol>
  </body>
</html>

```

Elementos lista

- Si deseamos establecer un orden personalizado para nuestro índice (decir a partir de qué número deseamos que empiece) debemos usar el atributo "start" dentro de la etiqueta .

Declaración: <ol start = ?> Elemento X ...

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista ordenada</h1>
    <ol start = 4>
      <li>Mates</li>
      <li>Lengua</li>
      <li>CC.Sociales</li>
      <li>Recreo</li>
    </ol>
  </body>
</html>

```

Elementos lista

- Si deseamos establecer el orden inverso del índice de la lista debemos usar el atributo "reversed" dentro de la etiqueta .

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista ordenada</h1>
    <ol type = A reversed>
      <li>Mates</li>
      <li>Lengua</li>
      <li>CC.Sociales</li>
      <li>Recreo</li>
    </ol>
  </body>
</html>

```

Declaración: <ol reversed> Elemento X ...

Elementos lista

- Las listas **sin índices** se crean con las etiquetas `` y ``.
- Entre estas, colocaremos mediante las etiquetas `` Nombre del elemento de la lista `` para generar una lista sin índices.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista no ordenada</h1>
    <ul>
      <li>Mates</li>
      <li>Lengua</li>
      <li>CC.Sociales</li>
      <li>Recreo</li>
    </ul>
  </body>
</html>

```

Declaración: `` `` Elemento X `` ... ``

Elementos lista

- Las listas de definiciones son aquellas en las que aparecen términos/conceptos y acto seguido las definiciones de estos.
- Se declaran con las etiquetas:
 - `<dl>` y `</dl>` para declarar la lista.
 - `<dt>` y `</dt>` para declarar el término.
 - `<dd>` y `</dd>` para declarar la definición.

Ejemplo de lista de definición

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista no ordenada</h1>
    <dl>
      <dt>Mates</dt>
      <dd> Asignatura donde se estudian operaciones algebraicas y trigonométricas </dd>
      <dt>Lengua</dt>
      <dd> Asignatura donde se estudia el análisis sintáctico y gramatical </dd>
    </dl>
  </body>
</html>

```

- Se pueden insertar listas (ordenadas o no) dentro de otras listas (ordenadas o no). A esto se le llaman listas anidadas (una lista dentro de otra).
- ¿Cómo se declaran?
 - Se declara el tipo de la lista padre (o).
 - A continuación, se declara un elemento en la lista padre () que ejercerá como padre de la segunda lista (lista hija) pero no se cierra con () aún.
 - Se declara la lista hija (o), cuyo padre es el elemento anterior que no hemos cerrado.
 - Una vez se termina de declarar la lista hija (o), se cierra el elemento que hacía como padre ().
 - Se siguen declarando elementos () o se cierra la lista padre (o).

- Ejemplo lista anidada:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista ordenada</h1>
    <ol start = 4>
      <li>Mates
        <ul>
          <li>Trigonometría</li>
          <li>Matrices</li>
          <li>Algebra</li>
          <li>Conjuntos</li>
        </ul>
      </li>
      <li>Lengua</li>
      <li>CC.Sociales</li>
      <li>Recreo</li>
    </ol>
  </body>
</html>

```

Elemento que hace como padre de la lista hija

Lista hija

Lista Padre

- Dentro de las listas podemos insertar enlaces en los elementos para redirigirnos hacia una página que tenga relación con dicho elemento (el elemento se vuelve un enlace).

Declaración: ` Elemento X `

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Lista ordenada</h1>
    <ol start = 4>
      <li><a href="https://es.wikipedia.org/wiki/Matem%C3%A1ticas">Mates</a></li>
      <li>Lengua</li>
      <li>CC.Sociales</li>
      <li>Recreo</li>
    </ol>
  </body>
</html>
```

- Las tablas son elementos html que se declaran con la etiqueta **<table>** y **</table>**.
- Puede haber varias tablas dentro de un html.
- Es necesario que al menos haya una fila en una tabla.
- Dentro de una tabla puede haber varios elementos html:
 - Imágenes.
 - Enlaces.
 - Texto.
 - Otras tablas
 - Etc.

- Se declara la tabla con las etiquetas `<table>` y `</table>`.
- Se declaran las filas de la tabla, utilizando las etiquetas `<tr>` y `</tr>`.
- Dentro de las etiquetas anteriores, debemos incluir los encabezados de las tablas en la primera fila si los tuviera con las etiquetas `<th>` y `</th>`.
- Para insertar celdas, debemos utilizar las etiquetas `<td>` y `</td>` dentro de `<tr>` y `</tr>`, como si fuera una lista.

Continente	Porcentaje
Asia	48,2 %
Europa	18,0 %
América Latina	10,2 %
África	9,8 %
América del Norte	9,3 %
Oriente Medio	3,7 %
Oceanía	0,8 %

- Ejemplo de tabla html:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
  </head>
  <body>
    <h1>Tabla Escuderías/Pilotos</h1>
    <table>
      <tr>
        <th>Mercedes</th>
        <th>McLaren</th>
        <th>Ferrari</th>
      </tr>
      <tr>
        <td>Hamilton</td>
        <td>Sainz</td>
        <td>Leclerc</td>
      </tr>
      <tr>
        <td>Bottas</td>
        <td>Norris</td>
        <td>Vettel</td>
      </tr>
    </table>
    <h1>Tabla Escuderías/Jefe equipo</h1>
    <table>
      <tr>
        <th>Mercedes</th>
        <th>McLaren</th>
        <th>Ferrari</th>
      </tr>
      <tr>
        <td>Totto Wolf</td>
        <td>Andreas Seidl</td>
        <td>Binotto</td>
      </tr>
    </table>
  </body>
</html>

```

Diagram annotations:

- Tabla 1**: Points to the first table structure.
- Tablas**: Points to the first table's rows.
- Cabeceras de la tabla**: Points to the header row of the first table.
- Celdas de la fila**: Points to the data cells of a row in the first table.
- Tabla 2**: Points to the second table structure.

Elementos HTML: contenedores

- Los contenedores sirven para dividir regiones de nuestro html.
- Se declaran con las etiquetas `<div>` y `</div>`.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h2 {color : red;}
      #CONTENEDOR1 {background-color:red;}
      #CONTENEDOR2 {background-color : green;}
      .parrafo {color : blue;}
    </style>
  </head>
  <body>
    <h1 class = "parrafo">Esto es un título que pertenece a la clase párrafo </h1>
    <div id = "CONTENEDOR1">
      <p class = "parrafo">Esto es un párrafo con la palabra <sub> informática </sub> por debajo del párrafo</p>
      <p class = "parrafo">Esto es un párrafo con la palabra <sup> informática </sup> por encima del párrafo</p>
      <p class = "parrafo">Esto es un párrafo con la palabra <mark> informática</mark> subrayada con fluorescente</p>
    </div>
    <div id = "CONTENEDOR2">
      <p class = "parrafo">Esto es un párrafo con la palabra <sub> informática </sub> por debajo del párrafo</p>
      <p class = "parrafo">Esto es un párrafo con la palabra <sup> informática </sup> por encima del párrafo</p>
      <p class = "parrafo">Esto es un párrafo con la palabra <mark> informática</mark> subrayada con fluorescente</p>
    </div>
  </body>
</html>

```

Región CONTENEDOR1

Región CONTENEDOR2

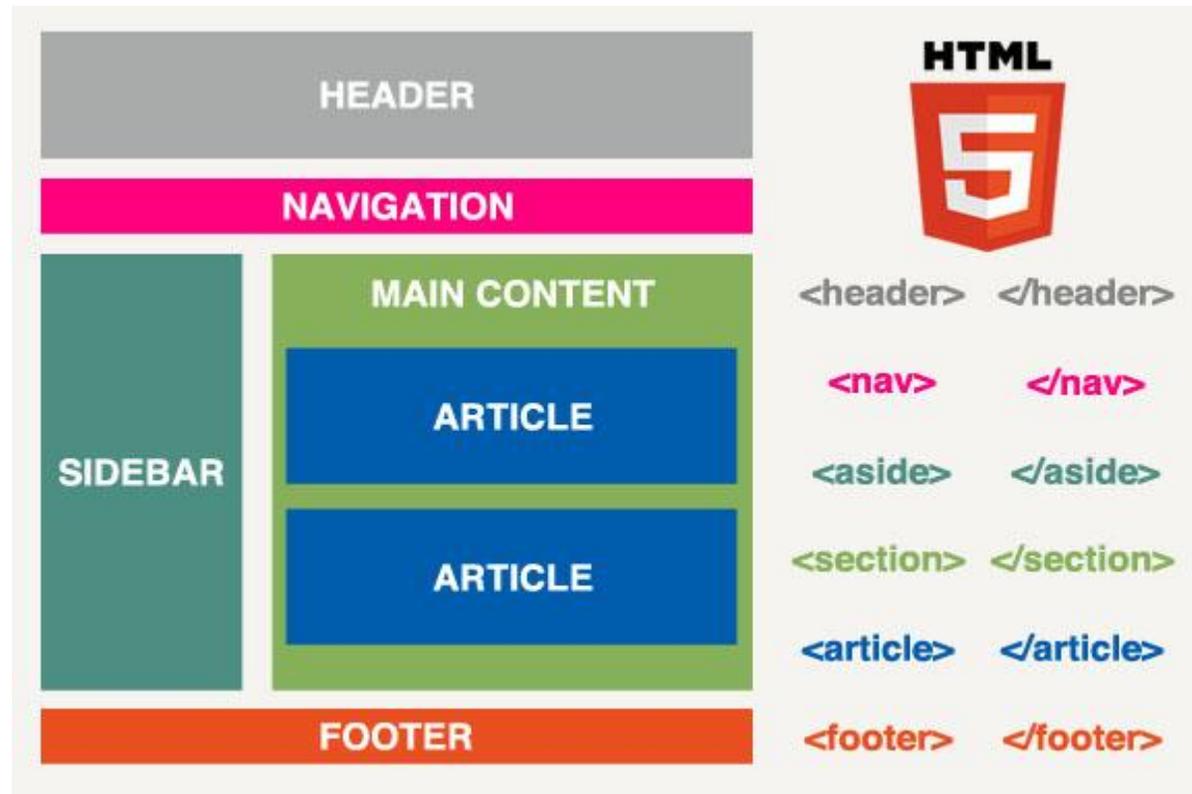
Los elementos html tienen estos atributos comunes:

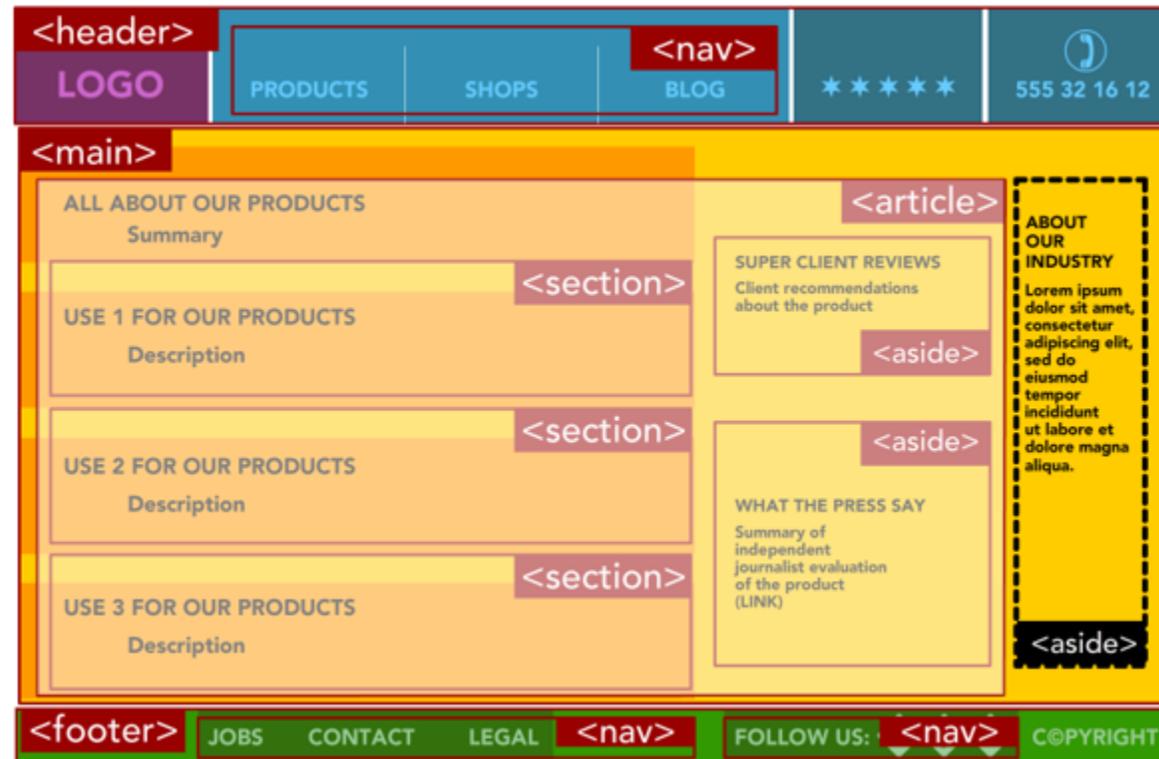
- **id**: identificador único de un elemento html.
- **class**: agrupación de elementos bajo una determinada clase.
- **style**: para definir estilos en CSS al elemento html.
- **name**: sirve para dar un nombre al elemento html que se enviará al servidor.

Echad un vistazo a la diapositiva anterior.

5. HTML 5

- HTML5 incorpora una serie de etiquetas que permiten definir la estructura de un archivo HTML:
 - **<header>** cabecera.
 - **<nav>** barra de navegación.
 - **<aside>** módulo lateral.
 - **<main>** principal.
 - **<article>** artículo.
 - **<section>** sección.
 - **<footer>** pie de página.
- No son obligatorias para construir una página HTML.





Fuente : <https://www.semrush.com/blog/semantic-html5-guide/>

- Barras de navegación `<nav>` `</nav>`

```
<nav> <ol><li><a href="#"> ? </a> ... </li>...</ol></nav>
```

```
<nav> <ul><li><a href="#"> ? </a> ... </li>...</ul></nav>
```

```
<nav> <a href="#"> ? </a> ... </nav>
```

Enlaces

```
<html>
  <body>
    <h1>Elemento nav</h1>
    <nav>
      <a href="/html/">HTML</a> |
      <a href="/css/">CSS</a> |
      <a href="/js/">JavaScript</a> |
      <a href="/css/">PHP</a> |
      <a href="/java/">Java</a> |
      <a href="/python/">Python</a>
    </nav>
    <p>Ejemplo de nav con enlaces</p>
  </body>
</html>
```

NAV

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Elemento nav</h1>
    <nav>
      <ol>
        <li> Cliente
          <a href="/html/">HTML</a> |
          <a href="/css/">CSS</a> |
          <a href="/js/">JavaScript</a> |
        </li>
        <li> Servidor
          <a href="/css/">PHP</a> |
          <a href="/java/">Java</a> |
          <a href="/python/">Python</a>
        </li>
      </ol>
    </nav>
    <p>Ejemplo de nav con enlaces</p>
  </body>
</html>
```

Lista

- Para poner un texto en formato dirección de correo postal podemos hacerlo mediante las etiquetas `<address>` y `</address>`.
- Se declara de la siguiente forma:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Prueba de elemento iframe</h1>
    <address>
      <p>Author: Cristian Gómez</p>
      <p>Dir: C/Joaquín Camarillo</p>
      <p>Telf: +34....</p>
    </address>
  </body>
</html>
```

- Para delimitar con una línea párrafos podemos usar la etiqueta `<hr>`.
- Se declara de la siguiente forma:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Prueba de elemento hr</h1>
    <p>Author: Cristian Gómez</p>
    <hr>
    <p>Dir: C/Joaquín Camarillo</p>
    <hr>
    <p>Telf: +34....</p>
  </body>
</html>
```

- Los iframes sirven para incorporar una página/documento a nuestro html `<iframe></iframe>`.
- Se declaran de la siguiente forma:

`<iframe src = "ruta/link" title="x"></iframe>`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      h1 {color : green;}
      h2 {color : red;}
      p {color: blue;}
    </style>
  </head>
  <body>
    <h1>Prueba de elemento iframe</h1>
    <iframe src="https://www.nationalgeographic.com.es/animales" title="Animales"></iframe>
  </body>
</html>
```

- Al igual que las imágenes, también se les puede añadir los atributos width and height para modificar la anchura y altura del iframe.
- Ejemplo:

```
<iframe src = "ruta" tittle="x" width="100%" height="200"></iframe>
```

- Ejercicio: utilizar estos atributos para acondicionar el iframe que habéis añadido.

- La manera de incorporar un archivo de audio es a través de las etiquetas **<audio controls></audio>**.
- Se declaran de la siguiente forma:

```
<audio controls src = "ruta/link"></audio>
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Título del documento</title>
6 </head>
7   <style>
8     h1 {color : green;}
9     h2 {color : red;}
10    p {color: blue;}
11  </style>
12 </head>
13 <body>
14   <h1>Prueba de elemento audio</h1>
15   <iframe src="https://www.nintendo.es/Juegos/Portal-de-Nintendo/Portal-para-Super-Mario/Portal-para-Super-Mario-627604.html" title="Super Mario" width="100%" height="600"></iframe>
16   <audio controls src = "C:/Users/Cristian/Dropbox/Docencia IFP/2020-2021/Lenguaje de Marcas/EjerciciosHTML/Tema 1/EjercicioIframe/audioSuperMario.mp3"></audio>
17 </body>
18 </html>
```

- Si no ponemos el atributo controls, no aparecerá la botonera para controlar el audio.



- La etiqueta audio también cuenta con los siguientes atributos:
 - autoplay: Reproduce automáticamente el audio.
 - loop: Reproduce en bucle el audio.
 - mute: Silencia el audio (Solo tiene sentido si lo colocamos después de autoplay).
 - preload: Inicia la descarga y almacenamiento en el búfer del audio.

- La manera de incorporar un archivo de audio es a través de las etiquetas **<video controls></video>**.
- Se declaran de la siguiente forma:

```
<video controls src = "ruta/link"></video>
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Título del documento</title>
6     <style>
7       h1 {color : green;}
8       h2 {color : red;}
9       p {color: blue;}
10    </style>
11  </head>
12  <body>
13    <h1>Prueba de elemento video</h1>
14    <video controls src = "C:/Users/Cristian/Dropbox/Docencia IFP/2020-2021/Lenguaje de Marcas/EjerciciosHTML/Tema 1/EjercicioVideo/superMario.mp4"></video>
15  </body>
16 </html>
```

- Si no ponemos el atributo controls, no aparecerá la botonera para controlar el vídeo:
- Recomendado usar como formato de vídeo mp4.
- La etiqueta video también cuenta con los siguientes atributos:
 - autoplay: Reproduce automáticamente el video.
 - poster: Muestra una imagen a modo de presentación del vídeo.
 - height: Altura de presentación del vídeo.
 - width: Anchura de presentación del vídeo.
 - mute: Silencia el vídeo.
 - loop: Reproduce en bucle el vídeo
 - preload: Inicia la descarga y almacenamiento en el búfer del vídeo.

- Si queremos reproducir el vídeo a partir de un determinado segundo, deberemos introducir después de la ruta del vídeo #t= seguido del segundo del vídeo en el que deseamos que comience. Ejemplo:

```
<video controls poster = "superMario.png" src = "C:/Users/Cristian/Dropbox/Docencia IFP/2020-2021/Lenguaje de Marcas/EjerciciosHTML/Tema 1/EjercicioVideo/superMario.mp4#t=50"></video>
```

- Si queremos reproducir solo un intervalo, separaremos por coma los segundos para reflejar el intervalo. Ejemplo:

```
<video controls poster = "superMario.png" src = "C:/Users/Cristian/Dropbox/Docencia IFP/2020-2021/Lenguaje de Marcas/EjerciciosHTML/Tema 1/EjercicioVideo/superMario.mp4#t=120,130"></video>
```

4. Formularios HTML

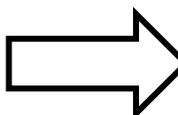
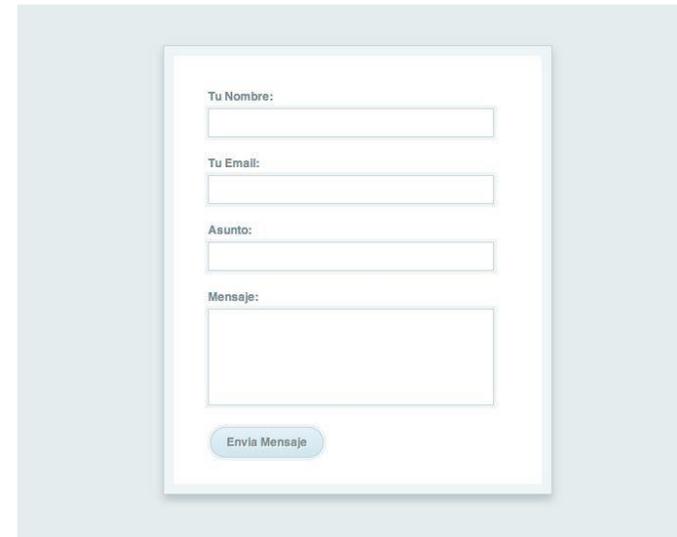
¿Qué es un formulario?

- Un formulario html es un elemento que se presenta al usuario para que rellene una serie de datos, con la intención de recoger su respuesta a estos y procesarla para alguna finalidad.
- Deben estar dentro del `<body>` de un archivo html.
- Se declaran con las etiquetas `<form>` `</form>`.

```

<html>
  <body>
    <form> ...
  </form>
  <body>
</html>

```

Tu Nombre:

Tu Email:

Asunto:

Mensaje:

Construcción de formularios: label

- Una etiqueta label es el texto que aparece asociado a una caja de entrada de datos que aparecen en un formulario. Solo sirve para indicar al usuario qué tipo de dato se le solicita.
- Se declara con las etiquetas `<label>` texto `</label>`.
- Tienen un atributo `for` cuyo valor coincide con el id de la caja de entrada de datos.

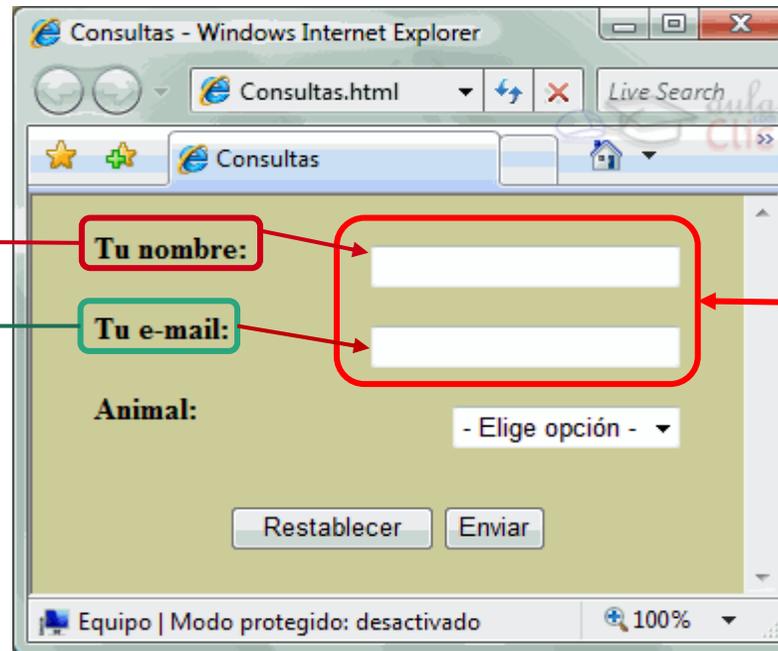


Construcción de formularios: input

- Una caja de entrada de datos es el espacio donde el usuario introduce datos.
- El label asocia información a la caja de entrada de datos el tipo de dato que se solicita al usuario.

Label asociado para indicar el nombre

Label asociado para indicar el mail



Cajas de texto

- Se declaran con las etiquetas `<input type = "X" />`, donde X puede ser:
 - text -> caja de texto normal.
 - password -> caja de texto para introducir una contraseña.
 - email -> caja de texto para introducir un email. Comprueba que el dato que introduce el usuario tiene formato de mail válido.
 - tel -> caja de texto para introducir un teléfono.
 - date -> caja de texto para seleccionar una fecha.
 - week -> caja de texto para seleccionar una semana.
 - file -> selección de un archivo con cualquier extensión.
 - url -> caja de texto para introducir un enlace.
 - hidden -> caja de texto oculta para el usuario.

- Ejemplo formulario básico:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
  </head>
  <body>
    <h1>Esto es un formulario </h1>
    <form>
      <label>Nombre del usuario</label><br>
      <input type = "text"/><br>
      <label>Password</label><br>
      <input type = "password"/>
    </form>
  </body>
</html>
```

Formulario

Caja de entrada de datos de tipo texto

Caja de entrada de datos de tipo password

Atributos importantes:

- id -> identificador único.
- name -> sirve para dar un nombre al campo que se enviará al servidor.
- required -> el formulario exige que se rellene ese campo.
- value -> si el usuario no introduce ningún valor, recoge el valor por defecto que se ha introducido en este atributo.
- placeholder -> lo que se introduzca como valor en este atributo aparece en la caja de texto.
- disabled -> la caja de texto se mantiene desactivada.

- Para agrupar varios elementos de un formulario en una sección, se utilizan las etiquetas **<fieldset>** y **</fieldset>**.
- Dentro de dichas etiquetas, se colocarán los elementos que deseamos agrupar.
- Si deseamos dar un nombre a dicha sección, debemos utilizar las etiquetas **<legend>** y **</legend>** de la siguiente forma:

`<legend> X </legend>` (donde X es el nombre de la sección).

- Los textarea son cajas de entrada de datos de texto más grandes que las normales.
- Se declaran del siguiente modo:

`<textarea rows="nº Filas" cols="nº Columnas" name=" nombre"></textarea>`

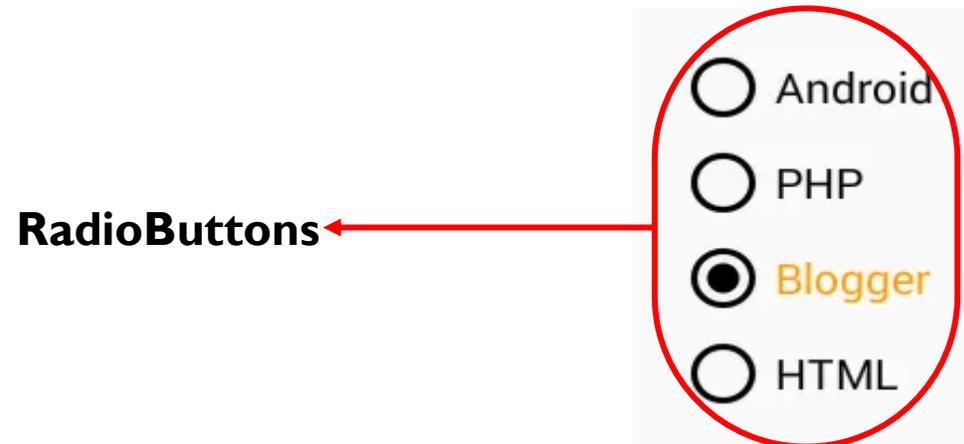
Textarea ←

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
  </head>
  <body>
    <h1>Esto es un formulario </h1>
    <form>
      <label for = "userName">Nombre del usuario</label><br>
      <input type = "text" id = "userName"/><br>
      <label for = "pwd">Password</label><br>
      <input type = "password" id = "pwd"/><br>
      <label for = "opinion">Opinión del usuario</label><br>
      <textarea rows = 10 cols = 50 name = "areaTexto"></textarea>
    </form>
  </body>
</html>

```

- Los elementos radiobutton sirven para poder seleccionar una opción (**sólo una**) de las que nos presenta el formulario.
- Pueden aparecer varios grupos de radiobutton en el formulario.
- Ejemplo visual de radiobuttons:



- Se declaran con las etiquetas `<input type = "radio" />`.
- Atributos importantes:
 - id -> identificador único.
 - name -> sirve para dar un nombre al campo que se enviará al servidor.
 - value -> el valor de la opción marcada que se enviará luego al servidor.
 - checked -> sirve para dejar una opción marcada por defecto.

- Ejemplo de formulario con radiobutton:

```
<!DOCTYPE html>
<html>
  <body>

    <h1>Ejemplo de raddiobuttons para Lenguaje de Marcas</h1>

    <form method="post">
      <p>Por favor escoja uno:</p>
      <input type="radio" id="pbulbasaur" name="pokemon" value="bulbasaur">
      <label for="pbulbasaur">Bulbasaur</label><br>
      <input type="radio" id="pcharmmander" name="pokemon" value="charmmander">
      <label for="pcharmmander">Charmander</label><br>
      <input type="radio" id="psquirtle" name="pokemon" value="squirtle">
      <label for="psquirtle">Squirtle</label>
    </form>

  </body>
</html>
```

Formulario

Radiobutton bulbasaur

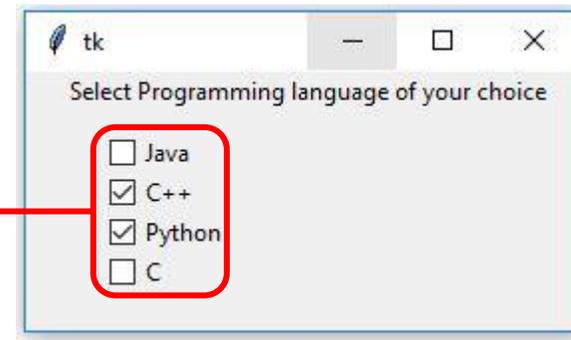
Radiobutton charmander

Radiobutton squirtle

Es importante que los *name* sean iguales para agrupar todos los radiobuttons y sólo se pueda seleccionar 1 opción

- Los elementos checkbox sirven para poder seleccionar una, varias, o ninguna opción de las que nos presenta el formulario.
- A diferencia de los radiobutton, podemos escoger más de una o ninguna respuesta.
- Ejemplo visual de checkbox en un formulario:

Checkbox



- Se declaran con las etiquetas `<input type = "checkbox" />`
- Atributos importantes:
 - id -> identificador único.
 - name -> sirve para dar un nombre al campo que se enviará al servidor.
 - value -> el valor de la opción marcada que se enviará luego al servidor.
 - checked -> sirve para dejar una opción marcada por defecto.

- Ejemplo de formulario con checkbox:

```

<!DOCTYPE html>
<html>
  <body>

    <h1>Ejemplo de checkbox para Lenguaje de Marcas</h1>

    <form action="get">
      <p>Selecciona tus animales favoritos:</p>
      <input type="checkbox" id="animPerro" name="animales" value="perro">
      <label for="animPerro">Perro</label><br>
      <input type="checkbox" id="animGato" name="animales" value="gato">
      <label for="animGato">Gato</label><br>
      <input type="checkbox" id="animCanario" name="animales" value="canario">
      <label for="animCanario">Canario</label><br>
      <input type="checkbox" id="animHuron" name="animales" value="huron">
      <label for="animHuron">Hurón</label><br>
    </form>

  </body>
</html>

```

Formulario ←

→ Checkbox perro

→ Checkbox gato

→ Checkbox canario

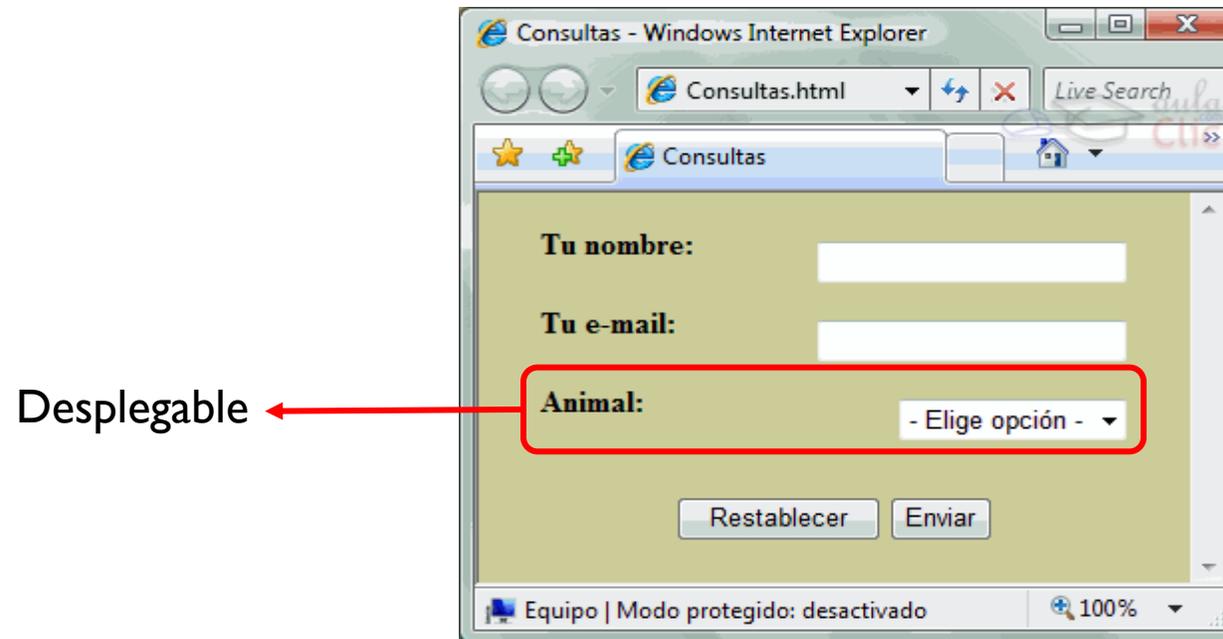
→ Checkbox huron

- Los rangos se declaran con las etiquetas `<input type = "range" min=X max=Y />`.
- Atributos importantes:
 - id -> identificador único.
 - name -> sirve para dar un nombre al campo que se enviará al servidor.
 - value -> el valor de la opción marcada que se enviará luego al servidor.
 - min -> sirve para establecer el valor mínimo del rango.
 - max -> sirve para establecer el valor máximo del rango.
 - step -> precisión (en decimales).



Construcción de formularios: select

- Sirven para mostrar una lista de opciones, ahorrando espacio en el formulario.
- Se pueden seleccionar una o varias opciones, pero lo normal es que se utilice para seleccionar una opción solo.



- Los desplegables se declaran con las etiquetas **<select>** y **<select/>**.
- Cada una de las opciones dentro del desplegable, se declara con las etiquetas **<option>** Texto de la opción **</option>**.
- Si deseamos agrupar un conjunto de opciones por un término utilizamos:
<optgroup label = "Nombre del término"> <option> Texto </option> ... **</optgroup>**
- Si deseamos que una opción aparezca marcada por defecto, deberemos usar el atributo *selected* dentro de la etiqueta.

<option selected> Texto de la opción **</option>**

- Ejemplo de formulario con select:

```

<!DOCTYPE html>
<html>
  <body>

    <h1>Ejemplo de select para Lenguaje de Marcas</h1>

    <form action="get">
      <p>Selecciona tu animal favorito:</p>
      <select id = "listAnimal" size = 1>
        <option>Perro</option>
        <option>Gato</option>
        <option>Hurón</option>
        <option>Canario</option>
        <option>Serpiente</option>
      </select>
    </form>

  </body>
</html>

```

Formulario

Select

Opciones del select

- Atributos importantes:
 - id -> identificador único.
 - name -> sirve para dar un nombre al campo que se enviará al servidor.
 - size -> el número de opciones que se muestran sin desplegar la lista de opciones.
 - multiple -> sirve para poder seleccionar + de 1 opción en el desplegable.

Construcción de formularios: select

- Ejemplo formulario con select con atributos y grupos:

```

<!DOCTYPE html>
<html>
  <body>

    <h1>Ejemplo de select para Lenguaje de Marcas</h1>

    <form action="get">
      <p>Selecciona tus animales favoritos:</p>
      <select id = "listAnimal" size = 1>
        <optgroup label="Mamiferos">
          <option>Perro</option>
          <option>Gato</option>
          <option selected>Hurón</option>
          <option>Conejo</option>
        </optgroup>
        <optgroup label="Aves">
          <option>Canario</option>
          <option>Gallina</option>
        </optgroup>
        <optgroup label="Reptiles">
          <option>Serpiente</option>
          <option>Iguana</option>
        </optgroup>
      </select>
    </form>

  </body>
</html>

```

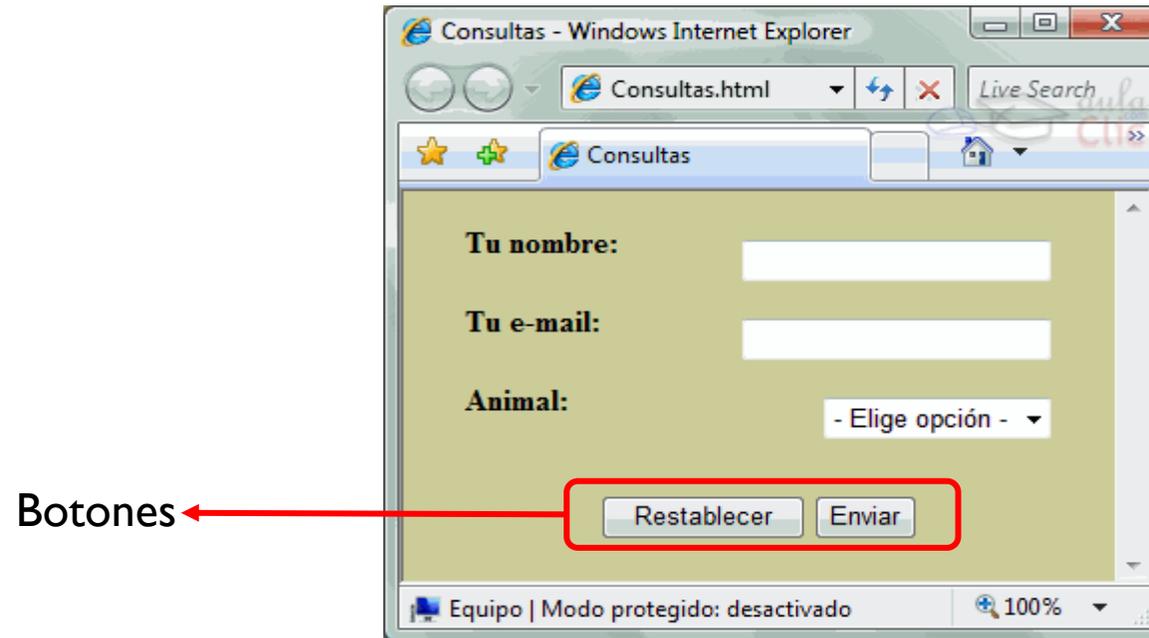
Formulario

Select

La declaración de los grupos es similar a las listas anidadas

Grupo Mamíferos

- Los botones se utilizan en los formularios para ejecutar alguna acción javascript sobre el formulario o enviar los datos introducidos por el usuario al servidor.



- Se declaran con las etiquetas `<button type = "X" />`, donde X puede ser:
 - button -> botón normal.
 - submit -> envía los datos introducidos por el usuario al controlador para procesarlos.
 - reset -> limpia el contenido introducido en el formulario.

Construcción de formularios: button

- Ejemplo formulario con botón submit:

Estilo del botón (CSS)

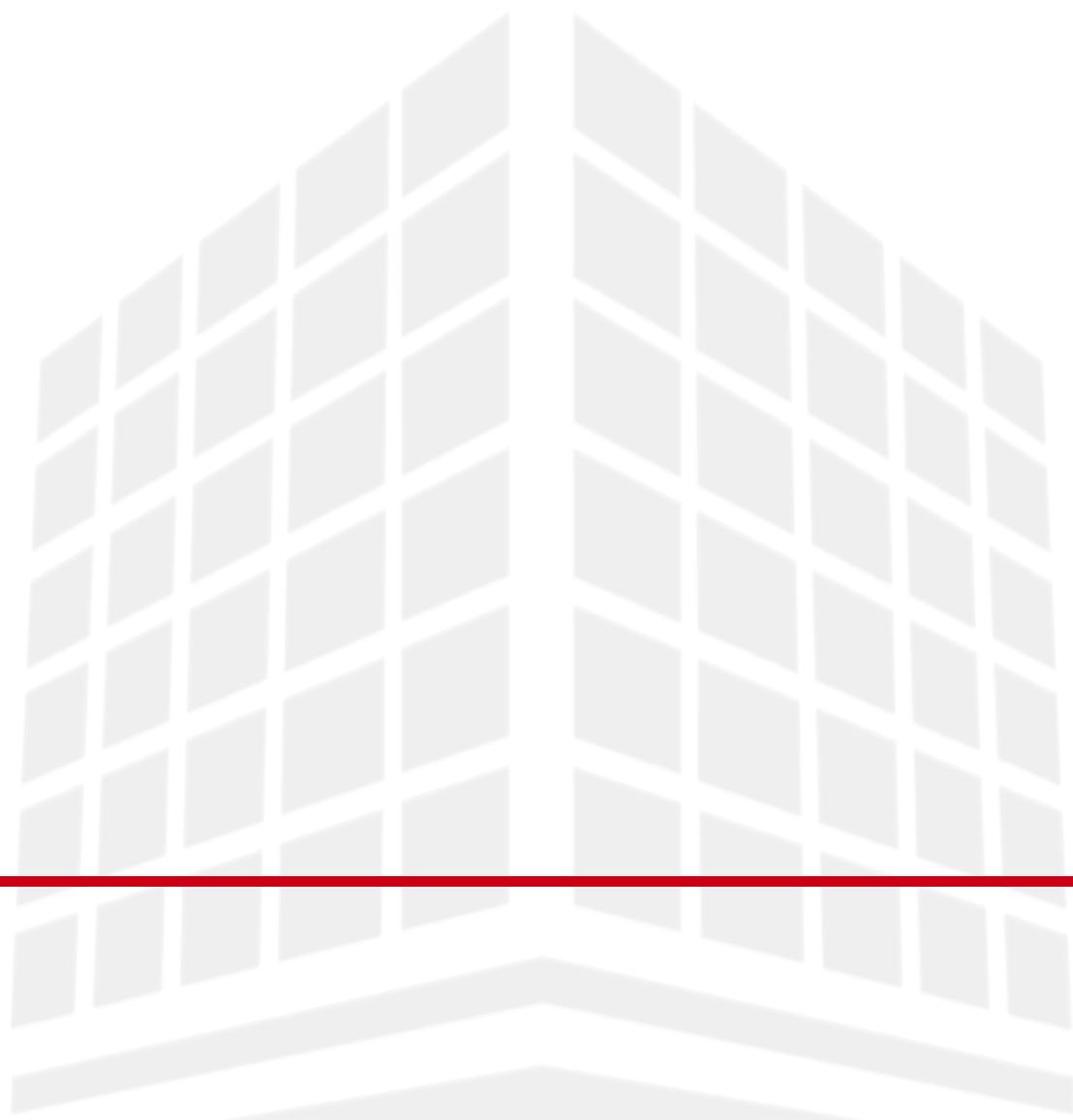
Formulario

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Título del documento</title>
    <style>
      .button {
        border: none;
        color: white;
        padding: 15px 32px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
        margin: 4px 2px;
        cursor: pointer;
        background-color: #008CBA;
      }
    </style>
  </head>
  <body>
    <h1>Esto es un formulario </h1>
    <form>
      <label for = "userName">Nombre del usuario</label><br>
      <input type = "text" id = "userName"/><br>
      <label for = "pwd">Password</label><br>
      <input type = "password" id = "pwd"/><br>
      <label for = "opinion">Opinión del usuario</label><br>
      <textarea rows = 10 cols = 50 name = "areaTexto"></textarea><br>
      <button type = "submit" class = "button">Enviar</button>
    </form>
  </body>
</html>

```

Botón enviar



Tecnología y Uso de Internet

Tema 4: Diseño web – CSS

1. ¿Qué es CSS?
2. Sintaxis CSS.
3. Modos de incorporar CSS.
4. Selectores.
5. Consideraciones adicionales.

I. ¿Qué es CSS?

- CSS (*Cascading Style Sheets* – Hojas Estilo en Cascada).
- En su interior, se encuentran un conjunto de declaraciones dirigidas a dar formato y presentación de los contenidos de una página web.
- Sólo sirve para definir la forma de representación de HTML.
- Permite controlar el diseño de una o varias páginas web.
- Los archivos de estilos tienen extensión `.css`.

HTML

JS

CSS

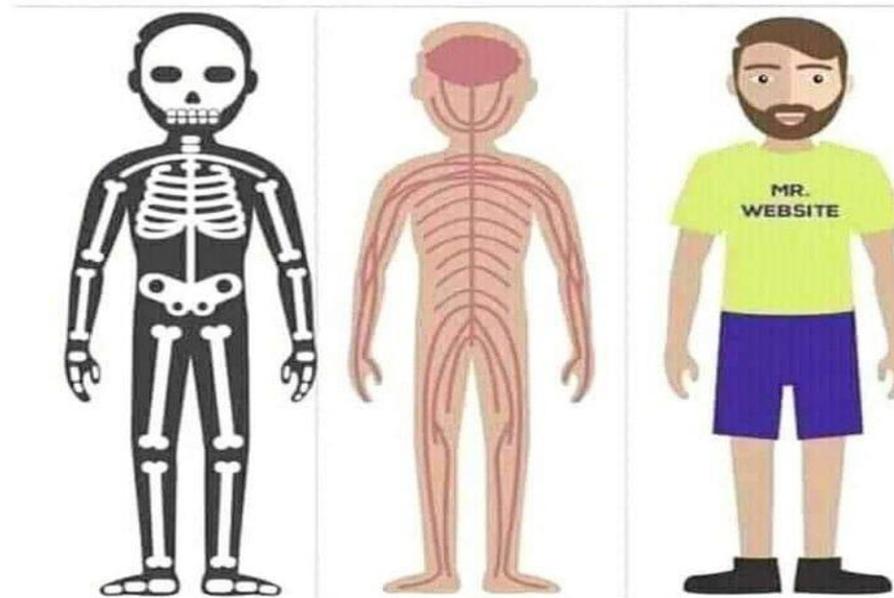
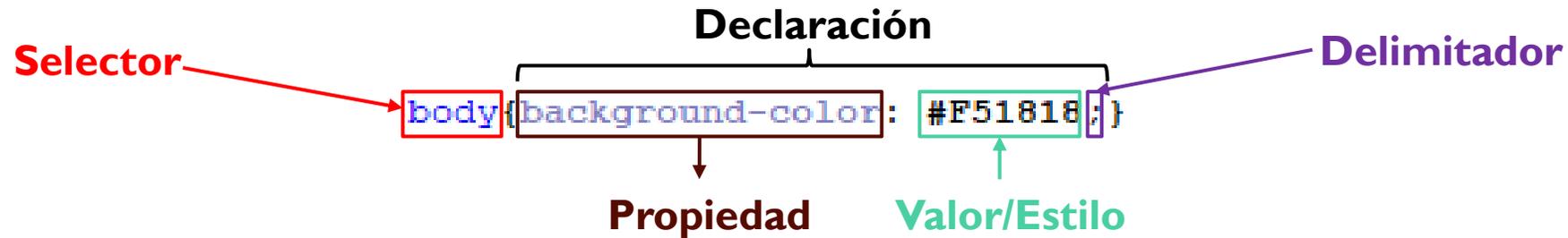


Imagen realizada por : Roberto Villamil

2. Sintaxis CSS



- El **selector** hace referencia al elemento HTML sobre el que se desea modificar una propiedad.
- Entre las llaves puede haber una o más **declaraciones** haciendo referencia al estilo de las propiedades del elemento HTML.
- Las declaraciones constan de:
 - El **nombre de la propiedad** del elemento HTML a modificar.
 - **Dos puntos**.
 - El **valor/estilo** que se desea aplicar a la propiedad.
 - **Punto y coma**, que actúa como delimitador de los diferentes estilos aplicados a las distintas propiedades del elemento HTML.

3. Modos de incorporar CSS

De forma directa (no recomendada):

- Se añade el atributo *style* al elemento HTML sobre el que se desea modificar su estilo.
- El estilo definido sólo afecta al elemento HTML sobre el que se ha añadido el atributo *style*.

```
<html lang = "es">
  <head>
    <title>Prueba</title>
  </head>
  <body>
    <h1 style = "color:green;">Hola a todos</h1>
    <p style = "color:black; background-color:blue;">Esta página es una prueba para la asignatura de Lenguajes de Marcas</p>
  </body>
</html>
```

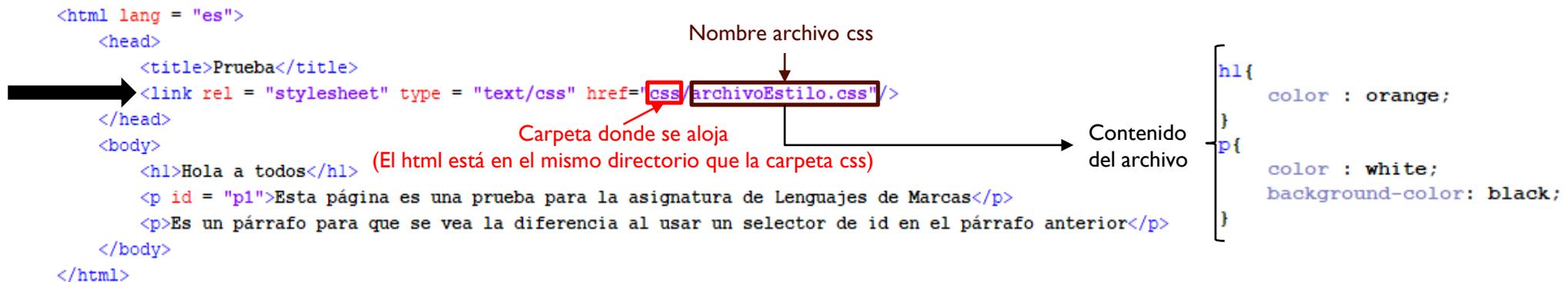
De forma interna (no recomendada):

- Se debe insertar el estilo de la propiedad del elemento HTML entre las etiquetas `<style>` y `</style>`.
- Las etiquetas `<style></style>` deben ir en la sección head (`<head>`) del archivo HTML.

```
<html lang = "es">
  <head>
    <title>Prueba</title>
    <style>
      h1 {color : green;}
      p {color:black;background-color : red;}
    </style>
  </head>
  <body>
    <h1>Hola a todos</h1>
    <p>Esta página es una prueba para la asignatura de Lenguajes de Marcas</p>
  </body>
</html>
```

De forma externa (forma recomendada):

- Se debe indicar en cada página HTML el archivo externo en cuyo interior se encuentren los estilos definidos para ella. Para ello:
- Incluir dentro del *head* la etiqueta `<link>` y definir los siguientes atributos:
 - `rel = "stylesheet"`.
 - `type = "text/css"`.
 - `href = "ruta donde se encuentra el archivo css"`.



4. Selectores

De etiqueta:

- El selector corresponde al elemento HTML que tiene el mismo nombre.

```
h1{  
  color : orange;  
}  
p{  
  color : white;  
  background-color: black;  
}
```

De clase:

- El selector corresponde al valor del atributo *class* del elemento HTML.
- Debe ir precedido de un punto “.” seguido del valor del atributo *class*.

Párrafo 1

Párrafo 2

Párrafo 3

Párrafo 4

Párrafo 5

```
<p>Párrafo 1</p>
<p class="pEspecial">Párrafo 2</p>
<p>Párrafo 3</p>
<p class="pEspecial">Párrafo 4</p>
<p>Párrafo 5</p>
```

```
p {
    background-color: lightblue;
}

.pEspecial {
    background-color: lightgreen;
}
```

De id:

- El selector corresponde con el valor del atributo *id* del elemento HTML.
- Debe ir precedido del símbolo #.

Párrafo 1

Párrafo 2

Párrafo 3

Párrafo 4

Párrafo 5

```
<p>Párrafo 1</p>
```

```
<p class="pEspecial">Párrafo 2</p>
```

```
<p>Párrafo 3</p>
```

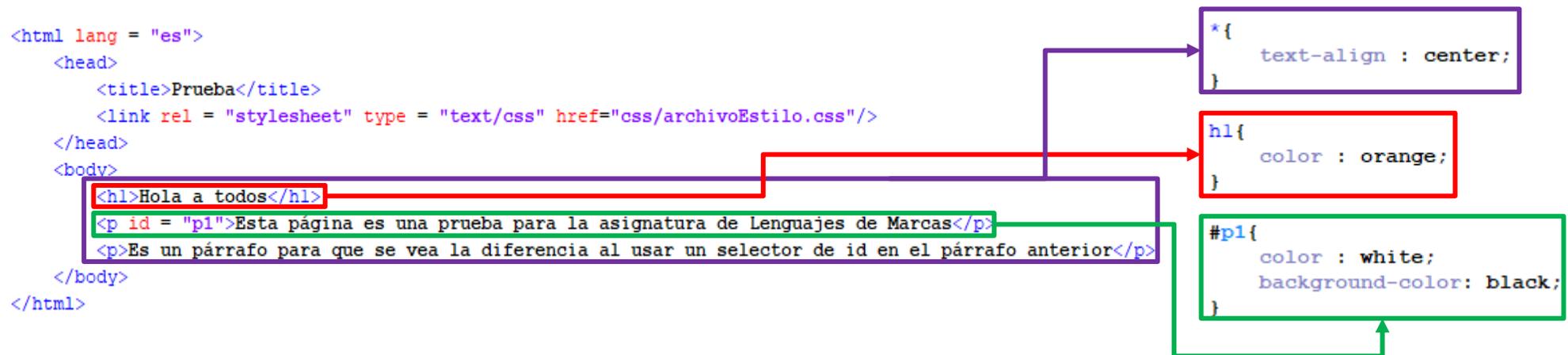
```
<p class="pEspecial">Párrafo 4</p>
```

```
<p id="miParrafo">Párrafo 5</p>
```

```
p {  
    background-color: lightblue;  
}  
  
.pEspecial {  
    background-color: lightgreen;  
}  
  
#miParrafo {  
    background-color: blue;  
}
```

Selector universal:

- Atribuye el estilo definido a todos los elementos del HTML.
- El selector es el símbolo *.



Descendente:

- Se utiliza en elementos que son hijos de otros elementos (cuando están dentro de las etiquetas de `<>` y `</>` de otro elemento). `p span { color: red; }`

Adyacente (+):

- Elementos que se encuentran a continuación de otros elementos (son hermanos y aparecen inmediatamente uno detrás del otro).

```

*{
  text-align : center;
}
h1{
  color : orange;
}
#p1{
  color : white;
  background-color: black;
}
h2 { color: green; }
h1 + h2 { color: red }

```

```

<html lang = "es">
  <head>
    <title>Prueba</title>
    <link rel = "stylesheet" type = "text/css" href="css/archivoEstilo.css"/>
  </head>
  <body>
    <h1>Hola a todos</h1>
    <h2>Esto es el título 2</h2>
    <h3>Esto es el título 3</h3>
    <h2>Esto es otro título 2</h2>
    <p id = "p1">Esta página es una prueba para la asignatura de Lenguajes de Marcas</p>
    <p>Es un párrafo para que se vea la diferencia al usar un selector de id en el párrafo ar
  </body>
</html>

```

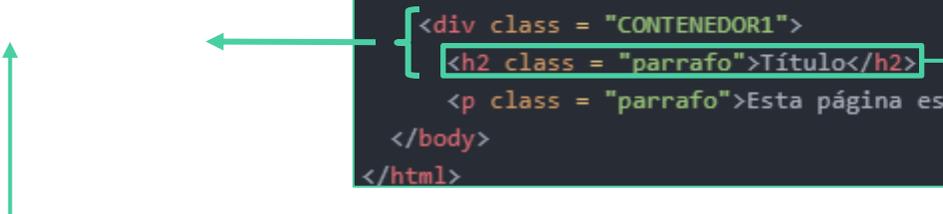
Hermanos y aparecen a continuación

Se puede combinar distintos tipos de selectores. Ejemplos:

- `.CONTENEDOR1 .parrafo {...}` -> Selecciona los elementos con `class = "parrafo"` que se encuentran dentro de cualquier elemento con `class = "CONTENEDOR1"`.
- `div.CONTENEDOR1 h2.parrafo {...}` -> Selecciona los elementos de tipo `<h2>` con un atributo `class = "parrafo"` que se encuentran dentro de cualquier elemento de tipo `<div>` con `class = "CONTENEDOR1"`.

```
/*Combinación de selectores */
div.CONTENEDOR1 h2.parrafo {
  color: yellow;
}
```

```
<html lang = "es">
  <head>
    <title>Prueba</title>
    <link rel = "stylesheet" type = "text/css" href="css/archivoEstilo.css"/>
  </head>
  <body>
    <h1>Hola a todos</h1>
    <div class = "CONTENEDOR1">
      <h2 class = "parrafo">Titulo</h2>
      <p class = "parrafo">Esta página es una prueba para la asignatura de Lenguajes de Marcas</p>
    </div>
  </body>
</html>
```



Selectores en función de atributos: permiten seleccionar elementos HTML en función de sus atributos y/o valores.

- `[nombre_atributo]`, selecciona los elementos que tienen el atributo llamado `nombre_atributo`, (no tiene en cuenta el valor).
- `[nombre_atributo=valor]`, selecciona los elementos que tienen un atributo llamado `nombre_atributo` con un valor igual a `valor`.
- `[nombre_atributo~=valor]`, selecciona los elementos que tienen atributo llamado `nombre_atributo` y al menos uno de los valores del atributo es `valor`.
- `[nombre_atributo|=valor]`, selecciona los elementos que tienen atributo llamado `nombre_atributo` y cuyo valor está entre una serie de palabras separadas con guiones. Sólo es útil para los atributos de tipo `lang` que indican el idioma del contenido del elemento.

```
/*Se muestran de color azul todos los enlaces que tengan un atributo "class", independientemente de su valor */  
a[class] { color: blue; }  
  
/*Se muestran de color azul todos los enlaces que tengan un atributo "class" con el valor "externo" */  
a[class="externo"] { color: blue; }  
  
/*Se muestran de color azul todos los enlaces que apunten al sitio "http://www.ifp.es" */  
a[href="http://www.ifp.es"] { color: blue; }  
  
/*Se muestran de color azul todos los enlaces que tengan un atributo "class" en el que al menos uno de sus valores sea "externo" */  
a[class~="externo"] { color: blue; }  
  
/*Selecciona todos los elementos de la página cuyo atributo "lang" sea igual a "en", es decir, todos los elementos en inglés */  
*[lang=en] { ... }  
  
/*Selecciona todos los elementos de la página cuyo atributo "lang" empiece por "es", es decir, "es", "es-ES", "es-AR", etc. */  
*[lang|="es"] { color : red }
```

5. Consideraciones adicionales

A lo largo de este tema tan solo se han utilizado dos propiedades CSS a modo de ejemplo: *color* (color de la fuente) y *background-color* (color del fondo). Sin embargo, **las posibilidades de CSS son ilimitadas**, existiendo otras propiedades referentes al tipo y tamaño de la fuente, posicionamiento de elementos, márgenes, transparencia de elementos, etc.

Para más información sobre CSS podéis consultar la W3Schools (<https://www.w3schools.com/css/default.asp>), así como el documento con información adicional.

Si bien en los comienzos de la web tan solo era posible navegar a través de ordenadores de escritorio, lo cierto es que **a día de hoy son muchos los dispositivos que incluyen exploradores web** (ordenadores, móviles, tablets, relojes, etc.).

A consecuencia de lo anterior, debemos asegurarnos de que las páginas web que construimos **se visualicen correctamente en cualquier dispositivo**, independientemente de las características del mismo (resolución y relación de aspecto de la pantalla).

Dos posibles soluciones:

- Construir una versión de la web para cada tipo de dispositivo (escritorio y móvil).
- Realizar **diseños responsivos**.

Para realizar diseños responsivos, utilizamos reglas CSS condicionales. Ejemplos:

- `<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />`
- ```
@media (max-width: 600px) {
 .facet_sidebar { display: none; }
}
```

Para facilitar la elaboración de diseños responsivos, podemos utilizar librerías como Bootstrap (<https://getbootstrap.com/>):

- Copiamos las siguientes líneas en el head de nuestro HTML:
  - `<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">`
  - `<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>`
- Copiamos el código de los componentes que nos interesen entre aquellos que contiene Bootstrap: <https://getbootstrap.com/docs/5.3/components/buttons/>



# Tecnología y Uso de Internet

---

Tema 5: Web dinámica – JavaScript

1. ¿Qué es JavaScript?
2. Modos de incorporar JavaScript.
3. Sintaxis JavaScript.

## I. ¿Qué es JavaScript?

---

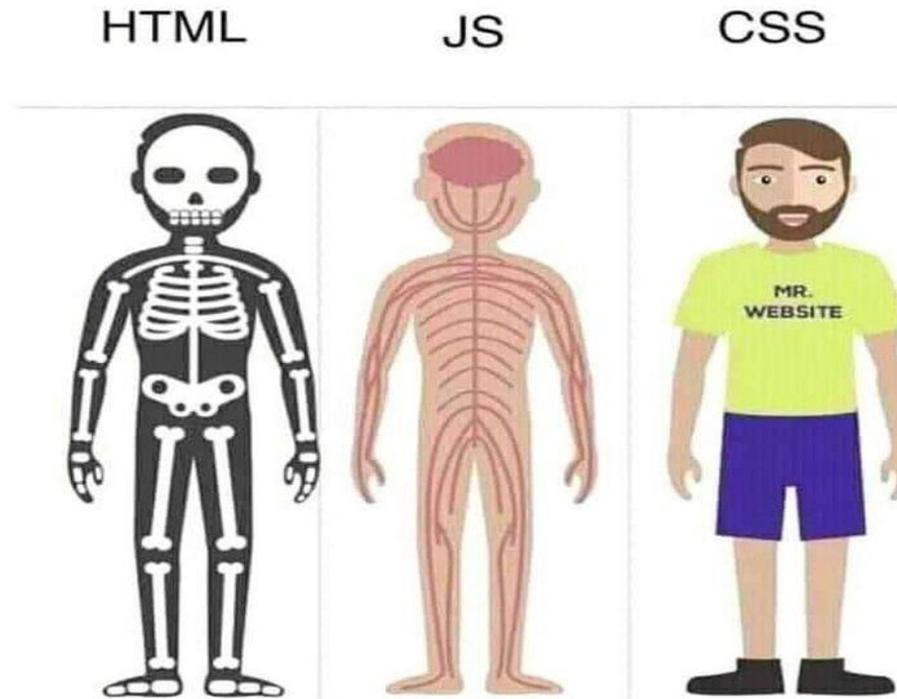


Imagen realizada por : Roberto Villamil

- **Lenguajes de programación y lenguajes de scripting:** se diferencian en que los lenguajes de scripting están pensados para pequeñas funcionalidades muy concretas.
- **JavaScript** surge como un **lenguaje de scripting**, pero su uso evoluciona hasta el punto de que **actualmente se considera un lenguaje de programación**.
- Principalmente se utiliza en el **lado del cliente** (*frontend*).
- Lenguaje interpretado.
- Lenguaje orientado a objetos.

- **Arquitectura cliente-servidor:** la página web se almacena en el servidor y se visualiza en el cliente (navegador).
- Origen de la web: **páginas web estáticas** (solo HTML y CSS).
- Evolución de la web: **páginas web dinámicas** (incluyen algún lenguaje que permite dotar de funcionalidad a la web).
  - **PHP:** funcionalidad en el **lado del servidor**.
  - **JavaScript:** funcionalidad en el **lado del cliente**, aunque actualmente también puede utilizarse en el lado del servidor.

# Lenguajes relacionados:

- **HTML**. Lenguaje de marcado. Define el **contenido** del documento.
- **CSS**. Lenguaje de diseño gráfico. Define la **presentación** del documento.

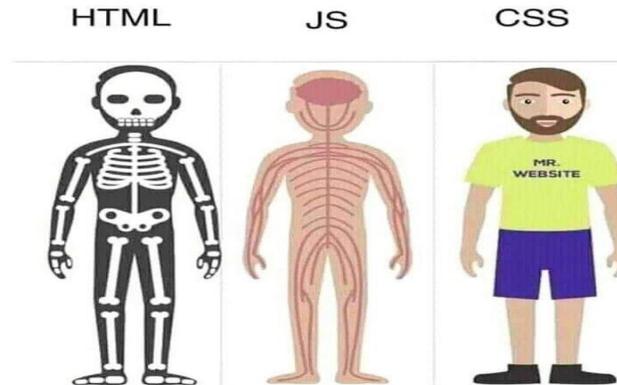
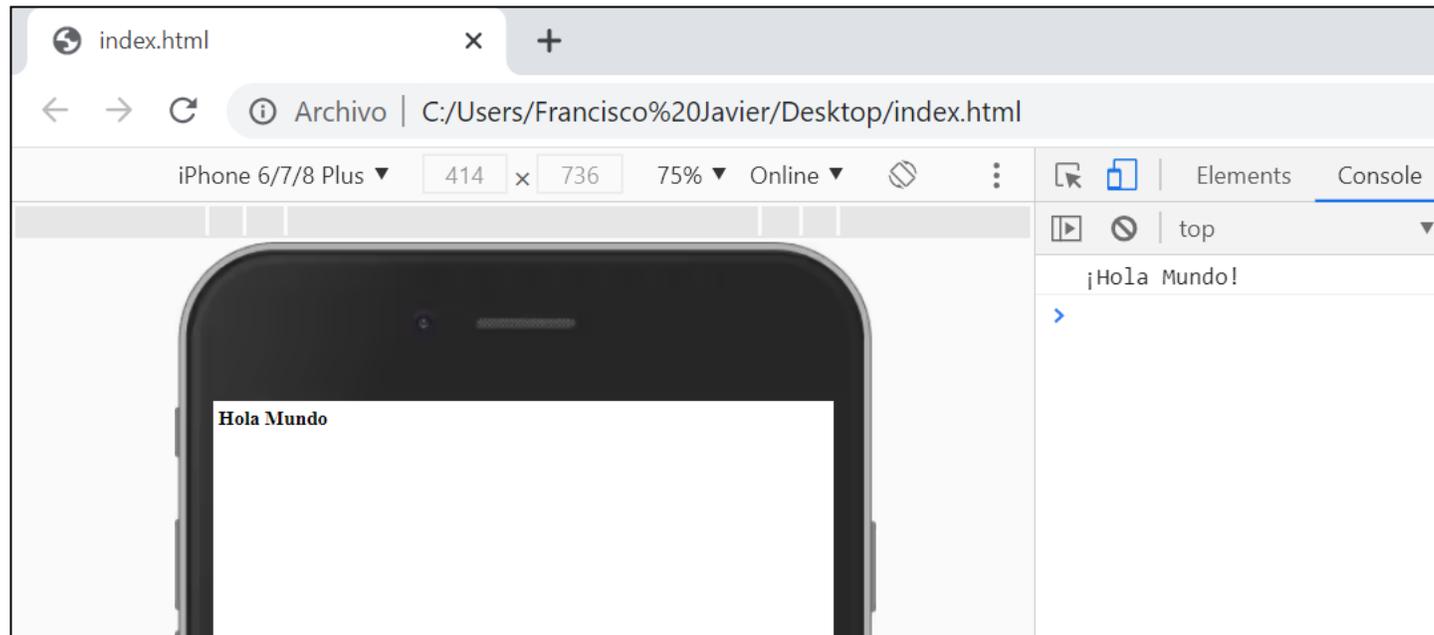


Imagen realizada por : Roberto Villamil

- Sintaxis sencilla.
- Mucha documentación.
- Ejecución rápida.
- Versátil.
- Multiplataforma.
- Seguridad.
- Ideado para desarrollos pequeños.



**Google Chrome:** abrir herramientas de desarrollador: **Ctrl + Mayús + I**



## 2. Modos de incorporar JavaScript

---

De forma interna:

```
1 <html>
2
3 <head>
4 <title>Primera App</title>
5 </head>
6
7 <body>
8
9 <h1>Hola Mundo</h1>
10
11 <script type="text/javascript">
12 // Esto es un comentario de una línea.
13 /*
14 Esto es
15 un comentario
16 de varias líneas.
17 */
18 alert("¡Hola Mundo!");
19 </script>
20
21 </body>
22
23 </html>
```

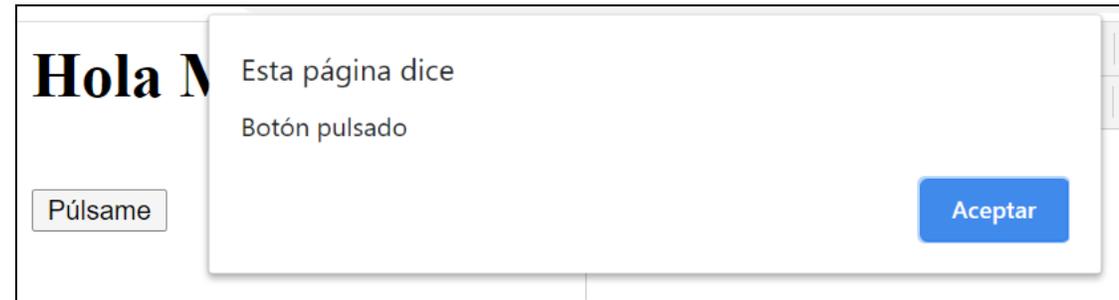
## De forma externa:

```
index.html × miScript.js
C: > Users > Francisco Javier > Desktop > index.html > htm
1 <html>
2
3 <head>
4 <title>Segunda App</title>
5 </head>
6
7 <body>
8
9 <h1>Hola Mundo</h1>
10
11 <script src="./miScript.js"></script>
12
13 </body>
14
15 </html>
```

```
index.html miScript.js ×
C: > Users > Francisco Javier > Desktop > miScript.js
1 alert("Hola Mundo!");
```

## De forma directa:

- Esta forma consiste en añadir **eventos** a elementos HTML.
- **Ejercicio:** incluir un botón que, al pulsarlo, la página nos muestre una alerta.



Atención a la alternancia de comillas dobles (“) y simples (‘).

## 3. Sintaxis JavaScript

---

- `console.log("mensaje de registro");`
- `console.info("mensaje informativo");`
- `console.error("mensaje de error");`

**Ejercicio:** al pulsar el botón del ejercicio anterior, hacer que se muestre por consola un mensaje de log, otro de información y otro de error.

Sintaxis básica de funciones:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Prueba</title>
5 </head>
6 <body>
7
8 <button type="button" id="miBoton" onclick="saludar();">
9 Saludar
10 </button>
11
12 <script type="text/javascript">
13
14 function saludar() {
15 alert("Hola Mundo");
16 alert("segunda alerta");
17 }
18
19 </script>
20
21 </body>
22 </html>

```

```

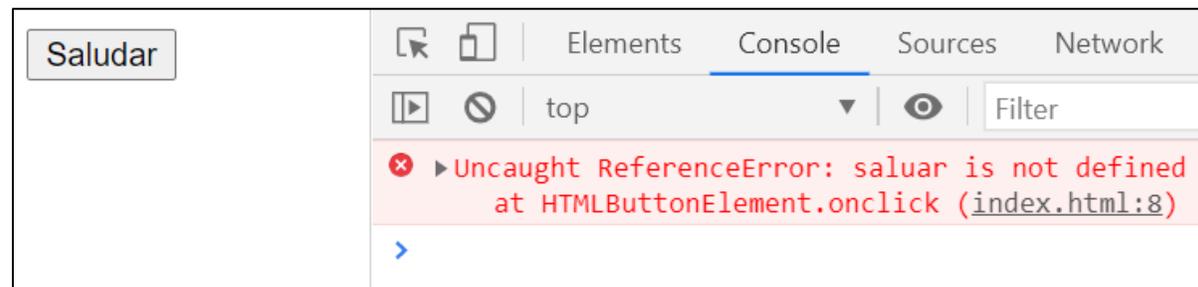
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Prueba</title>
5 </head>
6 <body>
7
8 <button type="button" id="miBoton" onclick="saludar();">
9 Saludar
10 </button>
11
12 <script type="text/javascript" src="./funciones.js"></script>
13
14 </body>
15 </html>

```

```

1 function saludar() {
2 alert("Hola Mundo");
3 alert("segunda alerta");
4 }

```



A partir de ahora, **los eventos llamarán a funciones** donde incluiremos el código.

**Estas funciones estarán en ficheros externos** que importaremos al final del *body* del HTML.

El objeto *document* representa el documento sobre el que estamos trabajando. Entre otras cosas, nos permite escribir contenido HTML:

```
document.write("<h1>Hola Mundo</h1>");
```

**Ejercicio:** Al pulsar el botón del ejercicio anterior, hacer que cambie el contenido HTML del documento y muestre, en negrita y cursiva, el texto botón pulsado.

Con el objeto *document*, también podemos recuperar cualquier elemento HTML...

```
document.getElementById("mi_ID")
```

**Importante:** el elemento HTML que queramos recuperar debe tener una propiedad id. Ej: `<p id="parrrafo1">...</p>`

...y sustituir su contenido si se trata de un contenedor div o párrafo (p):

```
document.getElementById("mi_ID").innerHTML = "Hola Mundo";
```

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>JavaScript Fundamentos</title>
</head>
<body>
 <h2>¿Qué podemos hacer con JavaScript?</h2>
 <p id="parrafo">JavaScript puede cambiar contenido HTML.</p>
 <button type="button"
 onclick="document.getElementById('parrafo').innerHTML = 'Hola Mundo, desde JavaScript!! '">Pulsamé!</button>
</body>
</html>
```

...modificar cualquiera de sus propiedades:

```
document.getElementById("mi_ID").src = "milmagen.png"
```

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>JavaScript Fundamentos</title>
</head>
<body>
 <h2>¿Qué podemos hacer con JavaScript?</h2>
 <p id="parrafo">JavaScript puede cambiar atributos HTML.</p>
 <button onclick="document.getElementById('imagenLuz').src='LuzEncendida.gif'">Encender luz !</button>

 <button onclick="document.getElementById('imagenLuz').src='LuzApagada.gif'">Apagar luz !</button>
</body>
</html>
```

...modificar sus propiedades CSS:

```
document.getElementById("mi_ID").style.fontSize = "35px"
```

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>JavaScript Fundamentos</title>
</head>
<body>
 <h2>¿Qué podemos hacer con JavaScript?</h2>
 <p id="parrafo">JavaScript puede cambiar estilos HTML(CSS).</p>
 <button type="button" onclick="document.getElementById('parrafo').style.fontSize='35px'">Pulsamé !!</button>
</body>
</html>
```

...o incluso recuperar el valor de un campo de un formulario:

```
<input type="text" id="mi_ID" placeholder="Introduce tu nombre">
```

```
... document.getElementById("mi_ID").value;
```

**Pero antes...**

```
1 function miFuncion() {
2
3 var nombre = "Agustín";
4 nombre = nombre + " García";
5 var edad = 20;
6 edad = edad + 1;
7 var estudiante = true;
8 const idPersona = "12345678B";
9
10 var mensaje = "ID Persona: " + idPersona +
11 "\nNombre: " + nombre +
12 "\nEdad: " + edad;
13
14 if (estudiante) {
15 mensaje = mensaje + "\nEs estudiante";
16 } else {
17 mensaje = mensaje + "\nNo es estudiante";
18 }
19
20 alert(mensaje);
21
22 }
```

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

Operator	Example	Same As
=	<code>x = y</code>	<code>x = y</code>
<code>+=</code>	<code>x += y</code>	<code>x = x + y</code>
<code>-=</code>	<code>x -= y</code>	<code>x = x - y</code>
<code>*=</code>	<code>x *= y</code>	<code>x = x * y</code>
<code>/=</code>	<code>x /= y</code>	<code>x = x / y</code>
<code>%=</code>	<code>x %= y</code>	<code>x = x % y</code>
<code>**=</code>	<code>x **= y</code>	<code>x = x ** y</code>

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

Operator	Description
&&	logical and
	logical or
!	logical not

```
1 function miFuncion() {
2
3 let operador1 = 2;
4 let operador2 = 3;
5 suma(operador1, operador2);
6 }
7
8 function suma(valor1, valor2) {
9 alert(valor1 + valor2);
10 }
```

```
1 function miFuncion() {
2 let operador1 = 2;
3 let operador2 = 1;
4 let resultado = suma(operador1, operador2);
5 alert(resultado);
6 }
7
8 function suma(valor1, valor2) {
9 return valor1 + valor2;
10 }
```



# Tecnología y Uso de Internet

---

Tema 6: Sistemas de gestión de contenido web

- 1. Introducción.**
- 2. WordPress.**

## I. Introducción

---

## Web estática:

- <https://www.xataka.com/tecnologiazan/las-webs-de-los-90-mas-miticas-vistas-en-2017-que-si-creetelo-que-esto-existio>
- Si bien en la mayoría de los casos se desarrollaban de manera completamente manual, existían determinadas herramientas que facilitaban este proceso: plantillas, editores como Adobe Dreamweaver, etc.

## Web dinámica:

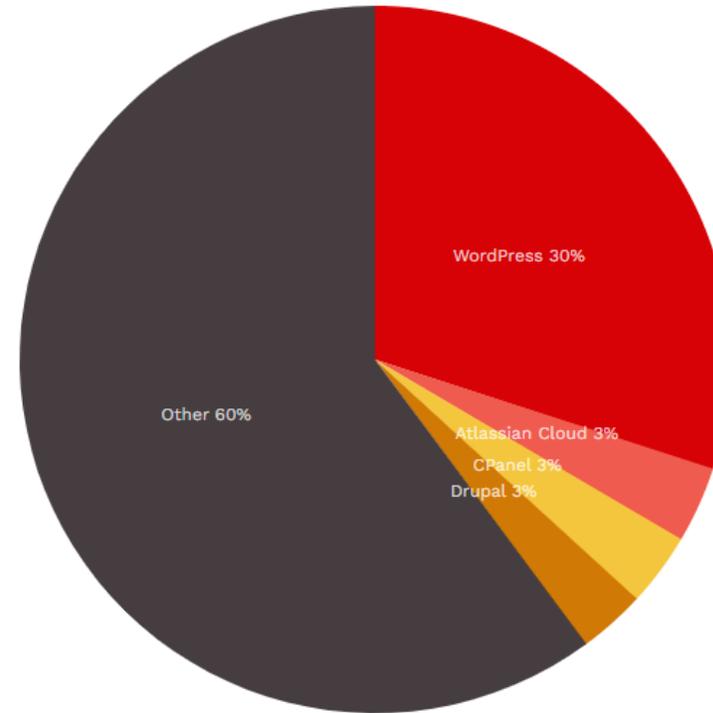
- Existe una gran variedad de métodos de desarrollo que pueden combinarse: de manera manual, utilizando *frameworks*, de manera gráfica o visual, mediante Sistemas de Gestión de Contenidos, etc.

## **Sistemas de Gestión de Contenidos (CMS, *Content Management System*):**

- Un CMS es un programa informático para la creación y administración de contenidos web por parte de administradores, editores y demás participantes.
- Generalmente no es necesario tener grandes conocimientos técnicos para su utilización, si bien es muy recomendable en algunos casos.
- Los CMS actuales permiten construir páginas web de manera visual, pero en algunos casos es posible editar el código fuente resultante.
- Ejemplos de CMS: Blogger, Wix, Joomla, Drupal, WordPress, Moodle (LMS, Learning Management System), etc.

## CMS más utilizados:

- <https://trends.builtwith.com/cms>



Explain Chart

Top In CMS Usage Distribution in the Top 1 Million Sites

Technology	Websites	%
WordPress	243,387	24.34
Atlassian Cloud	28,201	2.82
CPanel	27,113	2.71
Drupal	24,671	2.47

## 2. WordPress

---

- WordPress es un CMS para el desarrollo de cualquier tipo de página web: un pequeño blog, la página de una gran compañía, una tienda online, etc.
- Es gratuito (software libre), si bien pueden existir gastos en el caso de utilizarlo en un servidor externo (hosting).
- El desarrollo se lleva a cabo de manera visual, pudiendo acceder al código fuente de los distintos archivos.
- Existe una gran cantidad de documentación y una comunidad que desarrolla plugins, temas, etc. Estos plugins y temas pueden ser gratuitos o de pago.
- Para utilizar WordPress de manera externa: <https://wordpress.com/es/>
- Para descargar e instalar WordPress: <https://es.wordpress.org/>

