

**UNIVERSIDAD COMPLUTENSE DE MADRID**  
FACULTAD DE CIENCIAS DE LA INFORMACIÓN  
Departamento de Periodismo III



**TESIS DOCTORAL**

**Tecnologías de la web semántica**

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

**Adolfo Antón Bravo**

Director

**Gonzalo Abril Curto**

**Madrid, 2016**

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS DE LA INFORMACIÓN

# TECNOLOGÍAS DE LA WEB SEMÁNTICA

TESIS PRESENTADA POR ADOLFO ANTÓN BRAVO  
PARA OBTENER EL GRADO DE DOCTOR

2015

Departamento de Periodismo III



UNIVERSIDAD  
COMPLUTENSE  
MADRID

---

# Agradecimientos

Este trabajo no habría sido posible sin el apoyo y el estímulo de mi director de tesis y amigo Gonzalo Abril Curto bajo cuya supervisión escogí este tema y comencé la tesis. A tantas personas, amigxs que en las etapas iniciales, intermedias o finales del trabajo, también ha sido de ayuda de numerosos modos, incluyendo feroces críticas y apasionados comentarios sobre el presente y el futuro de las tecnologías web.

También me gustaría agradecerle a Ada Lovelace, Grace Hopper, Paul Otlet, Claude Shannon, Vannebar Bush, Tim Berners-Lee, Dennis Ritchie, Richard Stallman y tantxs otrxs que han realizado avances tecnológicos en pro de la mejora de las condiciones de vida de las personas.

No puedo terminar sin agradecer a mi familia, a mis padres y hermana y en especial a Carmen y a Aitana, por haberme acompañado en los últimos meses de culminación de un trabajo que comenzó años atrás. Es a ellas a quien dedico este trabajo.

---

# Índice general

Agradecimientos	2
1. Resumen	7
1.1. Abstract	8
1.2. Palabras clave	9
1.3. Keywords	9
2. Introducción, metodología y fuentes de esta investigación	10
2.1. Herramientas	16
2.2. Acerca del lenguaje sexista	18
2.3. Otras consideraciones	18
3. La Web Semántica desde mi interés por la comunicación	20
3.1. La Universidad	21
3.2. Experiencias	23
3.3. La Web	24
3.4. Doctorado	27
3.5. Tecnologías de la información	28
4. Sociedad de la información	33
4.1. Cibernética	34
4.2. La sociedad de la información	36

---

4.3.	Redes Sociales y P2P . . . . .	38
4.4.	Género . . . . .	42
4.5.	Periodismo . . . . .	48
4.5.1.	Minería de datos . . . . .	53
4.5.2.	API de noticias o contenidos . . . . .	56
4.6.	Características de los nuevos medios . . . . .	60
4.7.	Leyes . . . . .	64
5.	Web . . . . .	66
5.1.	Orígenes de la Web . . . . .	67
5.2.	Hipertexto . . . . .	74
5.3.	Principios de la Web . . . . .	80
5.4.	W3C . . . . .	84
5.5.	Web 2.0 . . . . .	89
5.6.	Web 3.0 . . . . .	93
5.7.	Giant Global Graph . . . . .	95
5.8.	Datos abiertos . . . . .	99
5.9.	Accesibilidad del contenido web . . . . .	102
6.	HTML . . . . .	105
6.1.	Inicios . . . . .	106
6.2.	Sintaxis HTML . . . . .	108
6.3.	Marcas . . . . .	112
6.4.	HTML5 . . . . .	114
6.5.	Documento básico . . . . .	116
6.6.	Caracteres . . . . .	119
6.7.	Nuevos elementos . . . . .	123
6.7.1.	Audio . . . . .	124
6.7.2.	Canvas . . . . .	124
6.7.3.	Video . . . . .	125
6.7.4.	Formulario . . . . .	125

---

6.7.5.	Elementos semánticos . . . . .	128
6.7.6.	Elementos obsoletos . . . . .	130
6.8.	Lenguaje de marcado SGML . . . . .	130
6.9.	DTD, definiciones de tipos de documentos . . . . .	131
6.10.	Lenguaje de estilo de texto CSS y XSL . . . . .	139
6.11.	Diseño responsivo . . . . .	146
7.	Redes . . . . .	149
7.1.	Internet . . . . .	150
7.2.	TCP/IP . . . . .	151
7.3.	Tipos de redes . . . . .	152
7.4.	IPv4, IPv6 . . . . .	155
7.5.	HTTP . . . . .	158
7.6.	URL . . . . .	161
7.7.	API REST . . . . .	163
7.8.	DNS . . . . .	168
8.	Hardware y software . . . . .	172
8.1.	Unix . . . . .	174
8.2.	Lenguajes informáticos . . . . .	176
8.3.	Semántica . . . . .	180
8.4.	Principios de la programación . . . . .	182
8.5.	JavaScript, XMLHttpRequest y AJAX . . . . .	186
8.6.	Arquitectura web cliente-servidor . . . . .	189
8.7.	Servidor web . . . . .	190
8.8.	La interfaz gráfica: el navegador . . . . .	193
9.	Web Semántica . . . . .	195
9.1.	Metadatos . . . . .	196
9.2.	Capas de la Web Semántica . . . . .	199
9.3.	La Web Social . . . . .	201

---

9.4.	XML . . . . .	205
9.5.	DTD XML . . . . .	212
9.6.	RDF . . . . .	214
9.7.	Notation 3 Logic . . . . .	226
9.8.	Ontologías OWL . . . . .	227
9.9.	Anotaciones y <i>folksonomías</i> . . . . .	234
9.10.	RSS . . . . .	236
9.11.	FOAF . . . . .	238
9.12.	SKOS y entidades . . . . .	240
9.13.	Microformats, Microdata y RDFa . . . . .	242
9.14.	XML Schema . . . . .	249
9.15.	Realidad aumentada . . . . .	252
9.16.	SPARQL . . . . .	257
10.	Conclusiones	266
A.	Glosario	275
B.	Cronología	300
	Bibliografía	322

---

# Capítulo I

## Resumen

La Web Semántica es un conjunto de tecnologías que están en constante evolución, un fenómeno que desarrolla o extiende la Web y pretende ir un paso más allá en la recuperación de información, una base de datos gigantesca y viva que supone un campo de experimentación y pruebas para las humanidades y para el periodismo. El espacio destinado a alojar documentos *HTML* se ha convertido en el canal de acceso a información, contenidos y datos, en la interfaz estándar de relación de los usuarios con las tecnologías.

Han sido muchos los avances tecnológicos que han posibilitado construir la Web Semántica, provenientes de amplios campos del conocimiento como la ingeniería de software, la cibernética, las ciencias de la información, la biblioteconomía y documentación, la inteligencia artificial, las matemáticas o la estadística. Este trabajo pretende trazar un mapa conceptual de este universo de saberes que engloba en la Web Semántica para que sirva de base para vislumbrar los retos y las oportunidades de la Web Semántica en algunas parcelas concretas, como el periodismo. Por ello realizo un recorrido en seis capítulos por la sociedad de la información, la Web, *HTML*, redes, hardware y software y la propia Web Semántica.

Para ello he empleado una metodología que mezcla la crónica histórica, la observación participante y la historia de vida cuyo objetivo es acercarse a este fenómeno contemporáneo del que la sociedad puede participar, una participación que no debe estar exenta de reflexión y de espíritu crítico. Este trabajo busca servir como base para desarrollar investigaciones que analicen o aborden cuestiones epistemológicas, filosóficas, ontológicas o de otro tipo relacionadas con la Web Semántica.

## 1.1. Abstract

The Semantic Web is a set of technologies that are constantly evolving, a phenomenon that develops or extends the Web and aims to go a step further in information retrieval, one giant and living database which is a field of experimentation and testing humanities and journalism. The space designed to host *HTML* documents has become the channel of access to information, content and data, but also the standard user interface between users and technology.

There have been many technological advances that have allowed building the Semantic Web from different fields of knowledge such as software engineering, cybernetics, the science of information, documentation, artificial intelligence, mathematics or statistics. This paper aims to draw a conceptual map of the universe of knowledge that are encompassed in the Semantic Web to become a basis for a glimpse of the challenges and opportunities of the Semantic Web in specific fields, such as journalism. So I propose a six chapters structure along information society, the Web, *HTML*, networks, hardware and software and the Semantic Web itself.

I have used a methodology that combines the historical chronicle, participant observation and life history which aims to approach this contemporary phenomenon for society to participate, a participation that should not be exempted for reflection and criti-

cal thinking. This paper seeks to serve as a basis for developing research that analyze or address epistemological, philosophical, ontological or other kind related to the Semantic Web.

## 1.2. Palabras clave

Comunicación, información, web semántica, organización del conocimiento, tecnologías digitales

## 1.3. Keywords

Communication, information, semantic web, knowledge organization, digital technologies

---

## Capítulo 2

# Introducción, metodología y fuentes de esta investigación

*Tecnologías de la Web Semántica* es una apuesta por definir un ámbito del conocimiento, el de la Web Semántica, de cara a realizar futuras investigaciones que aborden o planteen la Web Semántica, en su conjunto o por separado, y necesiten un marco conceptual, un mapa introductorio a las tecnologías de la Web Semántica, a las que están definidas en multitud de documentos técnicos y a las que no aparecen en relación con las mismas, a algunas cuestiones transversales como la organización del conocimiento o el género, y a nuevas disciplinas para el periodismo como el periodismo de datos.

No pretendo abordar todas las tecnologías de la Web Semántica ni tampoco están todas las relaciones que cabe encontrar entre unas tecnologías y otras, entre fenómenos contemporáneos y pasados, entre diversas teorías y proyecciones futuras. Pero sí quiero sentar una bases mínimas para que quienes investigamos sobre información, comunicación, periodismo, epistemología, semiótica, estructura de la información, teoría de la información, etc., tengamos un plano por el que empezar a movernos. Hay tecnologías y

conceptos que se explican con mayor o menor detalle según he considerado importante de cara a comprender el conjunto.

He elegido no explicarlo en el título, ni siquiera con la coletilla de *introducción*. Para muchxs investigadorxs, la mayor parte de las tecnologías que abordo son habituales en su campo de estudio. Creo que no lo son en absoluto para el conjunto de la comunidad de las Ciencias de la Información.

El conjunto, el mapa, comienza con mi aproximación a las tecnologías desde el periodismo para continuar con el periodismo y las cuestiones que abordo en el capítulo dedicado a la sociedad de la información. A continuación viene la Web, el proyecto que iniciara Tim Berners-Lee y que es culpable de que este fabuloso fenómeno merezca la pena ser analizado y conceptualizado. De la Web paso a *HTML*, la estructuración del contenido, el hipertexto, y vuelvo la mirada al pasado para comenzar por las redes de telecomunicaciones que dieron paso a Internet, esa infraestructura global de comunicaciones, en el capítulo de redes. Todo ello no habría sido posible sin avances continuos y acelerados en hardware y software que se abordan en ese capítulo, para llegar finalmente a la Web Semántica, la evolución natural de la Web.

Cuando comencé el doctorado era un entusiasta, como hoy, de la Web, y quise conjugar esa afición con la más reputada de las palabras que escuché en el departamento, la semiótica, por lo que pensé en un estudio semiótico de la Web Semántica, y comencé a pensar en cómo abordar esta investigación partiendo del objeto de estudio [96], pero concluí que no se hayaba definido, por lo que comencé a trazar un esquema de mi objeto de estudio.

Como toda tecnología que funciona bien, que es libre, abierta y cooperativa, hay mucha documentación técnica y muchas áreas desde donde investigadorxs de todo el mundo participan. Siendo un área tan cercana a las Ciencias de la Información y al Periodismo, sorprende comprobar que no haya más participación desde nuestro ámbito. En estos

cuatro años de investigación, la Web ha evolucionado mucho por cuanto se han consolidado *HTML5* como lenguaje de documentos web, los dispositivos móviles como vía de acceso a la Web –desplazando incluso de la primera posición al ordenador personal–, ha aumentado la cantidad de datos disponibles a través de portales de datos abiertos y/o de datos vinculados y han emergido multitud de *API* de acceso a datos que vislumbran un presente muy interesante para el periodismo, la recuperación de la información y la comunicación. Sin embargo, la Web sigue siendo en esencia lo que era, al igual que la Web Semántica, y se hace si cabe más acuciante la necesidad de contar con un mapa conceptual de la Web Semántica que aborde tecnologías y fenómenos diversos más allá de documentos técnicos exclusivamente o casos de éxito.

Si consideramos la Web como un medio donde buscar y por tanto recuperar información, que lo es, podría haber hecho partícipe de esta investigación a los parámetros de esta disciplina, la *information retrieval*, para evaluar el estado de la Web Semántica. Sin embargo, me alejaría de la cuestión epistemológica que me ocupa y encontraría obstáculos propios de la recuperación de información como la cuestión de la autoridad de la Web [46]. Sería muy interesante investigar en esa dirección, considerando determinadas páginas representativas, como por ejemplo la *Wikipedia*, portales de datos abiertos o buscadores específicos.

Desde un enfoque sociológico, el análisis de las estadísticas que ofrece *Google*, a través de *Google Trends*, también ofrece una visión de la diversidad de la Web en la actualidad: qué temas interesan a los usuarios, qué páginas son las más visitadas en cada campo, cómo están realizadas, qué acontecimientos mediáticos ocurren para que se produzca un aumento de las visitas en tal o cual sentido, etc. Tampoco pretendo observar las tendencias de los internautas, ni qué información buscan ni qué información es la que obtienen, por lo que no he orientado la investigación en este sentido. Por cierto que el servicio no está activo en España porque desde diciembre de 2014 *Google News* dejó de operar en España al exigirle el gobierno el pago a los editores de contenido de una tasa por la reutilización de esos contenidos. Sin negociaciones de por medio, *Google News* dejó de

operar en España sin que haya habido un debate público que considero necesario sobre el uso y reutilización de contenidos web. *Google Trends* se nutre de los datos de las búsquedas en *Google* pero también de los hábitos de lectura que analizan desde *Google News*, por lo que de momento -septiembre 2015- no hay *Google Trends* en España.

Un término relativamente nuevo en el ámbito científico es el de *web science* o ciencia de la Web, popularizado por el propio creador de la Web, Berners-Lee, quien se pregunta por una posible metodología para el estudio de la Web [46], que abunda en el sentido de la recuperación de la información, el uso de analítica web, de fuentes primarias que ofrezcan datos, número de páginas creadas, edad de las páginas, actualización de páginas por dominios, número de personas conectadas, volumen de datos transmitidos, etc. y en el estudio de la Web a través de la teoría de grafos, donde las webs son nodos y los enlaces son las flechas hacia donde apuntan los nodos, una estructura determinada por los creadores de contenido, autores, programadores o diseñadores.

En este sentido, un estudio que analizó doscientas millones de páginas [60] concluyó que el noventa por ciento del tráfico web estaba conectado si se tomaban enlaces no direccionales, y cincuenta y seis millones de esas páginas estaban muy conectadas. De estos números resulta la figura de una pajarita, donde el centro de la misma está ocupado por el conjunto de enlaces muy conectado o *Strongly Connected Clusters*, a la izquierda hay un conjunto de enlaces que van hacia el núcleo central pero no vuelven, mientras que a la derecha hay páginas a donde van enlaces del núcleo central pero que tampoco vuelven. Esa estructura se repite en conjuntos más pequeños y en análisis de páginas agrupadas por temas o *Thematically-Unified Clusters*, por lo sugiere una estructura fractal.

La ciencia de la web también pretende agrupar los estudios o reflexiones en torno a cómo esta infraestructura de la información descentralizada puede servir para los requerimientos científicos, representacionales y comunicacionales, y para diseñar y promover principios de diseño del gobierno de estas estructuras [46], dado el carácter multidisciplinar de las ciencias, tecnologías y saberes implicadas en la Web y en la Web que está

por venir: protocolos técnicos, infraestructura, leyes, tendencias, ubicuidad, movilidad, nuevos medios, datos disponibles, privacidad...

Por tanto, la propuesta de realización de un mapa conceptual de la Web Semántica podría enmarcarse en la ciencia de la web como punto de partida para abordar cuestiones que están relacionadas con la recuperación de la información, la teoría de redes o las cuestiones epistemológicas que las propias tecnologías de la Web Semántica sugieren. La producción científica sobre la ciencia de la web es relativamente nueva y fundamentalmente técnica. Como sujeto observador que participa de las tecnologías web, pretendo una definición de este ámbito de las tecnologías de la información y del conocimiento que bebe de las ciencias de la información, de la biblioteconomía y documentación o de la ingeniería ontológica, por lo que hago un repaso histórico a algunos avances que han permitido a la Web Semántica concebirse como tal e incorporo al periodismo de datos como fenómeno relacionado, donde se ponen en práctica o se utilizan las propuestas de la Web Semántica.

He realizado esta aportación con la misma filosofía pero distinto formato al que un periodista conecta datos de distintas realidades para crear una historia, un relato, una narración o un mapa que sirva para conocer más y mejor las tecnologías de la Web Semántica, un trabajo que combina el estudio histórico y la observación participante orientado a articular en un mapa provisional las tecnologías de la Web Semántica, sus usos y los posibles desarrollos. La etnografía propone un acercamiento al quehacer cotidiano desde el extrañamiento y desde el tratar de ser unx más [7], y desde mi experiencia profesional, educativa e investigadora he tratado de obtener descripciones que he incorporado a la propia investigación [227].

En el ámbito de las Ciencias de la Información, pero también de las humanidades, de las ciencias sociales e incluso de las ingenierías y otras ciencias, considero necesario definir el fenómeno de la Web Semántica más allá de sus consideraciones técnicas o de las técnicas consideradas como propias de la Web y los debates que vienen de éste y otros ámbi-

tos de las humanidades y las ciencias sociales que abordan cuestiones epistemológicas más allá de las consideraciones eminentemente técnicas. De esta manera la realización de este mapa pretende evidenciar las conexiones que se pueden producir entre distintos ámbitos del conocimiento y algunas consideraciones sobre las mismas, superar el extrañamiento que produce la falta de contexto [227] tanto para estudiar casos de uso como si queremos reflexionar de manera crítica sobre cualquier aspecto relacionado.

El salto tecnológico supone un salto cultural y social con muchas imbricaciones y ramificaciones, que van desde la organización de la información y del conocimiento registrado hasta su recuperación; del escenario para el desarrollo de los contenidos multimedia a la asunción de las interfaces gráficas de usuario, de las conexiones con la cibernética, la realidad virtual y la inteligencia artificial al nacimiento de la realidad aumentada; de los servicios de información personalizados a las redes sociales; de la Web Social al periodismo de datos, entre otros.

En este trabajo trato de realizar un mapa conceptual de la Web Semántica, compuesto por algunas de las tecnologías implicadas, un recorrido histórico y una mirada al periodismo de datos como forma de expresión contemporánea del periodismo que está muy relacionado con la Web Semántica. Considero muy importante saber los precedentes, las tecnologías pasadas que permiten que las tecnologías presentes funcionen, y que pueden dar pistas sobre los retos epistemológicos que se producen y qué futuro nos vislumbran. En la La Web Semántica se expresa la interacción hombre-máquina, la archivística y la recuperación de la información, la cibernética, el cambio de paradigma en la figura del creador de contenidos, las licencias sobre los contenidos y los datos, el acceso a los datos, las herramientas de visualización, la capacidad de aprendizaje, las formas de aprendizaje, la digitalización y archivo de las actividades realizadas a través de medios electrónicos, la evolución de la sociedad de la información, la brecha digital, la ciudadanía o las cuestiones de género. Sin profundizar en cada cual, componen también el panorama de la Web.

Me gustaría traspasar las barreras del análisis de las tecnologías de la información para navegar por las fronteras de sus aspectos técnicos, epistemológicos, sociales, culturales, semióticos y prácticos, tal como considero que hicieron en su tiempo Walter Benjamin y Bertolt Brecht, el primero con el análisis del cine y los cambios que provocaba en el *sensorium* con la fragmentación de la experiencia moderna producida por los nuevos medios de comunicación masivos [76], y el segundo con el estudio de la radio, al que tan solo le faltaba transmitir para ser el "más fabuloso aparato de comunicación"[172].

Considero esta tesis como el acercamiento necesario que las ciencias de la información debíamos hacer de un proyecto amplio y diverso como es la Web Semántica, fenómeno en el que podemos poner en práctica el paradigma de la complejidad [165] donde como investigador reflexiono sobre la materia [?] y sintetizo el mapa en las coordenadas cuyos límites he marcado, reconociendo la provisionalidad e incertidumbre del conocimiento científico que propongo y abriendo el campo a diversas direcciones y relaciones situadas en el ámbito de la organización de la información y del conocimiento.

## 2.1. Herramientas

Para escribir este trabajo he utilizado un ordenador que corre *Debian GNU/Linux* con el escritorio de *Gnome* y como editor de texto *Emacs*. Este programa escrito por Richard Stallman en 1975 y lanzado como *GNU/Emacs* en 1984 en el marco del proyecto *GNU* se utiliza sobre todo por personas con alta cualificación técnica por las funcionalidades que ofrece para programadores, y también en ámbitos académicos y científicos por trabajar muy bien con *L<sup>A</sup>T<sub>E</sub>X*, un sistema de composición de texto orientado a la creación de documentos con una alta calidad tipográfica, que muestra muy bien las expresiones matemáticas además del propio texto y es capaz de trabajar con código fuente insertado y ejecutarlo y transcribir los resultados.

*L<sup>A</sup>T<sub>E</sub>X* se basa en *T<sub>E</sub>X*, un sistema de tipografía escrito por Donald E. Knuth para publicar su libro *El arte de programar ordenadores*, descontento con la calidad de la tipografía que presentaba. Al disponer de una licencia libre, enseguida contó con una comunidad de usuarios que ampliaron sus capacidades y hoy en día es un estándar de facto en la documentación técnica ampliamente utilizado entre la comunidad científica.

*Emacs* permite muchos modos de edición, que sirven tanto para escribir distintos lenguajes informáticos como para otros fines. En mi caso he elegido el modo *org-mode*, que permite organizar documentación, listados, esquemas, proyectos e incluso una agenda. Su elección ha venido dada por la facilidad para escribir textos extensos y crear estructuras jerárquicas que se despliegan o colapsan fácilmente para visualizar el esquema del documento, sus posibilidades de exportación a distintos formatos, entre ellos *HTML*, *L<sup>A</sup>T<sub>E</sub>X*, *Markdown*, *ODT* o *PDF*. Uno de sus puntos fuertes es el tratamiento de la bibliografía y las referencias bibliográficas, que se realiza con *BibT<sub>E</sub>X* a través de *RefT<sub>E</sub>X*, de tal forma que se trabaja con un archivo con extensión *.bib* que es texto plano y que contiene toda la bibliografía y desde el documento se enlaza con cada entrada de la bibliografía. Al realizar la exportación a otro formato y según el estilo de referencias bibliográficas elegido, *Emacs* incorpora la bibliografía. Otra característica relevante es la inclusión y ejecución de bloques de código, que he utilizado para las consultas *SPARQL*, cuyo resultado se incorpora al documento..

La radio pública de EE.UU., *NPR* (*National Public Radio*), creó el paradigma de escritura de contenidos periodísticos denominado *COPE* [145], *create once, publish everywhere* o *escribelo una vez y publícalo donde quieras*, para explicar la creación de un software de gestión de contenidos que permitía la distribución del contenido a lo largo y ancho de las emisoras de radio públicas de EE.UU., de tal forma que pudieran elegir el medio de salida de datos de su elección, ya fuera radio, web, papel o una aplicación web. *Org-mode* creo que es la aplicación de ese paradigma al ordenador doméstico y a los textos personales, del tipo que sea, ya que se realiza una escritura de formato sencillo que facilita la escritura y también su lectura. De esta forma se dispone del código fuente de cada

texto en todo momento y se convierte, según las necesidades, a *PDF*, por ejemplo, para producir el documento de la tesis, o a *HTML* si quisiera publicarlo como documento web. Tanto *org-mode* como *RefTeX* han sido creados por Carsten Dominik.

## 2.2. Acerca del lenguaje sexista

En aras de utilizar un lenguaje no sexista, he optado por utilizar "xs" como terminación del plural de todos los términos cuyo género puede ser masculino o femenino. Esta práctica que se ha denominado *desobediencia lingüística* [260] pretende no reforzar el masculino en el singular o el plural, de tal forma que se escribe "lxs autorxs." "lxs programadorxs", que leído se diría los y las autoras, los autores y las autoras o incluso las autoras, para revertir el masculino.

## 2.3. Otras consideraciones

Internet, Web y Web Semántica se escriben en mayúscula para referirme a cada uno de esos proyectos en su conjunto, mientras que página web, sitio web o webs van en minúsculas.

Las empresas o instituciones van en cursiva.

Las siglas y tecnologías compuestas por siglas van en cursiva y la primera vez que aparecen se explica el acrónimo. Para facilitar la lectura del texto, se ha incluido un glosario de términos donde se incluyen las siglas.

Las fechas más destacadas aparecen en la cronología adjunta.

Para las referencias bibliográficas se utiliza el estilo *APA*, elaborado por la *American Psychological Association*, asociación estadounidense de psicología.

---

## Capítulo 3

# La Web Semántica desde mi interés por la comunicación

Comencé a estudiar Periodismo en la Facultad de Ciencias de la Información de la Universidad Complutense de Madrid (*UCM*) en el curso 1993-1994. Mi apuesta por los estudios de periodismo eran una apuesta por la comunicación como herramienta de transformación social. Anteriormente había querido ser ciclista o banquero, pero los libros, y sobre todo los periódicos, fueron una fuente de información muy importante en mi vida adolescente que buscaba encontrar un camino que seguir. Siempre había tenido una idea romántica de los medios de comunicación en el amplio sentido, desde el periodista hasta quien realizaba la tirada hasta quien repartía el periódico o lo vendía, quizás por ser mi padre un linotipista de formación, con abundantes libros, revistas y otros formatos por casa, fruto de su anterior "afición" por las vietnamitas, multicopistas o los libros clandestinos, siempre por su trabajo. Y también, por tanto, una gran consideración de todos esos oficios.

### 3.1. La Universidad

La universidad parecía destinada a quienes sacábamos buenas notas en la antigua EGB y no habíamos ido a la formación profesional sino al instituto. Allí me apasionaron las ciencias, las matemáticas, la física, y la química, hasta que un año la docencia se relacionó mal conmigo, o yo con ellas, y brillaron lxs profesores de humanidades: Lengua, Literatura, Ética, Historia, Filosofía... En esta época se produjo un cambio del panorama mediático español, donde se sumaron nuevas propuestas periodísticas como *El Sol* o *El Independiente* durante el instituto, un canal autonómico llamado Telemadrid que todavía sigue aunque no es ni la sombra de lo que fue y nuevas cadenas de televisión, privadas, que nos sacaban de la televisión de dos canales. Sin duda todo estaba cambiando y me sentía parte del cambio.

Los medios de comunicación parecían un buen vehículo donde desarrollar mis ansias por transformar la sociedad y hacerla más justa. La decisión estaba tomada. En 1991, con la *primera Guerra del Golfo*, la *CNN* emitía su *Breaking News* (últimas noticias) las 24 horas del día, un formato nuevo, sorprendente, fascinante, y en inglés, y Telemadrid en vez de cortar la conexión de madrugada, reproducía la de Atlanta. Veíamos los bombardeos o los lanzamientos de misiles en directo, o varias veces en diferido, acompañados de gráficos, de reporteros de guerra y de periodistas de estudio. El diario *El Mundo* completó la parrilla mediática y los primeros años de facultad, si bien no me animaron a seguir siendo periodista, sí que me descubrieron que esto de la información y la comunicación era algo mucho más complejo y más interesante de lo que cabía imaginar. Las películas como la del caso *Watergate* quedaban en mera anécdota. Ángela Rodicio, entre otras reporteras de guerra, demostró que había dos tipos de periodistas, lxs que hacían periodismo y lxs que hacían entretenimiento, aunque luego esta distinción también la puse en cuarentena.

Entre huelgas y manifestaciones, corría la vida universitaria con la atención que algunas clases o algunos profesores merecían, y que han sido determinantes para llegar hasta

aquí y para hacer este trabajo. La *estructura de la información* explicada por el doctor Fernando Quirós, me impactó desde el primer momento, ya que nos situó rápidamente en el lugar que nos encontrábamos a través de las concentraciones mediáticas y de las personas que lideraban los conglomerados, los magnates de la comunicación mundial. Quienes transmitían la información tenían unos jefes, unos empresarios, unos intereses, y el periodismo no siempre salía adelante. Schiller, Mattelart, Ramonet o Chomsky fueron apellidos que pasaron a formar parte de mi vida, junto con los conceptos de *agenda setting* o *guardianes de la libertad*. La crítica al periodismo me hizo cosechar poco ánimo para practicar más adelante el periodismo, ya que pensaba que de herramienta de transformación pasaba a herramienta de transmisión del control. por lo que igual no había que hacer periodismo sino explicar lo que el periodismo nos estaba haciendo a todos y proponer formas distintas de hacer periodismo.

Participaba en una asociación de alumnxs llamada *Información y Libertad* y en unas jornadas de contrainformación que realizamos tuve la suerte de conocer a Gonzalo Abril, con quien no coincidí en clases durante la licenciatura. Quirós y Abril compartieron estrado y nos ilustraron sobre concentración mediática y sobre concentración del conocimiento. Si Quirós nos había situado en el juego del *Quién es Quién* y del *Mediapoly*, ahora se trataba de saber qué significados se encontraban detrás de todos los signos de la sociedad postmoderna, del pasado y el presente, de los estudios de semiótica, epistemología, filosofía, lingüística, literatura, imagen... No tuve la suerte de disfrutarlo como profesor en los más de siete años que tardé en completar la licenciatura y no dudé ni un segundo en apuntarme al programa de doctorado en el que estoy matriculado para disfrutar de él y de lxs otrxs profesorxs que he tenido la suerte de conocer, como Eva Aladro, Asunción Bernárdez o Wenceslao Castañares.

### 3.2. Experiencias

Pero antes de pasar al doctorado, probé a hacer periodismo en proyectos de todo tipo, en calidad de prácticas o becario, en revistas, fanzines e incluso en la primera -y no sé si única- *Televisión Complutense*. Y sobre todo, practiqué con muchxs de lxs que estábamos en esa asociación de la facultad periodismo alternativo en un proyecto de contrainformación que luego fue el germen del actual periódico quincenal *Diagonal*. Pretendíamos cubrir las noticias que no salían en los medios o que salían manipuladas en cinco ejes: antimilitarismo, antifascismo, okupación, antipatriarcal y laboral, todo con una perspectiva anticapitalista. Contábamos con medios limitados para cubrir las noticias pero muchas ganas y mucha gente que en distintos puntos de Madrid, de España e incluso del resto del mundo se ponía en contacto con nosotrxs para que transmitiéramos su lucha, su experiencia o su saber. Era realmente estimulante formar parte de esa especie de red social contrainformativa que permitía conocer muchas noticias que no salían en los medios u otros enfoques más "objetivos" de algunas que sí salían. Disponíamos de un pequeño local con dirección postal y un código postal, dos líneas telefónicas, una con fax y otra con teléfono que por la noche se convertía en un contestador telefónico que narraba en tres minutos la actualidad del día. Escribíamos un boletín diario que imprimíamos y enviábamos por fax a radios libres y otros proyectos de contrainformación del estado, grabábamos el contestador automático y, cada quince días, componíamos un boletín de noticias que se repartía gratuitamente en bares, librerías y espacios contraculturales de toda España, el boletín *Molotov*. Previo a ese reparto, había que doblar manualmente toda la tirada de boletines, que solíamos hacer entre muchxs mientras nos reuníamos, y luego empaquetarlo y con un carro de la compra ir a la oficina de *Correos* a enviarlo a tantas direcciones como peticiones tuviéramos. Cuento todo esto no solo como relato vital, sino también para ejemplificar los cambios que se producen cuando llegan los medios digitales, Internet y la Web.

He sido usuario de aparatos electrónicos, consolas de videojuego y ordenadores desde que se popularizaron para el ámbito doméstico, ya que siempre algún amigx o amigx

de amigx contaba con una, desde el *Ping Pong* de *Atari* a las cintas de cassette del *ZX Spectrum* de 48K, el *Amstrad* de 64K, *Commodore 64*, *Amiga*, *MSX* o el primer *PC*, pero nunca habían formado parte del ámbito de la educación o la información sino del entretenimiento. Los trabajos del instituto y de la facultad los realizaba en máquina de escribir y gracias a ello mecanografía, que no es poco.

### 3.3. La Web

También la facultad y la universidad disponía de ordenadores. La primera cuenta de correo electrónico la conseguí en el *Centro de Cálculo* gracias a la autorización de Quirós. Cuando la guerra llegó al corazón de Europa, en Yugoslavia, participé con esa dirección en conversaciones con todas las *ONG* que desarrollaban un compromiso contra la guerra, por la paz, de ayuda a refugiados o de envío de ayuda humanitaria, ya que trabajé de voluntario en la *ONG Paz Ahora*. Los ordenadores del Centro de Cálculo eran terminales virtuales en modo texto sobre *Unix* y el programa de lectura, *Pine*. Aquello era como la nube de ahora pero en modo texto, letras blancas sobre pantalla oscura. En la facultad pusieron un par de redes de ordenadores, una de ellas con *Macintosh* que todos queríamos utilizar porque era el ordenador que se utilizaba para maquetación, y muchos estábamos viendo una posible salida profesional en el campo del diseño, como de hecho así ocurrió. Con el navegador *Netscape Navigator* disfrutabas de la Web a color, con imágenes, tablas y enlaces que iban a otras partes. Vaya, ¡la Web!, un gusto conocerla, era como una película que uno vivía en primera persona. Recuerdo una de mis primeras navegaciones, tenía un aura de moderno y a la vez antiguo, como la *Metrópolis* de Fritz Lang, ya que este acceso al ciberespacio era con velocidades realmente bajas, expresadas en *kilobytes* en vez de *Megabytes*, con blanco y negro, 256 colores o algunos miles en el mejor de los casos. Pero, viniendo del mundo de la máquina de escribir, los paquetes de correos, el fax, los ordenadores de 48K y las cintas de cassette, estaba habituado a la espera y más si sabía que merecía la pena. Conectarse era algo que no hacías todo

el tiempo como ahora, que estamos multiconectados por varios dispositivos a la vez, a veces sin querer. Antes le dedicabas un rato, una hora, y apuntabas la información en el cuaderno y la compartías ávidamente con lxs demás que se habían quedado en clase o en la cafetería o en una reunión. Esperaba pacientemente que cargaran las páginas, viendo cómo se componían las imágenes y finalmente el documento electrónico, con tal cantidad de información alternativa disponible que me sentía tan extasiado como Stendal y fantaseaba sobre su potencial en el ámbito de los medios de información alternativa. Era como una gran biblioteca, o más bien, una gran hemeroteca de artículos de acontecimientos que estaban pasando en ese momento en otra parte del globo y cuyo único límite para su acceso era conocer el idioma, aunque normalmente siempre había una versión inglesa.

En 1994 el *EZLN* se presenta en Chiapas el 1 de enero, y las *ONG* y simpatizantes de todo el mundo empiezan a difundir el mensaje por Internet. También en 1994 se produjo en Madrid la reunión del *FMI*, Banco Mundial y *GATT* y las *ONG* españolas montaron una cumbre alternativa *las otras voces del planeta*, que, entre otras cosas, se tradujo en el nacimiento del proveedor de servicios de Internet (*ISP, Internet Service Provider*) *Nodoso.org*, que ofreció información y conexión a la cumbre, y a partir de entonces se podía contratar conexión a Internet y alojamiento de páginas web y otros servicios como el correo electrónico o *FTP (File Transmission Protocol* o protocolo de transmisión de ficheros). La Web e Internet iban a ser los medios de comunicación del futuro y estos parecían al alcance de la mano de cualquiera. De un boletín impreso, una radio libre local o un contestador automático, saltábamos a la red de redes, a pensar en global y actuar local. Si la primera navegación web me fascinó, el reto por conocer y participar de estas tecnologías me cautivó. Encontré que éramos unxs cuantxs lxs que, viniendo de distintos lugares, de distintos estudios, distintas edades e intereses, confluíamos en lo que ahora manejamos con relativa facilidad: una *comunidad* de personas interesadas por aprender y por compartir conocimiento de las tecnologías web. Aprender de la Web significaba muchas cosas, como por ejemplo aprender un lenguaje de edición de páginas *HTML*, un programa de edición del lenguaje, otro para manipular imágenes,

un sistema operativo que exprimir y por tanto un hardware que mejorar, una red a la que conectarse pero también una intrared en la que interconectarnos, *DNS*, protocolos y servicios, configuraciones diversas, comandos, órdenes, interfaces gráficas, formatos gráficos, seguridad... todo aquello que no había visto, ni por encima, en tres años de asignatura de *Tecnologías de la Información* de la facultad, donde solo manejamos el tipómetro y ni siquiera hicimos la transición del tipómetro a la maquetación asistida por ordenador.

Además, me sorprendió y me sigue sorprendiendo, lo cerca que estábamos en Ciencias de la Información de otros saberes como la Informática, la Filología, la Psicología o la Estadística, y lo lejos que nos colocaba el plan de estudios, la vida académica, la vida universitaria y las salidas profesionales, tan compartimentados y dominados por el Periodismo. No obstante, considero que manejábamos un conjunto de asignaturas que han aportado un amplio espectro de saberes relacionados con las ciencias de la información, como la teoría de la imagen, la historia del periodismo, opinión pública o lengua, pero también derecho o economía. Lástima que no se amplíe aún más.

Después de trabajar diseñando carteles para los movimientos sociales con parte de ese grupo de personas de la por entonces denominada *comunidad internauta* y de trabajar haciendo páginas web para distintas empresas, entre ellas *RTVE*, los movimientos sociales que trabajaban específicamente la información alternativa o interesados en difundir su información, montamos nuestro propio servidor conectado a Internet, *sindominio.net*, que todavía subsiste. En esa aventura del saber teórico y práctico apareció un nuevo elemento que se sumó a la sopa tecnológica, el ingrediente que también revolucionó la informática de consumo y mi relación con ella, el software libre, un conjunto de programas, incluido el sistema operativo, que son de libre uso, con un sistema de licencias que protege este uso universal. *Sindominio.net* fue el primer ordenador que vi con *Debian GNU/Linux*, y desde hace unos años no ha dejado de correr en todos mis equipos, incluido el que ha sido utilizado para este trabajo. Si disponíamos de un servidor propio, conectado a Internet, reconocido por un dominio y con un sistema que

podíamos controlar, no debían existir los límites para desarrollar proyectos de comunicación alternativa propios. Así nacieron la agenda de *Actualidad en Centros Sociales ACS* que sirvió de impulso a la *Agencia en Construcción Permanente* (posteriormente *Indymedia Madrid*), la radio libre por Internet *Luther Blisset*, los *Hackmeetings* (encuentros de experimentación y apropiación colectiva de las tecnologías libres), los *hacklabs* (estructuras continuas de experimentación y diversión) y las redes inalámbricas como *Madridwireless.net*.

### 3.4. Doctorado

En ese contexto fue cuando me matriculé en el programa de doctorado *Aspectos teóricos, estructurales y éticos de la comunicación de masas*, y donde Abril me volvió a fascinar con la *fragmentación verbovisual e integración simbólica, hacia una genealogía del texto informativo de la textualidad moderna*. Desde entonces quise realizar un estudio sobre la Web que contemplara los saberes que estaba adquiriendo durante los cursos de doctorado. Sin embargo, volví al mundo periodístico, casualmente a revistas de informática, por lo que seguí ampliando mi conocimiento del medio. Allí viví la crisis de las *puntocom*, que fueron las primeras empresas que se enfocaron a la Web, protagonistas y víctimas de la burbuja que cuando explotó parecía que iba a acabar con todo, aunque lejos de producirse este pronóstico, la Web evolucionó hacia lo que se denominó Web 2.0. También viví un acontecimiento trágico, los atentados del *11S* (conocidos en EE.UU. como 9/11). Entramos, sin quererlo, en la guerra global contra el terrorismo que también supuso el inicio del fin del romanticismo en Internet.

Posteriormente, retazos de esa comunidad que nos habíamos encontrado unos años antes decidimos montar una empresa para dar servicios de tecnologías de la información, basada en software libre, y así seguir practicando y extendiendo las tecnologías libres en proyectos propios y ajenos. En ese periodo enfoqué mi participación en la Web en el co-

nocimiento y difusión de los estándares y la accesibilidad en la Web, por lo que participé en la traducción al español [127] de los borradores de la segunda versión de las directrices de accesibilidad del contenido web, que finalmente se publicó como *recomendación W3C* [66] y alcanzó el estatus de estándar internacional *ISO/IEC 40500:2012* en 2012.

No fue hasta 2011 cuando volví a matricularme para realizar la tesis, y entonces di forma a mis ideas e inicié la investigación que denominé *Tecnologías de la Web Semántica*, por ser la Web Semántica lo que articula los avances tecnológicos de la Web y porque, si quería estudiar el signo y los significados de la Web, debía conocer su funcionamiento, sus orígenes y las tecnologías implicadas. Consideré y sigo considerando que mi objeto de estudio no se hallaba definido, y me propuse hacerlo no solo para realizar mi investigación [96], sino como objetivo de mi investigación. Hay estudios sobre la sociedad de la información, las redes sociales, distintas aplicaciones de la Web, muchos documentos técnicos y artículos técnicos sobre las distintas tecnologías, documentos *RFC*, recomendaciones *W3C*, estudios sobre redes, sobre hipertexto o sobre la Web y la Web Semántica desde sus principios técnicos, pero no he encontrado nada que realice lo que propongo, un mapa conceptual de la Web Semántica, conceptos, debates y tecnologías relacionados a través de los siguientes apartados: sociedad de la información, Web, *HTML*, redes, hardware y software y la propia Web Semántica.

### 3.5. Tecnologías de la información

Por primera vez en la historia la gente y las máquinas están trabajando juntos cumpliendo un sueño. Una unión de fuerzas que no conoce límites geográficos ni repara en raza, creencia o color. Una nueva era donde la comunicación verdaderamente lleva a unir a la gente. Esto es, el amanecer en la red.

[98]

La Web es una gran base de datos de documentos *HTML* con una gran cantidad de datos que se pueden comunicar a través de lenguajes semánticos comunes: *XML*, *RDF* y *OWL*.

El concepto de *Web Semántica* suele ir asociado a la evolución de la Web, ya que la Web cuando nació sería una web hipertextual, la web 2.0 sería una web que favorecía la participación de los usuarios y la Web 3.0 apostaba por ofrecer conocimiento. Lo que posibilita ese sueño de la Web inteligente, de las máquinas inteligentes por tanto, es la Web Semántica, tecnologías que semantizan el contenido web. La semantización ha ido evolucionando desde el documento HTML y los metadatos hacia los datos abiertos vinculados de *RDF*, *OWL* y *SPARQL*.

Se habla de sociedad de la información [70] por ser la información o los procesos de información automática los que predominan en todos los ámbitos de decisión de la sociedad. El término, auspiciado por los foros mundiales y diversas instituciones, ha sufrido cierto rechazo y se ha preferido el de sociedad del conocimiento [219], lo cual enlaza con la minería de datos y la Web Semántica, que buscan o modelan el conocimiento.

Nacido hacia 1973 [17] como constatación de la crisis de las industrias extractivas, crisis del capitalismo de estado o del socialismo de estado, acto de defunción de la *sociedad industrial* precedente. Es decir, la materia prima de esta sociedad, en buena medida, va a ser la información, transita de los bienes materiales a los inmateriales, de lo tangible a lo intangible, de lo sólido a lo líquido, y se reproduce periódicamente la metáfora cuando se enuncia la minería de datos [132], los datos abiertos [37] o el reciente *big data* [206].

Anterior a sociedad de la información es el término sociedad del conocimiento, formulado por Peter Drucker en 1969 en *La era de la discontinuidad* [151]. En el ámbito del aprendizaje automático se estudia también la optimización de las búsquedas en bases de

datos, de donde procede el término *Knowledge Discovery Databases* u obtención de conocimiento a partir de bases de datos [186]. La *Unesco* trata el tema del conocimiento en su informe *Hacia las sociedades del conocimiento*, lo considera la *evolución de la sociedad de la información*. En la Web, si al principio consiste en poner en común información y contenidos dispersos en programas, ordenadores y redes distintas, la Web Semántica pretende que humanxs y máquinas sean capaces de analizar, criticar o interpretar la Web, se pasa de una Web de significantes a una Web de significados.

En la cumbre del *G7* de Bruselas de 1995, donde también se reunieron los miembros de la *Comisión Europea*, el vicepresidente de los EE.UU. Al Gore propone una apertura del mercado de las telecomunicaciones estadounidenses si se produce reciprocidad en los otros países. Sería el paso definitivo a la liberalización del mercado de la industria de la información y del entretenimiento. De las reuniones del *G7* -luego *G8*- se traslada a la *ONU* (*UNGIS, United Nations Group on the Information Society*, grupo de las *Naciones Unidas* para la sociedad de la información) o a las recomendaciones del *FMI*. En las conclusiones del encuentro de la *Comisión Europea* se afirma:

El progreso en las tecnologías de la información y la comunicación está cambiando la forma en que vivimos: cómo trabajamos y hacemos negocios, cómo educamos a nuestros hijxs, estudiamos e investigamos, cómo nos entrenamos y cómo nos entretenemos. La sociedad de la información no solo afecta a la forma en la que las personas interactúan sino que también demanda más flexibilidad, participación y descentralización de las estructuras tradicionales.

[78]

*Flexibilidad, participación y descentralización de las estructuras tradicionales* aparecen como características de la sociedad de la información, cuestiones tratadas en la teoría matemática de la información [207] o en la cibernética [250]. Coincide el discurso de Gore

con el lanzamiento de *Windows 95*, el sistema operativo de *Microsoft* orientado para el incipiente mercado de consumo de ordenadores personales domésticos, que presentaba algunas características en la interfaz gráfica de usuario que se han mantenido hasta la aparición de las tabletas digitales: la barra de tareas, el botón de *Inicio* o la arquitectura multitarea de 32 bits. También llevaba incorporado el navegador *Internet Explorer*, que señalaba la importancia de la Web. El lanzamiento de *Windows 95* hereda los modos de anunciarse de *Apple*, cuya publicidad, bien la suya propia o la de las empresas tecnológicas, merecería investigación aparte. En esta ocasión, *Microsoft* elige la canción *Start me up* (ponme en marcha, si me pones en marcha nunca pararé) de *Rolling Stones* y lanzan una vídeo-guía que emula una *sitcom* (telecomedia de situación) con Matthew Perry y Jennifer Aniston, protagonistas de la por entonces popular serie *Friends*.

La Web llevaba poco más de cinco años en activo, desde aquel día en que fue inaugurada por Tim Berners-Lee en el laboratorio del *CERN* [23], pero estaba destinada a convertirse en el vehículo principal de la sociedad de la información o del conocimiento y contaminar los avances tecnológicos que iban a modificar la televisión o el cine, la llegada de la alta definición, el 3D (las tres dimensiones), la variedad de dispositivos de acceso a contenidos multimedia: portátiles, ultraportátiles, tabletas, *ebooks* o libros electrónicos, *smartphones* o teléfonos inteligentes, etc. Este conjunto de tecnologías que sirven para que accedamos a contenidos textuales, hipertextuales, multimedia o hipermedia (hipertexto y multimedia) [149] posibilita el gran logro de la Web no es solo técnico sino también social [34, p. 115]

El objetivo último del Web es apoyar y mejorar nuestra entretrejida existencia en el mundo. Nos agrupamos en familias, asociaciones y empresas. Tenemos confianza en cosas que están a kilómetros y no las tenemos en cosas que están a la vuelta de la esquina. Lo que creemos, aprobamos, aceptamos y de lo que dependemos es representable y, cada vez más, está representado en el Web. Tenemos que asegurar que la sociedad que construimos con el Web es la que pretendemos construir.

La evolución de la Web es concebida como Web Semántica [33], la base para la futura evolución de la Web que facilite las tareas de búsqueda, compartición y recombinación de información disponible, donde los ordenadores tengan "primero capacidad de describir, luego de suponer, y por último de razonar"[34, p. 169]. Para ello se ha producido una discretización de la información hasta convertirlo en fragmentos o unidades mínimas de sentido o datos, identificables por una *URI* y enlazables con otros datos a través de los datos vinculados abiertos. En este sentido, la Web pasa de ser la Web de documentos a la Web de datos, desde donde crear un flujo continuo de contenidos vinculados llenos de significado, de conocimiento.

---

## Capítulo 4

# Sociedad de la información

Los avances en el tratamiento automatizado de la información se precipitan a finales de los años 1940', tras la II Guerra Mundial, a consecuencia del desarrollo tecnológico que impulsan los distintos países para ganar el conflicto. Las máquinas se utilizan para realizar cálculos en la trayectoria de misiles o en el encriptado y desencriptado de mensajes pero no en el intercambio de información. Las bases se desarrollan en dos sentidos [70]: desarrollo de Internet, una red de interconexión de ordenadores y desarrollo de software aplicado a procesos de tratamiento de información.

Con la publicación 1948 *Una teoría matemática de la comunicación* en la revista *Bell System Technical Journal* [207], se sientan las bases para optimizar la transmisión de información cuando se emite una cantidad de información por un canal que cuenta con determinada capacidad. Se apoyaba en los estudios precedentes de principios del siglo XX varios autores: Andrey Markov, John Tukey, John von Neumann, Alan Turing y Norbert Wiener. Markov da nombre a las *cadena de Markov*, un tipo de proceso estocástico discreto en el que la probabilidad de que ocurra un evento depende solamente del evento inmediatamente anterior; Tukey definió el término *bit* (*binary digit* o dígito

binario) cuando trabajaba con von Neumann en el desarrollo de los primeros ordenadores; Turing realiza el esquema de la máquina universal, capaz de manipular símbolos sobre una cinta de acuerdo con unas reglas y en Norbert Wiener, quien sienta las bases de la *cibernética* en su ensayo *Cybernetics, or Control and Communication in the Animal and Machina* [250].

En la actualidad dos fenómenos muy relacionados, el de los datos masivos o *big data* y el de los datos abiertos u *open data*, apelan por copar el protagonismo. *Big data* pretende expresar el aluvión –/stream/– de datos digitalizados –discretos– que se producen en la actualidad. El mundo de las ciencias de la información y la comunicación no debería permanecer ajena a estas realidades, ni por asunción de que lo entiende ni por considerar que no le es propio.

Fuente/codificador/mensaje canal/descodificador/destino

#### 4.1. Cibernética

Cibernética, del griego *kybernetes* (piloto, timonel o regulador), es utilizada por Platón y en el siglo XIX por André Marie Ampère para referirse a los modos de gobierno. Wiener utilizó el término para referirse a la ciencia que se ocupa de los mecanismos de control y de comunicación en el animal –el ser vivo, sistema natural– y la máquina –el ordenador–, sistema artificial [82]. El nacimiento de la cibernética moderna se estableció en un congreso sobre la inhibición cerebral celebrado en Nueva York en 1942, del cual surgió la idea de plasmar una ciencia que se fija en el intercambio de conocimiento entre organismos vivos y artificiales a través de mecanismos de control.

Wiener trabajó junto al neurofisiólogo Arturo Rosenblueth en estudios sobre el cerebro humano, robots y sistemas automáticos, donde introducen el concepto de *retroalimentación*.

tación o *feedback*, y de *inteligente*, en tanto en cuanto un sistema procesa información, recibe información –recepción– y emite una respuesta –emisión–. La retroalimentación es un concepto fundamental para el desarrollo de la automatización.

La cibernética estudia el cuerpo humano como una máquina que cuenta con sistemas complejos de información –sistema nervioso, sistema hormonal– donde se produce comunicación eléctrica. Al contemplar la información, la retroalimentación y la regulación en todo tipo de sistemas, abren la puerta a las interpretaciones que Shannon y Weaver hicieron con los sistemas de información. Para obtener la respuesta deseada en un organismo, habrá que proporcionarle la información relativa a los resultados reales de la acción prevista para guiarle sobre las acciones que ha de realizar.

La cibernética se aplica a cualquier sistema considerado como sistema de comunicación. Esta cibernética de los años 1940 se le conoce como *cibernética de primer orden* para diferenciarla de la *cibernética de segundo orden* o *cibernética de la cibernética* que Heinz von Foerster expone en 1970 para referirse a los sistemas que son capaces de modificarse por sí mismos, sin necesidad de ser guiados (*driven*) desde fuera [176, cap. 1].

Numerosos autores han visto en Wiener un precursor de las teorías de redes, del espacio virtual de la comunicación, al proponer relaciones horizontales, descentralizadas, dinámicas y circulares, lo cual alcanza su culmen con el concepto de sistema autoorganizado en función de su experiencia y de su entorno en la cibernética de segundo orden [?]. Estos sistemas autoorganizados son los precursores de las ciudades inteligentes o *smart cities*, donde la ciudad se gobierna apoyada en miles de sensores que facilitan su regulación [216, p. 42]. Y más allá, se piensa en una *hipótesis cibernética* que ha reemplazado como hipótesis política a la *hipótesis liberal* [216, p. 68] al concebir los comportamientos biológicos, físicos y sociales como programados y reprogramables.

## 4.2. La sociedad de la información

Los conceptos de la teoría matemática de la información se trasladan a la producción industrial y se produce una revolución en los modos de producción de manufacturas, pero también favorece el desarrollo de las industrias culturales: la radio, la televisión, el cine, la prensa y la informática de consumo empresarial y gubernamental. La generación de riqueza vira de los sectores industriales a los sectores de servicios [17] y cobran más importancia las tareas de servicio en los sectores industriales. Las tecnologías de la información y la comunicación (*TIC*) se convierten en la una nueva fuente de ingresos, motor del desarrollo económico, social, cultural y científico. Según Harvey Brooks y Bell, tecnología significa el uso del conocimiento científico para especificar modos de hacer cosas de una manera reproducible [70, p. 60]. No se incluyen todavía la Web o la Web Semántica, pero sí a la microelectrónica, la informática (máquinas y software), las telecomunicaciones, televisión, radio y la optoelectrónica, además de la ingeniería genética.

En línea con la importancia de la tecnologías se piensa en una sociedad tecnológica basada en soportes y conexiones, frente al término de sociedad de la información, a no ser que se tome el término información en su "sentido shannoniano" [113, p. 67], equivalente a señal radioeléctrica, desprovista de significado. Esta sociedad tecnológica tiene en la Web uno de sus principales soportes.

El impulso a la sociedad de la información iniciado por Al Gore se distribuyó por toda la cadena de organizaciones creadas tras la segunda guerra mundial, como la *Comunidad Europea* –ahora Unión Europea–, la *OCDE*, las *Naciones Unidas*, el *Banco Mundial*, el *Fondo Monetario Internacional* o la *Organización Mundial del Comercio (OMC)*.

La *Unión Internacional de Telecomunicaciones (UIT)* de la *ONU* decide en 1998 realizar sendas cumbres mundiales con ese nombre, la primera en 2003 y la segunda en 2005. De acuerdo con la declaración de principios de la *Cumbre de la Sociedad de la Infor-*

mación, llevada a cabo en Ginebra (Suiza) en 2003, la sociedad de la información debe estar centrada en la persona, integradora y orientada al desarrollo, en que todos puedan crear, consultar, utilizar y compartir la información y el conocimiento, para que las personas, las comunidades y los pueblos puedan emplear plenamente sus posibilidades en la promoción de su desarrollo sostenible y en la mejora de su calidad de vida, sobre la base de los propósitos y principios de la *Carta de las Naciones Unidas* [221].

El ordenador y los dispositivos móviles se han erigido en pocos años en tótem de nuestra época y en prueba de capacitación tecnológica, al igual que el acceso a la lectura y la escritura fue en otros siglos la clave del ascenso cultural y socioeconómico. En la sociedad hipertecnificada y teleconectada la fábula de que las naciones y los estados van dejando de ser las formas determinantes de la vida social [95] se hace más real. El ámbito privado deviene público [88], el ocio se convierte en trabajo [134] y el consumo en producción [208]. Con adjetivos como inteligentes, ecológicos, de bajo consumo, eficientes o limpias se esconde un gasto energético enorme [18] en la producción de dispositivos electrónicos, además de los conflictos extremos –guerras, violaciones de derechos humanos, golpes de estado– que viven los territorios donde las industrias extractivas consiguen los materiales que componen estos productos y de los residuos altamente contaminantes que se originan [115].

Otra fábula que mezcla consumo energético, sistemas de control, hiperrealidad e inteligencia artificial es la película *Matrix*, donde se produce tras un conflicto entre máquinas y humanos por la crisis energética [245]. Los humanos se convierten en fuentes de energía y su vida se desarrolla en una aplicación de hiperrealidad virtual, *Matrix*, donde cuentan con una vida "normal": trabajo, casa, ocio, consumo, etc. La conciencia de existir se ha programado en un no-vivir, algo que de forma más abrupta y real sufren quienes habitan o trabajan en las zonas de extracción de coltan, uno de los minerales más codiciados por las empresas de tecnología, en la República Democrática del Congo; o en los vertederos de chatarra tecnológica de Costa de Marfil. Pertenecen, aunque será su mayor problema el de definirse así, al cuarto mundo [70, t. 2], allí donde la brecha

no es digital y produce heridas peores, muertes. También están incluidos en este cuarto mundo las personas de los países desarrollados que no tienen la opción de conectarse o que pudiendo conectarse, carecen de las destrezas necesarias.

Otro término muy empleado para referirnos al desequilibrio o la desigualdad en el acceso y uso de las tecnologías es el de brecha digital, del inglés *digital gap*, lo cual impide el acceso a la información y el conocimiento. Tiene múltiples dimensiones, no obedece a aspectos exclusivamente de carácter tecnológico, sino que es un reflejo de los factores socioeconómicos que afectan a los países empobrecidos que se traduce en una dependencia tecnológica de los países más desarrollados.

Más que brecha digital habría que hablar de brechas digitales, como la que se produce en el acceso a los contenidos de calidad [113, p. 42], por ejemplo, en televisión, pero también en el acceso a publicaciones científicas o en medios de información web de pago. La información de calidad se mezcla con la información de baja calidad y la abundancia de información produce desinformación, los contenidos gratuitos, que no libres, esconden el pago indirecto en términos de publicidad o datos personales para reforzar la analítica de las empresas, parte de ese universo de datos que corresponde al *Big Data*.

### 4.3. Redes Sociales y P2P

Las redes sociales en la Web son sitios web y/o aplicaciones para móvil que permiten a los usuarios compartir información. El término *red social* se acuñó en los años 1950 y son varios lxs investigadorxs que utilizan el término red como metáfora para representar las relaciones que se producen entre miembros de una comunidad de estudio [182], ya sea una familia, una población u otro tipo de comunidades.

Una de las teorías inmersas en las redes sociales es la de los *seis grados de separación*, que

sostiene que todo el mundo está conectado entre sí a través de no más de seis personas intermediarias [147]. Fue propuesta en 1929 para evidenciar lo interrelacionados e interconectados que comenzaban a estar lxs ciudadanxs de las sociedades modernas en el siglo XX, con los avances en las comunicaciones de transporte de personas y mercancías y el rápido intercambio de información. Karinthy no se refería solo a personas, sino también a pensamientos, conocimientos, cosas o acontecimientos, de tal forma que la red pudiera representar no solo las relaciones entre personas, sino también lo que hacen o piensan, todo un anticipo de las redes sociales del presente.

La demostración de la teoría la intentaron dos matemáticos en la década de los 1950' del siglo XX, Ithiel de Sola Pool (*MIT*) y Manfred Kochen (*IBM*), y no tuvieron éxito. En 1967, el psicólogo Stanley Milgram realizó un experimento de campo donde un número de personas debía hacer llegar una carta a otra a través de conocidos que pudieran conocer al destinatario sólo utilizaron un promedio de entre cinco y siete intermediarios. De ahí viene la expresión *6 grados de separación* en una teoría que él llamo *el problema del mundo-pequeño*.

La expresión se popularizó en la Web después de que Brett C. Tjaden publicara un juego de ordenador donde usaba la base de datos de *IMDB* (*Internet Movie DataBase*, base de datos en Internet de películas) para representar las conexiones entre los actores. La revista *Time* lo consideró uno de los diez mejores sitios del año y lo denominó *el oráculo de Bacon en Virginia*, por las conexiones que alcanzaba a tener el actor Kevin Bacon. Duncan Watts realizó un experimento en 2001 con el correo electrónico y después de recopilar datos de 48.000 personas en 157 países y 19 objetivos comprobó que en número de pasos era, efectivamente, seis.

En la primera era de la Web, en la *Web 1.0*, una de las primeras redes fue *classmates.com*, compañeros de clase, creada por Randy Conrads en 1995 para que los antiguos estudiantes pudiesen recuperar el contacto con sus compañeros de clase. Otra fue *Six Degrees* (seis grados), que tomaba el nombre de la citada teoría y pretendía ser una red social de

intercambio de contenidos entre personas. Hubo muchos sitios similares que surgieron para conectar personas con todo tipo de aficiones o de diversas procedencias, aunque sin la trascendencia de las actuales, dado que las condiciones de implantación de la tecnología en todos los sentidos –experiencia, redes, software y hardware– era muy diferente. Muchas duraron lo que la *burbuja de las puntocom* [254] tardó en explotar.

Luego vinieron *MySpace* para grupos o aficionados a la música, *LinkedIn* para contactos profesionales, *Facebook* para intercambio de contenidos o *Twitter* para *micro blogging*. En algunos países se usan más unas u otras y suelen estar sujetas a un feroz mercado de compras y ventas por parte de sus competidores naturales o de otros actores de Internet, aparecen nuevas que aprovechan nichos desconocidos o de alguna funcionalidad de las tecnologías que antes no permitían en la misma medida.

Normalmente todas ofrecen una *URL* del usuario que puede utilizarse como página personal, desde donde crear o enlazar contenidos, relacionarse con otros usuarios de la misma red y compartir intereses comunes. Del listado de redes más utilizadas [261] destacan en la actualidad *Google+* y *Facebook* para compartir información, *Twitter* para *microblogging* o *Instagr.am* para fotografías. Todas ellas están por encima de los cuatrocientos millones de usuarios.

Destacan también, no por número de usuarios sino por la filosofía de los proyectos, redes sociales que no obedecen a intereses comerciales como las anteriores, que favorecen la privacidad y el uso de los propios datos, que fomentan la ética de compartir y de acceso a los datos y que están construidas sobre software libre y pretenden el intercambio de datos con otras redes que cumplan protocolos abiertos, como *n-1.cc* o *GNU social*

En línea con las redes distribuidas y horizontales se encuentran las teorías *peer to peer* o *P2P* (igual a igual, par a par), concepto que se expresa en el intercambio de archivos en redes de intercambio como *torrent* [15] que se extiende a las nuevas relaciones sociales que se producen en la Web y otras redes o aplicaciones, como por ejemplo la *Wikipedia*

y su modelo de edición colaborativa. Si Marx identificó en las plantas productivas de Manchester el sello de la nueva sociedad capitalista, en las redes *P2P* se transforma la vida social, política y económica [14]. Las personas o *agentes humanos* se conectan con otras a través de *agentes máquina* voluntariamente, de forma abierta, libre, participativa y compartida para producir valores orientados al bien común o *commons* [204].

Para que se produzcan procesos *P2P* [14] se necesitan cinco requerimientos:

1. Infraestructura de conexión que permita el acceso a cualquier tipo de objeto de datos, como las redes ciudadanas inalámbricas.
2. Sistemas de información y comunicación alternativa que permitan comunicación autónoma entre agentes humanos que cooperan y que no dependen de los canales clásicos de publicación o retransmisión.
3. Infraestructura de software que permita cooperación autónoma global, como los *blogs*, *wikis* o redes sociales, para crear capital social y confianza, conformando grupos globales que generen contenido de valor en cuya producción y distribución no medien intereses comerciales.
4. Infraestructura legal que fomente el valor de uso y lo proteja de la apropiación privada, como el proyecto *GNU* y la licencia *GPL*, utilizada en proyectos de software libre e inspiradora de otras licencias de software y de otros tipos de contenidos, como las *Creative Commons* o las de datos abiertos.
5. Infraestructura cultural.

Este último es el punto más importante ya que los anteriores se sustentan en una infraestructura física –redes conectadas, hardware– y lógica –software y licencias– que paradójicamente han sido creadas en el sistema capitalista desde el que se propone una

alternativa, mientras que para que el último exista se debe pasar por una cultura de masas que haya distribuido el conocimiento y el saber y que haya propiciado cambios ontológicos del ser, epistemológicos del conocer y axiológicos del valorar, sin los cuales no habría comenzado el *individualismo cooperativo* que funciona en los proyectos *P2P*.

#### 4.4. Género

En la historia de la Web, de la informática, las redes, el software o el hardware, han participado muchas mujeres y han tenido una gran importancia en muchas situaciones. La cuestión de género no se basa tanto en el porcentaje de mujeres que han participado o de cuán importante ha sido su contribución, o si han dominado unas tecnologías u otras, sino cómo se han separado los roles de género en la sociedad capitalista que propone los avances tecnológicos, qué papel han podido ocupar con respecto al otro género, el masculino, cuál ha sido su participación en un sistema que le excluye, y qué representaciones se producen a consecuencia de o a pesar de su participación o su ausencia.

Una de las paradojas en la historia de la informática se da con Augusta Ada King, condesa de Lovelace, hija del poeta Lord Byron y conocida como *Ada Lovelace*, una anomalía de la sociedad victoriana, quien se considera autora del primer lenguaje de programación. Como matemática, trabajó amistad con Charles Babbage, quien proyectó una máquina analítica, un ordenador mecánico de propósito general. En 1842 se publicó un libro en francés de Luigi Federico Menabrea sobre el invento, tras lo cual Babbage le propuso a Lovelace que lo tradujera, y añadiera sus propias notas. Lovelace escribió tres veces más contenido que el original [158], donde se incluía un algoritmo que debía correr la máquina para calcular los *números de Bernoulli*, lo cual se considera el primer documento donde se detalla un lenguaje de programación [218]. Pero además Lovelace pensó que esa máquina podría servir para muchas más cosas que para realizar cálculos matemáticos [187, p: 152], que también se podría componer música, producir gráficos y

utilizarse para otros usos prácticos y científicos.

En 1979, el *Departamento de Defensa* de EE.UU. de Norteamérica desarrolló un lenguaje de programación orientado a objetos, concurrente y fuertemente tipado. que llamó *Ada* en su honor. *Ada* se utiliza mucho en la aeronáutica, la gestión del tráfico aéreo y la industria aeroespacial. Otro invento importante fue el que realizó Grace Hopper en 1952, quien desarrolló el primer compilador para un ordenador electrónico, el *AO*, un traductor del lenguaje máquina al lenguaje matemático, lo cual abría la puerta a la programación independiente de la máquina, como era habitual hasta entonces. También se le considera la madre de los lenguajes de programación al inventar *COBOL* en 1959 y de popularizar el término *bug* (insecto) para referirse a un error informático, ya que encontró un insecto en un relé de su ordenador *Harvard Mark II* que le hizo cometer un fallo. Desarrollar el compilador y el lenguaje de programación fueron dos caras de la misma moneda, la evolución del paradigma de programación ensamblador a lenguajes independientes de la arquitectura del ordenador.

De Jean Jennings Bartik pasó de realizar cálculos a mano de las trayectorias de misiles a ser una de las cinco mujeres que trabajaron como programadoras y operadoras del primer *ENIAC* en 1945. Hasta entonces los hombres no se interesaban por este tipo de trabajos [212] de operadora o programadora, preferían construir el hardware, hacer los circuitos, diseñar la maquinaria, etc. Después de la *IIGM* se fueron a trabajar en el *UNIVAC*, el primer ordenador comercial, donde coincidieron con Hopper.

Jean Sammet, que participó con Hopper en el desarrollo de *COBOL*, desarrolla el lenguaje *FORMAC* (*FO*rmula *MA*nipulation *C*ompiler, compilador de manipulación de fórmulas) en *IBM* que es considerado el primer lenguaje de programación que interpreta símbolos algebraicos y los traduce en código. Ida Rhodes diseñó el lenguaje *C-10* para el *UNIVAC I*, un nuevo ordenador que se utilizó para calcular el censo de EE.UU. También Elisabeth Holberton estuvo en *ENIAC*, *COBOL*, *UNIVAC* y en *BINAC*, *Binary Automatic Computer*, primer ordenador con programa almacenado en EE.UU.

*COBOL* fue el punto de inflexión [212] de la consideración del software como algo tanto o más importante que el hardware, ya que permitía dar instrucciones en lenguaje natural, en inglés, a un ordenador, que después de compiladas servían para cualquier tipo de hardware.

Mary Allen Wilkes, desde el *MIT*, escribe el sistema operativo para *LINC* (*Laboratory Instrument Computer*, ordenador instrumento de laboratorio), el primer ordenador de tamaño doméstico, en 1961. Poco después, en 1965, construye su propio ordenador doméstico y es considerada la primera usuaria de un ordenador doméstico. En 1985 Radia Perlman de *Digital* desarrolla el algoritmo del árbol de expansión o *spanning tree protocol*, que mejora las conexiones de una red amplia como Internet, por lo que se le ha llamado *la madre de Internet*.

En este siglo, Corinna Bath se ha interesado en las conexiones entre los robots sociales y los agentes de software emocionales desde una perspectiva feminista, de crítica a la tecnología y de crítica de las condiciones previas que crearon y aplicaron estos artefactos [276, p. 53]. En este sentido, también se ha preguntado cómo esta representando el conocimiento la Web Semántica [184], si tiene género y si desde una epistemología objetiva, puede representar un único y verdadero punto de vista de la cultura dominante. Dado que las construcciones simbólicas relacionadas con el género que existen en nuestra cultura tienen su reflejo en los desarrollos tecnológicos y en el uso que hagamos de éstos, cómo ha se planteado la arquitectura que sustenta la representación del lenguaje y su significado son cruciales para conocer si tiene en cuenta distintas perspectivas en su desarrollo que permitan no dejar de lado la experiencia del mundo de otras personas y evita imponer visiones homogeneizadoras, lo cual contribuiría a crear *mundos vivibles* [129].

El trabajo de Donna Haraway nos recuerda que la ciencia es uno de los productos culturales de la civilización occidental y como tal no puede ser analizada al margen de las relaciones de poder que le dieron origen [131, p. 76]. Una clase de procesos de género se

basan en los estereotipos de género y en la división del trabajo, sin cuestionamiento sobre las políticas de dominio y las relaciones de poder implícitas [184], el actual orden de género estructural y simbólico se inscribe en los dispositivos informáticos y se reproduce por el uso de los mismos, por lo que la ruptura de la reproducción de la desigualdad de género en los dispositivos informáticos ha de llegar al nivel epistemológico y ontológico en el que los conceptos tecnológicos se basan.

En resumen, las dos líneas de trabajo conectadas, la de los productos y la epistemológica, se encuentran en consonancia con las teóricas feministas como Haraway, Karen Barad y Lucy Suchman que se basan en las implicaciones políticas de las tecnologías y en las imbricaciones del género en los procesos de construcción de significado de las tecnologías semánticas.

El concepto de *Conocimiento situado* [128] hace referencia a la postura epistemológica que nace de una crítica feminista a la misma epistemología feminista, propone hablar de los objetos de estudio desde el lugar del que cada cual parte, es decir, del contexto cultural y también de la subjetividad. El conocimiento es real, objetivo, en tanto lo es desde el punto de vista de quien lo genera [128, p. 327], por lo que trata de romper la desigualdad que surge a la hora de producir conocimiento entre qué sujeto puede conocer y qué sujeto no.

Haraway, al igual que Jacques Derrida o Jean-François Lyotard, contemplan una concepción de la escritura que resiste las prácticas de autoría falocéntricas y academicistas [178, p. 1], situando en primer término las condiciones de lxs autorxs para hacer visible el aparato de producción de autoridad que cualquiera lleva consigo en su discurso, la *escritura cyborg* [130].

Si la escritura es inseparable de la propia situacionalidad y de los sistemas de liberación y dominación, y por tanto la literatura es también un sistema de liberación, por qué no la Web Semántica. Si el enfoque de la formalización de las tecnologías de la Web Semántica

se basa sobre todo en una epistemología objetiva, la pregunta de Bath [184] es si la Web Semántica representa la *única y verdadera* visión de la cultura dominante, es decir, el conocimiento hegemónico del mundo occidental, blanco, masculino y de clase alta.

Desde instituciones oficiales como la *UNESCO*, en su informe *Ciencia, Tecnología y Género* abunda en la línea del feminismo por la igualdad que busca la igualdad de oportunidades, por ejemplo, para incorporar un mayor número de mujeres a la fuerza de trabajo científica y tecnológica a través de estrategias, políticas, programas e indicadores a nivel internacional, nacional o regional [220]. Las políticas científicas y tecnológicas que buscan la igualdad no se plantean ni qué igualdad es la que se busca ni qué lenguaje emplea ni que situaciones y construcciones se generan en esa igualdad. Además, de alguna manera, se parte de la consideración de que la participación de las mujeres en las tecnologías no ha ocurrido, se han ausentado, y ahora tienen la oportunidad de hacerlo, cuando hemos recorrido algunos ejemplos significativos de la historia reciente donde se constata que han estado muy presentes [187, p. 17], que rompen con una visión predominantemente masculina de las tecnologías.

La cibernética supuso una revolución para las identidades de género, expresado en el término redundante *cyborg*, *cybernetic organism* u organismo cibernético, pues todos los organismos son comprendidos por la cibernética. Acuñado en 1960 por Manfred E. Clynes y Nathan S. Kline, se referían a un ser humano mejorado que pudiera vivir en entornos extraterrestres ante la posibilidad de que la carrera espacial obligara al humano a habitar en otros planetas. Lo interesante para las teorías feministas o ciberfeministas es que los *cyborg* no son ni hombres ni mujeres en el sentido de género, es decir, que no tenían género definido. Esta nueva identidad *cyborg* podía romper con la cultura de las mujeres sumisas y esclavas, sometidas al hombre, y cuya máxima expresión se muestra en la parábola de *Poseídas* [187], donde la última mujer real es asesinada por una doble artificial que servía para cumplir el sueño de los maridos de Stepford de una feminidad cautiva.

La cuestión de género está en relación con el nivel de conciencia, de experiencia propia y por tanto de entendimiento, comunes todos a la inteligencia artificial y la cibernética. En 1950 Alan Turing creó el test que lleva su nombre, el *test de Turing*, que se utiliza para medir la inteligencia de una máquina, modificando un viejo juego de salón en el cual dos jugadores – *A*, hombre; *B*, mujer– se ocultan a un tercero que hace preguntas sobre los dos para descubrir cuál es la mujer. En el test de Turing, un jugador es sustituido por *C*, una máquina, que ha de hacerse pasar por *A*. Ninguna máquina lo ha pasado.

Sobre el género indefinido, los datos de *DBpedia*, datos semánticos en *RDF* extraídos de los datos estructurados de la *Wikipedia*, no aportan información del género de las personas que recoge porque no los contempla *Wikipedia*, por lo que son los agentes humanos y no los agentes máquina quienes han de discernir sobre el género de quienes aparecen en la *Wikipedia*. El vocabulario *FOAF* recoge la propiedad `foaf:gender` para referirse al "género del agente, normalmente *masculino* o *femenino*, aunque no necesariamente"[59]. Se trata de una propiedad en estado de pruebas `-/testing/-` y que si se utiliza implica ser un *Agent*, es decir, pertenece a ese dominio. Un *agent* puede ser una persona, un grupo, un software u otro tipo de artefacto físico. *Agente* es una clase con *status* de estable (`stable`), y tiene como subclases a *Group* (grupo), *Person* (persona) y *Organization* (organización). Según la descripción de la propiedad `foaf:gender`, relaciona un agente, normalmente persona, con un literal que representa su género, y avisa que aunque `male` (masculino) y `female` (femenino) son los valores típicos, se pueden usar otros valores porque esta propiedad no pretende capturar la gran variedad de conceptos biológicos, sociales y sexuales asociados con el término *género*. Tampoco la consideran una propiedad estática, ya que la misma persona puede tener distintos valores en tiempos distintos, y explican que han tratado de ser respetuosos con la diversidad sin adentrarse en catalogar o enumerar esa diversidad. Advierten, finalmente, que se debe hacer un uso cauteloso de la propiedad, de los valores que se permiten, dada la publicidad de esos datos y lo sensibles que pueden ser en determinados contextos, máxime cuando es el agente de *usuarix* quien se define y por tanto también puede mentir interesadamente en su definición.

La Web ha revolucionado la epistemología en cuanto supone la Web Semántica una posibilidad para la automatización del procesamiento de información, el almacenamiento de contenido y la creación de contenido re combinado. Los metadatos y las ontologías ofrecen una apreciación del contexto, con qué métodos se crearon y a qué fines responden, ya sean políticos, económicos o sociales, pero cabe preguntarse si las tecnologías actuales permiten abordar la distinción entre conocimiento y superstición, entre pensamiento único y pensamiento marginado, entre distintas visiones de la realidad y qué restricciones encuentran estos discursos [184].

## 4.5. Periodismo

El cambio tecnológico de la Web coincide con un largo periodo, inacabado, de crisis del periodismo [2]. La máxima de que cuando se producen convulsiones culturales, el periodismo se convierte en la excusa para hablar de muchas otras cosas [100] no debe esconder que en la práctica del oficio, la preocupación por su *making of* se ha de meditar sobre los modos de producción de la noticia, desde las cuestiones éticas hasta las puramente técnicas. Se produce un cúmulo de circunstancias [117] que van desde la audiencia a los ingresos, la atención de lxs lectorxs, la (pérdida de) confianza o la autoridad (cuestionada), razones más que probables por las que distintos periodistas han apostado por un cambio en la forma de publicar noticias [135].

El fenómeno del periodismo de datos resulta uno de los más interesantes de los últimos años, pues sienta las bases en la misma época y en las mismas tecnologías que permitieron el salto tecnológico que ha supuesto la Web. Aunque se puede retroceder hasta los años 1960 para encontrar a su precedente, el periodismo de precisión, se trata de una disciplina eminentemente asociada al cambio tecnológico de la Web. El periodismo de precisión nace con Philip Meyer, quien estudiaba métodos de investigación social durante su estancia en *Niemen Lab* de la *Universidad de Harvard* en el curso 1966-67.

En un viaje a Detroit con *Free Press* para informar sobre las revueltas de la ciudad de 1967 [8], se propuso utilizar métodos de análisis de las ciencias sociales, software de análisis estadístico y los últimos ordenadores disponibles. Construyó una historia periodística que rompía con el discurso oficial sobre las personas implicadas en la revuelta y el método se convirtió en una nueva disciplina periodística del periodismo estadounidense que se definió como tal durante el curso 1969-1970 [160]. El nombre proviene de Everette E. Dennis de la *Universidad del Estado de Kansas (Kansas State University)*, quien realizó un seminario sobre *nuevo periodismo* en la *Universidad de Oregon*, y citó el trabajo de Meyer como *periodismo de precisión*, para contrastarlo con el trabajo más literario de periodistas como Tom Wolfe o Jimmy Breslin.

Aunque la base es similar, ya no se necesitan habitaciones enteras para alojar un *mainframe* ni varias horas para complejas operaciones, pues las herramientas caben en la palma de la mano y la abundancia de datos, a menudo abiertos y a veces vinculados, facilita una parte del proces. Luego, conceptos como las *API* y la multitud de herramientas de representación y visualización de los datos hacen el resto.

Los periodistas han estado habituados a trabajar siempre con datos, pero terminaban con los datos una vez que se publicaba la historia [121], mientras que ahora, la publicación de la historia es tan solo el principio: los datos se pueden convertir en atractivas visualizaciones o en aplicaciones de noticias que los usuarios pueden usar tiempo después de que se haya publicado [203]. También por ello, el periodismo con datos, que era trabajo para un solo periodista, es ahora una práctica de equipo. El periodista de datos ha de tener competencias informáticas, ser capaz de desarrollar aplicaciones de noticias y visualizar los datos de alguna manera [121], lo cual se traduce en siete perfiles distintos, según maneje una, dos o todas las habilidades. El saber transversal de todos los perfiles es el de conocer lenguajes de la web y de las tecnologías web. Se denomina periodismo guiado por datos o *data driven journalism*, que ha terminado acortándose por periodismo de datos o *data journalism*.

El periodismo de datos ha devuelto al periodismo no solo el interés por las nuevas tecnologías, en el periodismo, sino también por la alfabetización tecnológica y capacitación del periodista en distintas competencias tecnológicas. Los medios de comunicación suelen, bien los más consolidados o bien los más innovadores, abordar procesos de cambio que afectan a su infraestructura tecnológica, y esto suele trasladarse al periodista en competencias, prácticas, formatos y relatos. El periodista con competencias informáticas, del original *Computer-assisted reporter* (literalmente, *periodista asistido por ordenador*), es el periodista que desarrolla competencias en la obtención, análisis y puesta en valor de los datos. Del periodismo de precisión de Meyer se pasa al periodismo asistido por ordenador y de ahí al análisis de datos que sienta las bases del futuro del periodismo [9].

En Europa, *The Guardian* ha sido abanderado en el periodismo de datos. Bajo la dirección de Simon Rogers, ahora en *Google* y entre medias en *Twitter* y conscientes de la necesidad de practicar periodismo de datos [56] en vez de debatir sobre qué es y que no es periodismo de datos [136], dispusieron de una sección denominada *Data Store* o almacén de datos que con el subtítulo *Facts are sacred*, jugaba con la ambivalencia de *facts* para referirse a datos y a hechos, el clásico tótem de la objetividad periodística de *los hechos son sagrados*. En un escenario con tantos datos disponibles, imaginaban el periodismo de datos como el nuevo *punk* [202], en vez de tres acordes para conseguir una canción de menos de tres minutos se disponen cientos de datos, tan solo hay que explorarlos y buscar una buena historia que contar tú mismx, filosofía *do it yourself* o *DIY* en estado puro. Además, desarrollaron su propia *API* que permite acceder a todo su contenido, más de un millón de artículos digitalizados,

Otrxs abogan por hacerlo colectivamente [215], *do it with others* o *DIWO*, equipos multidisciplinares donde participan periodistas, programadores, diseñadores e incluso personas de otras disciplinas como la sociología, la estadística o la psicología, en sintonía con los principios que están detrás de los datos abiertos y de la comunidad que los defiende. Ambas tendencias son plausibles. La tentación de lxs periodistas por mantener en secreto y en solitario sus investigaciones se define como la del *periodista Gollum* [64]

que quiere proteger su anillo, su tesoro, y que nadie más disfrute de él.

Para tratar los datos en brutos se puede dar un tratamiento similar al que necesitan las entrevistas [157], compuestas por palabras, frases, preguntas, respuestas... lxs periodistas lo realizan persiguiendo ciertos objetivos, con un sesgo determinado, con una o varias intenciones. En resumen, al igual que el contenido en bruto de una entrevista está mediado por la persona que entrevista y no solo por la persona entrevistada, los datos no son objetivos porque su recogida, análisis, limpieza, interpretación y disposición no son objetivos. Antes de tomar los datos como tal, lxs periodistas deben investigar qué implicaciones tienen esos datos, por qué se recogieron de esa manera [75]. Se pone en tela de juicio la afirmación de que *los datos cuentan historias*, una metáfora sobre las posibilidades que ofrecen los datos en la actualidad para el periodismo, ya que si se "leen" de determinada manera se podrán contar historias basadas en esos datos, pero son otras personas las que los han producido y lxs periodistas lxs que hacen las interpretaciones, cuando menos.

Una característica del periodismo de datos que lo relaciona directamente con la Web, con la investigación científica y con el software es el acceso al código fuente, a las fuentes de datos empleadas, a la metodología utilizada para el análisis y la visualización. No se produce siempre pero sí en muchas ocasiones, allí donde lxs productoras, lxs periodistas, no esconden la materia prima que lo compone: algoritmos, código, números, conjuntos de datos, herramientas, software, hipertexto, Web.

El término periodista de datos comienza a utilizarse a partir de 2011 [135], aunque actualmente se considera tan amplio que surgen otros [215] que intentan afinar o definir las competencias requeridas o las procedencias y destinos:

- *Data Journalist*, periodista de datos, el término más utilizado en la actualidad.
- *Computer-assisted reporter*, reportero asistido por ordenador, del original esta-

dounidense previo a Philip Meyer pero consolidado con su *precision journalism* o periodismo de precisión.

- *Newsroom developer* o desarrollador de redacción.
- *Journo-geek* o periodista geek, en inglés el juego de palabras funciona mejor.

Y se añaden algunos más [154]:

- *Journo-coder*, una especie de *perio-programador*
- *Programmer-journalist*: programador-periodista.
- *Hacker-journalist*: hacker-periodista
- *Journo-programmer*: *perio-programador*

La confusión terminológica se explica en parte por las distintas acciones que un periodista de datos realiza y que algunas rozan los límites del periodismo de datos para entrar en la programación, la visualización de datos o la estadística. Contar con una infraestructura propia de trabajo y de recopilación de datos [271], en línea con la filosofía *DIY* permite tener una visión panorámica completa del proceso de experimentar con herramientas en las distintas fases del proyecto que invite a la siguiente fase, la participación en proyectos de software. Como fuente de información, el agregador de noticias puede ser la primera herramienta de recopilación de datos [271], una vía para encontrar otros recursos, a través de simples herramientas libres que se ejecutan en el espacio de trabajo y desarrollan el trabajo de forma independiente y única a como lo están desarrollando otros.

#### 4.5.1. Minería de datos

Si el periodismo de datos se basa en la búsqueda de patrones y tendencias en los datos, no parece extraño que se empleen técnicas de minería de datos, disciplina que abarca a la inteligencia artificial, las herramientas de procesado de lenguaje natural, las técnicas de aprendizaje automático, los sistemas de extracción del conocimiento de las bases de datos (*Knowledge Discovery Databases* o *KDD*) la estadística o las matemáticas. La minería de datos se ocupa de la fase analítica de la *KDD*. Para ello se recopilan conjuntos de datos o *datasets*, datos digitalizados, recopilados, ya sean estructurados, como los de una base de datos con tablas de filas y columnas, o sin estructurar. Los datos describen características de objetos, eventos, personas, acontecimientos, recursos, etc. *KDD* supone [104, p. 40-41] el "proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos"(traducción libre), lo cual resume las características del conocimiento extraído:

- Válido, ya que los patrones deben valer para otros datos y no solo para los que se probaron.
- Novedoso, porque aporta algo desconocido.
- Potencialmente útil, ya que debe conducir a acciones que reporten algún beneficio.
- Comprensible, que facilite la interpretación, revisión, validación y toma de decisiones.

Allá donde la minería de datos busca la obtención de conocimiento, el periodismo busca la obtención de historias periodísticas. Las tareas que se realizan en *KDD* son una buena base para el trabajo que se ha de realizar en periodismo de datos:

- Selección, limpieza, transformación y proyección de los datos
- Análisis de datos para extraer patrones y modelos adecuados
- Evaluación e interpretación de los patrones para convertirlos en conocimiento
- Consolidación del conocimiento resolviendo posibles conflictos con el extraído previamente.
- Dejarlo disponible para su uso, por ejemplo, a través de visualizaciones.

A veces los datos interesantes son aquellos que no se ajustan al comportamiento general o *outliers*, otras veces son los redundantes o irrelevantes y otras los que no tienen valores.

Conocimiento es una palabra muy imponente para un árbol de decisión o una colección de reglas, y por usarla no quiere decir que tenga que ver con el conocimiento que llevamos en nuestras cabezas. Tan solo que necesitamos alguna palabra que se refiera estas estructuras que producen los métodos de aprendizaje.

[272, p. 61] (traducción libre)

En el campo de la inteligencia artificial, una parcela se ocupa del aprendizaje automático o *machine learning*, donde se estudian y desarrollan sistemas capaces de tomar decisiones, basados en la experiencia, a partir de la información –denominada .ejemplos- que se aporta. A los ejemplos se les denomina base de datos, corpus o colección. El aprendizaje automático también se ha traducido como sistemas de predicción o predictivos en cuanto que se desarrollan algoritmos y programas capaces de aprender y por tanto de predecir. Está muy extendido su uso en la detección de correo electrónico basura o en la búsqueda de rostros en una fotografía, por ejemplo.

Si en una biblioteca los recursos se identifican por la información del catálogo, una jerarquía de categorías temáticas definidas y limitadas, se puede considerar la Web como lo imaginara Paul Otlet sin definirlo así, un espacio donde buscar contenido relacionado, sin obedecer a un árbol jerárquico, creando una narración determinada. El contenido web cuenta con una carga semántica básica propia del *HTML* (*HyperText Markup Language*, lenguaje de marcado de hipertexto) y una semántica más desarrollada si participa de las tecnologías de la Web Semántica. Al contenido general se le pueden aplicar técnicas de minería de datos para extraer conocimiento de la Web, lo que se conoce como minería web o *web mining*,<sup>el</sup> conjunto de técnicas para la recuperación de información en grandes espacios hipertextuales y desestructurados que se centran más en la información incluida y latente en los hiperenlaces y en la estructura del documento que en lo que actualmente se entiende como contenido textual de las páginas web"[116, p. 12]. Se utiliza para describir el proceso de descubrir de recursos informativos y/o documentales de la Web, analizar los accesos a la Web y otras formas de visualización y acceso de los usuarios a la información web y para "definir el proceso de inferencia, detección y representación de estructuras informativas creadas por las relaciones que forman las propias fuentes web"[116, p. 13]. Se analiza la topología de enlaces mediante técnicas cuantitativas, el contenido del texto o *text mining*, o el número de enlaces por usuario, página, sitio, etc.

En este escenario no existen distancias con fuentes remotas y desaparece la barrera del tiempo. Como contrapartida, las fuentes pueden cambiar la *URI* o desaparecer, aunque también pueden surgir nuevos recursos que analizar. Por tanto, la Web es distinta para cada momento, para cada usuario y permite encontrar información incluso por casualidad [116, p. 41].

Son varios los ejemplos en la historia del periodismo en los que la recopilación, análisis y visualización de datos han estado presentes. Florence Nightingale, enfermera inglesa que participó en la *Guerra de Crimea*, donde estuvieron involucrados Inglaterra y otros países contra Rusia a principios de siglo XIX, se convirtió en una analista de datos de

la época al apuntar en un cuaderno las bajas de heridos y muertos [201]. John Snow, físico inglés que lideró la adopción de la anestesia y está considerado como el padre de la epidemiología moderna, realizó un mapa de los casos de la epidemia de cólera de 1854 en Londres [259] que le llevó a situar el foco de la enfermedad alrededor de las fuentes de agua, y descubrió que éstas eran precisamente las causantes de la enfermedad. Pocos años después y con cierta perspectiva, el francés Charles Minard, ingeniero de caminos, relata en 1869 la campaña rusa de Napoleón de 1812 en un solo gráfico, donde se recogen seis tipos de datos en dos dimensiones: el tamaño del ejército, la distancia recorrida, la temperatura, latitud y longitud, dirección del viaje y localización en fechas concretas [255].

En periodismo se utiliza la minería de datos al menos desde tres enfoques diferentes. En un primer enfoque se aplican las técnicas de minería de datos a la investigación periodística, tanto en los datos que se disponen como extendiendo las técnicas al mundo web. Un segundo enfoque más laxo traduce esas técnicas a ámbitos o tecnologías más usadas por los periodistas, como pueden ser algunas técnicas de recopilación de información de documentos que, sin llegar a tener la complejidad o los pasos del *KDD*, sirve de inspiración y el periodismo se ha permitido la licencia de tomar prestado esos conceptos. Por último, también se emplea también para analizar el propio mensaje periodístico.

#### 4.5.2. API de noticias o contenidos

Otra de las relaciones entre periodismo de datos y Web Semántica se produce a través de *API*. Buena parte de la minería de datos en el periodismo ha venido de la mano de la minería de datos de la Web o *Web Mining*, dado que la Web es uno de los almacenes de datos más grandes y abiertos que existen. Aunque *HTML* ofrece una primera estructuración del contenido, la mayor parte de los datos de contenido se encuentran sin estructura, si bien conviene señalar el esfuerzo colectivo y diverso de datos abiertos y vinculados como parte de la Web Semántica y de determinados repositorios de datos.

Los que son accesibles a través de *API* ofrecen un sistema estándar no solo para la recuperación de datos estructurados sino para su interconexión y generación de aplicaciones web que beban de estos datos. Se trata de un compendio de funciones, procedimientos, métodos o subrutinas intermedias entre un conjunto de datos y quienes quieran acceder a los mismos, ya sean agentes humanos o máquinas. Ofrece un conjunto de datos para ser utilizado por otros, para acceder a esos datos, una capa intermedia que explica cómo están dispuestos esos datos y cómo pueden recuperarse. Cualquier periodista de datos ha de saber trabajar con *API* como antes trabajaba con teletipos, notas de prensa o una biblioteca de fuentes especializadas. Las *API* permiten a los programas y a las personas que usan los programas comunicarse entre sí porque facilita el movimiento de datos de un lado a otro, como tuberías de información.

Medios como *The New York Times*, *The Guardian*, *NPR* o *BBC* han entendido que el futuro del medio de comunicación pasa por implicarse en las nuevas tecnologías y han incorporado equipos de periodistas de datos compuestos por ingenieros, analistas, diseñadores o responsables de producto que entienden de datos. Han sido pioneros del periodismo de datos con las noticias que han creado y con las *API* de acceso a sus datos.

*The Guardian* fue el aventajado europeo. Con Simon Rogers como editor del *Blog de Datos*, en 2010 desarrollaron herramientas que permitían no solo que el trabajo periodístico de *The Guardian* utilice técnicas de minería de datos sino que abrieron su plataforma a los lectores, permitiéndoles incluso su participación. Así, desarrollaron dos proyectos, ahora abandonados o en desuso: el *Índice de datos de gobiernos del mundo* (abandonado, ya no existe), y el repositorio de aplicaciones de terceros.

El *Global Data Index* extraía los datos de los sitios de datos abiertos de distintos gobiernos para transformarlos en conjuntos de datos útiles, donde se permitía la búsqueda y recuperación de información. La participación del arquitecto de información Martin Belam fue fundamental para el diseño de la información que se encontraba por debajo de las historias que se publicaron en el *Data Blog*. Tras consultar los portales de datos abier-

tos de EE.UU. y Australia y del por entonces en desarrollo portal de datos del Reino Unido, determinó qué categorías y terminología podría adaptarse o mapearse al vocabulario del Reino Unido. Además, estudió los registros de las búsquedas del sitio de *The Guardian*, para ver qué términos eran los más buscados desde 2009 y así encontrar patrones de búsqueda en asuntos como crimen o salud. Después de crear algunos prototipos, lanzaron el servicio donde cada país tenía su propia página que recolectaba datos procedentes no solo del *The Guardian* sino también de los sitios de datos de Reino Unido, Estados Unidos, Australia, Nueva Zelanda o Londres, es decir, de aquellos portales de datos abiertos con los que podía enlazar la información. Aunque cada conjunto de datos estuviera en un formato distinto, la *API* los ofrecía en un formulario unificado. Todos los datos disponibles se podían consultar, descargar o puntuar.

La otra aplicación (en desuso, cuya última actualización fue realizada en noviembre de 2010) consistía en un repositorio de aplicaciones que elaboraban lectores del periódico con los datos disponibles. El proyecto que ha permanecido vivo se trata de *Open Platform*, la *Plataforma Abierta*, que no es más ni menos que una *API* de todo el trabajo de *The Guardian*, actualizado en tiempo real y de acceso abierto, que cuenta con más de millón y medio de ítems publicados desde 1999.

En el *The New York Times* (*NYT*) también han desarrollado una *API*. Christopher H. Wiggins, director del equipo de analistas de datos explica las posibilidades que ofrece [185]: puede buscar artículos del periódico que están digitalizados desde 1851 hasta la actualidad, con titulares, resúmenes y enlaces a contenido multimedia asociado; libros, críticas de libros y de las listas de libros más vendidos; campañas electorales a la presidencia de los EE.UU. de Norteamérica, con los datos y gastos relatados por la *Comisión de Elecciones Federales*; datos geográficos, con datos vinculados sobre los conceptos usados en el vocabulario controlado del *NYT*; enlaces y metadatos de los artículos más leídos; crítica de películas, que enlaza con las películas y busca por palabras clave; personas, lugares u organizaciones. Además, se conecta con la *DBPedia* [205] participando de los *Datos Abiertos Vinculados* o *Linked Open Data* (*LOD*). En el mismo sentido funciona

la *API* de la *BBC*, lanzada para el *Mundial de Fútbol de Suráfrica 2010* por ser un acontecimiento donde se podía experimentar", que permitió lanzar contenidos que estaban archivados y no se habrían publicado de otra manera [214], lo cual ejemplifica también la posibilidad de innovación y reutilización en el periodismo.

Periodistas de *American Public Media*, *Public Radio Exchange*, *Public Radio International*, *PBS* (*Public Broadcasting System*, la televisión pública de EE.UU. de Norteamérica) y *NPR* decidieron crear un canal para compartir y distribuir contenido a través de cualquier plataforma [214], una *API* que permitiese a medios públicos de todo EE.UU. compartir contenidos con otros miembros de sus medios, ya fueran desarrolladores, creadores independientes o editores. Esto permitiría, por ejemplo, agregar y producir reportajes periodísticos amplios en acontecimientos de gran envergadura, como el desastre del vertido de *BP* en el Golfo de México, el terremoto en Haití o el huracán *Katrina*, pero también intercambiar cualquier tipo de noticia o formato y su reutilización en la otra punta del país. La *API* enlaza con más de 250.000 historias desde 1995.

Cuando *NPR* lanza el proyecto *Argo* con la ayuda de la *Knight Foundation* [148], una aventura periodística que pretende cubrir noticias locales con mayor profundidad y con nuevos formatos y temáticas, a través de sitios creados con *WordPress* en la que participan doce estaciones miembros de *NPR*, comparten contenidos a través de la *API*.

En la edición de 2009 de la conferencia *South by Southwest Interactive (SXSW)* hubo un panel titulado *Get me rewrite! Developing APIs and the Changing Face of News*, "¡déjame utilizar lo que escribes! desarrollo de *API* y cambio de rostro en las noticias", que contó con periodistas de *NPR*, *The Guardian*, *The New York Times* y *Daylife*, donde discutieron sus experiencias en el lanzamiento de *API* para publicar sus contenidos y se convirtió en un alegato a la reutilización de los contenidos periodísticos [211], ya que permite lanzar nuevos productos y secciones más rápido, se alimenta el sitio constantemente, se pueden crear plantillas de secciones que se alimenten de contenidos distintos, suministra contenido a todos los medios, secciones o formatos que tenga un medio o un

grupo mediático: sistemas de paginación, sistemas de gestión de contenido, aplicaciones móviles, diseños orientados, RSS, plataformas específicas o cualquier otra. Además, deja de ser el medio el único capaz de hacerlo ya que permite a otrxs, incluidos lxs lectores, utilizarlo de las maneras que decidan lxs creadorxs de la *API*.

Los ejemplos en España están muy lejos de estos usos pero cabe destacar el camino iniciado por *El País* en 2010 [198], con un gestor de contenidos que apuesta por la semántica, con un *colabulario* o vocabulario de etiquetas o entidades en el gestor de contenidos interno que tiene su reflejo en la categorización de sus contenidos y un índice de temas y autores público [199], además de algunas *API* de acceso a datos, como por ejemplo los resultados de la lotería nacional o de las distintas elecciones de España.

## 4.6. Características de los nuevos medios

Radio, cine, televisión fueron nuevos medios al nacer, contemporáneos de la modernidad. La Web y en general las nuevas tecnologías digitales han reformulado los nuevos medios, dado que la Web es un medio y también un metamedio, y el cine, la radio, la televisión y la prensa encuentran nuevas formas de expresión y difusión. Al ser digitales, los nuevos medios guardan algunas características comunes: representación numérica, modularidad, automatización, variabilidad y transcodificación [155, p. 72] .

Con la representación numérica, la digitalización, los medios se vuelven programables y se reproduce la lógica del modelo de producción industrial donde se organiza el trabajo procesos productivos separados y estandarizados, actividades simples, repetitivas y en secuencia que son capaces de realizarse por parte de obreros especializados en esa tarea concreta, sin tener que dominar el proceso completo y que se podían reemplazar con facilidad [155, p. 75]. En este universo de datos digitales, aparecen siete conceptos de diseño para la navegación en un espacio de datos: enlace, búsqueda, secuenciación,

jerarquía, similitud, cartografía, guías y agentes

El proceso de estandarización se da en los distintos niveles de las tecnologías, desde las dimensiones de la imagen (tamaño, relación de aspecto, contraste) o la frecuencia de muestreo, a los protocolos de comunicación en Internet, la codificación del texto o los lenguajes de marcado. Con la Web Semántica se pretende el acceso global al conocimiento humano [159, p. 189], lo cual requiere un establecimiento de guías, normas y procedimientos estandarizados para hacer los sistemas compatibles y facilitar la difusión y el intercambio automatizado de datos e informaciones a través de las redes de comunicaciones. Los estándares han sido impulsados por organizaciones nacionales e internacionales de normalización, como *ANSI* o *ISO*, pero también por entidades privadas sin ánimo de lucro como el *Consortio W<sub>3</sub>C*, *RFC* (*Request for Comments*, solicitud de comentarios) o *IETF* (*Internet Engineering Task Force*, fuerza de trabajo de la ingeniería de Internet).

*W<sub>3</sub>C* cuenta con 4 tipos de documentos que indican su estado:

1. Borrador de trabajo, *WD* o *Working Draft*.
2. Recomendación candidata, *CR* o *Candidate Recommendation*.
3. Propuesta de recomendación, *PR* o *Proposal of Recommendation*
4. Recomendación, *REC* o *W<sub>3</sub>C Recommendation*.

Por su parte, *IETF* establece distintos tipos de estadios de sus normas:

1. *Internet Draft*, borrador de Internet.
2. *Request for Comment*, *RFC*, solicitud de comentarios.

3. *Proposed Standard*, estándar propuesto
4. *IETF Standard*, estándar *IETF*.

También hay entidades orientadas a la creación de estándares para el mercado, como por ejemplo *OASIS*, *Organization for the Advancement of Structured Information Standards*, organización para el avance de los estándares de información estructurada, que diseña y desarrolla estándares industriales para la interoperabilidad basada en *XML*. Sus normas siguen dos pasos:

1. *Oasis TR*, *Technical Report*, informe técnico.
2. *OASIS Standard*, estándar.

Para que los metadatos mantengan un grado aceptable de eficacia y aplicación es necesario que cuenten con entornos normalizados [159, p. 210], ya sean estándares formales *ISO*, especificaciones de acceso público *W3C* o estándares de facto, tecnologías que sin seguir un proceso de normalización se convierten en las más utilizadas en su campo.

El lenguaje de los nuevos medios también responde a la lógica de la distribución postindustrial, a la producción a petición de los usuarios, bajo demanda, que solo son posibles por los procesos automatizados en todos los niveles. Dada la estructura modular de los nuevos medios [155, p. 75-76], se representan como colecciones de muestras discretas, para lo cual existe una automatización de alto nivel. El ordenador entiende hasta cierto punto los significados que incluyen los objetos que se generan. No solo se crean sino que se recuperan y reutilizan los objetos mediáticos a partir de la información contenida en bases de datos, que puede ser actualizada periódicamente, lo cual afecta de nuevo a la presentación de los contenidos.

Se reproduce el principio de la cibernética, donde se escucha a los usuarios como pro-

ductorxs de información para adaptar los contenidos, y se modifica la interfaz para que su experiencia de usuarix sea satisfactoria, donde se valora el individualismo por encima del conformismo [155, p. 88]

Pero sin duda el concepto que se considera más importante pasa por la transcodificación (cultural) [155, p. 341], de qué manera el funcionamiento de la tecnología influye en nuestra lógica cultural de los medios (organización, géneros y contenidos), en la mezcla de significados que propone el ordenador como dispositivo de salida de la Web, y en la base de datos como base para la narración. Al igual que el *flâneur* de Benjamin transformaba el espacio de la ciudad, la Web es un espacio subjetivo y su arquitectura responde a la navegación que realiza y su emoción [?]. Curiosamente, son metáforas espaciales las que aparecen en los nombres de los ordenadores más populares de la Web i.o: *Microsoft Internet Explorer* y *Netscape Navigator*, sin olvidarnos de otros como *Konqueror* o *Epiphany*.

La Web también ha favorecido nuevas narrativas como la transmedia, concepto lanzado por Henry Jenkins [190] para explicar la narrativa donde distintos elementos integran una ficción transmitida por varios canales que pretenden crear una experiencia unificada y coordinada. La Web es un canal multicanal, un metacanal, un metamedio, un medio multimedia que ha servido de campo de experimentación para lo transmedia, de tal forma que incluso el periodismo se ha visto influenciado, habiéndose beneficiado también de la facilidad para la producción multimedia y del público consumidor de narrativa transmediática de información. En la narrativa transmedia hay una estructura narrativa que se puede dividir en fragmentos [190], que puede utilizar distintos medios, que ofrece contenido expandido, circula por las redes sociales, a menudo en un proceso viral, y se puede producir desde dispositivos móviles.

## 4.7. Leyes

Desde que Internet y sobre todo la Web comenzó a ser un lugar donde hacer negocio, han sido muchos los intentos de sectores empresariales o gobiernos por regular un espacio sin aparente regulación que han conseguido normalmente restringir derechos y libertades fundamentales como la libertad de expresión o la libertad de información.

Los movimientos empresariales –normalmente expresados en grupos de presión– encuentran el respaldo de los gobiernos, quienes además de favorecer los intereses de los sectores económicos anteriores, ven una oportunidad de regulación de los contenidos en un campo que tenía las puertas abiertas a la libertad de expresión.

En España el primero de los intentos de regulación se conoció como *LSSI, Ley de Servicios de la Sociedad de la Información* o *Ley 34/2002*, y fue la traslación a la legislación española de la directiva europea sobre comercio electrónico. Coincidió con el fin de la crisis de *las puntocom*, la burbuja de los valores de sociedades de la *Web 1.0*. Sin embargo, lejos de regular el comercio electrónico, regulaba cualquier contenido transmitido a través de Internet, y por tanto de la Web [153], por lo que más que pretender su aplicación a las iniciativas comerciales de Internet, regulaba cualquier tipo de actividad que se realice a través de la Web ya que el concepto material en torno al que gira su ámbito de aplicación se denomina *servicios de la sociedad de la información*. Cualquier persona que disponga de una web se convierte en responsable de los contenidos que aparecen en su espacio.

Otro movimiento se produjo en 2008 con el nacimiento de la *Coalición de Creadores e Industrias de Contenidos*, formada por *EGEDA, SGAE, Promusicae, FAP, Adican* y *Adivan*. Días después se celebra en Madrid el *Seminario internacional El futuro de las industrias culturales y la lucha contra la piratería en internet*, donde participan el presidente de *IFPI (International Federation of Phonographic Industry, federación internacional de la industria fonográfica)*, John Kennedy y la responsable de piratería en

Internet de la *Oficina de Propiedad Intelectual* del Reino Unido, Liz Coleman, además de la ministra de cultura y los miembros de la coalición.

Mientras, en EE.UU. la regulación se llamaba *ACTA, Anti Counterfeiting Trade Agreement* o acuerdo comercial contra la falsificación, creado para luchar contra la piratería comercial que promueve un marco legal donde se pueden ver afectados derechos fundamentales, ya que entre las medidas que proponía destacaban la de solicitar a los proveedores de servicios de Internet que supervisen todas las comunicaciones de sus usuarios y obstaculicen el uso de materiales con copyright, incluidas las redes *P2P*, donde además se convertía a los prestadores de servicio en ejecutores de procesos administrativos sin intervención judicial previa. Estos principios básicos parecen una copia calcada de la *DMCA, Digital Millenium Copyright Act*, o ley de derechos digitales del milenio, con la que Bill Clinton blindaba los derechos de autor en Internet en 1998.

La *Ley de Economía Sostenible* de 2011, conocida popularmente como *Ley Sinde-Wert* por ser auspiciada bajo el mandato de la ministra de cultura Ángeles Sinde y rematada por el sucesor Ignacio Wert, contemplaba en su disposición final 43ª la regulación de las descargas de Internet, instrumento legal para perseguir las descargas de contenido protegido por copyright restrictivo, traslación de la directiva europea del acceso a Internet, que elimina el control judicial previo al corte de la conexión para aquellas personas que realizan descargas de archivos o la gestión del tráfico en función de las necesidades de los propietarios de las infraestructuras. Parece una copia del párrafo anterior, pero también regulaba, entre otras cosas, la solicitud previa para instalar *cookies* en los navegadores al visitar páginas web, lo que permite que la navegación pueda rastrearse. En el ámbito técnico, también superaba la consideración de *stateless* o sin estado de *HTTP (HyperText Transmission Protocol)*, protocolo de transmisión de hipertexto).

---

## Capítulo 5

### Web

La Web es el nombre por el que conocemos a la *World Wide Web* o gran telaraña mundial, un conjunto de tecnologías que sirven para que accedamos a contenido multimedia, hipermedia y sobre todo, hipertexto, escrito fundamentalmente en *HTML*, pero el contenido no tiene por qué venir de documentos *HTML*, pueden ser datos en formato *XML* (*eXtensible Markup Language*, lenguaje de marcas extensible), *JSON* (*JavaScript Object Notation*, notación de objetos *JS*), *CSV* (*Comma Separated Values*, valores separados por comas), o *RDF* (*Resource Description Framework*, marco de descripción de recursos), entre otros.

Una página web se trata de un texto maquetado/formateado en el lenguaje de marcas de hipertexto *HTML* que está alojado en un servidor web, se transmite por el protocolo de transmisión de hipertexto *HTTP*, viaja por una red *TCP/IP*, cuanta con una *URL* que lo identifica, un servidor *DNS* lo asocia con una dirección *IP* y finalmente se muestra por un navegador/renderizador en un dispositivo de salida, la pantalla del ordenador personal, del portátil o del móvil. Todos estos procesos suceden de forma transparente para los usuarios, pero se trata de un conjunto de tecnologías que traducimos como *ver*

*una página web*. Esta es la potencia de la Web, disponer de contenido que proviene de múltiples fuentes en un formato de salida estándar, el documento *HTML*, la Web. Si la Web nos permite acceder a información, la Web Semántica pretende dar significado a los contenidos de la Web, de forma que podamos recuperarla, convertirla, utilizarla o reutilizarla mejor.

La Web también solemos asociarla a cosas distintas, metáforas de sus posibilidades:

- Medio de comunicación de masas, como la televisión o la radio.
- Medio de información, como un libro o una enciclopedia.
- Medio de comunicación personal, como un teléfono.
- Dispositivo de acceso a la información, como un ordenador o una exposición.
- Sinónimo de Internet, aunque la Web use Internet como soporte, para muchas personas son sinónimos.
- Como buscador de información, para cualquier tipo de consulta.

## 5.1. Orígenes de la Web

Tim Berners-Lee, científico británico que desarrollaba su trabajo en el *CERN* (*Conseil Européen pour la Recherche Nucléaire*, organización europea para la investigación nuclear) de 1989 a 1994, se sorprendió de que la producción científica de lxs distintos investigadorxs, más de cinco mil personas, no pudiera consultarse de una forma estándar, accesible y multiplataforma [34, p. 9], por lo que se dedicó a desarrollar un método para intercambiar datos, un prototipo de sistema de información distribuida que presentó

en marzo de 1999.

Berners-Lee utiliza tecnologías o conceptos que ya existían, como el hipertexto, el protocolo de comunicaciones de Internet o los lenguajes de marcas, con otras nuevas y todas ellas re combinadas y sintetizadas en lo que conocemos como la Web [34]. Berners-Lee comienza a trabajar con código para el procesador de hipertexto en 1990 [21], tras lo que pudo mostrar en su ordenador *NeXT* la primera página web que informaba sobre el proyecto *WWW* a través de un programa de visualización, *browser* o navegador, creado para tal efecto. Se trataba de un documento de texto escrito en el nuevo lenguaje *HTML* alojado en un servidor de páginas web creado igualmente a propósito, que era el propio ordenador, que disponía de una dirección única *URL* (*Uniform Resource Locator*, localizador uniforme de recursos), transmitida por *HTTP* [183].

[http://timeline.web.cern.ch/sites/timeline.web.cern.ch/files/info.web\\_.jpg](http://timeline.web.cern.ch/sites/timeline.web.cern.ch/files/info.web_.jpg)

En 1991 el *CERN* incorpora a la Web a la comunidad de físicos que están investigando en la institución, a quienes ofrecen un navegador, el software del servidor web y una librería (software) que les permita crear su propio software. Poco después, a mediados de 1991, Berners-Lee avisa de la existencia de la Web a la comunidad de usuarios y entusiastas del hipertexto que se reunían en el grupo de noticias *alt.hypertext*. Este fue el primer aviso público de la Web como uno de los servicios de Internet. En 1993 el *CERN* decide liberar el código de la Web con licencia de *Dominio Público*, para asegurar su pervivencia como estándar abierto. A finales de año ya eran más de 500 servidores web y 10.000 a finales de 1994. Por entonces se había creado el consorcio *World Wide Web* en el laboratorio de ciencia informática del *MIT* (*Massachusetts Institute of Technology*, instituto de tecnología de Massachusetts, EE.UU. de Norteamérica) en colaboración con el *CERN* y el apoyo de *DARPA* (*Defense Advanced Research Projects Agency*, agencia de proyectos de investigación avanzados de defensa de EE.UU. de Norteamérica) y la *Comisión Europea*.

El documento que describe la *World Wide Web*, el hipertexto y su aplicabilidad en el entorno del *CERN* [43] sienta las bases de la primera Web y apunta algunos problemas de la Web actual:

- Pretende realizar un programa que provea acceso al hipertexto o navegador. Cuando se abre, aparecerá la página de inicio por defecto, la que se configure o cualquier otra configuración predeterminada o configurada por lxs usuarixs.
- Los textos están enlazados de tal forma que lxs usuarixs pueden ir de un concepto a otro, por enlaces, y ampliar la información.
- Esta red de enlaces es lo que da el nombre a la Web o telaraña.
- La Web no necesita jerarquía, no requiere escalar un índice para encontrar la información deseada, no cumple una tipología de red arbórea, sino que permite navegar por la estructura de enlaces, por sus nodos, siempre que tengan una *URI*.
- La Web no es un objeto u obra completa. Por un lado, cualquiera puede detentar la autoría de contenidos en la Web, por lo que se puede encontrar más o menos información de determinadxs autorxs. Por otro, puede haber algunos enlaces en el texto pero podría haber más, menos u otros. Además, el camino que se produce al navegar por el hipertexto puede variar según cada usuarix o cada momento.
- Al concebirse como red, a cada documento *HTML* se le denomina *nodo* y al proceso de pasar de un nodo a otro, navegar. Los nodos no necesitan estar en el mismo ordenador o en el mismo servidor web.
- La Web resuelve el problema de acceso a contenidos en redes que funcionan con distintos protocolos y documentos escritos en distintos formatos. Ofrece un formato estándar, un formato de intercambio de documentos.

- Los nodos o documentos *HTML* pueden contener información no textual como diagramas, imágenes, sonidos o animaciones. En este sentido, en la *Web 1.0* el término hipermedia es la extensión de la idea del hipertexto a estos otros tipos de medios.
- Aunque se permita el intercambio de archivos, la Web se concentra en la lectura universal del texto.

Entre los primeros documentos del *W3C* que describen la Web se encuentra una especie de guía de estilo que explica cómo debería ser la estructura del documento para favorecer su lectura [30], tanto a lectorxs que son la audiencia potencial como a otrxs que les pueda interesar. Se recomiendan estructuras arbóreas [29] con un tamaño determinado [26] que no se extiendan más de lo imprescindible para permitir su lectura, recuperación y entendimiento, acompañado de citas [31] si fuera necesario.

El primer sitio de la Web [23] la define como resultado de un conjunto de tecnologías, una iniciativa de recuperación de la información hipermedia de área amplia que pretende dar acceso universal a una gran cantidad de documentos.

The WorldWideWeb (W<sub>3</sub>) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

[23]

Entre las tecnologías o saberes se encuentran:

- Las redes, con las palabras *wide* y *web*, dos tipos de red, insertas en el propio nombre, además de *wide area*, para referirse al alcance de la red.
- Organización y recuperación de la información.

- Hipermedia.
- Acceso universal al conocimiento.
- Biblioteconomía y documentación.

En un principio, la Web era tan pequeña que se permitía un listado con todos los servidores que la componían [32], con toda la información clasificada por temas [28], por tipo de servicio [25] y por organización participante [24].

Por hipermedia se refiere a un enlace a otro documento [22], mientras que hipertexto consiste en el texto que contiene enlaces a otros textos, el texto que no está obligado a ser lineal. Aunque el término había sido acuñado por Ted Nelson en 1965 [65], cuenta con una historia previa y posterior independiente de Nelson muy ligada a la Web [79].

En el listado de términos que se incluye como glosario del proyecto WWW, hay varios que provienen de la cultura del hipertexto [27]:

- *Anchor*, ancla, un área de contenido en el interior de un nodo que es el origen o el destino de un enlace. Puede ser todo el contenido del nodo o una parte. Resaltan sobre el resto del texto de alguna manera y pueden pincharse para seguirlos.
- *Back link*, enlace de vuelta, un enlace en una dirección que implica que haya un enlace de ida.
- *Browser*, navegador, un programa que permite a una persona leer hipertexto, debe ofrecer formas para ver los contenidos de los nodos y para navegar de uno a otro.
- *Button*, botón, un *anchor* que es el origen de un enlace.

- *Card*, tarjeta, término alternativo a nodo donde su tamaño se limita a una página simple de tamaño limitado.
- *External*, externo, un enlace a un *nodo* en otra web.
- *Hypermedia*, hipermedia, se refiere al hipertexto multimedia, que incluya también sonido, gráficos o vídeo.
- *Hypertext*, hipertexto, texto que no está limitado a ser lineal en su lectura.
- *Internal*, interno, un enlace a una parte del documento.

Otros provienen de la cultura de la documentación:

- *Annotation*, anotación, el enlace de un comentario de un nodo a un nodo existente, pensado para que los usuarios de la Web interactúen con el contenido y lo mejoren.
- *Authoring*, autoría, para el proceso de escritura del documento más que para el texto en sí.
- *Database*, base de datos, una colección de nodos, contenidos informativos.
- *Document*, documento, un término que se utilizaba en Intermedia para un nodo y también como una colección de enlaces sobre asuntos relacionados, es el término preferido en  $W_3$ .
- *Node*, nodo, una unidad de información. Ha dejado de utilizarse como tal aunque en la comunidad de hipertexto se refería a partes de información como podría encontrarse en una *hypercard* o una *notecard*, sistemas hipertextuales previos a la Web.

- *Reader*, lector, la persona que navega por la información.
- *Versioning*, versiones, la gestión y almacenamiento de versiones previas de información por seguridad, diagnóstico o interés. Considerado muy útil cuando muchos usuarios pueden editar el mismo material. Abre la puerta a los sistemas de gestión de versiones.

Y otros de la cultura de red:

- *Client*, cliente, un programa que solicita servicios de otro programa, el servidor. En la Web, el navegador es un cliente de un servidor –de datos– web.
- *Daemon*, demonio, un programa que *corre* –que actúa, que se ejecuta– de forma independiente para realizar tareas como, por ejemplo, construir índices, realizar copias de seguridad o solicitar determinada información. En *Unix*, el demonio se utiliza también como sinónimo de servidor en cuanto que funciona de forma independiente.
- *Host*, un ordenador de la red que aloja servicios. Se utiliza este término en vez de *nodo* para diferenciarlo de los documentos de la primera Web, que también se denominaron nodos. Con el paso del tiempo, este significado ha perdido peso y se ha mantenido el de nodo de la red.
- *Protection*, protección, sobre los permisos para acceder –leer– o escribir ciertos datos.
- *Path*, ruta, un conjunto ordenados de nodos o anclas que una web puede leer.
- *Server*, servidor, un programa que ofrece un servicio a otro, conocido como cliente.

- *Tracing*, rastro, se refiere a buscar nodos para la navegación automática.
- *Topology*, topología, la conectividad permitida entre nodos, anclas y enlaces: 1 a 1, 1 a mucho, etc.
- *Web*, un conjunto de nodos interconectados por enlaces.

## 5.2. Hipertexto

El precedente más conocido del hipertexto se halla en el concepto de *Memex* [63], un dispositivo físico ideado por Vannebar Bush en 1945 en el que se almacenarían todo tipo de documentos que constaría de una mesa con un teclado y palancas que permitirían la consulta de datos almacenados en *microfilms* proyectados en unas pantallas translúcidas. El aparato incluiría también una opción para que el usuario pudiera tomar anotaciones en los márgenes, de manera que los usuarios se convirtieran, a su vez, en coautores del documento.

El significado de *Memex* no está del todo claro. Mientras la Wikipedia recoge que el nombre es una construcción de *memory index* (*índice de memoria*) [262], Andy Carvin apunta *memory extender* (extensor de memoria) [69]. Bush creía que la gran cantidad de trabajos de investigación que se disponían en esa época de mediados de siglo XX había crecido más allá de la capacidad de aprovechamiento de los humanos. Los métodos para almacenar, modificar y categorizar la información, de estructura arbórea, donde las búsquedas han de volver siempre al punto origen para seguir descendiendo por las ramas y no se permite la interconexión de contenidos relacionados, no se corresponde con el modo de hacer de la mente humana que funciona por asociación y permite enlaces entre contenidos relacionados. Por tanto, frente a la indexación, un sistema de recuperación de información debería ser capaz de conectar temas relacionados de forma ágil, realizar

anotaciones y crear nexos.

De esta manera, Bush redefine la forma de leer y anticipa la Web, conecta la lectura con la escritura, la autoría con la obra colectiva y los enlaces de hipertexto como permiten ahora la Web para realizar trayectos infinitos.

Al elaborar un trayecto, el usuario primero le da un nombre, luego introduce dicho nombre en su libro de códigos y lo teclea en el teclado. Delante de él están los dos artículos que han de unirse proyectados en dos superficies de visionado adyacente. Debajo de ellos, hay unos espacios para códigos en blanco y un puntero para designarlos. El usuario sólo tiene que tocar una tecla y los dos artículos se encuentran unidos. En cada espacio para códigos consta el código pertinente del código asociado. También en el espacio para códigos, pero sin que se vea, hay una serie de puntos que serán leídos por una célula fotoeléctrica; estos indican, con su posición relativa, el número de índice del otro artículo. Más adelante, cada vez que se visiona uno de los artículos, el otro podrá ser recuperado con simplemente apretar un botón situado debajo del correspondiente código.

[63]

Bush reconoce que estos trayectos también conforman una nueva forma de textualidad, de narrativa, de escritura, ya que al unir artículos de fuentes distantes se realiza un nuevo texto que contiene a ambos, donde un artículo, página, bloque de texto, imagen u otra información puede estar en uno o muchos textos a la vez, incluidas "formas enteramente nuevas de enciclopedias".

Esta idea de Bush es la que reivindica Nelson como *transclusión* [173] ya que mientras un enlace o *link* crea nexos entre textos que podían ser distintos, la *transclusion* enlaza textos que son iguales, es decir, incluye una copia del documento en otro o de parte del mismo

en otro documento [175], como, por ejemplo, un bloque de cita o una fotografía, lo que se denomina *transdelivery* (entrega), *explicit quotation* (cita explícita) o *transquotation* (transcitación).

Hipertexto y también hipermedia fueron acuñados por Nelson cuando bosquejó el proyecto *Xanadu* en la década de los 1960' [175]. Nelson. Tras visitar el centro de investigación de *Xerox* en Palo Alto (California, EE.UU.), propuso la idea de crear una biblioteca en línea con todos los *documentos* existentes [173]. La idea de Nelson difiere con la de la Web en varias cuestiones, además de la *transclusión*, porque Nelson no pretendía el acceso a toda la información sino a documentos. Su preocupación por la gestión de los derechos de autor se tradujo en el *transcopyright* un sistema de gestión y cobro de derechos de autor que funcionara automáticamente cuando alguien utilizaba una obra ajena, la citaba o la incorporaba a una suya. El sistema se encargaría de rastrear la reutilización, cobrar por ella y hacer llegar la cantidad devengada. Eso no impedía que cualquiera incluyese documentos de libre uso, con licencias libres.

En 2014 Nelson lanzó una versión de la transclusión tal como lo había definido 50 años antes [174]. Independientemente de las polémicas en las que se ha visto implicado a Nelson [173], lo cierto es que en la actualidad, a partir de la *Web 2.0* o *Web 3.0*, algo parecido a la transclusión –con permiso de Nelson– se produce en al menos tres prácticas de la Web. En primer lugar, con bastante profusión, se embeben bloques de información de todo tipo, gráficos, mapas incluso documentos completos, una hoja de cálculo o cualquier otro documento *HTML* en otros documentos *HTML*, de tal forma que el contenido no hay que almacenarlo en dos espacios distintos sino solo en el original, a través del elemento de *HTML iframe*. También ha popularizado el término tal cual el entorno de aplicaciones *JavaScript* liderado por *Google AngularJS*, que lo utiliza para anidar operaciones. Por último, en el mundo de los datos abiertos, distintos proyectos pueden utilizar la misma fuente de datos, en otra forma de *transclusión*.

Nelson también se pregunta por otras cuestiones interesantes compartidas con la Web,

como el problema de generar nombres únicos para que los documentos puedan ser encontrados fácilmente [175], incidió que los desarrollos de las interfaces de usuario *GUI* (*Graphical User Interface*, interfaz gráfica de usuario) habían de ser simples, y, si Berners-Lee se refiere a un *GGG* (*Giant Global Graph*, gráfico global gigante) [36], *Xanadu* concebía un documento global y único, el *docuverse*, que cubre todo lo escrito en el mundo mediante una gran cantidad de ordenadores interconectados, información relacionada mediante hipertexto –enlaces hipertextuales–, todos disponibles.

Si bien el hipertexto permite muchas lecturas posibles, una parte muy importante de su uso corresponde a la modificación de la interfaz entre usuario y el texto [79], por lo que se concibe el hipertexto como un medio para la información que se apoya en un soporte informático en el que muchos documentos interrelacionados se muestran con sus enlaces en una pantalla de alta resolución.

En el siglo XIX, cincuenta años antes de que Bush proyectara *Memex*, Paul Otlet y Henri La Fontaine diseñan en Bélgica el *Repertoire Bibliographique Universel* o repertorio bibliográfico universal, un intento de desarrollar una bibliografía del conocimiento acumulado en el mundo, para lo que necesitaban un software como sistema de clasificación y un hardware que lo llevara a cabo. Tomando como punto de partida el sistema decimal Dewey y observando las limitaciones de estos sistemas para guiar a los usuarios al libro de búsqueda, Otlet desarrolló la clasificación decimal universal *CDU*, la primera implementación de un sistema de clasificación facetado que contempla:

- Hechos, observaciones empíricas y aseveraciones.
- Interpretación, análisis o conclusiones, derivadas de los hechos.
- Estadísticas, datos medibles, cuantificables.
- Fuentes (*sources*), citas o referencias.

Comprende más de 62.000 clasificaciones individuales, aunque cuenta con diez clases de alto nivel:

- 0, generalidades. Ciencia, conocimiento, organización, ciencias de la computación.
- 1, Filosofía. Psicología
- 2, Religión. Teología
- 3, Ciencias Sociales. Derecho
- 4, en desarrollo
- 5, Matemáticas y ciencias naturales.
- 6, Ciencias aplicadas
- 7, Artes. Recreo. Entretenimiento. Deporte
- 8, Lenguaje. Lingüística. Literatura
- 9, Geografía. Biografía. Historia.

Además de las tablas principales de materias, soportaba tablas auxiliares para añadir facetas que ofrecían notaciones para lugar, idioma, características físicas y relaciones entre temas a través de los conectores +, / y :.

El sistema de clasificación buscaba servir la información, por lo que en 1910 diseñó el *Mundaneum*, sede del archivo, que permitía responder más de mil quinientas peticiones por año por correo postal [189]. Pero no solo se quedó en las tecnologías asentadas sino que bosquejó el uso de reconocimiento de voz, redes inalámbricas para subir docu-

mentos y comunidades virtuales que participaran del proyecto y una red de insituciones mundiales interconectadas que permitiera el libre flujo de información que permitiera llevar transformaciones sociales, culturales y políticas.

También Doug Engelbart planteó un sistema abierto de *hiperdocumentos*, el *OHS* (*Open Hiperdocument System*, sistema abierto de hiperdocumentos) [85] que incidía más en facilitar el trabajo en grupo y la inteligencia colectiva que en el propio acceso a los documentos.

En los inicios de la informática de consumo, en los años 1980', *Apple* contaba con el departamento llamado *Navigation Information* (navegación de la información) donde trabajaban más de cien personas en los conceptos de navegación por la información y en aplicaciones basadas en la navegación, precursores de los actuales *sitios web*. En 1987 lanzaron la aplicación *Hypercard*, incluida en todos los *Macintosh*, que permitía navegación y enlaces. Estaba basada en el concepto de una pila de tarjetas virtuales que contienen datos, texto e imágenes, al estilo de un *rolodex* (sistema de almacenamiento de tarjetas de visita). Contaba con un formulario para la presentación de datos similar a los de otros entornos de desarrollo rápido de aplicaciones (*RAD* o *Rapid Application Development*) y se conectaba con sistemas complejos de bases de datos, se desarrollaba como entorno para juegos estilo *elige tu propia aventura*, etc.

La influencia de *HyperCard* llegó a la prehistoria de los *Wikis* [111] ya que su inventor, Ward Cunningham, se basó en *hypercard* para desarrollar un navegador que visualizara tarjetas de ideas, tarjetas de personas que sostienen ideas y tarjetas de proyectos donde las personas comparten ideas. También al primer navegador gráfico, en blanco y negro, *Viola* (luego *ViolaWWW*) se basaba en las ideas de *Hypercard* [269]

En la actualidad, con la evolución de la Web y de las *API* de acceso a datos, se piensa en el hipertexto como independiente del navegador y del documento web [107], dado que otras aplicaciones, y no solo el navegador, pueden seguir enlaces si entienden los

formatos de datos y los tipos de relaciones. El hipertexto se convierte, por tanto, en el formato de presentación de la información e interfaz de usuario regulada mediante acciones que agentes humanos o máquinas pueden manipular para elegir y seleccionar acciones.

### 5.3. Principios de la Web

En 1998 Berners-Lee recopila los principios de buen diseño de la Web [67]:

- Simplicidad, que se expresa en el acrónimo *KISS* (beso) o *keep it simple, stupid*, *mántelo simple, idiota*.
- Diseño modular, ya que es bueno tanto para encontrar la pieza que falla como para cambiar algo que podría funcionar mejor y permite que más personas trabajen en cosas distintas a la vez. Esta metodología conviene mantenerla hacia dentro del proyecto y también hacia fuera, como si el proyecto en su conjunto pudiera ser un módulo de otro proyecto.
- Tolerancia, expresada en el axioma *ser liberal con lo que necesitas pero conservador con lo que haces*. Cuando los navegadores son laxos con los fallos de escritura de *HTML*, también se corre el peligro de hacer perezosos a quienes cometen el error. En cualquier caso, no impide que haya estándares claramente definidos y que si el error se produce, no cumpla el estándar.
- Descentralización, principio de diseño de los sistemas distribuidos que se puede aplicar también a la construcción de vocabularios que definan cosas. Los vocabularios serán útiles mientras no bloqueen otros significados en otros dominios, de lo contrario se pueden convertir en cuellos de botella de una red, bloquearía el

desarrollo de conceptos en otros dominios.

- Invención independiente, relacionado con la modularidad. El sistema debe hacer una cosa bien y dejar que otros módulos hagan otras cosas, lo que permite la descentralización y construir un sistema más grande y robusto.
- Principio de mínimo poder, aplicado a los lenguajes informáticos. Frente a los lenguajes potentes, los lenguajes simples permiten hacer mucho con los datos almacenados. La Web Semántica es un intento de mapear grandes cantidades de datos existentes en un lenguaje común para que pueda analizarse, descargarse, visualizarse y combinarse con otras fuentes de datos.

El *W<sub>3</sub>C* define en 2003 la Web en 7 puntos que explican sus objetivos y principios [234]:

1. Acceso universal. El *W<sub>3</sub>C* define la Web como el universo de información accesible en la red, disponible desde cualquier dispositivo, que beneficia a la sociedad al permitir nuevas formas de comunicación humana y oportunidades para compartir el conocimiento. La universalidad se practica si se puede acceder desde cualquier hardware, software, infraestructura de red, idioma, cultura, localización geográfica o habilidad mental o física. Hay grupos de trabajo en el consorcio de la Web, el *W<sub>3</sub>C*, encargados de esta misión.
2. Web Semántica. Cuando se comparte el conocimiento en la Web, se puede aportar semántica de cara a que los ordenadores pueden interpretarlo e intercambiarlo, recuperarlo y accederlo, a través de lenguajes de la Web Semántica.
3. Confianza. La Web se pensó no solo como un medio de lectura sino también de escritura, de interacción con el texto. Para favorecer la interacción la Web debe generar confianza y aportar confidencialidad, lo que permite a los usuarios ser responsables del contenido que publican.

4. Interoperabilidad. Afecta tanto al software como al hardware. *W<sub>3</sub>C* promueve la interoperabilidad diseñando lenguajes y protocolos no propietarios que impidan la fragmentación del mercado, a través del consenso con la industria y de foros de discusión abierta.
5. Evolución. Aunque se reconoce la búsqueda de la excelencia técnica, también se admite que lo que se sabe y se necesita hoy puede ser insuficiente para solventar los problemas del mañana. Se pretende mejorar la Web haciendo que los avances sean compatibles con lo que ahora funciona, basados en la simplicidad, modularidad, compatibilidad y extensibilidad.
6. Descentralización. Un principio de los sistemas modernos distribuidos y cada vez más presente en las sociedades, dado que evita cuellos de botella cuando se incrementa el tráfico de información por la autoridad central.
7. Más y mejor multimedia. El *W<sub>3</sub>C* quiere ser un medio donde mostrar imágenes, sonido, vídeo, efectos 3D, animaciones, con el consenso de los proveedores de contenido, los usuarios finales y la industria. Proyectos como *SVG* (*Scalable Vector Graphics*, gráficos vectoriales escalables) y *SMIL* (*Synchronized Multimedia Integration Language*, lenguaje de integración multimedia sincronizada), *HTML5* y otros.

En 2004 *W<sub>3</sub>C* publica la recomendación oficial de la arquitectura de la *World Wide Web* donde afirman que la Web usa tecnologías relativamente simples con suficiente escalabilidad, eficiencia y utilidad como para que den resultado en un espacio de información de recursos interrelacionados y un número creciente de idiomas, culturas y medios [144]. En el grupo discuten sobre la identificación de los recursos, la representación del estado del recurso y los protocolos que soportan la interacción entre agentes y recursos en el espacio web.

El *W3C* se encargó en 2012 de renovar, actualizar y resumir estos principios [243] y los resumió en tres: estándares abiertos, diseño para todos y visión global:

- Estándares abiertos, firmados por cinco organizaciones que se adhieren a un conjunto de principios que abogan por los estándares de cara al desarrollo de nuevas tecnologías e innovación.
- Diseño para todos en una Web para todas las personas y todos los dispositivos.
- Visión global, lo que incluye participación, compartir el conocimiento fomentando la interacción en el diseño de aplicaciones y en la arquitectura de la Web, a través de los datos y los servicios generados por los datos vinculados, y generar confianza a escala global a través del incremento en la seguridad y la privacidad.

Sobre los enlaces hay dos cuestiones no se consideran principios pero son tanto o más importantes: *permalink* y *deep link*. *Permalink* es una contracción de *permanent* y *link* (enlace permanente), fue creado para su uso en *Blogger* [51] (ahora de *Google*) para asignar una *URL* permanente a cada entrada del *blog*, de cara a que aportara semántica a la *URL*. Cuando la *URL* es nombrada manualmente, su propietario se suele hacer una *URL* semántica, pero un *blog*, al ser un motor de generación de contenidos el que realiza esta acción, suele identificar a la página web con un identificador único, un número, o a través de otras técnicas como la fecha, las etiquetas o las palabras del título. Para mejorar la semántica del *permalink*, se puede configurar el motor para que cree la *URL* conforme a uno o varios de estos elementos, separados por barras /: categoría, autor, fecha y título, donde el título suele ser el conjunto de palabras del título del artículo, separado por guiones, y sin caracteres no *ASCII*.

Otro tipo de enlace es el enlace profundo o *deep linking*, un enlace que lleva a una página o imagen específica de otro sitio web, en vez de llevar a la página principal de ese sitio. Esto no es nada extraño en la Web, pero algunas empresas, por cuestiones de derechos

de autor, por la explotación de la publicidad, por simple desconocimiento de la Web o por otras razones, evitan este tipo de enlaces. El *Technical Architecture Group* del *W3C* advierte que prohibir su práctica es no entender la tecnología y además poner en riesgo el funcionamiento de la Web en su conjunto [233]. Quien quiera establecer algún tipo de restricción al contenido lo debería hacer de otra forma, bien con sistemas de usuario/contraseña, a través de las posibilidades que ofrece *HTTP* [106] o a través del archivo *.htaccess*, que contiene las páginas web que no se permiten utilizar.

También es esencial para la Web la neutralidad de Internet, la no intervención interesada de gobiernos, empresas o instituciones, salvo que sea para mejorar la propia infraestructura. Los agentes implicados y los *ISP* han de tratar a sus clientes por igual, sean personas, empresas o instituciones. El peligro del control sobre la Web y/o Internet es continuo a escala global, incluso en los países más democráticos [39], donde se suceden noticias de espionaje a los ciudadanos [39]. En parte por ello, Berners-Lee impulsó el *W3C World Wide Web Consortium* (Consortio de la Web), la organización internacional que publica los documentos técnicos en los que se basa la Web, y en 2008 ha lanzado la *Fundación de la World Wide Web* (*World Wide Web Foundation* o *Web Foundation*), organización dedicada a la mejora de la Web como un bien público, abierto y participativo, como un derecho básico, que asegure su acceso universal y libre, en un contexto democrático. También defiende un sistema para dar a los sitios web que están probados que contienen información veraz una insignia.

#### 5.4. W3C

El *Consortio World Wide Web* o *W3C* fue creado el 1 de octubre de 1994 desde el *laboratorio de Ciencias de la Computación e Inteligencia Artificial* del *MIT*, con la colaboración del *CERN* y el apoyo de *DARPA* y la *Comisión Europea*. Un año después se unía el *Instituto Nacional de Investigación en Informática y Automatismos* (*INRIA*, *Institut*

*National de Recherche en Informatique et Automatique*, Francia) y al año siguiente la *Universidad de Keio* en Japón. En 2003 el *Consortio Europeo para la Investigación en Informática y Matemáticas* (*European Research Consortium in Informatics and Mathematics*) heredó el rol de *INRIA* y en 2013 se sumó la *Universidad de Beihang* (China), como cuarta entidad. Existen además oficinas regionales en muchos países, entre ellos España. La sede española se inauguró en 2003 en la *Fundación CTIC*, en el *Parque Científico Tecnológico* de Gijón.

La versión estable de los documentos que rigen la Web se denominan *recomendaciones W<sub>3</sub>C*, y funcionan como tal, es decir, *W<sub>3</sub>C* recomienda que se utilicen de esa manera, pero no obliga ni prevé ningún tipo de medida sancionatoria o de otro tipo para quienes no lo hagan. Sin embargo, sí que aboga por la libre y abierta participación de todos los actores implicados para crear conjuntamente los estándares de la web en el espíritu de accesibilidad y universalidad. Si lxs autorxs de contenido, creadorxs de programa de navegación o cualquier otra persona que participa en la Web de alguna manera quisiera realizar una modificación, podría hacerlo, pero su alcance estaría limitado. Si una persona introduce, por ejemplo, un elemento *HTML* en una página web, como los navegadores no lo saben interpretar, lo mostrarán como texto y no como elemento *HTML*.

Sin embargo, si los cambios o modificaciones provienen de otras partes, como, por ejemplo, del navegador, puede ocurrir que una web se vea de determinada forma en un navegador y de forma muy distinta en otra. Así ocurrió durante un tiempo con las implementaciones que hubo de *HTML* y de *JavaScript* por parte de *Netscape* e *Internet Explorer* [253].

El incumplimiento de las recomendaciones del *W<sub>3</sub>C* es mucho más extendido de lo que podríamos pensar, y no molesta más que a una minoría de usuarixs que son o bien creadores de contenido, diseñadorxs web o usuarixs de software que respeta más escrupulosamente la recomendación. El problema de la falta de código compatible con *HTML* se ha producido desde el origen de la Web. Las causas son variadas, aunque podríamos

resumirlas en tres: desconocimiento, desinterés e inercia. Muchas personas, empresas o entidades desconocen las tecnologías web con la profundidad necesaria y no tienen ningún interés en aprender más en cuanto que su modo de realizar webs funciona, lo que produce una inercia en los procesos de creación y mantenimiento de contenidos web que va en perjuicio de los propios contenidos.

La cuestión de la normativización y la estandarización es una cuestión fundamental tanto en el desarrollo industrial y tecnológico como en la organización del contenido y la clasificación de la información. En el caso de Internet, no se realizó bajo el amparo de las *Naciones Unidas*, a través de su *Asociación Internacional de las Telecomunicaciones (International Telecommunications Union, ITU)*, como habría sido más lógico, por el temor de que el proceso de toma de decisiones de los países que lo conforman fuera árduo y lento [72]. Apostaron, en cambio, por tomar un camino independiente y crearon *Internet Engineering Task Force* o *IETF* donde cualquier interesado podía participar en reuniones, pruebas o decisiones. Paradójicamente, cuando Berners-Lee acudió al *IETF*, lo consideró demasiado lento, por lo que creó su propio organismo, el *W3C* [34, p. XI]

La flexibilidad en la normativización del *W3C* se basa en el uso de tecnologías relativamente sencillas que poseen la suficiente escalabilidad, eficiencia y utilidad como para que funcionen en un notable espacio de información de recursos interrelacionados y de idiomas, culturas y medios en crecimiento [143]. En ese proceso, los usuarios han visto distintas denominaciones o etiquetas como *Web 1.0*, *Web 2.0*, *Web 3.0*, amén de otras definiciones nominativas, como la propia Web Semántica, la *web sintáctica*, la *web pragmática*, la *web de datos* o *GGG*. Al ser la Web software, es habitual utilizar en las distintas versiones del software etiquetas que indiquen su estado. Normalmente se utilizan:

- *Alfa*, primera versión del programa, inestable, requiere pruebas que encuentren inconsistencias.

- *Beta*, se han eliminado los errores detectados.
- *Release Candidate (RC)* o versión candidata a lanzarse, se considera muy estable y pendiente de aprobación final.
- *Release to manufacturing (RTM)*, versión lista para lanzarse, de disponibilidad general, versión dorada o versión final. Es la que se va a constituir como la versión maestra, de la que se harán copias.

Al versionar el software, se asigna un código normalmente de dos números, en inglés, *major.minor*, aunque puede haber incluso un tercero llamado *micro*. El número depende de:

- *Major*: si el software ha cambiado mucho.
- *Minor*: si el software tiene algunas modificaciones, mejoras o ha arreglado errores.
- *Micro*: para modificaciones menores.

También se puede nombrar el software según la fase de desarrollo:

- Final o estable, *stable*, si ya es una versión *RTM* (*release to manufacturing* o versión lista para comercializarse).
- Inestable o *unstable*, si todavía está en fase de pruebas.

Todo ello se traduce en nombres como los siguientes de versiones:

- *1.1*: *major./minor/* (mayor o menor)

- *1.1.1: major. minor. micro*
- *1.1-alpha: major./minor/-/fase/*
- *1.1 Beta: mayor. menor fase.* Cuando se separa con un espacio, va en mayúsculas.
- *1.1-rc1: mayor. menor- versión<sub>fase</sub>, release candidate* o versión candidada a ser lanzada.
- *1.1 RC1: mayor. menor versión fase.*

En el caso del *W<sub>3</sub>C*, las recomendaciones siguen cuatro estados:

- *Working Draft, WD*, borrador de trabajo, documento publicado por *W<sub>3</sub>C* para que la comunidad, incluidos miembros de *W<sub>3</sub>C*, lo comenten.
- *Candidate Recommendation, CR*, recomendación candidata, cuando *W<sub>3</sub>C* cree que se ha revisado lo suficiente y satisface los requerimientos técnicos del *Working Group*. Ahora se busca retroalimentación para su puesta en práctica.
- *Proposed Recommendation, PR*, recomendación propuesta es un documento técnico maduro que ha sido ampliamente revisado y que se envía al comité consultivo *W<sub>3</sub>C Advisory Committee* para su aprobación final.
- *W<sub>3</sub>C Recommendation, REC*, recomendación *W<sub>3</sub>C*, un conjunto de directrices que constituyen una especificación construida con un amplio consenso y que ha recibido el apoyo de los miembros del consorcio y del director.

## 5.5. Web 2.0

El nacimiento del término *Web 2.0* proviene de los miembros de *O'Reilly Media*, en una lluvia de ideas con *MediaLive International* donde abogan por la importancia de la Web pese al pinchazo de la burbuja de las *puntocom* en 2001 [181]. Dado que habían sobrevivido empresas y seguían surgiendo ideas, por lo que decidieron crear la *Conferencia Web 2.0* donde tratar estos asuntos de la evolución de la Web y conocer las nuevas iniciativas y tecnologías.

En la lluvia de ideas apuntan algunas cuestiones sobre la evolución de la Web 1.0 a la 2.0:

#+ATTR<sub>LATEX</sub>:caption Evolución de la Web 1.0 a la 2.0 :width 0.30

Web 1.0	Web 2.0	Ámbito
DoubleClick	Google Adsense	Publicidad
Ofoto	Flickr	Imágenes
Akamai	BitTorrent	Compartir y servir contenido
mp3.com	Napster	Música
Sitios web personales	Blogs	Contenido personal
evite	upcoming.org y EVDB	ns
Especulación sobre el nombre del dominio	Optimización del motor de búsqueda	ns
páginas vistas	coste por click	Retribución, pago
<i>screen scraping</i>	servicios web	servicios web
publicación	participación	contenidos
sistemas de gestión de contenido	wikis	contenidos
directorios, taxonomía	etiquetado, folksonomía	organización del contenido
stickyness	syndication	recuperación del contenido

Las empresas *necesitaban* seguir pensando en la Web como espacio donde realizar negocios [181], por lo que el término *Web 2.0* también pretendió maquillar la Web, quitarle la máscara de fracaso que habían sufrido las empresas que lanzaron negocios anteriormente y dar valor a las innovaciones tecnológicas que se estaban realizando y que apostaban por una mayor participación de las personas en la creación de contenido web.

Si en la *Web 1.0* las personas eran principalmente lectorxs, la *Web 2.0* pretende que sean creadorxs de contenido, añade participación de lxs usarixs. Es el territorio del contenido generado por lxs usarixso, *User Generated Content* o *UGC*, de la especialización y la adaptación, pero también de:

- Globalidad. Si en 1996 había unos doscientos cincuenta mil sitios web y cuarentaycinco millones de usarixs mientras que en 2006 eran ochenta millones de sitios web y más de mil millones de usarixs. La cifra alcanza los 1900 millones de usuarios en 2010. En septiembre de 2014 se alcanzó la cifra de mil millones de sitios web [140]. Un año después, la cifra ha bajado de los mil millones pero se mantiene por encima de novecientos mil.
- Movilidad. La forma de acceso ya no es solo a través del ordenador personal, el -ahora- viejo *PC*, sino desde dispositivos portátiles y móviles, cada vez más desde el teléfono móvil y tabletas. En julio de 2014 el acceso a la Web comenzó a ser mayor desde dispositivos móviles que desde entornos de escritorio [168].
- Redes sociales. Podría parecer de perogrullo pretender que las personas no sean solo lectorxs sino también creadorxs de contenido, algo que está en los fundamentos de la Web [34].
- Medios. Los medios de comunicación observan también un desplazamiento de la lectura de la prensa a la Web, lo cual es una oportunidad para los medios que quieran participar y es considerado una amenaza por quienes pretenden mante-

ner el modelo de negocio anterior.

El mapa mental de la *Web 2.0* señalaba estas ideas:

- Posición estratégica: la Web como plataforma.
- Desde el punto de vista de los usuarios: control sobre tus datos.
- Competencias principales:
  - Se promueven más los servicios que las aplicaciones o programas informáticos.
  - Se potencia una arquitectura de participación.
  - Escalabilidad sostenible.
  - Fuentes de datos que permiten la mezcla y la transformación.
  - Software por encima de un solo dispositivo.
  - Aprovechamiento de la inteligencia colectiva (como si fuera tan fácil como la comida rápida [77]).

La Web como plataforma ya se había intentado en la *Web 1.0* en la batalla que mantuvieron *Netscape* vs *Microsoft* [181]. Al contrario que *Netscape*, que pretendió aprovechar su posición de dominio en la *Web 1.0* para crear una plataforma de acceso a servicios especializados y así competir en el escritorio con *Microsoft*, *Google* comienza como una aplicación web, una base de datos especializada, y termina comprendiendo una serie de programas y un sistema operativo en el móvil, *Android*.

Algunas empresas o iniciativas que aprovecharon este paradigma de la inteligencia colectiva basada en la participación de lxs usuarixs fueron *Yahoo!*, *Google*, *Amazon*, *Wikipedia* o *Flickr*. *Yahoo!* pasa de ser un directorio de enlaces y sitios agregados a un conjunto de servicios para la generación de contenido por parte de lxs usuarixs. Los hiperenlaces creadxs por lxs usuarixs generan una web de conexiones que *Google* incorpora a *pagerank*, una valoración de la página que no se basa tanto en el su contenido, en la estructura del documento, como en su posicionamiento en la web. *Amazon* consigue, con el mismo contenido que cualquier otra tienda de libros, ser la preferida por lxs usuarixs por su propuesta de interacción y contenidos relacionados. La *Wikipedia* permite la edición colaborativa y multidioma para su proyecto de enciclopedia libre y abierta. *Flickr* introduce el concepto de *folksonomía* frente a taxonomía, donde lxs usuarixs proponen la categorización a través de sus propias palabras clave o etiquetas, lo que permite atender mejor a las conexiones cerebrales que se producen entre palabras en vez de amoldarse a las rígidas taxonomías [181].

Los blogs se desarrollan en la *Web 2.0*. Suponen una evolución de las páginas personales hacia el *UGC* y se alían con una de las tecnologías más importante de la Web, *RSS* (*Really Simple Syndication* o *sindicación -de contenidos- muy simple*). *Blog* es la contracción de *web log* o registro web, traducido en un primer momento como *cuaderno de bitácora* (diario del barco). Los registros o *logs* son habituales en todo programa informático, sirven para conocer la actividad del programa, están organizados temporalmente, y también para depurar errores. Los *blogs* recogen este concepto, lo enlazan con la bitácora y lo publican a través de *RSS*, donde el contenido aparece ordenado de más nuevo a más viejo.

*RSS* cuenta con una ventaja sobre los buscadores tradicionales [209], ya que la información, los contenidos que encuentra, no están organizados cronológicamente sino por otras consideraciones como el peso específico que ese buscador otorga a determinada web. Con *RSS* se puede acceder al contenido ordenado cronológicamente, convenientemente etiquetado y plenamente semántico (*RDF*).

Ricardo Domínguez recuerda la *Web 1.0* como un espacio para la realización de la persona sin necesidad de identidad, género o raza, que posibilitaba una existencia anónima [91], mientras que la *Web 2.0* apuesta por la identidad y la identificación de lxs usuarixs, en vez de una mayor transparencia, una profundización en la relación entre los cuerpos digitales y los cuerpos reales en acción.

El concepto de la *Web 2.0* se utilizó hasta 2011, año en el que se realiza el último congreso *Web 2.0* donde se vuelve a hablar de lo mismo, por última vez, bajo el paraguas de *Web 2.0*

## 5.6. Web 3.0

El concepto *Web 3.0* responde a la necesidad de actualizar la Web una vez superadas las tecnologías que más relevancia tuvieron –y que siguen utilizándose– en la Web 2.0. El diseñador web Jeffrey Zeldman emite una crítica en 2006 sobre el uso de los términos por parte de iniciativas comerciales [274] que parecen alejadas del mundo web que él habita, donde aplicaciones como *Flickr* o *Basecamp* no parecían responder al esquema de la *Web 2.0*. Por tanto se plantea por qué no dar un paso adelante y englobarla esas aplicaciones en otra etiqueta distinta, la *Web 3.0*. *Flickr*, una web donde compartir imágenes, una red social de usuarixs y aficionadxs de la fotografía, ponía el acento, al igual que *Basecamp*, en algunas características novedosas que podían caracterizar la *Web 3.0*: diseño simple, limpio, respeto por los estándares de la Web, innovación en la experiencia de usuarix a través de *JavaScript*, visualización en distintos navegadores y dispositivos y un modelo de negocio donde lo *freemium* resultaba suficiente para lxs usuarixs normales y lo *premium* no era excesivamente caro. *Flickr* fue comprada en 2005 por *Yahoo!*

A estas características se unen las máquinas como creadoras de contenido. Si la *Web 1.0* supuso el acceso al conocimiento humano y la *Web 2.0* lo puso en común, la *Web 3.0* pre-

tendía aplicar el conocimiento que se encuentra en la Web [126]. Ya no solo lxs humanxs participan y aportan contenido semántico sino que también las máquinas –los ordenadores personales, los servidores– son capaces de hacerlo. Recuperar el conocimiento de la Web, esa inteligencia colectiva que se ha puesto en común en la *Web 2.0*, era el reto de la *3.0*. El conocimiento colectivo se expresa en:

- Foros de ayuda y de opinión, evolución de los grupos de noticias.
- Periodismo ciudadano a través de *blogs*.
- Filtros colaborativos para la industria cultural (libros, música, películas).

Los sistemas de conocimiento colectivo [125] están basados en:

- Ofrecer información nueva y de valor, de entre grandes cantidades de datos.
- Contribuciones de personas.
- Mejorar en función de la participación de lxs usuarixs.

La *Web 3.0* añade a estos sistemas algunas características propias:

- Las colecciones de datos son múltiples
- Las contribuciones personales se ven mejoradas por los datos estructurados.
- Se ofrece respuestas, soluciones o cualquier otro resultado extraído de los datos originales.

Cualquier usuarix de una red social o aplicación web o de móvil crea a diario datos estructurados y semánticos que utilizan las empresas para aumentar el conocimiento so-

bre sus consumidorxs. El reto de la *Web 3.0* es convertir a lxs usuarixs en productorex conscientes de datos estructurados, compartir los datos de su elección y aprovecharlos para beneficios múltiples.

## 5.7. Giant Global Graph

Tan solo un año después de aparecer el concepto *Web 3.0*, en 2007, nació otro que explicaba el estado de la Web y de la Web Semántica: *Giant Global Graph* (Grafo Global Gigante) [36], que pone el acento en la distinción entre la naturaleza y la importancia de los contenidos de la Web ya existente y de la Web que, aunque ya existía, estaba por venir de manera masiva, la Web Semántica.

El concepto de grafo se encuentra en el propio logo del lenguaje base de la Web Semántica: *RDF*. La *Web 1.0* y la *Web 2.0* se puede representar como un gráfico de red, donde un enlace establece una relación entre una página y otra, por lo que se podría saltar también entre webs y construir otro gráfico gigante mundial. Sin embargo, *RDF* introduce tres conceptos novedosos:

1. La estructura de tripleta de datos, similar a las oraciones con estructura *Sujeto - Predicado - Objeto*.
2. La semantización del contenido, lo cual permite no solo dotar de semántica al documento web sino a cada pieza de contenido que quiera escribirse en *RDF*.
3. La *URI* que identifica cada pieza de contenido, cada dato.

Por tanto, no solo visualizamos documentos o enlaces sino datos, todos los datos que contengan las páginas web en formatos semánticos con otros datos de otras webs. Si la

*Web 1.0* conectaba ordenadores, la *Web 2.0* conectaba personas e Internet conecta los ordenadores, la *GGG* conecta lo que dicen los documentos, por lo que también se le ha denominado *Web de Datos (Web of Data)*.

En relación al contenido de las redes sociales, lamentablemente, parece haber quedado en manos de las comunidades de internautas más activxs la demanda de intercambio de datos entre aplicaciones, de forma que día tras día diseminamos datos enriquecidos en aplicaciones web o redes sociales y lxs usuarixs no pueden interconectarlos ni las aplicaciones promueven su interconexión. Algunas comunidades lanzaron la *Carta de derechos de los usuarios de la Web social* [180], publicada por primera vez en [socialweb.org](http://socialweb.org), y que pedía:

- Propiedad sobre los datos personales, incluidos:
  - Los del perfil
  - La lista de contactos
  - El flujo de contenidos que se crea (el *stream*)
- Control sobre cómo la información se comparte con otrxs usuarixs y libertad para ofrecer acceso a su información personal desde sitios de confianza externos.

Los sitios que cumplan estos derechos deberían permitir a lxs usuarixs:

- Sindicar contenidos de su propio perfil, de su lista de amistades y de los contenidos que se comparten a través del servicio, bien con una *URL* persistente o una clave de acceso a la *API* y formatos de datos abiertos.
- Sindicar su propio caudal de actividad fuera del sitio.

- Enlazar desde su perfil páginas externas.
- Descubrir otras personas que conozca en la red social a través de los mismos identificadores que se utilizan para buscar en la propia red.

En este manifiesto aparecen varias cuestiones muy importantes:

- La propiedad de los contenidos web.
- La propiedad de los contenidos de las redes sociales.
- Las *API*.
- Los datos abiertos.

La *GGG* también habla, al contrario que la *Web 3.0*, de datos vinculados abiertos, donde el contenido semántico pueda ser leído por distintos sistemas y organizada dinámicamente en distintos formatos legibles por lxs usuarixs. Que pueda ser leído por cualquiera también remite a otro concepto, el de la descentralización de los contenidos de la Web donde objetos de datos correctamente formateados se puedan organizar y sus relaciones discernir por cualquier ordenador y no sólo por grandes sistemas centralizados o empresas.

Hay algunos ejemplos, como el protocolo *FOAF* (*Friend Of A Friend*, amigx de un amigx), que permite organizar la información de los sitios web u otros nodos de Internet sin requerir necesariamente la intervención de sistemas centralizados. La *GGG* es una especie de red o web social en el interior de la Web. De una infraestructura que transmite información de documentos entre distintos ordenadores se pasa a una infraestructura de catalogación de los contenidos y de enlaces entre los documentos para llegar a los datos vinculados abiertos expresados en *RDF*. Se puede empezar expresando la red personal de contactos y los direcciones de los distintos servicios de Internet (web,

correo electrónico, mensajería instantánea) a través de un archivo *FOAF*, "el comienzo de la revolución" de los datos vinculados [36].

*Linked Data (LD)* o *Datos Vinculados* es una propuesta para construir la Web Semántica mediante el establecimiento de relaciones o vínculos entre los distintos datos que participan en la Web Semántica. Un precedente se encuentra en el proyecto *SOLO (Structure of Linked Objects)* o estructura de objetos vinculados), el software en que se basaron la gama de productos de la empresa *NetObjects* creada por Samir Arora, antiguo miembro del *Navigation Department* (departamento de navegación) de *Apple*.

Las cuatro reglas para vincular datos [35] van encaminadas a crear enlaces entre los datos, hacer datos vinculados, para que personas o máquinas puedan expresar la *Web de Datos* y datos relacionados, para lo cual:

1. Hay que utilizar *URI* como forma de nombrar o identificar cosas.
2. El uso de *HTTP URI* permite que se puedan buscar esos nombres.
3. Al buscar una *URI*, con *RDF* y *SPARQL* se puede ofrecer información adicional.
4. Se incluyen enlaces a otras *URI* para descubrir y relacionar contenidos, lo cual es la potencia de los datos vinculados.

El documento se actualizó en 2010 [35] y se añadió una escala de cumplimiento con cinco puntos o estrellas que son también un esquema de desarrollo de políticas relacionadas con datos.

- Una estrella si los datos están disponibles en la Web con una licencia abierta, por ejemplo, un *PDF*, aunque sea una imagen. Esto permite ver los datos, imprimirlos, almacenarlos, ponerlos en otro formato, cambiarlos y compartirlos.

- Dos estrellas, si los datos además están en un formato de datos estructurado, como, por ejemplo, *XLS* (*Microsoft Excell*).
- Tres estrellas si además están en un formato abierto como *Open Document Spreadsheet* o *CSV*.
- Cuatro estrellas, si se utiliza *URI*, por lo que se permite a otrxs apuntar a esos datos.
- Cinco estrellas, si se enlazan los datos con otros para ofrecer un contexto más amplio. Estos son los datos vinculados abiertos.

## 5.8. Datos abiertos

Uno de los fenómenos más interesantes de la Web es el movimiento por los datos abiertos u *Open Data*. La comunidad de los datos abiertos se compone de ciudadanxs pero también de empresas, instituciones e incluso gobiernos, ya que además de promoverlo y difundirlo es fundamental practicarlo. Cuando Barack Obama llega a la presidencia de los EE.UU. de Norteamérica emite un memorándum, la *Ley de Libertad de Información* (*Freedom of Information Act*) dirigido a los departamentos federales [38] donde anuncia la apertura de datos del gobierno para favorecer la rendición de cuentas del gobierno, la transparencia en la toma de decisiones y la claridad frente a las sombras. En 2009 lanzaron el sitio web de datos abiertos del gobierno, *data.gov*, con cientos de miles de conjuntos de datos de 295 *API* gubernamentales y 175 agencias federales. Su política ha sido imitada por otros países desarrollados como Reino Unido, Australia, Japón, Corea del Sur, Alemania o Francia.

España ha tardado tiempo en incorporarse a las buenas prácticas de apertura de datos

y transparencia. Comenzaron los portales autonómicos de Aragón, Euskadi y Asturias, siendo el de Aragón y también el de la ciudad de Zaragoza los mejores portales de datos abiertos en cuanto a cantidad y calidad de los datos. En 2013 se aprobó la *Ley de Transparencia* de España y en 2014 comenzó sus actividades el portal de la transparencia en *transparencia.gob.es*

El movimiento de los datos abiertos bebe de las fuentes de los orígenes de Internet, de Unix y de la Web, y entronca con los valores democráticos de transparencia, rendición de cuentas, participación ciudadana y bienes comunes. Estos portales de datos promueven o favorecen la existencia de aplicaciones y herramientas web que ponen en práctica esos valores. La Web permite la disposición universal de información, por lo que comunidades de usuarios interesados en el acceso a los datos apuestan por políticas que protejan e incentiven el uso de datos abiertos, de tal forma que estén disponibles para cualquiera que quiera utilizarlos, sin restricciones de copyright, patentes o cualquier otro mecanismo de control, en el ámbito académico, de investigación, periodístico, público o empresarial.

El grupo de difusión y educación en Web Semántica (*SWEO, Semantic Web Education and Outreach*) ha trabajado en el desarrollo de estrategias y materiales para aumentar la preocupación en la comunidad Web de la necesidad y la utilidad de la Web Semántica y de educar a la comunidad Web en soluciones y tecnologías relacionadas [236]. Aunque este grupo de interés cerró en 2008, realizó proyectos como *Linking Open Data on the Semantic Web, vinculando datos abiertos en la Web Semántica*, donde se recogían conjuntos de datos *RDF* provenientes de iniciativas de datos vinculados o *Linked Data*.

La iniciativa que continuó el trabajo relacionado en *W3C* se denominó *Linking Open Data W3C SWEO Community Project* y pretendía también extender la Web con los datos del *procomún*, los conjuntos de datos que cuentan con cinco estrellas [35], en *RDF*, con enlaces con otras fuentes de datos [242] y disponibles normalmente con licencias abiertas como *Creative Commons*. Entre estos conjuntos de datos se incluyen la enciclo-

pedia colaborativa *Wikipedia*, la biblioteca *Wikibook*, la base de datos musical *MusicBrainz*, el diccionario *Wordnet*, la base de datos geográfica *GeoNames* o la base de datos de bibliografía científica *DBLP* (*Digital Bibliography & Library Project*).

El proyecto *Open Knowledge* (conocimiento abierto. *OK*) es una red mundial sin ánimo de lucro de personas apasionadas por el conocimiento abierto y participan activamente en el movimiento de los datos abiertos. Ha sido pionero en la creación, difusión y promoción de licencias libres para conjuntos de datos, de cara a que cualquier tipo de dato y contenido pueda ser publicado, utilizado, reutilizado y redistribuido libremente. Si el proyecto *GNU* realiza licencias libres para el software y *Creative Commons* para obras artísticas, los conjuntos de datos que quieran tener licencias abiertas pueden contar con las licencias de *OK*.

Realizan iniciativas de promoción y difusión de los conjuntos de datos, como *DataPortals.org* que recoge 431 portales de datos en todo el mundo, o *School of Data*, que ofrece formación y alfabetización en datos, incluidas sesiones de periodismo de datos para activistas sociales o ciudadanxs. Entre sus proyectos web destacan *Government Data Camp* (campamento de datos del gobierno), *Where Does My Money Go* (dónde va mi dinero) y *Open Shakespeare* (Shakespeare abierto). También desarrollan software, servicios de conversión, limpieza y estandarización de datos, y mantienen el sistema de gestión de contenidos para portales de datos abiertos *CKAN*, uno de los más empleados para recoger conjuntos de datos abiertos.

También destaca en el mundo de los datos abiertos vinculados *DBpedia*, un esfuerzo comunitario para extraer información estructurada de la *Wikipedia* y tenerla disponible –recuperable– en la Web con estructura de datos semánticos. *DBpedia* (*DB* por *database* o base de datos y *pedia* de enciclopedia), permite realizar búsquedas de los datos estructurados de Wikipedia y enlazarlo con otros conjuntos de datos disponibles en la Web en formato *RDF*, por lo que forman parte de los datos abiertos vinculados o *Linked Open Data*.

*Wikipedia* dispone en muchos contenidos desestructurados, pero la información que visualmente aparece como una caja gris al margen derecho, denominada *infobox* (caja de información), contiene información estructurada acorde con las normas de estilo y de formato de *Wikipedia*. Así, por ejemplo, cada río que aparece en la *Wikipedia* tiene la misma caja base donde se muestra la información disponible de ese río con la misma estructura. *DBpedia* convierte esa información a formato *RDF*. De esta manera, *DBpedia* puede considerarse un espejo web semántico de *Wikipedia*. Además, comparte esta información con licencia *Creative Commons BY-SA 3.0* y *GFDL* (*GNU Free Documentation License*, licencia de documentación libre de *GNU*).

La versión 3.9 de *DBpedia* ofrece datos semánticos de más de cuatro millones de entidades [99], de las que más de tres millones están clasificadas conforme a la ontología *DBpedia*, incluidas 832.000 personas, 639.000 lugares o 116.000 álbumes musicales. Ofrece etiquetas y resúmenes de esos datos en 119 idiomas. Toda la base de conocimientos de *DBpedia* se compone de más de 2.460 millones de datos semánticos, información en tripletas *RDF*, de los cuales 470 corresponden a la edición inglesa de la *Wikipedia* y 100 millones a la versión española. La base de conocimientos de *DBpedia* posee varias ventajas sobre otras bases de conocimiento, como, por ejemplo, que cubre muchos dominios, evoluciona automáticamente al tiempo que cambia la *Wikipedia* y es multi idioma.

## 5.9. Accesibilidad del contenido web

El acceso al contenido fue uno de los motivos por los que Berners-Lee creó la web, para acceder de una forma estándar y universal al contenido que antes se encontraba en documentos escritos en distintos formatos, en distintos sistemas operativos, distintas aplicaciones y distintos tipos de red. Sin embargo, la accesibilidad web se refiere también al acceso real a los contenidos web, es decir, a que cumplan los estándares y tengan en cuenta a distintos escenarios, resoluciones, dispositivos, situaciones y a distintas

personas con distintas condiciones, motoras o funcionales, temporales o permanentes [101].

El W<sub>3</sub>C creó la *WAI Web Accessibility Initiative* o *Iniciativa por la Accesibilidad de la Web* para reforzar el trabajo realizado y por realizar en cuestiones de accesibilidad del contenido web. Como resultado de su trabajo, lanzó *WCAG (Web Content Accessibility Guidelines)* o directrices de accesibilidad del contenido web, cuya versión 1.0 se convirtió en recomendación oficial en 1999 y en 2008 se publicó segunda versión, *WCAG 2.0* [66].

Los cuatro principios básicos que proporcionan los fundamentos de la accesibilidad web [127] son perceptible, operable, comprensible y robusto:

1. Perceptible, el contenido y los componentes de la interfaz de usuario deben presentarse de modo que puedan ser percibidos por los usuarios.
2. Operable, los componentes de la interfaz deben ser operables, permitiendo el acceso a toda funcionalidad mediante el teclado, el tiempo suficiente para leer y usar el contenido.
3. Comprensible, el contenido y la interfaz deben ser comprensibles.
4. Robusto, de forma que pueda interpretarse el contenido por cualquier aplicación de usuario, conforme a los estándares de la Web.

Estos principios se traducen en doce pautas que ofrecen los objetivos básicos para crear un contenido más accesible para los usuarios. No son verificables pero sí los criterios de conformidad que desarrolla cada pauta, que establece una serie de valores de accesibilidad en una escala de tres niveles:

- A, el nivel básico, se suele alcanzar si el documento web cumple con los estándares

de la Web.

- AA, el nivel intermedio, requiere algunas consideraciones extras.
- AAA, el nivel más alto de conformidad y el que ofrece mayor accesibilidad.

---

## Capítulo 6

# HTML

*HTML* es el acrónimo para *HyperText Markup Language*, literalmente, Lenguaje de Marcado de Hipertexto, y es el primer lenguaje de los documentos de la Web, si bien no es el único. *HTML* no se trata de un lenguaje de programación sino de un lenguaje de marcado del contenido. Nace para permitir intercambiar información entre distintos equipos con distintos sistemas operativos y distintas aplicaciones y codificaciones. *HTML* ofrece una lengua franca, un estándar para compartir contenidos.

Se basa en el lenguaje *SGML* (*Standard Generalized Markup Language*, lenguaje de marcas estándar generalizadas), un metalenguaje para la creación de lenguajes a partir de plantillas de documentos expresados en *DTD* (*Document Type Definition*, definición de tipo de documento). *HTML* es por tanto una *DTD* de *SGML*, de donde hereda la sintaxis y algunos elementos.

El término *HTML* también se utiliza para referirse al contenido del tipo de *MIME* (*Multipurpose Internet Mail Extensions*, extensiones de correo de Internet multipropósito), `text/html`, o como un término genérico [40] para *HTML*, ya sea en las ver-

siones compatibles con *XML* como *XHTML 1.0* o en las posteriores descendientes de *SGML*.

*HTML5* ha supuesto una revolución en el mercado del contenido. Entre medias quedan los intentos de hacer *HTML* compatible con *XML*, de donde surgió *XHTML*, y el denunciado inmovilismo del *W3C* en el desarrollo de *HTML* [246] que llevó a un grupo de desarrolladores a dejar de lado a *W3C* para desarrollar su trabajo en *WHATGW* (*Web Hypertext Application Technology Working Group* o grupo de trabajo de Web, hipertexto, aplicaciones y tecnología), aunque luego han confluído de nuevo con *HTML5* adoptado oficialmente por *W3C*.

## 6.1. Inicios

La primera descripción de *HTML* disponible públicamente se encuentra en *HTML Tags* [20] (*etiquetas HTML*) de 1991 y cuenta con veintodós elementos, trece de los cuales todavía están disponibles en *HTML5*:

- A, de *anchor* o ancla, utilizado para enlazar hipertexto a través de su atributo HREF. Ya contemplaba el uso de enlaces internos enlazando con otros elementos A que emplearan el atributo NAME para identificarse. Así, por ejemplo:

```
<P>Esto es un <A HREF=#prueba>Enlace</A> a otro párrafo donde una palabra está con  
<P>En este párrafo está el destino del párrafo anterior, <A NAME=prueba>aquí</A>
```

- P, de *paragraph* o párrafo, para indicar un nuevo párrafo.
- *Headings* o encabezamientos: H1, H2, H3, H4, H5, H6, según su importancia, donde solo puede haber un H1

- *Address* o dirección, para añadir información sobre la dirección de postal o la firma del documento.
- *Glossaries* glosarios o listas de definición, para definir términos, con los elementos:
  - DL, *definition list* o lista de definición
  - DT, *definition term* o término a definir.
  - DD, *definition description* o descripción del término definido.
- *Lists* listas, con dos elementos:
  - UL, *unordered list*, lista desordenada.
  - LI, *list item*, elemento de la lista.

Las diferencias principales de este *HTML* con los que vinieron y definitivamente con el actual son:

- La sintaxis en minúsculas de los elementos, lo cual es más sencillo y rápido.
- Los valores de los atributos están entrecomillados con comillas dobles.

Con respecto a la característica de realizar enlaces internos, cualquier elemento que contenga el atributo `id` puede enlazarse de la misma manera que el anterior ejemplo:

```
<p>Esto es un <a href="#prueba">enlace</a> a otro párrafo donde una palabra está c  
<p>En este párrafo está el destino del párrafo anterior, <span id="prueba">aquí</a
```

El camino de la estandarización lo emprendieron en 1993 en el *IETF*, donde se definieron etiquetas para imágenes, tablas y formularios, pero no consiguieron convertirlo en estándar oficial, no fue formalmente reconocido como tal hasta la publicación por *IETF* en 1995 de una primera proposición para una especificación de *HTML*: el boceto *Hypertext Markup Language*, realizado junto con Dan Connolly, convertido en *RFC 1866* [41] y conocido como *HTML 2.0*, el cual incluía una *DTD SGML* para definir la gramática. La versión *HTML 3.2* se convirtió en *Recomendación W3C* en 1997 [196], mientras que la versión *HTML 4.0* lo fue en 1998 [194] y la *4.01* en 1999 [195].

## 6.2. Sintaxis HTML

*HTML* se utiliza tanto para describir la estructura como el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. *HTML* puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (normalmente, *JavaScript*), el cual puede modificar el comportamiento de navegadores web y otros procesadores de *HTML*.

De cara a la realización de la Web Semántica, *HTML* conllevaba algunos problemas porque definía más la presentación que el contenido, no escalaba conforme a las necesidades de la industria y al permitir inconsistencias en el marcado (cierre de elementos ausentes, anidación incorrecta o falta de elementos estructurales), presentaba dificultades para ser procesado por máquinas. *HTML5* conserva muchos elementos de *HTML* y añade algunos nuevos que definen más la estructura del contenido, por lo que aportan más semántica.

*HTML* es un lenguaje de marcas, que se representan por los signos de menor que < y mayor que >. Los elementos *HTML* tienen etiqueta de inicio y etiqueta de fin. Por ejemplo, el elemento `p` corresponde a párrafo, y se escribiría así:

```
<p>Esto es un párrafo</p>
```

La etiqueta de cierre es la misma que la de inicio más una barra / antes del elemento. Sin embargo, hay elementos que no disponen de etiquetas de cierre porque son *autocontenidos*, como, por ejemplo, una imagen, donde no se *pega* la imagen sino que se enlaza la imagen con un atributo del elemento `img`:

```

```

En estos elementos la propia etiqueta de inicio es también de cierre, y por eso se escribe la barra invertida antes del *mayor que* final.

Hay elementos que a su vez pueden contener otros elementos, como, por ejemplo, un elemento párrafo `p` que contiene una palabra enfatizada con el elemento `strong`:

```
<p>Esto es un <strong>párrafo</strong></p>
```

Otra clasificación que entiende *HTML* es la de elementos de línea y elementos de bloque. Los elementos de línea son los que se insertan en una línea de texto, y pueden estar dentro de elementos de bloque. Siguiendo el ejemplo anterior, `strong` es un elemento de línea y `p` es un elemento de bloque.

El documento *HTML* consta de dos partes bien diferenciadas, que son a su vez dos elementos de bloque: `head` y `body`.

La cabecera del documento *HTML* es el elemento `head`. Los elementos que incluimos en la cabecera no están visibles, salvo en el caso de `title`, que da título a la página web y suele ser el texto que aparece en la parte superior del navegador. El resto de elementos que se incluyen en `head` se utilizan para declarar qué tipo de documento es con el elemento `doctype`, el elemento `meta` y toda la información que puede contener en varias líneas, utilizados luego por buscadores o indexadores de información), o los documen-

tos relacionados a partir de líneas del elemento `link`, como, por ejemplo, una hoja de estilo.

El cuerpo del documento *HTML* es el elemento `body`. Casi todo está visible, aunque también se puede escribir código que no se va a mostrar, como el *CSS*, *JavaScript* o los comentarios.

Los elementos disponen de tres partes:

- Nombre de elemento
- Nombre de atributo
- Contenido de atributo

En el ejemplo anterior del elemento `img`, el nombre del elemento es `img`, el nombre del atributo es `src` (*source*, origen) y el contenido del atributo es `ruta-imagen.png`.

Cada elemento dispone de algunos atributos y valores válidos. Algunos elementos, tales como `br`, no tienen contenido ni llevan una etiqueta de cierre en esa versión de *HTML*, se incluyó en *XHTML* y en *HTML 4.01*

La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual que otorga el valor y escritos en la etiqueta de comienzo de un elemento, después del nombre de éste.

El formato de un nombre de hipertexto consiste en el nombre del esquema que utiliza, luego un nombre en un formato particular, por último un identificador de enlace opcional:

```
esquema : // equipo.dominio:puerto / ruta / ruta # ancla
```

Donde:

- Equipo (*host*) es el nombre del ordenador. En una red interna, se solía utilizar el nombre del ordenador como subdominio del dominio principal.
- Dominio (*domain*) es el nombre del dominio.
- Puerto (*port*) es el puerto que utiliza. La comunicación *HTTP* sobre *TCP/IP* utiliza por defecto el puerto 80 [106] pero si se quisiera utilizar otro puerto, se puede explicitar.
- Ruta (*path*) es la ruta, el directorio donde se encuentra el documento. Cada directorio se separa con la barra /.
- Ancla (*anchor*) el ancla que hemos realizado en alguna palabra o frase.

En el primer documento sobre hipertexto se definió la sintaxis del elemento A de HTML en forma de descripción *BNF*, que utiliza:

- Líneas verticales | para indicar alternativas
- Corchetes [] para indicar partes opcionales
- Los espacios son representacionales

Por entonces se definían las direcciones en función de los protocolos, ya que no solo se utilizaba `http` sino también `doc`, `file`, `news`, `wais`, etc. El ejemplo con dirección `http` y sintaxis *BNF*:

```
http:// puerto del host [/ruta] [?búsqueda]
```

### 6.3. Marcas

Históricamente, el término marcado se ha usado para describir anotaciones u otro tipo de marcas en un texto de cara a enseñar al corrector del texto las correcciones que había realizado y cómo debía transformarse, o al linotipista cómo debería imprimirse una palabra, frase, párrafo, una corrección de estilo o de contenido. En los textos electrónicos, las marcas son una forma de explicitar una interpretación del texto.

Hay marcas que nos dicen qué estructura tienen, otras nos dicen qué estilo, otras indican notas al pie de página... Las marcas también codifican el texto del documento y por tanto se deben definir marcas aceptadas, marcas necesarias, marcas opcionales, cómo se diferencian del texto, qué significado tienen, que atributos y valores aceptan...

Los lenguajes de marcado electrónico se encuadran en tres categorías:

1. Marcado de presentación
2. Marcado procesal
3. Marcado descriptivo

El marcado de presentación es el que usan los sistemas de procesado de texto tradicionales, código embebido en el documento que produce un efecto visual. Este marcado suele estar oculto para los usuarios. Así, cuando en un editor de texto como *LibreOffice* o *Microsoft Word* se selecciona una palabra y se pincha en el botón de *texto en negrita*, el texto se pone en negrita. Se aplica el concepto *wysiwyg*, acrónimo de *what you see is what you get*, lo que ves es lo que obtienes, y no vemos ningún código del marcado de presentación que demuestre que ese texto está en negrita, solo la apariencia visual.

El marcado procesal está embebido en el texto y ofrece instrucciones para los programas

que deben procesar el texto, y siguen la metáfora *wysiwym*, acrónimo de *what you see is what you mean*, lo que ves es lo que significa. Entre ellos se cuentan el lenguaje *LaTeX* o *PostScript* (el lenguaje que entienden las impresoras), y se produce una diferencia total entre el texto –el contenido– y la presentación. Cuando se quiere generar una *salida* (*output*) del documento, el procesador leerá el documento de principio a fin siguiendo las instrucciones que se encuentre y se obtendrá el resultado esperado. Este tipo de lenguajes suele incluir partes de programación, como macros o subrutinas. Otra de sus características es que no hay que preocuparse del estilo o del aspecto del documento hasta que se elija la salida, es decir, lo importante es contar con una buena estructuración del documento.

El marcado descriptivo se usa más para etiquetar partes del documento que para ofrecer instrucciones específicas sobre cómo debería procesarse. El objetivo es identificar la estructura inherente de los documentos, por lo que se le conoce como marcado semántico. *HTML*, *SGML* y *XML* son los más conocidos y utilizados, aunque en los últimos años también ha cobrado importancia formatos como *Markdown* de marcas simples.

En *HTML*, un lenguaje de marcado descriptivo, encontramos a su vez tres tipos de marcado:

1. Marcado estructural o semántico
2. Marcado de presentación o de estilo.
3. Marcado hipertextual

El marcado estructural describe el propósito del texto, tiene una naturaleza semántica. Por ejemplo, `<h2>Semiótica</h2>` establece *Semiótica* como un encabezamiento de segundo nivel. No define cómo se verá el elemento pero sí que cualquier navegador interpretará un h2 con un tamaño de texto menor al del h1 y mayor que los que tiene

por debajo, <h3>, <h4>, etc. Si *semiótica* estuviera con el elemento `strong`:

```
<strong>Semiótica</strong>
```

Enfatiza la palabra semiótica y por defecto cualquier navegador la mostrará en negrita, aunque el aspecto no es lo importante, sino la marca. El aspecto siempre se puede modificar por *CSS*. Otro ejemplo es el elemento `blockquote` que se utiliza para una cita en bloque:

```
<blockquote>Cualquier persona que pone en una página web una etiqueta diciendo “es
```

El marcado de presentación ya está en desuso por la separación que se produce en un documento web entre el contenido, escrito en *HTML*, y su presentación, escrita en el lenguaje de presentación de contenidos *CSS* (*Cascading Style Sheets*, hojas de estilo en cascada). Uno de los pocos medios donde se mantiene este marcado es el de los boletines de noticias o *newsletters*, debido al pobre y diferente soporte de *HTML* y *CSS* que realizan algunos clientes de correo.

Por último, el marcado hipertextual se refiere a los elementos que enlazan partes del documento con otras partes o con otros documentos, como son los elementos `a`, `link` o `img`, por ejemplo.

## 6.4. HTML5

*HTML5*, como su nombre indica, es la quinta versión de *HTML*, publicada como recomendación *W3C* en octubre de 2014. *HTML5* especifica dos variantes de sintaxis para *HTML* además de una evolución de *DOM* (*Document Object Model*, modelo de objetos del documento), integrada en *HTML5*:

- *HTML* (*text/html*) o *HTML5*
- *XHTML*, sintaxis *XHTML5*, con sintaxis *XML* (*application/xhtml+xml*, *application/xml*)

*HTML5* hereda cierta sintaxis de *XHTML* y se amplía con elementos nuevos que producen mejor marcado sintáctico, lo que intenta aportar más semántica al documento y a sus partes, que explotan las posibilidades de la Web Semántica. Pero también introduce algo fundamental, *API* asociadas para trabajar con aplicaciones o tipos de datos.

El caso de *HTML5* ha sido un punto de inflexión en la historia de la Web y de los desarrollos coordinados por el *W3C*. El desarrollo de la siguiente versión de *HTML*, *XHTML2*, lo llevaba el grupo de trabajo *Grupo de trabajo de HTML* (*HTML Working Group*) [240], quienes se ocupan también de otras recomendaciones y publicaciones [241]. Un grupo de personas de este grupo, miembros de *Apple*, *Mozilla Foundation* y *Opera*, decidieron montar en 2004 el proyecto alternativo al *W3C WHATWG* (*Web Hypertext Application Technology Working Group*, motivados por la lentitud en el desarrollo de *XHTML 2*, la preocupación por la falta de compatibilidad con las recomendaciones anteriores y la despreocupación por las necesidades de los autores de contenido web [246]. *WHATWG* también se puede leer como *what wg, ¿qué grupo de trabajo?*. Desarrollaron tres líneas de trabajo de estándares web:

- *HTML*, lo que incluye *Web Workers*, *Web Storage*, *Web Sockets API*, *Server-Sent Events*, *Microdata* y *Canvas 2D*.
- *DOM*, incluidos los eventos *DOM*, rango *DOM* y *mutation observers*.
- *URL*, incluida una *API* para URL.
- Otras especificaciones [247]

En 2006 el *W3C* comenzó a reconocer que mover *HTML* a *XHTML* no había sido una gran idea y finalmente en 2009 anunciaron que dejaban de desarrollar *XHTML2* para centrar los esfuerzos en *HTML5*.

El borrador de trabajo del *W3C* en *HTML5* se tituló *HTML5 A vocabulary and associated APIs for HTML and XHTML* (*HTML5*, un vocabulario y *API* asociadas para *HTML* y *XHTML*) [133], y así se mantiene. Sin embargo, la tensión entre ambos grupos de desarrollo no desapareció, y pese a la publicación de la recomendación por el *W3C* en octubre de 2014, ambos documentos pueden diferir y en caso de contradicción, para *Mozilla*, prevalece documento de *WHATWG* [13].

*WHATWG* ha cambiado el modelo de desarrollo y fruto de ello, el nombre de su documentación ha pasado de *HTML5* a *HTML*, añadiendo a continuación *living standard* (estándar activo), lo que indica que no va a tener versiones sino que el documento irá evolucionando permanentemente [248] hasta 2022, año de su nuevo lanzamiento. *W3C* sigue con su modelo de versiones, y el borrador de *HTML 5.1* se lanzó el 23 de marzo de 2015 [241].

## 6.5. Documento básico

El documento básico de *HTML* consta de la definición del tipo de documento, el elemento `html` que contiene todos los demás, el elemento cabecera `head` que contiene el título *title*, y el cuerpo del documento, el elemento *body*.

```
<!doctype html>
<html>
  <head>
    <title>Documento de prueba</title>
```

```
</head>
<body>
<p>Hola!</p>
</body>
</html>
```

- Tipo de documento con `<!doctype html>`
- Elemento `html` para indicar que comienza el código HTML
- Elemento `head` para los elementos de cabecera
- Elemento `title`, uno de los elementos de cabecera, donde se sitúa el título del documento.
- Elemento `body` para los elementos del cuerpo del documento.
- Elemento `p` para indicar un párrafo.

Los elementos son la estructura básica de *HTML*, y pueden incluir atributos con estructura de pares nombre/valor. Se sitúan en la etiqueta de inicio, a continuación del nombre del elemento. En algunos elementos, los atributos son imprescindibles para funcionar, como ocurre con el elemento `img` de imagen, en cuyo atributo `src` se ubica la ruta a la imagen:

```

```

Otros elementos imprescindibles en la cabecera son:

- `link` para vincular hojas de estilo, iconos o relaciones entre documentos

- `style` para incluir CSS en el documento
- `meta` para incluir todos los metadatos.

Y en el cuerpo:

- Elementos de `h1` a `h6`, encabezamientos de mayor a menor importancia. El `h1` era, antes de *HTML5*, uno de los elementos más importantes de un documento web y debía ser único. A partir de *HTML5*, se puede tener un `h1` por cada elemento `section` que se incluya en el documento.
- `a` para los enlaces, con los atributos:
  - `href` de *hypertext reference* o referencia de hipertexto, cuyo valor es la URL.
  - `title` de título, para indicar otra información sobre el enlace
- `div`, de *division* o parte del documento, se utiliza para agrupar contenido. En *HTML5* buena parte de la división del documento se puede identificar con los nuevos elementos semánticos `header`, `nav`, `footer` o `section`.
- `img`, de *images*, para imágenes, donde destacan los atributos:
  - `src`, de *source*, fuente, origen o ruta de la imagen.
  - `alt`, de *alternative* o texto alternativo, para ofrecer información textual sobre la imagen.
- `ul`, de *unordered list* o lista desordenada, para una lista de elementos sin orden explícito.

- `ol`, de *ordered list*, lista ordenada.
- `li`, de *list item*, elemento de la lista, sea ordenada, desordenada o un menú.

## 6.6. Caracteres

La codificación de caracteres ha sido algo fundamental en la Web que hunde sus raíces también en la historia de la informática. El contenido electrónico está codificado de una manera determinada. En el intercambio de documentos y de datos la codificación de caracteres correcta marca la diferencia entre el éxito y el fracaso.

Un carácter es una letra con tilde como la é, pero también ideogramas chinos como o caracteres devanagari como , el espacio, la coma o el salto de línea. La codificación de caracteres es la llave que mapea puntos de código o *code points* a *bytes* en la memoria del ordenador, y hace el camino inverso para leer del ordenador [235].

Los caracteres se agrupan en conjuntos de caracteres, *character sets*, *charsets* o *repertorios*. Un conjunto de caracteres codificado asigna cada carácter a un *punto de código* que se representan en la memoria con uno o más *bytes*. Si no compartimos la codificación de caracteres, los datos no se muestran, no se pueden compartir, leer o escribir.

*HTML* empezó con *ASCII* (*American Standard Code for Information Interchange*, código estándar americano para el intercambio de información) dado que era el lenguaje de los ordenadores, por lo que para representar otros caracteres como los signos de puntuación, letras con tilde, diéresis o símbolos como el de la ñ, es decir, todos aquellos símbolos que no están en la codificación de *ASCII*, había que utilizar entidades *HTML* (*HTML entities*). Desde *HTML 2.0* hasta *HTML 4.01*, el conjunto de caracteres por defecto ha sido *ISO-8859-1* [195], que incluía más caracteres latinos, mientras que con

*XML* y *HTML5* lo ideal es utilizar *UTF-8* (*8-bit Unicode Transformation Format*, formato de transformación Unicode de 8 bits) [244], la codificación universal.

Los códigos *ISO-8859-1* que van de 0 a 127 se pueden nombrar por su código *ASCII* [230] mientras que en el grupo superior, los códigos 160 a 255, se puede utilizar el nombre del carácter [229]. Anteriormente, para escribir símbolos *ISO-8859-1*, caracteres *ISO-8859-1* y otros caracteres especiales en *HTML*, había que escribir *entidades HTML*. Al utilizar caracteres normales, el comienzo de la entidad se indica con un símbolo de inicio o de escape, en este caso el ampersand &, seguido del carácter origen de la entidad, el código correspondiente y finalmente un punto y coma ; que marcaba el final de la entidad.

En el caso de las palabras con tilde, la expresión era *acute* o *grave* en los acentos=. También se podían escribir los caracteres en código decimal o en hexadecimal.

Por ejemplo, la letra Á:

- Entidad HTML: &Aacute;
- Código decimal: &#193;
- Código hexadecimal: &#xc0;

*ASCII* nació en 1963 como *código estadounidense estándar para el intercambio de información*, un código de caracteres basado en el alfabeto latino. Utiliza 7 bits para representar 128 valores (dos elevado a siete), donde los 32 primeros (0-31) y el último (127) están reservados para caracteres de control, como los que utilizaban impresoras, tarjetas perforadas o cintas de grabación. Lo crearon en la *ASA* (*American Standard Association* o *comité estadounidense de estándares*) (llamado *USASI* (*United States of America Standards Institute*, *Instituto Estadounidense de Estándares* desde 1969 y finalmente

*ANSI* (*American National Standards Institute, Instituto Estadounidense de Estándares Nacionales*) [252]). La *IANA* (*Internet Assigned Numbers Authority, autoridad para números asignados en Internet*) también lo conoce como *US-ASCII* [112].

Los caracteres especiales se utilizan igualmente en las direcciones *URL* [231], ya que solo pueden contener caracteres de código *ASCII*. En este caso el escape se realiza comenzando con el símbolo de porcentaje % seguido del valor hexadecimal de dos dígitos. El espacio en blanco se sustituye por el símbolo de más +.

*ASCII* permitió que distintos ordenadores pudieran comunicarse antes que la Web en tanto que cada empresa, entidad o persona que desarrollara software podía usar su propia aplicación para representar letras y números del alfabeto y códigos de control[57]. El escenario de Tim Berners-Lee funcionaba en *ASCII*, su desafío era comunicar personas, poner en común documentación, ordenadores. El de *ASCII* fue un paso previo, necesario e imprescindible para que la informática se desarrollara. Solo en 1960 *IBM* utilizaba nueve conjuntos diferentes de caracteres, y pese a *ASCII* no pudieron incluirlo en su *System/360* lanzado en 1964, por lo que el único ordenador con *ASCII* era el *Univac 1050* y no fue hasta 1981 con el primer *PC*, el modelo *5150*, cuando *IBM* utilizó *ASCII*. El *PC IBM 5150* además contaba con una extensión llamada *code page 437* que reemplazaba algunos caracteres de control y añadía 128 caracteres más: nuevo símbolos y letras latinas de otros idiomas como el español, pasando de 128 caracteres a 255.

El juego de caracteres más utilizado de Internet [112] dejó de reinar en exclusividad en diciembre de 2007 [87]. Por entonces *Google* ya utilizaba *Unicode* para su sistema de búsqueda pero ahora también soportaría *Unicode*, lo cual permite manejar textos en francés, chino, árabe, ruso o cualquier otro idioma con un conjunto de caracteres distinto a *ASCII*.

La Web se amplía con *Unicode*, donde caben más de un millón de caracteres. Lanzado por primera vez en 1991, pretendía resolver los conflictos de la codificación de caracteres

electrónicos y hacerlo de una manera universal y estándar, como el escenario de la Web demandaba [6]. Ser el conjunto de caracteres de la Web también lo convierte en la lengua franca entre aplicaciones distintas en distintos idiomas. Un año y medio después de su lanzamiento, el uso de *UTF-8* se acercaba al cincuenta por ciento [50]

Aunque haya sido una de las últimas tecnologías en adaptarse, contemporánea de *HTML5*, *UTF-8* lo desarrollaron en los años 1980 Ken Thompson y Rob Pike cuando trabajaban en los laboratorios *Bell* en el sistema operativo *Plan 9*, con la ayuda de Dennis Richie, con quien habían colaborado en la creación del sistema operativo *Unix*, el cual también cambió la forma de trabajar de los ordenadores.

*UTF-8* es un formato de codificación de caracteres *Unicode*, estándar *ISO 10646* y *RFC 3629 (IETF)*. Representa cualquier carácter *Unicode*, incluidos los de *US-ASCII* de 7 bits, y cuenta con otras características técnicas de interés [268]:

- Los símbolos tienen longitud variable, de 1 a 4 bytes por carácter *Unicode*.
- Incluye sincronía, por lo que se puede determinar el inicio de cada símbolo sin reiniciar la lectura.
- No se superpone, ya que los conjuntos de valores que puede tomar cada *byte* de un carácter *multibyte* son distintos.

Divide los caracteres en grupos según el número de *bytes* necesarios para codificarlos:

- 1 *byte*, los caracteres de *US-ASCII*, 128 en total.
- 2 *bytes*, 1920 caracteres de lenguas romances y de los alfabetos griego, cirílico, copto, armenio, hebreo, árabe.
- 3 *bytes*, considerados los caracteres del plano básico multilingüe de *Unicode*, in-

cluyen la práctica totalidad de los idiomas como el chino, el japonés o el coreano.

- 4 *bytes*, caracteres del plano suplementario multilingüe, símbolos matemáticos y alfabetos clásicos.

*UTF-8* ya se recomendaba su uso en correos electrónicos desde 1999 [139], identificando el contenido con el `content-type` en *UTF-8*:

```
Content-type: text/plain; charset="utf-8"
```

En un documento *HTML* se declara la codificación del documento a través del elemento `meta` y su atributo `charset` en la cabecera del documento, inmediatamente después del elemento `head`, en el primer sitio que puede aparecer de hecho, para que se incluya entre los primeros 1024 *bytes* de comienzo del archivo:

```
<!doctype html>
<html lang="es">
<head>
<meta charset="utf-8">
...
```

## 6.7. Nuevos elementos

En *HTML* faltaban muchos elementos para acomodar el contenido a los documentos web de una forma más semántica. Se utilizaba mucho, por ejemplo, del elemento `div` (división) que acomodaba el contenido en secciones, para identificarlo a través del atributo `id`. Ahora un conjunto de elementos permiten identificar el contenido de forma universal. Además soporta formatos de audio, vídeo y vectoriales y mejora la identifica-

ción de los distintos elementos que suelen aparecer en los formularios.

### 6.7.1. Audio

La inclusión de audio se hacía complicada en *HTML*, pues había importar un reproductor con *JS* o *Flash*, no trabajaba de forma nativa con audio. Ahora cuenta con el elemento `audio`, donde la ruta al archivo similar a las imágenes:

- Con el atributo `src` del elemento `audio`:

```
<audio id="audioTestElem" autobuffer controls >  
<source src="prueba.m4a">  
<source src="prueba.ogg" type="audio/ogg; codecs=vorbis">  
<source src="url">  
no escuchas audio  
</audio>
```

- Con el atributo `src` en el elemento `audio` y opcionalmente con el elemento `track` (pista):

```
<audio src="prueba.ogg">  
<track kind="captions" src="prueba.es.vtt" srclang="es" label="Español">  
</audio>
```

### 6.7.2. Canvas

El elemento `canvas` se utiliza para mostrar gráficos, hacer composiciones de fotografías o realizar animaciones, a través de *scripting* normalmente de *JavaScript*.

### 6.7.3. Vídeo

El elemento video resuelve como audio una deficiencia de *HTML*, permite embeber contenido audiovisual con un atributo `src` para incluir las rutas a los archivos.

```
<video src="prueba.ogg" autoplay poster="imagen-portada.jpg">  
<track kind="subtitles" src="prueba.es.vtt" srclang="es" label="Español">  
</video>
```

### 6.7.4. Formulario

Los formularios son uno de los elementos que más necesitados de mejoras e implementaciones estaban antes de *HTML5*. Una de las principales novedades está en el atributo `type` de `input`, el elemento que especifica una casilla de un formulario donde los usuarios han de introducir los datos, que amplía sus posibles valores. Ahora los valores del atributo `type` explican más qué tipo de dato se introduce en cada elemento del formulario:

- `datetime`, `=datetime-local`, `date`, `month`, `week` y `time` para fechas y horas.
- `email`, para indicar el correo electrónico.
- `color`, para un color.
- `range`, para un rango entre dos números.
- `url`, para una *URL*.

- `search`, para una búsqueda.
- `tel`, para un teléfono.
- `pattern`, permite contrastar el contenido del texto introducido con una expresión regular.

Por ejemplo:

```
<form>
<input name="form_numero" id="form_numero" type="number" min="1" max="10" >
<input name="form_fecha" id="form_fecha" type="date">
<input name="form_mes" id="form_mes" type="month">
<input name="form_semana" id="form_semana" type="week">
<input name="form_hora" id="form_hora" type="time">
<input name="form_url" id="form_url" type="url" list="url_list">

<datalist id="url_list">
<option value="https://www.startpage.com" label="Startpage">
<option value="https://www.duckduckgo.com" label="Duck">
</datalist>

<input name="form_email" id="form_email" type="email" list="email_list" multiple>

  <datalist id="email_list">
    <option value="fulano@prueba.com" label="fulano">
    <option value="mengana@prueba.com" label="mengana">
  </datalist>

  <input name="form_telefono" id="form_telefono" type="tel">
```

```
<input name="form_color" id="form_color" type="color">
</label>
Attachments:
<input type="file" multiple name="att">
</label>
<input name="x" type="range" min="100" max="700" step="10" value="500">
</form>
```

Además, se mantienen los anteriores valores posibles:

- `text`, para cadenas de texto.
- `password`, para contraseñas.
- `submit`, para validar el formulario.
- `radio`, para indicar una casilla circular que marcar, entre varias opciones
- `checkbox`, para un requisito que se cumple
- `button`, para un botón.

También ha incluido más elementos del formulario:

- `meter`, para medidas y escalas.
- `progress`, para barras de progreso.
- `datalist`, para campos con funcionalidad de autocompletar.
- `keygen`, para generar claves pública y privada para encriptación.

- `output`, para realizar y mostrar cálculos matemáticos.

### 6.7.5. Elementos semánticos

Los elementos más ansiados por quienes diseñan y organizan el contenido de un documento *HTML*. Ahora los contenidos tienen una identificación semántica mayor, que indica tanto su posición jerárquica en el documento como una posible ubicación espacial.

- `nav`, de *navigation* o *navegación*, representa una sección del documento que contiene enlaces de navegación del documento, puede enlazar con partes del propio documento o de otros documentos.
- `footer`, *pie*, indica el pie del documento *HTML*, suele contener información sobre quién ha escrito el documento, enlaces a otros documentos –pero no los que deban estar en `nav`–, información del *copyright*, etc. No tienen que estar al final de la sección aunque es lo normal. También puede contener apéndices, índices, acuerdos de uso del documento, glosarios...
- `section`, sirve para dividir el documento en secciones, lo cual antes había que hacer con `div`, siendo este elemento utilizado para cualquier tipo de división del documento. Sin embargo, una sección del documento permite que un solo documento *HTML* organice mejor su contenido, siguiendo el estilo de *SGML*. De hecho, esto provoca un cambio fundamental en el uso de `h1` ya que si antes solo podía haber un `h1` en todo el documento, ahora cada `section` puede contar con un `h1`. En los documentos escritos las secciones son una parte de una aplicación web, cualquier agrupación temática de un contenido, capítulos de un artículo o un libro o las partes de un capítulo. Esto implica que donde antes había distintas páginas o documentos web en un sitio web, ahora podría haber un solo do-

cumento web con distintas secciones: introducción, metodología, capítulo uno, conclusiones, etc.

- `article`, *artículo*, agrupa un contenido dentro de un documento web. Puede contener o ser parte de `section`. Al igual que con `section`, podría ser un documento web en sí mismo, pero su inclusión dentro del mismo documento permite una relación directa de todo el contenido del documento. Puede tratarse de un mensaje de un foro, un artículo de un blog, un comentario de un usuario, una parte de una aplicación u otro contenido tratado de forma independiente. Al igual que en `section`, debe contar con un encabezamiento, uno de los elementos `h1` a `h6`
- `aside`, *a un lado* o *aparte*, elemento que representa una sección de una página que consta de contenido que se sitúa a un lado de otro contenido. Por ejemplo, se puede utilizar para un menú, anuncios publicitarios, contenido relacionado, biografías, aplicaciones web, información del perfil o para agrupar contenido de cualquier otro tipo. Lo podría utilizar la *Wikipedia* para mostrar los *infobox*, pero utiliza una tabla `table`.
- `hgroup`, elemento que agrupa encabezamientos. No figura en la especificación sobre *HTML5* de *W3C* pero se mantiene en la versión de *HTML* de *WHATWG* [167].
- `header` o cabecera, se utiliza para situar un grupo de enlaces que ayude a la navegación o que introduzcan el contenido del documento. Puede contener también un logo, un formulario de búsqueda u otros elementos informativos que se suelen situar en la cabecera de la web.
- `time`, *hora*, sirve para mostrar tanto la hora como una fecha precisa del calendario gregoriano, y también información sobre la zona horaria.

- `mark`, *marca*, elemento que se utiliza para resaltar una parte del texto, como, por ejemplo, resultados de búsqueda, una parte de una cita que se quiere resaltar. Es fácil confundir su uso con `strong`, `span` o `em`.

### 6.7.6. Elementos obsoletos

Algunos elementos de *HTML 4.01* que han quedado obsoletos ya se habían dejado de utilizar al estar dedicado *HTML* para la estructura del contenido y *CSS* para la presentación. Es el caso de *font* y *center*.

## 6.8. Lenguaje de marcado SGML

*SGML* es la norma *ISO 8879:1986* que sirve para definir modos de representar el texto de forma electrónica independientes del dispositivo y del sistema operativo. Se trata de un metalenguaje que permite crear tu propia forma de representar el contenido a partir de la definición de la sintaxis en la *DTD* o *Document Type Definition*, definición de tipo de documento. Una de las *DTD* más conocidas es *HTML*, pero también *DocBook* se utiliza para documentación de software y en ámbitos científicos.

Se considera que la primera vez que se habló de separación de información contenida en un documento de su presentación fue en 1967 cuando William Tunnicliffe, presidente del comité de composición (*Composition Committee*) de la *Graphic Communications Association* (asociación de comunicaciones gráficas), se reunió con la oficina de impresión del gobierno de Canadá [119]. También a finales de los 1960', el diseñador de libros Stanley Rice propuso contar con un catálogo universal de etiquetas editoriales parametrizadas para facilitar la revisión de los libros antes de publicarlos [123]. Norman Scharpf, director de *GCA*, creó un proyecto que desarrolló el concepto *GenCode(R)*: diferentes

códigos genéricos para distintos tipos de documentos.

En 1969 Charles Goldfarb desarrolla *GML* (*Generalized Markup Language*, lenguaje de marcas generalizadas) en *IBM*, que consiste en marcar un documento con etiquetas que definen lo que es el texto en cada párrafo, encabezamiento, lista, tabla, etc. [120], y luego se puede dar formato para el dispositivo de salida elegida, distinguiendo así entre una impresora láser o una pantalla de ordenador, sin tener que rehacer el documento. Al igual que en *GML*, *SGML* permite que la estructura de un documento pueda ser definida en base a la relación lógica de sus partes, documentada en la *DTD*.

La norma *SGML* define la sintaxis del documento y la sintaxis y semántica de *DTD*. De *SGML* derivan el conjunto de lenguajes *XML*. Tanto *SGML* como *XML* son metalenguajes que se utilizan para definir lenguajes de marcas con vocabularios específicos, etiquetas para elementos y atributos, una sintaxis declarada y una gramática que define la jerarquía y otras características.

## 6.9. DTD, definiciones de tipos de documentos

*DTD* es un documento que describe la estructura y sintaxis de un documento *XML* o *SGML*. Si *XML* y *SGML* son metalenguajes que definen reglas, atributos y relaciones de las mismas, *DTD* es una de las formas de escribir estas reglas. *HTML* está basado en *SGML*, es una *DTD* de *SGML*. La descripción del formato de datos busca un formato común y mantener la consistencia entre todos los documentos que la utilicen y pueden validarse. De esta forma se pueden leer, escribir, reutilizar, verificar o intercambiar datos.

Las reglas se refieren a qué elementos y atributos se declaran y qué contenido se permite, qué orden puede llevar y el anidamiento, elementos que pueden contener a otros. La *DTD* se puede incluir en el propio archivo *XML* o se puede referenciar externamente.

Si el *DTD* se declara dentro de un documento *XML*, debería estar con la estructura:

```
<!DOCTYPE elemento-raiz [declaraciones-de-elementos]>
```

El contenido de los elementos puede ser:

- **ELEMENT**, si solo puede contener otros elementos, en ese caso separados por espacios en blanco.
- **TEXT**, si solo contiene texto
- **EMPTY**, vacío, no puede contener otros elementos.
- **ANY**, si tiene elementos hijos.
- **MIXED**, si puede contener texto u otros elementos
- Modelo de grupo, elementos hijos, donde la secuencia de elementos se define en:
  - Una lista, con elementos en secuencia, en un orden determinado, separados por comas =,= o bien con operadores de elección |.
  - Cuantificadores: al estilo *BNF*, ? si es 0 o 1 vez; +, si es 1 o más veces; \*, si es 0 o más veces.

Por ejemplo:

```
<?xml version="1.0" encoding='utf-8' standalone='yes'?>
<!DOCTYPE nota [
<!ELEMENT nota (para+, (de|desde), encabezamiento, cuerpo, comentario?, br*)>
<!ELEMENT para (#PCDATA)>
```

```
<!ELEMENT de (#PCDATA)>
<!ELEMENT desde (#PCDATA)>
<!ELEMENT encabezamiento (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
<!ELEMENT comentario ANY>
<!ELEMENT br EMPTY>
]>
<nota>
<para>Adolfo</para>
<br />
<de>Gonzalo</de>
<encabezamiento>Recordatorio</encabezamiento>
<cuerpo>Trabajo de Tesis</cuerpo>
</nota>
```

Donde:

- Como es un documento *XML*, en la primera línea del prólogo del documento se puede declarar en el prólogo del documento, donde se especifica que se utiliza la versión 1.0, que la codificación de caracteres es *Unicode UTF-8* y que se permiten entidades externas, es decir, entidades provenientes de otras *dtd* que habría que declarar mediante ENTITY.
- Si esta primera línea no se declara, se utilizarán estos mismos valores por defecto.
- !DOCTYPE define que el elemento *root* (raíz) de este documento es nota.
- !ELEMENT nota especifica que el elemento nota contiene los elementos: para, de, desde, encabezamiento, cuerpo, comentario y br.

- El elemento `para`, al ir seguido de un `+`, indica que puede aparecer más de una vez.
- Los elementos `de` y `desde` se pueden poner uno u otro, ya que van separados por condicionales opcionales.
- El elemento `comentario`, al ir seguido de un `?`, indica que puede aparecer cero o una veces.
- El elemento `br`, al ir seguido de un `*`, indica que puede aparecer cero o más veces.
- `!ELEMENT` seguido de cada uno de esos cuatro elementos, se declaran de tipo `#PCDATA`, que son *Parsed Character Data*, datos de caracteres que se analizan según la *DTD* y por tanto siguen las restricciones que llevan los *DTD*.
- El elemento `comentario` puede tener cualquier combinación de datos analizables.
- El elemento `br`, al ser `EMPTY`, se contiene a sí mismo.

El nombre del elemento tiene unas reglas definidas:

- El primer carácter puede ser una letra, un guión bajo `_` y dos puntos `:`
- El resto de caracteres puede contener, además de los anteriores, dígitos, guiones `-` y puntos `.`

Para enlazar con una declaración externa, se utiliza el atributo `SYSTEM` con la ruta como valor:

```
<!DOCTYPE elemento-raiz SYSTEM "ruta-archivo-dtd">
```

El mismo documento *XML* de antes puede utilizar la *DTD* externa:

```
<?xml version="1.0"? encoding="utf-8" standalone="yes">
<!DOCTYPE nota SYSTEM "nota.dtd">
<nota>
<para>Adolfo</para>
<de>Gonzalo</de>
<encabezamiento>Recordatorio</encabezamiento>
<cuerpo>La presentación de la tesina es mañana</cuerpo>
</nota>
```

El archivo `nota.dtd` contendrá la misma información que antes estaba escrita en el interior de los corchetes de la expresión `<!DOCTYPE NOTA []>`:

```
<!ELEMENT nota (para,de,encabezamiento,cuerpo)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT encabezamiento (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
```

La declaración del *DOCTYPE* puede contener un *FPI* (*Formal Public Identifier*, identificador formal público) y la localización de *DTD* que describe el vocabulario. La localización es imprescindible si se utiliza el identificador `PUBLIC`, pero no es necesario si se utiliza `SYSTEM`, ya que indica un ámbito local. En este ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/str
```

El *FPI* es:

```
"-//W3C//DTD HTML 4.01//EN"
```

Que consta de dos partes:

- El identificador de propiedad que identifica el emisor del *FPI*: `-//W3C`, donde el signo `-` indica que no es un *FPI* registrado en la *ISO*. Paradójicamente, *W3C* no lo es.
- El identificador de texto que indica el documento u objeto que identifica el *FPI*. En este caso, `DTD HTML 4.01//EN`, que consta de tres partes, clase, descripción e idioma, donde:
  - Clase es `DTD`
  - Descripción es `HTML 4.01`
  - Idioma es `EN` de *English*. Podría tener una versión, que se incluiría después del idioma, separado por barras dobles `//`

Para incluir atributos en la *DTD* =, *utilizamos la orden ATTLIST, por /attributes list* o lista de atributos, con la estructura:

```
<! ATTLIST elemento
atributo1 contenido1 valor-por-defecto1
atributo2 contenido2 valor-por-defecto2 >
```

Los valores pueden ser de varios tipos:

- `PCDATA`, cadenas de caracteres.
- `CDATA`, por *Characters Data*, cadenas de caracteres que no se parsean.
- `NMTOKEN`, cadenas de caracteres pero compuesto por caracteres de la *a* a la *z*, ma-

yúsculas o minúsculas ([a-z], [A-Z]), números ([0-9]) y guiones bajos (\_), puntos (.) o guiones (-).

- NMTOKENS, lista de NMTOKEN, separados por espacios en blanco.
- ENTITY, define valores de reemplazo.
- ENTITIES, lista de ENTITY, separadas por espacios en blanco.
- ID, valor único, identificador separada por espacios en blanco.
- IDREFS, lista de ID, separados por espacios en blanco.
- NOTATION, se ajusta a una notación declarada previamente.
- (en1 | en2 | . . .), uno de los valores de la lista.

Los atributos pueden ser:

- #IMPLIED, si es opcional
- #FIXED, si es de valor fijo.
- REQUIRED, si es de valor obligatorio

Podemos definir los atributos para cualquier elemento. Por ejemplo, definimos que el elemento nota tengo cuatro atributos: tipo, align, fecha-escrita y fecha:

```
<! ATTLIST nota
tipo (informativa|ejecutiva) #IMPLIED "informativa"
align (left|center|right) #FIXED "left"
```

```
fecha-escrita CDATA #IMPLIED
fecha NMTOKEN #REQUIRED>
```

Donde:

- Los valores opcionales se escriben entre paréntesis, separados por barras verticales
- #IMPLIED indica que se trata de un atributo opcional
- Si se trata de un atributo #FIXED, fijo, al final se escribe el valor por defecto que tiene el atributo.
- El valor `fecha-escrita` es de tipo CDATA, por lo que admite cadenas de caracteres.
- El valor `fecha` admite algunos caracteres

Con esos nuevos atributos, la línea del elemento `nota` quedaría así:

```
<nota tipo="ejecutiva" fecha-escrita="14 de septiembre de 2015" fecha="2015-09-14"
<!-- subelementos -->
</nota>
```

Nótese la diferencia entre los caracteres admitidos por `fecha-escrita`, de tipo CDATA, y los admitidos por `fecha`, de tipo NMTOKEN

Hay caracteres reservados, que tienen un significado especial para la DTD, por lo que si se quieren incluir en el texto, han de traducirse a sus *entity references* o referencias de entidad:

- `&lt;`, para `<`

- `&gt;`, para `>`
- `&amp;`, para `&`
- `&quot;`, para `=`
- `&apos;`, para `'`

Aunque se puede realizar declaraciones muy complejas, una de las limitaciones de las *DTD* es que no permite jugar con los espacios de nombres de *XML*, por lo que si un mismo elemento se quiere utilizar en dos subelementos, hay que modificar la sintaxis para diferenciarlos, dejando de ser el mismo elemento. Tampoco es posible indicar a qué tipo de dato (número, fecha, moneda) ha de corresponder un atributo o el texto de un elemento.

Para superar estas limitaciones se crearon lenguajes de esquema como *XML Schema* que permiten una descripción más compleja y funcionan como una alternativa a *DTD*.

## 6.10. Lenguaje de estilo de texto CSS y XSL

Para la representación visual de los datos comprendidos en los documentos *(x)HTML* o *XML* se utilizan las hojas de estilo *CSS* (*Cascading Style Sheets*, hojas de estilo en cascada) en el caso de *(x)HTML* y *XSL* (*eXtensible Style Sheet*, hoja de estilos extensible) en *XML*.

Las hojas de estilo aparecen poco después del lenguaje de etiquetas *SGML* [97], en los 1970', para definir un mecanismo que aplicara diferentes estilos a diferentes documentos electrónicos.

Del seno del *W3C* nacen dos propuestas que confluyen en *CSS* y se convierte en recomendación oficial de *W3C* en diciembre de 1996 *CSS Nivel 1* [273]. En 1997 *W3C* separa el grupo de *HTML* en tres secciones: *HTML*, *DOM* y *CSS* y en 1998 se publica la segunda recomendación oficial conocida como *CSS nivel 2* [54]. En 2009 se publicó una revisión, *CSS2.1* [55], y desde 1998 se comenzó a trabajar en *CSS3* [200].

El nombre de *CSS* viene por el funcionamiento de las hojas de estilo, ya que las declaraciones afectan a todo el contenido como si de una cascada se tratara. El estilo en una página web es lo que da aspecto a la presentación de un documento estructurado en *HTML* o *XML*. Los agentes indicados, es decir, los *navegadores*, interpretarán ese código y darán el aspecto indicado al contenido, dispondrán de diseño.

En las primeras versiones de *HTML* se permitían atributos extra dentro de una etiqueta *HTML* para darle estilo, como, por ejemplo, elegir el color o el tamaño de la fuente, pero se entendió que resultaba confuso para el propio documento de texto incluir código que no tenía que ver con el documento en sí sino con su presentación, por lo que se decidió separar la presentación del contenido totalmente.

El estilo se puede dar con diferentes métodos, algunos más *limpios* que otros, aunque se pueden combinar y no se exige un método u otro como no se exige estilo alguno:

- Hoja de estilo externa: es el método más limpio, ya que toda la información del estilo está en un archivo *css* distinto del *html*. Para modificar el estilo del documento *HTML* tan solo hay que modificar el archivo *CSS*. Cuando se trata de todo un sitio web con decenas, cientos o miles de documentos que comparten el mismo archivo *CSS*, es una ventaja sobre los otros métodos.
- Hoja de estilo interna: declaraciones de estilo incrustadas en el documento *HTML*, en el elemento `style`, en la cabecera del documento, en el interior del elemento `head`. En general, se suele usar la hoja de estilo interna cuando se quiere aportar

alguna característica de estilo específica para el documento *HTML* en cuestión.

- Estilo *en línea*, se inserta el estilo dentro del propio elemento *HTML* al que afecta, con el atributo `style`, cuyos valores son las declaraciones de *CSS*. No es lo más recomendable, de forma general, ya que convierte al documento *HTML* en algo tedioso de leer para lxs humanxs, aunque se utiliza excepcionalmente, al igual que la hoja de estilo interna, para aportar alguna característica de estilo específica o bien de ese documento *HTML* o bien de una parte del documento.

Desde que comenzara el desarrollo de *CSS3*, en 1998, *CSS2.1* ha sido la recomendación oficial de *W3C*. Dado el desarrollo modular de *CSS3*, hay módulos que son propuestas de recomendación y otros borradores de trabajo, lo cual no afecta para que se utilice plenamente *CSS3*.

Por el lado de la industria y del usuario, *CSS3* permite que las páginas web sean más dinámicas. Por el lado de quien crea, introduce claramente conceptos de programación que ya aparecían esbozados en *CSS 2.1*

En una hoja de estilos se utilizan reglas que consisten en elegir selectores a los que se asignan una serie de propiedades o atributos que tienen un valor, donde cada término significa:

- Regla: se entiende por regla cada uno de los estilos que componen una hoja *CSS*. Cada regla está compuesta de una parte de "selectores", un símbolo de llave de apertura {, otra parte denominada "declaración" por último, un símbolo de llave de cierre }.
- Selector: indica el elemento o elementos *HTML* a los que se aplica la regla *CSS*.
- Declaración: estilos que se aplican al selector y caben una o más propiedades *CSS*

separadas por punto y coma.

- Propiedad: característica CSS que modifica elemento seleccionado con su valor.
- Valor: establece el nuevo valor al selector

Por ejemplo, un archivo CSS con una regla consta de:

```
body {  
background-color: #ff0000;  
color: black;  
font-family: Arial, sans-serif;  
font-weight: bold;  
font-size: 12pt;  
}
```

Donde:

- El selector `body` se separa de las propiedades y los valores con una llave `{}`.
- Dentro de la llave cada declaración se separa de la siguiente con un punto y coma `;`
- Si hay más de un valor, se separan por comas, como en el caso de `font-family`.
- Los colores pueden ser en hexadecimal, como `#ff0000`, que es color rojo; o bien en *RGB* o como cadena de caracteres, algunos pocos, como `black` o `red`.
- En `font-family` siempre conviene poner, al final, `serif` o `sans-serif`, según la familia de fuente que hayamos elegido.

- La fuente será toda en negrita, como se declara en `font-weight: bold`

y el tamaño, 12pt. Se puede expresar en px por pixels, porcentaje %, relativo em o en puntos pt.

Para relacionar al *HTML* con ese archivo *CSS*, se escribe la ruta en el atributo `href` del elemento `link`, especificando que se trata de una relación `rel` de estilo `stylesheet`, que el tipo *MIME* es `text/css` y que es un estilo se utiliza en los medios `media` definidos, en todos `all`.

```
<link rel="stylesheet" type="text/css" href="css/principal.css" media="all" />
```

También se puede incluir el código en el documento:

```
<!DOCTYPE HTML>
<html lang="es">
<head>
<meta charset="utf-8">
<title>Mis primeros estilos CSS</title>
<style type="text/css">
  h1 { color: red; font-family: Arial; font-size: large; }
  p { color: gray; font-family: Verdana; font-size: medium; }
</style>
</head>
<body>
  <h1>El titular</h1>
  <p>Un párrafo corto</p>
</body>
</html>
```

Otra forma de enlazar los documentos CSS es con el modo `@import`, dentro del elemento `style`:

```
<style type="text/css" media="screen">
  @import '/css/principal.css';
</style>
```

El modelo de caja de CSS o *CSS box model* describe las cajas rectangulares invisibles pero presentes que son generadas por los elementos en la estructura del documento y compuestas de acuerdo al modelo de formato visual. Cada caja tiene un área de contenido y las áreas circundantes de *padding*, *border* y *margin*, que pueden dividirse en los segmentos *top*, *right*, *bottom*, y *left*, de acuerdo con la lectura que realiza el analizador de CSS, siguiendo el sentido de las agujas del reloj.

El perímetro de cada una de las cuatro áreas (contenido, *padding*, *border* y *margin*) se conoce como *límite*, de tal forma que cada caja dispone de cuatro límites:

1. Límite del contenido o límite interno, el límite del contenido rodea al contenido procesado del elemento.
2. Límite de *padding* o relleno, rodea a la caja de relleno. *Padding* es el espacio del contenido hasta el *border*.
3. Límite de *border*, rodea el borde de la caja.
4. Límite de *margin*, margen o límite externo, rodea el margen de la caja.

Cada regla de CSS consta de uno o más selectores, y un bloque de declaración. Los selectores se utilizan para declarar en qué parte del marcado se aplica un estilo por las etiquetas y atributos del marcado. Los selectores pueden aplicarse a:

- Todos los elementos de un tipo específico. Por ejemplo, los `h2`
- Elementos de un atributo en particular, identificado por un `id` o una `class`.
- Elementos que son relativos a otros elementos.

Hay dos tipos de atributos, cuyos valores son sensibles a las mayúsculas y minúsculas y pueden empezar por letras y contener también números y guiones bajos:

- `id`, identificador único en el documento, no puede repetirse.
- `class`, un identificador que agrupa múltiples elementos de un documento.

Las pseudoclasas se usan en los selectores *CSS* para permitir el formateado basado en la información que no se contiene en el árbol del documento. Un ejemplo es `:hover`, que identifica contenido solo cuando el usuario señala un elemento visible. Se suele utilizar de esta manera:

```
a:hover {}  
#id_de_elemento:hover {}
```

Una pseudoclase clasifica elementos del documento como `:link` (enlace) o `:visited` (visitado) mientras que un pseudoelemento hace una selección que se basa en elemento parciales como `:first-line` (primera línea) o `:first-letter` (primer carácter). Los selectores se pueden combinar de muchas maneras para conseguir mucha especificidad y flexibilidad.

*XSL* es una familia de lenguajes basados en *XML* que permiten la transformación o formateo de un documento *XML*. Tres lenguajes componen la familia, las tres recomendaciones del *W3C*:

- *XSLT*
- *XSL-FO*, *XSL Formatting Objects*, *XSL que da formato a objetos*, especifica cómo se va a dar formato a los datos para presentarlos en pantalla, papel u otros medios. En el documento figuran tanto los datos como el formato que se les va a aplicar.
- *XPath* o *XML Path Language*, lenguaje de rutas *XML*, es una sintaxis no basada en *XML* para acceder o referirse a porciones de un documento *XML*.

## 6.II. Diseño responsivo

El diseño *responsivo* es la traducción literal término *responsive design*, referido al diseño web que tiene en cuenta los distintos dispositivos, las distintas resoluciones, se adapta y fluye. Responsivo es un adjetivo que significa *perteneciente o relativo a la respuesta*, y anteriormente se había hablado de diseño web líquido, adaptable o adaptativo, por ser un diseño que se adapta según el dispositivo donde se va a mostrar el contenido web. En cualquier caso, la esencia sigue siendo la misma: escribir buen código, y adaptarse a distintos escenarios donde se puede mostrar ese código.

Antes de *CSS3*, para hacer un diseño adaptable, que se viera en todos los navegadores, había que hacerlo de forma modular, con un *HTML* bien construido y una hoja de estilo *CSS* que no forzara determinada disposición del contenido sino que propusiera disposiciones, de tal forma que los bloques de contenido se adaptaran al ancho propuesto por el dispositivo de salida. También contaba *CSS2* con *media-type* [55], donde se podía especificar un estilo distinto para cada dispositivo de salida, como, por ejemplo, *screen* para pantalla y *print* para impresora. Las primeras resoluciones estándar fueron de *640x480 pixels*, pasando a *800x600* y *1024x768* y de proporción cuatro tercios *4:3*. Era la resolución de la mayoría de los monitores. Luego comenzaron a aumentar a *1280*

y más, pero también, con los ordenadores portátiles, a tener distinta proporción 5:4; con los monitores panorámicos 16:9 y 16:10; y en los dispositivos móviles 480px...

W<sub>3</sub>C ha auspiciado el concepto *One Web* [193], recomendación oficial, para que se produzca en la medida de lo posible, el mismo contenido y se dispongan de los mismos servicios todos los usuarios de la Web, independientemente del dispositivo que utilicen, lo cual no quiere decir que dispongan de la misma forma de representar los contenidos, que puede modificarse en función de las capacidades de los dispositivos, el ancho de banda y las características de las redes móviles. De hecho, puede haber servicios o contenidos con un objetivo distinto según estos contextos.

Si en CSS<sub>2.1</sub> se disponía de `media-type`:

```
<link rel="stylesheet" type="text/css" href="principal.css" media="screen" />
<link rel="stylesheet" type="text/css" href="impresion.css" media="print" />
```

En CSS<sub>3</sub> se introducen `media-query` para realizar consultas al dispositivo [200] y ofrecerle un estilo u otro según sus características físicas:

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 480px)
```

La consulta comprende dos componentes:

1. El tipo de medio o `media type`, en este caso `screen`.
2. El atributo `max-device-width` de máxima anchura del dispositivo, en este caso `480px`

Si hay una pantalla con una resolución máxima de 480px, se usa la hoja de estilo `smartphone.css`. Si no, se ignora.

La consulta permite preguntar más de una propiedad de una sola vez con el operador lógico and:

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 480px)
```

Donde se añade la condición de que la resolución sea de al menos 163dpi (*dots per inch* o puntos por pulgada).

Se puede añadir el código al archivo CSS directamente, con la regla @media:

```
@media screen and (max-device-width: 480px) {  
  .column {  
    float: none;  
  }  
}
```

O como parte de la regla de importación de CSS:

```
@import url("movil.css") screen and (max-device-width: 480px);
```

---

## Capítulo 7

### Redes

Las redes de comunicación que utilizamos en el siglo XXI comenzaron a diseñarse en el XIX y se desarrollaron durante el XX, aunque una arqueología de las redes de la información y de los sistemas de comunicación remite a rutas comerciales y geopolíticas muy anteriores. En el siglo XIX se produce un punto de inflexión con las primeras redes de comunicación telegráfica, que permitían transmisión de mensajes de texto que eran enviados en lenguaje *morse* de forma unidireccional. Posteriormente, las redes telefónicas permiten la comunicación bidireccional de voz; la radio, la comunicación de uno a muchos, con sonido; y la televisión, con imágenes.

La irrupción de Internet supuso otra revolución en las redes de comunicación, no solo por la capacidad para comunicar un ordenador con otro sino también por la separación entre la capa física –la infraestructura– y la capa lógica –el software–, los protocolos de transmisión, el carácter reticular de la red, la capacidad de la transmisión para descomponerse en paquetes que toman rutas diversas que se recomponen en destino y su constitución como metamedio.

## 7.1. Internet

Internet es un conjunto de redes de comunicación interconectadas a través de la familia de protocolos *TCP/IP* (*Transmission Control Protocol/Internet Protocol*, protocolo de control de la transmisión/protocolo de Internet). Los puertos están codificados en 16 *bits*, por lo que se dispone de 65535 puertos de comunicación [263], cada uno de los cuales se puede utilizar para un servicio de comunicación determinado. Por ejemplo, la Web, utiliza por defecto el puerto 80, la recepción de correo *POP3* (*Post Office Protocol 3*, protocolo de oficina postal 3), el 11 y el envío de correo *SMTP* (*Simple Mail Transfer Protocol*, protocolo de transferencia de correo simple), el 25. La *IANA* (*Internet Assigned Numbers Authority*, agencia de asignación de números de Internet) desarrolló una aplicación estándar para ayudar con las configuraciones de red, de tal forma que dividió el uso de los puertos [81] en:

- Puertos conocidos o reservados, los que van del 0 al 1023 (1024 en total), utilizados por procesos del sistema, *daemons* o programas que ejecutan usuarios con privilegios.
- Puertos registrados, del 1024 al 49151.
- Puertos dinámicos y/o privados, del 49152 al 65535.

La historia de Internet comienza en 1962 cuando las *Fuerzas Aéreas* de EE.UU., que disponían en exclusiva del arsenal de armas atómicas, solicitan a un grupo de científicos el estudio de una red de comunicaciones que sea capaz de funcionar en el hipotético caso de que un ataque nuclear del enemigo –la URSS, Unión de Repúblicas Socialistas Soviéticas– inutilice una o varias de las bases o nodos de la red de comunicaciones. En 1964 Paul Baran elige un diseño de red en forma de tela de araña [72] (*web*) y la conmutación de paquetes [12] para transmitir los datos.

En 1969 la agencia para la investigación avanzada de proyectos, *ARPA* (*Advanced Research Projects Agency*) lanza la red *ARPANET* conectando cuatro universidades: Stanford, Universidad de California en Los Ángeles, Universidad de California en Santa Bárbara y Utah. En 1971 Ray Tomlinson escribe el primer correo electrónico, donde utiliza el carácter @ para separar al nombre del usuario del nombre del servidor, con el siguiente texto:

QWERTYUIOP

## 7.2. TCP/IP

Un año después Lawrence Roberts realiza una aplicación para leer correo y se presenta *ARPANET* en el congreso internacional de comunicaciones por ordenador (*ICCC*, *International Computer Communications Conference*) donde surge la palabra *internetting* para referirse a *ARPANET*. *ARPA* pasa a llamarse *DARPA* (la *d* es de *defense*, defensa) y Bob Kahn comienza a trabajar en *TCP* para reemplazar al protocolo *NCP* (*Network Control Protocol*, protocolo de control de la red). Con la ayuda de Vinton Cerf, en 1975 publican el *RFC 675* [74] y en 1976 *DARPA* asume *TCP* en la red *ARPANET*, constituida ya por III equipos. En 1978 *TCP* se divide en *TCP* e *IP* para conocerse a partir de entonces como protocolos *TCP/IP*.

Las dirección de cada equipo de la red relacionada con un nombre se escribía en un archivo llamado *hosts* (equipos, alojamientos), por ejemplo:

```
127.0.1.1      fantasmagoria
5.9.9.206     eiiic
```

Donde la primera línea con la dirección 127.0.1.1, que es una dirección reservada para

el *localhost* –el equipo propio–, se corresponde con el nombre del equipo *fantasmagoria*, mientras que la dirección 5 . 9 . 9 . 206 se corresponde con el equipo *iiiic*. El archivo *hosts* contendría todas las direcciones *IP* y nombres de equipos. Debido al crecimiento exponencial de los equipos conectados, resultaba inmanejable su actualización por lo que en 1984 surge un nuevo software denominado *DNS* (*Domain Name System*, sistema de nombres de dominio), que permite escalar el número de equipos conectados a Internet al traducir automáticamente entre nombres de dominio y direcciones *IP*.

### 7.3. Tipos de redes

Las redes pueden tener distintas topologías, tanto en el plano físico como en el lógico, según la forma en la que se conectan los nodos de la red. Se reconocen ocho tipos:

- Punto a punto, el enlace más simple entre dos puntos, que puede ser permanente o dedicada; conmutada, que se utiliza cuando es necesaria, o convergente, que transmite voz, audio y vídeo.
- *Bus*, lineal o distribuida, donde la comunicación se realiza sobre el canal de comunicaciones, llamado bus, troncal o backbone, al que se conectan los dispositivos.
- En estrella, donde se conectan los nodos con un nodo central, que utiliza un concentrador *hub* o *switch*.
- En anillo *ring* o circular, como la lineal pero el último nodo enlaza con el primero.
- De malla o *mesh*, puede ser una malla parcial o una red de araña *web*.
- En árbol *tree* o jerárquica, como si fueran redes de estrella concatenadas como redes *bus*.

- Híbrida o mixta, donde se combinan dos topologías básicas. La más habitual es la red circular de estrella donde una *MAU* (*multistation access unit* o unidad de acceso a multiestación) funciona como concentrador centralizado.
- *Daisy chain*, conectados en serie, bien en línea o en anillo, habiendo un nodo de inicio y uno de fin.

Aunque *TCP/IP* se ha convertido en un estándar de red, en el inicio de las redes informáticas cada fabricante contaba con su propio sistema de red, por lo que la *ISO* decidió armonizar la interconexión de sistemas abiertos a través del estándar *ISO/IEC 7498-1* [142], más conocido como *modelo OSI* (por *Open System Interconnection*), que define un marco de trabajo *framework* de la redes compuesto por siete capas o niveles [275]: aplicación, presentación, sesión, transporte, red, enlace de datos y físico.

- Físico, las características físicas de la red, la topología de red y de las conexiones del dispositivo con la red. Se ocupa tanto del medio físico como de la forma en la que se transmite la información.
- Enlace de datos, capa de enlace de datos, encargada del direccionamiento físico, el acceso al medio, la detección de errores, la distribución ordenada de las tramas y el control del flujo.
- Red, para enrutar entre una o más redes. Los dispositivos encargados son los *routers* o enrutadores.
- Transporte, se encarga del transporte de datos, los datagramas *TCP* o *UDP* (*User Datagram Protocol*, protocolo de datagrama del usuario).
- Sesión, mantiene y controla el enlace establecido.
- Presentación, representa la información, que dos equipos con distintas repre-

sentaciones internas de caracteres de datos lo reconozcan igualmente. Se ocupa de la semántica y la sintaxis. También puede cifrar y comprimir datos.

- Aplicación, ofrece acceso a los servicios de las otras capas y define protocolos para el intercambio de datos: *SMTP* para envío de correo electrónico, *FTP* para transferencia de archivos, etc. Cada aplicación puede usar uno o más protocolos.

Según el tamaño, también se distinguen redes locales o *LAN* (*local area network*, red de área local), *MAN* (*Metropolitan Area Network*, red de área metropolitana) y *WAN* (*Wide Area Network*, red de área extensa). Las redes domésticas y de entidades son redes *LAN*, las redes *MAN* son redes que cubren áreas ciudadanas y como ejemplo de redes *WAN*, Internet.

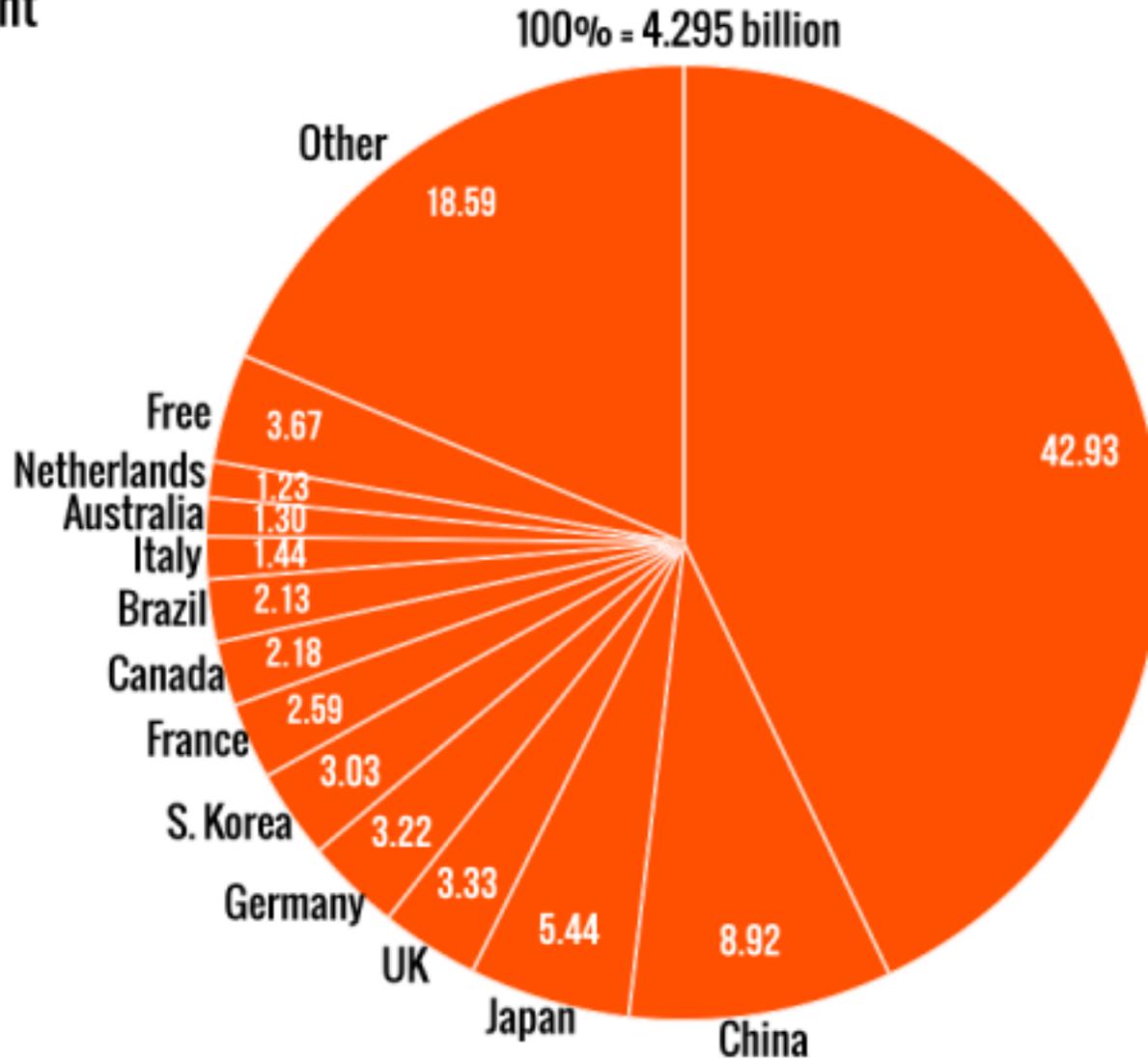
Según la tecnología que utilizan, las redes locales actuales suelen ser inalámbricas con el protocolo *802.11x* o *Ethernet* (*10/100* o incluso *Gigabit*, como velocidad de conexión); las redes metropolitanas pueden utilizar cable telefónico, fibra óptica, cable coaxial, conexión inalámbrica, etc.

*TCP* descompone la información en paquetes, bloques de datos llamados datagramas, un paquete de *bits* [3, p. 17] que va desde el origen hasta el destino, donde el protocolo recompone los datagramas. *TCP* es la capa intermedia entre el protocolo *IP* y la aplicación. Dado que *IP* aporta los datagramas pero no se centra en la fiabilidad de la comunicación, *TCP* añade el protocolo para que la comunicación sea fiable, sin errores o pérdidas y con seguridad, así como un mecanismo para distinguir aplicaciones a través de los puertos [251].

#### 7.4. IPv4, IPv6

Cada ordenador cuenta con una dirección *IPv4*, donde v4 se refiere a la versión cuarta del protocolo [191], que usa direcciones de 32 *bits* limitando el número de direcciones a dos elevado a treintaidós, lo que resulta un total de cuatro mil doscientos noventa y cuatro millones, novecientos sesenta y siete mil doscientas noventa y seis direcciones únicas *IPv4*. Es de plena actualidad el ocaso de *IPv4* por el agotamiento del número de direcciones únicas, cuya alarma saltó en febrero de 2011 [177]. A partir de esa alarma han sido los registros regionales los que se han encargado de administrar las direcciones que restan. De hecho, la autoridad estadounidense encargada de otorgar direcciones *IP*, *ARIN* (*American Registry for Internet Numbers*, registro estadounidense para números de Internet) ha tenido que negar una petición de direcciones [146] ya que solo disponen de 134.000 direcciones libres. En *ARIN* se encuentran *Phase 4* (fase 4) [224] desde 2014, cuando dejó de disponer de más de diecisiete millones de direcciones libres.

## IP ADDRESSES HELD BY COUNTRY Percent



Hay un desequilibrio evidente de direcciones *IP* a nivel mundial, tanto en su reparto como en su uso, debido a los orígenes de Internet y a quienes han utilizado primero las

tecnologías y el grado de desarrollo y de inversión de cada país. También ha habido cambios sorprendentes, como China, que contaba hace diez años con menos direcciones *IP* que la *Universidad de Stanford* [224] y ahora es el segundo país con más direcciones, un nueve por ciento del total, por detrás de EE.UU., que dispone de casi el cuarenta y tres por ciento. La *NRO* (*Number Resource Organization*, organización de números de recursos) cuenta con cinco registros de Internet regionales o *RIR* (*Regional Internet Registries*): *AFRINIC* para África, *APINIC* para Asia y el Pacífico, *ARIN* para Canadá, parte del Caribe, islas del Atlántico Norte y EE.UU. de Norteamérica; *LACNIC* para Latinoamérica y parte del Caribe; y *RIPE* para Europa, Oriente Medio y Asia Central.

En los últimos años, el auge de la conexión a Internet ha crecido a escala planetaria, aunque de manera desigual. Ha aumentado el número de hogares conectado a Internet permanentemente, ya que no solo se dispone de un ordenador sino varios dispositivos, como la tableta electrónica, el portátil, la televisión o el teléfono móvil. Si a esto le sumamos la llamada *Internet de las cosas* (*Internet Of Things* o *IoT*), que pretende que cualquier dispositivo electrónico esté conectado –frigorífico, lavadora, calefacción, casa, reloj, semáforo, farola, sensor, etc.–, la necesidad de direcciones es acuciante. Entre tanto, las casas particulares, empresas o entidades pueden suplir la conexión de múltiples equipos a través de *NAT* (*Network Address Translation*, traducción de direcciones de red), que intercambia paquetes entre dos redes, la interna e Internet.

El futuro se escribe en *IPv6*, la versión sexta del protocolo *IP*, *RFC 1883* de 1995 [89], reemplazado por *RFC 2460* de 1998 [90]. En vez de direcciones de 32 bits, *IPv6* dispone direcciones de 128 bits, lo que significa un rango de direcciones muchísimo más grande, dos elevado a 128, 340.282.366.920.938.463.463.374.607.431.768.211.456 o 340 sextillones de direcciones [258].

## 7.5. HTTP

*HTTP* es el protocolo usado en cada transacción de la Web [44]. Desarrollado por el *W3C* en colaboración con y la *IETF*, se publicaron una serie de *RFC*, el más importante de ellos el *RFC 2616* que especifica la versión *HTTP 1.1* [106]. Define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxis) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición, un navegador web, por ejemplo, se lo conoce como *agent* (*UA* o agente del usuario), mientras que a la información transmitida se le llama *recurso* y se identifica mediante un localizador uniforme de recursos (*Unified Resource Locator* o *URL*).

*HTTP* es un protocolo sin estado o *stateless*, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener el estado, tener información del mismo, por lo que se utilizan métodos como las *cookies*, que guardan información del servidor en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de *sesión*, y también permite rastrear usuarios ya que las *cookies* pueden guardarse en el cliente por tiempo indeterminado.

La versión *HTTP 1.0* definía tres métodos de uso: *GET*, *POST* y *HEAD*. La versión *1.1* añadió cinco nuevos métodos: *OPTIONS*, *PUT*, *DELETE*, *TRACE* y *CONNECT*. En total, ocho métodos o verbos que indican la acción que se ha de efectuar sobre el recurso:

- *HEAD*, *cabecera*, pide una respuesta idéntica a la que correspondería a una petición *GET*, pero sin el cuerpo de la respuesta. Esto es útil para la recuperación de metainformación escrita en los encabezados de respuesta, sin tener que transportar todo el contenido.
- *GET*, *obtener*, Pide una representación del recurso especificado. Transmite infor-

mación a través de la *URI* agregando parámetros a la *URL*:

```
GET /imagenes/logo.png HTTP/1.1
```

Obtiene un recurso llamado `logo.png`. Puede llevar parámetros, por ejemplo,

```
GET /index.php?pagina=principal&lang=es HTTP/1.1
```

- **POST**, *publicar*, hace que los datos que se incluirán en el cuerpo de la petición sean procesados para el recurso identificado, lo cual se puede producir en la creación de un nuevo recurso o actualizaciones de los recursos existentes o ambas cosas.
- **PUT**, *poner*, sube o carga un recurso especificado. A diferencia de *POST*, no utiliza un mensaje multiparte y por tanto el mensaje no tiene que ser decodificado por el servidor (*host*). Este método permite escribir un archivo en una conexión (*socket*, enchufe o encaje) establecida con el servidor. La desventaja del método *PUT* es que los servidores de alojamiento (*hosting*) compartido no lo tienen habilitado. Por ejemplo:

```
PUT /ruta/documento-nuevo.html HTTP/1.1
```

- **DELETE**, *borrar*, borra el recurso especificado.
- **TRACE**, *trazar*, solicita al servidor que envíe de vuelta los datos que reciba del mensaje de solicitud. Se usa para comprobar el servicio.
- **OPTIONS**, *opciones*, devuelve los métodos *HTTP* que el servidor soporta para una *URL*.
- **CONNECT**, *conectar*, crea una conexión con el servidor.

De estos ocho métodos, cuatro se identifican con las funciones básicas de una base de datos, recogidas en el acrónimo *CRUD* [256]: *Create*, *Read*, *Update* y *Delete* o crear, leer, actualizar y borrar. Se pueden utilizar en *HTTP* para los *recursos*, en cuanto que *Create* se puede realizar con los métodos *PUT* y *POST*; *Read* (leer, pero también *retrieve*, recuperar) se puede realizar con *GET*; *Update* (actualización, modificación), con *PUT*; y *Delete* con *DELETE*.

El servidor web o servidor *HTTP* debe responder con un código al servir el recurso [232], y según ese código aporta una información u otra:

- 1xx, mensaje de información, bien porque:
  - 100, *continúa*, solo una parte de la información solicitada ha sido recibida. No se rechaza, el cliente puede seguir con lo solicitado;
  - 101, o bien porque el servidor cambió de protocolo, muy utilizado en la primera época de la Web.
- 2xx, la operación se ha realizado con éxito:
  - 200, operación realizada con éxito
  - 201, *creada*, se completó la solicitud y se creó un nuevo recurso.
  - 202, *aceptada*, se aceptó la solicitud
- 3xx, redirección hacia otra *URL*.
- 4xx, error por parte del cliente, entre las que son más comunes:
  - 400, petición errónea.

- 403, recurso prohibido. La *URL* existe pero por no se permite el acceso público, suelen requerir algún tipo de autenticación.
- 404, URL no encontrada. Suele ocurrir que muchas URLs correspondan a dominios que no existan ya o bien que cambien dentro de su dominio.
- 5xx, error por parte del servidor, entre las que destacan:
  - 500, error interno, no se puede completar la operación.
  - 503, servicio no disponible, suele ocurrir por sobrecarga o baja del servidor.

Si crea el recurso de forma correcta con POST, se obtiene 201; si se accede al recurso con GET, se obtiene 200 y si se produce un error, 404 ó 400; si se actualiza con PUT, se obtiene 200 ó 204 y 201 si se crea; al borrar el recurso con DELETE, el mensaje que se obtiene es 200.

## 7.6. *URL*

Hay tres términos para definir a los recursos: *URL*, *URIs* (*Unified Resource Identifier*, identificador uniforme de recursos) o *URN* (*Uniform Resource Name*, nombre uniforme de recursos). *URN* se utilizaba desde 1997 [163] como complemento de la *URL*, pero desde 2005 [45] se utiliza el menos restrictivo *URI*, que engloba tanto a *URL* como a *URN*, y cuyo esquema se utiliza en multitud de protocolos oficiales de *IANA* [267]. Esta nomenclatura proviene de una diferencia clásica en computación entre nombres y direcciones: un nombre se asociaba a un objeto mientras que las direcciones eran localizaciones del objeto en memoria, que podían variar [46, pp. 25], por lo que un *URI* (identificador) podía tener una *URN* (nombre) y una *URL* (localización). En la

actualidad una *URI HTTP* es equivalente a una *URL*.

Una *URI* consta de las siguientes partes:

- Esquema, nombre que se refiere a una especificación para asignar los identificadores, en algunos casos resultan protocolos. Por ejemplo, son esquemas reconocidos *bitcoin*, para enviar dinero a una dirección *Bitcoin*, o *chrome*, para configurar *Google Chrome*, pero también *dns*, *feed*, para suscripciones a sitios de noticias, *ftp*, para recursos *FTP* (*File Transfer Protocol*, protocolo de transferencia de ficheros), o el más común, *http*. El esquema comienza con este nombre, seguido por dos puntos :, seguido por dos barras // y la dirección del dominio. Después de la dirección del dominio se puede especificar el puerto que utiliza, separado del resto por otros dos puntos :, y a continuación la ruta al recurso dentro del dominio (que identifica un *host* o servidor). Por ejemplo:

`http://es.wikipedia.org:8080/wiki/`

Donde:

- `http` es el esquema.
- `es.wikipedia.org` es el dominio del servidor.
- `8080` es el puerto (el protocolo *HTTP* va por el puerto *TCP* número 80 de forma predeterminada, por lo que no hay que especificarlo si se trata del 80).
- `/wiki` es la ruta dentro del servidor.
- Autoridad, elemento jerárquico que identifica la autoridad de nombres, como, por ejemplo, en `//infotics.es`.

- Ruta, la ruta del recurso, organizada de forma jerárquica, por ejemplo, /persona/anton
- Consulta, información con estructura no jerárquica, normalmente de pares clave/valor, que identifica el recurso y cuyo comienzo se indica mediante el componente ?, por ejemplo, ?url.
- Fragmento, que identifica una parte del recurso principal, o vista de una representación del mismo. El comienzo de este componente se indica mediante el carácter #.

Por recursos no hay que entender solo a direcciones de acceso a documentos *HTML*, sino que también cualquier elemento informacional disponible en la Web, como un archivo, el resultado de la ejecución de un programa, una consola a una base de datos, un comentario en una página web, la traducción automática de un documento, etc. Es decir, cualquier contenido identificable, independientemente del formato.

*W3C* advierte a los creadores de contenido que las *URI* bien hechas no tienen que cambiar [239]. Si el dominio sigue perteneciendo al creador de la *URI* no hay razón para ello y si se considera que su contenido está obsoleto, se pueden utilizar los metadatos apropiados que ofrecen fecha de creación del documento y fecha de expiración, por ejemplo, entre otros.

## 7.7. *API REST*

Cualquier recurso se puede acceder a través de aplicaciones que cumplan con el estándar *REST* (*Representational State Transfer* o transferencia de estado representacional) una interfaz para el acceso a los recursos. La *API* (*Application Programming Interface* o interfaz de programación de aplicaciones) es una capa de abstracción para la comunica-

ción entre aplicaciones. En el caso de las aplicaciones web, donde se realizan peticiones *HTTP*, se utilizan *API REST*, un estilo creado por Roy Fielding, uno de los autores de *HTTP* [107, cap. 5]. No es el único medio para hacerlo: se puede utilizar también *XML-RPC* (*XML-Remote Procedure Call*, llamada a procedimiento remoto), que usa *XML* para codificar las llamadas y *HTTP* para el transporte, o de su derivado *SOAP* (*Simple Object Access Protocol*, protocolo de acceso a objetos simple), que define cómo se pueden comunicar dos objetos en distintos procesos a través de intercambio de datos en *XML*. *SOAP* y *XML-RPC* invocan métodos sobre servicios remotos mientras que *REST* opera sobre recursos [217]. Sin embargo, se está convirtiendo en el estándar por su simplicidad y su utilidad ya que utilizamos la propia estructura y semántica del recurso para identificarlo, para relacionarlo y para interoperar con él por los métodos estándar de *HTTP*. De hecho, cuando ahora hablamos de *API*, nos referimos a una *API RESTful*, una *API* compatible con *REST*, aunque debe estar guiada por el hipertexto [108] y no por los servicios.

El estilo *REST* se basa en cinco mandatos obligatorios y uno opcional [152]:

1. Estructura cliente-servidor, que permite la separación de funciones, donde el cliente se ocupa de la interfaz de usuario y por tanto los datos se adaptan a las distintas plataformas, y el servidor se ocupa del almacenamiento de los datos.
2. Al no guardar el estado el servidor sino el cliente, cada petición del cliente ha de contener la información necesaria para que el servidor responda adecuadamente.
3. Se puede utilizar la caché en el lado del cliente para manejar los datos, o bien en intermediarios como proxis o cachés compartidas.
4. Interfaz uniforme, característica principal de *REST*, permite realizar desde pequeñas transferencias de datos hipermedia hasta grandes cantidades. La identi-

ficación de recursos se realiza por *URI* y su manipulación por representaciones (hiperenlaces *HTML*, formularios o con estilo *JavaScript AJAX*. Debe ser *HATEOAS*, un acrónimo para *Hypermedia as the Engine of Application State* o *hipermedia como el motor del estado de la aplicación* que indica que la *API* puede ser recorrida por completo sin documentación.

5. Sistema de capas, donde cada componente solo conoce el componente o la capa con la que interacciona, lo que permite filtros de la información que se pueden combinar con la restricción de la interfaz. Pueden utilizarse también componentes intermedios como balanceadores de carga, proxis inversos o cachés compartidas.
6. Código bajo demanda, opcional, donde el cliente no sabe cómo procesar los recursos a los que accede. El código que representa este conocimiento lo obtiene el cliente y lo ejecuta localmente.

El *TAG* (*Technical Architecture Group*, grupo de arquitectura técnica) de *W3C* desarrolló un estilo de arquitectura *REST* en paralelo con *HTTP 1.1*. Se puede considerar la Web como un gran sistema que funciona al estilo *REST* [264].

Las *API* de servicios web se definen por tener una *URI* base, un tipo de datos (*JSON*, *SML*, *Atom*, *Microformats*, etc.), métodos *HTTP* estándar, enlaces para las referencias de estado y para las referencias a recursos seleccionados. Cuando se envía *GET* sobre el recurso *URI* base, se muestra el listado y otros detalles de la colección, mientras que si se emplea sobre la *URI* de un elemento, se muestra una representación del elemento.

La estructura de la *URI* debe ser:

```
{protocolo}://{dominio o hostname}[:puerto (opcional)]/{ruta del recurso}?{consult
```

Los nombres de las *URI/s* no deben implicar una acción, por lo que no se suelen incluir verbos. En su estructura se explicita una jerarquía lógica sobre la que operan los métodos */HTTP*. Para añadir información al recurso se utilizan los parámetros *HTTP*.

En una *API* con una *URI* como esta, `http://infotics.es/personas`, a través de *GET* se podría obtener el listado de recursos correspondientes, personas. Si los recursos fueron objetos *JSON*, podrían tener esta estructura:

```
{
  "Nombre" : "Adolfo",
  "Apellido" : "Antón",
  "Edad" : "39",
  "link" : [ {
    "rel" : "self",
    "href" : "http://infotics.es/persona/anton"
  } ]
}
```

Esta misma estructura, en *XML*, se escribe:

```
<?xml version="1.0"?>
<persona>
  <nombre>Adolfo</nombre>
  <apellido>Antón</apellido>
  <edad>39</edad>
  <link rel="self" href="http://infotics.es/persona/anton" />
</persona>
```

Luego esta información, Anton, se puede obtener con una petición *GET* a `/personas/anton`. Para editar el recurso, se utiliza el método *PUT* con `PUT /personas/anton`, mientras

que para añadir una persona nueva, se utiliza `POST /personas`. Para borrar se utiliza `DELETE`, por ejemplo, `DELETE /personas/anton`. Para cada acción, cada operación, se obtiene un código de la *API* y representación del recurso. Para cumplir con el principio de *hipermedia*, debe incluir la *URI* de la persona, donde `rel` indica la relación con el enlace `href`, de cara a ofrecer información sobre la navegación por la interfaz.

Aunque en la *URI* no se ha especificado el formato, en la petición *HTTP* sí que se puede especificar en qué formato se prefiere recibir la representación del recurso, admitiendo varios, en orden de preferencia, con la cabecera `Accept`. La *API* devolverá la representación del recurso en el primer formato disponible o un `406` en caso de no disponer de ninguno de esos formatos:

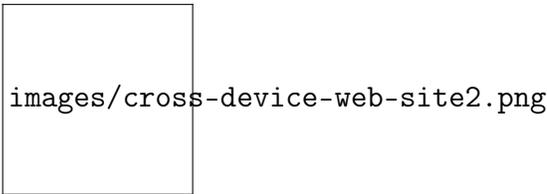
```
GET /personas/abril
Accept: application/json, application/xml, application/pdf
```

Si el recurso existe:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

El éxito de *REST* en el desarrollo de aplicaciones web y servicios web *RESTful* llevó a Leonard Richardson y Sam Ruby a trasladar esta filosofía a la ingeniería de software y a la creación de *ROA Resource-Oriented Architecture*, un paradigma de programación que apuesta por interfaces *RESTful* [197], donde los recursos son componentes de software.

En resumen, en una estructura *RESTful* se ha de usar correctamente *URI*, *HTTP* e implementar hipermedia. La Web Semántica y sus *media types* relacionados encuentran en estas estructuras un medio donde desarrollarse [152].



## 7.8. DNS

Las *URL* se apoyan en el sistema de gestión de dominios, que resolvió el problema de cómo identificar de una forma nominativa el crecimiento del número de equipos conectados a Internet. El *DNS*, *Domain Name System* o sistema de nombres de dominio, pone una máscara a las direcciones *IPv4*, traduce a un nombre la sucesión de números *IP*, permitiendo además utilizarlo en los distintos servicios de Internet, entre ellos la Web.

La entidad encargada de los dominios se denomina *IANA* (*Internet Assigned Numbers Authority*, *Agencia para números de Internet asignados*), y establece una jerarquía de los dominios que se visualiza en las direcciones, siendo la parte más alta de la jerarquía la situada a la derecha, donde cada parte se denomina *etiqueta*. Los dominios de alto nivel, *TLD* (*Top Level Domain*) o principales, son las terminaciones que de los dominios, en una lectura de derecha a izquierda, la última etiqueta del dominio. Los denominados dominios son subdominios de uno de los *TLD*. Un dominio es, por ejemplo, *ucm.es*, donde *ucm* (*Universidad Complutense de Madrid*), es subdominio de *es*, el *TLD* reservado para indentificar a *España* y gestionado por *nic.es*. También se suele configurar con el subdominio *www*, como en *www.ucm.es*, aunque se suele utilizar uno solo de cara a las *URL*. En teoría, la subdivisión puede alcanzar los 127 niveles y cada etiqueta puede llegar a 63 caracteres, pero siempre que el nombre del dominio no exceda los 255 caracteres.

*DNS* funciona con una arquitectura cliente-servidor donde además de clientes y servidores entran en juego las zonas de autoridad o *ZOA* (*Zones of Authority*). Cuando se realizan peticiones *HTTP* sobre recursos con *URL*, el cliente envía una consulta al servidor *DNS* que tenga configurado el equipo local o *localhost* para que traduzca el dominio a dirección *IPv4*. Si el servidor es recursivo, puede reenviar la petición a otro servidor si no dispone de la información solicitada. *ZOA* es el componente matriz donde el nombre de dominio almacena los datos de al menos un dominio y de los subdominios que estén configurados en esa *ZOA*.

Los *TLD* han evolucionado tanto como la Web o los servicios web. Al principio solo existía el dominio *arpa*, que pretendía ser transitorio mientras se implementaba el sistema de nombres de dominio, aunque se sigue utilizando con fines técnicos. Los siguientes fueron, en 1980, los siete *TLD* genéricos (*Generic top-level domains*) o categorías, que se correspondían con el tipo de institución que pretendía un dominio, por lo que había *org* (*organizations*) para organizaciones, *int* (*internet*) para organizaciones de Internet, *net* (*red*) para entidades encargadas de servicios de red, *gov* (*government*) para gobiernos, *com* (*company*) para empresas, *mil* (*military*), para entidades militares y *edu* (*education*) para ámbitos educativos; *com*, *net* y *org* se pueden utilizar sin restricciones. Luego llegaron los *ccTLDs* (*country code TLD*, código de país de dos caracteres según estándar *ISO 3166*), *sTLD* (*sponsored TLD*, *TLD* patrocinados) y otros.

Muchos dominios patrocinados se añadieron en 2000, como, por ejemplo, *coop* patrocinado por la *Cooperative League of the USA* o *geo* por *SRI International* [137]. Desde 2009 se pueden usar caracteres no *ASCII* que están internacionalizando el nombre de los *TLD*, no solo con los nombres chinos, indios, árabes o cirílicos sino también los que incluyen la letra ñ, por ejemplo.

*IANA* se ocupa de la gestión de la *ZOA root* [5], de la gestión del registro de los números de Internet y de la gestión del registro de parámetros del protocolo, dependiente en la actualidad de *ICANN* (*Internet Corporation for Assigned Names and Numbers*,

entidad de Internet para los nombres y los números asignados) [138], una organización sin ánimo de lucro que busca el beneficio público y mantener Internet seguro, estable e interoperable.

El término o etiqueta situada más a la izquierda suele dar información sobre el uso que se da a ese subdominio. En algunos casos responde al servicio de Internet, como, por ejemplo, `mail.ucm.es`, donde `mail` indica que se trata del dominio que utiliza el servicio web; o `www.ucm.es`, donde `www` indica que se trata del dominio que utiliza el servicio web. También puede indicar la estructura de la entidad, como, por ejemplo, `biblioteca.ucm.es`, donde se refiere a la web de la biblioteca de la *UCM*. Otras veces representa al nombre del *host* (ordenador), para la administración de los servicios o de la red. Estas estructuras de dominio se conocen como *FQDN Fully Qualified Domain Name*, nombre de dominio cualificado.

Las etiquetas son cadenas alfanuméricas (con `-` como único símbolo permitido), deben contar con al menos un carácter y un máximo de 63 caracteres de longitud, y deberá comenzar con una letra y no con un guión [164]. Las etiquetas individuales están separadas por puntos y el nombre de dominio termina con un punto, aunque este último punto generalmente se omite, ya que es puramente formal. Un *FQDN* correcto incluiría el punto final `www.example.com.`

El tipo de registro *DNS* más habitual es *A* (*all*, todos), que traduce nombres de *hosts* a direcciones *IPv4*. Cuando utiliza subdominios, se crean alternativas con *CNAME* (*Canonical Name* o nombre canónico). También se utilizan mucho *NS* (*Name Server*, servidor de nombres), que define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio, donde cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres; *MX*, (*Mail Exchange*, registro de intercambio de correo), asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio; *SOA* (*Start of authority*, autoridad de zona), proporciona información sobre el servidor *DNS* primario de

la zona; y SPF (*Sender Policy Framework*, esquema de las políticas de envío), uno de los más actuales ya que ayuda a combatir el *spam* (correo electrónico no deseado).

En una estructura arbórea como la de los documentos web, igual que la estructura de directorios de *Unix* [213], pueden existir distintos documentos en distintos ordenadores *hosts* que compartan el mismo nombre de documento, pero si cada equipo (*host*) utiliza un dominio o subdominio, la *URL* de cada documento será única, uno (<http://undominio/documento.html/> y otro <http://otrodominio.documento.html>. Al igual que en *XML* se utiliza el *namespace* o *espacio de nombres* para referirse a a un conjunto de nombres que es único en un contexto determinado, en el *DNS*, el espacio de nombres se refiere al *FQDN*, que son las que permiten que haya *URL* únicas.

---

## Capítulo 8

### Hardware y software

Un ordenador es *hardware*, la chatarra, los dispositivos físicos, ya sean ordenadores, móviles, servidores, *routers* o partes de esos dispositivos, microelectrónica, tarjetas de red, tarjetas gráficas, discos duros, memoria *RAM*... Los avances en *hardware* se han sucedido uno tras otro, y no es motivo de este trabajo repasarlos, pero conviene recordar algunos como el transistor o el microchip. Se descubre el transistor en 1947 en los laboratorios *Bell* de la mano de John Barden, Walter H. Brattain y William Shockley. Con ese nombre tan ligado a la historia de los medios de comunicación, el transistor reemplaza a las válvulas termoiónicas de tres electrodos o triodo. con lo que se produce un ahorro de peso, energía y tensión que se traduce inmediatamente en los ordenadores de la época. *Transistor* es una contracción de *transferecia de resistencia* o *transference resistor*, y la imagen del pequeño transistor para escuchar la radio es un buen ejemplo de lo que permite: regular el voltaje o amplificarlo.

En otros laboratorios, los de *Texas Instrument*, Jack St. Clair Kilby lanza en 1958 el primer microchip o circuito integrado: todos los materiales podían fabricarse con germanio como material semiconductor, por lo que todos los componentes se sitúan en el mis-

mo material. Los actuales microprocesadores integran miles de transistores. Gordon E. Moore, fundador de *Intel*, defendió en 1965 desde los laboratorios *Fairchild Semiconductor* que cada año se duplicaría el número de transistores en un circuito integrado, lo que se conoce como *ley de Moore*. En 1971 se lanzó el primer microprocesador, el *Intel 4004*, con 2.300 transistores y arquitectura de cuatro *bits*. Con este primer desarrollo propuso cambiar el periodo temporal de duplicación a dos años. El primer microprocesador de 8 *bits* fue el *Intel 8008* de mediados de 1972, con 3.300 transistores y frecuencias de hasta 800 kHz. El *8080* se considera el primer microprocesador para el uso general, con 4500 transistores y 200.000 instrucciones por segundo a 2 Mhz. Poco después, el primer procesador de 16 bits, el *8086*, es el precursor de la arquitectura *x86*.

También es relevante la llamada *arquitectura von Neumann*, diseñada en 1945, que recoge las aportaciones realizadas por Babbage o Turing para definir una arquitectura de los ordenadores. Se aplicó por primera vez en *EDVAC* (*Electronic Discrete Variable Automatic Computer*), sucesora de *ENIAC*.

El software es el conjunto de programas que funcionan en esos dispositivos, incluidos el *kernel* –núcleo del sistema operativo– y el sistema operativo. El *kernel* es software que intermedia entre el hardware y el software que el sistema operativo lleva. Para que el ordenador haga algo se necesita un lenguaje de programación, un algoritmo que dé unas órdenes pertinentes al hardware. Se considera el primer código o software a las anotaciones que realizó Ada Lovelace de la traducción del libro de Luis Menabrea sobre la máquina de Charles Babbage. Son fundamentales también el primer lenguaje de programación independiente de arquitectura *COBOL* o el nacimiento del sistema operativo *Unix*.

## 8.1. Unix

Un ordenador es un dispositivo sin utilidad de no ser por los lenguajes de programación que componen el software que corre en los dispositivos: sistema operativo (que es un programa en sí) y programas.

El nacimiento de *Unix* y su estándar de jerarquía de ficheros *FHS*, *File Hierarchy Standard* o *estándar de jerarquía de ficheros* [73] fue una revolución informática al proponer un ordenador que trabajaba de una forma completamente nueva, con una estructura de tres partes:

1. *Kernel* o núcleo del sistema operativo, con los programas que se relacionan con el hardware.
2. *Shell* o *command line interface*, que es la forma de relacionarnos con el *kernel*.
3. Software, todas las aplicaciones con las que cuenta el sistema operativo, incluido el propio sistema.

Anterior a *Unix* fue importante el trabajo liderado por Fernando Corbato en el centro de computación del *MIT*, quienes crearon el primer sistema operativo multiusuario llamado *Compatible Time-Sharing System (CTSS)* que sirvió de base de *MULTICS Multiplexed Information and Computing System*, desarrollado entre los laboratorios *AT&T Bell*, *General Electric* y *MIT*. Pero Dennis Ritchie (alias *dmr*) y Ken Thompson se separaron del anterior desarrollo en 1969 y realizaron desde los laboratorios *Bell* el desarrollo de este sistema operativo bautizado como *UNICS (Uniplexed Information and Computing System* o sistema de información y computación uniplexada), rápidamente rebautizado como *UNIX* en 1970. Entre sus novedades destacan el sistema de ficheros, las tuberías, el sistema multiusuario, ser multitarea y multiprocesador. En 1971 se publicó el manual de programación de *UNIX Unix programmer's manual* [94].

Como el sistema no era portable, pues estaba escrito en código ensamblador y no se podía ejecutar en otro ordenador que no fuera el *PDP-11*, Ritchie desarrolla con Brian Kernighan el código *C*, basado en *B*, que permitía una programación estructurada y economización de las expresiones. Entonces el equipo de *Bell* decide en 1972 reescribir *Unix* en lenguaje *C* [210], con lo que podría instalarse en otros ordenadores y abría el proyecto a más personas, entidades o instituciones que quisieran participar. Es la semilla de los sistemas abiertos, de código abierto u *open source* [228].

A partir de 1973 se popularizó mucho en el ambiente universitario y también entre empresas de ordenadores, y en 1991 un programador finlandés, Linus Torvalds, anunció el desarrollo de un kernel para un sistema operativo que llamaría *Linux*. Entre la comunidad de usuarios de *Unix* se encontraba Richard Stallman, quien después de algunos problemas de licencia con un software que había desarrollado decidió comenzar el proyecto *GNU* (acrónimo recursivo para *GNU Not Unix*, *GNU no es Unix*), un sistema operativo que contaría con una licencia denominada *GPL* (*General Public Licence*, *licencia pública general*) para asegurar que todo el software del sistema operativo fuera de libre uso. Cuando Torvalds comenzó *Linux*, lo licenció con *GPL* en agradecimiento al compilador *GC++* que había utilizado, que fue desarrollado anteriormente por Stallman como parte del proyecto. En ese momento se produjo la confluencia de los proyectos *GNU* y *Linux* y nació el sistema operativo *GNU/Linux*, presente en la mayor parte de los servidores web del mundo.

Ritchie fue también uno de los creadores del lenguaje *C*, junto con Brian Kernighan. De él derivan o se inspiran *C++*, *C#*, *Objective-C*, *D*, *Java*, *JavaScript*, *Limbo*, *Perl* o *PHP*.

## 8.2. Lenguajes informáticos

Se considera que el primer lenguaje de programación fue inventado por Augusta Ada King, condesa de Lovelace, hija del poeta Lord Byron. Como matemática, trabó amistad con Charles Babbage, quien estaba desarrollando una máquina analítica, un ordenador mecánico de propósito general. En 1842 tradujo un artículo en francés sobre la máquina analítica que había escrito el ingeniero militar italiano Luigi Menabrea. Babbage le sugirió a Ada que escribiera sus propias notas sobre la máquina, y escribió tres veces más contenido que el original. En este texto se incluía un algoritmo que debía correr la máquina para calcular los números de Bernoulli, lo cual se considera el primer documento donde se detalla un lenguaje de programación [218]. Pero además, Ada pensó que esa máquina sería capaz de componer música, producir gráficos y utilizarse para usos prácticos y también científicos, por lo que anticipó el ordenador personal.

En 1979, el Departamento de Defensa de EE.UU. de Norteamérica desarrolló un lenguaje de programación orientado a objetos, concurrente y fuertemente tipado, que llamó Ada en su honor. Ada se utiliza mucho en la aeronáutica, la gestión del tráfico aéreo y la industria aeroespacial.

Los ordenadores entienden ceros y unos, apagado y encendido en caso de transistores o lámparas de las primeras máquinas electrónicas, conducción o cortes en caso de semiconductores o bajo voltaje y alto de los tubos de vacío. Por eso los lenguajes de programación se compilan para traducirlos al lenguaje máquina.

Los lenguajes informáticos se componen de lenguajes de programación, lenguajes de marcado y lenguajes de presentación:

- Los lenguajes de programación son lenguajes formales diseñados para expresar procesos que pueden realizar los ordenadores como el control del comportamiento físico y lógico, expresar algoritmos de precisión o como método de comunica-

ción humana.

- Los lenguajes de marcado son un conjunto de instrucciones que permiten estructurar el contenido de un documento a través de etiquetas o marcas que contienen información adicional acerca de la estructura del texto o de su presentación.
- Los lenguajes de presentación se apoyan en los lenguajes de marcado para transformar el documento produciendo cambios de estilo o de presentación.

La responsable del primer compilador para un ordenador electrónico fue Grace Hopper, que además del lenguaje *A-0*, se le considera la madre del lenguaje de programación *COBOL* y del término *bug* (error) informático, ya que encontró un *bug* (insecto) en un relé de su ordenador *Harvard Mark II* que le hizo cometer un fallo. Poco después, Ida Rhodes diseñó el lenguaje *C-10* para el *UNIVAC I* que se utilizaba para calcular el censo. También diseñó el ordenador que usaría el sistema de Seguridad Social.

La creación de *COBOL* fue crucial para el desarrollo de la informática y para la manera de verla por los hombres también, ya que permitía dar instrucciones en inglés a un ordenador, que después de compiladas servían para cualquier tipo de hardware. Isaacson cree que fue el momento en el que el software empezó a ser más importante que el hardware [212].

La *shell* (la concha, el caparazón) es el intérprete de comandos que lanzaron como parte del sistema operativo *Unix*. Es la interfaz más directa entre usuarios y sistema operativo. La *shell* lee las líneas de código introducidas, interpreta su significado, lo transmite al sistema que ejecuta el comando y devuelve el resultado a través de las salidas posibles o definidas en la orden. Cuando se ejecuta un comando, se crea un proceso que abre tres flujos:

- `stdin`, por *standard input* o *entrada estándar*, donde el proceso lee los datos

de entrada. De forma predeterminada, se refiere al teclado y se identifica con el número 0.

- `stdout`, por *standard output* o *salida estándar*, donde el proceso escribe los datos de salida, de forma predeterminada, en pantalla. Se identifica con el número 1.
- `stderr`, por *standard error* o *error estándar*, donde el proceso escribirá los mensajes de error, de forma predeterminada, en pantalla. Se identifica con el número 2.

Los dispositivos de entrada y de salida pueden cambiarse. Unix permite enviar la entrada o la salida hacia archivos nuevos o ya creados:

- `>` de esta forma se crea un nuevo archivo, y si ya existiera se sobrescribe. Por ejemplo, escribimos el resultado del comando `ls`, la salida `stdin`, en el archivo `listado.txt`

```
:~$ ls Descargas/ > listado.txt
```

- `,` de esta forma añadimos la salida `stdin` al archivo, sin sobrescribirlo, y si no existiera se crea:

```
ls Descargas/ > listado.txt
```

Hay muchos tipos de programas para usar la *shell* aunque el más común es `bash` (*bourne again shell*).

La línea de comandos o *prompt* (consola de aviso) es la línea de la *shell* donde se introducen las órdenes de entrada. Normalmente muestra el nombre de usuario, el nombre de la máquina y la ruta, como por ejemplo:

```
adolflow@fantasmagoria:~$
```

Indica que `adolflow`, que es el nombre de usuario del ordenador llamado `fantasmagoria`, `:~` indica que se encuentra en el directorio del usuario, cuya ruta o dirección del ordenador completa sería `/home/adolflow/`; `$` indica que se trata de un usuario normal. Si fuera `root` o superusuario, en vez de `:~$` aparecería `#`.

Los comandos son nombres, y se pueden escribir argumentos opcionales. Por ejemplo, para listar los archivos y directorios del directorio en el que se encuentra, se escribe:

```
:~$ ls
```

`ls` es el comando para listar. Si hubiera un directorio dentro de la carpeta personal llamado *Descargas*, se podría su contenido con:

```
:~$ ls Descargas/
```

Puede aparecer más información del listado con argumentos opcionales:

```
:~$ ls -la Descargas/
```

Uno de los hitos de Unix son las tuberías [213], que permite enlazar la salida estándar de un programa como entrada de otro programa, y así sucesivamente hasta que obtener el resultado deseado. La tubería se representa con la barra vertical `|`. Además, esta sucesión de comandos se puede escribir en un archivo, hacerlo ejecutable y convertirlo en un programa que se ejecute siempre que se desee.

### 8.3. Semántica

Los lenguajes de programación son lenguajes formales diseñados para expresar procesos a través de un conjunto de reglas sintácticas y semánticas que definen la estructura y el significado de sus elementos y expresiones. Hay dos tipos de lenguajes claramente diferenciados: lenguajes de bajo nivel y de alto nivel. Los lenguajes más próximos a la arquitectura de hardware se denominan de bajo nivel, mientras que los más cercanos a los usuarios, de alto nivel.

Entre los lenguajes de bajo nivel se diferencian los lenguajes máquina y los lenguajes ensamblador. El Lenguaje máquina, que utiliza las operaciones fundamentales para el funcionamiento del ordenador, combina ceros y unos es muy rápido pero también difícil de escribir y de depurar. El lenguaje ensamblador, derivado del anterior pero formado por abreviaturas de letras y números mnemotécnicos, es algo más sencillo, los códigos son más ligeros y ocupan poco en memoria, aunque necesitan un lenguaje traductor y es igualmente complejo. Los lenguajes de alto nivel, más cercanos al lenguaje natural que al lenguaje máquina, utilizan estructuras dinámicas de datos, son lenguajes independientes de la arquitectura hardware de la máquina y se traducen en cada ordenador donde se ejecutan. Los hay de propósito general, como *Python* y *Ruby*, o de propósito específico, como *FORTRAN*.

Un programa es la descripción de un algoritmo en un lenguaje de programación que el ordenador es capaz de interpretar. El programa y los datos que manejan se conocen como proceso. En un ordenador corren a la vez varios procesos, decenas o cientos, según la actividad del equipo. Algunos programas utilizan solo un proceso pero otros producen muchos.

Un algoritmo es como una receta de cocina, resuelve un problema con una secuencia de operaciones que permiten obtener un resultado a través de los datos de entrada - ingredientes, palabras, comandos. Sus características son:

- Finitud, ha de concluir en algún momento, aunque tarde mucho tiempo.
- Bien definido, se refiere a los pasos que ha de dar, deben estar bien definidos, ser precisos.
- Entrada, el algoritmo recibe cero o más datos de entrada.
- Salida, el algoritmo produce una o más salidas.
- Factibilidad, cada paso debe poder llevarse a cabo.

Los lenguajes de programación son lenguajes artificiales definidos mediante un conjunto de reglas:

- Léxicas, según las palabras que se pueden usar.
- Sintaxis, según las frases que se pueden formar.
- Semánticas, según el significado de las frases que se pueden formar, denotativa, operacional o axiomática.

En la sintaxis los lenguajes de programación permiten escribir instrucciones y programas que entienden los ordenadores, pero hay un proceso previo, la compilación, el proceso que traduce el lenguaje de programación a lenguaje máquina, una descripción binaria de instrucciones elementales que ha de realizar el microprocesador.

La sintaxis determina el código, la escritura del lenguaje de programación, describe las combinaciones posibles de símbolos que forman un programa. La mayoría de las sintaxis son textuales, utilizan una combinación de expresiones regulares para la estructura léxica, y la notación de *Backus-Nauer* (*BNF* o *Backus-Nauer Form*, forma *Backus Naeur* o forma *BN*) para la estructura gramatical. La notación de *BN* es un metalen-

guaje usado para expresar gramáticas libres de contexto, lenguajes formales.

John Backus desarrolló desde *IBM FORTRAN* (*Formula Translating System*, sistema de traducción de fórmulas), lenguaje de programación alto nivel de propósito general, procedimental e imperativo, que está especialmente adaptado al cálculo numérico y a la computación científica. También desarrolló *IAL* (luego *ALGOL 58*) [10] presentado en el primer *Congreso Mundial de Computación Mundial* (*World Computer Congress*) de 1958 [10], donde adoptaba las reglas generativas de Chomsky para describir la sintaxis del nuevo lenguaje de programación. Por su parte, Peter Nauer identificó la notación de Backus como la *Forma Normal de Backus* (*Backus Normal Form, BNF*), y la simplificó para usar un conjunto de símbolos menor [171]. Posteriormente, a sugerencia de Donald Knuth, el creador de *T<sub>E</sub>X* y de *Metafont*, se agregó el apellido de Peter Nauer por su contribución a la definición, reemplazando *Nauer* a *Normal*. La notación de *Backus-Nauer* se convierte en un la metasintaxis utilizada para expresar gramáticas libres de contexto: es decir, una manera formal de describir lenguajes formales, gramáticas de los lenguajes de programación, de los sistemas de comandos y de los protocolos de comunicación, así como una notación para representar partes de las gramáticas de la lengua natural. En el caso de *HTML* esto se puede comprobar en la sintaxis de la *DTD*, heredera de *SGML*

## 8.4. Principios de la programación

Los lenguajes de programación son lenguajes que están formados por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Son de programación porque permiten especificar de manera precisa sobre qué datos debe operar el ordenador, cómo deben ser almacenados y transmitidos y qué acciones debe tomar bajo una serie de circunstancias posibles.

Las variables contenedores de datos específicos, de distintos tipos. En algunos lenguajes

se debe especificar qué tipo de datos se guardan en la variable. Hay un tipo de variable que se denomina vector porque es una variable compuesta. Los tipos de variables y vectores más comunes son:

- *Char*, por *character* o carácter, contienen un único carácter.
- *Int*, por *integer* o entero, contienen un número entero.
- *Float*: contienen un número decimal.
- *String*, contienen cadenas de texto, es decir, un vector con varias variables de tipo *Char*.
- *Boolean*: contienen 0 o 1. En muchos lenguajes 0 es sinónimo de `false` y 1, de `true`.

Los condicionantes son estructuras de código que indican que para que cierta parte del programa se ejecute deben cumplirse ciertas premisas. Por ejemplo:

- Que dos valores sean iguales
- Que un valor exista
- Que sea mayor o menor que algo

Los condicionantes más conocidos y empleados son:

- `if`, indica una condición para que se ejecute una parte del programa.
- `Else if`, indica una condición, siempre que no cumpla con el `if` previo y sí con la que especifica.

- `Else`, indica que debe ejecutarse cuando se cumplen las condiciones previas.

Un bucle ejecuta constantemente un código mientras se cumpla una determinada condición:

- `For`, ejecuta un código mientras una variable se encuentra entre dos parámetros determinados.
- `While`, ejecuta un código mientras se cumpla la condición que se indica.

Las funciones sirven para hacer cosas que el lenguaje de programación sabe hacer de determinada manera, bien porque venga con unas funciones definidas o bien porque se incorporen nuevas funciones de terceras partes. Las sentencias definen la lógica de un programa o un subprograma (subrutina), manipulan datos descritos anteriormente para producir el resultado deseado. Hay sentencias de asignación o de llamada a un procedimiento, compuestas o de control. Se utilizan para no tener que repetir fragmentos de código. Es un fragmento de código con un nombre asociado que realiza las tareas definidas y devuelve un valor. Si no devuelven valores se les suele llamar procedimientos.

Se escribe un programa o un pequeño *script* cuando se efectúa una misma tarea en distintos momentos. También se puede incorporar esa tarea en otros *scripts*. Para no tener que escribir cada vez el mismo conjunto de instrucciones, se agrupan para poder reusarlas cuando se necesiten de nuevo. Las agrupaciones de código las inventó Maurice Wilkes en 1951. Había construido en 1949 el *EDSAC*, el primer ordenador que almacenaba los programas en la propia circuitería del ordenador. Wilkes bautizó a las agrupaciones de código con el nombre de subrutinas, pero los distintos lenguajes de programación han nombrado a las subrutinas como procedimientos, funciones o métodos con distintas características.

Las subrutinas aportan muchas ventajas:

- Ahorran trabajo: si se hace una vez, no se debe programar de nuevo.
- Facilita el mantenimiento: si descubre errores en una subrutina, se corrige en un sitio y se aplica allí donde se emplee.
- Simplifica el código: la longitud y complejidad del programa se reducen porque las tareas repetitivas ya no aparecen en el cuerpo del programa.
- Facilita la creación de programas: el trabajo se puede dividir entre varios.

Cuando la rutina se utiliza en más de un programa, se incluye en un fichero aparte llamado librería (del inglés *library*). En *Python* se utiliza el término función para referirse a las subrutinas y el término módulo para referirse a las bibliotecas. Las sentencias son un tipo de estructuras de control. Permiten alterar el flujo de ejecución de las órdenes de un programa. Son de varios tipos:

- Sentencias condicionales, donde actúa una condición para ejecutar otra/s sentencias. Son del tipo `if then else`.
- Sentencias de acuerdo al valor de una variable, del tipo `select case`
- Sentencias que se ejecutan mientras se cumpla una condición, `do while`
- Sentencias que se ejecutan hasta que se cumpla una condición, `do until`
- Sentencias que se ejecutan un número determinado de veces, `for next, times`

Las estructuras tienen un único punto de entrada, aunque pueden ser secuenciales, iterativas y avanzadas. Todos los lenguajes de programación tienen sentencias similares pero la sintaxis de cada cual varía.

## 8.5. *JavaScript, XMLHttpRequest y AJAX*

Los lenguajes de programación del lado cliente son los que se ejecutan en el ordenador del propio usuario. El lenguaje *HTML* no se dedica a proporcionar elementos dinámicos en la página web sino a estructurar el contenido; para eso se utilizan *scripts* (pequeñas sentencias de un programa) embebidos en el código fuente de la página, de modo que cuando el navegador la página web la interpreta para que la visualicemos, analiza estos *scripts* a través de un plugin (extensión del programa) específico del navegador.

Lo interesante de los lenguajes del lado del cliente es que son independientes del servidor; y lo negativo es que si nuestro ordenador no los soporta o el navegador no cuenta con el plugin adecuado, no disfrutaremos de ese dinamismo y/o de ese contenido. Además, los *scripts* también pueden ser puertas abiertas a algún virus o a cualquier otro fallo de seguridad, ya que cuando visitamos la página no vemos el contenido de ese *script* pero sí que estamos aceptando su ejecución.

Los *scripts* más populares están basados en Javascript y utilizan *DOM* (*Document Object Model* o modelo de objetos del documento), un modelo de objetos para la representación de documentos, una interfaz de programación de aplicaciones (*API, Access Programming Interface*) que proporciona un conjunto estándar de objetos para representar documentos *HTML* y *XML*, cómo pueden combinarse dichos objetos, y una interfaz para acceder a ellos y manipularlos. *DOM* define la manera en que objetos y elementos se relacionan entre sí en el navegador y en el documento, por lo que los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos *HTML* y *XML* a través de lenguajes como *ECMAScript (JS)*. *DOM* es Recomendación del *W3C* desde 1998 [141], con versión de nivel 2 en 2000 y nivel 3 en 2003.

*DOM* pinta un documento en una estructura arbórea, como una red en árbol donde cada nodo es un objeto que tiene funciones e identidad. Como modelo de objetos, *DOM* identifica:

- Interfaces y objetos usados para representar y manipular un documento.
- Semántica de estas interfaces y objetos, incluyendo comportamiento y atributos.
- Relaciones y colaboraciones entre estas interfaces y objetos

En *JS*, cada objeto tiene un nombre, el cual es exclusivo y único. Cuando existen más de un objeto del mismo tipo en un documento web, estos se organizan en un vector o *array* de objetos. Es posible asignarle una identificación a un objeto, y luego usarla para hacer referencia a éste o bien directamente a través de:

```
document.div["nombre_de_objeto"]
```

O bien a través de la función `getElementById`:

```
document.getElementById("nombre_de_objeto")
```

Los objetos tienen como propiedades dimensiones, color y peso, y se pueden declarar en pares de atributos propiedad/valor. Muchas funciones que afectan al navegador o a la ventana de navegación han sido definidas previamente, aunque se pueden definir nuevas funciones de acuerdo a necesidades concretas.

En el comportamiento de los objetos entra en juego el concepto de *evento*, cualquier comportamiento que afecta al programa que se utiliza, el navegador, en la relación usuario-navegador. Por ejemplo, cambiar la posición del puntero del ratón por la ventana de navegación es un evento, o presionar una tecla o pasar el ratón por encima -poner el foco en- un enlace. Para tratar con los eventos existen los *event handlers* o manejadores de eventos, que actúan en alguna de las tres etapas del evento:

- I. Captura del evento, cuando el evento se traslada al destino.

2. Blanco, cuando llega al destino.
3. Burbujeo, cuando el evento vuelve a la posición original.

EVENTO → Ventana → Document → HTML → BODY → DIV → DESTINO  
RESPUESTA → DIV → BODY → HTML → Document → Ventana → EVENTO

Algunos tipos de objetos pueden estar pendientes de ciertos eventos si añadimos la función `addEventListener`, con el valor `true` si ha de ocurrir en la etapa de captura y `false` si ha de ocurrir en la etapa de burbujeo.

Para trabajar con la estructura de *DOM* se utilizan *frameworks* (entornos de trabajo) de *JS* ya que permiten simplificar la manera de interactuar con los documentos *HTML*, manipular el árbol *DOM*, manejar eventos, desarrollar animaciones y agregar interacción con la técnica *AJAX* a páginas web, como *jQuery*, *Prototype* o *AngularJS*.

Para el intercambio de datos entre la web y el servidor se puede utilizar *XMLHttpRequest*, petición *XML HTTP* también referida como *XHR* o *XMLHTTP*, una especie de interfaz empleada para realizar peticiones *HTTP* y *HTTPS* a servidores Web cuyos datos transferidos pueden ir en cualquier codificación basada en texto como texto plano, *XML*, *JSON*, *HTML* o codificaciones particulares específicas. La interfaz se presenta como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor. Es un borrador de trabajo [225] aunque se utiliza con normalidad en el mundo web para ofrecer contenido dinámico y actualizaciones asíncronas en páginas web mediante otras tecnologías que lo toman como base, como por ejemplo *AJAX*.

*AJAX*, acrónimo de *Asynchronous JavaScript And XML* (*JavaScript* y *XML* asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas o, en la jerga de la Web, *ricas en contenido* (de *RIA*, *Rich Internet Applications*). Se ejecutan en el

cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. *JavaScript* es el lenguaje interpretado que hace las funciones de llamada a *AJAX* mientras que el acceso a los datos se realiza por *XMLHttpRequest*.

Las cuatro tecnologías que entran en juego en *AJAX* son:

- Entorno web, con *HTML* para la estructura de contenidos, *CSS* para el estilo y *DOM* para la disposición de los datos e interacción con los mismo.
- Objeto *XMLHttpRequest* para intercambiar datos de forma asíncrona con el servidor web.
- *XML* como formato de datos, aunque caben otros, como hemos visto.
- *JS* para trabajar con las anteriores.

## 8.6. Arquitectura web cliente-servidor

Si antes la función principal del navegador era descargar documentos *HTML* y mostrarlos en pantalla, en la actualidad la Web es una especie de sustituto a la interfaz gráfica tradicional del *PC* donde se visualizan imágenes, sonidos o vídeos gracias al soporte de distintos formatos y protocolos, entre otras características, gracias al soporte de *HTML5* y a los dispositivos móviles que hacen uso de navegadores o de *HTML5* en las aplicaciones.

La Web, como cualquier otro servicio de Internet o de los ordenadores, funciona con

la arquitectura *cliente-servidor*, donde un programa *cliente*, el que maneja el usuario, realiza peticiones a otro programa *servidor*, el que tiene alojada la página web, que le responde. Para ver una página web, servidor web y navegador se comunican mediante el protocolo *HTTP* o *HTTPS* (la *S* es de *secure*, ya que utiliza *SSL*, *Secure Socket Layer* o capa de conexión segura). La separación entre cliente y servidor es una separación de tipo lógico, el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Al ser el cliente quien inicia las solicitudes o peticiones, se le considera un papel activo y en lenguaje informático se le puede llamar *maestro*. Por ejemplo, es una persona quien desde un navegador solicita una página web, espera y recibe las respuestas del servidor. Normalmente el cliente puede conectarse a varios servidores a la vez, es decir, podemos ver varias páginas web que están en distintos servidores. El programa cliente suele contar con una interfaz gráfica de usuario, que es la que realiza la interacción persona-ordenador (*HCI*, *Human Computer Interface*).

Por su parte, el receptor espera a que lleguen las solicitudes de los clientes, por lo que se le considera un papel pasivo y en lenguaje informático se le conoce como dispositivo *esclavo* (*slave*). Después de recibir la solicitud, la procesa y envía una respuesta al cliente. Los servidores suelen aceptar conexiones desde un gran número de clientes, aunque puede restringirse según las condiciones. Los servidores no suelen interactuar con los usuarios finales pero sí suelen hacerlo con las personas que mantienen el servicio.

## 8.7. Servidor web

En los lenguajes de programación del lado del servidor, la ejecución del *script* se produce en el servidor que ofrece la página web, de forma opaca –sin que la vea– y transparente –sin que afecten– para el navegador y los usuarios. Al recibir la petición de los clientes,

el *script* es ejecutado en el servidor y envía un documento *HTML*. En este caso, el navegador no necesita un *plugin* para visualizar correctamente el contenido de la web ya que el código está oculto para el cliente.

En la *Web 1.0* se utilizaban tecnologías *CGI* (*Common Gateway Interface*, interfaz de entrada común) para realizar las labores que en la actualidad realizan lenguajes que cohabitan con servidores web. Consistía en un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución eran objetos *MIME*. El servidor web pasa las solicitudes del cliente a un programa externo, escrito normalmente en los lenguajes *C* o *Perl*.

*CGI* se consideraba inseguro porque tenía permisos de ejecución del servidor, y pesado, porque cada petición lanzaba una vez el programa. Surgieron alternativas desde el lado de la tecnología *Java*, con *servlets* (programas que en *Java* extienden las posibilidades del servidor), y filtros *ISAPI* (*Internet Server API*, *API* de servidor de Internet) de *Microsoft* que permitían ejecutar un código en un único proceso externo que gestionaba todas las llamadas realizadas por el servidor web, impidiendo a su vez que el servidor ejecutara otros programas.

Luego surgieron lenguajes que se podían incluir en el *HTML* y que se podían interpretar, como *ASP* (*Active Server Pages*, servidor de páginas activo, motor de *scripts* del lado del cliente) de *Microsoft* o *PHP* (*PHP Hypertext Preprocessor*, acrónimo recursivo para procesador de hipertexto *PHP*) o precompilados como *JSP* (*Java Server Pages*, servidor de páginas de *Java*) y *ASP.NET* (entorno de aplicaciones web del lado del servidor) de *Microsoft*.

En los lenguajes del lado del servidor, cuando se realiza una petición de una página web, una petición *HTTP*, el servidor ejecuta el intérprete del lenguaje, lo procesa y genera el contenido de forma dinámica, obteniendo información de una base de datos. Normalmente se utiliza un framework para la gestión de los contenidos, un *CMS* (*Content*

*Management System*, sistema de gestión de contenidos) que se comunica con una base de datos donde se aloja el contenido web, un modelo de tres capas compuesto por plantillas de presentación de contenido *HTML + CSS + JS*, un motor generador de contenido y consultas al contenido en bases de datos.

Algunos *frameworks* siguen el patrón de diseño de software *Modelo Vista Controlador* (*MVC, Model View Controller*) que separan el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes. En las aplicaciones web, la vista es la página *HTML* y el código que provee los datos dinámicos a la página; el modelo es el sistema de gestión de base de datos y la lógica de funcionamiento y el controlador es el responsable de recibir los eventos de entrada desde la vista. Por ejemplo:

1. Cuando los usuarios acceden a una tienda en un sitio web y pincha en el botón de compra, el controlador recibe la notificación de la acción y gestiona el evento a través de un *events handler* (manejador de eventos) o *callback* (llamada de retorno).
2. El controlador accede al modelo, actualizándolo o modificándolo de forma adecuada a la acción solicitada por los usuarios, es decir, actualiza el carro de la compra.
3. La vista despliega la interfaz con el nuevo modelo.
4. La interfaz espera nuevas acciones para volver a comenzar el ciclo.

## 8.8. La interfaz gráfica: el navegador

Para acceder al contenido de la página web se utiliza un navegador, software, que interpreta y muestra (*renderiza*) el código *HTML*, *CSS* y *JS* y lo presenta en pantalla, hardware dispositivo de salida de datos, permitiendo a lxs usuarixs visualizar su contenido. La interfaz permite disfrutar de la navegación, y por ello fue parte del conjunto de tecnologías que formaron la Web desde el inicio. El primer navegador, desarrollado en el *CERN* a finales de 1990 y principios de 1991 por Tim Berners-Lee, se llamó *WorldWideWeb* y sólo funcionaba en estaciones *NeXT*, como era el de Berners-Lee, donde mostró la primera web que visualizaba cualquier documento con formato *HTML*. Luego vino *Erwise*, con algunas características como mejor soporte de fuentes, subrayado de enlaces y múltiples ventanas [47], pero el que se considera primer navegador gráfico es *ViolaWWW* porque ya mostraba gráficos, precursor de *Mosaic*, también gráfico y primer navegador multiplataforma.

Aparecieron otros, como *Amaya*, *Opera*, *Netscape Navigator* o *Microsoft Internet Explorer* [266], aunque lo que alteró la convivencia de la Web fue la introducción de elementos que funcionaban solo en un navegador por parte de *Netscape* y *Microsoft*. La batalla por el navegador o *browser wars* no fue solo comercial a finales de los 1990 y primeros 2000 sino que la Web dejaba de ser un espacio compartido, común y neutral [253], por lo que Berners-Lee alertó de estas prácticas que parecían "extrañar los viejos malos tiempos, anteriores a la Web, cuando uno tenía muy pocas oportunidades de leer un documento escrito en otro ordenador, con otro procesador de palabras o en otra red" [270]. También hubo respuestas desde la comunidad de diseñadorxs o usuarixs comprometidxs con el espíritu de la Web, algunxs de lxs cuales lanzaron la campaña *Best viewed with any browser* [62], *se ve mejor con cualquier ordenador*, como respuesta a los lemas que algunas páginas anunciaban *se ve mejor con ...*. En la misma línea, el *manifiesto sobre la esencia de la Web* enfatizaba que el gran salto comunicativo que se estaba logrando con la Web, se estaba pervirtiendo con la guerra de navegadores, ante lo cual los creadores de páginas web debían asumir un papel activo y apostar por la

estandarización [188].

Pasando por encima de las disputas, de las innumerables situaciones incómodas que ha provocado a diseñadorxs, programadorxs y personas u organizaciones que querían disponer de una web y no querían que la experiencia de lxs usuarixs se viera alterada por el navegador que tuvieran, la guerra actual solo es comercial y ha pegado un vuelco monumental con factores interrelacionados. Por un lado, cuando *Netscape* perdió la batalla del escritorio en el terreno comercial, liberó el código y se creó el proyecto *Fundación Mozilla* en 2003 que cuenta con uno de los navegadores más utilizados en todos los dispositivos, *Firefox*, multiplataforma y con un gran número de extensiones, y que ganó claramente a *Internet Explorer* su cuota de mercado. Por otro, en 2008 *Google* lanzó *Chrome* [49], un navegador que pretendía convertirse también en herramienta de escritorio, con muchas extensiones y multiplataforma, y *Android*, un sistema operativo para los teléfonos móviles, lo que ha supuesto que en la actualidad más de la mitad del tráfico web mundial provenga de *Chrome* [257].

---

## Capítulo 9

### Web Semántica

Los contenidos web, agrupados en documentos web, páginas web y sitios web, los pueden leer lxs usuarixs de la Web, ya sea texto, audio, imágenes, gráficos o vídeos. Para que las máquinas puedan "leer" ese mismo contenido, interpretarlo, se deben emplear lenguajes que entiendan las propias máquinas. Por un lado hay constantes avances en el procesado de lenguaje natural (*natural language processing* o *NLP*), mientras que por otra, compatible con los resultados de la primera, los lenguajes de la Web Semántica pretenden facilitar la lectura, interconexión y interpretación del contenido web, de tal forma que puedan responder a preguntas de lxs humanxs y ofrecerles contenidos relacionados [46]. Por tanto, la Web Semántica es una extensión de la Web en su modo más hipertextual, una hipertextualidad que entiendan las máquinas y no solo que recorran las personas.

Al igual que con *NLP*, la Web Semántica se relaciona con la inteligencia artificial (*Artificial Intelligence* o *AI*), que busca la creación y el diseño de entidades capaces de tomar decisiones, de emular el pensamiento y la inteligencia humana. La primera vez que aparece la expresión se refiere a "la ciencia e ingenio de hacer máquinas inteligentes, es-

pecialmente programas de cómputo inteligentes"[156]

En la actualidad existen distintos métodos de entrada y salida de datos, como pueden ser los sensores físicos, los sensores mecánicos, los pulsos eléctricos u ópticos, así como los producidos por un software determinado. En la época de los datos masivos y de *IoT*, las grandes máquinas de computación necesarias para realizar operaciones complejas caben ahora en la palma de la mano y se encuentran dispositivos en forma de ordenadores portátiles, dispositivos móviles, coches, electrodomésticos, relojes o prendas de vestir, con datos que viajan de los dispositivos a la nube, de ida y vuelta.

## 9.1. Metadatos

La forma inicial de aportar más semántica al documento se basa en los metadatos ofrecidos a través del elemento `meta` y sus distintos atributos. *HTML* también aporta con su sintaxis algo de semántica, sobre todo en su última versión *HTML5*, donde ha incorporado elementos (etiquetas) que dan más información sobre el contenido del documento. Se pueden utilizar además lenguajes nativos de la Web Semántica como *XML* para la codificación de las páginas, *RDF* para aportar datos vinculados y ontologías para especificar conceptos del dominio con *OWL*. Los metadatos fueron definidos por Philip R. Bagley en 1968 para asociar unos datos con otros [11].

Metadato, en singular, procede del griego , meta (*después de o más allá de*) y del latín *datum* (*lo que se da*) y se refiere a los datos que describen otros datos. Sirven para describir o anotar un recurso de cara a hacerlo más inteligible para lxs usuarixs [46, p. 35]. Si los usuarixs, o lxs agentes de usuarixs, son humanxs, entonces podrán escribirse sin estructura, pero si se trata de agentes máquinas, entonces deben estar estructurados para que los puedan leer también. Los metadatos son importantes en la recuperación de información, aportan estructura a contenidos no textuales, organizan los recursos, ayudan a

su archivo e identifican información. Hay esquemas de metadatos como *Dublin Core* (*Dublin Core Metadata Initiative*, *DCMI*, iniciativa Dublin Core de metadatos) y *TEI* (*Text Encoding Initiative*, iniciativa de codificación de texto) que sirven para integración de metadatos de distintos documentos.

Los metadatos plantean al menos dos cuestiones: cuántos metadatos son necesarios explicar el contenido –el documento– y cómo afectará esa descripción del contenido a la inferencia. En *HTML* el elemento `meta` es un elemento multipropósito para representar metadatos. Los detalles del elemento se organizan según sus atributos:

- `name`, nombre, representa metadatos del documento en forma de pares nombre/valor
- `charset`, conjunto de caracteres, declara la codificación de caracteres.
- `http-equiv=content-type` o `content`, el valor del elemento.

Se permiten atributos globales, `name` y `content` donde:

- `name`, de tipo *string* (cadena de caracteres), especifica el nombre de un par nombre/valor. Debe ser un nombre estándar de metadatos definido en la especificación *HTML5* o una extensión registrada del conjunto predefinido de nombres de metadatos.
- `content`, de tipo *string*, especifica la parte del valor del par *nombre/valor*.

El atributo `name` debe ser o un nombre de metadatos definido (*defined metadata name*) o un nombre de metadatos registrado (*registered metadata name*). Los nombres definidos son:

- `application-name`, lo que obliga a que el atributo `content` sea una cadena con

el nombre de la aplicación web que la página representa.

- `author`, el atributo `content` especifica el nombre del/de la autor/a o de un/x de los autorxs del documento.
- `description`, una descripción de la página de no más de 150 caracteres (espacios en blanco incluidos).
- `generator`, identifica el software que genera el documento.
- `keywords`, donde `content` será una lista de palabras clave separadas por comas.

En el documento *HTML* se escribe:

```
1 <meta name="application-name" content="Aplicación de la tesis
  "/>
2 <meta name="author" content="Adolfo Ant n Bravo ,
  adolflow@sindominio.net"/>
3 <meta name="description" content="Tesis sobre Tecnolog as de la
  Web Sem ntica"/>
4 <meta name="generator" content="Emacs"/>
5 <meta name="keywords" content="epistemolog a , sem ntica , web
  sem ntica , tecnolog as de la informaci n , web, www, rdf,
  xml, html, sparql , owl, ontolog a , taxonom a , folksonom a
  "/>
```

El número de nombres permitidos se ha extendido para reflejar distintos usos de los metadatos y se denominan extensiones de metadatos. Algunos utilizan un espacio de nombres propio para identificar los valores posibles de `name` y así evitar posibles duplicidades [249]. Algunos nombres permitidos como `dc.description` corresponden a *DCMI*, un modelo de metadatos para describir recursos, estándar *ISO-15836:2003* que usa *XML* y se basa en *RDF*.

## 9.2. Capas de la Web Semántica

El *W<sub>3</sub>C* ha explicado la estructura, el funcionamiento y el futuro de la Web Semántica a través de la ilustración *Semantic Web Stack* (pila de la Web Semántica), también conocido como *Semantic Web Cake* (pastel de la Web Semántica) o *Semantic Web Layer Cake* (pastel de capas de la Web Semántica) que muestra la jerarquía de los lenguajes y las tecnologías involucrados por capas, donde cada capa utiliza y aprovecha las capacidades de las capas que tiene por debajo.

La capa inferior o principal está compuesta por *Identifiers: URI* y *Character set: UNICODE*. *URI* ofrece identificación -semántica- única para recursos web. Se ha actualizado y generalizado como *IRI* (*Internationalized Resource Identifier* o identificación internacional de recursos), definido por el *IETF* como nuevo estándar en 2005, publicado como *RFC 3987* [93]. La diferencia entre *URI* e *IRI* es que mientras que las *URI* se limitan al subconjunto de caracteres *ASCII*, *IRI* puede contener caracteres de *UNICODE/ISO 10646*, lo que incluye chinos, japoneses, coreanos, cirílicos, etc. Las *IRI* se mapean a *URI* para mantener la compatibilidad con los sistemas que no soportan el nuevo formato. En ese sentido, el conjunto de caracteres propuesto, *Unicode*, se utiliza por su capacidad para representar y manipular texto en cualquier idioma. La potencia de *URI* es la de ofrecer una sintaxis uniforme para toda la Web de cara a enlazar, recuperar o representar el contenido [46, p. 24]

Las capas siguientes contienen las tecnologías más conocidas del hipertexto y la Web, y son la base de la Web Semántica:

- *XML*, responde al lenguaje de marcas extensible que permite la creación de documentos compuestos de datos estructurados.
- *XML NS*, *XML Namespace*, espacio de nombres *XML*, ofrece una manera de utilizar las marcas –las etiquetas– de distintas fuentes, identificándolas por su

espacio de nombres. La Web Semántica trata de conectar datos -objetos de datos- y los relaciona creando nuevos objetos de datos compuestos por el contenido que procede de fuentes diversas.

- *RDF, Resource Description Framework*, marco de descripción de recursos, es un esquema para la creación de declaraciones en forma de tripletas de valores. Permite representar información sobre recursos en forma de grafos, trata de aportar significado (semántica) a los datos estructurados (*XML*) y es el formato estándar de intercambio de datos semánticos.
- *RDFS, o RDF Schema*, esquema *RDF*, aporta un vocabulario básico para *RDF*, y permite crear jerarquías de clases y propiedades.
- *OWL, Ontology Web Language*, lenguaje de ontologías web, extiende *RDFS* añadiendo construcciones más avanzadas para describir semánticamente las declaraciones *RDF*. Permite declarar limitaciones adicionales, restricciones de valores o características de propiedades como la transitividad. Se basa en la lógica de la descripción y aporta lógica de razonamiento a la Web Semántica.
- *SPARQL*, acrónimo recursivo para *Sparql Protocol And RDF Query Language*, lenguaje del protocolo *SPARQL* y de consultas *RDF*, es un lenguaje para realizar consultas o *queries* sobre datos *RDF*, recupera información de contenido semántico en *OWL* o *RDF*.
- *RIF, Rule Interchange Format* o formato de intercambio de reglas, para expresar en *XML* reglas que pueden ejecutar los ordenadores.
- *SWRL, Semantic Web Rule Language* o lenguaje de reglas de la Web Semántica, utilizado para expresar reglas y lógica, combinando los dialectos de *OWL* como *OWL DL (OWL Description Logic*, lógica de descripción de *OWL*) y *OWL*

*Lite*, una versión reducida de *OWL*.

- *RuleML*, *Rule Markup Language* o lenguaje de marcas de reglas, definido por la *Rule Markup Initiative*, como un lenguaje para expresar reglas hacia adelante y hacia atrás para la deducción, la reescritura y posteriores tareas de transformación. Se utiliza en *MathML* (*Math Markup Language*, lenguaje de marcas para expresar las matemáticas), *DAML* (*DARPA Agent Markup Language*, lenguaje de marcas del agente *DARPA*) o *XSLT* (*eXtensible Stylesheet Language Transformations*), lenguaje de transformaciones de hojas de estilo extensibles, para transformar documentos *XML*.
- La encriptación y la firma segura es importante para asegurar y verificar que las declaraciones de la Web Semántica provienen de una fuente confiable o para firmar digitalmente las declaraciones *RDF*. La confianza de las declaraciones derivadas se realizará verificando que las premisas proviene de una fuente de confianza y añadiéndolo en la relación de dependencia de la nueva información derivada a través de la lógica formal.
- La interfaz es la capa final que permitirá utilizar fácilmente a los usuarios las aplicaciones de la Web Semántica.

### 9.3. La Web Social

La Web Semántica puede considerarse un ecosistema de datos -estructurados- donde el valor se origina por la integración de los datos que proceden de fuentes diversas, lo que definiría una *Web Semántica Social* [124], aquella donde la inteligencia colectiva de los sistemas de conocimiento colectivo de la *Web Social* (redes sociales) se relaciona con la representación del conocimiento y las técnicas de razonamiento de la Web Semántica

[125]

Los datos producidos por los usuarios (*User Generated Content* o contenido generado por lxs usuarixs) en distintas redes sociales podrían crear una *Federated Social Web* o *Web Social Federada* [192], uno de cuyos ejemplos fue el servicio de *microblogging* o micropublicación web *Identi.ca*. *Identi.ca* surgió como una alternativa a *Twitter*, pero por diversas cuestiones sus creadores decidieron renombrarlo como *Status.net* y después bautizarlo y relanzarlo de nuevo como *Pump.io*. El adjetivo *federada* pone el énfasis en la interconexión de las redes para que lxs usuarixs de una pueden relacionarse con lxs de otras, bien compartiendo la información de sus perfiles o bien intercambiándose contenidos propios o ajenos. Según la estructura de redes de Baran [12] se consideraría una red descentralizada [170].

El Consorcio *W3C* desarrolló estas ideas en el *Social Web Incubator Group Charter* [238] y en *Federated Social Web Incubator Group* [192]. *Incubator Activity* finalizó sus actividades en 2012 y los frutos de los grupos anteriores se concretaron en 2014 en la constitución de un grupo oficial de *W3C Social Web Working Group* [238] para el desarrollo de estándares que puedan integrar las aplicaciones sociales con la plataforma *Web Abierta*: vocabularios para aplicaciones, flujos de actividad o protocolos para consolidar el estado de la actividad [237]. Además, se creó el grupo *Social Interest Group* [236] para coordinar las acciones relativas en el *W3C* y formular una estrategia que permita el desarrollo de iniciativas de negocio y de federación de redes sociales.

La *Web Social* tendría estas características [192]:

1. Identidad (*identity*) o identificación única (*uniqueness*), no puede haber dos identidades iguales, que podrían crearse a partir del dominio de la red social favorita. Por ejemplo, *Fulano* en *Redsocial.net* podría ser *fulano@redsocial.net*, por lo que no podría haber otros *fulano* en *redsocial.net* pero sí en otras redes sociales. También hay iniciativas como *OpenID* que se presentan como un sistema de identi-

ficación única en cualquier tipo de red social [109], siempre que la red social soporte *OpenID* [122]. *Mozilla* desarrolló por un tiempo *BrowserID* [162] y luego *Persona*, sin conseguir los objetivos propuestos [166],

2. Perfil (*profile*), toda la información concerniente a lxs usuarixs. El sistema pionero *FOAF* sigue siendo una referencia [59] y se ha actualizado a través del proyecto *schema.org*, impulsado por *Google*, *Yahoo!*, *Bing* o *Yandex* para dotar de datos semánticos al contenido web. Otro proyecto que abordaba el perfil de usuarix fue *hCard* [71].
3. Relaciones (*relations*), parte fundamental de las redes sociales, saber o mostrar qué relaciones tienen lxs usuarixs y de qué tipo son, donde cada red utiliza una nomenclatura distinta.
4. Compartir archivos o *sharing*: fotos, audios, enlaces, vídeos o textos. Se pueden calificar de distintas formas, comentar y volver a compartir.
5. Eventos, relacionado con la agenda, el calendario y la asistencia a eventos, donde se puede confirmar la asistencia y aparecer en el evento.
6. Actividad, es el registro de acciones públicas realizadas por lxs usuarixs. Prodro-mou realizó el protocolo *Ostatus* para la actualización en tiempo real de lxs distintxs usuarixs de las redes sociales, pero solo se implementó en *status.net* e *identi.ca*
7. Mensajes, el uso de mensajería instantánea ha sido una de las aplicaciones más utilizadas por lxs usuarixs de redes sociales y ha desbordado a la Web para convertirse en un medio en sí, como es el caso de *Whatsap*.
8. Grupos, especie de mensajería entre usuarixs, se realiza una comunicación de uno

a muchxs, privada o pública.

9. Búsquedas, referidas a la información de los usuarixs, de cara a favorecer el encuentro con otrxs usuarixs.
10. *API*, la interfaz de comunicación entre máquinas con la que otrxs usuarixs o redes sociales pueden interactuar. Al principio estaban abiertas, pero en la actualidad exigen autenticación y tienen limitadas las peticiones de consulta de datos.
11. Portabilidad de los datos, algo que las redes sociales deberían permitir, bien para usar los datos en otras redes [110] o para su descarga. Los datos propios son una buena fuente de datos para experimentar en diversas áreas como el análisis de redes, minería de texto o técnicas de procesado del lenguaje natural.

*Facebook* ha publicado el protocolo *Open Graph* que permite que un sitio web cualquiera disponga de algunas de las posibilidades que la propia web de *Facebook* ofrece, por lo que aporta información semántica al documento *HTML* que *Facebook* entiende. *Google* desarrolló *Social Graph API* en un sentido parecido pero ha dejado de anunciarlo. *Open Graph* se basa en la versión inicial del protocolo *RDFa*, por lo que utiliza los elementos meta en la cabecera head del documento. Las cuatro propiedades mínimas del elemento meta que requiere son:

- `og:title`, título del objeto.
- `og:type`, tipo de objeto (imagen, audio, vídeo...)
- `og:image`, *URL* de una imagen que pueda representar el objeto en el gráfico.
- `og:url`, *URL* para el objeto.

*Facebook* se refiere a objetos y no a documentos dado que en los documentos *HTML*

puede haber muchos objetos distintos, muchas agrupaciones de datos. En el primer taller internacional de *Social Network Objects* de 2011 se referían a *objetos* de las redes sociales [1], considerando al objeto como una unidad de información, lo que la *Web 1.0* llamaría *nodo*, un contenido independiente.

## 9.4. XML

*XML* es un estándar *W3C* y un metalenguaje para la creación de esquemas de lenguajes de marcas *XML*, muy relacionado con las bases de datos dado que permite crear etiquetas específicas para que los datos se pueden procesar de distintas maneras, bien recuperándolos o bien mezclándolos con otros datos o contrastándolos. Si *HTML* es un lenguaje de estructuración de documentos de texto, *XML* es un lenguaje de datos de cualquier tipo de documento.

Se puede utilizar *HTML* para la estructura del documento y los contenidos disponerlos en documentos *XML*, de tal forma que si cambian los datos se cambia el documento, sin necesidad de tocar su estructura. De la presentación de los contenidos se encarga *CSS* y del acceso a los datos se puede realizar con *JS*.

*XML* ha sido inspirador para otros lenguajes muy conocidos de la web, donde destacan, entre otros:

- *XHTML*, el lenguaje que aúnaba *HTML* con *XML*.
- *WSDL Web Services Description Language*, lenguaje de descripción de servicios web.
- *WML, Wireless Markup Language* o lenguaje de marcas para dispositivos inalám-

bricos, una especie de *HTML* enfocado a dispositivos móviles que se comunicaban en red con *WAP* (*Wireless Application Protocol* o protocolo de aplicaciones inalámbricas), estándar de protocolo para comunicaciones inalámbricas en el año 2000, muy limitado.

- *RSS*, el estándar de la sindicación de contenidos web.
- *RDF*, para la estructuración de contenidos web.
- *OWL*, para ontologías web.
- *SMIL*, *Synchronized Multimedia Integration Language*, lenguaje para la integración de contenido multimedia sincronizado.

Entre las principales diferencias de *XML* y *SGML* destacan:

- Para procesar documentos *XML* no es necesaria una *DTD*.
- Los modelos de documento pueden incluir instrucciones de procesamiento.
- En los modelos de contenido de las declaraciones de elementos no se puede emplear el conector *&*, no se permite utilizar *exception* y no se utilizan los indicadores de minimización.
- Se admite el uso generalizado de elementos vacíos.
- Se emplean las secciones *CDATA* y las *notation declarations*.
- Los espacios en blanco, cambios de línea o saltos de carro son significativos.
- *XML* usa el lenguaje *EBNF* para expresar la sintaxis.

Diferencias *XML* y *HTML*:

- Siempre hay etiqueta de principio y final.
- Se permiten etiquetas vacías.
- Los valores de los atributos han de ir entre comillas siempre.
- Distingue entre mayúsculas y minúsculas (*case sensitive*).
- Todos los nombres de etiquetas y atributos deben empezar con un carácter alfabético no acentuado o con el signo de subrayado y no pueden contener ningún espacio.
- Se pueden preservar los espacios en blanco con el atributo `xml:space`, *HTML* los reduce a uno.
- La anidación de elementos debe ser correcta.
- En la cabecera de un documento *XML* se declara el tipo de *XML*, la codificación y una *DTD* entre símbolos de cierre de interrogación ?, para que el analizador entienda que ahí se recogen las instrucciones de procesamiento:

```
<?xml version= 1 .0 encoding= ISO -8859-1 standalone=  
yes ?>
```

Donde `standalone` igual a `yes` indica al documento el uso de una *DTD* interna.

El documento *XML* está formado por los bloques *elements* (elementos), *tags* (etiquetas), *attributes* (atributos), *entities* (entidades), *PCDATA* y *CDATA*.

- El bloque *elementos* es el principal, cuenta con un elemento raíz que no puede duplicarse, el primero que se abre y el último que se cierra. El conjunto de elementos forma un árbol del documento:

```
1 <?xml version= 1 .0 encoding= ISO -8859-1 standalone=  
   yes ?>  
2 <mi-documento >  
3 </mi-documento >
```

- Los elementos circundados por < y > componen la etiqueta de inicio. En la etiqueta de cierre se añade una barra antes del comienzo del elemento.
- Los *atributos* añaden información al elemento y se sitúan en la etiqueta de inicio del elemento y se estructuran como pares *nombre-valor*. separados por el símbolo de igual declarativo. El valor se escribe entrecomillado.
- *PCDATA*, donde van los caracteres que se identifican como propios del contenido.
- *CDATA*, donde van los caracteres que no se interpretan.
- Las *entidades* se utilizan para sustituir a caracteres que por formar parte de las etiquetas o tener otros significados, no pueden emplearse escritas y se transforman en entidades que el analizador entiende.

Entre las características de la sintaxis de *XML*, destacan:

- Los comentarios se marcan como en HTML, con <!-- -->
- Los nombres de los elementos pueden contener letras, números y tres signos de puntuación: guión -, guión bajo \_ y punto .. No pueden empezar por un número o un signo de puntuación (excepto el guión bajo), y tampoco por las letras

*xml*. No pueden contener espacios.

Para evitar conflictos de nombres y para identificar los elementos se utiliza *XML namespaces* o espacios de nombre *XML*, lo que ofrece un vocabulario que se puede utilizar en el documento. Así, si se utilizan dos nombres de elementos iguales en documentos diferentes de distinto propósito, cada elemento se identifica por su *namespace*, que se declara como propiedad del elemento, donde el nombre está compuesto por *xmlns* y el valor por la *URI*, aunque el nombre también puede indicar el nombre del elemento al que se refiere, separado del nombre por dos puntos.

La declaración del *DOCTYPE* ha de contener un *FPI* (*formal public identifier*, identificador público formal) y la localización del *DTD* que describe el vocabulario. La localización es imprescindible si se utiliza el identificador *PUBLIC*, pero no es necesario si se utiliza *SYSTEM* (sistema, local).

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE wsbib PUBLIC "-//ADOLFLOW//DTD wsbib//ES" "wsbib.dtd
   "/>
3 <wsbib xmlns="http://sindominio.net/adolflow/tesis/wsbib">
4   <tecnologia>
5     <nombre atributo="valor">RDF</nombre>
6     <lanzamiento atributo="valor">RDF</lanzamiento>
7     <nsi:elemento-nsi xmlns:nsi="http://es.wikipedia.org/wiki/
      Espacio_de_nombres_XML/ns" nsi:atributo="valor">Hola, Mundo</
      nsi:elemento-nsi>
8   </tecnologia>
9 </wsbib>
```

Con *XMLns* se permiten identificadores para las nociones abstractas de vocabularios y espacios de nombre, sin enlazar directamente los identificadores con las implementaciones técnicas, como ocurre con *DTD* o *schema*, que definen o fuerzan lo que son. Los identificadores no son *FPI* como los usados en declaraciones *DTD* sino referencias *URI*. Se pueden aplicar a cada elemento o atributo del documento.

El identificador del *xmlns* es la cadena <http://sindominio.net/adolflow/tesis/wsbib>, en cuya *URL* no necesita haber un documento, es solo una dirección, una etiqueta. Aunque parezca un enlace, no se utiliza en ese sentido sino como identificador que da un indicio sobre su propiedad, con su dominio, como sí hace *RDDL* (*Resource Directory Description Language* o lenguaje de descripción de recursos del directorio). Lo importante en *xmlns* es utilizar un dominio propio y la coherencia, no utilizar esa referencia *URI* para identificar cosas distintas.

Del ejemplo anterior, el elemento *tecnología* se ve afectado por el *xmlns* señalado y también sus elementos hijo, a no ser que provean o bien los elementos o bien los atributos su propio espacio de nombres y lo sobrescriban.

Otro objetivo de los espacios de nombre de *XML* es el de ofrecer una forma de mezclar elementos y atributos de diferentes espacios de nombre en un mismo documento. Así, el atributo *atributo* puede pertenecer a dos vocabularios distintos, siempre que aparezcan identificados en su propio espacio de nombres.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <bibliografia xmlns="http://sindominio.net/adolflow/tesis/
   bibliografia">
3   <libro id="abril1996teoria" available="true">
4     <isbn>972209394</isbn>
5     <titulo>Teoría general de la información</titulo>
6     <autoria id="abrilGonzalo" xmlns="http://sindominio.net/tesis/
   autoria">
7       <nombre>Gonzalo Abril</nombre>
8       <titulo>Catedrático</titulo>
9     </autoria>
10  </libro>
11 </bibliografia>
```

Se emplea dos veces el elemento *titulo*, la primera se refiere al título de un libro de la bibliografía y la segunda se refiere al título que ostenta el autor del libro. Para no escribir

tantas veces el espacio de nombres se utiliza un prefijo que se aplica a los nombres y atributos de los elementos para identificar sus espacios de nombres. El espacio de nombres declarado con el atributo *xmlns* se llama *default namespace* porque se asigna a elementos que no tienen prefijo.

En el ejemplo anterior, se pueden especificar los espacios de nombre como atributo y añadir un prefijo a uno o a ambos. En este ejemplo se muestra solo en uno para observar las diferencias:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <bibliografia
3 xmlns="http://sindominio.net/adolflow/tesis/bibliografia"
4 xmlns:aa="http://sindominio.net/tesis/autoria"
5 xmlns:dc="http://purl.org/dc/elements/1.1/" >
6 <libro id="abril1996teoria" available="true">
7   <isbn>972209394</isbn>
8   <aa:autoria id="abrilGonzalo">
9     <aa:nombre>Gonzalo Abril</aa:nombre>
10    <aa:titulo>catedratico</aa:titulo>
11   </aa:autoria>
12   <dc:publisher>Editorial Ctedra</dc:publisher>
13 </libro>
14 </bibliografia>
```

Las propiedades *id* y *available* no están asociadas con ningún namespace porque se utilizan para dar información sobre el documento, no sobre los datos. Los espacios de nombre *XML* permiten el desarrollo de vocabularios independientes que se pueden utilizar para construir bloques de datos. Las aplicaciones ignoran los nombres de elementos y atributos que no entienden, por lo que se pueden realizar mejoras o incorporaciones de espacios de nombres en el futuro sin que afecte a lo demás.

Todo el texto dentro de un documento *XML* se analiza (*PCDATA*) y el texto incluido en *CDATA* no se analizará. Esto resulta de utilidad para incluir *JS* o cualquier otro

lenguaje que contenga los caracteres >, < o &.

```
1 <![CDATA[
2 Texto que no se analiza
3 ]]>
```

Dentro de *CDATA* no puede haber caracteres corchete, han de sustituirse por entidades.

## 9.5. DTD XML

Para escribir documentos *XML* se puede elegir entre *DTD* o *XML Schema*. Con la *DTD*, cada archivo *XML* describe su propio formato y otras personas o entidades pueden usar esa *DTD* para leer o intercambiar datos, reutilizarlos o verificarlos, ya que se explicita qué elementos, etiquetas y contenido de las etiquetas se permite, qué estructura tienen, qué orden pueden llevar y su anidamiento. La *DTD* se incluye en el *XML* o se enlaza desde el documento:

```
1 <!DOCTYPE root-element [ element-declarations ]>
2 Ejemplo de documento XML con una DTD interna:
3 <?xml version="1.0"?>
4 <!DOCTYPE nota [
5 <!ELEMENT nota ( para , de , encabezamiento , cuerpo )>
6 <!ELEMENT para (#PCDATA)>
7 <!ELEMENT de (#PCDATA)>
8 <!ELEMENT encabezamiento (#PCDATA)>
9 <!ELEMENT cuerpo (#PCDATA)>
10 ]>
11 <nota>
12 <para>Adolfo</para>
13 <de>Gonzalo</de>
14 <encabezamiento>Recordatorio</encabezamiento>
15 <cuerpo>La presentaci n de la tesis es ma ana</cuerpo>
```

```
16 </nota >
```

Donde:

- !DOCTYPE define que el elemento *root* (raíz) de este documento es *nota*.
- !ELEMENT *nota* dice que el elemento *nota* contiene cuatro elementos: *para*, *de*, *encabezamiento* y *cuerpo*.
- !ELEMENT seguido de *para*, *de*, *encabezamiento* y *cuerpo* se consideran del #PCDATA, por lo que se analizan.

Si la declaración es externa, seguirá la siguiente sintaxis:

```
1 <!DOCTYPE root-element SYSTEM "filename">
```

El mismo documento *XML* puede referirse a una *DTD* externa:

```
1 <?xml version="1.0"? >
2 <!DOCTYPE nota SYSTEM "nota.dtd">
3 <nota >
4   <para >Adolfo </para >
5   <de >Gonzalo </de >
6   <encabezamiento >Recordatorio </encabezamiento >
7   <cuerpo >La presentaci n de la tesis es ma ana </cuerpo >
8 </nota >
```

La *DTD* enlazada *nota.dtd* contiene la misma información que antes se escribía en `<!DOCTYPE NOTA []>`. Las *DTD* presentan algunas limitaciones, no permiten definir elementos locales que sólo sean válidos dentro de otros elementos. Por ejemplo, un elemento `<encargado>` que describe a quien se encarga de una sede de la empresa y también a quien se encarga de una sección de la sede de la empresa, dado que no hay jerarquía, se tendría que utilizar una *jerarquía* basada en guiones o en el espacio

de nombres: <encargadoSede> y <encargadoSeccion> para evitar el conflicto de nombres. También ocurre que es poco flexible la definición de elementos que incluyan otros elementos no textuales. Además no es posible indicar a qué tipo de dato (número, fecha, moneda) ha de corresponder un atributo o el texto de un elemento. Para superar estas limitaciones se crearon lenguajes de esquema como *XML Schema* que permiten una descripción más compleja y funcionan como una alternativa a *DTD*.

## 9.6. RDF

*RDF* (*Resource Description Framework* o marco de descripción de recursos) pretende identificar los datos en la Web como si fueran recursos a través de *URI*, describiéndolos en términos de propiedades simples y valores que componen una sentencia o declaración, compuesta por una tripleta de elementos asemejables a la estructura gramatical *sujeto - objeto - predicado*. El conjunto de sentencias o declaración compone la descripción *RDF*. Se puede mostrar cada sentencia mediante grafos dirigidos donde cada nodo lanza arcos hacia las propiedades y los valores.

Al ser el lenguaje de la *Web de Datos*, Berners-Lee denomina a la Web como *GGG* o *Giant Global Graph* (gráfico gigante mundial) [36], inspirado por las ideas de Fitzpatrick sobre la *Web Social* [110] expresada en grafos que resalta las características de *RDF* como lenguaje capaz de representarse en grafos. De esta manera huye de las etiquetas versionadas de la Web que la consideraban *1.0*, *2.0* o *3.0*.

*RDF* surge del *W3C*, pero se nutre de trabajos previos como *PICS* (*Platform for Internet Content Selection*, plataforma para la selección de contenido en Internet) y *Dublin Core/Warwick Framework* (que actualmente vincula su desarrollo a *RDF*), así como de *MCF* o *CDF* (*Channel Definition Format*, formato de definición de canal, similar y previo a *RSS* aunque no llegó a utilizarse tanto.). *RDF* cuenta con tres aspectos de la

semántica funcional: ser un modelo de datos, una sintaxis y un esquema [159, p. 178].

*Meta Content Framework (MCF)* o *marco de metacontenido* se trata de un modelo de datos y las expresiones sintácticas que lo definen creado por Ramanathan V. Guha cuando trabajaba en *Apple Computer* y más tarde en *Netscape*, donde pretendía hacer entendibles por las máquinas -ordenadores- descripciones de conjuntos de información distribuida, describir objetos, sus atributos y las relaciones entre ellos. Está basado en el trabajo de sistemas de representación del conocimiento como *CycL*, *KRL (Knowledge Representation Language)*, lenguaje de representación del conocimiento, creado por Daniel G. Bobrow y Terry Winograd en el centro de investigación de *Xerox PARC* o *KIF (Knowledge Interchange Format)*, formato de intercambio de conocimiento, lenguaje diseñado para compartir conocimiento creado en los laboratorios de *DARPA*). *Cyc* era un proyecto de Douglas Bruce Lenat en *MCC (Microelectronics and Computer Corporation)* y *Cycorp, Inc.*, donde había colaborado Guha, que pretendía sentar las bases de la inteligencia artificial mediante la representación del conocimiento en lenguaje formal a través de una *lógica de primer orden (lógica de predicados o cálculo de predicados)*, un sistema formal para estudiar la inferencia en los lenguajes de primer orden.

En los lenguajes de primer orden los predicados son tratados como funciones en programación, donde pueden recibir argumentos para devolver valores o imágenes. Los predicados conectan una o varias expresiones para formar una oración. En lógica matemática, cuando un predicado conecta con una expresión, expresa una propiedad, mientras que si conecta dos o más expresiones, conforma relaciones. En la lógica de primer orden no se habla de propiedades o relaciones sino que estudia el modo como se habla con expresiones lingüísticas.

En la función matemática  $f(x) = 2x$

Se toma un número como argumento y devuelve otro número como valor. De la misma forma, en la lógica de primer orden, los predicados toman expresiones. Por ejemplo, en

la oración *el árbol es un manzano*, se puede transcribir como:

Manzano( árbol ) o m(a)

Se puede tomar más de un argumento, por ejemplo, *el peral y el manzano son árboles*, que se expresa:

rboles ( peral , manzano ) o (p,m)

En estos casos, *peral* y *manzano* son *constantes de individuo*, entidades lingüísticas. Las *variables de individuo* se refieren a entidades que no están determinadas, como las expresiones él, ella, esto, eso o aquello, cuyo referente varía según el contexto. Por ejemplo, la expresión *Esto está limpio* se representa con:

Limpio(x) o L(x)

Igualmente, también se formalizan relaciones. Por ejemplo, *Esto está más limpio que aquello*, sería:

L(x, y)

Y se pueden combinar constantes de individuo con variables, por ejemplo, *esto está más limpio que la casa de Mengana y Perantana*:

L(x, m, p)

Del laboratorio de *Apple* nació *Project X* que fue renombrado como *HotSauce*, una aplicación de *MCF* que permitía una visualización 3D de los contenidos de un archivo *MCF*, el cual podía ser tanto la estructura de contenidos de un sitio web como el árbol de directorios y archivos del disco duro. Se podía usar como extensión (*plug-in*) de *Netscape 3.0* y *Microsoft Explorer*, incluso *Netscape* lo incorporó a *Constellation*, el primer intento de una empresa de convertir al navegador en la pieza central del entorno de escritorio, algo que ahora recuerda a la estrategia que inició *Google* con *Google Chrome* [92].

*Constellation* funcionaba sobre dieciocho sistemas operativos diferentes al estar creado con *HTML*, *JavaScript* y *Java*, permitía su uso conectado o sin conexión, conectarse a bases de datos remotas, locales, hojas de cálculo o documentos de texto, fuentes de información externas -precedentes de *RSS*-, crear copias locales encriptadas de los datos, ver *streaming* de vídeo –el ordenador como televisión– o trabajar en lo que ahora se denomina *la nube* (*cloud computing*). Pretendía desbancar a *Windows*, pero *Microsoft* preinstaló *Internet Explorer* en *Windows* y fue el principio del fin de *Netscape*. *Netscape* se reconvirtió posteriormente, como empresa de código abierto, en *Mozilla*, liberando el software y creando *Mozilla Firefox* y *Mozilla Thunderbird*, entre otros productos e iniciativas.

En *RDF* los datos expuestos en documentos *XML* se transforman en expresiones con la forma *sujeto - predicado - objeto*, conocidas como *tripletas RDF* o *sentencias*, donde:

- Sujeto es el recurso, la entidad, lo que se está describiendo. Puede ser una *URI* o un nodo en blanco.
- Predicado es la propiedad o relación que se establece entre recursos y que aporta propiedades al recurso o entidad. Cada propiedad tiene unos valores y define un significado, qué tipos de objetos se pueden describir y relaciones entre los recursos o sujetos, con entidades de múltiples clases. Puede ser una *URI*.
- Objeto es otra entidad con la que se desea establecer la relación o el valor de la propiedad del recurso o entidad. Puede ser una *URI*, un valor literal o un nodo en blanco.
- Tanto el sujeto como el predicado son nodos del grafo. Predicado y propiedades son sinónimos.

*RDF* se basa en las *URI* para describir los términos, sus propiedades y sus valores. Una

referencia *URI* utilizada como nodo en un gráfico *RDF* identifica lo que el nodo representa. Una *URI* utilizada como predicado identifica la relación entre las cosas identificadas por los nodos que conecta. Se representa con el espacio de nombres *XML* y permite expresar un espacio o esquema inequívoco a la declaración, donde se permite utilizar varios conjuntos de elementos de metadatos de forma modular, eliminando las colisiones entre términos, y también asociar un prefijo a cada vocabulario eliminando la ambigüedad en el caso de que varios vocabularios tuviesen algún término en común.

Antes de elegir un prefijo, de cara a publicarlo, conviene consultar en *prefix.cc* si el prefijo elegido se encuentra disponible. No ha de tener más de ocho caracteres. Para estos ejemplos se utiliza el prefijo *flow*. Este es el código que ofrece *prefix.cc* en *XML* para comenzar con el vocabulario:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:flow="http://www.w3.org/2005/01/wf/flow#">
5   <rdf:Description rdf:about="">
6   </rdf:Description>
7 </rdf:RDF>
```

Donde:

- La *URI* es <http://www.w3.org/2005/01/wf/flow#>.
- Se puede realizar una validación de las tripletas *RDF* u *OWL* en el validador de *RDF* del *W3C*.
- En las propiedades o predicados se puede restringir el dominio mediante `rdfs:domain` y su rango mediante `rdfs:range`.
- Al ser relaciones y atributos, se pueden considerar propiedades del objeto, *object*

*properties*, definidas con `rdf:Property`, o propiedades del tipo de datos o *datatype properties*, definidas con `owl:DatatypeProperty`. Por ejemplo, escrito en  $N_3$ :

```

1 @PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @PREFIX owl: <http://www.w3.org/2002/07/owl#> .
3 @PREFIX flow: <http://sindominio.net/adolflow/> .
4
5 flow:adolflow flow:nacimiento "1975-10-14" ; flow:esAlumno flow:
   gonzaloAbril .
6
7 flow:nacimiento a rdf:Property .
8 flow:esAlumno a rdf:Property .
9 flow:nacimiento a owl:DatatypeProperty .
10 flow:esAlumno a owl:ObjectProperty .

```

El objeto establece el valor de la propiedad del recurso, la descripción del recurso o la relación con otra entidad, que puede ser un literal u otro recurso. Por ejemplo, si fuera un recurso, podría ser:

```

1 <rdfs:seeAlso rdf:resource="http://dbpedia.org/resource/
   Cuatro_Vientos"/>

```

Mientras que si fuera un literal, Cuatro Vientos. Son entidades de tipo `rdf:Property`. Pertenecen a las entidades, no a las clases.

Todo el grafo puede declararse como un subconjunto mediante *URI*, por ejemplo

```

1 http://sindominio.net/adolflow/dondeVivo

```

El ejemplo anterior podría tener un prefijo o espacio de nombre. escrito en  $N_3$ :

```

1 @PREFIX flow: <http://sindominio.net/adolflow> .

```

O utilizar la sintaxis *N triples*:

```
1 <http://sindominio.net/adolfoflow <http://sindominio.net/adolfoflow/
  viveEn > "Madrid" .
```

O también:

```
1 flow:adolflow flow:viveEn "Cuatro Vientos" ;
```

La primera definición de *RDF* se publica en 1999 [150] y su espacio de nombres se declara así:

```
1 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

*RDF* dispone de clases y propiedades, que son instancias de la clase. Las clases son equivalentes a expresiones *E-R*. Se definen con el predicado y el tipo de objeto:

- `rdfs:Resource`, la clase *para todo*, ya que todo lo descrito por *RDF* se considera un recurso.
- `rdfs:Class`, declara un recurso como una clase para otros recursos. Por ejemplo:

```
1 flow:Libro rdf:type rdfs:Class .
2 flow:Libro a owl:Class
3 flow:LibroBibliografia a rdfs:Class .
4 flow:LibroTesis a owl:Class .
5 flow:autoria a ej:Libro .
6 flow:paginas a ej:Libro .
```

- `rdfs:Literal`, valores literales, ya sean número enteros (*integers*) o cadenas (*strings*)
- `rdfs:Datatype`, clase para todo tipo de datos, es una instancia y una subclase de `rdfs:Class`. Cada instancia de `rdfs:Datatype` es una subclase de `rdfs:Literal`

- `rdf:XMLLiteral`, clase para los valores literales de *XML*, instancia de `rdfs:Datatype` y ambas subclase de `rdfs:Literal`
- `rdf:Property`, clase para las propiedades.

Las propiedades son instancias de la clase `rdf:Property` y describen una relación entre recursos sujeto y recursos objeto. *RDF* define cuatro propiedades para modelar sentencias [149]:

- *subject*, sujeto, identifica el recurso que describe la sentencia modelada; es decir, el valor de la propiedad *subject* es el recurso sobre el cual se hizo la sentencia original.
- *predicate*, predicado, identifica la propiedad original en la declaración modelada. El valor de la propiedad *predicate* es un recurso que representa la propiedad específica en la sentencia original.
- *object*, objeto, identifica el valor de la propiedad en la sentencia modelada. El valor de la propiedad *object* es el objeto en la sentencia original.
- *type*, tipo, describe el tipo del nuevo recurso. Todas las sentencias transformadas, de lo abstracto a lo concreto, son instancias de *RDF*, sentencias que tienen una propiedad *type* cuyo objeto es `RDF:sentencia`. La propiedad *type* se usa más comúnmente también para declarar el tipo del recurso.

Cuenta con las siguientes propiedades:

- `rdfs:domain` de un `rdf:predicate`, declara la clase del sujeto en una tripleta de la cual el segundo componente es el predicado.
- `rdfs:range` de un `rdf:predicate` declara la clase o tipo de datos del objeto en una tripleta cuyo segundo componente es el predicado.

- `rdf:type`, una propiedad usada para declarar que un recurso es una instancia de una clase.

La declaración entera se puede utilizar como predicado a través de la reificación de *RDF*.

Las jerarquías de las clases soportan herencia de una propiedad de un dominio y un rango de su clase a sus subclases, donde el dominio indica de qué clase sale la propiedad y el rango a qué clase alcanza.

En el caso del ejemplo:

```
1 flow:adolflow foaf:knows flow:gonzaloAbril
```

`foaf:knows` tiene dominio y rango `foaf:Person`:

```
1 Si X foaf:knows flow:gonzaloAbril
2 => X rdf:type foaf:Person
3 Si flow:adolflow foaf:knows Y
4 => Y rdf:type foaf:Person
```

Puede haber clases más generales y otras más específicas. Para definir la jerarquía de clases y propiedades se cuenta con `rdfs:subPropertyOf` y `rdfs:subClassOf`:

- `rdfs:subPropertyOf` es una instancia de `rdf:Property` y se usa para establecer que todos los recursos relacionados con una propiedad también se relacionan con la otra.
- `rdfs:subClassOf`, permite declarar jerarquías de clases. Por ejemplo, en *FOAF*, para decir que *toda persona es un animal*, se escribe:

```
1 foaf:Person rdfs:subClassof foaf:Animal
```

Propiedades de la utilidad:

- `rdfs:seeAlso`, instancia de `rdf:Property`, indica al recurso que puede ofrecer información adicional sobre el tema del recurso.
- `rdfs:isDefinedBy`, instancia de `rdf:Property`, indica a un recurso asunto del recurso. Esta propiedad puede usarse para indicar un vocabulario *RDF* en el que se describe el recurso.

Se han de especificar los espacios de nombre usados:

```

1 @PREFIX rdf: <http://www.w3.org/1999/02-22-rdf-syntax-ns#> .
2 @PREFIX flow: <http://sindominio.net/adolflow/ > .
3
4 flow:abril1993Teoria a flow:Libro; flow:paginas "220"; flow:
   autoria "Gonzalo Abril Curto" .
5 flow:paginas a rdf:Property .
6 flow:autoria rdf:type flow:Persona .

```

La definición de las entidades del dominio es uno de los aspectos más cruciales. Si las autorxs de los libros son personas, se puede crear la clase `flow:Persona`, como en el ejemplo anterior. Una misma entidad puede pertenecer a distintas clases, por ejemplo *Gonzalo Abril* puede pertenecer a la clase `flow:Persona` y a la clase `flow:Autor`. También es importante en el desarrollo de la ontología documentar el trabajo realizado para entender el modelo y el contexto por parte de quienes lo usen, informar del idioma o los idiomas empleados y aportar información extra de las clases y las propiedades a través de `rdfs:label` y `rdfs:comment`:

- `rdfs:label` es una instancia de `rdf:Property` que se utiliza para ofrecer una versión legible del nombre de un recurso.
- `rdfs:comment` es una instancia de `rdf:Property` para describir de forma legible un recurso.

```

1 flow:Libro rdfs:label "Libro"@es ;

```

```
2 rdfs:label "Book"@en ;  
3 rdfs:comment "Un libro , subclase de la clase bibliografía"@es ;  
4 rdf:comment "A book , subclass of bibliography"@en ;
```

Se pueden utilizar vocabularios que ya modelan otros dominios:

- *FOAF, Friend of a friend*, relaciones personales.
- *DC, Dublin Core*, para aportar metadatos de biblioteca.
- *GN, GeoNames*, para nombres de localización geográfica.
- *SDMX, Statistical Data and Metadata eXchange*, de información estadística.
- *XBRL, eXtensible Business Report Language*, lenguaje de información financiera y económica extensible.
- *Datacube/Scovo, cubo de datos* o información multidimensional y *Statistical Core Vocabulary* o vocabulario estadístico.
- *TIME, tiempo*, para fechas y periodos.
- *BIO*, para información biográfica.
- *FRBR, Functional Requirements for Bibliographic Records*, registros funcionales de los registros bibliográficos, para información bibliográfica

*RDF* no describe la semántica o el significado de los datos, para ello se ha de utilizar *RDF Schema* (esquema *RDF*) y *OWL*. Si *XML* sirve para modelar datos, *RDF* especifica metadatos [149]. En el mismo sentido, "*RDF* es un metamodelo de metadatos, una infraestructura de descripción de recursos que trata de contener otros esquemas

de metadatos"[159, p. 176], mientras que *XML* es un metalenguaje que permite definir otros lenguajes de estructuración de contenidos.

El modelo *recurso - propiedad - valor/atributo* se asemeja al esquema o diagrama *entidad-relación E-R* que se utiliza en las bases de datos relacionales, donde las entidades son cosas u objetos, donde los atributos definen las características de la entidad y la relación es el vínculo de distintas entidades con algún o algunos atributos. Las relaciones son de uno a uno, uno a varios o varios a uno y varios a varios.

En el modelo realacional (lógico):

- La entidad es objeto real o abstracto del cual se puede recoger información
- El atributo es la propiedad asociada a una entidad
- La relación es la correspondencia entre dos o más entidades con una función determinada.

Entre las características del modelo entidad-relación *E-R* destacan que:

- Refleja la existencia de datos, no lo que se hace con ellos
- No se orienta a aplicaciones específicas.
- Define elementos conceptuales.

También guarda similitud con el diseño orientado a objetos, donde los recursos serían los objetos y las propiedades los ejemplos de variables. El uso de metadatos para describir recursos se basa en considerar al documento como un objeto, *document-like object*, tal como sugirieron en *DCMI* para referirse a documentos digitales o recursos de información [159].

*RDF* también aporta un nuevo concepto de base de datos, muy ligada a las bases de datos relacionales: la base de datos gráfica. En sistemas de almacenamiento de la información, como es la Web, existe una relación entre los objetos de datos que se puede representar de distintas maneras. En el caso de las bases de datos relacionales, hay tablas de datos o nodos de datos más importantes que otros; en el caso de las bases de datos jerárquicas, como un documento *HTML* o *XML*, se produce una estructura arbórea. En el paradigma de base de datos *RDF*, un concepto se relaciona con otro sin jerarquía, se relacionan recursos unos con otros, como si se representaran los enlaces de los documentos *HTML*. El alcance va más allá de cualquier otra base de datos ya que se trata de conectar los datos de la Web.

Por tanto, la Web Semántica es una extensión de la Web, una de las capas donde se produce el enlace entre los datos de los documentos [46, p. 20]. No se trata de tratar con los documentos o sus metadatos o clasificar las páginas web de cara a buscarlas mejor, sino de relacionar datos. Comparándolo con una base de datos, el esquema de *OWL* es más extenso y complejo, la información menos estructurada pero en cambio propone una conceptualización de un dominio concreto que pueden distribuirse a través de la Web y conectarse con otras ontologías de otros dominios. Al tratarse de una estructura estándar, abierta, extensible y escalable, cualquiera puede participar, aunque respondan a distintos propósitos y procesos, desde una conceptualización básica hasta un gran consenso sobre determinado dominio [46, p. 28].

## 9.7. Notation 3 Logic

*Notation 3 Logic* o *N<sub>3</sub> Logic*, *lógica N<sub>3</sub>* es un lenguaje *RDF* para la Web Semántica, otra forma de escribir *RDF*. La sintaxis *N<sub>3</sub>* no permite el uso de prefijos. La tripleta siempre termina en punto . . Los nombres de clases se escriben con caracteres alfanuméricos, con la primera letra de cada palabra en mayúsculas y sin espacios:

```
1 flow:LibroBibliografia a rdfs:Class .
```

En los nombres de entidades, la primera palabra va en minúsculas:

```
1 flow:abril1993Teoria a flow:LibroBibliografia .
```

Múltiples recursos se separan por comas:

```
1 flow:abril1993Teoria a flow:LibroBibliografia , flow:LibroTesis ,
   flow:LibroPeriodismo flow:Libro
```

Múltiple declaraciones para un mismo recurso se pueden separar por punto y coma:

```
1 flow:abril1993Teoria a flow:LibroBibliografia , flow:Libro; flow:
   autoria "Gonzalo Abril Curto"; flow:editorial flow:Ctedra .
```

Lo cual es igual que decir:

```
1 flow:abril1993Teoria a flow:LibroBibliografia .
2 flow:abril1993Teoria a flow:Libro .
3 flow:abril1993Teoria flow:autoria "Gonzalo Abril Curto" .
4 flow:abril1993Teoria flow:editorial "Ctedra" .
```

## 9.8. Ontologías OWL

En los modelos semánticos, son importantes las ontologías y los vocabularios. Los vocabularios son un listado de términos de significado definido que es consistente en diferentes contextos, mientras que las ontologías permiten definir relaciones contextuales detras de un determinado vocabulario. La ontología es la piedra angular de la definición del conocimiento de un dominio. *OWL* es la propuesta de sintaxis formal para la definición de ontologías de la Web Semántica, una extensión de *RDFS*. Vocabularios y

ontologías, Ambas son recomendaciones de *W<sub>3</sub>C* y son la capa por encima de *RDF* en el esquema de capas de la Web Semántica..

El término ontología se ha visto afectado por su utilización en la Web Semántica. Se considera ontología como la especificación explícita de una conceptualización, una descripción formal de los conceptos y de sus relaciones [124] o como una especificación formal de una conceptualización compartida [53]. Si se suman las dos, se considera la ontología como una especificación explícita y formal de una conceptualización compartida [86], donde:

- Conceptualización se refiere a un modelo abstracto de algún fenómeno del mundo construido mediante la identificación de los conceptos relevantes a ese fenómeno o dominio (del conocimiento).
- Explícito, porque los conceptos utilizados y las restricciones para su uso están claramente definidos.
- Formal, porque deben ser comprensibles en su forma.
- Compartida, porque existe algún grado de consenso.

En la ontología se presentan elementos y relaciones entre los mismos, por lo que añade a la taxonomía relaciones, razonamiento y semántica [159, p. 217]

Se pueden clasificar las ontologías según el dominio y propósito que cubren:

- Ontologías de nivel superior, para conceptos genéricos, como *CyC* o *DCMI*.
- Ontologías generales, para conceptos como el tiempo, el espacio, eventos. Se pueden utilizar en diversos dominios, como *FOAF*.

- Ontologías de dominio, específicas de un dominio o para aplicaciones concretas.

En esta ontología se incluyen dos nuevos espacios de nombre en la cabecera del documento *RDF: RDFS* y *OWL*. Una ontología no tiene por que incluir cabecera pero es información útil que ayudará a entender el contenido de la ontología. En la cabecera está incluido el título `dc:title`, la descripción `dc:description` de la ontología, la autoría `dc:author` e información sobre la versión y fecha de la última actualización. También en la cabecera es el lugar para importar otras ontologías:

```
1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4   xmlns:owl="http://www.w3.org/2002/07/owl#"
5   xmlns:dc="http://purl.org/dc/elements/1.1/" >
6
7   <!-- OWL Header Ejemplo -->
8   <owl:Ontology rdf:about="http://www.sindominio.net/adolflow/
9     tesis/citas">
10     <dc:title>Referencias bibliogr ficas de la tesis</dc:title >
11     <dc:description>Un ejemplo de ontolog a con la referencias
12     bibliogr ficas empleadas en el trabajo de tesis</dc:
13     description >
14     <dc:author>Adolfo Ant n Bravo</dc:author >
15     <owl:versionInfo rdf:datatype="xsd:string">0.1, 20150825</
16     owl:versionInfo >
17   </owl:Ontology >
18
19   <!-- OWL Class Definition Ejemplo -->
20   <owl:Class rdf:about="http://www.sindominio.net/adolflow/tesis
21     /citas#libros">
22     <rdfs:label>Libros</rdfs:label >
23     <rdfs:comment>La clase de referencias bibliogr ficas que
24     son libros.</rdfs:comment >
25   </owl:Class >
26 </rdf:RDF>
```

Para clasificar cosas en términos de semántica, *OWL* utiliza clases, subclases e instancias individuales. Una clase es una clasificación de instancias individuales, luego las instancias individuales son miembros de una clase y se consideran *class extension* o extensión de clase:

```
1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:owl="http://www.w3.org/2002/07/owl#"
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   xmlns:citas="http://www.sindominio.net/adolflow/tesis/citas#">
6
7   <!-- Se omite la cabecera -->
8
9   <!-- Se omiten las clases -->
10
11  <!-- Se define la propiedad autor a -->
12  <owl:DatatypeProperty rdf:about="http://www.sindominio.net/
13    adolflow/tesis/citas#author"/>
14
15  <!-- Se define la propiedad titulo -->
16  <owl:ObjectProperty rdf:about="http://www.sindominio.net/
17    adolflow/tesis/citas#title"/>
18
19  <!-- Se define la instancia de clase Gonzalo Abril -->
20  <rdf:Description rdf:about="http://www.sindominio.net/adolflow/
21    tesis/citas#gonzaloAbril">
22
23    <!-- Gonzalo Abril es una instancia individual de la clase
24    autor -->
25    <rdf:type rdf:resource="http://www.sindominio.net/adolflow/
26      tesis/citas#author"/>
27
28    <!-- Gonzalo Abril es parte de la Universidad Complutense
29    -->
30    <citas:universidad>Universidad Complutense</citas:
31      universidad>
```

```
26 <!-- Gonzalo Abril es un autor tan destacado como Umberto
    Eco -->
27 < citas : tanImportanteComo rdf:resource="http://www.sindominio
    .net/adolflow/tesis/citas#umbertoEco"/>
28
29 </rdf:Description >
30
31 <!-- Se define la instancia de clase Umberto Eco -->
32 <rdf:Description rdf:about="http://www.sindominio.net/adolflow
    /tesis/citas#umbertoEco">
33
34 <!-- Umberto Eco es una instancia individual de la clase
    author -->
35 <rdf:type rdf:resource="http://www.sindominio.net/adolflow/
    tesis/citas#author"/>
36
37 <!-- Umberto Eco es parte de la Universidad de Bolonia -->
38 < citas : universidad >Universidad de Bolonia </ citas : universidad
    >
39
40 <!-- Umberto Eco es un autor tan destacado como Gonzalo
    Abril -->
41 < citas : tanImportanteComo rdf:resource="http://www.sindominio
    .net/adolflow/tesis/citas#gonzaloAbril"/>
42
43 </rdf:Description >
44
45 </rdf:RDF>
```

Se ha definido una nueva *individual* o instancia de la clase libros que representa a Gonzalo Abril, y la instancia Umberto Eco de la misma clase. La clase es la descripción formal de una entidad del universo o dominio que se representa. Puede contar con subclases y se componen de instancias. También puede haber clases abstractas que no permiten instancias y se utilizan para agrupar conceptos e introducir jerarquía. Las instancias representan objetos concretos del dominio, pertenecientes a una clase. La co-

lección de instancias conforma la base de hechos (base de datos o base del conocimiento) del modelo.

Las instancias y las clases tienen propiedades (atributos, *slot*) que contienen valores, que pueden ser cadenas de caracteres o *strings*, números y también otras clases o instancias. Las propiedades tienen restricciones denominadas facetas, que son propiedades de las propiedades, y que pueden indicar si son obligatorias o no, por ejemplo.

*OWL* utiliza buena parte de la terminología de la lógica de primer orden (*First Order Logic* o *FOL*) a pesar de ser una implementación de la lógica descriptiva (*Description Logic* o *DL*). Si *FOL* utiliza clases, propiedades o predicados y objetos, *DL* utiliza conceptos, roles e individuales. El enfoque de *DL* se ha sugerido para la gestión de versión de ontologías, debido a las características de la Web y su crecimiento y distintas versiones del contenido que se pueden tener.

Siguiendo el ejemplo anterior se puede realizar un documento *RDF*:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
4 </rdf:RDF>
```

Las dos primeras líneas son las fundamentales de un documento *RDF*. En la primera se declara el comienzo del documento con el elemento `rdf:RDF`, mientras que en la segunda con su atributo `xmlns` se explicita el espacio de nombres empleado.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   xmlns:referencias="http://sindominio.net/adolflow/tesis-referencias/">
6 <rdf:Description rdf:about="http://sindominio.net/adolflow/
   tesis/bibliografia#referencia">
```

```
7 <referencia:apellido>Abril</referencia:apellido>  
8 <referencia:categoría red:resource="http://sindominio.net/  
   adolflow/tesis/bibliografia/categoría#catedratico"/>  
9 <dc:author>Gonzalo Abril Curto</dc:title>  
10 <dc:publisher>Ctedra</dc:publisher>  
11 </rdf:Description>  
12 </rdf:RDF>
```

Para conocer la *URI* del predicado `referencia:apellido`, se sustituye `referencia` por la *URI* del espacio de nombres y se obtiene la siguiente *URI*:

```
1 http://sindominio.net/adolflow/tesis-referencias#apellido
```

Por eso se dice que las *URI* con espacio de nombres *XML* en *RDF* se emplean para distinguir propiedades que utilicen la misma etiqueta. Los esquemas *RDF* son parecidos a las *DTD* de *XML*, con su propia sintaxis. Si una *DTD* de *XML* proporciona las restricciones específicas en la estructura de un documento, en un esquema *RDF* se proporciona información sobre la interpretación dada en un modelo de datos *RDF*. *RDFS* define clases de jerarquías y restricciones de uso para esas clases, y proporciona un mecanismo para declarar las propiedades de los elementos y las relaciones entre las propiedades y los recursos, aportando una semántica, una forma de crear vocabularios.

Al incluir semántica en los datos, esta semántica se puede extender por otros sitios de la Web que compartan estos datos. A estos lugares se les denomina *dominios del conocimiento* o dominio, y se puede ramificar esa semántica a lo largo de otros dominios. Esto es lo que permite la relación de datos de la Web.

## 9.9. Anotaciones y *folksonomías*

Las anotaciones se refieren a otra forma de interactuar con el contenido, bien el momento de creación o de recuperación. Se pueden realizar anotaciones semánticas basadas en ontologías, información sobre las entidades y conceptos de una ontología que aparecen en el texto, estableciendo una relación entre el contenido y el identificador conceptual *URI*. Consideran la Web como una gran base de datos, pero también como una infraestructura hipertextual donde autores y agentes de usuario pueden aportar anotaciones [68]. Las anotaciones se pueden incrustar a través de *XML* o *RDF* con herramientas de anotación de autor o bien almacenar en bases de datos externas *RDF* a través de herramientas de anotación externa. Se guardan como objetos o propiedades de un concepto de una ontología [169].

Se considera a la semántica ontológica la base para la anotación de los recursos web. Dentro del campo de la inteligencia artificial se ha promovido el concepto de *ontological semantics* (ontologías semánticas) de Niremburg [48] que sitúan a la ontología en el centro del proceso de conocimiento: razona con el conocimiento que se deriva a partir de los textos en lenguaje natural y a su vez es el eje central para generar texto en lenguaje natural basados en representaciones de su significado.

La semántica ontológica estudia el significado del lenguaje humano o lenguaje natural y realiza una aproximación al procesamiento de lenguaje natural que utiliza una ontología para extraer y representar el significado de los textos, al razonar con el conocimiento de los mismos [48]. Las ontologías semánticas se pueden adaptar, modificar y/o crecer para cumplir con las necesidades de cualquier aplicación *NLP* (*natural language processing*, procesado de lenguaje natural), independientemente de su orientación teórica y práctica.

En España se desarrolló el proyecto *RODA* (Red de Conocimiento Descentralizada a través de Anotaciones) cuyo objetivo era validar herramientas de anotación que faci-

litasen la creación de repositorios de conocimientos claves para probar su efectividad en distintas áreas, entornos y perfiles de usuario. Según Idioa Murua, investigadora del proyecto RODA, "las herramientas de anotación externa son útiles para realizar comentarios personales sobre páginas web, para mantener discusiones en grupo sobre estas páginas, para compartir información, como herramientas de ayuda para el mantenimiento de *bookmarks* compartidos, etc"[169]. Probaron varias herramientas, entre las que destacan el conjunto de aplicaciones surgidas del *W3C* para uso en el lado del cliente y también para el lado del servidor, con el navegador *Amaya* con soporte para anotaciones; *Annozilla*, extensión para *Mozilla*; *Snuffkin*, extensión para *Internet Explorer*; el servidor de aplicaciones basado en *ZOPE Zannot (Zope Annotation Server)*, servidor de aplicaciones basado en *Zope*; y *Annotea Server*, servidor de aplicaciones para funcionar con el servidor web *Apache*.

También probaron *COHSE (The Conceptual Open Hypermedia Project* o proyecto conceptual del hipermedia abierto) iniciativa de investigación sobre métodos que mejoraran la calidad, consistencia y amplitud de los documentos en el proceso de creación y también de recuperación. Utilizaba un servicio de razonamiento de ontologías para representar un modelo conceptual de términos documentales y sus relaciones, un servicio abierto de enlaces hipermedia y la integración de estos para poder enlazar documentos según los metadatos que describían sus contenidos [118].

Una forma diferente de aportar información a un dominio es a través de las taxonomías y de su variante *Web 2.0* las folksonomías [226]. La taxonomía (de taxos ordenamiento y nomos norma o regla) es la ciencia de la clasificación, un sistema de clasificación jerárquica de taxones y categorías taxonómicas, que para realizarlas exige conocer al detalle la materia, por lo que está normalmente reservado a personas expertas.

Jugando con el concepto de taxonomía surge la folksonomía de su fusión con *folk* o popular, lo que sugiere tanto una vulgarización del término como de la práctica de indexación, al hacerlo cualquiera, de forma colaborativa, por medio de etiquetas *tags* cua-

lesquiera, sin jerarquías ni relaciones de parentesco predeterminadas. Por esta razón se asocia con el término *etiquetar*. Si el etiquetado es totalmente abierto, se considera una *folksonomía ancha*, mientras que si está limitado a un grupo de personas se considera *estrecha*.

Relacionado con folksonomía se encuentra el concepto de *collabulary* o *colabulario*, una especie de vocabulario colaborativo para etiquetar y explicar el contenido web. A diferencia de la *folksonomía*, el *colabulario* examina las palabras de forma automática buscando sinónimos con el objetivo de mantener la coherencia.

## 9.10. RSS

RSS es la sigla de *Really Simple Syndication* o *sindicación -de contenidos- muy simple*, en su versión 2.0, pero también *RDF Site Summary* o *sumario del sitio en RDF*, en su versión 0.9 y 1.0 y *Rich Site Summary* en su versión 0.91. RSS se utiliza para syndicar o compartir actualización de contenido de páginas o sitios web. Para leer esta información se utiliza un lector o agregador de RSS y permite suscripción. A esta práctica se la conoce también como redifusión o *sindicación web* y la URL suele contener la palabra *feed* de *alimento*. Por ejemplo, el archivo RSS de <http://infotics.es> se encuentra en <http://infotics.es/feed>:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <rss version="2.0"
3   xmlns:content="http://purl.org/rss/1.0/modules/content/"
4   xmlns:wfw="http://wellformedweb.org/CommentAPI/"
5   xmlns:dc="http://purl.org/dc/elements/1.1/"
6   xmlns:atom="http://www.w3.org/2005/Atom"
7   xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
8   xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
9   >
10
```

```
11 <channel >
12   <title >Infotics </title >
13   <atom:link href="http://infotics.es/feed/" rel="self" type="
14     application/rss+xml" />
15   <link>http://infotics.es</link >
16   <description>transversalidad de la informaci n y las
17     tecnolog as </description >
18   <lastBuildDate>Sat , 29 Aug 2015 09:34:53 +0000</lastBuildDate >
19   <language>es-ES</language >
20   <sy:updatePeriod>hourly</sy:updatePeriod >
21   <sy:updateFrequency >1</sy:updateFrequency >
22   <generator>http://wordpress.org/?v=4.3</generator >
23   <item >
24     <title >Convocatoria III Taller de Periodismo de Datos 2015:
25       explorando la desigualdad </title >
26     <link>http://infotics.es/2015/08/29/convocatoria-iii-taller-de-
27       periodismo-de-datos-2015-explorando-la-desigualdad/</link >
28     <comments>http://infotics.es/2015/08/29/convocatoria-iii-
29       taller-de-periodismo-de-datos-2015-explorando-la-desigualdad
30       /#comments </comments >
31     <pubDate>Fri , 28 Aug 2015 22:00:00 +0000</pubDate >
32     <dc:creator ><![CDATA[ Adolfo Ant n Bravo ]]></dc:creator >
33     <category ><![CDATA[ evento ]]></category >
34     <category ><![CDATA[ medialabprado ]]></category >
35     <category ><![CDATA[ periodismodatos ]]></category >
36     <category ><![CDATA[ DDJ ]]></category >
37     <category ><![CDATA[ desigualdad ]]></category >
38     <category ><![CDATA[ inequality ]]></category >
39     <category ><![CDATA[ periodismodedatos ]]></category >
40     <guid isPermaLink="false">http://infotics.es/?p=1261</guid >
41     <description ><![CDATA[ Convocatoria del Taller de Producci n
42       de periodismo de datos de Oto o 2015 en Medialab Prado
43       . ]]></description >
44     <content:encoded ><![CDATA[
```

```

39     <!-- contenido del post -->
40     ]]></content:encoded>
41     <wfw:commentRss>http://infotics.es/2015/08/29/convocatoria-
    -iii-taller-de-periodismo-de-datos-2015-explorando-la-
    desigualdad/feed/</wfw:commentRss>
42     <slash:comments>0</slash:comments>
43 </item>
44 </channel>
45 </rss>

```

Donde:

- La primera línea explica que se trata de un documento *XML* en su versión *1.0* y codificado como *UTF-8*.
- La segunda línea explica que se trata de un archivo *RSS* en su versión *2.0*, y los otros atributos son espacios de nombre *XML* `xmlns` que utiliza.
- La tercera línea es del elemento `channel`, donde comienza la información de las actualizaciones, que contiene los subelementos con el contenido, entre ellos `item` que se corresponde con cada entrada o noticia de *RSS*.

## 9.II. FOAF

*FOAF* se trata de un vocabulario con definiciones que usa el lenguaje *RDFS/OWL* [149] para describir las relaciones entre las personas.

```

1 <rdf:RDF <!-- aqu se declara RDF -->
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   <!-- URI del espacio de nombres XML de RDF -->
4   xmlns:foaf="http://xmlns.com/foaf/0.1/"

```

```
5 <!-- URI del namespace XML de FOAF -->
6 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
7 <!-- URI del namespace XML de RDF-Schema -->
8 <foaf:Person>
9 <!-- comienza a describir una persona con el vocabulario FOAF
   -->
10 <foaf:name>Adolfo Ant n Bravo</foaf:name>
11 <!-- Nombre -->
12 <foaf:mbox rdf:resource="mailto:adolflow@sindominio.net" />
13 <!-- direcci n de correo electr nico , se pueden poner
   varios -->
14 <foaf:homepage rdf:resource="http://www.sindominio.net/
   adolflow/" />
15 <!-- P gina web personal -->
16 <foaf:nick>adolflow</foaf:nick>
17 <!-- Nick o apodo -->
18 <foaf:depiction rdf:resource="http://userserve-ak.last.fm/
   serve/126/1541940.gif" />
19 <!-- Peque a imagen personal -->
20 <foaf:interest>
21 <!-- Informaci n sobre intereses personales , se pueden
   poner m s -->
22 <rdfs:Description rdf:about="http://www.sindominio.net"
   rdfs:label="Sindominio.net" />
23 </foaf:interest>
24 <foaf:knows>
25 <!-- Informaci n sobre personas a las que conoces -->
26 <foaf:Person>
27 <foaf:name>Dan Brickley</foaf:name>
28 <foaf:nick>danbri</foaf:nick>
29 </foaf:Person>
30 </foaf:knows>
31 </foaf:Person>
32 </rdf:RDF>
```

La cuestión de la identificación personal es muy importante para la Web Semántica.

*FOAF* es un sistema de identificación de relaciones estándar, abierto y libre, que puede conectarse con otros servicios como *OpenID* o los de las redes sociales..

Antes de *FOAF* se expresaban las relaciones con usuarios a través de *XFN* (*XHTML Friends Network* o red de amigos en *XHTML*) que utilizaba el atributo `rel` del elemento `a` con un vocabulario propio que expresaba el grado de relación un un significado determinado. Para indicar que se utiliza *XFN*, hay que utilizar el atributo `profile` del elemento `head`.

```
1 <html >
2 <head profile="http://gmpg.org/xfn/11" >
3 </head >
4 <body >
5 <p><a rel="colleague met" href="http://www.sindominio.net/
   adolflow">Adolflow </a> est escribiendo su tesis </p>
6 </p>
7 </body >
8 </html >
```

## 9.12. SKOS y entidades

*SKOS* (*Simple Knowledge Organization System*, sistema de organización simple del conocimiento) es un modelo de datos comunes para compartir y enlazar sistemas de organización del conocimiento como tesauros, taxonomías, esquemas de clasificación y sistemas de cabeceras, a través de la Web [161].

Los conceptos se identifican a través de *URI*, en uno o varios idiomas y se documentan y estructuran a través de relaciones semánticas de diversa tipología. Soporta mapeo de conceptos de distintos esquemas, colecciones ordenadas, agrupaciones de conceptos y relaciones entre las etiquetas asociadas a los conceptos.

Como lenguaje *RDF*, los elementos con los que cuenta son clases y propiedades, y la estructura e integridad del modelo de datos se define por las características lógicas y las interdependencias entre esas clases y propiedades.

Se puede utilizar únicamente *SKOS* o junto con *OWL*. *SKOS* no se considera un lenguaje formal de representación del conocimiento sino a caballo entre la web de documentos y la web de datos, ya que el conocimiento explícito de una ontología se expresa como conjunto de axiomas y hechos, mientras que el tesoro es de una naturaleza distinta que identifica y describe, a través del lenguaje natural y otros medios informales, un conjunto de ideas y de significados o conceptos. Los conceptos se pueden organizar en diversas estructuras, como redes de asociación y jerarquías.

El concepto de identidad está relacionado con el de entidad [222], algo identificable y distintivo, lo cual puede ser una persona, un libro, un edificio o una ciudad, por ejemplo. La entidad se define, se nombra, por la combinación de una denotación basada en el identificador –significado– y de la connotación basada en el contenido del documento – la descripción. Un identificador es un signo, un símbolo o conjunto de signos que representan la entidad. Hay distintos tipos de identificadores:

- Encerrados en comillas simples o dobles, como por ejemplo, =.Adolfo Antón-
- Referencias relativas como, por ejemplo, <#AdolfoAnton>
- *URI HTTP* absoluto:

```
<http://infotics.es/adolfoanton>
```

Lo cual sería también un *WebID*.

- *URI LDAP*:

```
<ldap://infotics.es/cn\=Adolfo%2oAnton%2Cou\=Accounts%2Co\=
```

```
Info tics es %2Cc\=ES>
```

Lo cual sería también un *NetID*, una tarjeta de identidad que describe o connota al sujeto.

### 9.13. Microformats, Microdata y RDFa

*Microformats* es un proyecto que pretendía ofrecer significado en la propia Web de documentos *HTML*, una alternativa ligera y más sencilla a *XML* y *RDF*. Los microformatos son etiquetas con una sintaxis y una semántica definida que se insertan en *HTML* en los atributos `class`, `rel` y `rev` y dan información extra sobre personas, eventos, geolocalización, productos, recetas o archivos de imagen, audio y vídeo. Al estar publicada la especificación cualquiera que quiera compartir o intercambiar datos en *microformats* puede hacerlo. También se pueden incrustar en archivos *(X)HTML/XML* o *Atom/RSS* y hay exportadores a *JSON*.

El vocabulario de los microformatos se inserta en los valores de los atributos. Este uso del atributo `class` está contemplado en *HTML 4.01* ya que explicita que se puede utilizar para un uso general por los agentes [195].

Se pueden utilizar para datos personales de una agenda de contactos, donde se elegiría el microformato *hCard* [71], que está basado en el estándar de intercambio de datos de agenda de contactos *vCard*:

```
1 <p class="h-card">
2   
3   <a class="p-name u-url" href="http://sindominio.net/adolflow">
   Adolfo Ant n Bravo </a>
```

```

4 <a class="u-email" href="mailto:adolflow@sindominio.net">
  adolflow@sindominio.net </a>,
5 <span class="p-adr">Las guilas </span>
6 <span class="p-locality">Madrid </span>
7 <span class="p-country-name">España </span>
8 </p>

```

Donde se emplea:

- Un elemento párrafo `p` con la clase (class) raíz `h-card` para los subelementos *HTML*.
- El elemento imagen `img`, con la clase `u-photo` (*user photo*)
- El elemento `a` (*anchor*, ancla, enlace) con las clases `p-name` (*person name*, nombre de persona) y `u-url` (*user url*, *url* de usuario) que enlaza con la web-
- Otro elemento `a` que enlaza con el correo electrónico a través del valor `u-email`.
- Tres filas con elementos `span` donde hay información sobre la dirección `p-adr`, la localidad `p-locality` y el país `p-country-name`

También se puede emplear el microformato *vcard*:

```

1 <dl class="vcard">
2 <dt class="fn"><a href="http://www.sindominio.net/adolflow"
  class="url">Adolfo Ant n Bravo </a></dt>
3 <dd class="title">Doctorando </dd>
4 <dd class="nickname">adolflow </dd>
5 <dd class="adr"><span class="locality">Madrid </span>, <abbr
  title="Comunidad de Madrid" class="region">Comunidad de
  Madrid </abbr> <span class="postal-code">28044 </span></dd>
6 <dd class="u-photo"> </dd>  
</dl>
```

En este caso:

- Se utiliza el elemento de *HTML* `dl` de *description list* o lista de descripción de elementos que contiene los subelementos `dt` de *description term* o descripción del término y `dd` de *description definition* o descripción de la definición. Se utiliza `vcard` como propiedad raíz en vez de `hcard`
- El `dt` tiene la propiedad `fn` de *friendly* (amigable), *formal* (formal) o *family* (de familia) `name` (nombre).
- El enlace `a` lleva el valor `url` para indicar la URL.
- `nickname` para el alias.
- `adr` para dirección, `locality` para localidad, `region` para comunidad autónoma y `postal-code` para código postal.

*HTML5* presenta algunos cambios, como por ejemplo se ha eliminado el atributo `rev` que era utilizado por *VoteLinks*, y tampoco existe el elemento `profile` que utilizaba el perfil URI.

*Microdata* es una especificación *HTML* de *WHATWG* para anidar metadatos en el contenido de las páginas web. Al igual que *microformats*, dispone de un vocabulario que describe elementos y pares nombre/valor para asignar valores a sus propiedades. Dispone de los siguientes atributos globales:

- *itemscope* crea un elemento `e` indica que los elementos que descienden contienen

información sobre él.

- *itemtype*, una *URL* válida de un vocabulario que describe el ítem y el contexto de sus propiedades.
- *itemid* indica un identificador único para el ítem.
- *itemprop* indica que la etiqueta que lo contiene lleva el valor de la propiedad del ítem especificado. El nombre de la propiedad y el contexto del valor se describen en el vocabulario del ítem. Los valores de las propiedades normalmente consisten en valores de cadenas de caracteres, pero también pueden usar *URL* a través del elemento *a* y de su atributo *href*, del elemento *img* y su atributo *src* o cualquier otro elemento que enlace a o embeba recursos externos.
- *itemref*, propiedades que no descienden del elemento, como ocurre con el atributo *itemscope*, se pueden asociar con el ítem a través de este atributo.

Con respecto a *Microformats*, *Microdata* es parte de la especificación *HTML5* y utiliza vocabularios personalizados con nuevos atributos para aplicar metadatos a cualquier contenido web, no sólo al contenido especificado en *Microformats*. Además, puede incluso anotar el *DOM*, lo que facilita el análisis de los metadatos.

El mismo ejemplo de *vcard* con *Microformats*, marcado con *Microdata*, donde con *itemscope* se especifica que se va a utilizar el vocabulario con la propiedad *itemtype* con *DataVocabulary.org* [84], que dispone de definiciones para *Person* y *Address*, entre otros.

```
1 <dl itemscope itemtype="http://data-vocabulary.org/Person">
2   <dt itemprop="name"><a href="http://www.sindominio.net/adolfo"
   " itemprop="url">Adolfo Ant n Bravo </a></dt>
3   <dd itemprop="title">Estudiante de Doctorado </dd>
4   <dd itemprop="address" itemscope itemtype="http://data-
   vocabulary.org/Address">
```

```

5 <span itemprop="locality">Madrid</span>,
6 <span title="Comunidad de Madrid" itemprop="region">Comunidad
  deMadrid</span>
7 <span itemprop="postal-code">28044</span>
8 </dd>
9 <dd itemprop="nationality">Habitante del mundo</dd>
10 <dd itemprop="nationality">Español</dd>
11 </dl>

```

Donde:

- Del elemento dl desciende información con `itemscope` y se declara que se trata de una persona con el `itemtype`.
- El elemento dt tiene la propiedad `itemprop name` y contiene el elemento a con la propiedad `itemprop url`
- El primer elemento dd define el título `title` con la propiedad `itemprop`.
- El segundo elemento dd informa de que se trata de una dirección con el `itemprop address`, que además es un `itemscope` que va a contener más información relacionada y que elige la definición de `Address` del vocabulario de *DataVocabulary.org*, de donde se utiliza `locality`, `region` y `postal-code`
- El último dd utiliza la propiedad de elemento `nationality` para la nacionalidad. Un ítem puede múltiples propiedades con el mismo nombre y distintos valores.

Para usar el atributo `itemref`, se definen los elementos siguientes que están relacionados con el elemento raíz con propiedad `id=blakey`:

```

1 <div itemscope id="blakey" itemref="a b"></div>
2 <p id="a">Nombre: <span itemprop="name">Art Blakey</span></p>

```

```

3 <div id="b" itemprop="band" itemscope itemref="c"></div>
4 <div id="c">
5   <p>Grupo: <span itemprop="name">Jazz Messengers</span></p>
6   <p>Componentes: <span itemprop="size">5</span> m sicxs</p>
7 </div>

```

Un elemento también puede tener dos propiedades, que atienden a significados distintos del nombre. Por ejemplo, la palabra *castaña*, tiene como significado el fruto del castaño pero también otros, como sinónimo de bofetada, golpe o cosa fastidiosa. Entonces se puede representar así:

```

1 <div itemscope>
2   <span itemprop="calidad-producto fruto-casta o">casta a</span>
3 </div>

```

Desde 2011 el vocabulario de *Data-Vocabulary* ha dejado de mantenerse en pro del proyecto *schema.org*, impulsado por los principales buscadores del mercado, entre los que se encuentran *Google*, *Bing*, *Yahoo!* o *Yandex*. Se puede utilizar el vocabulario *schema.org* para dar valor de `itemid`, con lo que se dispondrá de toda la información compatible con el idioma que entienden los buscadores:

```

1 <dl itemscope itemtype="http://schema.org/Book">
2 <dt><a itemprop="url" href="https://books.google.es/books/about/
   Teor%C3%ADa_general_de_la_informaci%C3%B3n.html?id=qpH-3
   vX2RuUC"><span itemprop="name"><strong>Teor a General de la
   Informaci n: datos, relatos y ritos</strong></span></a></dt>
3 <dd itemprop="author" itemscope itemtype="http://schema.org/
   Person">
4 Escrito por: <span itemprop="name">Gonzalo Abril</span></dd>
5 <dd itemprop="publisher" itemscope itemtype="http://schema.org/
   Organization">
6 Published by: <span itemprop="name">C tedra</span></dd>
7 <dd>ISBN: <span itemprop="isbn">8437614953, 9788437614953</span>
   </dd>

```

```
8 </dl>
```

*RDFa* corresponde con *RDF in Attributes*, RDF en atributos [4]. En su propósito resulta similar a *Microformats* y *Microdata*, ya que añade un conjunto de atributos y reglas de procesamiento para que *HTML* soporte el modelo de representación *RDF* y produzca documentos enriquecidos. Sin embargo, al adaptar la lógica de triples de *RDF* a *HTML* ofrece una compatibilidad mayor y además permite crear vocabularios propios, con lo que se dispone de mayor flexibilidad que con *Microformats* o *Microdata*.

Un documento *HTML* sencillo:

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 <title>Página personal de Adolfo Antón Bravo</title>
4 <meta name="author" content="Adolfo Antón Bravo" />
5 </head>
6 <body>... </body>
7 </html>
```

Se expresaría en *RDFa* así:

```
1 <html
2   xmlns="http://www.w3.org/1999/xhtml"
3   xmlns:foaf="http://xmlns.com/foaf/0.1/"
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   >
6 <head>
7 <title>Página personal de Adolfo Antón Bravo</title>
8 <meta property="dc:creator" content="Adolfo Antón Bravo" />
9   <link rel="foaf:topic" href="http://www.sindominio.net/
10     adolfoflow/" />
11 </head>
12 <body>... </body>
13 </html>
```

En el elemento `html` se especifica que se utilizan los espacios de nombres de *xhtml*, *foaf* y *dc*.

## 9.14. XML Schema

*XML Schema* es una alternativa o evolución del uso de *DTD* en *XML*, se utiliza para describir la estructura y las restricciones de los documentos XML y los tipos de datos (*datatypes*) válidos para cada elemento y atributo, lo cual es una de sus principales diferencias. Al realizar esta restricción del tipo de datos, se considera muy seguro para intercambiar información entre aplicaciones ya que no hay que comprobar el tipo de datos.

Desarrollado por el *W3C* desde 1998, alcanzó el nivel de *Recomendación* en 2001 [102] y la segunda revisión es de 2004 [103]. Consta de tres partes:

1. Parte 0, *Primer*, introducción a la normativa del lenguaje, con ejemplos y explicaciones.
2. Parte 1, *Structures*, descripción de los elementos del lenguaje.
3. Parte 2, *Datatypes*, tipos de datos y restricciones.

Su enfoque es modular, lo que le asemeja a la programación orientada a objetos *POO* que potencia la reutilización de código. Si *XML Schema* tiene *datatypes*, la *POO* tiene clases. Se pueden crear *datatypes* a partir de datos predefinidos, agrupando elementos y atributos, pudiendo ser simples si no disponen ni elementos ni atributos, como, por ejemplo, *string*, *double*, *boolean*, *list*, *union*), o complejos, con elementos hijos y/o atributos, nombrados o anónimos, pero si tienen nombre se pueden reutilizar, ya que

también se utilizan *namespaces*.

Algunos tipos de datos son iguales o parecidos a los de los lenguajes de programación. Los más comunes son:

- `xs:string` cadenas de caracteres, texto.

```
1 <xs:element name="persona" type="xs:string"/>
2 <persona>Adolfo Ant n Bravo</persona>
```

- `xs:decimal`, números decimales.

- `xs:integer`, números enteros.

- `xs:boolean`, valores *booleanos*, verdadero o falso, *true* or *false*.

- `xs:date`, fechas, en formato año-mes-día, AAAA-MM-DD. Para especificar la zona horaria en *UTC*, se añade una Z detras de la fecha. Se pueden indicar usos horarios por encima o por debajo de *UTC* con los signos de menos y más.

```
1 <xs:element name="limite" type="xs:date"/>
2 <limite>2015-09-15+02:00</limite>
```

- `xs:time`, hora, en formato horas-minutos-segundos, hh:mm:ss, donde se puede indicar la zona horaria igual que en `xs:date`,

```
1 <xs:element name="comida" type="xs:time"/>
2 <comida>01:30:00+02:00</comida>
```

- `xs:dateTime`, fusiona los dos anteriores en la estructura AAAA-MM-DDThh:mm:ss-

```
1 <xs:element name="plazo" type="xs:dateTime"/>
2 <plazo>2015-09-15T14:00+02:00</plazo>
```

- `xs:duration`, para especificar un intervalo de duración, con el esquema periodo-nº años-nº meses-nº días-zona horaria-nº horas-nº minutos-nº segundos, PnAnMnDTnHnMnS:

```

1 <xs:element name="periodo" type="xs:duration"/>
2 <periodo>P6M</periodo>
3 <!-- un periodo de seis meses -->

```

Un ejemplo de archivo *xsd* (*XML Schema Definition*, definición de *XML Schema*), sería:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="Libro">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="Titulo" type="xs:string"/>
7         <xs:element name="Autor a" type="xs:string" maxOccurs
8           ="5"/>
9         <xs:element name="Editorial" type="xs:string"/>
10        </xs:sequence>
11        <xs:attribute name="precio" type="xsd:double"/>
12      </xs:complexType>
13    </xs:element>
14  </xs:schema>

```

Donde se declara:

- El elemento `schema`, raíz del documento, junto con el espacio de nombres `xmlns`.
- Se define el elemento `Libro` que tiene tres hijos, `Título`, `Autoría` y `Editorial`, de tipo `string`
- Un atributo (`precio`) de tipo `double`.

- Título y Editorial solo pueden aparecer una vez pero Autoría puede aparecer hasta cinco veces con `maxOccurs`
- Al estar ubicados en una secuencia `sequence`, deben estar aparecer en ese orden.

Y un archivo *XML* que utiliza este *XML Schema*:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Libro xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi
   :noNamespaceSchemaLocation="libro.xsd" precio="20">
3   <Titulo>Teoría General de la Información </Titulo>
4   <Autora>Gonzalo Abril </Autora>
5   <Editorial>Ctedra </Editorial>
6 </Libro>
```

No hay que confundir *XML Schema* con *XML schemas*, aunque es habitual referirse a los lenguajes de esquema como *XML schema*, que es como se conoce al conjunto de los lenguajes de esquema entre los que están *DTD* y *XML Schema*, además de *RELAX NG*, *Schematron*, *NRL* (*Namespace Routing Language*, lenguaje de rutas del espacio de nombres), *DSDL* (*Document Schema Definition Languages*, lenguajes de definición del esquema de documentos), *DDML* (*Document Definition Markup Language*, lenguaje de marcas de definición del documento), *DSD* (*Document Structure Description*, descripción de la estructura del documento), *SGML* o *SOX*.

## 9.15. Realidad aumentada

La realidad aumentada *augmented reality* es la visión de la realidad a la que se agregan elementos virtuales [223], un concepto que se utiliza para explicar la interacción que se puede tener con la realidad interponiendo entre la visión ocular y la realidad un dispositivo móvil que consta de un dispositivo de entrada de datos, como por ejemplo una

cámara, y un dispositivo de salida, como es la pantalla. Si la cámara o el dispositivo es capaz de saber sobre lo que está viendo o la posición en la que se encuentra, se puede interactuar con la realidad a través de una aplicación de *realidad aumentada*, un navegador específico del dispositivo móvil. Para introducir datos a la aplicación se permiten distintas técnicas, como, por ejemplo, dispositivos *RFID* (*Radio Frequency Identification*, identificación por radio frecuencia), códigos *QR* (*Quick Response Code*), códigos de barras o geolocalización. De esta manera se añade a la visualización de la pantalla una o varias capas con información sobre la que interactuar.

Con el propósito de disponer de un lenguaje unificado para el uso en aplicaciones de realidad aumentada nació *ARML* (*Augmented Reality Markup Language*), el lenguaje de marcas de la realidad aumentada, basado en *KML* (*Keyhole Markup Language*), un lenguaje de marcado basado en *XML* para representar datos geográficos en tres dimensiones. *ARML* es un formato de datos descriptivos basado también en *XML* que se centra en mapear *puntos de interés* (*POI*, *Points of Interest*) y añadirle metadatos. Fue creado en 2009 por los miembros de *Wikitude World Browsers* para permitir a desarrolladores crear contenidos para los buscadores de realidad aumentada.

La versión *ARML 1.0* y *KML* son formatos de datos similares, descriptivos y propietarios mientras que la versión *ARML 2.0* está impulsada por el *OGC* (*Open Geospatial Consortium* o consorcio geoespacial abierto) y su *ARML 2.0 Standards Working Group* (grupo de trabajo de estándares *ARML 2.0*), y permite partes dinámicas que modifican las propiedades definidas en la parte descriptiva, define un conjunto de eventos con los que trabajar en la interacción, extiende las opciones de presentación de puntos de interés *POI* a visualizaciones como objetos *3D*, líneas y polígonos, y ofrece puerto de conexión para métodos de seguimiento (visual y audio). Aunque ha quedado fuera de *ARML 2.0* una forma de estandarizar el seguimiento (*tracking*), reconocen que tanto el seguimiento, los sensores o cualquier otra interacción hardware-software, los algoritmos de proyección en pantalla, las consultas de *POI*, los formatos de almacenamiento de *POI* y los protocolos de transmisión de datos están fuera del alcance del trabajo del grupo

[179].

Un ejemplo de archivo *ARML 1.0* consta de:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://www.opengis.net/kml/2.2"
3   xmlns:ar="http://www.openarml.org/arml/1.0"
4   xmlns:wiktitude="http://www.openarml.org/wiktitude/1.0">
5
6   <Document>
7     <ar:provider id="mis-lugares.net">
8       <ar:name>Los lugares que me gustan</ar:name>
9       <ar:description>Campo, ciudad, playa y montaña, las
10      cuatro estaciones, lugares donde disfrutar de la vida.</ar:
11      description>
12       <wiktitude:providerUrl>http://www.providerhomepage.com </
13       wiktitude:providerUrl>
14       <wiktitude:logo>http://www.mountain-tours-I-love.com/
15       wiktitude-logo.png </wiktitude:logo>
16     </ar:provider>
17     <Placemark id="123">
18       <ar:provider>mountain-tours-I-love.com</ar:provider>
19       <name>Gaisberg</name>
20       <description>Gaisberg is a mountain to the east of
21       Salzburg, Austria</description>
22       <wiktitude:info>
23       <wiktitude:thumbnail>
24         http://www.mountain-tours-I-love.com/gaisberg-thumb.png
25       </wiktitude:thumbnail>
26       <wiktitude:phone>555-9943</wiktitude:phone>
27       <wiktitude:url>http://en.wikipedia.org/wiki/Gaisberg </
28       wiktitude:url>
29       <wiktitude:email>info@mountain-tours-I-love.com</wiktitude:
30       email>
31       <wiktitude:address>Jakob-Haringer-Str. 5a, 5020 Salzburg,
32       Austria</wiktitude:address>
33     </Placemark>
34   </Document>
35 </kml>
```

```
26     <Point >
27     <coordinates >13.11,47.81,1158 </coordinates >
28     </Point >
29 </Placemark >
30 </Document >
31 </kml >
```

El documento comienza con la cabecera, la declaración de *XML*, y en la segunda línea el elemento *kml*, que es el elemento *root* o raíz descrito por el estándar *KML*. En la misma línea se incluyen los espacios de nombre de *ARML 1.0* y de *Wikitude*, que también se utilizan en este documento.

El cuerpo del documento se inicia, con el elemento nativo de *kml* *Document*, que contiene los subelementos *ar:provider*, del espacio de nombres de *ARML 1.0* y *Placemark*, elemento nativo de *kml*.

*ar:provider* identifica el proveedor de información. Cada proveedor debe tener un identificador único como atributo, en este caso *mis-lugares.net*. También utiliza de *ARML 1.0* los subelementos *ar:name*, que indica el nombre del lugar, y *ar:description*, que es una descripción del mismo. Se incluyen los subelementos del espacio de nombres de *Wikitude* *wikitude:providerUrl*, que informan de la *URL* del proveedor, y *wikitude:logo*, que contiene la *URL* de un logo, una imagen o archivo vectorial normalmente.

Después de *ar:provider* se encuentra el elemento nativo de *kml*, *Placemark* *wikitude:info* que ofrece información extra con los subelementos *wikitude:thumbnail*, para la *URI* de una imagen, *wikitude:phone*, para un teléfono de contacto, *wikitude:url*, para una *URL* de información, *wikitude:email*, para un correo electrónico de información y *wikitude:address*, para una dirección postal de la información.

Finalmente, después del elemento *Document* el otro elemento de *kml* *Point* contiene

el elemento `coordinates` con las coordenadas del punto.

*W3C* ha desarrollado el vocabulario *Basic Geo WGS84 lat/long vocabulary* [58] que ofrece datos de localización en *RDF* para *POI*, lo cual permite intercambiar estos datos entre aplicaciones. Gracias a los datos vinculados estos datos se enlazan con otros repositorios de información geográfica como *Geonames*. En este ejemplo, el *POI* representa la latitud y longitud concreta:

```

1 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2   xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
3   <geo:Point>
4     <geo:lat>40.3789412622424</geo:lat>
5     <geo:long>-3.75768437087443</geo:long>
6   </geo:Point>
7 </rdf:RDF>

```

Al principio se declara el espacio de nombres `xmlns:geo` junto con el espacio de nombres `rdf` dado que se trata de un documento *RDF*. El elemento raíz es `geo:Point` que cuenta con los subelementos `geo:lat` para latitud y `geo:long` para longitud.

Se puede combinar *Geo* con otros vocabularios como *FOAF*:

```

1 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
3   xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   xmlns="http://xmlns.com/foaf/0.1/">
6 <Person>
7   <name>Adolfo Ant n Bravo</name>
8   <homepage dc:title="P gina de adolflow"
9     rdf:resource="http://sindominio.net/adolflow"/>
10  <based_near geo:lat="40.3789412622424" geo:long
11    ="-3.75768437087443"/>
12  <rdfs:seeAlso rdf:resource="http://sindominio.net/adolflow/foaf.
    rdf"/>

```

```
12 <!-- aqu podr a ir m s c digo RDF con cualquier  
    vocabulario RDF -->  
13 </Person >  
14 </rdf:RDF>
```

La intersección de la realidad aumentada con la Web se denomina *augmented web* (web aumentada), una combinación de *HTML5*, *Web Audio*, *WebGL* y *WebRTC* que aumenta la experiencia de lxs usuarixs que visitan páginas web. El término fue acuñado por Alex Young en 2011 en una reunión de *ISMAR* (*international Symposium on Mixed and Augmented Reality*, simposio internacional sobre realidad mixta y aumentada).

## 9.16. SPARQL

Para acceder a los datos se necesita un lenguaje de consultas como *SPARQL* y un *endpoint* o punto final *SPARQL* que es una *URL* desde donde realizar consultas. *SPARQL*, como otros lenguajes de consultas, selecciona datos del conjunto de datos con la declaración *SELECT*, lo que determina de qué subconjunto de datos se produce la selección. También se utiliza *WHERE* para definir patrones de un gráfico para encontrar una coincidencia. Un patrón gráfico consta de sujeto, predicado y objeto. La declaración *SELECT* busca sobre la variable *?name*

*SPARQL* es el lenguaje de consultas de la Web Semántica. Permite, con ciertas técnicas, obtener valores de datos estructurados y semi-estructurados, explorar los datos consultando posibles relaciones desconocidas, realizar enlaces complejos entre distintas bases de datos en una simple consulta o transformar datos en *RDF* de un vocabulario a otro.

*SPARQL 1.0* se convirtió en estándar en enero de 2008 [16] como *lenguaje de consultas SPARQL 1.0* para encontrar patrones en datos *RDF* en consultas sobre *HTTP*. Los resultados se podían devolver en *XML* y *JSON*. En 2013 se actualizó a la versión *SPARQL*

1.1 [105] que actualizaba las versiones del protocolo y del lenguaje de consultas donde se incluía una forma de insertar, borrar y modificar datos *RDF*, se permitía acceso *REST* para grafos *RDF*, consultar varios *endpoints* a la vez y posibilidad de obtener resultados en *CSV/TSV*.

Una consulta *SPARQL* comprende, en este orden:

- Declaraciones de prefijos, *prefix declaration*, para abreviar las *URI*.
- Definición de conjuntos de datos, *dataset definition*, se declaran los grafos *RDF* que se consultan.
- Un pronombre de resultado, *result clause*, que identifica qué información devuelve la consulta.
- Un patrón de búsqueda, *query pattern*, que especifica qué consultar en el conjunto de datos.
- Modificadores de consulta, *query modifiers*, que eligen, ordenan y disponen los resultados de la consulta.

Para realizar consultas *SPARQL* se necesita un punto final o *endpoint* de los muchos disponibles en la Web. Una lista actualizada de puntos finales está albergada por el *W3C* en su sitio web <http://www.w3.org/wiki/SparqlEndpoints>. Algunos puntos finales permiten especificar el conjunto de datos sobre el que se realiza la consulta, *Default Data Set Name (Graph IRI)* o nombre del conjunto de datos (del grafo *IRI*) sobre el que se quiere realizar la consulta.

Además, si en la consulta se declara el prefijo, se puede utilizar el espacio de nombres propio. Si no, se ha de realizar la consulta con la *URI* completa. Hay muchos prefijos predefinidos, <http://dbpedia.org/sparql?nsdecl>. Algunos prefijos

jos que se suelen utilizar:

- PREFIX dbpedia-owl: <<http://dbpedia.org/ontology/>>, para consultar la ontología de la *DBpedia*.
- PREFIX nobel: <<http://data.nobelprize.org/terms/>>, para consultar premios *Nobel*.
- PREFIX foaf: <<http://xmlns.com/foaf/0.1/>>, para consultas del vocabulario *FOAF*.
- PREFIX yago: <<http://yago-knowledge.org/resource/>>, para consultar la base *YAGO*.
- PREFIX viaf: <<http://viaf.org/viaf/>>, para la consulta del fichero de autoridades virtual *VIAF*.
- PREFIX meta: <<http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/metadata#>>, consultas de
- PREFIX dcterms: <<http://purl.org/dc/terms/>>, para los términos de *Dublin Core*
- PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>, consultas con el esquema *RDFS*.
- PREFIX dbpedia: <<http://dbpedia.org/resource/>>, consultas de la *DBpedia*.
- PREFIX owl: <<http://www.w3.org/2002/07/owl#>>, consultas con el esquema *OWL*.

- PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>, para consultas con el esquema *XMLSchema*
- PREFIX map: <http://data.nobelprize.org/resource/#>
- PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- PREFIX freebase: <http://rdf.freebase.com/ns/>
- PREFIX dbpprop: <http://dbpedia.org/property/>
- PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

Si el *endpoint* ofrece un formulario que incluye los prefijos que se van a utilizar, no hay que especificarlos en la consulta:

```

1 SELECT DISTINCT ?Concept WHERE {
2   [] a ?Concept
3 } LIMIT 5

```

Hay *endpoints* como el de Aragón *Open Data* que permite realizar consultas en su grafo <http://opendata.aragon.es/graph/Aragopedia/latest>, donde se pueden utilizar un prefijo propio además de los prefijos de *DBpedia*, ya que hay entidades como *comarca* o *comunidad autónoma* que no están definidas en *DBpedia* que están definidas en su propia ontología *Aragopedia Ontology* [80].

```

1 PREFIX aragodef: <http://opendata.aragon.es/def/Aragopedia#>

```

Se pueden realizar consultas como el listado de municipios de Aragón:

```

1 PREFIX aragodef: <http://opendata.aragon.es/def/Aragopedia#>
2 SELECT DISTINCT ?x
3   WHERE {
4     ?x a <http://dbpedia.org/ontology/Municipality >}

```

El listado es muy largo y además el uso del prefijo no era necesario puesto que la búsqueda es de municipios, entidad que sí que comprende el vocabulario de *DBpedia*.

Se puede hacer lo mismo con las comarcas y preguntar el número de comarcas que hay en Aragón:

```

1 PREFIX aragodef: <http://opendata.aragon.es/def/Aragopedia#>
2 SELECT COUNT(DISTINCT ?x) AS ?numComarcas
3 WHERE {
4   ?x a aragodef:Comarca }

```

*Aragopedia* utiliza *data cube* [83], una recomendación que se considera una extensión de las tablas de dos dimensiones para llegar a las tres o más dimensiones, el hipercubo, que forma cada conjunto de datos. Cada una de las celdas del cubo de datos se denomina *qb:Observation*. Cada observación tiene asociadas dimensiones o atributos. *qb* es el prefijo para <http://purl.org/linked-data/cube#>

```

1 PREFIX qb: <http://purl.org/linked-data/cube#>
2 SELECT DISTINCT ?y
3 WHERE {
4   ?x qb:dataSet ?y }

```

Cada uno de ellos dispone de varias dimensiones a las que se puede acceder. Por ejemplo, el cubo de datos de *Reciclaje*:

```

1 PREFIX qb: <http://purl.org/linked-data/cube#>
2 SELECT DISTINCT ?p
3 WHERE {
4   ?x a qb:Observation .
5   ?x qb:dataSet <http://opendata.aragon.es/recurso/DataSet/
6     Reciclaje > .
7   ?x ?p ?y }

```

Estos portales de datos como el oficial del gobierno de Aragón ofrece además una cierta fiabilidad de sus datos. En el caso de las consultas sobre la *DBpedia*, los datos provienen

de la *Wikipedia*, por lo que pueden estar necesitados de limpieza. Si se consultan los datos de cantantes españolas:

```
1 PREFIX dcterms: <http://purl.org/dc/terms/>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX dbp: <http://dbpedia.org/ontology/>
4 SELECT ?musico ?nombreMusico ?fechaNacimiento ?
   fechaFallecimiento
5 WHERE {
6   ?musico dcterms:subject
7   <http://dbpedia.org/resource/Category:Spanish_female_singers
8   >;
9   rdfs:label ?nombreMusico ;
10  dbp:birthDate ?fechaNacimiento ;
11  dbp:deathDate ?fechaFallecimiento .
12 }
```

Si se encuentran inconsistencias, se puede participar de *Wikipedia* o de *DBpedia* para mejorar los datos.

El esquema de la consulta *SPARQL* consta de [265]:

- PREFIX, declara el esquema o esquemas que se usan en la consulta.
- FROM, cláusula que identifica las fuentes de datos que han de consultarse.
- SELECT, cláusula que identifica los valores que han de ser devueltos.
- WHERE, cláusula expresada en patrones triplas/grafos que se contrastarán contra las triplas/grafos en *RDF*.

Los tipos de consultas pueden ser [265]:

- **SELECT**: devuelve el resultado en forma tabular sobre XML, mostrando las vinculaciones encontradas de las variables
- **CONSTRUCT**: genera y devuelve el grafo RDF siguiendo el patrón definido en la consulta.
- **ASK**: devuelve un resultado booleano dependiente de si existe una solución a la consulta.
- **DESCRIBE**: devuelve información sobre un recurso específico. Depende de la configuración del procesador.

Se modifica la consulta con los modificadores [265]:

- **ORDER BY**, ordena los resultados según el criterio especificado.
- **DISTINCT**, busca resultados únicos.
- **REDUCED**, elimina algunos, todos o ningún duplicado.
- **LIMIT**, restringe el número de resultados.
- **OFFSET**, controla el punto de inicio de los resultados.

Las consultas se realizan sobre conjuntos de datos *RDF* consistentes en grafos *RDF*. Un punto final endpoint *SPARQL* acepta la consulta y devuelve los resultados por *HTTP*. Los resultados de las consultas se suelen devolver en *HTML*, *XML*, *JSON*, *CSV/TSV*, *RDF/XML*, *N-Triples*, *Turtle*, etc.

Además, se pueden realizar filtros sobre los datos con **FILTER** que admite tipos de datos booleanos `Bool`, enteros `Int`, decimales `Float`, fechas y horarios `dateTime` y operado-

res mayor que  $>$ , menor que  $<$ , menor o igual que  $<=$ , mayor o igual que  $>=$ , igual  $\backslash=$ , distinto de  $!=$ , y AND  $\&\&$ , o OR  $||$

Como otros lenguajes de programación, cuenta con funciones para la conversión o creación de tipos de datos [265]:

- `str(arg)`, *str* por *string*, cadena, y *arg* por *arguments*, argumentos o valores. Convierte a cadena de caracteres.
- `lang(arg)`, *lang* por *language*, idioma, devuelve el idioma del argumento que se le pasa (`.en`, `.es`, `"fr"`, etc.).
- `datatype(arg)`, tipo de dato del argumento.
- `uri(arg)`, convierte el argumento en *URI*.
- `iri(arg)`, convierte el argumento en *IRI*
- `bnode(arg)`, por *blank node*, nodo anónimo.
- `strdt(literal, tipo)`, por *string data type*, tipo de dato de la cadena de caracteres o literal, genera literal con el tipo de datos especificado.
- `strlang(literal, tipo)`, por *string language*, idioma de la cadena de caracteres, genera literal con el idioma especificado.

Funciones condicionales [265]:

- `bound(arg)`, devuelve `true` si tiene un valor, y `false` si no.
- `exists(patron)`, responde `true` si cumple un patrón.

- `not exists (patron)`, responde `true` si no se cumple el patrón.
- `if (cond, expr1, expr2)`, devuelve `expr1` si se cumple la condición y `expr2` si no se cumple.
- `coalesce (expresion1, expresion2, ...)`, responde con la primera expresión que evalúe sin error.

Funciones de comprobación de tipo de datos:

- `isNumeric (arg)`, contesta `true` si es un número.
- `isBlank (arg)`, responde `true` si es un nodo anónimo.
- `isLiteral (arg)`, devuelve `true` si es un literal.
- `isIRI (arg)`, devuelve `true` si es una *IRI*.

---

## Capítulo 10

### Conclusiones

En el transcurso de mi investigación he comentado con muchas personas de muchos ámbitos sobre el mapa conceptual que estaba desarrollando de la Web Semántica y he disfrutado de comentarios de todo tipo. Personas que preguntan qué es, que preguntan si realmente existe, si va a realizarse alguna vez o si eso tiene alguna relación con el periodismo, una vez que habían objetado sobre lo impreciso de la propuesta de tesis y había explicado en el ámbito en el que la desarrollaba.

Creo que este trabajo da respuesta a todas las dudas: a qué es la Web Semántica, qué tecnologías y conceptos forman parte o están relacionados y por dónde pueden continuar investigando. Esto es lo más importante, no en función de su trascendencia sino de su valor de uso y de su lugar inicial en todo proceso de aprendizaje o crítica, la definición de la materia, del objeto de estudio o del fenómeno tecnológico.

Quienes preguntan si realmente existe parten de distintos supuestos a su vez, al menos tres: personas escépticas con el alcance de los objetivos o logros de las tecnologías de la Web Semántica, y por tanto de la Web Semántica en sí, pero que no han participado de

ellas; personas que sí que participan plenamente de las tecnologías y han vivido y viven distintos estados; y personas que han dado por amortizada a la Web Semántica y han vuelto a considerar la Web el marco, tecnología o fenómeno de referencia. Esta última matización es importante y merece la pena matizarla ya que siempre he considerado la Web Semántica un estado de la Web. Las definiciones y los mapas tienen sentido en cuanto los nombres y las tecnologías que engloban lo tienen también. En este caso, tanto si consideras a la Web como a la Web Semántica, el mapa es igual de válido, es el mismo. Me parece que ha ocurrido algo parecido con *WHATWG*, que ahora no considera conveniente utilizar *HTML5* para referirse a *HTML*, una vez que es recomendación *W3C*, cuando fue el promotor del término.

Quienes dudan razonablemente de la Web Semántica por no haber alcanzado sus últimos objetivos forman parte, sin quererlo, de quienes usan el meme de que la Web Semántica no existe, aunque he de considerarlos por separado. Los primeros son conscientes de la extensión y desarrollo de las tecnologías semánticas; los segundos no. A los primeros espero reafirmar en ese grado de consciencia y ayudar a despejar dudas, y a los segundos convencerles del extraordinario proceso tecnológico que estamos viviendo y que exige de nuestra visión crítica, necesaria para que las tecnologías sean realmente útiles.

Por último, la relación con el periodismo no solo se da en el periodismo de datos, aunque es ahí donde encuentra su campo de expresión natural, participado de otro de los fenómenos citados y relacionados como es el de los datos abiertos. El periodismo, lxs profesionales de las Ciencias de la Información, utilizamos como el que más las tecnologías como herramientas de trabajo. Formamos parte de las comunidades *dateras*, trabajamos con datos y son los datos los que conforman la *Web de Datos*. La Web Semántica podrá realizarse o no en sus últimos objetivos de razonamiento, confiabilidad y seguridad, pero quienes participamos de las Ciencias de la Información somos lxs que razonamos cotidianamente con los datos, los contenidos, los formatos, quienes podemos aportar una visión crítica, de conjunto y englobada en las teorías de la información,

de la comunicación, del periodismo.

Me atrevería a proponer una pregunta difícil por cuanto parece obvia su respuesta: ¿existe la Web? Yo respondería inequívocamente que sí, pero también me gustaría incorporar a mi respuesta la gran cantidad de sitios web que no son accesibles, que no cumplen las recomendaciones y que dejan mucho que desear en cuanto a tecnologías web, salvo que podemos ver su contenido en una pantalla. Creo que el *W3C* adoptó en su momento la decisión de permitir a contenido que no cumpliera los estándares ser parte de la Web, y eso incluía desde al software generador del contenido como a los diseñadores y/o desarrolladores web y a lxs productorxs de contenido, en aras de la universalización de la Web. Se asumió que en la Web hubiera contenido semántico más limpio y contenido más sucio, contenido accesible y en su mayor parte inaccesible en los términos de la *accesibilidad Web*. Es decir, para mí muchas páginas web son muy poco web, y no por ello no son parte de la Web.

Creo que esta es una de las razones por las que la Wikipedia ha sido un éxito, porque ha sabido conjugar la comunidad abierta de productorxs que pueden aportar contenido de mayor o menor calidad y semántica con ciertos contenidos estructurados que son utilizados, por ejemplo, por proyectos como la *DBpedia* y que son reutilizables para cualquiera.

Con la Web Semántica no se ha producido una universalización de sus lenguajes de la forma como se ha producido con la Web (*HTML*, *CSS*). Muchxs productorxs de contenido no los conocerán, muchxs no sabrían incorporarlos y muchxs no estarán interesadosxs. A cambio, el contenido semántico procede de productorxs que se esfuerzan más que con una web en *HTML* en el contenido. Gobiernos, bibliotecas, empresas, universidades, medios de comunicación, entidades de todo tipo aportan contenido semántico que participa de la Web Semántica. El contenido de la Web permanece en la Web, las *URL* permanecen [239]. Una *URL* de hace 20 años no deja de perder valor si su semántica es realmente buena, y esto básicamente quiere decir que incorpore el título del

documento. Lo mismo ocurre en la citación de libros y artículos, se comienza con el título, y a continuación el autor, la fecha, la edición, la traducción, etc. En la *URL*, el título del documento identifica su contenido, la *URL* identifica este recurso, antaño nodo y también documento, de la Web.

Si el contenido web permanece y está bien construido, si cumple los estándares de *W3C*, se puede reutilizar. Los contenidos de la Web en *RDF* son parte del conjunto de datos abiertos, pero no todos los datos abiertos son datos compatibles con la Web, por lo que se está dedicando un gran esfuerzo en este reto de tener formatos de datos que no son estándares compatibles con la Web. Desde que el movimiento de datos abiertos lograra que los distintos gobiernos, instituciones y empresas dispusieran de sus datos en portales web, la Web se ha enriquecido con un gran número de conjuntos de datos. Los datos semánticos conforman la *Web de Datos* [35], donde los contenidos semánticos son identificables, son datos accesibles. Si antes identificábamos el nodo, el documento, la página, ahora cada pedazo de información puede ser un dato identificado por una *URI* [45].

Una de las claves de la Web Semántica no es la creación de datos enriquecidos, sino la conexión de los mismos, los datos vinculados, el camino donde la Web Semántica se encuentra con los datos abiertos y se habla de datos vinculados abiertos, estructurados, abiertos y semánticos. Los paradigmas de almacenamiento de información, recuperación de información o búsqueda de información se ven alterados por este conjunto de tecnologías. Por tanto, toda aquella persona que trabaje con datos, con información, va a modificar su forma de trabajar. Para quien investiga, ya no hace falta estar suscrito a los mejores repositorios de información científica; para quien hace periodismo, ya no es necesario bucear en la Web, en archivos desconectados; para los usuarios de bibliotecas, etc.

En general, para cualquiera que busque datos, información o contenidos, la Web Semántica puede considerarse un desarrollo de los prototipos de Otlet, Bush, Engelbart o

Nelson en un formato que ya utilizamos, el navegador. El navegador es la interfaz estándar, permite que se utilicemos cualquier tipo de ordenador o dispositivo. El navegador es una meta aplicación ya que permite la existencia de múltiples aplicaciones basadas en los paradigmas de interacción persona-máquina que ha estandarizado el navegador. Como usuarixs de la Web, del navegador, somos además consumidorxs y productorxs de datos abiertos vinculados.

En este punto, hemos asumido colectivamente que la mayor parte de estos datos no nos pertenecen, no podemos acceder a ellos y no podemos hacer nada al respecto. Las tecnologías de la Web no pertenecen a ninguna mega corporación, son estándares libres y abiertos impulsados por el *W3C*, la ciudadanía, gobiernos o instituciones públicas pueden impulsar redes sociales, aplicaciones, proyectos que apuesten por la alfabetización y la participación como hace la *Wikipedia*, uno de los casos más claros de éxito, y también uno de los pocos, de conocimiento común colaborativo, con todos sus defectos y todas sus virtudes. Y sin embargo, no es la *Wikipedia* sino *Facebook* o *Whatsap* quienes dominan el intercambio de información, los depositarios de los datos personales de lxs usuarixs de sus servicios que narran su vida diaria. Toda esa información acompañada de geolocalización y horarios, dos tipos de datos muy relevantes y clásicos: tiempo y espacio asociado a personas. De esta forma, no necesitan realizar encuestas para saber cuántas horas te conectas a Internet o en qué horario lo haces o si lo haces en el trabajo, en casa o fuera de casa: los datos están respondiendo esas preguntas que antes realizaban las encuestas presenciales o telefónicas. Además, toda esa información pone en relación dos o más personas, en cuanto destinatarias, lectoras o valoradoras de los mensajes, y se acompaña de entidades que se pueden identificar, como una película, un libro, una ciudad, una comida, etc.

La nube de datos vinculados abiertos cuenta con la *Wikipedia* como una de sus fuentes públicas más importantes, pero hay mucha información pública que puede y debe ir incorporándose a esa nube de datos. Normalmente son formatos de almacenamiento de información tradicional y colaborativa los que se han volcado en el movimiento de los

datos vinculados abiertos. La *Wikipedia*, como ejemplo de enciclopedia; *musicbrainz*, de discoteca; *archive.org*, de biblioteca web; *last.fm*, de consumo musical; etc. No hay, sin embargo, muchos medios que participen, aunque sí algunos ejemplos como hemos mencionado, *NYT*, *BBC* o *The Guardian*. Los medios lo utilizan de muchas formas, bien para ofrecer la programación de un canal televisivo o para disponer de los artículos publicados. A veces incluso se permite la reutilización de estos datos. La cuestión de las licencias depende del proyecto, de la entidad que los ponga a disposición de la nube de datos vinculados. Lo importante es que vincule datos, que permita, como en el hipertexto, posibilidades de uso y/o reutilización de la información. El campo de experimentación está igualmente abierto a investigadorxs sociales, de humanidades... cualquiera puede aplicarlo a su ámbito práctico. Y convendría que se realizaran más reflexiones críticas, incluso desde quienes aplican en primer lugar las tecnologías, lxs llamadx *técnicxs*, programadorxs, desarrolladorxs de software, ingenierxs... quienes pueden traducir de primera mano las tecnologías si las consideran también en términos sociales y no solo técnicos [42].

El mapa conceptual excede de la propia Web Semántica para abordar fenómenos que han permitido su definición o extensión y que están relacionados con los principios de la inteligencia artificial, la organización del conocimiento, el procesamiento de la información, la intercomunicación de dispositivos, las redes de comunicaciones. De esta manera, hardware, software, redes o hipertexto son parte del mapa. Creo que cualquier estudio sobre la Web Semántica debería tener en cuenta estos aspectos, los cuales no se encuentran tan alejados de las Ciencias de la Información como los planes de estudio nos han indicado.

Esta postura epistemológica pretende trascender el uso instrumental de la tecnología que va a servir para abordar cualquier estudio que se realice sobre estas tecnologías. Si cuando se investiga sobre periodismo conviene abordar no solo el uso instrumental o el análisis de los mensajes sino también los medios como tales, las tecnologías implicadas, los lenguajes periodísticos o la estructura de los medios, con la Web Semántica no debe-

ríamos detenernos solo en una página web, un contenido con formato de página web, sino también en los lenguajes, protocolos, vocabularios, tecnologías hardware y software que permiten su existencia, desarrollo y alcance. ¿Se puede hacer un uso instrumental sin conocer las tecnologías que lo posibilitan? ¿Modifica los contenidos esta tecnología? ¿Modifican el uso de la información? ¿Se pueden utilizar esas tecnologías de otras formas? El único cuestionamiento similar, pensamiento crítico y abordaje epistemológico sobre la Web lo he encontrado en Corinna Bath cuando investigaba sobre tecnología y género. El resto de estudios son relatos técnicos, usos posibles, casos de éxito, etc.

Me sorprende no haber encontrado en la amplísima producción científica anglosajona en nuevas tecnologías y por ende en la Web, alguna investigación relacionada, aunque no descarto que pueda existir, porque así nos lo demuestra la historia, que existan propuestas similares, complementarias o incluso iguales que con el tiempo y la difusión de las ideas, se encuentran y confluyen.

Los estudios sobre cibernética de segundo orden [?] aportan una base metodológica y epistemológica para la realización de estos mapas conceptuales, y debería seguir el camino marcado en ese sentido para evaluar la producción de "conocimiento" que se declara con la Web Semántica y combinarlo con los estudios que se han realizado desde biblioteconomía y documentación e ingeniería ontológica, pero también para construir un mapa invariante a la descripción"[61].

Lxs investigadorxs del ámbito de la biblioteconomía y la documentación han continuado la misión que Otlet imaginaba y desarrollan una apasionante actividad por el conocimiento abierto, el acceso abierto u *Open Access*, la reutilización de los datos, el estudio de los metadatos, la recuperación de la información y la organización del conocimiento [159]. También desde la ingeniería ontológica se abordan proyectos y desarrollos concretos con tecnologías semánticas [19].

Como navegante de la Web y como investigador, he intentado no contaminar este mapa

con el zapping o cognición basada en retales"[113] y he buscado mostrar las "potencialidades libertadoras y regenerativas aun por explotar" de la puesta en común de saberes. Esta potencialidad está sin duda, también, en la Web Semántica, este conjunto de tecnologías que pueden poner orden en la confusión, un escenario idóneo para experimentar esa estrategia social de organización del conocimiento y recuperar la utopía de quien quiere compartirlo con sus iguales, que son las personas que habitamos este mundo, aportando reflexividad, crítica, enfoque de género, atención a las minorías y componiendo los cimientos que permitan las narraciones más completas y complejas [165].

Utilizar la tecnología desde estrategias sociales de organización del conocimiento, crear mecanismos lógico-semánticos para superar la reducción de la fragmentación provocada por el hipertexto restaurando el texto y el contexto, instaurar el trabajo en red contributiva socializando los avances y los acervos, asumiendo la complejidad como única aproximación que permite lo global sin dañar lo particular, lo local sin negar lo general. Le queda el camino de restituir un total dominio y responsabilidad de sus actos de aparecer como identificado y digno hermeneuta, recuperar la utopía del bibliotecario ilustrado y hacerlo compatible con los dispositivos de la tecnocultura, introducir la reflexividad en los procesos de análisis, representación, clasificación y registro imprimiendo de metacognición y explicitación su labor, ser el genuino y prestigioso gestor de la exomemoria humana

[113]

Se confía mucho en la Web Semántica, demasiado. La confianza viene de las posibilidades de representar información, contenidos, datos, primero con *SGML* como meta lenguaje de marcas, luego con *XML* un metalenguaje para definir reglas, atributos y las relaciones (una gramática), *RDF* para mostrar las relaciones y finalmente *OWL* para formalizar el significado. Pero si el significado no está en las palabras sino en la mente de quien las procesa, si el significado no es un fenómeno representacional (lingüístico-

gramatical) sino de naturaleza metarrepresentacional (cognitivo-pragmático) [52], ¿estaremos ante un término tan discutido como *KDD*, *Knowledge Discovery in Databases* o extracción del conocimiento de las bases de datos [272, p. 61]? Una lectura del término Web Semántica no debería alertarnos tanto como el anterior. Sin embargo, los retos son tan ambiciosos que dejan de lado las prácticas que ya se producen con la semantización de los contenidos, con la descomposición en datos cargados de significado. Quizás ocurra como con *inteligencia artificial*, que proponían el término alternativo *simulación cognitiva* [52] para escapar de las dudas que producía. La fascinación por las palabras poderosas y por los términos compuestos resulta muy habitual en la tradición científica anglosajona.

Otras metáforas que entran en juego son las espaciales, formalizadas en los programas de visualización de contenido web *Internet Explorer* y *Netscape Navigator*, para precisamente eliminar la consideración fundamental de la percepción humana, la distancia espacial entre el sujeto y el objeto [155, p. 143]. La Web tal como la concebimos es el medio ideal para la representación, visual a través de su interfaz gráfica y su imitación en las aplicaciones para móviles, del conocimiento a través de su forma de funcionar. Si la fotografía, el cine, la televisión, el ordenador, el hipermedia y los videojuegos son los orígenes de la estética de los nuevos medios [155], ¿dónde están los orígenes para interactuar con la gestión de los contenidos que nos propone la Web Semántica? En las bibliotecas, en las hemerotecas, en las bases de datos, en las redes de comunicación, en la historia de la programación, en los artefactos que diseñaron Otlet, Bush o Engelbart.

Debería observarse la recuperación (del conocimiento) desde una perspectiva discursiva y cultural dado que el conocimiento es una construcción que implica una lógico-semántica compleja articulada desde usos reales en marcos de sensibilidad colectiva [114, p. 15]. Espero que mi trabajo sirva al menos para que otrxs observadorxs puedan participar en la realización del criterio de validación de explicaciones científicas" [61].

---

# Apéndice A

## Glosario

Acerca del glosario:

- Las palabras normalmente comienzan por mayúscula y las demás letras son minúsculas.
- Si todas las letras son mayúsculas es porque se trata de una sigla o un término especial
- Si la primera letra es en minúscula es porque es un término especial que se escribe así, en minúsculas.

ABNF *Augmented Backus-Naur Form*, forma Backus-Naur aumentada, ver *BNF*

ACTA *Anti Counterfeiting Trade Agreement*, acuerdo comercial contra la falsificación.

Ada lenguaje de programación utilizado en industria aeroespacial, llamado así en ho-

menaje a la autora del primer lenguaje informático, Ada Lovelace.

**Agente** ser refiere a un software que funciona de forma continua y autónoma según unas reglas que le definen, en un medio particular donde habitan más agentes y procesos. Cumple una función específica de representación de un usuario o de un software. También se habla de *agente humano* para referirse a un usuario.

**AI** *Artificial Intelligence*, inteligencia artificial.

**AJAX** *Asynchronous JavaScript + XML*, *JavaScript* asíncrono + *XML*, es una forma de actualizar contenidos en páginas web a través de distintas tecnologías como *HTML*, *CSS*, *JS*, *DOM*, *XML*, *XSLT* y *XMLHttpRequest*. Se utiliza mucho con datos en *JSON*.

**ANSI** *American National Standards Institute*, instituto estadounidense de estándares nacionales.

**API** *Application Programming Interface*, interfaz de programación de la aplicación, conjunto de protocolos, rutinas y herramientas para relacionarse con una aplicación.

**API REST** *API* que sigue el estándar *REST*.

**AR** *Augmented Reality*, realidad aumentada, la disposición de una capa intermedia entre la realidad y las personas a través de dispositivos móviles que permiten incorporar a las aplicaciones datos de la realidad.

**ARIN** *American Registry for Internet Numbers*, registro americano para números de Internet.

**ARML** *Augmented Reality Markup Language*, lenguaje de marcas de la realidad au-

mentada.

ARPA *Advanced Research Projects Agency*, agencia de proyectos de investigación avanzados de EE.UU. de Norteamérica.

ARPANET sistema de red de trabajo interconectado, la primera Internet.

Array sinónimo de vector o matriz, tipo de estructura de datos que contiene varios elementos, ordenados de alguna manera.

ASA *American Standard Association*, comité estadounidense de estándares, llamado *USASI* desde 1969 y finalmente *ANSI*.

ASCII *American Standard Code for Information Interchange*, código estadounidense estándar para el intercambio de información, código de caracteres basado en alfabeto latino, usa 7 *bits* para representar 128 valores.

ASP *Active Server Pages*, motor de *scripts* del lado del cliente para generar páginas web, realizado por *Microsoft*.

ASP.NET entorno de aplicaciones web del lado del servidor diseñado para el desarrollo web, realizado por *Microsoft* con licencia *Open Source*.

Blog contracción de *web log*, literalmente, registro web, sinónimo de diario personal en la web o cuaderno de bitácora, una de las aplicaciones que nacieron en la *Web 2.0* y que han perdurado hasta la actualidad.

BM Banco Mundial.

BNF *Backus-Nauer Form*, forma Backus-Nauer, una metasintaxis para expresar gramáticas libres de contexto, una manera de representar lenguajes formales.

- Boolean** booleano, tipo de valores *true* o *false*, verdadero o falso, o 0 o 1.
- Browser** literalmente, buscador, lo conocemos como navegador porque hemos aprehendido la misma metáfora de navegación por la Web que nos sugería el programa *Netscape Navigator* de la empresa *Netscape*, ahora *Fundación Mozilla*.
- BY** tipo de licencia *CC* que implica atribución de la autoría.
- CAR** *Computer Assisted Reporting*, periodismo asistido por ordenador, definido por primera vez en 1952 por la *CBS* para hablar de resultados electorales.
- Carpeta** en un ordenador, en un programa gráfico de la estructura de archivos, entendemos por carpeta o *folder* al directorio de Unix, es una metáfora para entender los directorios. En una web es una carpeta o directorio del sitio. Puede ser un directorio más, por ejemplo, para albergar todas las imágenes *img*, o puede ser un espacio organizativo propio que lo convierte en un sitio web.
- CC** *Creative Commons*, licencias para trabajos artísticos, como por ejemplo el texto de una página web.
- cçTLD** *country code Top Level Domain*, código de país de *TLD*.
- CDATA** *Character Data*, datos de caracteres que no se interpretan según la *DTD*, por lo que no siguen sus restricciones. Lo contrario son los *PCDATA*
- CERN** *Conseil Européen pour la Recherche Nucléaire*, organización europea para la investigación nuclear, donde trabajó como investigador a Tim Berners-Lee, donde desarrollo la Web.
- Colabulario** del inglés *collabulary*, la fusión de *collaborative vocabulary*, vocabulario colaborativo.

- CGI *Common Gateway Interface*, interfaz de acceso común a datos, permite a un cliente solicitar datos de un programa ejecutado en el servidor. Las aplicaciones que se ejecutan en el servidor también se denominan así.
- Charset conjunto de caracteres que se establecen en una web.
- CLI *Command Line Interface*, interfaz de línea de comandos, permite interactuar con los dispositivos electrónicos a través de comandos, órdenes.
- Cliente en una arquitectura cliente/servidor, el usuario que pide, solicita o usa un servicio.
- CMS *Content Management System*, sistema de gestión de contenidos, aplicación web que utiliza una interfaz web para crear contenido que se almacena en una base de datos y se genera a partir de un motor de la aplicación y de plantillas de contenido.
- COBOL *COmmon Business-Oriented Language*, lenguaje común orientado a los negocios, creado en 1959 por Grace Hooper, es un lenguaje de programación compilado, imperativo, procedural y, desde 2002, orientado a objetos. Se utiliza en empresas, finanzas, administración.
- Colabulario traducción de *collabulary*
- Collabulary *colabulario*, contricción de *collaborative vocabulary*, se refiere a la creación de un vocabulario de términos, una taxonomía empleada por un sitio web donde no hay una sola persona autora sino que se permite la colaboración de un grupo amplio.
- Cookie literalmente, galleta, se refiere a un código que mandan algunas páginas web a los navegadores para guardar información de la sesión en el navegador, por lo que permite mantener una sesión abierta entre esa página web y la navegación de lxs

usuarixs, incluso cuando éstxs no utilizan esa página.

CRUD *Create, Read, Update, Delete*, crear, leer, actualizar y borrar, las cuatro operaciones básicas de una base de datos, también presentes en *REST*.

CSS *Cascading Style Sheet*, hoja de estilo en cascada, utilizado para dar estilo de presentación a las páginas web.

CSV *Comma Separated Values*, valores separados por comas, un tipo de archivo de datos que almacena en texto plano datos de una tabla, donde los valores de las filas se separan por comas. Como sufijo de tipo de archivo, se escribe en minúsculas.

Cyborg *cybernetic organism*, organismo cibernético.

DAML *DARPA Agent Markup Language*, lenguaje de marcas de agente *DARPA*

DARPA *Defense Advanced Research Projects Agency*, agencia de proyectos de investigación avanzados de defensa de EE.UU. de Norteamérica.

Datacube cubo de datos, se refiere a datos multidimensionales.

Datatype tipo de datos.

Date fecha, tipo de datos para fechas.

DBpedia *DataBase pedia*, enciclopedia basada en los datos semánticos estructurados de la *Wikipedia*.

DC *Dublin Core*, véase *DCMI*.

DCMI *Dublin Core Metadata Initiative*, es una iniciativa de metadatos llamada así por

el primer lugar donde se reunieron, Dublin, en el estado de Ohio (EE.UU. de Norteamérica)

DDML *Document Definition Markup Language*, lenguaje de esquema *XML*. En desuso sobre *XML Schema* o *RELAX NG*.

Deep link enlace profundo, nombre con el que se conoce a la práctica de enlazar un documento o parte de un documento de una web que no es la portada del sitio web, lo cual es perfectamente compatible con la Web pero no con la política de algunos sitios web.

DIY *do it yourself*, hazlo tú mismx, filosofía *punk* que recoge el periodismo de datos como seña de identidad.

DIWO *do it with others*, hazlo con otrxs, filosofía colaborativa que recoge el periodismo de datos en contraposición a la figura del lobo solitario.

Documento web es un documento de texto escrito en lenguaje *HTML* y declarado como *HTML*, es sinónimo de página web.

DL *Description Logic*, lógica de descripción.

DLO *Document-like Object*, literalmente "objeto documental" "objeto como si fuera un documento", se utiliza en bibliotecas digitales y en la Web en general para referirse a objetos de información discreta que se pueden describir con metadatos, tal como propuso el *DCMI* en su primer taller de Ohio (EE.UU.).

DMCA *Digital Millenium Copyright Act*, ley sobre derechos digitales del milenio, donde se blindaban los derechos de los autores en Internet.

DNS *Domain Name System*, sistema de nombres de dominio, la identificación nomi-

nal de las *IP* de Internet.

DoD *Department of Defense*, departamento de defensa de EE.UU.

DOM *Document Object Model*, modelo de objetos del documento, describe el documento *HTML*. Se utiliza en *JS*.

Double doble, tipo de elemento o atributo de *XML Schema*.

dpi *dots per inch*, puntos por pulgada, véase *ppp*

DSD *Document Structure Description*, descripción de la estructura de documento, es un lenguaje de esquema para *XML*.

DSDL *Document Schema Definition Languages*, lenguajes de definición de esquemas de documentos, entorno para realizar tareas de validación de distintos tipos en documentos *XML*.

DTD *Document Type Definition*, definición de tipo de documento, se utiliza para declarar la sintaxis de un documento *SGML* o *XML*.

EBNF *Extended Backus-Naur Form*, forma Backus-Naur extendida, ver *BNF*.

ECMA *European Computers Manufacturers Association*, asociación europea de fabricantes de ordenadores, cambio su nombre a *Ecma International* al convertirse en entidad global.

ECMAScript lenguaje estandarizado por *Ecma International* como *ECMA-262*, más conocido como *JavaScript*.

EndPoint punto final, se refiere a una *URL* donde se pueden realizar búsquedas *SPARQL*.

Entidad en *HTML*, sirve para representar caracteres no *ASCII*.

Equipo sinónimo de host, albergador, ordenador de una red que alberga uno o varios servicios de la red.

ENIAC *Electronic Numerical Integrator And Computer*, primer ordenador electrónico de propósito general, anunciado en 1946 y utilizado para usos militares.

Endpoint literalmente, *punto final*, lugar donde se realiza la consulta *SPARQL*

Event evento, acción que realizamos al relacionarnos con el navegador y que *JS* puede capturar para permitir la interacción con el usuario.

Event handler manejador de eventos, la forma como *JS* maneja los eventos para permitir la interacción con el usuario.

Evento véase *event*.

FHS *File Hierarchy Standard*, estándar de jerarquía de archivos, método utilizado por Unix para organizar el sistema operativo.

FQDN *Fully Qualified Domain Name*, nombre de dominio cualificado.

FMI Fondo Monetario Internacional.

FOAF *Friend Of A Friend*, amigo de un amigo, vocabulario *RDF* para expresar datos personales y relaciones entre personas.

Folksonomy folksonomía, la unión de *folk* con taxonomía.

FPI *Formal Public Identifier*, un formato de texto dividido en dos partes que puede

identificar una especificación o un documento. Se utiliza en *DTD*. Consta de dos partes, el identificador de propiedad y el texto. Por ejemplo:

```
//W3C//DTD HTML 4.01//EN
```

Framework marco de trabajo, de desarrollo, entorno.

FTP *File Transmission Protocol*, protocolo de transmisión de archivos.

G8 grupo de los 8 países más ricos del mundo según su *PIB* o producto interior bruto.

GATT *General Agreement on Tariffs and Trade*, acuerdo general sobre aranceles aduaneros y comercio.

GPL *General Free Documentation License*, licencia general de documentación libre.

GML *Generalized Markup Language*, lenguaje de marcas generalizado, lo desarrolló *IBM* con una sintaxis *EBNF* para organizar documentos.

GN *GeoNames*, geo nombres, para localización geográfica.

GNU *GNU Not Unix*, acrónimo recursivo que dice que *GNU* no es *Unix*, proyecto de sistema operativo que suele utilizar *Linux* como *kernel* o núcleo del sistema.

GPL *General Public License*, licencia pública general, una licencia para software libre del proyecto *GNU*.

GRAPH grafo, suele referirse a grafos *RDF*.

GRAPH IRI una *IRI* involucrada en el protocolo *SPARQL*

GUI *Graphical User Interface*, interfaz de usuario gráfico, permite interactuar con dispositivos electrónicos a través de iconos visuales.

HATEOAS *Hypermedia As The Engine of Application State*, hipermedia como motor del estado de la aplicación, una característica de *REST* que indica que el cliente interactúa con una aplicación de red a través de hipermedia.

Hardware conjunto de los componentes que integran la parte material de un ordenador u otros dispositivos electrónicos.

hcard formato de *Microformats* para publicar información personal de personas, empresas u organizaciones en *HTML*. Basado en *vCard*.

Hiperenlace o enlace de hipertexto, la forma de saltar de un contenido a otro en la Web.

Hipermedia el medio más allá del medio, la lógica hipertextual en los contenidos audiovisuales; lógica hipertextual para representar relaciones de un recurso con otros recursos en la Web Semántica.

Hipertexto el texto que está más allá del texto, posible gracias a los hiperenlaces o enlaces de hipertexto. El concepto lo realizó el inventor de Xanadú.

HTML *HyperText Markup Language*, lenguaje de marcado de hipertexto, el lenguaje de las páginas web. Como sufijo de tipo de archivo, se escribe en minúsculas.

HTTP *HyperText Transmission Protocol*, protocolo de transmisión de hipertexto, una de las bases de la Web.

HTTPS *HTTP Secure*, *HTTP* seguro, que utiliza el protocolo *SSL*.

Host ordenador *anfitrión* que aloja servicio(s) en una red, también se puede conocer

como equipo. De aquí provienen los términos *alojamiento* (*hosting*).

Hosting alojamiento, en servicios de Internet.

Hyper Card aplicación de *Macintosh* consistente basada en tarjetas que podían contener enlaces de hipertexto.

IAL *International Algebraic Language*, escrito por John Backus a partir de las reglas generativas de Chomsky.

IANA *Internet Assigned Names Authority*, autoridad para los nombres asignados de Internet, responsable de la coordinación del *DNS* raíz, direcciones *IP* y otros recursos. Actualmente depende de *ICANN*.

IBM *International Business Machines corporation*, empresa internacional de máquinas de negocios.

ICANN *Internet Corporation for Assigned Names and Numbers*, corporación de Internet para los nombres y los números asignados

IEEE *Institute of Electrical and Electronics Engineers*, instituto de ingeniería eléctrica y electrónica, dedicada a la estandarización y desarrollo en áreas técnicas, fundado en 1884.

IETF *Internet Engineering Task Force*, grupo de ingeniería de Internet, organismo que elabora normas técnicas para el funcionamiento de Internet.

Integer enteros, tipo de valores que son números enteros.

Internet of Things Internet de las cosas.

IoT *Internet of things*, Internet de las cosas.

IP *Internet Protocol*, se refiere al protocolo de Internet y también es sinónimo de *dirección IP*, la dirección *IPv4* de un equipo conectado a Internet.

IPv4 protocolo *IP*, versión 4.

IPv6 protocolo *IP*, versión 6.

IRI *Internationalized Resource Identifier*, identificador de recursos internacionalizado, lo definió *IETF* en 2005 como un nuevo estándar de Internet sobre el que definir el esquema *URI*, *RFC 3987*. *URI* está limitado a caracteres *ASCII* mientras que *IRI* contiene *Universal Character Set (Unicode/ISO 10646)* que incluye caracteres chinos, japoneses, coreanos o cirílicos.

ISAPI *Internet Server API*, *API* de servidor de Internet.

ISP *Internet Service Provider*, proveedor de servicios de Internet.

JavaScript o *JS*, lenguaje interpretado para ofrecer funcionalidades, dinamismo e interacción a los documentos web.

JS acrónimo de *JavaScript*. Como sufijo de tipo de archivo, se escribe en minúsculas.

JSON *JavaScript Object Notation*, notación de objeto *JS*, se utiliza como formato de datos. Como sufijo de tipo de archivo, se escribe en minúsculas.

JSP *Java Server Pages*, servidor de páginas de *Java*.

Kernel núcleo del sistema operativo de un ordenador o dispositivo electrónico.

KDD *Knowledge Discovery Databases*, descubrir el conocimiento de bases de datos.

KMS *Knowledge Management System*, sistema de gestión del conocimiento realizado en 1981 que permitía escritura colaborativa y compartición de información a través de hiperenlaces.

LAN *Local Area Network*, red de área local.

Librería traducción literal de *library*, biblioteca, se ha popularizado librería. Se trata de código que puede agruparse para su posterior reutilización.

Linked Data datos vinculados.

Linked Open Data datos abiertos vinculados.

List lista, listado, tipos de datos que son listados; elemento de *HTML*.

Literals literales, valores literales, cadenas de caracteres que se toman como tipo de datos literales. Por ejemplo, un literal 1984 no es un número ni un año.

Literales ver Literals.

LOD *Linked Open Data*.

Matriz véase *array*.

MAN *Metropolitan Area Network*, red de área metropolitana.

MAU *multistation access unit* o unidad de acceso a multiestación.

Metadata o metadatos, término para referirse a los datos sobre los datos, sirve para apor-

tar más información semántica a un documento. Fue explicado por primera vez por Philip R. Bagley en 1968 en su tesis *Extension of Programming Language Concepts*.

**Microblogging** el acto de escribir en un *microblog*, un blog que permite un número de caracteres más limitado. El ejemplo más conocido es *Twitter*.

**Microdata** especificación *WHATWG HTML* para anidar metadatos en el contenido de las páginas web.

**Microformats** patrones para incluir metadatos en *HTML*.

**MIME** *Multipurpose Internet Mail Extensions*, extensiones de correo de Internet multipropósito, convenciones para el intercambio de archivos a través de Internet, el tipo de contenido se especifica en el atributo *Content-Type*.

**MIT** *Massachusetts Institute of Technology*, instituto de tecnología de Massachusetts.

**MVC** *Model View Controller*, modelo vista controlador, un estilo de desarrollo web.

**Namespace** espacio de nombre, conjunto de nombres iguales en un dominio determinado que se pueden identificar de alguna manera. Ver también *XMLns*.

**NAT** *Network Address Translation*, traducción de direcciones de red.

**Navegador** ver *Browser*.

**NC** *Non Comercial*, no comercial, tipo de licencia *CC* que indica que no se permite uso comercial de la obra.

**NLP** *Natural Language Processing*, procesado de lenguaje natural.

**Nodo** En informática se considera a un nodo un elemento de una lista enlazada, un árbol o un grafo en una estructura de datos con sus propias características y con relaciones de referencia con otros nodos, muchos o pocos, pero al menos siempre con uno. Esta relación se denomina puntero. Si solo tiene un puntero, la única estructura de datos que se puede construir es una lista, pero si tiene más, se pueden construir estructuras más complejas como árboles o grafos. En redes se habla de un nodo de la red, uno de los equipos conectados. En la Web 1.0 se llamaba nodo a cada documento de un sitio web; esta acepción ha vuelto en la Web Semántica a actualizarse para llamar nodos a *URI* en un grafo *RDF*.

**NRL** *Natural Rule Language*, lenguaje de reglas naturales, utilizado para modelar proyectos de integración.

**ns** *namespace*, véase *Namespace*

**NS** *NameServer*, servidor de nombres.

**OData** *Open Data Protocol*, permite consultas e interoperabilidad en *API RESTful*.

**OK** *Open Knowledge*, organización antes conocida como *OKFN*.

**OKFN** *Open Knowledge Foundation Network*, red de la fundación del conocimiento abierto.

**ONG** Organización no gubernamental.

**Ontología** colección de enunciados que define relaciones entre conceptos y especifica reglas lógicas para razonar con ellos.

**Open Source** código abierto, se refiere al software que muestra el código fuente.

OWL *Ontology Web Language*, lenguaje de ontologías de la web, en realidad debería escribirse como *WOL* pero por alguna razón se ha difundido *OWL*. Son una de las capas del pastel de la Web Semántica. Como sufijo de tipo de archivo, se escribe en minúsculas.

P2P *peer to peer*, de igual a igual, se refiere a redes donde la conexión se produce entre pares iguales.

Página web traducción de *web page*, se trata de una de las páginas web o documentos *HTML* de un sitio web.

Parser analizador sintáctico.

PC *Personal Computer*, ordenador personal.

PCDATA *Parsed Character Data*, datos de caracteres parseados, son datos que se ajustan a las restricciones de la DTD, por lo que requerirán de entidades para la representación de algunos caracteres.. Frente a estos, los *CDATA* permiten escribir sin restricciones.

PD *Public Domain*, dominio público, licencia *CC* de dominio público.

Permalink *permanent link* o enlace permanente, se refiere a la *URL* que un *CMS* construye para el contenido.

PHP *PHP Hypertext Preprocessor*, acrónimo recursivo para procesador de hipertexto *PHP*.

POI *Point of Interest*, puntos de interés.

POO Programación orientada a objetos.

- POP<sub>3</sub> *Post Office Protocol 3*, protocolo de oficina postal 3, protocolo de recepción de correo electrónico.
- ppp puntos por pulgada, sistema de medida de las resoluciones de las pantallas, dispositivos de salida de datos.
- PPV *pay per view* o pago por visión, sistema de monetización de los contenidos culturales en televisión, también para Internet.
- QR *Quick Response Code*, código de respuesta rápida.
- RDDL *Resource Directory Description Language* o lenguaje de descripción de recursos del directorio.
- RDF *Resource Description Framework*, marco de descripción de recursos, recomendación *W<sub>3</sub>C* diseñada como metamodelo de metadatos. Como sufijo de tipo de archivo, se escribe en minúsculas.
- RDFa *RDF in attributes*, *RDF* en atributos, una forma de incluir *RDF* en las páginas web.
- RDFS *RDF Schema*, esquema *RDF*.
- RELAX NG *REgular LAnguage for XML Next Generation* lenguaje de esquema para *XML*, especifica el patrón de la estructura y contenidos de un documento *XML*. Aunque *RELAX NG* es un documento *XML*, ofrece una sintaxis distinta, inspirada en *BNF* y expresiones regulares, comparte sintaxis de *DTD*.
- REST *REpresentational State Transfer*, transferencia de estado representacional, arquitectura de desarrollo que se apoya en *HTTP*. Se utiliza para crear *API*.

RESTful una arquitectura cliente/servidor que funciona plenamente con el estándar *REST*.

RIA *Rich Internet Application*, aplicación de Internet rica, se refiere a una aplicación web que por sus características o funcionalidades se asemeja con una aplicación de escritorio.

RIF *Rule Interchange Format*, formato de intercambio de reglas.

RFC es el acrónimo de *Request for Comments* (solicitud o petición de comentarios), y se trata de especificaciones abiertas -de ahí su nombre, quienes lo escriben solicitan comentarios al respecto- de cómo hacer determinadas cosas en Internet. Los documentos los aprueba finalmente el *IETF*, momento en el que se consideran estándares. *IETF* reúne los *RFC* en su sitio web <http://tools.ietf.org/html/> Fueron creados por Steve Crocker en 1969.

RFID *Radio Frequency Identification*, identificador de radio frecuencia.

RSDL *RESTful Service Description Language*, lenguaje de descripción de servicios *RESTful*.

ROA *Resource-Oriented Architecture*, arquitectura orientada a recursos.

RSS *Really Simple Syndication* o *sindicación -de contenidos- muy simple*, en su versión 2.0, pero también *RDF Site Summary* o *sumario del sitio en RDF*, en su versión 0.9 y 1.0 y *Rich Site Summary* en su versión 0.91.

RTM *Release to manufacturing*, estado de desarrollo de software cuya versión se encuentra lista para comercialización.

RuleML *Rule Markup Language*, lenguaje de marcas de reglas.

SA *Share Alike*, compartir igual, tipo de licencia *CC* que indica que las obras derivadas deben compartirse con la misma licencia.

Schema.org conjunto de esquemas para incluir datos estructurados en las páginas web.

Schematron lenguaje de validación basado en reglas para hacer aserciones sobre la presencia o ausencia de patrones en árboles XML.

Scovo *Statistical Core Vocabulary*, vocabulario estadístico.

Script guión, conjunto de instrucciones que operan de determinada manera. Cuando un *script* es muy grande, suele llamarse "programa".

SDMX *Statistical Data and Metadata eXchange*, intercambio de metadatos de datos estadísticos.

Servidor en una estructura cliente/servidor, el prestador del servicio.

Servlet programa en *Java* que extiende las posibilidades de un servidor.

Shell interfaz de usuario para acceder a servicios del sistema operativo.

Sitio web traducción de *website*, un conjunto de páginas web o documentos *HTML* que componen una unidad organizativa y accesible a través de un dominio, un subdominio o una carpeta del dominio o del subdominio. Por tanto, dentro de un mismo dominio puede haber varios sitios web.

SKOS *Simple Knowledge Organization System*, sistema de organización simple del conocimiento.

SGML *Standard Generalized Markup Language*, lenguaje de marcas estándar genera-

lizadas, precursor de *HTML* y *XML*, metalenguaje para la creación de lenguajes de descripción, como por ejemplo lo anteriores o *DocBook* para documentación científica. Se basa en *GML*.

**SMTP** *SMTP* (*Simple Mail Transfer Protocol*, protocolo de transferencia de correo simple, protocolo de envío de correo electrónico.

**SOA** *Service-Oriented Architecture*, arquitectura orientada a servicios, donde clientes y servidores interactúan por una interfaz compartida.

**Software** los programas que funcionan en un ordenador u otros dispositivos electrónicos.

**SOLO** *Structure of Linked Objects* o estructura de objetos vinculados, el software en que se basaron la gama de productos de la empresa *NetObjects*.

**SOX** *Schema for Object-Oriented XML*, esquema para *XML* orientado a objetos.

**SPARQL** acrónimo recursivo para *Sparql Protocol And RDF Query Language*, protocolo *Sparql* y lenguaje de consultas *RDF*.

**SSL** *Secure Sockets Layer*, capa segura de conexión para protocolos de Internet como *HTTP*.

**sTLD** *sponsored TDL*, *TLD* patrocinado.

**String** cadena, por cadena de caracteres. Ver literales.

**SWRL** *Semantic Web Rule Language*, lenguaje de reglas de la Web Semántica.

**TCP** *Transmission Control Protocol*, protocolo de control de la transmisión, utilizado

en redes *TCP/IP* como Internet.

TLD *Top Level Domain*, dominio de alto nivel.

TXT abreviatura de *text*, se refiere a datos que están en formato de texto plano. Como sufijo de tipo de archivo, se escribe en minúsculas.

UDC *Universal Decimal Classification*, clasificación decimal universal, creada por Paul Otlet y Henri La Fontaine a finales de siglo XIX.

UDP *User Datagram Protocol*, protocolo de datagrama del usuario.

Unicode estándar de codificación de caracteres que permite el tratamiento informático de casi todos los caracteres de lenguas conocidas.

Union en inglés, asociación, tipo de elementos y atributos de *XML Schema*.

UNIVAC *Universal Automatic Computer*, ordenador de propósito general creado por *Eckert-Mauchly Computer Corporation*

URI *Unified Resource Identifier*, identificador de recursos unificado.

URL *Uniform Resource Locator*, localizador uniforme de recursos.

URN *Uniform Resource Name*, nombre uniforme de recursos.

US-ASCII *United States ASCII*, *ASCII* de los EE.UU.

USASI *United States of America Standard Institute*, instituto estadounidense de estándares.

UTC *Coordinated Universal Time*, tiempo universal coordinado.

UTF-8 codificación de caracteres basada en Unicode, contempla casi la totalidad de los caracteres de lenguas conocidas.

Vector véase *array*.

W<sub>3</sub> sinónimo de *World Wide Web*

W<sub>3</sub>C de *World Wide Web Consortium* o Consorcio de la Web, es la organización encargada de regular la Web, las tecnologías que la componen.

WAI *Web Accessibility Initiative*, iniciativa de accesibilidad en la Web, iniciativa del W<sub>3</sub>C para promover y aumentar la accesibilidad del contenido web.

WAN *Wide Area Network*, red de área extensa.

WCAG *Web Content Accessibility Guidelines*, directrices de accesibilidad del contenido web

Web la Web con mayúsculas es el nombre corto para referirnos a la *World Wide Web*, se refiere al conjunto de todas las páginas web, sitios web, documentos HTML, documentos web, a la Web en su totalidad.

web la web con minúsculas es el nombre corto para referirnos a *página web* pero es el sinónimo de *sitio web*.

Web page página web.

Website sitio web.

WHATWG *Web Hypertext Application Technology Working Group*, grupo de trabajo de la tecnología de aplicaciones de hipertexto para la Web, salió de entidades que participaban en el desarrollo de *XHTML2* en el *W3C* y no estaban de acuerdo con el desarrollo del proyecto, por lo que iniciaron este grupo para desarrollar *HTML5*, ahora conocido por ellos como *HTML* y en el *W3C* como recomendación. *HTML5*

Wiki rápido en hawaiano, un modo de edición y creación rápida de páginas web.

Wikimedia uno de los software más utilizados para *wikis*, utilizado por la *Wikipedia*

WWW de *World Wide Web* o gran tela de araña mundial, es el conjunto de documentos web interconectados por *HTTP* e identificados por URL gracias a DNS.

XFN *XHTML Friends Network*, rede de amigxs en *XHTML*.

XHR abreviatura de *XMLHttpRequest*.

XML *eXtensible Markup Language*, lenguaje de marcas extensible.

XMLHttpRequest o *XHR*, objeto *JavaScript* que ofrece una *API* para el acceso a datos de una *URL* a través de *HTTP*. Se usa con *AJAX*.

XML Schema Esquema *XML*

XMLns espacio de nombres *XML*, recomendación de *W3C* que aporta elementos y atributos en un archivo *XML*. Se puede crear un vocabulario *XML* con el espacio de nombres que se quiera para un dominio, referenciado a una *URI* e identificado con un prefijo.

XRBL *eXtensible Business Reporting Language*, lenguaje de informes de negocio ex-

tensible.

XSDL *XML Schema Definition Language*, lenguaje de definición de esquemas *XML*, otro lenguaje de esquemas.

xsi prefijo convencional para una instancia de *XML Schema*.

XSLT *eXtensible Stylesheet Language Transformations*, lenguaje extensible de transformaciones de estilo, se utiliza en documentos *XML* para aplicar transformaciones de estilo.

YAGO *Yet Another Great Ontology*, otra gran antología.

ZOA *Zones of Authority*, zonas de autoridad.

---

## Apéndice B

### Cronología

1801

- Tèlar de Jacquard Joseph Marie Jackquar inventa en 1801 un telar que utiliza tarjetas perforadas para tejer patrones de tela

1842

- Ada Traduce un artículo sobre la máquina analítica de Charles Babagge y sus notas se consideran el primer documento donde se detalla un lenguaje de programación

1866

- Cable transatlántico Se logra la conexión telegráfica entre América y Europa.

1895

- Repertoire Bibliographique Universel El repertorio bibliográfico universal *RBU* era un intento de Paul Otlet y Henri La Fontaine de desarrollar una bibliografía del conocimiento acumulado en el mundo.

1906

- Microficha Otlet y Robert Goldschmidt inventan la microficha estándar para generar documentación. Se utiliza para reproducir libros, revistas o periódicos.
- Mobile telephone Otlet escribe en *Les aspects du livre* que la telefonía será inalámbrica, como la telegrafía, por lo que seremos testigos de la transformación del libro: todos llevaremos en el bolsillo un auricular que se conectará al centro emisor.

1910

- Mundaneum El sistema de clasificación universal de Otlet buscaba servir la información, por lo que diseñó el *Mundaneum*, sede del archivo, que permitía responder más de mil quinientas peticiones por año por correo postal y proyectó otras formas de acceso, cableado o inalámbrico.

1929

- 6 grados de separación Idea propuesta por Frigyes Karinthy en un relato para evidenciar los avances en la comunicación del siglo XX.

1934

- El libro universal Paul Otlet publica *Traité de documentation* donde propone, entre otras innovaciones, una especie de ordenador de sobremesa que permitiera obtener documentos y anotar las relaciones con otros documentos, por lo que se conformaría el *libro universal* enlaces anotados entre documentos. También

imaginó el acceso remoto a través de un telescopio eléctrico conectado a una línea telefónica que transmitiera una imagen proyectada en una pantalla plana.

1941

- El jardín de los senderos que se bifurcan Cuento de Jorge Luis Borges considerado inspirador para el hipertexto.

1942

- Cibernética Se propone por primera vez el término en un congreso sobre la inhibición cerebral.

1944

- Harvard Mark I O *Automatic Sequence Controlled Calculator ASCC*, ordenador electro-mecánico de propósito general de *IBM*.

1945

- Memex Vannevar Bush escribe en el *Atlantic Monthly* el artículo *As we may think* que describe un dispositivo foto-eléctrico-mecánico llamado *Memex* que amplía la memoria al poder crear y seguir enlaces entre documentos de microfichas.
- Programadoras de ENIAC Jean Jennings Bartik y otras cinco mujeres comienzan a trabajar como operadoras de ENIAC
- Silicio Texas Instruments comienza a trabajar con este material.

1946

- ENIAC Se lanza *Electronic Numerical Integrator and Computer*, primer ordenador electrónico de propósito general, mil veces más rápido que los ordenadores electro-mecánicos, creado por John Adam Presper Eckert y John Mauchly

1947

- Transistor Contracción de *transresistencia*, término acuñado por John Pierce, fue lanzado por John Bardeen, Walter Brattain y William Shockley.

1948

- Bit Aparece el término bit en el artículo de Claude Shannon *Una teoría matemática de la información*, atribuido a John Tukey. También se le atribuyen a él los términos *box-plot* (1977) o diagrama de cajas, muy utilizados en estadística, y *software* (1958).
- Cibernética Norbert Wiener define la cibernética como el estudio científico del control y la comunicación en el animal y la máquina. y sienta las bases para el estudio de todo sistema en términos de comunicación.
- Manchester Mark I Primer ordenador con programa almacenado.
- Teoría matemática de la información Wiener sienta las bases de la cibernética: información, mensajes, líneas de conducta, mecanismos de control o automatización. Claude Elwood Shannon y Warren Weaver dan nombre al modelo *Shannon-Weaver* o *teoría matemática de la información* que integra fuente de información, mensaje, transmisor, señal, canal, ruido, receptor, información en destino, probabilidad de error, codificación, decodificación porcentaje de información o capacidad del canal. Abordan la información desde la estadística y las leyes de probabilidad y definen la entropía o neguentropía de la información.

1949

- EDSAC *Electronic Delay Storage Automatic Calculator*, segundo ordenador digital electrónico con programas almacenados, primer ordenador con comandos y se considera el inicio de la industria de software.

1950

- 6 grados de separación Ithiel de Sola Pool y Manfred Kochen intentar una demostración matemática de la teoría.
- Test de Turing Creado por el propio Alan Turing para medir la inteligencia de una máquina.

1951

- Agrupaciones de código Maurice Wilkes, David Wheeler y Stanley Gill inventan las agrupaciones de código o subrutinas, prolegómeno de las librerías.
- Transistor de contacto Shockley avanza en el transistor con el transistor de contacto
- UNIVAC I Se lanza ya dentro de *Remington Rand* y alcanza la fama por predecir la elección presidencial del año siguiente.

1952

- Computer Assisted Reporting La *CBS* utiliza el término *CAR* o *Computer Assisted REporting* para referirse a la predicción sobre el resultado electoral.

1954

- Transistor de silicio El primer transistor lo desarrolló Morris Tanenbaum en Bell Labs, pero el primero usado comercialmente lo produjo Gordon Teal en Texas Instruments

1955

- FLOW-MATIC Grace Hopper lanza lenguaje de programación basado en compilador.

1957

- Circuito integrado Invento de Jack Kilby de Texas Instruments y Bob Noyce de Fairchild
- FORTRAN *Formula Translating System*, sistema de traducción de fórmulas, lenguaje de programación imperativo de propósito general orientado al cálculo numérico y científico, desarrollado por John Backus en *IBM*.

1958

- IAL Conocido como *ALGOL 58*, desarrollado por John Backus a partir de las reglas generativas de Chomsky

1959

- COBOL Creado por Grace Hopper a partir de *FLOW-MATIC* y de *COMTRAN*, era multiplataforma y con sentencias cercanas al inglés
- Proceso planar Fairchild Semiconductors de Silicon Valley inventa el proceso planar para integrar componentes miniaturizados

1960

- Cyborg Acuñado por Manfred E. Clynes y Nathan S. Kline, se referían a un ser humano mejorado que pudiera vivir en entornos extraterrestres
- Man-Computer Symbiosis Joseph Carls Robnett Licklider, alias *JCR* o *Lick*, publica *Man-Computer Symbiosis*, simbiosis hombre-ordenador, donde expone la interacción necesaria entre ordenadores y usuarios y las tareas que debían realizar los ordenadores. Se le considera pionero de la cibernética, de la inteligencia artificial.

1962

- Conocimiento aumentado Douglas Englebart publica *Augmenting Human Intellect. A conceptual Framework*
- Sistema de comunicaciones robusto a ataques Las Fuerzas Áreas de EE.UU. de Norteamérica piden a un grupo de científicos el diseño de un sistema de comunicaciones capaz de resistir el ataque a una o varias de sus sedes.

1963

- ARPA El Departamento de Defensa (DoD) de los EE.UU. de Norteamérica crea ARPA, luego DARPA, de donde surge ARPANET.
- ASCII Nace el *American Standard Code for Information Interchange*, código de caracteres basado en alfabeto latino.
- BNF Notación de Backus-Naur por las contribuciones de Hohn Backus y Peter Naur a esta metasintaxis para representar gramáticas libres de contexto.

- Intergalactic Computer Network Licklider, director de *Information Processing Techniques Office (IPTO*, oficina de técnicas de procesado de información) de *DARPA*, define una red planetaria abierta a gobiernos, instituciones y ciudadanos.

1964

- Red *Web* Paul Baran elige un diseño de red de tela de araña para ARPA

1965

- Hipertexto Ted Nelson acuña el término Hipertexto en el texto *.A file structure for the complex, the changing and the indeterminate.*<sup>en</sup> la vigésima edición de la conferencia nacional de la Association for Computing Machinery en Nueva York

1967

- 6 grados de separación Stanley Milgram realiza un experimento de campo para comprobar la teoría, que resulta con éxito, y propone *el problema del mundo-pequeño*
- HES Andrew van Dam y Ted Nelson realizan el experimento *Hypertext Editing System HES*, sistema de edición de hipertexto, en la Universidad de Brown con el IBM 2250 donde se muestra la potencia del hipertexto.
- Detroit riot Revueltas que ocurrieron en Detroit donde cuarenta personas murieron y más de mil resultaron heridas. Philip Meyer investiga el suceso con técnicas de análisis de las ciencias sociales y nace el periodismo de precisión.
- Separación información/presentación William Tunnicliffe habla de separación de la información de su presentación. Stanley Rice propone un catálogo universal

de etiquetas para la industria editorial.

1968

- OLS Demostración que Douglas Englebart realizó en 1968 denominada *la madre de todas las demostraciones* por ser varios los inventos que había realizado de su trabajo en el *Augmentation Research Center*, agrupados en el *Online System*: pantalla de mapas de *bits*, un ratón, un teclado o hipertexto. Su influencia es crucial en las interfaces gráficas de usuario.
- Metadatos Definidos por Philip R. Babley para asociar unos datos con otros.

1969

- ARPANET ARPA (Advanced Research Projects Agency) aprueba la creación de ARPANET para realizar investigaciones sobre redes. Conecta cuatro universidades.
- FRESS *File Retrieval and Editing System FRESS*, recuperación de archivos y sistema de edición, desarrollado por Andries van Dam y Bob Wallace en la Universidad de Brown.
- Periodismo de Precisión Se publica el término periodismo de precisión, práctica iniciada por Philip Meyer
- RFC Steve Croker de *UCLA* lanza *RFC 1*. El *RFC 3* es sobre *RFC*. Postel sería el editor de *RFC*
- Sociedad del conocimiento Peter Drucker escribe en *La era de la discontinuidad* sobre la sociedad del conocimiento por la relevancia en el PIB de las empresas del mundo de la cultura.

1970

- Cibernética de segundo orden Se refiere a la cibernética de Heinz von Foerster que habla de sistemas que son capaces de modificarse a sí mismos sin necesidad de guía externa.
- Conmutación de paquetes Donald Davies acuña el término *packet switching* para dividir los mensajes del ordenador en paquetes y enrutarlos independientemente, reconstruidos en destino. Era lo que Paul Baran había denominado *distributed adaptive message block switching* lanza la conmutación de paquetes o *packet switching*, un método que agrupa los datos transmitidos en bloques de tamaño manejable llamados paquetes.
- Unix Dennis Ritchie y Ken Thompson lanzan *Unix* en *Bell Labs*
- Pascal Niklaus Wirth publica el lenguaje de programación *Pascal*, imperativo y procedural, utiliza programación estructural y estructuración de datos. Se denomina así en honor a Blaise Pascal, filósofo y matemático del siglo XVII, creador de máquinas calculadoras.

1971

- Correo electrónico Ray Tomlinson crea un programa de correo electrónico para enviar mensajes a través de redes distribuidas. Utiliza la arroba para identificar la dirección, disponible para los usuarios de *ARPANET* en 1972.
- Microprocesador Ted Hoff en Silicon Valley descubre el microprocesador, un cambio en la microelectrónica y la informática.
- Project Gutenberg Proyecto voluntario para digitalizar libros y distribuir libros electrónicos, fundado por Michael S. Hart con la primera digitalización, la *De-*

*claración de Independencia de EE.UU.*

- RFC John Postel se encarga de los RFC.

1972

- ZOG Investigadores de *Carnegie-Mellon University* crean ZOG, un sistema de base de datos multiusuario que consiste en piezas de información que cuenta con enlaces a otras piezas.
- Quality-control of information Tesis de Kristo Ivanov que contiene las bases teóricas de los que corresponde a la idea de wiki.
- RFC
- IANA *IANA, Internet Assigned Numbers Authority*, asigna números a los distintos protocolos de Internet, administrado por Jon Postel.
- NRD Aplicación para leer correo implementada por Lawrence Roberts.
- TCP Robert Kahn comienza a trabajar en TCP en *ARPA*
- C Lenguaje de programación multiproósito diseñado por Dennis Ritchie en los laboratorios *Bell*.

1973

- Sociedad Postindustrial Concepto creado por Daniel Bell en *El advenimiento de la sociedad postindustrial* y que se concreta en la *sociedad de la información*.
- TCP Vinton Cerf y Robert Kahn definen TCP en RFC 675. En 1974 publican *A*

*protocol for Packet Network Interconnection*, un protocolo para la interconexión de redes por paquetes que especifica en detalle *TCP*.

- Unix Ritchie y Thompson reescriben *Unix* en *C*. *Bell Labs* distribuye su código fuente.

1974

- SQL Nace el lenguaje de programación multipropósito destinado a manejar bases de datos relacionales.

1975

- Emacs Richard Stallman comienza a escribir Emacs, el editor multipropósito.
- Transmedia music Stuart Saunders Smith propone la música transmedfia, música que viene de mezcla de sonidos de instrumenso musicales distintos con ritmos diversos.

1976

- DARPA Se utiliza *TCP* en *ARPANET*, ciento once equipos.

1977

- BSD Se lanza *BSD Berkeley Software Distribution*, versión del sistema operativo Unix realizado por el *Computer Systems Research Group* de la Universidad de California en Berkeley.

1978

- IP Parte del protocolo *TCP* se publica de forma separada como protocolo de Internet *IP*
- Aspen Movie Map Sistema hipermedia desarrollado en el MIT por Andrew Lippman que permitía un tour virtual por la ciudad de Aspen.

1979

- Ada El *Departamento de Defensa* de EE.UU. de Norteamérica desarrolla un lenguaje de programación que llama Ada en honor de Ada Lovelace.

1980

- Enquire Mientras trabaja como consultor del *CERN*, Tim Berners-Lee escribe con Robert Caillau un programa llamado ".Enquire-within-upon-everything" (pregunte cualquier cosa) que permite crear enlaces entre nodos arbitrarios.
- Note Cards Tecnología desarrollada por Xerox que permitía objetos escritos y enlaces. Cada pieza del hipertexto, la *card* o tarjeta, se disponía en una ventana separada de la pantalla.
- Index Thomisticus Proyecto del jesuita Roberto Busa para realizar búsquedas en los trabajos de Santo Tomás de Aquino, considerado uno de los primeros trabajos sobre hipertexto.
- IANA Se definen los siete *TLD* o *top level domains*, dominios de alto nivel.

1981

- KMS Dos miembros del equipo de *ZOG* crean *KMS*, *Knowledge Management System*, un sistema multiusuario para crear colaborativamente contenido y com-

partirlo a través de hipertexto. Podía representar presentaciones, documentos, bases de datos, programas de ordenador, correo electrónicos u otro tipo de contenido electrónico.

- ASCII en IBM Uso de *ASCII* en *PC 5150* de *IBM*

1982

- C++ Nace el lenguaje de programación multiparadigma, procedural, funcional, genérico y orientado a objetos C++, diseñado por Bjarne Stroustrup.

1983

- CRUD Aparece el término *CRUD* en el libro de James Martin *Managing the Data-base Environment*
- HyperTies Sistema creado por Ben Shneiderman de la universidad de Maryland. Se utilizó en 1988 para el que se ha considerado primer libro electrónico, con las conferencias de la ACM.

1984

- DNS Paul Mockapetris presenta el sistema de nombres de dominio *Domain Name System* o *DNS*.
- GNU Stallman lanza el proyecto *GNU*, entre cuyas aplicaciones se encuentra Emacs.
- TELOS Aplicación *Filevision*, una base de datos hipermedia de Macintosh
- POSIX *IEEE* publica *POSIX, Portable Operating System Interface Unix*, inter-

faz de sistema operativo portable, *IEEE p 1003*

1985

- Intermedia Proyecto de Norman Meyrowitz de la Universidad de Brown
- Minix Andrew Tannenbaum desarrolla Minix utilizando *FHS*
- Object Pascal Versión de *Pascal* que contempla la programación orientada a objetos.

1986

- SGML SGML se convierte en norma ISO 8879:1986

1987

- Hypercard Apple presenta *Hypercard*, la tarjeta con información hipertextual.
- Hypertext'87 Primera gran conferencia sobre hipertexto.
- Extracting Knowledge from Data En la conferencia internacional de 1987 sobre ingeniería de bases de datos, Gio Wiederhold y R. Blum hablaron sobre *Extracting Knowledge From Data*, extracción del conocimiento de datos".
- Perl Nace el lenguaje de programación multiparadigma, funcional, imperativo, orientado a objetos y procedural Perl, diseñado por Larry Wall.

1989

- Information Management: A proposal Berners-Lee busca comentarios por el *CERN*

de su documento *Information Management: A proposal* (gestión de la información: una propuesta).

- KDD-89 Primera conferencia sobre *KDD* en 1989, organizada por Gregory Piatetsky-Shapiro

1990

- ECHT *European Conference on HyperText*, primera conferencia europea en hipertexto.

1991

- CERN A los físicos del *CERN* se les ofrece un navegador, software de servidor web y librería para crear su propio software.
- Linux Linus Torvalds presenta el kernel *Linux*, basado en *Minix*, que cumple también con *FHS*.
- Nuevo periodismo de precisión Meyer actualiza su libro *Periodismo de precisión*
- Python Nace el lenguaje de programación de alto nivel, de propósito general, Python, diseñado por Guido van Rossum.
- Transmedia Marsha Kinder propone el término transmedia para referirse a la narrativa que se puede dividir en fragmentos y utilizar distintos medios, que ofrece contenido expandido por la Web y se puede producir desde nuevos dispositivos como los móviles.
- WorldWideWeb Nombre del primer navegador web del W<sub>3</sub>C desarrollado por Berners-Lee.

- Unicode Lanzamiento de *Unicode UTF-8*, desarrollado por Ken Thompson y Rob Pike en los laboratorio Bell, cuando desarrollaban el sistema operativo *Plan 9* en los 1980' con el apoyo de Dennis Ritchie.
- World Wide Web Tim Berners-Lee presenta en el *CERN* la *World Wide Web*, primera aplicación global de hipertexto. Cuenta con 22 elementos de HTML.

1993

- CERN Libera el código de la web con licencia de dominio público. A finales de año son más de 500 servidores.
- IETF Camino de estandarización de la Web. Se crean los elementos para imágenes, tablas y formularios.

1994

- EZLN Se presenta en Chiapas el *EZLN*, Ejército Zapatista de Liberación Nacional, y una ola de simpatizantes difunde su mensaje por Internet.
- Nodos50 Durante la campaña alternativa a la reunión en Madrid del *FMI*, *BM* y *GATT* se presenta *Nodos50*, un *ISP* promovido por *ONG*.
- Mozilla *Netscape* lanza la primera versión de su navegador, *Mozilla 096b*, y poco después la versión *Mozilla 1.0*. La versión de código abierto de *Netscape Navigator* lanzada en 2002 se llamó Mozilla en homenaje a esta versión.
- Opera Se crea la compañía que desarrolla uno de los navegadores más reputados aunque poco usados.
- W3C Comienza sus actividades el *W3C* en el *CERN* con el apoyo del *MIT*, *IN-*

*RIA*, *DARPA* y la Comisión Europea.

- Web 10.000 servidores web a finales de año
- WikiWikiWeb Ward Cunningham lanza el primer *wiki*, *WikiWiki*, donde se crea artículos a través de palabras que componen hiperenlaces.

1995

- Classmates Una de las primeras redes sociales de la *Web 1.0*, *classmates.com*, por Randy Conrads.
- Dublin Core Primera reunión de *Dublin Core* sobre metadatos.
- G7 En la reunión del G7 Al Gore propone una apertura del mercado de las telecomunicaciones estadounidenses si se produce reciprocidad en otros países, la liberalización del mercado de la industria de la información y del conocimiento.
- IETF Se define el *HTML 2.0* en *IETF* con *DTD* para definir la gramática.
- Ruby Nace el lenguaje de programación dinámico, multipropósito y orientado a objetos Ruby, diseñado por Yukihiro Matsumoto.
- IPv6 La versión 6 de *IP* se convierte en *RFC 1883*, reemplazado por *RFC 2460* en 1998
- Windows95 *Microsoft* lanza el sistema operativo *Windows 95* que incluía el navegador *Internet Explorer*, por lo que superó a *Netscape Navigator* en 1999 como navegador más utilizado.
- Java Nace Java, lenguaje de programación de propósito general concurrente, ba-

sado en clases y orientado a objetos. Aplicó el paradigma "wrote once run anywhere", ya que no había que compilar el código mientras la máquina virtual de Java (*JVM*, *Java Virtual Machine*) esté instalado.

1996

- 6 grados de separación Juego de Brett Tjaden con datos de *IMDB* representa las relaciones entre actores y actrices.
- Opera Se lanza la primera versión del navegador.
- W<sub>3</sub>C W<sub>3</sub>C mueve sus oficinas al *MIT* y su dominio a [www.w3c.org](http://www.w3c.org)

1997

- HTML 3.2 W<sub>3</sub>C lanza *HTML 3.2* y recomendación *HTML 4.0*: soporte para *CSS*, *scripts*, tablas complejas o formularios.
- URN Se utiliza *URN Unified Resource Name* como complemento de *URL* hasta 2005, que se utiliza *URI* para *URL* o *URN*

1998

- DOM Recomendación W<sub>3</sub>C en *DOM Level 1*, en 2000 la de Nivel 2 y en 2003 Nivel 3.
- HTML W<sub>3</sub>C actualiza recomendación *HTML 4.0*. Berners-Lee lanza los principios del buen diseño de la web.
- ICANN Se crea *ICANN, Internet Corporation for Assigned Names and Numbers*, realiza la conversión entre números /IP/(v4) y nombres de dominio.

- UIT La *Unión Internacional de Telecomunicaciones*, agencia de las *Naciones Unidas*, decide realizar una cumbre mundial sobre la sociedad de la información que se realiza en dos fases: 2003 en Suiza y 2005 en Túnez.
- XML Recomendación *W<sub>3</sub>C*

1999

- Correo en UTF-8 Se recomienda el uso de UTF-8 en el correo electrónico.
- HTML 4.01 *W<sub>3</sub>C* lanza recomendación *HTML 4.01*
- RDF Recomendación *RDF* del *W<sub>3</sub>C*.
- WCAG *W<sub>3</sub>C* crea la *Web Accessibility Initiative*, iniciativa por la accesibilidad en la Web y en 1999 publican la primera recomendación oficial de *WCAG/*

2000

- REST Roy Fielding define *REST*.
- XHTML *W<sub>3</sub>C* lanza *XHTML 1.0*

2001

- 6 grados de separación Duncan Watts realizó experimento por correo electrónico y comprobó el número de pasos.
- Wikipedia Nace *Wikipedia*, primera enciclopedia de hipertexto colaborativa.
- XML Schema Recomendación *W<sub>3</sub>C*, segunda revisión de 2004.

2002

- Mozilla Se lanza *Mozilla*, la versión de código abierto de *Netscape Navigator*.
- XHTML2 W3C comienza el desarrollo de *XHTML2*

2003

- Cumbre Mundial Sociedad de la Información
- W3C Spain Se inaugura la oficina del W3C en España.

2004

- Firefox *Mozilla* lanza *Firefox*, el navegador que destronó a *Internet Explorer* y luego se vio superado por *Google Chrome*.
- WHATWG *Apple*, *Mozilla* y *Opera* crean *WHATWG*, preocupados por el desarrollo de *XHTML2* en *W3C*

2005

- URI Comienza a extenderse el uso de *URI* para referirse a *URN* y *URL*

2006

- Periodismo de Datos Adrian Holovaty, fundador de *EveryBlock*, realiza una de las primeras definiciones de periodismo de datos.

2007

- HTML Vuelve el *HTML* a *W<sub>3</sub>C*

2008

- HTML<sub>5</sub> Primer borrador oficial de *HTML<sub>5</sub>*
- ACTA *Anti Counterfeiting Trade Agreement*, acuerdo comercial contra la falsificación, en el sSeminario internacional sobre *el futuro de las industrias culturales y la lucha contra la piratería in Internet*

2009

- XHTML<sub>2</sub> *W<sub>3</sub>C* abandona *XHTML<sub>2</sub>* en pro de *HTML<sub>5</sub>*.<sup>o</sup>
- The Web Foundation Creada por Tim Berners-Lee, fundador de la Web, para trabajar en el incremento del acceso a la Web, la mejora de la participación, los datos abiertos y la consideración de la Web como derecho fundamental.
- 2014
  - Cube El *W<sub>3</sub>C* publica la recomendación de *Vocabulary Data Cube*, que permite visiones multidimensionales de datos *RDF*

---

## Bibliografía

- [1] 2011, S. O. (2011). 1st international workshop on social object networks (socialobjects 2011). <http://ir.ii.uam.es/socialobjects2011/>, acceso en 2015-09-04.
- [2] AA., V. (2013). Crisis del periodismo. <http://s.coop/1wv81>, acceso en 2015-09-04.
- [3] Abril, G. (1997). *Teoría general de la información*. Cátedra.
- [4] Adida, B. . M. B. . S. M. . S. P. (2008). RDFa in XHTML: Syntax and Processing. Technical report, W3C. <http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014>, acceso en 2015-08-24.
- [5] aiana (2015). Root zone database. web. <https://www.iana.org/domains/root/db>, acceso en 2015-08-18.
- [6] Allen, J. . U. C. (2007). *The Unicode Standard 5.0*. Addison-Wesley.
- [7] Álvarez Álvarez, C. (2008). La etnografía como modelo de investigación en educación. [http://www.ugr.es/~pwlac/G24\\_10Carmen\\_Alvarez\\_Alvarez.html](http://www.ugr.es/~pwlac/G24_10Carmen_Alvarez_Alvarez.html), acceso en 2015-09-22.
- [8] Antón Bravo, A. (2013). El periodismo de datos y la web semántica. *CIC, Cuadernos de Información y Comunicación*, 18:99–116. <http://revistas.ucm.es/index.php/CIYC/article/view/41718>, acceso en 2015-09-04.

- [9] Arthur, C. (2010). Analysing data is the future for journalists, says tim berners-lee. <http://cort.as/XNP5>, accedida en 2015-08-23.
- [10] Backus, J. W. (1959). The syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference. In *IFIP Congress*, pages 125–131.
- [11] Bagley, P. R. (1968). Extension of programming language concepts. Technical report, University City Science Center, Philadelphia, PA. <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=AD0680815>, accedido 2015-08-26.
- [12] Baran, P. (1964). On distributed communications: Viii. the multiplexing station. Technical report, United States Air Force Project RAND. [https://www.rand.org/content/dam/rand/pubs/research\\_memoranda/2006/RM3764.pdf](https://www.rand.org/content/dam/rand/pubs/research_memoranda/2006/RM3764.pdf), accedido 20150824.
- [13] Baron, D. (2014). W<sub>3</sub>C proposed recommendation: Html5. [https://groups.google.com/forum/#!msg/mozilla.dev.platform/BnY1261cNJo/MdkaT\\_EX6MOJ](https://groups.google.com/forum/#!msg/mozilla.dev.platform/BnY1261cNJo/MdkaT_EX6MOJ), acceso en 2015-08-15.
- [14] Bauwens, M. (2005). The political economy of peer production. <http://www.ctheory.net/articles.aspx?id=499>, acceso en 2015-08-25.
- [15] Bauwens, M. (2014). La economía política de la producción entre iguales. *Hipertextos, Capitalismo, Técnica y Sociedad en debate*, 1(2). <http://s.coop/1wttm>, accedido 2015-08-25.
- [16] Beckett, D. . J. B. (2008). SPARQL Query Results XML Format. Recommendation, W<sub>3</sub>C. <http://www.w3.org/TR/2008/REC-rdf-sparql-XMLres-20080115/>, acceso en 2015-08-21.
- [17] Bell, D. (2001). *El advenimiento de la sociedad post-industrial: un intento de prognosis social*. Alianza Universidad. Alianza.
- [18] Bellver, J. (diciembre 2013-febrero 2014). Lo pequeño no es tan hermoso. *Boletín ECOS*, pages 5–15. <https://www.fuhem.es/media/cdv/file/biblioteca/>

- Dossier/Dossier\_La-huella-del-consumismo-tecnologico.pdf, acceso en 2015-08-26.
- [19] Benjamins, R. . J. C. . O. C. . A. G.-P. (2002). The six challenges of the semantic web.
- [20] Berners-Lee, T. (1991a). HTML Tags. <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>, acceso en 2015-07-20.
- [21] Berners-Lee, T. (1991b). Hypertext implementation. <http://www.w3.org/History/1991-WWW-NeXT/Implementation/HyperText.m>, acceso en 2015-08-16.
- [22] Berners-Lee, T. (1991c). What is hypertext. <http://info.cern.ch/hypertext/WWW/WhatIs.html>, acceso en 2015-08-18.
- [23] Berners-Lee, T. (1991d). World wide web. <http://info.cern.ch/hypertext/WWW/TheProject.html>, acceso en 2015-08-18.
- [24] Berners-Lee, T. (1992a). Academic information by organisation. <http://www.w3.org/History/19921103-hypertext/hypertext/DataSources/byOrganisation/Overview.html>, acceso en 2015-08-18.
- [25] Berners-Lee, T. (1992b). Data sources classified by type of service. <http://www.w3.org/History/19921103-hypertext/hypertext/DataSources/ByAccess.html>, acceso en 2015-08-18.
- [26] Berners-Lee, T. (1992c). How big to make each document. <http://www.w3.org/Provider/Style/DocSize.html>, acceso en 2015-08-18.
- [27] Berners-Lee, T. (1992d). Hypertext Terms. <http://info.cern.ch/hypertext/WWW/Terms.html>, acceso en 2015-08-18.

- [28] Berners-Lee, T. (1992e). Information by subject. <http://www.w3.org/History/19921103-hypertext/hypertext/DataSources/bySubject/Overview.html>, acceso en 2015-08-18.
- [29] Berners-Lee, T. (1992f). Overlapping Trees. <http://www.w3.org/Provider/Style/Trees.html>, acceso en 2015-08-18.
- [30] Berners-Lee, T. (1992g). The reader's structure. <http://www.w3.org/Provider/Style/ReaderStructure.html>, acceso en 2015-08-18.
- [31] Berners-Lee, T. (1992h). Refer or copy. <http://www.w3.org/Provider/Style/ReferOrCopy.html>, acceso en 2015-08-18.
- [32] Berners-Lee, T. (1992i). W3 servers. <http://info.cern.ch/hypertext/DataSources/WWW/Servers.html>, acceso en 2015-08-18.
- [33] Berners-Lee, T. (1998). Semantic Web Road map. Technical report, W3C. <http://www.w3.org/DesignIssues/Semantic.html>, consultada en 2015-09-05.
- [34] Berners-Lee, T. (2000a). *Tejiendo la red*. Siglo XXI de España, Madrid, siglo xxi de españa general edition.
- [35] Berners-Lee, T. (2006a). Linked data. <http://www.w3.org/DesignIssues/LinkedData.html>, acceso en 2015-09-06.
- [36] Berners-Lee, T. (2007). Giant Global Graph. web. <http://dig.csail.mit.edu/breadcrumbs/node/215>, accedida en 2015-07-24.
- [37] Berners-Lee, T. (2010). The year open data went worldwide. web. [http://www.ted.com/talks/tim\\_berniers\\_lee\\_the\\_year\\_open\\_data\\_went\\_worldwide](http://www.ted.com/talks/tim_berniers_lee_the_year_open_data_went_worldwide), acceso en 2015-09-23.
- [38] Berners-Lee, T. (2013a). Barack obama tim berners-lee open data. <http://www.theguardian.com/news/datablog/2013/oct/25/barack-obama-tim-berniers-lee-open-data>, acceso en 2015-08-23.

- [39] Berners-Lee, T. (2013b). Web founder berners-lee slams hypocrisy of western governments on internet spying. <http://s.coop/1ww3g>, accedido 2015-08-24.
- [40] Berners-Lee, T. . D. C. (1995a). Hypertext Markup Language - 2.0. RFC 1866, Internet Engineering Task Force. <http://www.rfc-editor.org/rfc/rfc1866.txt>, acceso en 2015-09-04.
- [41] Berners-Lee, T. . D. C. (1995b). Hypertext Markup Language - 2.0. <http://www.ietf.org/rfc/rfc1866.txt>, acceso en en 2015-09-06.
- [42] Berners-Lee, T. . J. H. . L. O. (2001). The semantic web. a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*. <http://www.scientificamerican.com/article.cfm?id=the-semantic-web>, acceso en 2015-08-24.
- [43] Berners-Lee, T. . R. C. (2000b). WorldWideWeb: Proposal for a HyperText Project. Technical report, CERN. <http://www.w3.org/Proposal.html>, acceso en 2015-09-02.
- [44] Berners-Lee, T. . R. F. . H. F. (1996). Hypertext Transfer Protocol – HTTP/1.0. <http://www.ietf.org/rfc/rfc1945.txt>, acceso en en 2015-09-06.
- [45] Berners-Lee, T. . R. F. . L. M. (2005). Uniform resource identifier (uri): Generic syntax. <http://www.ietf.org/rfc/rfc3986.txt>, acceso en en 2015-09-06.
- [46] Berners-Lee, T. . W. H. . J. A. H. . K. O. . N. S. . D. J. W. (2006b). *A Framework for Web Science*, volume 1. now, the essence of knowledge. <http://www.nowpublishers.com/article/Download/WEB-001>, acceso en 2015-09-09.
- [47] Berners-Lee, Tim (1992). A quick look at Erwise. <http://s.coop/1wtrv>, acceso en 2015-09-06.
- [48] Blanco Suárez, S. (2005). *Anotaciones semánticas en WebQuest*. PhD thesis, Universidad de Valladolid. <http://www.infor.uva.es/~sblanco/tesis/>, acceso en 2015-09-04.

- [49] Blog, G. O. (2008). A fresh take on the browser. <http://s.coop/1wtsr>, acceso en 2015-08-15.
- [50] Blog, G. O. (2010). Unicode nearing 50 % of the web. <http://s.coop/1wtsx>, acceso en 2015-08-15.
- [51] Blogger (2001). Permanent Links. <http://s.coop/1wtrs>, acceso en 2015-09-11.
- [52] Bonilla, S. (2006). Web semántica, marcadores discursivos y metarrepresentación. *RAEL, Revista electrónica de lingüística aplicada*.
- [53] Borst, W. N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. University of Twente. <http://purl.utwente.nl/publications/17864>, acceso en 2015-09-02.
- [54] Bos, B., Lie, H. W., Lilley, C., and Jacobs, I. (1998). Cascading Style Sheets, level 2 (CSS2) Specification. Recommendation, W3C. <http://www.w3.org/TR/1998/REC-CSS2-19980512>, acceso en 2015-09-22.
- [55] Bos, B. . T. Ç. . I. H. . H. W. L. (2011). Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. Technical report, W3C. <http://www.w3.org/TR/2011/REC-CSS2-20110607>, acceso en 2015-08-24.
- [56] Bradshaw, P. (2010). How to be a data journalist. <http://cort.as/XNV0>, accedida en 2015-08-22.
- [57] Brandel, M. (1999). 1963: The debut of ascii. <http://edition.cnn.com/TECH/computing/9907/06/1963.idg/>, acceso en 2015-08-15.
- [58] Brickley, D. (2006). Basic Geo WGS84 lat/long Vocabulary. <http://www.w3.org/2003/01/geo/>, acceso en 2015-09-05.
- [59] Brickley, D. . L. M. (2014). FOAF Vocabulary Specification 0.99. Technical report, XMLNS. <http://xmlns.com/foaf/spec/>, accedido 2015-08-24.

- [60] Broder, A. . R. K. . F. M. . P. R. . S. R. . R. S. . A. T. . J. W. (2000). Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference on Computer Networks : The International Journal of Computer and Telecommunications Netowrking*, pages 309–320, Amsterdam, The Netherlands. North-Holland Publishing Co. <http://dl.acm.org/citation.cfm?id=347319.346290>, acceso en 2015-09-04.
- [61] Brunet Icart, I. . A. M. B. (2001). Epistemología y cibernética. <http://papers.uab.cat/article/view/v65-brunet-morell/pdf-es>, acceso en 2015-09-22.
- [62] Burstein (2006). Accessible design tools. <http://s.coop/1wv87>, acceso en 2015-09-09.
- [63] Bush, V. (1945). As we may think. *The Atlantic Monthly*. <http://web.mit.edu/STS.035/www/PDFs/think.pdf>, accedida en 2015-08-23.
- [64] Cabra, M. (2015). Swissleaks apm 20150605. <http://s.coop/1wv8d>, acceso en 2015-08-22.
- [65] Caillau, R. (1991). Short history of hypertext. <http://info.cern.ch/hypertext/History.html>, acceso en 2015-08-18.
- [66] Caldwell, B. . M. C. . L. G. R. . G. V. (2008). Web Content Accessibility Guidelines (WCAG) 2.0. Technical report, W3C. <http://www.w3.org/TR/2008/REC-WCAG20-20081211/>, acceso en 2015-08-24.
- [67] Carpenter, B. (1996). Architectural Principles of the Internet. =<http://www.ietf.org/rfc/rfc1958.txt>, accedida en 2015-08-28.
- [68] Carr, L. . S. B. . C. G. . W. H. (2001). Conceptual Linking: Ontology-based Open Hypermedia. <http://s.coop/1wv86>, acceso en 2015-09-06.

- [69] Carvin, A. (2007). The Wonders of Hypertext: Non-Linear Informational Adventures. <http://www.edwebproject.org/web.hypertext.html>, accedida en 2015-09-08.
- [70] Castells, M. and Gimeno, C. (2004). *La era de la información: economía, sociedad y cultura*. Number v. 1 in *La era de la información: economía, sociedad y cultura*. Siglo Veintiuno.
- [71] celik, T. (2015). h-card. <http://microformats.org/wiki/h-card>, acceso en 2015-09-08.
- [72] CCM (2015a). Historia de internet. <http://s.coop/1wv83>, acceso en 2015-08-15.
- [73] CCM (2015b). Unix: archivos. <http://s.coop/1wv84>, acceso en 2015-08-15.
- [74] Cerf, V. . Y. D. . C. S. (1974). Specification of Internet Transmission Control Program. <http://www.ietf.org/rfc/rfc675.txt>, acceso en en 2015-07-24.
- [75] Chua, R. (2012). A question of trust. <http://s.coop/1wv82>, acceso en 2015-09-04.
- [76] Cisneros Torres, M. J. (2009-2010). Walter benjamin: cultura de masas y esteticismo político. *Especulo. Revista de estudios literarios*, 1(43). <https://pendientedemigracion.ucm.es/info/especulo/numero43/wbenjam.html>, accedida en 2015-08-24.
- [77] Cobo Romani, C. . H. K. (2007). *Planeta Web 2.0. Inteligencia colectiva o medios*. Flacso México. [http://online.upaep.mx/campusTest/ebooks/planetaWeb2\\_0.pdf](http://online.upaep.mx/campusTest/ebooks/planetaWeb2_0.pdf), acceso en 2015-08-15.
- [78] Commission, E. (1995). Conclusions of g7 summit information society conference". [http://web.archive.org/web/20040710155058/http://europa.eu.int/ISPO/docs/services/docs/1997/doc\\_95\\_2\\_en.pdf](http://web.archive.org/web/20040710155058/http://europa.eu.int/ISPO/docs/services/docs/1997/doc_95_2_en.pdf), consultado en 2015-09-04.

- [79] Conklin, J. (1987). Hypertext: An Introduction and Survey. <http://cort.as/X01r>, accedida en 2015-09-08.
- [80] Corcho, Ó. (2014). Aragopedia Ontology. Technical report, Aragopedia Project. <http://opendata.aragon.es/def/Aragopedia>, acceso en 2015-09-21.
- [81] Cotton, M. . L. E. . J. T. . M. W. . S. C. (2011). Internet assigned numbers authority (iana) procedures for the management of the service name and transport protocol port number registry. <http://www.ietf.org/rfc/rfc6335.txt>, acceso en en 2015-09-06.
- [82] Cruz, É. G. (2007). *Las metáforas de Internet*. Tic . Cero. Editorial UOC, S.L.
- [83] Cyganiak, R. . D. R. (2014). The RDF Data Cube Vocabulary. Recommendation, W<sub>3</sub>C. <http://www.w3.org/TR/2014/REC-vocab-data-cube-20140116/>, acceso en 2015-09-21.
- [84] Data-vocabulary.org (2011). Data-vocabulary.org. <http://www.data-vocabulary.org/>, acceso en 2015-09-08.
- [85] Daul, W. (1991). Seminar by doug engelbart. <http://info.cern.ch/hypertext/Products/Engelbart/Seminar.html>, acceso en 2015-08-18.
- [86] Davies, J. . R. S. . P. W. (2006). *Semantic Web technologies: trends and research in ontology-based systems*. John Wiley & Sons.
- [87] Davis, M. (2008). Moving to unicode 5.1. <http://googleblog.blogspot.com.es/2008/05/moving-to-unicode-51.html>, acceso en 2015-08-15.
- [88] Debord, G. (1995). *La sociedad del espectáculo*. Biblioteca de la mirada. La Marca.
- [89] Deering, S. . R. H. (1995). Internet Protocol, Version 6 (IPv6) Specification. <http://www.ietf.org/rfc/rfc1883.txt>, acceso en en 2015-09-06.
- [90] Deering, S. . R. H. (1998). Internet Protocol, Version 6 (IPv6) Specification. <http://www.ietf.org/rfc/rfc2460.txt>, acceso en en 2015-09-06.

- [91] Del Olmo, C. (2010). La ciencia de los oprimidos. <http://www.revistaminerva.com/articulo.php?id=417>, acceso en 2015-08-25.
- [92] DeLong, B. (2009). John gordon: Why google loves chrome: Netscape constellation. [http://delong.typepad.com/egregious\\_moderation/2009/01/john-gordon-why-google-loves-chrome-netscape-constellation.html](http://delong.typepad.com/egregious_moderation/2009/01/john-gordon-why-google-loves-chrome-netscape-constellation.html), acceso en 2015-08-25.
- [93] Duerst, M. . M. S. (2005). Internationalized Resource Identifiers (IRIs). RFC 3987 (Proposed Standard). <http://www.ietf.org/rfc/rfc3987.txt>, acceso en 2015-09-23.
- [94] Earhart, S., Telephone, A., and Company, T. (1986). *UNIX Programmer's Manual: Languages and support tools*. UNIX Programmer's Manual. Holt, Rinehart, and Winston.
- [95] Echeverria Ezponda, J. (1999). *Los señores del aire: Télépolis y el Tercer Entorno*. Destino.
- [96] Eco, U. (2014). *Cómo se hace una tesis*. GEDISA.
- [97] Eguiluz, J. (2015). Introducción a CSS. <https://librosweb.es/libro/css/>, acceso en 2015-07-24.
- [98] Elam, G. . T. S. . N. H. (1999). Warriors of the net. <http://www.warriorsofthe.net>, acceso en 2015-09-02.
- [99] esDBpedia (2014). esdbpedia. <http://mayor2.dia.fi.upm.es/oeg-upm/index.php/es/services/322-spanish-chapter-dbpedia>, acceso en 2015-09-23.
- [100] Espada, A. (2009). La noticia posmoderna. <http://www.puroperiodismo.cl/?p=3176>, accedida en 2015-07-23.
- [101] española de W3C, O. (2014). Guía breve de accesibilidad web. <http://cort.as/XJa5>, acceso en 2015-08-24.

- [102] Fallside, D. (2001). XML Schema Part 0: Primer. Technical report, W3C. <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>, acceso en 2015-09-06.
- [103] Fallside, D. . P. W. (2004). XML Schema Part 0: Primer Second Edition. Recommendation, W3C. <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>, acceso en 2015-09-24.
- [104] Fayyad, U. . G. P.-S. . P. S. (1997). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 1(Fall):37–51.
- [105] Feigenbaum, L. . G. W. . K. C. . E. T. (2013). SPARQL 1.1 Protocol. Recommendation, W3C. <http://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/>. Latest version available at <http://www.w3.org/TR/sparql11-protocol/>.
- [106] Fielding, R. . J. G. . J. M. . H. F. . L. M. . P. L. . T. B.-L. (1999). Rfc 2616, hypertext transfer protocol – http/1.1. <http://www.rfc.net/rfc2616.html>, accedida en 2015-09-12.
- [107] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine. <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>, acceso en 2015-08-16.
- [108] Fielding, R. T. (2008). Rest apis must be hypertext-driven. web. <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>, acceso en 2015-08-16.
- [109] Fitzpatrick, B. . D. R. (2006). Openid authentication 1.1. Technical report, OpenID. [http://openid.net/specs/openid-authentication-1\\_1.txt](http://openid.net/specs/openid-authentication-1_1.txt), acceso en 2015-08-24.
- [110] Fitzpatrick, B. . D. R. (2007). Thoughts on the social graph. <http://bradfitz.com/social-graph-problem/>, acceso en 2015-08-24.

- [111] Fletcher, J. (2014). Wiki History. <http://c2.com/cgi/wiki?WikiHistory>, accedida en 2015-09-06.
- [112] Freed, Ned Dürst, M. (2013). Character sets. Technical report, IANA. <http://s.coop/1wtt1>, acceso en 2015-08-15.
- [113] García Gutiérrez, A. (2002). *La Memoria Subrogada: Medicación, Cultura y Conciencia en la Red Digital*. Biblioteca de Bolsillo (Universidad de Granada): Limitanea. Universidad de Granada, Editorial.
- [114] García Gutiérrez, A. L. (2004). *Otra memoria es posible. Estrategias descolonizadoras del archivo mundial*. Buenos Aires: La Crujía.
- [115] García-Luengos, J. (2014). Extracción de los recursos naturales en África: la industria extractiva. <http://www.africacuestiondevida.org/wp-content/uploads/2014/01/ExplotacionRecursosNaturales.pdf>, acceso en 2015-09-23.
- [116] García Santiago, L. (2003). *Extraer y visualizar información en Internet: el Web Mining*. Ediciones Trea, Gijón.
- [117] Gitlin, T. (2013). Un cúmulo de crisis: circulación, ingresos, atención, autoridad y confianza. *CIC, Cuadernos de Información y Comunicación*, 18:11–26. <http://s.coop/1wv81>, acceso en 2015-09-04.
- [118] Goble, C. . S. B. . D. L. C. . D. d. R. . W. H. (2001). Conceptual Open Hypermedia, The Semantic Web. <http://ceur-ws.org/Vol-40/Goble-et-al.pdf>, acceso en 2015-09-06.
- [119] Goldfarb, C. (1996a). Sgml: in memory of william w. tunnicliffe. <http://xml.coverpages.org/tunnicliffe.html>, acceso en 2015-09-03.
- [120] Goldfarb, C. F. (1996b). The roots of sgml – a personal recollection. <http://www.sgmlsource.com/history/roots.htm>, acceso en 2015-09-03.

- [121] Gordon, R. (2013). Want to build a data journalism? you'll need these three people. <http://cort.as/XNNe>, accedida en 2015-06-28.
- [122] Granqvist, H. (2006). Openid authentication security profiles. Technical report, Verisign, Inc. <http://s.coop/1wtsz>, acceso en 2015-08-24.
- [123] Group, S. U. (1990). <http://www.sgmlsource.com/history/sgmlhist.htm>, acceso en 2015-09-03.
- [124] Gruber, T. (2006). Social web meets semantic web. <http://tomgruber.org/writing/social-web-meets-semantic-web.pdf>, acceso en 2015-07-15.
- [125] Gruber, T. (2007). Collective knowledge systems: Where the social web meets the semantic web. <http://tomgruber.org/writing/collective-knowledge-systems.htm>, acceso en 2015-08-29.
- [126] Gruber, T. (2008). Why ontologies are not the hard part of semantic web. <http://tomgruber.org/writing/aaai-ss08.pdf>, consultada en 2015-07-24,.
- [127] Gutiérrez y Restrepo, E. . C. B. . S. B. (2009). Web content accessibility guidelines wcag 2.0. Technical report, Fundación Sidar - Acceso Universal. <http://www.sidar.org/traducciones/wcag20/es/>, acceso en 2015-08-30.
- [128] Haraway, D. J. (1995). *Ciencia, cyborgs y mujeres: la reinención de la naturaleza*. Cátedra.
- [129] Haraway, D. J. (1997). *Modest-witness second millenium: feminism and technoscience; FemaleMan-Meets-Oncomouse*. Routledge, Nueva York.
- [130] Harawayó, D. (2013). *Simians, Cyborgs, and Women: The Reinvention of Nature*. Taylor Francis.

- [131] Hernández Castillo, A. (2005). Ciencia, cyborgs y mujeres: la reinención de la naturaleza de donna j. haraway. *Ciencias, Universidad Nacional Autónoma de México. Distrito Federal, México*, 1(77):76–77. <http://www.redalyc.org/pdf/644/64407713.pdf>, acceso en 2015-07-25.
- [132] Hernández Orallo, J. . M. J. R. Q. . C. F. R. (2005). *Introducción a la minería de datos*. Pearson Educación, Madrid.
- [133] Hickson, I. . R. B. . S. F. . T. L. . E. D. N. . E. O. . S. P. (2008). RDFa in XHTML: Syntax and Processing. Technical report, W3C. <http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014>, accedida en 2015-08-24.
- [134] Hispano, A. S. (2015). Archivo situacionista hispano. [www.sindominio.net/ash](http://www.sindominio.net/ash), acceso en 2015-09-30.
- [135] Holovaty, A. (2006). A fundamental way newspaper sites need to change. <http://www.holovaty.com/writing/fundamental-change/>, acceso en 2015-09-04.
- [136] Holovaty, A. (2009). The definitive, two-part answer to "is data journalism?". <http://www.holovaty.com/writing/data-is-journalism/>, acceso en 2015-08-22.
- ñez, 1990]ibanez1990sociologia Ib{áñez, J. (1990). *Sociología Crítica de la Cotidianidad Urbana*. Anthropos Editorial.
- [137] ICANN (2000). New tld program application process archive. <http://s.coop/1wtt2>, acceso en 2015-08-18.
- [138] ICANN (2014). Overview and history of the iana functions. Technical report, ICANN. <https://www.icann.org/en/system/files/files/sac-067-en.pdf>, acceso en 2015-08-18.
- [139] IMC (1998). Using International Characters in Internet Mail. <http://www.imc.org/mail-i18n.html>, acceso en 2015-09-02.

- [140] Internetlivestats (2015). Total number of Websites. <http://cort.as/XOQe>, acceso en 2015-09-12.
- [141] Isaacson, S. . S. B. B. . M. C. . I. J. . A. L. H. . G. N. . J. R. . R. S. S. . C. W. . L. W. (1998). Document Object Model (DOM) Level 1. Technical report, W3C. <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>, acceso en 2015-09-06.
- [142] ISO, I. (1996). ISO/IEC 7498-1 Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. Technical report, ISO. <http://s.coop/1wtt0>, acceso en 2015-08-17.
- [143] Jacob, I. . N. W. (2004). Architecture of the World Wide Web, Volume One. Technical report, W3C. <http://www.w3.org/TR/webarch/>, accedida en 2015-09-04.
- [144] Jacobs, I. . N. W. (2004). Architecture of the World Wide Web, Volume One. Technical report, W3C. <http://www.w3.org/TR/2004/REC-webarch-20041215/>, acceso en 2015-09-09.
- [145] Jacobson, D. (2009). COPE: Create Once, Publish Everywhere. <http://www.programmableweb.com/news/cope-create-once-publish-everywhere/2009/10/13>, acceso en 2015-09-30.
- [146] Johnston, C. (2015). Time's up for ipv4 as north america runs out of addresses. <http://s.coop/1wtt4>, acceso en 2015-08-16.
- [147] Karinthy, F. (1929). Chain-links. <http://s.coop/1wtrg>, acceso en 2015-09-09.
- [148] Kempf, R. (2012). Argo Project Milestone: Launch of Open Source Site. <http://digitalservices.npr.org/post/argo-project-milestone-launch-open-source-site>, acceso en 2015-08-22.

- [149] Lamarca Lapuente, M. J. (2006). *Hipertexto: el nuevo concepto de documento en la cultura de la imagen*. PhD thesis, Universidad Complutense de Madrid. Departamento de Biblioteconomía y Documentación. <http://www.hipertexto.info>, accedida 2015-08-20.
- [150] Lassila, O. (1999). Resource Description Framework (RDF) Model and Syntax Specification. Technical report, W3C. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>, acceso en 2015-08-24.
- [151] López Amat, J. (2010). De la Sociedad de la información a la(s) Sociedad(es) del Conocimiento. Master's thesis, Dpto. de Historia de la Comunicación Social. Universidad Complutense de Madrid UCM, Madrid, Spain. <http://cort.as/XKw8>, accedida en 2015-08-23.
- [152] Louvel, J. (2013). How much rest should your web api get? <http://s.coop/1wtt5>, acceso en 2015-08-18.
- [153] Maestre, J. A. . C. S. A. (2001). Lssi: análisis legal. <http://www.nodo50.org/kriptopolis-lssi/lssi/02.html>, accedida en 2015-08-25.
- [154] Mair, J. . R. L. K. (2014). *Data journalism. Mapping the future*. Arima Publishing.
- [155] Manovich, L. (2005). *El lenguaje de los nuevos medios de comunicación: la imagen en la era digital*. Paidós comunicacion/ Communication. Paidós.
- [156] McCarthy, J. (2007). What is artificial intelligence? <http://www-formal.stanford.edu/jmc/whatisai/node1.html>, acceso en 2015-09-02.
- [157] McGregor, S. (2012). On data literacy. <https://www.youtube.com/watch?v=qGNv45xYnno>, acceso en 2015-09-11.
- [158] Menabrea, L. F. (1841). Sketch of the Analytical Engine Invented by Charles Babbage. <https://www.fourmilab.ch/babbage/sketch.html>, accedida en 2015-08-24.

- [159] Méndez Rodríguez, E. (2002). *Metadatos y recuperación de información: estándares, problemas y aplicabilidad en bibliotecas digitales*. Trea, Gijón, Spain.
- [160] Meyer, P. (2002). *Precision Journalism: A Reporter's Introduction to Social Science Methods*. G - Reference, Information and Interdisciplinary Subjects Series. Rowman Littlefield Publishers.
- [161] Miles, A. . S. B. (2009). SKOS Simple Knowledge Organization System Reference. Technical report, W3C. <http://www.w3.org/TR/2009/REC-skos-reference-20090818/>, acceso en 2015-08-24.
- [162] Mills, C. D. (2011). Introducing BrowserID: A better way to sign in. <http://s.coop/1wtt7>, acceso en 2015-08-24.
- [163] Moats, R. (1997). URN Syntax. <http://www.ietf.org/rfc/rfc2141.txt>, acceso en en 2015-09-06.
- [164] Mockapetris, P. (1987). Domain names - implementation and specification. <http://www.ietf.org/rfc/rfc1035.txt>, acceso en en 2015-09-06.
- [165] Morin, E. (1998). *Introducción al pensamiento complejo*. Ciencias cognitivas. Gedisa.
- [166] Mozilla (2013). Disclosure: the identity team at mozilla work for me. <https://news.ycombinator.com/item?id=7362613>, acceso en 2015-08-24.
- [167] Mozilla, D. (2015). hgroup. <http://s.coop/1wtt8>, acceso en 2015-08-15.
- [168] Murtagh, R. (2014). Mobile Now Exceeds PC: The Biggest Shift Since the Internet Began. <http://cort.as/XOPV>, acceso en 2015-09-10.
- [169] Murua, I. (2009). Herramientas de anotación para el desarrollo de la web semántica. <http://s.coop/1wtt6>, acceso en 2015-09-06.
- [170] Narayanan, A. . V. S. (2009). De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE.

- [171] Nauer, P. (1963). Report on the algorithmic language algol 60. [https://web.eecs.umich.edu/~bchandra/courses/papers/Naure\\_Algol60.pdf](https://web.eecs.umich.edu/~bchandra/courses/papers/Naure_Algol60.pdf), acceso en 2015-08-24.
- [172] Navarrete, A. (1993). *La crítica de la modernidad como crítica a la autonomía del arte*. Colección Tesis doctorales. Universidad de Castilla-La Mancha, Servicio de Publicaciones.
- [173] Nelson, T. (2010). Tim berners-lee re ted nelson. <http://hyperland.com/TBLpage>, acceso en 2015-08-16.
- [174] Nelson, T. (2014). Xanadoc. <http://xanadu.com/xanademos/MoeJusteOrigins.html>, acceso en 2015-08-18.
- [175] Nelson, T. H. (1987). *Literary Machines*. Theodor H. Nelson. <https://archive.org/details/Literary.Machines.1987>, acceso en 2015-08-29.
- [176] Nemiche, M. (2002). *Un modelo sistémico de evolución social dual*. PhD thesis, Universitat de Valencia. <http://www.uv.es/nemiche/capitulo1>.
- [177] NRO, T. N. R. O. (2011). Free Pool of IPv4 Address Space Depleted. <https://www.nro.net/news/ipv4-free-pool-depleted>, acceso en 2015-08-16.
- [178] Olson, G. A. (2010). Writing, literacy and technology: Toward a cyborg writing. *JAC, a journal of rhetoric, culture politics*, 16.1:1-26. <http://www.jaconlinejournal.com/archives/vol16.1/olson-writing.pdf>, acceso en 2015-09-16.
- [179] Open Geospatial, C. (2015). Arml 2.0 swg. <http://www.opengeospatial.org/projects/groups/arml2.0swg>, acceso en 2015-09-04.
- [180] OpenSocialWeb.org (2007). A Bill of Rights for Users of the Social Web. <http://s.coop/1wtt9>, acceso en 2015-09-08.
- [181] O'Reilly, T. (2005). What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. <http://cort.as/XJUE>, acceso en 2015-06-09.

- [182] Otero Puime, Ángel, M. V. Z. P. . F. B. . Á. R. L. . M. J. G. d. Y. y. P. (2006). *Relaciones sociales y envejecimiento saludable*. Fundación BBVA. [http://www.fbbva.es/TLFU/dat/DT\\_2006\\_09.pdf](http://www.fbbva.es/TLFU/dat/DT_2006_09.pdf), acceso en 2015-09-06.
- [183] Palmer, S. B. (1990). The early history of html. <http://infomesh.net/html/history/early/>, acceso en 2015-06-29.
- [184] Pellegrini, T. (2009). Every technology is socially and culturally shaped. <http://s.coop/1wttd>, acceso en 2015-09-06.
- [185] Piatetsky, G. (2015). Interview with Chris Wiggins, NYTimes Chief Data Scientist. <http://www.kdnuggets.com/2015/01/exclusive-interview-chris-wiggins-nytimes-chief-data-scientist.html>, acceso en 2015-07-23.
- [186] Piatetsky-Shapiro, G. (2000). Knowledge Discovery in Databases: 10 years after. [http://cort.as/XL\\_n](http://cort.as/XL_n), accedida en 2015-09-02.
- [187] Plant, S. (1998). *Ceros + Unos: Mujeres Digitales + la Nueva Tecnocultura*. Ciencias sociales. Ediciones Destino.
- [188] Pooley, S. (2003). Web Matters. The Essence of the Web. <http://sbpooley.home.xs4all.nl/webmatters/essence.html>, acceso en 2015-09-09.
- [189] Popova, M. (2014). The Birth of the Information Age: How Paul Otlet's Vision for Cataloging and Connecting Humanity Shaped Our World. <https://www.brainpickings.org/2014/06/09/paul-otlet-alex-wright/>.
- [190] Porto Renó, D. . J. F. (2012). *Periodismo transmedia*. Fragua.
- [191] Postel, J. (1981). Internet Protocol. <http://www.ietf.org/rfc/rfc791.txt>, acceso en en 2015-09-05.
- [192] Prodromou, E. (2010). What is the federated social web? <http://s.coop/1wtte>, acceso en 2015-09-04.

- [193] Rabin, J. . C. M. (2008). Mobile Web Best Practices 1.0. Technical report, W3C. <http://www.w3.org/TR/2008/REC-mobile-bp-20080729/>, acceso en 2015-09-09.
- [194] Ragget, D. . A. L. H. . I. J. (1998). HTML 4.0 Recommendation. Technical report, W3C. <http://www.w3.org/TR/1998/REC-html40-19980424>, acceso en 2015-08-24.
- [195] Ragget, D. . A. L. H. . I. J. (1999). HTML 4.01 Specification. Technical report, W3C. <http://www.w3.org/TR/1999/REC-html401-19991224>, accedida en 2015-08-24.
- [196] Raggett, D. (1997). HTML 3.2 Reference Specification. Recommendation, W3C. <http://www.w3.org/TR/REC-html32-19970114>, acceso en 2015-08-22.
- [197] Richardson, L. . S. R. (2007). *RESTful Web Services*. O'Reilly Media. Editor: Loukides, Mike.
- [198] Rivero, R. (2010). API para la Lotería de Navidad. [http://tecnologia.elpais.com/tecnologia/2010/12/21/actualidad/1292925666\\_850215.html](http://tecnologia.elpais.com/tecnologia/2010/12/21/actualidad/1292925666_850215.html), acceso en 2015-09-22.
- [199] Rivero, R. (2012). Bienvenido a la Revolución. <http://blogs.elpais.com/el-cambio-por-dentro/2012/02/bienvenido-a-la-revolucion.html>, acceso en 2015-09-23.
- [200] Rivoal, F. (2012). Media Queries. Technical report, W3C. <http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>, acceso en 2015-08.24.
- [201] Rogers, S. (2010). Florence nightingale, data journalist: information has always been beautiful. <http://www.theguardian.com/news/datablog/2010/aug/13/florence-nightingale-graphics>, acceso en 2015-08-20.
- [202] Rogers, S. (2012a). Anyone can do it. data journalism is the new punk. <http://s.coop/1wv89>, acceso en 2015-08-12.

- [203] Rogers, S. (2012b). Open data journalism. <http://s.coop/1wv8b>, acceso en 2015-08-22.
- [204] Sádaba, I. (2008). El p2p: ¿más allá del capitalismo? *Minerva*, 1(9):42-45. <http://www.revistaminerva.com/articulo.php?id=275>, acceso en 2015-08-25.
- [205] Sandhaus, E. (2010). Build your own NYT Linked Data Application. <http://s.coop/nytdapp>.
- [206] Schönberger, V. . K. C. . A. J. (2013). *Big data : la revolución de los datos masivos*. Turner Publicaciones, S.L.
- [207] Shannon, C., Weaver, W., Machado, T., and Pérez-Amat, R. (1981). *Teoría matemática de la comunicación*. Forja.
- [208] Situacionista, I. (1991). *Textos sobre la Internacional Situacionista*. Archivo Situacionista Hispano.
- [209] Skrenta (2005). The incremental web. <http://blog.topix.com/2005/02/the-incremental-web.html>, acceso en 2015-08-24.
- [210] Stewart, W. (2014). Living internet. <http://www.livinginternet.com/>, acceso en 2015-08-15.
- [211] Sullivan, W. (2009). Four reasons your news org should use apis. <http://s.coop/mediaapi>, acceso en 2015-08-24.
- [212] Sydell, L. (2014). The forgotten female programmers who created modern tech. <http://cort.as/XJWh>, acceso en 2015-08-18.
- [213] Tech Channel, A. (2015). The UNIX System: Making Computers More Productive. <https://www.youtube.com/watch?v=tc4R0CJYbm0>, accedido 2015-08-15.
- [214] Tenore, M. J. (2010). Public media api could be engine of innovation'for journalism. <http://s.coop/mediaapiengine>, acceso en 2015-08-02.

- [215] Thibodeaux, T. (2011). 5 tips for getting started in data journalism. <http://s.coop/1wv8e>, acceso en 2015-08-22.
- [216] Tiqqun (2015). *La hipótesis cibernética*. Acuarela Libros Antonio Machado Libros.
- [217] Tomàs, E. (2014). Qué es rest. <http://s.coop/1wttg>, acceso en 2015-08-16.
- [218] Toole, B. (2014). Ada Byron, Lady Lovelace. <http://www.agnesscott.edu/lriddle/women/love.htm>, acceso en 2015-08-21.
- [219] Torres, R. M. (2005). Sociedad de la información / sociedad del conocimiento. <http://www.ub.edu/prometheus21/articulos/obsciberprome/socinfocon.pdf>, acceso en 2015-08-25.
- [220] UNESCO (2007). Ciencia, tecnología y género: informe internacional. [http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/SC/pdf/sc\\_stg\\_executive\\_summary-es.pdf](http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/SC/pdf/sc_stg_executive_summary-es.pdf), acceso en 2015-09-24.
- [221] Unión Internacional de Telecomunicaciones UIT (2005). Cumbre Mundial sobre la Sociedad de la Información. Documentos Finales. <https://www.itu.int/net/wsis/outcome/booklet-es.pdf>, acceso en 2015-09-20.
- [222] Uyi Idehen, K. (2014). Federated identity attribute based resource access control. <http://www.slideshare.net/kidehen/how-virtuoso-enables-attributed-based-access-controls>, acceso en 2015-09-01.
- [223] Vallejo Martín-Albo, C. (2010). Realidad aumentada. <http://s.coop/1wtth>, acceso en 2015-09-04.
- [224] Van Beijnum, I. (2014). With the americas running out of ipv4, it's official: The internet is full. <http://s.coop/1wtti>, acceso en 2015-08-16.

- [225] van Kesteren, A. . J. A. . J. S. . H. R. M. S. (2014). XMLHttpRequest Level 1. Technical report, W3C. <http://www.w3.org/TR/XMLHttpRequest/>, acceso en 2015-09-06.
- [226] Vander Wal, T. (2007). Folksonomy Coinage and Definition. <http://vanderwal.net/folksonomy.html>, acceso en 2015-08-26.
- [227] Velasco, H. M. . H. M. V. M. . Á. D. d. R. (2004). *La lógica de la investigación etnográfica: un modelo de trabajo para etnógrafos de la escuela*. Colección Estructuras y procesos. Trotta.
- [228] Velasco, J. J. (2011). Historia del Software: Dennis Ritchie. <http://s.coop/1wttj>, acceso en 2015-08-15.
- [229] Virtualnauta (2015a). Caracteres especiales. <http://cort.as/XJYr>, acceso en 2015-08-15.
- [230] Virtualnauta (2015b). Guía de referencia código ASCII. <http://cort.as/XJYo>, acceso en 2015-08-15.
- [231] Virtualnauta (2015c). Html código url. <http://cort.as/XJYt>, acceso en 2015-08-15.
- [232] Virtualnauta (2015d). Mensajes de status HTTP. <http://cort.as/XJYy>, acceso en 2015-08-15.
- [233] W3C (2001). Deep Linking in the World Wide Web. <http://www.w3.org/2001/tag/doc/deeplinking.html>, acceso en 2015-09-06.
- [234] W3C (2005). W3C in 7 points. <http://www.w3.org/Consortium/Points/>, acceso en 2015-08-20.
- [235] W3C (2010). Character encodings for beginners. <http://cort.as/XJZm>, acceso en 2015-08-15.

- [236] W3C (2011). Semantic Web Education an Outreach (SWEO) Interest Group. <http://www.w3.org/blog/SWEO/>, acceso en 2015-09-06.
- [237] W3C (2014a). W3C launches push for social web application interoperability. <http://www.w3.org/2014/06/social.html.en>, acceso en 2015-09-03.
- [238] W3C (2014b). W3C launches push for social web application interoperability. <http://www.w3.org/blog/news/archives/3958>, acceso en 2015-09-03.
- [239] W3C (2015). Cool URIs don't change. <http://www.w3.org/Provider/Style/URI.html>, acceso en 2015-08-18.
- [240] W3C (2015). HTML Working Group. <http://www.w3.org/html/wg/>, acceso en 2015-08-15.
- [241] W3C (2015a). Html/publications. <http://www.w3.org/wiki/HTML/Publications>, acceso en 2015-08-15.
- [242] W3C (2015b). SweoIG/TaskForces/CommunityProjects/LinkingOpenData. <http://cort.as/XJZh>, acceso en 2015-09-06.
- [243] W3C (2015c). W3C Mission. <http://www.w3.org/Consortium/mission>, acceso en 2015-08-20.
- [244] w3schools (2015). HTML Character Sets. [http://www.w3schools.com/tags/ref\\_charactersets.asp](http://www.w3schools.com/tags/ref_charactersets.asp), acceso en 2015-08-15.
- [245] Wachoswski, Andy Lana Wachowski, (Directors) (1999). The matrix. Warner Bros. <http://www.imdb.com/title/tt0133093>, acceso en 2015-08-23.
- [246] WHATWG (2004a). FAQ. <https://whatwg.org/specs/>, acceso en 2015-08-15.
- [247] WHATWG (2004b). Web Hypertext Application Technology Working Group Specifications. <https://whatwg.org/specs/>, acceso en 2015-08-15.

- [248] WHATWG (2011). Html is the new html5. <https://blog.whatwg.org/html-is-the-new-html5>, acceso en 2015-08-15.
- [249] WHATWG (2015). MetaExtensions. <https://wiki.whatwg.org/wiki/MetaExtensions>, acceso en 2015-09-30.
- [250] Wiener, N. (1985). *Cibernética o el control y comunicación en animales y máquinas*. Metatemas: Superínfimos. Tusquets Editores.
- [251] Wikilibros (2015). Redes informáticas/protocolos tcp y udp en el nivel de transporte. <http://s.coop/1wttl>, acceso en 2015-08-17.
- [252] Wikipedia (2015a). American national standards institute. [https://en.wikipedia.org/wiki/American\\_National\\_Standards\\_Institute](https://en.wikipedia.org/wiki/American_National_Standards_Institute), acceso en 2015-08-15.
- [253] Wikipedia (2015b). Browser wars. [https://en.wikipedia.org/wiki/Browser\\_wars](https://en.wikipedia.org/wiki/Browser_wars), acceso en 2015-09-09.
- [254] Wikipedia (2015c). Burbuja punto com. [https://es.wikipedia.org/wiki/Burbuja\\_punto\\_com](https://es.wikipedia.org/wiki/Burbuja_punto_com), acceso en 2015-09-08.
- [255] Wikipedia (2015d). Charles joseph minard. [https://en.wikipedia.org/wiki/Charles\\_Joseph\\_Minard](https://en.wikipedia.org/wiki/Charles_Joseph_Minard), acceso en 2015-08-15.
- [256] Wikipedia (2015e). Create, read, update and delete. <http://cort.as/XJbC>, accedida 2015-08-18.
- [257] Wikipedia (2015f). Google Chrome. [https://en.wikipedia.org/wiki/Google\\_Chrome](https://en.wikipedia.org/wiki/Google_Chrome), accedida en 2015-08-15.
- [258] Wikipedia (2015g). Ipv6. <https://es.wikipedia.org/wiki/IPv6>, accedida en 2015-09-06.

- [259] Wikipedia (2015h). John snow (physician). [https://es.wikipedia.org/wiki/IPv6https://en.wikipedia.org/wiki/John\\_Snow\\_%28physician%29](https://es.wikipedia.org/wiki/IPv6https://en.wikipedia.org/wiki/John_Snow_%28physician%29), acceso en 2015-09-26.
- [260] Wikipedia (2015i). Lenguaje no sexista. [https://es.wikipedia.org/wiki/Lenguaje\\_no\\_sexista](https://es.wikipedia.org/wiki/Lenguaje_no_sexista), acceso en 2015-08-18.
- [261] Wikipedia (2015j). List of social networking websites. [https://en.wikipedia.org/wiki/List\\_of\\_social\\_networking\\_websites](https://en.wikipedia.org/wiki/List_of_social_networking_websites), acceso en 2015-09-06.
- [262] Wikipedia (2015k). Memex. <https://en.wikipedia.org/wiki/Memex>, acceso en 2015-09-06.
- [263] Wikipedia (2015l). Números de puerto. [https://es.wikipedia.org/wiki/Anexo:N{C3}BAmoros\\_de\\_puerto](https://es.wikipedia.org/wiki/Anexo:N{C3}BAmoros_de_puerto), acceso en 2015-08-15.
- [264] Wikipedia (2015m). Representational state transfer. [https://es.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://es.wikipedia.org/wiki/Representational_State_Transfer), accedida 2015-08-16.
- [265] Wikipedia (2015n). Sparql. <https://es.wikipedia.org/wiki/SPARQL>, acceso en 2015-08-17.
- [266] Wikipedia (2015o). Timeline of Web Browsers. <http://s.coop/1wts2>, acceso en 2015-09-09.
- [267] Wikipedia (2015p). Uri scheme. [https://es.wikipedia.org/wiki/URI\\_scheme](https://es.wikipedia.org/wiki/URI_scheme), acceso en 2015-08-17.
- [268] Wikipedia (2015q). Utf-8. <https://es.wikipedia.org/wiki/UTF-8>, accedida en 2015-08-15.
- [269] Wikipedia (2015r). Violawww. <https://en.wikipedia.org/wiki/ViolaWWW>, accedida en 2015-08-15.

- [270] Wikiquote (2015). Tim berners lee. [https://en.wikiquote.org/wiki/Tim\\_Berners-Lee](https://en.wikiquote.org/wiki/Tim_Berners-Lee), acceso en 2015-09-09.
- [271] Winer, D. (2012). Ec2 for poets. <http://ec2.forpoets.org/>, acceso en 2015-09-21.
- [272] Witten, I. H. . E. F. (2005). *Data Mining. Practical Machine Learning Tools and Techniques*. Elsevier, San Francisco.
- [273] Wium Lie, H. . B. B. (1999). Cascading Style Sheets (CSS1) Level 1 Specification. Recommendation, W3C. <http://www.w3.org/TR/1999/REC-CSS1-19990111>, acceso en 2015-09-22.
- [274] Zeldman, J. (2006). Web 3.0. <http://alistapart.com/article/web3point0>, acceso en 2015-09-06.
- [275] Zimmermann, H. (1980). OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, COM-28(4). <https://www.cs.cmu.edu/~srini/15-744/F02/readings/Zim80.pdf>, acceso en 2015-08-16.
- [276] Zorn, I. . S. M. . E. R. . C. S. . H. S. (2007). *Gender Designs IT: Construction and Deconstruction of Information Society Technology*. Studien Interdisziplinäre Geschlechterforschung. VS Verlag für Sozialwissenschaften.