

TRABAJO DE FIN DE MÁSTER SOBRE SCHEMA.ORG

MÁSTER UNIVERSITARIO DE INFORMÁTICA

Consultor: SAMIR KANAAN

06/2017

Rafael Dombidau



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>TRABAJO DE FIN DE MÁSTER SOBRE SCHEMA.ORG</i>
Nombre del autor:	<i>Rafael Dombidau</i>
Nombre del consultor:	<i>SAMIR KANAAN IZQUIERDO</i>
Fecha de entrega (mm/aaaa):	<i>06/2017</i>
Área del Trabajo Final:	<i>Web Semántica y Representación del Conocimiento</i>
Titulación:	<i>Máster Universitario de Informática</i>
Resumen del trabajo (máximo 250 palabras):	
<p>Schema.org es una iniciativa lanzada en 2011 por los principales buscadores: Bing, Google, Yahoo y más adelante Yandex. Proponen el uso del vocabulario schema.org y la sintaxis de Microdatos, RDFa y JSON-LD, para marcar el contenido de los sitios web con metadatos sobre ellos. Tal marcado hace disponible el significado de los datos web a servicios tales como buscadores web, motores de reserva, comparadores de precios, etc.</p>	
Abstract (in English, 250 words or less):	
<p>Schema.org is an initiative launched in 2011 by the major search engines: Bing, Google, Yahoo and later Yandex. They propose using the schema.org vocabulary and the Microdata, RDFa, or JSON-LD syntax to mark up website content with metadata about itself. Such mark up gives availability of the meaning of the web data to services as Web search, price comparison, reservation engines, etc.</p>	
Palabras clave (entre 4 y 8):	
<p>Schema.org, Web Semántica (Semantic Web), HTML, Marcado de datos (Data markup), Microdatos (Microdata), RDFa, JSON-LD.</p>	

ÍNDICE

FICHA DEL TRABAJO FINAL.....	3
ÍNDICE	4
1. Introducción	7
1.1 Contexto y justificación del trabajo.....	7
1.2 Objetivos del trabajo.....	7
1.3 Enfoque y método seguido	7
1.4 Planificación del trabajo.....	7
1.4.1. MATERIAL.....	7
1.4.2. TAREAS E HITOS.....	8
1.4.3. DIAGRAMA DE GANTT.....	9
1.5 Breve resumen de los productos obtenidos	11
2. INTRODUCCIÓN.....	12
3. Uso de Schema.org.....	12
3.1. Modelo de datos	20
3.1.1. Clases (Tipos).....	20
3.1.2. Propiedades.....	21
3.1.3. Tipos de datos	21
3.1.4. Extensión	22
3.2. SINTAXIS	23
3.2.1. Microdatos	24
3.2.2. RDFa (Resource Description Framework in Attributes)	24
3.2.3. JSON-LD JavaScript Object Notation for Linked Data	25
3.2.3.a. Elementos Sintácticos y Palabras clave.....	25
3.3. Ejemplo.....	26
3.3.1. Sin marcado.....	26
3.3.2. Marcado	26
3.3.2.a. Microdata	26
3.3.2.b. RDFa	26
3.3.2.c. JSON-LD	27
3.4. FORMAS DE MARCADO DE DATOS ESTRUCTURADOS	27
3.4.1. MANUAL.....	27

3.4.2. MEDIANTE HERRAMIENTAS	27
3.4.3. UTILIZANDO PLUGINS	27
4. OTROS VOCABULARIOS	28
4.1. FOAF (Friend of a Friend)	28
4.2. DUBLIN CORE.....	31
4.3. SKOS	33
4.3.1. Categorías de elementos.....	33
4.4. Open Graph protocol (OGP).....	35
4.4.1. Metadatos Básicos	35
4.4.2. Otros Metadatos de OGP	36
5. VENTAJAS, INCONVENIENTES Y RETOS DE SCHEMA.ORG	36
5.1. VENTAJAS	36
5.2. DESVENTAJAS	36
5.3. RETOS.....	37
6. Guía Práctica.....	39
6.1. Marcado de una película (Movie).....	40
6.1.1. HTML básico.....	40
6.1.2. Marcado mediante intercalado de Microdatos: itemscope e itemtype	41
6.1.3. Añadimos itemprop.....	41
6.1.4. Añadimos el marcado de la calificación promedio (AggregateRating)	42
6.1.5. Marcado de RDFa.....	43
6.1.6. JSON-LD.....	45
6.2. Marcado de un Producto (Product).....	46
6.2.1. HTML básico	46
6.2.2. Marcado mediante intercalado de Microdatos: itemscope, itemtype e itemprop	46
6.2.3. Marcado de RDFa.....	47
6.2.4. JSON-LD.....	48
6.3. Marcado de un Evento (Event)	49
6.3.1. HTML básico	49
6.3.2. Marcado mediante intercalado de Microdatos: itemscope, itemtype e itemprop	50
6.3.3. Marcado de RDFa.....	50
6.3.4. JSON-LD.....	51
7. CONCLUSIONES	51

GLOSARIO	53
Fuentes	55
Índice de Figuras	56
Tablas	57
Índice de entradas	57

1. Introducción

1.1 Contexto y justificación del trabajo

El contenido en código HTML puro publicado en la web no refleja los datos estructurados subyacentes. Las aplicaciones que requieran acceso a esos datos deben hacer un complejo esfuerzo para extraerlos del HTML. Para facilitar el trabajo de intercambio de información entre aplicaciones y web, entre publicadores y usuarios de los datos, surgieron una serie de vocabularios distintos e independientes. Esta variedad de vocabularios creó confusión entre los webmasters sobre qué vocabularios utilizar y cómo. Para intentar remediar esta situación surge Schema.org, proporcionando un único vocabulario común.

1.2 Objetivos del trabajo

Este Trabajo de Fin de Máster (TFM) tratará sobre Schema.org, investigando:

- Qué es, por qué surge, quiénes y para qué lo utilizan.
- Estadísticas de uso.
- Aplicaciones que lo utilizan: Rich Snippets (Fragmentos enriquecidos) de Google, Knowledge Graph (Grafo de Conocimiento), Cortana de Microsoft, Pinterest, Searchlight/Siri de Apple, correos electrónicos como Gmail de Google.
- Su vocabulario y su extensión.
- Su codificación mediante las especificaciones Microdata, RDFa y JSON-LD.
- Cómo utilizarlo. Herramientas para generar codificación de Schema.org
- Ventajas e inconvenientes de su uso. Retos a los que se enfrenta.
- Otros vocabularios similares: Dublin Core, SKOS (Simple Knowledge Organization System), FOAF (Friend of a Friend), OGP (Open Graph Protocol).

1.3 Enfoque y método seguido

El TFM consistirá en un trabajo de investigación sobre Schema.org. Para ello, se buscará información de varias fuentes, sobre todo a través de internet.

1.4 Planificación del trabajo

1.4.1. MATERIAL

- Punto de trabajo (PC de sobremesa) con acceso a internet para la búsqueda de información.
- Microsoft Project 2013 para la planificación.
- Procesador de textos (Microsoft Word 2010) para redactar la memoria.

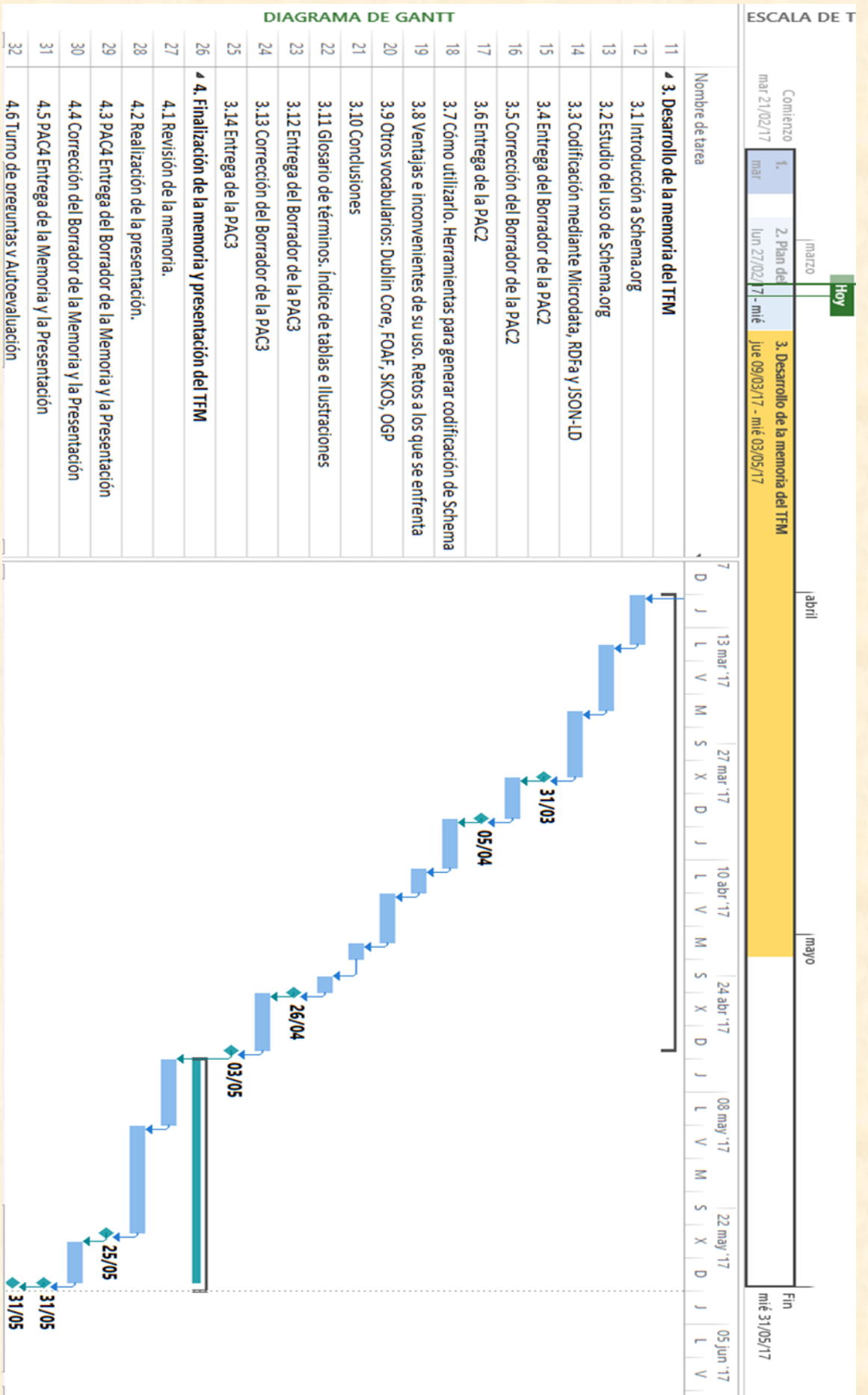
-Microsoft Power Point 2010 y Camtasia Studio 8 para la presentación.

1.4.2. TAREAS E HITOS

	Nombre de tarea	Duración	Comienzo	Fin
1	1. Definición del proyecto	4 días	mar 21/02/17	vie 24/02/17
2	1.1 Descarga de documentación disponible en el aula virtual	0 días	mar 21/02/17	mar 21/02/17
3	1.2 Estudio de la documentación	2 días	mar 21/02/17	mié 22/02/17
4	1.3 Preparación de las herramientas para la realización del proyecto	2 días	jue 23/02/17	vie 24/02/17
5	2. Plan del proyecto	8 días	lun 27/02/17	mié 08/03/17
6	2.1 Búsqueda de información sobre el TFM en internet mediante buscadores y biblioteca virtual.	1 día	lun 27/02/17	lun 27/02/17
7	2.2 Planificación del trabajo. Establecer objetivos, apartados, incidencias, riesgos y sus planes de contingencia; entregas de las PAC, evaluación del material necesario.	1 día	mar 28/02/17	mar 28/02/17
8	2.3 Temporalización	2 días	mié 01/03/17	jue 02/03/17
9	2.4 Borrador de Plan de trabajo	2 días	vie 03/03/17	lun 06/03/17
10	2.5 Corrección de Plan de trabajo	2 días	mar 07/03/17	mié 08/03/17
11	3. Desarrollo de la memoria del TFM	39 días	jue 09/03/17	mié 03/05/17
12	3.1 Introducción a Schema.org	4 días	jue 09/03/17	mar 14/03/17
13	3.2 Estudio del uso de Schema.org	6 días	mié 15/03/17	mié 22/03/17
14	3.3 Codificación mediante Microdata, RDFa y JSON-LD	6 días	jue 23/03/17	jue 30/03/17
15	3.4 Entrega del Borrador de la PAC2	0 días	vie 31/03/17	vie 31/03/17
16	3.5 Corrección del Borrador de la PAC2	3 días	vie 31/03/17	mar 04/04/17
17	3.6 Entrega de la PAC2	0 días	mié 05/04/17	mié 05/04/17
18	3.7 Cómo utilizarlo. Herramientas para generar codificación de Schema	4 días	mié 05/04/17	lun 10/04/17
19	3.8 Ventajas e inconvenientes de su uso. Retos a los que se enfrenta.	3 días	mar 11/04/17	jue 13/04/17
20	3.9 Otros vocabularios: Dublin Core, FOAF, SKOS, OGP	4 días	vie 14/04/17	mié 19/04/17
21	3.10 Conclusiones	2 días	jue 20/04/17	vie 21/04/17
22	3.11 Glosario de términos. Índice de tablas e ilustraciones.	2 días	lun 24/04/17	mar 25/04/17
23	3.12 Entrega del Borrador de la PAC3	0 días	mié 26/04/17	mié 26/04/17
24	3.13 Corrección del Borrador de la PAC3	5 días	mié 26/04/17	mar 02/05/17
25	3.14 Entrega de la PAC3	0 días	mié 03/05/17	mié 03/05/17
26	4. Finalización de la memoria y presentación del TFM	20 días	jue 04/05/17	mié 31/05/17
27	4.1 Revisión de la memoria.	6 días	jue 04/05/17	jue 11/05/17
28	4.2 Realización de la presentación.	9 días	vie 12/05/17	mié 24/05/17
29	4.3 PAC4 Entrega del Borrador de la Memoria y la Presentación	0 días	jue 25/05/17	jue 25/05/17

30	4.4 Corrección del Borrador de la Memoria y la Presentación	3 días	vie 26/05/17	mar 30/05/17
31	4.5 PAC4 Entrega de la Memoria y la Presentación	0 días	mié 31/05/17	mié 31/05/17
32	4.6 Turno de preguntas y Autoevaluación	0 días	mié 31/05/17	mié 31/05/17

1.4.3. DIAGRAMA DE GANTT



1.5 Breve resumen de los productos obtenidos

- Estudio sobre el vocabulario Schema.org: ¿Qué es?, ¿quiénes, para qué y cómo lo usan?
- Codificación mediante Microdatos, RDFa (Resource Description Framework in Attributes) y JSON-LD (JavaScript Object Notation for Linked Data)
- Otros vocabularios: Dublin Core, FOAF (Friend Of A Friend), SKOS (Simple Knowledge Organization System), OGP (Open Graph Protocol).

2. INTRODUCCIÓN

Schema.org está patrocinada por Google, Microsoft, Yahoo y Yandex y consiste en una actividad colaborativa, comunitaria y abierta dedicada a crear, mantener y promover esquemas para marcado de datos estructurados en Internet, páginas web, mensajes de correo y otros.

Su vocabulario puede ser codificado mediante RDFa, Microdata y JSON-LD y cubre entidades, relaciones entre entidades y acciones, y es ampliable mediante un modelo de extensión bien documentado. Más de 10 millones de sitios utilizan Schema.org para el marcado de sus páginas web y mensajes de correo electrónico. Muchas de las aplicaciones de Google, Microsoft, Pinterest, Yandex y otros ya utilizan estos vocabularios para potenciarlos.

Schema.org surge por la necesidad de aunar esfuerzos para resolver el caos de la multiplicidad de vocabularios que surgieron para el marcado de datos estructurados en internet. Los vocabularios se comenzaron a usar para facilitar el acceso a los datos web por parte de las aplicaciones que necesitaban construir analizadores para poder extraer esos datos mediante un proceso complejo y poco fiable.

3. Uso de Schema.org

Generalmente, el código HTML utilizado en el diseño Web se utiliza para la presentación de información. Sin embargo, no nos da acceso al contenido de la información. No conocemos los datos estructurados subyacentes de la Web.

Por regla general, las etiquetas (*tags*) HTML de páginas web le indican al navegador (*browser*) de internet cómo mostrar la información contenida en la etiqueta. Por ejemplo, `<h1>Nombre</h1>` indica al navegador que muestre la cadena de texto “Nombre” en el formato de cabecera 1 (*heading 1*). Sin embargo, la etiqueta HTML no ofrece información del significado de la cadena de texto.

Las aplicaciones que quieran acceder a la información subyacente en páginas Web deben extraerla, convirtiendo el HTML en datos estructurados. Este proceso de extracción de datos es complicado y propenso a errores.

Para facilitar el intercambio de información entre publicadores y consumidores de datos de la web, surgieron diversas especificaciones:

-XML (eXtensible Markup Language): lenguaje de marcas Extensible que define un conjunto de reglas para codificar documentos que puedan ser entendidos por humanos y por máquinas.

-MFC (Meta Content Framework): Marco de Meta Contenido, utilizado para representación de conocimiento en la web, describiendo objetos, sus atributos y las relaciones entre ellos.

-RDF (Resource Description Framework), RDFS (Resource Description Framework Schema) y OWL (Web Ontology Language): Desarrollados para la sintaxis y el modelo de datos.

-Vocabularios como RSS (Rich Site Summary) para personalizar páginas de inicio, vCard/hCard para intercambio de información de contacto, hCalendar basado en iCalendar para intercambio de calendario. FOAF (Friend of a Friend) para datos de redes sociales.

Las aplicaciones de búsqueda web fueron las que más consumían los datos estructurados publicados. Los buscadores web comenzaron a mejorar los resultados de búsquedas, aumentando los fragmentos de resultados con los datos estructurados obtenidos. Aunque ello beneficiaba tanto a usuarios como a desarrolladores web (*webmasters*), se utilizaban diferentes vocabularios y cada motor de búsqueda recomendaba uno distinto.

Con la intención de resolver esta confusión creada por los diversos vocabularios surgió Schema.org.

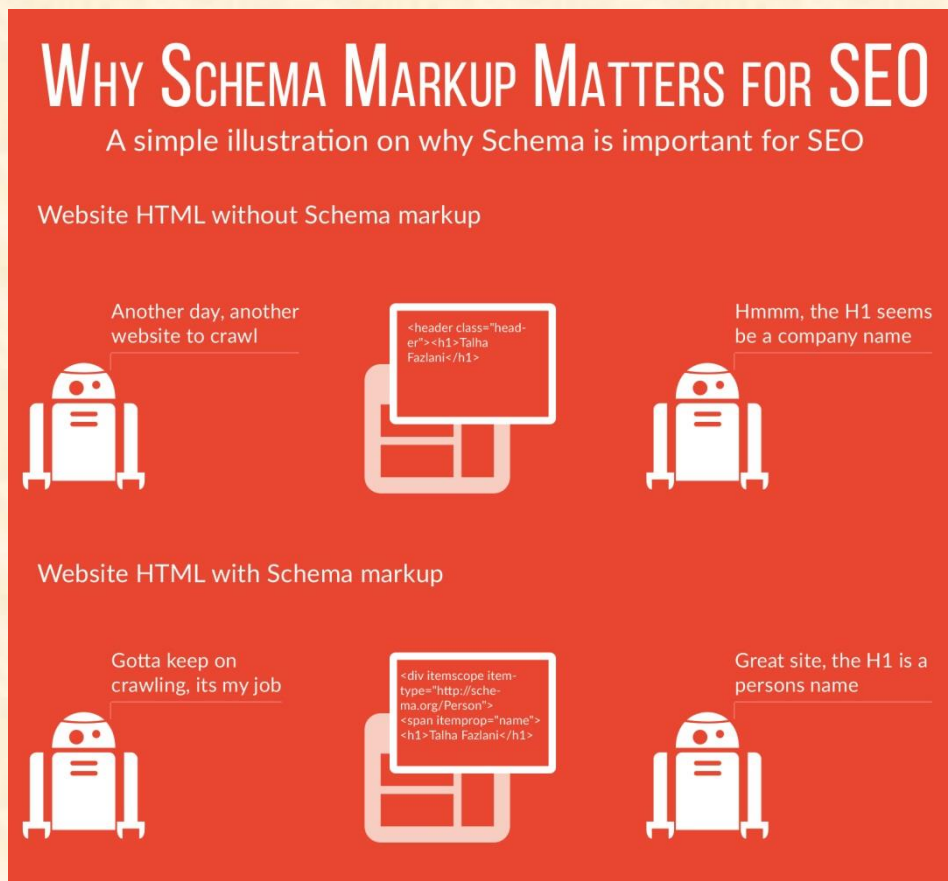


Fig. 1. seopressor.com

La Fig. 1 ilustra cómo funciona un buscador de información web. Facilitar el acceso a los datos estructurados subyacentes mediante Schema puede conseguir que nuestra web consiga un mejor posicionamiento (Search Engine Optimization o SEO) en las páginas de resultados (Search Engine Results Pages o SERPs) mejorando su tasa de clicks (Click Through Rate o CTR).

Schema proporciona una colección de vocabularios compartidos para el etiquetado (*markup*) semántico de páginas web que puedan ser entendidos por los principales buscadores como Google, Microsoft, Yandex y Yahoo! y por otras aplicaciones.

“Una gran parte del vocabulario en schema.org se inspiró en trabajos anteriores como FOAF, OpenCyc, Microformatos, etc... Muchos términos de schema.org provienen de colaboraciones.” (1)

El vocabulario se puede añadir como Microdatos (de WHATWG HTML), RDFa (Resource Description Framework in Attributes de W3C) o JSON-LD (JavaScript Object Notation for Linked Data).

Los esquemas son un conjunto de clases o "tipos" jerárquicos, cada uno asociado a un conjunto de propiedades. En la Fig. 3 y la Fig. 2 vemos un ejemplo de cómo funciona la jerarquía de tipos y alguna de las relaciones que pueden establecerse para la clase CreativeWork (Trabajo Creativo).

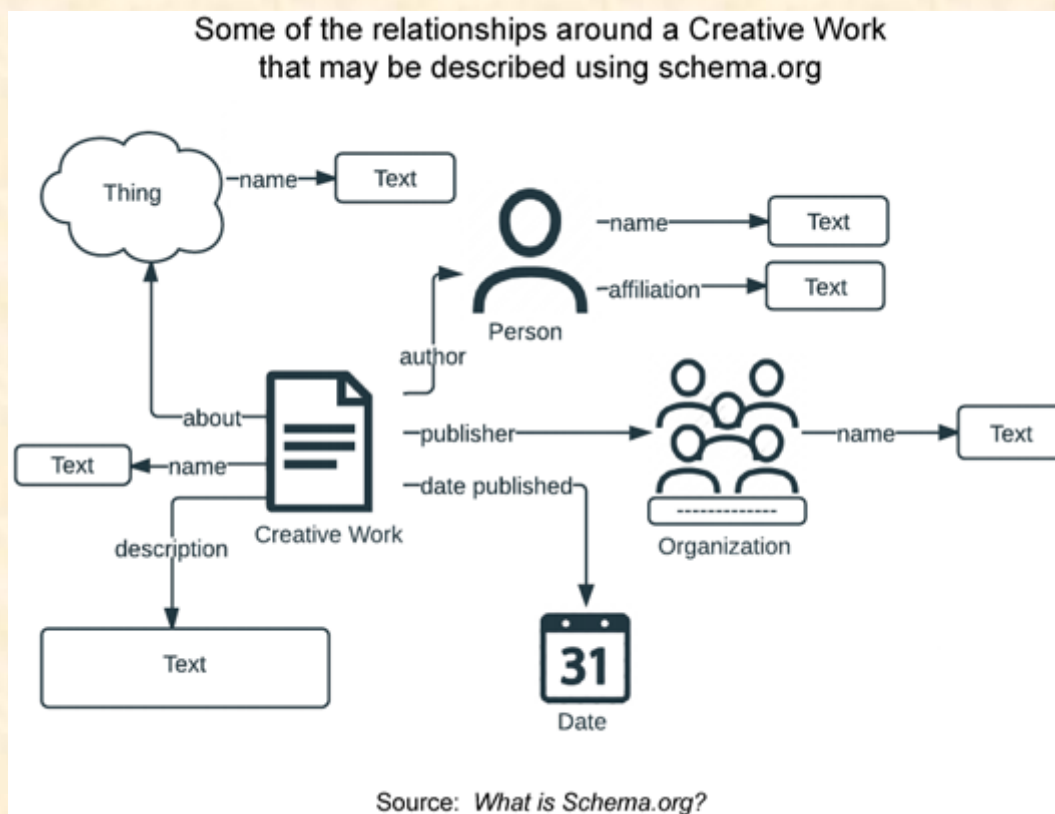


Fig. 2. Creative Work (Trabajo Creativo) <http://publications.cetis.org.uk/2014/960>

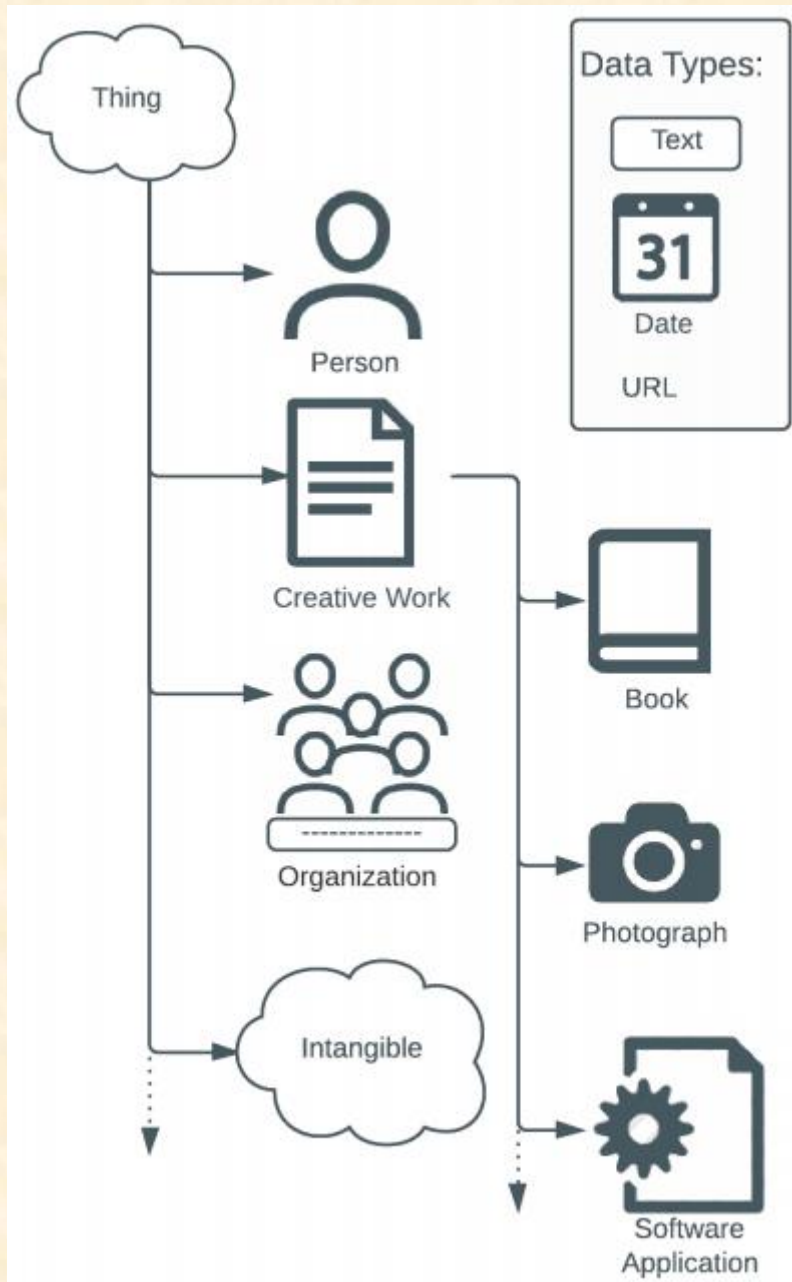


Fig. 3. Jerarquía de Tipos de Schema.org. blogs.pjjk.net/phil/topics/schema-org/

El vocabulario básico se compone actualmente de 589 clases o tipos (cuya primera letra es mayúscula), 860 Propiedades (cuya primera letra es minúscula), y 114 valores de enumeración¹. (2)

¹ Enumeración: propiedad que puede tomar uno de sólo un pequeño número de valores posibles.

En la Fig. 4 observamos un ejemplo de cómo funciona la Ontología de Schema.org: en los círculos se muestran las clases, las flechas discontinuas representan las relaciones de herencia entre clases y las flechas continuas representan las relaciones entre clases mediante propiedades.

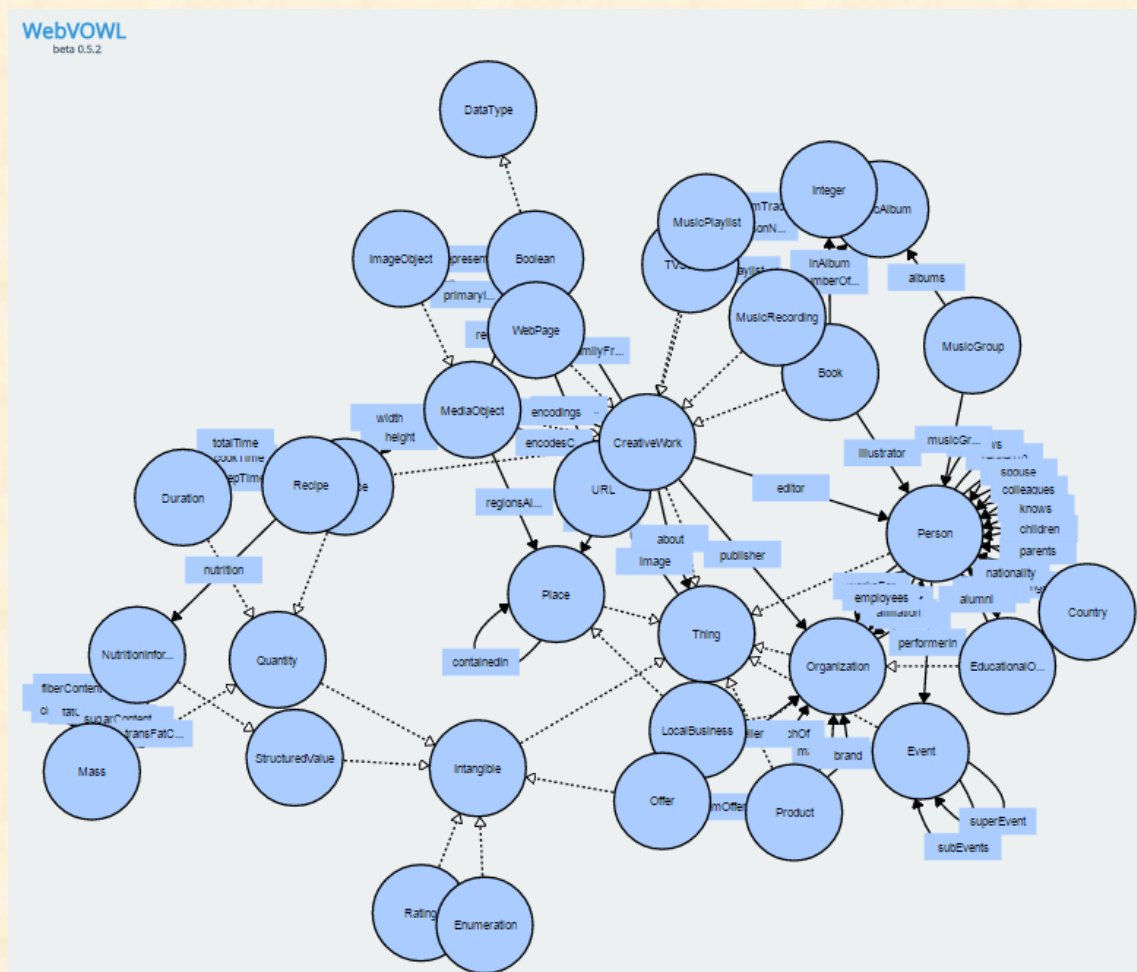


Fig. 4. Ontología [Schema.org-2013.03.25](#) representada mediante [WebVOWL](#)

Aplicaciones que hacen uso de Schema.org (3) y (4):

-Grafo de Conocimiento (Knowledge Graph). El mercado le proporciona datos para entidades conocidas como logo, contacto e información social (Fig. 5).

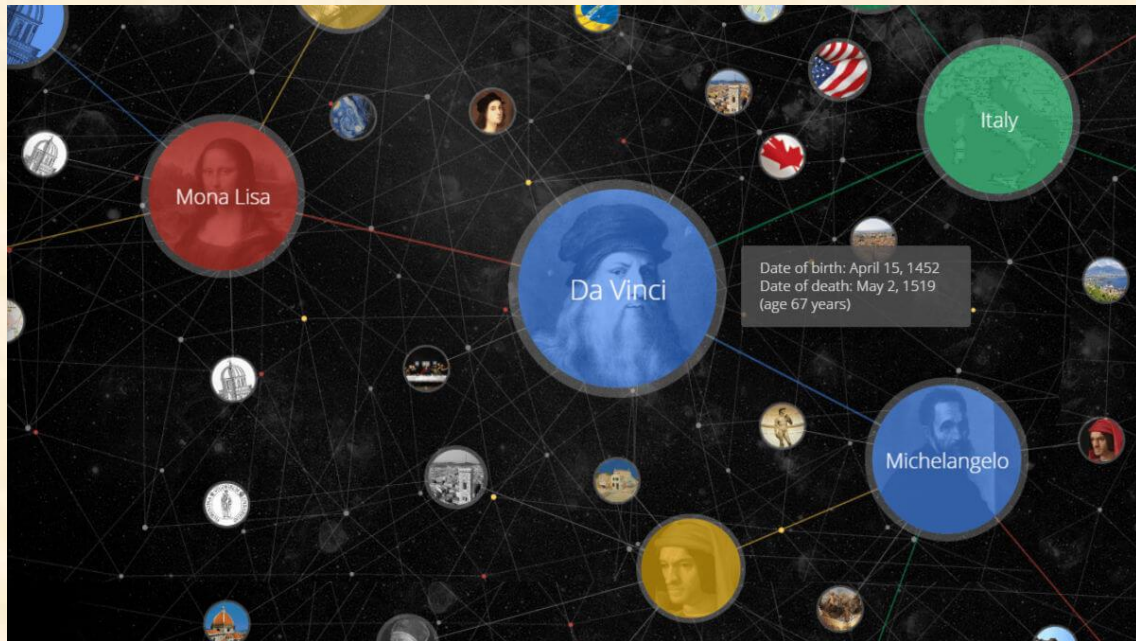


Fig. 5. Knowledge Graph de Google. searchengineland.com

-Fragmentos Enriquecidos (Rich Snippets) de Google. Fue la primera en usarla. Supone una representación más rica de los resultados de búsqueda, tal como muestra la Fig. 6.

La mejor tarta de manzana, por Martha Stewart | Telva.com
www.telva.com › Inicio › Cocina ▾
 12 ene. 2017 - La tarta de manzana es el ejemplo más conocido de tarta de fruta, y para muchos sirve como introducción a las tartas cubiertas. Para empezar, ...


La mejor tarta de manzana (4.4/5) - Rebañando
 www.rebanando.com/receta-56751-tarta-de-manzana.htm ▾
 ★★★★★ Valoración: 4,4 - 400 votos
 La mejor receta de tarta de manzana explicada en 4 pasos. Con una masa casera y un toque de mermelada, esta fácil tarta es un éxito garantizado.

Fig. 6. Comparación entre un resultado de búsqueda normal y un resultado enriquecido de una Receta (Recipe).

-Fragmentos Interactivos (Yandex Islands) de Yandex. Parecido a los enriquecidos pero admite interacción (Fig. 7).

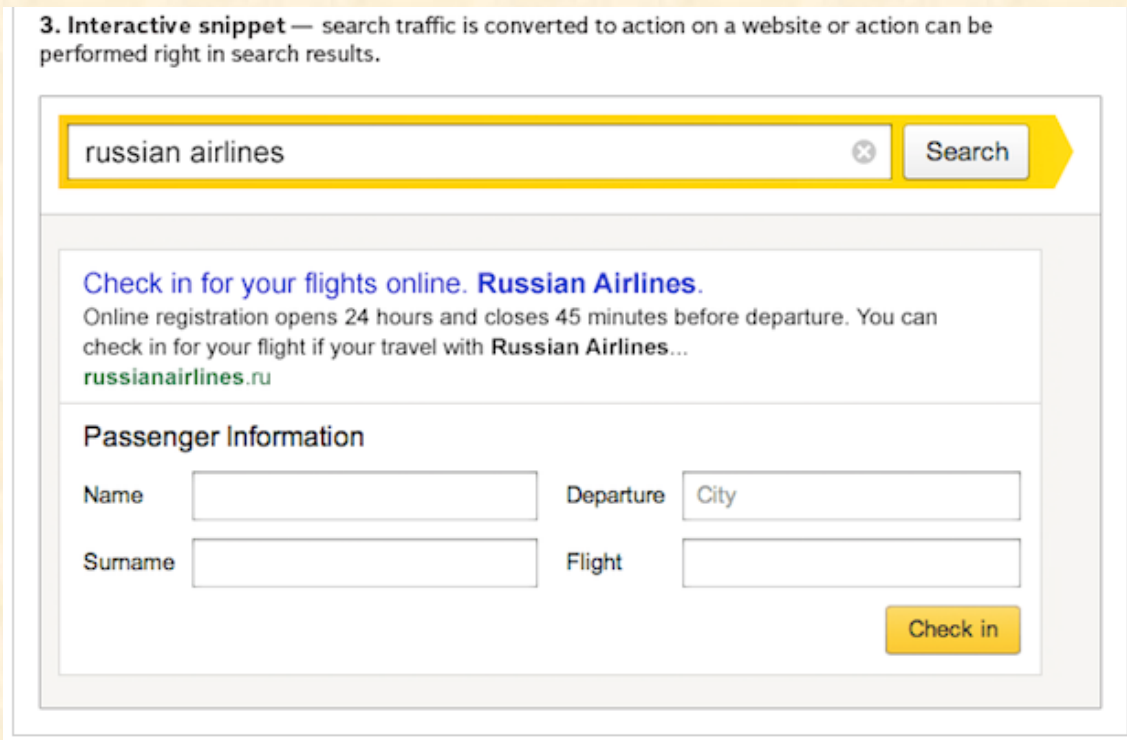


Fig. 7. Búsqueda interactiva de Yandex que permite registrarse en un vuelo. searchenginewatch.com

-Pins enriquecidos de Pinterest. Utiliza Schema para *pins* enriquecidos para recetas, películas, artículos, productos o lugares.

-Correo electrónico. Para recibos de compra, confirmación de reservas en eventos, restaurantes, hoteles o líneas aéreas. El marcado de Schema embebido hace posible a las herramientas de correo electrónico acceder a los datos estructurados y los hace disponibles mediante notificaciones o recordatorios de móvil, mapas o calendarios.

Aplicaciones que actúan como asistentes digitales personales:

-Google Now, mediante *Confirmation Cards* ofrece información actualizada a los usuarios sobre reservas de eventos, restaurantes, hoteles y vuelos. Esta información es obtenida del marcado de los mensajes enviados a Gmail por los proveedores de esos servicios.

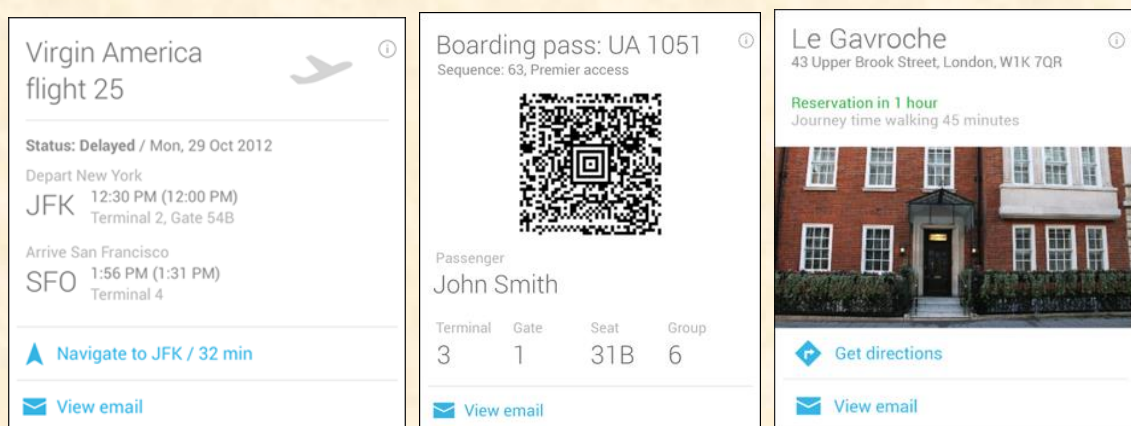


Fig. 8. Ejemplos de Confirmation Cards de Google Now. <https://developers.google.com/schemas/now/cards>.

-Cortana de Microsoft (para Windows 10 y teléfonos Windows) obtiene información de Schema enviada por las líneas aéreas al correo electrónico para trazar vuelos (Fig. 9).

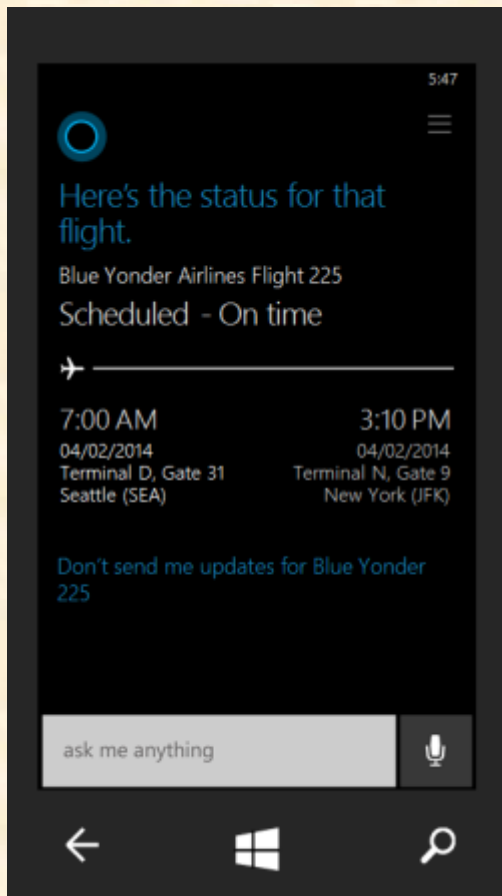


Fig. 9. Información de Cortana sobre estado del vuelo. msdn.microsoft.com.

-Searchlight/Siri del iOS 9 de Apple. Lo utiliza para búsquedas de tipo puntuaciones agregadas, ofertas, productos, precios, organizaciones, imágenes, números de teléfono y acciones de búsqueda web. También junto a RSS para marcado de noticias.

ESTADÍSTICAS DE ADOPCIÓN

El 30% de las páginas web contienen datos embebidos en HTML (Webdata Commons 2015).

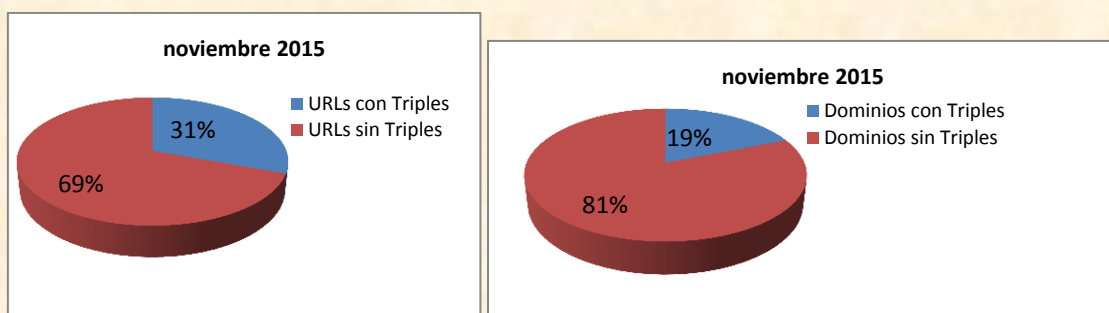


Fig. 10. URLs y Dominios con y sin triples (sujeto-predicado-objeto). Web Data Commons.

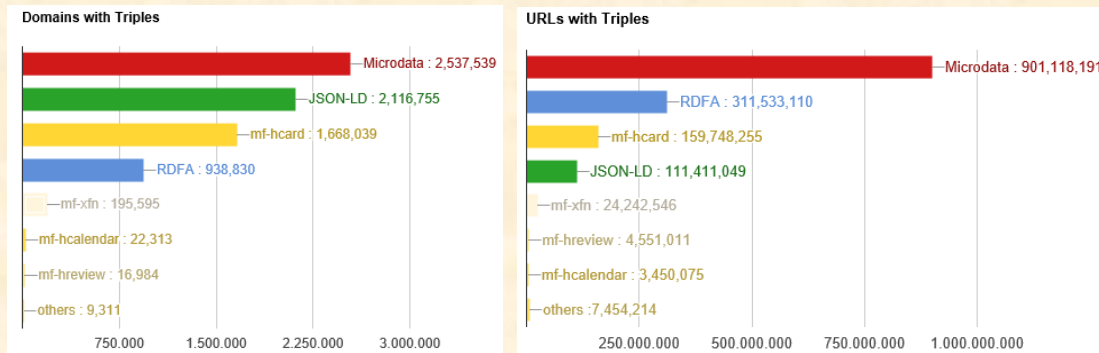


Fig. 11. Porcentaje de las distintas sintaxis en Dominios y URLs en Octubre de 2016. [Web Data Commons](#).

El nivel de adopción por los desarrolladores web, medidos en una muestra de una combinación del índice de Google y Web Data Commons de diez mil millones de páginas en 2014 fue del 22% y del 31.3% en 2015. Al menos 12 millones de sitios usan el marcado de Schema.org (60% de los mejores sitios de comercio electrónico, el 86% de los máximos sitios web de viajes y el 70% de los mayores portales de empleo) (5). En la Fig. 10 observamos URLs y Dominios con y sin triples (sujeto-predicado-objeto) y en la Fig. 11, el porcentaje de las distintas sintaxis en URLs y Dominios. En la Fig. 12 vemos el ascenso comparativo de Schema.org.

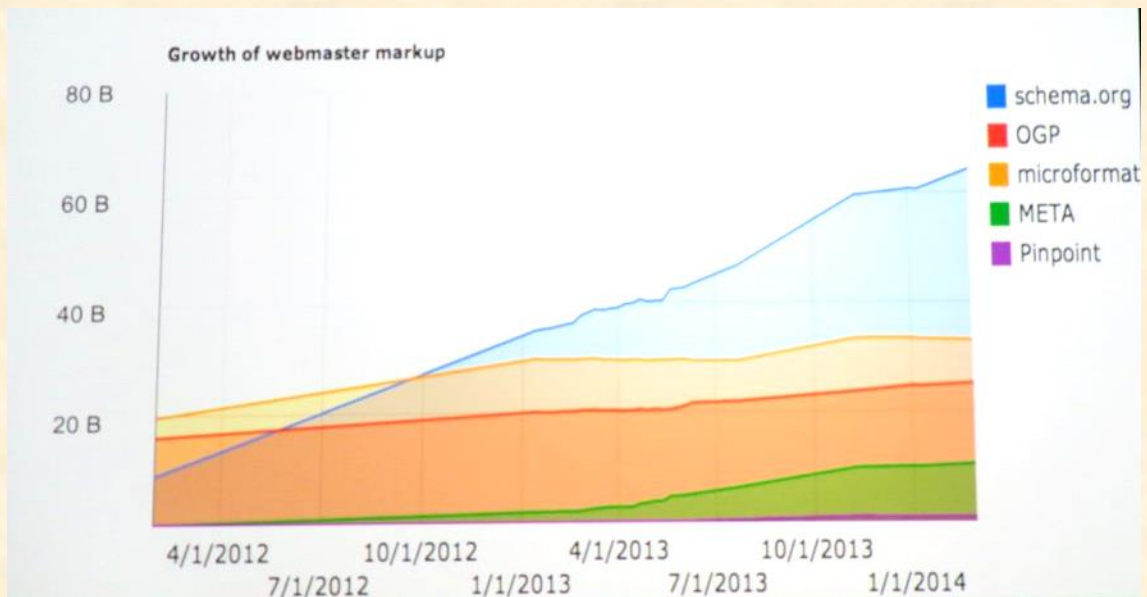


Fig. 12. Estadística comparativa de Schema.org. R.V. Guha (Google) [pbs.twimg.com](#)

3.1. Modelo de datos

“El modelo de datos utilizado es muy genérico y derivado de RDF Schema (que a su vez se deriva de CYCL)”. (6)

3.1.1. Clases (Tipos)

-Comienzan con letra mayúscula (Person, Organization, CreativeWork,...).

-Cada clase tiene una etiqueta y descripciones.

-Forman una jerarquía de clases donde se permite la herencia múltiple (una clase con dos superclases) pero es rara. (Fig. 13).

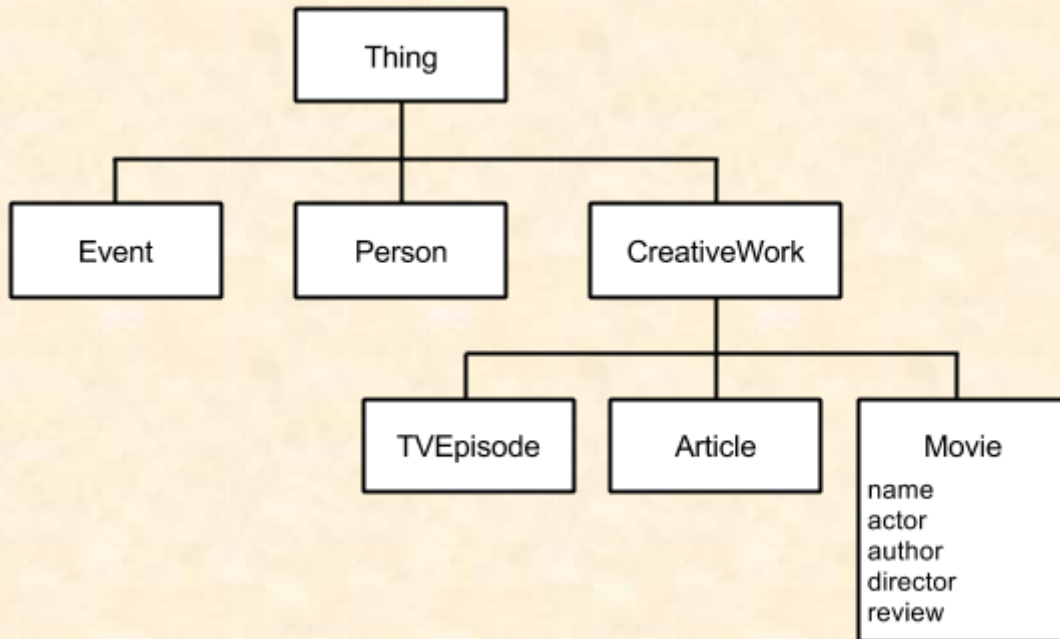


Fig. 13. Jerarquía de clases. support.google.com

3.1.2. Propiedades

-Comienzan con letra minúscula.

-Tienen etiqueta y descripción.

-Tienen dominios y rangos, y propiedades inversas. Las relaciones son polimórficas: tienen uno o más dominios y uno o más rangos.

3.1.3. Tipos de datos

Boolean (True o False), Date (formato de fecha ISO 8601), DateTime (fecha y hora), Number (Número), Text (Texto), Time (hora que se repite múltiples días).

El éxito de Schema radica en haber conseguido estandarizar y simplificar el proceso de marcar el contenido web, a la vez que proporcionar beneficios por ello, como son los fragmentos enriquecidos.

Los fragmentos enriquecidos (del inglés, rich snippets) son la información destacada que aparece bajo los resultados de búsqueda, como puede ser el precio de un producto, estrellas de valoración y opiniones de los clientes.

Es un tipo de información adicional que hace que el resultado sea más destacado en las búsquedas (Search Engine Optimization), lo que ayuda a captar la atención de los usuarios y recibir más clics.

Google muestra fragmentos enriquecidos para productos, recetas, reseñas, eventos, apps, videos y artículos. Para optar a ellos, el buscador necesita entender el contenido de la página, algo que consigue gracias al uso combinado de datos estructurados y Schema.org.

Para conseguir fragmentos enriquecidos hay que implementar datos estructurados que describan el contenido web mediante marcado o etiquetado semántico.

3.1.4. Extensión

Schema.org proporciona un núcleo, un vocabulario básico que describe las entidades más comunes. Cuando se necesitan vocabularios más específicos, los mecanismos de extensión facilitan su creación. (7)

Hay dos tipos de extensiones: Revisadas/Alojadas y Externas. Ambos tipos de extensiones suelen añadir subclases y propiedades al núcleo.

Extensiones Externas

Son extensiones independientes no aprobadas oficialmente. Aquí se muestran algunos ejemplos: [GS1 Web Vocabulary](#).

Extensiones Alojadas

Están administradas y publicadas como parte del proyecto schema.org y pueden ser usadas junto con los términos del núcleo. Existen varias extensiones alojadas:

- [auto.schema.org](#): Términos relativos a los automóviles.
- [bib.schema.org](#): Relativo a libros y bibliografía.
- [health-lifesci.schema.org](#): Relacionado con la salud, la medicina y las ciencias de la vida.
- [iot.schema.org](#): Exploración de Schema.org para IoT (*Internet of Things*)
- [meta.schema.org](#): Términos utilizados en schema.org para las definiciones de esquema.
- [pending.schema.org](#): Son términos pendientes de una revisión más amplia.
- [attic.schema.org](#): Términos obsoletos del núcleo o de las extensiones, eliminado de los pendientes o no aceptados en el vocabulario completo. (2)

Uso de las extensiones

Por ejemplo, para usar *MotorizedBicycle*, tipo definido en la extensión [auto.schema.org](#) y documentada en la página <http://auto.schema.org/MotorizedBicycle>, usaremos la URI canónica <http://schema.org/MotorizedBicycle>.

```
<div itemscope itemtype = "http://schema.org/MotorizedBicycle">
```

(7)

3.2. SINTAXIS

Para Schema.org se consideró más apropiado admitir múltiples sintaxis. Se acepta RDFa (Resource Description Framework in Attributes) y JSON-LD (JavaScript Object Notation for Linked Data). Debido a la inicial complejidad de RDFa 1.0 se introdujo Microdatos, que fue desarrollado como parte de HTML5, y fue diseñado como un subconjunto simple de Microformatos y de RDFa, pero rompiendo la compatibilidad con RDFa. Posteriores revisiones han hecho a RDFa menos complejo. (4)

RDFa Lite

-También es posible utilizar RDFa Lite, que al igual que Microdatos, es un subconjunto de RDFa que posee la simplicidad de los Microdatos pero añade la extensibilidad y compatibilidad con RDFa.

-W3C declaró a RDFa Lite como alternativa estándar única y definitiva a Microdatos.

-Casi todo el marcado de Microdatos puede ser expresado directamente mediante RDFa Lite.

Comparación de RDFa / RDFa Lite con Microdata:		
	RDFa / RDFa Lite	Microdata
Admitido por los mayores buscadores y por Facebook	Sí	No aceptado por Facebook
Lenguajes objetivo	(XHTML1, HTML 4, HTML5, XHTML5, XML, SVG, ePub, OpenDocument)	(HTML5, XHTML5)
Atributos nuevos	about, datatype, profile, prefix, property, resource, typeof, vocab	itemid, itemprop, itemref, itemscope, itemtype
Atributos reutilizados	content, href, rel, rev, src	content, src, href, data, datetime
Mezclar vocabularios	Se puede usar schema.org y Open Graph Protocol (OGP) de Facebook con sólo un lenguaje de marcado	Muy complicado
Invertir la dirección de la relación de una propiedad	Sí	Muy complicado

Tabla 1. Comparación de RDFa / RDFa Lite con Microdata (8)

Ejemplo:

```
<div>
  
  Dell UltraSharp 30" LCD Monitor
</div>
```

Microdata 1.0	RDFa Lite 1.1
<pre><div itemscope itemtype="http://schema.org/Product"> Dell UltraSharp 30" LCD Monitor </div></pre>	<pre><div vocab="http://schema.org/" typeof="Product"> Dell UltraSharp 30" LCD Monitor </div></pre>

Las similitudes son evidentes y la traducción es directa. (9)

Métodos para implementar datos estructurados

Hay dos formas de llevarlo a cabo:

-Intercalando descripciones en el contenido (Microdatos y RDFa) Tanto los Microdatos como RDFa son una serie de etiquetas HTML (no confundir con meta etiquetas o *meta tags* que se suelen usar para resumir el contenido de la página para buscadores y navegadores web e indicaciones a los robots cómo se desea que traten la página) que asocian el contenido de la página con el vocabulario de Schema.org. Microdatos y RDFa funcionan mejor en sitios generados con plantillas del lado del servidor.

-Añadiendo un bloque de código con todas las descripciones (JSON-LD) En lugar de intercalar etiquetas HTML entre el contenido, JSON-LD incrusta la descripción en un fragmento de código separado (*script*). De este modo la implementación es más fácil, ya que el código puede situarse en bloque sobre cualquier parte de la página. Es apropiado para sitios que son generados usando *JavaScript* del lado del cliente y también en correos electrónicos personalizados dónde las estructuras de datos pueden ser más prolijas. Sistemas de gestión de contenidos para eventos (como conciertos) que proporcionan *widgets* que están incrustados en otros sitios. JSON-LD permite que estos *widgets* integrados lleven datos estructurados en Schema.org.

A continuación veremos cada uno de estos métodos:

3.2.1. Microdatos

Es una especificación WHATWG HTML utilizada para anidar metadatos. Utiliza un vocabulario como soporte para describir un ítem y pares nombre-valor para asignar valores a sus propiedades. Surge como un intento de conseguir una forma más sencilla que RDFa y Microformatos de marcar elementos HTML con etiquetas legibles para las máquinas (10).

Atributos globales	
itemscope	Crea el elemento (ítem) cuyos descendientes contienen información sobre él.
itemtype	URL del vocabulario que describe el ítem y sus propiedades
itemid	Identificador único del elemento,
itemprop	Nombre y valor de contexto de la propiedad descrita por el vocabulario. El nombre de la propiedad y el contexto de valor son descritos por vocabulario del elemento. Los valores de las propiedades generalmente consisten en valores de cadena, pero también pueden utilizar direcciones URL utilizando el elemento <code>a</code> y su atributo <code>href</code> , el elemento <code>img</code> y su atributo <code>src</code> u otros elementos que vinculan o incrustan recursos externos.
itemref	Provee una lista de elementos de referencia con propiedades adicionales

Tabla 2. Atributos de Microdatos.

3.2.2. RDFa (Resource Description Framework in Attributes)

La esencia de RDFa (Marco de descripción de recursos en atributos) es proporcionar un conjunto de atributos que pueden usarse para transportar metadatos en lenguaje XML (por lo tanto, la 'a' en RDFa).

Estos atributos son:

vocab: indica el IRI (Internationalized Resource Identifier) del vocabulario por defecto, en nuestro caso es schema.org
typeof: indica de que tipo es la instancia descrita.
about: una URI que indica el recurso que describen los metadatos y que remite al documento actual por defecto
rel, rev, href y resource: atributos que establecen un relación o relación inversa con otro recurso
property: aporta una propiedad para el contenido de un elemento
content: atributo opcional que se sobrepone al contenido del elemento cuando se usa el atributo property
datatype: atributo opcional que indica el tipo de datos del contenido

Comparando los atributos de RDFa y Microdata:

Microdata 1.0	RDFa Lite 1.1	Uso
Itemid	resource	Identificar exactamente lo que se está describiendo mediante una dirección URL, como una determinada persona, evento o lugar.
Itemprop	property	Identifica una propiedad de la cosa descrita, como nombre, fecha o localización.
Itemscope	no se necesita	Señala una nueva cosa descrita.
Itemtype	typeof	Identifica el tipo de cosa descrita como persona, evento o lugar.
Itemref	no se necesita	Permite copiar y pegar un fragmento de datos y asociarlo con varias cosas.
no soportado	vocab	Se utiliza para especificar un vocabulario predeterminado que contiene términos que son utilizados para marcado.
no soportado	prefix	Utilizado para mezclar diferentes vocabularios en el mismo documento, como los proporcionados por <i>Facebook</i> , <i>Google</i> y proyectos <i>open source</i> .

1. itemprop se sustituye con property.
2. itemscope se omite.
3. itemtype se sustituye con typeof.
4. itemid se sustituye con resource.

Además, el valor del atributo vocab = "http://schema.org/" se añade a la etiqueta *body* o a alguna otra etiqueta.

3.2.3. JSON-LD JavaScript Object Notation for Linked Data

En lugar de intercalar etiquetas HTML entre el contenido, [JSON-LD](#) incrusta la descripción en un fragmento de código separado (*script*). De este modo la implementación es más fácil, ya que el código puede situarse en bloque sobre cualquier parte de la página. El método es más sencillo de implementar y queda más claro lo que describen las etiquetas (*name, image, url*).

El mayor problema de JSON-LD es que obliga a repetir todo el contenido que va a marcarse. Esto implica que si son muchos los datos a marcar, la página crecerá y la velocidad de carga podría resentirse. (11)

3.2.3.a. Elementos Sintácticos y Palabras clave

JSON-LD especifica un número de elementos de sintaxis y **palabras clave** que son una parte fundamental del lenguaje:

@context	Se utiliza para definir los nombres que se utilizan en un documento JSON-LD. Estos nombres se llaman términos y ayudan a expresar los identificadores específicos de una manera compacta.
@id	Se utiliza para identificar de forma única <i>las cosas</i> que se están descritas en el documento con los IRI o identificadores de nodos en blanco .
@value	Se utiliza para especificar los datos que se asocian con una determinada propiedad en el grafo.
@language	Se utiliza para especificar el idioma para un valor de cadena en particular o el idioma predeterminado de un documento JSON-LD.
@type	Define el tipo de datos de un nodo o valor .
@container	Define el tipo de contenedor predeterminado para un término .
@list	Expresa un conjunto ordenado de datos.
@set	Expresa un conjunto desordenado de datos y para asegurar que los valores siempre se representan como matrices.
@reverse	Se usa para expresar propiedades inversas.
@index	Se utiliza para especificar que un contenedor se utilice para indexar información y que el procesamiento deba continuar más profundamente en una estructura de datos JSON.
@base	Se utiliza para establecer la base IRI a partir de la cual los IRIs relativos se resuelven.
@vocab	Se utiliza para ampliar propiedades y valores en @type con un prefijo común IRI .
@graph	Se usa para expresar un grafo .
:	Separador de claves y valores JSON que usan IRIs compactos .

Tabla 3. Elementos de sintaxis y palabras clave <https://www.w3.org/TR/json-ld/>

En todas las claves, palabras clave y valores en JSON-LD se distingue entre mayúsculas y minúsculas.

3.3. Ejemplo

3.3.1. Sin marcado

```
<p>Christopher Froome fue patrocinado por
<a href="http://www.skysports.com/">Sky</a>
en el Tour de Francia.
</p>
```

3.3.2. Marcado

Para proceder al marcado debemos identificar todos los elementos (Tipos y propiedades):

-Christopher Froome: nombre (propiedad "name") de una Persona (Tipo Person).

-Sky: Es el nombre (propiedad "name") de su patrocinador (propiedad "sponsor") y es una Organización (Tipo Organization) con URL (propiedad "url") <http://www.skysports.com/>.

3.3.2.a. Microdata

```
<p itemscope itemtype="http://schema.org/Person">
  <span itemprop="name">Christopher Froome</span> fue patrocinado por
  <span itemprop="sponsor" itemtype="http://schema.org/Organization">
    <a itemprop="url" href="http://www.skysports.com/">Sky</a>
  </span> en el Tour de Francia.
</p>
```

3.3.2.b. RDFa

```
<p vocab="http://schema.org/" typeof="Person">
  <span property="name">Christopher Froome</span> fue patrocinado por
```

```
<span property="sponsor" typeof="http://schema.org/Organization">
  <a property="url" href="http://www.skysports.com/">Sky</a>
</span> en el Tour de Francia.
```

```
</p>
```

3.3.2.c. JSON-LD

```
<script type="application/ld+json">
{
  "@context": "http://schema.org/",
  "@type": "Person",
  "name": "Christopher Froome",
  "sponsor":
  {
    "@type": "Organization",
    "name": "Sky",
    "url": "http://www.skysports.com/"
  }
}
</script>
```

3.4. FORMAS DE MARCADO DE DATOS ESTRUCTURADOS

3.4.1. MANUAL

La forma más simple pero tediosa de marcado consiste en utilizar un esquema para los tipos que necesitemos y rellenarlo con los datos deseados, copiando el código en la página web. Se pueden conseguir esquemas y ejemplos en: [Schema.org Full Hierarchy](#), [getSchema](#), [Builtvisible Microdata](#), [JSON-LD & Schema.org](#) y [Google Structured Data Guide](#) (11).

3.4.2. MEDIANTE HERRAMIENTAS

Otra forma más sencilla que a mano, consiste en rellenar un formulario con los datos a marcar para obtener el código que copiaremos en nuestra web:

[Asistente de marcado de datos estructurados de Google](#): introduciendo URL o código HTML para marcado en Microdatos y JSON-LD de sitios web (Apps de software, Artículos, Empresas locales, Episodios de TV (con valoración), Eventos, Películas, Productos, Reseñas de libros, Restaurantes) y correo electrónico (Envío de paquetes, Pedidos, Reservas de alojamiento, alquiler de coche, autobús, eventos, restaurantes, tren y vuelos)

[Raven Schema Creator](#) (de pago), **[Micro Data Generator](#)**, **[Sistrix Video Schema Markup Generator](#)**, **[Schema-generators \(SEO Chat\)](#)**, **[Schema Generator \(IT Solutions\)](#)**.

3.4.3. UTILIZANDO PLUGINS

Pueden realizar el marcado de forma automática o mediante un formulario.

WordPress	Schema Creator by Raven – La misma funcionalidad de la herramienta online de Raven integrada en WordPress. Permite añadir Schema.org y microdatos a entradas y páginas.
-----------	--

	<p>All in One Schema Rich Snippets – Inserta los datos estructurados necesarios para optar a los principales tipos de fragmentos enriquecidos en buscadores, además de añadir los meta datos Open Graph de Facebook.</p> <p>Yoast SEO – Añade Schema.org a diversos elementos, en concreto, el marcado RDFa a los <i>breadcrumbs</i> y JSON+LD al buscador interno. También inserta los metadatos sociales de Facebook, Twitter y Google+. Los plugins de pago Video SEO, Local SEO y WooCommerce SEO incorporan marcado adicional.</p> <p>WordLift – Permite organizar contenidos mediante conceptos o <i>entidades</i> siguiendo las especificaciones de Schema.org.</p>
Joomla	<p>J4Schema – Incorpora un editor visual de atributos para Schema.org. Tras seleccionar un texto permite añadir los atributos oportunos. Se integra con Virtuemart y K2.</p> <p>Rich Snippets para jEvents – Gratis. Añade nombre/descripción de eventos para búsqueda.</p>
Drupal	<p>Schema.org – Permite asociar contenido a diferentes esquemas de Schema.org, así como especificar los atributos correspondientes.</p>
Magento	<p>Schema.org – De pago. Añade metadatos (nombre, descripción, precio, valoraciones) a una página de productos.</p> <p>Gmail Actions in the Inbox - Schema.org – Gratis. Marcado para correo electrónico saliente en Gmail.</p> <p>Google Rich Cards – Gratis. Mejora la apariencia del sitio web en las búsquedas orgánicas de móvil mediante Cartas Enriquecidas (<i>Rich Cards</i>).</p> <p>SEO Rich Snippets - Google, Bing, Yahoo - schema.org – De pago. Para Fragmentos Enriquecidos (<i>Rich Snippets</i>), <i>Breadcrumbs</i> de navegación, Valoraciones, Productos (nombre, imagen, descripción, categoría,...).</p>

4. OTROS VOCABULARIOS

4.1. FOAF (Friend of a Friend)

El vocabulario Amigo de un Amigo surgió a principios del año 2000 y es un vocabulario adecuado para definir metadatos sobre las personas y sus intereses, relaciones y actividades. El vocabulario tiene un conjunto básico de clases (primera letra mayúscula) y propiedades (primera letra en minúscula). A continuación, vemos una tabla de sus componentes, un gráfico de sus clases y un ejemplo:

FOAF Basics	<p>name, nick, title, homepage, mbox, mbox_sha1sum, img, depiction (depicts), surname, family_name, givenname, firstName,</p>
-------------	---

Agent	<i>Agente</i> - Un agente (Ej.: persona, grupo, software o artefacto físico). foaf:Agent clase es la clase de agentes; cosas que hacer cosas. Una subclase es foaf:Person , que representa la gente. Otros tipos de agentes incluyen foaf:Organization y foaf:Group .
Person	Una persona. La clase foaf:Person representa la gente. Algo es un foaf:Person si se trata de una persona. Puede estar viva, muerta, ser real o imaginaria. foaf:Person es una subclase de foaf:Agent , ya que todas las personas son consideradas "agentes" en FOAF.
Personal Info	weblog , knows , interest , currentProject , pastProject , plan , based_near , workplaceHomepage , workInfoHomepage , schoolHomepage , topic_interest , publications , geekcode , myersBriggs , dnaChecksum
Online Accounts / IM	holdsAccount , accountServiceHomepage , accountName , icqChatID , msnChatID , aimChatID , jabberID , yahooChatID
OnlineAccount	Una foaf: OnlineAccount representa el establecimiento de algún tipo de servicio en línea, por alguien (indicado indirectamente a través de un foaf: accountServiceHomepage) con algún foaf:Agente . La propiedad foaf:holdsAccount del agente se usa para indicar las cuentas que están asociadas con el agente.
OnlineChatAccount	Una foaf: OnlineChatAccount es una foaf: OnlineAccount dedicada a chat de mensajería / instantánea. Esta es una generalización de las propiedades de identificación FOAF de Chat, foaf: jabberId , foaf: aimChatID , foaf: msnChatID , foaf: icqChatID y foaf: yahooChatID .
OnlineEcommerceAccount	Una foaf: OnlineEcommerceAccount es una foaf: OnlineAccount dedicado a la compra y/o venta de bienes, servicios, etc. Los ejemplos incluyen Amazon , eBay , PayPal , thinkgeek , etc.
OnlineGamingAccount	Una foaf: OnlineGamingAccount es una foaf: OnlineAccount dedicado a los juegos en línea. Los ejemplos pueden incluir EverQuest , Xbox Live , Neverwinter Nights , etc., así como los sistemas antiguos basados en texto (Moos, MUD y cosas por el estilo).
Projects and Groups	member , membershipClass , fundedBy , theme
Project	Un proyecto (una tarea colectiva de algún tipo). La clase foaf:Project representa la clase de cosas que son "proyectos". Estos pueden ser formales o informales, colectivos o individuales. A menudo es útil para indicar la foaf:homepage de un foaf:Project .
Organization	Una organización. foaf:Organization representa un tipo de foaf:Agent correspondiente a instituciones sociales tales como empresas, sociedades, etc.
Group	Una clase de agentes. foaf:Group representa una colección de agentes individuales (y puede en sí desempeñar el papel de un foaf:Agent , es decir, algo que puede llevar a cabo acciones.).
Documents and Images	topic (page), primaryTopic , tipjar , sha1 , made (maker), thumbnail , logo
Document	La clase foaf:Document representa lo que consideramos «documentos».
Image	La clase foaf:Image es una subclase de foaf:Document correspondiente a esos documentos que son imágenes. Las imágenes digitales (tales como JPEG, PNG, GIF mapas de bits, diagramas SVG etc.) son ejemplos de foaf:Image .

<p>PersonalProfileDocument</p>	<p>Un documento RDF de perfil personal.</p> <p>La clase foaf: PersonalProfileDocument representa las cosas que son un foaf: Document, y que hace uso de RDF para describir las propiedades de la persona que es el foaf: maker (creador) del documento. Sólo hay una foaf: Person que se describe en el documento, es decir, la persona que lo foaf: made (hizo) y qué será su foaf: primaryTopic (tema principal)</p>
--------------------------------	--

Tabla 4. Vocabulario FOAF (12)

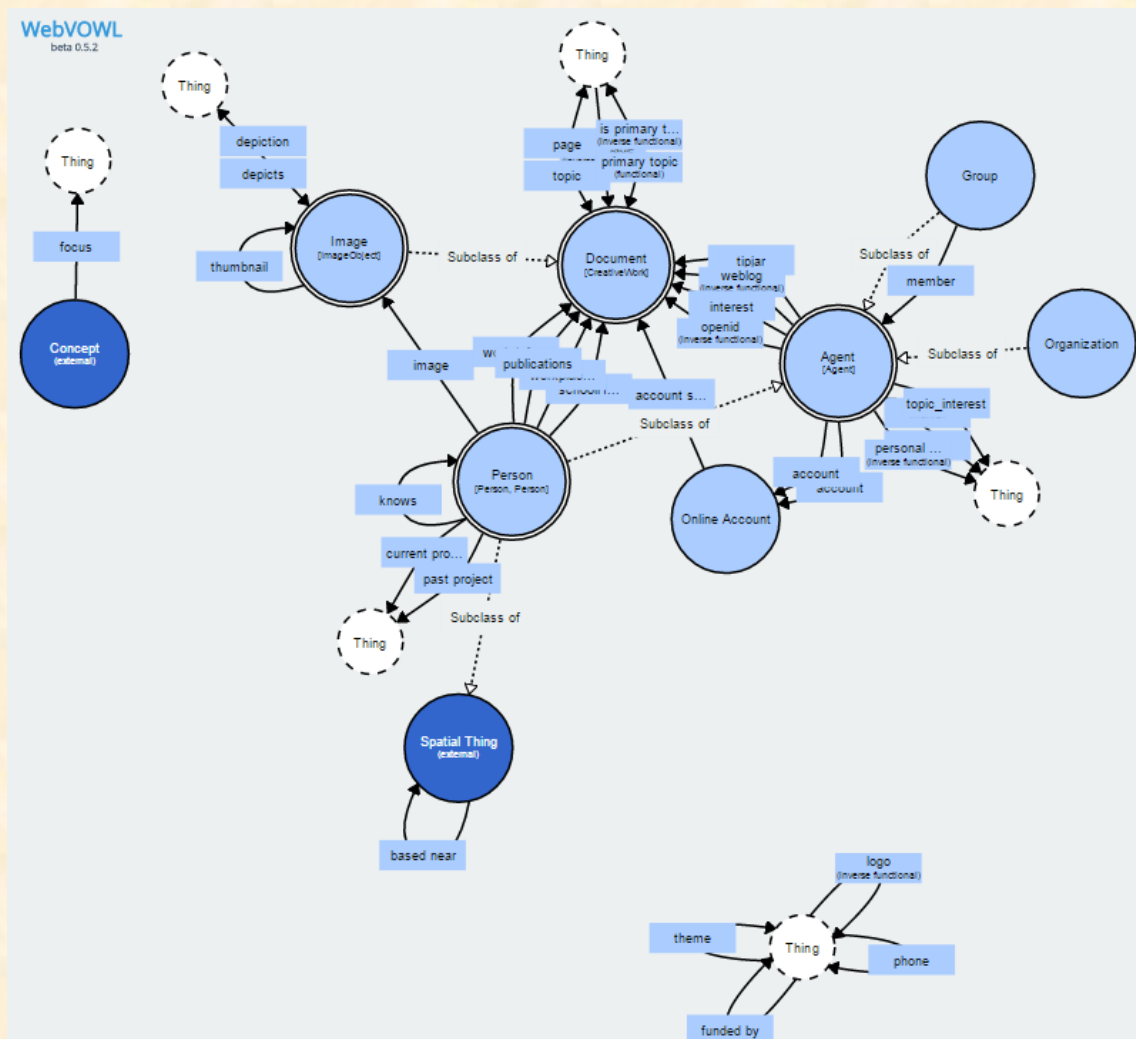


Fig. 14. FOAF mediante WebVOWL. <http://vowl.visualdataweb.org/webvowl/#foaf>

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <foaf:Person>
    <foaf:name>Jimmy Wales</foaf:name>
  
```

```

<foaf:mbox rdf:resource="mailto:jwales@bomis.com" />
<foaf:homepage rdf:resource="http://www.jimmywales.com/" />
<foaf:nick>Jimbo</foaf:nick>
<foaf:depiction
rdf:resource="http://www.jimmywales.com/aus_img_small.jpg" />
<foaf:interest>
  <rdf:Description rdf:about="http://www.wikimedia.org"
rdfs:label="Wikipedia" />
</foaf:interest>
<foaf:knows>
  <foaf:Person>
    <foaf:name>Angela Beesley</foaf:name>
  </foaf:Person>
</foaf:knows>
</foaf:Person>
</rdf:RDF>

```

Tabla 5. Un ejemplo sencillo de FOAF serializado en RDF/XML. (13)

4.2. DUBLIN CORE

“Es un vocabulario adecuado para la descripción de recursos (documentos), habiendo comenzado como un conjunto básico de 15 propiedades basados en una idea análoga a la de los elementos de catalogación de bibliotecas” (14). A continuación, vemos una comparativa con Schema, una tabla de sus componentes y un ejemplo:

Dublin Core:	Schema.org:
Se originó en un taller de OCLC y NCSA en Dublin, Ohio, en 1995	Reciente, fue fundada en 2011
Fue hecho a propósito con base amplia	Fue la creación de los principales proveedores de motores de búsqueda
Es un esquema de metadatos descriptivos inspirado en un sistema que fue utilizado en bibliotecas llamado MARC (MACHINE-Readable Cataloging).	Es una colección de esquemas de metadatos que se centran en SEO (Search Engine Optimization u optimización en buscadores) y se dirige especialmente a los webmasters
Se utiliza en muchos dominios, idiomas y modelos de negocio	Es prevalente en Web
Se basa en gran medida en sus calificadores para ampliar su vocabulario	Es la elección número uno para el SEO
Es especialmente favorable por las descripciones de colecciones físicas	Sobresale con los sitios web generados
Es interoperable con la nube de Datos Vinculados y la Web semántica	Destaca en el tratamiento adecuado de cada tipo de datos
A menudo es confiable, ya que está bien establecida y tiene mucho respaldo	
Es utilizado por muchas aplicaciones destacadas y es especialmente eficaz para las redes sociales	

Tabla 6. Comparación Dublin Core/Schema.org. Fuente: seopressor.com

DUBLIN CORE	
Etiqueta	Contenido:
DC.Title	Título: el nombre dado a un recurso, habitualmente por el autor
DC.Subject	Claves: los temas del recurso Típicamente, Subject expresará las claves o frases que describen el título o el contenido del recurso Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales
DC.Description	Descripción: una descripción textual del recurso Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual
DC.Source	Fuente: secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual
DC.Type	Tipo del Recurso: la categoría del recurso Por ejemplo, página personal, romance, poema, diccionario, etc
DC.Relation	Relación: es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos
DC.Coverage	Cobertura: es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado (que ya lo encontramos en el elemento Date)
Propiedad Intelectual:	
DC.Creator	Autor o Creador: la persona u organización responsable de la creación del contenido intelectual del recurso Por ejemplo, los autores en el caso de documentos escritos; artistas, fotógrafos e ilustradores en el caso de recursos visuales
DC.Publisher	Editor: la entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual
DC.Contributor	Otros Colaboradores: una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento Creator(por ejemplo: editor, ilustrador y traductor)
DC.Rights	Derechos: son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso
Instanciación:	
DC.Date	Fecha: una fecha en la cual el recurso se puso a disposición del usuario en su forma actual Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la medida que el contenido intelectual está de alguna manera relacionado con aquella fecha
DC.Format	Formato: es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso
DC.Identifier	Identificador del Recurso: secuencia de caracteres

	utilizados para identificar unívocamente un recurso Ejemplos para recursos en línea pueden ser <u>URLs</u> y <u>URNs</u> Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo <u>ISBN</u> ("International Standard Book Number")
DC.Language	Lengua: lengua/s del contenido intelectual del recurso

Tabla 7. DUBLIN CORE (15)

```
<head>
<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/">
<meta name="DC.title" content="Guia de Web Semântica">
<meta name="DC.creator" content="Carlos Laufer">
<meta name="DC.subject" content="Web Semântica">
<meta name="DC.subject" content="Dados Conectados">
<meta name="DC.description" content="Introdução ao Ecosystema e às
Tecnologias de Web Semântica e Dados Conectados">
<meta name="DC.publisher" content="W3C Brasil"> <meta name="DC.date"
content="21/01/2015">
</head>
```

Tabla 8. Ejemplo Dublin Core. [Book-Semantic-Web-Guideline.pdf](#)

4.3. SKOS

SKOS (siglas de Simple Knowledge Organization System) es una iniciativa del [W3C](#) en forma de aplicación de [RDF](#) que proporciona un modelo para representar la estructura básica y el contenido de esquemas conceptuales como listas, [encabezamientos de materia](#), [taxonomías](#), esquemas de [clasificación](#), [tesauros](#) y cualquier tipo de vocabulario controlado. El origen del proyecto se remonta a la elaboración de un tesauro de actividades dentro del proyecto SWAD-Europe2. La primera versión de SKOS Core se presentó en el año 2003.

4.3.1. Categorías de elementos

Las principales categorías de elementos de SKOS son:

-Conceptos. El vocabulario de SKOS se basa en conceptos. Los conceptos son las unidades de pensamiento -ideas, significados, u objetos y eventos (instancias o categorías)- que subyacen en muchos sistemas de organización del conocimiento.

-Etiquetas, cadena de caracteres Unicode, opcionalmente con etiquetas de idioma, que se asocian con un concepto.

-Anotaciones, similar a etiqueta, pero la cadena literal tiene un tipo de datos, como entero, flotante, o fecha; incluso el tipo de datos puede ser compuesto.

-Documentación, proporciona información básica acerca de los conceptos de SKOS.

-Relaciones semánticas, proporciona formas de declarar las relaciones entre conceptos dentro del esquema conceptual.

-Propiedades de asignación, expresan correspondencia (exacta o difusa) entre conceptos de diferentes esquemas conceptuales.

-Las colecciones, son grupos de conceptos etiquetados y/u ordenados.

Los conceptos asociados se enumeran en la tabla siguiente:

Vocabulario SKOS organizado por temas					
Conceptos	Etiquetas y notación	Documentación	Relaciones semánticas	Propiedades de mapeo	Colecciones
Concept	prefLabel	note	broader	broadMatch	Collection
ConceptScheme	altLabel	changeNote	narrower	narrowMatch	orderedCollection
inScheme	hiddenLabel	definition	related	relatedMatch	member
hasTopConcept	notation	editorialNote	broaderTransitive	closeMatch	memberList
topConceptOf		example	narrowerTransitive	exactMatch	
		historyNote	semanticRelation	mappingRelation	
		scopeNote			

Tabla 9. Vocabulario SKOS organizado por temas. [Wikipedia](#)

```

<rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core">
  <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#tesauro">
    <skos:prefLabel xml:lang="es">Tesauro</skos:prefLabel>
    <skos:prefLabel xml:lang="en">Thesauri</skos:prefLabel>
    <skos:prefLabel xml:lang="fr">Thésaurus</skos:prefLabel>
    <skos:prefLabel xml:lang="ru">Тезаурысы</skos:prefLabel>
    <skos:scopeNote xml:lang="es">Lenguaje documental controlado y dinámico
      que contiene términos relacionados semántica y genéricamente
      que
      abarcan de manera exhaustiva una esfera concreta del
      conocimiento.
    </skos:scopeNote>
    <skos:altLabel xml:lang="es">Descriptores</skos:altLabel>
    <skos:altLabel xml:lang="es">Tesauro monolingüe</skos:altLabel>
    <skos:altLabel xml:lang="es">Tesauro multilingüe</skos:altLabel>
    <skos:altLabel xml:lang="es">Thesauri</skos:altLabel>
    <skos:broader rdf:resource="http://www.ejemplo.org/conceptos#lengindex"/>
    <skos:narrower rdf:resource="http://www.ejemplo.org/conceptos#comptes"/>
    <skos:related rdf:resource="http://www.ejemplo.org/conceptos#controlterm"/>
    <skos:related rdf:resource="http://www.ejemplo.org/conceptos#encmateria"/>
    <skos:related rdf:resource="http://www.ejemplo.org/conceptos#matref"/>
    <skos:related rdf:resource="http://www.ejemplo.org/conceptos#terminologia"/>
  </skos:Concept>
  <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#lengindex">
    <skos:prefLabel xml:lang="es">Lenguaje de indización</skos:prefLabel>
  </skos:Concept>

```

```

<skos:Concept rdf:about="http://www.ejemplo.org/conceptos#comptes">
  <skos:prefLabel xml:lang="es">Compilación de tesauro</skos:prefLabel>
</skos:Concept>
<skos:Concept rdf:about="http://www.ejemplo.org/conceptos#controlterm">
  <skos:prefLabel xml:lang="es">Control terminológico</skos:prefLabel>
</skos:Concept>
<skos:Concept rdf:about="http://www.ejemplo.org/conceptos#encmateria">
  <skos:prefLabel xml:lang="es">Encabezamiento de materia</skos:prefLabel>
</skos:Concept>
<skos:Concept rdf:about="http://www.ejemplo.org/conceptos#matref">
  <skos:prefLabel xml:lang="es">Material de referencia</skos:prefLabel>
</skos:Concept>
<skos:Concept rdf:about="http://www.ejemplo.org/conceptos#terminologia">
  <skos:prefLabel xml:lang="es">Terminología</skos:prefLabel>
</skos:Concept>
</rdf:RDF>

```

Fig. 15. Ejemplo SKOS. es.wikipedia.org/wiki/Simple_Knowledge_Organization_System

4.4. Open Graph protocol (OGP)

El protocolo Open Graph se utiliza para integrar cualquier página en un grafo social.

OGP se utiliza en páginas web para convertirlas en objetos enriquecidos del grafo social. A una web se le puede proporcionar en Facebook la misma funcionalidad que cualquiera de sus objetos.

4.4.1. Metadatos Básicos

Para convertir una web en un objeto grafo, debemos añadirle metadatos básicos. La versión inicial del protocolo está basada en RDFa, por lo que añadiremos las <meta> etiquetas (tags) en la cabecera (<head>) de la página web. Las 4 propiedades básicas son: **og:title** (título), **og:type** (tipo de objeto), **og:image** (URL de imagen) y **og:url** (URL canónica del objeto que será usada como su identificador permanente dentro del grafo).

Ejemplo de marcado con el protocolo Open Graph para la película [La Roca en IMDB](#):

```

<html prefix="og: http://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url"
content="http://www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="http://ia.media-
imdb.com/images/rock.jpg" />
...
</head>
...
</html>

```

4.4.2. Otros Metadatos de OGP

`og:audio` (Una URL de audio), `og:description` (Descripción del objeto), `og:determiner` (La palabra que aparece antes del título del objeto), `og:locale` (La configuración regional de las etiquetas), `og:locale:alternate` (Un vector con otras configuraciones regionales disponibles para la página), `og:site_name` (Nombre general del sitio web), `og:video` (URL de un archivo de video).

5. VENTAJAS, INCONVENIENTES Y RETOS DE SCHEMA.ORG

5.1. VENTAJAS

En general, **más datos buenos es mejor que menos datos buenos**, Los consumidores (ya sean humanos o robots) pueden analizar sus documentos y decidir por sí mismos cuáles de los datos estructurados les interesa, e ignorar el resto. Esa es la ventaja: cuanto más datos estructurados, posiblemente supondrá más y mejor uso para los consumidores (16). Sin embargo, como regla general, sólo se debe marcar el contenido visible para los visitantes de la web y no el contenido oculto (17).

Schema.org proporciona a los desarrolladores web (18):

-Muchas ventajas por añadir marcado de datos (mejor posicionamiento y más atractivos resultados de búsqueda)

-Permite a los autores avanzados añadir marcado más rico procedente de varios vocabularios distintos.

-Una forma fácil de llevar a cabo el marcado.

Los motores de búsqueda y muchas aplicaciones están utilizando marcado en páginas y correos electrónicos de una gran variedad de formas. Estos proyectos ayudan a que se refleje su contenido con mayor claridad o de manera más prominente en los resultados de búsqueda o en avisos y notificaciones de móvil. No todos los tipos de información en Schema.org aparecen en los resultados de búsqueda, pero con el tiempo se puede esperar que un mayor número de datos sean reflejados en ellas. Además, dado que el marcado de las páginas web y correos es accesible al público, se encontrarán nuevas e interesantes aplicaciones.

5.2. DESVENTAJAS

Los tamaños de archivo (y por lo tanto, los tiempos de carga) aumentan.

Hay que invertir más trabajo (para su aplicación, así como para su mantenimiento), por lo que es posible que haya que hacer un análisis de coste-beneficio. (16)

- Los sitios web construidos con AJAX son difíciles para los rastreadores. El contenido que utiliza un servicio web y se carga mientras la página se carga, debe realizar un esfuerzo significativo para crear instantáneas HTML (snapshots). Las instantáneas se proporcionan sólo para los rastreadores de búsqueda y esto sólo es factible para las empresas medianas o grandes que tienen desarrolladores que pueden pasar unas semanas haciendo esto.
- Los editores hacen el esfuerzo de suministrar los metadatos adicionales y sus principales consumidores, los motores de búsqueda no proporcionan ninguna referencia de los datos que han recogido.
- Los editores corren el riesgo de proporcionar datos tan buenos que los motores de búsqueda puedan responder directamente a las consultas de los usuarios sin enviar al usuario a su sitio web o sin atribuir la respuesta.
- Los cambios en Schema.org (en su mayoría extensiones, pero algunas cosas se vuelven obsoletas) y los editores tendrán que actualizar su marcado de vez en cuando
- Schema.org fue desarrollado por Google, Yahoo!, Bing y Yandex y la comunidad más amplia fue informada después del hecho. El desarrollo continuo de la norma sigue siendo un producto de estas empresas. Sin embargo, hay un proceso de solicitud formal para que se pueda influir en la dirección de las extensiones si se cuenta con un amplio apoyo de la industria.
- El mayor beneficio hoy en día son fragmentos enriquecidos y sólo algunos tipos de entidades los generan. (19)

5.3. RETOS

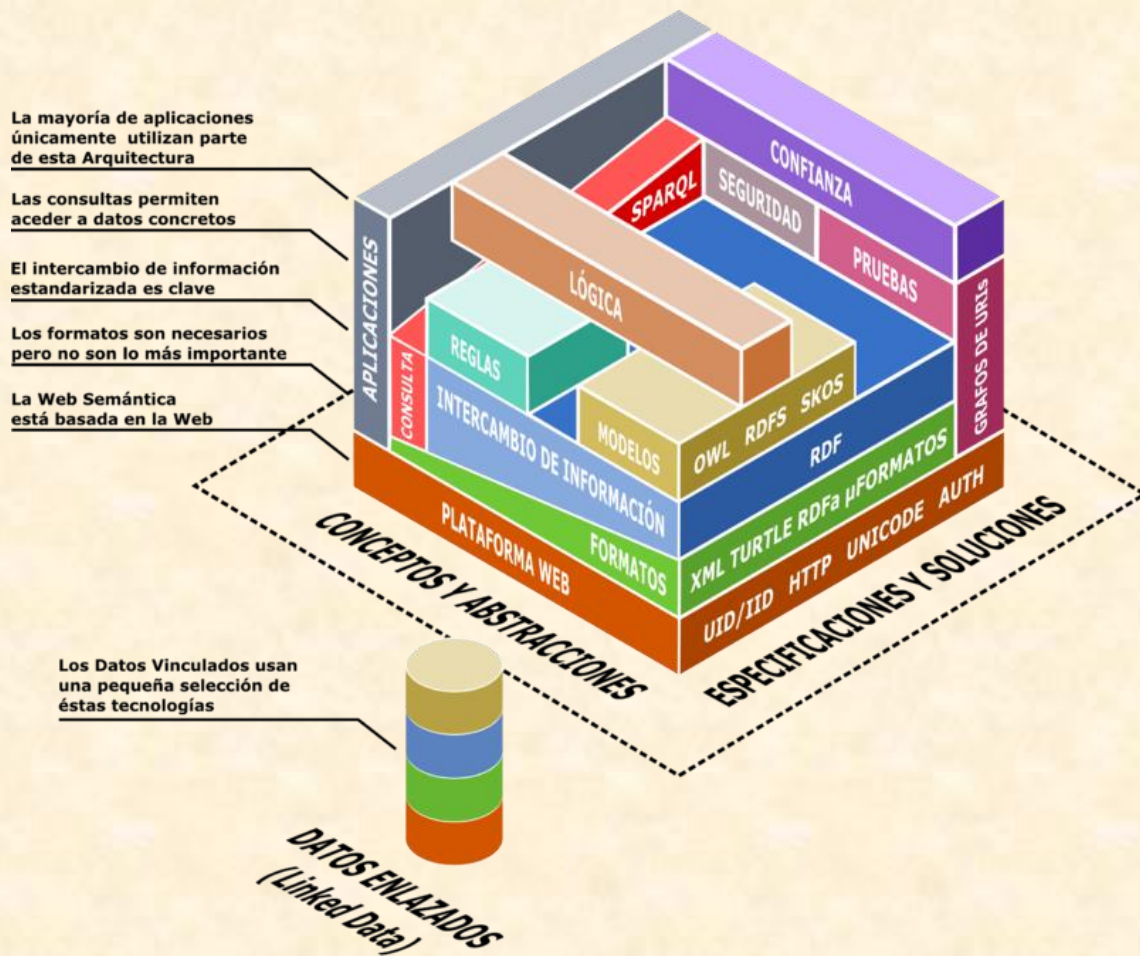


Fig. 16. Arquitectura tecnológica de la web semántica. <https://es.wikipedia.org>

Algunos de los retos a los que se enfrenta Schema.org que comparte con la Web Semántica (Fig. 16) incluyen amplitud, vaguedad, incertidumbre, inconsistencia y engaño. Los sistemas de razonamiento automatizado tendrán que lidiar con todos estos temas con la finalidad de cumplir con los objetivos de la Web Semántica que conduzca a una web inteligente (Fig. 17):

Retos	Descripción	Posibles soluciones
INMENSIDAD	Hay una enorme cantidad de webs y los grafos que se generan son inmensos. Por ahora, ninguna tecnología es capaz de eliminar términos semánticos duplicados.	Técnicas altamente escalables o modulares
IMPRECISIÓN	Conceptos imprecisos como: "joven" o "alto".	Lógica borrosa (Fuzzy)
INCERTIDUMBRE	Conceptos precisos pero con valores inciertos.	Técnicas de razonamiento probabilístico
INCONSISTENCIA	Contradicciones lógicas que inevitablemente surgen durante el desarrollo de grandes ontologías, y cuando las ontologías se combinan a partir de fuentes separadas. El razonamiento deductivo falla catastróficamente cuando se enfrentan a la incoherencia	razonamiento rebatible y paraconsistente
ENGAÑO	El productor de la información intenta engañar intencionadamente al consumidor	Técnicas de criptografía.

	de la información	Mediante verificación de hechos utilizando schema.org/ClaimReview
--	-------------------	--

Tabla 10. Retos a los que se enfrenta la web semántica. https://en.wikipedia.org/wiki/Semantic_Web



Fig. 17. Evolución Web: A medida que la cantidad de datos aumenta, la productividad de las búsquedas por palabra clave disminuye (gráfica roja). Las tecnologías semánticas ayudan a remontar la productividad afrontando el rápido crecimiento de la información (gráfica verde). web-evolution-nova-spivack-twine-3-728.jpg

6. Guía Práctica

Para marcar datos de una página web o un correo electrónico nos pueden ayudar herramientas como el Asistente para el marcado de datos estructurados de Google (<https://www.google.com/webmasters/markup-helper/>) en el que debemos marcar las propiedades y genera el código en Microdatos y JSON-LD, o Schema Generator (<http://schema.pythonanywhere.com/>) que permite rellenar un formulario y genera el código en Microdatos, RDFa y JSON-LD. Para probar que el marcado esté correcto se

puede utilizar la Herramienta de pruebas de datos estructurados de Google (<https://search.google.com/structured-data/testing-tool>).

Los Tipos y sus propiedades los consultamos en <http://schema.org/>, por ejemplo: el Tipo Película está en <http://schema.org/Movie>, el Tipo Producto en <http://schema.org/Product>, el Tipo Evento en <http://schema.org/Event>,...

Las propiedades pueden ser específicas o heredadas para cada tipo (clase), por ejemplo, para Película (Movie):



6.1. Marcado de una película (Movie)

6.1.1. HTML básico

```

<div>


<h1>Avatar</h1> (2009)
<div>
Puntuación 8/10
Usuarios: 354,477 votos
</div>
<span>Un marine parapléjico es enviado a una misión a la luna
Pandora...</span>
Director:
<a href="james-cameron.html">James Cameron</a>

Actores:
<a href="sam-worthington.html">Sam Worthington</a>,
<a href="zoe-saldana.html">Zoe Saldana</a> and
<a href="sigourney-weaver.html">Sigourney Weaver</a>

<a href="avatar-trailer.html">Watch Trailer</a>
</div>
(20)
  
```


6.1.2. Marcado mediante intercalado de Microdatos: itemscope e itemtype

Como regla general, delimitamos el ítem principal con *div*: `<div itemscope itemtype="http://schema.org/Tipo">...</div>`, los subítems `<div itemprop="propiedad" itemscope itemtype="http://schema.org/Tipo">...</div>` y las propiedades con *span*: `...`

```
<div itemscope itemtype="http://schema.org/Movie">
<!-- Marcamos el ítem como Película (Movie) -->



<h1>Avatar</h1> (2009)

<div>
Puntuación 8/10
Usuarios: 354,477 votos
</div>

<span>Un marine parapléjico es enviado a una misión a la luna
Pandora...</span>

Director:
<a href="james-cameron.html">James Cameron</a>

Actores:
<a href="sam-worthington.html">Sam Worthington</a>,
<a href="zoe-saldana.html">Zoe Saldana</a> and
<a href="sigourney-weaver.html">Sigourney Weaver</a>

<a href="avatar-trailer.html">Watch Trailer</a>
</div>
```

6.1.3. Añadimos itemprop

```
<div itemscope itemtype="http://schema.org/Movie">
<!-- Marcamos el tipo Película (Movie) -->


<!-- Marcamos la propiedad "imagen (image)" de la película -->

<h1 itemprop="name">Avatar</h1> (2009)
<!-- Marcamos la propiedad "nombre (name)" de la película -->

<meta itemprop="datePublished" content="2009-12-18">
<!-- fecha de publicación: RDFa y Microdatos permiten el uso del atributo
"content=" para publicar valores simples legibles por máquinas junto con
otros formatos más amigables para humanos.

Fecha/hora con formato ISO 8601 aaaa-mm-ddThh:mm. Ejemplos: 2015-07-27 o
2015-07-27T15:30-->

<div>
Puntuación 8/10
Usuarios: 354,477 votos
</div>
```

```
<span itemprop="description">Un marine parapléjico es enviado a una misión a la luna Pandora...</span>
```

```
<!-- Propiedad "descripción (description)" de la película -->
```

Director:

```
<div itemprop="director" itemscope itemtype="http://schema.org/Person">
  <a href="james-cameron.html" itemprop="url">
    <span itemprop="name">James Cameron</span>
  </a>
</div>
```

```
<!-- Propiedad "director" de Movie con url y nombre (name) de Persona (Person) -->
```

Actores:

```
<a href="sam-worthington.html" itemprop="actor">Sam Worthington</a>,
<a href="zoe-saldana.html" itemprop="actor">Zoe Saldana</a> and
<!-- Por simplicidad, Schema admite marcar de esta forma la propiedad "actor" de Movie, sobreentendiendo que es el url de una Persona (Person) -->
```

```
<div itemprop="actor" itemscope itemtype="http://schema.org/Person">
  <a href="sigourney-weaver.html" itemprop="url">
    <span itemprop="name">Sigourney Weaver</span>
  </a>
</div>
```

```
<!-- Propiedad "actor" de Movie con url y nombre (name) de Persona (Person) -->
```

```
<a href="avatar-trailer.html" itemprop="trailer">Watch Trailer</a>
```

```
<!-- Propiedad "trailer" de la Película -->
```

```
</div>
```

6.1.4. Añadimos el marcado de la calificación promedio (AggregateRating)

```
<div itemscope itemtype="http://schema.org/Movie">
```

```
<!-- Marcamos el tipo Película (Movie) -->
```

```

```

```
<!-- Marcamos la propiedad "imagen (image)" de la Película -->
```

```
<h1 itemprop="name">Avatar</h1> (2009)
```

```
<!-- Marcamos la propiedad "nombre (name)" de la Película -->
```

```
<meta itemprop="datePublished" content="2009-12-18">
```

```
<!-- fecha de publicación: RDFa y Microdatos permiten el uso del atributo "content=" para publicar valores simples legibles por máquinas junto con otros formatos más amigables para humanos.
```

```
Fecha/hora con formato ISO 8601 aaaa-mm-ddThh:mm. Ejemplos: 2015-07-27 o 2015-07-27T15:30-->
```

```
<!-- Las propiedades pueden ser ítems -->
```

```
<div itemprop="aggregateRating"
```

```

itemscope itemtype="http://schema.org/AggregateRating">
Puntuación <span itemprop="ratingValue">8</span> / <span
itemprop="bestRating">10</span>

```

```

Usuarios: <span itemprop="ratingCount">354,477</span> votos
</div>

```

```

<!-- AggregateRating: Calificación basada en múltiples valoraciones o
comentarios, ratingValue: Puntuación, bestRating: Máxima puntuación posible
(5 por defecto), ratingCount: Número de calificaciones -->

```

```

<span itemprop="description">Un marine parapléjico es enviado a una misión a
la luna Pandora...</span>
<!-- Propiedad "descripción (description)" de la Película -->

```

Director:

```

<div itemprop="director" itemscope itemtype="http://schema.org/Person">
  <a href="james-cameron.html" itemprop="url">
    <span itemprop="name">James Cameron</span>
  </a>
</div>
<!-- Propiedad "director" de Movie con url y nombre (name) de Persona
(Person) -->

```

Actores:

```

<a href="sam-worthington.html" itemprop="actor">Sam Worthington</a>,
<a href="zoe-saldana.html" itemprop="actor">Zoe Saldana</a> and
<!-- Por simplicidad, Schema admite marcar de esta forma la propiedad "actor"
de Movie, sobreentendiendo que es el url de una Persona (Person) -->

```

```

<div itemprop="actor" itemscope itemtype="http://schema.org/Person">
<a href="sigourney-weaver.html" itemprop="url">
<span itemprop="name">Sigourney Weaver</span>
</a>
</div>
<!-- Propiedad "actor" de Movie con nombre (name) y url de Persona (Person) -
->

```

```

<a href="avatar-trailer.html" itemprop="trailer">Watch Trailer</a>
<!-- Propiedad "trailer" de la Película -->
</div>

```

6.1.5. Marcado de RDFa

Partimos del marcado con Microdatos anterior. Añadimos el vocabulario *vocab*, *itemscope* se omite, *itemtype* se sustituye con *typeof*, sustituimos *itemprop* con *property* e *itemid* cambia a *resource*.

Como regla general, delimitamos el ítem principal con *div* para indicar el vocabulario y el tipo: `<div vocab="http://schema.org/" typeof="Tipo">...</div>`, los subítems `<div property ="propiedad" typeof="Tipo">...</div>` y las propiedades con *span*: `...`

Tendríamos:

```

<div vocab="http://schema.org/" typeof="Movie">
<!-- Marcamos el tipo Película (Movie) -->

<!-- Marcamos la propiedad "imagen (image)" de la Película -->

<h1 property="name">Avatar</h1> (2009)
<!-- Marcamos la propiedad "nombre (name)" de la Película -->

<meta property="datePublished" content = "2009-12-18">
<!-- fecha de publicación: RDFa y Microdatos permiten el uso del atributo
"content=" para publicar valores simples legibles por máquinas junto con
otros formatos más amigables para humanos.

Fecha/hora con formato ISO 8601 aaaa-mm-ddThh:mm. Ejemplos: 2015-07-27 o
2015-07-27T15:30 -->

<!-- Las propiedades pueden ser ítems -->
<div property="aggregateRating" typeof="AggregateRating">
Puntuación <span property="ratingValue">8</span> / <span
property="bestRating">10</span>

Usuarios: <span property="ratingCount">354,477</span> votos
</div>
<!-- AggregateRating: Calificación basada en múltiples valoraciones o
comentarios, ratingValue: Puntuación, bestRating: Máxima puntuación posible
(5 por defecto), ratingCount: Número de calificaciones -->

<span property="description">Un marine parapléjico es enviado a una misión a
la luna Pandora...</span>
<!-- Propiedad "descripción (description)" de la Película -->

Director:
<div property="director" typeof="Person">
<a href="james-cameron.html" property="url">
<span property="name">James Cameron</span>
</a>
</div>
<!-- Propiedad "director" de Movie con url y nombre (name) de Persona
(Person) -->

Actores:
<a href="sam-worthington.html" property="actor">Sam Worthington</a>,
<a href="zoe-saldana.html" property="actor">Zoe Saldana</a> and
<!-- Por simplicidad, Schema admite marcar de esta forma la propiedad "actor"
de Movie, sobreentendiendo que es el url de una Persona (Person) -->

<div property="actor" typeof="Person">
<a href="sigourney-weaver.html" property="url">
<span property="name">Sigourney Weaver</span>
</a>
</div>
<!-- Propiedad "actor" de Película con propiedad url y nombre (name) del tipo
Persona (Person) -->

<a href="avatar-trailer.html" property="trailer">Watch Trailer</a>
<!-- Propiedad "trailer" de la Película -->

```

```
</div>
```

6.1.6. JSON-LD

Marcamos el objeto de Javascript para datos enlazados de esta forma:

```
<script type="application/ld+json">
{
  "@context" : "http://schema.org",
  "@type" : "Tipo",
  "propiedad" : "valor"
  "propiedad1" : [
    {
      "@type" : "Tipo2",
      "propiedad2" : "valor2"
    },
    {...}
  ],
  ...
}
</script>
```

Quedaría así:

```
<!-- Marcado JSON-LD generado con ayuda del Asistente para el marcado de
datos estructurados de Google (https://www.google.com/webmasters/markup-
helper/) y la herramienta de pruebas de datos estructurados de Google
(https://search.google.com/structured-data/testing-tool). -->
<script type="application/ld+json">
{
  "@context" : "http://schema.org",
  "@type" : "Movie",
  "name" : "Avatar",
  "image" : "avatar.jpg",
  "datePublished" : "2009-12-18", // Fecha/hora con formato ISO 8601 aaaa-mm-
ddThh:mm. Ejemplos: 2015-07-27 o 2015-07-27T15:30

  "description" : " Un marine parapléjico es enviado a una misión a la luna
Pandora...",

  "actor" : [ {
    "@type" : "Person",
    "name" : "Sam Worthington",
    "url" : "sam-worthington.html"
  }, {
    "@type" : "Person",
    "name" : "Zoe Saldana",
    "url" : "zoe-saldana.html"
  }, "Sigourney Weaver" // Schema admite sólo el nombre del actor y
sobreentiende que es un nombre de Persona
  ],
  "director" : {
    "@type" : "Person",
    "name" : "James Cameron",
```

```

    "url" : "james-cameron.html"
  },
  "aggregateRating" : {
    "@type" : "AggregateRating",
    "ratingValue" : "8",
    "bestRating" : "10",
    "ratingCount" : "354,477"
  },
  "trailer" : "avatar-trailer.html"
}
</script>

```

6.2. Mercado de un Producto (Product)

6.2.1. HTML básico

Kenmore White 17" Microondas

```

```

Clasificado 3,5/5 basado en 11 Comentarios

54,55€ en stock

Descripción del producto: microondas de encimera de 20 l. Tiene seis categorías de cocción preestablecidas y características como Add-A-Minute y bloqueo para niños.

Valoración:

No me gustó - por Ellie, 01 de abril de 2011

1/5 estrellas

La lámpara se quemó y ahora tengo que reemplazarlo.

Valor de compra - por Lucas, 25 de marzo de 2011

4/5 estrellas

Gran microondas para el precio. Es pequeño y encaja en mi apartamento.

...

6.2.2. Marcado mediante intercalado de Microdatos: itemscope, itemtype e itemprop

```
<div itemscope itemtype="http://schema.org/Product">
<span itemprop="name">Kenmore White 17" Microondas</span>
```

```

```

```
<div itemprop="aggregateRating"
itemscope itemtype="http://schema.org/AggregateRating">
Clasificado <span itemprop="ratingValue">3,5</span>/5 basado en <span
itemprop="reviewCount">11</span> Comentarios
</div>
```

```
<div itemprop="offers" itemscope itemtype="http://schema.org/Offer">
  <span itemprop="price" content="54.55">54,55</span>
  <!-- Utilizaremos los valores de 0123456789 (Unicode 'dígito cero'
(U+0030) a 'dígito nueve' (U+0039)) en lugar de símbolos de Unicode
superficialmente similares. -->
```

```
<!-- Usaremos '.' ("Punto" de Unicode (U+002E)) en lugar de "," para
indicar un punto decimal. Evitaremos usar estos símbolos como
separador de legibilidad. -->
```

```
<span itemprop="priceCurrency" content="EUR">€</span>
<!-- La divisa (en formato ISO 4217 de 3 letras) del precio en lugar
de incluir en el valor el considerado como símbolo ambiguo € -->
```

```
<link itemprop="availability" href="http://schema.org/InStock"/> en stock
<!-- Forma de especificar un valor de Enumeración (que puede tomar un
número limitado de valores). -->
```

```
<!-- Propiedad "availability" o disponibilidad del Producto, puede tomar
valores de Enumeración de ItemAvailability: Discontinued, InStock,
InStoreOnly, LimitedAvailability, OnlineOnly, OutOfStock, PreOrder, PreSale,
SoldOut -->
</div>
```

Descripción del producto:

```
<span itemprop="description">Microondas de encimera de 17". Tiene seis
categorías de cocción preestablecidas y características como Add-A-Minute y
bloqueo para niños.</span>
```

Valoración:

```
<div itemprop="review" itemscope itemtype="http://schema.org/Review">
  <span itemprop="name">No me gustó</span> -
  por <span itemprop="author">Ellie</span>,
  <meta itemprop="datePublished" content="2011-04-01">01 de abril de
2011
```

```
<div itemprop="reviewRating" itemscope itemtype="http://schema.org/Rating">
  <meta itemprop="worstRating" content = "1">
  <span itemprop="ratingValue">1</span>/
  <span itemprop="bestRating">5</span> estrellas
</div>
```

```
<span itemprop="description"> La lámpara se quemó y ahora tengo que
reemplazarlo. </span>
</div>
```

```
<div itemprop="review" itemscope itemtype="http://schema.org/Review">
  <span itemprop="name">Valor de compra</span> -
  por <span itemprop="author">Lucas</span>,
  <meta itemprop="datePublished" content="2011-03-25">25 de marzo de 2011
  <div itemprop="reviewRating" itemscope
    itemtype="http://schema.org/Rating">
    <meta itemprop="worstRating" content = "1">
    <span itemprop="ratingValue">4</span>/
    <span itemprop="bestRating">5</span> estrellas
  </div>
  <span itemprop="description">Gran microondas para el precio. Es
pequeño y encaja en mi apartamento.</span>
</div>
</div>
```

6.2.3. Mercado de RDFa

```
<div vocab="http://schema.org/" typeof="Product">
```

```

<span property="name">Kenmore White 17" Microondas</span>



<div property="aggregateRating" typeof="AggregateRating">
Clasificado <span property="ratingValue">3,5</span>/5 basado en <span
property="reviewCount">11</span> Comentarios
</div>

<div property="offers" typeof="Offer">
  <span property="price" content="54.55">54,55</span>
  <span property="priceCurrency" content="EUR">€</span>
  <link property="availability" href="http://schema.org/InStock"/> en stock
</div>

```

Descripción del producto:

```

<span property="description">Microondas de encimera de 17". Tiene seis
categorías de cocción preestablecidas y características como Add-A-Minute y
bloqueo para niños.</span>

```

Valoración:

```

<div property="review" typeof="Review">
  <span property="name">No me gustó</span> -
  por <span property="author">Ellie</span>,
  <meta property="datePublished" content="2011-04-01">01 de abril de
2011
  <div property="reviewRating" typeof="Rating">
    <meta property="worstRating" content = "1">
    <span property="ratingValue">1</span>/
    <span property="bestRating">5</span> estrellas
  </div>

<span property="description"> La lámpara se quemó y ahora tengo que
reemplazarlo. </span>
</div>

<div property="review" typeof="Review">
  <span property="name">Valor de compra</span> -
  por <span property="name">Lucas</span>,
  <meta property="datePublished" content="2011-03-25">25 de marzo de 2011
  <div property="reviewRating" typeof="Rating">
    <meta property="worstRating" content = "1">
    <span property="ratingValue">4</span>/
    <span property="bestRating">5</span> estrellas
  </div>
  <span property="description">Gran microondas para el precio. Es
pequeño y encaja en mi apartamento.</span>
</div>
...
</div>

```

6.2.4. JSON-LD

```

<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "Product",

```



```

"aggregateRating": {
  "@type": "AggregateRating",
  "ratingValue": "3.5",
  "reviewCount": "11"
},

"description": "Microondas de encimera de 17". Tiene seis categorías de
cocción preestablecidas y características como Add-A-Minute y bloqueo para
niños.",

"name": "Kenmore White 17\" Microwave",

"image": "kenmore-microwave-17in.jpg",

"offers": {
  "@type": "Offer",
  "availability": "http://schema.org/InStock",
  "price": "54.55",
  "priceCurrency": "EUR"
},

"review": [
  {
    "@type": "Review",
    "author": "Ellie",
    "datePublished": "2011-04-01",
    "description": "La lámpara se quemó y ahora tengo que reemplazarla.",
    "name": "No me gustó",

    "reviewRating": {
      "@type": "Rating",
      "bestRating": "5",
      "ratingValue": "1",
      "worstRating": "1"
    }
  },
  {
    "@type": "Review",
    "author": "Lucas",
    "datePublished": "2011-03-25",
    "description": "Gran microondas por el precio. Es pequeño y encaja en
mi apartamento.",
    "name": "Value purchase",
    "reviewRating": {
      "@type": "Rating",
      "bestRating": "5",
      "ratingValue": "4",
      "worstRating": "1"
    }
  }
]
}
</script>

```

6.3. Mercado de un Evento (Event)

6.3.1. HTML básico

```

<a href="acb.html">
Real Madrid - Estudiantes
</a>
Jue, 21/04/16
8:00 p.m.
<a href="palacio-deportes.html">Palacio de Deportes</a>
Madrid
Precios desde 35€
1938 entradas disponibles

```

6.3.2. Marcado mediante intercalado de Microdatos: `itemscope`, `itemtype` e `itemprop`

```

<div itemscope itemtype="http://schema.org/Event">
  <a itemprop="url" href="acb.html">
  <span itemprop="name"> Real Madrid - Estudiantes</span>
  </a>
  <meta itemprop="startDate" content="2016-04-21T20:00">
    Jue, 21/04/16
    8:00 p.m.

  <div itemprop="location" itemscope itemtype="http://schema.org/Place">
    <a itemprop="url" href="palacio-deportes.html">
      Palacio de Deportes
    </a>
    <div itemprop="address" itemscope
itemtype="http://schema.org/PostalAddress">
      <span itemprop="addressLocality">Madrid</span>,
    </div>
  </div>

  <div itemprop="offers" itemscope
itemtype="http://schema.org/AggregateOffer">
    Precios desde <span itemprop="lowPrice">35€</span>
    <span itemprop="offerCount">1938</span> entradas disponibles
  </div>
</div>

```

6.3.3. Marcado de RDFa

```

<div vocab="http://schema.org/" typeof="Event">
  <a property="url" href="acb.html">
  <span property="name"> Real Madrid - Estudiantes </span>
  </a>
  <meta property="startDate" content="2016-04-21T20:00">
    Jue, 21/04/16
    8:00 p.m.

  <div property="location" typeof="Place">
    <a property="url" href="palacio-deportes.html">
      Palacio de Deportes
    </a>
    <div property="address" typeof="PostalAddress">
      <span property="addressLocality"> Madrid </span>,
    </div>
  </div>

  <div property="offers" typeof="AggregateOffer">
    Precios desde <span property="lowPrice">35€</span>

```

```
<span property="offerCount">1938</span> entradas disponibles
</div>
```

```
</div>
```

6.3.4. JSON-LD

```
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "Event",
  "name": "Real Madrid - Estudiantes",

  "location": {
    "@type": "Place",

    "address": {
      "@type": "PostalAddress",
      "addressLocality": "Madrid",
    },
    "url": "palacio-deportes.html"
  },

  "offers": {
    "@type": "AggregateOffer",
    "lowPrice": "$35",
    "offerCount": "1938"
  },
  "startDate": "2016-04-21T20:00",
  "url": "acb.html"
}
</script>
```

7. CONCLUSIONES

Históricamente, la Web ha publicado su contenido en código HTML y ha estado destinada a humanos (y no a máquinas). Había múltiples dificultades que surgían en el proceso de extracción de datos estructurados a partir del HTML por aplicaciones como: búsqueda Web, comparadores de precios o reservas.

Para facilitar el intercambio de información Web surgieron varias iniciativas: Meta Content Framework (MCF) de Apple, RDF, RDFS, DAML, OWL, Microformatos, SPARQL, Turtle, N3, GRDDL, R2RML, FOAF, SIOC, SKOS,...

Schema.org supuso el lanzamiento de una iniciativa de metadatos dirigida específicamente a lograr óptimos resultados y visualización de los motores de búsqueda. **El objetivo era crear un esquema de metadatos que los webmasters pudieran utilizar óptimamente**, continuando con el espíritu de sitemaps.org.

Por lo tanto, **los motores de búsqueda más grandes del mundo, conjuntamente, consolidaron sus esfuerzos para proporcionar para**

optimización de búsqueda (SEO) el mejor conjunto posible de esquemas de metadatos.

Gran parte del vocabulario fue inspirado por los predecesores como microformatos, FOAF, y GoodRelations. En un primer momento, la cantidad de formatos disponibles era escasa, pero Schema.org ha crecido en número de esquemas, así como en popularidad.

Un método de extensión integrado permite la incorporación continua de muchas propiedades nuevas.

Schema.org facilita el marcado de contenido de los sitios web por medio de la información de metadatos auto-descriptiva. Se utilizan Microdatos y Ontología en HTML5, que es reconocible por los analizadores (parsers).

Supone una iniciativa muy interesante porque es un proyecto abierto que ha puesto de acuerdo a las principales empresas de búsqueda, dejando aparte su rivalidad. Su uso es muy sencillo e intuitivo y trata de conseguir un equilibrio entre formatos amigables para humanos y para máquinas, aunque se fomenta la orientación a máquinas para mejorar la usabilidad de los publicadores de datos.

Posee una jerarquía de Clases o Tipos con propiedades que admiten herencia y polimorfismo. Por ejemplo: el polimorfismo de la propiedad startDate de los tipos Events, Reservations y Offers.

Permite utilizar extensiones propias y externas para añadir expresividad.

Los publicadores (con mejores resultados de búsqueda), las aplicaciones consumidoras de datos (con mejor acceso a ellos) y los usuarios (con mejor servicio) se benefician de su uso.

La gran proliferación de dispositivos ha facilitado el uso de asistentes personales como Google Now, Cortana (Microsoft) o Siri (Apple). Estos asistentes pueden avisarnos de reservas de vuelos, restaurantes o de eventos (deportivos o musicales), haciendo uso del marcado de los correos electrónicos enviados por las empresas de reserva.

Por lo tanto, Schema.org ha resultado ser una excelente herramienta para la representación de conocimiento y su intercambio.

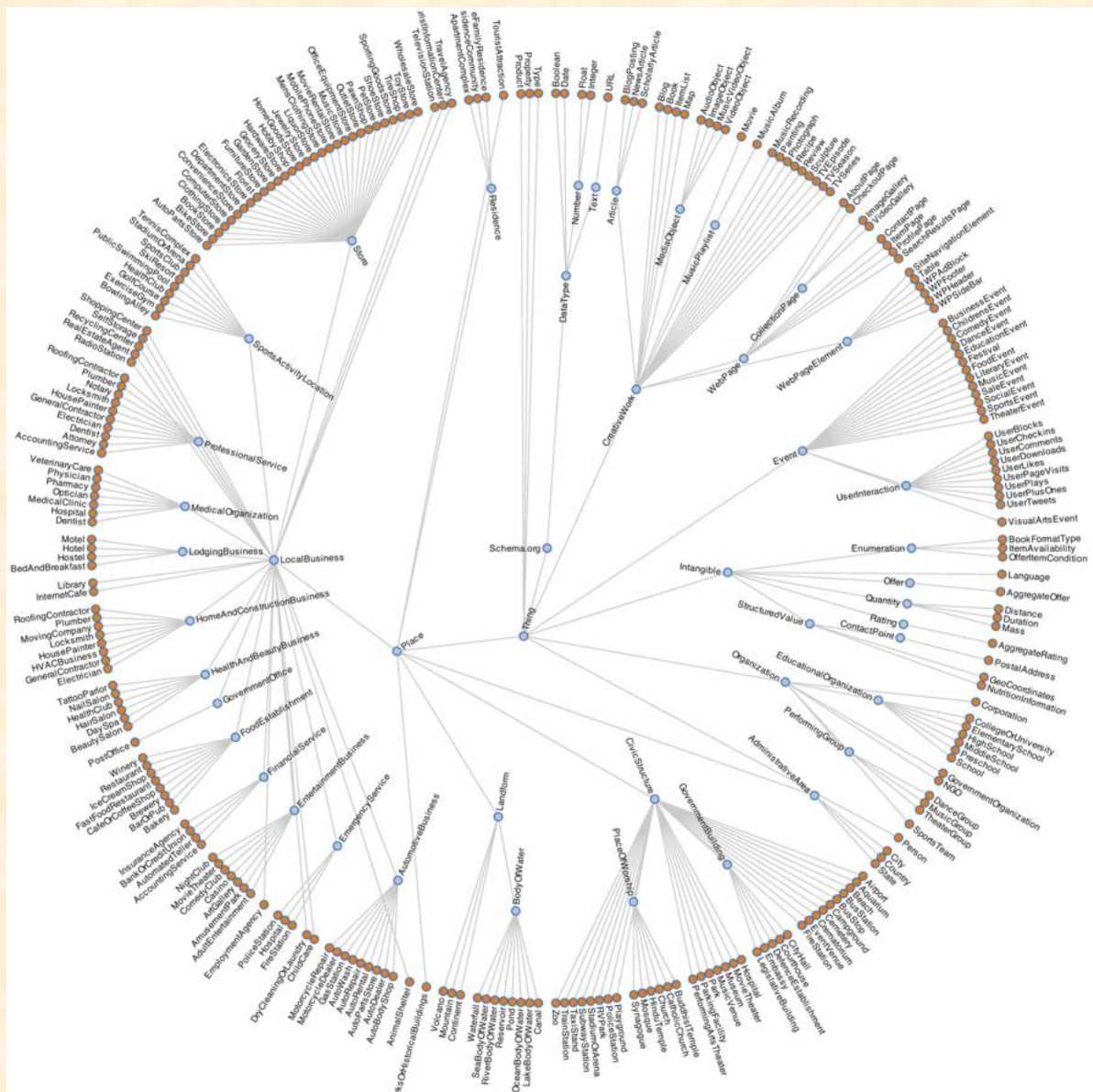


Fig. 18. Representación de la Jerarquía de Schema.org <https://www.flickr.com/photos/danbri/5925660995>

GLOSARIO

HTML *m* **HyperText Markup Language**, comúnmente abreviado como **HTML**, es el estándar de [lenguaje de marcado](#) utilizado para crear [páginas web](#). Junto con [CSS](#) y [JavaScript](#) , HTML es una tecnología fundamental utilizada para crear páginas web

JSON *m* (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del [Lenguaje de Programación JavaScript](#), [Standard ECMA-262 3rd Edition - Diciembre 1999](#). JSON es un

formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON-LD *m* Siglas de JavaScript Object Notation for Linked Data, es un método de codificación de [datos enlazados](#) usando [JSON](#). **JSON-LD** está diseñado en torno al concepto de un "contexto" para proporcionar asignaciones adicionales de JSON a un modelo [RDF](#). El contexto enlaza las propiedades de objetos en un documento JSON a conceptos de una [ontología](#).

Markup *m* Un **lenguaje de marcado** o **lenguaje de marcas** es una forma de codificar un documento que, junto con el texto, incorpora [etiquetas](#) o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

Metadatos *m* Son datos que describen otros datos. Para varios campos de la [informática](#), como la [recuperación de información](#) o la [web semántica](#), los **metadatos** en [etiquetas](#) son un enfoque importante para construir un puente sobre el [intervalo semántico](#).

Ontología *f* Definición formal de tipos, propiedades, y relaciones entre [entidades](#) que realmente o fundamentalmente existen para un [dominio de discusión](#) en particular. En los campos de la [inteligencia artificial](#), la [Web Semántica](#), [ingeniería de sistemas](#), [ingeniería de software](#), [informática biomédica](#), [bibliotecología](#) y [arquitectura de la información](#) se crean ontologías para limitar la complejidad y para organizar la información. La **ontología** puede entonces ser aplicada para [resolver problemas](#).

OWL *m* Sigla de Ontology Web Language, un lenguaje estándar para la descripción de ontologías inspirado en la lógica descriptiva y lenguajes previos.

RDF *m* Sigla de Resource Description Framework, un estándar de [W3C](#) que describe un modelo de datos para la web semántica basado en enunciados de la forma sujeto-predicado-objeto.

RDFa *m* Sigla de Resource Description Framework in attributes, es una recomendación [W3C](#) que añade un conjunto de extensiones en el nivel de atributo a [HTML](#), [XHTML](#) y diversos tipos de documentos basados en XML para incrustar ricos [metadatos](#) RDF dentro de los documentos Web.

Rich snippets o **fragmentos enriquecidos** *m* hacen referencia a lo que es extraído de algo, específicamente de las [páginas de resultados de búsqueda](#). Los fragmentos enriquecidos aparecieron en mayo del 2009 debido a la necesidad del [buscador](#), ([Google](#), [Yahoo!](#), [Bing](#), etc.) de mostrar más datos al usuario y de enriquecer las páginas de resultados de búsqueda. Para

SEO (Search Engine Optimization) Optimización de motores de búsqueda es el proceso de afectar a la visibilidad de un [sitio web](#) o una [página web](#) en los resultados no pagados en un [motor de búsqueda en la web](#) - a menudo referido como "naturales", "[orgánicos](#)" o resultados "obtenidos". En general, cuanto más pronto (o con mayor puntuación en la página de resultados de búsqueda), y con mayor frecuencia un sitio aparece en la lista de resultados

de búsqueda, más visitantes recibirá de los usuarios del motor de búsqueda, y estos visitantes se pueden convertir en clientes.

Search engine optimization (SEO) is the process of affecting the visibility of a [website](#) or a [web page](#) in a [web search engine](#)'s unpaid results — often referred to as "natural," "[organic](#)," or "earned" results. In general, the earlier (or higher ranked on the **search** results page), and more frequently a site appears in the **search** results list, the more visitors it will receive from the **search** engine's users, and these visitors can be converted into customers.

W3C *m* Siglas de World Wide Web Consortium, una comunidad internacional dedicada al desarrollo de estándares para la web.

Web Semántica (*Semantic Web*) *f* Conjunto de actividades desarrolladas en el seno de [World Wide Web Consortium](#) con tendencia a la creación de tecnologías para publicar datos legibles por aplicaciones informáticas (máquinas en la terminología de la Web semántica).¹Se basa en la idea de añadir [metadatos semánticos](#) y [ontológicos](#) a la [World Wide Web](#).

Web Semántica: La **web semántica** (del [inglés](#) *semantic web*) es un conjunto de actividades desarrolladas en el seno de [World Wide Web Consortium](#) con tendencia a la creación de tecnologías para publicar datos legibles por aplicaciones informáticas (máquinas en la terminología de la Web semántica).¹Se basa en la idea de añadir [metadatos semánticos](#) y [ontológicos](#) a la [World Wide Web](#). Esas informaciones adicionales —que describen el contenido, el significado y la relación de los datos— se deben proporcionar de manera formal, para que así sea posible evaluarlas automáticamente por máquinas de procesamiento. El objetivo es mejorar Internet ampliando la [interoperabilidad](#) entre los sistemas informáticos usando "agentes inteligentes". Los agentes inteligentes son programas en las computadoras que buscan información sin operadores humanos.

Fuentes

1. **schema.org**. [En línea] <https://schema.org/docs/faq.html>.
2. —. [En línea] <http://schema.org/docs/schemas.html>.
3. **Jovanovic, Jelena**. [En línea] <http://ai.fon.bg.ac.rs/wp-content/uploads/2016/10/Linking-Web-Data-eng.pdf>.
4. **Guha, R.V. (Google), Brickley, Dan (Google) y Macbeth, Steve (Microsoft)**. [En línea] <http://queue.acm.org/detail.cfm?id=2857276>.
5. **Cudre-Mauroux, Philippe**. [En línea] http://www.wirtschaft.bfh.ch/fileadmin/wgs_upload/wirtschaft_und_verwaltung/6_forschung/eGov_Fokus/Hand-Outs/eGF_2_16/05_Presentation_Cudre-Mauroux.pdf.
6. **schema.org**. [En línea] schema.org/docs/datamodel.html.

7. —. [En línea] <http://schema.org/docs/extension.html>.
8. **Sporny, Manu**. [En línea] manu.sporny.org/2011/uber-comparison-rdfa-md-uf/.
9. —. [En línea] manu.sporny.org/2012/mythical-differences/.
10. **Wikipedia**. *Wikipedia*. [En línea] 2017. [https://en.wikipedia.org/wiki/Microdata_\(HTML\)](https://en.wikipedia.org/wiki/Microdata_(HTML)).
11. **De Teresa, Tomás**. [En línea] deteresa.com/datos-estructurados/.
12. [En línea] xmlns.com/foaf/spec/20070524.html.
13. **Wikipedia**. *Wikipedia*. [En línea] es.wikipedia.org/wiki/FOAF.
14. **Laufer, Carlos**. [En línea] ceweb.br/guias/web-semantic//es/capitulo-6/.
15. [En línea] es.wikipedia.org/wiki/Dublin_Core.
16. **Unor**. [En línea] <http://webmasters.stackexchange.com/questions/91623/advantages-disadvantages-of-adding-schema-org-markups-in-excess>.
17. **Schema.org**. [En línea] schema.org/docs/gs.html#schemaorg_expected.
18. **Guha, R.V**. [En línea] videolectures.net/site/normal_dl/tag=817824/iswc2013_guha_tunnel_01.pdf.
19. **van Berkel, Mark**. [En línea] http://onlinepresence.coach/marketing/understanding-schema-org/#Challenges_with_Schemaorg.
20. **Goel, Kavi (Google)**. [En línea] schema.org/docs/kickoff-workshop/sw1109_Implementation.pdf.
21. *Wikipedia*. [En línea] <https://www.wikipedia.org/>.
22. **Barker, Phil y Campbell, Lorna M**. [En línea] <http://publications.cetis.org.uk/wp-content/uploads/2014/06/schemaBriefing.pdf>.
23. **Antoniazzi, Matteo**. [En línea] <https://docs.google.com/viewer?url=https%3A%2F%2Fodia.archive.knowledgearc.net%2Fbitstream%2Fhandle%2F10642%2F2733%2FAntoniazzi.pdf%3Fsequence%3D2%26isAllowed%3Dy>.

Índice de Figuras

Fig. 1. seopressor.com.....	13
Fig. 2. Creative Work (Trabajo Creativo) http://publications.cetis.org.uk/2014/960	14
Fig. 3. Jerarquía de Tipos de Schema.org. blogs.pjjk.net/phil/topics/schema-org/	15
Fig. 4. Ontología Schema.org-2013.03.25 representada mediante WebVOWL	16

Fig. 5. Knowledge Graph de Google. searchengineland.com..... 17

Fig. 6. Comparación entre un resultado de búsqueda normal y un resultado enriquecido de una Receta (Recipe). 17

Fig. 7. Búsqueda interactiva de Yandex que permite registrarse en un vuelo. searchenginewatch.com 18

Fig. 8. Ejemplos de Confirmation Cards de Google Now. <https://developers.google.com/schemas/now/cards>. 18

Fig. 9. Información de Cortana sobre estado del vuelo. msdn.microsoft.com. 19

Fig. 10. URLs y Dominios con y sin triples (sujeto-predicado-objeto). Web Data Commons. 19

Fig. 11. Porcentaje de las distintas sintaxis en Dominios y URLs en Octubre de 2016. Web Data Commons. 20

Fig. 12. Estadística comparativa de Schema.org. R.V. Guha (Google) pbs.twimg.com 20

Fig. 13. Jerarquía de clases. support.google.com 21

Fig. 14. FOAF mediante WebVOWL. <http://vowl.visualdataweb.org/webvowl/#foaf> 30

Fig. 15. Ejemplo SKOS. es.wikipedia.org/wiki/Simple_Knowledge_Organization_System..... 35

Fig. 16. Arquitectura tecnológica de la web semántica. <https://es.wikipedia.org>..... 38

Fig. 17. Evolución Web: A medida que la cantidad de datos aumenta, la productividad de las búsquedas por palabra clave disminuye (gráfica roja). Las tecnologías semánticas ayudan a remontar la productividad afrontando el rápido crecimiento de la información (gráfica verde). web-evolution-nova-spivack-twine-3-728.jpg 39

Fig. 18. Representación de la Jerarquía de Schema.org <https://www.flickr.com/photos/danbri/5925660995> 53

Tablas

Tabla 1. Comparación de RDFa / RDFa Lite con Microdata (8) 23

Tabla 2. Atributos de Microdatos. 24

Tabla 3. Elementos de sintaxis y palabras clave <https://www.w3.org/TR/json-ld/> 26

Tabla 4. Vocabulario FOAF (12) 30

Tabla 5. Un ejemplo sencillo de FOAF serializado en RDF/XML. (13) 31

Tabla 6. Comparación Dublin Core/Schema.org. Fuente: seopressor.com 31

Tabla 7. DUBLIN CORE (15) 33

Tabla 8. Ejemplo Dublin Core. Book-Semantic-Web-Guideline.pdf..... 33

Tabla 9. Vocabulario SKOS organizado por temas. Wikipedia 34

Tabla 10. Retos a los que se enfrenta la web semántica. https://en.wikipedia.org/wiki/Semantic_Web 39

Índice de entradas

fragmentos enriquecidos, 21, 22
Fragmentos Enriquecidos, 17

rich snippets, 21
Rich Snippets, 7, 17