

---

# Formats d'intercanvi

---

PID\_00272094

Blas Torregrosa García

---

Temps mínim de dedicació recomanat: 2 hores

---



**Blas Torregrosa García**

Enginyer en Informàtica i màster universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions (MISTIC) per la Universitat Oberta de Catalunya (UOC). Especialitzat en ciberseguretat. Professor col·laborador del màster de Ciència de Dades de la UOC i professor associat a la Universitat de Valladolid (UVA).

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Ferran Prados Carrasco (2020)

Primera edició: febrer 2020  
© Blas Torregrosa García  
Tots els drets reservats  
© d'aquesta edició, FUOC, 2020  
Av. Tibidabo, 39-43, 08035 Barcelona  
Realització editorial: FUOC

*Cap part d'aquesta publicació, incloent-hi el disseny general i la coberta, no pot ser copiada, reproduïda, emmagatzemada o transmesa de cap manera ni per cap mitjà, tant si és elèctric com químic, mecànic, òptic, de gravació, de fotocòpia o per altres mètodes, sense l'autorització prèvia per escrit dels titulars dels drets.*

# Índex

<b>Introducció</b> .....	5
<b>1. XML</b> .....	7
1.1. Validació de documents XML .....	9
<b>2. JSON</b> .....	10
2.1. JSON-LD .....	11
<b>3. Dades integrades amb HTML</b> .....	14
3.1. HTML5 Microdata .....	14
3.1.1. LocalBusiness .....	14
3.2. Microformats .....	15
3.2.1. Microformat hCalendar.....	16
3.2.2. Microformat hCard.....	16
3.2.3. rel-license.....	17
3.3. RDFa .....	18
<b>4. CSV</b> .....	20



## **Introducció**

En aquest mòdul s'introdueixen els formats més freqüents a la web de dades, com ara XML i JSON, i també els formats pensats per estar integrats en pàgines HTML, com són MicroFormats, MicroData o RDFa. També el format d'intercanvi CSV.



## 1. XML

XML (*eXtensible Markup Language*) va ser proposat pel W3C com un llenguatge de marques extensible per a la web el 1996. XML deriva d'SGML (*Standard Generalized Markup Language*) un metallenguatge que proporciona una sintaxi comuna per a sistemes de marques textual i del qual van derivar les primeres versions d'HTML. El model XML està adaptat per representar informació textual que combina text i elements de marques.

El model XML consta d'una estructura en **arbre** en què cada node de l'arbre es defineix com un **element** d'un tipus de dada. Un element és una part del document delimitat per una etiqueta d'inici (com <nom>) i una etiqueta de finalització (com </nom>). Les etiquetes d'inici i les etiquetes finals envolten el contingut i possiblement altres elements de marques, i també hi poden haver elements buits (com <pagina/>).

Tota l'estructura del document està fixada a un element arrel (l'element superior). Mentre que amb HTML l'element arrel és <html>, amb XML és possible triar l'element arrel del document.

Figura 1. Exemple de dades d'un curs representat amb XML

```
<?xml version="1.0"?>
<curs nom="Dades">
  <estudiant id="alicia">
    <nom>Alicia</nom>
    <genere>Dona</genere>
  </estudiant>
  <estudiant id="benito">
    <nom>Benito</nom>
    <genere>Home</genere>
    <dataNaixement>1981-11-24</dataNaixement>
  </estudiant>
</curs>
```

XML ha desenvolupat tecnologies de consulta i transformació a altres formats. Entre ells XPath, que és un llenguatge per seleccionar parts d'XML, i que està integrat a altres tecnologies com XSLT o XQuery.

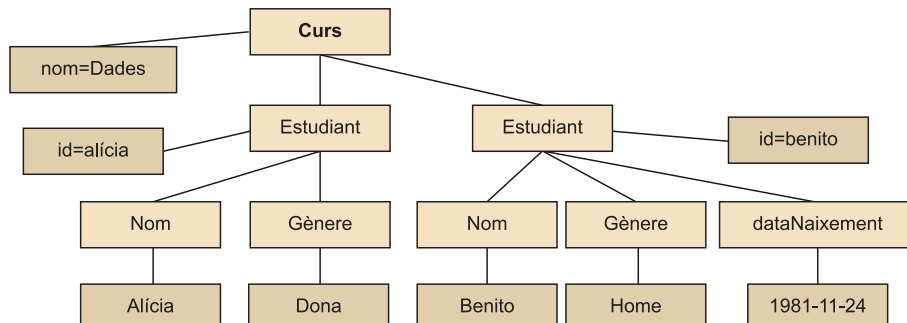
El fragment XPath següent busca els noms de tots els estudiants d'un curs que són dones:

Figura 2. Exemple de consulta amb XPath

```
// estudiant[genere = "Dona"]/ nom
```

XML defineix el concepte de documents ben formats i documents vàlids. Els **documents ben formats** són documents XML amb una sintaxi correcta, mentre que els **documents vàlids** són documents que, a més d'estar ben formats, s'ajusten a una definició d'esquema.

Figura 3. Arbre del document XML



Per definir un esquema amb XML hi ha diverses opcions:

1) **Document Type Definition (DTD)**. L'especificació XML conté un mecanisme bàsic per definir un esquema per a documents XML, heretat d'SGML i anomenat *DTD*. Un *DTD* genera l'estructura d'una família de documents XML.

DTD utilitza expressions regulars per definir l'estructura dels documents. Però DTD té un control limitat sobre els tipus de dades. Per exemple, no és possible validar que una data tingui el format de data.

Figura 4. Exemple de DTD

```
<!ELEMENT curs (estudiant*)>
<!ELEMENT estudiant (nom,genere,dataNaixement?)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT genere (#PCDATA)>
<!ELEMENT dataNaixement (#PCDATA)>
<!ATTLIST estudiant id ID #REQUIRED>
<!ATTLIST curs name CDATA #IMPLIED>
```

2) **XML Schema**. Aquesta especificació es divideix en dues parts: la primera defineix l'estructura dels documents XML i la segona és un repertori de tipus de dades d'XML Schema.

Un validador d'XML Schema afegeix a cada estructura del document XML informació addicional denominada PSVI (*Post-Schema Validation Infoset*). Aquesta estructura conté informació sobre el procés de validació.



Figura 5. Exemple d'XML Schema

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="curs">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="estudiant" minOccurs='1'
                    maxOccurs='100' type="Estudiant"/>
      </xs:sequence>
      <xs:attribute name="nom" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:complexType name="Estudiant">
    <xs:sequence>
      <xs:element name="nom" type="xs:string" />
      <xs:element name="genere" type="Genere" />
      <xs:element name="dataNaixement" type="xs:date" minOccurs='0' />
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use='required' />
  </xs:complexType>
  <xs:simpleType name="Genere">
    <xs:restriction base="xs:token">
      <xs:enumeration value="Home" />
      <xs:enumeration value="Dona" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

## 1.1. Validació de documents XML

Hi ha diferents enfocaments per indicar com s'han de validar els documents XML respecte d'un esquema. Aquests en són alguns:

1) **Esquema integrat.** Els DTD es poden integrar directament en els documents XML.

Figura 6. DTD integrat en un document XML

```

<!DOCTYPE curs [
  <!ELEMENT curs (estudiant*)>
  <!ELEMENT estudiant (nom,genere,dataNaixement?)>
  ...
]>
<curs name="Dades">
  ...
</curs>

```

2) **Associació directa amb l'esquema XML.** Es pot fer utilitzant els atributs `xsi: SchemaLocation` o `xsi: noNamespaceSchemaLocation`.

Figura 7. Document XML amb declaració d'esquema i localització

```

<curs xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://exemple.co.ca/ns/Curs
                        http://exemple.co.ca/curs.xsd">
  ...
</curs>

```

## 2. JSON

Douglas Crockford va proposar JSON (*JavaScript Object Notation*) el 2001 com un subconjunt de JavaScript. El JSON ha evolucionat cap a un format d'intercanvi de dades independent amb la seva pròpia especificació de l'ECMA (**European Computer Manufacturers Association**).

Un document JSON es pot definir recursivament de la manera següent:

- `true`, `false` i `null` són valors JSON.
- Qualsevol nombre decimal també és un valor JSON.
- Qualsevol cadena de caràcters Unicode entre cometes (") també és un valor JSON, denominat *valor de cadena*.
- Si  $k_1, k_2, \dots, k_n$  són  $n$  valors de cadena diferents i  $v_1, v_2, \dots, v_n$  són valors JSON, llavors  $\{k_1: v_1, k_2: v_2, \dots, k_n: v_n\}$  són valors JSON, denominats *objectes*. En aquest cas, cada  $k_i : v_i$  és un parell atribut-valor. L'ordre dels parells no és significatiu.
- Si  $v_1, v_2, \dots, v_n$  són valors JSON, llavors  $[v_1, v_2, \dots, v_n]$  és un valor JSON denominat *array*. L'ordre dels elements de l'*array* no és significatiu.

Cal tenir en compte que en el cas d'*arrays* i objectes, els valors  $v_i$  també poden ser objectes o matrius, la qual cosa permet als documents un nivell arbitrari d'imbricació. D'aquesta forma el model de dades JSON es pot representar com un arbre.

Figura 8. Exemple de JSON amb dues claus: nom i estudiants

```
{
  "nom": "Dades" ,
  "estudiants": [
    {
      "nom": "Alicia",
      "genere": "Dona",
      "edat": 21
    },
    {
      "nom": "Benito",
      "genere": "Home",
      "dataNaixement": "1981-11-24"
    }
  ]
}
```

JSON Schema es va proposar com un llenguatge d'esquema per JSON amb una funcionalitat semblant a XML Schema per a XML. Està escrit utilitzant la sintaxi JSON i és independent del llenguatge de programació. Conté els tipus

### ECMA Script

**ECMA Script** és una especificació de llenguatge de programació basat en JavaScript proposat com a estàndard per Netscape. Actualment està acceptat com l'estàndard ISO 16262.

de dades predefinides següents: `null`, `boolean`, `object`, `array`, `number` i `string`, i permet definir restriccions sobre cadascun d'aquests. Les següents són paraules reservades amb JSON Schema:

- `$schema`. Indica que l'esquema està definit segons una versió de l'estàndard.
- `$id`. Defineix l'URI per a l'esquema.
- `Title` i `description`. No afegeixen restriccions, solament descriuen l'esquema.
- `Type`. Defineix la restricció de validació de les dades JSON.

Amb JSON Schema és possible tenir definicions reutilitzables a les quals després es pot referenciar.

Figura 9. Exemple de JSON Schema

```
{ "$schema": "http://json-schema.org/draft-04/schema#",  
  
  "definicions": {  
    "estudiant": { "type": "object",  
      "properties": {  
        "nom": { "type": "string" },  
        "genere": { "type": "string", "enum": ["Dona", "Home"] },  
        "dataNaixement": { "type": "string", "format": "date" },  
        "edat": { "type": "integer", "minimum": 1 }  
      },  
      "required": ["nom", "genere"]  
    }  
  },  
  
  "type": "object",  
  "properties": {  
    "nom": { "type": "string" },  
    "estudiants": { "type": "array",  
      "items": { "$ref": "#/definicions/estudiant" }  
    }  
  },  
  "required": ["nom", "estudiants"]  
}
```

## 2.1. JSON-LD

JSON-LD (*JSON for Linking Data*) és un format per estructurar dades en pàgines web, anàleg a Microdata i RDFa. Per a això JSON-LD es basa en la notació JSON i l'amplia amb una sintaxi mitjançant la qual les dades s'anoten en funció d'esquemes de validesa universal. L'especificació de JSON-LD figura com a recomanació oficial del W3C des de 2014.

Figura 10. Anotació amb JSON-LD

```
<script type="application/ld+json">{
  "@context": "https://schema.org",
  "@type": "Course",
  "name": "Introducció a les dades",
  "description": "Introducció als tipus i orígens de les dades.",
  "provider": {
    "@type": "Organization",
    "name": "Universitat Oberta de Catalunya",
    "sameAs": "http://www.uoc.edu"
  }
}</script>
```

JSON-LD no s'associa, en principi, a cap vocabulari en particular, si bé el projecte schema.org està considerat *de facto* com l'estàndard per a l'anotació.

Els elements següents són de JSON-LD:

- Etiqueta `<script>`. Indica que el contingut de l'*script* és JSON-LD (entre claus {}). Tot i que JSON s'anota en etiquetes *script*, no constitueix un codi executable.
- `@context`. Defineix el vocabulari que s'utilitza per marcar les dades, en aquest cas schema.org.
- `@type`. Especifica el tipus d'element que s'està anotant.
- **Parells propietat-valor**. Constitueixen les propietats del tipus d'element descrit. Els components són:
  - **Propietat**: prové del vocabulari del context i ha d'anar entre cometes dobles (""). Ha de pertànyer a les propietats permeses per al tipus d'element.
  - **Valor**: és el valor que té la propietat i ha de concordar amb el tipus de la propietat. Així, valors simples s'assignen per separat mentre que els valors múltiples són *arrays* que van entre claudàtors ([]). Els nombres enters, punt flotant i en doble precisió no necessiten cometes, i els objectes van entre claus ({}).

L'avantatge que ofereix JSON-LD enfront d'altres formats és que les dades no s'integren en el codi HTML, sinó que es poden implementar com un bloc de codi autònom. Això contribueix a la llegibilitat del codi i a la seva independència respecte al contingut visible.

Google i altres cercadors donen suport a les anotacions JSON-LD per a les extensions següents:

1) **Dades corporatives de contacte:** si les organitzacions marquen semànticament la seva informació de contacte, Google i altres cercadors poden mostrar-les en els SERP.

2) **Logotips:** quan es marquen els logotips, el cercador sap quin gràfic ha d'utilitzar com a imatge corporativa oficial.

3) **Vincles a perfils socials:** si s'han marcat els enllaços als perfils socials com a dades estructurades, els cercadors amplien els resultats de persones i organitzacions amb els botons corresponents a les xarxes socials (Facebook, Twitter, Instagram, YouTube, LinkedIn, etc.).

Google i altres cercadors suporten les anotacions JSON-LD per als tipus següents: articles periodístics, llibres, música, cursos, ofertes de treball, podcasts, vídeos, pel·lícules, receptes de cuina, crítiques, productes, conjunts de dades i més.

**SERP**

**SERP** (*Search Engine Results Page*) fa referència als resultats que mostra una pàgina de cercador com ara Google, Bing o Yahoo! entre els més destacats.

## 3. Dades integrades amb HTML

### 3.1. HTML5 Microdata

**Microdata** és una alternativa per integrar dades estructurades amb HTML 5. Microdata utilitza etiquetes específiques per afegir contingut semàntic:

- `Itemscope`. Indica què defineix una entitat o contingut en un bloc i informa al navegador que totes les propietats definides dins d'aquest element li pertanyen.
- `Itemtype`. Indica el tipus de contingut. Per exemple, una persona, un esdeveniment, etc.
- `Itemprop`. Indica les propietats per a aquest tipus de contingut.
- `Itemid`. Indica un identificador únic per a un element. Per exemple, un URI o URN.

#### Microdata

A l'enllaç «Com funcionen les dades estructurades» descobrireu com usa Google les microdades en els resultats de cerca.

#### 3.1.1. LocalBusiness

Per facilitar als motors de cerca la ubicació de negocis resulta útil marcar les dades amb microdades que afegeixin més informació i, per tant, que augmentin les possibilitats de mostrar aquesta informació addicional juntament amb el llistat dels resultats de la cerca.

Figura 11. Exemple de microdada per a LocalBusiness

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

Algunes propietats que es poden incloure serien:

- `name`. El nom del negoci.
- `description`. Conté una breu descripció del negoci.
- `image`. Una imatge del negoci o el logotip.
- `url`. Adreça del lloc web.
- `address`. Inclou l'adreça en què, a més, es pot marcar separatament la localitat o el codi postal.
- `map`. Mapa de la localització (per exemple, enllaç amb Google Maps).
- `telephone`. El número de telèfon.
- `openingHours`. L'horari.

#### Propietats

La llista completa de propietats està a <http://schema.org/local-business>.

Figura 12. Microdades Local Business

```

<div itemscope itemtype="http://schema.org/LocalBusiness"></div>
  <span itemprop="name">Exemple Co.</span>
  <p itemprop="description">Breu descripció Exemple Co. </p>
  
  <a itemprop="url" href="https://exemple.co.ca"> Exemple Co.</a>
  <div itemprop="address" itemscope
    itemtype="http://schema.org/PostalAddress">
    <span itemprop="streetAddress">Carrer Carrer 1</span>
    <span itemprop="addressLocality">Ciutat</span>,
    <span itemprop="addressRegion">Província</span>,
    <span itemprop="postalCode">99123</span>
  </div>
  <a itemprop="map" href="http://maps.google.com">Vegeu a mapa</a>
  <span itemprop="telephone">999 012345</span>
  <a itemprop="email" href="mailto:exemple@exemple.co.ca">Correu</a>
  <time itemprop="openingHours" datetime="Mo-Fr 09:00-21:00">
    De dilluns a divendres de 9 a 21 h.
  </time>
</div>

```

### 3.2. Microformats

Els microformats (µF) són una especificació de patrons HTML per publicar dades estructurades sobre conceptes freqüents, com ara persones, esdeveniments, publicacions de blogs o ressenyes.

Els **microformats** són elements de marques semàntiques que utilitzen POSH (*Plain Old Semantic HTML*) amb un conjunt de valors per a les etiquetes `class`, `rel` i `rev`. Els microformats són oberts i estan disponibles lliurement perquè qualsevol els pugui usar.

Taula 1. Alguns tipus de microformats

Nom	Utilitat
hCalendar	Marcar esdeveniments.
hCard	Representa persones, companyies, organitzacions i llocs.
XFN	Representa relacions socials.
hReview	Representa ressenyes de productes, negocis, esdeveniments, etc.
hAtom	Representa entrades de blogs.
rel-license	Representa llicències d'ús.
hResume	Representa el <i>curriculum vitae</i> .
hRecipe	Representa receptes de cuina.
Geo	Representa localitzacions geogràfiques.

#### Microformats

A [microformats.org/wiki/main\\_page](http://microformats.org/wiki/main_page) es poden trobar tots els microformats existents.

El principal inconvenient dels microformats és que no poden representar qual-sevol tipus de dades.

### 3.2.1. Microformat hCalendar

El microformat hCalendar s'utilitza per a entrades de calendari per a esdeveniments esportius, aniversaris, recordatoris, reunions, conferències i altres esdeveniments. La classe principal d'hCalendar és vCalendar i la classe per a esdeveniments és vEvent, que cal per a tots els esdeveniments. Les propietats es representen amb els elements d'hCalendar que necessita les classes dtstart i summary.

dtstart

Hauria d'estar en format de data ISO 8601 per a representacions de data i hora.

Figura 13. Conferència representada a hCalendar

```
<div class="vevent">
  <h1 class="summary">Conferència Dades 2020</h1>
  <div class="description">La Conferència Dades 2020 ha estat
anunciada.</div>
  <div>Publicat :
    <abbr class="dtstamp" title="20200525T000000Z">
      25 de maig de 2020
    </abbr>
  </div>
  <div>Organitzada per:
    <a class="organizer" href="mailto:dades2020@exemple.co.ca">
      dades2020@exemple.co.ca
    </a>
  </div>
  <div>Dates:
    <abbr class="dtstart" title="20200525T093000Z">
      25 de maig de 2020, 9.30 am
    </abbr> -
    <abbr class="dtend" title="20200526T200000Z">
      26 de maig de 2020, 8.00 pm
    </abbr>
  </div>
</div>
```

Com a propietats opcionals, entre d'altres, es podrien afegir location, duration (en format durada de dates ISO), category, description, geo (latitud i longitud), contact, organizer o status.

### 3.2.2. Microformat hCard

El microformat hCard s'utilitza per representar dades de contactes de persones, companyies o organitzacions i es basa en l'estàndard vCard (RFC 2426), per la qual cosa la marca vCard és convertible a hCard.

La classe base d'hCard és vCard i els únics atributs obligatoris d'hCard són fn i n.<sup>1</sup>

vCard

L'estàndard vCard s'utilitza per a les targetes de visita d'empreses. Molts mòbils usen vCard per emmagatzemar els contactes en el dispositiu.

<sup>(1)</sup>Si s'omet n però està definit fn, llavors el valor d'n serà el d'fn.



Figura 14. Targeta de visita amb hCard

```

<link rel="profile" href="http://microformats.org/profile/hcard" />
...
<div id="hcard-Persona-Ficticia" class="vcard">
  
  <a class="url fn" href="http://exemple.co.ca">Persona Fictícia</a>
  <div class="org">Empresa Fictícia</div>
  <a class="email" href="mailto:persona_ficticia@exemple.co.ca">
    persona_ficticia@exemple.co.ca
  </a>
  <div class="adr">
    <div class="street-address">Avinguda Principal</div>
    <span class="locality">Ciutat</span>,
    <span class="region">Província</span>,
    <span class="postal-code">99123</span>
    <span class="country-name">Espanya</span>
  </div>
  <div class="tel">+99 123456789</div>
</div>

```

### 3.2.3. rel-license

Hi ha milions de recursos web amb drets reservats i també molts tipus de llicències associades a documents i objectes. Amb el microformat `rel="license"` es poden afegir hipervincles que apunten a la descripció de la llicència, la qual cosa és especialment útil per a imatges, encara que es pot usar per a qualsevol tipus de recurs.

Figura 15. Imatge amb llicència Creative Commons Attribution-Share Alike

```

<link rel="profile" href="http://microformats.org/profile/rel-license" />
...


```

El valor d'atribut `href` proporciona l'URI associat al recurs que descriu la llicència. Algunes de les llicències més freqüents són:

- Creative Commons Attribution (cc by)
- Creative Commons Attribution Share Alike (cc by-sa)
- Creative Commons Attribution No Derivatives (cc by-nd)
- Creative Commons Attribution Non-Commercial (cc by-nc)
- Creative Commons Attribution Non-Commercial Share Alike (cc by-nc-sa)
- Creative Commons Attribution Non-Commercial No Derivatives (cc by-nc-nd)

### 3.3. RDFa

RDFa (*RDF in Attributes*) permet posar tripletes RDF en HTML com a valors d'atributs. La sintaxi completa (RDFa *Core*) proporciona característiques per expressar dades estructurades complexes com ara relacions humanes, llocs o esdeveniments. Hi ha un subconjunt mínim (RDFa *Lite*) més senzill i adequat per a molts propòsits.

RDFa *Lite* té els atributs següents:

- `vocab`. Defineix el vocabulari en què es basen els elements de marques d'RDFa. Per exemple, `schema.org`.
- `typeof`. Indica de quin tipus és l'element descrit.
- `property`. Proporciona una propietat de l'element.
- `resource`. Permet assignar identificadors als elements.
- `prefix`. Permet prefixos per representar recursos i abreujar.

Amb RDFa la relació entre dos recursos es pot expressar amb l'atribut `rel`, mentre que les relacions inverses s'expressen amb l'atribut `rev`.

Figura 16. RDFa per anotar persones

```
<div vocab="http://xmlns.com/foaf/0.1/" typeof="Person">
  <p>
    <span property="name">Alicia</span>,
    Correu: <a property="mbox" href="mailto:alicia@exemple.co.ca">
      alicia@exemple.co.ca
    </a>,
    Telèfon: <a property="phone" href="tel:+99 12345678">
      +99 12345678
    </a>
  </p>
  <ul rel="knows">
    <li resource="http://exemple.co.ca/benito/" typeof="Person">
      <span property="name">Benito</span>
    </li>
    <li resource="http://exemple.co.ca/carlos/" typeof="Person">
      <span property="name">Carlos</span>
    </li>
  </ul>
</div>
```

Taula 2. Vocabularis d'ús freqüent amb RDFa

Prefix	URI	Vocabulari
cc	http://creativecommons.org/ns#	Creative Commons Rights Expression Language
dcterms	http://purl.org/dc/terms/	Dublin Core Metadata Terms
dc	http://purl.org/dc/elements/1.1/	Dublin Core Metadata Element Set, V. 1.1
foaf	http://xmlns.com/foaf/0.1/	Friend of a Friend (FOAF)
og	http://ogp.me/ns#	Facebook OpenGraph

<b>Prefix</b>	<b>URI</b>	<b>Vocabulari</b>
<b>sioc</b>	<a href="http://rdfs.org/sioc/ns#">http://rdfs.org/sioc/ns#</a>	SIOC Core
<b>vcard</b>	<a href="http://www.w3.org/2006/vcard/ns#">http://www.w3.org/2006/vcard/ns#</a>	vCard in RDF
<b>schema</b>	<a href="http://schema.org/">http://schema.org/</a>	schema.org

RDFa proporciona una API (*Application Programming Interface*) que accedeix al DOM (*Document Object Model*) per extreure i utilitzar dades estructurades a la pàgina web.

## 4. CSV

CSV (sigles de *comma-separated values*) és un arxiu de text que s'utilitza per **transferir dades** d'una aplicació a una altra. Els arxius CSV no tenen un esquema predeterminat. Normalment per CSV (i similars, com TSV, *tab-separated values*) s'exporta des de les taules d'una base de dades relacional. La capçalera i el significat de les columnes solen aparèixer com una fila dins de l'arxiu CSV, encara que es poden ometre.

Les característiques següents són pròpies d'un arxiu CSV:

1) Un caràcter per **separar registres** individuals (normalment retorn del carro CRLF).

Figura 17. Exemple de registres en CSV

```
aaa,bbb,ccc CRLF  
zzz,yyy,xxx CRLF
```

2) Un caràcter per **separar columnes** individuals (comes, tabulacions o espais).

Figura 18. Exemple de camps separats per coma

```
aaa,bbb,ccc
```

3) **Delimitador de camps** que és un caràcter per evitar confusions amb els separadors (normalment cometes dobles o simples).

Figura 19. Exemple de camps delimitats per cometes

```
"aaa","bbb","ccc" CRLF  
"zzz","yyy","xxx" CRLF
```

4) Opcionalment, en la primera línia de l'arxiu pot aparèixer una **capçalera** amb els noms dels camps en el mateix format que les línies de registres.

Figura 20. Exemple de capçalera en CSV

```
camp1,camp2,camp3 CRLF  
aaa,bbb,ccc CRLF  
zzz,yyy,xxx CRLF
```