

Treball Fi de Carrera

Enginyeria Tècnica en Informàtica de Gestió

Curs 2006/2007 – Primer semestre

Alumne: **Ignasi Pinto Iglesias**

Consultor: **Òscar Celma Herrada**

Índex

Índex de figures	3
1. Introducció	4
1.1. Descripció del Treball Fi de Carrera	4
1.2. Objectius del Treball Fi de Carrera	4
1.3. Contingut	5
1.4. Planificació	5
1.5. Productes obtinguts	7
1.6. Estructura de la memòria	7
2. Web Semàntica	9
2.1. Introducció	9
2.2. Estructura de la Web Semàntica	9
2.3. XML (eXtensible Markup Language)	11
2.4. RDF (Resource Description Framework)	13
2.4.1. Esquemes RDF	15
2.5. OWL (Web Ontology Language)	16
3. Sistemes Gestors de Bases de Dades per a documents RDF	18
3.1. Introducció	18
3.2. Estructura i organització dels SGBDs	18
3.2.1. Sesame	20
3.2.2. Jena	23
3.2.3. Redland	25
3.2.4. Kaon	27
3.3. Comparativa entre els SGBDs: Jena vs. Sesame	29
3.3.1. Model RDF	30
3.3.2. Models persistents	31
3.3.3. Inferència	32
3.3.4. Llenguatges de consulta	33
3.3.5. Interoperabilitat Jena – Sesame	34
4. Llenguatge de consulta SPARQL per a RDF	36
4.1. Introducció	36
4.2. Realització de consultes simples	37
4.2.1. Restriccions de valors	38
4.2.2. Nodes en blanc en el resultat de la consulta	39
4.3. Grafs	40
4.3.1. Graf bàsic	40
4.3.2. Grup de Grafs	41
4.4. Informació opcional	41
4.5. Combinacions alternatives	42
4.6. Documents RDF	43
4.6.1. Especificar Document RDF	44
4.6.2. Consultar	46
4.7. Consultes	47
4.7.1. Solucions i resultats	47
4.7.2. Seleccionar variables	49

5. Aplicació pràctica: CercarFOAF	50
5.1. Introducció	50
5.2. Anàlisi i disseny	51
5.2.1. Descripció	51
5.2.2. Programari	52
5.2.3. Funcionalitats	52
5.3. Casos d'ús	53
5.4. Implementació	58
5.4.1. Relació de servlets	58
5.4.2. Disseny de pantalles	59
6. Conclusions i possibles línies de futur	64
7. Bibliografia	66

Índex de figures

Figura 1.1 - Diagrama de Gantt del TFC	6
Figura 2.1 - Estructura en capes de la Web Semàntica	9
Figura 2.2 - Relació SGML / XML / HTML	12
Figura 2.3 - Representació del format d'una sentència RDF	14
Figura 2.4 - Representació sentències RDF	14
Figura 2.5 - Jerarquia de classes d'RDF-Schema	15
Figura 3.1 - Arquitectura integrada d'SGBDR	19
Figura 3.2 - Graf d'RDF	19
Figura 3.3 - Arquitectura Sesame	21
Figura 3.4 - Sesame en el seu context	22
Figura 3.5 - Arquitectura Jena2	23
Figura 3.6 - Graf contenidor Grafs especialitzats sobre taules	24
Figura 3.7 - Classes de Redland	25
Figura 3.8 - Redland Model Layers	26
Figura 3.9 - Graf Redland RDF amb context	27
Figura 3.10 - Arquitectura KAON	27
Figura 3.11 - Implementació KAON API	28
Figura 3.12 - Esquema per emmagatzemament d'RDF	29
Figura 3.13 - Arquitectura Sesame 2	30
Figura 3.14 - Jena2 RDF API: Interfícies principals	30
Figura 3.15 - Sesame. Diagrama de classes pel model RDF	31
Figura 3.16 - Sesame 2 Repository API	32
Figura 3.17 - Estructura de la maquinària d'inferència en Jena	32
Figura 5.1 - Logo de FOAF	52
Figura 5.2 - Diagrama de casos d'ús de l'aplicació web	54

Capítol 1

Introducció

1.1 Descripció del Treball Fi de Carrera

La Web Semàntica proporciona una plataforma comú que permet compartir i reutilitzar les dades a través de les aplicacions, empreses i comunitats. Té com a objectiu crear un seguit de llenguatges expressius per a l'intercanvi d'informació significativa (o semàntica), d'una forma comprensible per a les màquines, del contingut dels documents de la Web. La Web Semàntica és una extensió de la web actual en què la informació aporta un significat ben definit, permetent així una millor cooperació entre persones, i sobretot entre màquines.

El projecte es basa en estudiar i avaluar diferents sistemes gestors de bases de dades (SGBDs) per desar informació dins el context de la Web Semàntica. Els SGBDs hauran de tractar, emmagatzemar i gestionar la informació classificada segons uns criteris semàntics i interrelacionada amb conceptes afins, alhora que permetin la comunicació entre sistemes de manera transparent a l'usuari.

La informació a la Web Semàntica es codifica seguint la notació RDF (*Resource Description Framework*). El llenguatge RDF, desenvolupat pel World Wide Web Consortium (W3C), permet la interoperabilitat entre aplicacions mitjançant l'intercanvi d'informació la través de la Web. L'RDF defineix un model de dades que suporta la fàcil i ràpida integració de diferents fonts d'informació utilitzant conceptes semàntics.

En aquest projecte s'estudiarà el llenguatge de consultes SPARQL (*Simple Protocol and RDF Query Language*), així com els diferents SGBDs que el suporten. SPARQL és un llenguatge de consulta per a informació representada en RDF. El protocol SPARQL per a RDF descriu l'accés remot de dades i la transmissió de consultes dels clients als processadors.

Com a cas d'estudi concret, es proposa que el conjunt de dades sobre els quals es realitzin les consultes SPARQL siguin dades relacionades amb el projecte FOAF (*Friend of a Friend*). L'objectiu de la iniciativa FOAF és descriure la informació referent a les persones i les relacions que s'estableixen entre elles, destinada a ésser processada a l'entorn web.

1.2 Objectius del Treball Fi de Carrera

Els objectius principals del projecte són, per una banda, analitzar el funcionament de les bases de dades que suporten RDF i, d'altra banda, estudiar el llenguatge de consulta estàndard SPARQL que ha estat desenvolupat per W3C.

En concret, els objectius que es volen assolir amb el projecte són els següents:

- Conèixer l'estructura i organització dels SGBDs que treballen amb informació basada en RDF.
- Avaluar l'adequació d'utilització dels SGBDs per desar i recuperar descripcions en RDF.
- Anàlisi comparatiu exhaustiu dels diferents SGBDs estudiats a l'objectiu anterior.
- Estudi del llenguatge de consulta SPARQL.

- Realització de casos pràctics d'aquestes tecnologies, implementant una aplicació web senzilla que permeti fer cerques sobre documents RDF, amb dades relacionades amb l'iniciativa FOAF.

Per últim, en l'elaboració de la memòria del treball es volen assolir els objectius d'agrupar, seleccionar, jerarquitzar i ordenar tota la informació i documentació, per tal que el conjunt del text tingui una estructura coherent i cohesionada.

1.3 Contingut

En funció dels objectius definits a l'apartat anterior es detalla a continuació la previsió d'apartats i subapartats de la memòria. Cal esmentar que el capítol tercer és una breu introducció de l'estructura de la Web Semàntica, així com del llenguatge XML. Els següents punts són provisionals, i s'aniran adequant en funció de l'evolució del projecte.

M	1. Índex	2 planes
	2. Introducció	4 planes
	3. Web Semàntica	
	3.1. Estructura de la Web Semàntica	2 planes
	3.2. XML	2 planes
	3.3. RDF	2 planes
	3.4. OWL	2 planes
	4. SGBDs i documents RDF	
E	4.1. Estructura i organització dels SGBDs	5 planes
	4.2. Emmagatzemant de documents RDF	6 planes
	4.3. Comparativa entre els SGBDs	6 planes
M	5. Llenguatge de consulta SPARQL per a RDF	
	5.1. Introducció	1 plana
Ò	5.2. Realització de consultes simples	4 planes
	5.3. Treball amb literals RDF	4 planes
	5.4. RDF Database	4 planes
	5.5. Consultes	4 planes
	6. Aplicació pràctica	
R	6.1. Anàlisi i Disseny	3 planes
	6.2. Casos d'us	3 planes
	6.3. Implementació	2 planes
I	7. Conclusions	2 planes
	8. Bibliografia	2 planes
A		

1.4 Planificació

En aquest punt es presenten de forma resumida les tasques planificades, així como les fites establertes. Cal tenir present que la descripció amb detall de les tasques ha estat objecte del pla de treball específic [1].

La llista de tasques identificades és la següent:

- Definició del projecte.
- Breu introducció a la Web Semàntica, XML, RDF i OWL.
- Estudi dels sistemes de gestors de bases de dades
- Estudi del llenguatge de consultes SPARQL.
- Productes resultants.

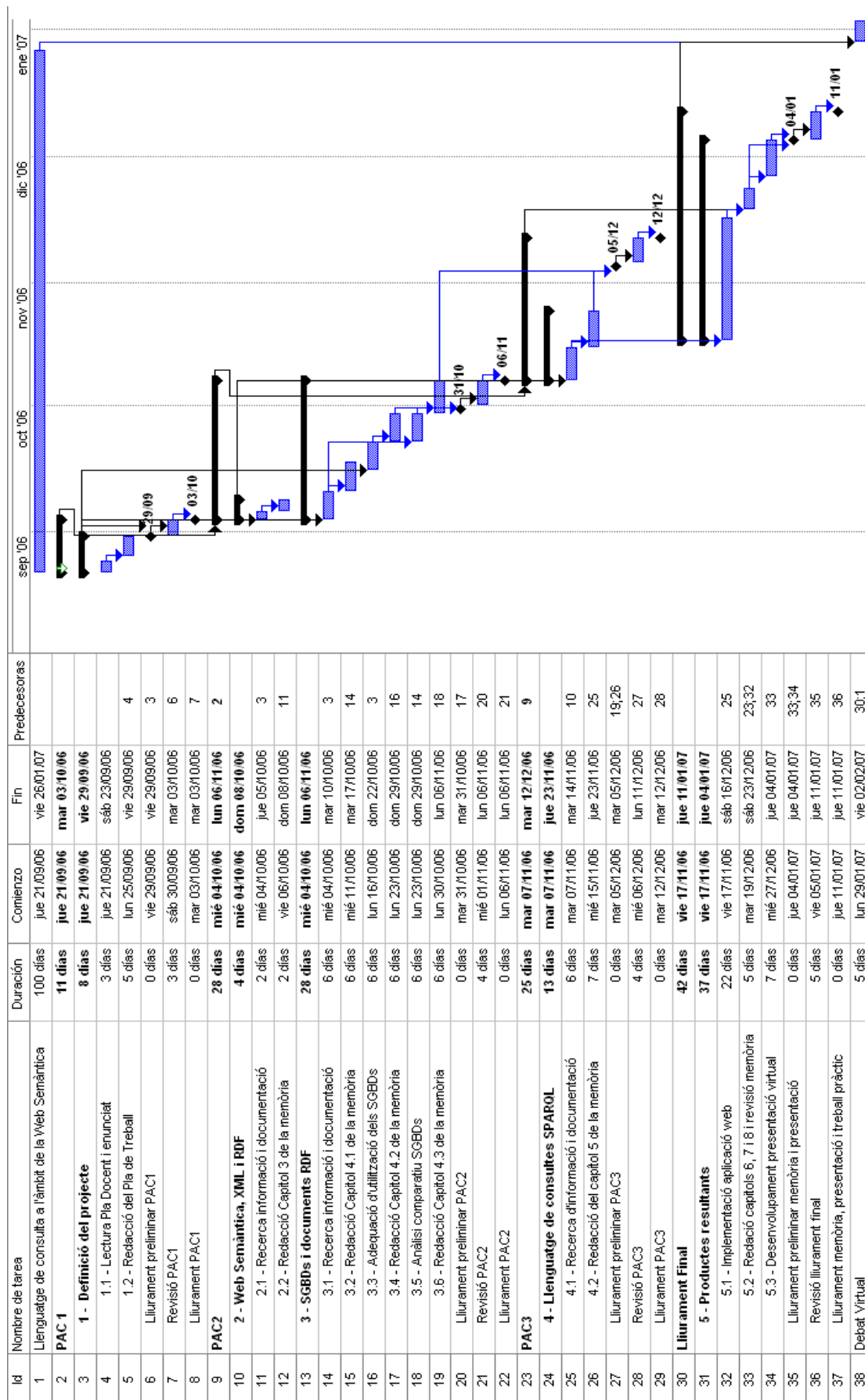


Figura 1.1 - Diagrama de Gantt

A més de les dates claus prèviament definides en el Pla Docent de l'assignatura, i amb l'objectiu d'assolir els objectius fixats es farà un lliurament preliminar al consultor, per tal que faci els comentaris i correccions pertinents.

Fita	Contingut	Data
<i>Lliurament preliminar de la PAC1</i>	Esborrany del pla de treball	29/09/2006
<i>Lliurament de la PAC1</i>	Pla de treball	03/10/2006
<i>Lliurament preliminar de la PAC2</i>	Esborrany dels capítols 1, 2, 3, 4.1 i 4.2 de la memòria	31/10/2006
<i>Lliurament de la PAC2</i>	Capítols 1, 2, 3, 4.1 i 4.2 de la memòria	06/11/2006
<i>Lliurament preliminar de la PAC3</i>	Esborrany del capítol 4.3 i 5 de la memòria	05/12/2006
<i>Lliurament de la PAC3</i>	Capítol 4.3 i 5 de la memòria	12/12/2006
<i>Lliurament preliminar de la memòria i de la presentació</i>	Esborrany de la memòria complerta del treball i de la presentació	04/01/2007
<i>Lliurament final de la memòria, de la presentació i del treball pràctic</i>	Memòria complerta del treball, de la presentació i del producte del cas pràctic	11/01/2007
<i>Inici Debat Virtual</i>	El Tribunal d'avaluació adreçarà, si és necessari, preguntes als estudiants	22/01/2007
<i>Final Debat Virtual</i>		26/01/2007

1.5 Productes obtinguts

Al finalitzar aquest projecte els productes obtinguts són el següents:

- **Memòria del projecte.** És aquest document. És lliurarà en format *Word*.
- **Presentació del projecte.** Es tracta d'una presentació del contingut del treball en format de diapositives per a la seva exposició. Es lliurarà en format *PowerPoint*.
- **Productes del cas pràctic.** Es tracta dels fitxer produïts amb el programari utilitzat per dissenyar i implementar una aplicació web que permeti

1.6 Estructura de la memòria

La memòria està estructurada amb l'objectiu de proporcionar una aproximació gradual al objectius principals del projecte, per una banda, el funcionament de les bases de dades que suporten RDF i, d'altra banda, estudiar el llenguatge de consulta estàndard SPARQL que ha estat desenvolupat per W3C.

Així, en el capítol 2 es fa una introducció a la Web Semàntica, amb una explicació de la seva estructura i dels seus components, els llenguatges XML, RDF i OWL, els quals permeten dotar a la web d'una infraestructura global per a la compartició de dades i documents.

En el capítol 3 s'analitzen els sistemes gestors de bases de dades que tracten, emmagatzemen i gestionen la informació basada en RDF. Es proporciona una visió de la seva estructura i organització.

En el capítol 4 es presenta el cas d'estudi d'aquest treball, el llenguatge SPARQL. Aquest és un llenguatge de consulta per a informació representada en RDF. S'expliquen la seva aportació a la web semàntica i els conceptes que han de permetre conèixer i comprendre tots els fonaments d'aquest llenguatge.

En el capítol 5 es descriu la realització del cas pràctic, una aplicació web que permet fer cerques sobre documents, amb dades relacionades amb l'iniciativa FOAF, tant des del punt de vista del seu disseny com de la seva implementació. El projecte FOAF és una aplicació que pretén descriure la informació referent a les persones –dades personals, vincles entre les persones, els interessos, etc.- destinada a ser processada a l'entorn Web

Finalment, en el capítol 6 es presenten les conclusions finals i possibles línies de futur.

Capítol 2

Web Semàntica

2.1 Introducció

La Web Semàntica és una Web estesa i dotada de major significat, recolzada en llenguatges universals, que permetran que els usuaris (i les aplicacions) puguin trobar respostes a les seves preguntes de forma més ràpida i senzilla, gràcies a una informació més ben definida. Amb aquesta Web, els usuaris podran delegar tasques en el programari que serà capaç de processar el contingut de la informació, raonar amb aquest, combinar-lo i realitzar deduccions lògiques per resoldre automàticament problemes quotidians.

Els principis bàsics de la Web Semàntica definits pel W3C [3] són els següents:

- Qualsevol cosa pot ser identificada mitjançant un URI.
- Els recursos i els enllaços poden tenir un tipus concret.
- La informació parcial és permesa.
- No hi ha una necessitat d'una veritat absoluta.
- Cal tenir en compte l'evolució de la informació.
- Disseny simplista.

2.2 Estructura de la Web Semàntica

L'objectiu de la Web Semàntica és que la Web passi de ser una col·lecció de documents a convertir-se en una base del coneixement. El desenvolupament de la Web Semàntica s'estructura en diverses capes o nivells. Aquesta estructura es pot observar a la figura 2.1.

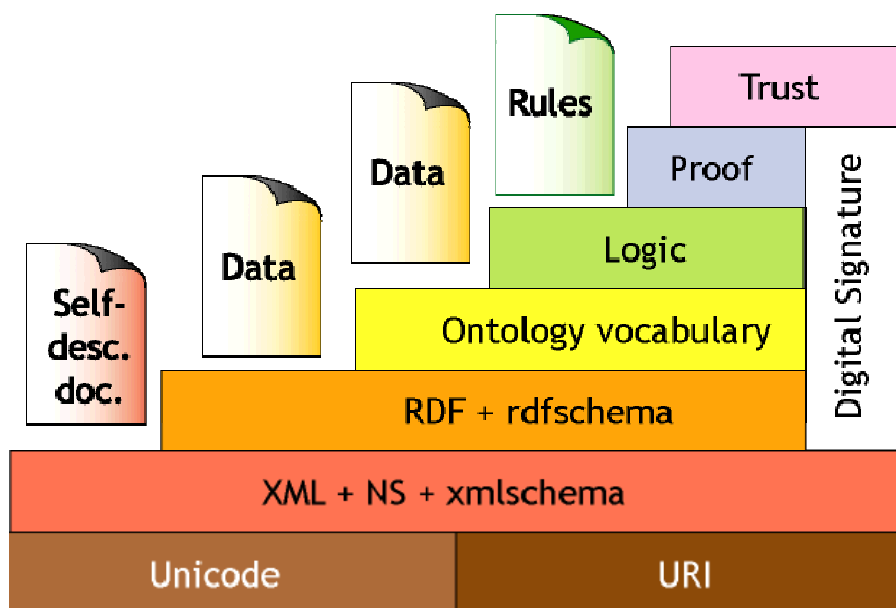


Figura 2.1 – Estructura en capes de la Web Semàntica [2]

A l'esquema de la figura es pot distingir el següent:

El nivell més bàsic de la Web Semàntica s'identifiquen els recursos. Amb l'*URI (Unified Resource Identifier)* els recursos -ja siguin documents, imatges, arxius descarregables, serveis o inclús una entitat externa a la web com una persona-, es poden identificar de manera global en la web. La sintaxi d'un URI és similar a la d'un URL (*Unified Resource Locator*). Amb l' *Unicode* s'assegura que es faran servir conjunts de caràcters internacionals.

En el nivell següent es defineix la sintaxi. El llenguatge XML (*eXtensible Markup Language*) és un metallenguatge que estructura la informació mitjançant etiquetes, permetent descriure altres llenguatges. Mitjançant XML Schema es defineixen les restriccions d'aquests llenguatges. I per agrupar els URIs pertanyents a cert domini d'aplicació, l'XML preveu la creació de Namespaces (*espais de noms*).

El nivell *RDF (Resource Description Framework)* i *rdfschema* permet descriure els recursos que la Web Semàntica preveurà, relacionant-los amb altres recursos. L'RDF es basa en un model de graf per representar dades en la Web. Permet intercanviar informació a través de diferents aplicacions sense que aquestes dades perdin significat, la qual cosa facilita la reutilització dels recursos en la Web. Igualment és possible la consulta d'aquestes dades mitjançant llenguatges de consulta.

El nivell d'*Ontologies* suporta l'evolució dels vocabularis definits al nivell inferior, ja que permet la definició de relacions entre diferents conceptes. Les ontologies proporcionen tot el coneixement necessari sobre la naturalesa d'un domini perquè una màquina pugui raonar sobre l'esmentat domini.

Una de les coses que fan molt poderosa una ontologia són les regles d'inferència, que corresponen al següent nivell *Logic*.

Finalment, un nivell de seguretat, *Digital Signature* permet assignar nivells de fiabilitat a determinats recursos, comprovant que la informació prové d'una font de confiança (nivells *Proof* i *Trust*).

La Web Semàntica es construeix a través de canvis incrementals, portant als documents i a dades ja disponibles en la web actual a descripcions llegibles per màquines. XML, RDF i OWL permeten a la web ser una infraestructura global per a la compartició de dades i documents, la qual cosa fa més fàcil i fiable la recerca i reutilització d'informació.

- **L'XML:** Proporciona les regles i la sintaxi per a la creació de documents estructurats. Aquest llenguatge no imposa cap tipus de restricció semàntica al significat dels documents creats i, per tant, necessita l'aportació d'altres eines per tal de poder proporcionar un sentit semàntic a una determinada informació.
- **L'RDF:** Facilita una infraestructura de dades que permet conferir semàntica a la web. És un estàndard per realitzar descripcions senzilles d'informació. Els esquemes RDF proporcionen una manera de combinar aquestes descripcions i formar un vocabulari únic, que defineix cadascuna de les propietats i relacions d'un recurs determinat.
- **L'OWL:** Ofereix la gestió d'ontologies dins la web. Proporciona un llenguatge per a la definició d'ontologies estructurades basades en la web, que ofereix una integració i interoperabilitat de dades entre diverses comunitats descriptives. L'OWL suposa un gran pas endavant en la representació i organització del coneixement en la web, i per tant, per a la creació de la Web Semàntica.

Per tant, l'objectiu de la Web Semàntica és que la web actual passi de ser una col·lecció de documents a convertir-se en una base de coneixement.

2.3 XML (*eXtensible Markup Language*)

El llenguatge extensible de marques XML és un metallenguatge, és a dir, una llenguatge que permet descriure altres llenguatges. Aquests llenguatges generats en XML estaran estructurats i seran fàcilment processables per ordinadors.

La definició actual d'XML [4] és una recomanació del W3C i està basat en l'estàndard SGML. XML s'ha creat per enriquir l'estructura dels documents que poden ser utilitzats a la Web, donat que les altres alternatives viables, HTML (orientat a la presentació de la informació) i SGML (massa complex), no eren massa pràctiques per a aquest propòsit.

Segons l'especificació, els objectius de dissenyar l'XML van ser:

- XML ha de ser utilitzable directament a Internet
- XML ha de suportar una gran quantitat d'aplicacions
- XML ha de ser compatible amb SGML
- Ha de ser fàcil escriure programes de processat de documents XML
- Mantenir mínim el nombre de característiques opcionals a ML, idealment zero
- Els documents XML han de ser llegibles per persones i raonablement clars
- El disseny d'XML ha de ser ràpid
- El disseny d'XML ha de ser formal i concís
- Els documents XML han de ser fàcils de crear
- No es gens important l'abreviació en les marques XML

Com assenyala el W3C, els documents XML són llegibles i generables per persones i per màquines. Els conceptes bàsics d'XML [5] es poden resumir en els punts següents.

XML és per estructurar dades

L'XML és un llenguatge que permet jerarquitzar i estructurar la informació i descriure els continguts dins el propi document, així com la reutilització de parts del mateix. La informació estructurada presenta diversos continguts (text, imatges, àudio, etc.) i formes: fulls de càlcul, llibretes d'adreces, paràmetres de configuració, dibuixos tècnics, etc. L'XML és un conjunt de regles per a dissenyar documents de text, les quals permeten l'estructuració de les dades, amb una estructura d'arbre. L'XML facilita que l'ordinador generi dades, en llegeixi, o s'asseguri que l'estructura de les dades no és ambigua. L'XML evita els problemes típics del disseny de llenguatges: és extensible, independent de la plataforma, i suporta la internacionalització i la traducció.

XML s'assembla una mica a HTML

En HTML les etiquetes semàntiques i el conjunt d'etiquetes són fixes. Un `<h1>` és sempre un primer nivell d'encapçalament. Igual que l'HTML, l'XML fa servir marques, que són paraules entre '`<`' i '`>`', i atributs; però mentre l'HTML especifica què vol dir cada etiqueta i cada atribut i, sovint, quina aparença tindrà el text en un navegador, l'XML usa les etiquetes només per delimitar porcions de les dades, i deixa la interpretació de les dades completament en mans de l'aplicació que les llegeix. Així, una etiqueta com `<p>` pot ser un paràgraf, persona, etc. depenent del context.

XML és text, però no està pensat per a ser llegit

Els programes que produeixen "dades estructurades" poden guardar aquestes dades en el disc usant un format binari o un format de text. Un avantatge dels formats de text és que permet veure les dades sense el programa que les ha produït. El llenguatge XML és basa en el llenguatge *Unicode* (amb un conjunt de caràcters de 16 bits, més que el format ASCII)

L'XML consisteix d'una sèrie de regles, pautes o convencions per planificar formats de text per aquestes dades, de manera que produeixin arxius que siguin fàcilment generats i llegits per un ordinador, que siguin inequívocs i que evitin els problemes més comuns como la falta d'extensibilitat, la falta d'interoperabilitat entre plataformes o la falta de suport per universalitzar el seu tractament. Els arxius XML són arxius de text, es poden visualitzar. Es pot usar un editor de text per programar XML, però qualsevol errada o oblit d'una marca deixarà inservible l'esmentat arxiu. El llenguatge XML és més estricte que l'HTML, ja que els navegadors no són tant restrictius a l'hora d'analitzar la validesa del document HTML.

XML és una família de tecnologies

Les tecnologies XML són un conjunt de mòduls que ofereixen serveis útils a les peticions més freqüents dels usuaris. Entre les tecnologies disponibles es poden destacar:

- [XSL](#) (*Extensible Stylesheet Language*). Defineix el full d'estils estàndard per a XML. Funciona com un llenguatge avançat per a crear fulls d'estil. És capaç de transformar, ordenar i filtrar dades XML, i donar-les format basant-se en el seus valors.
- [XPath](#) (*XML Path Language*). És un llenguatge per accedir a parts d'un document XML. Identifica parts d'un document concret, com poden ser els seus atributs, elements, etc.
- [XLink](#) (*XML Linking Language*). És un llenguatge que permet inserir elements en documents XML per crear enllaços entre recursos XML. Permet crear vincles bidireccionals, la qual cosa implica la possibilitat de moure's en dues direccions. Això facilita l'obtenció d'informació remota com recursos en lloc de simplement com pàgines Web.
- [XPointer](#) (*XML Pointer Language*). És un llenguatge que permet l'accés a l'estructura interna d'un document XML, és a dir, als seus elements, atributs i contingut.
- [XQuery](#) (*XML Query*). Llenguatge de consulta XML que proporciona una manera flexible de consulta per extreure dades dels documents en la Web.

XML és nou, però no tant

XML va sorgir com un llenguatge de marques per substituir a HTML. Ambdós llenguatges són hereus d' SGML, el llenguatge de marques estàndard per a la descripció formal i de contingut dels documents, no només per a la presentació d'aquests documents. El desenvolupament d'XML va començar a l'any 1996 i des de llavors ha tingut un desenvolupament exponencial. Amb la intenció de millorar l'HTML i prenent com a punt de partida l' SGML, però simplificant-lo per poder treballar en la web, es va crear XML, que ha estat una norma W3C des de febrer de l'any 1998. La primera definició d' XML va ser: "*sistema per definir, validar i compartir formats de documents en la Web*".

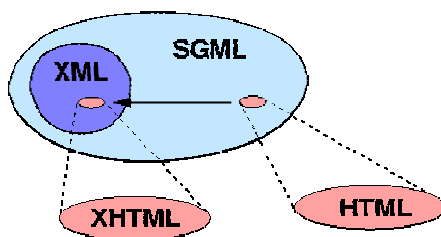


Figura 2.2 - Relació SGML / XML / HTML

XML converteix HTML en XHTML

HTML té la seva pròpia especificació basada en l'XML, la del llenguatge XHTML (*eXtensible Hypertext Markup Language*) que és, en realitat, un pas intermedi de la migració d'HTML

cap a XML. El llenguatge XHTML té molts dels elements d'HTML, i és un llenguatge que permet separar la presentació del contingut web, fent aquest compatible amb XML i permetent crear noves etiquetes. Un document que està basat en XML hereta la seva sintaxi i la restringeix en determinats aspectes (per exemple, XHTML permet '<p>', però no '<r>'); també afegeix significat a aquesta sintaxi (en XHTML '<p>' vol dir 'paràgraf', no 'preu', ni 'persona', ni cap altra cosa).

XML és modular

XML permet definir un nou format de document tot combinant i reutilitzant d'altres formats. Tanmateix, combinar elements definits de forma independent pot donar lloc a ambigüitats en els noms. Per a resoldre això, XML ofereix un sistema de espai de noms (*Namespace*), que és una col·lecció de noms identificats mitjançant un URIs, que és prefix de tots els noms continguts en l'esmentat Namespace. XSL i RDF són exemples de formats basats en l'XML que usen espais de noms.

XML Schema [6] permet definir elements i atributs imposant també restriccions pels valors que poden prendre. Els XML Schemas, desenvolupats per Microsoft i d'ús lliure, són l'aposta del W3C per definir documents XML en la Web Semàntica. Algunes característiques que motiven això són:

- Són documents XML
- Permeten definir tipus de dades
- Són extensibles

El llenguatge XML, juntament amb XML Schema i els Namespaces, ofereixen una base estàndard, senzilla i extensible per començar a construir la Web Semàntica.

```
<?xml version="1.0" encoding='iso-8859-1' ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Person>
    <foaf:name>Dan Brickley</foaf:name>
    <foaf:mbox_sha1sum>241021fb0e6289f92815fc210f9e9137262c252e</foaf:mbox_sha1sum>
    <foaf:homepage rdf:resource="http://rdfweb.org/people/danbri/" />
    <foaf:img rdf:resource="http://rdfweb.org/people/danbri/mugshot/danbri-small.jpeg" />
  </foaf:Person>

</rdf:RDF>
```

Exemple 1 – Document XML amb 3 declaracions de Namespaces

2.4 RDF (*Resource Description Framework*)

RDF [7] defineix un model que permet la descripció de recursos i el processament de metadades – *dades que descriuen unes altres dades* -. RDF no és un llenguatge en sí mateix, sinó un model que podem emprar-ho en diferents llenguatges. La sintaxi bàsica d'RDF definida a la seva especificació és XML, i és per això que parlem d'RDF/XML com a llenguatge semàntic; cal esmentar que hi ha d'altres maneres per representar RDF, com per exemple N3, N-Triples o Turtle. Aquestes sintaxis d'RDF s'han gestionat pel W3C a través del [RDF Core](#) (*RDF Core Working Group*).

El model RDF [8] el constitueixen tres tipus d'objectes:

- **Recursos** (*Resources*): Totes les coses descrites per expressions RDF es denominen recursos. Un recurs pot ser una pàgina web completa, una part d'una plana web, una col·lecció de planes web, un objecte que no sigui directament accessible via web, etc. Cada recurs utilitzat ha d'estar identificat inequívocament mitjançant un URI. L'extensibilitat dels URIs permet la utilització d'indicadors per qualsevol entitat imaginable.
- **Propietats** (*Properties*): Una propietat és qualsevol característica, atribut o relació utilitzat per a descriure un recurs. A cada propietat se li assigna un valor que li permet tenir un significat específic, definir els valors permesos, limitat els tipus de recursos que pot descriure, així com determinar les seves relacions amb altres propietats.
- **Sentències** (*Statement*): Al conjunt format per un recurs específic, una propietat determinada i el valor de l'esmentada propietat per aquest recurs se l'anomena *sentència RDF*. Les sentències RDF són les **triples** SUBJECTE – PREDICAT - OBJECTE. La representació gràfica d'una sentència es fa utilitzant nodes i arcs: el·lipsis per representar els subjectes, rectangles per representar objectes i fletxes per representar predicats.

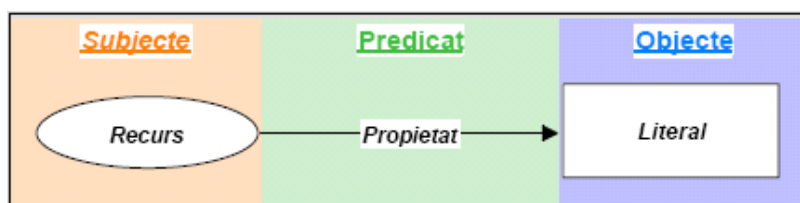


Figura 2.3 - Representació del format d'una sentència RDF

En RDF el Subjecte d'una sentència és un recurs, el qual s'identifica mitjançant URIs. Les propietats, o els predicats de les sentències, també son recursos. L'objecte de la sentència pot ser un valor simple, como string, enter, flotant, data, etc., els quals s'anomenen *literals*; a més l'objecte també pot ser un recurs, essent subjecte d'una altra sentència i formant així l'extens graf del coneixement que pretén el model RDF.

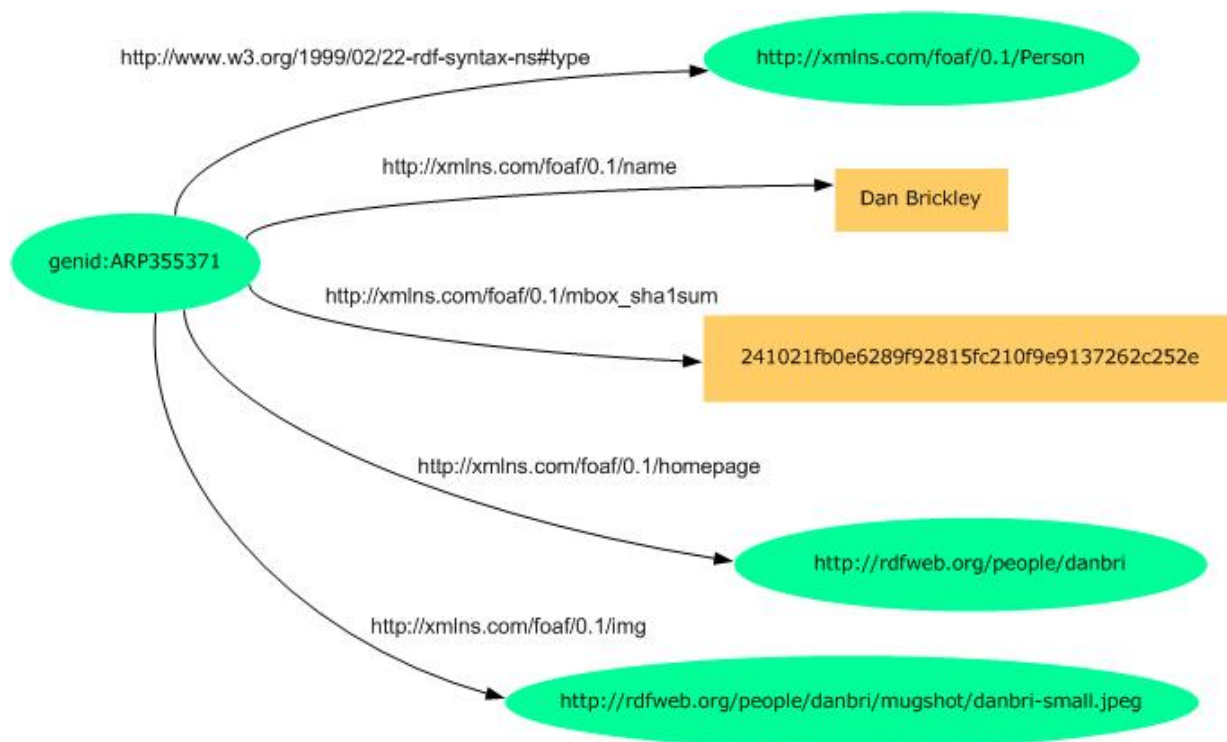


Figura 2.4 - Representació de 5 sentències RDF diferents, amb el mateix Subjecte


```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

  <rdfs:Class rdf:ID="AgentProfileDocument">
    <rdfs:label xml:lang="en">AgentProfileDocument</rdfs:label>
    <rdfs:comment xml:lang="en">A document with a foaf:Agent as
foaf:primarySubject</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
  </rdfs:Class>

  <rdf:Description rdf:about="http://xmlns.com/foaf/0.1/PersonalProfileDocument">
    <rdfs:subClassOf rdf:resource="#AgentProfileDocument"/>
  </rdf:Description>

  <rdf:Property rdf:ID="knownBy">
    <rdfs:label xml:lang="en">knownBy</rdfs:label>
    <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#seeAlso"/>
    <rdfs:comment xml:lang="en">inverse of foaf:knows</rdfs:comment>
    <owl:inverseOf rdf:resource="http://xmlns.com/foaf/0.1/knows"/>
    <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
  </rdf:Property>

  <rdf:Property rdf:ID="personalProfileDocument">
    <rdfs:label xml:lang="en">agentDescriptionDocument</rdfs:label>
    <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#seeAlso"/>
    <rdfs:subPropertyOf rdf:resource="#agentDescriptionDocument"/>
    <rdfs:comment xml:lang="en">Indicates a document which has the subject as
primary- topic and is a foaf:PersonalProfileDocument</rdfs:comment>
    <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
  </rdf:Property>

</rdf:RDF>

```

Exemple 2 - RDF descrivint perfils de persones

2.5 OWL (Web Ontology Language)

La funció d'una ontologia és proporcionar una manera de desenvolupar vocabularis específics per a un domini. Una ontologia defineix els termes utilitzats per a descriure i representar una àrea de coneixement, i són utilitzades per les persones, base de dades i aplicacions que necessiten compartir un tipus d'informació específica sobre un determinat assumpte o domini. Per exemple, es poden crear ontologies sobre medicina, immobiliàries, reparació de cotxes, administració financera, agències de viatges, etc. [12].

L'OWL és un llenguatge ontològic especialment pensat per al seu ús a la Web. És la recomanació del W3C per a l'expressió d'ontologies a la Web Semàntica [13]. En el document "[OWL Web Ontology Language Overview](#)" hi ha explicacions dels requisits d'aquest llenguatge i dels objectius que es volen assolir en el seu desenvolupament.

L'OWL es construeix sobre l'RDF, i també sobre els esquemes RDF, afegint a aquestes tecnologies un ampli vocabulari per a la descripció de classes i propietats: relacions entre classes, cardinalitat, igualtat, classes enumerades, etc. Podem dir que l'OWL defineix un nou univers paral·lel al d'RDF; aquest univers OWL està dividit en tres parts: els *individus* OWL, les *classes* OWL i les *propietats* OWL. Per representar cadascú d'aquest tres conjunts es defineixen classes.

L' OWL proporciona tres subllenguatges dissenyats per a ser utilitzats específicament per diverses comunitats de desenvolupadors i usuaris [14]. Són els següents:

- **OWL Lite.** Està pensat per proporcionar un suport bàsic als usuaris. Permet definir i declarar estructures jeràrquiques i aplicar restriccions simples. Dels tres llenguatges és el que ofereix una complexitat més baixa.
- **OWL DL o Description Logic.** Està destinat als usuaris que desitgen la màxima expressivitat del llenguatge, però sense perdre capacitat de concreció i definició per part dels sistemes que l'utilitzin.
- **OWL Full.** És la versió més complexa de les tres, i està destinada als usuaris que també volen la màxima expressivitat, i a més, la llibertat sintàctica que oferia l'RDF.

```
<!-- here we see a FOAF group described.
      each foaf group may be associated with an OWL definition
      specifying the class of agents that constitute the group's membership -->
<foaf:Group>
  <foaf:name>ILRT staff</foaf:name>
  <foaf:membershipClass>
    <owl:Class rdf:about="http://ilrt.example.com/groups#ILRTStaffPerson">
      <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="http://xmlns.com/foaf/0.1/workplaceHomepage"/>
          <owl:hasValue rdf:resource="http://www.ilrt.bris.ac.uk/">
        </owl:Restriction>
      </rdfs:subClassOf>
    </owl:Class>
  </foaf:membershipClass>
</foaf:Group>
```

Exemple 3 - Definició OWL que especifica la classe d'agents que pertanyen al grup FOAF descrit

Capítol 3

Sistemes Gestors de Bases de Dades per a documents RDF

3.1 Introducció

La informació que conté la Web Semàntica ha de poder ser processada automàticament per les màquines, a més de per les persones. La informació que es diposita a la Web ha d'estar d'una forma estructurada, i a la informació ja existent s'hauria d'afegir la informació estructurada necessària per a permetre el processament automàtic per les màquines. Per tant, és necessària una interoperabilitat semàntica entre les diferents aplicacions de la Web Semàntica.

Les aplicacions, partint d'una ontologia prèviament definida, permeten incloure la informació estructurada dins de la pàgina Web. La informació s'inclou normalment en el llenguatge RDF, promogut pel W3C. Per a la definició d'ontologies està el llenguatge OWL. I per a la consulta de la informació el W3C està treballant actualment en la definició d'un llenguatge de consulta estàndard, SPARQL. Aquesta informació no s'emmagatzema dins de la mateixa pàgina Web, sinó de forma externa en un repositori destinat específicament a mantenir les anotacions. Aquests són les **bases de dades RDF**, que es poden implementar sobre sistemes gestors de bases de dades relacionals (SGBDR).

Les dades de la Web Semàntica estan destinades a ser llegides per les màquines i són fàcils de reutilitzar per tenir una forma d'autodescripció de les dades. L'autodescripció és en forma d'esquema que es pot definir junt amb les dades. Aquesta informació s'ha de guardar amb les dades. Tots els sistemes d'emmagatzematge permeten consultar totes les bases de dades, però els sistemes d'emmagatzematge de la Web Semàntica també han de proporcionar accés per altres mètodes a l'esquema emmagatzemat i a la procedència de la informació que podria estar disponible.

3.2 Estructura i organització dels SGBDs

Un Sistema de Gestió de Bases de Dades és un programa o aplicació capaç de gestionar adequadament les bases de dades. Actualment quasi tots els SGBDs es basen en el model relacional. Aquest model relacional està basat en la utilització d'una sèrie de taules que emmagatzemen la informació relativa a un conjunt de dades relacionades; les taules s'organitzen en files i columnes que defineixen els registres i els camps, i es poden relacionar-se entre elles a partir de camps comuns, la qual cosa garanteix la seva versatilitat.

Un SGBDR (Sistema Gestor de Base de Dades Relacional) emmagatzema la informació en taules organitzades lògicament que s'enllacen definint relacions i contenen dades. El llenguatge de consulta [SQL](#) (*Structured Query Language*), estandarditzat per l'ISO, proporciona la recuperació i gestió d'aquestes dades.

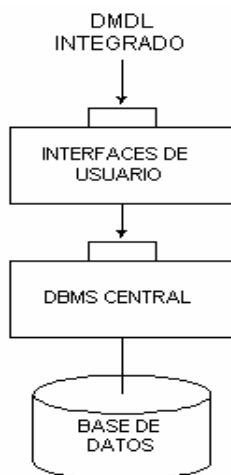


Figura 3.1 – Arquitectura integrada d' SGBDR

De tots els SGBD ens interessen els que treballen amb informació basada en RDF i, per tant, tracten, emmagatzemen i gestionen la informació classificada segons uns criteris semàntics i interrelacionada amb conceptes afins, alhora que permeten la comunicació entre sistemes de manera transparent a l'usuari.

En aquest treball analitzarem alguns d'aquests SGBD i que a més suporten a l'actualitat el llenguatge de consultes SPARQL. Concretament, els SGBD que avaluarem són els següents:

- Sesame
- Jena
- Redland
- Kaon

Emmagatzematge de documents RDF

El bloc bàsic d'un document RDF es basa en una *triple*: subjecte, predicat i objecte. On aquest objecte pot ser també subjecte o objecte d'una altra *triple*, i així els grafs poden niats com també encadenats. RDF i les seves especificacions relacionades, RDF Schema i OWL proporcionen algunes propietats bàsiques com a tipus, classe, subclasse, etc.

```
:dan foaf:mbox <mailto:connolly@w3.org>.
:dan foaf:name "Dan Connolly".

:daniel foaf:mbox <mailto:connolly@w3.org>.
:daniel foaf:name "Daniel W. Connolly".
```

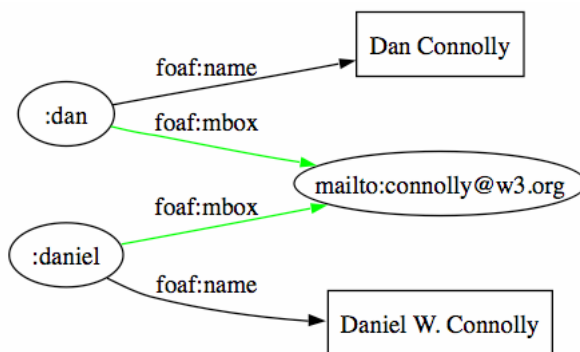


Figura 3.2 - Graf RDF

Necessitem ser capaços d'emmagatzemar persistentment i tractar gran quantitats de dades RDF, aquest es caracteritza per ser un model de dades molt flexible i dinàmic. Els recursos poden adquirir propietats i tipus en qualsevol moment, sense tenir en compte el tipus de recurs o de propietat. Aquesta flexibilitat fa que RDF sigui una tecnologia atractiva per l'especificació i l'intercanvi de metadades, ja que les descripcions dels recursos no estan en esquemes fixos. Molts sistemes d' RDF utilitzen les bases de dades relacionals per a l'emmagatzematge persistent i per a la recuperació de la informació.

Hi ha dos aspectes que sorgeixen quan analitzem l'emmagatzemen de dades RDF i les bases de dades relacionals:

- Cóm es pot implementar un repositori de dades RDF en una base de dades relacional?
- I si ja tenim una base de dades relacional, cóm podem expressar l'RDF?

Aquestes dues preguntes formen part d'una pregunta més general de com fer correspondre dades d' RDF i el seu esquema i una base de dades relacional.

Requisits del repositori de dades d'RDF

- Tractament dels components *strings* de les triples d' RDF com si fossin un objecte binari, permetent la recerca de text utilitzant operadors com conjunció.
- Suport eficient per a URIs, ja que aquests s'utilitzen per identificar tots els termes del graf d' RDF i estan contingudes en totes les parts de les *triples* RDF.
- Suport en la utilització del tipus dades. RDF dóna suport als tipus de dades compatible amb l'XML Schema.
- Els contenidors d'RDF (`rdf:Seq`, `rdf:Bag`, `rdf:Alt`, `rdf:_n`) es construeixen com a múltiples *triples*, essent prèviament desconeguda la mida.
- Suport de l'RDF Schema.
- Suport de les Ontologies. Les sentències RDF són principalment conjuntives juntament amb la descripció del vocabulari utilitzat.
- Suport per a fer un seguiment de les triples RDF, ja que la informació per fer el seguiment d'on venen i com s'afegeixen al graf RDF no està emmagatzema en aquest.

3.2.1 SESAME

Sesame és una arquitectura desenvolupada per a un eficient emmagatzemament i consulta de grans quantitats de metadades en RDF i RDF Schema. Un dels seus components principals és SAIL (*Storage And Inference Layer*) que és la capa que s'encarrega d'interactuar amb els SGBDs, permetent d'aquesta manera mantenir l'arquitectura Sesame independent dels SGBDs. Per tant, permet el seu ús en una varietat de dispositius d'emmagatzemament com base de dades relacionals, emmagatzemament de triples o bases de dades orientades a l'objecte, sense haver de canviar el motor de consulta o uns altres mòduls funcionals [15].

Sesame ofereix suport de control de concurrència, exportació independent de RDF i RDFS i consultes en llenguatge RQL i SPARQL. Així mateix, ofereix diverses formes comunicació: via HTTP pel context Web; o RMI (*Remote Method Invocation*) o SOAP (*Simple Object Acces Protocol*).

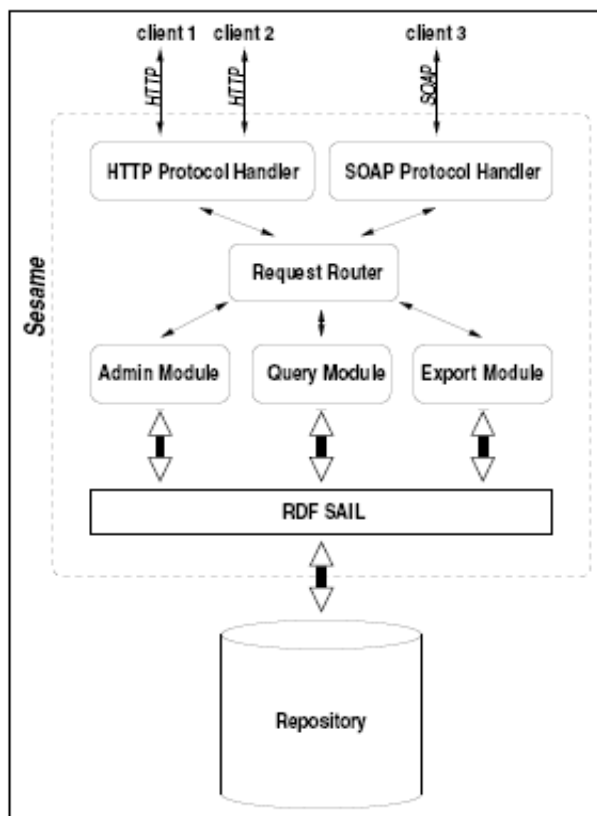


Figura 3.3 - Arquitectura SESAME

Mòduls funcionals de SESAME

- **Mòdul de consulta.** Per realitzar les consultes. Els llenguatges de consulta que implementa, entre d'altres, són l'SeRQL i l' SPARQL. Hi ha tres nivells d'abstracció: el nivell sintàctic, documents XML; el nivell estructural, que considera un conjunt de triples, és a dir, el model RDF; i el nivell semàntic, que constitueix una o més grafs amb semàntiques parcialment definides.
- **Mòdul d'administració.** Per possibilitar la inserció de dades RDF i informació d'esquemes en un repositori. Hi ha dos operacions fonamentals: l'addició incremental de dades i esquemes RDF i la neteja del repositori. Aquest mòdul comprova la consistència de les sentències afegides i realitza inferències.
- **Mòdul d'exportació RDF.** Permet exportar els continguts d'un repositori (esquemes, dades). Utilitza transformadors de triples RDF a documents XML. La idea és que serveixi per alimentar altres aplicacions basades en RDF.

SAIL RDF

La SAIL és una API que ofereix mètodes RDF específics als seus "clients" i traduir aquests mètodes a crides a un SGBD específic. Consta d'un conjunt d'interfícies Java que han estat específicament dissenyades per l'emmagatzemament i l'extracció d'informació basada en RDF. Els seus principis més rellevants són:

- Definir una interfície bàsica per emmagatzemar, esborrar i recuperar RDF i RDFS.
- Una abstracció del mecanisme d'emmagatzematge: bases de dades relacionals (PostgreSQL, MySQL, ...), sistema de fitxers, emmagatzemament en memòria, ...
- Aplicable a equips petits (PDAs) o grans.

- Extensible a d'altres llenguatges basats en RDF, com l' OWL

Un dels aspectes més importants de Sesame és que permet la seva implementació en una gran varietat de repositoris, ja sigui en bases de dades relacionals, emmagatzemament de triples RDF o a nivell remot, atesa la independència en relació al SGBD.

Emmagatzematge de documents RDF en Sesame

Sesame va ser dissenyat per a utilitzar sistemes d'emmagatzematge existents com els diversos SGBDR existents. Aquests SGBDs es poden utilitzar via Sesame en diverses formes d'emmagatzematge de dades d'RDF. Aquesta varietat es extreta pel codi específic del SGBD cap el SAIL (*Storage And Inference Layer*) com a interfície entre els mètodes d'RDF i l'API de la base de dades. Hi ha també, una implementació SAIL en memòria la qual utilitza arxius per l'emmagatzematge persistent.

El SAIL proporciona interfícies a mòduls que implementen llenguatges de consulta, administració (incloent l'addició de dades i esquemes RDF, eliminació) i exportant RDF com a RDF/XML. El SAIL és un API Java de alt nivell que inclou suport per a esquemes semàntics RDF i mètodes per a operar amb les dades.

El SAIL pot treballar per sobre d'alguns SGBDRs, SFBDOs, repositoris d'RDF existents, arxius RDF o serveis de xarxa. Sesame dona suport utilitzant alguns SGBD amb JDBC i explícitament dona suport a versions PostgreSQL, MySQL i Oracle.

Amb PostgreSQL, utilitzant les característiques de ser un SGBD relacional orientat a objectes, es permeten relacions entre les seves taules. Noves taules són adicionades a la base de dades cada vegada que una nova classe o propietat és adicionada al repositori. Amb MySQL s'utilitza un esquema estrictament relacional, i la gran diferència amb PostreSQL és que l'esquema de la base de dades no canvia quan un RDFS canvia.

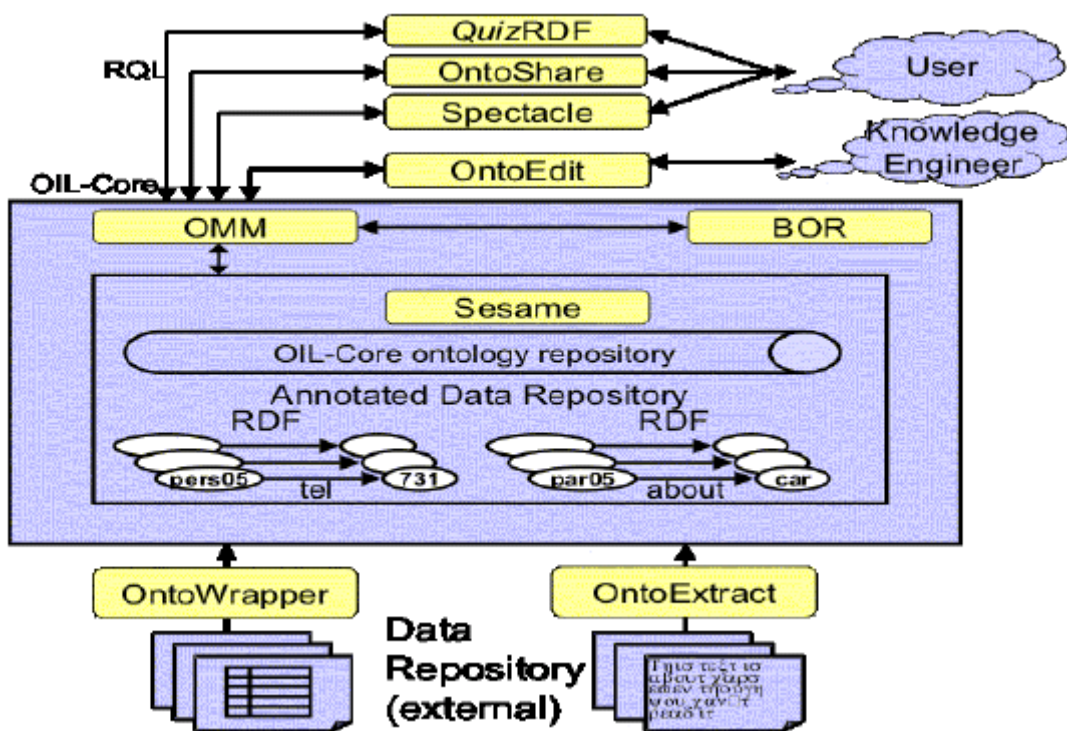


Figura 3.3 Sesame en el seu context

3.2.2 JENA

Jena [16] és un *framework* desenvolupat en Java per construir aplicacions per la Web Semàntica. És un projecte de codi obert i creix amb l'ajuda del grup d'investigació de Web Semàntica de "HP Labs". Ofereix una llibreria de programació per a RDF, RDFS i OWL, incloent-hi un motor d'inferència basat en regles.

L'estructura de Jena conté:

- Una API per a RDF.
- Lectura i escriptura de documents en format RDF/XML, N3 i N-Triples.
- Una API per a OWL.
- Emmagatzemament persistent i en memòria.
- RDQL i SPARQL com a llenguatges de consultes per a RDF.

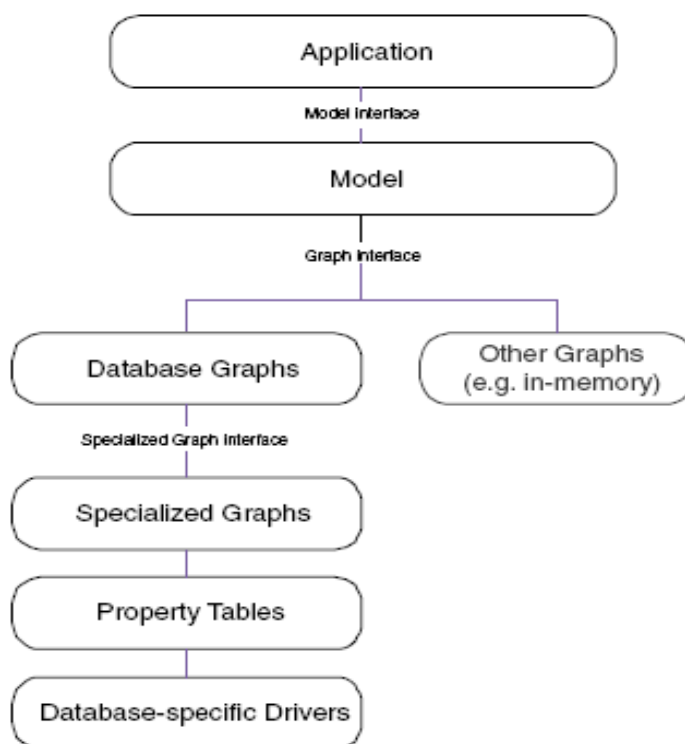


Figura 3.5 - Arquitectura JENA2

Les aplicacions interaccionen amb un model abstracte, el qual tradueix operacions d'alt nivell a operacions de nivell baix sobre triples emmagatzemades en un graf RDF. Hi ha diverses implementacions de grafs, però ens interessen els que proporcionen un emmagatzemen persistent.

En un nivell abstracte, el subsistema d'emmagatzematge de Jena2 només necessita implementar tres operacions: afegir sentències emmagatzemant una sentència RDF en una base de dades, suprimir sentències traient una sentència RDF des de la base de dades, i operació de cerca. Aquesta operació de cerca recupera totes les sentències. Una implementació utilitzada és emmagatzemar sentències de RDF en una base de dades relacional, utilitzant una taula de sentències anomenada "triple-store". Això és una taula que emmagatzema cada sentència RDF com a fila i té columnes per al subjecte, predicat i objecte.

- **API RDF.** Permet crear i manipular models RDF des d'una aplicació Java. Proporciona classes Java per representar models, recursos, propietats literals i sentències.

- **API d'Ontologies.** Jena suporta RDF Schema, DAML i OWL i és independent del llenguatge. Els recursos no estan lligats estàticament a una classe Java en particular.
- **Models persistents.** Jena proporciona emmagatzematge persistents de dades d'RDF en base de dades relacionals. L'aplicació no accedeix directament a la base de dades ni necessita conèixer l'esquema de la base de dades. Es proporcionen aplicacions per a MySQL, HSQLDB, PostgreSQL, Oracle i Microsoft SQL Server.

Emmagatzematge de documents RDF en Jena

El subsistema de persistència de Jena2 [17] implementa la interfície *Jena Model* que proporciona persistència al model RDF a través de la utilització de bases de dades relacionals. Utilitza un esquema en el qual els literals i els recursos URIs s'emmagatzemen directament en taules de sentències. Permet insercions i recuperacions ràpides per a les operacions de l'API a compte d'emmagatzemar sobre un esquema normalitzat de *triples* i nusos. També dona suport a *Jena Fastpath* per a consultes de SPARQL que dinàmicament genera les consultes SQL per a *SPARQL Basic Graph Patterns*, executant parts significants de consultes SPARQL dins de la base de dades.

La capa de persistència de Jena2 presenta una interfície de graf donant suport a les operacions habitual d'afegir, suprimir i cercar (figura 3.9). S'implementa cada graf lògic utilitzant una llista ordenada de grafs especialitzats, cada un dels quals és optimitzat per emmagatzemar un estil particular de sentència. Per exemple, en la figura el primer graf lògic s'implementa utilitzant-ne tres grafs especialitzats.

Una operació en el graf lògic com afegir sentència, suprimir sentència o cercar, és processada, al seu torn, cridant afegir, suprimir i cerca sobre cada gràfic especialitzat. Els resultats de les operacions individuals són combinats i retornats com el resultat per al graf sencer.

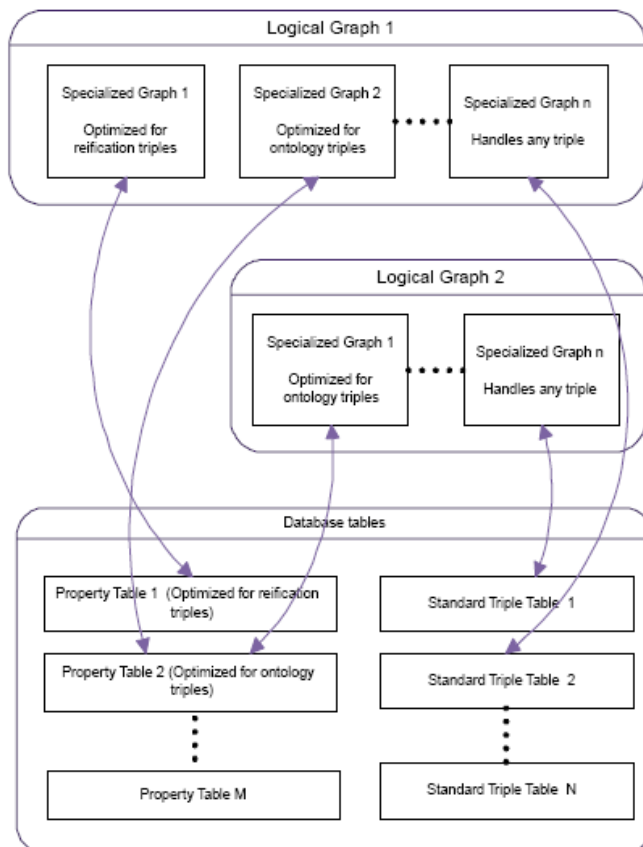


Figura 3.6 - Graf contenidor Grafs especialitzats sobre taules

El *driver* de la base de dades proporciona una interfície d'emmagatzematge abstracta que aïlla els grafs especialitzats de diferències de com les bases de dades donen suport a expressions, nulls, taules i creació d'índex, etc. Hi ha una implementació de driver genèrica per a bases de dades de SQL i *drivers* específics per a PostgreSQL, MySQL, Oracle, etc. El *driver* és responsable d'operacions de definició de dades com la inicialització de la base de dades, la creació i supressió de taules, de l'assignació d'identificadors.

Aquesta implementació de *drivers* específics, junt amb diverses formes d'esquema de base de dades, permeten a l'usuari triar-ne un SGBD apropiat per l'aplicació, com admetre models múltiples en un SGBDR, o seleccionar la utilització de procediments emmagatzemats.

3.2.3 REDLAND

Redland [18] proveeix una interfície d'alt nivell per a RDF. És una aplicació flexible i eficaç d'RDF que proporciona interfícies d'alt nivell permetent emmagatzemar, consultar i tractar informació basada en el model RDF. Redland s'implementa utilitzant una API orientada a l'objecte, proporcionant unes classes d'aplicació; com ara mòduls que es poden afegir, treure's o reemplaçar-se per permetre optimitzacions de funcionalitat.

L'estructura proporciona la tecnologia bàsica per desenvolupar aplicacions noves basades en RDF. Redland ofereix l'emmagatzematge dels grafs RDF en memòria, bases de dades relacionals (com per exemple MySQL, o Berkeley DB), arxius o URIs. Les consultes són amb SPARQL i RDQL, les quals s'executen amb el paquet *Rasqal RDF Query Library*.

Classes de Redland

Es defineixen un conjunt de classes per cobrir els conceptes exigits per Redland. Les classes s'utilitzen i estan associades l'una amb l'altra com mostra la figura 3.4. La classe *Stream* és utilitzada quan s'accepta una seqüència de sentències o són generades per les classes *Model*, *Storage* o *Parser*. La classe *Model* utilitza *Stream* per serialitzar i deserialitzar *Model* i retorna llistes de sentències des de consultes. La classe *Parser* només s'utilitza per proporcionar una seqüència de sentències com a resultat d'una anàlisi.

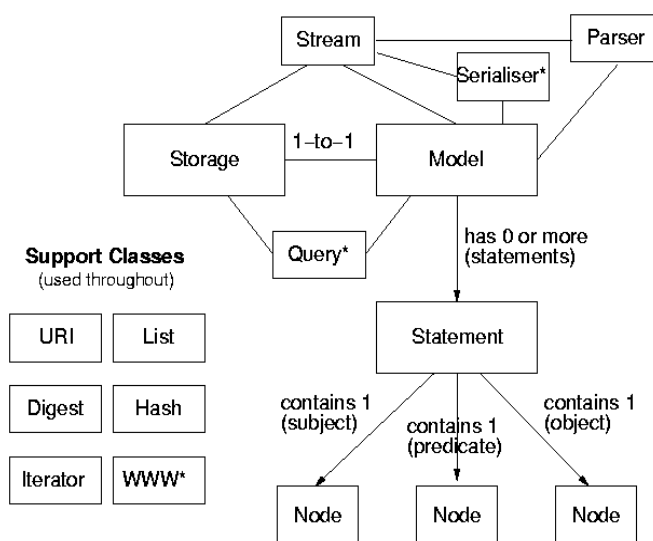


Figura 3.7 - Classes de Redland

Classes *Model* i *Storage*

La classe *Model* és la principal interfície d'aplicació per Redland, com es mostra a la figura 3.5. Malgrat això, gran part de la funcionalitat de la classe és proporcionada per unes altres classes. La classe *Storage* implementa la gestió dels mòduls d'emmagatzematge i també gestiona parts de l'API d'aquests mòduls.

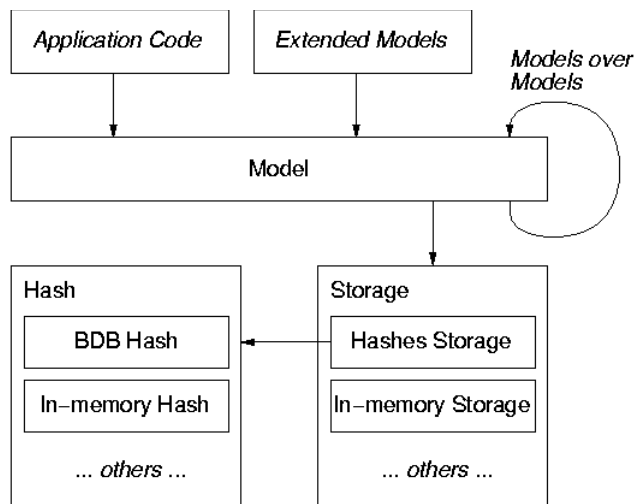


Figura 3.8 - Redland Model Layers

Redland, descrit en *Designing and Building the Redland RDF Application Framework* [19] és una biblioteca en C que implementa el model dels grafs de RDF dissenyat per la portabilitat, flexibilitat i actuació. Redland s'ha desenvolupat des del 2000 i l'API C té unes quantes interfícies de llenguatge d'alt nivell que permeten el desenvolupament ràpid de sistemes que utilitzen els APIs: *Redland Java API*, *Redland Perl API*, *Redland Python*, *Redland Ruby API*, *Redland PHP API* i *Redland Tcl API*. Redland utilitza la biblioteca **Raptor** que tracta de l'anàlisi de la sintaxis d'RDF (RDF/XML i N-Triples).

Emmagatzematge de documents RDF en Redland

Redland proporciona múltiples maneres d'entrar dades d' RDF, des de qualsevol sintaxis d'RDF o directament per *triples* RDF. Aquestes dades es poden tractar utilitzant l'API principal i els resultats solen generar formats nous. El suport de llenguatge de consulta que dóna SPARQL permet accedir i tractar les dades.

És important que les entrades i sortides de dades RDF es realitzi fàcilment des d'un sistema, i Redland proporciona *parsers* que poden llegir sintaxis d'RDF com RDF/XML i generar *triples* d'RDF; així també Redland proporciona *serialisers* que fan l'invers i generen sintaxis d'un graf d'RDF. Aquests s'integren a l'API, així que llegir i escriure a i des de strings, arxius locals i URIs és una crida de mètode senzilla.

Redland proporciona programes de suport per emmagatzematge de grafs d'RDF, utilitzant una abstracció d'interfície d'emmagatzematge. L'elecció d'emmagatzematge pot dependre no només del sistema que Redland es configura i es fonamenta, sinó de les característiques que proporciona com emmagatzematge: dins de memòria, dipòsits persistents en arxius estàtics, arxius de base de dades simples, fins a utilitzar una base de dades relacional (MySQL).

Un ús freqüent de dades d'RDF ha d'afegir contingut des d'una varietat de fonts de dades a un graf d'RDF (magatzem de dades) i llavors tractar i consultar el resultat. Redland proporciona el *context*, el qual permet que un conjunt de triples RDF emmagatzemades en

un dipòsit de dades ser marcadades amb una URI quan afegim dades, i a més donen suport a aquesta URI quan cerquem dades. Això permet que quan afegim dades RDF actualitzar el graf en cas que el contingut de la font original canvia, i també permet la supressió de totes les *tripletes* associades amb un context, de forma que, si cal, les noves dades es puguin carregar.

	Triple	Contexts
1.	(s1, p1, o1)	C1, C2
2.	(s2, p2, o2)	C1
3.	(s3, p3, o3)	C2
4.	(s4, p4, o4)	C2
5.	(C1, p5, o5)	
6.	(C2, p6, o6)	

Figura 3.9 - Graf Redland RDF amb contexts

Els mètodes de la classe *Model* proporcionen el control de *contexts* incloent-hi llistar tots els nusos de context en un graf, llistar totes les *triples* amb un *context* donat i afegir/suprimir *triples* associades a un *context*.

3.2.4 KAON

KAON [20] proporciona un repositori d’RDF utilitzant EJB (*Enterprise Java Beans*) i amb les dades persistents en una base de dades relacional mitjançant JDBC. La persistència és proporcionada per mecanismes de contenidors d’EJB, i l’esquema relacional és proporcionat pel servidor per a construir ontologies, amb un esquema d’indexació complex sobre el contingut. Funciona amb qualsevol base de dades relacionals compatible amb SQL2 i executa PostgreSQL, IBM DB2 i Oracle 9i.

El **KAON Server** és el repositori d’ontologies pel projecte KAON. Es va crear amb els requeriments de persistència, d’actualització semàntica, concurrència i seguretat que no es veien disponibles, en el moment del seu disseny, en altres repositoris d’RDF. KAON també proporciona una interfície separada a motors d’inferència que es poden registrar dinàmicament. També proporciona connectors com Java RMI, IIOP i SOAP.

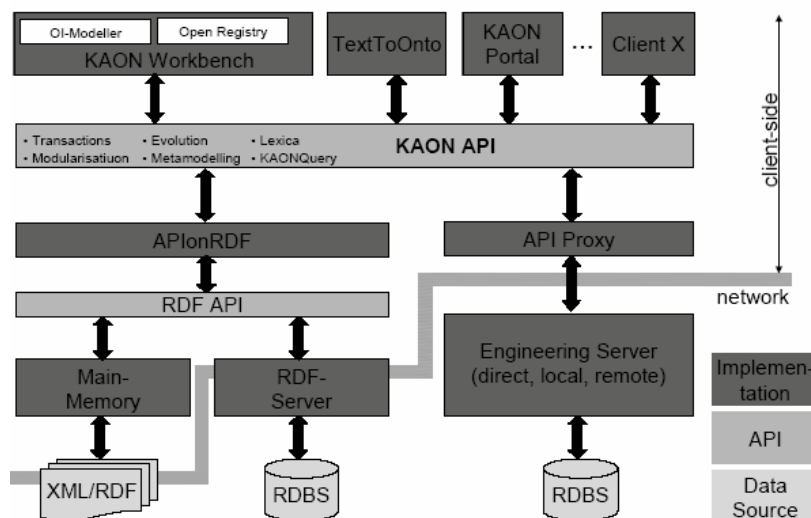


Figura 3.10 - Arquitectura KAON

KAON2

És el successor del projecte KAON (*KAON1*). La diferència principal respecta a KAON1 és el llenguatge d'ontologia a que es dona suport: KAON1 utilitzava una ampliació d'RDFS, mentre que KAON2 es basa en *OWL-DL* i *F-Logic*.

KAON2 té les característiques següents:

- Un API per manegar ontologies en OWL-DL, regles basades en la notació SWRL i F-Logic.
- Un servidor autònom que proporciona accés a ontologies distribuïdes que utilitza RMI.
- Un motor d'inferències per contestar a consultes conjuntives. S'utilitza el llenguatge de consulta SPARQL.
- Una interfície que permet l'accés des d'eines com Protegé.
- Un mòdul per extreure exemples d'ontologia de bases de dades relacionals.

Emmagatzematge de documents RDF en KAON

A l'estructura de KAON [21] es defineixen dues APIs per l'actualització i consultes de dades de la Web Semàntica, un RDF API i una interfície ontològica anomenada KAON API. Les diferents eines implementen aquest APIs de manera diferent com descriu la figura 3.11 i es convertiran en components funcionals.

L'RDF API consta d'interfícies pel tractament transaccional de models d'RDF amb la possibilitat de modularització, un RDF parser i un RDF serializer per escriure models RDF. Aquest consta d'un conjunt de sentències, i cada sentència es representa com una *triple* (subjecte – predicat -- objecte), essent els elements recursos o literals. Les corresponents interfícies presenten mètodes per consultar i actualitzar aquestes entitats.

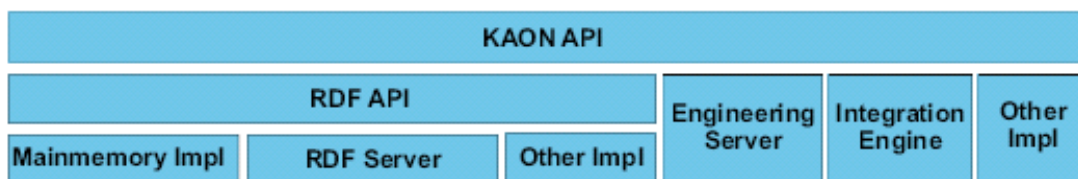


Figura 3.11 - Implementació KAON API

RDF Main Memory Implementation.

La implementació del RDF API és útil per l'accés als models d'RDF en memòria, i aquest model RDF es carrega a partir d'una serialització d'XML. Després d'això, les sentències RDF poden ser afegides, modificades i suprimides, encapsulat tot en una transacció. Finalment, el model RDF en memòria s'ha de serialitzar una altra vegada.

RDF Server

RDF Server és una implementació de l'RDF API que permet emmagatzematge persistent i la gestió dels models d'RDF. Aquesta solució es recolza en una estructura física que correspon al model d'RDF. Les dades es representen utilitzant quatre taules, una representa models i les altres representen sentències contingudes en el model.

Tanmateix, aquestes estructures simples són molt òptimes per a RDF. Normalment es fan un gran nombre de "join" per recórrer els enllaços de l'objecte. Per això, les estructures d'indexació són requerides en tots els elements de la sentència per oferir un eficient accés a les dades. Això es podria aconseguir utilitzant índexs *hashed*, i que podria facilitar la consulta en un temps constant. Tanmateix, les *hashtables* requereixen una bona distribució de claus per als seus valors. Això no és una realitat per URIs (els quals tenen prefixos similars) ni per incrementar automàticament identificadors.

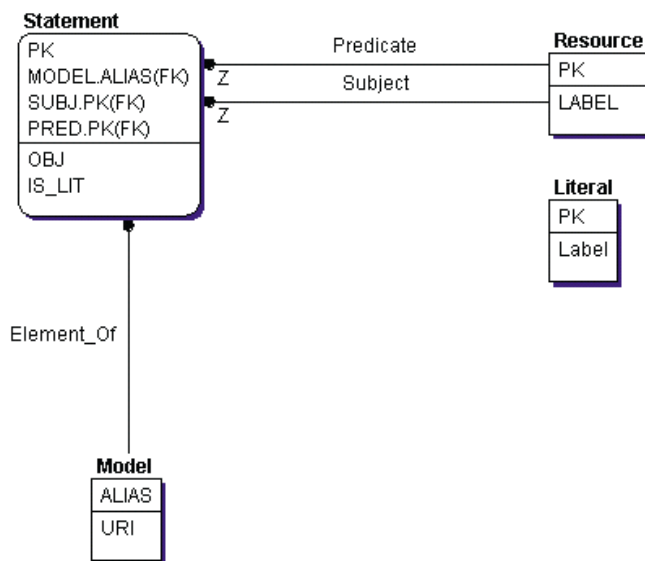


Figura 3.12 - Esquema per emmagatzemament d' RDF

Aquesta representació requereix que les altres relacions representin recursos i literals, i proporcionin un identificador intern que mostri bones propietats d'indexació per proporcionar índexs eficaços sobre la relació de la sentència.

L'RDF Server utilitza un SGBD relacional i es recolza en el JBoss Application Server que tracta la comunicació entre client i SGBD.

3.3 Comparativa entre els SGBDs: Jena vs. Sesame

En aquest apartat fem un anàlisi comparatiu de dos dels SGBDs que hem vist anteriorment. Aquest anàlisi s'ha basat en les versions de Jena i Sesame que donen suport actualment el llenguatge de consultes SPARQL, concretament **Jena 2.4** i **Sesame 2.0-alpha-3**.

Sesame : <http://www.openrdf.org/>

Jena : <http://jena.sourceforge.net/>

Jena és un *framework* de codi obert en Java. Inclou moltes APIs i utilitats que es requereixen en treballs d'aplicacions semàntiques. A més Jena, no es limita a les seves pròpies solucions, inclou components de tipus *Plug-In* i interfícies amb d'altres solucions externes, i així poder connectar a Jena amb un raonador independent si es requereix. A més Jena dóna suport per a RDF, OWL i d'altres estàndards de semàntica informàtica, incloent un motor de SPARQL per a consultes d'aquest tipus anomenat ARQ

Sesame és un *framework* de codi obert desenvolupat en Java per emmagatzemar, consultar i raonar amb RDF i RDF Schema [22]. Pot ser utilitzat com a base de dades per a RDF, o com a llibreria Java per a aplicacions que necessitin funcionar internament amb RDF. Sesame proporciona les eines necessàries per analitzar, interpretar, consultar i emmagatzemar tota aquesta informació, possiblement inclosa en aplicacions, o en bases de dades separades o en servidors remots.

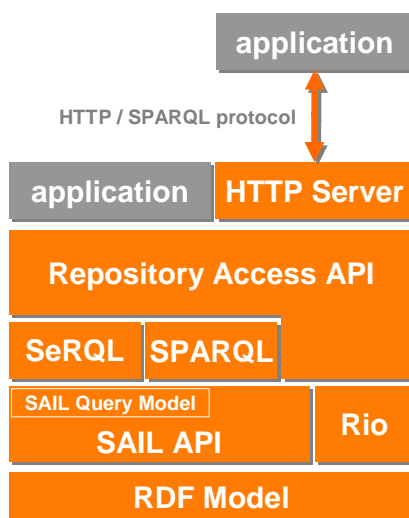


Figura 3.13 - Arquitectura Sesame 2

3.3.1 Model RDF

Jena - És un API de Java per a aplicacions semàntiques [23]. El paquet d’RDF clau per desenvolupar l’aplicació es `com.hp.hpl.jena.rdf.model` [24]. Aquest paquet conté interfícies per a representar models, recursos, propietats, literals, sentències i tots els altres conceptes claus d’RDF, i una classe `ModelFactory` que permet la creació de models.

- `ModelFactory.createDefaultModel()` permet la creació d’un model en memòria. Retorna una instància de `Model` sobre la qual podem crear `Resources`.
- `Model.createProperty()` permet la creació de la relació.
- Afegir sentències al model: `Resource.addProperty()` o `Model.createStatement()` i `Model.add()`.

En Jena una sentència està composta per:

- Un subjecte en la forma d’un `Resource`.
- Un predicat representat per una classe `Property`.
- Un objecte, o un `Literal` o un `Resource`.

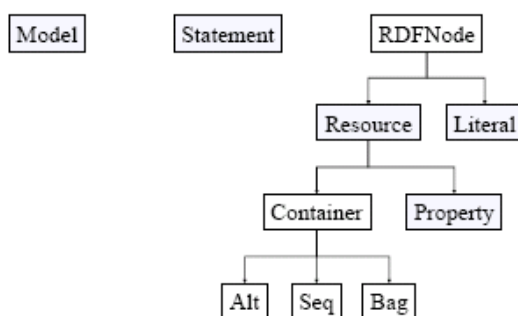


Figura 3.14 - Jena2 RDF API: Interfícies principals

Sesame - El paquet `org.openrdf.model` [25]. conté les interfícies que són definides en el model RDF de Sesame. Les classes `URI`, `BNode` i `Literal` estan organitzades en una jerarquia reflectint la seva utilització en RDF. La interfície `ValueFactory` permet la creació dels objectes d’aquest paquet [26].

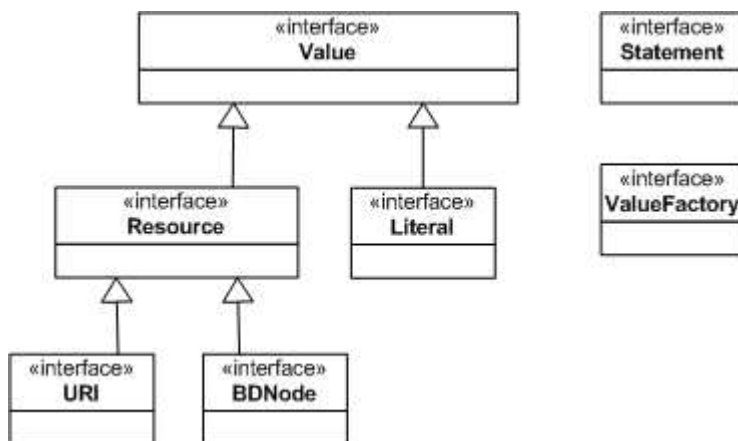


Figura 3.15 - Sesame. Diagrama de classes pel model RDF

3.3.2 Models persistents

Jena - El subsistema d'emmagatzematge persistent de Jena2 és una extensió de la classe `Model` que proporciona persistència transparent per models mitjançant la utilització de bases de dades (MySQL, Oracle, PostgreSQL i Microsoft SQL Server).

En Jena 2 totes les bases de dades són multimodel i cada model és, per defecte, emmagatzemat en taules separades. Els models poden compartir taules de base de dades utilitzant l'opció `StoreWithModel`. El nom dels model pot ser qualsevol string, tanmateix els usuaris són potenciats a utilitzar URIs com a nom de models per la compatibilitat amb futures característiques de Jena.

Hi ha dos mecanismes per a crear models persistents, un utilitzant mètodes de `ModelFactory`, i un altre utilitzant constructors per la classe `ModelRDB`. Els mètodes de `ModelFactory` no retornen una instància de `ModelRDB`, Conseqüentment, els mètodes definits en `ModelRDB` no estan disponibles per a models creats per `ModelFactory`; per exemple, `remove`, `setDoDuplicateCheck`. Si alguna aplicació necessita aquests mètodes, els constructors de `ModelRDB` podrien ser utilitzats. Aquesta és una mesura temporal fins que els models creats per `ModelFactory` suportin tot el potencia de `ModelRDB`.

Sesame suporta diverses formes d'emmagatzemament de repositoris, SGBDR, memòria o arxius. Tanmateix, la més usuals són el SGBDR. Sesame pot utilitzar PostgreSQL, MySQL, Oracle (9i o posterior) i SQL Server.

El SAIL API permet a Sesame utilitzar els diversos mecanismes d'emmagatzematge. Fa una abstracció del format d'emmagatzematge utilitzat i proporciona el suport al raonament. Les principals interfícies SAIL estan en el paquet `org.openrdf.sesame.sail`. El SAIL API defineix un conjunt de interfícies per magatzems RDF i inferències. La principal interfície és `sail`, la qual defineix mètodes per iniciar, consultar, modificar i tancar l'objecte `sail`.

El *Repository API* és l'accés central pels repositoris Sesame. Pot ser utilitzat per consultar i actualitzar el contingut dels repositoris local i remot. El *Repository API* tracta tots els detalls de la comunicació client-servidor, permetent manipular els repositoris remots tan fàcilment com els locals. Les classes i interfícies principals pel *Repository API* poden ser trobats en el paquet `org.openrdf.sesame.repository`. Les classes d'aquest paquet tenen mètodes per avaluar una consulta i retornar un resultat, afegir i suprimir sentències RDF, començar i acabar transaccions, etc.

La classe `Repository` és una implementació la interfície `Transaction`. La qual permet directament operacions de modificacions en el repositori, sense la necessitat de començar i acabar una transacció. Si es prefereix, pel mètode `startTransaction()` és possible començar explícitament una transacció.

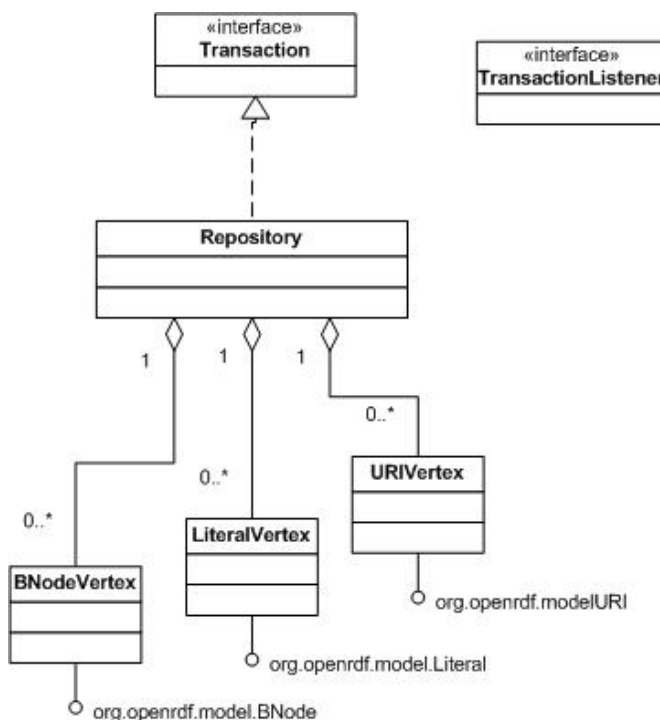


Figura 3.16 - Sesame 2 Repository API

3.3.3 Inferència

Jena - El subsistema d'inferència de Jena2 està dissenyat per permetre una varietat de motors d'inferència o raonadors. La utilització d'aquests mecanismes és donar suport als llenguatges com RDFS i OWL.

Normalment, les aplicacions tenen accés a la maquinària d'inferència utilitzant `ModelFactory` per associar dades amb algú raonadora per crear un nou mode. Les consultes al model creat tornaran no només aquelles sentències que hi eren presents en les dades originals, sinó també les sentències addicionals que poden ser deduïdes de les dades utilitzant les regles o uns altres mecanismes d'inferència implementats pel raonador.

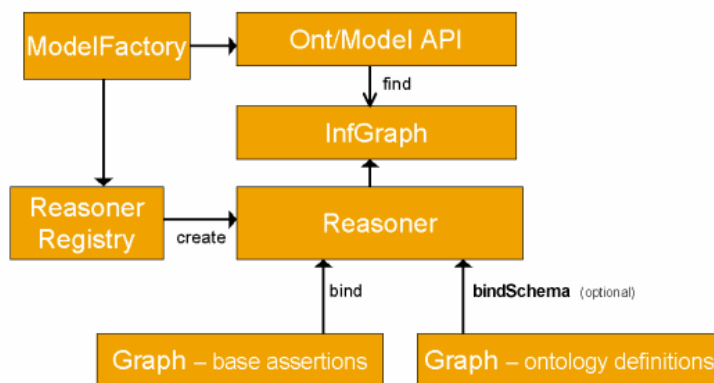


Figura 3.17 - Estructura de la maquinària d'inferència en Jena

En la distribució de Jena estan inclosos un raonadors predefinits:

- **Raonador transitiu:** Proporciona suport per emmagatzemar i recórrer l'entramat de classes i propietats. Aquest implementa les propietats transitiva i simètrica de `rdfs:subPropertyOf` i `rdfs:subClassOf`.
- **RDFS ruler reasoner:** Implementa un subconjunt configurable de vinculacions d'RDF.
- **OWL, OWL Mini, OWL MicroReasoners:** Un conjunt pràctic però amb implementació incompleta de OWL Lite.
- **DAML Micro Reasoner:** Utilitzat internament per facilitar a DAML API proporcionar una mínima inferència.

Sesame - Un concepte central en el Sesame és el *repository*. Un repositori és un contenidor d'emmagatzematge per a RDF. Aquest pot significar simplement un objecte de Java en la memòria, o pot significar una base de dades relacional. Independentment de l'opció d'emmagatzematge escollida, és important entendre que a cada operació en el Sesame té lloc amb respecte un repositori: quan s'afegeix una dada RDF, quan s'afegeix aquesta al repositori. Quan es fa una consulta, es consulta un repositori particular.

Sesame dona suport a la inferència de RDF Schema. Això significa que donat un conjunt d'RDF i RDF Schema, Sesame pot trobar la informació implícita en les dades. Sesame dona suport afegint tota la informació implícita en el repositori quan les dades s'estan afegint.

La inferència en Sesame està associada amb el tipus de repositori que estem utilitzant. Sesame suporta diversos tipus de repositoris. Alguns d'aquests suporten la inferència, d'altres no.

Sesame inclou els següents components de tipus *plug-in*:

- **BOR** és un raonador de descripció lògica (DL) que pot ser integrat en Sesame. BOR és un SHQ(D) *reasoner*. Això proporciona un suport DAML+OIL complet i és compilat amb les últimes especificacions d'OWL.
- L'**OWLIM** és un repositori semàntic d'alt rendiment, empaquetat en el SAIL per la base de dades RDF de Sesame. OWLIM utilitza el motor IRRE per mostra el raonador OWL DLP. El raonador i l'avaluació de consultes són realitzats en memòria, mentre s'assegura la preservació, consistència i integració de les dades

3.3.4 Llenguatges de consulta

Jena - El llenguatge de consulta SPARQL s'afegeix als llenguatges existents com `rdfBD`, `RDQL` i `SeRQL`. Les consultes en el llenguatge SPARQL són creades i executades amb Jena mitjançant classes en el paquet `com.hp.hpljena.query`. Utilitzant `QueryFactory` és la forma més simple. `QueryFactory` té diversos mètodes `create()` per consultar des d'un arxiu o d'un string; aquests mètodes retornen un objecte `Query` que encapsula la consulta analitzada.

`QueryExecution` és una classe que representa l'execució d'una consulta. S'ha de cridar al mètode `QueryExecutionFactory.create(query,model)`, el qual retorna un objecte `QueryExecution`. Hi ha diferents mètodes que donen suport a `QueryExecution`. Per exemple, per fer una consulta SELECT simple es crida al mètode `execSelect()` que retorna un `ResultSet`, aquest permet la iteració sobre cada `QuerySolution` tornat per la consulta, proporcionant l'accés al valor de cada variable.

```
// Open the bloggers RDF graph from the filesystem
InputStream in = new FileInputStream(new File("bloggers.rdf"));

// Create an empty in-memory model and populate it from the graph
Model model = ModelFactory.createMemModelMaker().createModel();
model.read(in,null); // null base URI, since model URIs are absolute
in.close();

// Create a new query
String queryString =
    "PREFIX foaf: <http://xmlns.com/foaf/0.1/> " +
    "SELECT ?url " +
    "WHERE {" +
    "    ?contributor foaf:name \"Jon Foobar\" . " +
    "    ?contributor foaf:weblog ?url . " +
    "}";

Query query = QueryFactory.create(queryString);

// Execute the query and obtain results
QueryExecution qe = QueryExecutionFactory.create(query, model);
ResultSet results = qe.execSelect();

// Output query results
ResultSetFormatter.out(System.out, results, query);

// Important - free up resources used running the query
qe.close();
```

Exemple 3 - Execució d'una consulta simple utilitzant API de Jena

Sesame - També en Sesame 2.0 el llenguatge de consultes SPARQL s'afegeix als llenguatges de consulta anteriors: SeRQL, RQL i RDQL. Per realitzar les consultes en llenguatge SPARQL hem d'utilitzar el paquet `org.openrdf.sesame.query`, el qual conté les classes `SPARQLResultsReader` i `SPARQLResultsWriter`.

El SAIL Query modela el paquet `org.openrdf.sesame.sail.query`. El *Sail Query Model* és un model genèric per a consultes RDF. Les consultes en diversos llenguatges, entre ells l'SPARQL, són analitzades sintàcticament en models construïts des d'aquestes classes.

Un SAIL que operi en un SGBDR, el qual dona suport SQL, pot traduir parts del model de consulta en consultes SQL. També el model de consultes és capaç d'avaluar-se: la interfície `TripleSource` defineix tots els mètodes necessaris pel model de consulta per fer aquesta . Implementat aquesta interfície, un SAIL pot utilitzar l'autoavaluació. `QueryOptimizer` pot ser utilitzat per aplicar algunes optimitzacions generals per preguntar models, resultant temps d'avaluació inferiors.

3.3.5 Interoperabilitat Jena – Sesame

Actualment, Jena i Sesame són les dues implementacions més populars per emmagatzemar RDF. Les diferències entre els dos els trobem en aspectes com l'ajuda a la inferència, funcionament i, fins a l'actualitat, en els llenguatges de consulta.

S'utilitza el Jena API o Sesame API per publicar, consultar i raonar sobre una *triple* RDF. La interoperabilitat entre Sesame i Jena pot ser un problema quan una aplicació basada en Jena API ha de tenir accés a un *backend* de Sesame, o una aplicació basada en Sesame API ha de tenir accés a un *backend* de Jena. Aquest problema està solventat parcialment pel

Sesame Jena Adapter, que proporciona l'accés a un model de Jena per l'SAIL API de Sesame.

Jena Sesame Model (JSM) aborda el problema de la interoperabilitat entre Jena i Sesame per altra via. El JSM proporciona l'accés a un repositori de Sesame pel Jena API. El JSM té les característiques següents:

- El JSM es conforma a Jena Model API.
- El JSM es conforma a Jena Graph SPI.
- Internament, JSM funciona en un repositori de Sesame (actualment només el repositori local és acceptat).

L'accés d'un model Jena pel SAIL API de Sesame permet a un usuari executar RQL i consultes de SeRQL en un model Jena, tant pel Sesame API com pel Jena API. Un model de Jena també pot ser afegit com un repositori a un servidor de Sesame i consultar per la interfície Web de Sesame.

Capítol 4

Llenguatge de consultes SPARQL per a RDF

4.1 Introducció

El llenguatge de consultes SPARQL (*Simple Protocol and RDF Query Language*) és un llenguatge de consulta per a informació representada en RDF, dins el context de la Web Semàntica. A l'actualitat està en procés d'estandardització pel DAWG (*RDF Data Access Working Group*) del W3C, essent candidat de recomanació el 6 d'abril de 2006 i havent-ne fet públic l'últim esborrany de treball en data 4 d'octubre de 2006 [27].

En els capítols anteriors hem vist que un graf RDF és un conjunt de *triples*, i que cada *triple* consisteix en un subjecte (*URI o Node en blanc*), un predicat (*URI*) i un objecte (*URI o Node en blanc o Literal*). Aquestes *triples* poden provenir des de diverses fonts. Per exemple, poden provenir directament des d'un document RDF; poden ser inferides des d'una altra *triple* RDF; o poden ser expressions d'RDF de dades emmagatzemades en altres formats, com XML o base de dades relacionals.

L'SPARQL és un llenguatge de consulta per aconseguir la informació dels graf RDF, i ens proporciona facilitats per:

- Extreure informació en forma d'URIs, Nodes en blanc i literals
- Extreure subgrafs RDF
- Construir nous grafs RDF fonamentats en la informació del graf consultat.

Com a llenguatge d'accés a dades, l'SPARQL és adequat tant per la seva utilització local com per la remota. El document *SPARQL Protocol for RDF* [28] descriu l'accés remot de dades i la transmissió de consultes dels clients als processadors. El Protocol SPARQL ha estat dissenyat per ser compatible amb el llenguatge de consultes SPARQL. Està descrit de dues maneres: la primera, com a una interfície abstracta independents de qualsevol execució, implementació o vinculació a un altre protocol; i la segona, com HTTP i SOAP units en aquesta interfície.

Els resultats de la consulta poden ser retornats en XML utilitzant l'*SPARQL Query Results XML Format* [29]. Aquest és un vocabulari XML que estandarditza la representació dels resultats d'una consulta SPARQL. El format consisteix en diversos elements en un únic namespace. A diferència de la sintaxi RDF, que ofereix varies formes de representar les mateixes dades, el format de resultats d'SPARQL és més regular.

Els prefixos i namespaces comuns utilitzats en SPARQL són els següents:

Namespace	Prefixos	URIs
RDF	rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
RDFS	rdfs:	http://www.w3.org/2000/01/rdf-schema#
XML Schema Datatypes	xsd:	http://www.w3.org/2001/XMLSchema#
Dublin Core	dc:	http://purl.org/dc/elements/1.1/
FOAF	foaf:	http://xmlns.com/foaf/0.1/
OWL	owl:	http://www.w3.org/2002/07/owl#

El format de les dades estan en llenguatge **Turtle**, derivat del llenguatge *Notation 3*. Turtle dóna la possibilitat d'expressar diverses sentències per a un recurs, o diversos valors per a una propietat, compactant la sintaxi. Les URIs poden abreujar-se mitjançant la directiva @prefix.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
...
```

Els exemples explicatius d'aquest capítol estan basats en el document *SPARQL Query Language for RDF, W3C Working Draft 4 October 2006* [27]. Per cada exemple es mostren les dades, la consulta sobre aquestes dades i el seu resultat.

4.2 Realització de consultes simples

El llenguatge de consulta SPARQL està basat en el model de grafs. El model de graf més simple és el model triple, que es sembla a una triple RDF però amb la possibilitat d'una variable en lloc d'un terme RDF en la posició del subjecte, predicat i objecte.

Consulta simple

La següent consulta consisteix de dues parts, la clàusula **SELECT**, que identifica la variable que apareixerà en el resultat de la consulta, i la clàusula **WHERE**, que té una *triple*. La solució de la consulta és un conjunt de parelles d'una variable amb un valor.

Dades:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name
WHERE {
  ?x foaf:name ?name .
}
```

Resultat de la consulta:

name
"Ignasi Pinto"

Múltiples correspondències

El resultat de la consulta són una seqüència de solucions. Pot haver-hi zero, una o múltiples solucions amb una consulta.

Dades:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .

_:b foaf:name "Guillem Bassas" .
_:b foaf:mbox <mailto:guillem@exemple.com> .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?mbox
WHERE {
  ?x foaf:name ?name .
  ?x foaf:mbox ?mbox .
}
```

Resultat de la consulta:

name	mbox
"Ignasi Pinto"	<mailto:ipinto@uoc.edu>
"Guillem Bassas"	<mailto:guillem@exemple.com>

Així també, es poden fer combinacions de literals RDF.

4.2.1 Restriccions de valors

La clàusula **FILTER** restringeix els resultats d'una consulta imposant restriccions als valors de les variables. Aquestes restriccions dels valors son expressions booleans i poden ser combinades amb els operadors lògics && i ||. Per exemple, una consulta que retorni una llista de noms podria ser modificada amb un **FILTER** per retornar només els noms que corresponguin a una expressió regular donada.

Dades:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:dateOfBirth "1960-04-03" .

_:b foaf:name "Guillem Bassas" .
_:b foaf:dateOfBirth "1973-06-10" .
```

Restriccions de valors de tipus strings.

S'utilitza l'operador **regex**. Aquest operador pot ser utilitzat per combinar formes lèxiques d'altres literals utilitzant l'operador **str**.

Consulta:

```

PREFIX foaf:    <http://xmlns.com/foaf/0.1/>

SELECT ?name
WHERE {
  ?x foaf:name ?name .
  FILTER regex(?name, "Ignasi")
}

```

Resultat de la consulta:

name
"Ignasi Pinto"

Restriccions de valors de literals d'altres tipus

També és possible el valors dels literals d'altres tipus. El filter s'aplica al valor del literal, no a la seva forma lèxica.

Consulta:

```

PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#>

SELECT ?name ?dateOfBirth
WHERE {
  ?x foaf:dateOfBirth ?dateOfBirth .
  FILTER (xsd:dateTime(?dateOfBirth) < "1970-01-01"^^xsd:dateTime)
}

```

Resultat de la consulta:

name	dateOfBirth
"Ignasi Pinto "	"1960-04-03"

4.2.2 Nodes en blanc en el resultat de la consulta

La presència de nodes en blanc poden ser indicats per etiquetes en la serialització del resultat de la consulta. Però aquesta informació no pot ser utilitzada per una aplicació o un client que rebí aquests resultats, ja que aquests nodes en blanc són tractats com a locals en les subseqüents consultes.

Dades:

```

@prefix foaf:    <http://xmlns.com/foaf.0.1/> .

_:a foaf:name    "Ignasi Pinto" .
_:b foaf:name    "Guillem Bassas" .

```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf.0.1/>

SELECT ?x ?name
WHERE {
  ?x foaf:name ?name .
}
```

Resultat de la consulta:

x	name
_:c	"Ignasi Pinto"
_:d	"Guillem Bassas"

4.3 Grafs

Les consultes d'SPARQL són realitzades sobre un o diversos grafs.

4.3.1 Graf Bàsic

El Graf Bàsic és un conjunt de triples i forma la base de la combinació de consultes SPARQL. L'operació de consultar un graf RDF proporciona l'accés a l'estructura del graf.

Dades:

```
@prefix foaf: <http://xmlns.com/foaf.0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .

_:b foaf:name "Guillem Bassas" .
_:b foaf:mbox <mailto:guillem@example.com> .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?mbox
WHERE {
  ?x foaf:name "Ignasi Pinto" .
  ?x foaf:mbox ?mbox .
}
```

Resulta de la consulta:

mbox
<mailto:ipinto@uoc.edu>

4.3.2 Grup de Grafs

El Grup de Grafs és un conjunt de grafs. En el text de la consulta de SPARQL, cada graf del Grup de Grafs està delimitat amb '{ }'.

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>

SELECT ?name ?mbox
WHERE {
  { ?x foaf:name ?name . }
  { ?x foaf:mbox ?mbox . }
}
```

4.4 Informació opcional

SPARQL té la capacitat de consultar les dades i no suspendre la consulta quan aquestes dades no existeixen, retornant sempre la informació no opcional. La consulta utilitza una part opcional per ampliar la informació trobada en la solució de la consulta.

OPTIONALS

La clàusula **OPTIONAL** és un operador binari que combina dos grafs

Dades:

```
@prefix foaf:    <http://xmlns.com/foaf.0.1/> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

_:a  rdf:type    foaf:Person .
_:a  foaf:name   "Ignasi Pinto" .
_:a  foaf:mbox   <mailto:ipinto@uoc.edu> .
_:a  foaf:mbox   <mailto:ignasi@exemple.org> .

_:b  rdf:type    foaf:Person .
_:b  foaf:name   "Guillem Bassas" .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?mbox
WHERE {
  ?x foaf:name ?name .
  OPTIONAL { ?x foaf:mbox ?mbox }
}
```

Resultat de la consulta:

name	mbox
"Ignasi Pinto"	<mailto:ipinto@uoc.edu>
"Ignasi Pinto"	<mailto:ignasi@exemple.org>
"Guillem Bassas"	

Les restriccions també poden ser donades en la consulta del graf, mitjançant la clàusula **FILTER** dins de la clàusula **OPTIONAL**.

També permet utilitzar la mateixa variable en dos o més clàusules OPTIONAL.

Dades:

```
@prefix foaf:      <http://xmlns.com/foaf/0.1/> .

_:a foaf:name      "Ignasi Pinto" .
_:a foaf:homepage  <http://memoria.exemple.org/ignasi/> .

_:b foaf:name      "Guillem Bassas" .
_:b foaf:mbox      <mailto:guillem@exemple.com> .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name ?mbox ?hpage
WHERE {
    ?x foaf:name ?name .
    OPTIONAL { ?x foaf:mbox ?mbox } .
    OPTIONAL { ?x foaf:homepage ?hpage }
}
```

Resultat de la consulta:

name	mbox	hpage
"Ignasi Pinto"		<http:// exemple.org/ignasi/>
"Guillem Bassas"	<mailto:guillem@exemple.com>	

4.5 Combinacions alternatives

SPARQL proporciona un mètode de combinar grafs. Si n'hi ha més d'una combinació, es troben totes les possibles solucions. La sintaxi utilitzada és la clàusula **UNION**.

Dades:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix vcard: <http://www.w3.org/vcard-rdf/3.0#> .

_:a foaf:name      "Ignasi Pinto" .
_:b foaf:name      "Guillem Bassas" .
_:c vcard:FN       "Sara Brown" .
_:d vcard:FN       "Andrea Martinez" .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/vcard-rdf/3.0#> .

SELECT ?name1 ?name2
WHERE {
    { [] foaf:name ?name1 } UNION { [] vcard:FN ?name2 }
}
```

Resultat de la consulta:

Name1	name2
"Ignasi Pinto"	
"Guillem Bassas"	
	"Sara Brown"
	"Andrea Martinez"

OPTIONAL i UNION

A la pràctica, la clàusula `OPTIONAL` és més comú que la clàusula `UNION`, però ambdós tenen les seves utilitats. `OPTIONAL` és útil per augmentar les solucions trobades. `UNION` és útil per concatenar les solucions de dos grafs. Tampoc és necessari que retornin la informació de la mateixa manera.

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/vcard-rdf/3.0#> .

SELECT ?name1 ?name2
WHERE {
    { [] foaf:name ?name } UNION { [] vcard:FN ?name }
}
```

Resultat de la consulta:

name1	name2
"Ignasi Pinto"	
"Guillem Bassas"	
	"Sara Brown"
	"Andrea Martinez"

4.6 Documents RDF

Un document RDF és la unitat que és consultada per la consulta SPARQL. Comprèn un graf, el graf per defecte (*default graph*) el qual no té nom, i zero o més grafs que estan anomenats (*named graphs*), cadascú identificat per una URI. La dada RDF no restringeix les relacions entre aquets grafs. La informació pot ser repetida en grafs diferents, mostrant-se la relació entre grafs. Així podem:

- Tenir la informació en el graf per defecte que conté la informació de procedència respecte els grafs anomenats.
- Incloure, igualment, la informació continguda en els grafs anomenats en el graf per defecte

En el següent exemple, el graf per defecte conté els noms i els editors de dos grafs amb noms. Les triples en els grafs amb noms, en aquest cas, no són visibles en el graf per defecte.

```
# Default graph
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://exemple.org/guillem> dc:publisher "Guillem Bassas" .
<http://exemple.org/ignasi> dc:publisher "Ignasi Pinto" .
```

```
# Named graph: http://exemple.org/guillem
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Guillem Bassas" .
_:a foaf:mbox <mailto:guillem@exemple.com> .
```

```
# Named graph: http://exemple.org/ignasi
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .
```

4.6.1 Especificar Document RDF

Una consulta SPARQL pot especificar les dades que siguin utilitzades en les correspondències fent ús de la clàusula FROM, i la clàusula FROM NAMED per descriure el document RDF.

Especificar el graf per defecte (*default graph*)

Cada clàusula FROM té un URI que indica el graf a utilitzar per formar el graf per defecte.

Dades:

```
# Default graph (stored at http://exemple.org/foaf/ignasiFoaf)
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
:a foaf:mbox <mailto:ipinto@uoc.edu> .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name
FROM <http://exemple.org/foaf/ignasiFoaf>
```

Resultat de la consulta:

name
"Ignasi Pinto"

Especificar els grafs amb noms (*named graphs*)

Una consulta pot proporcionar URIs dels grafs amb nom en el Document RDF utilitzant la clàusula FROM NAMED.

```
# Graph: http://exemple.org/guillem
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Guillem Bassas" .
_:a foaf:mbox <mailto:guillem@exemple.com> .
```

```
# Graph: http://exemple.org/ignasi
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .
```

```
...
FROM NAMED <http://exemple.org/ignasi>
FROM NAMED <http://exemple.org/guillem>
...
```

Combinació de FROM i FROM NAMED

Les clàusules FROM i FROM NAMED poden ser utilitzades en la mateixa consulta.

Dades:

```
# Default graph (stored at http://exemple.org/dft.ttl)
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://exemple.org/guillem> dc:publisher "Guillem Bassas" .
<http://exemple.org/ignasi> dc:publisher "Ignasi Pinto" .
```

```
# Named graph: http://exemple.org/guillem
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Guillem Bassas" .
_:a foaf:mbox <mailto:guillem@exemple.com> .
```

```
# Named graph: http://exemple.org/ignasi
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?who ?g ?mbox
FROM <http://exemple.org/dft.ttl>
FROM NAMED <http://exemple.org/ignasi>
FROM NAMED <http://exemple.org/guillem>
WHERE
{
  ?g dc:publisher ?who .
  GRAPH ?g { ?x foaf:mbox ?mbox }
}
```

Resultat de la consulta:

who	g	mbox
"Guillem Bassas"	<http://exemple.org/guillem>	<mailto:guillem@exemple.com>
"Ignasi Pinto"	<http://exemple.org/ignasi>	<mailto:ipinto@uoc.edu>

4.6.2 Consultar

Quan es consulta un conjunt de grafs s'utilitza la clàusula GRAPH.

Dades:

```
# Named graph: http://example.org/foaf/ignasiFoaf
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .
_:a foaf:knows _:b .

_:b foaf:name "Guillem Bassas" .
_:b foaf:mbox <mailto:guillem@exemple.com> .
_:b foaf:nick "Guille" .
_:b rdfs:seeAlso <http://exemple.org/foaf/guillemFoaf> .

<http://example.org/foaf/guillemFoaf>
  rdf:type foaf:PersonalProfileDocument .
```

```
# Named graph: http://example.org/foaf/guillemFoaf
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

_:z foaf:mbox <mailto:guillem@exemple.com> .
_:z rdfs:seeAlso <http://exemple.org/foaf/guillemFoaf> .
_:z foaf:nick "William" .

<http://example.org/foaf/guillemFoaf>
  rdf:type foaf:PersonalProfileDocument .
```

Consulta:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?src ?guillemNick
WHERE
{
  GRAPH ?src
  { ?x foaf:mbox <mailto:guillem@exemple.com> .
    ?x foaf:nick ?guillemNick }
}
```

Resultat de la consulta:

src	guillemNick
<http://example.org/foaf/ignasiFoaf>	"Guille"
<http://example.org/foaf/guillemFoaf>	"William"

4.7 Consultes

SPARQL té quatre tipus de consultes. Aquests tipus utilitzen les solucions des de parelles de resultats a *resultsets* o RDF grafs. Els tipus de consulta són:

- **SELECT**.- Retorna una taula de resultats.
- **CONSTRUCT**.- Retorna un graf RDF, basat en la consulta en una plantilla; permet construir un graf de resposta.
- **DESCRIBE**.- Retorna un graf RDF que descriu els recursos trobats; permet recuperar totes les metadades associades a un recurs (URI).
- **ASK**.- Retorna un booleà.

4.7.1 Solucions i resultats

Les dades de les solucions poden ser modificades per:

ORDER BY

Posa les soluciones en ordre. S'aplica una condició d'ordenació de les solucions basades en l'expressió especificada en la clàusula `ORDER BY`. La ordenació és ascendent per defecte, però es pot especificar explícitament que sigui ascendent o descendent mitjançant la condició `ASC()` o `DESC()`.

```
PREFIX foaf:      <http://xmlns.com/foaf/0.1/>

SELECT ?name
WHERE { ?x foaf:name ?name }
ORDER BY ?name
```

```
PREFIX      :      <http://exemple.org/ns#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX xsd:   <http://www.w3.org/2001/XMLSchema#>

SELECT ?name
WHERE { ?x foaf:name ?name ; :empId ?emp }
ORDER BY DESC(?emp)
```

PROJECTION

Guarda només les variables seleccionades. La solució pot ser transformada en un subconjunt de les variables.

Dades:

```
@prefix foaf:      <http://xmlns.com/foaf/0.1/> .

_:a foaf:name      "Ignasi Pinto" .
_:a foaf:mbox      <mailto:ipinto@uoc.edu> .

_:b foaf:name      "Guillem Bassas" .
:b foaf:mbox      <mailto:quillem@exemple.com> .
```

Consulta:

```

PREFIX foaf:      <http://xmlns.com/foaf/0.1/>

SELECT ?name
WHERE {
    ?x foaf:name ?name .
}

```

Resultat de la consulta:

name
"Guillem Bassas"
"Ignasi Pinto"

DISTINCT.- La solució pot ser modificada afegint la paraula clau `DISTINCT`, la qual assegura que cada solució en la seqüència és única.

Dades:

```

@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:mbox <mailto:ipinto@uoc.edu> .

_:z foaf:name "Ignasi Pinto" .
_:z foaf:mbox <mailto:ignasi@exemple.org> .

_:b foaf:name "Guillem Bassas" .
_:b foaf:mbox <mailto:guillem@exemple.com> .

```

Consulta:

```

PREFIX foaf:      <http://xmlns.com/foaf/0.1/>

SELECT DISTINCT ?name
WHERE {
    ?x foaf:name ?name
}

```

Resultat de la consulta:

name
"Ignasi Pinto"

OFFSET

Origina les solucions que comencen després d'un nombre especificat de solucions. Un `OFFSET` de zero no té cap efecte. En la utilització de `LIMIT` i `OFFSET` per seleccionar diferents subconjunts de solucions de la consulta s'ha d'utilitzar `ORDER BY`.

```

PREFIX foaf:      <http://xmlns.com/foaf/0.1/>

SELECT ?name
WHERE { ?x foaf:name ?name }
ORDER BY ?name
LIMIT 5
OFFSET 10

```


LIMIT

Restringeix el nombre de solucions tractades pel resultat de la consulta.

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>

SELECT ?name
WHERE { ?x foaf:name ?name }
LIMIT 20
```

4.7.2 Seleccionar variables

SELECT retorna directament les variables. La sintaxi `SELECT *` és una abreviació que selecciona totes les variables en una consulta.

Dades:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Ignasi Pinto" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .

_:b foaf:name "Guillem Bassas" .

_:c foaf:name "Andrea Martinez" .
_:c foaf:nick "AM" .
```

Consulta:

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>

SELECT ?nameX ?nameY ?nickY
WHERE
{ ?x foaf:knows ?y ;
  foaf:name ?nameX .
  ?y foaf:name ?nameY .
  OPTIONAL { ?y foaf:nick ?nickY }
}
```

Resultat de la consulta:

nameX	nameY	nickY
"Ignasi Pinto"	"Guillem Bassas"	
"Ignasi Pinto"	"Andrea Martinez"	"AM"

Capítol 5

Aplicació pràctica : *CercarFOAF*

5.1 Introducció

L'aplicació pràctica d'aquest projecte consisteix en la implementació d'una aplicació web senzilla, **CercarFOAF**, que permeti fer cerques sobre documents RDF, mitjançant el llenguatge de consulta SPARQL, amb dades relacionades amb la iniciativa FOAF.



Figura 5.1 - Logo de FOAF

FOAF (*Friend of a Friend*) [30] es una aplicació d'XML i RDF que pretén descriure la informació referent a les persones –dades personals, vincles entre les persones, els interessos, etc.- destinada a ser processada a l'entorn Web. Bàsicament és un arxiu RDF escrit en XML on es desa la descripció d'una persona, com si d'una pàgina web personal es tractés.

```
<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 rdf:nodeID="harry">
  <foaf:name>Harry Osborn</foaf:name>
  <rdfs:seeAlso rdf:resource="http://www.osborn.com/harry.rdf"/>
</foaf:Person>
<foaf:Person>
  <foaf:name>Peter Parker</foaf:name>
  <foaf:knows rdf:nodeID="harry"/>
  <foaf:knows>
    <foaf:Person>
      <foaf:name>Aunt May</foaf:name>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>
</rdf:RDF
```

Exemple 4 - Exemple d'un arxiu FOAF

A l'actualitat hi ha aplicacions de i per a FOAF:

- [FOAF People Map](#). Presenta una mapa complet de la Terra situant a tots aquells usuaris que van completar les seves dades de nearestAirport a la latitud i longitud corresponents. Necessita tenir un navegador amb suport d'SVG.

- [Foafanaut](#). Permet veure un usuari i tots aquells als quals coneix o quines persones el coneixen. Permet veure, també quin és el “camí més curt” entre dos usuaris diferents. També necessita tenir un navegador amb suport d'SVG.
- [Foaf Explorer](#). Permet veure la informació continguda en un arxiu FOAF sense necessitat d'instal·lar cap plug-in ni enviar l'arxiu pel seu processament. Tot es fa en temps real.
- [Foaf Web View](#). Mostra les dades d'una persona de forma similar a Foaf Explorer, però en aquesta aplicació s'ha d'enviar primer a URL de l'arxiu FOAF, ja que les dades són emmagatzemades i processades en el servidor.

Cal destacar [FOAF-a-Matic](#), que es tracta d'una aplicació Javascript que permet crear una descripció d'un mateix. Amb aquesta descripció, les dades personals seran compartides a la Web formant part d'un motor de cerca on serà possible cercar informació d'una persona en concret i de les comunitats de les quals es membre d'una forma senzilla i ràpida.

FOAF també es pot aplicar a la música. Per exemple, [Foafing the Music](#), desenvolupat dins del projecte europeu SIMAC (*Semantic Interaction with Audio Contents*) en el qual participa l'equip Music Technology Group (MTG) de la Universitat Pompeu Fabra de Barcelona. El sistema permet recomanar música a l'usuari en funció del seu perfil FOAF.

5.2 Anàlisi i Disseny

5.2.1 Descripció

CercarFOAF ha de permetre a un usuari fer consultes sobre un conjunt de dades relacionades amb el projecte FOAF, amb les quals es descriuen informació referent persones i les seves relacions, indicant algunes dades personals, quines altres persones coneix, quines aficions i interessos tenen, etc.

Tota aquesta informació s'expressarà mitjançant les propietats de FOAF, entre les quals podem destacar les següents:

- `mbox` : direcció correu electrònic
- `mbox_sha1sum` : direcció correu electrònic codificada
- `nick` : àlies
- `workplaceHomepage`: pàgina web del lloc de treball
- `homepage` : pàgina web personal
- `knows` : coneix
- `interest` : interès i/o afició

Poden existir diferents persones que comparteixen determinats atributs, i per aquest motiu s'utilitza el correu electrònic com a identificador únic d'una persona.

CercarFOAF es limita, principalment, a realitzar la cerca de les dades que es volen consultar, mitjançant la utilització d'un formulari que l'usuari ha d'omplir. A més a més l'aplicació web ha de permetre navegar a través de les dades resultants de la cerca, les quals que es troben emmagatzemades en memòria en arxius RDF i que són específics per a aquesta aplicació.

5.2.2 Programari

Per la realització de l'aplicació web s'ha utilitzat com a sistema gestor de bases de dades [Jena 2](#), el qual ha estat analitzat en els capítols anteriors; i com a servidor [Joseki](#), el qual implementa una interfície de consulta SPARQL per a Jena.

Així mateix, cal destacar que la programació de l'aplicació web s'ha realitzat amb Servlets de Java. Un Servlet és un programa que s'executa en un servidor web i que construeix una pàgina web que és retornada a l'usuari. El servidor web utilitzat és [Apache HTTP Server 2.2.3](#) i el servidor de Servlets/JSP (*contenedors de Servlets*) que s'ha fet servir en la present aplicació és [Apache Tomcat](#).

Atès que l'aplicació va adreçada a un usuari final, aquesta ha de ser accessible mitjançant una URL. En conseqüència, no es requereix cap instal·lació de programari específic a la màquina de l'usuari.

5.2.3 Funcionalitats

Atès que **CercarFOAF** és, bàsicament, una interfície per a que els usuaris puguin cercar informació referent a persones i les seves relacions mitjançant un formulari, el tipus d'accions que es poden realitzar són limitades.

A continuació s'entra en detall en les diferents accions i el comportament de l'aplicació davant de cada una.

Cercar persones

Un cop s'ha omplert el formulari amb les dades interessades per l'usuari i prémer el botó *Cercar* aquesta acció l'aplicació mostrarà una nova pàgina amb el nom de la persona cercada; cas que n'hi hagi més d'una persona l'aplicació mostrarà un llistat amb els noms de les persones cercades. Cas que no hi hagi cap persona que compleixi els paràmetres introduïts en el formulari l'aplicació ho notificarà.

Mostrar dades personals

Aquesta acció mostrarà en una nova pàgina la informació referent a les dades personals de la persona cercada, que quedaran restringides al seu correu electrònic, al seu àlies, al seu lloc web, al seu telèfon i al lloc web de la seva feina.

Mostrar correu electrònic

Aquesta acció mostrarà en una nova pàgina la informació referent al correu electrònic de la persona cercada. Aquesta dada s'utilitza com a identificador únic d'una persona.

Mostrar l' àlies

Aquesta acció mostrarà en una nova pàgina la informació referent a l' àlies de la persona cercada. Cas que no hi hagi cap informació l'aplicació ho notificarà

Mostrar el lloc web

Aquesta acció mostrarà en una nova pàgina la informació referent al lloc web de la persona cercada. Cas que no hi hagi cap informació l'aplicació ho notificarà

Mostrar l'adreça web del lloc de treball

Aquesta acció mostrarà en una nova pàgina la informació referent a l'adreça web del lloc de treball de la persona cercada. Cas que no hi hagi cap informació l'aplicació ho notificarà

Mostrar les persones que coneix

Aquesta acció mostrarà en una nova pàgina un llistat de les persones que coneix la persona cercada. En seleccionar una, mitjançant els diferents botons de la pàgina podrem accedir a una nova pàgina amb la informació referent a les dades personals, a les persones que coneix i a les aficions de la persona seleccionada. Cas que no hi hagi cap informació l'aplicació ho notificarà.

Mostrar les aficions i/o interessos

Aquesta acció mostrarà en una nova pàgina un llistat de les aficions i/o interessos de la persona cercada. En seleccionar una, i prémer el botó *Comuna a ...*, es mostrarà una nova pàgina amb un llistat de les persones que tinguin aquesta afició i/o interès. Cas que no hi hagi cap informació l'aplicació ho notificarà.

5.3 Casos d'ús

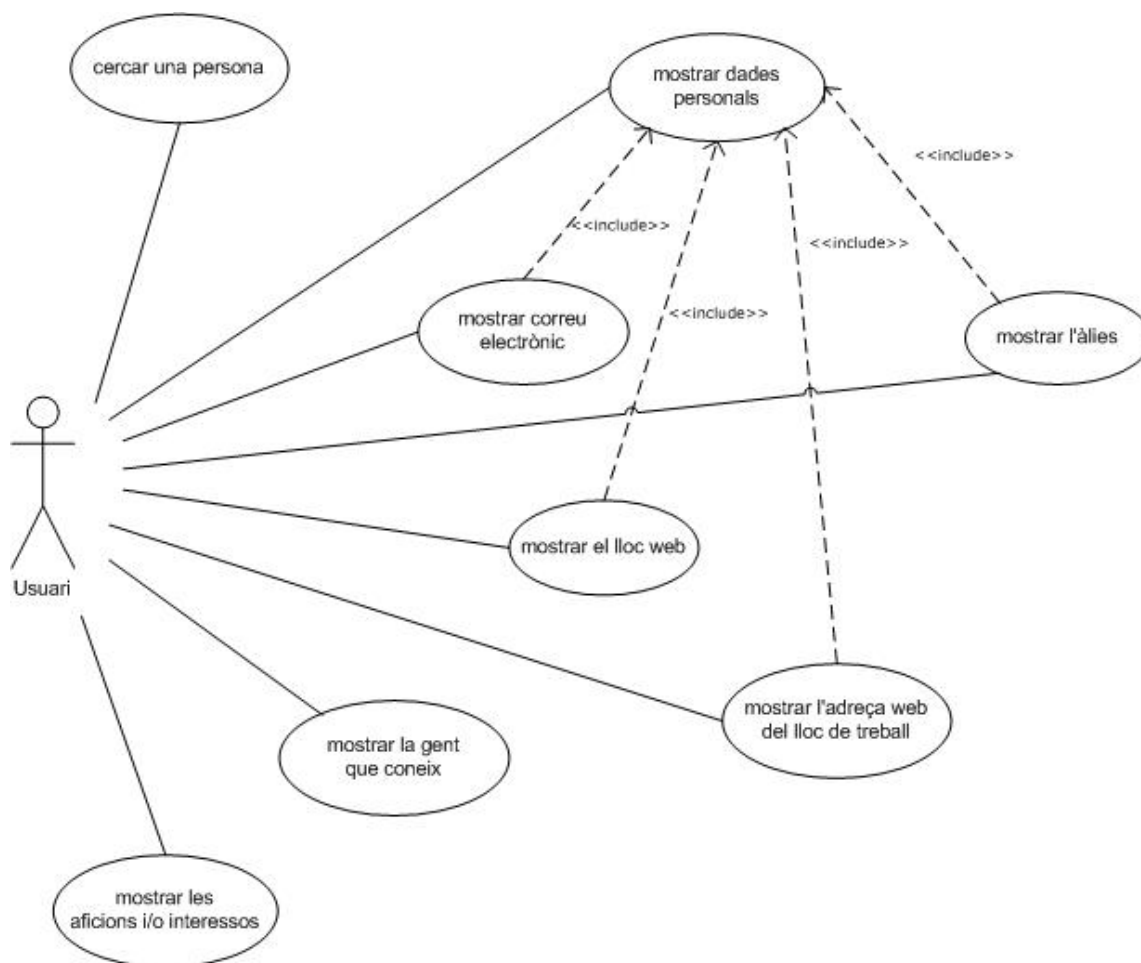


Figura 5.2 - Diagrama de casos d'ús de l'aplicació web

Cas d'ús : Cercar una persona	
Resum funcionalitat general	Permet la cerca d'una persona en funció de les dades introduïdes per l'usuari. Aquesta cerca donarà com a resultat el nom de la persona cercada. Cas que n'hi hagi més d'una persona permet mostrar un llistat amb els noms de les persones cercades.
Paper dins el treball de l'usuari	És el cas d'ús principal de l'usuari.
Actors	Usuari.
Casos d'ús relacionats	Cap
Precondició	L'usuari ha accedit a l'aplicació web i ha omplert adequadament el formulari de l'aplicació.
Postcondició	S'ha mostrat el nom de la persona cercada, o si és el cas, s'ha mostrat el llistat amb els noms de les persones cercades
Descripció	Mostra el nom de la persona cercada o bé, un llistat ordenat amb les persones cercades, les dades de les quals es troben emmagatzemades en memòria en arxius rdf.
Observacions	Cas que no hi hagi cap persona que compleixi els paràmetres introduïts en el formulari es notificarà a l'usuari.

Cas d'ús : Mostrar dades personals	
Resum funcionalitat general	Permet mostrar les dades personals corresponents al correu electrònic, l' àlies, el lloc web i l'adreça web del lloc de treball d'una persona.
Paper dins el treball de l'usuari	És un cas d'ús molt important, donat que permet conèixer totes les dades personals de la persona.
Actors	Usuari.
Casos d'ús relacionats	Mostrar el correu electrònic, mostrar el renom, mostrar el lloc web, mostrar l'adreça web del lloc de treball.
Precondició	L'usuari ha seleccionat el nom de la persona.
Postcondició	S'ha mostrat les dades personals de la persona seleccionada
Descripció	Mostra el nom, el cognom i el correu electrònic de la persona seleccionada, i també, cas que n'hi hagi, mostra l' àlies, el lloc web i l'adreça web del lloc de treball de la mateixa.
Observacions	Cas que no hi consti les dades corresponents al renom, o al lloc web, o a l'adreça web del lloc de treball, o bé a totes elles quedarà en blanc el camp corresponent a aquestes dades.

Cas d'ús : Mostrar el correu electrònic	
Resum funcionalitat general	Permet mostrar l'adreça del correu electrònic de la persona seleccionada. Cas que n'hi hagi més d'una adreça de correu electrònic permet mostrar un llistat amb aquestes.
Paper dins el treball de l'usuari	És un cas d'ús complementari que permet conèixer només les dades que ens interessin, el correu electrònic, de la persona seleccionada.
Actors	Usuari.
Casos d'ús relacionats	Mostrar dades personals.
Precondició	L'usuari ha seleccionat el nom de la persona.
Postcondició	S'ha mostrat l'adreça de correu electrònic de la persona seleccionada, o si és el cas, s'ha mostrat un llistat amb les adreces de correu electrònic de la mateixa.
Descripció	Mostra el correu electrònic de la persona seleccionada, o bé un llistat amb les adreces de correu electrònic de la mateixa.
Observacions	Cap

Cas d'ús : Mostrar l'àlies	
Resum funcionalitat general	Permet mostrar l' àlies de la persona seleccionada. Cas que n'hi hagi més d'un renom permet mostrar un llistat amb aquests.
Paper dins el treball de l'usuari	És un cas d'ús complementari que permet conèixer només les dades que ens interessin, l'àlies, de la persona seleccionada.
Actors	Usuari.
Casos d'ús relacionats	Mostrar dades personals.
Precondició	L'usuari ha seleccionat el nom de la persona.
Postcondició	S'ha mostrat l' àlies de la persona seleccionada, o si és el cas, s'ha mostrat un llistat amb els àlies de la mateixa
Descripció	Mostra l' àlies de la persona seleccionada, o bé un llistat ordenat amb els àlies de la mateixa.
Observacions	Cas que no hi hagi cap informació es notificarà a l'usuari.

Cas d'ús : Mostrar el lloc web	
Resum funcionalitat general	Permet mostrar l'adreça de la web personal de la persona seleccionada.
Paper dins el treball de l'usuari	És un cas d'ús complementari que permet conèixer només les dades que ens interessin, el lloc web, de la persona seleccionada.
Actors	Usuari.
Casos d'ús relacionats	Mostrar dades personals.
Precondició	L'usuari ha seleccionat el nom de la persona.
Postcondició	S'ha mostrat l'adreça de la web personal de la persona seleccionada.
Descripció	Mostra l'adreça de la web personal de la persona seleccionada.
Observacions	Cas que no hi hagi cap informació es notificarà a l'usuari.

Cas d'ús : Mostrar l'adreça web del lloc de treball	
Resum funcionalitat general	Permet mostrar l'adreça web del lloc de treball de la persona seleccionada.
Paper dins el treball de l'usuari	És un cas d'ús complementari que permet conèixer només les dades que ens interessin, l'adreça web del lloc de treball, de la persona seleccionada.
Actors	Usuari.
Casos d'ús relacionats	Mostrar dades personals.
Precondició	L'usuari ha seleccionat el nom de la persona.
Postcondició	S'ha mostrat l'adreça web del lloc de treball de la persona seleccionada.
Descripció	Mostra l'adreça web del lloc de treball de la persona seleccionada.
Observacions	Cas que no hi hagi cap informació es notificarà a l'usuari.

Cas d'ús : Mostrar les persones que coneix	
Resum funcionalitat general	Permet mostrar un llistat amb els noms o renoms de las persones que coneix la persona seleccionada. En seleccionar una d'aquest llistat ens permetrà accedir a les seves dades personals, a les persones que coneix i a les seves aficions.
Paper dins el treball de l'usuari	És un cas d'ús molt important, donat que permet conèixer dades que interessin de la persona.
Actors	Usuari.
Casos d'ús relacionats	Cap.
Precondició	L'usuari ha seleccionat el nom de la persona.
Postcondició	S'ha mostrat un llistat amb el nom o renom de les persones que coneix la persona seleccionada.
Descripció	Mostra un llistat ordenat amb els noms o renoms de les persones que coneix la persona seleccionada. Així mateix, en seleccionar una persona del llistat ens permetrà accedir a les dades relacionades amb aquesta.
Observacions	Cas que no hi hagi cap informació es notificarà a l'usuari.

Cas d'ús : Mostrar les aficions i/o interessos	
Resum funcionalitat general	Permet mostrar un llistat amb les aficions i/o interessos de la persona seleccionada.
Paper dins el treball de l'usuari	És un cas d'ús molt important, donat que permet conèixer dades relatives a aficions i/o interessos de la persona.
Actors	Usuari.
Casos d'ús relacionats	Cap.
Precondició	L'usuari ha seleccionat el nom de la persona.
Postcondició	S'ha mostrat un llistat amb les aficions i/o interessos de la persona seleccionada.
Descripció	Mostra un llistat ordenat amb les aficions i/o interessos de la persona seleccionada. Així mateix, en seleccionar una del llistat ens permetrà mostrar un llistat de les persones que tinguin la mateixa afició i/o interès.
Observacions	Cas que no hi hagi cap informació es notificarà a l'usuari.

5.4 Implementació

L'aplicació web **CercarFOAF** està implementada amb Servlets de Java. Un Servlet és un programa que s'executa en un servidor web i que construeix una pàgina web que és retornada a l'usuari. Aquesta pàgina construïda dinàmicament, en el cas de CercarFOAF, conté informació procedent dels documents RDF emmagatzemats a la base de dades.

S'ha pres la decisió d'utilitzar aquesta tecnologia pels següents motius:

- *Facilitat d'us*: L'estàndard de Servlets ofereix una infraestructura de desenvolupament d'aplicacions web proporcionant mètodes per anàlisi automàtic i descodificació de dades de formularis d'HTML. Accés a les capçaleres de peticions HTTP, seguiment, control i gestió de sessions, entre d'altres facilitats.
- *Potència*: Els Servlets poden compartir dades entre si, permetent compartir dades, connexions a bases de dades, etc.
- *Portabilitat*: Els Servlets poden executar-se en totes les plataformes que ofereixen suport de Java Servlets i amb independència del sistema operatiu i arquitectura *hardware*.

I, juntament amb Jena 2, com a sistema gestor de bases de dades per a documents RDF, i el servidor Joseki, que implementa una interfície de consulta SPARQL per a Jena, ens ha permet fer la implementació de l'aplicació web CercarFOAF per tal que sigui fàcil i intuïtiva per l'usuari final.

Cal remarcar que s'han implementat totes les funcionalitats descrites a l'apartat 5.2.3 d'aquest document. Les dades introduïdes al document RDF consultat i que es troba emmagatzemat en memòria, **foaf.rdf**, permet tractar totes les excepcions descrites a les funcionalitats, és a dir, es notifica a l'usuari quan no hi hagi cap informació de la persona seleccionada en relació a: l'àlies, l'adreça de la web personal, l'adreça de la web del lloc de treball, les persones conegudes i les aficions i interessos.

A l'aplicació s'accedeix mitjançant la URL: <http://ignasiuoc.no-ip.org/CercarFOAF/index.htm>, o bé mitjançant la instal·lació de l'aplicació amb el programari corresponent que l'usuari faci a la seva màquina. En aquest cas, s'ha creat l'arxiu **connexio.properties** per tal que es permeti modificar el host i el port per accedir als servlets emmagatzemats en el contenidor de servlets que tingui l'usuari, sense haver de modificar el codi dels diferents servlets.

5.4.1 Relació de servlets

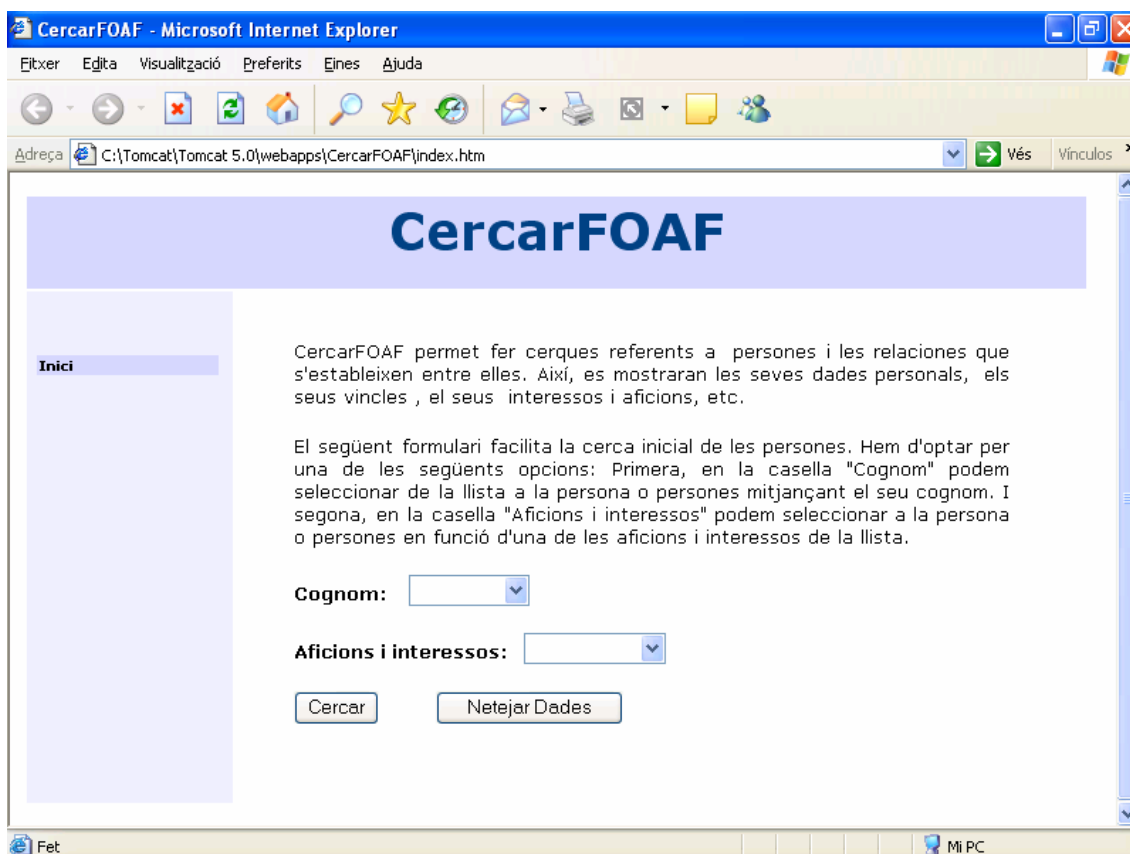
A continuació es detallen els servlets utilitzats a l'aplicació web, en que '*' correspon al nom de cada persona introduïda en el document RDF o, si és el cas, al nom de les aficions o interessos, per exemple `DadesPersonalsIgnasi.java` o `LlistaPersonesCine.java`. En funció del disseny gràfic de l'aplicació, hi ha servlets amb el nom de la persona que tenim en l'esmentat document RDF (per exemple, `Ignasi.java`) per tal que es mostri una barra de navegació personalitzada per a cada persona.

Formulari	Servlet que gestiona tota la informació del formulari i que respon a l'usuari amb una pàgina HTML que mostra el nom de la persona cercada, o si és el cas, el llistat amb les noms de les persones cercades.
DadesPersonals*	Servlet que realitza la consulta de la informació referent a les dades personals de la persona '*' i que respon a l'usuari amb una pàgina HTML amb aquestes dades.

Correu*	Servlet que realitza la consulta de la informació referent al correu electrònic de la persona '*' i que respon a l'usuari amb una pàgina HTML amb aquestes dades.
Alies*	Servlet que realitza la consulta de la informació referent a l'àlies de la persona '*' i que respon a l'usuari amb una pàgina HTML amb aquesta dada.
Web*	Servlet que realitza la consulta de la informació referent a l'adreça de la web personal de la persona '*' i que respon a l'usuari amb una pàgina HTML amb aquesta dada.
WebFeina*	Servlet que realitza la consulta la informació referent a l'adreça web del lloc de treball de la persona '*' i que respon a l'usuari amb una pàgina HTML amb aquesta dada.
Coneguts*	Servlet que realitza la consulta de la informació referent a la informació de les persones conegudes de la persona '*' i que respon a l'usuari amb una pàgina HTML amb aquestes dades.
Aficions*	Servlet que realitza de la consulta la informació referent a la informació de les aficions de la persona '*' i que respon a l'usuari amb una pàgina HTML amb aquestes dades.
LlistaPersonas*	Servlet que realitza la consulta de la informació referent a les aficions i interessos '*' i que respon a l'usuari amb una pàgina HTML amb un llistat de les persones que comparteixen aficions i interessos.

5.4.2 Disseny de les pantalles

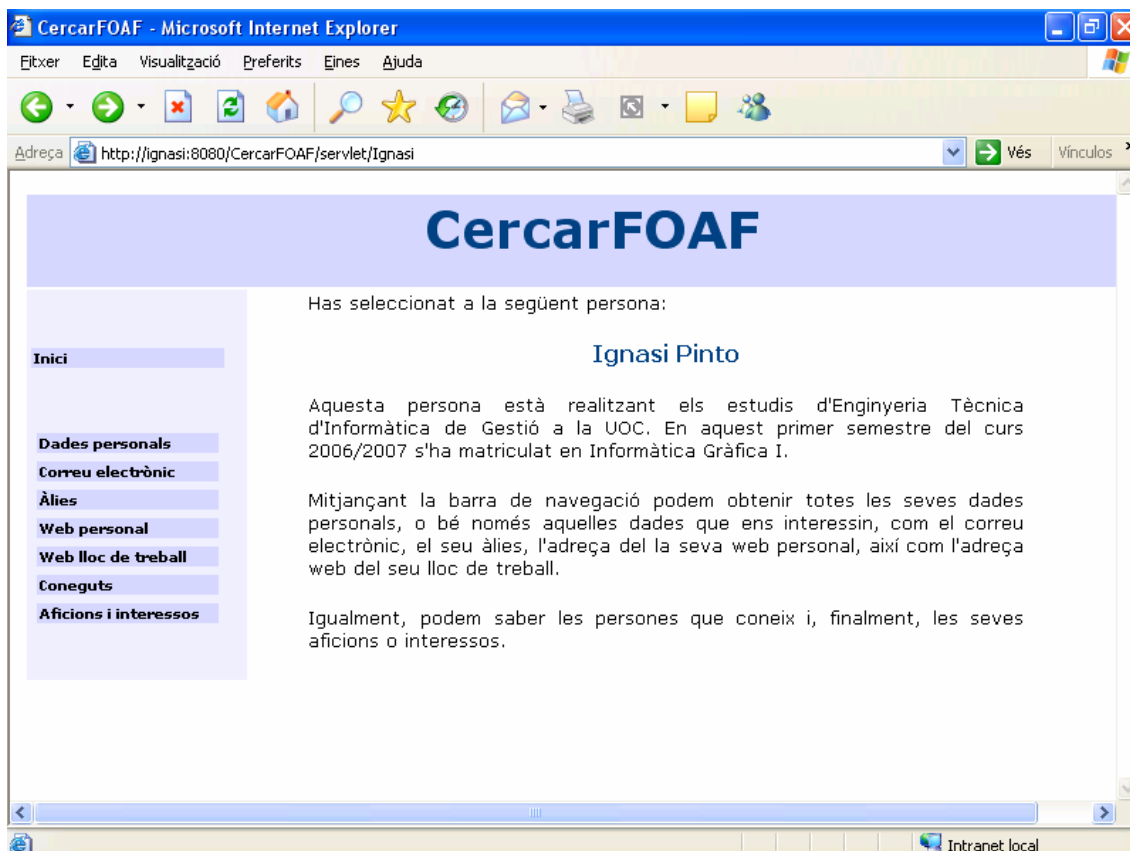
Pantalla corresponent a la pàgina principal de l'aplicació.



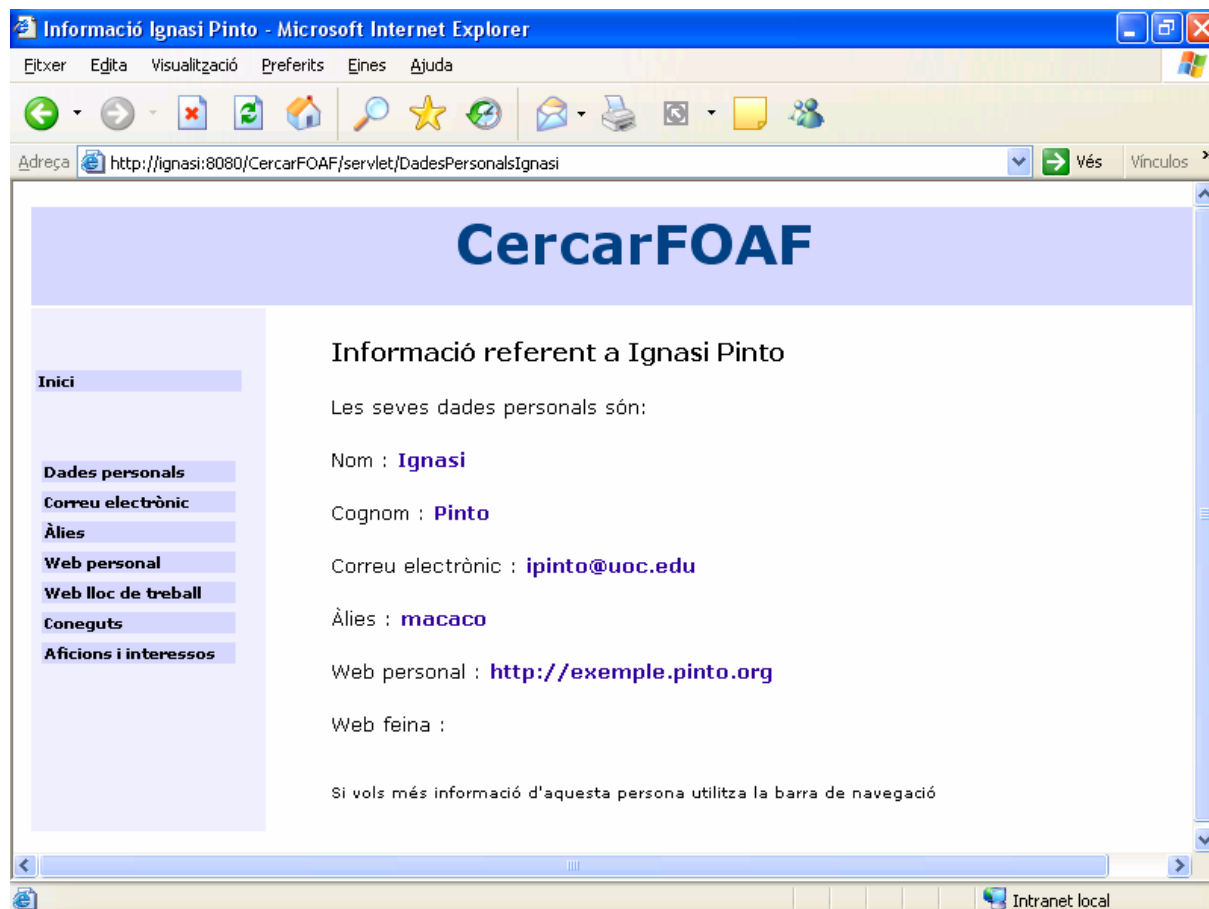
Pantalla que mostra la llista de persones cercades en funció de la seves aficions i interessos.



Pantalla que es mostra la pàgina de presentació de la persona cercada .



Pantalla que mostra les dades personals.



Pantalla que mostra l'adreça de correu electrònic.



Pantalla que mostra l'àlies.



Pantalla que mostra l'adreça de la web personal.



Pantalla que mostra l'adreça web del lloc de treball, observem que a la base de dades no existeix aquesta informació en relació a la persona seleccionada.



Pantalla que mostra els coneguts i les seves adreces web.

Informació Ignasi Pinto - Microsoft Internet Explorer

Fitxer Edita Visualització Preferits Eines Ajuda

Adreça <http://ignasi:8080/CercarFOAF/servlet/ConegutsIgnasi> Vés Vínculos >>

CercarFOAF

Inici

- Dades personals
- Correu electrònic
- Àlies
- Web personal
- Web lloc de treball
- Coneguts
- Aficions i interessos

Informació referent a Ignasi Pinto

Els seus coneguts i les seves adreces web són:

- **Frederic Segui** : <http://exemple.segui.org>
- **Ivan Benaiges** : <http://exemple.benaiges.org>
- **Montserrat Aran** : <http://exemple.aran.org>

Si vols informació dels seus coneguts dica sobre el seu nom, o bé si vols més informació d'aquesta persona utilitza la barra de navegació

Pantalla que mostra les aficions i interessos.

Informació Ignasi Pinto - Microsoft Internet Explorer

Fitxer Edita Visualització Preferits Eines Ajuda

Adreça <http://ignasi:8080/CercarFOAF/servlet/AficionsIgnasi> Vés Vínculos >>

CercarFOAF

Inici

- Dades personals
- Correu electrònic
- Àlies
- Web personal
- Web lloc de treball
- Coneguts
- Aficions i interessos

Informació referent a Ignasi Pinto

Les seves aficions i interessos són:

- **Lectura**
- **Musica**
- **Senderisme**

Si vols informació de les persones que comparteixen aquesta afició, clica sobre la mateixa.

Si vols més informació d'aquesta persona utilitza la barra de navegació

Capítol 6

Conclusions i possibles línies de futur

En aquest projecte s'ha fet una anàlisi introductori de la Web Semàntica, tant de la seva estructura com dels seus components, els llenguatges XML, RDF i OWL. La Web Semàntica és un projecte que pretén canviar la manera amb què fem servir la web amb l'objectiu que la Web passi de ser una col·lecció de documents a convertir-se en una base del coneixement.

Per a poder fer possible la Web Semàntica ha estat necessari el desenvolupament d'un conjunt de tecnologies que s'han analitzat en aquest treball. L'XML que proporciona les regles i la sintaxi per a la creació de documents estructurats dins la web; l'RDF, que facilita una infraestructura de dades que permet conferir semàntica a la web; i l'OWL, que permet la creació i definició d'ontologies estructurades dins la web per tal de descriure i representar les diverses àrees de coneixement.

Tot i que la Web Semàntica té molt pocs anys, està implantada en comunitats específiques dedicades a la medicina, comunicacions, informàtica, etc. Però aquests dominis només inclouen una petita part de coneixement estructurat del tot que representa la World Wide Web. La consolidació de la Web Semàntica significarà, doncs, un salt qualitatiu sobre el potencial de la web "actual"; però serà del tot necessari l'existència d'aplicacions que permetin la migració de tota la informació "actual" en un format estructurat que sigui compatible amb els estàndards de la Web Semàntica.

En el camí de la consolidació de la Web Semàntica es troba el desenvolupament dels sistemes gestors de bases de dades per a documents RDF. Actualment, hi ha aplicacions que permeten incloure la informació estructurada dins de la pàgina Web. Aquesta informació es troba escrita en llenguatge RDF i no s'emmagatzema dins de la mateixa pàgina web, sinó de forma externa en un repositori destinat específicament a mantenir les anotacions. Aquests repositoris són les bases de dades RDF i que es poden implementar sobre sistemes gestors de bases de dades per a documents RDF.

Hem analitzat diversos sistemes gestors de bases de dades, tals com Sesame, Jena, Redland i Kaon. Tots aquets SGBDs tracten, emmagatzemen i gestionen la informació classificada segons uns criteris semàntics i interrelacionada amb conceptes afins, alhora que permeten la comunicació entre sistemes de manera transparent a l'usuari. Un dels aspectes comuns a tots ells és l'emmagatzematge dels documents RDF. Ofereixen emmagatzematge en memòria i en bases de dades relacionals, com MySQL, PostgreSQL i Oracle, entre d'altres.

El desenvolupament i expansió de la Web Semàntica provocarà que les bases de dades que treballin amb documents RDF continui creixent. I possiblement les grans companyies de bases de dades relacionals ofereixin eines i funcionalitats d'emmagatzemament i manipulació de documents RDF, conjuntament amb les capacitats SQL i relacionals que ja ofereixen a l'actualitat.

Un altra element de consolidació de la Web Semàntica és el llenguatge de consultes SPARQL per a informació representada en RDF, i que ha estat el cas d'estudi d'aquest projecte. També hem vist que tots els sistemes gestors de bases de dades analitzats suporten l'SPARQL com a llenguatge de consultes, entre d'altres llenguatges com RDQL o SeRQL.

Amb l'SPARQL es pretén proporcionar un llenguatge de consultes per buscar informació específica de documents RDF mitjançant qualsevol eina, en comptes que sigui necessari aprendre el llenguatge específic de cada eina. A l'actualitat l'SPARQL està en procés d'estandardització pel DAWG (RDF Data Acces Working Group) del W3C, havent estat candidata de recomanació el d'abril de i havent-ne publicat l'últim esborrany de treball en data 4 d'octubre de 2006.

L'SPARQL ofereix un camí per presentar i utilitzar els resultats de recerques a través d'una gran varietat d'informació com pot ser dades personals (com a exemple, l'aplicació web CercarFOAF implementada en aquest projecte), xarxes socials i metadades sobre recursos digitals, com música o imatges. L'SPARQL també proporciona un camí d'integració sobre recursos diferents. A més, permet extreure informació en diverses formes, incloent URIs, extreure subgrafs RDF i construir nous grafs RDF basats en la informació dels grafs consultats.

CercarFOAF és una aplicació basada en la idea d'un grup de persones que tenen o realitzen alguna activitat comuna. En un document RDF s'han recopilat dades basades en FOAF sobre aquestes persones, i a partir dels enllaços persona a persona es pot crear una xarxa social representada en metadades i que és processable amb la tecnologia basada en la Web Semàntica.

Aquesta aplicació constitueix un exemple il·lustratiu d'ús de tecnologia basada en la Web Semàntica, ja que a partir de la recopilació dades, aquestes s'emmagatzemen en documents RDF i posteriorment es poden analitzar per extreure més coneixement de la informació

Possibles línies de futur

En aquest projecte el cas d'estudi concret ha estat el llenguatge de consultes SPARQL, el qual serà l'estàndard per a informació representada en RDF. Aprofitant tota la potencialitat que té aquest llenguatge, una possible línia de futur podria ser la realització d'un estudi sobre la interacció entre els sistemes gestors de bases de dades per a documents RDF en recerques d'informació entre diferents comunitats que tinguin punts en comú.

És previsible que amb l'estandardització de l'SPARQL els sistemes gestors de base de dades relacionals (Oracle, MySQL, MS SQL Server, PostgreSQL, DB2, ...) ofereixin eines i funcionalitats d'emmagatzemament i manipulació de documents RDF, conjuntament amb les capacitats SQL i relacionals que ja ofereixen a l'actualitat. Per tant, una altra possible línia de futur podria ser analitzar com les principals bases de dades relacionals treballen i permeten la manipulació de documents RDF.

La implementació de l'aplicació CercarFOAF ha estat centrada fonamentalment en el cas d'estudi objecte d'aquest projecte: el llenguatge de consultes SPARQL. Per la qual cosa, la seva única funcionalitat, en termes generals, es la consulta de documents RDF. Una possible extensió de CercarFOAF seria poder afegir, modificar i suprimir dades d'aquests documents RDF, i amb diferents nivells de permisos, per tal que alguns usuaris només puguin fer consultes, i d'altres, a més, puguin afegir, modificar o suprimir dades dels documents RDF.

Bibliografia

- [1] *Llenguatges de consulta a l'àmbit de la Web Semàntica. Cas d'estudi SPARQL / Pla de treball*. Ignasi Pinto. Setembre 2006
- [2] <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html> - *Semantic Web – XML 2000. Architecture*. Tim Bernes-Lee
- [3] <http://www.w3.org/2001/12/semweb-fin/w3csw> - Marja-Riitta Koivunen and Eric Miller, *W3C Semantic Web Activity*, Proceedings of the Semantic Web Kick-off Seminar in Finland. November 2001
- [4] <http://www.w3.org/TR/2006/REC-xml-20060816/> - W3C Recommendation, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, 29 September 2006
- [5] <http://www.w3.org/XML/1999/XML-in-10-points.ca> - XML en 10 punts
- [6] <http://www.w3.org/XML/Schema> - XML Schema
- [7] <http://www.w3.org/RDF> - Secció del *W3 Consortium* encarregada de la part que correspon a RDF.
- [8] <http://www.w3.org/TR/rdf-primer/> - RDF Primer. W3C Recommendation 10 February 2004
- [9] <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/> - RDF/XML Syntax Specification (Revised)
- [10] <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/> - RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004
- [11] <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/> - RDF Schema Specification 1.0. W3C Candidate Recommendation 27 March 2000
- [12] <http://www.w3.org/2004/01/sws-pressrelease.html.en> - World Wide Web Consortium Issues RDF and OWL Recommendations
- [13] <http://www.w3.org/TR/owl-ref/> - OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004
- [14] <http://www.w3.org/TR/owl-guide/> - OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004
- [15] <http://www.cs.vu.nl/~frankh/postscript/ISWC02.pdf> - Sesame: A Generic Architecture for Storing and Querying RDF and RDFSchemas. Jeen Broekstra, Arjohn Kampman i Frank Van Harmelen
- [16] <http://jena.sourceforge.net/> - Jena: A Semantic Web Framework for Java
- [17] <http://www.hpl.hp.com/techreports/2003/HPL-2003-266.pdf> - Efficient RDF Storage and Retrieval in Jena2. Kevin Wilkinson, Craig Sayers, Harumi Kuno i Dave Reynolds.

- [18] <http://librdf.org/index.html/> - Redland RDF Application Framework
- [19] <http://www10.org/cdrom/papers/490/> - The Design and Implementations of the Redland RDF Application Framework - David Beckett
- [20] <http://kaon2.semanticweb.org/> - KAON2
- [21] <http://www2003.org/cdrom/papers/alternate/P029/p29-volz.html> - KAON SERVER - A Semantic Web Management System.
- [22] <http://www.openrdf.org/doc/sesame/users/index.html> - User Guide for Sesame
- [23] http://jena.sourceforge.net/tutorial/RDF_API/index.html - An Introduction to RDF and the Jena RDF API
- [24] <http://jena.sourceforge.net/javadoc/index.html> - The Jena Javadoc (Public APIs)
- [25] <http://www.openrdf.org/doc/sesame2/system/index.html> - System Documentation for Sesame 2.0 (DRAFT)
- [26] <http://www.openrdf.org/doc/sesame2/api/> - Sesame 2 API Javadoc
- [27] <http://www.w3.org/TR/rdf-sparql-query/> - SPARQL Query Language for RDF. W3C Working Draft 4 October 2006
- [28] <http://www.w3.org/TR/2006/CR-rdf-sparql-protocol-20060406/> - SPARQL Protocol for RDF. Candidate Recommendation 6 April 2006
- [29] <http://www.w3.org/TR/2006/CR-rdf-sparql-XMLres-20060406/> - SPARQL Query Results XML Format. RDF. Candidate Recommendation 6 April 2006
- [30] <http://xmlns.com/foaf/0.1/> - FOAF Vocabulary Specification