

Ontologies Turístiques Geogràfiques

XML i Web semàntica

Alumna: **Gemma Pallàs Saura**
Dirigit per: **Joan Anton Pérez Braña**

Treball Fi de Carrera
Enginyeria Tècnica en Informàtica de Gestió
UOC. Segon semestre 2010-2011

Resum

En els propers anys, la Web semàntica i les tecnologies que porta associades sembla que constituïran els eixos vertebradors d'una nova manera d'entendre la Xarxa. Aquesta deixarà de ser un mer repositori d'informació per convertir-se en un ens capaç de generar coneixement per si sol, on màquines i usuaris treballin de manera col·laborativa.

En aquest treball s'analitzarà l'evolució que ha sofert el mitjà i el significat d'una Web semàntica, passant per algunes de les tecnologies que hi donen suport. Per a entendre com es confereix significat a l'estructura d'una web, s'estudiaran de manera incremental els diferents llenguatges que permeten construir-la: *XML*, *RDF* i *OWL*. L'estructura conceptual que pren la informació, la representació de la qual és codificada amb els llenguatges mencionats anteriorment, es tractarà tant en un vessant teòric com en un de pràctic. Així, s'analitzarà el concepte i el procés de creació d'una ontologia i, mitjançant el programari *Protégé*, es crearà una taxonomia adequada a l'àmbit del turisme o lleure de muntanya.

Un altre aspecte que s'analitzarà en el document és la nova generació de bases de dades que ha sorgit com a conseqüència del nou suport de la informació, on preval la relació que s'estableix entre les dades per sobre d'aquestes. Així doncs, s'estudiaran les característiques bàsiques de les bases de dades *XML* natives i es farà una breu introducció als llenguatges de modificació i consulta que suporten. Aquests aspectes s'aplicaran de manera pràctica sobre la base de dades *eXist*, mitjançant l'ús de les instàncies de l'ontologia creada.

Finalment es desenvoluparà una aplicació web que permeti portar a la pràctica algunes de les tecnologies estudiades. El context en què se centrarà el cas pràctic serà el mateix que s'ha representat mitjançant l'ontologia i permetrà comprovar el funcionament d'una web explotada de manera semàntica.

Índex de continguts

1. Introducció	5
1.1. Justificació del treball	5
1.2. Objectius del TFC	6
1.3. Enfocament i mètode seguit	6
1.4. Planificació del projecte	7
1.5. Productes obtinguts	8
1.6. Estructura de la memòria	8
2. Evolució històrica de la Web	9
2.1. La Web tradicional	9
2.2. La Web social	9
2.3. La Web semàntica	10
3. La Web semàntica	11
3.1. Limitacions que presenta la Web tradicional	11
3.2. La Web semàntica i les tecnologies que té associades	12
4. El llenguatge XML	15
4.1. El concepte d'XML	15
4.2. Estructura d'un document XML	15
4.3. Validesa d'un document XML	17
4.3.1. DTD	17
4.3.2. XML Schema	18
4.4. L'espai de noms	18
5. El model RDF	20
5.1. El concepte d'RDF	20
5.2. La identificació de recursos: URI	21
5.3. El vocabulari principal d'RDF/XML	22
5.4. RDF Schema	24
6. El llenguatge OWL	25
6.1. El concepte d'OWL	25
6.2. Estructura d'un document OWL	26
6.3. Els subllenguatges d'OWL	27
6.3.1. Els components del subllenguatge OWL Lite	28
6.3.2. Els components dels subllenguatges OWL DL i OWL Full	29
7. L'ontologia	30
7.1. El concepte d'ontologia	30
7.2. Tipus d'ontologies	31
7.3. Components de l'ontologia	31
7.4. Cicle de desenvolupament d'una ontologia	32
7.5. Ontologies turístiques geogràfiques	33
7.6. Protégé: un editor d'ontologies OWL	34
7.6.1. Instal·lació del programari	34
7.6.2. Construcció de l'ontologia amb Protégé	35
8. Les bases de dades XML natives	40
8.1. Què és una base de dades XML	40
8.2. Característiques de les bases de dades XML natives	40
8.3. Els llenguatges de consulta de documents: XPath i XQuery	41
8.3.1. XPath	41
8.3.2. XQuery	42
8.4. Els llenguatges de modificació de documents: XUpdate i XQuery Update	43
8.4.1. XUpdate	43
8.4.2. XQuery Update	44
8.5. El SGBD eXist	45
8.5.1. Instal·lació	45
8.5.2. Emmagatzemament, consulta i modificació d'instàncies de l'ontologia	45
9. Aplicació pràctica	49
9.1. Anàlisi de requeriments	49
9.2. Anàlisi funcional	50
9.2.1. Consultar una instància	51

9.2.2. Donar d'alta una instància	54
9.2.3. Eliminar una instància	55
9.2.4. Modificar una instància	56
9.3. Arquitectura de l'aplicació i tecnologies emprades	57
10. Conclusions i línies de futur	58
10.1. Conclusions	58
10.2. Línies de Futur	59
11. Bibliografia	60

Índex de figures

Figura 1 – Taula de fases de treball	7
Figura 2 – Calendari de desenvolupament del TFC	7
Figura 3 – Comparació web 1.0, 2.0 i 3.0	10
Figura 4 – Estructura de la Web semàntica	14
Figura 5 – Exemple de document XML	16
Figura 6 – Estructura conceptual d'un document XML	16
Figura 7 – Exemple de document XML que fa servir un DTD	17
Figura 8 – Exemple de DTD	17
Figura 9 – Exemple de document XML Schema	18
Figura 10 – Exemple de declaració de namespace	19
Figura 11 – Exemple de traducció de llenguatge natural a triplets RDF	21
Figura 12 – Graf dirigit que representa triplets RDF	21
Figura 13 – Graf dirigit que substitueix literals per URIs	22
Figura 14 – Taula de cims	22
Figura 15 – Exemple de document RDF sobre la taula de cims	22
Figura 16 – Exemple de document RDF on el predicat és entès com a atribut	23
Figura 17 – Exemple de document RDF on el predicat és entès com a recurs	23
Figura 18 – Exemple de document RDF Schema	24
Figura 19 – Exemple de codificació d'una ontologia mitjançant OWL	26
Figura 20 – Nivell de complexitat dels llenguatges de representació de coneixement	27
Figura 21 – Taula de components del llenguatge OWL Lite	28
Figura 22 – Taula de components addicionals dels llenguatges OWL DL i OWL Full	29
Figura 23 – Captura de Protégé (versió 4)	34
Figura 24 – Primera aproximació als elements del domini	35
Figura 25 – Ontologia de Muntanya representada gràficament amb OWLViz	36
Figura 26 – Llistat de relacions dins l'ontologia	37
Figura 27 – Llistat d'atributs de l'ontologia	37
Figura 28 – Ontologia de muntanya representada gràficament amb OntoViz	38
Figura 29 – Llistat d'instàncies de l'ontologia	39
Figura 30 – Expressions XPath	41
Figura 31 – Travesses.xml	42
Figura 32 – Taula d'exemples de consultes XPath	42
Figura 33 – Exemple de consulta XQuery FLWOR	43
Figura 34 – Exemple de modificació de dades amb XUpdate	44
Figura 35 – Exemple de modificació de dades amb XQuery Update	44
Figura 36 – Captura d'eXist	45
Figura 37 – Extracte d'una instància XML de l'ontologia de muntanya	46
Figura 38 – Distribució de la informació dins d'eXist	46
Figura 39 – Realització d'una consulta a eXist mitjançant XQuery (1)	47
Figura 40 – Realització d'una consulta a eXist mitjançant XQuery (2)	48
Figura 41 – Realització d'una inserció a eXist mitjançant XQuery Update	48
Figura 42 – Diagrama de casos d'ús de l'aplicació	50
Figures 43 – 54 – Esquemes de disseny i captures de l'aplicació	51
Figura 55 – Diagrama de desplegament	57

1. Introducció

El Treball Fi de Carrera (TFC) és una assignatura que té com a objectiu realitzar un treball de síntesi que requereixi posar en pràctica els coneixements adquirits al llarg dels estudis. Mitjançant l'anàlisi d'un problema complex de tipus pràctic i establint una metodologia i una planificació de treball adequades es pretén sintetitzar una solució viable i realista al problema proposat, tot elaborant una memòria que permeti presentar el plantejament i desenvolupament del projecte i que en reculli els resultats.

En aquest capítol veurem quins són els objectius que persegueix aquest treball (*apartat 1.2*) i quina és la metodologia i la planificació que s'ha seguit per a la seva elaboració (*apartats 1.3 i 1.4*). Finalment descriurem els productes que s'han obtingut com a resultat del treball (*apartat 1.5*) i farem una breu explicació sobre l'estructura amb què es desenvoluparà el present document (*apartat 1.6*).

1.1. Justificació del treball

Des que es va originar a principis dels anys noranta, la *World Wide Web* ha esdevingut un gran repositori d'informació que ha permès contenir el coneixement humà de manera col·lectiva. Si bé des del principi la seva vasta extensió ha estat el seu punt fort, el cert és que darrerament aquest mateix paràmetre ha posat de manifest la insuficient existència d'eines que facilitin la localització d'un coneixement concret dins l'entorn que el conté. Per aquest motiu, i com a resposta al repte de gestionar eficientment l'extensa quantitat d'informació de la Xarxa, en la darrera dècada ha sorgit una nova línia de treball en l'àmbit web basada en el tractament de dades semi-estructurades. Mitjançant el concepte de Web semàntica i les tecnologies que porta associades, es pretén capacitar els ordinadors semànticament, de manera que no tant sols puguin manipular informació, sinó que també puguin comprendre-la.

Amb el sorgiment d'aquesta nova manera d'entendre la informació, és inevitable haver de plantejar noves maneres de tractar les dades contingudes a la Web, amb independència de la font o del format que tinguin. Per aquesta raó calen nous llenguatges i nous models que puguin definir i tractar les dades de manera semàntica. D'igual manera, es fa palès que serà necessari un concepte evolucionat de base de dades, la qual sigui capaç de gestionar l'esmentada informació semi-estructurada i emmagatzemar-la de forma eficient traient-ne el màxim de potencial.

Deixant de banda el vessant més pràctic, aquest treball busca estudiar els conceptes bàsics de la Web semàntica i conèixer tant l'estructura d'alguns llenguatges i models de representació d'informació a la Web, com els sistemes d'emmagatzemament d'aquesta informació en bases de dades. I és que el futur immediat de la Xarxa sembla que es veurà plenament condicionat per aquesta redefinició tecnològica i conceptual del propi mitjà.

1.2. Objectius del TFC

Un cop determinat quin serà el marc de treball general del projecte, podem establir els objectius que es pretenen assolir mitjançant l'elaboració del present estudi:

- **Analitzar l'estat de l'art de la Web semàntica.** Conèixer l'evolució que ha sofert la Web, determinar quins elements són propis de la Web semàntica i buscar exemples concrets enfocats a turisme.
- **Conèixer i aprendre a crear ontologies.** Estudiar què és una ontologia i treballar amb el programari *Protégé* per tal de construir-ne una de pròpia sobre la temàtica associada al cas pràctic i aprendre a realitzar-ne instàncies.
- **Conèixer el llenguatge de modelat de dades XML.** Entendre la seva estructura i funcionament i analitzar les característiques que té un document XML. Fer una visió sobre els elements que hi tenen relació per donar-hi estructura: el llenguatge XML Schema i la definició DTD. Conèixer els llenguatges que confereixen major significat sobre XML: RDF i OWL.
- **Estudiar les bases de dades XML natives.** Analitzar quina estructura tenen i com emmagatzemen les instàncies, particularitzant-ho de manera pràctica mitjançant la base de dades *eXist*.
- **Analitzar els llenguatges de consulta de les bases de dades XML natives.** Estudiar els llenguatges *Xpath*, *Xquery* i *Xupdate* i determinar com s'utilitzaran en l'aplicació que es desenvoluparà.
- **Desenvolupar un cas pràctic on s'apliquin els conceptes estudiats.** Implementar una aplicació web que mitjançant una interacció inicial amb l'usuari, on es pugui especificar la zona geogràfica, el tipus d'informació a la qual es vol accedir i el seu format, recuperi d'una base de dades XML nativa aquells fitxers que presentin concordança amb les preferències mostrades. La temàtica del treball es particularitzarà implementant una aplicació per a excursionistes que permeti a l'usuari consultar informació sobre elements propers a la seva localització geogràfica. Els elements que es recuperaran de la base de dades podran ser rutes, poblacions, monuments, etc. i el format de la informació podrà ser text, vídeo, fotografia, *track GPS*...

1.3. Enfocament i mètode seguit

Partint dels objectius definits en l'apartat anterior es poden detectar quatre grans àmbits de treball que determinaran, en bona mesura, la metodologia que se seguirà per al desenvolupament del projecte. La primera d'elles té un component eminentment teòric i la resta d'àrees apliquen, de manera pràctica i sobre un cas real, els conceptes estudiats. Aquestes àrees són:

- a) **Estudi dels conceptes teòrics sobre la Web semàntica** (*capítols 2, 3, 4, 5, 6, primera part del 7 i primera part del 8*). Àmbit que correspon a l'anàlisi genèrica dels conceptes que defineixen la Web semàntica i les tecnologies que té associades.

- b) **Disseny d'una ontologia pròpia** (*segona part del capítol 7*). Desenvolupada mitjançant el programari *Protégé* i que dóna resposta als requeriments de context establerts en el cas pràctic.
- c) **Treball amb una base de dades XML nativa** (*segona part del capítol 8*). Acció que es materialitza mitjançant l'elaboració d'un conjunt d'instàncies de l'ontologia i emmagatzemant els documents resultants en la base de dades *eXist* per, a continuació, procedir a la seva manipulació mitjançant llenguatges de consulta/modificació específics.
- d) **Implementació de l'aplicació** (*capítol 9*). Àrea que correspon a la definició i disseny de l'aplicació que posarà en pràctica els conceptes estudiats tot vinculant les estructures mencionades en els dos apartats precedents (ontologia i base de dades) amb un programari de gestió, control i visualització.

1.4. Planificació del projecte

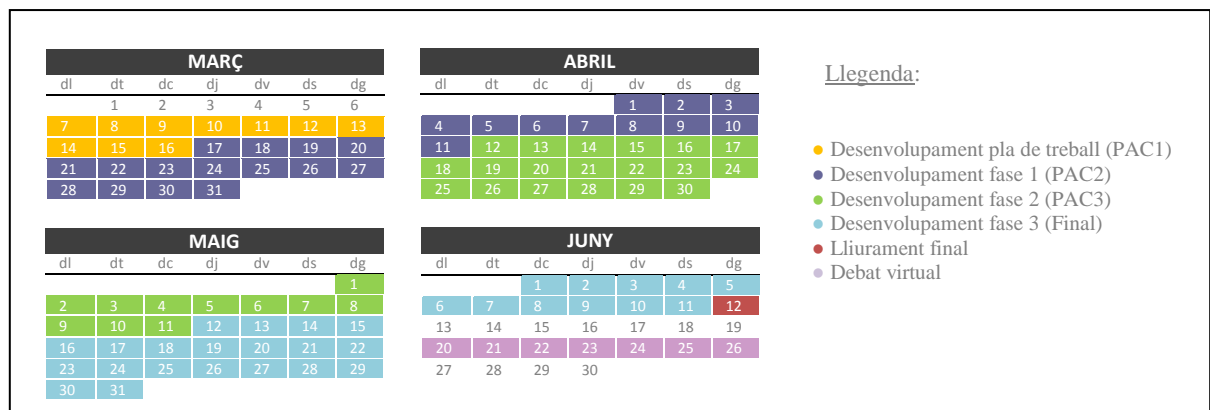
Per tal d'ordenar i sistematitzar totes les accions requerides per al desenvolupament del TFC, es va elaborar un pla de treball que distribuïa al llarg del semestre les quatre àrees de citades anteriorment. A causa de la considerable extensió del document –que incloïa calendaris, diagrames i taules– s'ha estimat preferible fer-ne una taula-calendari resum que té en compte els reajustaments efectuats:

Figura 1

Fase de treball	Tasques associades
Pla de treball	Elaboració de diagrames de <i>Gantt</i> i taules associades per planificar el treball i realització de les primeres cerques d'informació.
Fase 1	Desenvolupament del marc teòric del treball (<i>capítols 1, 2, 3, 4, 5, 6, part del 7 i part del 8</i>) i elaboració del primer esbós de l'ontologia.
Fase 2	Desenvolupament complet de l'ontologia, realització d'instàncies concretes i treball amb la base de dades <i>eXist</i> . Finalització dels <i>capítols 7 i 8</i> de la memòria i disseny de l'aplicació (<i>capítol 9</i>).
Fase 3	Implementació de l'aplicació, repàs d'aspectes formals de la memòria i elaboració de la conclusió. Realització del vídeo de presentació.

Taula de fases de treball.

Figura 2



Calendari de desenvolupament del TFC

1.5. Productes obtinguts

Com a resultat de l'elaboració del TFC s'han obtingut els següents productes:

- ✓ Memòria del projecte.
- ✓ Vídeo de presentació. Resum del treball realitzat i demostració de funcionament de l'aplicació.
- ✓ Projecte de l'ontologia.
- ✓ Col·lecció d'instàncies.
- ✓ Aplicació. Fitxers i programari necessari per al funcionament del cas pràctic.

1.6. Estructura de la memòria

En els primers capítols d'aquesta memòria s'establirà el context en què es desenvoluparà el projecte. Així en el *capítol 2* s'analitzarà breument l'evolució que ha sofert la Web des del seu origen i fins al darrer salt tecnològic (la Web semàntica), veient quines característiques han determinat cada etapa. En el *capítol 3* s'estudiarà detalladament la Web semàntica, analitzant el motiu pel qual era necessari redefinir la Web i s'enumeraran les diferents tecnologies que té associades.

A continuació s'introduiran, de manera incremental, les diferents tecnologies associades a la Web semàntica. Així, en el *capítol 4* s'estudiarà el llenguatge *XML*, en el *capítol 5* el model *RDF* i en el *capítol 6* el llenguatge *OWL*.

En els *capítols 7 i 8* es tractaran el concepte d'ontologia i les bases de dades *XML* natives, respectivament. Aquests dos capítols tindran al mateix temps un vessant teòric i un de pràctic. D'una banda s'hi estudiaran els conceptes teòrics associats a la tecnologia concreta que s'anuncia en el capítol, i de l'altra posaran en pràctica aquests coneixements mitjançant la realització d'un cas pràctic: el disseny d'una ontologia sobre el context del lleure de Muntanya, en el primer cas, i la instanciació d'aquesta ontologia mitjançant una col·lecció de documents *XML* que puguin ser emmagatzemats en una bases de dades *XML* nativa, en el segon cas.

Tot seguit es descriuran tant el disseny com la implementació del cas pràctic (*capítol 9*). Mitjançant l'ontologia creada en el *capítol 7* i treballant amb els documents *XML* obtinguts en el *capítol 8*, es dissenyarà una aplicació que permeti consultar la col·lecció de documents *XML* (que estaran emmagatzemats a la base de dades) en funció de certs paràmetres de cerca especificats per l'usuari.

Finalment, en el *capítol 10* es presentaran les conclusions i les possibles línies de futur del treball.

2. Evolució històrica de la Web

Des que el 1989 *Berners-Lee* va presentar davant del *CERN*¹ el document que esdevindria l'embrió del que avui coneixem com *World Wide Web*, l'evolució d'aquest instrument de compartició de coneixement ha estat imparable. La seva expansió i arrelament en la societat ha superat amb escreixos el d'altres mitjans coetanis (televisió, telefonia, etc) –dels quals ha acabat adoptant funcionalitats– esdevenint una de les revolucions tecnològiques més rellevants de la història.

En aquest capítol veurem l'evolució que ha sofert el mitjà i els diferents salts tecnològics que ha experimentat des de la seva creació i fins a dia d'avui.

2.1. La Web tradicional²

En un primer moment, abans que apareguessin els primers estàndards web, Internet era un amalgama d'imatges i d'arxius de text normal, als quals s'accedia mitjançant navegadors només text. Poc després però, amb l'especificació del llenguatge *HTML*³ –el qual permetia vertebrar l'estructura de la informació a nivell de presentació– l'estètica visual va passar a ser un factor determinant en la programació de pàgines: el contingut havia de fer-se visualment atractiu per a l'usuari. Com a conseqüència de la nova especificació tècnica de la Web, els navegadors *només text* aviat van veure's substituïts per navegadors gràfics (*Internet Explorer*, *Netscape*...), els quals permetien visualitzar la informació continguda en les pàgines. És justament en aquest punt històric on podem situar la veritable arrencada de la Web.

La ràpida expansió d'aquest mitjà i l'increment exponencial d'usuaris, va popularitzar-lo fent que aviat es fessin paleses les primeres limitacions: la Web presentava un contingut estàtic que només s'actualitzava quan ho decidia l'administrador i no permetia cap mena d'interactivitat amb l'usuari, és a dir, el contingut era únicament de lectura. En aquest punt sorgeix el primer salt evolutiu del mitjà: les webs es tornen dinàmiques –es construeixen dinàmicament a partir de contingut emmagatzemat en bases de dades– i permeten una certa interactivitat amb l'usuari, que pot introduir contingut mitjançant formularis o comentaris.

2.2 La Web social

L'any 2004, i arran d'una conferència feta per *Tim O'Reilly* sobre el que podria ser una nova evolució de la Web, va aparèixer per primer cop el concepte de Web 2.0. Si bé el nou terme podria suggerir un nou salt en les especificacions tècniques del *World Wide Web*, el cert és que més que definir tecnològicament l'entorn, determinava l'ús que se'n feia. Fonamentada en les relacions interpersonals, d'aquí que també sigui coneguda com a Web social, la nova generació de pàgines Web fomentava un entorn col·laboratiu i

¹ CERN: Organització Europea per la Recerca Nuclear

² També anomenada web 1.0, en contraposició a la que seria la seva successora, la web 2.0.

³ HTML (HyperText Markup Language)

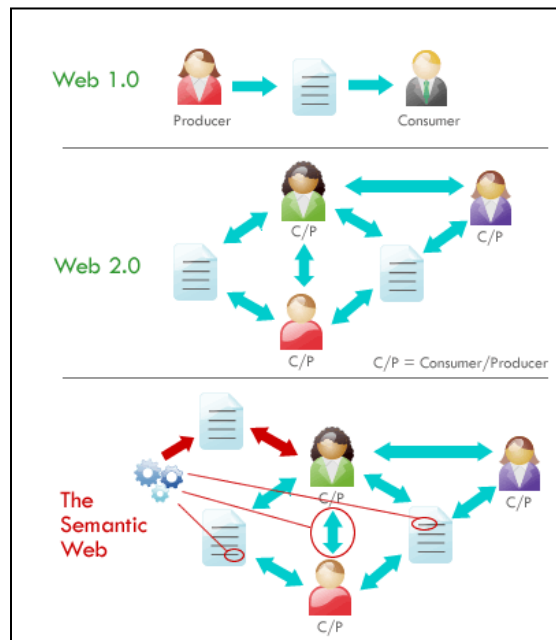
de compartició d'informació que permetia que els propis usuaris fossin alhora creadors de contingut, servint-se de recursos com les xarxes socials, els blocs, els serveis d'allotjament de vídeos o fotografies, etc. Per tant, a diferència de la unidireccionalitat que presentava la Web tradicional, la Web 2.0 fa de l'usuari un element actiu que ja no es limita a consumir continguts de manera passiva, sinó que també en produeix.

2.3. La Web semàntica

Després de més de dues dècades d'evolució de la Xarxa ens trobem a les portes d'un nou salt tecnològic: la Web semàntica. L'any 2001 *Berners-Lee*, el creador del propi *World Wide Web*, juntament amb *James Handler* i *Ora Lassila* van publicar un article a la revista *Scientific American* en què s'utilitzava per primera vegada el terme de Web semàntica. En ell es definia el concepte que hi ha sota aquesta extensió tecnològica de la Web tradicional, en què el significat de la informació traspasa els límits humans per fer-se comprensible per a les màquines.

A diferència de les webs anteriors, els actors generadors de coneixement ja no són únicament els desenvolupadors i els usuaris, sinó que les aplicacions prenen un paper actiu essent capaces de processar el coneixement humà de manera semàntica, convertint-se d'aquesta manera en productores i consumidores de continguts.

Figura 3



Comparació entre la web 1.0, la 2.0 i la semàntica a nivell de producció i consumició de recursos.

3. La Web semàntica

En els propers anys, la Web semàntica es projecta com una tendència a l'alça en l'àmbit del món virtual. Malgrat que a hores d'ara el volum de presència a la Xarxa que té la Web semàntica respecte les seves antecessores pot considerar-se incipient, sembla que en un futur immediat fixarà les línies tecnològiques a seguir, obtenint una *World Wide Web* molt més eficient i amb predominança del treball col·laboratiu entre persones i/o màquines.

En aquest capítol veurem les mancances que presenta la Web tradicional i quins són els aspectes que intenta resoldre la Web semàntica (*apartat 3.1*). A continuació farem una visió general sobre el concepte de Web semàntica i sobre les capes que la formen, tot introduint les tecnologies que té associades (*apartat 3.2*).

3.1. Limitacions que presenta la Web tradicional⁴

Popularment la Web ha estat considerada una gran enciclopèdia universal, on qualsevol coneixement humà o aspecte del món real pot ser-hi contingut. Tant en l'àmbit professional com en el personal utilitzem aquest mitjà per establir relacions, consumir serveis o generar continguts, ampliant amb cada acció aquest gran repositori d'informació. Si bé en una enciclopèdia la informació s'estructura mitjançant una jerarquia lògica en múltiples nivells –aspecte que en facilita la indexació–, els diversos continguts que componen la *World Wide Web* es troben dispersos en el que podríem considerar un núvol de coneixement. Aquest aspecte, juntament amb el gran volum d'informació que hi ha continguda –i que s'amplia a cada segon que passa– fa que la veritable dificultat amb què es pot trobar un usuari sigui la d'obtenir un resultat que satisfaci les seves expectatives.

A nivell de navegació d'usuari, la Web resulta una extensió inabastable. En el cas que un usuari tingui interès en cercar continguts delimitats per diferents paràmetres (per exemple, en el cas d'estudi que ens ocupa, trobar una ruta turística en funció d'un radi geogràfic determinat i amb unes característiques, quant a informació de punts d'interès i a tipus de suport d'aquesta informació, molt concretes) pot resultar una tasca que s'allargui molt en el temps o que resulti del tot infructuosa. I és que la gran quantitat d'informació a processar faria necessària l'automatització, sinó de tota, de com a mínim una bona part d'aquesta cerca per tal d'obtenir resultats satisfactoris en un temps raonable.

La gran dificultat que comporta aquest tipus d'automatització però, és que seria necessari dotar les màquines de capacitat de comprensió. D'aquesta manera podrien filtrar automàticament els continguts de la Web com ho faria un ésser humà: entenent el significat de la informació que s'hi conté i aprovant-la o descartant-la en funció de l'adequació que presenta enfront els paràmetres de cerca definits. Això ens porta a parlar dels motors de cerca.

⁴ Quan parlem de Web tradicional ens referim a tot allò aparegut abans de la Web semàntica (web 1.0 i 2.0), cal recordar que la Web 2.0 o social no representa una nova especificació tecnològica, sinó que descriu la manera com l'usuari utilitza el mitjà.

En certa mesura l'automatització de la cerca ja està implementada a través dels cercadors actuals, els quals actuen com a bases de dades de pàgines indexades a partir d'algunes dades concretes que aquestes contenen (títol, encapçalaments, etc.). Quan un usuari indica un perfil de cerca, el cercador retorna diverses pàgines de resultats amb les concordances entre el perfil especificat i les pàgines indexades però, a causa de la pròpia estructura del cercador, la informació no ha estat processada, sinó que simplement s'ha indexat.

Així, a tall d'exemple, si en un motor de cerca actual establim com a perfil de cerca:

“Ruta per Montserrat passant per Sant Jeroni fetes per la FEEC”

Veurem com obtenim diverses pàgines de resultats sobre la muntanya de *Montserrat*, sobre el cim de *Sant Jeroni*, sobre els grups excursionistes associats a la *FEEC*⁵ que organitzen caminades per *Montserrat*, etc. Però no obtenim allò que en llenguatge humà hem especificat de manera tan precisa: un *track* publicat per la *FEEC* amb la ruta per *Montserrat*. De manera que podem observar que el cercador no ha processat la informació continguda a les pàgines, sinó que s'ha limitat a servir-nos les pàgines per tal que en fem un processament humà.

Aquesta qüestió ens porta al següent aspecte dins del panorama web actual, i no és res més que la codificació que presenta. La major part de les pàgines estan programades orientant-se exclusivament a la presentació d'informació a l'usuari, és a dir, basen el seu codi en el llenguatge *HTML*, de manera que la informació que contenen només és interpretable de forma humana i per part d'un ésser humà.

Mitjançant el codi *HTML* s'estructura el contingut que es visualitzarà amb el navegador, però no s'identifica de cap forma les dades o la tipologia d'informació que conté un determinat document. Per aquest motiu els motors de cerca o els agents web –que són els seus equivalents en l'àmbit semàntic– no poden “comprendre’n” el significat i manipular les dades en conseqüència. Així que, per tal que es pugui automatitzar, i per tant optimitzar, el procés de cerca a la Web cal que aquesta codifiqui la informació amb un llenguatge específic capaç de conferir un significat a la informació que resulti interpretable per part d'una màquina.

3.2. La Web semàntica i les tecnologies que té associades

Quan *Berners-Lee*, *James Handler* i *Ora Lassila* van definir el concepte de Web semàntica, no ho van fer pensant en crear una especificació tècnica que redissenyés el món web des de zero, sinó que pretenia ser una extensió que superés les limitacions de la Web actual o tradicional. Així, la principal característica a nivell global que aporta la nova especificació és precisament referent a l'àmbit semàntic de la Web i, en conseqüència, a la manera com s'hi estructura i com es manipula la informació continguda, que busca afavorir la cooperació entre persones i ordinadors.

⁵ Federació d'Entitats Excursionistes de Catalunya

La Web semàntica pot ser considerada com una Web de dades, ja que no tant sols aporta estructura visual a la informació, com fa la Web tradicional mitjançant el llenguatge *HTML*, sinó que també li proporciona una estructura conceptual –la qual especifica un conjunt de conceptes i defineix la relacions que s'estableixen entre ells (ontologia)– que és capaç de ser compartida, processada i interpretada per agents de programari.

Amb aquest nou entorn de la Web, resulta imprescindible emprar eines específiques que permetin tant dissenyar l'estructura semàntica de les dades contingudes com manipular pròpiament les dades per tal d'extreure'n coneixement. Per tal que això sigui possible, juntament amb la Web semàntica, ha sorgit una col·lecció de tecnologies semàntiques que es poden agrupar en les següents categories:

1. ***Eines de desenvolupament d'ontologies.*** Són eines que faciliten la creació d'ontologies o la reutilització d'ontologies ja existents (per exemple el programari *Protégé*).
2. ***Llenguatges de representació i consulta*** (llenguatges semàntics). Són llenguatges que d'una banda permeten millorar la definició d'ontologies, i de l'altra permeten realitzar consultes sobre els recursos de la Web (per exemple *XML*, *RDF*, *OWL*, etc.).
3. ***Entorns de treball amb agents.*** Són entorns de desenvolupament que faciliten la implementació d'aplicacions basades en sistemes d'agents (per exemple l'entorn *Jade*).

A nivell de llenguatges semàntics trobem dues eines que conformen la base de la Web semàntica: el llenguatge *XML* (detallat al capítol 4) i el llenguatge *RDF* (detallat al capítol 5). A grans trets, podríem dir que *XML* permet, mitjançant la creació d'etiquetes pròpies, dotar d'estructura un document. L'estructura de per si sola però, no atorga cap mena de significat al document, sinó que qui ho fa és el llenguatge *RDF*, el qual codifica la relació entre un element i el valor que prenen les seves propietats (per exemple si l'element és “*Tim Berners-Lee*” la propietat podria ser “*és autor de*” i podria prendre el valor de “*Weaving the web*”). El mateix creador de la Web, *Berners-Lee* fa el símil entre aquesta codificació i l'estructura gramatical d'una oració: en aquest cas l'element esdevindria el *subjecte*, la propietat seria el *predicat* i el valor que pren la propietat seria l'*objecte*.

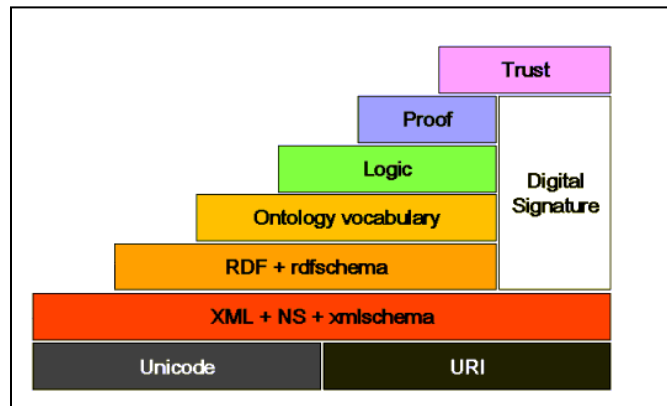
Cada recurs definit per *RDF* –una pàgina web, un document, una adreça de correu...– pot ser codificat mitjançant l'*URI* (*Uniform Resource Identifier*), el qual l'identifica de manera inequívoca. Com que la Web semàntica es basa en la cooperació entre aplicacions, pot donar-se el cas que en diversos emplaçaments hi hagi un recurs amb un significat idèntic però un identificador diferent. Per aquest motiu resulta necessari tenir un darrer element semàntic que permeti als agents de programari deduir la relació entre ambdós recursos: l'ontologia.

Mitjançant la definició d'una taxonomia (classes i subclasses d'objectes i les seves propietats) i d'un conjunt de regles d'inferència s'especifica un esquema conceptual, conegut com ontologia (detallat al

capítol 7), que permet establir relacions entre els diferents conceptes creant un vocabulari formal capaç de ser processat per una màquina.

Vegem a continuació l'esquema per capes proposat per *Berners-Lee* per tal de definir l'estructura de la Web semàntica:

Figura 4



Estructura de la web semàntica.

En l'esquema per capes podem observar que a la base que conforma la Web semàntica es troben els identificadors de recursos (*URI*) i l'estàndard *Unicode* per a la codificació de caràcters. En una segona capa trobem el llenguatge i els mecanismes que permeten donar estructura a la informació (*el significat d'aquesta capa es desenvoluparà de manera més extensa al capítol 4*). Per sobre d'aquesta hi trobem el model de dades que permetrà atorgar significat als diferents termes que componen el coneixement (*consultar el capítol 5*). La quarta capa, correspon a l'esquema conceptual del coneixement, on s'establiran les relacions entre tots els termes formant un vocabulari (*consultar el capítol 7*).

La capa *Logic* correspon a un desenvolupament superior de les ontologies, on a més d'esquemes conceptuals, són necessàries regles d'inferència. Per sobre seu, la capa *Proof* permet executar les regles d'inferència i comprovar-ne la validesa. El segment superior, anomenat *Trust*, correspon al grau de confiança que han de mostrar els agents de programari davant les fonts d'informació semàntica. Finalment, la capa transversal *Digital Signature* permet als agents de programari verificar que la font de la informació sigui realment de confiança.

4. El llenguatge XML

El llenguatge *XML* constitueix el component més elemental de la Web semàntica. Mitjançant un conjunt d'etiquetes que conformen una jerarquia entre objectes, permet descriure i estructurar de manera simple un àmbit concret de coneixement. *XML* és un llenguatge que proveeix una sintaxi superficial per als documents estructurats, però que no especifica el significat de les dades que conté.

En aquest capítol veurem què és el llenguatge *XML* (*apartat 4.1*), com s'estructura un document *XML* (*apartat 4.2*) i quins mecanismes hi ha per determinar-ne la validesa (*apartat 4.3*). Finalment introduïrem el concepte d'espai de noms (*apartat 4.4*).

4.1. El concepte d' XML

El llenguatge *XML* (*eXtensible Markup Language*) és un llenguatge d'etiquetatge derivat del *SGML*⁶ que permet incloure marques en un document amb la finalitat de definir la seva estructura. Com que es tracta d'un llenguatge extensible o ampliable, pot considerar-se un metallenguatge, ja que permet crear llenguatges d'etiquetatge específics per a un domini d'aplicació concret.

Desenvolupat per l'*W3C*⁷, l'*XML* es proposa com a format estàndard per a representar qualsevol tipus d'informació, centrant-se més en els aspectes semàntics que en la presentació. El seu principal avantatge és que facilita la col·laboració, i és que si un conjunt d'entitats arriben a un acord sobre un llenguatge d'etiquetatge basat en *XML* i orientat al seu àmbit de treball, podran intercanviar fàcilment informació.

4.2. Estructura d'un document XML

La principal característica d'aquest llenguatge de marcatge és que a diferència del llenguatge *HTML*, el qual s'especifica mitjançant un conjunt fix d'etiquetes (`<h1>`, `<p>`, `<title>`...), l'*XML* permet a l'usuari definir les seves pròpies marques, adequant-les –en termes semàntics– al tipus d'informació que conté el document (per exemple, en el cas d'estudi que ens ocupa, podríem emprar etiquetes com: `<ruta>`, `<refugi>`,...).

Un document *XML* té dos components bàsics: els elements i els atributs. Mentre que els primers esdevenen els diferents components lògics del document (*ruta*, *refugi*...), i per tant l'estructuren, els segons són els que defineixen informació addicional sobre aquests (*distància*, *dificultat*...*altitud*, *nom*...).

⁶ Standard Generalized Markup Language

⁷ World Wide Web Consortium

Les etiquetes es distribueixen en el document de manera jeràrquica seguint un arbre que parteix d'un element arrel únic –component semàntic bàsic– i s'estén amb diferents nivells d'etiquetes. Cada etiqueta s'especifica entre claus angulars (<refugi>) i la seva especificació es tanca mitjançant claus amb una barra obliqua (</refugi>). Per cada element poden definir-se diversos atributs, cadascun dels quals representa una parella de *nom* i *valor*, que s'especifiquen a banda i banda del signe “igual” amb el valor de l'atribut entre cometes dobles o simples (seguint amb l'exemple: <refugi nom="Colomers" altitud="2100">....</refugi>).

Partint d'aquestes premisses, com a exemple de document *XML* podríem tenir el que es mostra en el següent quadre. Cal notar que la primera línia correspon a la declaració del document i especifica que aquest es tracta d'un *XML*.

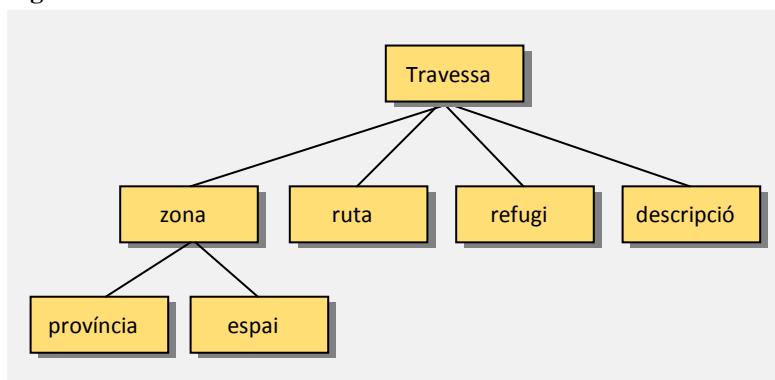
Figura 5

```
<?xml version="1.0" encoding="UTF-8" ?>
<Travessa>
  <zona>
    <província codi="03"> Lleida </província>
    <espai tipus="PN"> Parc Natural </espai>
  </zona>
  <ruta distància="55" dificultat="5"> Carros de Foc </ruta>
  <refugi nom="Colomers" altitud="2100"> Refugi de Colomers </refugi>
  <descripció>
    Aquesta travessa uneix els nou refugis guardats que hi ha
    dins el parc Nacional d'Aigües Tortes i Estany de Sant
    Maurici.
  </descripció>
</Travessa>
```

Exemple de document XML.

Com que la finalitat bàsica del llenguatge és estructurar el document, podríem determinar la seva estructura conceptual, que seria la que es mostra a continuació, on l'element arrel seria *Travessa*, hi hauria una branca que seria *zona* i els demés elements correspondrien a les fulles de l'arbre jeràrquic:

Figura 6



Estructura conceptual del document XML de la figura anterior.

4.3. Validesa d'un document XML

Per tal de complir totes les definicions de format d'un document *XML* s'han de tenir en compte certs aspectes que es recullen en l'especificació⁸ del propi llenguatge, publicada pel *W3C*. El compliment d'aquestes especificacions però, només determina que el document està ben format, és a dir, que empra una sintaxi correcta. Per tal d'assegurar-ne la validesa i, per tant, verificar la relació entre els diferents elements que el formen, cal que alhora compleixi un conjunt de regles i restriccions especificades en una definició externa.

Aquesta definició descriu el format dels diversos elements que conformen el document, determinant-ne tant la jerarquia com la sintaxi. Els models de validació es poden definir de dues maneres: mitjançant una *DTD* (*Document Type Definition*) o bé fent servir un *XML Schema*.

4.3.1. DTD

La *DTD* (*Document Type Definition*) és una descripció d'un document de tipus *SGML* o *XML* que especifica les restriccions que aquest pot tenir quant a estructura i sintaxi. Com que la definició pot fer-se de manera externa, i enllaçar-se posteriorment des del *DOCTYPE* (*declaració de tipus de document*) feta al pròleg de l'*XML*, pot ser compartida per múltiples documents, afavorint així la cooperació.

El processador del document *XML*, també conegut com a *parser*, empra la informació continguda en la definició per tal de validar la gramàtica especificada a la *DTD*. Aquest mecanisme de validació determina, a nivell d'elements, quines etiquetes són reconegudes i quin contingut poden tenir. A més, estructuralment, indica l'ordre que seguiran aquestes etiquetes dins del document i el seu grau d'aniuament respecte la resta. Vegem a continuació l'aspecte que podria tenir un *DTD* que descriu l'estructura del document *XML* que l'acompanya:

Figura 7

```
<?xml version="1.0"?>
<!DOCTYPE Travessa SYSTEM
    "http://.../Travessa.dtd">
<Travessa>
  <ruta> Carros de Foc </ruta>
  <refugi> Colomers </refugi>
</Travessa>
```

Exemple de document XML que fa servir un DTD.

Figura 8

```
<!ELEMENT Travessa (ruta, refugi)>
<!ELEMENT ruta (#PCDATA)>
<!ELEMENT refugi (#PCDATA)>
```

Exemple de document DTD (*Travessa.dtd*). La primera línia especifica que l'element *Travessa* té dos fills, la resta de línies indiquen que les etiquetes *ruta* i *refugi* contenen dades que no han de ser interpretades pel *parser*.

Tot i que la *DTD* continua resultant un model vàlid, les seves limitacions i falta de flexibilitat han impulsat l'ús dels *XML Schema*, que és el mecanisme que recomana el *W3C*.

⁸ <http://www.w3.org/TR/2006/REC-xml11-20060816/>

4.3.2 XML Schema

*XML Schema*⁹ és un llenguatge d'esquema extensible que estableix el conjunt de regles que ha de complir la gramàtica d'un document *XML*. A diferència del la *DTD*, aquest document d'esquema fa servir *XML* per a codificar la gramàtica, aspecte que facilita la seva validació per part dels processadors de documents *XML*. A més, el fet que es defineixi amb un llenguatge extensible com és l'*XML*, fa que el mateix esquema pugui reutilitzar-se en altres esquemes, així com referenciar múltiples esquemes en un mateix document. A nivell de components, i com a conseqüència d'aquesta mateixa extensibilitat, l'*XML Schema* també permet la creació de tipus de dades derivades dels tipus estàndards, els quals al seu temps formen part d'un conjunt més ric i variat que no pas el que constituïa la *DTD*.

Vegem a continuació l'exemple de document *XML* citat a l'apartat anterior, però emprant un *XML Schema* per a definir la seva gramàtica:

Figura 9

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.senderisme.org/muntanya"
xmlns="http://www.senderisme.org"
elementFormDefault="qualified">

  <xs:element name="Travessa">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ruta" type="xs:string"/>
        <xs:element name="refugi" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Exemple de document *XML Schema*. L'element *travessa* és de tipus complex, ja que conté els subelements *ruta* i *refugi*. Aquests darrers són de tipus cadèna de caràcters.

4.4. L'espai de noms

Els documents *XML* permeten en la seva declaració l'ús de qualsevol etiqueta per tal de referenciar elements i atributs, cosa que fomenta la creació de vocabularis rics i precisos. Si bé la idea fonamental d'aquest llenguatge és que un mateix document pugui ser compartit per diversos mòduls de programari, potenciant la reutilització de vocabularis ja definits, pot passar que la manca de restriccions en termes de declaracions provoqui col·lisions entre les marques del vocabulari original —que es reutilitza— i les del vocabulari particular que hem creat.

⁹ Especificacions del llenguatge W3C (primera i segona part):
<http://www.w3.org/TR/xmlschema11-2/>
<http://www.w3.org/TR/xmlschema11-1/>

Per exemple:

Si treballéssim amb un vocabulari de muntanya, una etiqueta `<track>` podria significar una ruta per a GPS, mentre que si parléssim d'un vocabulari musical, la mateixa etiqueta significaria una pista d'àudio.

Per tal d'evitar la poca precisió en la identificació d'elements que es dona en l'*XML*, la pròpia especificació del llenguatge proposa un mecanisme que permet identificar de manera inequívoca cada element: els espais de noms o *namespaces*¹⁰.

Els espais de noms permeten superar l'ambigüitat que poden produir les etiquetes que formen els vocabularis *XML* mitjançant l'ús d'una nomenclatura específica que identifica que un determinat element pertany a un grup conceptual definit per l'espai de noms. Aquest es declara a l'etiqueta arrel i s'utilitza com a prefix cada vegada que es creï una etiqueta associada a un determinat espai de noms –ja que en un mateix document *XML* poden coexistir diversos espais de noms–.

Figura 10

```
<M:Travessa xmlns:M="http://www.senderisme.org/muntanya">
  <M:ruta> Carros de Foc </M:ruta>
  <M:refugi> Colomers </M:refugi>
</M:Travessa>
```

Exemple de declaració de *namespace*. A l'etiqueta arrel es declara l'espai de noms que s'utilitzarà mitjançant l'expressió `xmlns:nom` i s'identifica el recurs amb l'URI.

¹⁰ Especificació W3C: <http://www.w3.org/TR/REC-xml-names/>

5. El model RDF

El model *RDF*¹¹ (*Resource Description Framework*) és un estàndard definit pel W3C per a l'intercanvi de dades a la Web que està dissenyat per a ser llegit i interpretat per ordinadors. Tot i que d'alguna manera es pot percebre com a llenguatge, el cert és que es tracta d'una forma de representar un determinat llenguatge o els conceptes semàntics que aquest defineix. Mitjançant objectes i la relació que s'estableix entre ells, aquest model de dades proveeix una semàntica simple per tal de conferir significat a l'estructura conceptual creada a través del llenguatge *XML* (*aspecte tractat a l'apartat 4*).

En aquest capítol veurem què és aquest model de dades i quina estructura pren la informació que conté (*apartat 5.1*), a continuació analitzarem el mecanisme que es fa servir per identificar inequívocament un recurs dins un determinat sistema (*apartat 5.2*) i descriurem els principals components del vocabulari que ofereix (*apartat 5.3*). Finalment analitzarem quins elements incorpora l'extensió *RDF Schema* (*apartat 5.4*).

5.1. El concepte d'RDF

RDF va ser dissenyat per processar metadades i afavorir-ne l'intercanvi. Com a llenguatge de modelat de coneixement permet definir les relacions semàntiques que s'estableixen entre diferents recursos – identificats per *URIs*– associant-les a un seguit de propietats i valors concrets.

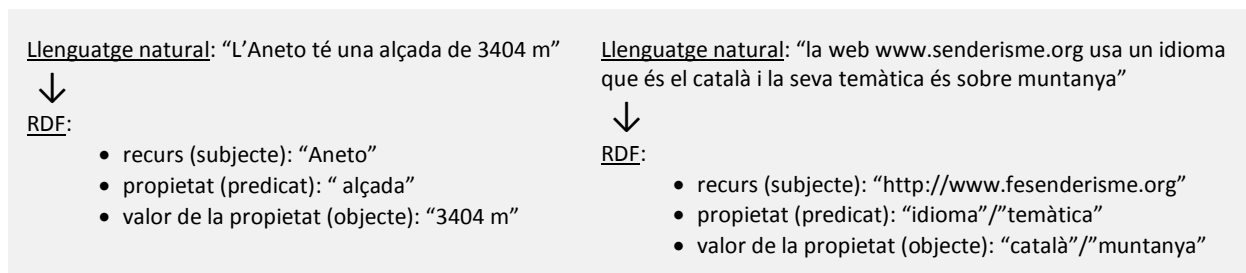
El llenguatge que fa servir és l'*XML*, de manera que aquest sol atribuir-se al model anomenant-se *RDF/XML*. El principal avantatge d'emprar aquest llenguatge de configuració del model *RDF* és que la informació continguda és fàcilment compartible entre diferents aplicacions –inclús si aquestes pertanyen a sistemes operatius diferents o fan servir un llenguatge, a nivell d'aplicació, diferent–.

Pel que fa referència a l'estructura del vocabulari, podem partir del símil que feia *Berners-Lee* amb l'oració gramatical (*veure apartat 3.2*), *RDF* es basa en la traducció dels recursos de la Web a expressions codificades en triplets –que s'anomenen sentències– que segueixen l'estructura *Subjecte-Predicat-Objecte*. D'aquesta manera podríem dir que la posició de *Subjecte* està ocupada pel recurs que definim, la del *Predicat* per la propietat d'aquest *Subjecte* i, finalment, l'*Objecte* correspon al valor concret que pren la propietat o a un altre recurs.

¹¹ Especificació W3C del model RDF: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

Per entendre el concepte en què es basa podríem fer la següents traduccions del llenguatge natural:

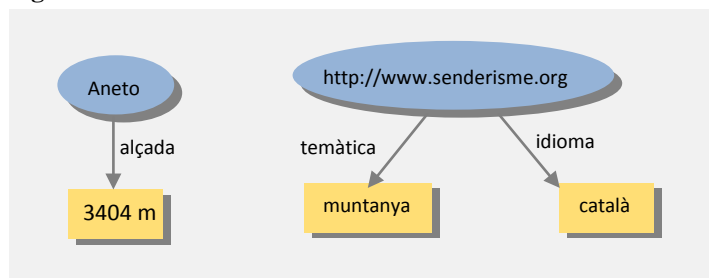
Figura 11



Exemple de traducció del llenguatge natural a triplet RDF.

Per tal de representar les sentències, *RDF* utilitza un graf dirigit on els nodes es corresponen amb els recursos o valors de propietat i les fletxes amb les propietats. D'aquesta manera podríem veure l'exemple anterior representat com:

Figura 12



Graf dirigit que representa gràficament els triplets de la figura anterior.

5.2. La identificació de recursos: URI

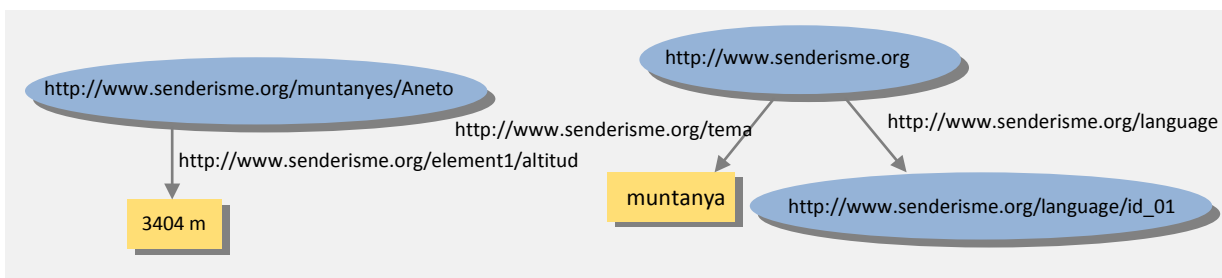
Les diverses sentències que es defineixen mitjançant els triplets *RDF* hauran de ser interpretades per un ordinador, de manera que el llenguatge humà emprat per definir-les pot resultar excessivament vague. Com que la finalitat dels llenguatges semàntics és precisament la de permetre processar la informació que hi ha distribuïda en diversos sistemes, podria passar que dos autors diferents anomenessin de la mateixa manera un recurs o propietat i que aquest recurs correspongués a un element diferent en cadascun dels casos. Si aquests autors decidissin treballar entre ells de manera cooperativa, podria haver-hi confusions entre objectes. Per evitar aquesta situació existeix un mecanisme que els permet identificar de manera inequívoca: l'*URI*.

L'*URI* (*Uniform Resource Identifier*) és una cadena de caràcters que identifica els objectes amb independència d'on es trobin –a la Xarxa o fora d'ella¹²– i n'especifica la localització. Un *URI* pot ser

¹² L' *URI* identifica un recurs amb independència de la seva localització (accessible o no des de la Xarxa), en canvi les *URLs* (*Uniform Resource Locator*) –com a subconjunt d'*URIs*– són identificadors de recursos que es troben únicament en línia.

aplicat a qualsevol dels elements que formen els triplets *RDF* (*Subjecte-Predicat-Objecte*). Vegem a continuació l'exemple anterior substituint alguns dels literals (cadena de caràcters) per *URIs*.

Figura 13



Graf dirigit que substitueix alguns literals per URIs.

5.3. El vocabulari principal d'*RDF/XML*

El model de dades *RDF* proporciona una sintaxi determinada que permet mapejar el contingut i significat del graf dirigit sobre un document amb llenguatge *XML*. L'objectiu d'aquesta traducció és el de permetre a les màquines interpretar el significat d'un model que representa una realitat humana. I és que el *RDF/XML* pot ser processat per un ordinador i mitjançant els *URIs* enllaçar dades contingudes en objectes dispersos per la Web o per d'altres sistemes. A continuació podem veure un exemple de document *RDF*, que mapeja el contingut representat a la taula a un llenguatge interpretable per part d'un ordinador:

Figura 14

Muntanya	Altitud	Zona	Espai
Aneto	3404	ARAG	PN Posets-Maladeta
Pica d'Estats	3143	CAT	PN Alt Pirineu

Taula de cims.

Figura 15

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:M="http://www.senderisme.org/muntanyes#">

  <rdf:Description rdf:about="http://www.senderisme.org/muntanyes/Aneto#">
    <M:altitud>3404 m</M:altitud>
    <M:zona>ARAG</M:zona>
    <M:espai>PN Posets-Maladeta</M:espai>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.senderisme.org/muntanyes/Pica_Estats#">
    <M:altitud>3143 m</M:altitud>
    <M:zona>CAT</M:zona>
    <M:espai>PN Alt Pirineu</M:espai>
  </rdf:Description>
</rdf:RDF>
```

Exemple de document *RDF*. La primera i segona línia són la declaració de document *XML* i de l'element arrel de *RDF*, respectivament. S'utilitzen dos espais de noms: un per a elements de tipus *RDF* (`xmlns:rdf`) i l'altre per a elements propis del context de muntanya (`xmlns:M`). L'etiqueta `rdf:Description` conté la descripció del recurs identificat per l'atribut `rdf:about`. La resta d'elements (`M:element`) són propietats del recurs.

Per tal de declarar els triplets es fa servir l'etiqueta `<rdf:Description>`, a més *RDF/XML* defineix un seguit d'atributs *RDF* que permeten identificar els diferents components de les sentències:

- **Subjecte.** El recurs que correspon al *subjecte* s'identifica mitjançant l'atribut `rdf:about`. Per tal de localitzar-lo dins del sistema s'acompanya de l'*URI* (en l'exemple anterior `rdf:about="http://www.senderisme.org/muntanyes/Aneto"`).
- **Predicat.** La propietat expressada pel *predicat* pot entendre's com un atribut o com un recurs. Si es fa servir com a atribut, l'exemple anterior –recordem que descrivia un determinat recurs amb la inclusió de diversos elements– prendria ara la forma:

Figura 16

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:M="http://www.senderisme.org/muntanyes#">

  <rdf:Description
    rdf:about="http://www.senderisme.org/muntanyes/Aneto#">
    M:altitud="3404 m" M:zona="ARAG" M:espai="PN Posets-Maladeta"/>
  </rdf:Description>
</rdf:RDF>
```

Predicat entès com a atribut.

En canvi, si es fa servir un recurs com a propietat, el mateix exemple s'expressaria:

Figura 17

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:M="http://www.senderisme.org/muntanyes#">

  <rdf:Description
    rdf:about="http://www.senderisme.org/muntanyes/Aneto#">
    ...
    <M:zona rdf:resource="http://www.senderisme.org/muntanyes/provincia"/>
    ...
  </rdf:Description>
</rdf:RDF>
```

Predicat entès com a recurs.

- **Objecte.** La manera com es representa un objecte depèn, en bona mesura, de com s'hagi definit el *predicat*. Així, si s'ha emprat un *predicat* en forma d'atribut el valor s'indica de manera literal (tal com es mostra a la *figura 16*), en canvi, si s'ha fet servir un recurs, s'especifica l'*URI* per tal de localitzar-lo en el sistema (tal com es mostra a la *figura 17*).

5.4. RDF Schema

L'*RDF* és un model que permet estructurar els recursos i les seves propietats de manera molt efectiva, però no permet agrupar els recursos en objectes majors –com per exemple en *classes*– per tal d'aplicar-hi una propietat de conjunt. Aquesta limitació, juntament amb la manca de mecanismes per establir una jerarquia clara entre elements, fa que sigui necessari trobar alguna eina que permeti donar estructura al vocabulari definit mitjançant l'*RDF*. Per aquest motiu el *W3C* especifica l'*RDF Schema*¹³.

L'*RDF Schema* és una extensió semàntica d'*RDF* que proporciona els elements necessaris per descriure vocabularis *RDF* –també anomenats ontologies–. La seva finalitat bàsica és descriure grups d'objectes relacionats entre sí, així com les relacions que s'hi estableixen. Basant-se en un model similar al que presenta l'*Orientació a Objectes*, *RDF Schema* ofereix un entorn de descripció de classes i relacions, en el qual poden definir-se tant *instàncies de classes* com *subclasses*.

Figura 18

```
<?xml version="1.0"?>

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

  <rdf:Description rdf:ID="muntanya">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>

  <rdf:Description rdf:ID="cim">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#muntanya"/>
  </rdf:Description>

</rdf:RDF>
```

Exemple de RDF Schema. Muntanya és un element del tipus *classe* i cim és una *subclasse* de muntanya.

Tot i la notable millora respecte *RDF*, el vocabulari que presenta *RDF Schema* pot resultar limitat per acabar de conferir significat a les dades que conté l'ontologia, de manera que des del *W3C* s'ha proposat un nou llenguatge –l'*OWL (Ontology Web Language)*– per tal d'ampliar la sintaxi semàntica existent.

¹³ Especificació W3C de RDF Schema: <http://www.w3.org/TR/rdf-schema/>

6. El llenguatge OWL

La Web semàntica es construeix sobre dos grans pilars. D'una banda tenim la capacitat que presenta el llenguatge *XML* per a definir esquemes conceptuals d'elements mitjançant un sistema d'etiquetes personalitzables. De l'altra, la flexibilitat que ofereix el model de dades *RDF* –i per extensió *RDF Schema*– per tal de representar les relacions que existeixen entre aquests elements. Ara bé, per tal que una màquina pugui processar el coneixement humà de manera com ho faria un ésser humà, cal que les relacions que connecten les dades que formen aquest coneixement estiguin descrites de forma minuciosa, i en certs aspectes *RDF Schema* no acaba d'assolir aquest fet.

En els següents apartats veurem com el llenguatge *OWL* ofereix la possibilitat de representar el coneixement d'una manera encara més descriptiva i lògica superant les limitacions dels seus predecessors però emprant-los com a base (*apartat 6.1*). A continuació veurem com s'estructura un document *OWL* (*apartat 6.2*) i finalment esmentarem la divisió en subllenguatges que presenta *OWL* amb els diferents nivells de complexitat, veient les característiques particulars de cadascun (*apartat 6.3*).

6.1. El concepte d'OWL

El W3C proposa el llenguatge *OWL*¹⁴ (*Ontology Web Language*) per tal d'ampliar la capacitat semàntica de models anteriors, com serien *RDF* o *RDF Schema*, amb la finalitat de descriure formalment el significat de l'ontologia d'una manera més precisa. Els llenguatges que van precedir *OWL* no van definir-se pensant en ser utilitzats en l'àmbit de la Web semàntica de manera que certs aspectes importants d'aquest entorn (com els predicats binaris o les jerarquies de propietats) queden molt limitats.

OWL pretén ampliar les capacitats del llenguatge *RDF* –que continua funcionant com a base– permetent realitzar una major quantitat d'accions sobre les ontologies. Per exemple, aquestes hauran de poder distribuir-se per múltiples sistemes i ser escalables. Com a punt addicional, el nou llenguatge proporciona més vocabulari sobre el ja existent per definir de forma més precisa les *classes*¹⁵ i les seves relacions. Per exemple:

- Afegeix el concepte de cardinalitat
- Incorpora el concepte d'igualtat
- Inclou major nombre de propietats i característiques de propietats
- Permet una descripció més precisa de les relacions que s'estableixen entre classes
- Afegeix l'opció de numerar classes

¹⁴ Especificació W3C d'OWL: <http://www.w3.org/TR/owl-features/>

¹⁵ Veure apartat 7.3.

6.2 Estructura d'un document OWL

El document *OWL* és la traducció de l'ontologia –que en el sistema conceptual humà es representa mitjançant un gràfic de jerarquies– a un llenguatge comprensible i semànticament interpretable per part d'un agent de programari. Com que *OWL* es desenvolupa sobre els llenguatges anteriors, no ens estendrem excessivament en aquest punt ja que molta de la terminologia vista en apartats anteriors continua sent vàlida per aquest.

Com que *OWL* utilitza el constructes ja definits per *RDF*, *RDF Schema* i *XML Schema* fa servir els prefixos *rdf*, *rdfs* i *xsd* (*XML Schema Datatype*) per referir-se als *namespaces* definits per cada llenguatge. A continuació podem veure un exemple de codificació d'una ontologia (correspon a un extracte de l'ontologia desenvolupada en l'apartat 7.6.2):

Figura 19

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
  <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
  ...
]>

<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2011/3/OntologiaMuntanya.owl#"
  xml:base="http://www.semanticweb.org/ontologies/2011/3/OntologiaMuntanya.owl"
  ...
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <owl:Ontology rdf:about=""/>

  <owl:Class rdf:ID="Allotjament">
    <rdfs:label rdf:datatype="&xsd:string">Allotjament</rdfs:label>
  </owl:Class>

  <owl:DatatypeProperty rdf:ID="Altitud">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Cim"/>
          <owl:Class rdf:about="#Poblacio"/>
          <owl:Class rdf:about="#Refugi"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:label rdf:datatype="&xsd:string">Altitud</rdfs:label>
    <rdfs:range rdf:resource="&xsd:int"/>
  </owl:DatatypeProperty>

  (segueix...)
```

Exemple de codificació d'una ontologia mitjançant OWL.

- ① Conjunts de definicions d'entitats fetes en un DOCTYPE.
- ② Espais de noms corresponents a owl, rdfs, rdf i xsd.
- ③ Exemple de declaració d'una classe (classe *Allotjament*)
- ④ Exemple de declaració d'una propietat de dades (atribut *altitud*)

```
(...)
```

```
<owl:ObjectProperty rdf:ID="Pertany_a">
  <rdfs:domain rdf:resource="#Cim"/>
  <rdfs:label rdf:datatype="&xsd:string">Pertany_a</rdfs:label>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Comarca"/>
        <owl:Class rdf:about="#Regio"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
```

```
<owl:Thing rdf:ID="Aiguafreda">
  <rdfs:type rdf:resource="#Poblacio"/>
  <rdfs:label rdf:datatype="&xsd:string">Aiguafreda</rdfs:label>
</owl:Thing>
```

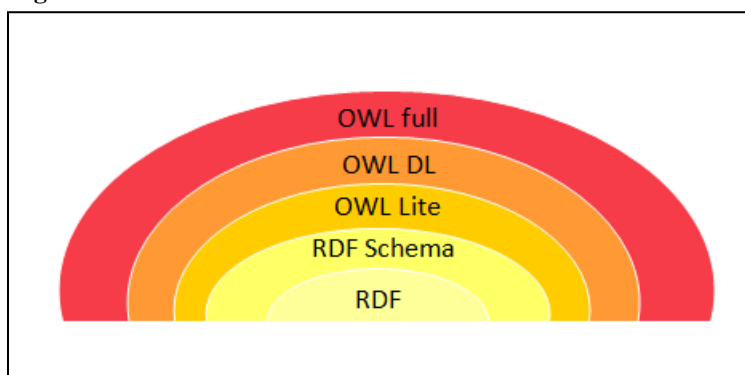
```
...
</rdf:RDF>
```

- 5 Exemple de declaració d'una propietat d'un objecte (relació *Pertany_a*)
- 6 Exemple de declaració d'una instància de tipus població (instància *Aiguafreda*)

6.3 Els subllenguatges d'OWL

El llenguatge *OWL* es divideix en tres subllenguatges que, de manera incremental, van ampliant les capacitats que presenta el seu predecessor. Per cada ontologia caldrà determinar quin és el subllenguatge que s'adapta millor a les necessitats de descripció que presenta, així quan es vulgui representar un model conceptual amb major significat, és a dir, emprant constructes més expressius, el desenvolupador s'haurà de decantar per un llenguatge *OWL* més complet. En la següent figura podem observar el nivell de complexitat de cada llenguatge:

Figura 20



Nivell de complexitat dels llenguatges de representació de coneixement.

Com es pot observar en la figura anterior, *OWL* representa una extensió dels llenguatges de modelat de dades previs. Els subllenguatges que el componen, al seu torn, es classifiquen en funció de la seva capacitat semàntica com segueix a continuació:

- **OWL Lite.** És la versió més simplificada d'*OWL* i dóna suport als desenvolupadors que necessiten jerarquies i restriccions simples.
- **OWL DL.** És la versió intermèdia, inclou tots els constructes del llenguatge però aquests han de ser usats amb certes restriccions. Segueix mantenint la computacionabilitat i la resolubilitat en el càlcul.
- **OWL Full.** És la versió més completa d'*OWL* i està indicada per a desenvolupadors que busquen la màxima expressivitat i llibertat sintàctica.

6.3.1 Els components del subllenguatge OWL Lite

A la taula que es mostra a continuació podem veure els diferents components o sintaxi del llenguatge més bàsic d'*OWL* i el seu significat:

Figura 21

Característiques d'RDF Schema	
<i>Class (Thing, Nothing)</i>	Defineix un conjunt d'elements que tenen propietats compartides.
<i>rdfs:subClassOf</i>	Defineix que una classe és subclasse d'una altra.
<i>rdf:Property</i>	Defineix una relació o un atribut.
<i>rdfs:subPropertyOf</i>	Permet definir subpropietats.
<i>rdfs:domain</i>	Determina sobre quins elements s'aplicarà una propietat.
<i>rdfs:range</i>	Determina quins elements poden ser valor d'una propietat.
<i>Individual</i>	Instància d'una classe.
Igualtat i desigualtat	
<i>equivalentClass</i>	Defineix que dues classes són equivalents o sinònimes.
<i>equivalentProperty</i>	Defineix que dues propietats són equivalents.
<i>sameAs</i>	Serveix per crear noms diferents que es refereixen a un mateix individu.
<i>differentFrom</i>	Defineix que un individu és diferent d'altres individus.
<i>AllDifferent</i>	Defineix que tots els individus són diferents entre ells.
Característiques de propietats	
<i>ObjectProperty</i>	Defineix la relació entre dos individus.
<i>DatatypeProperty</i>	Defineix un atribut d'un individu.
<i>inverseOf</i>	Defineix que una propietat és la inversa d'una altra.
<i>TransitiveProperty</i>	Estableix que una propietat és transitiva
<i>SymmetricProperty</i>	Estableix simetria entre propietats.
<i>FunctionalProperty</i>	Estableix que la propietat només accepta un valor únic per cada individu.
<i>InverseFunctionalProperty</i>	Inversa de l'anterior.
Restriccions de propietats	

Restriction onProperty allValuesFrom someValuesFrom	Restricció que s'aplica. Propietat on s'aplica la restricció. Restricció aplicada a tots els elements del rang d'una propietat. Restricció aplicada a alguns elements del rang d'una propietat.
Cardinalitat	
minCardinality maxCardinality cardinality	Cardinalitat mínima d'una propietat (val 0 o 1). Cardinalitat màxima d'una propietat (val 0 o 1). Indica que el valor de cardinalitat màxim i mínim és el mateix.
Altres components	
Anotacions Intersecció de classes Datatypes	

Taula de components del llenguatge OWL Lite.

6.3.2 Els components del subllenguatge OWL DL i OWL Full

Tant el llenguatge *OWL DL* com l'*OWL Full* utilitzen exactament el mateix vocabulari –que és una extensió del que emprava *OWL Lite*–. La diferència entre ambdós subllenguatges rau en el fet que el primer està subjecte a certes restriccions en el seu ús. A continuació podem veure una taula amb els components que incorporen, respecte la versió més simple del llenguatge, els dos subllenguatges de més complexitat:

Figura 22

Components addicionals	
<i>oneOf</i>	Es poden descriure classes mitjançant l'enumeració dels individus que la formen.
<i>hasValue</i>	Defineix que una propietat té un valor concret per als individus d'una classe.
<i>disjointWith</i>	Defineix que dues classes no tenen cap element en comú.
<i>UnionOf</i> <i>complementOf</i> <i>intersectionOf</i>	Permeten fer operacions booleanes sobre classes o propietats.
<i>minCardinality</i> <i>maxCardinality</i> <i>Cardinality</i>	A diferència d'OWL Lite, permet definir la cardinalitat amb nombres majors que 1.

Taula de components addicionals dels llenguatges OWL DL i OWL Full

7. L'ontologia

En el camp de la metafísica, l'ontologia és una estructura conceptual que busca determinar, a nivell filosòfic, el que existeix i el que no mitjançant la descripció d'un seguit d'entitats i de les relacions que s'estableixen entre elles. De manera paral·lela, si apliquem aquest concepte a l'àmbit de la informàtica l'ontologia esdevé un esquema conceptual que descriu de forma exhaustiva i formal els elements que formen part d'un domini concret, permetent d'aquesta manera que la informació continguda en l'estructura pugui ser processada per màquines com ho faria un ésser humà, és a dir, semànticament.

En aquest capítol veurem què és exactament una ontologia, quines aplicacions té en l'àmbit de la informàtica (*apartat 7.1*) i repassarem breument les diferents tipologies d'ontologia que existeixen (*apartat 7.2*). A continuació determinarem quins són els components fonamentals que formen una ontologia (*apartat 7.3*) i els passos que es requereixen per tal de construir-ne una (*apartat 7.4*). Finalment descriurem breument alguna de les ontologies turístiques existents (*apartat 7.5*) i treballarem amb el programari Protégé amb l'objectiu desenvolupar la nostra pròpia ontologia, particularitzant-ne el domini en l'àmbit del turisme o lleure de Muntanya (*apartat 7.5*).

7.1. El concepte d'ontologia

Amb l'arribada de la Web semàntica, ha sorgit la necessitat de crear mecanismes específics per tal de representar el coneixement i facilitar-ne la localització i compartició. Per a que això sigui possible, la informació ha de ser introduïda en una estructura que la descrigui a nivell semàntic i estableixi les diferents relacions entre els conceptes que hi apareixen. L'esquema conceptual que representa l'ontologia serveix per definir de manera formal els termes que descriuen una àrea particular de coneixement: cada terme representa un element dins l'estructura de l'ontologia i, al mateix temps, configura un seguit de relacions establertes amb altres elements que formen part del domini, donant com a resultat un vocabulari específic que descriu un coneixement humà.

Les ontologies s'utilitzen tant per part d'éssers humans com d'aplicacions i bases de dades amb la finalitat de compartir informació sobre àrees de coneixement particulars. El fet que una mateixa estructura hagi de ser interpretada per dos subjectes amb un raonament tan diferent fa que sigui necessari emprar un llenguatge humà prou explícit i lògic com perquè aquest resulti processable per part d'un ordinador.

Aquest tipus de representació semàntica presenta un gran potencial i se'n poden destacar un seguit d'aspectes:

- Permet compartir la interpretació d'informació entre agents evitant l'ambigüitat
- Permet reutilitzar el coneixement

- Separa el coneixement del domini de l'operacional (el coneixement associat a un problema és independent a les tècniques per resoldre'l)
- Permet analitzar la informació que es té sobre un determinat domini (per exemple verificar la validesa de les dades) i explicitar suposicions.

7.2. Tipus d'ontologies

La tipologia dels conceptes que es representen mitjançant l'ontologia determina, en bona mesura, el tipus de vocabulari que s'obté. Per aquest motiu podem distingir quatre tipus fonamentals d'ontologies:

- **Ontologies d'un sol domini.** Són aquelles estructures mitjançant les quals es representa el coneixement específic d'una àrea concreta –ja sigui un domini o un subdomini– (per exemple *Medicina*, *Oncologia*, etc.).
- **Ontologies genèriques.** Són estructures que representen conceptes generals i fundacionals del coneixement (estructures tot/part, processos, tipus d'objectes, etc.).
- **Ontologies representacionals o metaontologies.** Especifiquen d'una manera més clara les conceptualitzacions del coneixement definint les metadates.
- **Ontologies creades per a una tasca específica o “task ontologies”.** Són estructures creades per a una activitat molt concreta (per exemple fer el diagnòstic d'una malaltia).

7.3. Components l'ontologia

La web semàntica necessita recolzar-se en ontologies amb un alt grau d'expressivitat que descriguin les diferents àrees de coneixement mitjançant una estructura molt detallada. Aquest aspecte fa que certs conceptes del domini hagin de descriure's amb minuciositat, esdevenint elements fonamentals de l'ontologia. Aquests elements són:

- **Les classes.** Són els elements generals del domini (contenidors d'individus). Serveixen per representar de manera formal els conceptes més genèrics.
- **Els individus.** Són objectes dins d'un domini determinat (les instàncies concretes d'una classe).
- **Les propietats.** Són les relacions que poden existir entre les diverses classes i el conjunt de característiques i atributs d'aquestes. De manera addicional poden presentar restriccions quant a cardinalitat, tipologia, etc.

7.4. Cicle de desenvolupament d'una ontologia

Per tal de desenvolupar una ontologia, no existeix una metodologia concreta, sinó que hi ha diferents tècniques que s'adeqüen de manera específica segons el domini tractat. Cal tenir en compte que la construcció d'una ontologia no té per què fer-se de manera lineal, sinó que sol desenvolupar-se de manera iterativa tractant quatre aspectes fonamentals: la definició de les classes que componen un determinat domini, l'organització jeràrquica d'aquestes classes, la definició de les propietats de cada classe –indicant, en cas que sigui necessari, les restriccions que presenten– i l'assignació de valors concrets a les propietats per tal de generar instàncies.

A continuació podem veure un *cicle de desenvolupament*¹⁶ d'una ontologia que inclou, a part dels quatre aspectes esmentats, tres passos previs que en garanteixen un desenvolupament òptim:

- **Pas 1: Determinar el domini i la cobertura.** Caldrà determinar quin domini descriurà l'ontologia i quina finalitat tindrà. Pot resultar útil preguntar-se qui haurà d'emprar o mantenir l'estructura resultant o a quines preguntes ha de poder donar resposta.
- **Pas 2: Considerar la reutilització d'ontologies existents.** L'objectiu fonamental de les ontologies és esdevenir un mecanisme de compartició de coneixement, de manera que poden existir ontologies implementades que s'ajustin suficientment a les nostres necessitats. D'aquesta manera pot resultar interessant treballar sobre un esquema ja creat i ampliar-lo en funció dels nostres requeriments particulars.
- **Pas 3: Determinar els termes importants que haurà de contenir.** Cal enumerar tots aquells termes que permeten descriure el nostre domini, així com crear frases que permetin fer-nos preguntes per explicar informació sobre ell.
- **Pas 4: Definir les classes i la seva jerarquia.**
- **Pas 5: Definir les propietats de cada classe.** Cal tenir en compte que existeix el mecanisme de l'herència, de manera que les propietats definides en classes generals es transmetran de manera automàtica a les classes particulars.
- **Pas 6: Determinar les propietats.** Especificar, en cas que correspongui: la cardinalitat, tipologia, obligatorietat, els valors que pot tenir per defecte, etc.
- **Pas 7: Generar instàncies a partir de l'assignació de valors concrets.**

¹⁶ Cicle de desenvolupament presentat per Javier Béjar (LSI-FIB-UPC)

7.5. Ontologies turístiques geogràfiques

En la metodologia de desenvolupament d'ontologies es contempla el fet d'utilitzar-ne de ja existents per a crear-ne de pròpies. Partint de la base que l'aplicació final a desenvolupar tractarà sobre el turisme (de manera particular sobre el turisme de muntanya), s'ha realitzat una cerca d'ontologies sobre l'àmbit turístic en general per mirar de trobar taxonomies en què basar-se.

1) Ontologia Travel.owl

Es tracta d'una ontologia implementada amb Protégé que estructura i relaciona conceptes del context turístic: tipus d'allotjament, platges, activitats de lleure...

Web: <http://protege.cim3.net/file/pub/ontologies/travel/travel.owl>

2) Web de la ciutat de Saragossa

Es tracta d'un projecte amb el nom “*Un visitante, una ruta*”, finançat per l'Ajuntament de Saragossa, que té com a objectiu esdevenir una aplicació intel·ligent que calculi les rutes turístiques per la ciutat de Saragossa en funció del perfil i el context de l'usuari.

Web: <http://www.zaragoza.es/ciudad/turismo/es/visitar/ontologia.htm>

Ontologia: <http://www.zaragoza.es/cont/paginas/turismo/es/visitar/turismo.owl>

3) Illes Balears Innovació Tecnològica (IBIT)

És un projecte que té com a objectiu la transformació de continguts turístics a metadades semàntiques per tal de millorar seva distribució.

Link PDF: http://www.ibit.org/dades/doc/955_es.pdf

Article: http://www.iadis.net/dl/final_uploads/200508P002.pdf

4) Ontoguate

Ontologia de Turisme a Guatemala desenvolupada per la Universitat de San Carlos (Guatemala). A la pestanya *options* permet veure l'estructura de classes de l'ontologia.

Link: <http://ontoguate.comze.com/index.php>

5) Centre d'Investigació en Computació, Mèxic

Article que explica el disseny d'una aplicació Web híbrida aplicada a serveis turístics de la zona d'Acapulco descrits semànticament.

Link a l'article: <http://www.tonahtiu.com/SIS032.pdf>

6) Projecte GODO (Generación inteligente de Objetivos para el Descubrimiento de servicios web semánticos)

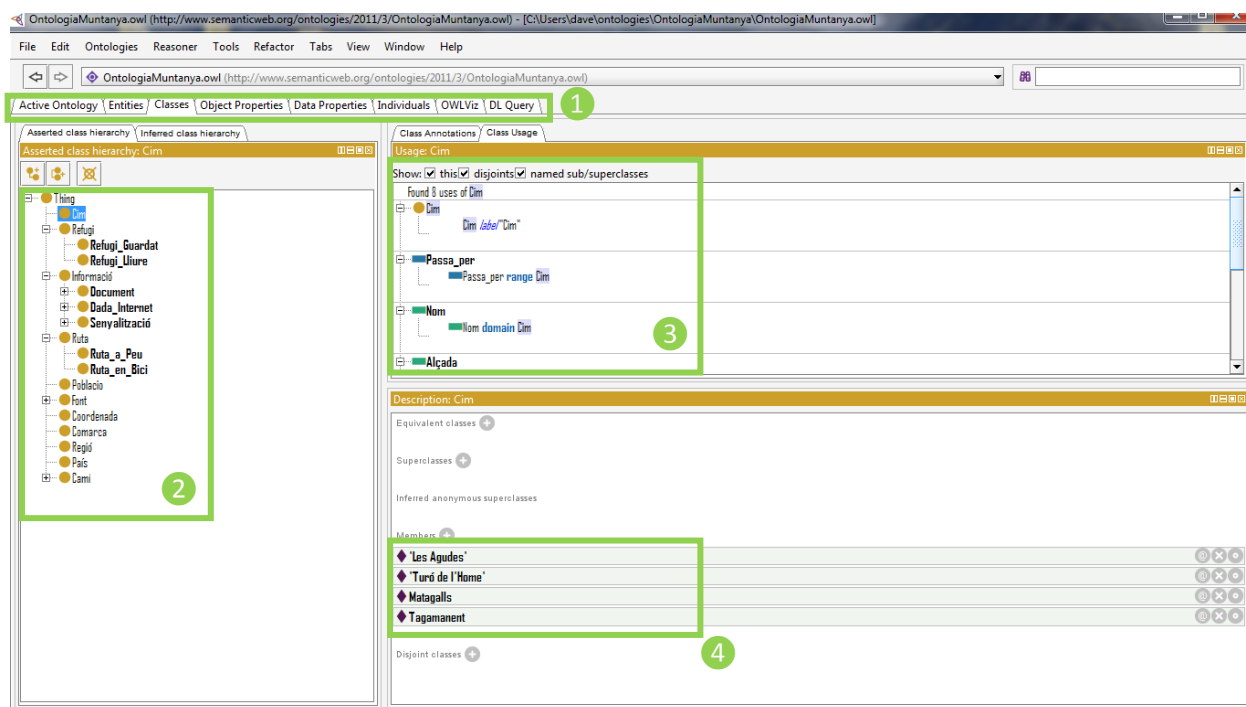
Presenta un model ontològic per a la representació del coneixement en el domini del turisme.

Link: <http://godo.atosorigin.es/wp-content/uploads/2010/05/r52v1-modelo-ontologico-antiguo-d21-godo-2007.pdf>

7.6. Protégé: un editor d'ontologies OWL

El programari *Protégé* és un entorn de treball *open source* que permet construir models de coneixement sobre dominis mitjançant l'ús d'ontologies. Aquest editor proporciona una gran varietat de funcions per tal de crear, manipular i visualitzar les diferents estructures associades a l'ontologia. A continuació podem veure la captura d'una de les pantalles que incorpora l'eina:

Figura 23



Captura de Protégé (versió 4)

- 1 Pestanyes per accedir als diferents components de l'ontologia i als modes de visualització
- 2 Finestra on es mostra la jerarquia de classes i subclasses (en aquest cas tenim seleccionada la classe *cim*)
- 3 Finestra on podem veure les propietats de la classe: les relacions i els atributs.
- 4 Finestra on podem veure les instàncies de la classe seleccionada.

7.6.1. Instal·lació del programari

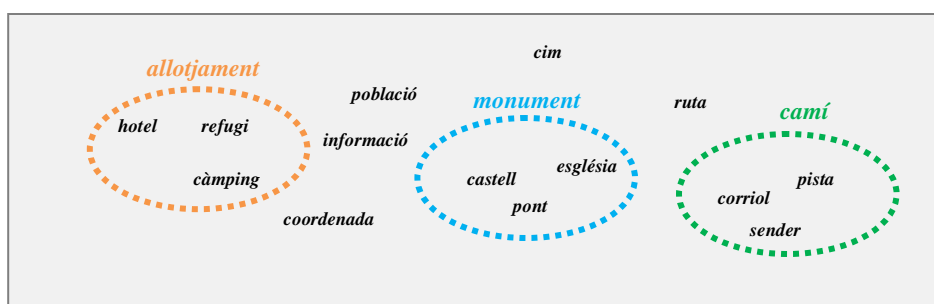
Per tal d'instal·lar aquesta eina d'edició d'ontologies cal anar a la pàgina <http://protege.stanford.edu> i descarregar el software pertinent. Tot i que en un principi hem instal·lat i iniciat el disseny de l'ontologia amb la versió 4, la incompatibilitat amb un dels *plugins* de visualització (*OntoViz*), ha fet que finalment ens haguem decantat per la seva versió anterior (*Protégé v3.4.6*).

7.6.2 Construcció de l'ontologia de muntanya amb Protégé

El primer pas a fer per tal de construir l'ontologia pròpia és determinar el domini i la cobertura que aquesta tindrà. L'objecte d'estudi d'aquest treball són les ontologies turístiques geogràfiques de manera que se'n pot delimitar una primera àrea. Per tal de particularitzar aquest àmbit tan genèric, i motivats per aficions de caràcter personal, s'ha establert que la subàrea que conformaria el domini seria el turisme o lleure de muntanya.

Respecte la qüestió de considerar la possibilitat de partir d'una ontologia ja existent, després de fer una mica de recerca (*veure apartat 7.5*) s'ha determinat que els esquemes trobats no eren adequats al propòsit, ja que la majoria se centraven en el turisme de caràcter urbà o hoteler (com per exemple *turismo.owl*¹⁷ o *travel.owl*¹⁸). Per això s'ha procedit a construir una ontologia des de zero. Per a fer-ho s'han enumerat els termes importants que hauria de contenir l'ontologia. Com que el domini ha quedat ben definit, l'ontologia se centrarà en conceptes d'aquest àmbit que puguin tenir un determinat interès per tal de descriure el context de la muntanya:

Figura 24



Primera aproximació als elements del domini. Ja pot detectar-se alguna classe i s'intueixen agrupacions d'elements que donaran lloc a classes més genèriques.

Un cop determinat el domini i els termes que permeten particularitzar conceptes dins l'àrea escollida, es procedeix a la creació de l'ontologia mitjançant *Protégé*. Després d'executar el programa, se selecciona l'opció de *crear una ontologia OWL* i s'hi especifica el *URI* per tal d'identificar-la en el sistema (en el nostre cas, aquest és <http://www.semanticweb.org/ontologies/2011/3/OntologiaMuntanya.owl>).

A continuació es creen el conjunt de *classes* que formaran l'esquema: comencem per les principals, que pegen de la classe *Thing* –que ve per defecte i és la *Superclasse*– i després anem creant les subclasses utilitzant l'eina d'edició gràfica que ens proporciona el programa. Cal fer notar que les classes creades tenen una relació anomenada “*is-a*” obligatòriament, és a dir, una relació “*és-un*”. Això implica que és molt important escollir correctament el nom de les classes i subclasses per tal que el disseny sigui correcte.

¹⁷ Ontologia turística de la ciutat de Saragossa (www.zaragoza.es/cont/.../turismo/es/.../turismo.owl)

¹⁸ Ontologia Travel: <http://protege.cim3.net/file/pub/ontologies/travel/travel.owl>

En la figura següent podem observar la jerarquia de classes creada amb *Protégé* i representada gràficament mitjançant el *plugin OWLViz*¹⁹. En l'estructura que presenta es pot detectar un primer nivell de classes (just a continuació de la superclasse *Thing*), que corresponen als elements més genèrics de l'ontologia, i a mesura que s'avança cap a la dreta els elements es van particularitzant en subclasses que permeten definir de manera concreta les característiques de cada tipologia d'element del context.

Figura 25



Ontologia de muntanya representada gràficament amb OWLViz.

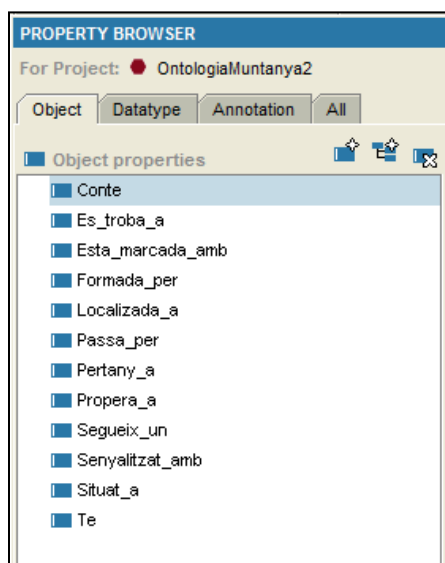
Com podem observar en la figura anterior, hi ha classes declarades en l'ontologia que poden resultar excessivament específiques pel cas de treball que ens ocupa (*veure requeriments i casos d'ús de l'aplicació pràctica a l'apartat 9.1 i 9.2, respectivament*), com per exemple la classe *camí* (que té les subclasses *corriol*, *sender*, *pista* i *camp_a_través*) o la classe *senyalització* (amb subclasses *pals*, *cinta* i *pintura*). El motiu pel qual s'han inclòs ha estat per permetre, en un futur, la integració de dades d'aquest tipus en l'aplicació web, ja que en el món de muntanya –i del senderisme en particular– la majoria de

¹⁹ OWLViz: <http://protege.wiki.stanford.edu/wiki/OWLViz>

rutes solen estar definides mitjançant dues estructures diferenciades: un *track GPS* i un document que descriu narrativament el camí indicant els trencalls que hi ha, la senyalització, el tipus de superfície, els monuments per on es passa....

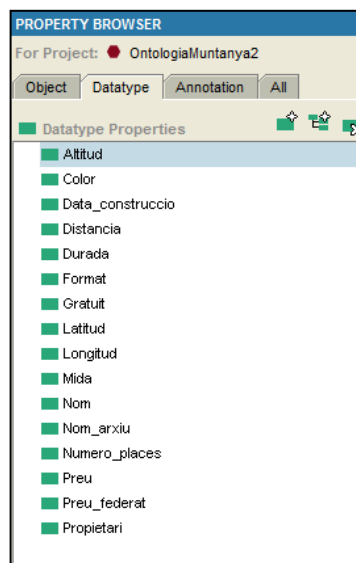
Un cop definides les classes que constituïran els elements de l'ontologia cal especificar les propietats dels objectes. Les propietats dels objectes són les relacions existents entre les classes (*figura 26*) i les característiques i atributs d'aquestes (*figura 27*). Si es vol aconseguir relacions diferents a la “és-un” cal fer servir aquesta opció. Un cop etiquetada la relació –normalment amb un verb–, s’ha de vincular a un rang i a un domini específics: el *domini* correspon a la classe que actua de “subjecte” del verb definit, mentre que el *rang* és la classe que fa de “complement o objecte”. Les classes que es trien per a la relació han de ser sempre les més genèriques ja que, tenint en compte que hi ha herència implícita, les relacions afecten a totes les subclasses que intervenen.

Figura 26



Llistat de relacions dins l'ontologia

Figura 27

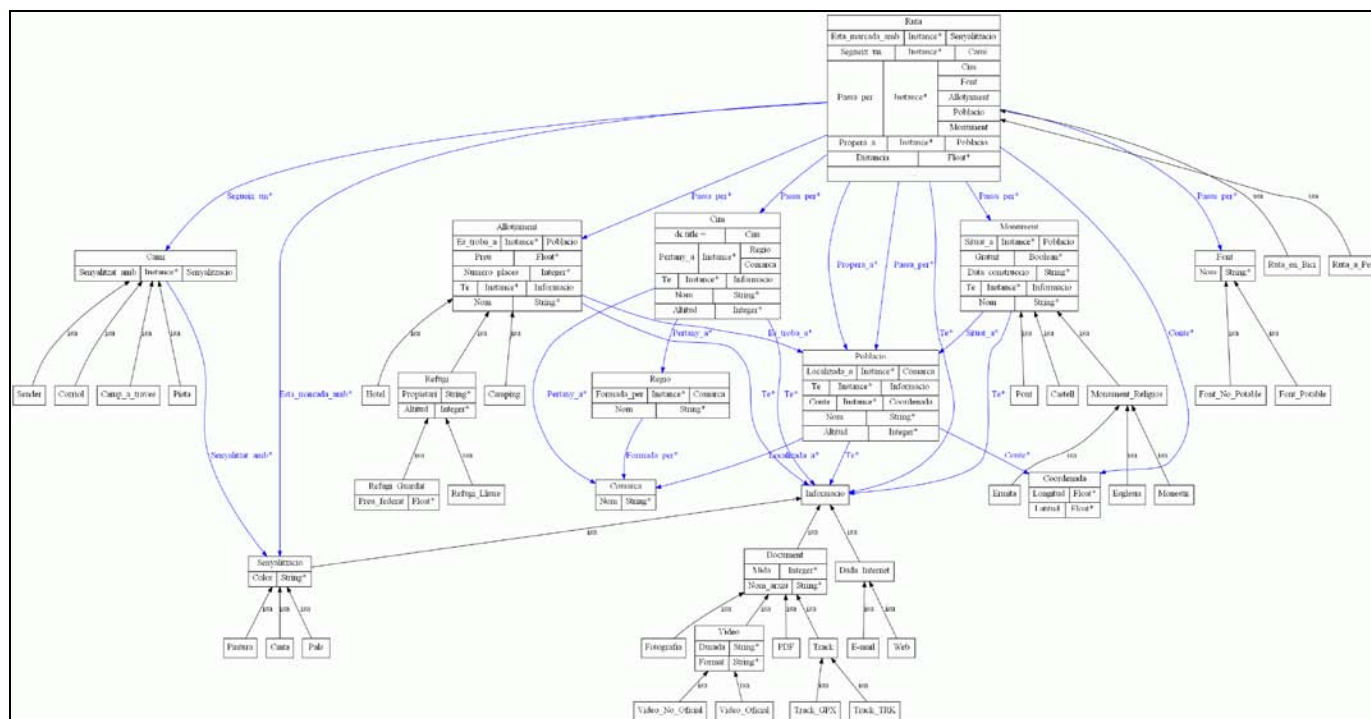


Llistat d'atributs de l'ontologia

A continuació es defineixen els atributs associats a cada classe, creant els diferents elements i vinculant-los a la classe o classes corresponents mitjançant el paràmetre *domini*. El tipus de dades d'aquests atributs s'especificaran amb el paràmetre *rang*. Cal notar que els atributs han de col·locar-se a les classes més generals ja que, igual que passava amb les relacions, són transmesos automàticament a les classes derivades. A través de la següent figura, que correspon a la representació gràfica de l'ontologia mitjançant el *plugin OntoViz*²⁰, podem observar les diferents relacions que s'estableixen entre classes i quins són els atributs que descriuen cada element.

²⁰ OntoViz: <http://protegewiki.stanford.edu/wiki/OntoViz>

Figura 28



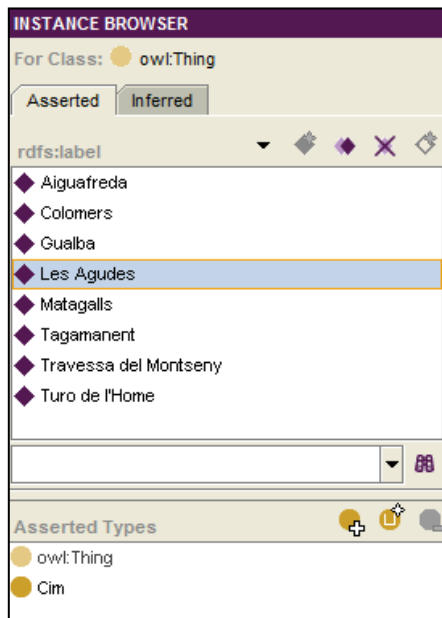
Ontologia de muntanya representada gràficament amb *OntoViz*.

Si fem un *zoom* sobre la figura anterior podem enumerar les propietats (relacions i atributs) de les classes més significatives de l'ontologia:

- **Ruta.** Té els atributs *nom* i *distància*. Les seves relacions són: *Ruta Passa_per* Cim/Allotjament/Monument/Població/Font, *Ruta Segueix_un* Camí, *Ruta Està_marcada_amb* senyalització, *Ruta conté* Coordenada, *Ruta Té* informació.
- **Cim.** Té els atributs *nom* i *altitud*. Les seves relacions són: *Cim Pertany_a* Regió, *Cim Té* informació.
- **Població.** Té els atributs *nom* i *altitud*. Les seves relacions són: *Població Té* informació, *Població Localitzada_a* Comarca, *Població Conté* coordenada.
- **Monument.** Té els atributs *nom*, *Data construcció* i *gratuït*. Les seves relacions són: *Monument Situat_a* Població, *Monument Té* informació.
- **Allotjament.** Té els atributs *nom*, *preu* i *Numero_places*. Les seves relacions són: *Allotjament Es_troba_a* Població, *Allotjament Té* informació.

Per acabar, es poden definir algunes instàncies de l'ontologia. Per a fer-ho s'indica el nom que tindrà la instància, se n'especifica el tipus seleccionant la classe que particularitzarà i s'omplen els valors dels diversos atributs que la descriuen. A continuació podem veure un llistat amb algunes instàncies, en aquest cas la instància anomenada *Les Agudes* està seleccionada, de manera que a la part inferior de la finestra es pot veure que la classe que instancia és Cim.

Figura 29



Llistat d'instàncies de l'ontologia

8. Bases de dades XML natives

L'aparició de l'*XML* i el seu gran potencial per a la gestió de coneixement ha fet que, paral·lelament al sorgiment d'altres tecnologies que el complementen augmentant-ne la capacitat semàntica (*RDF*, *OWL*...), hagi sorgit una nova generació de bases de dades capaces d'emmagatzemar els nous elements semi-estructurats. Tot i trobar-se encara en fase de desenvolupament, aquest nou model de base de dades pot constituir una bona alternativa a les bases de dades relacionals, permetent establir relacions entre elements sense fer ús de taules.

En aquest capítol veurem en què està basada una base de dades *XML* nativa (*apartat 8.1*), quines característiques presenta (*apartat 8.2*) i quins llenguatges específics hi ha per a la seva manipulació (*apartat 8.3 i 8.4*). Finalment, estudiarem el sistema de bases de dades *eXist* tot creant col·leccions de documents que instanciïn l'ontologia dissenyada en el capítol anterior i realitzarem consultes específiques que donin resposta als casos d'ús de l'aplicació pràctica final (*apartat 8.5*).

8.1. Què és una base de dades XML

Existeixen dos tipus de bases de dades *XML*: les basades en les dades i les basades en el document. El primer dels casos correspon a aquelles bases de dades en les quals s'usa el document *XML* com a mitjà de transport de les dades més que com a suport d'emmagatzematge. És a dir, les dades que a l'entrada tenen un format *XML* s'extreuen i es guarden a la base de dades en un format de model relacional tradicional (mitjançant taules que especifiquen les relacions entre elements). Posteriorment, quan s'extreu la informació es recupera el format original transformant les diferents relacions en documents de tipus *XML*. Les diverses traduccions de format es fan de manera automàtica per part de la pròpia base de dades.

El segon tipus de bases de dades, el corresponent a les basades en el document, atorguen la màxima importància a l'estructura original de les dades, que en aquest cas és el document *XML*. En aquest tipus de model l'estructura física del document preval per sobre de les dades que conté, de manera que n'esdevé la unitat mínima de suport i gestió. De forma clara, aquesta segona tipologia de base de dades presenta més puresa quant a tractament de coneixements, ja que aquests no són traduïts al format relacional tradicional per a emmagatzemar-se. Per tant, a partir d'ara ens centrarem exclusivament en aquest model (bases de dades *XML* natives).

8.2. Característiques de les bases de dades XML natives

Com s'ha esmentat anteriorment, les bases de dades natives se centren únicament en documents de tipus *XML*. Aquests documents poden emmagatzemar-se dins la base de dades formant col·leccions, que no són

més que agrupacions que segueixen una determinada lògica, i que desenvolupen el mateix paper que el que fan les taules en les bases de dades relacionals o els directoris en els sistemes de fitxers.

Totes les bases de dades *XML* natives suporten com a mínim un llenguatge de consulta (*XPath*, *Xquery*...) que permet realitzar *queries* sobre nodes de documents o col·leccions de documents, i aquestes es realitzen en funció de la coincidència entre un node i un cert criteri de cerca. Pel que fa a l'actualització i eliminació de documents, existeixen diverses estratègies que van des del simple reemplaçament de document, eliminant el document original i creant un nou document actualitzat, fins a l'ús de llenguatges específics de modificació, com podrien ser el llenguatge *XUpdate* o alguna extensió específica per a *XQuery* (*Update extensions*).

A banda de les característiques ja esmentades, les bases de dades natives també haurien de permetre validar els documents que contenen, indexar-los per ta l'accelerar el procés de cerca de dades i assignar-hi un identificador únic.

8.3. Els llenguatges de consulta de documents: XPath i XQuery

Els llenguatges de consulta de documents *XML* parteixen de la base que la informació que aquest conté s'estructura de forma arbòria, és a dir, hi ha un conjunt de nodes que es relacionen entre si mitjançant una jerarquia. Com podrem veure a continuació, aquesta distribució determinarà el procediment per a realitzar cerques dins del document, que diferiran substancialment respecte les consultes pròpies de model relacional quant a localització d'elements.

8.3.1. XPath

El llenguatge *XPath*²¹ és un estàndard del W3C que té com a finalitat bàsica permetre l'adreçament de parts concretes d'un document *XML*. Aquest llenguatge modela el document *XML* com un arbre de nodes, de manera que mitjançant un seguit d'expressions s'hi pot navegar com si es tractés d'un sistema de fitxers. Els diferents nodes del document *XML* es relacionen amb la resta de manera jeràrquica, és a dir, un node respecte un altre pot ser-ne el pare, el fill, un germà, un ascendent o un descendent. Per tal de seleccionar un node concret, *XPath* utilitza una sèrie d'expressions que n'indiquen la ruta o el camí:

Figura 30

Expressió	Comportament
<i>nom_del_node</i>	Selecciona tots els fills del node indicat per <i>nom_del_node</i>
/	Selecciona nodes des del node arrel
//	Selecciona nodes des del node actual
.	Selecciona el node actual
..	Selecciona el pare del node actual
@	Selecciona atributs

Expressions XPath

²¹ Estàndard XPath: <http://www.w3.org/TR/xpath/>

A continuació podem veure un exemple en què es pot apreciar el funcionament de la navegació per un document *XML* mitjançant *XPath*.

Figura 31

```
<?xml version="1.0"?>
<Travessa>
  <Ruta tipus='a_peu'>
    <nom> Carros de Foc </nom>
    <km> 55 </km>
  </Ruta>
  <Ruta>
    <nom tipus='en_bici'>
      Cavalls del vent
    </nom>
    <km> 97 </km>
  </Ruta>
</Travessa>
```

“Travesses.xml”. Document *XML* sobre el que es faran consultes *XPath*

Figura 32

Expressió	Comportament
/Travessa/Ruta	Selecciona tots els elements <i>Ruta</i> que són fills de <i>Travessa</i>
//Ruta	Selecciona tots els elements <i>Ruta</i> de l'arbre
//@tipus	Selecciona tots els atributs anomenats <i>tipus</i>
//Ruta[@tipus]	Selecciona tots els elements <i>Ruta</i> que tenen l'atribut <i>tipus</i>
// Ruta[@tipus='en_bici']	Selecciona tots els elements <i>Ruta</i> amb l'atribut <i>tipus=en_bici</i> .
//Ruta[km<50]	Selecciona totes les rutes amb distància inferior a 50 km
//nom //km	Selecciona tots els elements <i>nom</i> i tots els elements <i>km</i>

Taula d'exemples de consultes *XPath*

8.3.2 XQuery

El llenguatge de consulta *XML Query*, anomenat de manera abreujada *XQuery*²², és un estàndard publicat pel *W3C* que té com a finalitat extreure informació de col·leccions de dades expressades en *XML*. Aquest llenguatge utilitza expressions *XPath* per tal accedir a determinades parts del document i afegeix alguna expressió més, amb semàntica molt similar al llenguatge *SQL*²³, que permet ampliar el nivell d'expressió i transformació de resultats; aspecte que fa que es puguin realitzar dos tipologies de consulta: usant expressions *XPath* o bé usant el conjunt de clàusules *FLWOR*.

Les consultes mitjançant *XPath* constitueixen l'expressió més simple de consulta i serveixen per a localitzar nodes dins de documents *XML*. Mitjançant el terme `doc("nom_fitxer.xml")` s'indica sobre quin document es realitzarà la cerca i fent ús de les expressions pròpies d'*XPath* se'n defineix la ruta. Per tant, en el cas del document *XML* anterior (*Travesses.xml*) podríem fer la següent consulta:

```
doc("Travesses.xml")//Ruta/nom
```

Pel que fa referència a la segona tipologia de consultes, i a diferència del llenguatge *SQL* del model relacional (que empra les clàusules *select-form-where*), *XQuery* pot fer servir el conjunt de clàusules *FLWOR*, paraula formada amb les sigles de les cinc instruccions que admet: *For*, *Let*, *Where*, *Order by* i *Return*. La funció d'aquestes instruccions és la que s'indica a continuació:

- **For.** Vincula una o més variables (que s'expressa amb la forma `$nom_variable`) a expressions *XPath*. Cada node se selecciona i s'emmagatzema en una variable.
- **Let.** Vincula una variable al resultat d'una expressió.

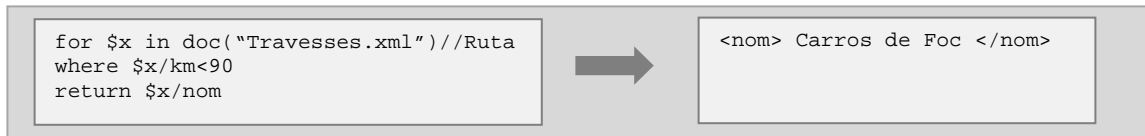
²² Especificació *XQuery*: <http://www.w3.org/TR/xquery/>

²³ *SQL* (Structured Query Language)

- **Where.** Permet discriminar alguns dels nodes seleccionats.
- **Order by.** Ordena els nodes resultants segons un criteri determinat.
- **Return.** Construeix el resultat de la consulta.

Vegem a continuació un exemple del funcionament d'aquest tipus de consulta sobre l'exemple de document *XML* de la figura 31:

Figura 33



Exemple de consulta XQuery FLWOR. A l'esquerra la consulta sobre el document "Travesses.xml". A la dreta el resultat.

8.4. Els llenguatges de modificació de documents: XUpdate i XQuery Update

Els llenguatges de modificació de documents *XML* permeten accedir a l'estructura arbòria que modela les dades dins d'un document i efectuar determinades operacions sobre els seus nodes per tal de provocar-ne una modificació. Com podem veure a continuació a través dels llenguatges *XUpdate* i *XQuery Update*, aquest tipus de tecnologia fa ús d'expressions *XPath* per navegar per l'estructura jeràrquica de la informació i seleccionar nodes, i empra un conjunt d'operacions específiques per tal d'aplicar-hi canvis.

8.4.1. XUpdate

El llenguatge *XUpdate*²⁴ empra les expressions definides mitjançant *XPath* per tal de seleccionar nodes que seran posteriorment processats. Aquest llenguatge es troba dins la iniciativa *XML:DB*²⁵, de manera que es pot emprar en qualsevol base de dades *XML* que tingui un *driver* d'aquest tipus (*XML:DB*). Les operacions de modificació (*xupdate:modifications*) que permet fer són les següents:

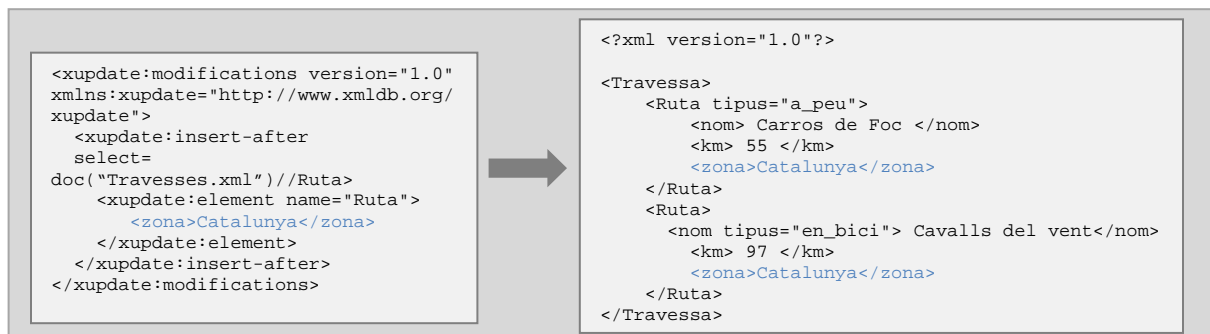
- Inserir nodes dins l'arbre. La seva localització s'indicarà mitjançant la selecció d'un node ja existent i situant el nou node abans (*xupdate:insert-before*) o després d'aquest (*xupdate:insert-after*).
- Afegir (*xupdate:append*) o eliminar un node de l'arbre (*xupdate:remove*).
- Modificar el contingut d'un node existent (*xupdate:update*)
- Reanomenar atributs o elements després d'haver-los creat (*xupdate:rename*)
- Altres operacions

²⁴ XUpdate: <http://xmldb-org.sourceforge.net/xupdate/xupdate-wd.html>

²⁵ XML:DB: <http://xmldb-org.sourceforge.net/xupdate/index.html>

Vegem a continuació un exemple d'ús d'aquest llenguatge de modificació de documents *XML*:

Figura 34



Exemple de modificació de nodes amb XUpdate. A l'esquerra, l'ordre d'inserir l'element *zona* en el document "Travesses.xml". A la dreta, el resultat d'aquesta modificació sobre el document.

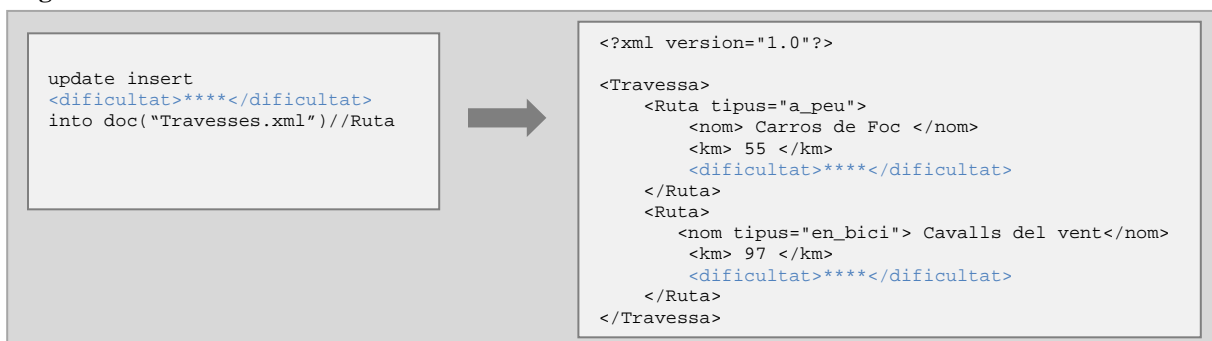
8.4.2. XQuery Update

*XQuery Update*²⁶ és una extensió del llenguatge *XQuery* que proporciona expressions específiques per tal que es puguin modificar documents de tipus *XML*. Es tracta d'una iniciativa creada pel W3C i recentment ha estat aprovada com a recomanació (Març 2011). Les funcions bàsiques de modificació que permet són:

- Inserir un nou node abans, després o dins d'un node especificat
- Eliminar un node
- Reemplaçar un node per un altre
- Reanomenar un node

Per a dur a terme les funcions esmentades, aquesta extensió d'*XQuery* introdueix cinc tipus d'expressions: insert, delete, replace, rename i transform. Vegem a continuació un exemple:

Figura 35



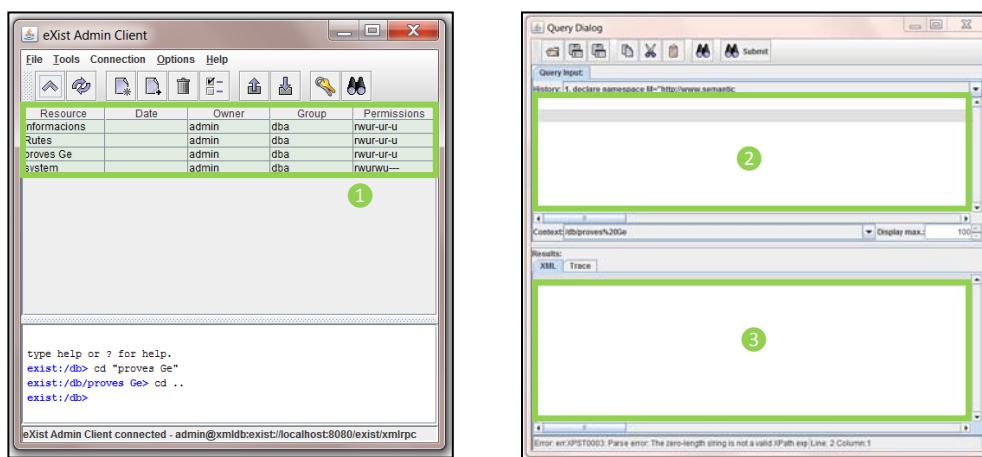
Exemple de modificació de nodes amb XQuery Update. A l'esquerra, l'ordre d'inserir l'element *dificultat* en el document "Travesses.xml". A la dreta, el resultat d'aquesta modificació sobre el document.

²⁶ XQuery Update: <http://www.w3.org/TR/xquery-update-10/#id-update-operations>

8.5. El SGBD eXist

El SGBD *eXist*²⁷ és una eina *open source* per administrar bases de dades de tipus *XML*. El seu element bàsic és el document i aquest pot agrupar-se amb d'altres individus mitjançant les col·leccions, constituint un conjunt d'elements relacionats per un criteri lògic. A diferència de les taules emprades en el model relacional, *eXist* es fonamenta en un model jeràrquic i basat en documents per a estructurar la informació, de manera que les relacions entre elements han de fer-se de manera explícita.

Figura 36



Captura del client d'eXist.

- 1 Conjunts de col·leccions de documents que hi ha al sistema.
- 2 Finestra on s'especifiquen les sentències de consulta o modificació.
- 3 Finestra on es mostren els resultats de la *query*.

8.5.1. Instal·lació

Per tal d'instal·lar aquesta eina de gestió de bases de dades cal anar a la pàgina web <http://exist.sourceforge.net/download.html> i descarregar el software pertinent. En el nostre cas hem fet servir la versió 1.4.0 (*eXist-setup-1.4.0-rev10440.jar*).

8.5.2. Emmagatzemament, consulta i modificació d'instàncies de l'ontologia

Després d'haver dissenyat l'ontologia de Muntanya (veure apartat 7.6.2), procedirem a emmagatzemar les instàncies en el sistema de bases de dades *eXist*. Com que el programari *Protégé* no incorpora cap eina automàtica per a generar instàncies susceptibles de ser carregades a *eXist*, aquestes han estat generades de manera manual. Per a fer-ho s'han creat arxius de tipus *XML* que utilitzen les mateixes etiquetes que les declarades en les classes de l'ontologia i s'han omplert amb els valors específics que creen una instància segons els recursos disponibles (les rutes específiques i els elements que les formen). Finalment, s'ha vinculat l'ontologia amb les instàncies generades mitjançant la declaració d'un *namespace*.

²⁷ SGBD eXist: <http://exist.sourceforge.net/>

Vegem a continuació l'extracte d'un d'aquests documents:

Figura 37

```
<?xml version="1.0" encoding="windows-1252"?>

<M:Ruta_a_Peu
xmlns:M="http://www.semanticweb.org/ontologies/2011/3/OntologiaMuntanya.owl">
  <M:Nom>Matagalls-Montserrat</M:Nom>
  <M:Distancia>83.3</M:Distancia>
  <M:Passa_per>
    <M:Cim>
      <M:Nom>Matagalls</M:Nom>
    </M:Cim>
    <M:Monument>
      <M:Nom>Monestir de Montserrat</M:Nom>
    </M:Monument>
    <M:Poblacio>
      <M:Nom>Aiguafreda</M:Nom>
    </M:Poblacio>
  </M:Passa_per>
</M:Ruta_a_Peu>
(...)
```

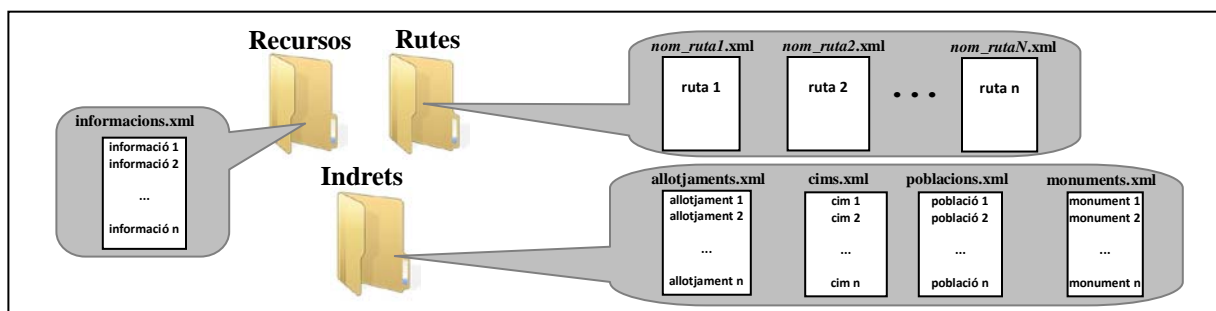
Extracte d'una instància XML de l'ontologia de muntanya.

Com es pot observar en l'ontologia creada (veure apartat 7.6.2), existeixen atributs que són compartits per múltiples classes, com per exemple l'atribut *nom*. Aquesta característica es mapeja directament al document XML, on s'observa l'existència d'aquesta etiqueta per cada element amb entitat de l'arbre. Com podem veure més endavant, la compartició d'etiquetes implicarà haver de declarar amb major detall les expressions XPath per tal d'accedir correctament a l'element que es vol seleccionar.

Com que *eXist* no permet declarar relacions de manera implícita, aquestes s'han establert explícitament emprant l'atribut *nom* (que actua a mode d'identificador). Així, per exemple, es pot comprovar que en la figura anterior la relació de l'ontologia anomenada *Passa_per* vincula una ruta amb un seguit d'elements (cims, monuments, poblacions...) identificats per un nom. Aquests elements, al seu torn, estaran instanciats de manera individual en d'altres documents i se'n permetrà la identificació mitjançant aquest mateix atribut. El motiu pel qual s'ha decidit emprar un identificador no numèric ha estat el fet de garantir la facilitat tant de comprensió com de gestió de continguts.

L'emmagatzemament de les diverses instàncies dins *eXist* s'ha fet agrupant per col·leccions els diferents documents. Inicialment es va plantejar una col·lecció per cada tipologia d'element amb entitat (rutes, poblacions, monuments...) i dins d'aquestes un document per cada instància de l'element, però la reduïda extensió d'alguna de les instàncies va fer replantejar-ne la distribució resultant-ne la següent arquitectura de col·leccions:

Figura 38



Distribució de la informació de l'ontologia dins d'eXist.

Un cop emmagatzemades unes quantes instàncies, s'ha procedit a fer les proves de consulta i modificació de documents amb la finalitat de trobar aquelles sentències que donaran resposta als casos d'ús de l'aplicació final (*veure apartat 9.2*). Per tal de validar el resultat de les diferents cerques s'ha treballat directament sobre el *client d'eXist*, executant una *query* específica que s'ajusti al que es vol obtenir en el cas d'ús concret i contrastant el resultat obtingut amb els fitxers originals. Com que el nombre de *queries* a realitzar per donar resposta a totes les combinacions possibles de preferències d'usuari –quant a paràmetres de cerca– és considerable, s'ha exemplificat el correcte funcionament mitjançant una selecció de mostres: dues que permeten comprovar la consulta i una que permet validar la modificació.

A la figura següent podem observar la interacció que s'efectua amb la base de dades quan un usuari selecciona l'opció de realitzar una **consulta per ruta**. Aquesta opció permet escollir una ruta dins un conjunt donat (que s'extreu de la base de dades) i seleccionar, d'una banda, sobre quins dels elements que conté es vol conèixer informació i, de l'altra, amb quin tipus de format es vol visualitzar aquesta informació.

Figura 39

En accedir a l'opció de **cercar per ruta**, automàticament es carreguen des de la base de dades totes les rutes disponibles.

Com podem veure en aquesta captura d'eXist, de moment tenim les rutes "Roger Hug III", "Travessa del Montseny" i "Matagalls-Montserrat".

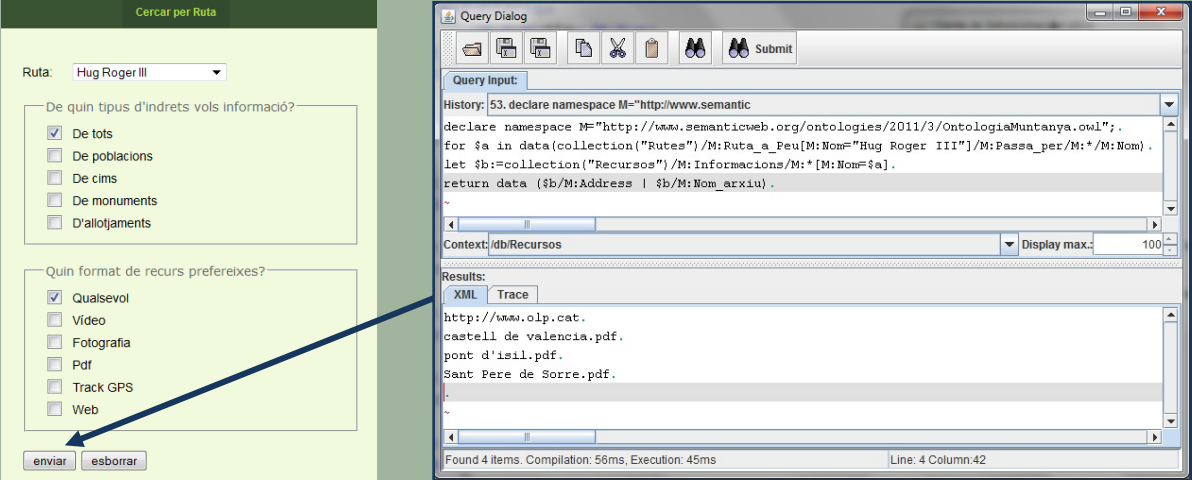
En prémer el botó **enviar**, s'obté com a resultat de la cerca els documents pdf associats a monuments de la ruta "Ruger Hug III": "castell de valencia.pdf", "pont d'isil.pdf" i "Sant Pere de Sorre.pdf".

Realització d'una consulta a eXist mitjançant XQuery (1).

Seguint amb la mateixa modalitat de consulta, a l'exemple de funcionament següent podrem veure quin resultat s'obté si, donada la mateixa ruta que en la figura anterior, se selecciona l'opció de consultar tots els recursos associats a tots els elements d'una ruta i amb qualsevol mena de suport. Com podrà observar-se, en aquest cas no s'ha exemplificat la càrrega de rutes (que funcionaria igual que en l'exemple anterior) i l'única captura que es mostra fa referència al resultat de la cerca segons els paràmetres especificats.

Figura 40

Si de la ruta seleccionada (*Hug Roger III*) marquessim les caselles de **TOTS** els tipus d'elements i **QUALSEVOL** recurs, obtindríem: Les webs associades als elements per on passa la ruta (en aquest cas la web de la població d'Olp) i els recursos de què disposa (en aquest cas només hi teníem pdfs), que és el que hi ha en aquest moment en la BD.

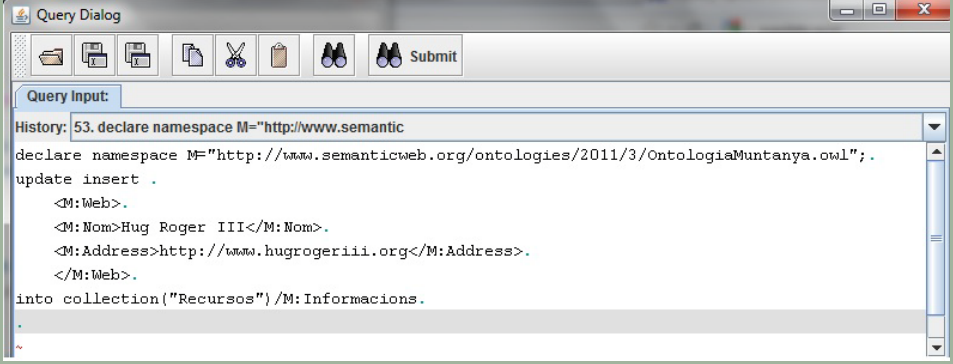


Realització d'una consulta a eXist mitjançant XQuery (2).

Finalment, podem veure un exemple de modificació sobre un document de la base de dades. Concretament en el fitxer corresponent a les informacions (dades sobre els recursos), s'afegirà una nova entrada corresponent al recurs web d'una ruta.

Figura 41

Inserir un element de tipus web al fitxer que hi ha a la col·lecció *Recursos* amb el nom de la ruta al qual s'associarà i l'adreça.



Realització s'una inserció mitjançant XQuery Update.

9. Aplicació pràctica

El TFC té un vessant pràctic que es manifesta mitjançant el disseny i la implementació d'un prototip simple que posi en ús algunes de les tecnologies estudiades fins al moment. Tot i que certament la implementació constituirà un producte independent de la memòria, la part del disseny de l'aplicació quedarà recollida en aquest document. En aquest capítol, per tant, detallarem els requeriments que presenta l'aplicació a desenvolupar (*apartat 9.1*) i farem una anàlisi funcional per tal de determinar les accions que permetrà fer el *software* resultant tot acompanyant l'explicació d'exemples de funcionament (*apartat 9.2*). Finalment analitzarem l'arquitectura de l'aplicació i les tecnologies que fa servir (*capítol 9.3*).

9.1. Anàlisi de requeriments

El programa a implementar consistirà en una aplicació web que contemplarà els requeriments que es detallen a continuació:

Si s'accedeix amb el perfil d'usuari, l'aplicació haurà de permetre donar informació i servir un conjunt de fitxers en funció de les preferències especificades. Mitjançant la selecció d'opcions a través d'un formulari, l'usuari seleccionarà un element geogràfic, escollirà els elements turístics o indrets sobre els quals vol conèixer informació (població, monument, allotjament...) i el format amb què vol que aquesta li sigui mostrada (vídeo, *pdf*...). Aquestes dades determinaran la consulta que es farà sobre la base de dades, el resultat de la qual es mostrarà a l'usuari per tal que pugui accedir als enllaços dels recursos. Concretament es vol oferir a l'usuari la possibilitat de realitzar quatre modalitats de cerca:

- **Per ruta.** Aquesta modalitat ha de permetre escollir una ruta d'un llistat i possibilitar la selecció de diferents elements sobre els quals es pot sol·licitar informació (indrets d'una ruta), tot escollint el format amb què aquesta serà mostrada. El resultat de la cerca serà un llistat amb tots els indrets i els enllaços a informacions associades.
- **Per població.** Ha de permetre escollir una població d'un llistat i possibilitar tant la selecció de diferents elements sobre els quals es pot sol·licitar informació com el format amb què aquesta serà mostrada. El resultat serà el mateix que en el cas anterior i, de manera addicional, es mostrarà en un mapa la localització de la població mitjançant *Google Maps*.
- **Per coordenades.** Ha de permetre introduir una latitud i una longitud i possibilitar la selecció de diferents formats amb què es pot mostrar una informació. Els resultats de la cerca mostraran un llistat amb totes les informacions (segons el format especificat) de qualsevol població que es trobi a menys de 25 km de la coordenada introduïda. Per cada ítem trobat es llistaran les informacions associades i s'indicarà la distància a què es troba des de la coordenada introduïda. De manera addicional es permetrà de situar totes les poblacions en un mapa mitjançant *Google Maps*.

- **Per monuments.** Ha de permetre seleccionar diferents tipologies de monument (església, pont, castell...) i determinar el format d'informació que es vol consultar. El resultat de la cerca mostrarà totes les informacions associades a monuments (segons tipologia seleccionada) de totes les àrees geogràfiques presents en la base de dades.

Si s'accedeix amb el perfil d'administrador, a més de les opcions citades anteriorment, l'usuari podrà donar d'alta, eliminar o modificar instàncies de la base de dades.

9.2. Anàlisi funcional

A continuació es mostra el diagrama de casos d'ús on es mostren les funcionalitats que tindrà l'aplicació (figura 42). Posteriorment es procedirà a descriure amb més detall els casos d'ús principals (*Consultar instància*, *Donar d'alta instància*, *Eliminar instància* i *Modificar instància*) mitjançant la seva especificació i a través d'un diagrama de flux que en defineixi el comportament.

Figura 42

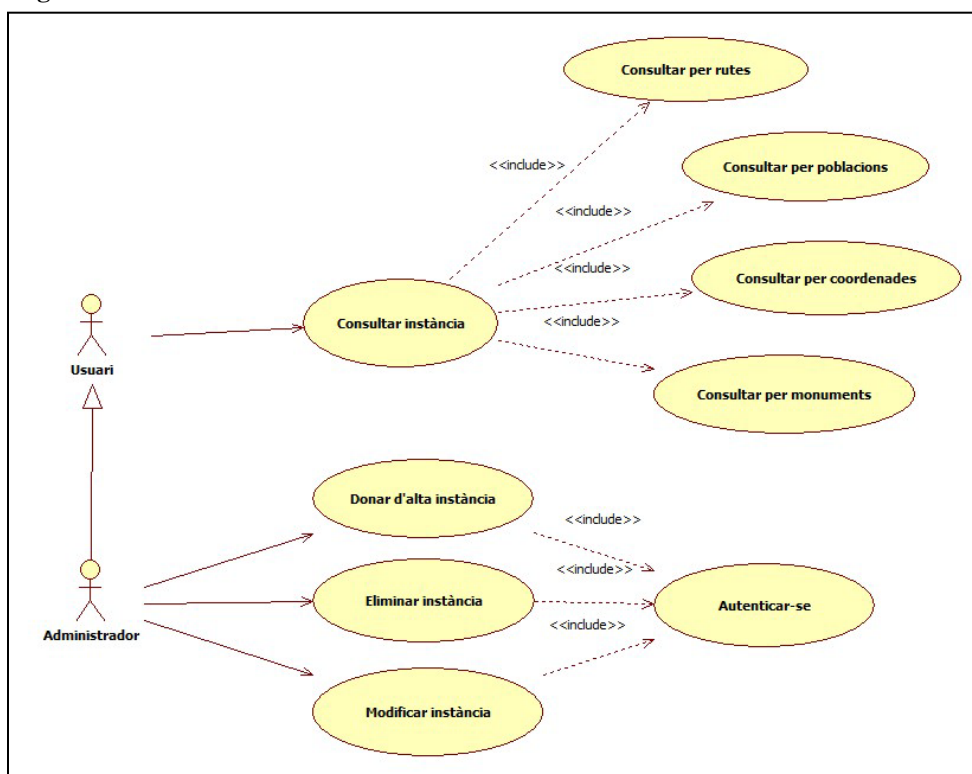


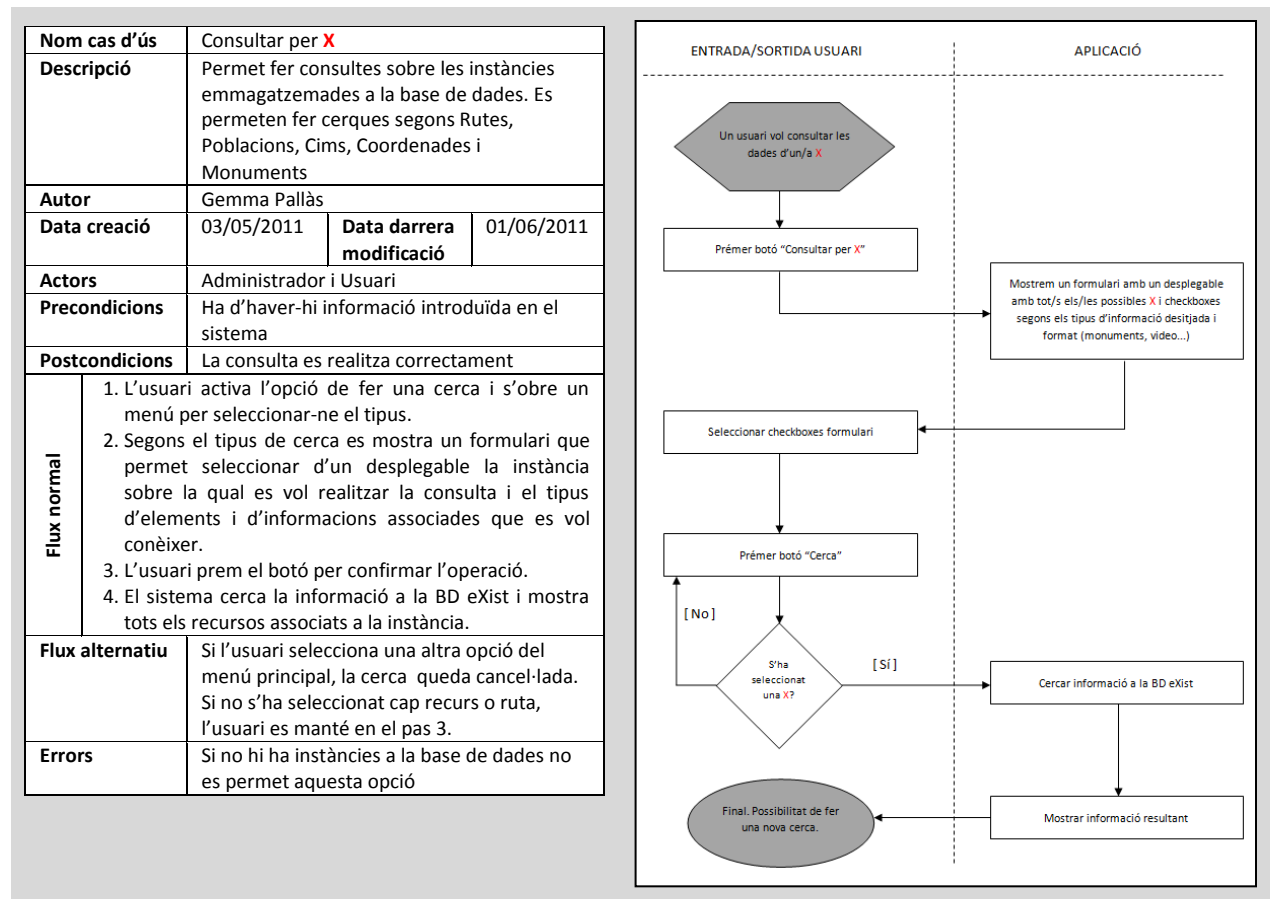
Diagrama de casos d'ús de l'aplicació

Com es pot veure en la figura anterior l'aplicació admet dos possibles actors: un usuari qualsevol i l'usuari administrador. Com que l'administrador hereta les funcionalitats de l'usuari normal, podrà realitzar tant les consultes públiques com les funcionalitats pròpies de gestió. Per tal de realitzar aquestes últimes accions, l'usuari caldrà que passi per la funcionalitat d'autenticar-se.

9.2.1. Consultar una instància

Aquest cas d'ús representa de manera genèrica les quatre modalitats de cerca que s'ofereixen a l'usuari. En els diferents diagrames es fa servir la variable **X**, que en el cas que ens ocupa pot prendre els valors [Ruta, Població, Coordenades, Monument]. Tot seguit podem veure l'especificació genèrica d'aquest cas d'ús (a l'esquerra) i el seu diagrama de flux (a la dreta).

Figura 43



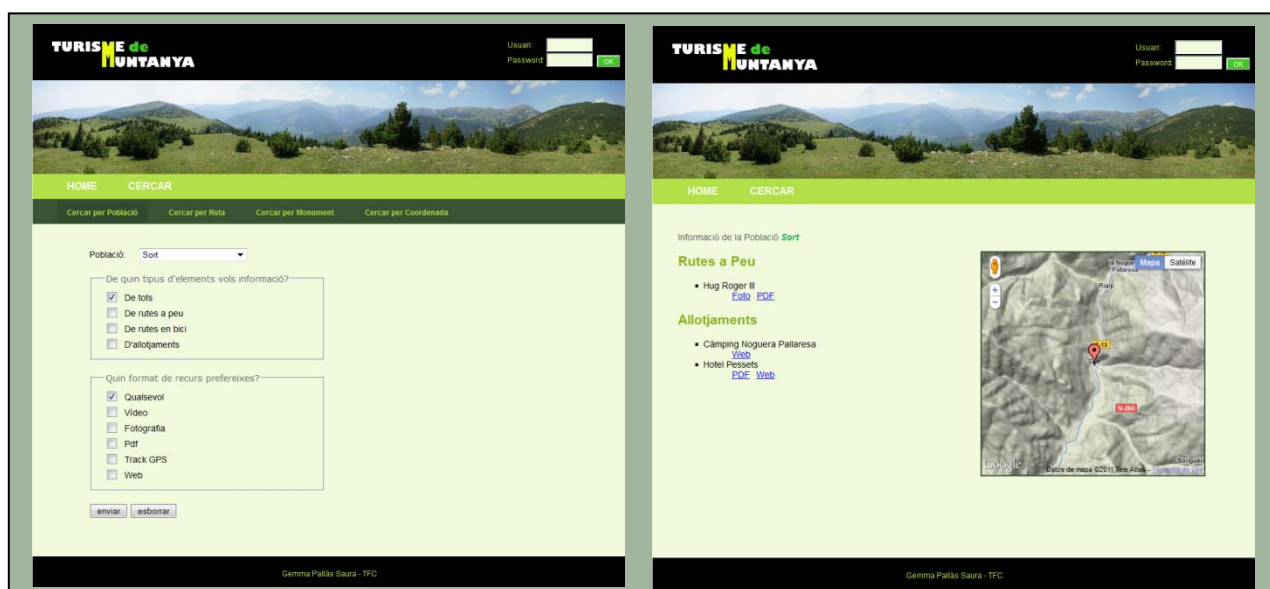
Especificació i diagrama de flux (1)

Com s'ha esmentat anteriorment, l'especificació i el diagrama de flux fan referència a una consulta genèrica. Podem veure quins paràmetres pot seleccionar l'usuari i quins resultats obtindrà en funció de cada modalitat de cerca a les captures de pantalla de l'aplicació que es mostraran a continuació.

En el primer dels casos podem observar com es comporta l'aplicació quan l'usuari selecciona l'opció de **Cercar per Població**. En la captura d'especificació dels paràmetres de cerca (*imatge de l'esquerra*) podem observar que en el desplegable de **Població** s'ha seleccionat terme *Sort* i que s'han deixat activats els *checkboxes* corresponents a mostrar tota la informació amb qualsevol format. En confirmar l'operació es presenten els resultats que coincideixen amb els paràmetres especificats (*imatge de la dreta*): s'ha trobat una ruta a peu que passa per *Sort* –que té associades dues informacions: un *pdf* i una fotografia– i dos allotjaments que es troben en aquesta població –cadascun dels quals mostra els enllaços als recursos

que tenen associats—. Acompanyant el llistat de resultats es presenta un mapa que, mitjançant *Google Maps*, mostra la localització de la població seleccionada.

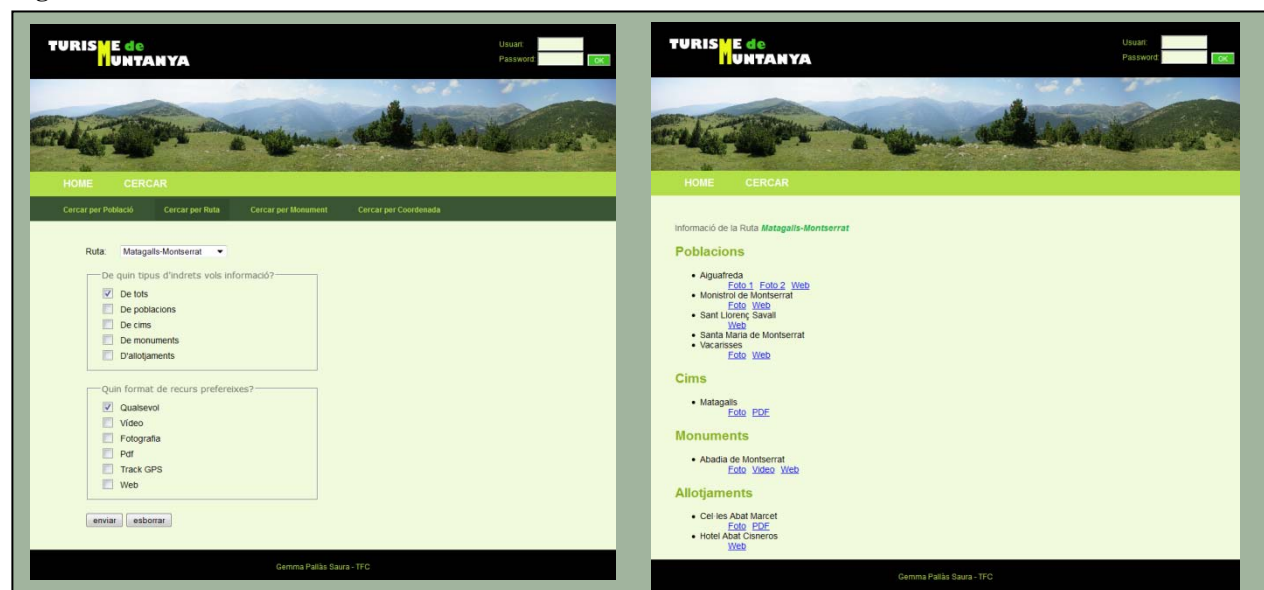
Figura 44



Captures de la CERCA (1)

L'opció de **Cercar per Ruta** permet fer una consulta de tots els indrets –i de les informacions que tenen associades– que poden trobar-se en una ruta. Com podem veure a l'especificació de la cerca d'aquest exemple de funcionament (*imatge de l'esquerra*), s'ha seleccionat la ruta *Matagalls-Montserrat* i s'han deixat activats els *checkboxes* que permeten consultar tots els recursos amb qualsevol format. En la captura dels resultats (*imatge de la dreta*) es llisten les diverses poblacions, cims, monuments i allotjaments per on passa la ruta, acompanyant-los dels enllaços als diversos recursos disponibles que tenen associats (pàgines web, fotografies, documents *pdf*, vídeos, *tracks*...).

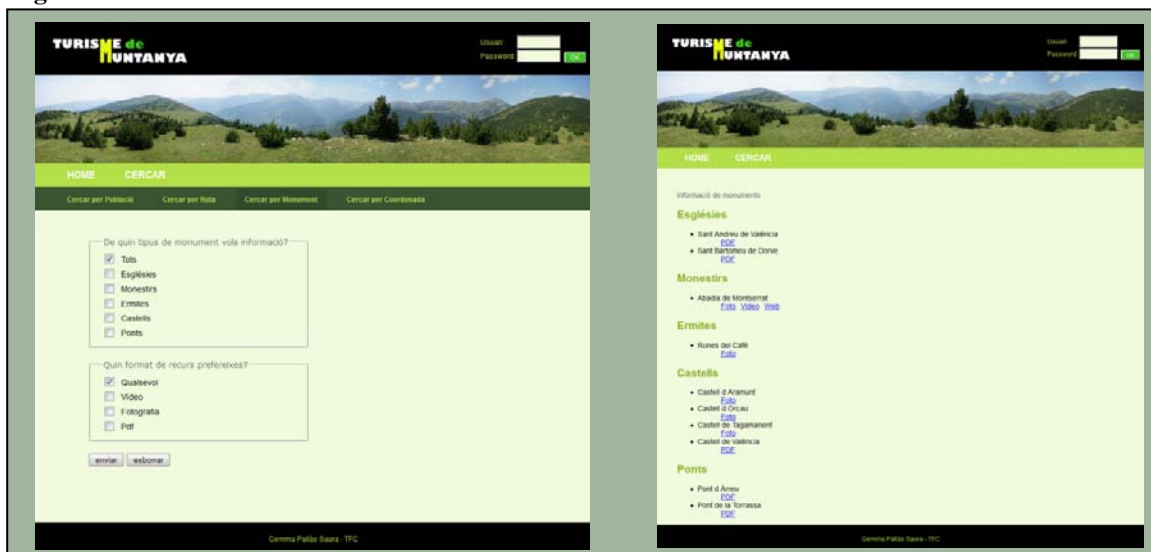
Figura 45



Captures de la CERCA (2)

L'opció de **Cercar per Monument** permet llistar tots els monuments d'una determinada tipologia (castell, ermita, pont...) que hi ha presents en la base de dades. En la captura de l'especificació de la cerca (*imatge de l'esquerra*) podem seleccionar el tipus de monument i el format de la informació que volem consultar. En aquest cas hem deixat activats els *checkboxes* que permeten mostrar tots els monuments amb informacions associades de qualsevol format. En la captura dels resultats (*imatge de la dreta*) podem observar el llistat de monuments classificats per tipologia i els enllaços als recursos associats.

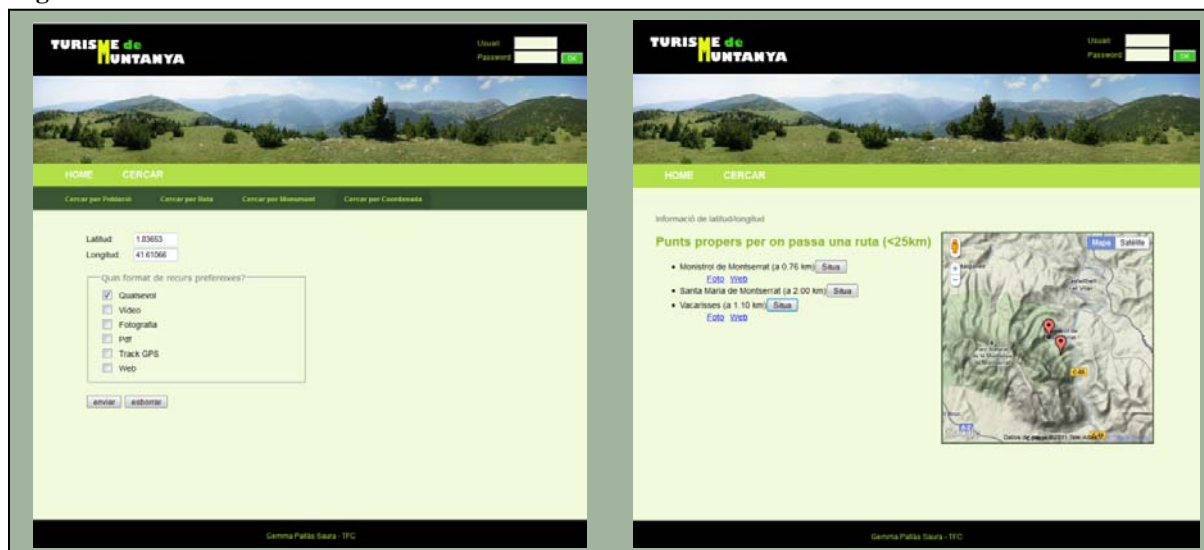
Figura 46



Captures de la CERCA (3)

Finalment, l'opció de **Cercar per Coordenades** permet situar les poblacions que es troben properes a un punt geogràfic especificat per l'usuari. En aquest cas hem especificat les coordenades d'un punt de la *Muntanya de Montserrat* (*imatge de l'esquerra*). En la captura dels resultats (*imatge de la dreta*) podem veure el llistat de poblacions que es troben a menys de 25 km del punt especificat –acompanyats de la distància fins al punt– i els enllaços als recursos que tenen associats. De manera addicional podem situar, mitjançant *Google Maps*, aquests indrets en el mapa que acompanya els resultats.

Figura 47

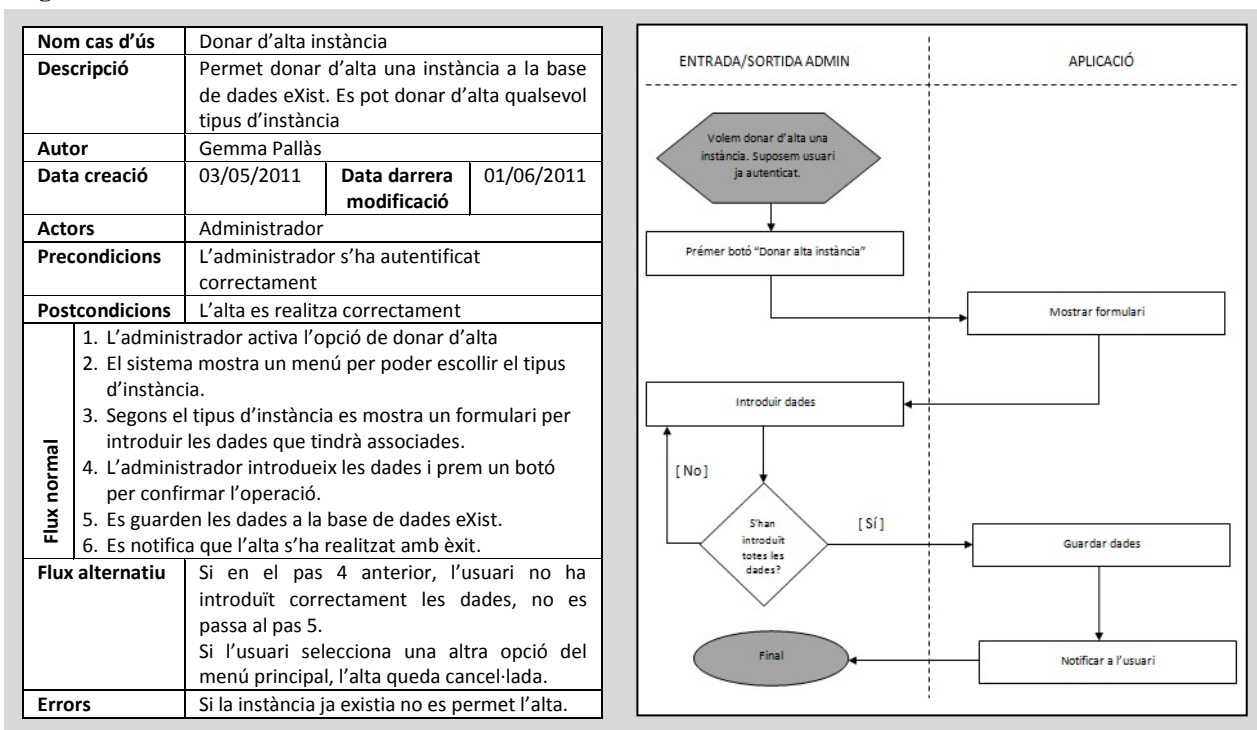


Captures de la CERCA (4)

9.2.2. Donar d'alta una instància

Aquest cas d'ús correspon a la inserció d'un nou element dins la base de dades. Per a poder executar-se cal que s'accedeixi a l'aplicació amb perfil d'administrador. Tot seguit en podem veure l'especificació (a l'esquerra) i el diagrama de flux, que permetrà d'entendre'n el funcionament (a la dreta).

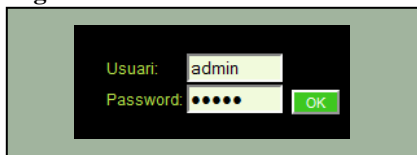
Figura 48



Especificació i diagrama de flux (2)

Com s'ha esmentat, per accedir a aquesta opció cal autenticar-se amb el perfil d'administrador. Per a fer-ho emprarem el terme “admin” tant pel nom d'usuari com per la paraula de pas:

Figura 49



Accés amb perfil d'administrador

En accedir amb aquest perfil el nombre de funcionalitats de l'aplicació s'incrementa per tal de permetre gestionar la informació continguda a la base de dades. En la captura que es mostra a continuació podem veure les diferents opcions disponibles per **donar d'alta una instància**.

Figura 50

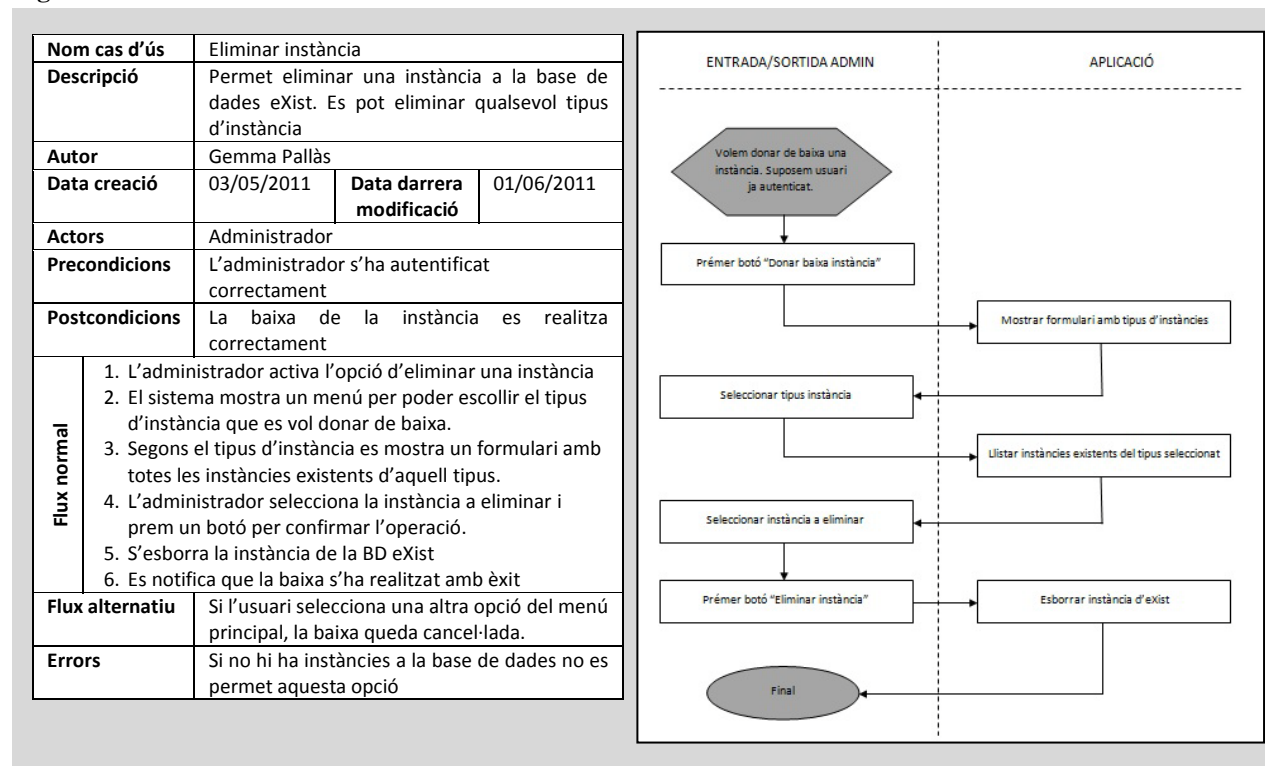
HOME	CERCAR	ALTA	BAIXA	MODIFICAR RUTA	
Alta Població	Alta Ruta	Alta Monument	Alta Recurs	Alta Cim	Alta Allotjament

Opcions de l'ALTA

9.2.3. Eliminar una instància

Aquest cas d'ús correspon a l'eliminació d'un element existent dins la base de dades. Per a poder executar-se cal que s'accedeixi a l'aplicació amb perfil d'administrador. Tot seguit en podem veure l'especificació (a l'esquerra) i el diagrama de flux, que permetrà d'entendre'n el funcionament (a la dreta).

Figura 51



Especificació i diagrama de flux (3)

De manera similar al cas d'ús anterior, la **baixa d'una instància** es realitza mitjançant la selecció d'un tipus de baixa concreta en un menú d'opcions. Un cop escollit el tipus d'instància a eliminar es pot seleccionar l'element concret en un desplegable que mostra, com podem veure a continuació, totes les instàncies d'un mateix tipus contingudes en la base de dades:

Figura 52

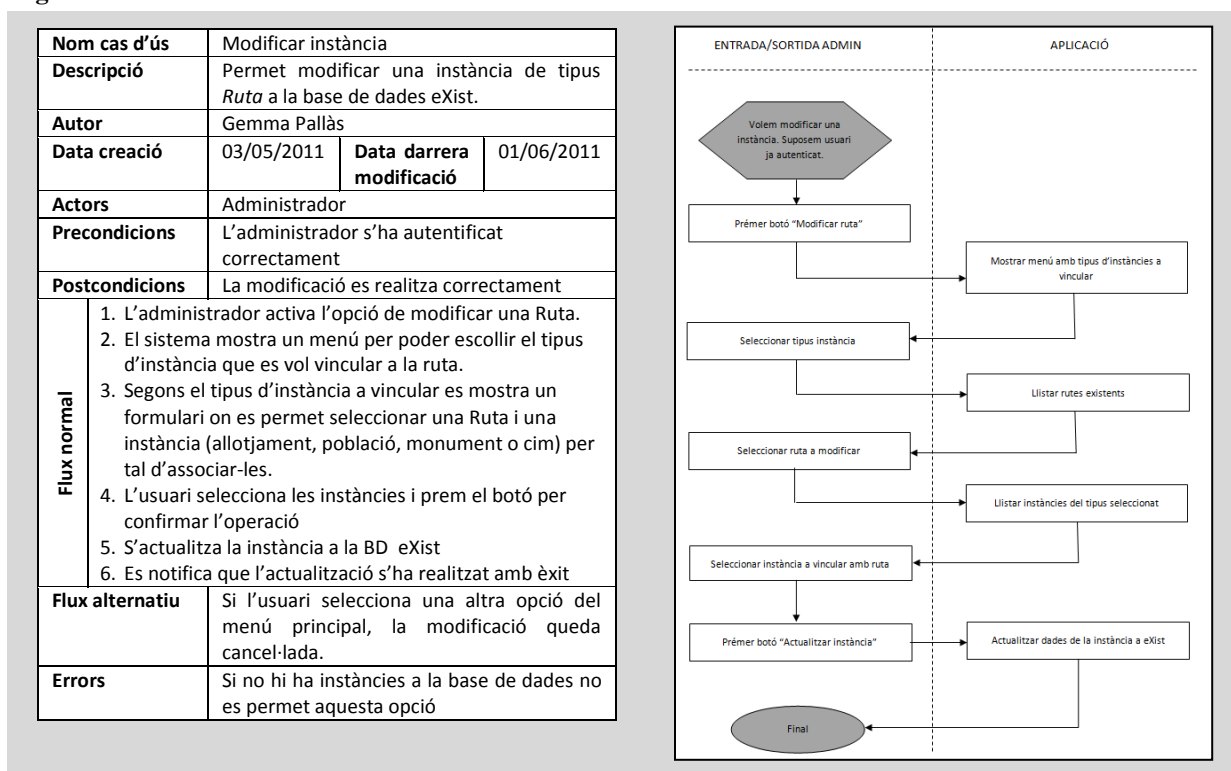


Opcions de la BAIXA

9.2.4. Modificar una instància

Aquest cas d'ús correspon a la modificació d'un element existent dins la base de dades. Concretament permet la modificació d'una instància de tipus *Ruta*, i serveix per poder vincular un nombre indeterminat d'indrets (poblacions, monuments,...) a una determinada *Ruta*. Per a poder executar-se cal que s'accedeixi a l'aplicació amb perfil d'administrador. Tot seguit en podem veure l'especificació (a l'esquerra) i el diagrama de flux, que permetrà entendre'n el funcionament (a la dreta).

Figura 53



Especificació i diagrama de flux (4)

A continuació podem veure un exemple de funcionament de la vinculació d'una Ruta (en aquest cas hem seleccionat la ruta *Matagalls-Montserrat*) amb una població que podem seleccionar d'entre les que existeixen actualment a la base de dades:

Figura 54

Vinculació d'una Població amb una Ruta.

9.4. Arquitectura de l'aplicació i tecnologies usades

Mitjançant el diagrama de desplegament que es mostra a continuació, es pot veure com es distribuïrien les diferents màquines que *hostatjarien* el projecte en un entorn real. En el seu desenvolupament però s'ha treballat de manera local simulant el conjunt de servidors en una mateixa màquina.

Figura 55

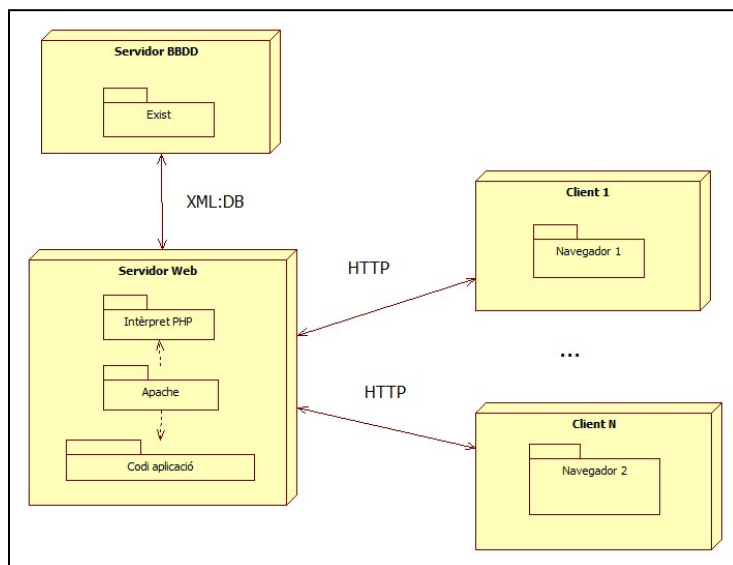


Diagrama de desplegament

Els paquets de software que hi hauria a cada servidor estarien distribuïts de la següent manera:

- **Servidor web:**
 - ✓ Intèrpret de PHP
 - ✓ Apache (Servidor web)
 - ✓ Codi aplicació
- **Servidor BDD:** eXist (Base de dades XML nativa)
- **Màquina client:** Navegador

Pel que fa referència a les tecnologies emprades per desenvolupar l'aplicació, s'ha fet servir el llenguatge *PHP 5*²⁸ *Orientat a Objectes* i s'ha fet ús de la llibreria *query-exist*²⁹ per a la connexió i realització de consultes i modificacions contra la base de dades eXist. Aquestes operacions s'han realitzat amb els llenguatges XQuery, XPath i XUpdate.

²⁸ PHP 5: <http://www.php.net/>

²⁹ Llibreria query-exist: <http://query-exist.sourceforge.net>

10. Conclusions i línies de futur

En aquest treball s'ha estudiat el concepte de Web semàntica i el conjunt de tecnologies que la constitueixen i/o hi donen suport. Partint des d'una revisió de la perspectiva evolutiva del mitjà, en què s'han estudiat breument els diferents salts tecnològics que ha sofert la Xarxa, s'ha analitzat quins són els elements que permeten d'entendre'n la darrera reconceptualització que ha sofert i que ha donat lloc a una nova manera de presentar, gestionar i generar informació, basada en la comprensió del significat de les dades contingudes.

En aquest capítol final veurem quines conclusions es poden extreure de la realització d'aquest estudi i de la posada en pràctica dels coneixements adquirits mitjançant la implementació de l'aplicació (*apartat 10.1*) i determinarem quins poden ser els passos de continuació que es podrien seguir per tal de treure el màxim de potencial al conjunt de tecnologies associades a la Web semàntica (*apartat 10.2*).

10.1. Conclusions

Un dels objectius fixats a l'inici d'aquest treball era el de conèixer i aprendre a crear ontologies tot adaptant-ne el disseny a un àmbit d'estudi concret. Mitjançant la utilització del programari de creació i gestió d'ontologies *Protégé* s'ha pogut vehicular l'adquisició de coneixements en aquesta matèria tot construint un esquema conceptual d'acord amb la temàtica suggerida. S'ha pogut observar el format del fitxer resultant de l'ontologia i comprendre'n el significat gràcies a l'estudi incremental previ dels diferents llenguatges que constitueixen la Web semàntica (*XML, RDF, OWL*).

Sense deixar de banda aquest darrer aspecte i assolint un altre dels objectius marcats, s'ha estudiat el format estàndard –i amb clara tendència a l'alça– dels fitxers d'intercanvi de dades a la Xarxa (*XML*). La seva facilitat de comprensió i de gestió en constitueixen el seu gran potencial, motiu pel qual s'ha determinat que aquest format de document esdevingués la base –almenys quant a suport de dades– de l'aplicació implementada. D'aquesta manera s'ha pogut instanciar l'ontologia creada tot distribuint les diferents informacions en conjunts conceptualment lògics per tal d'adequar-los al sistema d'emmagatzemament de les bases de dades natives *XML*.

Mitjançant l'estudi d'aquest tipus de bases de dades i dels diferents llenguatges de consulta i modificació que en permeten la gestió, s'ha après a navegar per un document *XML* i a realitzar-hi tota mena d'operacions (consultes, insercions,...). Aquest aspecte s'ha posat en pràctica mitjançant la base de dades *eXist*, establint inicialment una interacció directa amb la base de dades des de l'aplicació *client d'eXist* i posteriorment emprant la interacció indirecta que proporciona el llenguatge *PHP* integrant, d'aquesta manera, la base de dades en l'aplicació web.

En un vessant eminentment pràctic, s'ha après a dissenyar una aplicació web des de zero –seguint un cicle de vida en espiral basat en prototip– que s'adeqüés a la nova forma de gestionar la informació de la

Xarxa. És a dir, determinant la informació a presentar segons les preferències de l'usuari i gestionant aquesta informació segons el seu significat. D'aquesta manera s'ha pogut constatar que la Web semàntica constitueix una eina de gran potencial que permet a l'usuari personalitzar les cerques d'informació en funció de les seves necessitats, o de les seves preferències, aconseguint així una navegació per la *World Wide Web* més delimitada i eficient.

Pel que fa a les carències o dificultats observades durant la realització d'aquest treball es pot dir que probablement moltes d'elles són fruit de la poca maduresa de la tecnologia, aspectes que amb tota seguretat evolucionaran en els propers anys per tal de consolidar la Web semàntica. De manera general s'han detectat dificultats o mancances en els següents punts:

- L'eina de disseny d'ontologies *Protégé* no permet generar de manera automàtica instàncies de l'ontologia adequades per a ser emmagatzemades i gestionades mitjançant la base de dades *eXist*. El procés de traducció és manual i, per tant, pot estar subjecte a errors o a inconsistències.
- En les bases de dades natives *XML* no existeix integritat referencial –almenys en *eXist*–, aspecte que implica haver de realitzar un control acurat d'errors (especialment en cas d'eliminació d'instàncies).
- La classe existent (*API*) per a gestionar la base de dades *eXist* mitjançant *PHP* presenta mancances pel que fa al tractament de dades, com ara la manera poc consistent de retornar-les.
- Els llenguatges de consulta i modificació de dades no són prou madurs –cal tenir en compte que fa pocs mesos que *XQuery Update* ha estat publicat com a estàndard– i provoquen dificultats en la interacció amb *eXist*. Per exemple des del *client d'eXist* es treballa molt fàcilment amb *XQuery Update* però no amb *XUpdate*, i en canvi amb l'*API* per a *PHP* la interacció amb la base de dades es restringeix a *XUpdate*.
- En l'actualitat no existeixen gaire *hostings* que permetin integrar una base de dades com *eXist*.

10.2. Línies de futur

Tot seguit es descriuen un conjunt d'aspectes que s'han identificat com a possibles millores per tal de treure el màxim de potencial a la tecnologia i a l'aplicació:

- Refer la classe existent (*API*) que connecta *PHP* i *eXist* per fer-la més consistent (per exemple aconseguint que les dades que retornen les consultes realitzades amb aquesta classe tinguin un format estàndard).
- Incorporar la resta d'informació contemplada en l'ontologia per tal d'ampliar la possibilitat de personalització de cerca (per exemple en funció d'altituds, preus...) o augmentar-ne funcionalitats (com per exemple poder consultar el tipus de camí, la senyalització o afegir rutes amb bicicleta).
- Permetre incorporar i obtenir informació de l'aplicació des de dispositius mòbils.
- Lligar el contingut de l'aplicació amb les xarxes socials per tal de potenciar-ne les visites.

Bibliografia

1. **Berners-Lee, T; Hendler, J; Lassila, O** (2001). *The semantic Web*. Scientific American Magazine. [en línia]. http://nopiedra.files.wordpress.com/2008/05/the-semantic-web_-_scientific-american.pdf
2. **Boag, S; Chamberlin, D; i altres** (2010). *XQuery 1.0: An XML Query Language*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/xquery/>
3. **Bourret, Ronald** (2005). *XML and Databases*. [en línia].
4. <http://www.rpbourret.com/xml/XMLAndDatabases.htm#nativedefinition>
5. **Bray, T; Hollander, D; i altres** (1999). *Espacios de nombres en XML*. World Wide Web Consortium. [en línia]. <http://html.conclase.net/w3c/xml-names-es/>
6. **Bray, T; Hollander, D; i altres** (2009). *Namespaces in XML 1.0 (Third Edition)*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/REC-xml-names/>
7. **Bray, T; Paoli, J; i altres** (2006). *Extensible Markup Language (XML)*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/2006/REC-xml11-20060816/>
8. **Calderón, Marcos** (2011). *Web 3.0*. Universidad Carlos III de Madrid. [en línia]. <http://web30websemantica.comuf.com/index.html>
9. **Clark, J; DeRose, S** (1999). *XML Path Language (XPath)*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/xpath/>
10. **Eriksson, Henrik**. *Semantic Web description*. Linköping University. [en línia]. http://sirnet.metamatrix.se/material/SIRNET_08/semantic_web.doc
11. **Gao, S; Sperverg, CM; i altres** (2009). *XML Schema Definition Language (XSD) - Part 1*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/xmlschema11-1/>
12. **Heflin, Jeff** (2009). *OWL Web Ontology Language – Use Cases and Requirements*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/webont-req/>
13. **Laux, A; Martin, L** (2000). *XUpdate WD*. XML:DB Project. [en línia]. <http://xmldb-org.sourceforge.net/xupdate/xupdate-wd.html#N995a79>
14. **Manola, F; Miller, E** (2004). *RDF Primer*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/rdf-primer/>
15. **McGuinness, D; van Harmelen, F** (2004). *OWL Web Ontology Language*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/owl-features/>
16. **Peterson, D; Gao, S; i altres** (2009). *XML Schema Definition Language (XSD) - Part 2*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/xmlschema11-2/>
17. **Robie, J; Chamberlin, D; i altres** (2011). *XQuery Update Facility 1.0*. World Wide Web Consortium. [en línia]. <http://www.w3.org/TR/xquery-update-10/#id-update-operations>
18. **W3C** (2010). *Guía breve de web semántica*. World Wide Web Consortium. [en línia]. <http://www.w3c.es/divulgacion/guiasbreves/websemantica>
19. **Socorro, R; Simón, A; i altres** (2008). *Las ontologías en la representación del conocimiento*. Centro de Estudios de Ingeniería de Sistemas (CEIS), Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE). [en línia]. <http://nopiedra.files.wordpress.com/2008/05/rep-con-ontologias.pdf>
20. **W3C Schools** (2009-2011). *Introduction to RDF*. W3C Schools. [en línia]. http://www.w3schools.com/rdf/rdf_intro.asp
21. **W3C Schools** (2009-2011). *XML Schema Tutorial*. W3C Schools. [en línia]. <http://www.w3schools.com/schema/default.asp>
22. **W3C Schools** (2009-2011). *XPath Syntax*. W3C Schools. [en línia]. http://www.w3schools.com/xpath/xpath_syntax.asp
23. **W3C Schools** (2009-2011). *XQuery Tutorial*. W3C Schools. [en línia]. <http://www.w3schools.com/xquery/default.asp>