

Guia per visitar Barcelona

Judit Cabezas Cabiol
Enginyeria Tècnica de Gestió

Treball Final de Carrera – XML i Web Semàntica

Consultor: Joan Anton Perez Braña
Segon Semestre Curs 2011
UOC

1. Índex

1. Índex	2
2. Presentació	4
2.1. El document.....	4
2.2. El Projecte	4
3. Objectius	5
3.1. Objectius de l'assignatura	5
3.2. Objectius del TFC	5
4. Apartats	6
5. Temporització i fites	6
5.1. Tasques a realitzar	7
5.2. Gràfic de precedències	9
5.3. Fites.....	9
5.4. Calendari.....	10
5.5. Diagrama de Gannt.....	11
6. Avaluació del material necessari	12
6.1. Coneixements previs	12
6.2. Requeriments de maquinari	12
6.3. Requeriments de programari	12
7. Incidències i riscos – Pla de contingències	13
7.1. Desplaçaments per motius laborals	13
7.2. Avaries	13
7.3. Compromisos.....	14
8. Evolució de la web	14
8.1. Web tradicional	14
8.1.1. Funcionament de la WWW	15
8.2. Web 2.0.....	16
8.2.1. Funcionament de la Web 2.0	17
8.3. Web Semàntica.....	19
8.3.1. Funcionament de la Web Semàntica.....	21
8.4. Diferències entre els diferents tipus de Webs i conclusions.....	22
9. TECNOLOGIES DE LA WEB SEMÀNTICA	23
10. XML.....	24
10.1. Característiques d'un document XML	24
10.2. Format d'un document XML.....	24
10.3. Tecnologies del XML i com funcionen	26
11. RDF.....	27
11.1. Característiques del RDF.....	27
11.2. Format d'un document RDF	27
12. RDF Schema	28
12.1. Característiques principals	28
12.2. Classes RDFS	28
12.2.1. Bàsiques.....	28
12.2.2. Defineixen relacions	29
12.2.3. Restriccions de Propietats.....	29
12.2.4. Altres.....	29

13. OWL	29
13.1. Característiques de OWL	29
13.2. Per què OWL?	30
13.3. Els 3 subllenguatges de OWL.....	30
13.4. Escollir un sub-llenguatge o un altre	31
14. ONTOLOGIA	32
14.1. Què és una ontologia?.....	32
14.2. Procés de creació d'una ontologia	32
14.3. Llenguatges d'ontologies.....	33
14.4. Propietats en una Ontologia.....	34
15. Anàlisi d'ontologies ja existents	34
16. Bases de dades natives XML.....	35
16.1. Característiques.....	35
16.1.1. Processament de dades.....	35
16.1.2. Emmagatzemant.....	36
16.1.3. Cerques.....	36
16.2. Àrees d'aplicació.....	36
16.3. eXist	37
16.4. Realitzar consultes: XPath	37
16.4.1. Model de dades XPath.....	38
16.4.2. Tipus de nodes.....	39
16.5. Realitzar consultes: XQuery.....	41
16.5.1. Característiques.....	41
16.5.2. Aplicacions	42
17. Aplicació.....	42
17.1. Requeriments	42
17.2. Descripció de l'aplicació	43
17.3. Tecnologia a emprar	43
18. Creació d'una Ontologia pròpia.....	43
18.1. Classes de la ontologia	44
19. Mapa conceptual	45
20. Casos d'ús	46
21. Conclusions.....	46
22. Webs d'on extreure informació de l'ontologia	47
23. Bibliografia.....	48

2. Presentació

L'exposició d'aquest document serà una presentació tant del contingut d'aquest document com del projecte a realitzar.

2.1. El document

En aquest document es podrà trobar una presentació del projecte final de carrera, i estarà dividit en 3 parts diferenciades.

1. Es definiran l'abast i els objectius que es pretenen assolir amb l'elaboració d'aquest projecte des del punt de vista d'elaboració com de la temàtica a tractar.
2. Un cop definits aquests objectius es definiran les tasques a portar a terme, es realitzarà una planificació del temps i es dividirà en les diferents fites a portar a terme.
3. Finalment es detallarà tota la infraestructura necessària per l'elaboració, així com els possibles problemes que puguin sorgir en l'elaboració.

2.2. El Projecte

L'objectiu d'aquest treball és conèixer els mecanismes de què es disposa per aportar significat i semàntica als documents i construir un model que permeti obtenir la informació desitjada.

El projecte es dividirà en diferents fases:

- Conèixer el llenguatge XML i XML-Schema que ens servirà com a mecanisme d'intercanvi d'informació.
- Conèixer RDF i RDF-Schema
- Conèixer OWL i els nous elements que aporta respecte a RDF
- Conèixer i aprendre a treballar amb PROTÉGÉ.
- Dissenyar una ontologia que permetrà agrupar les paraules pels conceptes lèxics.

Els canvis que estan succeint a la Web comporta la necessitat de tractar dades semiestructurades.

XML és un llenguatge que es fa servir per intercanviar informació entre diferents sistemes d'informació degut a la flexibilitat i independència de les fonts de dades i de formats fixes que proporciona.

La pretensió és que els ordinadors puguin entendre la semàntica de la informació que hi ha a la Web i sigui possible integrar diverses fonts d'informació malgrat la procedència i formats de cadascuna d'elles.

3. Objectius

Ara es procedirà a descriure els objectius que pretén assolir aquest treball, diferenciant-los en dos parts, els objectius propis de l'assignatura, és a dir el que es pretén aprendre, i els objectius del Projecte que seria tot allò que volem que es realitzi i sigui capaç de fer la nostra aplicació.

3.1. Objectius de l'assignatura

L'objectiu d'aquesta assignatura és realitzar un treball que mostri els coneixements adquirits durant la realització de les diverses assignatures que hi han al pla docent d'Enginyeria Tècnica en Informàtica de Gestió.

Els objectius a assolir són:

- A partir d'uns requisits que ens donen s'haurà de realitzar una aplicació Informàtica que compleixi aquests requisits.
- Planificar el desenvolupament del projecte de forma que es faci de forma ordenada.
- Proposar una solució al problema que ens han donat.
- Treballar a fons els aspectes formals que s'han de tindre en compte i realitzar quan es desenvolupa un projecte.
- Elaborar una memòria del projecte a on estiguin tots els aspectes de documentació així com els problemes que han pogut sorgir i els canvis de planificació que s'han anat fent durant el procés.
- Elaborar una presentació del projecte per tal de poder mostrar el resultat final de l'aplicació.

3.2. Objectius del TFC

Els objectius del Treball Final de Carrera són:

- Conèixer el llenguatge XML: És un llenguatge d'Etiquetat Extensible molt simple, però estricte que facilita el intercanvi d'una gran varietat de dades. És un llenguatge similar a HTML però la seva funció principal és la de descriure dades i mostrar-los.

XML és un format que permet la lectura de dades mitjançant diferents aplicacions, serveix per estructurar, emmagatzemar i intercanviar informació.

- Conèixer el llenguatge RDF: són les sigles de **Resource Description Framework**, un llenguatge de descripció del W3C.

- Conèixer OWL: vol dir **Ontology Web Language**, un llenguatge de marcat per publicar i compartir dades utilitzant ontologies en la WWW. Té com a objectiu facilitar un model de marcat construït mitjançant RDF i codificat en XML

- Realitzar casos pràctics de les tecnologies abans esmentades.
Un altre objectiu que hi ha al realitzar aquest projecte és conèixer les implicacions i com s'ha d'elaborar un treball d'aquest tipus, aprendre a planificar el projecte, buscar informació del tema en concret que s'ha de desenvolupar, com s'ha de fer la presentació del projecte, resoldre els problemes que van sorgint...

4. Apartats

Ara es procedirà a fer una aproximació dels diversos apartats que hi hauran al treball i de quina extensió aproximada seran aquests apartats:

Apartat	Extensió
1. Índex	2 planes
2. Presentació	1 plana
3. Objectius	2 planes
4. Apartats	1 plana
5. Temporització i fites	5 planes
6. Avaluació del material necessari	1 plana
7. Incidències i riscos	2 planes
8. Evolució de la Web	9 planes
9. Tecnologia de la Web Semàntica	1 plana
10. XML	4 planes
11. RDF	1 plana
12. RDF Schema	2 planes
13. OWL	3 planes
14. Ontologia	3 planes
15. Anàlisi d'ontologies ja existents	1 plana
16. Bases de Dades natives	12 planes
17. Aplicació	2 planes
18. Creació d'ontologia pròpia	2 planes
19. Mapa conceptual	1 plana
20. Casos d'ús	1 plana
21. Webs d'on extreure informació de l'ontologia	1 plana
22. Bibliografia	1 plana

5. Temporització i fites

S'ha de fer una planificació de les diferents tasques a realitzar així com estructurar que s'haurà de fer en cada una de les entregues que s'han d'anar realitzant.

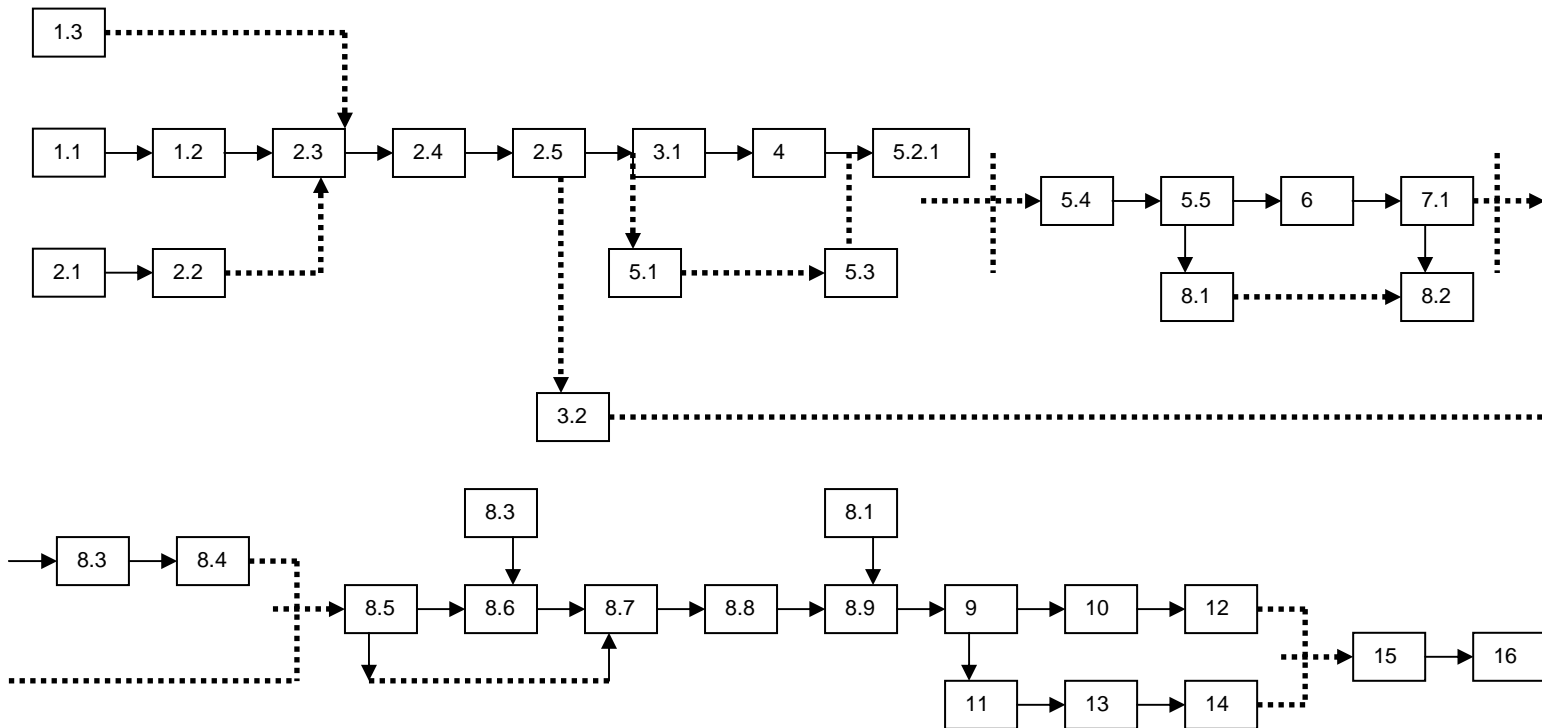
En aquest treball les dates de lliurament les marcaran les PAC's en canvi en un projecte de cara a un client les marcarà ell mateix.

5.1. Tasques a realitzar

Tasca	Precedents
1. Definició: definició del projecte	
1.1. Descarregar documentació: descarregar de les bústies de l'assignatura l'enunciat i tota la documentació facilitada pel consultor	
1.2. Llegir documentació: lectura detallada del pla d'estudis i de l'enunciat del treball, així com de tota la informació addicional aportada pel consultor	1.1
1.3. Trobada de síntesi: assistència a la primera trobada presencial del semestre. Aclariment de dubtes amb el consultor	
2. Esborrany PAC1: elaboració de l'esborrany del pla de treball	
2.1. Recerca d'informació: cerca a biblioteques, catàlegs digitals, cercador d'Internet, etc. De tots els temes a tractar en el treball. Intentar tenir tota la documentació necessària abans de començar el desenvolupament del treball pròpiament dit.	1.1
2.2. Recollir bibliografia: recollir a la biblioteca la bibliografia recomanada al pla d'estudis i a l'enunciat del TFC, així com la bibliografia trobada a la tasca anterior. La bibliografia recomanada al pla d'estudis i a l'enunciat del TFC, així com la bibliografia trobada a la tasca anterior. La bibliografia recomanada al pla d'estudis i al propi enunciat del treball és considerat material necessari pel bon desenvolupament d'aquest treball. Recollir també altres llibres sobre els temes a tractar que es puguin considerar d'interès	2.1
2.3. Pla de treball: redacció dels diferents punt que formen el pla de treball. Això és, objectius, guió, apartats que contindrà el treball, número de planes per apartat; incidències, riscos i els corresponents plans de contingència; fites i avaluació del material necessari. Deixar la temporització per quan estiguin definits tots els altres punts.	1.2, 1.3, 2.2
2.4. Temporització: un cop avaluat tot l'abast del treball, analitzar el temps necessari per realitzar cadascuna de les tasques, el temps disponible per portar-les a terme i fer la temporització, amb el corresponent diagrama de Gannt.	2.3
2.5. Lliurament de l'esborrany de la PAC1: lliurament de l'esborrany del pla de treball	2.4
3. Correcció PAC1: revisió de les correccions indicades a l'esborrany de la PAC1	
3.1. Revisió PAC1: en funció de les anotacions fetes pel consultor sobre l'esborrany de la PAC1, fer les correccions corresponents. Si cal, acabar de completar els punts anteriors.	2.5
3.2. Instal·lació: instal·lació del programari necessari pel bon desenvolupament d'aquest treball. Donat que al punt de treball ja es disposa de Microsoft Word, Microsoft Power point i MS Project, només restarà instal·lar el programa PROTÉGÉ. A l'ordinador del meu company, com a mesura de precaució, s'haurà d'instal·lar el Microsoft Project i el PROTÉGÉ	2.5
4. Lliurament PAC1: Lliurament del pla de treball	3.1
5. Esborrany PAC2: elaboració de l'esborrany de la PAC2	
5.1. Llegir XML: lectura acurada de tota la documentació recopilada sobre XML i XML-Schema. Si cal, cerca més informació per aprofundir.	2.5
5.2. Apartat 3: redacció del capítol 3 del treball, referent a XML i XML-Schema	4, 5.1
5.3. Llegir RDF: lectura acurada de tota la documentació recopilada sobre RDF i RDF-Schema. Si cal, cercar més informació per aprofundir	4, 5.1
5.4. Apartat 4: redacció del capítol 4 del treball, referent de RDF i RDF-Schema.	5.2, 5.3
5.5. Lliurament esborrany PAC2	5.4

6. Correcció PAC2: en funció de les anotacions fetes pel consultor sobre l'esborrany de la PAC2, fer les correccions corresponents. Si cal, acabar de completar els punts anteriors	5.5
7. Lliurament PAC2	6
8. Esborrany PAC3: elaboració de l'esborrany de la PAC3	
8.1. Apartat 2: un cop encaminat el desenvolupament dl treball, redacció del capítol 2 del treball, que constitueix la introducció	5.5
8.2. Llegir OWL: lectura acurada de tota la documentació recopilada sobre OWL. Si cal, cercar més informació per aprofundir	5.5
8.3. Apartat 5: redacció del capítol 5 del treball, referent a OWL	7, 8.2
8.4. Llegir PROTÉGÉ: lectura acurada de tota la documentació recopilada sobre PROTÉGÉ. Si cal, cercar més informació per aprofundir	8.3
8.5. Practicar PROTÉGÉ: practicar amb l'ús de PROTÉGÉ	8.4
8.6. Cas pràctic: disseny de l'ontologia proposada als Objectius del TFC. Desenvolupament amb PROTÉGÉ	8.5
8.7. Apartat 6: redacció del capítol 6 del treball, referent a PROTÉGÉ, el propi cas pràctic	8.6
8.9. Lliurament esborrany PAC3	8.1, 8.8
9. Correcció PAC3: en funció de les anotacions fetes pel consultor sobre l'esborrany de la PAC3, fer les correccions corresponents. Si cal, acabar de completar els punts anteriors	8.9
10. Lliurament PAC3	9
11. Conclusions: redacció de les conclusions del treball	8.9
12. Revisió final: lectura acurada del treball, revisant ortografia, redacció i els propis continguts	10
13. Síntesi: selecció de les parts del treball que seran d'utilitat per a la realització d'una presentació	11
14. Presentació: desenvolupament d'una presentació de 20 diapositives fetes amb Microsoft Power Point, amb una síntesi del treball	13
15. Lliurament memòria i presentació	12, 14
16. Debat: el tribunal plantejarà preguntes i situacions a l'alumne per mitjà del correu electrònic. Aquests missatges s'hauran de contestar en el termini de 24 hores	15

5.2. Gràfic de precedències



5.3. Fites

Per tal de realitzar el treball hi ha una sèrie de lliuraments ja definits que aniran marcant el ritme amb el que s'ha d'anar realitzant el projecte.

Tenint en compte les dates que tenim de lliurament podrem donar per data d'ini del Projecte el dia 03/03/2011 i data de lliurament 26/06/2011

Hi hauran 3 tipus diferents de "lliuraments":

- Lliurament d'esborranys de les entregues finals de les PAC's que vindria a ser un avançament al consultor del que finalment s'entregarà de forma que ens pugui fer correccions i comentaris al que hem estat fent fins al moment.
- Lliuraments oficials, les quals ja estan definides al Pla d'estudis.
- Debat entre els dies 20 i 26 de Juny, un període de temps en el que mitjançant unes preguntes realitzades per e-mail es passarà a avaluar el treball realitzat conjuntament amb les entregues ja realitzades.

- La PAC1, la qual serà aquest Pla de Treball tindrà com a data de lliurament el 16/03/2011
- La segona part, PAC2, tindrà com a data de lliurament 11/04/2011
- La entrega de la tercera PAC serà el dia 11/05/2011
- Finalment vindrà l'entrega final que serà el dia 12/06/2011

Fites	Data de lliurament
Esborrany PAC1	12/03/2011
Lliurament PAC1	16/03/2011
Esborrany PAC2	04/04/2011
Lliurament PAC2	11/04/2011
Esborrany PAC3	04/05/2011
Lliurament PAC3	11/05/2011
Lliurament Presentació i memòria final	12/06/2011
Inici Debat	20/06/2011
Final Debat	26/06/2011

5.4. Calendari

En termes generals els dies marcats en negreta són dies de cada dia i dedicaria 2 hores per la realització del treball encara que puntualment podria ser més en cas de fer falta.

Els dies en ombrejat vermell són festius i per tant podria dedicar un mínim de 5 hores, i els dies en ombrejat groc són dies amb lliurament marcat tant d'esborrany com de lliurament oficial.

Març							Abril						
	1	2	3	4	5	6					1	2	3
7	8	9	10	11	12	13	4	5	6	7	8	9	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24
28	29	30	31				25	26	27	28	29	30	

Maig							Juny						
						1			1	2	3	4	5
2	3	4	5	6	7	8	6	7	8	9	10	11	12
9	10	11	12	13	14	15	13	14	15	16	17	18	19
16	17	18	19	20	21	22	20	21	22	23	24	25	26
23	24	25	26	27	28	29	27	28	29	30			
30	31												

Total hores mes de Març: 35 + 26 = 61 hores

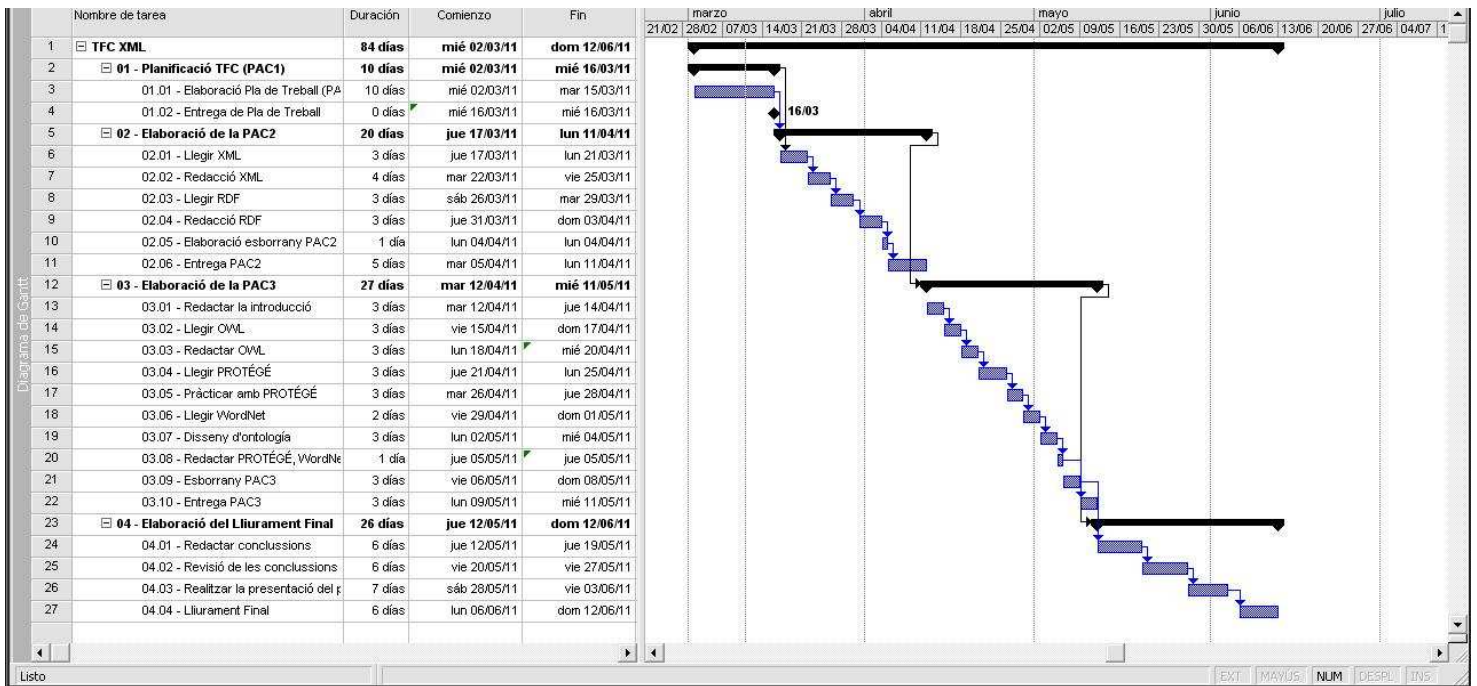
Total hores mes d'Abril: 55 + 16 = 71 hores

Total hores mes de Maig: 45 + 26 = 71 hores

Total hores mes de Juny: 15 + 8 = 23 hores

En definitiva hi haurà un mínim de 226 hores dedicades al projecte.

5.5. Diagrama de Gannt



6. Avaluació del material necessari

6.1. Coneixements previs

Encara que prèviament a la realització d'aquest treball no tinc coneixements previs en aquesta àrea espero a la finalització tindre com a mínim coneixements bàsics del XML.

6.2. Requeriments de maquinari

Estarà la possibilitat de utilitzar dos ordinadors indistintament, encara que el primer que es detalla serà el que s'utilitzarà com a principal:

- PC de sobretaula: AMD Phenom (tm) 6950 Quad-Core, amb processador 1'20 GHz, 3'25 GB de RAM, i com a Sistema Operatiu: Windows XP Professional
- Portàtil: Intel(R) Atom(TM) CPU N450 amb processador de 1'67 GHz, 1 GB de RAM, i com a Sistema Operatiu Windows 7.

6.3. Requeriments de programari

Per la realització d'aquest projecte com a Software s'utilitzarà com a Sistema Operatiu Windows XP Professional com ja s'ha descrit encara que en casos puntuals es farà servir Windows 7.

Per a la realització dels diagrames pertinents, planificació i seguiment del Projecte, el Microsoft Office Project 2003.

Així com per la realització de documents com aquest, i d'altres que s'hauran d'entregar es farà servir el paquet de Microsoft Office 2003 (Word, PowerPoint...)

També es farà servir el Protégée i eXist.

7. Incidències I riscos – Pla de contingències

7.1. Desplaçaments per motius laborals

En principi d'aquí a la data de lliurament final del projecte no hi ha previst cap desplaçament per motius laborals, en el cas de que sorgeixi algun desplaçament el procediment que faria servir seria avisar al consultor si aquest desplaçament pugues perjudicar la puntualitat d'un lliurament.

7.2. Avaries

Les incidències que poden anar sorgint poden ser diferents:

- Problemes amb qualsevol dels equips amb que es realitzarà el projecte
- Problema amb el subministrament d'electricitat el que pot portar a no poder realitzar el treball en moments en que estava programat.
- Incidència amb el subministrament d'Internet el que pot impossibilitar l'entrega d'alguna part del projecte dintre del termini establert.
- Infermetat de l'autor del treball, o familiar directe que pugui interferir amb el desenvolupament del projecte
- Projectes assignats a la feina que pugui treure temps del treball durant la setmana (de dilluns a divendres)

Per evitar o resoldre aquestes incidències, s'haurien de solucionar per exemple:

- Si hi ha cap problema amb els equips, s'hauria de resoldre buscant un altre equip amb el que realitzar el treball. Això és possible tenint còpies de seguretat del mateix en: correu personal (Gmail en el meu cas), correu de la UOC, Pen Drive d'ús personal, i tenint una còpia també al Dropbox, així com també tot el software necessari per poder instal·lar-lo en un altre equip.
- Si hi ha algun problema amb la electricitat que sigui per causa externa (de la companyia per exemple) s'haurà de fer un desplaçament a un altre lloc a on no hi hagi aquest problema i així poder treballar.
- En cas d'un problema amb Internet, igual que en el cas de problemes amb l'electricitat aniré a un altre domicili a on no hi hagi aquest problema, i en cas de no poder-se fer, mitjançant Internet que dispo en el mòbil (Google Nexus One) poder enviar un missatge al consultor del Projecte per avisar del motiu pel qual no s'ha fet l'entrega.
- En cas d'una infermetat pròpia o d'un familiar, s'haurà de prioritzar la realització del Projecte en la mida possible, i sens dubte primer de tot avisar al consultor.
- Per últim en cas que la feina pugui dificultar la realització del Projecte s'haurà de fer un treball de més hores durant el cap de setmana per poder recuperar el temps perdut durant la setmana.

7.3. Compromisos

L'únic compromís personal que tinc que podria interferir en la realització del treball és un casament el dia 25 de Juny durant el període de Debat.

Encara que tinc també l'entrega d'unes PACs d'altra assignatura intentaré que no perjudiqui a l'elaboració del treball.

8. Evolució de la web

La Web 1.0 o Web Tradicional era un tipus de web estàtica amb documents que mai s'actualitzaven. Els continguts eren rígids a la navegació (HTML i GIF). El disseny i producció estava a càrrec de qui coneixen sobre informàtica. Llocs amb finalitats generalment comercials i amb continguts d'alta i baixa qualitat administrats per un webmaster.

La Web 2.0 amplia la diversitat en continguts administrats per usuaris. Softwares i aplicacions que requereixen de la instal·lació al PC per a utilitzar-los. Disseny i producció sense necessitat de grans coneixements de informàtica. Accessibles i pràctics. Llocs amb fins diversos; a la majoria dels casos la construcció de comunitats que comparteixen interessos, pràctiques, informació...etc. Funció: produir, dissenyar, construir i compartir informació en diferents suports.

La Web 3.0 és el triomf del món virtual i de la intel·ligència de les màquines. A la web 3.0 es parla de conceptes tals com: Web 3D, Web centrada en multimèdia i Web permanent. La Web permanent és una Web que està a totes parts no només al PC o mòbil sinó a la roba, joies, automòbils, etc. L'usuari tindria una identitat a Internet amb una reordenació de totes les comunitats virtuals. Connexió il·limitada al ordinador, mòbils...a preus accessibles.

8.1. Web tradicional

La web tradicional va ser creada per Tim Berners-Lee, anomenada com WWW (World Wide Web).

Al principi eren poques persones les que tenien accés al contingut de la xarxa, encara que amb el pas del temps es va observar el gran potencial que tenia la WWW, en aquell moment es va crear el navegador Mosaic per poder facilitar el intercanvi d'informació.

Bàsicament podem entendre com a Web Tradicional (si parlem de versions seria la 1.0) bàsicament era un lloc de lectura encara que anava poc a poc evolucionant i introduint imatges, animacions...

Amb les companyies .COM (2001) es van evolucionar les pàgines HTML estàtiques fins a obtenir un nou nivell a on gran part del contingut era generat dinàmicament amb diversos sistemes i bases de dades, es tenien molt en compte factors com són l'estètica i funcionalitat.

8.1.1. Funcionament de la WWW

El primer pas consisteix en traduir la part nom del servidor de la URL en una adreça IP utilitzant la base de dades distribuïda de Internet coneguda com a DNS. Aquesta adreça IP es necessària per a contactar amb el servidor Web i poder enviar-li paquets de dades.

El següent pas és enviar una petició HTTP al servidor Web sol·licitant el recurs.

Al cas d'una pàgina web típica, primer se sol·licitarà el text HTML i després és immediatament analitzat pel navegador, el qual després fa peticions addicionals per als gràfics i altres fitxers que formen part de la pàgina. Les estadístiques de popularitat d'una web normalment estan basades en el número de pàgines vistes o de les peticions de servidors associades, o peticions de fitxer que tenen lloc.

Al rebre els fitxers sol·licitats des del servidor web, el navegador enredereix la pàgina tal i com es descriu en el codi HTML, el CSS i altres llenguatges Web. Al final s'incorporen imatges i altres recursos per a produir la pàgina que veu l'usuari a la seva pantalla.

La majoria de les pàgines web contenen hiperenllaços a altres pàgines relacionades i algunes també contenen descarregues, documents font, definicions i altres recursos web.

Aquesta col·lecció de recursos útils i relacions, interconnectats mitjançant enllaços de hipertext és el anomenat xarxa d'informació. Si un usuari accedeix de nou a una pàgina després d'un petit interval és probable que no es tornin a recuperar les dades del servidor web de la forma en que es va fer la primera vegada. Per defecte, els navegadors emmagatzemen en una cache del disc dur local tots els recursos web als que l'usuari va accedint. El navegador enviarà una petició HTTP només si la pàgina ha estat actualitzada des de la última carrega, en altre cas, la versió emmagatzemada es reutilitzarà per agilitzar la visualització de la pàgina.

Això és particularment important per a reduir la quantitat de tràfic Web en Internet.

8.2. Web 2.0

Una definició que es podria donar de les Webs 2.0 es que són més punts de trobada o webs que depenen de usuaris.

Els principis bàsics de la Web 2.0 segons la Web 2.0 Conference:

- La Web és la Plataforma
- La informació és el processador
- L'arquitectura participativa produeix efectes diferents a la xarxa
- La innovació sorgeix de característiques distribuïdes per desenvolupadors independents.
- Sempre serveis en Beta, és a dir, el final del cercle de l'adopció del Software

En definitiva la Web 2.0 ofereix eines amb la finalitat de beneficiar a l'usuari final, mitjançant els avantatges de les aplicacions d'escriptori amb la facilitat de la Web.

Serveis associats a la Web 2.0:

- Blogs: Per exemple Blogger
- Wikis: Wikipedia
- Recursos compartits: Youtube, Flickr
- Plataformes educatives: Moodle
- Xarxes socials: Facebook, Twiter

Tècniques:

- CSS
- Java Web Start
- XUL
- AJAX (exemple d'aplicacions no intrusives)
- RSS/ATOM (Redifusió/Agregació de dades)
- URLS senzilles
- Suport per postejar a un blog
- JCC i APIs REST o XML
- JSON
- Alguns aspectes de xarxes socials
- Mashup

En resum la Web 2.0 podríem dir:

- La informació s'ha de poder introduir i extreure fàcilment
- Els usuaris haurien de poder controlar la seva informació
- Les pàgines Web 2.0 amb més èxit poden ser utilitzats des d'un navegador
- L'existència de links és requisit imprescindible

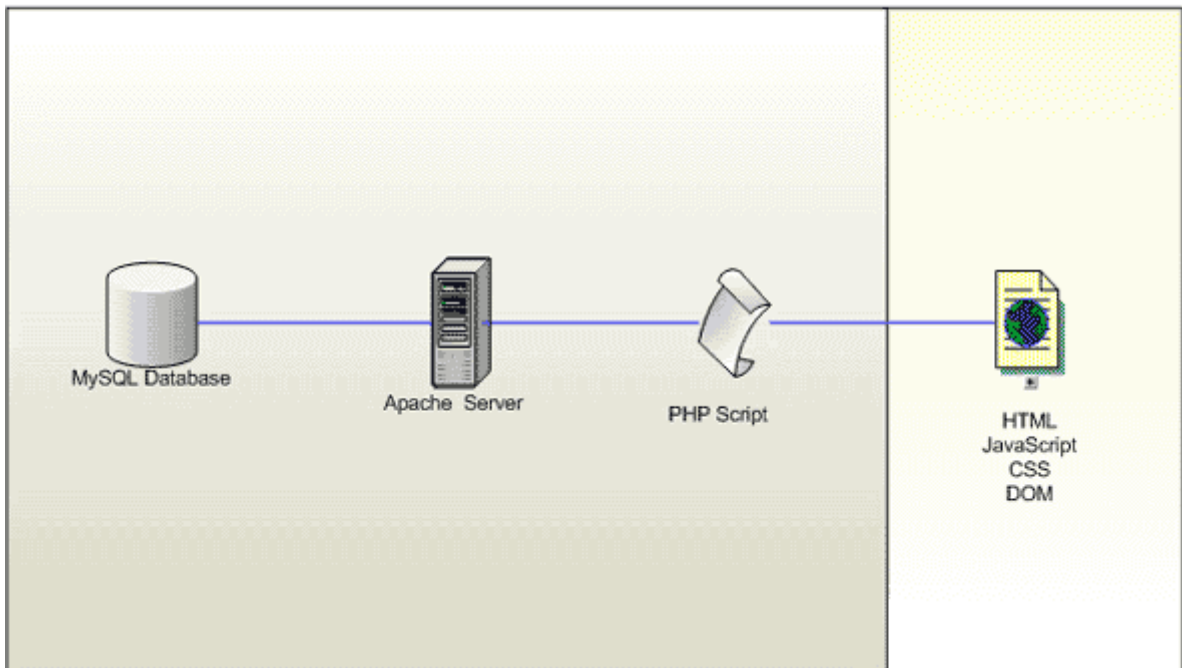
8.2.1. Funcionament de la Web 2.0

La Web 2.0 és possible gràcies a la combinació de diverses tecnologies a on les pàgines web antigues passen a convertir-se en aplicacions que són executades remotament als servidors des d'un navegador dels usuaris. Els llocs 2.0 o aplicacions web 2.0 estan composades per un nucli format per poques o milers de línies de codi escrites en un llenguatge específic com **Phyton**, **ASP.NET**, **JSP**, **RubyonRails** i com no el popular **PHP** que es utilitza en llocs com Facebook o Wikipedia, aquest codi conté les instruccions de com han de funcionar els llocs 2.0 i és executat del costat del servidor.

Quan un usuari entra a un lloc 2.0 per a registrar-se, afegir contingut, buscar, etc., s'executen una sèrie d'instruccions que comuniquen al lloc amb un altre element clau en la web 2.0 "les Bases de Dades" també localitzades als servidors, aquestes emmagatzemen i administren una gran quantitat de dades, un exemple popular d'administrador de bases de dades és MySQL.

Per comprendre el procés d'interacció Usuari (navegador) – Codi (servidor) – Base de dades (servidor), posarem un exemple del registre d'un usuari en una aplicació web 2.0 construïda amb una combinació molt estesa actualment PHP (Llenguatge de programació) + MySQL (Base de dades), primer l'usuari accedeix a la interfaz del lloc 2.0 al teclejar l'adreça URL al navegador, per a realitzar el registre l'usuari ha d'omplir els camps que li sol·liciten com nom d'usuari, email, contrasenya, país...etc.

Una vegada l'usuari ha posat les dades el servidor PHP els processa i envia a la base de dades, el nom d'usuari l'emmagatzema en una taula anomenada 'usuari' el mateix farà amb l'email i país, a aquesta aplicació la contrasenya té un processament diferent PHP i fa el següent, llegeix els caràcters introduïts al camp i mitjançant un algorisme el transformarà en una cadena llarga de caràcters per a que no puguin ser llegits directament per persones, aquest procés se'l coneix com **encriptació**, després la cadena de caràcters resultant és emmagatzemada a altre taula anomenada 'contrasenya', totes les dades del usuari es troben relacionats entre ells mateixos amb un ID únic de forma que no hi ha risc de que es mostrin dades d'altres usuaris, per aquest motiu es diuen bases de dades relacionals.



Quan un usuari intenta iniciar sessió al lloc ha d'ingressar el seu usuari i contrasenya, en aquest procés es portaran a terme els següents passos: l'usuari ingressa les seves dades a la aplicació, aquesta determina la seva id per a realitzar els següents processos, els caràcters introduïts com a contrasenya són encriptats de nou amb el mateix algorisme utilitzat al registrar-se, la cadena de caràcters resultants són comparats amb els existents a la taula "contrasenya" i si ambdues cadenes coincideixen llavors se li dona accés al usuari de la compta.

Bàsicament aquest és el principi del funcionament del motor intern dels llocs 2.0 que permet que la web sigui social a on els usuaris són el més important, des de realitzar una cerca, afegir un comentari, pujar una fotografia, escriure un article, interactuar amb amics...etc.

Però els usuaris que ens trobem al costat del navegador estem en contacte amb la capa més externa de l'aplicació, la interfície que en poques paraules és el que veiem i està dissenyada a mida per mostrar les dades del lloc de forma organitzada, lògica, senzilla d'entendre i visualment atractiva, aquesta capa està composta per diferents tecnologies com XHTML, CSS, Javascript, etc..

Als llocs web 2.0 la interfície o disseny del lloc es troba separada del seu funcionament del lloc, pel que es pot modificar el disseny sense canviar les funcionalitats del lloc, és com la pell de l'aplicació web 2.0.

En general aquesta és l'estructura bàsica dels llocs 2.0 incloent els que s'han fet molt famosos com YouTube, Facebook, Twitter, o Wikipedia exceptuant que aquests són molt complexes ja que manegen quantitats massives de dades cada segon de contingut generat pels usuaris.

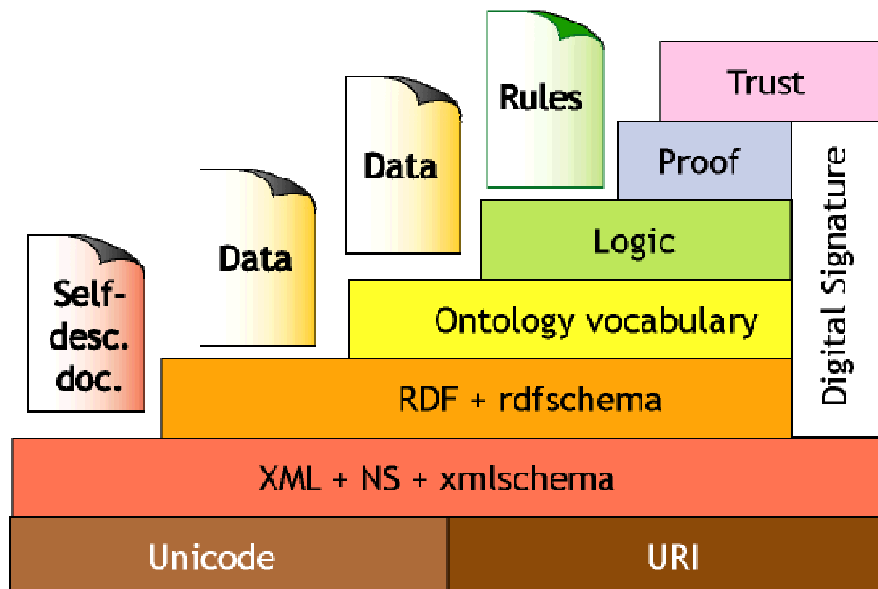
8.3. Web Semàntica

La Web Semàntica vindrà a ser una extensió de la Web actual amb un significat que això és un espai a on la informació tindrà un significat ben definit, de forma que pugui ser interpretada tant per agents humans com per a agents computats.

A l'any 2000 Berners-Lee va oferir una conferència al marc del W3C a on va proposar: "La nova informació ha de ser reunida de forma que un cercador pugui comprendre, en lloc de posar-la només en una llista". La Web Semàntica seria una xarxa de documents "més intel·ligents" que permetin cerques més intel·ligents. La idea seria augmentar la intel·ligència dels continguts de les pàgines web dotant-les de contingut semàntic.

La Web actual posseeix una gran capacitat per emmagatzemar dades i poder llegir i visualitzar els continguts, però no és capaç de pensar ni d'entendre tot el que conte. Es precisa una nova Web (la semàntica) que farà possible no només emmagatzemar dades, sinó entendre e interpretar el sentit d'aquesta informació. D'aquesta forma, Berners-Lee presenta la nova arquitectura en que es basarà la Web Semàntica, no entesa com una nova Web sinó com una extensió de la Web existent.

Segons Berners-Lee, l'arquitectura de la Web Semàntica es podria representar de la següent forma:



El desenvolupament de la Web Semàntica requereix la utilització d'altres llenguatges com el llenguatge estructurat XML (Extensible Markup Language) i el llenguatge RDF (Resource Description Framework) que pugui dotar a cada pàgina, a cada arxiu i a cada recurs o contingut de la xarxa d'una lògica i un significat, i que permetin als ordinadors conèixer el significat de la informació

que fan servir amb la finalitat de que aquesta informació pugui no només ser presentada en pantalla, sinó que també pugui ser integrada i reutilitzada.

XML ha aconseguit convertir-se avui en un llenguatge estàndard. Es tracta d'un subconjunt del complex i sofisticat llenguatge SGML que aporta dades estructurats a la Web i que s'ha convertit en la infraestructura preferida per al intercanvi de dades.

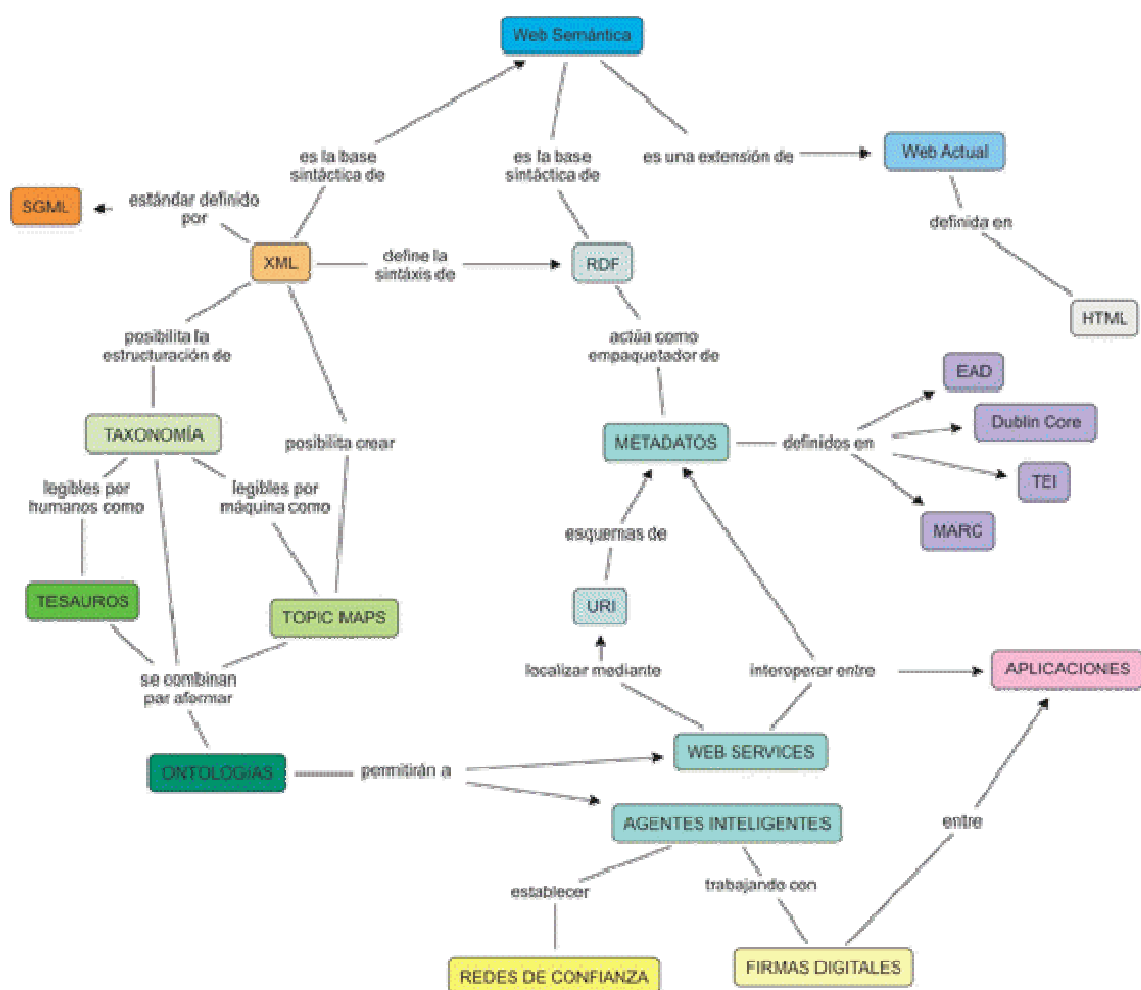
A més les pàgines XML puguin ubicar metadades, esquemes XML i esquemes RDF, que aporten un mecanisme per a que els programes puguin interpretar i comprendre documents amb un vocabulari descriptiu.

Per a poder explotar la Web Semàntica, es necessiten llenguatges semàntics més potents, això és, llenguatges de marcat capaços de representar el coneixement basant-se en la utilització de metadades i ontologies. Utilitzant anotacions RDF i RDF Schema es poden presentar algunes facetes dels conceptes d'un domini del coneixement i es pugui mitjançant relacions taxonòmiques crear una jerarquia de conceptes. Però es precisen llenguatges de marcat (basats en RDF) amb una major expressivitat i capacitat de raonament per a representar els coneixements que continguin les ontologies. A més, aquests llenguatges han de ser estandarditzats i formalitzats per a que la seva utilització sigui universal, reutilitzable i compartit a la Web.

Es necessita un llenguatge comú basat en web, amb suficient capacitat expressiva i de raonament per a representar la semàntica de les ontologies. D'aquesta forma la utilització de llenguatges tals com OWL són un pas més en la consecució de la Web Semàntica.

Es necessari doncs, crear una ontologia o biblioteca de vocabularis descriptius/semàntics, definits en format RDF i ubicats en la Web per a determinar el significat contextual d'una paraula pel mig de la consulta a la ontologia apropiada. D'aquesta forma, agents intel·ligents i programes autònoms podrien buscar la Web de forma automàtica i localitzar, exclusivament les pàgines que es refereixen a la paraula buscada amb el significat i concepte precís amb el que interpretem el terme. Per tant, per a potenciar l'ús d'ontologies a la Web, es necessiten aplicacions específiques de cerca d'ontologies que indiquen als usuaris les ontologies existents i les seves característiques per a utilitzar-les al seu sistema.

L'objectiu de la Web Semàntica és que la Web passa de ser una col·lecció de documents a convertir-se en una base de coneixement:



8.3.1. Funcionament de la Web Semàntica

La forma en la que es processarà aquesta informació no només serà en termes d'entrada i sortida de paràmetres sinó en termes de la seva Semàntica. La Web Semàntica com a infraestructura basada en metadades aporta un camí per a raonar en la Web, estenent així les seves capacitats.

No es tracta d'una intel·ligència artificial màgica que permeti a les màquines entendre les paraules dels usuaris, és només la habilitat d'una màquina per a resoldre problemes ben definits amb operacions ben definides que es portaran a terme mitjançant dades existents ben definits.

Per a obtenir aquesta adequada definició de les dades, la Web Semàntica utilitza RDF i OWL, dos estàndards que ajuden a convertir la Web en una infraestructura global en la que es possible compartir, i reutilitzar dades i documents entre diferents tipus d'usuaris.

- RDF proporciona informació descriptiva simple dels recursos que es troben a la Web i que s'utilitza, per exemple, en catàlegs de llibres, directoris, col·leccions personals de música, fotos, esdeveniments...etc.
- OWL és un mecanisme per a desenvolupar temes o vocabularis específics en els que associar aquells recursos. El que fa OWL es proporcionar un llenguatge per a definir ontologies estructurades que puguin ser utilitzades mitjançant diferents sistemes. Les ontologies, que s'encarregaran de definir els termes utilitzats per a descriure i representar una àrea de coneixement, són utilitzades per als usuaris, les bases de dades, i les aplicacions que necessiten compartir informació específica, és a dir, en un camp determinat com pot ser el de les finances, medicina, esport, etc. Les ontologies inclouen definicions de conceptes bàsics en un camp determinat i la relació entre ells.

8.4. Diferències entre els diferents tipus de Webs i conclusions

Les principals característiques que podrien destacar-se de cadascuna de les Webs més concretament de la 3.0 i la 2.0 són:

Actualment la Web 2.0

- Un software d'escriptori transformat en una aplicació web
- Web que respecta els estàndards del XHTML
- Separació del contingut del disseny amb utilització de fulles d'estil
- Permet la sindicació de continguts
- S'utilitza Ajax (Asincronical Javascript and XML)
- Implementació de Flash, Flex o Lazlo
- Utilització de Ruby on Rails per a programar pàgines dinàmiques
- Utilització de xarxes socials al fer servir usuaris i comunitats
- Se li dona control total als usuaris en la manipulació de la seva informació
- Proveir Apis o XML perquè les aplicacions puguin ser manipulades per altres
- Facilitar el posicionament amb URL senzills.

El que serà la Web 3.0

- Aplicació Web amb molt AJAX
- Podran treballar totes juntes
- Seran relativament petites
- Gestionaran dades que estaran "en un núvol"
- Podran ser executades a qualsevol dispositiu (PC, telèfon mòbil, etc)
- Seran molt ràpides i molt personalitzades
- Es podran distribuir viralment (correu electrònic, xarxes socials, serveis de missatgeries, etc)

Si observem detingudament ambdós llistats es conclou que la Web 3.0 serà una revolució a Internet, no només serà el futur en el que a gestió de contingut es refereix, també aportarà una nova forma d'utilitzar la Web traient-la dels ordinadors i els mòbils i portant-la a quasi qualsevol objecte quotidià.

9. TECNOLOGIES DE LA WEB SEMÀNTICA

Com ja s'ha esmentat abans el XML és un format en el que la seva funció principal és descriure dades i no mostrar-los com fa el HTML.

Les tecnologies XML són un conjunt de mòduls que ofereixen serveis útils a les demandes més freqüents per part dels usuaris.

XML serveix per estructurar, emmagatzemar e intercanviar informació.

Entre les tecnologies disponibles es poden destacar:

→ XSL: Llenguatge extensible de Full d'estil, el seu objectiu és mostrar com deuria estar estructurat el contingut, com deuria ser dissenyat el contingut d'origen i com deuria ser paginat en un medi de presentació com pot ser una avantatge d'un navegador Web o un dispositiu mòbil, un conjunt de pàgines d'un catàleg, informe o llibre.

És capaç de transformar, ordenar i filtrar dades XML, i donar-li format basant-ho en els seus valors.

→ XPath: Llenguatge de rutes XML, és un llenguatge per accedir a parts d'un document XML. Identifica parts d'un document XML concret com poden ser els seus atributs, elements...

→ XLink: Llenguatge d'enllaç XML, és un llenguatge que permet inserir elements en documents XML per crear enllaços entre recursos XML. Descriu un camí estàndard per afegir hiperenllaços en un fitxer XML. Permet crear vincles bidireccionals, el que implica la possibilitat de moure's en dos direccions.

→ XPointer: Llenguatge de Adreçament XML, és un llenguatge que permet l'accés a l'estructura interna d'un document XML, això és, als seus elements, atributs i contingut. És com una extensió de XPath, s'utilitza per arribar a certes parts d'un document XML. Primer XLink permet establir l'enllaç amb el recurs SML i després XPointer va a un punt específic del document.

→ XQL: Llenguatge de Consulta XML, és un llenguatge que facilita l'extracció de dades des de documents XML. Ofereix la possibilitat de realitzar consultes flexibles per a extraure dades de documents XML en la Web. Pot realitzar consultes en infinitat de tipus de documents com són documents estructurats, col·leccions de documents, bases de dades...

10. XML

10.1. Característiques d'un document XML

Les principals avantatges del XML:

- És extensible, després d'haver dissenyat i posat en producció el XML es pot estendre afegint noves etiquetes de forma que podem continuar utilitzant sense complicació alguna el XML.
- L'analitzador és un component estàndard, no és necessari crear un analitzador específic per a cada versió de XML. D'aquesta manera s'eviten bugs i s'accelera el desenvolupament d'aplicacions.
- Si una altra persona decideix utilitzar un document XML és senzill d'entendre la seva estructura i processar-la. Millora la seva compatibilitat entre aplicacions. Podem comunicar aplicacions de distintes plataformes, sense que importi l'origen de les dades, és a dir crear una base de dades amb Postgres i comunicar-la amb una altra aplicació Windows.
- Transformem dades en informació en informació, se li afegeix un significat concret i els hi associem un context, tenim flexibilitat per estructurar documents.

10.2. Format d'un document XML

XML busca donar una solució al problema d'expressar informació estructurada de la manera més abstracte i reutilitzable possible.

Com la informació sigui estructurada vol dir que es compona de parts ben definides, i que aquestes parts es componen d'altres parts.

Una etiqueta consisteix en una marca feta en el document, que senyala una porció d'aquest com un element, un tros de informació amb sentit clar i definit. Les etiquetes tenen la forma **<nombre>**, a on "nombre" és el nom del element que està assenyalant.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail> Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```


Un document ben format seria:

- Els documents han de seguir una estructura estrictament jeràrquica amb el que respecta les etiquetes que delimiten els seus elements. Les etiquetes han d'estar correctament niades.
- Només poden tenir un element inicial.
- Els valors atribuïts en XML sempre han d'estar tancats entre cometes simples o dobles
- El XML és sensible a majúscules i minúscules. Existeixen un conjunt de caràcters anomenats espais en blanc que el processador XML tracten de forma diferent al XML.
- És necessari assignar noms a les estructures, tipus d'elements, entitats, elements particulars, etc. Al XML els noms tenen alguna característica en comú.
- Les construccions com etiquetes, referències d'entitat, i declaracions s'anomenen marques; són parts del document que el processador XML espera entendre.

Parts d'un document XML

- Pròleg:

No és obligatori, però els documents XML poden començar amb unes línies que descriuen la versió XML, el tipus de document i altres coses.

El pròleg conté:

- Una declaració XML, sentència que declara al document com un document XML
- Una declaració de tipus de document. Enllaça el document amb el seu DTD
- Un o més comentaris e instruccions de processament.

- Cos:

El cos no és opcional en un document XML, el cos ha de contenir només un element arrel, característica indispensable també perquè el document estigui ben format. És necessària l'adquisició de dades pel seu bon funcionament

- Elements:

Poden tindre contingut (més elements, caràcters o els dos) o be ser elements buits.

- Atributs:

Els elements poden tindre atributs, que són una manera de incorporar característiques o propietats als elements d'un document. Han d'anar entre cometes.

- Entitats predefinides:

Entitats per a representar caràcters especials per que d'aquesta forma no siguin interpretats com a marcat al processador XML.

- Seccions CDATA:

En una construcció XML per a especificar dades utilitzant qualsevol caràcter sense que s'interpreti com marcat XML. Permet que caràcters especials no trenquin l'estructura.

- Comentaris:

Són de caire informatiu pel programador que seran ignorats per al processador.

10.3. Tecnologies del XML i com funcionen

Entre les tecnologies XML disponibles es poden destacar:

- **XSL**: Llenguatge Extensible de Fulles d'Estil, el seu objectiu principal es mostrar com hauríem d'estar estructurat el contingut, com hauria d'estar dissenyat el contingut d'origen i com hauria de ser paginat en un medi de presentació com pot ser una finestra d'un navegador Web o un dispositiu mòbil, o un conjunt de pàgines d'un catàleg, informe o llibre.

Funciona com un llenguatge avançat per crear fulles d'estils. És capaç de transformar, ordenar i filtrar dades XML, i donar-los format basant-ho als seus valors.

- **XPath**: Llenguatge de Rutes XML, és un llenguatge per accedir a parts d'un document XML.

Identifica parts d'un document XML concret, com poden ser els seus atributs, elements...etc.

- **XLink**: Llenguatge d'enllaç XML, és un llenguatge que permet inserir elements a documents XML per crear enllaços entre recursos XML.

El seu funcionament és descrivint un camí estàndard per afegir hiperenllaços a un arxiu XML. És a dir, és un mecanisme de vinculació a altres documents XML. Funciona similar a un enllaç a una pàgina Web, és a dir, funciona com ho faria ``, només que "a href" és un enllaç unidireccional. XLink permet crear vincles bidireccionals, el que implica la possibilitat de moure's en dues direccions. Això facilita l'obtenció de informació remota com recursos en lloc de simplement com pàgines Web.

- **XPointer**: Llenguatge de Adreçament XML, és un llenguatge que permet l'accés a l'estructura interna d'un document XML, això és als seus elements, atributs i contingut.

Funciona com una sintaxis que apunta a parts d'un document XML, és com una extensió de XPath. S'utilitza per a arribar a parts d'un document XML. Primer, XLink permet establir l'enllaç amb el recurs XML i després és XPointer el que va a un punt específic del document. El seu funcionament és molt semblant al dels identificadors de fragments a un document HTML ja que s'afegeix al final d'una URL i després el que fa es trobar el lloc especificat al document XML. Al ser XPointer una extensió de XPath, XPointer té totes les avantatges de XPath i a més permet establir un rang al document XML, és a dir, amb XPointer és possible establir un punt final i un punt d'inici, el que inclou tots els elements XML entre aquests dos punts.

- **XQL**: Llenguatge de Consulta XML, és un llenguatge que facilita l'extracció de dades des de documents XML. Ofereix la possibilitat de realitzar consultes flexibles per a extraure dades de documents XML a la Web.

El seu funcionament és un llenguatge de consultes, es basa en operadors de cerca d'un model de dades per a documents XML que pot realitzar consultes en infinitat de tipus de documents com són documents estructurats, col·leccions de documents, bases de dades, estructures DOM, catàlegs...etc.

11. RDF

11.1. Característiques del RDF

Va ser creat a 1998 i recomanat per W3C a 1999.

És l'acrònim de Resource Description Framework i és un llenguatge per a la representació de la informació dels recursos en la web (autor d'una pàgina web, llicència, etc), particularment dirigit per a la representació de les metadades. És a dir, defineix la sintaxis i models de dades per a la representació semàntica de les dades.

RDF es basa en els estàndards de URIs i Unicode a més de que es pot representar en XML (pel que es considera com una de les seves aplicacions)

RDF ens permetrà escriure "metainformació" per qualssevol tipus de dades i treballar amb ells. Podem descriure les propietats dels documents i de qualssevol recurs. A més ens permetrà la interoperabilitat entre diferents aplicacions sense pèrdua del significat de les dades.

En resum, RDF ens dona la forma de la representació de informació a prop dels recursos que puguin ser identificats en la web.

11.2. Format d'un document RDF

Aquest model ens obliga a descriure els recursos en termes de propietats i valors que ens força a crear enunciats simples dels recursos dels quals consten de tres elements: subjecte, propietat i valor de la propietat.

- Subjecte: és del que estem parlant
- Propietat: propietat del objecte de la qual estem parlant, per poder identificar-la ens podem preguntar "Què volem saber del subjecte?"
- Valor (Objecte): el valor que tindrà el subjecte de la propietat en concret de la qual parlem.

Per exemple del enunciat “La poma és de color vermell” els diferents elements són:

- Subjecte: Poma
- Propietat: Color
- Valor: Vermell

En el cas de que tot no es pugui expressar en un sol enunciat, pel que es requereix que es descrigui cada una de les propietats del objecte en termes d'altres enunciats.

Requereix especificar per separat cada un dels components en diferents enunciats.

12. RDF Schema

Es pot trobar identificat com RDFS o RDF Schema o Esquema RDF que és una extensió semàntica de RDF.

Un llenguatge primitiu d'ontologies que proporciona els elements bàsics per la descripció de vocabularis.

La primera versió va ser publicada en abril 1998 per la W3C, la versió actual de la recomanació va ser publicada al febrer de 2004 també per la W3C.

Existeixen actualment altres llenguatges d'ontologies més potents com pot ser OWL.

12.1. Característiques principals

- Un arxiu RDFS és un arxiu RDF, és a dir, es tracta d'un arxiu amb la mateixa sintaxis i la mateixa estructura que la que s'utilitza en RDF. La sintaxis està basada en XML.
- És extensible, cada desenvolupador pot estendre l'esquema RDF de forma independent.

12.2. Classes RDFS

12.2.1. Bàsiques

- **rdfs: Class** permet declarar recursos com classes per altres recursos. La definició de rdfs: Class és recursiva, és també la rdfs: Class de qualssevol rdfs:Class
- **rdfs: Resource** és la classe a la que pertanyen tots els recursos
- **rdfs: Literal** és la classe de tots els valors literals, cadenes i enters.
- **Rdfs: Datatype** és la classe que abasta els tipus de dades definits en el model RDF.

12.2.2. Defineixen relacions

- **rdfs: subclassOf** és una instància de rdf:Property que permet definir jerarquies. Relaciona una classes amb les seves superclasses.
- **Rdfs: subPropertyOf** és una instància de rdf:Property que permet definir jerarquies de propietats.

12.2.3. Restriccions de Propietats

- **rdfs: domain** és una instància de rdf:Property que especifica el domini d'una propietat P, rdfs:property. Això és, la classes dels recursos que apareixen com subjectes en les triplets on P és predicat.
- **Rdfs: range** és una instància de rdf:Property que especifica el rang d'una propietat P, rdfs:range. Això és, la classe dels recursos que apareixen com a objectes en les triplets a on P és predicat.

12.2.4. Altres

- **rdfs:ConstraintResource**, és la classe que agrupa a totes les restriccions.
- **rdfs:ConstraintProperties**, és la classe que conté totes les propietats que defineixen restriccions. Les seves instàncies són rdfs:range i rdfs:domain
- **rdfs:seeAlso**, és una instància de rdf:Property que relaciona un recurs amb altre que proporciona informació addicional sobre el primer
- **rdfs:isDefinedBy**, és una instància rdf:Property per a relacionar un recurs als llocs a on es troba el recurs subjecte.
- **rdfs:label**, és una instància de rdf:Property que s'utilitza per a proporcionar un versió clarament comprensible del nom d'un recurs.
- **rdfs:comment**, és una instància de rdf:Property que s'utilitza per a proporcionar una descripció d'un recurs.

13. OWL

13.1. Característiques de OWL

OWL està pensat per a ser utilitzat quan la informació continguda als documents necessita ser processada per les aplicacions, al contrari que en les situacions a on el contingut només necessita ser presentat als humans.

OWL pot ser utilitzat per a representar explícitament el significat de termes en vocabularis i relacions entre aquestes termes.

Aquesta representació de termes i les seves interrelacions s'anomena Ontologia.

OWL té una major capacitat per a expressar significat i semàntica de XML, RDF i RDF-S, i d'aquesta forma, OWL va més allà d'aquesta llenguatges en la seva capacitat per a representar contingut interpretable per un ordinador a la Web.

OWL és una revisió del Llenguatge d'ontologies Web DAML+OIL incorporant lliçons apreses a partir del disseny i aplicació de DAML+OIL

13.2. Per què OWL?

La Web Semàntica està basada en la capacitat de XML per a definir esquemes d'etiquetes a mida i a l'aproximació flexible de RDF per a representar dades. El primer nivell requerit per sobre de RDF per la Web Semàntica és un llenguatge d'ontologies que pot descriure formalment el significat de la terminologia utilitzada en els documents Web.

OWL ha estat dissenyat per cobrir aquesta necessitat d'un llenguatge d'ontologies Web.

OWL forma part d'un conjunt creixent de recomanacions del W3C relacionades amb la Web Semàntica.

- **XML** proporciona una sintaxis superficial per a documents estructurats, però no imposa restriccions semàntiques al significat d'aquests documents
- **XML Schema** és un llenguatge que s'utilitza per a restringir l'estructura dels documents XML, a més de per ampliar XML amb tipus de dades.
- **RDF** és un model de dades per a objectes ("recursos") i relacions entre ells, proporcionant una semàntica simple per aquest. Aquest tipus de model de dades pot ser representat en una sintaxis XML.
- **RDF Schema** és un vocabulari utilitzat per a descriure propietats i classes de recursos RDF, amb una semàntica per la generalització i jerarquitització tant de propietats com de claus.
- **OWL** afegeix més vocabulari per a descriure propietats i classes: entre altres, relacions entre classes, cardinalitat, igualtat, més tipus de propietats, característiques de propietats i classes enumerades.

13.3. Els 3 subllenguatges de OWL

OWL proporciona tres llenguatges, cada un amb nivell d'expressivitat més gran que l'anterior, dissenyats per a ser utilitzats per comunitats específiques de desenvolupadors i usuaris.

- **OWL Lite**, dissenyat per aquells usuaris que necessiten principalment una classificació jeràrquica i restriccions simples. Per

exemple, a la vegada que admet restriccions de cardinalitat, només permet establir valors cardinals de 0 o 1. Proporciona una ruta ràpida de migració per altres taxonomies. OWL Lite té també una menor complexitat formal que OWL DL.

- **OWL DL** està dissenyat per a aquells usuaris que vulguin ser la màxima expressivitat conservant completiu computacional, i resolubilitat. OWL DL inclou totes les construccions del llenguatge OWL però només poden ser utilitzats sota certes restriccions.
- **OWL FULL** està dirigit a usuari que vulguin màxima expressivitat i llibertat sintàctica de RDF sense garanties computacionals. Per exemple, en OWL Full una classe pot ser considerada simultàniament com una col·lecció de classes individuals i com una classe individual pròpiament dita.

Cada un d'aquests subllenguatges és una extensió del seu predecessor més simple, respecte al que pot ser expressat legalment i a la validació de les seves conclusions. El següent grup de relacions es manté, però les relacions inverses no es mantenen:

- Cada ontologia legal de OWL Lite és una ontologia legal de OWL DL
- Cada ontologia legal de OWL DL és una ontologia legal de OWL Full
- Cada conclusió vàlida de OWL Lite és una conclusió vàlida de OWL DL
- Cada conclusió vàlida de OWL DL és una conclusió vàlida de OWL Full

13.4. Escollir un sub-llenguatge o un altre

Els desenvolupadors d'ontologies que adopten OWL haurien de considerar qual és el subllenguatge que millor s'adapta a les seves necessitats.

L'elecció entre OWL Lite i OWL DL depèn de les necessitats dels usuaris sobre l'expressivitat de les construccions, proporcionant OWL DL les més expressives.

L'elecció entre OWL DL i OWL Full depèn principalment de les necessitats dels usuaris sobre els recursos de metamodelat del esquema RDF.

Quan s'utilitza OWL Full en comparació amb OWL DL, el suport al raonament és menys fàcil de predir, ja que no existeixen en aquest moment implementacions completes de OWL Full.

OWL Full pot ser considerada com una extensió de RDF, la OWL Lite i OWL DL poden ser considerades com extensions d'una visió restringida de RDF. Cada document OWL (Lite, DL, Full) és un document RDF, i cada document RDF és un document de OWL Full, però només alguns documents RDF seran

legalment documents OWL Lite o OWL DL. Per aquest motiu, s'ha de tenir cura quan un usuari vulgui migrar un document de RDF a OWL. Quan es consideri que l'expressivitat de OWL DL o OWL Lite és adequada, han de prendre algunes precaucions per assegurar-se que el document RDF original compleix amb les restriccions addicionals imposades per OWL DL i OWL Lite.

14. ONTOLOGIA

14.1. Què és una ontologia?

Ontologia a informàtica fa referència a la formulació d'un exhaustiu i rigorós esquema conceptual en un o diversos dominis de dades amb la finalitat de facilitar la comunicació i el intercanvi d'informació entre diferents sistemes i entitats.

Un us semàntic del concepte d'ontologia el podríem trobar a la Intel·ligència Artificial i a la representació del coneixement.

Els programes informàtics poden fer servir aquest punt de vista de la ontologia per una diversitat de propòsits, incloent el raonament inductiu, la classificació i una varietat de tècniques de resolució de problemes.

- * El raonament inductiu és una modalitat del raonament no deductiu que consisteix en obtenir conclusions generals a partir de premisses que contenen dades particulars.

14.2. Procés de creació d'una ontologia

Hi han diferents metodologies per crear una ontologia.

- *Diligence*: Metodologia de desenvolupament d'ontologies basada en la col·laboració de múltiples participants de forma que la creació d'una ontologia es tracta com un procés social, distribuït i poc controlat. Tota ontologia ha d'evolucionar amb el temps de forma que permeti introduir noves etapes en elles.
- *Competency Questions*: També se li diu metodologia de les "Preguntes Rellevants" o "Preguntes de verificació", és una de les metodologies més senzilles d'aplicar. Consisteix en determinar el domini i abast de la ontologia mitjançant la llista de preguntes que el sistema deuria ser capaç de contestar. Les respostes a aquestes preguntes suggereixen el que podrien ser les instàncies de la ontologia, a partir de les quals es deduiran les classes de la mateixa. Al mateix temps, aquestes preguntes serviran com factor decisiu a l'hora d'avaluar la pròpia ontologia, permetin comprovar si s'ha representat la suficient informació com per a poder respondre a qüestions especialment rellevants.
- *Methontology*: Proposa un cicle de vida de construcció de la ontologia basat en prototips evolutius per que això permet afegir, canviar i renovar termes en cada nova versió.

Els passos essencials són:

- *Especificació*: consisteix en delimitar els objectius de la seva creació, decidir el domini d'actuació de la ontologia, qui la farà servir i per a que, les preguntes a les quals haurà de respondre i quin s'encarregarà del seu manteniment.
- *Conceptualització*: Consisteix en crear un glossari de termes que pertanyen al domini, definir-los i crear una taxonomia
- *Formalització*: Consisteix en convertir el model anterior en un model formal o semi computable. En aquest pas es pot fer servir Protégé
- *Implementació*: Transforma el model formalitzat en un model computable mitjançant un llenguatge per a construcció d'ontologies. Es pot fer servir Protégé
- *Manteniment*: Aquest pas pot anar des del esborrat d'instàncies ja obsoletes o la incorporació de noves instàncies que es van produint amb el temps.

- *On-To-Knowledge*:

Els passos essencials són:

- *Estudi de viabilitat*: Prèviament al desenvolupament de la ontologia i base pel següent procés.
- *Inici*: Procés de delimitació del domini i objectiu de la ontologia, extracció de les fonts de coneixement juntament amb la descripció dels seus usuaris i aplicacions futures.
- *Refinament*: Producció d'una ontologia orientada a les seves aplicacions, segons les especificacions extremes del procés anterior.
- *Avaluació*: Proba de la utilitat de la ontologia i del entorn de software associat a ella a l'aplicació per la qual va ser dissenyada.
- *Manteniment*: Decisió sobre el responsable d'aquesta feina i dels processos possibles en ella.

14.3. Llenguatges d'ontologies

Per crear ontologies hi haurien diferents llenguatges que podríem fer servir:

- **RDF**: és el marc de Descripció de Recursos (Resource Description Framework, RDF) és un framework per a metadades en World Wide Web (WWW) desenvolupat per World Wide Web Consortium (W3C)

Aquest model es basa en la idea de convertir les declaracions en expressions amb la forma Subjecte-Predicat-Objecte.

- Subjecte és el recurs, és a dir allò que s'està descrivint
- Predicat la propietat o relació que es desitja establir del recurs
- Objecte és el valor de la propietat o l'altre recurs amb el que s'estableix la relació.

- **DAML**: (DARPA Agent Markup Language) era el nom d'un programa de EE.UU per a la Defense Advanced Research Projects Agency, començada al 1999 per James Hendler, controlada més tard per Murray Burke i Mark Greaves. Estava enfocat per a la creació de representacions llegibles per màquines per al web.

S'utilitza a la capa de descripció de la pila tecnològica de Web Services.

- OIL

- **OWL:** Acrònim de Ontology Web Language, un llenguatge marcat per a publicar i compartir dades utilitzant ontologies en la WWW. Té com a objectiu facilitar un model de marcat construït sobre RDF i codificat en XML.

Actualment OWL té tres variants:

- OWL Lite: està construït de forma que tota sentència pugui ser resolta en temps finit. Aquesta es basa en lògica menys expressiva.
- OWL DL: es basa en la lògica descriptiva
- OWL Full

14.4. Propietats en una Ontologia

- **DataProperties:** En aquest apartat es definiran els diferents atributs que tindrà cada classe. Es definirà quin tipus de dades és, i en el cas que tinguin valors predeterminats també es podran assignar a les DataProperties.

- **ObjectsProperties:** També es podran definir les propietats que tindran els objectes, que en realitat seran les diferents relacions que existeixen entre ells.

15. Anàlisi d'ontologies ja existents

Hi ha una ontologia ja existent que seria bastant útil pel nostre treball.

La ontologia en qüestió és una ontologia de Zaragoza:

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

En aquesta ontologia ens explica llocs que podríem visitar de la ciutat de Zaragoza, a on anar a menjar, llocs per allotjar-se, inclòs es poden calcular rutes de visites a la ciutat tenint en compte el lloc de l'allotjament.

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

Aquí ens podríem centrar més en la informació que dona de com organitzar les dades i el tipus de dades que es farien servir.

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

Tracta d'un cas pràctic de com transformar unes dades d'informació turística en metadades semàntics per poder fer una Ontologia.

16. Bases de dades natives XML

Les bases de dades són una part fonamental de totes les organitzacions, doncs en elles s'emmagatzema informació crucial pel bon funcionament. A més sabem que XML és el present i futur de l'administració de dades, doncs aquest llenguatge ha permès trencar barreres i crear una forma estàndard de processar la informació.

Aquest tipus de bases de dades són completament distintes a les relacionals, les quals a la actualitat tenen suport per XML, però encara segueixen emmagatzemen tota la informació de forma relacional, és a dir de forma tabular (taules, registres i columnes) o en el cas contrari emmagatzemen tot el document en format Binary Large Object (BLOB), però la principal característica que tenen aquestes base de dades és la capacitat d'obtenir els resultats de les consultes en format XML, és per això que aquestes base de dades pertanyen a la categoria de "XML-enabled database"

En què consisteixen les Bases de Dades natives en XML?

Les bases de dades natives defineixen un model lògic per al document XML, a més emmagatzema i recupera document de la mateix forma que els XML. Aquest model ha d'incloure atributs com PACTADA, i documents en ordre. Exemples d'aquest models son XPath, XML Infoset i models que impliquen DOM i SAX 1.0

Bases de dades centrades en documents

Totes les bases de dades relacions són centrades en les Dades, doncs el que emmagatzemen als seus camps són dades simples més conegudes com dades atòmiques. Una base de dades nativa en XML, no posseeix camps, ni emmagatzema dades atòmiques, el que emmagatzema són documents XML, per tant aquest tipus de bases de dades se'ls hi denomina bases de dades centrades en documents.

16.1. Característiques

16.1.1. Processament de dades

El processament de dades en aquest tipus de bases de dades semblaria ser molt beneficiós, però realment no és així, degut al format jeràrquic en el que està emmagatzemada la informació.

Moltes bases de dades necessiten que l'usuari recuperi tot el document XML, l'actualitzi amb el XML API de la seva preferència i posteriorment torni a emmagatzemar el document al transpositor. Això es deu a que encara no existeix un llenguatge estàndard que permeti l'actualització, inserció o eliminació d'elements d'un document XML.

Existeix un llenguatge que permet realitzar actualitzacions a un document XML però encara no és un estàndard i molts gestors d'aquest tipus de bases de dades no ho suporten, aquest llenguatge és Xopada.

16.1.2. Emmagatzemant

Per deducció lògica, una base de dades nativa en XML emmagatzema la informació en format XML, però aquest és només una deducció lògica, doncs aquest tipus de bases de dades tenen depositaris en un format "tipus XML", com pot ser DOM o Infosa.

En aquest mateix "depositari" (paquet d'arxius) s'emmagatzemen els índex que es generen per cada document XML emmagatzemat.

16.1.3. Cerques

Aquest tipus de bases de dades no utilitza SQL com llenguatge de consulta. En lloc d'allò utilitzen Xpath. Algunes bases de dades permeten seleccionar els elements que permeten tindre índex mentrestant que altres bases de dades indexen tot el contingut del document.

El problema que tenen les cerques en aquest tipus de bases de dades es que no permeten realitzar cerques molt complicades, com per exemple ordenació i cross join, degut a que Xpath no va ser creada realment per cerques en bases de dades, sinó simplement per cerques en un sol document.

Moltes bases de dades permeten realitzar cerques utilitzant la tecnologia Full-Text Search, d'aquesta manera es pugui agilitzar la cerca de dades en els documents XML.

16.2. Àrees d'aplicació

Només existeix un requisit per a qualsevol aplicació a on es requereix utilitzar NXD: l'aplicació ha d'utilitzar XML.

Després d'això no existeix cap restricció o regla per no utilitzar NXD en una aplicació.

En general, qualsevol NXD és una excel·lent eina per emmagatzemar documents orientats a dades, la informació que té una complexa estructuració niada profundament i la informació que està semi-estructurada a la natura. Bàsicament, si la informació es representada per XML un NXD probablement seria una bona solució. Un NXD pot emmagatzemar qualsevol tipus d'informació XML, però probablement no seria la millor reina per un sistema de comptabilitat a on les dades són molt ben definides i rígides.

Algunes àrees potencials d'aplicació podrien ser:

- Portals d'informació corporativa
- Informació de catàlegs
- BD en parts de manufactura
- Informació metge
- BD personalitzades.

16.3. eXist

En els últims anys han aparegut un gran nombre de bases de dades amb suport per a documents XML i algunes d'elles amb suport per a XQuery.

Existeixen principalment dos tipus de bases de dades XML:

1. XML-Enabled: les quals mapegen els documents XML en una base de dades relacional o objecte relacional.
2. Bases de dades natives XML (XNDB): a on s'utilitzen com unitat fonamental d'emmagatzemen d'estructures específiques per a documents XML.

eXist-db és un sistema de gestió de bases de dades lliure i de codi obert que emmagatzema dades XML d'acord a un model de dades XML.

Algunes de les seves característiques són:

- Suport per a diferents llenguatges de consultes XML com XQuery, XPath i XSLT
- Indexació de documents i suport per a l'actualització de les dades i per a multitud de protocols com SOAP, XML-RPC, WebDav i REST i en l'actualitat compleix el estàndard XQuery en un 99'4%

Es poden realitzar diferents instal·lacions:

- Mitjançant un fitxer d'instal·lació (.jar multiplataforma). Al executar el instal·lador (java .jar fichero.jar) sortirà un assistent guiant-te en el procés de instal·lació.
- Mitjançant un fitxer .war per incloure'l en un servidor Cocoon
- Mitjançant el codi font.

16.4. Realitzar consultes: XPath

XPath és un llenguatge que permet construir expressions que recorren i processen un document XML. La idea és semblant a les expressions regulars per seleccionar parts d'un text sense atributs (plain text). XPath va ser creat per la seva utilització al estàndard XSLT, a on s'utilitza per seleccionar i examinar l'estructura del document d'entrada de la transformació.

Tot el processament realitzat amb un fitxer XML està basat en la possibilitat de adreçar o accedir a cada una de les parts que ho componen, de forma que puguem tractar cada un dels elements de forma diferenciada.

El tractament del fitxer XML comença per la localització del mateix per tot el conjunt de documents existents al mon. Per portar a terme aquesta localització de forma unívoca, s'utilitzen els URI (Uniform Resource Identifiers) dels quals els URL (Uniform Resource Locators) són sens dubte els més coneguts.

Una vegada localitzat el document XML, la forma de seleccionar informació en ell és mitjançant la utilització de XPath, que és la abreviatura del que es coneix com XML Path Language. Amb XPath podem seleccionar i fer referència al text, elements, atributs i qualssevol altre informació continguda dintre d'un fitxer XML.

XPath és un llenguatge sofisticat i complexa, però diferents als llenguatges procedurals que fem servir (C, C++, Basic, Java...) però com el mon de XML encara està en estat de desenvolupament no es fàcil trobar eines que incorporin totes les seves funcionalitats i a la vegada és la base sobre la qual s'han especificat noves eines que aprofitar pel tractament de documents XML. Eines tal com XPointer, XLink i XQuery.

XPath serveix per dir com s'ha de processar una fulla d'estil el contingut d'una pàgina XML, però també per poder posar enllaços o carregar en un navegador zones determinades d'una pàgina XML en comptes de tota la pàgina.

16.4.1. Model de dades XPath

Un document XML és processat per un analitzador (o parser) construït un arbre de nodes. Aquest arbre comença amb un element arrel, que es diversifica per tot el llarg del elements que pengen d'ells i acaba en nodes fulla, que contenen només text, comentaris, instruccions de procés o fins i tot, que estan buits i només tenen atributs.

Les formes en que XPath selecciona parts del document XML es basa precisament en la representació arbòria que es genera del document. De fet, els "operadors" dels quals consta aquest llenguatge ens recordaran la terminologia que s'utilitza a l'hora de parlar d'arbres en informàtica: arrel, fill, ancestre, descendent, etc.

Un cas especial de node són els nodes atribut. Un node pot tindre tants atributs com desitgi, i per cada un se li crearà un node atribut. No obstant, aquests nodes atribut NO es consideren com fills seus, sinó com etiquetes afegides al node element.

A continuació es mostra un exemple de com es converteix en arbre un document XML.

Document XML

```
<?xml version="1.0" ?>
<allotjaments>
  <allotjament>
    <Nom_allotjament>Eurostars Grand Marina Hotel</Nom_allotjament>
    <Adresa_allotjament>Moll de Barcelona, s/n</Adresa_allotjament>
    <Codi_postal_allotjament>08039</Codi_postal_allotjament>
    <preu_allotjament>205</preu_allotjament>
    <Categoria_allotjament>5</Categoria_allotjament>
    <Quin_tipus_allotjament>Hotel</Quin_tipus_allotjament>
  </allotjament>
</allotjaments>
```

Arbre generat

```
+---allotjaments
|
+---allotjament
|
|   +---(Nom_allotjament) Eurostars Grand Marina Hotel
|   |
|   +---(Adresa_allotjante) Moll de Barcelona, s/n
|   |
|   +---(Codi_postal_allotjament) 08039
|   |
|   +---(preu_allotjament) 205
|   |
|   +---(Categoria_allotjament) 5
|
+---allotjaments
```

16.4.2. Tipus de nodes

Existeixen diferents tipus de nodes en un arbre generat a partir d'un document XML, a conèixer: arrel, element, atribut, text, comentari i instrucció de processament (respectivament: root, elements, attribute, text, comment i processing instruction)

Node arrel

S'identifica per /. No s'ha de confondre el node arrel amb l'element arrel del document. Així, si el document XML del nostre exemple té per element arrel hotel, aquest serà el primer node que pengi del node arrel del arbre, el qual és: /

Node Element

Qualsevol element d'un document XML es converteix en un node element dintre del arbre. Cada element té el seu node pare. El node pare de qualsevol element és, a la seva vegada, un element, excepte l'element arrel, el qual el seu pare és el node arrel.

Els nodes element tenen a la seva vegada fill, que són: nodes element, nodes text, nodes comentari i nodes d'instruccions de procés.

Els nodes element també tenen propietats com el seu nom, els seus atributs i informació dels "espais de nom" que tenen actiu.

Una propietat interessant dels nodes element es que poden tindre identificadors únics (per això han d'anar acompanyats d'un DTD que especifiqui que aquests atributs tindran valors únics), això permet referenciar als elements d'una forma molt més directa.

Nodes text

Farem referència a tots els caràcters del document que no estan marcats amb alguna etiqueta. Un node text no té fills, és a dir, els diferents caràcters que el formen no es consideren fills seus.

Nodes atribut

Els nodes atribut no són tant fills del node element que els conté com etiquetes afegides a aquest node element. Cada node atribut consta d'un nom, un valor (sempre una cadena) i un possible "espai de noms"

Aquells atributs que tenen per valor el valor per defecte assignat al DTD es tractaran com si el valor se'ls hi hagués assignat al escriure el document XML. Al contrari, no es crea node per atributs no especificats al document XML, i amb la propietat #IMPLIED definida al seu DTD. Tampoc es creen nodes atribut per les definicions dels espais de nom. Tot això és normal si tenim en compte que no es necessari tindre un DTD per processar un document XML.

Nodes comentari i d'instruccions de procés

A l'arbre també es generen nodes per cada node amb comentaris i amb instruccions de procés. Al contingut d'aquests nodes es pot accedir amb la propietat *string-value*.

16.5. Realitzar consultes: XQuery

XQuery és un llenguatge de consulta dissenyat per a consultar col·leccions de dades XML. És semànticament similar a SQL, però no inclou algunes capacitats de programació.

16.5.1. Característiques

- XQuery proporciona els medis per extraure i manipular informació de documents XML, o de qualssevol font de dades que pugui ser representada mitjançant XML, com per exemple Bases de Dades Relacions o documents ofimàtics.
- XQuery utilitza expressions XPath per accedir a determinades parts del document XML. Afegeix a més unes expressions similars a les utilitzades en SQL, conegudes com expressions **FLWOR**. Les expressions FLWOR prenen el seu nom dels 5 tipus de sentències de les que poden estar compostes: **FOR, LET, WHERE, ORDER BY** i **RETURN**.
- També inclou la possibilitat de construir nous documents XML a partir dels resultats de la consulta. Es pot utilitzar una sintaxis similar a XML si l'estructura (elements i atributs) es coneguda amb antelació, o utilitzar expressions de construcció dinàmica de nodes en cas contrari. Tots aquests constructors es defineixen com expressions dintre del llenguatge, i es poden nidar arbitràriament.
- El llenguatge es basa en el model en arbre de la informació continguda en el document XML, que consisteix en set tipus diferents de node: elements, atributs, nodes de text, comentaris, instruccions de processament, espais de noms i nodes de documents.
- El sistema de tipus utilitzar pel llenguatge considera tots els valors com seqüències, sumint-se un valor simple com una seqüència d'un sol element. Els elements d'una seqüència poden ser valors atòmics o nodes. Els valors atòmics poden ser números sencers, cadenes de text, valors booleans, etc. La llista completa dels tipus disponibles està basada en les primitives definides en XML Schema.
- XQuery 1.0 no inclou capacitat d'actualitzar els documents XML. Tampoc pot realitzar cerques textuales. Aquestes dos capacitats estan essent objecte de desenvolupament per la seva possible incorporació en la següent versió del llenguatge.
- XQuery és un llenguatge de programació funcional que consta en la seva totalitat d'expressions. No hi ha sentències, encara quan algunes de les paraules claus utilitzades puguin suggerir un comportament similar al d'una sentència. Per executar una funció, l'expressió dintre del cos de la mateixa s'avalua i es retorna el resultat obtingut. Per exemple, per escriure una funció que dupliqui el valor subministrat com argument escriuríem:

Declare function local:doubling(\$x) { \$x * 2}

Per a escriure una consulta XQuery que retorni el text "Hola Mundo", escriuríem l'expressió:

"Hola, Mundo"

16.5.2. Aplicacions

Una relació d'alguns exemples d'ús de XQuery:

- Extraure informació d'una base de dades per utilitzar-la en un Servei Web
- Generar un resum de la informació emmagatzemada en una base de dades XML
- Realitzar cerques textuais en la web i compilar els resultats de la mateixa
- Seleccionar i transformar dades de XML a XHTML de forma que es puguin publicar a la Web
- Obtenir dades des de diferents fonts amb vistes a ser integrades per l'aplicació
- Dividir un document XML que representa una sèrie de múltiples transaccions en diversos documents XML, un per cada transacció.

17. Aplicació

17.1. Requeriments

Els requeriments per la creació de l'aplicació serien:

- Conèixer en que consistirà l'aplicació a desenvolupar i quin són els requeriments mínims
- Tenir els coneixements de les tecnologies a emprar per desenvolupar-la, així com disposar d'elles i de l'equipament per instal·lar-les.
- Obtenir informació que després l'aplicació ens retornarà, és a dir, obtenir la informació necessària per introduir-la a la base de dades i així poder buscar mitjançant el cercador a desenvolupar.
- Tindrà una planificació que s'haurà de seguir pel bon desenvolupament de l'aplicació.
- Cercar informació per omplir la Base de Dades i així poder realitzar les cerques des de l'aplicació que es desenvoluparà.

17.2. Descripció de l'aplicació

L'aplicació que es desenvoluparà serà un cercador d'informació de cara a una persona que vulgui visitar una ciutat, en aquest cas Barcelona.

Des d'aquest cercador l'usuari podrà buscar tant informació de llocs a on allotjar-se i diferents activitats que podrà realitzar a la ciutat.

La cerca es podrà realitzar en 3 idiomes diferents de forma que encara que no sigui un usuari que conegui els idiomes de la ciutat podrà fer la cerca.

Tant es podrà realitzar aquesta cerca global o d'un tema en concret, així com també obtenir un pressupost aproximat del que suposaria fer la visita.

17.3. Tecnologia a emprar

La tecnologia a emprar en aquesta aplicació serà:

- Protege_4.1.: Per realitzar la ontologia de l'aplicació.
- Dreamweaver: Software que es farà servir per la confecció de la pàgina web des d'on es podran fer les cerques.
- eXist: Per la confecció de la Base de Dades mitjançant el qual es farà la importació de les dades (que estaran en arxius XML), per importar les dades a la Base de Dades es farà creant una col·lecció a eXist i en aquesta es pujaran els arxius XML amb les dades per la nostra aplicació.
- Per realitzar la cerca a la nostra Base de Dades es farà servir el llenguatge Xquery que permetrà realitzar diferents tipus de consulta segons les dades introduïdes.

18. Creació d'una Ontologia pròpia

L'ontologia en qüestió es farà per donar informació a diferents persones i possibles visitants de la ciutat de Barcelona, donant-li informació d'activitats a realitzar a la ciutat, i llocs a on allotjar-se.

Per tant el domini de l'ontologia podríem dir que seria la ciutat de Barcelona i els seus diferents serveis.

L'ontologia hauria de ser capaç de respondre sense cap problema a certes preguntes que donaran informació bàsica per al turista:

- A on m'allotjaré?
- Què puc fer a Barcelona?

Com ja he comentat l'ontologia la utilitzaran els visitants per trobar informació a l'hora de planificar-se la visita.

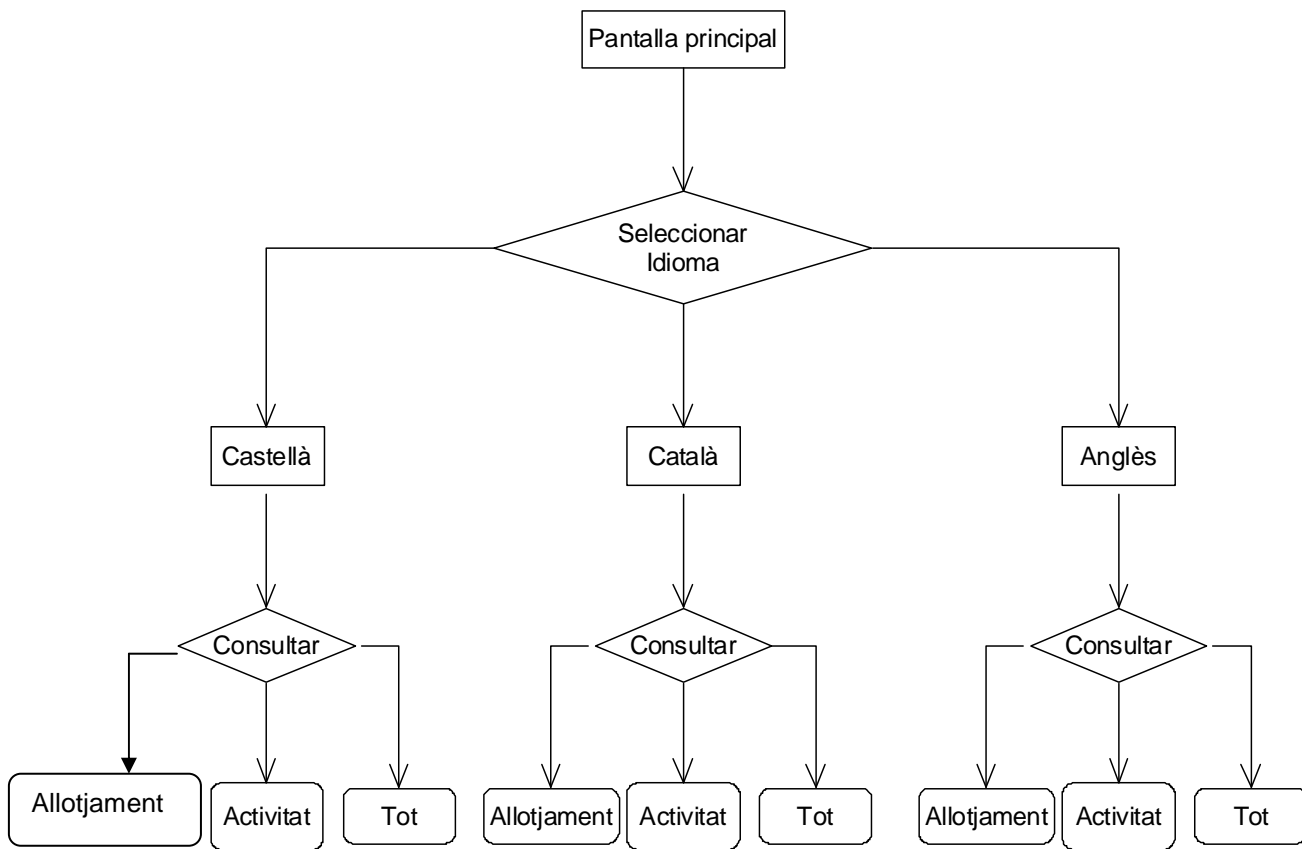
18.1. Classes de la ontologia

- Allotjament
- Activitats
- Cerca
 - Nomes_allotjament
 - Per_tipus_allotjament
 - Per_preu_allotjament
 - Per_categoria_allotjament
 - Per_nom_allotjament
 - Per_tots_atributs_allotjament
 - Nomes_activitat
 - Per_tipus_activitat
 - Per_nom_activitat
 - Per_preu_activitat
 - Per_tots_atributs_activitat
 - Cerca_global
 - Global_tipus_allotjament
 - Global_preu_allotjament
 - Global_categoria_allotjament
 - Global_nom_allotjament
 - Global_tots_allotjament
 - Global_tipus_activitat
 - Global_nom_activitat
 - Global_preu_activitat
 - Global_tots_activitat

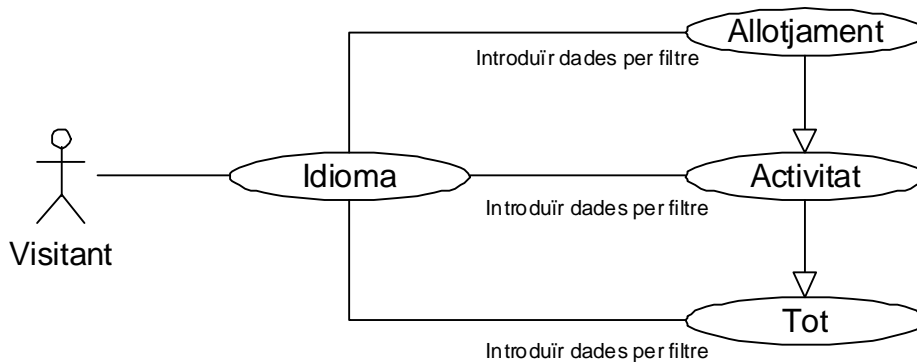
En la ontologia, creada amb Protege, s'especifiquen els diferents atributs de cada classe i les diferents propietats.

* Adjunto a aquest document la ontologia creada.

19. Mapa conceptual



20. Casos d'ús



21. Conclusions

Després d'haver realitzat aquest treball podria concloure:

- He après a realitzar la planificació d'un projecte, tenint en compte diferents aspectes com són els requeriments previs de maquinari, els coneixements que s'han de tenir per realitzar l'anomenada planificació tenint en compte diferents aspectes professionals i/o personals que poden interferir de forma negativa en la realització del projecte.
- He conegut el món de la Web Semàntica, així com els diferents tipus de llenguatges que el formen.
- He conegut què és una ontologia, com fer-la, així com la utilització de Protege que ens ajuda a formar-la.
- He après a crear documents XML que seran el que carregaran les dades a la nostra Base de Dades.
- Encara que ja tenia coneixements de php, gràcies a la realització d'aquest projecte he conegut com es poden fer servir conjuntament.
- M'ha permès aprendre a carregar una base de dades en eXist, així com realitzar diferents consultes a aquestes dades emmagatzemades.

M'ha semblat un tema interessant de realitzar, que encara que al no tindre coneixements previs hi ha hagut moments on semblava que no ho podria aconseguir, crec que he assolit els requeriments i realitzat una aplicació tal i com es demanava.

En definitiva, estic contenta amb el resultat que he aconseguit i tot el que he après al realitzar aquest projecte.

22. Webs d'on extreure informació de l'ontologia

Turisme en general

<http://www.barcelonaturisme.com/>

Subclasse Hotels

<http://barcelona.guia-de-hoteles.com/>

Subclasse Restaurants

<http://www.restaurantesbarcelona.com/>

Subclasse Metro

http://www.tmb.cat/ca_ES/home.jsp

Subclasse Tram

<http://www.trambcn.com/>

Subclasse Renfe

<http://www.renfe.com/>

Subclasse Ferrocarrils

<http://www.fgc.cat/cat/index.asp>

Subclasse Taxi

<http://www.taxis-barcelona.es/>

Subclasse En Bici

<http://www.bicing.cat/home/home.php>

23. Bibliografia

- Web semàntica:
http://es.wikipedia.org/wiki/Web_sem%C3%A1ntica
- Com crear una ontologia
http://es.wikipedia.org/wiki/Ontolog%C3%ADa_%28inform%C3%A1tica%29
<http://enciclopedia.us.es/index.php/Ontolog%C3%ADa>
- EXist
- <http://gim.unex.es/blogs/ljarevalo/2009/02/13/exist-base-de-datos-nativa-xml/XML>
<http://www.w3c.es/divulgacion/guiasbreves/tecnologiasxml>
- RDF
http://es.wikipedia.org/wiki/Resource_Description_Framework
- OWL
<http://es.wikipedia.org/wiki/OWL>
- Diferencias entre la web 1.0, 2.0 y 3.0
<http://web30websemantica.comuf.com/diferenciasweb3-0web2-0web1-0.htm>
- XPath
<http://es.wikipedia.org/wiki/XPath>
- XQuery
<http://es.wikipedia.org/wiki/XQuery>
- XQuery
<http://xmldb-org.sourceforge.net/xupdate/xupdate-wd.html>