Construcció d'una eina automàtica per extreure informació d'un document OpenOffice

Jesús Delgado Pons
Enginyeria en Informàtica

Jordi Ferrer Duran

8 de juny / 08

Fig. 1 - Portada de la revista Novatica, número 184

Dedicatòria i agraïments

Aquest projecte representa la culminació d'una empresa personal que vaig començar ara fa uns anys.

Durant aquest temps he gaudit amb l'estudi de diverses matèries a la UOC i amb l'excel·lent labor educativa del seu professorat, en especial l'atenció i explicacions del meu consultor, Jordi Ferrer Duran, que m'ha sabut adreçar quan ho necessitava.

Per acabar gràcies a tothom que m'ha engrescat a acabar aquesta empresa i en especial gràcies a tu Montse que tant m'has recolzat per a aconseguir-ho.

Resum

Aquest Projecte de Final de Carrera consisteix en estudiar el format dels fitxers OpenOffice i extreure'n certa informació continguda en els documents per satisfer una necessitat plantejada per la recentment constituïda Agencia Catalana de Seguretat, mitjançant una eina de software creada per a tal fi.

Complementàriament i aprofitant l'estudi del format dels fitxers OpenOffice s'estudiaran possibles formes de protegir les dades dins d'un document per tal de conèixer noves vies de frau que puguin ser utilitzades.

Índex de continguts

Índex de figures

Capítol 1 Introducció

1.1 Justificació del PFC i context en el qual es desenvolupa: punt de partida i aportació del PFC

Actualment el tractament d'informació és una peça clau en la rebotiga de les organitzacions, sobretot en les governamentals, en tant que el valor afegit dels seus processos es troben en els documents que generen.

Aquests processos consisteixen principalment en l'emmagatzemament i compartició de documents entre sistemes i ha estat des dels inicis una font de problemes a causa de cada sistema ha usat un format propietari de document que en dificultava el tractament en altres sistemes.

L'any 1999 es va produir un fet molt important en el camp dels document digitals: la constitució d'un projecte obert amb l'objectiu de crear un format de document digital lliure i independent de plataforma. Aquest format anomenat Open Document OASIS Standard (ODF) serà objecte d'estudi en aquest Projecte Final de Carrera, a més del format de marques XML en el qual està fonamentat.

La importància del format ODF radica en el fet de que es tracta d'un estàndard lliure, és a dir, un format del qual tothom en pot fer ús sense haver de pagar regalies. Actualment moltes organitzacions, tant públiques i privades, estan decantant-se per programari lliure i en el cas dels documents estan migrant cap a format Open Document el que suposa tant una obertura cap als ciutadans com una reducció de costs en els processos de tractament d'informació.

D'altra banda per a poder aprofundir en el format Open Document aquest projecte també fa un estudi previ del format XML en el qual es basa, mostrant els motius i avantatges de perquè ha esdevingut un èxit en el àrea de la informació i aplicacions informàtiques.

Finalment es presentarà una eina per al tractament de documents digitals mitjançant el format ODF alhora que mostrarà com treballar amb els processos d'automatització facilitats per aquesta tecnologia.

En un altre ordre de coses aquest projecte pretén ser un estudi profund del format Open Document, mostrant els avantatges del seu ús però també mostrar les amenaces de seguretat relacionades i un punt de partida a nous estudis o desenvolupaments mitjançant aquesta tecnologia. Creiem que els aspectes de tractament d'informació i de seguretat que s'exposaran en aquesta memòria poden ser origen de noves oportunitats que facilitaran l'emmagat-zemament i compartició d'informació mitjançant documents digitals de forma segura.

1.2 Objectius del PFC

Els objectius generals del projecte són:

- L'estudi del format de document de text OpenOffice, concretament el referent a la versió 1.1 de ODF (Open Document Format).
- Estudiar les alternatives existents per llegir i manipular documents digitals.
- Definir documents XML per a guardar informació sobre dades confidencials i els DTD associats.
- Implementar una eina capaç de fer el tractament descrit en el punt anterior sobre un document OpenOffice qualsevol i que generi un document HTML amb la presentació dels resultats prèviament guardats en un document XML.

Més específicament es treballaran els següents conceptes:

- Generació i manipulació d'expressions regulars que serveixin de patró per buscar la informació demanada dins del document.
- Definir el format dels diferents documents XML que cal generar.
- Construir els DTD que validen el diferents fitxer XML generats.
- Mitjançant transformacions XSL generar un document HTML per visualitzar els resultats.

1.3 Enfocament i mètode seguit

Si recordem l'objectiu proposat: l'estudi profund del format de document digital Open Document i els aspectes de seguretat relacionats,creiem oportú dividir aquest projecte en tres grans blocs: un primer d'investigació de totes les tecnologies necessàries, un de desenvolupament d'una eina que modelitzi l'etapa d'investigació i un últim bloc que aporti una reflexió de tot plegat a més de consideracions sobre els aspectes de seguretat per a propers projectes d'investigació.

El primer bloc consistirà en un conjunt de capítols que versaran sobre els format ODF i el format XML en el qual està basat, concloent amb les expressions regulars com a eina per a fer patrons de cerca i un annex al format Open Document on es tractaran les amenaces de seguretat aplicables.

El segon bloc consistirà en presentar un cas real on s'apliqui el format de document digitals desenvolupant un sistema informàtic on es permeti el tractament d'informació per a cercar dades compromeses a partir de patrons mitjançant el format Open Document.

Per a fer aquest desenvolupament seguirem el cicle tradicional del programari, és a dir, passarem per les fases d'especificació, anàlisi, disseny i implementació fins a arribar a l'objectiu proposat.

Finalment com a resultat dels blocs anteriors es presentarà una reflexió sobre el format Open Document, els avantatges que proporciona i també les amenaces relacionades que originaran nous punts de vista a desenvolupar en altres projectes d'investigació.

1.4 Planificació del projecte

La planificació del projecte ha consistit en dividir en tasques cadascuna de les fites proposades, per exemple en apartats anteriors s'ha comentat que hi haurà un primer bloc que consistirà en investigar certs formats i tecnologies, per tant dividirem aquest bloc en tantes fites com tecnologies s'estudiaran.

D'altra banda tenim un altre gran bloc que és la construcció de l'eina de programari que seguirà un cicle tradicional en cascada que comporta dividir aquest bloc en tasques ja conegudes: anàlisi, disseny i implementació.

Un cop feta aquesta divisió de tasques podríem haver pensat en usar programari específic per a realitzar la planificació però al tractar-se d'un projecte eminentment d'investigació s'ha optat per proposar una temporització adequada als estudis de les diverses tecnologies i a una administració del temps restant fins a l'entrega per al desenvolupament i reflexions finals.

1.4.1 Planificació amb fites i temporització

Per a poder identificar les fites haurem de fer-ho des de dos punts de vista: un a llarg termini que identificarà les principals tasques i un a més curt termini per a cadascuna d'elles que ens revelarà les tasques en que es divideix cadascuna de les fites.

Per tant considerarem cadascuna de les entregues a realitzar durant l'elaboració del projecte com les grans tasques a realitzar i a continuació les subdividirem en grups de tasques més petites. El resultat ha donat fruit al següent llistat de tasques:

1. Redacció del resum del treball
2. Redacció de la introducció
    a. Resum del projecte
    b. Redacció de l'estat de l'art: punt de partida i objectius.
    c. Planificació. Riscs i pla de contingències.
3. Capítol 2 – Estudi XML
    a. Estudi del format XML
        i. Breu història
        ii. Sintaxi del format
    b. Esquemes DTD
    c. Transformacions XSLT
    d. Redacció del capítol 2

4. Capítol 3 – Expressions regulars
    a. Redacció del capítol 3
5. Capítol 4 – Estudi ODF
    a. Estudi de l'arquitectura Open Document
        i. Breu història
    b. Estudi de les amenaces de seguretat a ODF
    c. Redacció del capítol 4
6. Capítol 5 – Eina de programari
    a. Anàlisi
    b. Disseny
    c. Implementació
7. Redacció de les conclusions, glossari i bibliografia.
8. Repàs final de la memòria
9. Elaboració de la presentació
10. Debat Virtual

En un segon terme s'ha considerat la temporització de cadascuna de les tasques tal i com es recull en la següent taula utilitzant el mètode de Gantt i considerant cadascuna de les Proves d'Avaluació Contínua com a fites del projecte. Tanmateix el nivell de detall es limita a les principals tasques a desenvolupar ja que altrament la taula perdria el seu objectiu d'informar.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Elaboració del Pla de Treball	13,5 días?	mar 11/03/08	mié 26/03/08	
2	**PAC 1: entrega del Pla de Traball**	**0 días**	**mié 26/03/08**	**mié 26/03/08**	
3	Redacció de la introducció	2 días	jue 27/03/08	vie 28/03/08	1
4	**Capítol 2 - XML**	**13 días**	**dom 30/03/08**	**lun 14/04/08**	**3**
5	Estudi XML	5 días	dom 30/03/08	vie 04/04/08	
6	Estudi transformacions XSLT	3 días	vie 04/04/08	mar 08/04/08	5
7	Redacció capítol 2	5 días	mar 08/04/08	lun 14/04/08	6
8	**Capítol 3 - Expressions regulars**	**3 días**	**lun 14/04/08**	**jue 17/04/08**	**4**
9	Estudi expressions regulars	2 días	lun 14/04/08	mié 16/04/08	
10	Redacció capítol 3	1 día	mié 16/04/08	jue 17/04/08	9
11	**Capítol 4 - Format OpenOffice (I)**	**9 días**	**jue 17/04/08**	**mar 29/04/08**	**8**
12	Part 1: format ODF	7 días	jue 17/04/08	vie 25/04/08	
13	Part 2: amenaces de seguretat	2 días	dom 27/04/08	mar 29/04/08	12
14	**PAC 2**	**0 días**	**mié 30/04/08**	**mié 30/04/08**	
15	**Capítol 4 - Format OpenOffice (i II)**	**5 días**	**mar 29/04/08**	**lun 05/05/08**	**11**
16	Redacció capítol 4	5 días	mar 29/04/08	lun 05/05/08	
17	**Capítol 5 - Eina de programari (I)**	**19 días**	**lun 05/05/08**	**mié 28/05/08**	**15**
18	**Especificació i anàlisi**	**5 días**	**lun 05/05/08**	**dom 11/05/08**	
19	Anàlisi de requeriments	2 días	lun 05/05/08	mié 07/05/08	
20	Generació XML / DTD	2 días	mié 07/05/08	vie 09/05/08	19
21	Generació dels jocs de proves	1 día	vie 09/05/08	dom 11/05/08	20
22	Disseny	3 días	dom 11/05/08	mié 14/05/08	18
23	**Implementació**	**8 días**	**mié 14/05/08**	**vie 23/05/08**	**22**
24	Implementació	6 días	mié 14/05/08	mié 21/05/08	
25	Validació dels jocs de proves	2 días	jue 22/05/08	vie 23/05/08	24
26	Redacció del capítol 5	3 días	dom 25/05/08	mié 28/05/08	23
27	**PAC 3**	**0 días**	**mié 28/05/08**	**mié 28/05/08**	
28	**Capítol 5 - Eina de programari (i II)**	**2 días**	**mié 28/05/08**	**vie 30/05/08**	**17**
29	Redacció del capítol 5	2 días	mié 28/05/08	vie 30/05/08	
30	Redacció del resum del treball	1 día	vie 30/05/08	dom 01/06/08	28
31	Repàs final de la memòria	1 día	dom 01/06/08	lun 02/06/08	30
32	Redacció de les conclussions, glossari i bibliografia.	2 días	lun 02/06/08	mié 04/06/08	31
33	Elaboració de la presentació	3 días	mié 04/06/08	dom 08/06/08	32
34	**PAC 4: entrega del PFC**	**0 días**	**dom 08/06/08**	**dom 08/06/08**	**33**
35	**Debat virtual**	5 días?	lun 23/06/08	vie 27/06/08	34

Fig. 2 - Tasques del projecte

La planificació temporal de cadascuna de les tasques segons el mètode de Gantt és la següent:

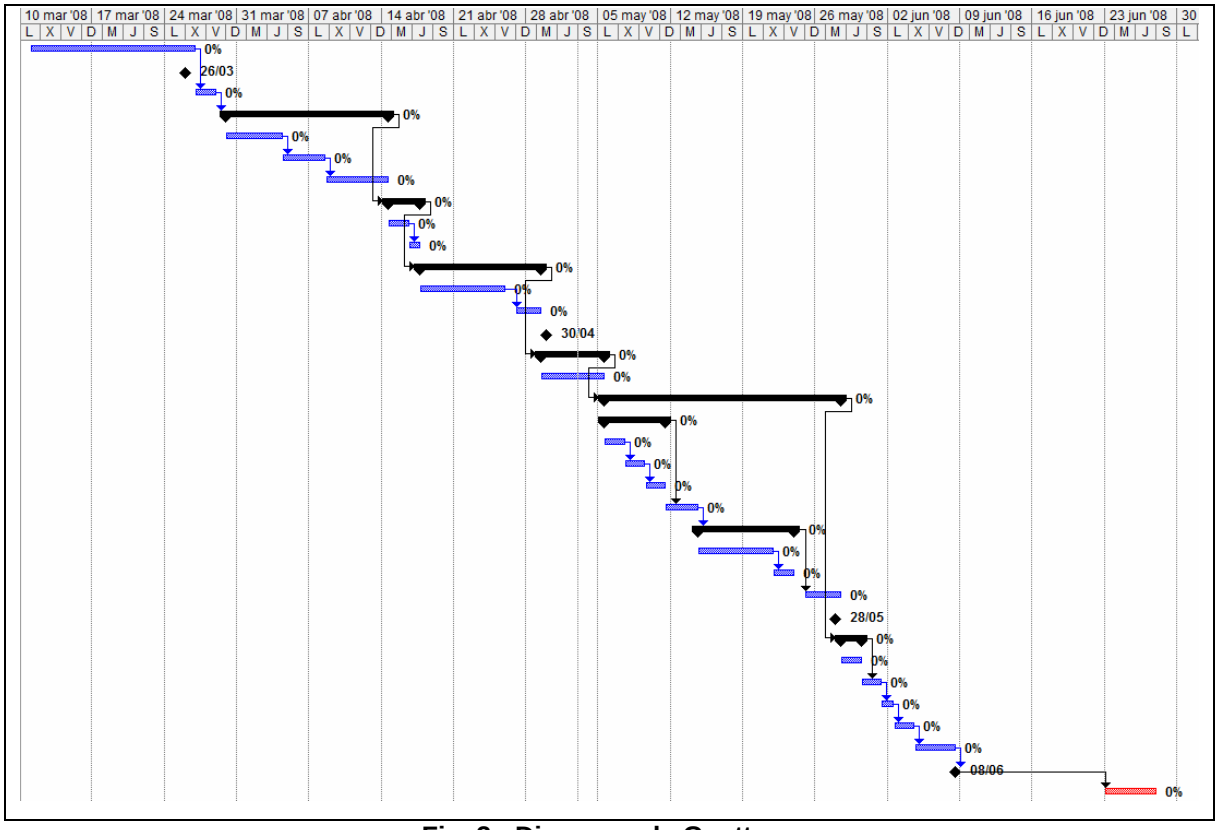


Fig. 3 - Diagrama de Gantt

1.5 Anàlisi de riscs

Per a la planificació de tot projecte hem de tenir en consideració els possibles riscs que es poden produir per a poder elaborar un pla de contingències que ens permeti actuar de forma ordenada en cas de produir-se.

Aquest anàlisi es divideix en diverses fases, que són:

- Identificació dels riscs, en aquesta fase identificarem els possibles riscs, el grau de repercussió sobre el projecte i les conseqüències que poden derivar-se.
- Pla de contingències, en aquesta fase es proposaran tasques d'actuació per a cadascun dels riscs anteriors indicant què s'ha de fer, la valoració temporal de l'actuació i altres riscs que se'n derivin.
- Control de riscs, en aquesta fase es catalogaran els riscs identificats i es farà un compendi de les fases anteriors afegint els actors responsables de les actuacions així com les seves prioritats d'execució.

1.5.1 Identificació dels riscs

1.5.1.1 Risc de problemes relacionats amb la investigació

- **Grau**: mig.
- **Conseqüències**: retràs de les etapes posteriors del projecte. No complir amb la fita: entrega de la PAC2.

1.5.1.2 Risc de problemes amb els recursos humans

- **Grau**: baix.
- **Conseqüències**: es podrien donar problemes de disponibilitat de recursos de personal que farien retardar el projecte.

1.5.1.3 Risc de problemes relacionats amb la tecnologia

- **Grau**: mig.
- **Conseqüències**: si apareguessin problemes de compatibilitat entre tecnologies es podria retardar el desenvolupament del projecte.

1.5.1.4 Risc de problemes amb els jocs de proves

- **Grau**: baix.
- **Conseqüències**: tornar a fer els jocs de proves.

1.5.1.5 Risc de problemes amb altres versions de documents ofimàtics

- **Grau:** baix.
- **Conseqüències:** no es podran usar documents d'altres formats, per exemple Office Word, versions anteriors del format, etc. o transformats d'altres formats.

1.5.2 Pla de contingències

El pla de contingències inclouria un esforç en personal i pressupost en cas de produir-se problemes de recursos humans o tecnològics respectivament.

Tot i que per el tipus de projecte no es preveuen problemes a la fase d'implementació si que cal tenir-los en consideració ja que el marge de maniobra és molt petit ja que la data d'entrega és inamovible. Així doncs es tindrà especial cura durant l'elaboració del pla de contingències.

Per a cadascun del problemes detectats a la fase d'anàlisi de riscs es proposen una sèrie de mesures:

1.5.2.1 Risc de problemes amb la investigació

- **Proposta d'actuació**: retardar la fita 'entrega de la PAC2' fins després del dia 1 de Maig.
- **Descripció**: es pacta una demora de la fita per a fer-la amb data posterior al dia que s'havia de fer efectiva amb la condició d'avançar la generació de la documentació del capítol posterior per a que consti també a la nova data d'entrega de la fita.
- **Impacte temporal**: es fixa com a dia màxim l'últim laborable de la setmana de la fita per a poder continuar les tasques amb normalitat el laboral següent. Això permet tenir els dos festius del cap de setmana hàbils per avançar les tasques enrederides.
- **Riscs associats**: no es preveu cap altre risc excepte que no es pugui dur a terme l'actuació proposada.

1.5.2.2 Risc de problemes amb els recursos humans

- **Proposta d'actuació**: augmentar l'esforç en les etapes posteriors al desviament fins a recuperar el desviament.
- **Descripció**: com que no es pot augmentar el nombre de recursos humans el que hi estan destinats hauran d'augmentar l'esforç per dia fins a recuperar el desviament.
- **Impacte temporal**: es preveu recuperar el temps i no haver d'endarrerir cap tasca.
- **Riscs associats**: el risc associat és l'endarreriment de tota la planificació.

1.5.2.3 Risc de problemes relacionats amb la tecnologia

- **Proposta d'actuació**: en cas de que sorgeixi algun problema amb la tecnologia augmentarem l'esforç dels recursos. En cas necessari s'haurà de revisar la planificació d'algunes tasques posteriors en el següent ordre: glossari, agraïments, valoració econòmica, conclusions, bibliografia, presentació.
- **Descripció**: s'augmentarà l'esforç per a aconseguir el bagatge tecnològic necessari mitjançant fòrums de consulta específics, llistes de distribució, recursos bibliogràfics, etc. En cas de ser necessari es plantejaria revisar la planificació d'algunes tasques menys importants per obtenir un augment de temps.
- **Impacte temporal**: en cas que sigui necessari la revisió de la planificació significarà que certes tasques veuran minvades les seves duracions.
- **Riscs associats**: en el pitjor dels casos pot comportar el no compliment de la fita 'entrega de la PAC3'.

1.5.2.4 Risc de problemes amb els jocs de proves

- **Proposta d'actuació**: reconstruir els jocs de proves.
- **Descripció**: es generaran nous jocs de proves adaptant-se als requeriments que siguin necessaris. En aquest cas s'haurà d'incloure una nova etapa després de la fita 'entrega de la PAC3' per a poder fer-ho.
- **Impacte temporal**: L'aparició d'una nova tasca implica que altres tasques es veuran afectades amb una reducció del temps disposat. Es proposa modificar les següents tasques: glossari, agraïments, valoració econòmica.
- **Riscs associats**: aquesta actuació pot comportar un retràs de les tasques afectades.

1.5.2.5 Risc de problemes amb altres versions de documents

- **Proposta d'actuació**: revisar el tractament dels documents.
- **Descripció**: es revisarà el funcionament de la classe destinada al tractament del document així com s'esbrinarà si les tecnologies usades (llibreries d'ODF) permeten fer la tasca amb versions anteriors o provinents d'altres formats.
- **Impacte temporal**: es concentrarà a l'etapa de desenvolupament i sobretot a la de proves ja que en aquesta sorgiran la majoria de problemes amb els formats no ODT.
- **Riscs associats**: aquesta actuació pot comportar un retràs de les tasques afectades.

1.5.3 Control de riscs

El control de riscs ens permet tenir totalment identificats els riscs que es preveuen pel projecte i també cadascuna de les propietats que els defineixen.

Un aspecte important és que per a cada risc es té constància de quins són els actors responsables, la categorització de prioritats, la categorització d'impacte i les actuacions que han de portar-se a terme.

A continuació es presenta el control de riscs del projecte:

1.5.3.1 Risc de problemes amb la investigació

Identificador del risc	R001
Títol del risc	Risc de problemes amb la investigació
Descripció	Hi ha un risc potencial si es produeix un endarreriment durant l'etapa d'investigació.
Probabilitat d'ocurrència (A/M/B)	M
Impacte sobre el projecte (A/M/B)	M
Prioritat (A/M/B)	M
Acció	Endarrerir la fita 'entraga PAC2', augmentar l'esforç i avançar a la fita part de la fita següent.
Responsable	Jordi Duran / Jesús Delgado
Data inici / Data acabament	1 de Maig / 4 de Maig
Objectius	Recuperar el temps de desviament.
Acabament	4 de Maig

1.5.3.2 Risc de problemes amb els recursos humans

Identificador del risc	R002
Títol del risc	Risc de problemes amb els recursos humans
Descripció	Es pot produir problemes de disponibilitat dels recursos humans.
Probabilitat d'ocurrència (A/M/B)	M
Impacte sobre el projecte (A/M/B)	A
Prioritat (A/M/B)	A
Acció	Augmentar l'esforç en etapes posteriors.
Responsable	Jesús Delgado
Data inici / Data acabament	Inici del projecte fins el 8 de Juny
Objectius	Recuperar el temps de les desviacions.
Acabament	8 de Juny

1.5.3.3 Risc de problemes relacionats amb la tecnologia

Identificador del risc	R003
Títol del risc	Risc de problemes relacionats amb la tecnologia
Descripció	Revisió de la planificació de l'etapa d'implementació.
Probabilitat d'ocurrència (A/M/B)	M
Impacte sobre el projecte (A/M/B)	A
Prioritat (A/M/B)	A
Acció	Augmentar l'esforç en la fase d'implementació. Revisar la planificació d'algunes etapes en el següent ordre: glossari, agraïments, valoració econòmica, conclusions, bibliografia, presentació.
Responsable	Jordi Duran / Jesús Delgado
Data inici / Data acabament	14 de Maig fins el 8 de Juny
Objectius	Aconseguir el bagatge tecnològic suficient. Recuperar la desviació de la tasca d'implementació.
Acabament	8 de Juny

1.5.3.4 Risc de problemes amb els jocs de proves

Identificador del risc	R004
Títol del risc	Risc de problemes amb els jocs de proves
Descripció	Els jocs de proves no són suficients per complir els requeriments. Es procedirà a generar uns nous jocs de proves.
Probabilitat d'ocurrència (A/M/B)	B
Impacte sobre el projecte (A/M/B)	B
Prioritat (A/M/B)	B
Acció	Generar uns nous jocs de proves.
Responsable	Jordi Duran / Jesús Delgado
Data inici / Data acabament	30 de Maig al 2 de Juny
Objectius	Complir la tasca d'elaboració dels jocs de proves i test de l'aplicació.
Acabament	2 de Juny

1.5.3.5 Risc de problemes amb altres versions de documents

Identificador del risc	R005
Títol del risc	Risc de problemes amb altres versions de documents ofimàtics
Descripció	Els jocs detecten que l'eina no pot tractar documents de versions anteriors o provinents d'altres formats (per exemple Word)
Probabilitat d'ocurrència (A/M/B)	M
Impacte sobre el projecte (A/M/B)	B
Prioritat (A/M/B)	B
Acció	Esbrinar si es pot transformar el document a la versió actual d'OpenOffice amb l'eina Writer
Responsable	Jesús Delgado
Data inici / Data acabament	30 de Maig al 2 de Juny
Objectius	Poder tractar documents provinents d'altres formats
Acabament	2 de Juny

1.6 Revisió de la planificació

Durant el desenvolupament de les diferents etapes del projecte s'han fet realitat alguns dels riscs que s'havien previst.

Amb l'ajuda del pla de contingències s'ha pogut revisar la planificació del projecte per a poder complir les fites proposades. Això a comportat augmentar l'esforç en alguna tasca a partir de la desviació ja que no es podien afegir més recursos humans.

La següent figura representa la revisió de les tasques de la planificació inicial on es poden veure en color verd les desviacions que s'han produït:

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Elaboració del Pla de Treball	13,5 días?	mar 11/03/08	mié 26/03/08	
PAC 1: entrega del Pla de Traball	**0 días**	**mié 26/03/08**	**mié 26/03/08**	
Redacció de la introducció	2 días	jue 27/03/08	vie 28/03/08	1
Capítol 2 - XML	**13 días**	**dom 30/03/08**	**lun 14/04/08**	**3**
Estudi XML	5 días	dom 30/03/08	vie 04/04/08	
Estudi transformacions XSLT	3 días	vie 04/04/08	mar 08/04/08	5
Redacció capítol 2	5 días	mar 08/04/08	lun 14/04/08	6
Capítol 3 - Expressions regulars	**3 días**	**lun 14/04/08**	**jue 17/04/08**	**4**
Estudi expressions regulars	2 días	lun 14/04/08	mié 16/04/08	
Redacció capítol 3	1 día	mié 16/04/08	jue 17/04/08	9
Capítol 4 - Format OpenOffice (I)	**9 días**	**jue 17/04/08**	**mar 29/04/08**	**8**
Part 1: format ODF	7 días	jue 17/04/08	vie 25/04/08	
Part 2: amenaces de seguretat	2 días	dom 27/04/08	mar 29/04/08	12
Capítol 4 - Format OpenOffice (i II)	**4 días**	**mar 29/04/08**	**dom 04/05/08**	**11**
Redacció capítol 4	4 días	mar 29/04/08	dom 04/05/08	
PAC 2	0 días	dom 04/05/08	dom 04/05/08	
Capítol 5 - Eina de programari (I)	**19 días**	**lun 05/05/08**	**mar 27/05/08**	**14**
Especificació i anàlisi	**5 días**	**lun 05/05/08**	**vie 09/05/08**	
Anàlisi de requeriments	2 días	lun 05/05/08	mar 06/05/08	
Generació XML / DTD	2 días	mié 07/05/08	jue 08/05/08	19
Generació dels jocs de proves	1 día	vie 09/05/08	vie 09/05/08	20
Disseny	3 días	dom 11/05/08	mié 14/05/08	18
Implementació	**8 días**	**mié 14/05/08**	**vie 23/05/08**	**22**
Implementació	6 días	mié 14/05/08	mié 21/05/08	
Validació dels jocs de proves	2 días	mié 21/05/08	vie 23/05/08	24
Redacció del capítol 5	3 días	vie 23/05/08	mar 27/05/08	23
PAC 3	**0 días**	**mié 28/05/08**	**mié 28/05/08**	
Capítol 5 - Eina de programari (i II)	**2 días**	**mar 27/05/08**	**jue 29/05/08**	**17**
Redacció del capítol 5	2 días	mar 27/05/08	jue 29/05/08	
Redacció del resum del treball	1 día	jue 29/05/08	vie 30/05/08	28
Revisió dels jocs de proves	**2 días**	**vie 30/05/08**	**lun 02/06/08**	**30**
Redacció de les conclussions, glossari i bibliografia.	1 día	mar 03/06/08	mar 03/06/08	31
Elaboració de la presentació	3 días	mié 04/06/08	vie 06/06/08	32
PAC 4: entrega del PFC	**0 días**	**dom 08/06/08**	**dom 08/06/08**	**33**
Debat virtual	5 días?	lun 23/06/08	vie 27/06/08	34

Fig. 4 - Revisió de la planificació

La següent figura es correspon al diagrama de Gannt associat:

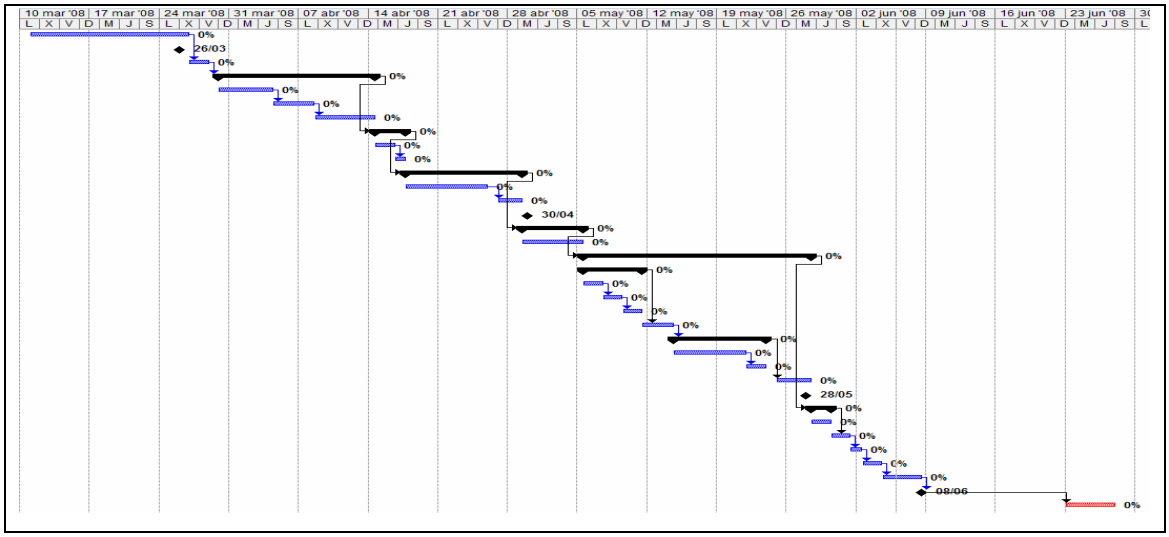


Fig. 5 - Diagrama de Gannt de la revisió

1.7 Productes obtinguts

El desenvolupament d'aquest projecte ha de permetre definir patrons de cerca per a l'Agencia Catalana de Seguretat basats en expressions regulars amb l'objectiu de que siguin emprats en sistemes informàtics en els seus processos de cerca d'informació compromesa.

D'altra banda el producte més important serà una aplicació que farà ús dels patrons anteriors per a cercar dades concretes dins de documents digitals basats en el format Open Document, generi un document XML amb els resultats del procés i finalment els presenti en format HMTL mitjançant transformacions XML.

1.8 Breu descripció dels altres capítols de la memòria

En l'apartat anterior s'han presentat dos desenvolupaments que seran part del segon bloc del projecte. En el primer bloc es tractaran els següents temes, cadascun dels quals es correspondrà a un capítol del segon bloc:

1) L'estudi del format XML i concretament la sintaxi, els esquemes DTD associats i el procés de validació dels documents que segueixen aquest format.
2) L'estudi de les expressions regulars com a eina fonamental de la informàtica en els processos de cerca mitjançant patrons.
3) L'estudi del format de document digital ODF a partir de la presentació de la seva arquitectura, les amenaces de seguretat relacionades i una comparativa amb les tecnologies competidores.
4) Per últim, el tercer bloc serà u Una reflexió de les amenaces de seguretat del format ODF i possibles aplicacions per a futurs projectes d'investigació.

Capítol 2 El format XML

2.1 Definició i objectius

El format XML (Extensible Markup Language) es un metallenguatge extensible de marques desenvolupat pel World Wide Web Consortium com a simplificació i adaptació del llenguatge SGML i que per definició permet definir la gramàtica de llenguatges específics **[1]**.

En conseqüència el format XML ens permet definir nous llenguatges per a diferents necessitats fins al punt d'establir-se com a estàndard d'intercanvi de dades entre diferents plataformes.

2.2 Breu història

Durant la dècada dels setanta IBM va crear un llenguatge anomenat Generalized Markup Language (GML) per a resoldre les pròpies necessitats d'emmagatzemar informació de forma estructurada que es va ser elevat a estàndard pel comitè ISO a l'any 1986 sota la norma Standard Generalized Markup Language (SGML). Aquesta norma definia un metallenguatge per a definir nous llenguatges basats en etiquetes i unes definicions de documents anomenades DTD que en validaven la consistència.

L'any 1989 Tim Berners Lee va crear Internet i amb ella el llenguatge HTML definit sota el marc del llenguatge SGML. Molt ràpidament aquest llenguatge es va convertir en la més important de les aplicacions creades a Internet però el seu ús va generar el caos: els navegadors Web enlloc de respectar la sintaxi del nou llenguatge van ser molt permissius fins al punt de no incorporar un analitzador genèric sinó lògica ad hoc per a validar la consistència dels documents.

Arrel d'aquest fet es va crear el format XML com a subconjunt del llenguatge SGML amb objectiu de facilitar la seva interpretació per part del programes.

2.3 Característiques del format XML

La principals característiques d'aquest llenguatge són:

- Es tracta d'un llenguatge extensible, és a dir, un cop dissenyat es possible afegir-hi noves marques de manera que els antics consumidors encara puguin entendre el nou format.
- És estructurat i per tant es senzill entendre la estructura i processar-la millorant així la compatibilitat entre aplicacions.
- Va acompanyat d'una definició de tipus que permet crear analitzadors genèrics.

Per a poder proporcionar aquestes característiques i, com hem vist en apartats anteriors, poder convertir-se en un estàndard d'intercanvi de dades entre sistemes, el format XML s'ha de basar en un format no dependent de plataforma, allunyat d'implementacions propietàries i de mida reduïda. Aquest format no és altra que el format de text pla el qual no es veu afectat per les limitacions de seguretat dels sistemes, al ser un format inofensiu pels sistemes basant-se en els mateixos principis que l'arquitectura SOA o dit d'altra forma l'arquitectura SOA es basa en el format XML per a aconseguir aquesta transparència.

2.3.1 Estructura del format XML

El format XML expressa la informació mitjançant una estructura jeràrquica, és a dir, es tracta d'una composició de parts que a la vegada es composen d'altres parts amb la condició que en tota l'estructura només existeix una única part arrel.

Cadascuna d'aquestes parts són anomenades element i expressades (o marcades) mitjançant marques amb la següent codificació:

<marca></marca>

Per exemple imaginem que volem emmagatzemar receptes de cuina. Potser podríem pensar que una recepta té una part corresponent als ingredients què necessitem composada d'altres parts que són els propis ingredients:

```
<recepta>
    <ingredients>
        <ingredient>Pà</ingredient>
        <ingredient>Oli</ingredient>
        <ingredient>Sal</ingredient>
        <ingredient>Tomàquet</ingredient>
    </ingredients>
</recepta>
```

Fig. 6 - Exemple de document XML

Podem visualitzar millor l'estructura jeràrquica de l'exemple anterior mitjançant la següent figura:

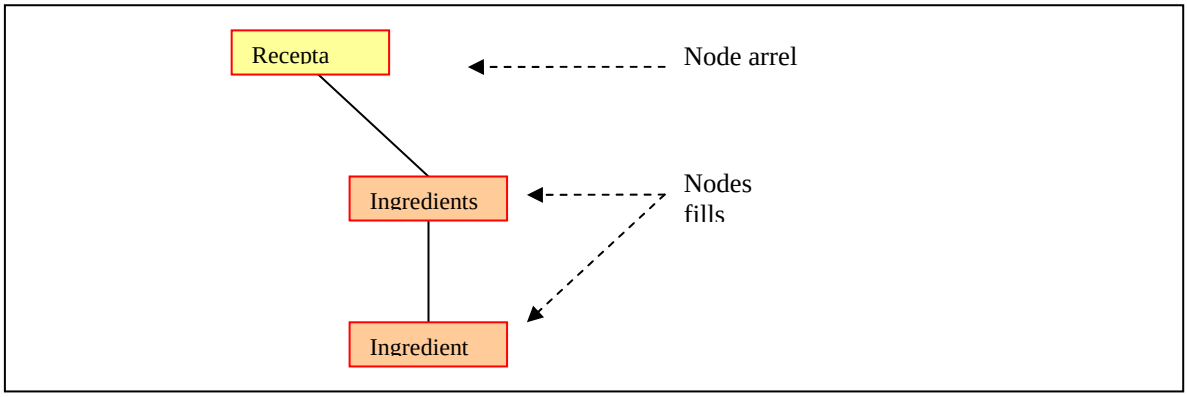


Fig. 7 - Estructura jeràrquica d'un document XML

Tal com hem vist en l'apartat precedent un document en format XML conté una sèrie d'elements que s'estructuren de forma jeràrquica dins d'un document. Addicionalment cadascun d'aquests elements pot contenir atributs que es definiran dins de la pròpia marca seguint la següent codificació:

<marca atribut='valor'></marca>

I en aquest cas l'element 'marca' conté un atribut anomenat 'atribut' amb valor 'valor'.

Reprenent l'exemple anterior podríem pensar en afegir-hi un títol per que una persona pugui identificar la recepta i un número intern destinat a identificar la recepta dins del sistema de receptes:

```
<recepta numero='0001'>
    <titol>Pà amb tomàquet</titol>
    <ingredients>
        <ingredient>Pà</ingredient>
        <ingredient>Oli</ingredient>
        <ingredient>Sal</ingredient>
        <ingredient>Tomàquet</ingredient>
    </ingredients>
</recepta>
```

Fig. 8 - Atributs en un document XML

2.3.2 Codificació dels documents

En un altre ordre de coses, degut a la naturalesa de la informació, ens podem trobar que aquesta és expressada amb diferents idiomes i per tant, a més baix nivell, el format XML ha de ser capaç de treballar amb diferents codificacions de caràcters. Actualment la versió 1.0 del format XML ens permet treballar amb gairebé tots els idiomes però amb la limitació de referir-se només als valors dels elements o atributs.

```
<?xml version="1.0" encoding="UTF-8"?>
<book>
<title>የማቴዎስ ወንጌል</title>
<chapter number="፩">
<title>የኢየሱስ የትው ልድ ሐረግ</title>
<verse number="፩">
  የጻዊት ልጅ፣ የአከርሃም ልጅ የሁ ነው የኢየሱስ ከርስቶስ የትው ልድ ሐረግ የሚከተለው ነው፤
</verse>
<verse number="፪">

አከርሃም ይሰሐቅን ወለደ፤

ይሰሐቅ ያዕቆብን ወለደ፤

ያዕቆብ ይሁዳንና ወነድሞቹን ወለደ፤

</verse>
</chapter>
</book>
```

Fig. 9 - Codificació d'un document XML versió 1.0

Per a idiomes menys freqüents, com l'arameu, o per a poder usar marques localitzades necessitarem usar la versió 1.1 del format XML però per norma general es desaconsella l'ús de la localització de marques ja que impedeix la compatibilitat entre aplicacions:

```
<?xml version="1.1" encoding="UTF-8"?>
<መጽሐፉ>
<አርእስት>የማቴዎስ ወንጌል</አርእስት>
<ምዕራፉ ቁጥር="፩">
<አርእስት>የኢየሱስ የትውልድ ሐረግ</አርእስት>
<ቤት ቁጥር="፩">
  የዳዊት ልጅ፣ የአከርሃም ልጅ የሁ ነው የኢየሱስ ከርስቶስ የትውልድ ሐረግ የሚከተለው ነው፤
</ቤት>
<ቤት ቁጥር="፪">

አከርሃም ይሰሐቅን ወለደ፤

ይሰሐቅ ያዕቆብን ወለደ፤

ያዕቆብ ይሁዳንና ወንድሞቹን ወለደ፤

</ቤት>
</ምዕራፉ>
</መጽሐፉ>
```

Fig. 10 - Codificació d'un document XML versió 1.1

Per a que els sistemes puguin treballar amb diferents idiomes i que aquests siguin usables en un document XML se n'haurà d'indicar la codificació utilitzada, d'aquesta manera un sistema receptor podrà determinar la codificació del document i llegir correctament els caràcters que hi són continguts.

Pel que fa a les codificacions usades les pròpies del format XML es basen en l'estàndard Unicode però es pot determinar-ne el codi mitjançant l'ús de l'atribut de codificació en la declaració del document. En els exemples anteriors se n'ha fet ús de la codificació Unicode UTF-8 i tanmateix per a cadascun s'ha establert també la versió de format XML com podem recordar a continuació:

<?xml version="1.0" encoding="UTF-8"?>

<?xml version="1.1" encoding="UTF-8"?>

2.3.3 Parts d'un document XML

Un document XML es divideix en dues grans entitats que contenen a la vegada entitats de més baix nivell com poden ser els elements o atributs; aquestes dues grans entitats són:

- El pròleg
- El cos

El pròleg informa que el document segueix el format XML i concretament la versió però també de forma opcional la codificació de caràcters usada i la tipologia de document.

<?xml version="1.0" encoding="UTF-8"?>

Respecte al cos del document ja hem vist que està format per elements que s'estructuren de forma jeràrquica a partir d'un element arrel i aquests elements a la vegada poden contenir d'altres elements.

Com també hem vist, en un segon nivell es poden articular atributs els quals permeten completar la definició de cadascun dels elements anteriors.

Completant la definició del cos d'un document XML trobem les entitats predefinides, les seccions CDATA i els comentaris. Tant les entitats predefinides com les seccions CDATA ens permeten expressar valors especials que no seran tractats pel processador XML de forma que les primeres es codifiquen d'una forma reconeguda pel processador i les segones tenen un testimoni d'inici i un altre de final que indica la part que no s'ha de processar. Per últim trobem els comentaris que fan ús de testimonis per a encapsular cadenes de text amb caràcter informatiu que no ser processades ja que únicament estan destinades a informar als desenvolupadors del document.

<!-- Comentari dins d'un document XML que no serà processat. -->

2.3.4 Sintaxi del format XML

Fins ara hem vist com s'organitza un document en format XML i quines entitats en formen part, però també necessitem conèixer les regles bàsiques que fan que un document compleixi el format XML o en paraules més formals "que un document sigui un document XML ben format".

En un primer nivell s'ha de destacar que el format XML és sensible a les minúscules i les majúscules, és a dir, que podem trobar el mateix nom de marca per a dos elements diferents si es que aquests es diferencien segons majúscules i minúscules. Per exemple:

<marca></marca> és un element diferent a <Marca></Marca>

En un segon nivell es necessari que el document informi una declaració:

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

Concretament la declaració informa el següent:

- El format del document mitjançant l'element xml
- La versió XML que se segueix mitjançant l'atribut version
- La codificació de caràcters usada mitjançant l'atribut encoding
- El tipus del document mitjançant l'atribut standalone. Si el valor d'aquest atribut es "no", opció per defecte, indica que el document fa referència a un document de definició de tipus; en cas contrari, valor de l'atribut "yes", indica que el document és independent.

En un tercer nivell el cos del document, per definició, ha de presentar un únic element com a arrel i tots els elements i atributs dels elements que hi depenen han de ser sintàcticament correctes. Les propietats sintàctiques variaran segons es tracti d'elements o atributs i s'enumeren a continuació:

- Els elements han de tenir una etiqueta d'inici i una altra de final o bé si tenen valor nul una marca d'inici seguida d'un caràcter de fi de marca:

<marca></marca> o bé <marca/>

- Els valors dels atributs han d'estar delimitats per cometes dobles o simples

<marca atribut="valor"/> o bé <marca atribut='valor'/>

Finalment, si un document compleix aquestes propietats sintàctiques direm que es tracta d'un document XML ben format tal com anunciàvem al inici de l'apartat.

2.4 La definició de tipus de document DTD

Que un document sigui un document XML ben format només garanteix que respecta la sintaxis bàsica del format XML però cada aplicació, és a dir, cada llenguatge XML creat necessitarà quelcom que defineixi les relacions que s'han de verificar entre els elements.

Aquest quelcom és el que anomenem definició de tipus de document[1] que permetrà analitzar a posteriori els documents d'aquest llenguatge i per tant direm que valida els documents.

[1] En endavant DTD (acrònim anglès de Document Type Definition).

2.4.1 Característiques i objectius

L'objectiu dels DTD és proporcionar una descripció d'estructura i sintaxi que pugui ser compartida per diferents documents i en conseqüència que aquests últims puguin ser validats, coneguin l'estructura dels elements i la descripció de les dades que hi contenen, facilitant la compartició d'informació (els propis documents) entre diferents sistemes (programes).

Per a poder assolir aquest objectiu el DTD defineix un nou llenguatge de marques que s'incorpora a una aplicació (llenguatge XML) determinada en un document XML i com hem vist en l'apartat anterior definir un atribut a la declaració de l'aplicació informant si el DTD és intern o se'n fa referència a un d'extern.

Concretant la definició de DTD aquest descriu:

- Els elements que són permesos dins l'aplicació i el contingut d'aquestes etiquetes.
- L'ordre en que es contindran els elements anterior o estructura.
- La jerarquia de les etiquetes o quines van dintre de quines.

2.4.2 Utilització d'un document DTD

Un document DTD constarà d'una declaració XML i a més enumerarà una sèrie d'entitats que definiran l'estructura dels documents XML. Entre aquestes entitats destaquem:

- Entitat <!ELEMENT> la qual permet definir elements de documents XML i per a cada element, segons sigui fulla o no, el tipus de dada o els elements que hi són continguts respectivament.

<!ELEMENT recepta (titol, ingredients, preparacio, dificultat?)>

<!ELEMENT ingredient (#PCDATA)>

- Entitat <!ATTLIST> la qual permet definir un atribut per a un element determinat podent indicar el tipus de data a usar, el valor per defecte i altres propietats com per exemple si es requerit o no.

<!ATTLIST recepta id CDATA #REQUIRED>

Reprenent l'exemple de les receptes i suposant que hem definit un document DTD que valida el document XML hauríem de procedir de la següent manera:

1. Definir un document DTD d'acord amb les nostres expectatives:

```
<?xml version="1.0" encoding="utf-8" ?>

<!ELEMENT receptes (recepta*)>
<!ELEMENT recepta (titol, ingredients, preparacio, dificultat?)>
<!ELEMENT titol (#PCDATA)>
<!ELEMENT ingredients (ingredient+)>
<!ELEMENT preparacio (#PCDATA)>
<!ELEMENT dificultat (#PCDATA)>
<!ELEMENT ingredient (#PCDATA)>

<!ATTLIST recepta id CDATA #IMPLIED>
```

Fig. 11 - Definició d'un document DTD

2. Informar una declaració de document DTD que podria ser tant interna com externa, però que en el nostre cas serà externa:

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<!DOCTYPE receptes SYSTEM "Receptes.dtd">
<receptes>
  <recepta id="0001">
    <titol>Pà amb tomàquet</titol>
    <ingredients>
      <ingredient>Pà</ingredient>
      <ingredient>Tomàquet</ingredient>
      <ingredient>Oli d'oliva</ingredient>
      <ingredient>Sal</ingredient>
    </ingredients>
    <preparacio>Es talla una llesca de pà, es suca el
    tomàquet i a continuació se li posa una mica de
    sal per sobre finalitzant amb un rajolí d'oli
    </preparacio>
    <dificultat>Baixa</dificultat>
  </recepta>
</receptes>
```

Fig. 12 - Document XML amb DTD associat

En el nostre exemple podem veure que en el DTD s'ha definit una llista de receptes i cadascuna d'elles té una sèrie d'elements comuns: un títol, una preparació, una llista de com a mínim un ingredient anomenada ingredients i opcionalment un nivell de dificultat.

A més una recepta té un atribut que l'identifica dins de l'aplicació.

Més a baix nivell podem observar com s'ha definit que un element és opcional mitjançant el caràcter "?", o bé com es pot indicar que un element és un conjunt d'elements amb l'ús del caràcter "+" el qual indica que com a mínim hi haurà una aparició.

En un altre ordre de coses per als elements fulla s'ha d'indicar el tipus de dades que es pot usar encara que per limitacions del format només podem usar tipus de dades alfanumèric (PCDATA) ja que no es permès indicar ni tipus de dates numèriques, ni dates, ni monedes.

2.4.3 Validació XML mitjançant DTD

El procés de validació consisteix en comprovar que un document XML donat està ben format i que és consistent amb l'estructura definida per el document DTD relacionat.

Completant aquesta definició direm que el procés de validació comporta un major grau de fiabilitat, consistència i precisió, facilitant el procés d'intercanvi de documents XML entre sistemes i augmentant la seva funcionalitat i utilitat.

Per a aconseguir això el procés de validació fa una sèrie de tasques, concretament verifica:

- La correctesa de les dades, el que permet detectar dades incorrectes com valors nuls no permesos o valors fora de rang.
- La integritat de les dades, es comprova que tota la informació requerida sigui present al document.
- L'enteniment compartit de les dades, on es comprova que tant l'emissor com el receptor interpretin per igual el document.

2.4.3.1 Inconvenients de la validació XML mitjançant DTD

Els documents DTD són una herència del llenguatge SGML i en tant que són la primera forma de validació mitjançant esquema del format XML presenten una sèrie de mancances, algunes de les quals són les següents:

- No és apte per a suportar noves ampliacions del format XML
- No és capaç de descriure certs aspectes formals en un document a nivell expressiu

Exemples de la segona limitació serien els següents:

- Un document DTD no pot definir elements locals que només són vàlids dins d'altres elements. Per a resoldre aquest problema s'ha de fer ús dels espais de noms que permeten afegir un context a un element per a diferenciar-lo d'un altre que usi la mateixa marca.
- Un document DTD no pot definir tipus de dades numèriques, dates o monedes com a valors per als seus elements o atributs.

2.5 Transformacions XSLT

2.5.1 Definició

Les transformacions XSL o XSLT són la part més important de l'Extensible Stylesheet Language o XSL i la seva finalitat és la de transformar documents XML en documents XHTML o altres documents XML.

2.5.2 Procés de transformació

Les transformacions XSL fan ús de la tecnologia XPath per a realitzar cerques d'informació a través del document XML amb l'objectiu d'identificar entitats a partir de les quals realitzar la transformació així com accions a realitzar amb l'entitat a tractar, entre les quals: recórrer l'estructura de l'entitat, obtenir el valor de la mateixa o d'algun dels seus atributs, obtenir el nombre d'elements que conté, etc.

Continuant amb l'exemple de les receptes de cuina imaginem que ens proposem fer una plana Web que funcioni com a compendi de planes relacionats amb les receptes de cuina. Com que volem emmagatzemar tots els enllaços d'una forma ordenada pensem en desar-los en un document en format XML ja que ens permetrà compartir-lo amb altres aplicacions i a més mitjançant una transformació XSL podrem generar una plana Web.

Mostrem aquí el document XML que conté els enllaços a diverses planes de cuina:

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet href="Enllaços.xsl" type="text/xsl"?>
<links>
  <link>
    <nom>
      Receptes personals
    </nom>
    <url>
      http://www.lesmevesreceptes.cat
    </url>
  </link>
  <link>
    <nom>
      Karlos Arguiñano
    </nom>
    <url>
      http://www.karlosnet.com
    </url>
  </link>
</links>
```

Fig. 13 - Exemple de document XML

A continuació el document XSL que ens permetrà fer la transformació desitjada:

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="links">
    <html>
      <head>
        <title>Receptes de cuina</title>
        <h1>Receptes de cuina</h1>
        <h3>Llistat de planes de receptes de cuina disponibles:</h3>
        <xsl:apply-templates select="link"></xsl:apply-templates>
      </head>
      <body>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="link">
    <xsl:element name="a">
          <xsl:attribute name='href'>
            <xsl:value-of select="url"/>
          </xsl:attribute>
          <xsl:value-of select="nom"/>
        </xsl:element>
    <br></br>
  </xsl:template>
 </xsl:stylesheet>
```

Fig. 14 - Document de transformació XSL

A continuació el resultat en format HTML:

```
<html>
  <head>
    <META http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Receptes de cuina</title>
    <h1>Receptes de cuina</h1>
    <h3>Llistat de planes de receptes de cuina disponibles:</h3><a href="&#xA;
http://www.lesmevesreceptes.cat&#xA;    ">
      Receptes personals
    </a><br><a href="&#xA;      http://www.karlosnet.com&#xA;    ">
      Karlos Arguiñano
    </a><br></head>
  <body></body>
</html>
```

Fig. 15 - Resultat de la transformació XSL

I finalment la visualització com a plana Web:

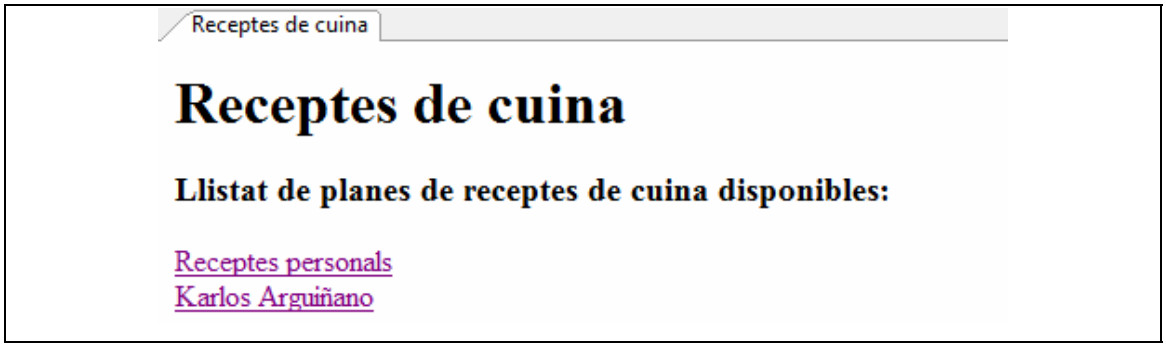


Fig. 16 - Vista prèvia document HTML

Com a cloenda d'aquest apartat afegirem que els detalls de les transformacions XSL queden fora de l'abast d'aquesta memòria i es recomana consultar la bibliografia al respecte indicada al final d'aquest volum.

Capítol 3 Expressions regulars

3.1 Definició

S'entén per expressió regular, també anomenada patró, a aquella expressió que descriu un conjunt de cadenes sense enumerar als seus elements i concretament a l'àrea de la programació permeten realitzar cerques dins de cadenes de caràcters.

Cal remarcar que les expressions regulars formen part del camp de la Teoria d'Autòmats i Llenguatges Formals que per a aquest projecte no es considera necessari aprofundir. Es recomana per a un coneixement més profund es faci referència de la bibliografia recomanada al final de la memòria.

3.2 Motors de cerca

Per a poder fer cerques mitjançant expressions regulars hem de disposar d'aplicacions amb capacitat d'usar-les; aquestes aplicacions són conegudes com motors de cerca i en distingirem dos tipus: les destinades a usuaris finals i les destinades a desenvolupadors.

Les primeres estan destinades a permetre als usuaris finals fer cerques sobre el contingut d'arxius o sobre un text extret i aplicat a un programa.

Respecte a les segones aquestes estan destinades a processaments repetitius de cerques, és a dir, poder desenvolupar processos automatitzats de cerques principalment en aplicacions. Aquests motors estan implementats en forma de llibreries que acompanyen a les plataformes sobre les que estan desenvolupades, per exemple destaquen:

- Package java.util.regex, llibreries d'expressions regulars per a la plataforma Java de SUN
- System.Text.RegularExpressions, llibreries d'expressions regulars per a la plataforma Microsoft .NET
- Perlre, llibreries d'expressions regulars per a PERL

El funcionament de les llibreries, centrant-nos en la de Java i .NET, en general és el mateix amb lleugeres diferències. Totes dues incorporen una classe que permet definir un patró o expressió regular que serà proporcionada a un motor de cerca implementat amb una altra classe. Aquesta classe proporcionarà mètodes per a fer la cerca sobre un objecte text segons el patró abans esmentat.

3.3 Avantatges i inconvenients

Amb l'ús d'expressions regulars som capaços de fer comparacions sobre texts amb l'objectiu de trobar determinades seqüències de caràcters però també podem extreure aquestes porcions de text o fins i tot reemplaçar-les, cosa que ens pot facilitar molt el desenvolupament d'aplicacions que fan tractaments extensius de text o en un altre ordre de coses poder fer validacions sobre les dades que introdueix un usuari fins al punt de poder evitar atacs com els d'injecció SQL mitjançant expressions regulars que analitzen el text parametritzat.

Com a principal inconvenient trobem que cada implementació dependent de plataforma té les seves especificitats en la codificació de les expressions regulars amb el que algunes capacitats les trobem a una plataforma però no a la resta. Per sort aquestes peculiaritats són ben documentades i no afecten al concepte propi de les expressions regulars.

> *Per exemple alguns motors de cerca permeten, addicionalment, indicar que es vol cercar únicament paraules complertes durant el procés de cerca.*

3.4 Construcció d'una expressió regular

Tot i que en un primer moment les expressions regulars poden ser críptiques i de fet ho són la construcció d'un patró és relativament fàcil.

De totes maneres la millor manera de construir bons patrons es basa en dos conceptes:

1. Estar familiaritzat amb les expressions regulars. Una gran idea és llegir bons manuals d'expressions regulars per familiaritzar-se amb elles.
2. Crear expressions regulars **pot ser confús** però llegir-les **realment sí que ho és de confús**. Les expressions regulars complexes poden portar a reptes molt difícils quan algú ha de validar la seva correctesa.

Canviant de registre els conceptes per a crear una expressió regular són següent:

1. Tot patró és una sèrie de caràcters amb possibles repeticions i on hi té importància si aquests estan en majúscules o no.
2. Els caràcters especials s'han d'indicar amb una posant al davant una contrabarra, per exemple, el retorn de carro seria representat per "\r".
3. Es poden indicar comodins que ens permeten consumir seqüències determinades dins dels texts. Per exemple el caràcter "." ens permet indicar que es consumirà qualsevol caràcter excepte el salt de línia "\n".

4. De comodins n'hi ha tres especialment importants: "*", "+" i "?" ja que cadascun d'ells representa el nombre de repeticions que s'aplica al caràcter o rang de caràcters que el precedeix. El caràcter "*" representa zero o més repeticions, el caràcter "+" representa com a mínim una repetició mentre que el caràcter "?" indica opció d'aparició.
5. Es poden indicar rangs de caràcters, de forma explícita, per exemple la construcció [a - z] equival a dir qualsevol caràcter en minúscula de l'alfabet anglès, és a dir, les vocals accentuades no en formarien part.

Els conceptes revisats en aquest apartat són una petita llista dels existents i es recomana l'ús de referències més complertes ja que no és l'objectiu d'aquesta memòria proporcionar un manual d'expressions regulars sinó oferir una introducció en tant que les expressions regulars han estat part de la implementació del programari que acompanya aquest document.

Capítol 4 El format ODF

4.1 Punt de partida

Actualment el format Open Document OASIS Standard **[3]** està estandarditzat per el comitè ISO/IEC segons la norma ISO/IEC 26300 i usat per multitud de programari lliure[2] per la qual cosa moltes organitzacions s'estan interessant en el seu ús, tant per la reducció de costs com perquè es pot usar independentment de plataforma.

Del format ODF existeixen dues versions: la 1.0 i la 1.1 essent la segona, segons la comunitat oficial, una revisió menor de la versió 1.0 que li aporta certes correccions, clarificacions i una revisió d'accessibilitats.

4.2 Tipus de documents suportats

El format ODF està destinat a tots els tipus de documents d'oficina disponibles: text, fulls de càlcul, presentació, imatges, bases de dades i fórmula matemàtica.

Com en el cas de la suite Office de Microsoft trobarem una extensió per a cadascun dels tipus de document. La relació de tipus i extensions és la següent:

Tipus	Extensió de l'arxiu	Tipus MIME
Document de text	.odt	application/vnd.oasis.opendocument.text
Full de càlcul	.ods	application/vnd.oasis.opendocument.spreadsheet
Presentació	.odp	application/vnd.oasis.opendocument.presentation
Imatge	.odi	application/vnd.oasis.opendocument.image
Base de dades	.odb	application/vnd.oasis.opendocument.database
Fórmula matemàtica	.odf	application/vnd.oasis.opendocument.formula
Gràfic	.odc	application/vnd.oasis.opendocument.chart
Dibuix	.odg	application/vnd.oasis.opendocument.graphics
Document mestres	.odm	application/vnd.oasis.opendocument.text-master

Fig. 17 - Tipus MIME dels documents OpenOffice

Altres tipus de documents són les plantilles:

Tipus	Extensió de l'arxiu	Tipus MIME
Document de text	.ott	application/vnd.oasis.opendocument.text-template
Full de càlcul	.ots	application/vnd.oasis.opendocument.spreadsheet-template
Presentació	.otp	application/vnd.oasis.opendocument.presentation-template
Dibuix	.otg	application/vnd.oasis.opendocument.graphics-template

Fig. 18 - Tipus MIME de les plantilles OpenOffice

[2] Definició de programari lliure a la Wikipedia

4.3 Formats competidors

Per a completar aquest apartat és necessari mostrar els formats competidors a ODF tant els directament competidors com el format de document digital Open XML com els que ho són a causa del seu ús com pot ser el format PDF / A del fabricant de software Adobe.

El format PDF / A es tracta d'un estàndard de document portable però amb certes limitacions, concretament pel que fa a la reproducció dels documents i al tractament que se'n pot fer. El fabricant Adobe Systems garanteix que un document basat en PDF / A serà accessible per la tecnologia apropiada fins d'aquí a cinquanta anys per la qual cosa no sembla ser un format adequat a la conservació de documents oficials en tant que la jurisprudència estableix que qualsevol document ha de poder ser consultat, preservant la seva integritat, en qualsevol moment futur **[4]**. D'altra banda el format és propietari de la marca i per tant el tractament digital d'aquests tipus de documents passen per adquirir el programari propietari necessari. Les seves principals avantatges són la independència de plataforma i que en l'actualitat el format es l'estàndard ISO/IEC 19005-1:2005 **[7]**.

Per la seva banda el format Open XML és el competidor directe del format ODF en tant que és un calc adaptat per Microsoft. Aquest format, segons alguns especialistes, pretén convertir-se en el format de document digital central a tots els processos de Microsoft i d'aquesta manera preservar el lligam dels usuaris finals a les plataformes de Microsoft **[5]**. Cal afegir aquest format va acompanyat de certa controvèrsia perquè l'abril de 2008 va rebre els vots necessaris per convertir-se en l'estàndard ISO/IEC DIS 29500 després d'haver estat rebutjat pel mateix comitè durant l'any 2007 a causa de que no reunia les mínimes condicions exigibles a un estàndard. Entre altres deficiències es va denunciar que l'especificació acceptava parts binàries el que es contradiu amb el format XML i representa que es poden afegir parts propietàries i dependents de plataforma **[8]**.

4.4 Breu història

L'origen del format ODF esdevé a l'any 1999 quan el fabricant de software SUN decideix deslligar-se dels formats propietaris de la suite ofimàtica de Microsoft. En aquesta decisió SUN pretén alleugerir costs en la seva cadena de producció, d'uns 50.000 llocs de treballs, en base a retallar el nombre de llicències. Per a aconseguir-ho fa un sondeig al mercat de la ofimàtica que li aconsella comprar una companyia alemanya anomenada StarDivision la qual estava desenvolupant una suite ofimàtica independent de plataforma.

La compra d'aquesta companyia però no va significar el naixement del format ODF sinó que va ser el lliurament del format usat per aquesta companyia, sota la llicència de codi obert, que origina el projecte, anomenat OpenOffice.org, i que té com a fites més importants l'estandardització del format per part del comitè Organization for the Advancement of Structured Information (OASI) el maig de 2005 i l'estandardització per part del comitè ISO/IEC el maig de 2006.

La cronologia del format es pot resumir en la següent taula facilitada per la revista informàtica Novatica **[6]**:

Fecha / Período	Acontecimiento / Hito
1999	Empieza en StarDivision el desarrollo de un formato de archivo predeterminado en XML. Las limitaciones de los antiguos formatos binarios y la necesidad de soporte de Unicode desencadenan el cambio. El objetivo es crear un formato de archivo abierto e interoperable que también pueda ser utilizado e implementado por otros fabricantes.
Agosto 1999	Sun Microsystems, Inc. adquiere StarDivision.
13 de octubre 2000	Sun Microsystems, Inc. publica el código abierto de StarOffice bajo licencias abiertas en el proyecto OpenOffice.org recientemente fundado (julio 2000).
13 de octubre 2000	Se establece en OpenOffice.org el proyecto comunitario XML con el objetivo de definir la especificación del formato de archivo XML OpenOffice.org como un esfuerzo de la comunidad abierta.
2002	Las definiciones para CJK (Chino, Japonés, Coreano) y otros idiomas con representaciones complejas se añaden a la especificación de formato OpenOffice.org XML.
2002	Se inicia la colaboración con el proyecto KOffice.
16 de diciembre 2002	El OASIS Open Office Technical Committee convoca su primera conferencia.
Mayo 2002	Se publican OpenOffice.org 1.0 y StarOffice 6. Ambos utilizan el formato de archivo OpenOffice.org XML como formato predeterminado.
Agosto 2003	KOffice decide utilizar ODF como formato predeterminado.
2003 / 2004	Se modifica la especificación original del formato OpenOffice.org XML para reflejar los últimos desarrollos en XML y en el área de aplicaciones ofimáticas: * Introducción de espacios de nombre conforme a las reglas de denominación de OASIS. * Cambio de las DTDs de XML a Relax-NG como lenguaje de esquema. * Mejoras en el esquema para soportar mejor la validación de documentos. * Adaptación del esquema para nuevas versiones de estándares. * Adaptación para otras aplicaciones ofimáticas (KOffice). * Adaptación para nuevas versiones de aplicaciones ofimáticas (OpenOffice.org 2.0). * Eliminación de inconsistencias en la especificación. * Corrección de errores.
Diciembre 2004	Se aprueba un segundo borrador del comité, cuyo título cambia de "OASIS Open Office Specification" a "OASIS Open Document Format Office Applications (OpenDocument)".
Enero 2005	El comité técnico cambia de nombre a OASIS Open Document Format for Office Applications (OpenDocument) TC.
Febrero 2005	El tercer borrador de la especificación de formato de archivo, incluyendo las reacciones de la evaluación pública, se aprueba como borrador del comité.
Mayo 2005	El OpenDocument Format (ODF) se aprueba como un estándar de OASIS.
Septiembre 2005	Sun Microsystems lanza StarOffice 8 con soporte ODF.
Septiembre 2005	ODF se envía a la Organización Internacional para la Normalización (ISO).
Septiembre 2005	El INdT (grupo de investigación perteneciente a Nokia) contribuye a ODF con filtros para Abiword y Gnumeric.
Octubre 2005	Se lanza OpenOffice 2.0 con soporte ODF.
Octubre 2005	Sun publica la siguiente declaración de convenio de patente: "La declaración pública de Sun de derechos por falta de afirmación puede resumirse de forma extraoficial como un acuerdo irrevocable para no aplicar ninguna de las patentes aplicables tanto de EE.UU. como del extranjero contra ninguna implementación de la especificación de OASIS OpenDocument". <http://xml.coverpages.org/ni2005-10-04-a.html>.
Diciembre 2005	Softmaker lanza Textmaker 2006 con soporte ODF.
Enero 2006	IBM lanza IBM Workplace con soporte ODF.
Marzo 2006	Se funda la ODF Alliance con 35 miembros iniciales para promover ODF en el sector público.
Marzo 2006	Se funda el OASIS ODF Adoption TC con el objetivo de educar al mercado sobre el valor de ODF.
Abril 2006	Se lanza KOffice 1.5, que utiliza ODF como formato predefinido.
Mayo 2006	ISO aprueba ODF como ISO/IEC 26300.
Junio 2006	La ODF Alliance ya tiene más de 200 miembros entre los que se incluyen empresas, organizaciones y ciudades como BBC, Corel EDS, EMC, IBM, Novell, Red Hat, Oracle, Software AG, Sun Microsystems, y la ciudad de Viena.
Septiembre 2006	La segunda edición de ODF 1.0 se completa con cambios editoriales identificados en el proceso de revisión de ISO.
Octubre 2006	Se aprueba ODF 1.1. como Especificación del Comité (Comittee Specification); está pendiente de ser enviado para ser sometido a votación como estándar OASIS en enero de 2007. Desarrollo continuo de fórmulas, accesibilidad y meta datos planeados para su publicación en 2007 como ODF 1.2. ODF Alliance supera los 300 miembros de más de 40 países.

Fig. 19 - La història del format ODF

4.5 Avantatges que aporta el format

L'ús del format ODF aporta al usuari final la independència de programari de base per a la generació, tractament i distribució de documents digitals. Això ratifica el pas de la informàtica dels formats propietaris a la informàtica distribuïda i lliure de la informació, on és possible no dependre de cap fabricant de software ni de cap aplicació propietària.

Per a poder realitzar aquest salt el format ODF aprofitarà com a base el format XML per a definir un format per al tractament de documents digitals de tota mena: arxius de text, presentacions, fulls de càlcul, bases de dades i gràfics.

D'altra banda el format aporta un tipus de llicència lliure el que permet ser usat lliure de regalies en múltiples propòsits com són:

- La generació i distribució de documents basats en aquest format
- L'automatització de processos de tractaments de documents basats en aquest format

Per tant, com s'ha esmentat en apartats anteriors, es disposa en l'actualitat de multitud de programes que fan un ús exhaustiu del format ODF donant la possibilitat de manegar documents portables independents de plataforma.

Completant aquesta idea l'Open Document dóna peu a les organitzacions governamentals a poder emmagatzemar els seus documents d'acord a les pròpies lleis que exigeixen poder guardar els documents amb un tipus de suport que permetin interpretar els mateixos amb qualsevol tecnologia que es disposi en un futur mantenint la seva integritat **[4]**.

4.6 Arquitectura del format

Per a implementar el format els creadors van establir dues màximes:

1. No tornar a inventar la roda i per tant reutilitzar els estàndards disponibles al mercat. Alguns d'aquests es resumeixen a la següent llista: XML, SGV, PNG, XSL i d'altres més específics com XForms, XLink o MathML.
2. Implementar un format portable de document digital. En aquest sentit es va pensar en basar-se en documents XML i empaquetar-los dins d'un contenidor que seria la compressió dels documents XML en un arxiu ZIP.

En la imatge següent, extreta de la revista Novatica **[6]**, es pot veure la composició interna d'un document de text en format Open Document.

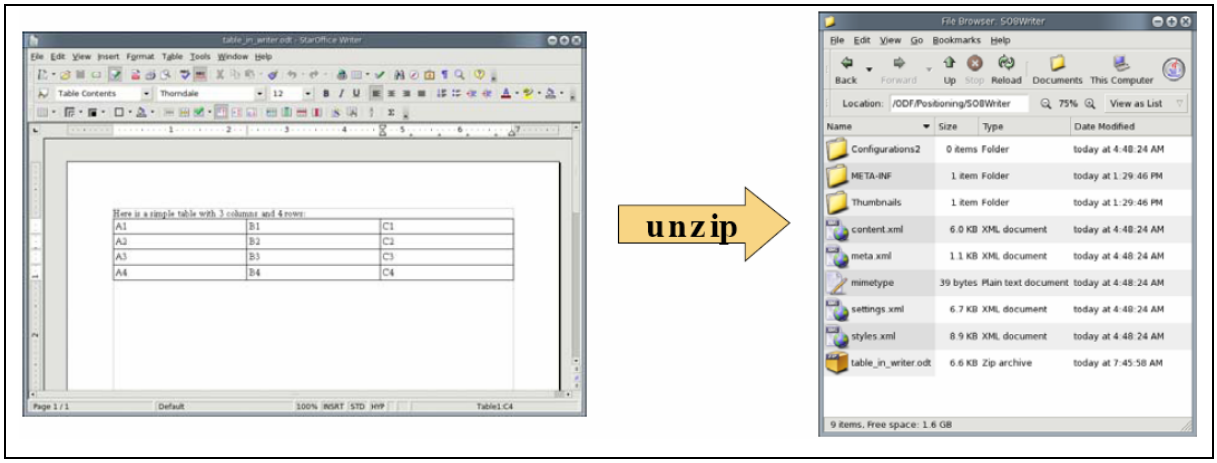


Fig. 20 - Document ODF

Concretament podem veure els documents principals del format: *content.xml*, *styles.xml*, *settings.xml* i *meta.xml*, *mimetype.xml* i els recursos relacionats organitzats en carpetes.

4.6.1 Arxiu content.xml

Entre tots els arxius el principal és el document anomenat *content.xml* que a part de contenir el contingut del document ODF aporta les referències als esquemes usats, les fonts i els estils que s'aplicaran:

```
<?xml version="1.0" encoding="UTF-8"?>
<office:document-content xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0" xmlns:style="urn:oasis:names:tc:op
                         xmlns:text="urn:oasis:names:tc:opendocument:xmlns:text:1.0" xmlns:table="urn:oasis:names:tc:opendo
                         xmlns:draw="urn:oasis:names:tc:opendocument:xmlns:drawing:1.0" xmlns:fo="urn:oasis:names:tc:opendo
                         xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:dc="http://purl.org/dc/elements/1.1/"
                         xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0" xmlns:number="urn:oasis:names:tc:openc
                         xmlns:svg="urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0" xmlns:chart="urn:oasis:names:
                         xmlns:dr3d="urn:oasis:names:tc:opendocument:xmlns:dr3d:1.0" xmlns:math="http://www.w3.org/1998/Mat
                         xmlns:form="urn:oasis:names:tc:opendocument:xmlns:form:1.0" xmlns:script="urn:oasis:names:tc:openc
                         xmlns:ooo="http://openoffice.org/2004/office" xmlns:ooow="http://openoffice.org/2004/writer" xmlns
                         xmlns:dom="http://www.w3.org/2001/xml-events" xmlns:xforms="http://www.w3.org/2002/xforms" xmlns:x
                         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" office:version="1.0">
    <office:scripts/>
    <office:font-face-decls>
        <style:font-face style:name="Tahoma1" svg:font-family="Tahoma"/>
        <style:font-face style:name="Times New Roman" svg:font-family="&apos;Times New Roman&apos;" style:font-family-gener
        <style:font-face style:name="Arial" svg:font-family="Arial" style:font-family-generic="swiss" style:font-pitch="var
        <style:font-face style:name="Lucida Sans Unicode" svg:font-family="&apos;Lucida Sans Unicode&apos;" style:font-fami
        <style:font-face style:name="MS Mincho" svg:font-family="&apos;MS Mincho&apos;" style:font-family-generic="system"
        <style:font-face style:name="Tahoma" svg:font-family="Tahoma" style:font-family-generic="system" style:font-pitch="
    </office:font-face-decls>
    <office:automatic-styles>
        <style:style style:name="P1" style:family="paragraph" style:parent-style-name="Standard">
            <style:text-properties fo:font-weight="bold" style:font-weight-asian="bold" style:font-weight-complex="bold"/>
        </style:style>
    </office:automatic-styles>
```

Fig. 21 - Arxiu de continguts (I)

Com es pot veure, a la figura anterior, al inici del document es fa referència a les definicions dels esquemes del format ODF usats i en un segon punt es poden observar les fonts particulars que s'usaran al document. Prosseguint amb el mateix arxiu trobem una entitat que té com a objectiu la definició dels estils, també particulars, que s'aplicaran al document de text, en aquest cas un per a representar els caràcters en negreta.

Per últim trobem el cos del document i per a cadascun dels elements els estils emprats:

```
    <office:body>
        <office:text>
            <office:forms form:automatic-focus="false" form:apply-design-mode="false"/>
            <text:sequence-decls>
                <text:sequence-decl text:display-outline-level="0" text:name="Illustration"/>
                <text:sequence-decl text:display-outline-level="0" text:name="Table"/>
                <text:sequence-decl text:display-outline-level="0" text:name="Text"/>
                <text:sequence-decl text:display-outline-level="0" text:name="Drawing"/>
            </text:sequence-decls>
            <text:h text:style-name="Heading_20_1" text:outline-level="1">Exemple de document OpenOffice</text:h>
            <text:p text:style-name="Standard">Hola he dit hola.</text:p>
            <text:p text:style-name="Standard"/>
            <text:p text:style-name="P1">Aquest text està en negreta.</text:p>
            <text:p text:style-name="P1"/>
            <text:p text:style-name="P1">
                <draw:frame draw:style-name="fr1" draw:name="gráficos1" text:anchor-type="paragraph"
                            svg:width="0.423cm" svg:height="0.423cm" draw:z-index="0">
                    <draw:image xlink:href="Pictures/10000018000000100000000105727C38A.gif"
                                xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad"/>
                </draw:frame>
            </text:p>
        </office:text>
    </office:body>
</office:document-content>
```

Fig. 22 - Arxiu de continguts (II)

4.6.2 Arxiu styles.xml

Aquest arxiu està destinat a contenir les fonts i els estils predeterminats, que no particulars, usats en el document Open Document. Quan es parla d'estil i font predeterminada ens referim als estils i fonts comunes a tots els documents ODF i que per motius d'eficiència es mantenen en un arxiu separat, l'arxiu *styles.xml*, al qual es fa referència des de l'arxiu *content.xml*.

Per adonar-nos de la diferència entre particular i predeterminat fixem-nos en aquestes dues entitats:

<text:p text:style-name="Standard">Hola he dit hola.</text:p>

<text:p text:style-name="P1">Aquest text està en negreta.</text:p>

La primera fa referència a un estil predeterminat ubicat a l'arxiu *styles.xml* mentre que la segona usa un estil particular ubicat a l'arxiu *content.xml*.

4.6.3 Arxiu settings.xml

Aquest arxiu està destinat a guardar els paràmetres de configuració del document entès com a aplicació, per exemple, la posició del cursor al obrir el document o bé el nivell de zoom.

4.6.4 Arxiu meta.xml

Conté les metadades del document. Per exemple el nom de l'autor del document, la identificació de la última persona que va modificar el document, etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<office:document-meta xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"
                      xmlns:xlink="http://www.w3.org/1999/xlink"
                      xmlns:dc="http://purl.org/dc/elements/1.1/"
                      xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0"
                      xmlns:ooo="http://openoffice.org/2004/office" office:version="1.0">
    <office:meta>
        <meta:generator>OpenOffice.org/2.3$Win32 OpenOffice.org_project/680m5$Build-9221</meta:generator>
        <meta:initial-creator>Jesús</meta:initial-creator>
        <meta:creation-date>2008-05-04T18:54:16</meta:creation-date>
        <dc:creator>Jesús</dc:creator>
        <dc:date>2008-05-04T19:37:48</dc:date>
        <meta:editing-cycles>3</meta:editing-cycles>
        <meta:editing-duration>PT1M5S</meta:editing-duration>
        <meta:user-defined meta:name="Info 1"/>
        <meta:user-defined meta:name="Info 2"/>
        <meta:user-defined meta:name="Info 3"/>
        <meta:user-defined meta:name="Info 4"/>
        <meta:document-statistic meta:table-count="0" meta:image-count="0" meta:object-count="0"
                                 meta:page-count="1" meta:paragraph-count="3" meta:word-count="13"
                                 meta:character-count="75"/>
    </office:meta>
</office:document-meta>
```

Fig. 23 - Arxiu de metadades

4.6.5 Arxiu mimetype.xml

Aquest arxiu conté una única línia que indica el tipus de document del que es tracta. La seva intenció és la de preservar el tipus de document encara que se'n modifiqui l'extensió.

Per exemple en el nostre cas que es tracta d'un document de text:

application/vnd.oasis.opendocument.text

4.6.6 Directori pictures

Conté les imatges usades en el document a les quals es fa referència des de l'arxiu content.xml. Per exemple, tornant a la part del cos de l'arxiu content.xml podem veure com hi ha una entitat paràgraf que conté una imatge. La imatge es denotada una marca *image* de l'espai de noms *draw* i que té un atribut *href* que fa referència a una imatge situada a la carpeta *Pictures*.

```
<text:p text:style-name="P1">
    <draw:frame draw:style-name="fr1" draw:name="gráficos1" text:anchor-type="paragraph"
                svg:width="0.423cm" svg:height="0.423cm" draw:z-index="0">
        <draw:image xlink:href="Pictures/10000018000000100000000105727C38A.gif"
                    xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad"/>
    </draw:frame>
</text:p>
```

Fig. 24 - Entitat imatge a ODF

A la següent imatge es pot veure la ubicació de la imatge dins del directori *Pictures*:

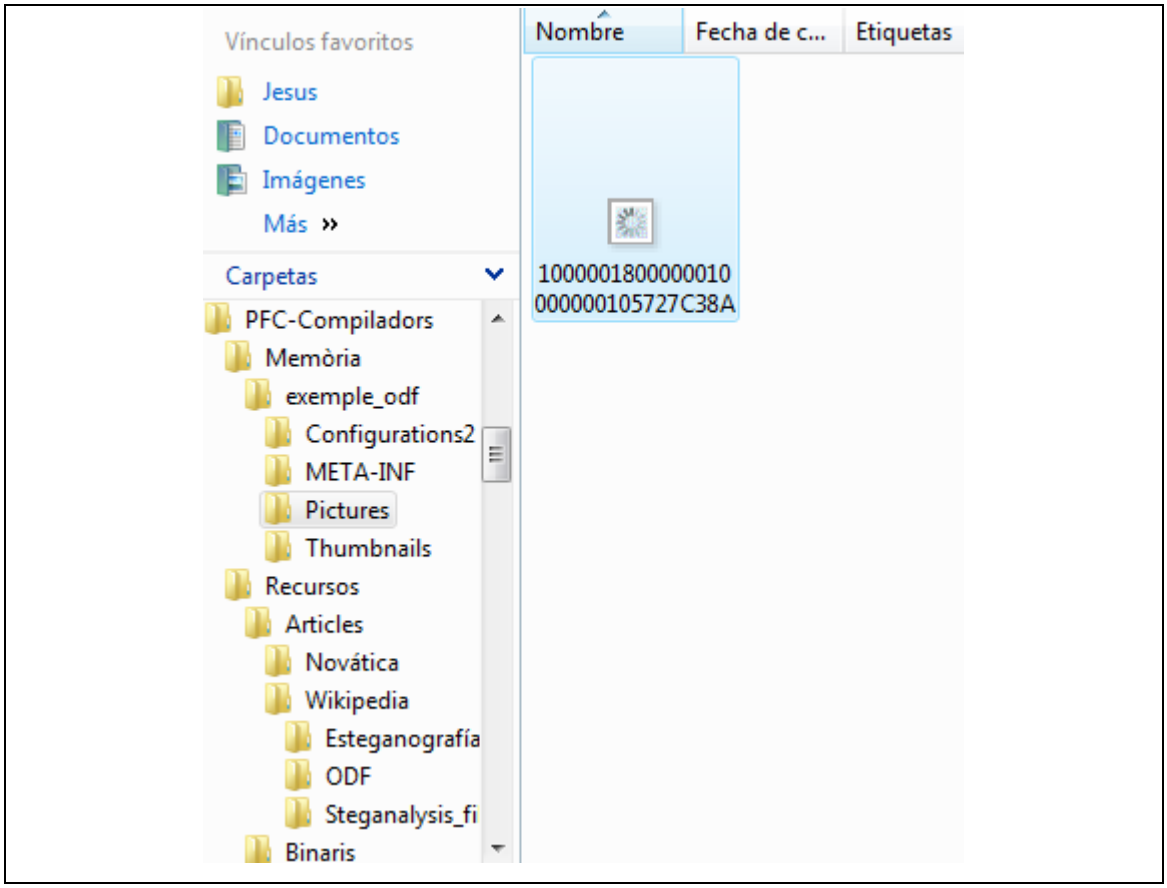


Fig. 25 - Directori Pictures

4.6.7 Entitats principals dels documents de text

Per a finalitzar la descripció de l'arquitectura presentarem les principals entitats que trobarem en un document de text, totes elles pertanyents a l'espai de noms *text* del format Open Document. Cal afegir que aquestes entitats poden ser usades en els processos d'automatització, per exemple, per a alterar l'estil de les mateixes, per inserir contingut, inserir entitats filles a altres entitats, etc i que

La llista de les entitats més importants és la següent:

- Paràgraf, és la entitat base del document de text i pot incloure tant text com altres entitats. L'estil del paràgraf, com s'ha vist abans, s'aplicarà al text contingut i pot ser heretat per les entitats filles. La marca usada per a definir un paràgraf és *<text:p/>*.
- Taula, aquesta entitat permet mostrar taules dins d'un document de text. La marca usada per a representar-la és *<text:table/>*.
- Fila, aquesta entitat permet mostrar files dins d'una taula. La marca usada per a representar-la és *<text:row/>*.

- Cel·la, representa una cel·la dins d'una fila. La marca usada per a representar-la és <*text:cell/*>.
- Llista, aquesta entitat permet representar llistes en un document de text. La marca usada per a representar-la és <*text:list/*>.
- Element de llista, aquesta entitat permet representar elements dins d'una llista. La marca usada és <*text:list-item/*>.

4.7 Comparatives amb els formats competidors

4.7.1 PDF / A

El format PDF / A és l'estàndard portable per a documents digitals, principalment destinat a la impressió dels mateixos.

Té com principals avantatges:

- És estàndard del comitè ISO/IEC
- És un format molt popular, primerament va ser un estàndard de facto i posteriorment va ser aprovat com a estàndard industrial
- Té capacitat per afegir-hi signatures digitals
- Pot incloure text, imatges i gràfics

Respecte als inconvenients:

- Està ideat per a ser un estàndard d'impressió en tant que qualsevol usuari final pot aconseguir visors gratuïts però si es vol treballar amb el document s'ha d'adquirir programari propietari.
- La propietat del format és de l'empresa Adobe Systems.
- Està limitat a documents de text.

ODF a més de proporcionar els mateixos avantatges resol els inconvenients ja que el format no és propietat de cap fabricant i per tant lliure de regalies, és obert i a més existeix molt programari obert que permet tant la impressió com modificació i admet, a part del format text, documents de tipus full de càlcul, presentacions, bases de dades i imatges.

4.7.2 Open XML

És el principal competidor perquè proposa una arquitectura similar. La diferència més notòria es que els documents XML que codifiquen els documents adopten una implementació en la que es barreja el contingut de les entitats i la seva presentació. Alguns experts atribueixen això a que possiblement la definició del format Open XML es una fotografia en XML dels formats propietaris de Microsoft.

Per exemple imaginem que volem representar el següent text en ambdós formats, ODF i Open XML:

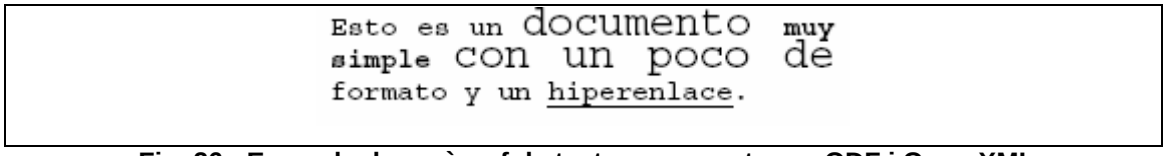


Fig. 26 - Exemple de paràgraf de text a representar en ODF i Open XML

En la figura representada a continuació es pot comprovar com la representació en format ODF es molt semblant a la representació d'un document HTML incorporant fulles d'estil i per tant molt senzilla d'interpretar per qualsevol desenvolupador:

```
<text:p text:style-name="Standard">
 Esto es un
    <text:span    text:style-
name="T1">documento</text:span>
 muy simple
    <text:span    text:style-
name="T2">con un poco de</text:span>
 formato y un
  <text:a  xlink:href="http://
www.EstandaresAbiertos.org">hiperenlace</
text:a>
</text:p>
```

Fig. 27 - Paràgraf en format ODF

En canvi en la figura següent es representa el mateix exemple en format Open XML que d'entrada resulta més difícil de comprendre:

```
<w:p>
  <w:r>
    <w:t>Esto es un </w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:b />
    </w:rPr>
    <w:t>documento</w:t>
  </w:r>
  <w:r>
    <w:t> muy simple </w:t>
  </w:r>
  <w:r>
    <w:rPr>
      <w:i />
    </w:rPr>
    <w:t>con un poco de</w:t>
  </w:r>
  <w:r>
    <w:t> formato y un </w:t>
  </w:r>
    <w:hyperlink  w:rel="rId4"
w:history="1">
    <w:r>
      <w:rPr>
       <w:rStyle w:val="Hyperlink"/>
      </w:rPr>
      <w:t>hiperenlace</w:t>
    </w:r>
  </w:hyperlink>
</w:p>
```

Fig. 28 - Paràgraf en format Open XML

En un altre ordre de coses Open XML s'està promocionant com el format que permetrà migrar els documents produïts amb els antics formats propietaris de Microsoft als nous formats.

A més Microsoft ha signat un acord de col·laboració amb Novell per desenvolupar una eina (Novell - Microsoft Clever Age) que permeti transformar documents Open XML a ODF però no a la inversa.

Respecte a aquest format ODF ofereix menys complexitat tal com acabem de veure en l'exemple anterior i al fet de que la definició del format Open Document ocupa 600 planes que contrasta amb les 4000 planes que ocupa la definició del format Open XML. Això últim es font de divergències entre els especialistes perquè alguns postulen que aquest volum dispar en les definicions implica una complexitat proporcional al volum mentre que altres opinions justifiquen aquest volum al nivell de detall i quantitat d'exemples introduïts.

Pel que fa als avantatges que ofereix ODF respecte a Open XML el primer és la maduresa del producte en tant que ODF té un cicle de vida de més de 8 anys mentre que segon format va néixer fa 2 anys. Una segona avantatge és la quantitat de programari que usen ODF en oposició al nombre d'aplicacions que fan ús del segon format. Aquest punt però no és totalment cert ja que des de la creació del format Open XML han estat nombrós el número d'aplicacions que fan ús d'aquest format, potser sí que es pot argumentar que aquestes pateixen de no complir amb la totalitat de la definició del format Open XML però aquest és un tema que queda fora de l'abast d'aquest projecte **[11]**.

4.7.3 Llibreries de desenvolupament

Actualment la fundació OpenOffice.org disposa d'eines per a ajudar al desenvolupament d'aplicacions que treballin amb el format Open Document. Aquestes llibreries inclouen la API del format i d'altres específiques que ofereixen objectes d'automatització per a realitzar tasques concretes com per exemple accedir i treballar amb el text d'un document.

La llista de programari i documentació és extensa però aquí nombrarem les que creiem més interessants, sense oblidar una altre font molt valuosa que és la plana WIKI d'OpenOffice.org i la llista de distribució de cadascuna de les eines:

- The OpenOffice.org API Project
- The OpenOffice.org ODF Toolkit Project
- ODF Toolkit (Wiki)
- OpenDocument for Java - odf4j (wiki)
- AODL (An Open Document Library)

De la llista anterior destacarem les dues últimes referències ja que proporcionen un conjunt de classes que permeten l'accés als documents en format ODF i la majoria d'operacions disponibles per a desenvolupar processos d'automatització sobre documents.

La primera de les llibreries, l'ODF4J, està desenvolupada sobre plataforma Java mentre que la segona, l'AODL, està desenvolupada sobre plataforma .NET, concretament en llenguatge C#.

4.8 Estat actual

Per tot el que hem vist fins ara preveiem una guerra de formats amb l'objectiu d'acaparar la major part del pastís. Els actors principals són clars: per una banda Microsoft juntament amb els seus aliats i per l'altra les empreses que formen part del comitè OASI. L'objectiu el pastís format per les organitzacions, principalment les governamentals, la gran majoria de les quals estan usant els formats propietaris de Microsoft.

La competència segons alguns autors és bona ja que tendeix a reduir costs i a aconseguir beneficis tecnològics a curt i mitjà termini, mentre que per altres la existència de més d'un format per a un mateix propòsit pot suposar més entrebancs que millores ja que cadascun mirrarà de posar impediments a la resta.

En un altre ordre de coses el fet de que Microsoft estigui desenvolupant totes les seves aplicacions al voltant del format Open XML pot suposar una barrera d'entrada per als formats competidors. Això és molt probable si apliquem la teoria de que els directors d'IT tendeixen a escollir aquelles tecnologies que els facilitin l'adaptació dels seus processos enfront d'aquelles que suposin un trasbals. Per tant la idea de que les organitzacions escullin Open XML per les garanties que dóna durant la els processos de refactorització pot fer que, tot i estar usant un format de document estàndard, aquestes continuïn depenent d'un fabricant de programari.

4.9 Amenaces de seguretat

La cloenda d'aquest capítol no es pot fer sense criticar el format Open Document. Això no implica que es vulgui desprestigiar el format ODF ni el contrari sinó, segons la opinió del que escriu, exposar possibles problemes i / o falses creences sobre aquest format.

Després d'haver llegit la definició del format ODF i en part la d'Open XML creiem que no es pot asseverar que el format ODF estigui lliure de parts propietàries en tant que es poden incrustar objectes OLE i parts dependents de fabricant gràcies als espais de noms del format XML com bé relata Marco Fioretti **[4]**.

La incrustació d'aquetes parts permetria la compatibilitat amb formats binaris durant els processos de migració, cosa que creiem oportuna, mentre que la incrustació de parts propietàries, com la iniciativa del format per a certificats mèdics del consorci MedBiquiotous[3], suposaria una alteració de les propietats del format.

Un altre fet és la possibilitat de xifrar totalment o en part el contingut d'un document. Aquesta propietat que en part suposa el poder ocultar parts de documents confidencials també pot ser negativa si es destina a un ús fraudulent com pot ser afavorir l'espionatge o el terrorisme.

[3] http://www.medbiq.org/

Prosseguint amb aquesta última idea es pot fer ús de l'esteganografia per a ocultar informació compromesa dins d'imatges incrustades dins d'un document Open Document. Una altra possibilitat seria emmagatzemar aquesta informació dins dels documents XML que formen el document ODF però això es podria resoldre aplicant cerques basades en patrons com la proposada en aquest projecte.

Aquests processos de cerca formen part del camp de l'esteganàlisi que consisteix en la detecció de tècniques d'esteganografia sobre documents.

Capítol 5 Enginyeria del software

5.1 Especificació i anàlisi

El procés d'especificació permet plasmar d'una manera oficial les característiques d'allò que es vol obtenir, indicant les característiques i tot el necessari per a assolir-ho.

Per tant en un projecte informàtic l'especificació detallarà les característiques que defineixen el sistema, els actors que hi participen i opcionalment indicarà els components que el conformen[4].

D'altra banda necessitarem un punt de partida per a començar el procés d'especificació i aquest correspon a les necessitats d'aquells que ens encomanen la tasca de construir el sistema. Complementàriament nosaltres em de saber escoltar aquestes necessitats i proporcionar allò que pugui donar un valor afegit al sistema.

En el nostre cas el punt de partida serà l'enunciat que motiva aquest projecte i que ens permetrà especificar:

- L'anàlisi de requeriments
- Els actors del sistema
- Els components que conformen el sistema

5.1.1 Anàlisi de requeriments

En un projecte tradicional aquesta etapa correspondria a descriure els requeriments funcionals i no funcionals del sistema a partir d'entrevistes amb el client però a causa de les limitacions que imposa el marc de treball d'aquest projecte entendrem com a necessitats del sistema les funcionalitats requerides en l'enunciat tal com s'ha comentat a l'apartat anterior.

En aquest sentit definirem en primer lloc els requeriments funcionals que definiran el comportament del sistema i en segon lloc els requeriments no funcionals que imposen restriccions al disseny o en el funcionament del sistema tals com requisits de funcionament, estàndards de qualitat o requisits en el disseny.

[4] En alguns sistemes inclús s'han especificat els components després de l'etapa d'implementació.

5.1.1.1 Requeriments funcionals

A partir de l'anàlisi de l'enunciat proposat pel projecte entenem que s'han de proporcionar les següents característiques:

- Un procediment automàtic de tractament de documents de text OpenOffice a partir d'uns patrons proporcionats.
- Implementar un magatzem de dades en format XML i un DTD que el validi. Aquest magatzem serà usat per l'Agència Mundial de Seguretat per a la transmissió de dades sospitoses.
- Un procés que indiqui si les dades extretes del document coincideixen o no amb dades sospitoses del magatzem de l'Agència Mundial de Seguretat.
- Generar un document XML i el DTD que el valida on guardar els resultats anteriors.
- Generar un document HTML mitjançant transformació XSL del document XML anterior. Aquest document HTML està destinat a la presentació dels resultats a l'usuari de l'aplicació.

5.1.1.2 Requeriment no funcionals

Com a requeriments no funcionals entenem que hi tenen cabuda tots aquells que aportin un valor afegit:

- Proveir un mecanisme d'actualització dels patrons a usar per l'eina.
- Proveir un mecanisme d'actualització de magatzems de l'Agència Mundial de Seguretat en tant que s'entén que aquesta organització enviarà de forma regular noves versions del magatzem.
- Gestió d'errors de forma centralitzada per a facilitar la informació a l'usuari en cas d'error.
- Notificar la inexistència dels arxius necessaris per a la validació dels documents de text en cas que procedeixi.
- Proporcionar una arquitectura que permeti l'anàlisi d'altres tipus de formats de la llibreria ofimàtica OpenOffice o similars.

5.1.2 Actors del sistema

Els actors del sistema, si ens limiten a aquells que en faran ús, seran totes aquelles persones que necessitin validar documents de text OpenOffice, però també podrien ser els administradors que mantenen els magatzems de dades sospitoses de l'aplicació o l'arxiu de patrons.

5.2 Disseny

Aquesta fase té com a objectiu definir un plànol que representi o modeli el sistema proporcionant una estructura de dades, arquitectura, etc. que a usar durant la darrera fase d'implementació. En aquesta fase podrem aportar el valor afegit al sistema (o qualitat del software) i es poden fer les millores pertinents si es fa necessari sense requerir proves o als clients de la fase d'especificació.

Per tant el següent apartat tindrà com a objectiu presentar les estructures de dades que compondran el sistema i les seves interaccions.

En un altre ordre de coses en aquesta etapa del cicle de vida també es decideix amb quines tecnologies es treballaran. De forma genèrica hom podria decidir quina infraestructura de xarxa s'usarà, la plataforma de software amb la que s'implementarà la solució, el maquinari que serà necessari, etc.

En el nostre cas el projecte està centrat en l'àrea del programari i per tant pel que fa al maquinari només serà necessari especificar el que s'usarà per a implementar el producte i aquell on pugui ser executat.

En referència a la plataforma de desenvolupament s'han considerat la plataforma Java de SUN i la plataforma .NET de Microsoft. La motivació per aquesta tria entre altres llenguatges i arquitectures es deu a l'existència d'unes llibreries de desenvolupament d'OpenOffice que permeten el tractament d'arxius en format OpenOffice i en especial OpenDocument format requerit en aquest projecte.

El descobriment i estudi d'aquestes llibreries es va fer durant l'etapa d'investigació del projecte tal i com es pot veure a l'apartat 'Llibreries de desenvolupament' del capítol dedicat al format ODF i la seva utilització es deu als següents criteris:

- Suporten tots els formats de la família OpenOffice.
- Ofereixen una API per al tractament de documents en qualsevol format OpenOffice.
- Redueixen el cost de desenvolupament de qualsevol aplicació que hagi de tractar documents OpenOffice.
- Usen el paradigma de l'orientació a objectes per a representar les diferents entitats d'un document ODF com són els elements de text: paràgraf, taula, llista, etc.
- Estan implementades mitjançant les plataformes Java i .NET amb el que es pot fer ús de les classes disponibles en cadascuna de les arquitectures respectivament.
- Totes dues estan supervisades per l'equip de desenvolupament d'OpenOffice.
- Són llibreries de codi obert i lliures.

En un altre ordre de coses, en aquesta etapa és quan s'ha d'optar per una de les dues tecnologies i ho farem a partir de l'experiència de programació, la portabilitat que ofereixen, etc.

En el meu cas, tot i que l'experiència sobre cadascuna de les plataformes per a fer una aplicació d'escriptori és similar, el fet de que actualment desenvolupi aplicacions sobre arquitectura .NET i la disponibilitat de productes lliure de costs com són les versions Express del Visual Studio 2008 m'ha portat a escollir la plataforma .NET com a tecnologia sobre la que desenvolupar el producte.

Abans d'acabar l'apartat m'agradaria fer un petit resum de les diferències o similituds que presenten per a que el lector pugui tenir una visió general de cadascuna:

- Totes dues es tracten d'una arquitectura amb implementacions de patrons de disseny estàndard, accés a sistema d'arxius, tractament d'excepcions, orientades a objecte, etc.
- Totes dues ofereixen portabilitat tot i que el desenvolupament fet sobre la plataforma Java és superior i per tant més portable en la actualitat. La plataforma .NET de Microsoft és exclusiva d'entorns Windows i en versions reduïdes en entorns Mac. Tot i així Novell està desenvolupant una arquitectura bastant completa de l'arquitectura .NET anomenada Mono. Les principals virtuts d'aquest projecte és que es tracta de programari lliure, veritablement portable a sistemes no Windows com són Mac, Linux, FreeBSD, etc.

- El desenvolupament sobre Java i .NET es bastant similar en quan a dificultat al oferir pràcticament les mateixes funcionalitats. Específicament totes dues arquitectures ofereixen classes per a la creació d'expressions regulars, tractament d'arxius XML i transformacions que ens han estat imprescindibles per a desenvolupar l'aplicació del projecte.

Com veiem doncs l'elecció final venia determinada bàsicament per l'experiència tot i que la plataforma Java actualment té una avantatge destacable com és la portabilitat en detriment de la .NET.

5.2.1 Arquitectura del sistema

5.2.1.1 Disseny del sistema

El sistema proposat per a l'eina de programari objectiu d'aquest projecte seguirà la arquitectura de tres capes tradicional:

- Una capa de presentació que gestionarà la interacció de l'usuari amb l'eina.
- La capa de lògica que conté totes les regles de negoci.
- Una capa de persistència basada en el sistema d'arxius del sistema operatiu.

A partir dels requeriments funcionals s'ha entès que és necessari oferir un mecanisme de creació de classes validadores sense que el sistema hagi de preocupar-se de com crear-los. D'aquesta manera podrem gestionar en un futur nous tipus de documents. Per a poder acomplir aquesta necessitat s'ha aplicat el patró de disseny factoria **[12]**.

En un altre ordre de coses trobem necessari l'existència d'una classe gestora que proporcioni un punt d'entrada a la interfície d'usuari oferint-li tots els mètodes necessaris. D'aquesta manera la creació de la instància validadora, el tractament del procés i la generació dels resultats romanen a la capa de lògica del sistema separant-la de la capa de presentació.

Per completar la definició, tal com s'ha comentat al principi de l'apartat, la capa de persistència farà ús del sistema d'arxius del sistema operatiu i en primer terme aprofitarà les classes d'accés a arxius que proporcioni el llenguatge de programació escollit per a la implementació.

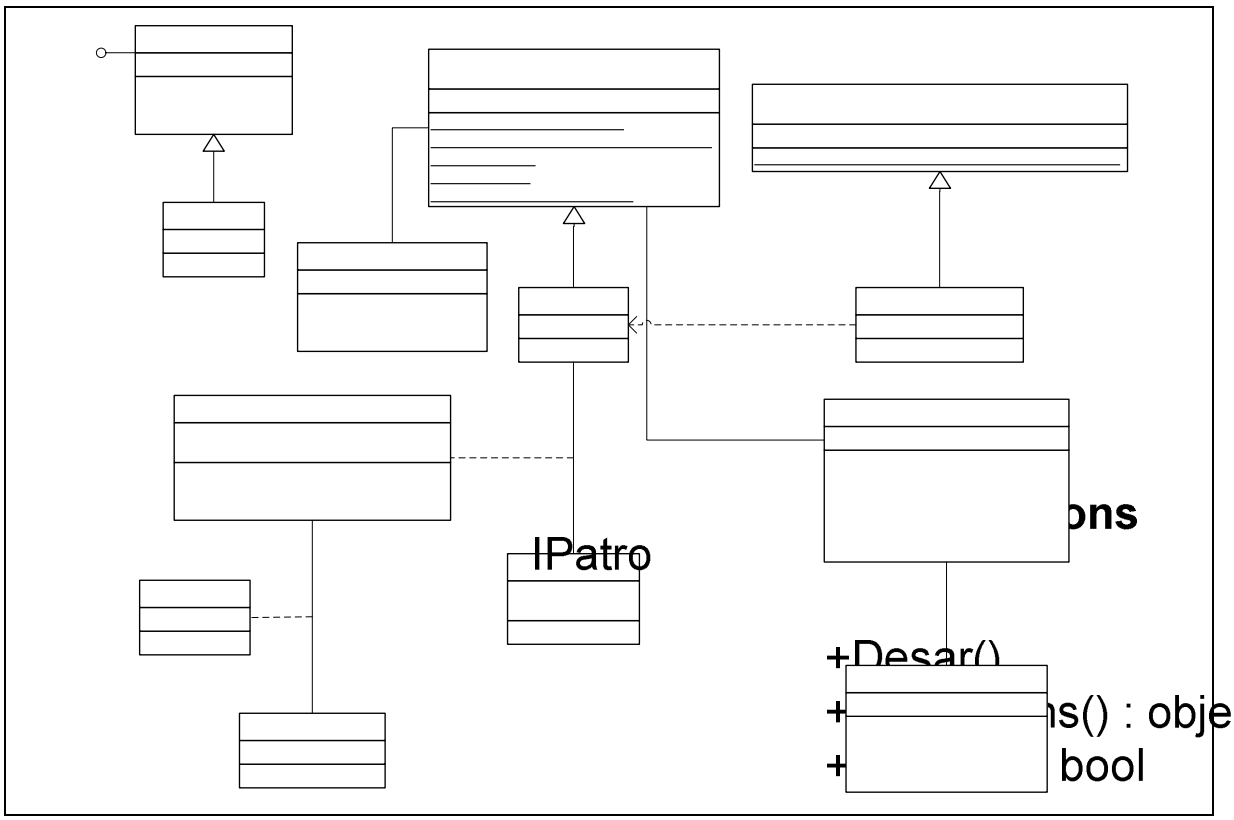


Fig. 29 – Especificació

5.2.2 Disseny de la interfície d'usuari

5.2.2.1 Finestra principal

Segons l'apartat d'especificació els usuaris poden comprendre tant usuaris finals com administradors de l'eina i en tots dos casos volem que el programa es presenti com quelcom integrat i de fàcil ús.

En aquest sentit s'ha considerat l'ús d'un component que ens permeti presentar als usuaris finals els resultats de les seves validacions però que alhora permeti presentar els patrons i el magatzem de l'Agència Mundial de Seguretat que s'està usant. Aquest component podria ser un navegador en tant que en tots tres casos tractem amb documents XML que poden ser transformats en HTML.

En la següent figura es mostra una possible implementació de la finestra principal: senzilla però alhora permet l'accés a totes les dades necessàries. La finestra es composa d'un menú amb les següents opcions:

- Arxiu, amb el qual poder validar i desar el resultat de la validació.
- Eines, amb el qual administrar l'aplicació.
- Ajuda, amb el qual obtenir informació de com usar l'eina i de la mateixa eina.

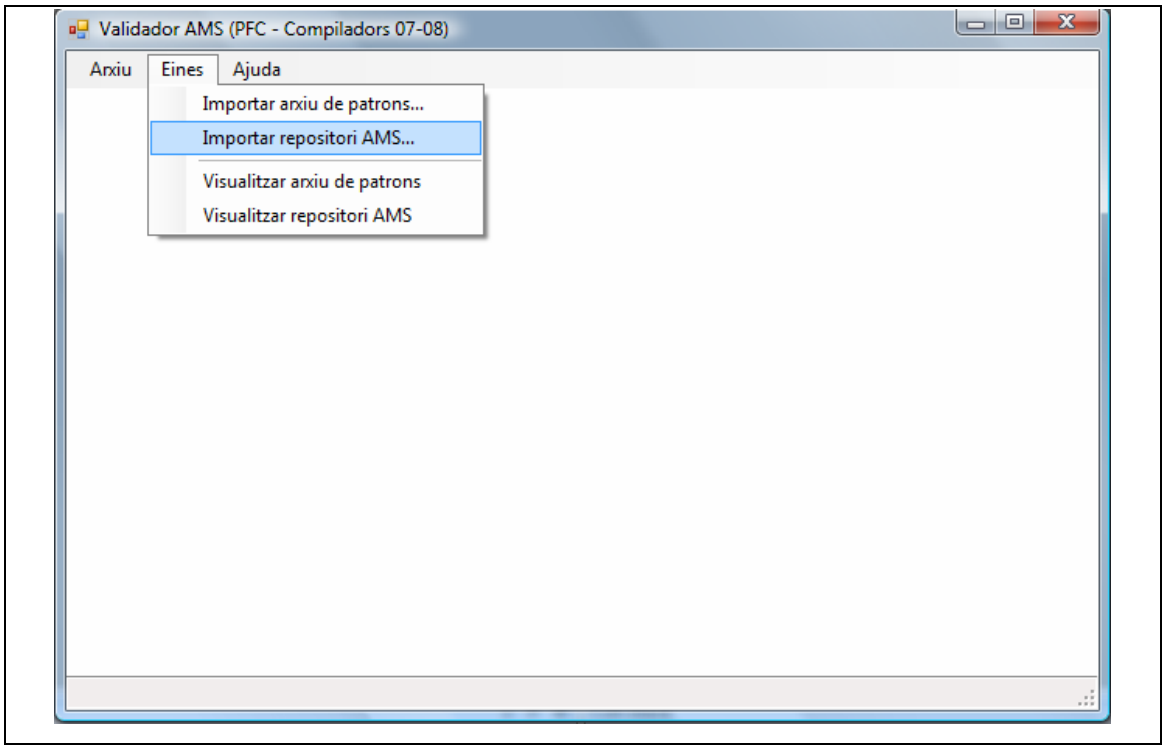


Fig. 30 - Finestra principal

A la part baixa de la finestra tenim una barra d'estat que ens informa del document que s'està validant o bé del document de patrons o magatzem de dades sospitoses carregat durant el manteniment dels mateixos. Es pot veure un exemple tot seguit:

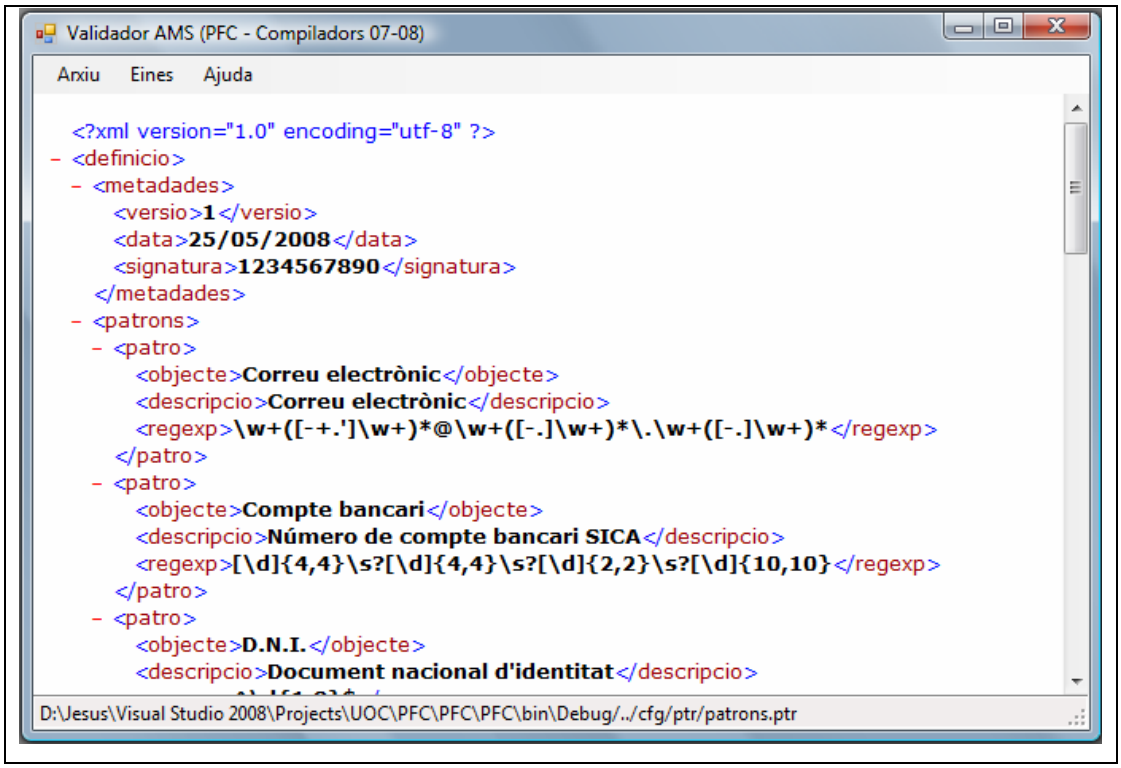


Fig. 31 - Component Web i barra d'estat mostrant un arxiu de patrons

5.2.2.2 Gestió d'arxius

Un dels requeriments no funcionals de l'etapa d'especificació és la reutilització de components per a facilitar l'ús del programa i també la implementació.

Com que en els processos de validació, manteniment de patrons i manteniment del magatzem de l'Agència Mundial de Seguretat tenen en comú un primer pas que és la selecció del document afectat s'ha estimat oportú crear un diàleg genèric que actuï com a gestor d'arxius i que es pugui adaptar. Aquesta adaptació consistiria en especificar una descripció i títol adequat a cadascun dels casos anteriors.

A continuació es mostra el diàleg genèric esmentat al paràgraf anterior. Com podem comprovar la finestra principal obre el diàleg genèric i aquest a la vegada, si es requerit per l'usuari, podrà fer ús d'un diàleg de gestió d'arxius proporcionat pel sistema operatiu.

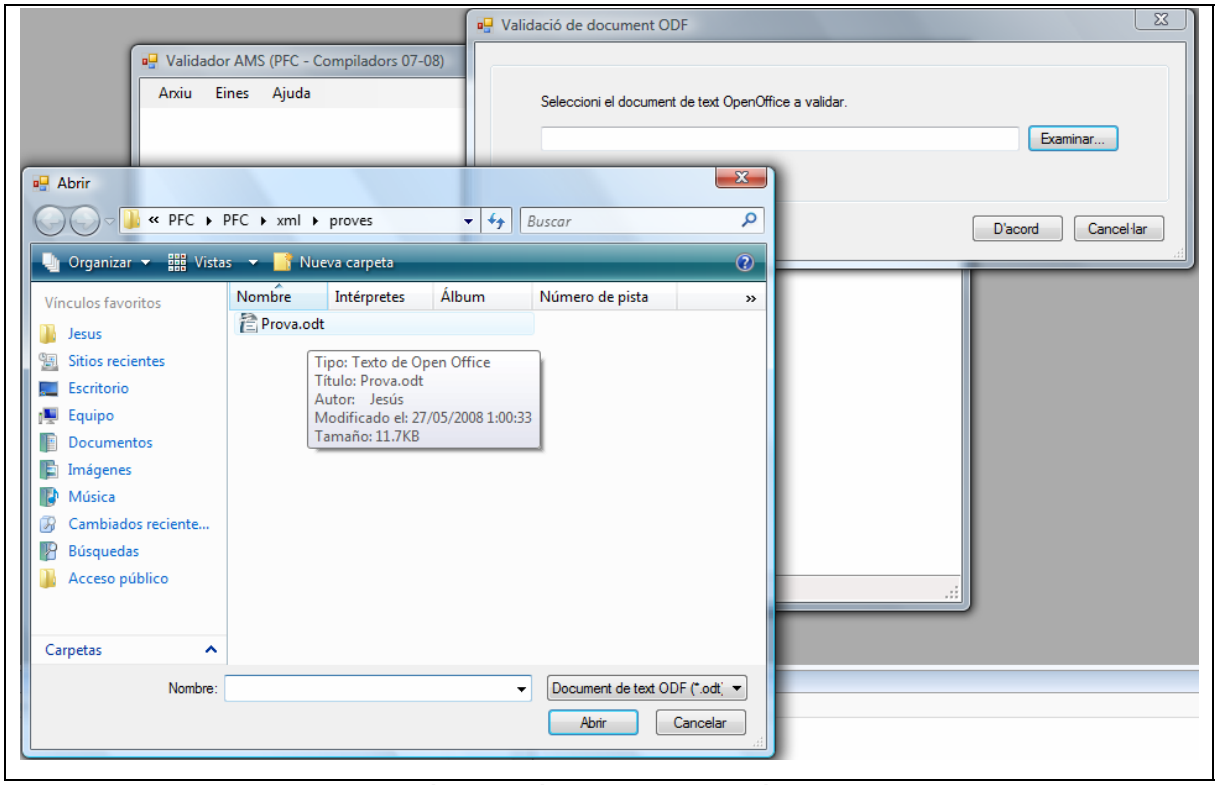


Fig. 32 - Diàleg genèric per a la gestió d'arxius

5.2.3 Disseny dels arxius XML de l'aplicació

Per acabar l'etapa de disseny és necessari definir els diferents arxius que ens serviran per a permetre la persistència de les dades. Aquestes dades comprenen tant els resultats com els patrons o el magatzem de dades proporcionat per l'Agència Mundial de Seguretat.

Cal afegir que l'enunciat només indica que s'hauran de definir els DTD que validen els diferents arxius en format XML i la implementació es deixa lliure a l'estudiant. Per tant el disseny proposat és una possible solució i no té perquè ser la única.

En aquest sentit s'ha optat per unes estructures el més senzilles possibles i formades per dues seccions:

1. Una primera secció que contindrà informació relativa al document com la data de creació, la versió, signatura digital per a verificar la integritat del document, etc.
2. Una segona secció amb el contingut del document.

5.2.3.1 Magatzem AMS

Aquest document en format XML ha de permetre la transmissió, primerament mitjançant correu electrònic, de dades sospitoses per part de l'Agència Mundial de Seguretat.

Tal com s'ha comentat a l'apartat anterior s'ha cregut oportú usar una estructura diferenciada en dues seccions: una que contingui la informació relativa al document i una segona amb el contingut pròpiament del document.

El DTD que validarà aquest document és el següent:

```
<?xml version="1.0" encoding="utf-8" ?>

<!ELEMENT Repositori (Metadades, DadesSospitoses)>

<!ELEMENT Metadades (Autor, Data, Versio, Descripcio, Signatura)>

<!ELEMENT DadesSospitoses (DadaSospitosa*)>

<!ELEMENT Autor (#PCDATA)>
<!ELEMENT Data (#PCDATA)>
<!ELEMENT Versio (#PCDATA)>
<!ELEMENT Descripcio (#PCDATA)>
<!ELEMENT Signatura (#PCDATA)>

<!ELEMENT DadaSospitosa (#PCDATA)>
```

Fig. 33 - DTD magatzem de l'AMS

Com es pot veure aquest document de definició requereix tots els valors excepte el en llistat de dades sospitoses que podria ser una llista buida. Això s'indica amb el caràcter '*' tot i que aquest cas sembla molt difícil d'aconseguir.

A continuació es presenta una taula amb el significat de cadascun dels elements principals:

Element	Significat
Metadades	Informació relativa al document
Autor	Organització que ha creat el document
Data	Data de la creació del document
Versió	Versió diària del document
Descripció	Propòsit del document. Podria haver-hi diversos com en el cas dels certificats digitals, CRL, etc.
Signatura	Signatura digital per a validar la integritat del document.
DadesSospitoses	Llista de dades sospitoses.
DadaSospitosa	Literal de dada sospitosa.

Taula 1 - Elements del DTD del magatzem de l'AMS

5.2.3.2 Implementació de l'arxiu de sortida XML

L'arxiu de sortida, segons l'anàlisi funcional, ha de proporcionar totes les dades resultants extretes del document OpenOffice, tant si la comparació amb les dades de l'AMS ha donat resultat positiu o no.

Continuant amb el mateix criteri que en el cas anterior s'ha estimat oportú usar una estructura diferenciada en dues seccions: una que contindrà informació relativa al document i una segona amb les dades pròpiament del resultat.

El DTD que validarà aquest document és el següent:

```
<?xml version="1.0" encoding="utf-8" ?>

<!ELEMENT Validacio (Metadades, Resultats)>

<!ELEMENT Metadades (NomDocument, DataValidacio)>
<!ELEMENT Resultats (Resultat*)>

<!ELEMENT NomDocument (#PCDATA)>
<!ELEMENT DataValidacio (#PCDATA)>
<!ELEMENT Resultat (Paragraf, Text, Patro)>
<!ELEMENT Paragraf (#PCDATA)>
<!ELEMENT Text (#PCDATA)>
<!ELEMENT Patro (#PCDATA)>
```

Fig. 34 - DTD del document de sortida

A continuació es presenta una taula amb el significat de cadascun dels elements principals:

Element	Significat
Metadades	Informació relativa al document
NomDocument	Nom del document de text OpenOffice validat
DataValidacio	Data de la validació
Resultats	Llista de dades resultants extretes
Resultat	Cadascuna de les dades resultants extretes
Paragraf	Número de paràgraf on es trobava la dada resultant
Text	La dada sospitosa
Patro	Identificador del patró del magatzem de l'AMS

Taula 2 - Elements del DTD de l'arxiu de sortida

5.2.3.3 Arxiu de transformació XSL

Per últim es defineix l'arxiu de transformació que permetrà generar a partir del document en format XML anterior un nou document en aquest cas en format HTML per a mostrar-lo a l'usuari que valida el document.

Abans de mostrar el document de transformació explicarem com s'ha articulat.

En un primer terme trobem la definició de document XML i a posteriori la de document de transformació.

A continuació s'ha implementat l'esquema d'una plana HTML que conté un títol i seguidament una taula. Aquesta taula contindrà dues taules més:

- Una primera que ens servirà per representar la informació del document validat i de la validació
- Una segona taula que detallarà els resultats de la validació

Com es pot veure les plantilles "Metadades" i "Resultats" tractaran la ocurrència homònima durant el processament del document XML.

En el cas de la plantilla "Resultat" trobem una bifurcació segons el resultat ha estat positiu o negatiu. Si el resultat es positiu la fila es farà notar, tal com diu l'especificació funcional, mitjançant un color de fons vermell i l'ús d'una etiqueta flotant[5] informativa.

El document de transformació XSL és el següent:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="Validacio">
    <html>
      <head>
        <title>Resultats de la validació</title>
      </head>
      <body>
        <font style="font-size:130%; font-weight:bold;">
          Resultats de la validació
        </font>
        <br></br>
        <br></br>
```

[5] En anglès *tooltip*.

```
        <table style="height:100%; width:100%;">
          <tr>
            <td>
              <font style="font-size:100%; border-bottom: 1px solid
                   #000000; width:100%;">
                    Dades del document tractat
              </font>
            </td>
          </tr>
          <tr>
            <td>
              <table style="width:100%">
                <xsl:apply-templates select="Metadades">
                </xsl:apply-templates>
              </table>
            </td>
          </tr>
          <tr>
            <td valign="bottom" style="height:50px;">
              <font style="font-size: 100%;
                    border-bottom: 1px solid #000000; width:100%;">
                  Dades compromeses detectades
              </font>
            </td>
          </tr>
          <tr>
            <td valign="top">
              <table style="height:100%; width:100%">
                <tr style="background-color: #CCCCCC;">
                  <td align="center"
                      style="font-weight:bold;">Paràgraf</td>
                  <td align="center" style="font-weight:bold;"
                      title="Positiu si la dada está
                      catalogada com a sospitosa per la AMS">Detecció
                  </td>
                  <td align="center" style="font-weight:bold;">Text</td>
                  <td align="center" style="font-weight:bold;">Patró</td>
                </tr>
                <xsl:apply-templates select="Resultats">
                </xsl:apply-templates>
              </table>
            </td>
          </tr>
        </table>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="Metadades">
    <tr>
      <td width="175px" style="font-weight:bold;">Document</td>
      <td>
        <xsl:value-of select="NomDocument"></xsl:value-of>
      </td>
    </tr>
    <tr>
      <td width="175px" style="font-weight:bold;">
         Data de la validació
      </td>
      <td>
        <xsl:value-of select="DataValidacio"></xsl:value-of>
      </td>
    </tr>
  </xsl:template>
  <xsl:template match="Resultats">
    <xsl:apply-templates select="Resultat"></xsl:apply-templates>
  </xsl:template>
  <xsl:template match="Resultat">
    <tr>
```

```
      <xsl:choose>
        <xsl:when test="DataSospitosaAMS = 'True'">
          <tr>
            <td align="center" style="background-color:#F5DEB3"
                 title="Aquesta dada està catalogada com a sospitosa per
                        l'AMS">
               <xsl:value-of select="Paragraf"></xsl:value-of>
            </td>
            <td align="center" style="background-color:#F5DEB3"
                 title="Aquesta dada està catalogada com a sospitosa per
                       l'AMS">
              Positiva
            </td>
            <td style="background-color:#F5DEB3"
                 title="Aquesta dada està catalogada com a sospitosa per
                        l'AMS">
              <xsl:value-of select="Text"></xsl:value-of>
            </td>
            <td style="background-color:#F5DEB3"
                title="Aquesta dada està catalogada com a sospitosa per
                      l'AMS">
              <xsl:value-of select="Patro"></xsl:value-of>
            </td>
          </tr>
        </xsl:when>
        <xsl:otherwise>
          <tr>
            <td align="center">
              <xsl:value-of select="Paragraf"></xsl:value-of>
            </td>
            <td align="center">
              Negativa
            </td>
            <td>
              <xsl:value-of select="Text"></xsl:value-of>
            </td>
                <td>
              <xsl:value-of select="Patro"></xsl:value-of>
            </td>
          </tr>
        </xsl:otherwise>
        </xsl:choose>
      </tr>
  </xsl:template>
</xsl:stylesheet>
```

Fig. 35 - Document de transformació XSL

5.3 Implementació

En aquest capítol s'explicaran els aspectes més tècnics del projecte ajudant-nos de fragments de codi de l'aplicació o estats de l'aplicació en format gràfic per a facilitar les descripcions.

Al final del capítol trobarem un apartat específic de proves on es mostraran diferents exemples usats en la fase de test del programa.

5.3.1 Aspectes tècnics

La arquitectura .NET es basa en arxius anomenats assemblats generats a partir del codi font de les aplicacions. Un assemblat és similar a una llibreria d'enllaç dinàmic però difereix en que aporta un arxiu de manifest que permet conèixer els tipus que són continguts. Això permet a altres assemblats de fer-hi referència i usar els tipus continguts.

Prosseguint amb aquesta definició trobem els executables que són idèntics als assemblats anteriors però aporten un punt d'entrada que permet la seva execució. En aquest cas és un mètode estàtic i públic anomenat Main al qual se li poden passar paràmetres d'entrada i que executaran una rutina[6]:

```
/// <summary>
/// Punt d'entrada de l'aplicació.
/// </summary>
[STAThread]
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new IU.FrmPrincipalEx());
}
```

Fig. 36 - Punt d'entrada de l'aplicació

En el nostre cas, al tractar-se d'una aplicació d'escriptori basada en formularis Windows, farà aixecar una interfície gràfica: la nostra finestra principal i a partir d'aquí la gestió d'esdeveniments permetrà obrir noves finestres i executar les rutines de tractament quan sigui precís.

El segon pas a tenir en consideració és la generació de codi. Per a fer-ho hem utilitzat les facilitats que ens aporta l'ús d'un entorn integrat de desenvolupament o IDE com és el Visual Studio 2008 Express.

[6] En el nostre cas usem el terme rutina d'una forma general per a referir-nos a subrutines i funcions.

A l'apartat de generació de codi ens em trobat una pestanya amb tots els paràmetres necessaris de configuració necessaris pel compilador de C# ja que hem usat aquest llenguatge.

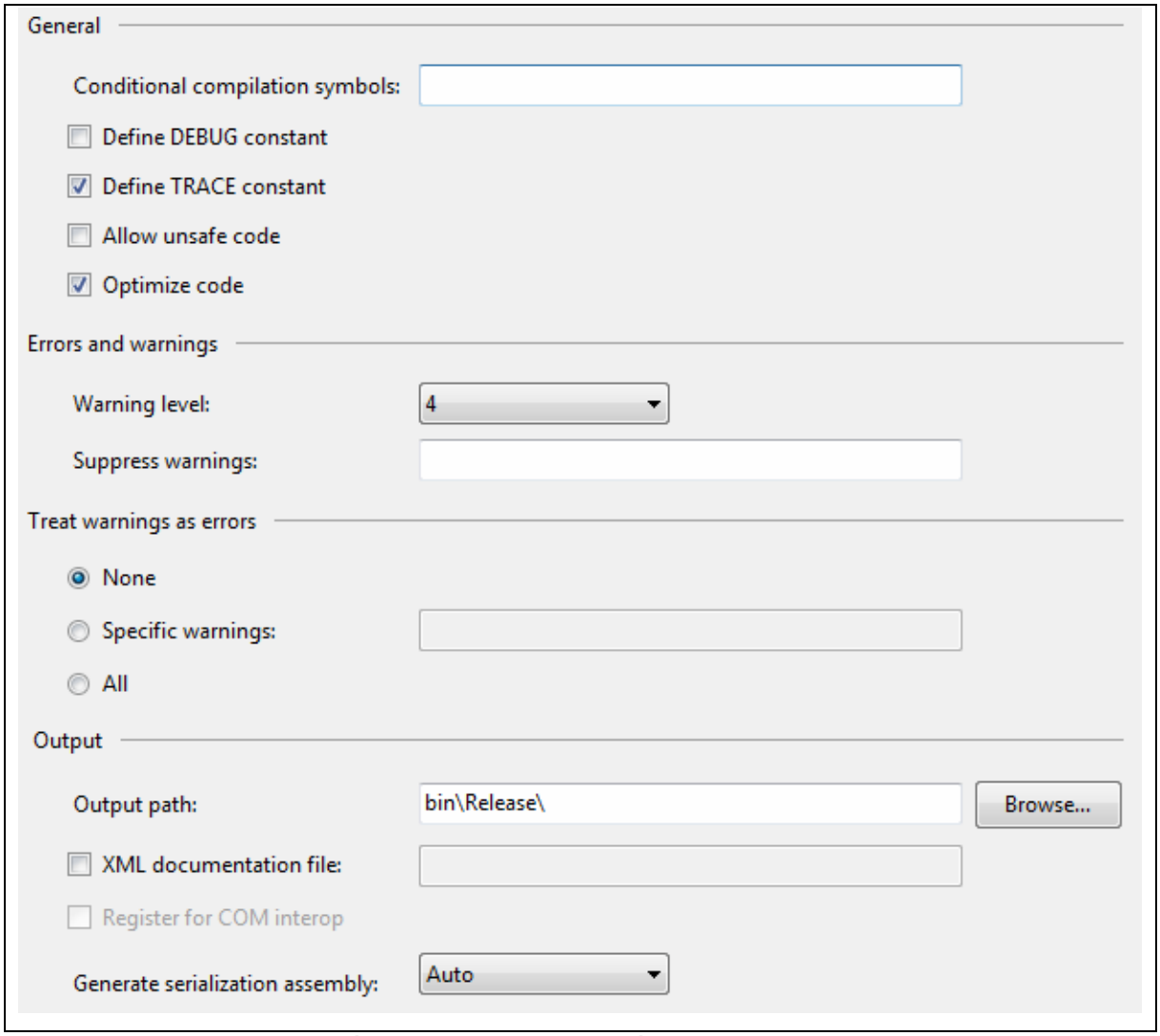


Fig. 37 - Paràmetres de configuració del compilador de C#

En aquesta figura podem veure els aspectes bàsics per a la generació: la definició dels objectes de traça per a poder inspeccionar el codi, el directori de generació i nivells d'avisos permesos pel compilador.

La pestanya important és la que es mostra a continuació ja que permet definir els atributs de l'aplicació tals com el nom, l'espai de noms, la classe que s'enllaçarà com a punt d'entrada, etc.

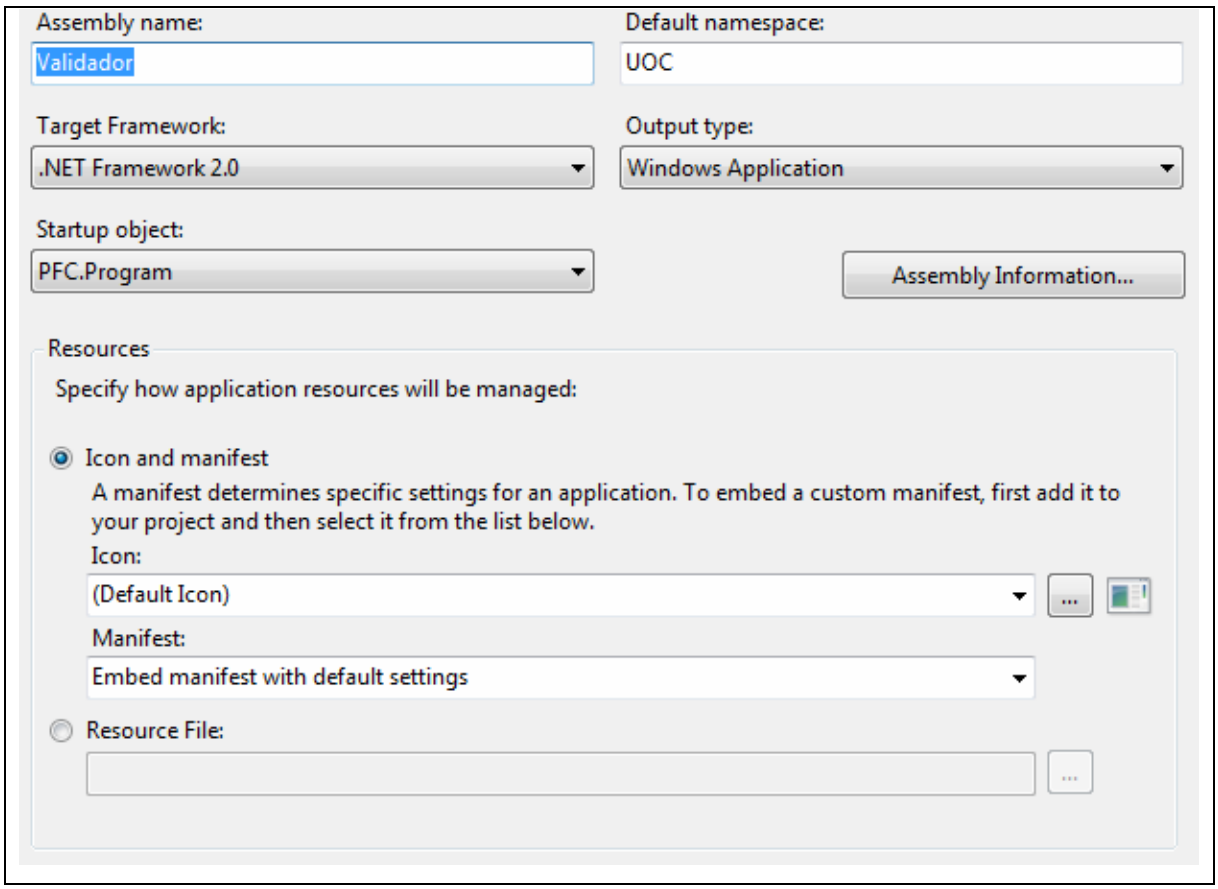


Fig. 38 - Configuració dels atributs de l'aplicació

A grans trets aquests han estat els passos necessaris per a generar la solució i els aspectes bàsics de la configuració. A partir d'aquí podem endinsar-nos en com s'ha implementat la rutina de tractament de documents ODT, la gestió dels patrons basats en expressions regulars i altres aspectes rellevants.

5.3.2 Flux d'execució

A continuació presentarem l'algorisme de tractament dels documents ODF. En aquest algoritme podrem veure el flux general de l'aplicació amb tots els processos i validacions relacionades. Completant aquesta definició trobarem també altres processos auxiliars com l'obtenció de dades externes i la persistència dels resultats.

A nivell intern l'algorisme rep les dades necessàries per a obrir un arxiu i valida si es tracta d'un document de text ODT vàlid. Arribats a aquest punt, si tot és correcte, comprovarà la existència de l'arxiu de patrons i seguidament el magatzem de dades sospitoses de l'AMS.

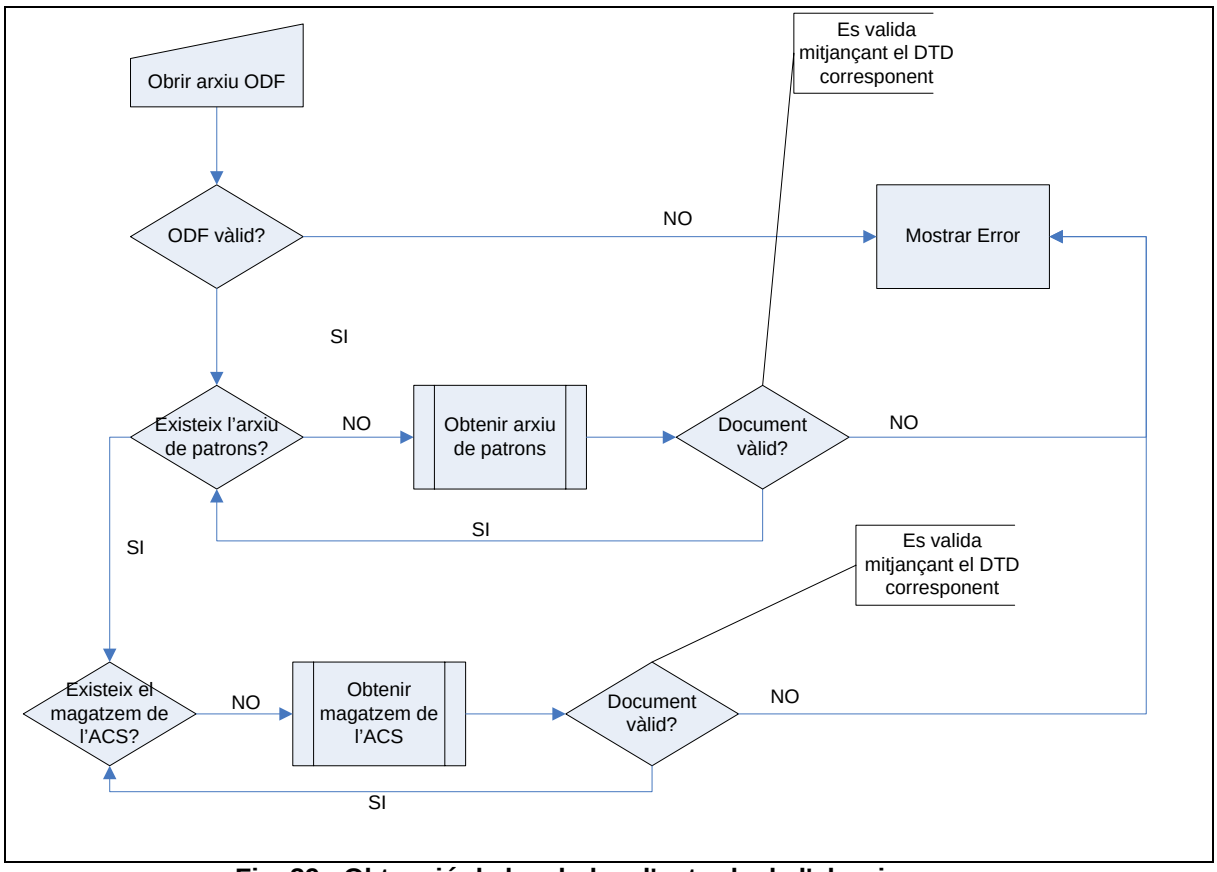


Fig. 39 - Obtenció de les dades d'entrada de l'algorisme

Un cop es disposa de totes les dades necessàries es procedirà al tractament del document que consisteix en un recorregut de tots els seus paràgrafs per a validar el contingut. Segons el resultat de la validació s'actuarà de la següent forma:

- Si el contingut és una dada compromesa, segons l'arxiu de patrons, es guarda a la cadena compromesa a l'objecte resultat. Si a més es sospitosa, es troba en el magatzem de l'AMS, es marcarà com a resultat positiu.
- Si el contingut no és una dada compromesa es tractarà el següent paràgraf.

Un cop recorregut tot el document i generat els resultats en format XML s'haurà de presentar a l'usuari en format HTML i per a fer-ho es farà ús d'una transformació XSL. Arribats a aquest punt l'usuari podrà desar els resultats en format XML.

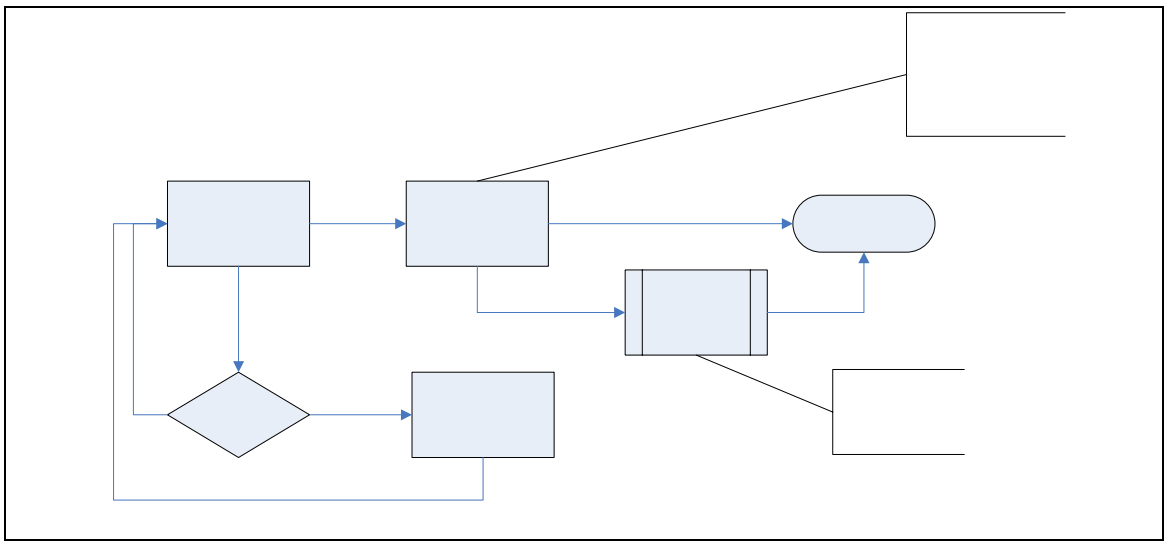


Fig. 40 - Tractament del document ODT

A continuació es presenta el diagrama de flux complet de l'algorisme:

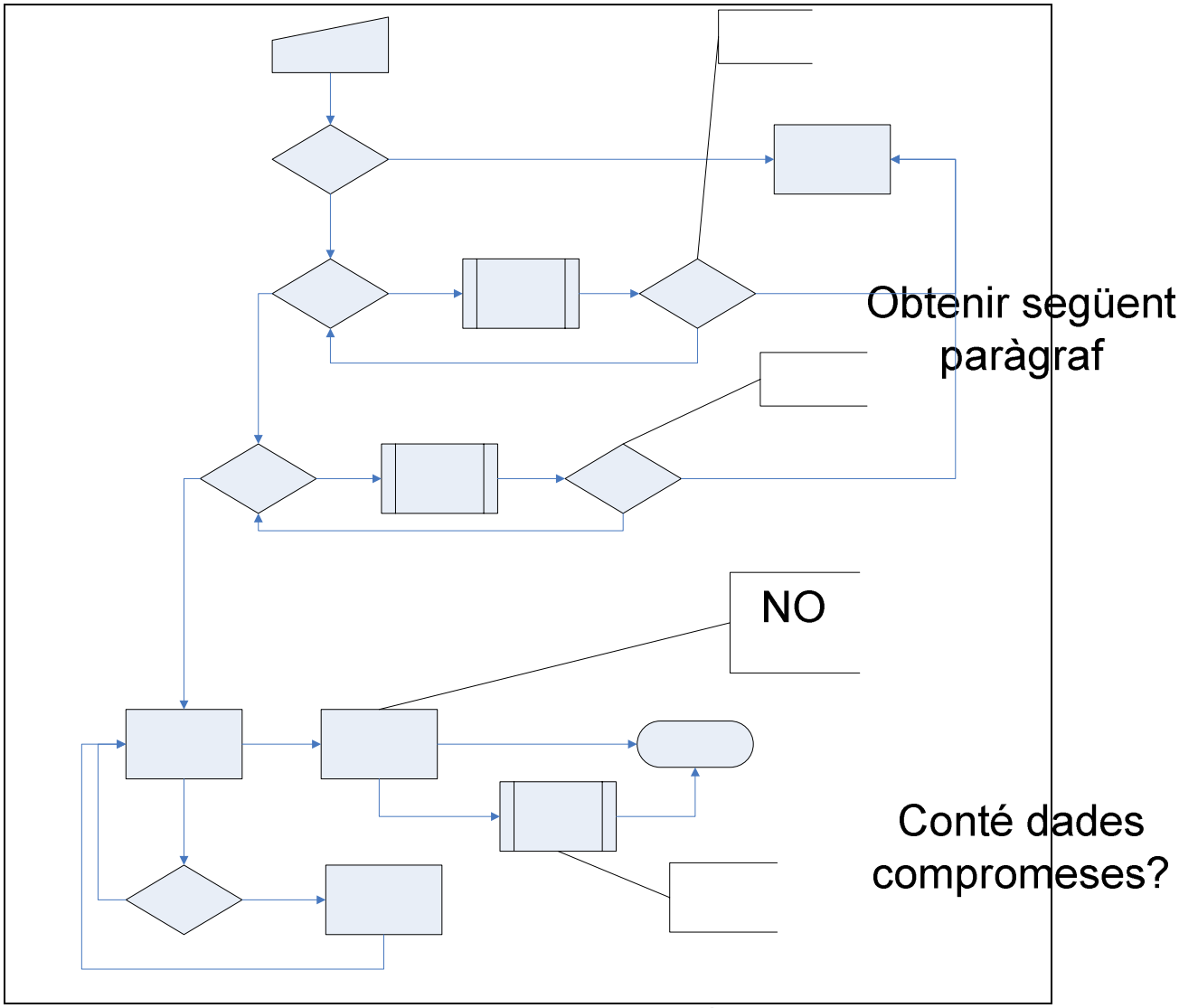


Fig. 41 - Algorisme de l'aplicació

5.3.3 Implementació

5.3.3.1 Classe GestorValidador

Aquesta classe fa de pont entre la interfície d'usuari i la lògica de negoci; per tant té la responsabilitat de la gestió del procés de validació.

Referent al codi exposa tres mètodes molt importants: el d'inicialització, el de tractament i el de generació de resultats.

A continuació es mostra la rutina d'inicialització on, a grans trets, es pot veure com:

1. Es crea un objecte PatroODT el qual llegeix els patrons de disc.
2. Es crea una instància de classe de ValidadorODT mitjançant la factoria de validacions i passant-li com a paràmetres l'objecte PatroODT i la ruta del magatzem de l'AMS.

```
public void Inicialitza(string uriPatrons, string uriDocumentODF, string uriRespositoriAMS)
{
    // Recolecció de les dades d'entrada.
    m_uriPatrons = uriPatrons;
    m_uriDocumentODF = uriDocumentODF;
    m_uriRepositoriAMS = uriRespositoriAMS;

    PatroODT patro = new PatroODT(m_uriPatrons);
    CreadorValidadorODT factoria = new CreadorValidadorODT();
    m_validador = factoria.GenerarValidador(patro, m_uriRepositoriAMS);
}
```

Fig. 42 - Mètode GestorValidador::Inicialitza

El mètode de tractament consisteix en invocar la rutina de validació de la instància validadora anterior passant-li per empenta el document ODF.

```
public void TractarDocument()
{
    Document document = new Document(m_uriDocumentODF);
    m_validador.Validar(document);
}
```

Fig. 43 - Mètode GestorValidador::TractarDocument

Finalment la rutina de processament de resultats aprofitant les classes bases de la plataforma .NET:

```
public void DesarResultats(string uri)
{
    XmlDocument document = new XmlDocument();
    document.LoadXml(m_validador.ResultatXml);
    document.Save(uri);
}
```

Fig. 44 - Mètode GestorValidador::DesarResultats

5.3.3.2 Interfície IPatrons

A continuació es detallen els mètodes de la interfície IPatrons.

```
interface IPatrons
{
    void Desar(string uri);
    Hashtable ObtenirPatrons();
    void Validar();
}
```

Fig. 45 - Interfície IPatrons

5.3.3.3 Classe Patrons

La classe Patro té la responsabilitat de llegir l'arxiu de patrons de l'aplicació a partir d'un arxiu XML i validar que sigui un arxiu XML vàlid per al DTD que l'acompanya.

```
public Patrons(string uri)
{
    if (!System.IO.File.Exists(uri))
    {
        throw new RN.Error.ClsArxiuPatronsNoTrobatException(uri);
    }
    m_uri = uri;
    m_patrons = new XmlDocument();
    m_patrons.Load(uri);
    this.Validar();
}
```

Fig. 46 - Constructor de la classe Patrons

```
public void Validar()
{
    try
    {
        // Estableix la configuració per a la validació.
        XmlReaderSettings configuracio = new XmlReaderSettings();
        configuracio.ProhibitDtd = false;
        configuracio.ValidationType = ValidationType.DTD;
        configuracio.ValidationEventHandler += new ValidationEventHandler(ValidationCallBack);

        // Crea l'objecte XmlReader.
        XmlReader reader = System.Xml.XmlReader.Create(m_uri, configuracio);

        // Parseja l'arxiu.
        while (reader.Read()) ;
    }
    catch (XmlException ex)
    {
        throw new RN.Error.ClsErrorDeValidacio(ex.Message, "Patro::Validar");
    }
}

// Mostra qualsevol error de validació.
private static void ValidationCallBack(object sender, ValidationEventArgs e)
{
    throw new RN.Error.ClsErrorDeValidacio(e.Message, "Patro::ValidationCallBack");
}
```

Fig. 47 - Validació de la classe Patrons

5.3.3.4 Classe ValidadorODT

Aquesta és la classe central de l'aplicació ja que té la responsabilitat de tractar el text del document a la recerca de dades compromeses que podran ser sospitoses si es troben identificades al magatzem de l'AMS.

El procés de validació, com es pot veure a la figura següent, consisteix en un recorregut sobre els elements que pertanyen a la propietat Contingut de l'objecte Document.

El recorregut té com a objectiu tractar tots els elements que es troben en el text del document. Aquests elements, si recordem el capítol corresponent al format ODT, poden ser des de paràgrafs a llistes d'elements passant per taules. Les taules i les llistes contindran altres elements que a la vegada contindran elements de tipus paràgraf i per tant el tractament consistirà en recórrer de forma descendent els elements de l'estructura a la recerca d'elements paràgraf i tractar el text que hi trobem a dins.

La següent figura presenta l'algorisme al qual aplicarem el disseny descendent per a arribar als elements paràgraf finals:

```
public override void Validar(Document document)
{
    m_resultat.AfegirMetadades(document.Nom);

    IEnumerator enumerador = document.Contingut.GetEnumerator();
    while (enumerador.MoveNext())
    {
        if (enumerador.Current is Paragraph)
        {
            this.TractarParagraf((Paragraph) enumerador.Current);
        }
        else if (enumerador.Current is Table)
        {
            this.TractarTaula((Table) enumerador.Current);
        }
        else if (enumerador.Current is List)
        {
            this.TractarLlista((List) enumerador.Current);
        }
        else if (enumerador.Current is UnknownContent)
        {
            this.TractarContingutDesconegut((UnknownContent)enumerador.Current);
        }
    }
}
```

Fig. 48 - Mètode ValidadorODT::Validar

Durant el procés d'implementació i jocs de proves es van tractar documents provinents de la suite Microsoft Office els quals eren transformats per l'aplicació Writer d'una forma especial: el procés acabava generant un element 'UnknowContent' que contenia tot el text del document Word en un objecte XmlDocument. Això significava que es podien fer consultes XPath per aconseguir tots els paràgrafs i no haver de recórrer tots els elements del document a fi i efecte de tractar cada paràgraf. Aquest punt es tractarà amb una mica més de detall a l'apartat de proves d'integració d'aquest mateix capítol.

A continuació la funció de tractament de paràgrafs que consisteix en fer un recorregut de la llista de patrons. Per a cadascun dels patrons crearem una expressió regular la qual aplicarem al text del paràgraf.

En cas de trobar-se alguna dada compromesa (match) validarem si es tracta d'una dada sospitosa i finalment desarem el resultat amb els següents valors:

- Número de paràgraf on s'ha trobat la dada compromesa.
- Resultat positiu o negatiu en funció de si la dada a més es sospitosa segons el magatzem de l'AMS.
- El valor de la dada compromesa.
- El patró que ha trobat l'ocurrència de dada compromesa.

```
private void TractarParagraf(Paragraph paragraf)
{
    Hashtable patrons = m_patro.ObtenirPatrons();
    Regex regex = null;
    Match match = null;
    bool resultat = false;

    foreach (DictionaryEntry item in patrons)
    {
        regex = new Regex(item.Value.ToString());
        if (regex.IsMatch(paragraf.Node.InnerText))
        {
            match = regex.Match(paragraf.Node.InnerText);
            resultat = this.ValidarDadaCompromesa(match.Value);
            m_resultat.AfegirResultat(m_numParagraf, resultat,
                                      match.Value, item.Key.ToString());
        }
    }
    m_numParagraf++;
}
```

Fig. 49 - Mètode ValidadorODT::TractarParagraf

Mètode que valida si la data compromesa es una data sospitosa de l'AMS.

```
private bool ValidarDadaCompromesa(string dada)
{
    return m_repositori.ValidarData(dada);
}
```

Fig. 50 - Mètode ValidadorODT::ValidarDadaCompromesa

5.3.3.5 Classe RepositoriAMS

La classe RepositoriAMS té la responsabilitat de validar si una cadena compromesa es tracta d'una dada sospitosa catalogada per l'AMS. Per a fer-ho aprofitem que hem usat un objecte HashTable on hem emmagatzemat les dades sospitoses de l'AMS considerant el valor sospitós com a clau de la HashTable. Amb això aconseguim millorar l'eficiència de la validació.

```
public bool ValidarData(string dada)
{
    return m_repositori.ContainsKey(dada);
}
```

Fig. 51 - Mètode RepositoriAMS::ValidarData

La creació del magatzem es basa en llegir l'arxiu XML que ens proporciona l'AMS mitjançant correu electrònic. A continuació el propi constructor valida que el contingut de l'arxiu XML sigui validat per el seu DTD corresponent.

```
public RepositoriAMS(string uri)
{
    if (!System.IO.File.Exists(uri))
    {
        throw new RN.Error.ClsArxiuRepositoriAMSNoTrobatException(uri);
    }
    m_uri = uri;
    m_repositori = new Hashtable();
    this.Validar();
}
```

Fig. 52 - Constructor RespositoriAMS

```
public bool ValidarData(string dada)
{
    return m_repositori.ContainsKey(dada);
}

private void Validar()
{
    // Estableix la configuració per a la validació.
    XmlReaderSettings configuracio = new XmlReaderSettings();
    configuracio.ProhibitDtd = false;
    configuracio.ValidationType = ValidationType.DTD;
    configuracio.ValidationEventHandler += new ValidationEventHandler(ValidationCallBack);

    // Crea l'objecte XmlReader.
    XmlReader reader = System.Xml.XmlReader.Create(m_uri, configuracio);

    // Parseja l'arxiu.
    while (reader.Read())
    {
        if (reader.NodeType == XmlNodeType.Text)
        {
            m_repositori.Add(reader.Value, reader.Value);
        }
    };
}

// Mostra qualsevol error de validació.
private static void ValidationCallBack(object sender, ValidationEventArgs e)
{
    throw new RN.Error.ClsErrorDeValidacio(e.Message, "RepositoriAMS::ValidationCallBack");
}
```

Fig. 53 - Mètode RepositoriAMS::ValidarData

5.3.3.6 Classe TransformadorXSL

Aquesta classe s'encarrega de transformar l'arxiu XML de resultats en un arxiu en format HTML que es pugui visualitzar en un navegador o un control Web com és en el nostre cas.

Per a poder fer la transformació es fa ús d'una classe de la plataforma .NET destinada a fer transformacions, en concret es crearà una instància d'aquesta classe que farà les següents tasques:

1. Carregarà l'arxiu de transformació presentat a l'etapa de disseny
2. Llegirà el contingut del document XML

3. Aplicarà la transformació
4. Retornarà el contingut de la transformació en format text

```
class TransformadorXSL
{
    public string Transformar(string xml)
    {
        XslCompiledTransform xslt = new XslCompiledTransform();

        xslt.Load(@"../cfg/xsl/sortida.xslt");
        XmlDocument document = new XmlDocument();
        document.LoadXml(xml);
        MemoryStream fluxe = new MemoryStream();
        XmlTextWriter writer = new XmlTextWriter(fluxe, Encoding.UTF8);
        writer.Formatting = Formatting.Indented;

        xslt.Transform(document, writer);

        fluxe.Flush();
        return UTF8Encoding.UTF8.GetString(fluxe.GetBuffer());
    }
}
```

Fig. 54 - Mètode TransformadorXSL::Transformar

5.4 Proves d'integració

La última etapa de la fase d'implementació és l'etapa de proves d'integració amb objectiu de descobrir possibles errors ocults i validar que l'aplicació es comporti com esperem.

A continuació es mostraran diverses proves efectuades amb l'aplicació i el resultat que han originat.

Respecte als documents usats hem usat dos mètodes per a crear-los:

- Usant l'eina Writer amb l'última versió d'OpenOffice, la 2.4
- Baixant algun formulari oficial de la Web de la Generalitat de Catalunya. En aquest cas la majoria de documents eren en format Office Word i per tant hem hagut de transformar-los amb l'eina Writer. Un cop transformats hem omplert els camps que hem considerat significatius per a les proves.

Abans d'explicar cadascun dels processos de prova mostrarem els aspectes de configuració usats, és a dir, els arxius de patrons i repositori de l'AMS que han estat utilitzats. Per motius d'espai adrecem al lector a consultar l'arxiu de patrons dins del directori de configuració de l'aplicació.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE Repositori SYSTEM "dtd_repositori_ams.dtd">
<Repositori>
  <Metadades>
    <Autor>Agència Catalana de Seguretat</Autor>
    <Data>25/05/2008</Data>
    <Versio>1</Versio>
    <Descripcio>Repositori de dades sospitoses detectades per l'Agència Mundial de Seguretat</Descripcio>
    <Signatura>1234567890</Signatura>
  </Metadades>
  <DadesSospitoses>
    <DadaSospitosa>correu@correu.es</DadaSospitosa>
    <DadaSospitosa>1111 2222 33 4444444444</DadaSospitosa>
    <DadaSospitosa>255.0.34.123</DadaSospitosa>
    <DadaSospitosa>12345678A</DadaSospitosa>
    <DadaSospitosa>33</DadaSospitosa>
    <DadaSospitosa>77</DadaSospitosa>
    <DadaSospitosa>1111-2222-33-4444444444</DadaSospitosa>
    <DadaSospitosa>http://www.worldbank.cat</DadaSospitosa>
    <DadaSospitosa>Tel. 93 444 33 22</DadaSospitosa>
  </DadesSospitoses>
</Repositori>
```

Fig. 55 - Magatzem de l'AMS per a les proves

5.4.1 Documents generats amb l'eina Writer

5.4.1.1 Prova 01

Aquesta va ser la primera prova que es va fer, bastant complexa, amb tots els elements que es poden generar amb l'aplicació Writer. Trobarem doncs a més de paràgrafs, taules, llistes, files, cel·les, elements de llistes, etc.

L'objectiu d'aquesta prova és fer una validació general del funcionament de l'aplicació.

Prova PFC-Compiladors 2007-2008

Prova dins d'un paràgraf:

DNI: 12345678A
Correu electrònic: correu@correu.es
Número de compte: 1111 2222 33 4444444444
Número de compte: 1111-2222-33-4444444444
Adreça Web: www.lacaixa.es
Adreça IP: 255.0.34.123

Prova dins d'una taula:

Titular	Número de compte
NIF	12345678A
Correu electrònic	correu@correu.es
Número de compte	1111.2222.33.4444444444
Adreça Web	www.lacaixa.es
Adreça IP	255.0.34.123

Prova dins d'una llista

- 1111 2222 33 4444444444
- correu@correu.com
- 12345678
- www.lacaixa.es
- 255.0.34.123

Fig. 56 - Document de prova 01

Després de l'execució hem obtingut el següent resultat:

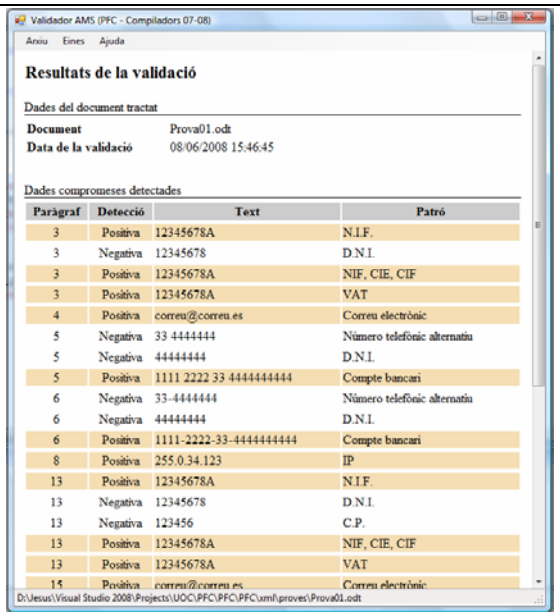


Fig. 57 - Resultat de la prova 01

5.4.1.2 Prova 02

Aquesta prova intenta simular una carta d'una entitat financera adreçada a un client. La prova hauria de concloure amb la identificació de la màxima quantitat d'informació possible en aquest tipus de document: NIF, adreça postal, adreça web de l'entitat, telèfons, correus electrònics, números de comptes bancaris, etc.

Banc Mundial
4 de juny de 2008

Estimat senyor Jesús Delgado,

en referència al seu escrit li transmetem les dades que disposa el Banc Mundial sobre les seves activitats a la nostra entitat:

	Número de Compte Corrent	Data d'obertura
Compte corrent número 1	1111-2222-33-4444444444	1 de maig de 2007
Compte corrent número 1	1111-2222-33-4444444455	1 de maig de 2007

Per a qualsevol dubte o aclariment pot emprar l'atenció al client a la nostra plana d'internet:

http://www.worldbank.cat

o bé adreçar-se a qualsevol de les nostres oficines o mitjançant trucada telefònica si ho prefereix. A continuació li mostrem les dades de les oficines més properes al seu domicili:

Banc Mundial
C/ exemple nº1
CP 08080
Tel. 93 444 33 22

Atentament, el Director del Banc Mundial
John Smith

Fig. 58 - Document de la prova 2

A més s'ha afegit alguns dels valors al magatzem de l'AMS per a que hi hagi deteccions positives tal com es pot contrastar en la següent imatge i amb la llista de dades sospitoses indicada al començament de l'apartat:

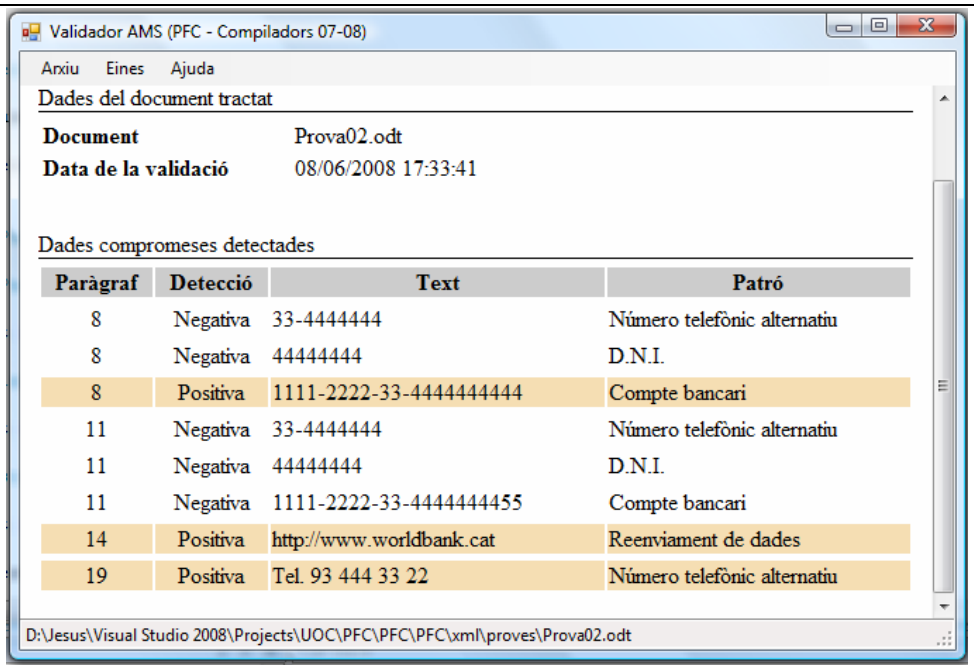


Fig. 59 - Resultat de la prova 02

5.4.2 Documents generats a partir de formularis oficials

Aquests documents provinents de la transformació de formularis oficials de la Generalitat de Cataluya són una prova bastant complexa a causa de que el procés de transformació genera nous espais de noms i per tant elements. A més el contingut d'un document Microsoft Office presenta la particularitat de que és accessible de forma completa i per tant podrem fer ús de cerques XPath cosa que no es pot fer amb els documents OpenOffice amb la llibreria utilitzada.

En un primer moment l'aplicació no va poder llegir el contingut dels documents transformats a OpenDocument i es va examinar el codi font per a esbrinar què succeïa. El problema estava en que la llibreria ODF creava un element nou anomenat 'UnknowContent' de tipus XmlElement i dins d'ell tot el contingut del document de text en format XML.

La forma de solucionar aquest problema ha estat afegir un nou criteri en el tractament i fer un recorregut sobre els elements de tipus paràgraf del element anterior per a tractar el contingut.

5.4.2.1 Prova 03

Es tracta del formulari “”. Aquest és un cas especial ja que la majoria de la informació es troba en el mateix paràgraf o dit d’una altra manera a la mateixa cadena. Aquí és important veure que el recorregut ens permet processar tots els patrons disponibles sobre la mateixa cadena.

CERTIFICACIÓ DE VINCULACIÓ

Jesús Delgado Pons amb DNI número 12345678, com a *President* de l'entitat *Entitat Primera*, inscrita al Cens de la Secretaria General de Joventut de la Generalitat amb el número *1234567890* i amb NIF 12345678A

CERTIFICO:

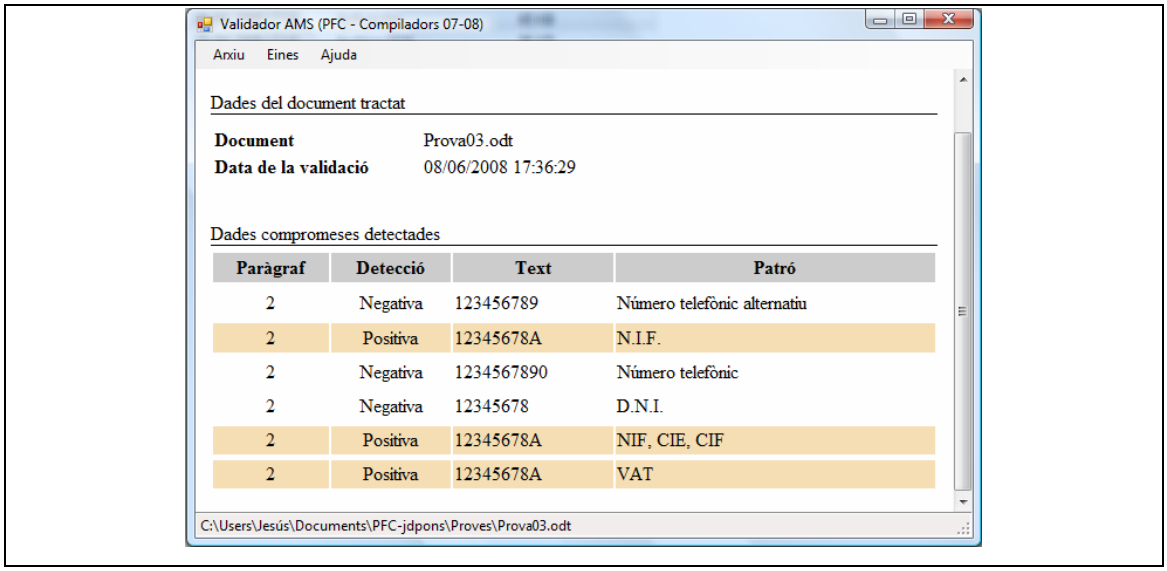
Que el grup juvenil *Entitat Juvenil* està vinculat a Entitat Primera

I, perquè així consti als efectes oportuns, signo aquesta certificació.

Barcelona, 6 de juny de 2008 *Segell de l'entitat*

Fig. 60 - Document de la prova 03

El resultat ha estat:



Validador AMS (PFC - Compiladors 07-08)

Arxiu Eines Ajuda

Dades del document tractat

Document Prova03.odt
Data de la validació 08/06/2008 17:36:29

Dades compromeses detectades

Paràgraf	Detecció	Text	Patró
2	Negativa	123456789	Número telefònic alternatiu
2	Positiva	12345678A	N.I.F.
2	Negativa	1234567890	Número telefònic
2	Negativa	12345678	D.N.I.
2	Positiva	12345678A	NIF, CIE, CIF
2	Positiva	12345678A	VAT

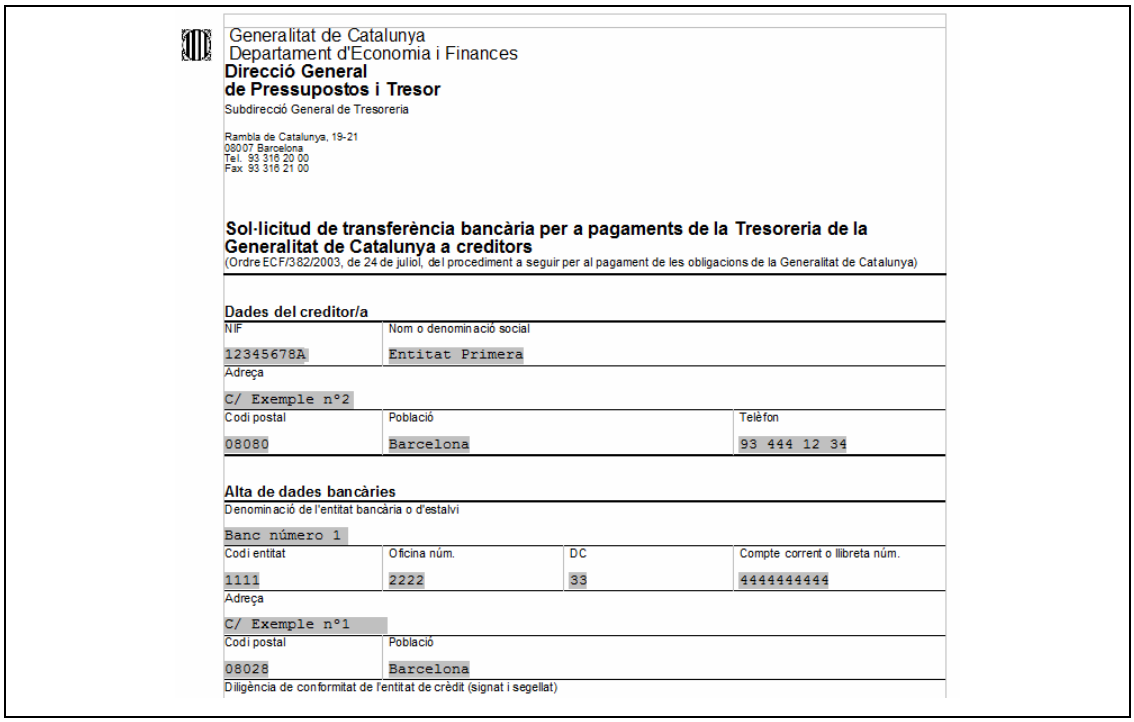
C:\Users\Jesús\Documents\PFC-jdpons\Proves\Prova03.odt

Fig. 61 - Resultat de la prova 03

Com podem comprovar hi ha algunes dades compromeses que són detectades més d’un cop per patrons diferents. Això es degut a l’expressió regular usada.

5.4.2.2 Prova 04

Es tracta del formulari " Sol·licitud de transferència bancària per a pagaments de la Tresoreria de la Generalitat de Catalunya a creditors". Aquest és un cas més complex que l'anterior ja que les dades estan distribuïdes en diferents paràgrafs i niuades dins d'altres elements.



Generalitat de Catalunya
Departament d'Economia i Finances
Direcció General de Pressupostos i Tresor
Subdirecció General de Tresoreria

Rambla de Catalunya, 19-21
08007 Barcelona
Tel. 93 316 20 00
Fax 93 316 21 00

Sol·licitud de transferència bancària per a pagaments de la Tresoreria de la Generalitat de Catalunya a creditors
(Ordre ECF/382/2003, de 24 de juliol, del procediment a seguir per al pagament de les obligacions de la Generalitat de Catalunya)

Dades del creditor/a

NIF	Nom o denominació social	
12345678A	Entitat Primera	
Adreça		
C/ Exemple nº2		
Codi postal	Població	Telèfon
08080	Barcelona	93 444 12 34

Alta de dades bancàries

Denominació de l'entitat bancària o d'estalvi			
Banc número 1			
Codi entitat	Oficina núm.	DC	Compte corrent o llibreta núm.
1111	2222	33	4444444444
Adreça			
C/ Exemple nº1			
Codi postal	Població		
08028	Barcelona		

Diligència de conformitat de l'entitat de crèdit (signat i segellat)

Fig. 62 - Document de la prova 04

A continuació es mostra un fragment de l'arxiu de resultats desat a disc amb la pròpia aplicació:

```
<?xml version="1.0"?>
<Validacio>
  <Resultats>
    <Resultat>
      <Paragraf>8</Paragraf>
      <DataSospitosaAMS>False</DataSospitosaAMS>
      <Text>Tel. 93 316 20 00</Text>
      <Patro>Número telefònic alternatiu</Patro>
    </Resultat>
    <Resultat>
      <Paragraf>9</Paragraf>
      <DataSospitosaAMS>False</DataSospitosaAMS>
      <Text> 93 316 21 00</Text>
      <Patro>Número telefònic alternatiu</Patro>
    </Resultat>
    <Resultat>
      <Paragraf>13</Paragraf>
      <DataSospitosaAMS>True</DataSospitosaAMS>
      <Text>12345678A</Text>
      <Patro>N.I.F.</Patro>
    </Resultat>
    <Resultat>
      <Paragraf>13</Paragraf>
      <DataSospitosaAMS>False</DataSospitosaAMS>
      <Text>12345678</Text>
      <Patro>D.N.I.</Patro>
    </Resultat>
```

Fig. 63 - Resultat de la prova 04

Capítol 6 Valoració econòmica

Aquest projecte no té com a objectiu la venda de cap producte sinó que és un estudi d'un format estàndard per a un millor coneixement d'aquest i les avantatges que pot oferir. Per tant no es fa necessari cap estudi de mercat per conèixer el cost d'oportunitat que tindria alliberar el producte del projecte i només donarem una visió global dels recursos i costs associats.

6.1 Recursos Humans

Es considera una mitja de 50€ / hora de treball.

Tasca	Temps (hores)	Personal	Cost (€)
Investigació	232	1	11.600
Anàlisi/Disseny	80	1	4000
Implementació	80	1	4000
Documentació	164	1	8200
		Total	27.800

Taula 3 - Relació de cost per recurs

6.2 Altres despeses

Partida	Cost
Estació de treball	1.000€
Visual Studio 2008 Express	0€
Microsoft Office 2003	Inclòs al maquinari
OpenOffice 2.04	0€
Connexió a Internet	5 mesos x 50€ = 250€
Material formatiu	75€ (aproximadament)
	1.325€

Taula 4 - Altres despeses

Capítol 7 Conclusions

7.1 Fites assolides

Si recordem les fites marcades al començament del projecte:

- Estudi del format Open Document
- Estudi de les expressions regulars
- Estudi del format XML i de les transformacions XML
- Elaboració d'una eina que apliqués tota la tecnologia anterior

ens adonarem de que aquestes s'han assolit en major o menor grau, possibilitant el desenvolupament del programari associat a aquest projecte respectant la planificació elaborada, malgrat algun petit desviament com veurem a continuació.

Recapitulant: entenem que el projecte s'ha dut a terme amb èxit al completar totes les tasques identificades segons la planificació prevista i amb tots els objectius complerts tant d'investigació com de desenvolupament.

7.2 Opinió personal

Quan vaig començar aquest projecte era conscient de la importància dels formats oberts però no de les seves repercussions ja que durant un cert temps vaig estar investigant sobre el format OpenXML i ràpidament vaig descobrir que existia un altre format de document digital: l'ODF.

Durant l'etapa d'investigació he descobert nous punts de vista, uns més propers a la tecnologia, d'altres més propers a la política... tots ells m'han permès formar-me una opinió al respecte de l'existència de formats estàndard d'oficina i la competència entre formats.

La meva opinió al respecte és que estic totalment d'acord amb la creació d'estàndards tecnològics que permetin un accés lliure a la seva definició, així com el seu us sense cap tipus d'aranzel. També estic d'acord amb l'existència de competència entre formats perquè només poden dur a una millora dels mateixos i a un benefici per a l'usuari final, sobretot perquè sempre podrà optar entre més d'un format. I sobretot a l'exigència de l'usuari per a que així sigui i cap organització imposi un format únic.

En la vessant tècnica aquest projecte m'ha permès treballar bastant profundament el format d'arxiu ODF cosa que m'ha originat noves idees relacionades amb altres projectes laborals. També m'ha permès veure que el format XML tot i que en moltes ocasions és molt adequat per a la transmissió d'informació en altres no és la millor opció: un exemple el trobem en el llenguatge JSON per a la transmissió de dades entre planes Web.

En l’apartat personal he gaudit molt durant l’elaboració d’aquest projecte i m’agradaria molt que també ho fos de profitós per a qualsevol que el llegeixi.

Glossari

DTD

De l’anglès Data Type Definition és un arxiu que valida el contingut del document XML que en fa referència. Valida les entitats i l’ordre en que apareixen al document.

Expressió regular

Expressió que descriu un conjunt de cadenes sense enumerar als seus elements i concretament a l’àrea de la programació permeten realitzar cerques dins de cadenes de caràcters.

HTML

Acrònim d’HyperText Markup Language, és el llenguatge de marcat predominant per a la construcció de planes Web. Es tracta d’un llenguatge en format XML.

Internet

Internet és una xarxa pública i global de computadors interconnectats mitjançant el protocol d'internet (Internet Protocol) i que transmeten les dades mitjançant commutació de paquets.

ISO/IEC

Acrònim de l’Organització Internacional per a l'Estandardització.

Java

El llenguatge de programació Java fou dissenyat per James Gosling i els seus companys a Sun Microsystems, a l'any 1990, a partir del C++. Des del seu naixement fou pensat com un llenguatge orientat a objectes, és a dir, que segueix la filosofia de programar mòduls senzills, per tal de crear aplicacions avançades quan tots treballen junts.

GML

El Generalized Markup Language és un conjunt de macros que implementen etiquetes per al formatador de text d'IBM SCRIPT, creat als anys 60.

Microsoft .NET

Alternativa a Java de Microsoft.

Microsoft Office Open XML

Office Open XML (conegut sovint com a OOXML o OpenXML) és un estàndard internacional per a presentar documents electrònics com fulls de càlcul, presentacions, bases de dades i documents de text desenvolupat per Microsoft.

Motor de cerca

Programes o subsistemes de desenvolupament que admeten expressions regulars per a realitzar cerques sobre un text donat.

ODF

L'OpenDocument és un format de fitxer estàndard, creat per OASIS, i també és conegut com a Format de Document Obert per Aplicacions Ofimàtiques.

Open Document OASIS Standard

Veure ODF

PDF / A

Document portable d'Adobe Systems.

Perl

Perl és un llenguatge de programació dissenyat per Larry Wall inspirat en els llenguatges awk, C, sed, shell entre d'altres.

Programari lliure

El programari lliure (en anglès free software) és el programari que, un cop obtingut, pot ser utilitzat, copiat, estudiat, modificat i redistribuït lliurement.

SGML

Acrònim de Standard Generalized Markup Language. Consisteix en un sistema per a la organització i etiquetat de documents.

Tim Berners Lee

Tim Berners-Lee és considerat com l'inventor de la World Wide Web (Internet).

XML

Acrònim de l'anglès eXtensible Markup Language , és un metallenguatge extensible, d'etiquetes, desenvolupat pel World Wide Web Consortium. És una simplificació i adaptació de l'experimentat SGML, i permet definir la gramàtica de llenguatges específics.

XSL

Acrònim d'Extensible Stylesheet Language. És una família de llenguatges basats en l'estàndard XML que permet descriure com la informació continguda en un document XML qualsevol ha de ser transformada o formatada per a la seva presentació en un mitjà específic.

XSLT

És un estàndard de l'organització W3C que presenta una forma de transformar documents XML en uns altres i fins i tot a formats que no són XML.

W3C

El World Wide Web Consortium és un consorci internacional que treballa per a desenvolupar i promocionar estàndards per al Web. El dirigeix Tim Berners-Lee, creador d'Internet i autor de les especificacions de l'URL, l'HTTP i l'HTML, que són les seves principals tecnologies d'aquesta xarxa.

Bibliografia

[1]

XML
Wikipedia (http://es.wikipedia.org/wiki/XML)
Data de la última consulta: 29 d'abril / 08

[2]

DTD
Wikipedia (http://es.wikipedia.org/wiki/DTD)
Data de la última consulta: 29 d'abril / 08

[3]

ISO/IEC 26300
Open Document Format for Office Applications (OpenDocument) v1.0
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485
Data de la última consulta: 03 de maig / 08

[4]

Trampas ocultas en Open Document y efectos secundarios en el software libre y de código abierto
Marco Fioretti
Revista Novatica Número 184, Novembre – desembre de 2006

[5]

Interoperabilidad: ¿se impondrá el verdadero formato universal de ficheros?
Sam Hiser
Gary Edwards
Revista Novatica Número 184, Novembre – desembre de 2006

[6]

Abierto desde el diseño: el Formato de Documento Abierto para aplicaciones ofimáticas
Erwin Tenhumberg, Donald
Harbison, Rob Weir
Comité Técnico de Adopción de ODF, OASIS
Revista Novatica Número 184, Novembre – desembre de 2006

[7]

ISO/IEC 19005-1:2005
Document management -- Electronic document file format for long-term preservation
Part 1: Use of PDF 1.4 (PDF/A-1)
http://www.iso.org/iso/catalogue_detail?csnumber=38920
Data de la última consulta: 03 de maig / 08

[8]

OpenXML.info
http://www.openxml.info/
Data de la última consulta: 03 de maig / 08

[9]

ISO-26300 (OpenDocument) vs. MS-Office Open XML
Alberto Barrionuevo García
Consultor de gestión documental, Coordinador de EstándaresAbiertos.org, Vicepresidente de la FFII.org
Revista Novatica Número 184, Novembre – desembre de 2006

[10]

Interoperabilidad: ¿se impondrá el verdadero formato universal de ficheros?
Sam Hiser, Gary Edwards
OpenDocument Foundation
Revista Novatica Número 184, plana 30, Novembre – desembre de 2006

[11]

Brian Jones' Blog
Data de la última consulta: 03 de maig / 08

[12]

Patró Factoria
http://es.wikipedia.org/wiki/Factory_Method_%28patr%C3%B3n_de_dise%C3%B1o%29
Data de la última consulta: 28 de maig / 08

[13]

.NET Framework 2.0 Application Development Foundation
Microsoft Press

Autor: Administrador del Sistema

Validador 1.0
Manual d'usuari

Índex de continguts

Autor: Administrador del Sistema

Instal·lació

L'eina es pot executar en qualsevol versió de Windows però caldrà tenir una sèrie de programari instal·lat. La guia de preparació de la màquina és la següent:

1. Actualitzar l'instal·lador "Windows Installer" a la versió 3.1 (mitjançant el Windows Update)

2. Instal·lar el framework NET 2.0

3. Opcionalment actualitzar el sistema altre cop mitjançant Windows Update.

Autor: Administrador del Sistema

Execució del "Validador"

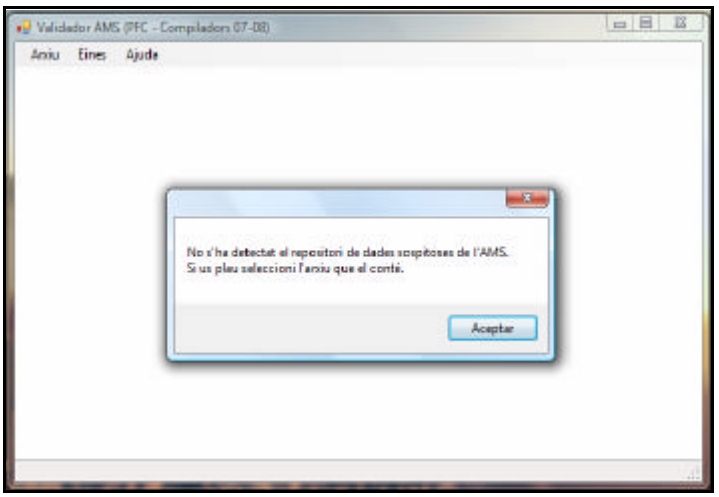
Preparació de l'aplicació

Un cop preparada la màquina procedirem a executar el validador. Aquest es troba al subdirectori "Executable" i concretament la ruta a l'aplicació és:

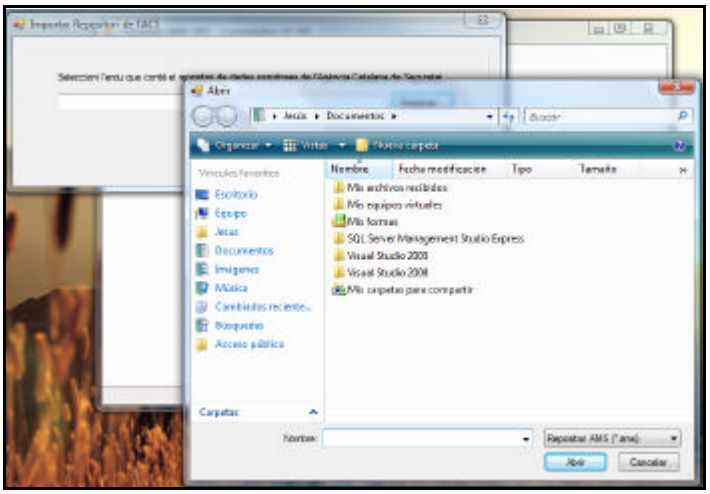
/jdpons_producte/Validador/Bin/Validador.exe

Haurem de fer doble clic sobre l'arxiu anterior.

Si és el primer cop que s'executa el Validador apareixerà el següent diàleg que ens indica que hem d'escollir el magatzem de dades sospitoses de l'AMS.



No ens hem de preocupar ja que seguidament l'aplicació ens permet escollir un arxiu a través d'un diàleg com el que es mostra a continuació:



La ubicació del magatzem de l'AMS per a les proves és la següent:

Autor: Administrador del Sistema

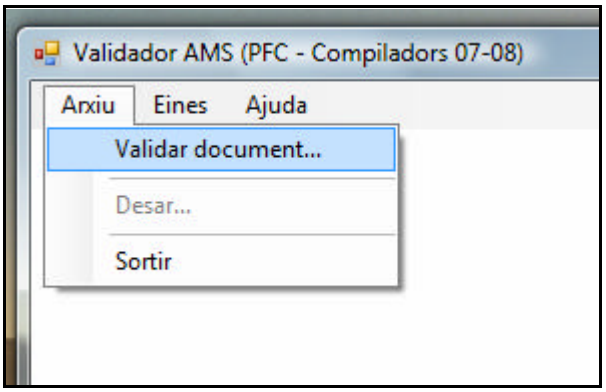
/jdpons_producte/Validador/AMS/repositori.ams

Nota: l'aplicació desa en el directori de configuració de l'usuari els paràmetres de configuració indicats.

Nota 2: el propòsit de poder escollir l'arxiu magatzem de l'AMS es deu a que aquest vindrà, segons la proposta del projecte, a través de correu electrònic i per tant s'ha entés que cal facilitar aquesta tasca. Per coherència també s'ha fet amb la configuració de l'arxiu de patrons per si hi haguessin actualitzacions de l'eina en un futur.

Validar un document

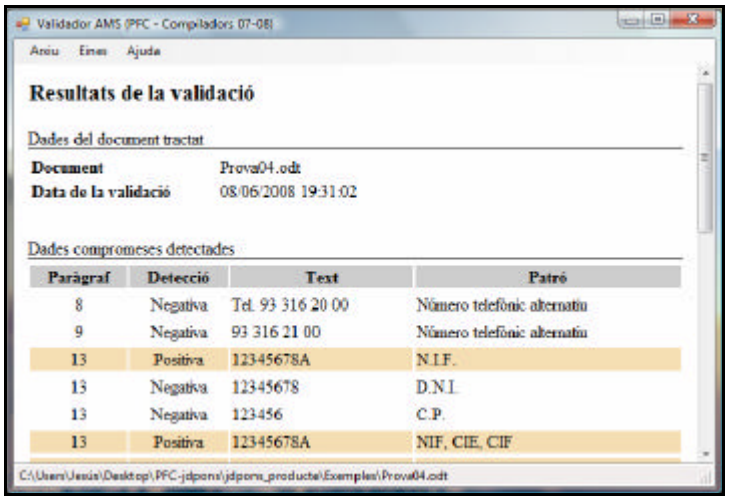
Per a validar un document només haurem d'anar al menú **ARXIU** i seleccionar l'opció **VALIDAR** que ens obrirà un diàleg com l'anterior per a escollir el document de text ODF a validar:



Les proves usades en el projecte es poden trobar a la següent ruta:

/jdpons_producte/Exemples

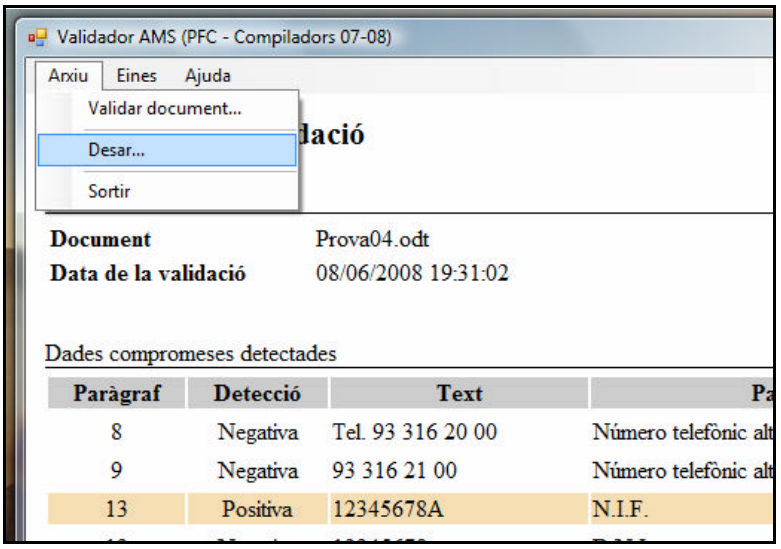
Un cop seleccionat l'arxiu el procés es durà a terme i veurem el resultat de l'execució dins de la pròpia aplicació:



Autor: Administrador del Sistema

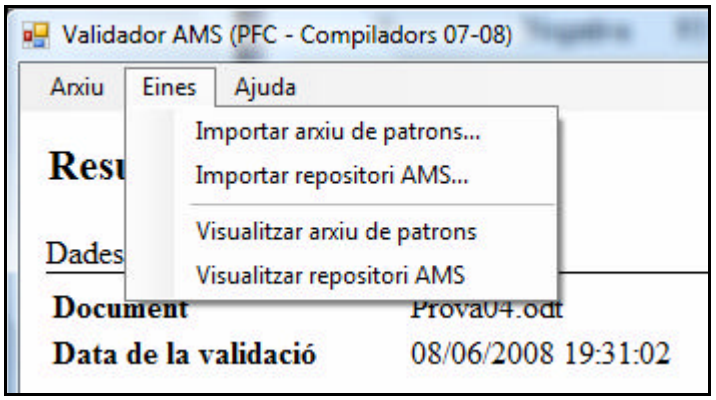
Guardar els resultats a disc

Podem guardar els resultats en un arxiu XML mitjançant el menú **ARXIU** i seleccionat l'opció **DESAR**. Això ens farà aparèixer un diàleg per a guardar l'arxiu amb el nom que nosaltres desitgem i en format XML:



Consultar els arxius de patrons i de dades sospitoses de l'AMS

També podem consultar el contingut dels arxius de patrons i dades sospitoses. Per a fer-ho seleccionarem dins del menú **EINES** l'opció corresponent:



Com es pot comprovar també podem importar nous arxius de dades i patrons. Això es una mica més complicat ja que tant un com l'altre han d'anar acompanyats dels seus respectius DTD.

Nota: els arxius de patrons i transformació els podem trobar a la següent ruta:

jdpons_producte/Validador/Cfg

Construcció d'una eina automàtica per extreure informació d'un document OpenOffice

Autor: Jesús Delgado Pons
Consultor: Jordi Ferrer Duran
Enginyeria en informàtica
8 de juny / 08

Índex de continguts

Format XML

Definició

- Llenguatge de marques extensible que permet estructurar dades de forma jeràrquica
- Propietats:
  - Al ser extensible permet definir nous llenguatges per a diferents necessitats
  - Va acompanyat d'una definició de tipus que permet crear analitzadors genèrics
  - Permet usar diferents codificacions
- En conseqüència:
  - Estàndard d'intercanvi de dades entre diferents plataformes

Format XML

Esquemes DTD

- “El format XML va acompanyat d’una definició de tipus”
  - enumerarà una sèrie d’entitats que definiran l’estructura dels documents XML
- Propietats
  - Els elements que són permesos dins l’aplicació i el contingut d’aquestes etiquetes.
  - L’ordre en que es contindran els elements anterior o estructura.
  - La jerarquia de les etiquetes o quines van dintre de quines.
- En conseqüència
  - Validen els documents XML que hi fan referència

Format XML

Transformacions XSL

- Permeten transformar documents XML en altres tipus de documents XML, per exemple en arxius HTML
- Procés de transformació
  - S'identifiquen entitats del document XML mitjançant XPath a les que s'apliquen transformacions o altres accions.
  - Les accions poden ser inclús funcions programàtiques que permeten fer tractaments més complexos.

Expressions Regulars

Definició

- Expressió que descriu un conjunt de cadenes sense enumerar als seus elements
- Permeten realitzar cerques dins de cadenes de caràcters mitjançant motors de cerca
- Dos tipus de motors de cerca
  - Destinats a usuaris finals
    - Aplicacions finals
  - Destinats a desenvolupadors
    - Framework .NET → System.Text.RegularExpressions
    - Java → Java.Util.Regex
    - Perl → Perlre

Format ODF

Definició

- Acrònim d'Open Document OASIS Standard
- Propietats
  - Està basat en el format XML
  - És l'estàndard ISO/IEC 26300 per a documents digitals d'oficina
- Avantatges
  - Es tracta de programari lliure i obert
  - Aporta al usuari final la independència al programari de propietat
  - Ratifica el pas de la informàtica dels formats propietaris a la informàtica distribuïda i lliure de la informació

Format ODF

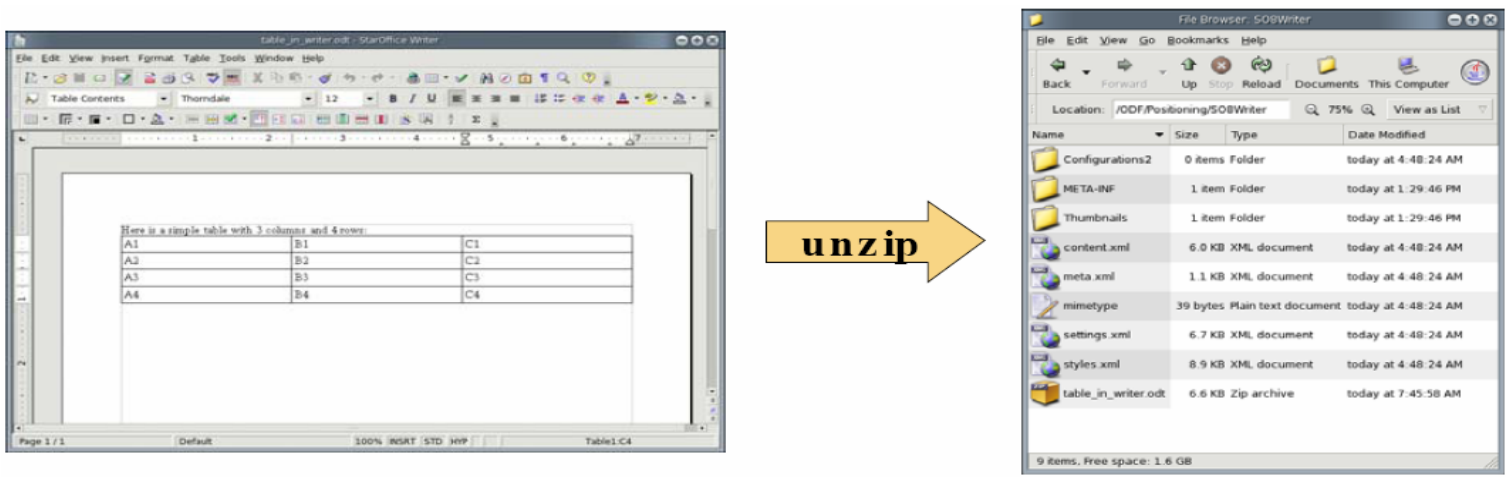
Breu història

Data / Període	Fet / Fita
1999	StarDivision desenvolupa un format d'arxiu basat en XML
Agost 1999	SUN adquireix StarDivision
13 d'octubre de 2000	SUN allibera StarOffice sota llicència lliure
13 d'octubre de 2000	S'estableix OpenOffice.org amb l'objectiu de definir l'especificació del format d'arxiu XML
2002	El comitè OASIS convoca la seva primera conferència
Maig de 2002	Es publica OpenOffice 1.0 i StarOffice 6
Maig de 2005	ODF es aprovat com a estàndard d'OASIS
2006	ODF es aprovat com a estàndard del comitè ISO/IEC

Format ODF

Arquitectura I

- Aprofita formats existents, no torna a inventar la roda!
- Tots els documents ODF es basen en un contenidor ZIP i una estructura comú.



Format ODF

Arquitectura II

- Estructura d'un document ODF
  - Arxiu **content.xml**
    - Emmagatzema el contingut del document
    - Conté les referències als esquemes usats
    - Conté les fonts i estils personalitzats per l'usuari
  - Arxiu **styles.xml**
    - Conté les fonts i estils predeterminats del document
  - Arxiu **settings.xml**
    - Conté els paràmetres de configuració
  - Arxiu **meta.xml**
    - Conté les metadades d'informació
  - Arxiu **mimetype.xml**
    - Indica el tipus de document
  - Directori **pictures**
    - Conté les imatges del document

Format ODF

Formats competidors

- PDF / A
  - Propietat d'Adobe Systems
  - Document portable però no indicat per emmagatzemament oficial al no complir tots els requeriments
  - Estàndard ISO/IEC 19005-1:2005
- OpenXML
  - El creador és Microsoft però és lliure de costs
  - Competidor directe d'ODF
  - Arquitectura similar: contenidor ZIP i estructura comú entre tots els documents
  - Estàndard ISO/IEC DIS 29500

Format ODF

Amenaces de seguretat

- Es poden incloure elements propietaris
  - La inclusió d'elements propietaris alteraria les propietats del format
- Es pot xifrar part o tot el contingut
  - Permetria ocultar dades o informació per a usos fraudulents
    - Mitjançant l'esteganografia
  - Permetria impossibilitar la lectura del missatge original
    - Mitjançant la criptografia
- Amb l'ajut de l'esteganalisi i patrons determinats es pot localitzar fragments on s'ha aplicat l'esteganografia com veurem en el programari del projecte.

Programari

Objectius

- Eina automàtica per extreure informació d'un document OpenOffice
- Usarà
  - Arxiu en format XML amb patrons de cerca per a extreure informació d'un document de text ODF
  - Magatzem en format XML vàlid que representen dades sospitoses detectades per l'Agència Mundial de Seguretat
- Generarà
  - Un arxiu en format XML vàlid amb els resultats
  - Mitjançant una transformació XSL generarà un arxiu en format HTML per a presentar els resultats a l'usuari

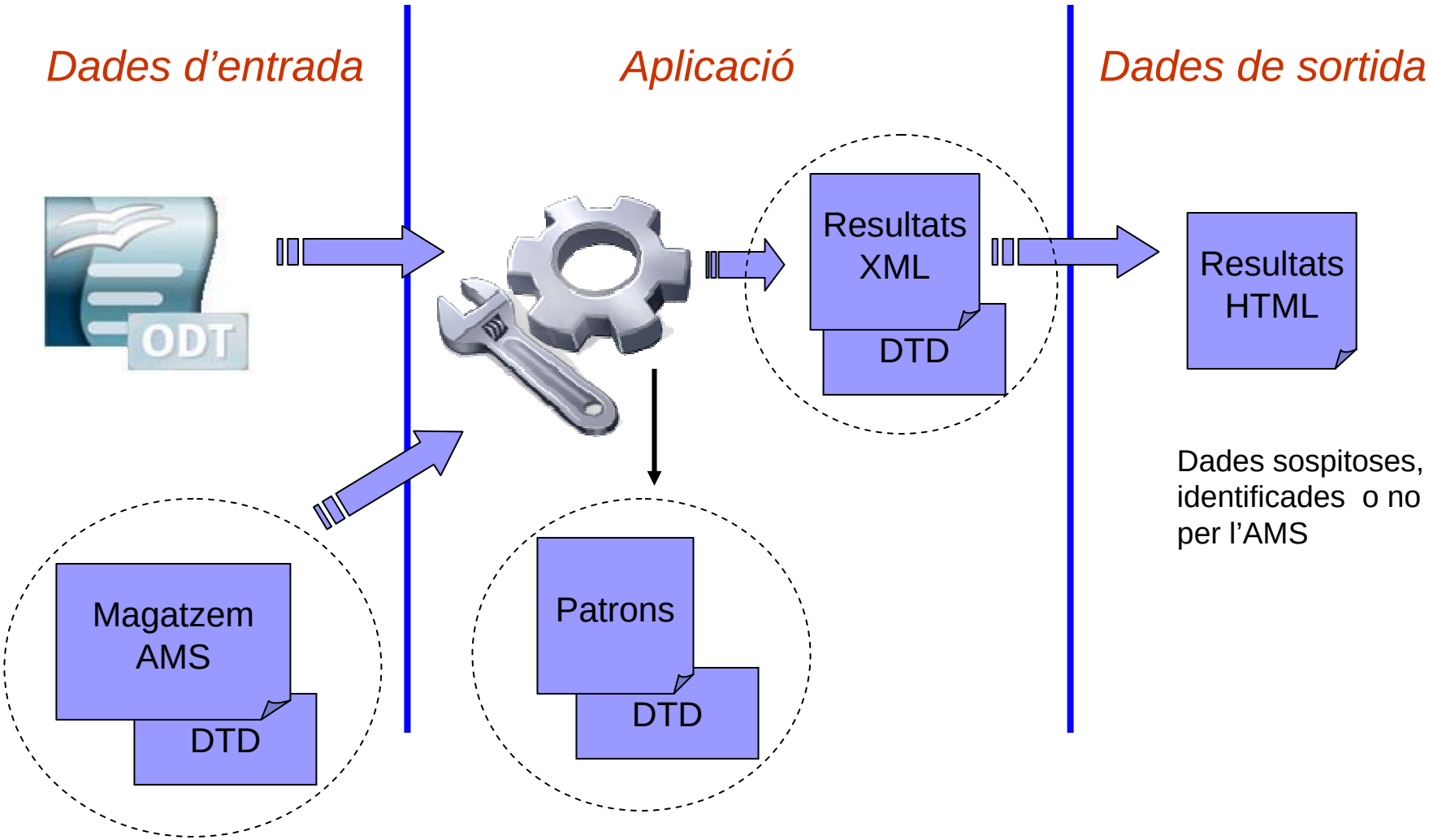
Programari

Arquitectura

- Aspectes tecnològics
  - L'eina s'ha desenvolupat sobre la plataforma .NET de Microsoft
  - C# és el llenguatge escollit de la plataforma
- Desplegament
  - Màquines amb sistemes operatius Windows
  - Amb la plataforma .NET 2.0 instal·lada

Programari

Disseny



Programari

Disseny

- Obertura del document a tractar

```
public Document(string uri)
{
    try
    {
        m_document = new AODL.Document.TextDocuments.TextDocument();
        m_document.Load(uri);
```

- Validació de les dades de l'AMS

```
// Estableix la configuració per a la validació.
XmlReaderSettings configuracio = new XmlReaderSettings();
configuracio.ProhibitDtd = false;
configuracio.ValidationType = ValidationType.DTD;
configuracio.ValidationEventHandler += new ValidationEventHandler(ValidationCallBack);

// Crea l'objecte XmlReader a partir de la uri del magatzem AMS.
XmlReader reader = System.Xml.XmlReader.Create(m_uri, configuracio);

// Parseja l'arxiu.
while (reader.Read())
{
    if (reader.NodeType == XmlNodeType.Text)
    {
        // S'afegeix la dada al magatzem (hashtable) de dades sospitoses.
        m_repositori.Add(reader.Value, reader.Value);
    }
};
```

Programari

Disseny

a) Recorregut en profunditat de tots els elements del contingut del document cercant els de tipus paràgraf

b) Per a tot element paràgraf es valida si és una ocurrència segons l'arxiu de patrons:

1. Si ho és → es valida si és una dada positiva de l'AMS
   - S'afegeix la informació generada a l'arxiu de resultats
2. Si no ho és passem al següent element paràgraf

```
public override void Validar(Document document)
{
    m_resultat.AfegirMetadades(document.Nom);

    IEnumerator enumerador = document.Contingut.GetEnumerator();
    while (enumerador.MoveNext())
    {
        if (enumerador.Current is Paragraph)
        {
            this.TractarParagraf((Paragraph) enumerador.Current);
        }
        else if (enumerador.Current is Table)
        {
            this.TractarTaula((Table) enumerador.Current);
        }
        else if (enumerador.Current is List)
        {
            this.TractarLlista((List) enumerador.Current);
        }
    }
}
```

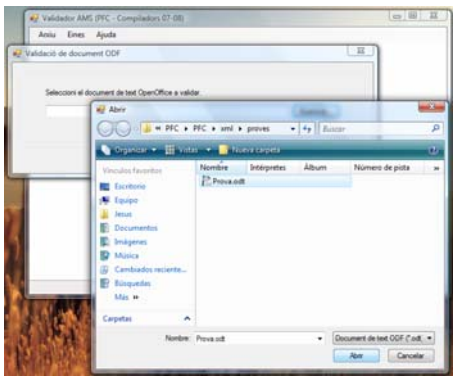
```
private void TractarParagraf(Paragraph paragraf)
{
    Hashtable patrons = m_patro.ObtenirPatrons();
    Regex regex = null;
    Match match = null;
    bool resultat = false;

    foreach (DictionaryEntry item in patrons)
    {
        regex = new Regex(item.Value.ToString());
        if (regex.IsMatch(paragraf.Node.InnerText))
        {
            match = regex.Match(paragraf.Node.InnerText);
            resultat = this.ValidarDadaCompromesa(match.Value);
            m_resultat.AfegirResultat(m_numParagraf, resultat,
                                      match.Value, item.Key.ToString());
        }
    }
    m_numParagraf++;
}
```

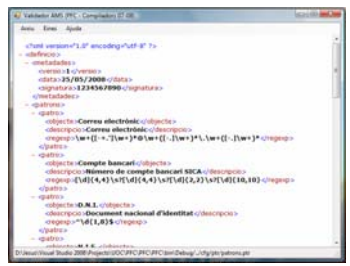
Programari

Execució de l'aplicació i resultat final

Obrir document

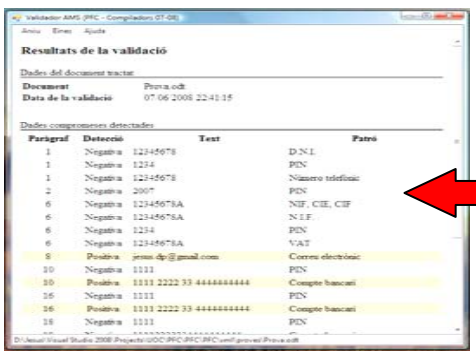


L'usuari pot configurar l'arxiu de patrons



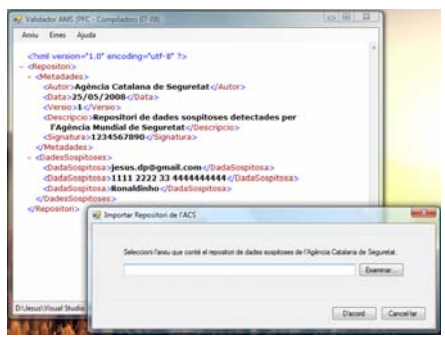
Resultats del procés en XML

```
<?xml version="1.0" ?>
- <Validacio>
  - <Resultats>
    - <Resultat>
        <Paragraf>10</Paragraf>
        <DataSospitosaAMS>True</DataSospitosaAMS>
        <Text>1111 2222 33 4444444444</Text>
        <Patro>Compte bancari</Patro>
      </Resultat>
    - <Resultat>
        <Paragraf>24</Paragraf>
        <DataSospitosaAMS>False</DataSospitosaAMS>
        <Text>correu@correu.com</Text>
        <Patro>Correu electrònic</Patro>
      </Resultat>
    - <Resultat>
        <Paragraf>25</Paragraf>
        <DataSospitosaAMS>False</DataSospitosaAMS>
        <Text>12345678</Text>
        <Patro>D.N.I.</Patro>
      </Resultat>
    - <Resultat>
        <Paragraf>27</Paragraf>
        <DataSospitosaAMS>False</DataSospitosaAMS>
        <Text>www.lacaixa.es</Text>
        <Patro>Noms de domini RFC 1035</Patro>
      </Resultat>
    </Resultats>
  - <Metadades>
      <NomDocument>Prova.odt</NomDocument>
      <DataValidacio>28/05/2008 23:41:24</DataValidacio>
    </Metadades>
</Validacio>
```



Resultats en format HTML després de transformar

L'usuari pot configurar el magatzem de l'AMS



Conclusions

- Format XMLS
  - És apte per a la transmissió de dades de forma estructurada
  - Els DTD ens permeten validar els documents en format XML
  - Les transformacions XSL permeten que un document XML es transformi en un altre tipus de document XML per exemple HTML
- Expressions regulars
  - Les expressions regulars possibiliten la cerca dins de cadenes de caràcters mitjançant motors de cerca
- Format ODF
  - És el primer estàndard per a documents digitals d'oficina
  - És lliure i obert i reutilitza estàndards ja existents
  - Es basa en un contenidor ZIP estructurat de forma comú a tots els documents
- Fites aconseguides
  - Hem investigat els formats XML, ODF, les transformacions XSL i les expressions regulars
  - Hem posat en pràctica aquests coneixements amb una aplicació que permet l'extracció de continguts d'un document de text ODF mitjançant patrons definits amb expressions regulars

Bibliografia

- Investigació
  - XML
    - http://es.wikipedia.org/wiki/XML
    - http://es.wikipedia.org/wiki/DTD
    - http://www.cafeconleche.org/books/effectivexml/
  - Transformacions XSL
    - http://msdn.microsoft.com/es-es/library/ms256069(VS.80).aspx
  - Expressions regulars
    - http://www.microsoft.com/spanish/msdn/articulos/archivo/201205/voices/regex.mspx
    - http://java.sun.com/docs/books/tutorial/essential/regex/index.html
    - http://www.regular-expressions.info/tutorial.html
- Programari
  - .NET
    - http://www.microsoft.com/express/support/faq/
  - C#
    - Microsoft Press, .NET Framework 2.0 Application Development Foundation
  - ODF Toolkit
    - http://wiki.services.openoffice.org/wiki/ODF_Toolkit