

Construcció d'una eina pel tractament automatitzat de fitxers OpenOffice

Xavier Graell Gras
Enginyeria en Informàtica

Jordi Ferrer Duran

15 de juny del 2007

1 Resum

Aquí comença la redacció d'aquest document, la memòria del PFC, que tracta sobre la *Construcció d'una eina pel tractament automatitzat de fitxers OpenOffice*. A continuació es mostren diferents apartats en els quals s'ha recollit tot el treball realitzat per l'alumne durant el semestre i que ha estat necessari per assolir els objectius marcats inicialment. Cal remarcar que també s'inclou part del treball que es va realitzar en el semestre anterior, doncs aquest PFC ja va ser planificat per a desenvolupar-se durant el semestre de tardor del 2006-2007. Malauradament, no es van arribar a les fites mínimes necessàries en les dates establertes, i durant aquest semestre s'ha prosseguit a finalitzar les que havien quedat pendents.

El projecte pot ser dividit en dues grans etapes. La primera d'elles és el resultat del primer treball realitzat durant el semestre anterior, el qual correspon a un anàlisi detallat dels fitxers que poden ser generats amb el paquet OpenOffice i del format que utilitzen tots ells, l'OASIS OpenDocument, el qual es basa en un esquema XML. Tal i com es podrà descobrir en els capítols posteriors d'aquest document, qualsevol fitxer OpenOffice, ja sigui de text, full de càlcul o un altre, està format per diferents subdocuments comprimits sota un de sol. Sobre el format OASIS OpenDocument se'n pot destacar la gran quantitat de tags que incorpora per a permetre totes les funcionalitats que ha d'oferir un programari potent de processament de text, full de càlcul i altres programaris d'oficina. També s'inclou dins aquesta primera etapa l'estudi i desenvolupament dels arxius DTD que s'encarreguen de validar els corresponents arxius XML resultants de la transformació realitzada pel programari.

La segona etapa correspon, principalment, al treball realitzat durant aquest últim semestre, el qual s'ha basat en dissenyar i construir el programari de la forma més adequada possible. Primerament es va realitzar un estudi de les diferents alternatives disponibles actualment en el mercat capacitades per tractar arxius XML. Es va poder observar que la gran majoria de plataformes i llenguatges de programació actuals disposen de funcionalitats per tractar arxius XML, la qual cosa deixa constància del gran poder d'acceptació que ha tingut l'XML. Entre les plataformes disponibles es va escollir la variant Visual J# de .NET que es troba dins el paquet Visual Studio 2005 Express Edition. Les raons principals d'aquesta elecció van ser la gran quantitat de mètodes i components disponibles, la fàcil integració amb el sistema operatiu Windows, i la interfície amigable que té per al desenvolupador. Tot seguit es va crear el diagrama de processos i es van començar a construir cada un d'ells. Entre els processos que inclou el programari se'n poden destacar els següents:

- Descompressió fitxer OpenOffice: L'arxiu OpenOffice subministrat es descomprimirà per generar tots els arxius que el componen. Aquests fitxers seran emmagatzemats en una carpeta temporal.
- Anàlisi fitxers: Lectura dels fitxers on es troben els termes que s'hauran d'indexar. Cada vegada que se'n llegeixi un, aquest s'haurà de registrar en una estructura interna, juntament amb informació addicional. L'anàlisi es realitzarà amb l'ajuda de l'API SAX disponible en el Visual J#.
- Generació XML amb índex de termes: A partir de l'estructura interna on s'han registrat tots els termes trobats, es crearà o modificarà el fitxer que emmagatzemarà físicament els termes a indexar amb un format específic.
- Modificació fitxer per incloure-hi l'índex: Amb l'ajuda de l'API DOM disponible al Visual J# es modificaran els arxius que componen l'arxiu OpenOffice per tal d'incloure l'índex de termes al final del document amb el format especificat per l'usuari.
- Compressió fitxer OpenOffice: Tots els arxius que havien estat descomprimits i que han patit modificacions, hauran de tornar a ser comprimits sota el mateix arxiu OpenOffice.

Evidentment, tots aquests processos seran descrits amb més detall en posteriors capítols d'aquest document.

2 Índex

1	Resum	2
2	Índex	3
3	Cos.....	4
3.1	Introducció	4
3.1.1	Definició del projecte	4
3.1.2	Estructura del projecte i descomposició en activitats.....	6
3.1.3	Organització del projecte	7
3.1.4	Descripció següents capítols.....	11
3.2	Anàlisi fitxers OpenOffice	12
3.2.1	Introducció	12
3.2.2	Estandarització i llicència	12
3.2.3	Tipus de fitxers	13
3.2.4	Formats interns.....	14
3.3	Anàlisi format OpenDocument	17
3.3.1	Introducció	17
3.3.2	Estructura	17
3.4	Alternatives per manipular documents OpenOffice	50
3.5	Construcció del programari	52
3.5.1	Diagrama de processos.....	52
3.5.2	Descripció de processos	53
3.6	Construcció DTDs	62
3.7	Conclusions	63
4	Bibliografia	64

3 Cos

3.1 Introducció

3.1.1 Definició del projecte

Marc de referència

El projecte "**Construcció d'una eina pel tractament automatitzat de fitxers OpenOffice**" neix fruit de la necessitat per part de l'alumne de desenvolupar el projecte de final de carrera dels estudis de "2n cicle de l'Enginyeria d'Informàtica", després d'haver obtingut la totalitat dels crèdits de cursar les assignatures de la carrera.

La idea és que el projecte serveixi com a treball per a posar en pràctica els coneixements que ha anat adquirint l'alumne al llarg de la carrera, posant èmfasi en aquells que estan relacionats amb les assignatures de Compiladors I i Compiladors II.

El projecte no parteix de zero, ja que el seu desenvolupament es va iniciar el semestre anterior. Per tant, aquest projecte és la continuació d'un primer treball que no va assolir els objectius esperats, i del qual se n'aprofitaran totes aquelles parts que ja estiguessin correctes i acabades amb anterioritat.

Terminis establerts

La planificació d'aquest projecte està basada en la necessitat de complir les dates d'entrega preestablertes ja des de la publicació de la proposta del projecte.

Aquestes dates estableixen que:

- Pel 15 de març del 2007 s'ha d'haver definit el pla de treball.
- Pel 16 d'abril del 2007 es farà entrega al consultor de la segona PAC.
- Pel 14 de maig del 2007 es farà entrega al consultor de la tercera PAC.
- Pel 15 de juny del 2007 s'ha d'haver finalitzat i entregat el projecte.
- Entre els dies 18 i 22 de juny del 2007 es farà el debat i la presentació virtual del projecte.

Anàlisi dels recursos

Abans de procedir a l'anàlisi de recursos que es necessitaran pel correcte desenvolupament del projecte, serà interessant fer un estudi de tots els recursos dels quals disposa l'alumne.

Recursos existents

- **Maquinari.** L'alumne aprofitarà l'ordinador amb connexió a Internet via ADSL del que disposa, amb el qual ja va realitzar un primera versió del projecte.
- **Programari.** L'alumne disposa del programari instal·lat a l'ordinador que ja va utilitzar en la primera versió del projecte.
- **Recursos humans.** Els recursos humans amb els que conta l'alumne són ell mateix. No hi ha la possibilitat de poder obtenir recursos humans addicionals, doncs aquest projecte l'ha de desenvolupar l'alumne de manera individual.

Necessitats de nous recursos

Degut al treball iniciat durant el semestre anterior, l'alumne ja disposa de gran quantitat de recursos per continuar amb el desenvolupament del projecte, entre ells el programari OpenOffice, el qual és programari de lliure distribució. De totes maneres, no es desestima la possibilitat de més recursos en programari, depenent de la tecnologia escollida per desenvolupar l'eina.

Objectius del projecte

Aquest projecte té com a objectiu principal la construcció d'una eina pel tractament automatitzat de fitxers OpenOffice. Però, per aconseguir aquesta eina, establirem una altra sèrie d'objectius, el conjunt dels quals ens faran obtenir l'objectiu principal. Aquests són:

- Conèixer el format de document OpenOffice
- Estudiar les alternatives existents per llegir i manipular documents OpenOffice
- Implementar una eina que llegeixi el document OpenOffice i hi realitzi una certa transformació
- Treballar amb documents XML i amb DTDs associats

Resultats esperats

Els resultats que s'esperen del treball de l'alumne per a la realització d'aquest projecte són:

- *Pla de treball.* Es tracta del document en el qual es descriu el projecte i s'especifica les tasques que formaran el projecte, planificades en el temps.
- *Memòria.* Es tracta d'aquest document. És on es recull, de manera sintetitzada, tota la informació rellevant de la feina que ha dut a terme l'alumne durant la realització del projecte.
- *Producte.* És l'eina de programari amb la qual es podrà tractar automatitzadament els fitxers d'OpenOffice.
- *Presentació virtual del projecte.* Es tracta d'un document de presentació on s'explicarà de forma clara i sintetitzada les característiques més importants del projecte i la manera com s'han realitzat.

Anàlisi de riscos

Amb la informació disponible s'identifica un risc, el qual pot ser objecte del no assoliment dels objectius planificats en aquest document. Val a dir que aquest risc ja va ser el responsable del no assoliment de l'èxit en el primer treball realitzat el semestre anterior. Estic parlant de:

- *Falta de temps*
L'alumne es troba en una situació laboral en la qual es pot veure obligat a donar-li prioritat a la feina i no disposar del temps suficient per complir els terminis establerts d'entrega de les diferents fites del projecte. L'endarreriment d'aquestes suposaria el fracàs del projecte.
- *Inexperiència amb el llenguatge de programació per construir el programari*
L'alumne no disposa d'experiència sobre la construcció i desenvolupaments amb el llenguatge de programació que s'ha escollit. La investigació de classes i funcions amb les que programar pot ser una font de pèrdua de temps.

3.1.2 Estructura del projecte i descomposició en activitats

Una vegada definit el projecte i coneguts alguns dels aspectes més rellevants d'aquest, procedirem a la seva descomposició en activitats.

Estructura del projecte

Tenint en compte els condicionants que s'exposen a continuació, utilitzarem una metodologia de desenvolupament en cascada:

- El projecte es pot definir amb bastant precisió, gracies a la proposta i a la descripció del projecte.
- Es tracta d'una aplicació transaccional en temps real.
- L'equip de treball és expert en aquesta metodologia, doncs el seu únic membre té un perfil qualificat.

Per a l'estimació de l'esforç ens basarem en una descomposició en activitats (WBS), baixant un únic nivell en la descomposició amb el volum d'activitats que tenim previst.

Descomposició en activitats

Les activitats que compondran el projecte són les següents:

Descomposició estructural d'activitats (WBS)		
Codi activitat	Activitat del nivell 1	Activitat del nivell 2
01	Inici del projecte	
02	Gestió del projecte	
03	Anàlisi fitxers OpenOffice	
04	Estudi alternatives manipulació documents OpenOffice	
05	Construcció programari	
05.01		Construcció interfície
05.02		Validacions paràmetres
05.03		Descompressió fitxer
05.04		Analitzador
05.05		Generació XML amb índex termes
05.06		Incloure índex termes final fitxer
05.07		Compressió fitxers
05.08		Neteja carpeta temporal
05.09		Proves
06	Construcció DTDs de validació	
07	Final del projecte	

El projecte té en compte estrictament la construcció del programari sol·licitat. També val a dir que la primera i la darrera activitat (Inici del projecte; Final del projecte) no tindran contingut, i es posen únicament per qüestions de claredat en l'etapa de planificació.

3.1.3 Organització del projecte

Relació d'activitats

A continuació es detallen cada una de les tasques, partint de la base que el projecte té una durada prevista de 135 hores, corresponents als 9 crèdits (15 hores per crèdit).

Amb això, i complint amb les dates preestablerts, tindrem un esforç de 10 hores per setmana.

1) Anàlisi fitxers OpenOffice

Esforç: 2 hores

Durada: 1 dia

Descripció: Aquesta tasca consistirà en donar una repassada a la documentació existent sobre l'anàlisi dels fitxers OpenOffice generada en el treball anterior.

Resultat: Document amb un estudi dels fitxers OpenOffice i del format OASIS OpenDocument.

2) Estudi alternatives manipulació documents OpenOffice

Esforç: 12 hores

Durada: 6 dies

Descripció: De les tecnologies existents al mercat capaces de manipular fitxers OpenOffice, fer un estudi de quina d'elles pot ser la més convenient per implementar aquest projecte.

Resultat: Document amb l'estudi i l'anàlisi de les possibles tecnologies amb les quals fer la implementació.

3) Construcció interfície

Esforç: 4 hores

Durada: 2 dies

Descripció: Construir la interfície gràfica del programari de manera usable i amigable per a l'usuari, i que a més, inclogui tota la informació necessària per a dur a terme la generació de l'índex.

Resultat: Arxiu del projecte que representa el formulari amb el qual interactuarà l'usuari

4) Validacions paràmetres

Esforç: 4 hores

Durada: 2 dies

Descripció: Dels paràmetres d'entrada proporcionats per l'usuari, pot ser que n'hi hagi d'incorrectes o que no s'hagin entrat. Abans de començar amb el procés, es validaran tots aquests paràmetres.

Resultat: Diferents funcions dins un arxiu del projecte que validaran els paràmetres d'entrada.

5) Descompressió fitxer

Esforç: 12 hores

Durada: 6 dies

Descripció: L'arxiu OpenOffice proporcionat per l'usuari i validat, serà descomprimit en una carpeta temporal per disposar de tots els arxius que el componen.

Resultat: Funció adjunta dins arxiu del projecte que s'encarrega de descomprimir l'arxiu que se li passa com a paràmetre en una carpeta temporal.

6) Analitzador

Esforç: 30 hores

Durada: 3 setmanes

Descripció: Analitzar els arxius que rebí com a entrada de principi a fi per trobar-hi termes. Cada terme cercat quedarà registrat en una estructura d'emmagatzematge, juntament amb informació addicional sobre el punt on s'ha trobat.

Resultat: Funció adjunta dins arxiu del projecte que s'encarrega de cercar i emmagatzemar tots els termes que s'hauran d'indexar posteriorment.

7) Generació XML amb índex de termes

Esforç: 12 hores

Durada: 6 dies

Descripció: Generació d'un fitxer XML, el nom del qual serà especificat per l'usuari, que contindrà tots els termes que s'indexaran al final del fitxer OpenOffice. L'estructura d'aquest fitxer XML és sempre la mateixa.

Resultat: Funció adjunta dins arxiu del projecte que s'encarrega de generar un fitxer XML a partir de l'estructura interna on s'han emmagatzemat tots els termes cercats.

8) Incloure índex termes final fitxer

Esforç: 25 hores

Durada: 2 setmanes i mitja

Descripció: Manipulació dels arxius descomprimits a partir de l'arxiu OpenOffice, de tal manera que s'inclourà l'índex de termes amb el format especificat per l'usuari i pel propi format OpenDocument.

Resultat: Funció adjunta dins arxiu del projecte que s'encarrega de crear tota aquella informació necessària dins els arxius corresponents per tal que el fitxer OpenOffice inclogui l'índex de termes al final.

9) Compressió fitxers

Esforç: 12 hores

Durada: 6 dies

Descripció: Compressió dels fitxers, abans descomprimits i que han estat modificats, dins el mateix arxiu OpenOffice que havia subministrat l'usuari.

Resultat: Funció adjunta dins arxiu del projecte que s'encarrega de comprimir tots els arxius descomprimits dins l'arxiu OpenOffice inicial.

10) Neteja carpeta temporal

Esforç: 6 hores

Durada: 3 dies

Descripció: Tots els arxius que s'havien generat de descomprimir l'arxiu OpenOffice, poden ser eliminats un cop ja s'han tornat a comprimir dins l'arxiu OpenOffice.

Resultat: Funció adjunta dins arxiu del projecte que s'encarrega d'eliminar els arxius que ja han tornat a ser comprimits dins l'arxiu OpenOffice.

11) Proves

Esforç: 10 hores

Durada: 1 setmana

Descripció: Realització d'un pla de proves de l'eina ja construïda amb els corresponents resultats.

Resultat: Document amb les proves sobre l'eina registrades.

12) Construcció DTDs

Esforç: 6 hores

Durada: 3 dia

Descripció: Construcció de dos arxius amb extensió dtd, els quals s'han d'utilitzar per validar els fitxers XML resultants.

Resultat: Dos fitxers amb extensió dtd.

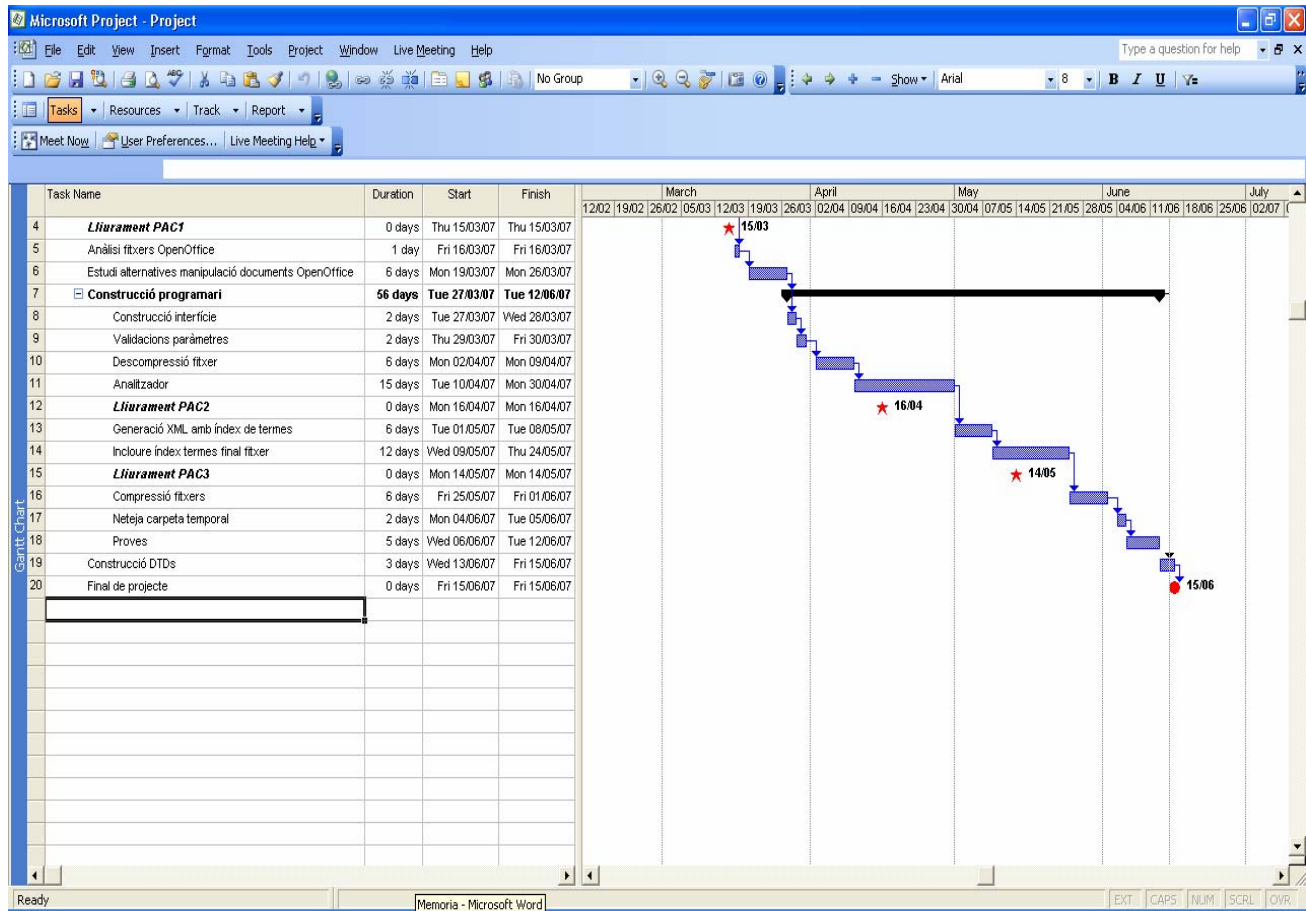
Planificació temporal del projecte

A continuació es mostra la planificació temporal del projecte per així tenir una guia de referència on veure com evoluciona el desenvolupament del projecte. Partim de la base que el projecte, pròpiament dit, comença el dia 1 de Març, quan es té la confirmació de poder continuar amb el treball realitzat el semestre anterior:

Setmana	Dates	Activitat	Esdeveniment
1	1 a 4 de Març	Presentació	
2	5 a 11 de Març	Pla de treball	
3	12 a 18 de Març	Pla de treball i Tasca 1	Lliurament Pla de treball a PAC1 el 15 de Març. Trobada presencial el 17 de Març
4	19 a 25 de Març	Tasca 2	
5	26 de Març a 1 d'Abril	Tasca 2, Tasca 3 i Tasca 4	
6	2 a 8 d'Abril	Tasca 5	
7	9 a 15 d'Abril	Tasca 5 i Tasca 6	
8	16 a 22 d'Abril	Tasca 6	Lliurament PAC2 el 16 d'Abril
9	23 a 29 d'Abril	Tasca 6	
10	30 d'Abril a 6 de Maig	Tasca 6 i Tasca 7	
11	7 a 13 de Maig	Tasca 7 i Tasca 8	
12	14 a 20 de Maig	Tasca 8	Lliurament PAC3 el 14 de Maig
13	21 a 27 de Maig	Tasca 8 i Tasca 9	
14	28 de Maig a 3 de Juny	Tasca 9	
15	4 a 10 de Juny	Tasca 10 i Tasca 11	
16	11 a 17 de Juny	Tasca 11 i Tasca 12. Revisió de la memòria i elaboració de la presentació	Lliurament final (memòria, presentació i producte) el 15 de Juny
17	18 a 24 de Juny	Defensa del projecte	Presentació virtual i debat

Calendari de treball

A continuació es mostra el calendari global del projecte:



Equip de treball

Tal i com ja s'ha comentat anteriorment en aquest document, l'alumne farà de tots els rols del projecte, és a dir, el de director del projecte, el de dissenyador funcional, el d'analista tècnic i el de programador i codificador.

3.1.4 Descripció següents capítols

Arribats a aquest punt ja es pot tenir una visió clara del que representa aquest projecte de final de carrera. S'ha fet una definició del projecte, descrivint-ne objectius, riscos i abast. S'han estructurat i organitzat totes les activitats i tasques, les quals s'han d'haver desenvolupat dins els marcs definits en espai i temps. I ara ens queda veure com s'han dut a la pràctica aquestes activitats amb un grau de detall més elevat.

Així, en els propers capítols podrem trobar les descripcions detallades de totes les tasques que ja han estat introduïdes en la definició del projecte. També s'inclouran comentaris sobre problemes que hi ha hagut i tot allò que s'ha cregut important de mencionar per cada una de les activitats realitzades.

Començarem el següent capítol dedicat a estudiar els fitxers OpenOffice i el format amb el qual es basen, l'OpenDocument. Es farà una introducció d'aquest format, el seu origen, els diferents tipus existents i la seva estructura. Durant el capítol posterior es farà un anàlisi més complet dels elements que componen l'estructura d'aquest format. Aquests dos primers capítols formen l'activitat d'*Anàlisi dels fitxers OpenOffice*.

El següent capítol correspon a la descripció de l'activitat d'*Estudiar alternatives per manipular documents OpenOffice*, i en ell es valoren les APIs actuals per tractar arxius XML i es decideix la plataforma o llenguatge de programació amb el qual s'implementarà el programari. També s'introdueix la manera com s'utilitzaran els mètodes de les API per a construir el programari.

El capítol següent està dedicat al treball realitzat durant la *Construcció del programari*. Comencem mostrant el diagrama de processos, i després es passa a analitzar cada un dels processos en més detall, oferint captures de pantalla i casuístiques de funcionament en funció de les accions que es facin.

Continuarem amb el capítol dedicat a la *Construcció de DTDs*. En ell s'analitza el format que s'ha utilitzat per emmagatzemar l'índex de termes dins un arxiu XML, i es comenta la manera com s'ha construït el DTD que valida un arxiu amb format OpenDocument.

Finalment, acabarem el document de memòria d'aquest projecte de final de carrera amb el capítol destinat a les conclusions que es poden extreure de reflexionar sobre la feina feta durant aquest projecte.

3.2 Anàlisi fitxers OpenOffice

3.2.1 Introducció

El propòsit d'aquest capítol és fer un anàlisi del format *OpenDocument*, el qual és un format de fitxer estàndard usat actualment per emmagatzemar documents d'ofimàtica com fulls de càlcul, gestors de textos o presentacions.

Aquest format és el que s'utilitza en els fitxers d'OpenOffice, un paquet de programari de lliure distribució que conté eines ofimàtiques com gestor de textos, full de càlcul o presentacions. En la versió 2.0 de l'OpenOffice, aquest format és anomenat *OASIS OpenDocument file format*.

El desenvolupament del format ha estat encomanat a l'organització OASIS (acrònim de *Organization for the Advancement of Structured Information Standards*, consorci internacional sense ànims de lucre que orienta el desenvolupament, la convergència i l'adopció d'estàndards d'e-business) i que es basa en un esquema XML (acrònim de *eXtensible Markup Language*, metallenguatge extensible d'etiquetes desenvolupat per la World Wide Web Consortium – W3C) inicialment creat per OpenOffice.org. (paquet d'ofimàtica de programari lliure i codi obert)

El format *OpenDocument* va ser aprovat com a estàndard OASIS l'1 de Maig del 2005. Així mateix va ser aprovat el 3 de Maig del 2006 com a esborrany de l'estàndard ISO 26300. A data de 9 d'octubre del 2006 es troba en la fase número 50 del procés d'estandarització, la fase final per a ser aprovat. L'estàndard, que va ser desenvolupar públicament per un grup d'organitzacions, és accessible públicament, i pot ser implementat per a qualsevol sense restricció.

El format *OpenDocument* té com a objectiu oferir una alternativa oberta als formats de documents propietat d'un fabricant, els quals tenen uns requisits de llicències que impedeixen el seu ús a varis competidors.

La motivació principal per utilitzar formats estàndard és que les organitzacions i individus que ho fan eviten la dependència d'un únic proveïdor de programari, permetent així canviar d'entorn informàtic si el seu proveïdor actual és expulsat del mercat o canvia el seu de model de llicència en termes menys favorables pel client.

OpenDocument és el primer estàndard per a documents d'ofimàtica implementat per diferents competidors, visat per organismes d'estandarització independents i susceptible de ser implementat per qualsevol proveïdor.

3.2.2 Estandarització i llicència

La versió 1.0 de l'especificació d'*OpenDocument* va ser aprovada com a estàndard OASIS al Maig del 2005, i està disponible per ser descarregat i utilitzat lliurement. L'especificació es pot llicenciar en "termes" recíprocs per qualsevol part, sempre i quan no estigui sotmesa a quotes. Els "termes" de la llicència són equivalents als promoguts per altres organitzacions de normalització (com el W3C), i pretén evitar el conflicte entre les qüestions relatives a la propietat intel·lectual i la promoció de la innovació tecnològica.

El procés d'estandarització inclou a desenvolupadors de moltes aplicacions ofimàtiques o relacionats amb sistemes de documentació, incloent:

- Adobe (Framemaker, Distiller)
- Corel (Word Perfect)
- IBM (Lotus 1-2-3, Workplace)
- Sun Microsystems / OpenOffice.org (StarOffice/OpenOffice.org)

El procés d'estandarització d'*OpenDocument* també inclou a molts usuaris, especialment aquells que tinguin la necessitat d'utilitzar documents complexos, o de poder recuperar-los bastant temps després d'haver estat creats. Alguns dels usuaris implicats en el procés d'estandarització van ser:

- Boeing (documents grans i complexos)
- Intel (documents grans i complexos; desenvolupen textos de prova)
- National Archive of Australia (accés a documents després de molt temps després de la seva composició)
- New York State Office of the Attorney General (documents grans i complexos que requereixen accés molt temps després de la seva composició)
- Society of Biblical Literature (grans documents políglotes, amb capacitat d'accés després de molt temps)

A més de la participació de molts membres formals, les versions provisionals de l'especificació van ser revelades al públic i subjectes a revisió mundial. Molts participants, que no eren membres formals del comitè d'estandarització, van enviar comentaris al comitè, i aquests van ser avaluats en públic per comitè.

En tot aquest procés, Microsoft va estar notablement absent, especialment perquè Microsoft és un membre d'OASIS i és el venedor dominant de programari d'oficina. Pel contrari, contra diverses convocatòries d'institucions públiques i privades, Microsoft va decidir desenvolupar per separat el seu propi format incompatible, sense consulta o revisió externa. Degut a aquesta carència de revisió independent i pública externa del format de Microsoft, molts creuen que serà més difícil que altres implementin el format de Microsoft, o pensen que el format de Microsoft té mancances de capacitats importants especificades en *OpenDocument*. Microsoft també imposa condicions addicionals de llicència per usuaris del seu format; molts creuen que aquestes condicions addicionals inhibeixen la competència, i formen part de l'intent de Microsoft de protegir el seu monopoli virtual en aplicacions ofimàtiques; altres esperen que els fets obligaran a Microsoft a desistir de l'intent d'imposar el seu propi format, de la mateixa manera que es va veure obligat a la dècada del 1990 a abandonar els seus estàndards d'Internet per adoptar estàndards oberts.

3.2.3 Tipus de fitxers

En la taula següent es mostren les diferents extensions, juntament amb el tipus de fitxer al qual corresponen els tipus de fitxers amb format *OpenDocument*.

Documents

Tipus de fitxer	Extensió	Tipus Mime
Text	.odt	application/vnd.oasis.opendocument.text
Fulla de càlcul	.ods	application/vnd.oasis.opendocument.spreadsheet
Presentació	.odp	application/vnd.oasis.opendocument.presentation
Dibuix	.odg	application/vnd.oasis.opendocument.graphics
Gràfica	.odc	application/vnd.oasis.opendocument.chart
Fórmula matemàtica	.odf	application/vnd.oasis.opendocument.formula
Base de dades	.odb	application/vnd.oasis.opendocument.database
Imatge	.odi	application/vnd.oasis.opendocument.image
Document mestre	.odm	application/vnd.oasis.opendocument.text-master

Plantilles

Tipus de fitxer	Extensió	Tipus Mime
Text	.ott	application/vnd.oasis.opendocument.text-template
Fulla 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

3.2.4 Formats interns

L'estructura d'un document amb format *OpenDocument* està formada per varis components, cada un dels quals és representat per un element juntament amb els seus atributs associats. Així, aquesta estructura s'aplica a qualsevol dels tipus de fitxers que utilitzen el format, és a dir, que l'estructura és independent del tipus de fitxer (text, fulla de càlcul, presentació, etc), el que varia és el seu contingut.

El format *OpenDocument* suporta dues maneres diferents per a representar un document:

- En un únic document XML
- En una col·lecció de varis subdocuments empaquetats, cada un dels quals emmagatzema una part del document complet

El motiu de que *OpenDocument* permeti empaquetar els subdocuments és degut a que XML no suporta objectes binaris, com imatges o objecte OLE, i a més, perquè un fitxer XML sense comprimir ocupa molt espai (per exemple un fitxer XML de 4 Mb passa a ser de 0'3 Mb quan es comprimeix).

D'aquesta manera *OpenDocument* aconsegueix no només empaquetar tots els subdocuments sota un únic fitxer, sinó que a més, aquest els conté de forma comprimida. No estudiarem la forma com s'empaqueten i es comprimeixen els subdocuments, tot i que sí podem dir que usant un descompressor típic com WinZip sobre un arxiu amb extensió *odt*, per exemple, obtindrem els subdocuments associats a aquest fitxer.

Actualment, sembla que la forma més utilitzada per OpenOffice per emmagatzemar els seus fitxers en format *OpenDocument* és mitjançant l'empaquetament dels diferents subdocuments.

Anem doncs a estudiar aquests subdocuments que s'obtenen a partir de descomprimir un fitxer d'OpenOffice:

Fitxers XML: content.xml; meta.xml; settings.xml; styles.xml

Altres fitxers: mimetype; layout-cache

Directoris: .META-INF/
.Thumbnails/
.Pictures/
.database/
./Configurations2/
./Configurations2/accelerator/
./Configurations2/floater/
./Configurations2/images/
./Configurations2/images/Bitmaps/
./Configurations2/menuubar/
./Configurations2/popupmenu/

```
./Configurations2/progressbar/  
./Configurations2/statusbar/  
./Configurations2/toolbar/
```

Com es pot observar per la distribució de directoris i arxius, el format *OpenDocument* ofereix una clara separació entre el contingut, el format i les metadades. Analitzem-ne els seus components:

- **content.xml**

Aquest és el fitxer més important. Emmagatzema el contingut real del document (excepte les dades binàries, com les imatges). El format de base utilitzat va ser inspirat per HTML, tot i que és bastant més complex que aquest, i hauria de ser raonablement llegible per a un humà. Exemple:

```
<text:h text:style-name="Heading_2">Capcelera</text:h>  
<text:p text:style-name="Text_body"/>  
<text:p text:style-name="Text_body">  
    Això és un paràgraf. La informació sobre el format  
    s'emmagatzema al fitxer d'estil. La marca buida text:p de més  
    amunt és un paràgraf en blanc (línia buida).  
</text:p>
```

- **styles.xml**

OpenDocument fa un ús intensiu dels estils per a donar format i disposar el contingut pel document. La major part de la informació d'estil s'emmagatzema en aquest fitxer (tot i que part d'aquesta apareix dins el fitxer *content.xml*). Tal i com estudiarem, hi ha diferents tipus d'estil, que inclouen els següents:

- Estils de paràgraf
- Estils de pàgina
- Estils de caràcters
- Estils de marc
- Estils de llista

El format *OpenDocument* és únic pel fet que no es poden evitar l'ús d'estils per donar format als documents. Inclús el format "manual" es realitza mitjançant estils (que l'aplicació ofimàtica ha de crear dinàmicament segons sigui necessari).

- **meta.xml**

Aquest fitxer conté les metadades del document. Per exemple, l'autor, la identificació de l'última persona que l'ha modificat, la data de l'última modificació, etc. El contingut té un aspecte semblant a això:

```
<meta:creation-date>2003-09-10T15:31:11</meta:creation-date>  
<dc:creator>Daniel Carrera</dc:creator>  
<dc:date>2005-06-29T22:02:06</dc:date>  
<dc:language>es-ES</dc:language>  
<meta:document-statistic  
    meta:table-count="6" meta:object-count="0"  
    meta:page-count="59" meta:paragraph-count="676"  
    meta:image-count="2" meta:word-count="16701"  
    meta:character-count="98757"/>
```

- **settings.xml**

Aquest fitxer inclou propietats com el factor de zoom o la posició del cursor que afecten al inici d'obrir el document, però no són continguts que afectin a la disposició d'aquest en el document.

- **mimetype**
Aquest fitxer inclou una única línia que conté el tipus MIME del document, i que correspon a la tercera columna de la taula de més amunt (tipus de fitxers per documents). Això implica que l'extensió del nom del fitxer és indiferent del format real, doncs preval la definida per aquest fitxer. Així, l'extensió del fitxer s'utilitza només per facilitar la identificació del tipus del fitxer a l'usuari.
- **layout-cache**
Aquest fitxer inclou informació sobre el format emmagatzemat en cache per a documents de text (extensió .odt).
- **./META-INF/**
Aquesta carpeta conté l'arxiu **manifest.xml**, el qual descriu l'estructura (directoris i fitxers) que té l'arxiu empaquetat en format *OpenDocument*.
- **./Thumbnails/**
Aquesta carpeta conté l'arxiu **thumbnail.png**, el qual és una imatge de la primera pàgina del document o de la última pàgina modificada, que s'utilitza per mostrar com a vista prèvia del document.
- **./Pictures/**
Aquesta carpeta conté totes les imatges del document. El fitxer *content.xml* conté referències a elles mitjançant l'ús de l'etiqueta <draw:image>, similar a l'etiqueta de HTML. Tot seguit podem veure un exemple d'aquestes referències:

```
<draw:image
  xlink:href="Pictures/100000000000005E80000049F21F631AB.tif"
  xlink:type="simple" xlink:show="embed"
  xlink:actuate="onLoad" />
```

La informació de posicionament (amplada, posició, etc) es dona mitjançant una etiqueta <draw:frame> que conté a la seva vegada l'etiqueta <draw:image>.

La majoria de les imatges es guarden en el seu format original (GIF, JPEG, PNG), tot i que els mapes de bits es converteixen a PNG per qüestions de grandària.

- **./database/**
Aquesta carpeta conté tots els fitxers que estan relacionat i formen part de la definició de la base de dades (**backup, data, properties, script**)
- **./Configurations2/**
Aquesta carpeta conté varies subcarpetes, les quals emmagatzemen fitxers de configuració de diferents objectes, tals com la barra de menú, la barra d'eines o la barra d'estat.

El format *OpenDocument* va se dissenyat per a que reutilitzés els estàndards oberts existents de XML quan aquests estiguessin disponibles, i crear etiquetes noves només quan no existís un estàndard que oferís la funcionalitat necessària. Així, *OpenDocument* utilitza el nucli de Dublín de XML per a metadades, MathML per a fórmules matemàtiques, SVG per a gràfics vectorials, SMIL per la multimèdia, i altres *namespaces*, tal i com veurem més endavant.

3.3 Anàlisi format *OpenDocument*

3.3.1 Introducció

Després de fer un primer anàlisi dels fitxers OpenOffice i una ullada al format *OpenDocument* veient part de la seva història i evolució, els tipus de fitxers on s'utilitza, l'estructura que tenen aquests i esmentar aplicacions que suporten el format, passem a analitzar més a fons i en detall l'estructura que tenen els subdocuments que es troben empaquetats i comprimits dins un document OpenOffice.

El focus d'aquest anàlisi recaurà principalment sobre els fitxers amb extensió xml (**meta.xml**; **settings.xml**; **styles.xml**; **content.xml**), dels quals exposarem l'estructura i els elements que la formen, ja que és imprescindible conèixer-los per al desenvolupament del programari que volem construir (eina pel tractament automatitzat de fitxers OpenOffice). Així doncs, el propòsit d'aquest capítol és definir l'esquema XML utilitzat en els documents amb format *OpenDocument*.

Comencem primer anomenant els diferents espais de noms (*namespaces*) que s'utilitzen en els documents amb format *OpenDocument*: **office**; **meta**; **config**; **text**; **table**; **draw**; **presentation**; **dr3d**; **anim**; **chart**; **form**; **script**; **style**; **number**; **manifest**; **xlink**; **fo**; **svg**; **smil**; **dc**; **math**; **xforms** (Per a més informació sobre la localització de cadascun d'ells, consulteu la bibliografia).

3.3.2 Estructura

Un document amb format *OpenDocument* podrà contenir elements i atributs que no s'hagin especificat en el seu esquema, permetent així que l'especificació *OpenDocument* sigui usada en un major nombre d'implementacions. En canvi, les típiques aplicacions ofimàtiques només poden suportar un subconjunt d'elements i atributs que sí han hagut de ser definits a l'especificació, essent aquestes així més limitades que les aplicacions que suporten el format *OpenDocument*, les quals són més flexibles.

Root element

Un document OpenOffice segueix una mateixa estructura, independentment de si el document és de text, una fulla de càlcul o un dibuix.

L'element principal o primari que té un document en format *OpenDocument* és el **root element**, el qual conté el document sencer. Tots els tipus de documents OpenOffice utilitzen el mateix tipus de **root element**.

Tal i com ja s'ha comentat anteriorment, el format *OpenDocument* suporta dues maneres diferents de representar els seus documents:

- Amb un únic document XML.
- Amb diferents subdocuments empaquetats i comprimits que contenen parts del document. Cada un dels subdocuments té un **root element** diferent i emmagatzema un aspecte particular del document únic XML.

Hi ha quatre tipus de subdocuments, cada un dels quals amb un **root element** diferent. Addicionalment hi ha el **root element** propi de quan es representa en un únic document XML. Així tenim un total de cinc elements primaris, que resumim en la taula següent:

Root element	Contingut del subdocument	Nom del subdocument
<office:document>	Complet en un únic document XML	<i>no afecta</i>
<office:document-content>	Contingut del document i estils automàtics usats en el contingut	content.xml
<office:document-styles>	Estils usats en el contingut i estils automàtics usats als mateixos estils	styles.xml
<office:document-meta>	Informació metadata, tal com autor, data de creació, etc.	meta.xml
<office:document-settings>	Característiques específiques de l'aplicació, com la dimensió de la finestra o informació de la impressora	settings.xml

Els elements que conté el **root element** <office:document> són tots:

```
<define name="office-document">
  <element name="office:document">
    <ref name="office-document-attrs"/>
    <ref name="office-document-common-attrs"/>
    <ref name="office-meta"/>
    <ref name="office-settings"/>
    <ref name="office-scripts"/>
    <ref name="office-font-face-decls"/>
    <ref name="office-styles"/>
    <ref name="office-automatic-styles"/>
    <ref name="office-master-styles"/>
    <ref name="office-body"/>
  </element>
</define>
```

El **root element** <office:document-content> inclou el contingut del document i els estils automàtics usats en el contingut, i són:

```
<define name="office-document-content">
  <element name="office:document-content">
    <ref name="office-document-common-attrs"/>
    <ref name="office-scripts"/>
    <ref name="office-font-face-decls"/>
    <ref name="office-automatic-styles"/>
    <ref name="office-body"/>
  </element>
</define>
```

El **root element** <office:document-styles> conté tots els estils usats en el document, juntament amb els estils automàtics usats en els estils:

```
<define name="office-document-styles">
  <element name="office:document-styles">
    <ref name="office-document-common-attrs"/>
    <ref name="office-font-face-decls"/>
    <ref name="office-styles"/>
    <ref name="office-automatic-styles"/>
    <ref name="office-master-styles"/>
  </element>
</define>
```

El **root element** <office:document-meta> conté la informació meta del document:

```
<define name="office-document-meta">
  <element name="office:document-meta">
    <ref name="office-document-common-attrs"/>
    <ref name="office-meta"/>
  </element>
</define>
```

El **root element** <office:document-settings> conté les característiques específiques de l'aplicació que han de ser aplicades en processar el document:

```
<define name="office-document-settings">
  <element name="office:document-settings">
    <ref name="office-document-common-attrs"/>
    <ref name="office-settings"/>
  </element>
</define>
```

Anem ara a estudiar cada una de les referències a les que fan menció els diferents **root elements**.

<office:document-attrs>

Aquesta referència només conté l'atribut MIMEType. Només està disponible pel **root element** <office:document>, és a dir, quan només s'usa un únic document XML, ja que quan s'usen subdocuments empaquetats, aquest atribut es presenta en un fitxer per separat (*mimetype*).

<office:mimetype> → indica el tipus de document al qual pertany el document en qüestió (text, fulla de càlcul, etc).

<office:document-common-attrs>

Aquesta referència conté els atributs que tenen en comú tots els **root elements**, que són:

<office:version> → indica la versió de l'especificació a la qual forma part l'element al qual apareix.

<office:meta>

Aquest element conté informació general del document. Els seus elements són:

<meta:generator> → identifica l'aplicació o eina que ha creat o ha modificat per última vegada el document

<dc:title> → identifica el títol del document

<dc:description> → breu descripció del document

<dc:subject> → tema del document

<meta:keywords> → identifica una paraula clau. Aquest element pot aparèixer varies vegades

<meta:initial-creator> → nom de la persona que ha creat el document

<dc:creator> → nom de la darrera persona que ha modificat el document

<meta:printed-by> → nom de la darrera persona que ha imprès el document

<meta:creation-date> → data i hora en la que es va crear el document

<dc:date> → data i hora en la que el document s'ha modificat per última vegada

<meta:print-date> → data i hora en la que el document s'ha imprès per última vegada

<meta:template> → conté una URL a la plantilla usada per crear el document. La URL s'especifica com un *XLink*. Els atributs que hi té associats són:

xlink:href → localització de la plantilla del document

xlink:title → nom de la plantilla del document

meta:date → data i hora de la darrera modificació patida per la plantilla abans que aquesta s'usés com a plantilla del document en qüestió.

<meta:auto-reload> → especifica quan s'ha de recarregar o substituir per un altre document després de cert període de temps. S'especifica com un *XLink*. Els atributs associats són:

xlink:href → identifica la URL del document substituïdor

meta:delay → especifica la *reload delay*.

<meta:hyperlink-behaviour> → especifica el comportament per defecte dels hiperlinks continguts en el document. L'atribut associat és:

meta:target-frame-name → nom del *target frame* per defecte en que es visualitzarà el document al que referència l'hiperlink.

<dc:language> → especifica el llenguatge per defecte del document

<meta:editing-cycles> → número de vegades que s'ha editat el document

<meta:editing-duration> → total de temps editant el document

<meta:user-defined> → especifica qualsevol altra *user-defined* metadada addicional pel document que especifica l'usuari. Els seus atributs associats són:

meta:name → identifica el nom de l'element metadada

meta:type → identifica el tipus d'element metadada

office:value → valor del metadada. Aquest element pot variar en funció del tipus del metadada definit per l'usuari (*office:value*; *office:currency*; *office:date-value*; *office:time-value*; *office:boolean-value*; *office:string-value*).

<meta:document-statistic> → estadístiques del document. Els seus atributs depenen del tipus de document d'OpenOffice i de l'aplicació usada per crear el document. Pel tipus de document de text, els seus atributs són:

meta:page-count → número de pàgines que té el document

meta:table-count → número de taules que conté el document

meta:draw-count → número de dibuixos que hi ha al document

meta:image-count → número d'imatges que conté el document

meta:ole-object-count → número d'objectes OLE que té el document

meta:paragraph-count → número de paràgrafs contingut en el document

meta:word-count → número de paraules que conté el document

meta:character-count → número de caràcters que té el document

meta:row-count → número de files que conté el document

meta:frame-count → número de *frames* que hi ha al document

meta:sentence-count → número de sentències contingudes en el document

meta:syllable-count → número de síl·labes que té el document

meta:non-whitespace-character-count → número de caràcters no blancs continguts en el document

Tot seguit es mostra un exemple de l'element **<office:meta>** d'un document OpenOffice:

```
<office:meta>
  <dc:title>Title of the document</dc:title>
  <dc:description>Description/Comment for the document</dc:description>
  <meta:initial-creator>User Name</meta:initial-creator>
  <meta:creation-date>1999-10-18T12:34:56</meta:creation-date>
  <dc:creator>User Name</dc:creator>
  <dc:date>1999-10-19T15:16:17</dc:date>
  <meta:printed-by>User Name</meta:printed-by>
  <meta:print-date>1999-10-20T16:17:18</meta:print-date>
  <dc:subject>Description of the document</dc:subject>
  <meta:editing-duration>PT5H10M10S</meta:editing-duration>
  <meta:keyword>First keyword</meta:keyword>
  <meta:keyword>Second keyword</meta:keyword>
  <meta:keyword>Third keyword</meta:keyword>
  <meta:template xlink:type="simple"
    xlink:href="file:///c:/office52/share/template/german/finance/budget.vor"
    xlink:title="Template name"
    meta:date="1999-10-15T10:11:12" />
  <meta:auto-reload
    xlink:type="simple"
    xlink:href="file:///..."
    meta:delay="P60S" />
  <dc:language>de-DE</dc:language>
  <meta:user-defined meta:name="Field 1"
    meta:type="string">Value 1</meta:user-defined>
  <meta:user-defined meta:name="Field 2"
    meta:type="float">1.234</meta:user-defined>
</office:meta>
```

<office:settings>

Els *settings* (característiques d'entorn) d'una aplicació OpenOffice.org es poden dividir en varies categories, cada una representada per l'element **<config:config-item-set>**. Per exemple tenim:

- *Document settings* (p.ex. impressora per defecte)
- *View settings* (p.ex. nivell de zoom)

<config:config-item-set> → Aquest element conté un conjunt de *settings* i té només un atribut, anomenat **<config:name>**. Disposa dels següents elements:

<config:config-item> → És l'element que conté totes les característiques base d'un *setting*. Té els atributs següents:

<config:name> → Indica el nom de la característica

<config:type> → Indica el tipus de dada del *setting*, podent tenir els valors: *boolean, short, int, long, double, string, datetime, base64Binary*.

<config:config-item-set> → Especifica una altra cadena de característiques dins de la corresponent característica

<config:config-item-map-indexed> → Aquest element és un contenidor d'altres característiques en seqüència. L'ordre especifica l'índex dels elements. Té com a atribut **<config:name>** i disposa del següent element:

<config:config-item-map-entry> → Aquest element representa una entrada en seqüència de característiques, els quals podran ser accedits amb un índex que correspon a l'ordre com s'hagin entrat.

<config:config-item-map-named> → Aquest element és un contenidor d'altres característiques en seqüència, cada un dels quals s'identifica pel seu nom. També té com a atribut **<config:name>** i disposa del següent element:

<config:config-item-map-entry> → Aquest element representa una entrada en seqüència de característiques, els quals seran accedits pel nom.

<office:scripts>

En aquest element s'hi emmagatzemen els diferents scripts que pot usar el document OpenOffice, ja siguin macros, o altres. Els elements que pot contenir són:

<office:script> → Aquest element conté un dels varis scripts. En molts casos contindrà el codi font de l'script tot i que pot ser que contingui una versió compilada de l'script.. Com a atribut té **<script:language>** per indicar el llenguatge usat.

<office:event-listeners> → Aquest element conté l'assignació dels scripts amb els seus corresponents esdeveniments que els activen. Els elements que pot contenir són:

<script:event-listener> → Element per representar els esdeveniments que són lligats a un script. Els atributs que pot tenir associats són:

<script:event-name> → Especifica el nom de l'esdeveniment

<script:language> → Especifica el llenguatge utilitzat a l'script

<script:macro-name> → Especifica la localització del codi de l'script. Es pot especificar amb l'ajuda d'un *XLink*. Si es fa així, aquest element també contindrà els següents atributs: **<xlink:href>**, **<xlink:type>** i **<xlink:actuate>**.

<office:font-face-decls>

Aquest element aporta informació sobre les fonts usades per l'autor del document, les quals poden estar localitzades en altres sistemes. Les fonts que s'hi descriuen corresponen directament amb la descripció de fonts de **@font-face** de CSS2 (*Cascading Style Sheets, level2*) i amb la de **<font-face>** de SVG (*Scalable Vector Graphics*).

Degut a la gran quantitat d'atributs aplicables a les declaracions de fonts que ofereixen CSS2 i SVG, tan sols mencionarem els que m'han semblat més significatius. Per a més informació i detalls, consulteu la bibliografia.

<style:font-face> → Descriu una font usada en el document. Pot contenir els següents elements:

<svg:font-face-src> → Indica l'origen de la font. Pot recórrer a l'ajuda d'*XLink*. Per tant, els elements que pot contenir són:

<svg:font-face-uri> → Indica l'origen de la font per mitjà d'*XLink*. Els atributs que pot tenir associats són: **<xlink:href>**, **<xlink:type>** i **<xlink:actuate>**. Com a element que pot tenir hi ha:

<svg:font-face-format> → El qual s'utilitza per indicar el format amb el seu atribut **<svg:string>**

<svg:font-face-name> → Indica el nom de l'origen de la font, mitjançant el seu atribut **<name>**.

<svg:definition-src> → Indica l'origen de la definició de la font, també amb l'ajuda d'una referència amb *XLink*. Per tant, els atributs que pot tenir associats són: **<xlink:href>**, **<xlink:type>** i **<xlink:actuate>**

Els atributs que utilitza l'element **<style:font-face>** són:

<style:name> → Identifica el nom de la font

<style:font-adornments> → Especifica adornos de la font (**bold** o *italic*).

<style:font-family-generic> → Especifica el nom de la família genèrica

<style:font-pitch> → Indica si la font té un ample fix o variable
<style:font-charset> → Especifica el grup de caràcters d'una font

A més d'aquest cinc atributs, aquest element també utilitza una sèrie d'atributs obtinguts de les declaracions que ofereixen CSS2 i SVG. Aquests són:

<svg:font-family>; **<svg:font-style>**; **<svg:font-variant>**; **<svg:font-weight>**;
<svg:font-stretch>; **<svg:font-size>**; **<svg:unicode-range>**; **<svg:units-per-em>**;
<svg:panose-1>; **<svg:stemh>**; **<svg:stemv>**; **<svg:slope>**; **<svg:cap-height>**; **<svg:x-height>**;
<svg:accent-height>; **<svg:ascent>**; **<svg:descent>**; **<svg:widths>**;
<svg:bbox>; **<svg:ideographic>**; **<svg:alphabetic>**; **<svg:mathematical>**;
<svg:hanging>; **<svg:v-ideographic>**; **<svg:v-alphabetic>**; **<svg:v-mathematical>**;
<svg:v-hanging>; **<svg:underline-position>**; **<svg:underline-thickness>**;
<svg:strikethrough-position>; **<svg:strikethrough-thickness>**; **<svg:overline-position>**; **<svg:overline-thickness>**

<office:styles> ; **<office:automatic-styles>** **<office:master-styles>**

El format OpenDocument suporta els següents tipus d'estils:

- **Common styles:** Terme usat per referir-se als estils i per diferenciar els estils en casos on pot haver confusió amb els *automatic styles*. Són suportats per la majoria d'aplicacions ofimàtiques amb les seves interfícies d'usuari.
- **Automatic styles:** Estils que contenen unes propietats de format, a la vista de la interfície d'usuari del document, que són assignats a un objecte (p.ex. paràgraf). Aquest estil és generat automàticament quan s'utilitza l'objecte que el té associat.
- **Master styles:** És un *common style* que conté informació sobre el format i contingut addicional que es mostra amb el contingut del document quan se li aplica l'estil. Com a exemple clar tenim les *master pages*, usades en aplicacions gràfiques.

Els diferents estils es representen amb el següents elements:

<office:styles> → per identificar els *common styles*.

<office:automatic-styles> → per identificar els *automatic styles*.

<office:master-styles> → per identificar els *master styles*.

Per als estils *common styles* i *automatic styles* s'utilitza la mateixa representació XML, és a dir, que els elements i atributs que poden tenir són els mateixos pels dos.

En *OpenDocument* les propietats de format s'emmagatzemen solament en els estils. Això difereix de les típiques aplicacions ofimàtiques, on les propietats de format poden ser assignades a l'objecte directament o bé indirectament aplicant l'estil a l'objecte.

Hi ha dos motius principals per usar estils on emmagatzemar propietats del format:

1. Format i estructura del document queden separats del contingut.
2. Si dos o més objectes tenen les mateixes propietats de format i estils assignats, les propietats de format assignades als objectes directament poden ser representades per un únic *automatic style* aplicable a tots els objectes.

L'element bàsic que s'utilitza dins els elements dels diferents estils és:

<style:style> → Identifica un estil, ja sigui per un paràgraf, text o gràfic. Els atributs que tenen en comú quan l'element s'utilitza en aquests tres objectes diferents són:

<style:name> → Identifica el nom de l'estil. Combinat amb l'atribut **<style-family>** identifica de forma única un estil

<style:display-name> → Especifica la manera com el nom de l'estil apareixerà a la interfície d'usuari

<style:family> → Identifica la família de l'estil, com per exemple: *paragraph*, *text* o *frame*, entre d'altres

<style:parent-style-name> → Identifica el nom de l'estil pare

<style:next-style-name> → Especifica l'estil a usar per al següent paràgraf en cas de que un trancament del paràgraf s'introdueixi a la interfície d'usuari.

<style:list-style-name> → Especifica l'estil al que es podran associar estils del tipus paràgraf o d'altres que vulguin un estil de llista.

<style:master-page-name> → Especifica l'estil al que es podran associar estils del tipus paràgraf o taula que vulguin un estil *master page* (estil de pàgina)

<style:auto-update> → Determina si un estil que s'utilitzi en varis objectes es modificarà automàticament quan les propietats de format d'un dels objectes que tingui l'estil assignat canviï

<style:data-style-name> → Especifica l'estil al que es podran associar estils de cel·les de les taules

<style:class> → Indica la classe a la que pertany l'estil

<style:default-outline-level> → Per a estils de la família paràgraf aquest atribut especifica el nivell per defecte del perfil.

A més d'aquests atributs comuns per tots tres tipus d'estils, un estil també pot tenir assignats elements de format. D'aquests n'existeixen molts de diferents. A continuació passem a anomenar-los, tot i que no s'entrarà a detallar-los, doncs comportaria excedir el nombre de pàgines recomanades per la memòria.

<style:page-layout-properties> → Especifica propietats per l'esquema de la pàgina. Els atributs que pot contenir són: *page size, page number format, paper tray, print orientation, margins, border, border line width, padding, shadow, background, columns, register-truth, print, print page order, first page number, scale, table centering, maximum footnote height, footnote separator.*

<style:header-footer-properties> → Especifica propietats per la capçalera i el peu de pàgina. Els atributs que pot contenir són: *fixed and minimum heights, left and right margins, bottom and top margins, borders, shadows, backgrounds, dynamic-spacing.*

<style:text-properties> → Especifica propietats pel text. Els atributs que pot contenir són: *font variant, text transformations, color, window font color, text outline, line through type, line through style, line through width, line through color, line through text, line through text style, text position, font name, font family, font family generic, font style, font pitch, font character set, font size, relative font size, script type, letter spacing, language, country, font relief, text shadow, underlining type, underlining style, underling width, underline color, font weight, text underline word mode, text line through word mode, letter kerning, text blinking, text background color, text combine, text combine start and end characters, text emphasis, text scale, text rotation angle, text rotation scale, hyphenation, hyphenation remain char count, hyphenation push char count, hidden or conditional text.*

<style:paragraph-properties> → Especifica propietats pel paràgraf. Els atributs que pot contenir són: *fixed line height, minimum line height, line distance, font-independent line spacing, text align, text align of last line, justify single word, keep together, widows, orphans, tab stops, tab stop distance, hyphenation keep, maximum hyphens, drop caps, register true, left and right margins, text indent, automatic text indent, top and bottom margins, margins, break before and break after, paragraph background color, paragraph background image, border, border line width, padding, shadow, keep with next, line numbering, line number start value, text autospace, punctuation wrap, line break, vertical alignment, writing mode, automatic writing mode, snap to layout, page number, background transparency.*

<style:section-properties> → Especifica propietats per la secció de text. Els seus atributs són: section background, margins, columns, column specification, column separator, protect, don't balance text columns, writing mode, notes configuration.

<style:ruby-properties> → Especifica propietats per la secció de vermell. Els seus atributs són: ruby position, ruby alignment.

<style:list-level-properties> → Especifica propietats per llistes. Els seus atributs són: label alignment, start indent, minimum label width, minimum label distance, font name, image size, vertical alignment.

<style:table-properties> → Especifica propietats per taules. Els seus atributs són: table width, table alignment, table left and right margin, table top and bottom margin, table margins, page number, break before and break after, table background and background image, table shadow, keep with next, may break between rows, border model, writing mode, display.

<style:table-column-properties> → Especifica propietats per columnes de taules. Els seus atributs són: column width, optimal table column width, break before and break after.

<style:table-row-properties> → Especifica propietats per files de taules. Els seus atributs són: row height, optimal table row height, row background, break before and break after, keep together.

<style:table-cell-properties> → Especifica propietats per cel·les de taules. Els seus atributs són: vertical alignment, text align source, direction, vertical glyph orientation, cell shadow, cell background, cell border, diagonal lines, border line width, padding, wrap option, rotation angle, rotation align, cell protect, print content, decimal places, repeat content, shrink to fit.

<style:graphic-properties> → Especifica propietats per objectes de dibuix. Els seus atributs estan formats per:

- **stroke properties** (propietats del moviment): stroke style, dash, multiple dashes, width, color, start marker, end marker, start marker width, end marker width, start marker center, end marker center, opacity, line join
- **fill properties** (propietats d'emplenar): fill style, color, secondary fill color, gradient, gradient step count, hatch, solid hatch, fill image, fill image rendering style, fill image size, fill image tile reference point, fill image tile translation, none and linear opacity, gradient opacity, fill rule, symbol color
- **text animation properties** (propietats d'animació de text): animation, animation direction, animation start inside, animation stop inside, animation repeat, animation delay, animation steps
- **text and text alignment properties** (propietats de text i d'alineació de text): auto grow width and height, fit to size, fit to contour, text area vertical align, text area horizontal align, word wrap, list styles
- **color properties** (propietats de color): color mode, color inversion, adjust luminance, adjust contrast, adjust gamma, adjust red, adjust green, adjust blue, adjust opacity
- **shadow properties** (propietats d'ombra): shadow, offset, color, opacity
- **connector properties** (propietats de connexió): start line spacing, end line spacing
- **measure properties** (propietats de mesura): line distance, guide overhang, guide distance, start guide, end guide, placing, parallel, text alignment, unit, show unit, decimal places

- **caption properties** (propietats de subtítol): type, angle type, angle, gap, escape direction, escape, line length, fit line length
- **3D geometry properties** (propietats de geometria 3D): horizontal segments, vertical segments, edge rounding, edge rounding mode, back scale, depth, backface culling, end angle, close front, close back
- **3D lighting properties** (propietats d'il·luminació 3D): mode, normal kind, normal direction
- **3D texture properties** (propietats de text 3D): generation mode, kind, filter, mode
- **3D material properties** (propietats de material 3D): colors, shininess
- **3D shadow properties** (propietats d'ombra 3D): shadow
- **frame formatting properties** (propietats de format del marc): frame widths, frame heights, maximum width and height, left and right margins, top and bottom margins, margins, print content, protect, horizontal position, horizontal relation, vertical position, vertical relation, frame anchor, border, border line width, padding, shadow, background, columns, editable, wrapping, dynamic wrap threshold, paragraph-only wrapping, contour wrapping, contour wrapping mode, run through, flow with text, overflow behavior, mirroring, clipping, wrap influence on position
- **floating frame formatting properties** (propietats de format de marc flotant): display scrollbar, display border, margins, visible area, draw aspect
- **chart formatting properties** (propietats de format de mapa): scale text
- **chart subtype properties** (propietats de subtipus de mapa): three-dimensional chart, chart depth, chart symbol, chart symbol size, bar chart properties, stock chart properties, line chart properties, pie chart properties, lines, solid charts bars, stacked chart bars
- **chart axes properties** (propietats d'eix de mapa): linked data formats, visibility, scaling, tickmarks, labels
- **common chart properties** (propietats comunes de mapa): stacked text, rotation angle, data labels
- **statistical properties** (propietats d'estadístiques): mean value, error category
- **plot area properties** (propietats d'àrea de parcel·la): series source
- **regression curve properties** (propietats de corba regressiva): regression type
- **presentation page attributes** (atributs de pàgines de presentació): transition type, transition style, transition speed, transition type or family, transition subtype, transition direction, fade color, page duration, page visibility, sound, background size, background objects visible, background visible, display header, display footer, display page number, display date and time

Després d'anomenar les propietats dels elements de format que hi pot haver dins un element **style**, seguim els elements que podrà contenir:

<style:map> → Especifica l'estil amb el qual es *mapeja* en el cas que es compleixin certes condicions. Els seus atributs són:

<style:condition> → Indica la condició que s'ha de complir perquè es produeixi el *mapeig*.

<style:apply-style-name> → Indica l'estil que s'aplicarà en cas que la condició s'avalui a cert.

<style:base-cell-address> → Indica la cel·la base per adreces relatives en fórmules.

<style:default-style> → Especifica les propietats de format per defecte per un cert estil de família.

<style:page-layout> → Especifica les propietats físiques d'una pàgina. Aquest element conté dos atributs:

<style:name> → On s'hi especifica el nom de l'estil amb propietats físiques

<style:page-usage> → Indica el tipus de pàgines que la pàgina màster hauria de generar.

A més d'aquests dos atributs, aquest element pot contenir els següents elements:

<style:header-style> → Especifica les propietats de capçalera de la pàgina

<style:footer-style> → Especifica les propietats del peu de pàgina

<style:master-page> → Aquest element especifica una plantilla per a les pàgines en un document. Conté una referència a un element **page-layout**. Tot document contindrà com a mínim un element **master-page**. Els seus atributs són:

<style:name> → Indica el nom de la pàgina màster

<style:default-name> → Indica el nom de la pàgina màster que apareixerà a la interfície d'usuari

<style:page-layout-name> → Indica la **page layout** que conté les dimensions, marges i orientació de la pàgina màster

<draw:style-name> → Defineix un emplenament de fons opcional. Usat per a aplicacions gràfies

<style:next-style-name> → Indica la pàgina màster que és usada per la pàgina següent si la pàgina actual està completa.

A més d'aquests atributs, l'element **<style:master-page>** també pot contenir els següents elements:

<style:header> → Especifica el contingut de la capçalera

<style:footer> → Especifica el contingut del peu de pàgina

Els elements que poden contenir els dos elements són:

<text:h>, **<text:p>**, **<text:list>**, **<table:table>**, **<text:section>**, **<text:table-of-content>**, **<text:illustration-index>**, **<text:table-index>**, **<text:object-index>**, **<text:user-index>**, **<text:alphabetical-index>**, **<text:bibliography>**, **<text:index-title>**, **<text:change-marks>**

A més de:

<style:region-left>, **<style:region-center>**, **<style:region-right>** els quals s'usen per especificar les tres regions d'una capçalera o peu de pàgina si són mostrats amb alineació a l'esquerra, dreta o centrada.

I de l'atribut **<style:display>** usat per indicar si es mostra o no es mostra la capçalera o peu de pàgina.

<presentation:notes> → Element usat en aplicacions de presentacions per a especificar una pàgina de notes de la presentació. Els seus atributs són:

<style:page-layout> → Indica la **page layout** que conté les dimensions, marges i orientació de la pàgina de notes

<draw:style-name> → Per assignar un atribut de format addicional a la pàgina de notes

<presentation:use-header-name> → Indica el nom de la declaració del camp per la capçalera que s'usa per tots els camps de capçalera que són mostrats a la pàgina de notes

<presentation:use-footer-name> → Indica el nom de la declaració del camp pel peu de pàgina que s'usa per tots els camps de peus de pàgina que són mostrats a la pàgina de notes

<presentation:use-date-time-name> → Indica el nom de la declaració del camp per la data i hora que s'usa per tots els camps de data i hora que són mostrats a la pàgina de notes

<table:table-template> → Aquest element és un conjunt de propietats de format, com marges, color de fons i propietats de text que poden ser aplicades a una taula quan aquesta es crea. Els atributs que té són:

<table:name> → Indica el nom de la plantilla per la taula

<table:first-row-start-column> → Indica si la cel·la situada a la primera fila i primera columna ha d'obtenir el seu estil de la fila o de la columna on es troba

<table:first-row-end-column> → Indica si la cel·la situada a la primera fila i darrera columna ha d'obtenir el seu estil de la fila o de la columna on es troba

<table:last-row-start-column> → Indica si la cel·la situada a la darrera fila i primera columna ha d'obtenir el seu estil de la fila o de la columna on es troba

<table:last-row-end-column> → Indica si la cel·la situada a la darrera fila i darrera columna ha d'obtenir el seu estil de la fila o de la columna on es troba

I els seus elements són:

<table:first-row> → Indica l'estil de la primera fila de la taula

<table:last-row> → Indica l'estil de la darrera fila de la taula

<table:first-column> → Indica l'estil de la primera columna de la taula

<table:last-column> → Indica l'estil de la darrera columna de la taula

<table:body> → Indica l'estil de les de més cel·les de la taula

<table:even-rows> → Indica l'estil de les files parelles de la taula

<table:odd-rows> → Indica l'estil de les files imparelles de la taula

<table:even-columns> → Indica l'estil de les columnes parelles de la taula

<table:odd-columns> → Indica l'estil de les columnes imparelles de la taula

Anem a veure tot seguit una sèrie d'estils que descriuen com es mostraran una sèrie de dades, com per exemple un número o una data:

<number:number-style> → Especifica l'estil que segueixen els números decimals. Pot contenir els següents elements:

<number:text> → Especifica la forma com es mostrarà el text

<number:number> → Especifica la forma com es mostrarà el número

<number:scientific-number> → Especifica la forma com es mostrarà un número científic

<number:fraction> → Especifica com es mostrarà un número que sigui fracció

<number:currency-style> → Especifica l'estil que segueixen els valors per diners. Els elements que pot contenir són:

<number:number> → Només pot aparèixer una vegada i especifica el format com es mostrarà el valor dels diners

<number:currency-symbol> → Només pot aparèixer una vegada. Aquest element especifica si un símbol de moneda és mostrat en l'estil actual o no

<number:percentage-style> → Especifica l'estil que segueixen els valors per percentatges.

Els elements que pot contenir són:

<number:text> → Especifica el text i número que es mostrarà. Ha de contenir el símbol %

<number:number> → Especifica com es mostrarà el format del percentatge

<number:data-style> → Especifica l'estil que segueixen els valors per dates. Aquest element pot contenir una instància d'algun d'aquests elements: **<number:day>**, **<number:month>**, **<number:year>**, **<number:era>**, **<number:day-of-week>**, **<number:week-of-year>**, **<number:quarter>**, **<number:hours>**, **<number:minutes>**, **<number:seconds>**, **<number:am-pm>**

A part, també podrà contenir elements **<number:text>** però no de forma consecutiva.

<number:time-style> → Especifica l'estil dels valors per temps. Aquest element pot contenir una instància d'algun d'aquests elements: **<number:hours>**, **<number:minutes>**, **<number:seconds>**, **<number:am-pm>**. A part, també podrà contenir elements **<number:text>**, però no de forma consecutiva.

<number:boolean-style> → Especifica l'estil dels valors per booleans. Els elements que pot contenir són: **<number:text>**, **<number:boolean>**

<number:text-style> → Especifica l'estil per mostrar el text. Els elements que pot contenir són:
<number:text-content> → Representa el contingut del text variable a mostrar
<number:text> → Conte qualsevol altra text fix addicional a mostrar

Hi ha un parell d'elements que són comuns a tots aquests elements d'estil de dades:

<style:text-properties> → Especifica les propietats del format del text que s'aplicarà a qualsevol text mostrat en l'estil de dada corresponent

<style:map> → Especifica l'estil de dades alternatiu amb el que es mapejarà si es donen certes condicions.

La majoria d'estils de dades no només tenen en comú aquests dos elements, sinó que també tenen atributs en comú:

<style:name> → Indica el nom de l'estil de dades

<style:display-name> → Indica el nom de l'estil de dades tal i com apareixerà a la interfície d'usuari

<number:language> → Indica el codi de llenguatge de l'estil de dades

<number:country> → Indica el codi del país de l'estil de dades

<number:title> → Indica el títol de l'estil de dades

<style:volatile> → Indica si l'estil de dades en qüestió pot ser ignorat degut a que no s'utilitzi habitualment.

<number:automatic-order> → Pot ser utilitzat per ordenar automàticament les dades per equiparar l'ordre per defecte del llenguatge i país de l'estil de dades. S'utilitza pels elements **<number:currency-style>**, **<number:data-style>**

<number:format-source> → Indica la font del format a mostrar. S'utilitza pels elements **<number:data-style>**, **<number:time-style>**

<number:transliteration-format> → Indica el número de caràcters que s'usen

<number:transliteration-language> → Indica el llenguatge al que pertany el sistema de numeració nadiu

<number:transliteration-country> → Indica el país al qual pertany el sistema de numeració nadiu

<number:transliteration-style> → Indica l'estil al qual pertany el sistema de numeració nadiu

A part dels dos elements i atributs que tenen en comú els elements d'estil de dades, aquests també tenen uns altres atributs d'estil numèric que són aplicables a més d'un element d'estil numèric. Aquests atributs són: **<number:decimal-places>**, **<number:min-integer-digits>**, **<number:grouping>**, **<number:calendar>**

Acabat l'anàlisi d'estils de dades, seguim amb l'estil de text:

<style:style> → Aquest element especifica els estils de text, tenint el valor a l'atribut família igual a *text*. També pot definir estils de paràgraf si l'atribut família té el valor *paragraph*. Una altra opció que hi ha és definir estils de secció, però en aquest cas l'atribut família haurà de contenir el valor *section*. Finalment, també es podrà definir estils de *ruby* quan el valor per l'atribut família sigui *ruby*.

Estils de text destacats:

<text:linenumbering-configuration> → Un document pot contenir un o cap element per configurar la línia de numeració. Els atributs que pot tenir associats són: **<text:number-lines>**, **<style:num-prefix>**, **<style:num-suffix>**, **<style:num-format>**, **<style:num-letter-sync>**, **<text:style-name>**, **<text:increment>**, **<text:position>**, **<text:offset>**, **<text:count-empty-lines>**, **<text:count-in-text-boxes>**, **<text:restart-on-page>**, **<text:linenumbering-separator>**

<text:notes-configuration> → Un document en format *OpenDocument* pot contenir com a molt un element de configuració de notes per cada classe de notes que s'usi. Els atributs que pot tenir associats són: **<text:note-class>**, **<text:citation-style-name>**, **<text:citation-body-style-name>**, **<text:default-style-name>**, **<text:master-page-name>**, **<text:start-value>**, **<style:num-prefix>**, **<style:num-suffix>**, **<style:num-format>**, **<style:num-letter-sync>**, **<text:start-numbering-at>**, **<text:footnotes-position>**, **<text:note-continuation-notice-forward>**

<text:bibliography-configuration> → Especifica la configuració de la bibliografia. Els atributs que pot contenir són: **<text:prefix>**, **<text:suffix>**, **<text:numbered-entry>**, **<text:sort-by-position>**, **<text:sort-key>**

<text:list-style> → Aquest element especifica les propietats de format per a llistes. Conté un conjunt d'elements d'estil per a cada nivell de la llista, els quals s'anomenen **estils de nivell de llista**. D'aquests estils de nivell de llista n'existeixen tres de diferents, depenent si el nivell de la llista en qüestió tingui una llista d'etiquetes formades per números, bales o imatges. Els atributs que té associats són:

<style:name> → Indica el nom de l'estil de llista

<style:display-name> → Indica el nom de l'estil de llista tal i com apareixerà a la interfície d'usuari

<text:consecutive-numbering> → Indica si l'estil de llista utilitza una numeració consecutiva per tots els seus nivells, o cada cop es reinicia la numeració

A més d'aquests tres elements, hi ha un element comú per a tots els estils de nivell de llista. Aquest és: **<text:level>** i indica el nivell de l'estil de llista.

<text:list-level-style-number> → Element que especifica l'estil d'una llista numerada. Els seus atributs associats són:

<text:style-name> → Indica el nom de l'estil de caràcters a utilitzar per donar format a la llista numèrica

<style:num-prefix> → Indica com es mostrarà abans del número

<style:num-suffix> → Indica com es mostrarà després del número

<style:num-format> → Indica el format del número de la mateixa manera que un atribut de format

<style:num-letter-sync> → Indica si la sincronització de lletres enlloc de números en una llista numèrica estarà habilitada

<text:display-levels> → Indica el número de nivells que hi haurà disponibles pel nivell actual

<text:start-value> → Indica el primer número d'un element de la llista pel nivell actual

Hi podrà haver propietats de format addicionals contingudes en els elements **<style:list-level-properties>** i **<style:text-properties>**.

<text:list-level-style-bullet> → Element que especifica l'estil d'una llista on els seus elements són precedits per bales(punts). Els seus atributs associats són:

<text:style-name> → Indica el nom de l'estil de caràcters a utilitzar per donar format a la llista de bales

<text:bullet-char> → Indica el caràcter UNICODE a utilitzar com a bala en una llista de bales

<style:num-format-prefix> → Indica el format a utilitzar pels caràcters d'abans del caràcter bala

<style:num-format-suffix> → Indica el format a utilitzar pels caràcters de després del caràcter bala

<text:bullet-relative-size> → Indica el valor en percentatge per la dimensió relativa de la bala en front de la dimensió dels paràgrafs en una llista de bales

Hi podrà haver propietats de format addicionals contingudes en els elements **<style:list-level-properties>** i **<style:text-properties>**.

<text:list-level-style-image> → Element que especifica l'estil d'una llista on els seus elements són precedits per imatges. Els seus atributs associats poden ser:

<xlink:href> → Per especificar la localització de la imatge a utilitzar

Aquest atribut no serà utilitzat en cas que s'utilitzi el següent element:

<office:binary-data> → Aquest element contindrà les dades de la imatge codificades en BASE64

<text:outline-style> → Aquest element especifica un estil de llista que s'aplica a totes les capçaleres d'un document de text on l'estil paràgraf de la capçalera no defineixi un estil per a les seves pròpies llistes. Pot contenir el següent element:

<text:outline-level-style> → Element que especifica l'estil per cada un dels nivells d'esquema. Els atributs que pot tenir associats aquest element són:

<text:level> → Indica el nivell de l'estil per l'esquema

<text:style-name> → Indica el nom de l'estil pels caràcters a utilitzar pel format del número de la capçalera

<style:num-prefix> → Indica l'estil a utilitzar abans dels números

<style:num-suffix> → Indica l'estil a utilitzar després dels números

<style:num-format> → Indica el format del número de la mateixa manera que un atribut de format

<style:num-letter-sync> → Indica si la sincronització de lletres enlloc de números estarà habilitada

<text:display-levels> → Indica el número de nivells que hi haurà disponibles pel nivell actual

<text:start-value> → Indica el primer número d'una capçalera pel nivell actual

Hi podrà haver propietats de format addicionals contingudes en els elements **<style:list-level-properties>** i **<style:text-properties>**.

Una vegada acabat l'anàlisi dels estils disponibles per a qualsevol tipus de llista, seguirem amb els estils que podrem aplicar a les taules. L'element que ho defineix és el **<style:style>** quan té el valor de **table** per a l'atribut família, i els elements que pot contenir són:

<style:style> → Element que quan conté el valor **table-column** per a l'atribut família especifica les propietats de les columnes de la taula. Pot suportar qualsevol propietat de format per columnes de taules de les vistes anteriorment

<style:style> → Element que quan conté el valor **table-row** per a l'atribut família especifica les propietats de les files de la taula. Pot suportar qualsevol propietat de format per files de taules de les vistes anteriorment

<style:style> → Element que quan conté el valor **table-cell** per a l'atribut família especifica les propietats de les cel·les de la taula. Pot suportar qualsevol propietat de format per cel·les de taules de les vistes anteriorment

Passem a veure els diferents estils possibles per a gràfics. N'hi ha tres i tots ells utilitzen com a element **<style:style>** però amb diferents valors a l'atribut família. Si té el valor **graphic** està indicant que es tracta d'un estil gràfic, si té el valor **presentation** està indicant que es tracta d'un estil gràfic per a presentacions, i si té el valor **drawing-page** estarà indicant que es tracta d'un estil gràfic per a pàgines de dibuixos.

Elements destacats per a estils gràfics, només localitzables dins l'element **<office:styles>**:

<draw:gradient> → Especifica un gradient per poder omplir un objecte de dibuix. Els seus atributs són: **<draw:name>**, **<draw:display-name>**, **<draw:style>**, **<draw:cx>**, **<draw:cy>**, **<draw:start-color>**, **<draw:end-color>**, **<draw:start-intensity>**, **<draw:end-intensity>**, **<draw:angle>**, **<draw:border>**

Adicionalment, els gradients també poden ser especificats mitjançant els elements **<svg:linearGradient>** i **<svg:radialGradient>**

<draw:hatch> → Aquest element especifica una finestreta per emplenar els objectes de dibuix. Els atributs que pot tenir associats són: **<draw:name>**, **<draw:display-name>**, **<draw:style>**, **<draw:color>**, **<draw:distance>**, **<draw:rotation>**

<draw:fill-image> → Element que especifica un enllaç a una font de mapa de bits, com per exemple, un arxiu amb extensió *png*. Com a atributs pot tenir: **<draw:name>**, **<draw:display-name>**, **<svg:width>**, **<svg:height>**

<draw:opacity> → Especifica la opacitat del gradient per a un objecte gràfic. Els atributs que té són: **<draw:name>**, **<draw:display-name>**, **<draw:style>**, **<draw:cx>**, **<draw:cy>**, **<draw:angle>**, **<draw:border>**, **<draw:start>**, **<draw:end>**

<draw:marker> → Aquest element representa un marcador que s'utilitza per dibuixar polígons al principi i final de punts traçats. Entre els seus atributs hi tenim: **<draw:name>**, **<draw:display-name>**

<draw:stroke-dash> → Representa un estil per fer un dibuix ràpid que es pot utilitzar per donar cops de formes. Els atributs que pot tenir associats són: **<draw:name>**, **<draw:display-name>**, **<draw:style>**, **<draw:dots1>**, **<draw:dots1-length>**, **<draw:dots2>**, **<draw:dots2-length>**, **<draw:distance>**

I amb aquest element acabem amb els elements disponibles per a estils gràfics destacats.

<style:presentation-page-layout> → Aquest és un element contenidor de places (*placeholders*) que defineixen un conjunt d'objectes de presentació buits, com per exemple, un títol. Els seus atributs són: **<style:name>**, **<style:display-name>**. Els elements que conté són:

<presentation:placeholder> → Especifica un *placeholder* per objectes de presentació. Els atributs que pot tenir són: **<presentation:object>**, **<svg:x>**, **<svg:y>**, **<svg:width>**, **<svg:height>**

L'estil existent per a definir un dibuix d'un mapa (*chart*) és el **<style:style>** utilitzant el valor **chart** dins l'atribut família.

Amb aquest estil acabem l'anàlisi del principal element per definir els estils, **<style:style>**.

<office:body>

Aquest element indica quin tipus de contingut conté el document. Actualment, els tipus de documents suportats pel format *OpenDocument* són:

- **Documents de text**
- **Documents de dibuix**
- **Documents de presentació**
- **Documents de full de càlcul**
- **Documents de gràfics**
- **Documents d'imatges**

Tots els tipus de documents comparteixen els mateixos elements que poden contenir, tot i que diferents tipus de documents tenen restriccions sobre els elements que contenen i les seves combinacions.

En el cas que s'està abordant en aquest projecte, només analitzarem l'estructura que segueixen els documents de text, doncs analitzar-los tots suposaria superar el límit de pàgines establertes per a la memòria.

Aquest tipus utilitza com a element principal:

<office:text> → El contingut d'un document de text està format principalment per un nombre de paràgrafs, taules, índexs, marcs de text (*frames*), seccions de text i elements gràfics. Addicionalment, un document de text pot contenir formularis, canvis de *tracking* (seguiment) i declaracions de variables. Cada un d'aquests es defineix al prelude del document i s'hi pot fer referència des del contingut del document. L'únic atribut que té opcional és **<text:global>**. Té quatre referències que són:

<office:text-content-prelude> → Conté la informació del document de text referent a formularis, canvis de *tracking* i declaració de variables. També conté elements per permetre implementar funcionalitats típiques de fulls de càlcul, però no els comentarem, ja que aquest anàlisi correspon a documents de tipus text.

Conté els següents elements:

<office:forms> → Aquest element especifica un formulari. Per a més informació, consulteu la bibliografia

<text:tracked-changes> → Element que especifica un canvi de *tracking*. Es troba dins un element de llista. Com a atribut utilitza:

<text:track-changes> → Indica si els agents d'usuari guardaran els canvis pel document

I els elements que pot contenir són:

<text:changed-region> → Especifica una entrada als canvis de *tracking*. Conté una llista amb tots els canvis a aplicar. Té un atribut **<text:id>** usat per identificar l'element. Els elements que pot contenir aquest són:

<text:insertion> → Especifica la informació requerida per identificar qualsevol inserció de contingut

<text:deletion> → Especifica la informació que serà eliminada mentre el canvi de *tracking* estigui habilitat

<text:format-change> → Especifica qualsevol canvi en el format dels atributs. Podrà contenir l'element **<office:change-info>** el qual conté meta informació sobre qui fa el canvi i quan

<text:change-start> → Especifica el principi d'una regió amb contingut on s'hi posarà text o s'hi modificarà el format

<text:change-end> → Especifica el final d'una regió amb contingut on s'hi posarà text o s'hi modificarà el format

<text:change> → Especifica una posició en un regió buida on el text ha estat eliminat

Aquests tres últims elements contenen l'atribut **<text:id>** per a especificar un identificador de la regió a la qual pertanyen

<text:decls> → Element que conté declaracions de text, les qual poden ser sobre variables, camps d'usuari, seqüències, connexions DDL o arxius d'auto marca. Els elements que pot contenir són:

<text:variable-decls> → Aquest element conté tants elements del tipus **<text:variable-decl>** com declaracions de variables tingui, i cada un d'ells contindrà els atributs:

<text:name> → Indica el nom de la variable

<office:value-type> → Indica el tipus de valor que té

<text:user-field-decls> → Aquest element conté tants elements del tipus **<text:user-field-decl>** com declaracions de camps d'usuari tingui, i cada un d'ells contindrà els atributs:

<text:name> → Indica el nom de la variable

<text:formula> → Indica la fórmula per computar el valor del camp de la variable d'usuari

<office:value-type> → Indica el tipus de valor que té

<text:sequence-decls> → Aquest element conté tants elements del tipus **<text:sequence-decl>** com declaracions de variables de seqüència tingui, i cada un d'ells contindrà els atributs:

<text:name> → Indica el nom de la variable

<text:display-outline-level> → Indica un nivell d'esquema que determina els capítols on fa referència

<text:separation-character> → Indica el caràcter amb el qual es realitza la separació del número del capítol

<text:dde-connection-decls> → Element per declarar els camps utilitzats per realitzar la connexió DDE

<text:alphabetical-index-auto-mark-file> → Element amb el qual es declaren els arxius de marca per índexs. Usa l'atribut **<xlink:href>** per realitzar els enllaços

<table:decls> → Element amb les declaracions de taula. S'utilitza per a arxius de tipus fulla de càlcul.

<office:text-content-main> → Conté la informació del document de text referent a paràgrafs (incloent capçaleres), seccions de text (i índexs), taules i formes gràfiques. Com a alternativa, un document de text pot contenir una seqüència d'una sola pàgina.

<text:h> → Element que especifica capçaleres. Juntament amb **<text:p>** són coneguts com els elements paràgraf. Els seus atributs són:

<text:outline-level> → Determina el nivell de la capçalera, començant per 1

<text:restart-numbering> → Determina si s'ha de restablir la numeració per a les capçaleres

<text:start-value> → S'utilitza per reiniciar la numeració de les capçaleres dins el nivell actual

<text:is-list-header> → Per a especificar que la capçalera en qüestió no sigui numerada

A més d'aquests quatre atributs, l'element **<text:h>** també podrà contenir els atributs de l'element **<text:p>**.

Com a elements, hi pot haver:

<text:number> → Si a la capçalera s'hi aplica numeració, el text del format dels números s'inclou en aquest element

<text:p> → Element que especifica un paràgraf. Els seus atributs són:

<text:style-name> → Indica l'estil per al paràgraf

<text:class-names> → Indica una llista amb espai en blanc separat per a estils de text

<text:cond-style-name> → Indica un estil condicional que s'utilitzarà com a estil si es compleixen les condicions que té

Abans de continuar amb el contingut principal del document, anem a veure el contingut que es pot trobar dins els elements paràgrafs.

Els fills dels elements paràgraf formen el contingut de text de qualsevol document. Tot el contingut d'un element paràgraf i fills és contingut de text, amb unes quantes excepcions que anem a veure:

Caràcters d'espai en blanc → Es processen per mitjà dels elements **<text:p>**, **<text:h>**, **<text:span>**, **<text:a>**, **<text:ref-point>**, **<text:ref-point-start>**, **<text:ref-point-end>**, **<text:bookmark>**, **<text:bookmark-start>**, **<text:bookmark-end>**. Amb l'atribut **<text:c>** es pot especificar un número de caràcters d'espai. També tenim l'element **<text:tab>** que representa el caràcter tabulador i l'element **<text:line-break>** per marcar el trencament d'una línia en una capçalera o paràgraf.

També hi ha caràcters com els guionets, que només poden ser representats mitjançant caràcters UNICODE.

Text atribuïts → Amb l'element **<text:span>** es representen porcions de text que són atribuïts usant un determinat estil o classe de text, amb lo qual, tindrà com atributs **<text:style-name>** i **<text:class-names>**.

Enllaços → Es representen amb l'element **<text:a>**. Conté l'element **<office:event-listeners>** dins el qual s'hi especifiquen els esdeveniments que té assignats l'enllaç i que ja s'ha analitzat anteriorment.

Els atributs que es poden assignar a l'element **<text:a>** són: **<office:name>**, **<xlink:href>**, **<office:target-frame-name>**, **<text:style-name>** i **<text:visited-style-name>**.

Uns altres elements que ens podem trobar dins els elements paràgraf són els de marcatge i els de referència. Els primers marcaran una posició de text o un rang de text. Els elements que utilitza són:

<text:bookmark> → Marca una posició de text

<text:bookmark-start> → Marca la posició inicial d'un rang de text

<text:bookmark-end> → Marca la posició final d'un rang de text

Per representar referències hi ha dues maneres, en funció de l'element:

<text:reference-mark> → Marca un punt de referència

<text:reference-mark-start> → Marca el punt inicial d'un rang de referència

<text:reference-mark-end> → Marca el punt final d'un rang de referència

Als elements paràgrafs s'hi poden incorporar notes, **<text:note>** tant de peu com de fi, les quals són mostrades fora del contingut de text principal i adjuntades a una certa posició del text. Mitjançant l'element **<text:notes-configuration>**, que incorpora **<office:styles>**, es podrà configurar el format de la nota. Els elements de **<text:note>** són:

<text:note-class> → Especifica el tipus de nota, que podrà ser *footnotes* o *endnotes*.

<text:id> → Especifica un identificador per la nota

<text:note-citation> → Especifica l'element de la citació de la nota amb el seu format. Pot tenir l'atribut **<text:label>** per indicar com s'etiqueta la citació de la nota

<text:note-body> → Especifica el contingut actual de la nota de peu

També podem tenir text adicional mostrat a sobre o a sota de la base de text, mitjançant l'element **<text:ruby>**. Pot contenir dos elements:

<text:ruby-base> → Especifica el text que serà anotat, contenint qualsevol element paràgraf. Disposa de l'atribut **<text:style-name>** per indicar l'estil que segueix el text

<text:ruby-text> → Especifica l'anotació del text, contenint text pla. També s'hi especifica l'atribut **<text:style-name>**

Parlant de textos addicionals, cal mencionar la possibilitat de realitzar anotacions per a que apareguin amb els elements paràgraf. L'element usat per això és **<office:annotation>**. Els atributs que hi pot tenir associats són ben variats, des de la posició i la grandària fins a l'estil. I dels elements continguts en destaquem **<text:p>** i **<text:list>**. Es descriu en més detall dins d'aquest mateix capítol.

Tenim les marques d'índex que s'utilitzen per construir un índex a partir d'aquestes. Hi ha els següents tipus de marques d'índex:

Taules de contingut → Es defineixen per mitjà dels elements:

<text:toc-mark-start> → Marca el principi d'una taula de contingut d'índex. Té els atributs **<text:id>** i **<text:outline-level>**

<text:toc-mark-end> → Marca el final d'una taula de contingut d'índex. Té l'atribut **<text:id>**

<text:toc-mark> → Marca una taula de contingut d'índex sense que quedi limitada. Com a atribut té **<text:string-value>**

Definides per l'usuari → Es defineixen per mitjà dels elements:

<text:user-index-mark-start> → Marca el principi d'un índex definit per l'usuari. Conté els atributs **<text:id>**, **<text:outline-level>** i **<text:index-name>**

<text:user-index-mark-end> → Marca el final d'un índex definit per l'usuari. Conté els atributs **<text:id>** i **<text:outline-level>**

<text:user-index-mark> → Marca un índex definit per l'usuari sense que quedi limitat. Com a atribut té **<text:string-value>**

Alfabètiques → Es defineixen per mitjà dels elements:

<text:alpha-index-mark-start> → Marca el principi d'un índex alfabètic. Els atributs que hi té associats són: **<text:id>**, **<text:key1>**, **<text:key2>** i **<text:main-entry>**

<text:alpha-index-mark-end> → Marca el final d'un índex alfabètic. Els atributs que hi té associats són: **<text:id>**

<text:alpha-index-mark> → Marca un índex alfabètic sense que quedi limitat. Com a atribut té **<text:string-value>**

Bibliogràfiques → Es defineixen per mitjà de l'element **<text:bibliography-mark>**.

Els elements paràgrafs poden contenir també marques de canvis de *tracking*, les quals ja s'han comentat anteriorment.

Més elements continguts dins els elements paràgrafs són els d'imatges i de dibuixos, mitjançant els elements **<shape>** i **<draw:a>**, però aquests no seran analitzats, ja que es tracten d'elements utilitzats en documents gràfics i de dibuixos.

Dins el contingut dels elements paràgrafs ens podem trobar **camp**s de text, és a dir, elements de text variables. S'utilitzen bàsicament per la numeració de pàgines, creació de dates i rangs de números. Els dividirem en les següents categories:

Camps de document → Són els que mostren informació sobre el document actual o sobre una part específica del document actual. Dins ells hi tenim els següents elements:

<text:date> → Mostra la data actual. Disposa dels atributs:

<text:date-value> → Indica el valor particular de la data

<text:date-adjust> → Indica la forma com s'ajusta el valor de la data a un cert període

<text:fixed> → Indica si el valor de l'element és fix, és a dir, que no varia cada cop que s'edita el document
<style:data-style-name> → Indica el nom de l'estil que dóna format al valor de l'element

<text:time> → Mostra l'hora actual. Disposa dels atributs:
<text:time-value> → Indica l'hora en que s'ha editat per última vegada el document
<text:time-adjust> → Indica la forma com s'ajusta el valor de l'hora a un cert període
<text:fixed> → Indica si el valor de l'element és fix
<style:data-style-name> → Indica el nom de l'estil que dóna format al valor de l'element

<text:page-number> → Mostra el número de la pàgina actual. Disposa dels atributs:
<text:page-adjust> → Indica el valor en com s'ajusta el número de la pàgina
<text:select-page> → Indica el número de la pàgina prèvia o següent enlloc del número de la pàgina actual. Els valors disponibles són: *previous, current, next*
<text:fixed> → Indica si el valor de l'element és fix
<style:data-style-name> → Indica el nom de l'estil que dóna format al valor de l'element

<text:page-continuation> → Mostra un recordatori al final de la pàgina a més a més del número de pàgina. D'atributs en té:
<text:select-page> → Indica si es comprova l'existència de la pàgina prèvia o següent per continuar imprimint
<text:string-value> → Indica el text de continuació

Elements diversos per representar camps del remitent. Tots ells disposen de l'atribut **<text:fixed>** per indicar si són elements fixos o no: **<text:sender-firstname>**, **<text:sender-lastname>**, **<text:sender-initials>**, **<text:sender-title>**, **<text:sender-position>**, **<text:sender-email>**, **<text:sender-phone-private>**, **<text:sender-fax>**, **<text:sender-company>**, **<text:sender-phone-work>**, **<text:sender-street>**, **<text:sender-city>**, **<text:sender-postal-code>**, **<text:sender-country>**, **<text:sender-state-or-province>**

Elements per mostrar informació de l'autor del document. Tots ells disposen de l'atribut **<text:field>** per indicar si són elements fixos o no: **<text:author-name>**, **<text:author-initials>**

<text:chapter> → Mostra el nom i/o número del capítol. Com atributs té:
<text:display> → Indica la informació que el camp hauria de mostrar
<text:outline-level> → Indica el nivell d'esquema a usar

<text:file-name> → Mostra el nom de l'arxiu que s'està editant actualment. Els atributs que té associats són:
<text:display> → Indica quanta informació de l'arxiu es mostrarà (nom amb extensió, només ruta, només nom, tot)
<text:fixed> → Indica si el valor canvia o no quan s'edita el document

<text:template-name> → Mostra informació sobre la plantilla que utilitza el document. Com a atribut té

<text:display> → Indica quanta informació sobre el document de la plantilla es mostrarà

Camps variables → Són els camps processats o mostrats usant camps variables. En format *OpenDocument* les variables han de ser declarades al principi del document. Hi ha variables de tres tipus:

De simples → Tenen diferents valors durant el document. Els diferents elements que utilitzen són:

<text:variable-decl> → Element que s'utilitza per declarar variables. Com a atributs associats té:

<text:name> i **<office:value-type>**

<text:variable-set> → Element que s'utilitza per assignar valor a les variables. Com a atributs associats té:

<text:name>, **<text:formula>**, **<office:value-type>**,
<text:display>, **<style:data-style-name>**

<text:variable-get> → Element per llegir i mostrar el valor de les variables. Els atributs que hi té associats són:

<text:name>, **<text:display>**, **<style:data-style-name>**

<text:variable-input> → Element alternatiu per assignar valor a les variables. Com a atributs associats té:

<text:name>, **<text:description>**, **<office:value-type>**,
<text:display>, **<style:data-style-name>**

D'usuari → Tenen el mateix valor durant tot el document. Els diferents elements que utilitzen són:

<text:user-field-decl> → Element que s'utilitza per declarar variables i assignar-li el valor que tindrà sempre. Com a atributs associats té:

<text:name>, **<text:formula>**, **<office:value-type>**

<text:user-field-get> → Elements per mostrar el contingut de les variables d'usuari. Els seus atributs són:

<text:name>, **<text:display>**, **<style:data-style-name>**

<text:user-field-input> → Element alternatiu per assignar a les variables d'usuari. Com a atributs té:

<text:name>, **<text:description>**, **<style:data-style-name>**

De seqüència → Tenen valors incrementals i s'utilitzen per numerar elements. Els diferents elements que utilitzen són:

<text:sequence-decl> → Element usat per declarar les variables de seqüència. Els atributs que hi té associats són:

<text:name>, **<text:display-outline-level>**, **<text:separation-character>**

<text:sequence> → Element amb el que s'usen els camps de seqüència després que aquests hagin estat declarats. Els atributs que té associats són:

<text:name>, **<text:formula>**, **<style:num-format>**, **<style:num-letter-sync>**, **<text:ref-name>**

A més d'aquestes tres grans categories de camps variables, tenim els camps expressió que són avaluats i acaben mostrant el valor del resultat. S'especifiquen amb l'element:

<text:expression> → I com a atributs té: **<text:formula>**,
<office:value-type>, **<text:display>**, **<style:data-style-name>**

Camps de metadades → Són camps que mostren meta informació sobre el document. Els seus noms corresponen als elements descrits dins l'element **<office:meta>**. Els elements que s'hi inclouen són:

<text:initial-creator>, **<text:creation-date>**, **<text:creation-time>**,
<text:description>, **<meta:user-defined>**, **<text:print-time>**, **<text:print-date>**,
<text:printed-by>, **<text:title>**, **<text:subject>**, **<text:keywords>**,
<text:editing-cycles>, **<text:editing-duration>**, **<text:modification-time>**,
<text:modification-date>, **<text:creator>**, **<text:page-count>**,
<text:paragraph-count>, **<text:word-count>**, **<text:character-count>**,
<text:table-count>, **<text:image-count>**, **<text:object-count>**

Camps de base de dades → Són camps que fan referència a bases de dades. Els elements que en formen part són:

<form:connection-resource> → Especifica la base de dades mitjançant un enllaç XLink

<text:database-display> → Mostra les dades d'una base de dades. Els elements que hi pot tenir associats són:

<text:database-name> → Indica el nom de la base de dades font

<text:table-name> → Indica el nom d'una taula

<text:table-type> → Indica si la taula es refereix a una taula SQL, a una consulta *OpenDocument* o al resultat d'una comanda SQL.

<text:database-column-name> → Indica la columna de la qual es mostrarà la informació

<style:data-style-name> → Indica el format amb el qual es mostren les dades

<text:database-next> → Canvia la fila de la selecció actual a la següent en el cas que es compleixi la condició específica. Els atributs que hi té associats són:

<text:database-name>, **<text:table-name>**, **<text:table-type>**

<text:condition> → Indica la condició a complir per passar a la següent fila

<text:database-row-select> → Selecciona una fila de les que hi ha en la selecció actual en el que cas que es compleixi la condició específica. Els atributs que hi té associats són:

<text:database-name>, **<text:table-name>**, **<text:table-type>**,
<text:condition>

<text:row-number> → Indica el número de fila a seleccionar quan la condició és certa.

<text:database-row-number> → Especifica el número de la fila actual per a una taula determinada. Els seus atributs associats són:

<text:database-name>, **<text:table-name>**, **<text:table-type>**

<style:num-format> → Indica el format numèric

<style:num-letter-sync> → Indica la sincronització de lletres amb els números

<text:value> → Indica el número de la fila actual

<text:database-name> → Mostra la base de dades i el nom de la taula de la taula utilitzada més recentment. Com a atributs té:

<text:database-name>, **<text:table-name>**, **<text:table-type>**

Altres camps → Tot seguit es nombren una sèrie de camps addicionals que són utilitzats en el format *OpenDocument*.

<text:page-variable-set> → Element que permet una numeració alternativa per a les pàgines. Els seus atributs associats són:

<text:active> → Indica si s'utilitza una numeració addicional per a les pàgines o no

<text:page-adjust> → Indica l'ajustament de la pàgina

A més d'aquests dos atributs, també té associat l'element:

<text:page-variable-get> → El qual especifica el valor de la variable pàgina. Els atributs que hi pot tenir associats són **<style:num-format>** i **<style:num-letter-sync>**

<text:placeholder> → Aquests tipus de camps s'utilitzen per indicar localitzacions dins el document on l'usuari ha d'emplenar alguna mena d'informació. Com a atributs associats té:

<text:placeholder-type> → Indica el tipus per representar el camp, que pot ser: *text, table, text-box, image, object*.

<text:description> → Indica una descripció més detallada del camp

<text:conditional-text> → Element per representar un camp de text però amb l'ajuda d'una condició. Els seus atributs són:

<text:condition> → Indica la condició que s'avalua

<text:string-value-if-true> → Indica la cadena de text que es mostrarà en el cas que la condició sigui certa

<text:string-value-if-false> → Indica la cadena de text que es mostrarà en el cas que la condició sigui falsa

<text:current-value> → Indica el resultat de l'avaluació de la condició donada

<text:hidden-text> → Aquest tipus de camp mostra un text de manera fixa, excepte quan una condició és certa, doncs llavors no mostra res. Els seus atributs associats són:

<text:condition> → Indica la condició que s'avalua

<text:string-value> → Indica el text que es mostra si la condició és falsa

<text:is-hidden> → Indica si el camp està actualment visible o no

Camps de referència → Aquests camps estan basats en el tipus d'elements als quals es refereixen: *notes, bookmarks, references, sequences*. Estem parlant dels elements: **<text:reference-ref>**, **<text:bookmark-ref>**, **<text:note-ref>**, **<text:sequence-ref>**.

<text:script> → Un camp *script* emmagatzema scripts o seccions per scripts. Com a atributs té:

<xlink:href> → Indica la localització de l'arxiu que conté el codi font de l'script.

<script:language> → Indica el llenguatge en què el codi font de l'script és escrit.

<text:execute-macro> → Un camp *macro* conté el nom d'una macro que és executada quan el camp és activat. L'atribut associat que té és **<text:name>** el qual indica la macro a invocar

<text:hidden-paragraph> → Aquest tipus de camp permet a un paràgraf de text formatat ser amagat o mostrat en funció de si es compleix una condició. Els seus atributs associats són:

<text:condition> → Indica la condició que s'avalua

<text:is-hidden> → Indica si el paràgraf està actualment visible o no

<text:dde-connection> → Aquest tipus de camp permet que informació DDE pugui ser mostrada. El seu atribut és:

<text:connection-name> → Indica el nom de la connexió DDE a la que es refereix el camp

<text:measure> → Conté la mesura per a objectes de dibuix. El seu atribut **<draw:kind>** indica quina part de la mesura es mostra

<text:table-formula> → Aquest tipus de camp emmagatzema una fórmula per se usada en taules. Pot tenir els següents atributs:

<text:formula> → Indica l'expressió actual utilitzada per computar el valor del camp

<text:display> → S'utilitza per mostrar el valor del camp o la fórmula usada per computar el valor

<style:data-style-name> → Indica l'estil utilitzat per donar format al camp

Amb aquests tipus de camps acabem de veure tots els elements que estan disponibles dins el contingut dels elements paràgrafs. Seguim doncs amb els elements continguts dins **<office:text-content-main>**.

<text:page-sequence> → Aquest element especifica una seqüència de pàgines màster que són instanciades en el mateix ordre com fan referència en aquest element. Està compost per elements com:

<text:page> → Indica una pàgina de la seqüència. Té l'atribut:

<text:master-page-name> → Especifica la pàgina màster que és instanciada

<text:list> → Element que representa una llista. Els seus atributs són:

<text:style-name> → Indica el nom de l'estil per la llista

<text:continue-numbering> → Indica si la numeració per la llista actual és heretada d'una llista anterior o no. Els elements que utilitza aquest element són:

<text:list-item> → Especifica el contingut textual de la llista. Pot contenir paràgrafs i llistes, però no taules ni capçaleres, amb lo qual pot contenir els elements **<text:h>** i **<text:p>**. Com a atribut:

<text:start-value> → Indica el número amb el que comença la llista

Si en un element de la llista s'hi aplica numeració, el format del text numèric pot ser inclòs dins l'element **<text:number>**

<text:list-header> → És un tipus especial d'element d'una llista. Conté un o més paràgrafs que són mostrats abans de la llista

<text:numbered-paragraph> → Element que permet la numeració d'un paràgraf individual, com si fos part d'una llista en un nivell específic. Com a atribut té **<text:level>** a més dels atributs dels elements **<text:list>** i **<text:list-item>**

<text:section> → Element que especifica una secció de text, és a dir, una regió de nivells de paràgraf que conté text. Pot contenir qualsevol nombre de paràgrafs. Com a atributs té:

<text:style-name> → Indica l'estil que utilitza la secció

<text:name> → Identifica únicament aquesta secció

<text:protected> → Indica si la secció queda protegida per a que cap usuari la pugui editar

<text:protection-key> → Indica la clau que protegeix la secció

<text:display> → Indica si la secció queda oculta o no

<text:condition> → S'utilitza juntament amb l'atribut anterior i especifica la condició per a que la secció quedi oculta o no

A part d'aquests atributs, l'element podrà contenir els elements:

<text:section-source> → Especifica la font amb la qual s'enllaça la secció, quan la informació de la secció és externa. Pot tenir els següents atributs:

<xlink:href> → Indica amb un *XLink* la referència

<text:section-name> Indica el nom de la secció a la qual fa referència

<text:filter-name> → Indica el tipus de filtre utilitzat per importar l'enllaç
<office:dde-source> → Especifica un enllaç amb informació externa mitjançant DDE.

<text:table-of-content> → Element que aporta una taula de contingut que serveix com a índex i guia del contingut que hi ha al document. Els atributs que suporta són els mateixos que l'element **<text:section>**, és a dir:

<text:style-name>, **<text:name>**, **<text:protected>**, **<text:protection-key>** i **<text:display>**

A part d'aquests atributs, els elements que conté són:

<text:table-of-content-source> → Especifica com la taula de continguts és generada. Els atributs que pot tenir associats són:

<text:outline-level> → Indica el nivell de l'esquema que s'utilitza quan es genera la taula de continguts

<text:use-outline-level> → Indica si les capçaleres s'utilitzen per generar les entrades de l'índex

<text:use-index-marks> → Indica si les marques d'índex s'utilitzen per generar les entrades de l'índex

<text:use-index-source-styles> → Indica si les entrades de l'índex són generades per paràgraf usant estils de paràgraf

<text:index-scope> → Indica si la taula de continguts és generada pel document o només pel capítol actual

<text:relative-tab-stop-position> → Indica si la posició de les parades de tabulador és relativa al marge esquerre o si es determina per l'estil del paràgraf

A més d'aquests atributs, també pot contenir els següents elements:

<text:index-title-template> → Determina l'estil i contingut del títol de l'índex.

<text:table-of-content-entry-template> → Determina el format d'una entrada d'índex per a un nivell particular d'esquema. Té els següents atributs associats:

<text:outline-level> → Especifica el nivell al que s'aplica

<text:style-name> → Especifica l'estil de paràgraf a usar

També suporta els següents tipus d'elements de text:

<text:index-entry-chapter> → Especifica el número de capítol de l'entrada de l'índex. Com a atributs associats té:

<text:style-name> i **<text:display>**

<text:index-entry-page-number> → Element que mostra el número de pàgina on l'entrada de l'índex es troba ubicada. L'element que té associat és **<text:style-name>**

<text:index-entry-text> → Element que mostra el text de l'entrada de l'índex. Té l'atribut associat **<text:style-name>**

<text:index-entry-span> → Element que representa una cadena fixa a l'entrada d'índex. Conté **<text:style-name>**

<text:index-entry-tab-stop> → Element que representa una parada de tabulació per a l'entrada d'índex. Els seus atributs:

<text:style-name> → Indica l'estil usat

<style:leader-char> → Indica el caràcter líder

<style:type> → Indica el tipus de parada de tabulació

<style:position> → Indica la posició del tabulador

<text:index-entry-link-start> → Element que marca el principi d'una entrada d'índex d'enllaç. Com a atribut associat té **<text:style-name>**

<text:index-entry-link-end> → Element que marca la fi d'una entrada d'índex d'enllaç. També té l'atribut **<text:style-name>**

<text:index-source-styles> → Aquest element conté totes les fonts d'estils per a un determinat nivell d'esquema. Conté l'atribut:

<text:outline-level> → Per especificar el nivell al que s'aplica
A més de poder contenir el següent element varies vegades:
<text:index-source-style> → Usat per indicar l'estil o classe que inclou
l'índex

<text:index-body> → Element per especificar el cos de l'índex. Els elements que
té associats són:

<text:content> → Especifica el contingut que té l'índex, incloent-hi elements
paràgraf
<text:index-title> → Especifica el títol de l'índex

<text:illustration-index> → Element que llista totes les imatges i gràfics del document
o capítol actual. Com a atribut té:

<text:style-name> → Indica l'estil de secció que utilitza l'índex

Els elements que pot contenir són:

<text:illustration-index-source> → Especifica com es genera l'índex
d'il·lustracions. Els atributs que té associats són:

<text:use-caption> → Indica si la il·lustració té títol

<text:caption-sequence-name> → Indica la seqüència a la que estan
associades els títols

<text:caption-sequence-format> → Si les entrades de l'índex s'obtenen de
títols d'imatges, aquest atribut indica el format que tenen les entrades

<text:index-scope> → Indica si l'índex s'aplica al document sencer o només
al capítol actual

<text:relative-tab-stop-position> → Indica si la posició de les parades dels
tabuladors són o no relatives al marge esquerre

A més d'aquests atributs, aquest element pot contenir els elements:

<text:index-title-template> → Determina l'estil i contingut del títol de l'índex.

<text:illustration-index-entry-template> → Determina el format d'una
entrada d'índex per a un nivell particular d'esquema. Aquest element conté
atributs i elements ja vistos en índexs anteriors

<text:index-body> → Element per especifica el cos de l'índex. Aquest element ja
s'ha analitzat amb d'altres índexs.

<text:table-index> → Element que llista totes les taules en el document o capítol
actual. S'utilitza igual que l'element **<text:illustration-index>**, és a dir, que conté els
elements:

<text:table-index-source> → Mateix comportament i propietats que l'element
<text:illustration-index-source> vist anteriorment, amb la diferència que enlloc
d'utilitzar l'element **<text:illustration-index-entry-template>** utilitza l'element
<text:table-index-entry-template>

<text:index-body> → Element amb les mateixes característiques que l'element de
mateix nom usat en índexs d'il·lustracions

<text:object-index> → Element que llista tots els objectes en el document o capítol
actual. Utilitza els mateixos atributs que l'element **<text:illustration-index>**. Com a
elements associats té:

<text:object-index-source> → Determina quin tipus d'objecte s'inclou a l'índex
d'objectes. Els atributs que pot tenir associats són:

<text:use-*-objects> → On s'inclouen els atributs **<text:use-spreadsheet-objects>**, **<text:use-draw-objects>**, **<text:use-chart-objects>**, **<text:use-math-objects>**, **<text:use-other-objects>**

<text:index-scope> → Indica si l'índex s'aplica al document sencer o només al capítol actual

<text:relative-tab-stop-position> → Indica si la posició de les parades dels tabuladors són o no relatives al marge esquerre

A més d'aquests atributs, els elements que pot contenir són:

<text:index-title-template>, **<text:object-index-entry-template>** amb similars característiques que els elements d'índex d'il·lustracions

<text:index-body> → Element amb les mateixes característiques que l'element de mateix nom usat en índexs d'il·lustracions

<text:user-index> → Element que combina les capacitats dels índexs vistos fins ara. Les seves entrades poden ser de diferents fonts. Els seus atributs són els mateixos que els usats per l'element **<text:illustration-index>**, i els elements que conté són:

<text:user-index-source> → Element que conté varis atributs per determinar l'origen de les entrades de l'índex. Aquests atributs són:

<text:use-*> → On s'inclouen els atributs **<text:use-index-marks>**, **<text:use-graphics>**, **<text:use-tables>**, **<text:use-floating-frames>**, **<text:use-objects>**

<text:copy-outline-levels> → Indica si les entrades es situen al nivell d'esquema de l'element font al qual es refereix, o si pel contrari es situen al nivell d'esquema més alt

<text:index-scope> → Ja comentat anteriorment

<text:index-name> → Indica quina marca d'índex s'aplica a l'índex actual

<text:relative-tab-stop-position> → Ja comentat anteriorment

A més d'aquests atributs, els elements que pot contenir són:

<text:index-title-template> → Ja comentat anteriorment

<text:user-index-entry-template> → Determina el format d'una entrada d'índex per a un nivell particular d'esquema. D'atributs té:

<text:outline-level> i **<text:style-name>**

<text:index-source-styles> → Ja comentat anteriorment

<text:index-body> → Element amb les mateixes característiques que l'element de mateix nom usat en índexs anteriors

<text:alphabetical-index> → Element per configurar un índex on totes les seves entrades provenen de marques d'índex. Els atributs que té associats són els mateixos que els usats per l'element **<text:illustration-index>** i els elements que conté són:

<text:alphabetical-index-source> → Especifica com es genera l'índex alfabètic. Els atributs que té associats són:

<text:ignore-case> → Indica si les lletres majúscules de les paraules són ignorades

<text:main-entry-style-name> → Indica l'estil de caràcters usat per les entrades principals

<text:alphabetical-separators> → Indica si les entrades que comencen amb la mateixa lletra són agrupades i separades de les entrades que comencen amb la lletra següent

<text:combine-entries> → Indica si quan hi ha múltiples entrades d'índex per la mateixa paraula o frase aquestes són combinades en una sola entrada

<text:combine-entries-with-dash> → Indica si el número de les pàgines on fan referència les entrades combinades es separen amb un guió

<text:combine-entries-with-pp> → Indica si el número de les pàgines on fan referència les entrades combinades comença amb una etiqueta escollida del llenguatge corresponent

<text:use-keys-as-entries> → Indica si les claus de les marques d'índex també són utilitzades com a entrades per l'índex

<text:capitalize-entries> → Indica si les entrades per l'índex són transformades a majúscules

<text:comma-separated> → Indica si les entrades múltiples són separades per coma o situades en diferents línies per separat

<fo:language> → Indica el llenguatge usat per l'índex

<fo:country> → Indica el país

<text:sort-algorithm> → Indica l'algoritme usat per ordenar

<text:index-scope> → Comentat anteriorment

<text:relative-tab-stop-position> → Comentat anteriorment

A més d'aquests atributs, també pot contenir els elements:

<text:index-title-template> → Ja comentat anteriorment

<text:alphabetical-index-entry-template> → Determina els nivells d'esquema de l'índex i l'estil de paràgraf. Per això utilitza els següents atributs:

<text:outline-level> → Indica el nivell, i podrà tenir els valors 1,2 o 3, doncs aquest tipus d'índex no suporta més nivells

<text:style-name> → Indica l'estil usat per la plantilla

A més d'aquests dos atributs, també contenir els següents elements, els quals ja s'han analitzat amb un tipus d'índex anterior:

<text:index-entry-chapter>, **<text:index-entry-page-number>**,

<text:index-entry-text>, **<text:index-entry-span>**, **<text:index-entry-tab-stop>**

<text:index-body> → Element amb les mateixes característiques que l'element de mateix nom usat en índexs anteriors

<text:bibliography> → Element per configurar un índex on totes les seves entrades provenen de marques d'índex bibliogràfiques. Els atributs que té associats són els mateixos que els usats per l'element **<text:illustration-index>** i els elements que conté són:

<text:bibliography-source> → Especifica com es genera la bibliografia. Els elements que pot contenir són:

<text:index-title-template> → Comentat anteriorment

<text:bibliography-entry-template> → Determina el format d'una entrada d'índex bibliogràfica. Els seus atributs associats són:

<text:bibliography-type> → Indica quin tipus d'entrada bibliogràfica aplica la plantilla. Per veure els valors possibles, consulteu la bibliografia

<text:style-name> → Indica l'estil usat per la plantilla

A més d'aquests dos atributs, els elements que pot contenir són:

<text:index-entry-span> → Comentat anteriorment

<text:index-entry-tab-stop> → Comentat anteriorment

<text:index-entry-bibliography> → Element que introdueix dades bibliogràfiques a les entrades d'índex de la plantilla. Els atributs que té associats són:

<text:style-name> → Indica l'estil de l'entrada d'índex

<text:bibliography-data-field> → Indica quina part del camp bibliogràfic serà mostrada. Per saber els valors que pot contenir, consulteu la bibliografia

<text:index-body> → Element amb les mateixes característiques que l'element de mateix nom usat en índexs anteriors

<table:table> → Element principal per definir una taula, tant per a document de text com per a fulles de càlcul, tot i que aquí només tindrem en compte els elements que s'utilitzen en document de text. Els atributs que té associats aquest element són:

<table:name> → Indica el nom de la taula

<table:style-name> → Indica les propietats de format de la taula

<table:protected> → Indica si la taula està protegida per editar o no

<table:print> → Indica si la taula és impresa

<table:print-ranges> → Indica si un rang de la taula és impresa o no

<table:is-subtable> → Indica si la taula és una taula continguda dins una cel·la d'una taula mare

A més d'aquests atributs, els elements que pot contenir són;

<table:table-row> → Especifica una fila en una taula. Com a atributs té:

<table:number-rows-repeated> → Indica el número de files a que un element fila s'aplica

<table:style-name> → Indica l'estil que s'aplica a la fila

<table:default-cell-style-name> → Indica l'estil per defecte de les cel·les de la fila

<table:visibility> → Indica si la fila està visible, filtrada o col·lapsada (per aquests dos últims, no estarà visible)

A part d'aquests atributs, els elements que pot contenir són:

<table:table-cell> → Especifica el contingut de les cel·les. Els atributs que té associats són:

<table:number-columns-repeated> → Indica el número de columnes successives en què es repeteix una cel·la

<table:number-columns-spanned> → Indica el número de columnes que té l'espai d'una cel·la

<table:number-rows-spanned> → Indica el número de files que té l'espai d'una cel·la

<table:style-name> → Indica l'estil de format de la cel·la, com el color de fons, el format de números o l'alineació vertical

<table:content-validation-name> → Indica si una cel·la conté una cadena de validació

<table:value-type> → Indica el tipus de valor que pot aparèixer a la cel·la

<office:value> → Indica el valor numèric actual d'una cel·la

<office:currency> → Indica el valor de la moneda actual

<office:date-value> → Indica el valor de data actual

<office:time-value> → Indica el valor de temps actual

<office:boolean-value> → Indica el valor booleà actual

<office:string-value> → Indica el valor de la cadena actual

<table:protected> → Indica si la cel·la de la fila està protegida

A més d'aquests atributs, l'element **<table:table-cell>** també disposa dels següents elements:

<table:cell-range-source> → Especifica la informació sobre el rang original quan hi fa un enllaç. Aquest element està contingut a la primera cel·la del rang. Els seus atributs són:

<table:name> → Indica el nom del rang origen

<table:last-column-spanned> → Indica la darrera dimensió coneguda per una columna del rang

<table:last-row-spanned> → Indica la darrera dimensió coneguda per una fila del rang

<xlink:href> → Indica la URL usada per l'enllaç *XLink*
<xlink:type> → Indica el tipus de *XLink* usat per l'enllaç
<xlink:actuate> → Indica l'actuació que farà l'enllaç
<table:filter-name> → Indica el tipus d'arxiu del document que conté el rang original
<table:filter-options> → Indica característiques opcionals sobre el tipus d'arxiu

<office:annotation> → Especifica anotacions per a la cel·la. Els atributs que hi pot tenir associats són:

<office:display> → Indica si l'anotació és o no visible
A més d'una gran quantitat d'atributs per especificar com es mostrarà l'anotació. Per saber més sobre ells, consulteu la bibliografia.

D'elements pot contenir els següents:

<dc:creator> → Especifica l'autor de l'anotació
<dc:date> → Especifica la data i el temps de creació de l'anotació
<meta:date-string> → Especifica una cadena per a la data en cas de que no s'hagi pogut transformar

<table:detective> → Aquest element conté informació sobre les relacions entre cel·les i si han d'estar ressaltades a la interfície d'usuari. També s'usa per contenir informació sobre les cel·les que són ressaltades actualment a la interfície d'usuari per relacions entre cel·les o per errors en condicions. Els elements que pot contenir són:

<table:operation> → Especifica les relacions entre cel·les que poden ser visibles o no a la interfície d'usuari. Els atributs que hi té associats són:

<table:name> → Indica el nom de l'operació detectiu
<table:index> → Indica l'ordre amb el qual les operacions detectiu s'apliquen

<table:highlighted-range> → Especifica un rang de cel·les ressaltades a la interfície d'usuari degut a operacions detectiu o bé perquè contenen algun error. Els atributs que pot tenir associats són:

<table:cell-range-address> → Indica l'adreça del rang actualment ressaltat
<table:direction> → Indica la direcció de la relació entre aquesta cel·la i el rang ressaltat
<table:contains-error> → Indica si el rang de cel·les conté algun error
<table:market-invalid> → Indica si la cel·la actual està marcada de manera invàlida

<text:content> → Especifica el contingut que té la cel·la

<table:table-column> → Especifica una columna en una taula. Com a atributs associats té:

<table:number-columns-repeated>, **<table:style-name>**,
<table:visibility>, **<table:default-cell-style-name>**

Els quals ja s'han comentat anteriorment

<table:table-header-columns> → Especifica aquelles columnes que són repetides automàticament en cada pàgina. L'únic element que pot contenir varies vegades és **<table:table-column>**.

<table:table-column-group> → Especifica una agrupació de columnes adjacents. Els atributs que té associats són:

<table:display> → Indica si l'agrupació és visible o no

Els elements que pot contenir són:

<table:table-column-group> → Especifica una agrupació de columnes dins l'agrupació de columnes corresponents

<table:table-column> → Especifica una columna de la taula

<table:table-header-columns> → Especifica les columnes que fan de capçalera automàticament en cada pàgina

<table:table-header-rows> → Especifica aquelles files que són repetides automàticament a cada pàgina. L'únic element que pot contenir varies vegades és **<table:table-row>**

<table:table-row-group> → Especifica una agrupació de files adjacents. Els atributs que té associats són:

<table:display> → Indica si l'agrupació és visible o no

Els elements que pot contenir són:

<table:table-row-group> → És a dir, una altra agrupació de files dins l'agrupació de files corresponent

<table:table-row> → Especifica una fila de la taula

<table:table-header-rows> → Especifica les files que fan de capçalera automàticament en cada pàgina

<table:table-source> → Especifica la font de la taula quan aquesta és un enllaç a una altra taula. Els atributs que pot tenir associats són:

<table:mode> → Indica quines dades han de ser copiades de la taula origen a la taula destí. Els valors que pot tenir són: *copy-all* o *copy-results-only*

<table:table-name> → Indica el nom de la taula origen en el document

<xlink:type> → Quan la taula font s'especifica amb un enllaç *XLink*, aquest atribut indica quin tipus utilitza

<xlink:actuate> → Quan la taula font s'especifica amb un enllaç *XLink*, aquest atribut indica quina actuació farà

<xlink:href> → Quan la taula font s'especifica amb un enllaç *XLink*, aquest atribut indica la URL

<table:filter-name> → Indica el tipus d'arxiu del document que conté la taula original

<table:filter-options> → Indica característiques opcionals sobre el tipus d'arxiu

<table:refresh-delay> → Indica el temps de retard entre les accions de refresc de la taula d'enllaç

<office:dde-source> → Especifica una connexió de dades DDE de taules. Els atributs que pot suportar són:

<office:dde-application> → Indica el nom de l'aplicació a usar per a la connexió DDE

<office:dde-topic> → Indica el nom del topic a usar per la connexió DDE

<office:dde-item> → Indica quina informació de l'aplicació s'hauria d'alliberar

<office:automatic-update> → Indica si la connexió DDE és actualitzada automàticament o no

<office:name> → Indica el nom pel qual es refereix la connexió

<office:conversion-mode> → Indica el mètode pel qual el servidor DDE converteix les dades en números

<table:scenario> → Especifica un escenari en una taula, és a dir, una àrea de la taula on les dades són enllaçades temporalment a una altra taula. Els atributs que pot tenir associats són:

<table:scenario-ranges> → Indica el rang de la taula que es mostra com un escenari

<table:is-active> → Indica si l'escenari es troba actiu o no

<table:display-border> → Indica si el marge que envolta l'escenari es mostra o no

<table:border-color> → Indica el color del marge que es mostra al voltant de l'escenari

<table:copy-back> → Indica si les dades es tornen a copiar a l'escenari si un altre escenari és activat

<table:copy-styles> → Indica si els estils de la taula origen són copiats juntament amb les dades a l'escenari corresponent

<table:copy-formulas> → Indica si les fórmules són copiades de la taula origen a la taula destí

<table:comments> → Conté comentaris sobre l'escenari

<table:protected> → Indica si les dades que són mostrades a l'escenari estan protegides de ser editades o no

<table:shapes> → Aquest element conté tots les figures gràfiques. No té cap atribut associat. Els elements que té continguts són els **<shape>**. Per a més informació sobre ells, consulteu la bibliografia.

<office:text-content-epilogue> → No hi ha elements específics d'epíleg per a un document de text, tot i que pot contenir elements que implementin característiques de taules. Per a més informació sobre aquesta referència, consulteu la bibliografia.

3.4 Alternatives per manipular documents OpenOffice

Actualment existeixen gran varietat d'analitzadors XML (en anglès *XML Parser*) que s'utilitzen per llegir les dades XML d'una font i fer-ne un anàlisi lèxic i sintàctic. La llista és molt llarga, però els més coneguts podrien ser MSXML (de Microsoft), Apache Xerces (en Java) o Expat.

Hi ha altres analitzadors XML que s'utilitzen en forma de components i que són cridats mitjançant mètodes o funcions quan són necessaris. Aquests solen estar implementats dins de funcions pròpies dels llenguatges de programació en els que es troben definits.

Però la realitat és que tots els analitzadors XML segueixen dues definicions d'**APIs** (*Application Programming Interface*) o interfícies per a programar l'aplicació. A continuació anem a detallar-ne les característiques principals de cada una d'elles:

- **API SAX** (*Simple API for XML*)
 - Programari de lliure distribució, gratuït i amb el codi obert
 - Lectura seqüencial del document XML, sense permetre l'accés aleatori
 - Generació d'un conjunt d'esdeveniments quan identifica els termes a cercar (*startDocument, startElement, endElement, characters, ...*)
 - Ràpid i amb poca càrrega a la memòria

Per a Java, l'API SAX són d'un conjunt d'interfícies i classes que defineixen els mètodes per manipular arxius XML, els quals poden ser redefinits en funció del programari i les seves necessitats.

- **API DOM** (*Document Object Model*)
 - Construcció d'una representació en memòria del document XML
 - Especificació de l'organisme W3C, generada en CORBA IDL (*Common Object Request Broker Architecture Interface Definition Language*). Per tant és independent de la plataforma i del llenguatge de programació
 - Establiment d'una estructura arbre amb mètodes i propietats on poder-hi accedir i obtenir-hi les dades emmagatzemades (*getNodeName, getAttributes, ...*)
 - Càrrega en memòria proporcional a la mida i complexitat del document XML, i operacions sobre arbres cares de temps

Per a Java, l'API DOM s'anomena JDOM. És una iniciativa de programari lliure en constant evolució per millorar el rendiment de l'anàlisi i processament de l'estructura DOM.

Fent un estudi d'uns quants llenguatges de programació actuals (Visual Basic .NET, C#, C++, Java), es pot observar que tots d'ells disposen de les funcionalitats necessàries per tractar i manipular documents XML. Ja sigui utilitzant alguna de les APIs descrites o utilitzant algun mètode propi del llenguatge de programació derivat i/o basat en alguna d'aquestes API.

Per tant, l'elecció d'un llenguatge de programació o un altre per a implementar el programari no dependrà única i exclusivament de la possibilitat de disposar d'alguna de les API, doncs com hem dit, tots ells disposen de mètodes que implementen funcionalitats per tractar fitxers XML. Considerarem també per a l'elecció el tipus d'entorn sobre el qual es vol executar l'eina (Windows, Linux, Web, etc).

En aquest cas, com que es vol implementar una eina sobre el sistema operatiu Windows, l'elecció del llenguatge de programació i plataforma queda acotada a uns quants llenguatges.

La tecnologia escollida finalment és la variant **Visual J#** del .NET, que es troba dins el paquet Visual Studio 2005 Express Edition. Aquest paquet es pot descarregar gratuïtament des de la pàgina web de Microsoft, i ha estat una de les raons per les quals s'ha escollit. Altres motius que han fet decantar la balança per aquesta tecnologia han estat els següents:

- Entorn de programació amigable
- Gran quantitat de components i classes per a implementar i generar una eina sobre el sistema operatiu Windows
- Disponibilitat de la classe **XMLReader**, la qual és una variant de l'API SAX, i que es troba dins la llibreria System.XML.
- Disponibilitat de la classe **XMLDocument**, la qual es basa amb l'API DOM, i també es troba dins la llibreria System.XML

Igual que la API SAX, XMLReader és un cursor de lectura que avança únicament endavant. Proporciona un sistema d'accés ràpid i sense emmagatzemament a memòria de les seqüències d'entrada. Pot llegir una seqüència o un document sencer. Permet a l'usuari extraure dades i ometre els registres que no interessin a l'aplicació. La diferència principal entre ells resideix en el fet que el model SAX és d'inserció, és a dir, l'analitzador introdueix esdeveniments a l'aplicació i li notifica cada vegada que s'ha llegit un nou node. En canvi, les aplicacions que utilitzen XMLReader poden extreure els nodes del lector directament.

Per tant sembla clar que durant la tasca de construcció del programari, la classe utilitzada per a fer l'anàlisi i buscar els termes a indexar serà la XMLReader. Ara bé, també s'ha de tenir en compte la necessitat de modificar els arxius XML oportuns per tal que el document OpenOffice incorpori l'índex de termes al final. I aquest requeriment no és satisfet per la classe XMLReader, la qual només permet llegir. Aleshores, durant la tasca per *Incloure l'índex de termes* al final del document haurem d'utilitzar la classe XMLDocument, la qual es basa en l'API DOM, i per tant permet modificar l'estructura de l'arxiu XML.

En el següent capítol entrarem en tots aquests aspectes en major detall, però ja podem afirmar que durant la construcció del programari utilitzarem les corresponents classes de Visual J# derivades tant de l'API SAX (XMLReader) com de l'API DOM (XMLDocument).

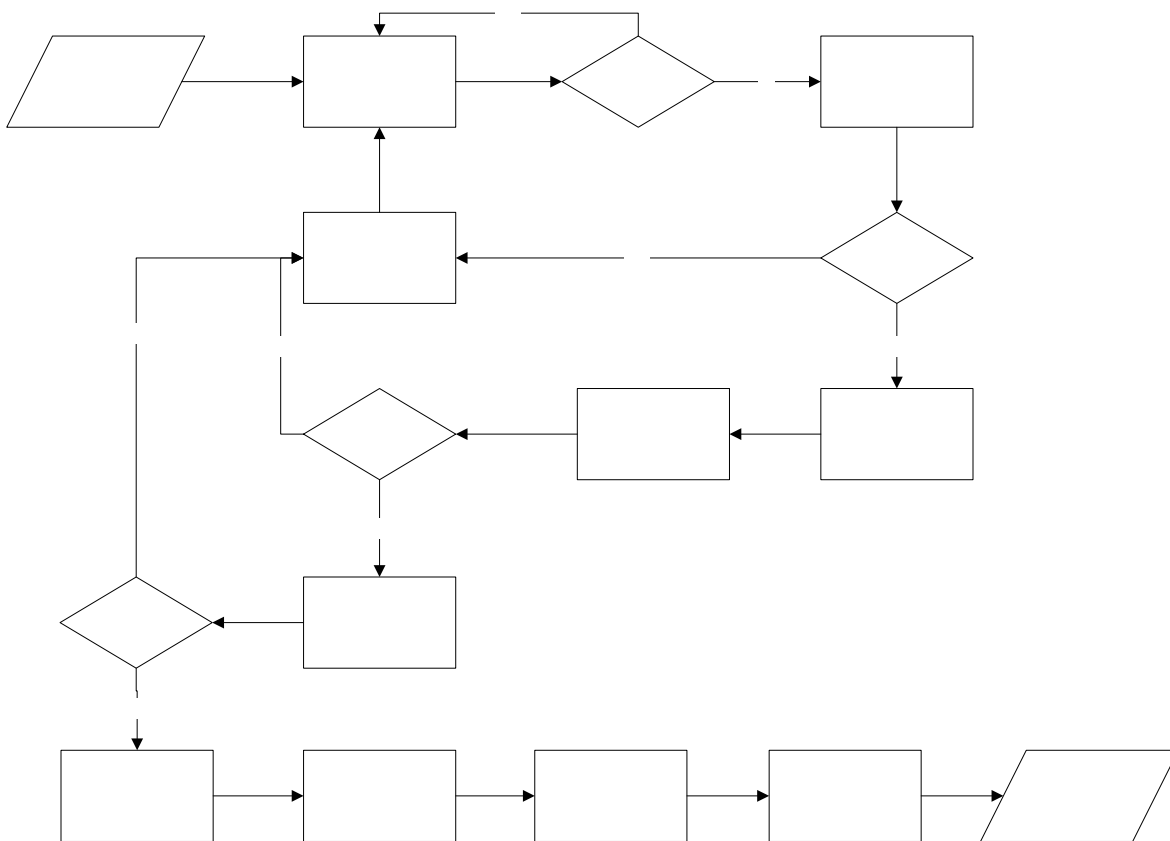
3.5 Construcció del programari

La fase de construcció del programari ha estat dividida en dues fases clarament diferenciades. En la primera d'elles s'han implementat tots els requeriments que no tenen a veure amb l'analitzador lèxic i sintàctic de l'arxiu que es vol tractar. És a dir, els processos de *construcció de l'interfície*, *validacions dels paràmetres*, *descompressió del fitxer*, *compressió dels fitxers* i *neteja de la carpeta temporal*.

La segona, en canvi, tracta els processos d'*anàlisi*, de *generació de l'arxiu XML amb l'índex de termes* i el d'*incloure l'índex de termes* al final del fitxer.

Abans d'explicar cada un dels processos, anem a veure el diagrama de processos per observar el flux d'execució del programari.

3.5.1 Diagrama de processos



Gràcies a aquest diagrama ja podem veure que l'anàlisi per trobar els termes a indexar es farà en els arxius **styles.xml** i **content.xml**.

Anem a veure ara la descripció de cada un dels processos.

3.5.2 Descripció de processos

1. Construcció interfície

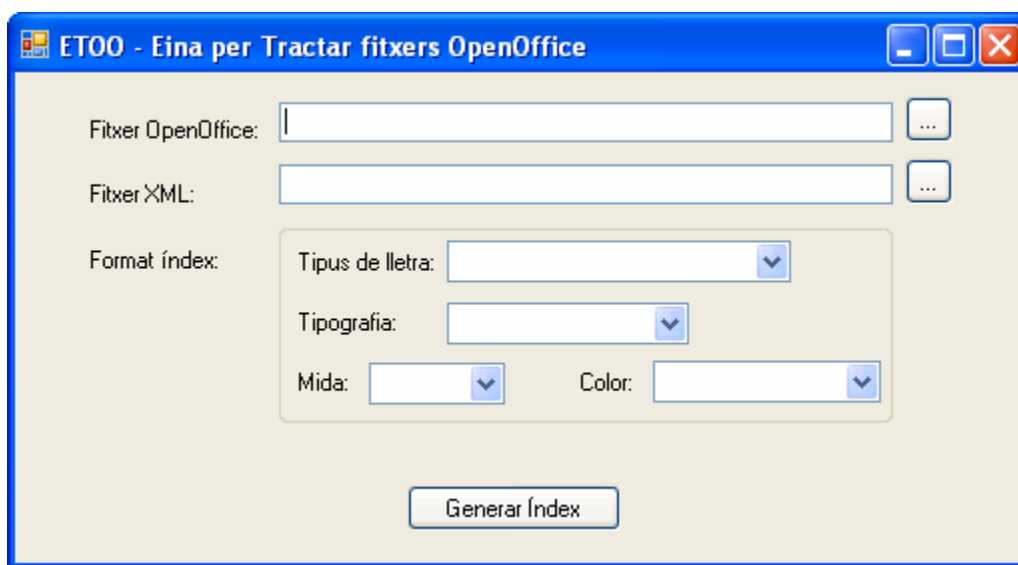
Es tractava de construir una interfície sobre la qual l'usuari no tingui cap problema per entendre el funcionament del programari i li faciliti la interacció per a que aquesta sigui intuïtiva.

La interfície constarà de dues parts. A la part superior hi haurà dos camps on l'usuari podrà informar els noms dels fitxers d'entrada de l'aplicació, cada un dels quals disposarà d'un botó auxiliar (a la seva dreta) que desplegarà una finestra d'exploració d'arxius típica de Windows.

A la part inferior es mostraran quatre camps desplegable on l'usuari podrà informar del format que desitja que tingui l'índex de termes. Aquest format quedarà definit pel tipus, la tipografia, la mida i el color de la lletra.

Finalment, aquesta interfície també disposarà del botó encarregat de generar l'arxiu amb l'índex de termes, *Generar Índex*.

Tot seguit es pot veure una mostra d'aquesta interfície:

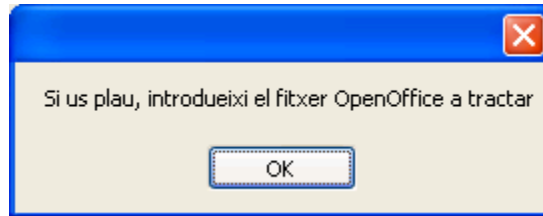


2. Validacions paràmetres

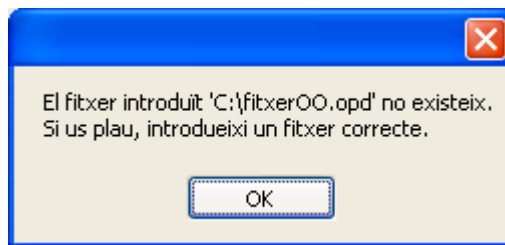
Abans de començar el procés, haurem d'assegurar-nos que l'usuari està facilitant les dades necessàries per a que s'executi l'eina correctament i pugui finalitzar amb èxit. Les validacions es duran a terme en el moment de prémer el botó de *Generar Índex*, i es poden agrupar en validacions realitzades sobre els arxius d'entrada (la funció s'anomena `ValidarTextBox` i retorna un enter) i validacions realitzades sobre el format de l'índex especificat per l'usuari (la funció s'anomena `ValidarFormatIndex` i retorna un enter també).

Anem a veure el comportament de l'aplicació per cada una de les validacions que es practica:

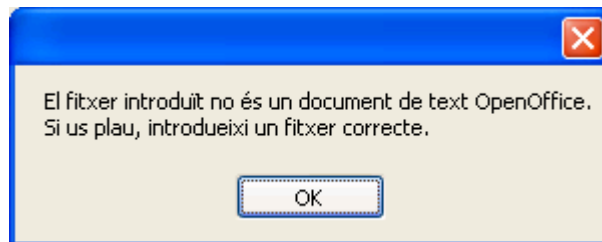
- a. L'usuari facilita l'arxiu OpenOffice a tractar. En cas contrari apareix per pantalla el següent missatge:



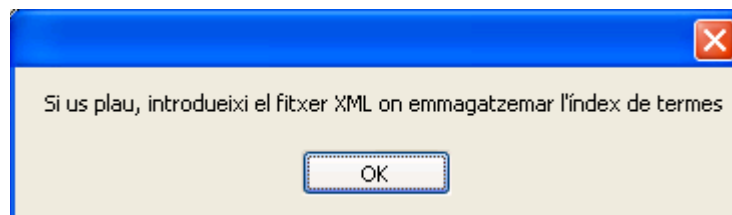
- b. L'arxiu OpenOffice que facilita l'usuari existeix. En cas contrari, apareix per pantalla el següent missatge:



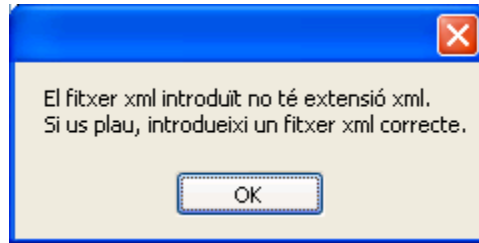
- c. L'arxiu OpenOffice té extensió opd (document de text OpenOffice). En cas contrari, apareix per pantalla el següent missatge:



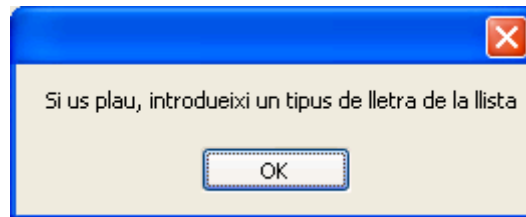
- d. L'usuari facilita un nom d'arxiu xml per a emmagatzemar l'índex de termes. En cas contrari, apareix per pantalla el següent missatge:



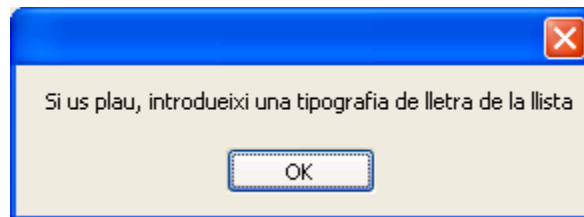
- e. El nom de l'arxiu xml facilitat per l'usuari té realment extensió xml. En cas contrari, apareix per pantalla el següent missatge:



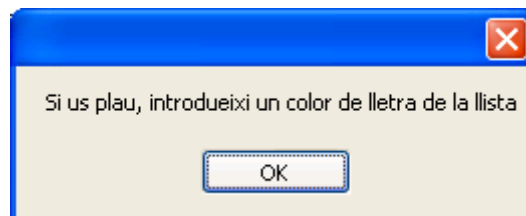
- f. L'usuari facilita un tipus de lletra per a l'índex, i a més, està inclòs en la llista dels possibles tipus. En cas contrari, apareix per pantalla el següent missatge:



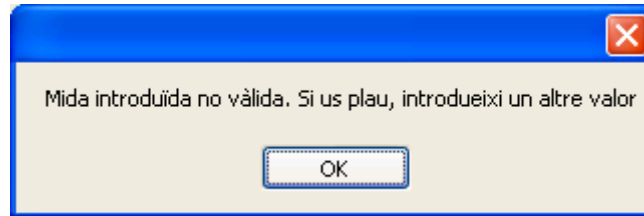
- g. L'usuari facilita una tipografia per a l'índex, i a més, està inclosa en la llista de les possibles tipografies. En cas contrari, apareix per pantalla el següent missatge:



- h. L'usuari facilita un color de lletra per a l'índex, i a més, està inclòs en la llista dels possibles colors. En cas contrari, apareix per pantalla el següent missatge:



- i. L'usuari facilita una mida de lletra per a l'índex vàlida. En cas contrari, apareix per pantalla el següent missatge:

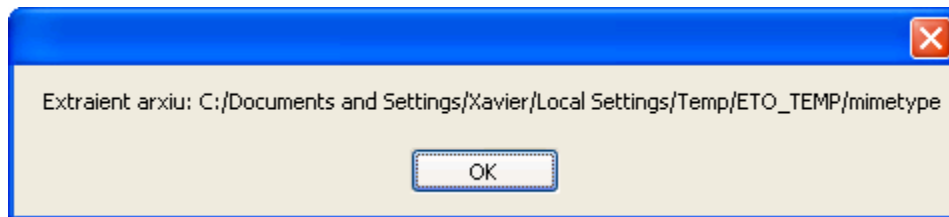


3. Descompressió de l'arxiu OpenOffice

Aquest procés és l'encarregat, com el seu nom indica, de descomprimir l'arxiu OpenOffice que es vol tractar. Tal i com s'ha explicat anteriorment, un fitxer amb format OpenOffice està format per varis arxius, els quals es troben comprimits sota un mateix arxiu.

Així doncs, aquest procés crearà físicament els arxius i directoris que formen l'arxiu OpenOffice en una carpeta temporal, per a que, posteriorment, el procés d'anàlisi pugui tractar directament els arxius *styles.xml* i *content.xml*.

El programari informarà en tot moment de l'arxiu que s'està descomprimint i la ubicació on aquest residirà temporalment. Tot seguit es pot observar el missatge tipus que apareixerà durant aquest procés:

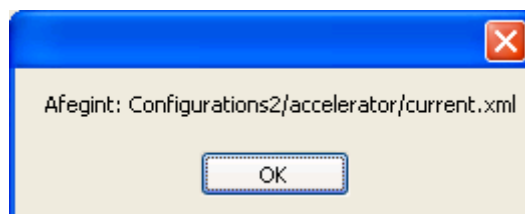


En aquest procés cal destacar l'ús de la classe ZipFile que incorpora Visual J# en una de les seves llibreries, i que ha ajudat a la implementació d'aquesta tasca. Es troba dins la funció amb nom *DesComprimirArxiu* i se li passa una cadena de caràcters com a paràmetre.

4. Comprensió dels arxius que componen l'arxiu OpenOffice

Aquest procés és l'encarregat, com el seu nom indica, de comprimir els arxius que han estat descomprimits anteriorment pel procés de descompressió. A partir de la carpeta temporal on han estat creats els arxius i directoris, aquest procés generarà de nou l'arxiu OpenOffice per a que pugui ser tractat com a tal.

Com en el procés anterior, el programari informarà en tot moment de l'arxiu que està sent comprimit dins el fitxer OpenOffice. Tot seguit es pot observar el missatge tipus que apareixerà durant aquest procés:

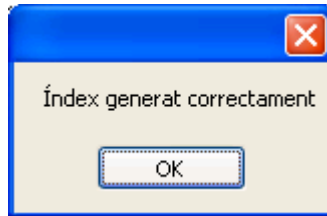


En aquest procés cal destacar l'ús de la classe `ZipEntry`. Aquesta tasca es troba dins la funció `ComprimirArxiu` i utilitza una altra funció anomenada `GeneraLlistaArxius` la qual obté en una llista tots els arxius d'una carpeta, la qual es dona com a paràmetre d'entrada.

5. Neteja de la carpeta temporal

Aquest procés és el responsable d'eliminar la carpeta temporal que ha creat el procés de descompressió, tant el seu contingut com la mateixa carpeta. Es troba dins la funció anomenada `NetejaCarpeta` rebent com a paràmetre una cadena de caràcters que representa la carpeta que s'ha d'eliminar.

Al finalitzar la neteja, apareixerà el següent missatge de confirmació que tot ha anat bé:



6. Analitzador

Aquest és un dels processos més importants i es troba dins la funció anomenada `LlegirArxiuXML`, la qual rep com a paràmetre d'entrada una cadena de caràcters que correspon a un arxiu XML. Tal i com es pot veure en el diagrama de processos, aquesta funció es crida dues vegades. Una per llegir l'arxiu `styles.xml`, i una altra per llegir l'arxiu `content.xml`.

El procés consisteix en utilitzar una instància de la classe `XMLReader`, que incorpora Visual J# i que ve derivada de l'API SAX, per anar llegint tag per tag (o node per node) fins a trobar aquells nodes que són d'interès perquè poden contenir termes. Els nodes que han estat seleccionats com d'interès i que marquen seccions de text d'un document d'OpenOffice són:

- `style:header`: Per indicar una capçalera de pàgina. Es troba dins l'arxiu `styles.xml`.
- `style:footer`: Per indicar un peu de pàgina. Es troba dins l'arxiu `styles.xml`
- `text:list-item`: Per indicar una llista.
- `text:list-header`: Per indicar una llista.
- `text:numbered-paragraph`: Per indicar un paràgraf numerat.
- `text:section`: Per indicar una secció.
- `text:table-of-content`: Per indicar un índex de continguts.
- `text:illustration-index`: Per indicar un índex d'il·lustracions.
- `text:table-index`: Per indicar un índex de taules.
- `text:object-index`: Per indicar un índex d'objectes.
- `text:user-index`: Per indicar un índex d'usuari
- `text:alphabetical-index`: Per indicar un índex alfabètic
- `text:bibliography`: Per indicar un índex bibliogràfic
- `table:table`: Per indicar una taula
- `text:p`: Per indicar un paràgraf
- `text:h`: Per indicar una capçalera

Com també sabem, els tags `text:p` i `text:h` poden contenir a l'hora altres tags que també poden ser d'interès. Aquest és el motiu pel qual aquest procés crida a una altra funció anomenada `tractamentElemPara`, la qual rep com a paràmetre d'entrada l'objecte `XMLReader` que està sent llegit. Els nodes que han estat seleccionats com d'interès dins elements paràgraf i que marquen seccions de text són:

- `text:a`: Per indicar un enllaç.
- `text:reference-ref`: Per indicar una referència.
- `text:bookmark-ref`: Per indicar una referència.
- `text:note-ref`: Per indicar una referència.
- `text:sequence-ref`: Per indicar una referència.
- `text:note`: Per indicar una nota.
- `text:ruby`: Per indicar un text en rubí.
- `text:title`: Per indicar un títol.
- `text:date`: Per indicar una data.
- `text:time`: Per indicar una hora.
- `text:page-number`: Per indicar un número de pàgina.
- `text:page-continuation`: Per indicar un recordatori de pàgina.
- `text:h`: Per indicar una capçalera

El procés incorpora una variable de la classe `Vector` per anar registrant cada una de les seccions on es troben termes, i que corresponen a les seccions dels tags que acabem de veure. Així, cada cop que el procés troba un terme dins el tag `text:p` o `text:h` crida a la funció anomenada `introTaulaContinguts`, la qual rep com a paràmetre d'entrada el terme llegit i la secció actual (gràcies a la variable de la classe `Vector`). Aquesta funció comprova si el terme ja està registrat. Si és així, només incrementa el nombre de citacions del terme, i inclou la secció on s'acaba de llegir. Si el terme no està registrat, crea una nova entrada amb el terme, les citacions (que serà 1 al ser la primera vegada) i la secció on s'ha llegit. Cal mencionar que els termes s'emmagatzemen en una estructura del tipus **Taula de Hash**. S'ha decidit aquesta estructura per la rapidesa i facilitat d'accés que té.

7. Generació XML amb índex de termes

Aquest procés s'encarrega de generar l'arxiu XML especificat per l'usuari amb l'índex de termes indexats. Es troba dins la funció `GeneraTermesXML`, la qual rep com a paràmetre d'entrada una cadena de caràcters que representa el nom de l'arxiu XML.

El procés consisteix en utilitzar una instància de la classe **XMLWriter**, que incorpora `Visual J#` per anar escrivint element per element amb el format que s'especifica més avall. Però per escriure, abans s'accedeix de manera seqüencial a l'estructura que conté tots els termes llegits, i mentre aquesta tingui elements, es van adjuntant a l'arxiu. Cal mencionar que si l'arxiu XML existeix i té contingut, aquest serà eliminat.

Tot seguit es mostra un exemple d'arxiu XML on poder veure el format.

```

<?xml version="1.0" encoding="utf-8" ?>
<index>
  <format>
    <lletra>Century</lletra>
    <tipografia>Negrita</tipografia>
    <tamany>9</tamany>
    <color>Negre</color>
  </format>
  <termes>
    <terme>
      <mot>describe</mot>
      <citacions>1</citacions>
      <seccions>
        <seccio>Paràgraf</seccio>
      </seccions>
    </terme>
    <terme>
      <mot>documento</mot>
      <citacions>3</citacions>
      <seccions>
        <seccio>Llista</seccio>
        <seccio>Paràgraf</seccio>
        <seccio>Paràgraf</seccio>
      </seccions>
    </terme>
  </termes>
</index>

```

8. Incloure índex de termes al final del fitxer

L'objectiu d'aquest procés és modificar l'arxiu content.xml de tal manera que:

- S'inclougui l'índex de termes al final del document OpenOffice
- El format del text de l'índex sigui l'especificat per l'usuari
- L'índex resideixi dins una taula de tres columnes

Val a dir que el tercer punt no era un requeriment específic d'aquest procés, però s'ha cregut oportú mostrar l'índex de termes dins una taula per així millorar la presentació del document final d'OpenOffice.

Per desenvolupar aquest procés s'ha creat una funció anomenada `IntroFormatXML` que rep com a paràmetre una cadena de caràcters que representa el nom del fitxer XML, en el nostre cas, el content.xml.

La seva construcció es basa en l'ús de les classes **XMLDocument**, **XMLNode** i **XMLAttribute**. La primera d'elles s'utilitza per representar un arxiu XML en un arbre i ve derivada de l'API DOM. Les altres dues classes són derivades de la primera. Una s'utilitza per manipular un node o element, i l'altre per manipular els atributs d'un node.

Primerament, el procés modifica el node principal *office:font-face-decls* afegint-hi l'estil amb nom "FormatIndex". A continuació es mostra una part del codi:

```
copiat = XDoc.CreateNode(XmlNodeType.Element, "style", "font-face",
"urn:oasis:names:tc:opendocument:xmlns:style:1.0");
    xatrib = XDoc.CreateAttribute("style", "name",
    "urn:oasis:names:tc:opendocument:xmlns:style:1.0");
    xatrib.set_Value("FormatIndex");
    copiat.get_Attributes().Append(xatrib);
```

Després li toca el torn al node principal *office:automatic-styles*. En aquest node se li afegeix també l'estil de text amb nom "EstilIndex" que utilitza com a estil font "FormatIndex", definit anteriorment dins el node *office:font-face-decls*. A continuació es mostra una part del codi:

```
copiat = XDoc.CreateNode(XmlNodeType.Element, "style", "style",
"urn:oasis:names:tc:opendocument:xmlns:style:1.0");
    xatrib = XDoc.CreateAttribute("style", "name",
    "urn:oasis:names:tc:opendocument:xmlns:style:1.0");
    xatrib.set_Value("EstilIndex");
    copiat.get_Attributes().Append(xatrib);
```

Però a més d'aquest nou estil, al node *office:automatic-styles* se li afegeixen els estils de la taula, els estils de les tres columnes de la taula, i l'estil de les cel·les de la taula. Tot seguit es mostra una part del codi:

```
copiat = XDoc.CreateNode(XmlNodeType.Element, "style", "style",
"urn:oasis:names:tc:opendocument:xmlns:style:1.0");
xatrib = XDoc.CreateAttribute("style", "name",
"urn:oasis:names:tc:opendocument:xmlns:style:1.0");
xatrib.set_Value("Taula_Index");
copiat.get_Attributes().Append(xatrib);

xatrib = XDoc.CreateAttribute("style", "family",
"urn:oasis:names:tc:opendocument:xmlns:style:1.0");
xatrib.set_Value("table");
copiat.get_Attributes().Append(xatrib);
```

Per acabar el procés, i quan es tenen els estils de text i de la taula definits, només queda modificar el node *office:body* per incloure la taula amb l'índex de termes al final del document.

Es comença introduint els nodes corresponents que defineixen la taula i les seves columnes. Després és el moment de llegir l'estructura on s'han emmagatzemat tots els termes, i per cada registre llegit es crea una nova fila dins de la taula. Aleshores s'omple la primera cel·la, corresponent a la primera columna, amb el nom del terme. La segona cel·la, corresponent a la segona columna, amb el nombre de citacions del terme. I la tercera cel·la, corresponent a la tercera columna, amb les seccions on s'ha mencionat el terme.

Quan s'acaben les entrades de l'estructura, passem a emmagatzemar el document modificat dins l'arxiu que havien passat com a paràmetre d'entrada a la funció.

A continuació es pot detallar una mostra del codi.

```

copiat = XDoc.CreateNode(XmlNodeType.Element, "table", "table",
"urn:oasis:names:tc:opendocument:xmlns:table:1.0");
atrib_fill = XDoc.CreateAttribute("table", "name",
"urn:oasis:names:tc:opendocument:xmlns:table:1.0");
atrib_fill.set_Value("Taula_Indexada");
copiat.get_Attributes().Append(atrib_fill);

.....

for (llistat = TaulaContinguts.keys(); llistat.hasMoreElements(); )
{
s = (simbol)TaulaContinguts.get((String)llistat.nextElement());
fila = XDoc.CreateNode(XmlNodeType.Element, "table", "table-row",
"urn:oasis:names:tc:opendocument:xmlns:table:1.0");

cela = XDoc.CreateNode(XmlNodeType.Element, "table", "table-cell",
"urn:oasis:names:tc:opendocument:xmlns:table:1.0");
atrib_fill = XDoc.CreateAttribute("table", "style-name",
"urn:oasis:names:tc:opendocument:xmlns:table:1.0");
atrib_fill.set_Value("Cella_Index");
cela.get_Attributes().Append(atrib_fill);
atrib_fill = XDoc.CreateAttribute("office", "value-type",
"urn:oasis:names:tc:opendocument:xmlns:office:1.0");
atrib_fill.set_Value("string");
cela.get_Attributes().Append(atrib_fill);
.....
}

```

3.6 Construcció DTDs

Un altre dels objectius d'aquest projecte de final de carrera és construir els DTDs que validin els fitxers XML resultants. Així, s'han hagut de crear dos DTDs:

- Un per validar l'arxiu XML especificat per l'usuari, on s'ha emmagatzemat l'índex de termes. El format d'aquest arxiu ha estat escollit a lliure elecció per l'autor d'aquest projecte, i ja se n'ha mostrat un exemple en un capítol anterior. L'arxiu DTD s'anomena **index.dtd**, i es pot trobar adjunt al lliurament del projecte. De totes maneres, i degut a la seva simplicitat, ara se'n mostra el contingut:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT index (format,termes)>
<!ELEMENT format (lletra,tipografia,tamany,color)>
<!ELEMENT lletra (#PCDATA)>
<!ELEMENT tipografia (#PCDATA)>
<!ELEMENT tamany (#PCDATA)>
<!ELEMENT color (#PCDATA)>
<!ELEMENT termes (terme*)>
<!ELEMENT terme (mot,citacions,seccions)>
<!ELEMENT mot (#PCDATA)>
<!ELEMENT citacions (#PCDATA)>
<!ELEMENT seccions (seccio+)>
<!ELEMENT seccio (#PCDATA)>
```

- L'altre DTD serveix per validar l'arxiu content.xml, el qual ha estat modificat durant el procés de generació de l'índex. El nom de l'arxiu és **content.dtd** i també es pot trobar adjunta al lliurament del projecte. Per a la seva construcció ha estat necessari un anàlisi molt exhaustiu dels elements, ja fossin opcionals o obligatoris, i dels possibles valors dels atributs de cada element. L'arxiu està estructurat en les següents parts: tipus de dades, espai de noms, definicions, elements i atributs office, estils, metadades, scripts, drawing, text, taules, chart, estils de dades, formularis i settings.

3.7 Conclusions

Arribats en aquest últim capítol ve el moment de reflexionar sobre la feina feta durant el semestre. Potser en el meu cas, la reflexió ha de ser dels dos últims semestres, doncs ha estat el temps que he necessitat per obtenir aquest treball, del qual ara n'escric les últimes línies.

Reflexionant crec que podria classificar les conclusions en dos tipus o categories. Unes són les que fan referència a temes educatius i acadèmics, i que tracten sobre els beneficis i el coneixement que he adquirit a mesura que anava desenvolupant i donant forma al projecte. Les altres fan referència a temes personals.

Així doncs, penso que aquest projecte m'ha servit per conèixer més a fons un producte informàtic, l'OpenOffice, cada dia més extès, i del qual no en tenia el gust de conèixer abans de començar amb aquest projecte. Ara no és moment de destacar-ne les qualitats, però crec que és un bon exemple del fort creixement que està tenint el programari lliure dins la informàtica. Sembla la gran incògnita a mitjà termini. Aconseguirà el programari lliure competir amb el programari propietari? L'aposta és complicada, però crec que entre tots els professionals que treballem en aquest sector hauríem de potenciar l'ús del programari lliure. Desgraciadament existeixen les grans empreses, a les quals potser no els hi interessa aquest tipus de programari.

Canviant de tema, també puc afirmar el plaer que obtingut al treballar amb Visual J# del Visual Studio 2005. Realment és una versió molt recomanable per treballar. Molta potència, fàcil i intuïtiu pels desenvolupadors. A més, tot i no ser de programari lliure, sí que és una versió gratuïta, la qual cosa la fa encara més recomanable.

A nivell personal només m'agradaria destacar que una bona planificació del projecte et pot salvar d'un desastre. Tots sabem que les planificacions no sempre es poden seguir, però si més no, fent el seguiment del projecte pots saber l'estat en el que es troba, i intuir si al final s'obtindrà l'èxit o el fracàs. Abans de desenvolupar aquest projecte pensava que l'èxit no depenia d'una bona planificació. Creia que si durant la recta final d'un projecte es poden dedicar tots els recursos i esforços a les feines incompletes s'aconsegueix l'èxit igualment. Ara puc afirmar que difícilment es pot obtenir l'èxit d'un projecte si no se'n fa un seguiment i una bona planificació.

4 Bibliografia

Enllaços a Internet:

<http://es.wikipedia.org/wiki/OpenDocument>

<http://www.oasis-open.org/committees/download.php/12575/OpenDocument-v1.0-os.pdf>

<http://msdn2.microsoft.com/es-es/default.aspx>

Documents electrònics:

ManualJLexiCup,pdf , *Francesc Bagés Vaqué*. Universitat Oberta de Catalunya

Anàlisi lèxica , *Joan Codina i Banti*. Universitat Oberta de Catalunya

Llenguatges de marques , *Francesc Santanach i Delisau*. Universitat Oberta de Catalunya