

"I will do this. Nothing in my life matters except this. No moment in my life exists except this moment. I am born in this moment, and if I fail, I will die in this moment."

Raistlin Majere.

1. Introducció	7
1.1. Descripció del Projecte.....	7
1.2. Objectius.....	8
1.3. Metodologia.....	9
1.3.1. Anàlisi de Requeriments	9
1.3.2. Anàlisi del SI	9
1.3.3. Disseny.....	9
1.3.4. Implementació.....	9
1.4. Sobre aquest document.....	10
2. Marc de treball	13
2.1. La informació a les organitzacions.....	13
2.1.1. Les metadades	13
2.2. Diferències en la terminologia.....	14
2.3. Gestió documental	15
2.3.1. Record management.....	15
2.3.2. Sistemes de Recuperació de la Informació.....	16
2.4. PH Systems.....	17
2.5. Utilització de les TIC	18
3. Requeriments del projecte.....	21
3.1. Arxiu Òptic.....	22
3.1.1. Arxiu central.....	22
3.1.2. Sistema de RI.....	23
3.1.3. Magatzem Virtual	23
3.2. Adquisició de documents i administració.....	24
3.2.1. Adquisició de documents.....	24
3.2.2. Administració de l'arxiu òptic.....	24
3.2.2.1. Gestió de plantilles.....	24
3.2.2.2. Gestió d'esquemes.....	25
3.2.2.3. Visualització de documents.....	25
3.3. Integració amb el Lirba	26
3.3.1. Marc de treball per a Record Management.....	26
3.3.2. Marc de treball per la factoria de serveis.....	27
4. Alternatives	29
4.1. Arxiu Òptic.....	29
4.2. Adquisició i administració de documents.....	30
4.3. Integració amb el Lirba	30
5. Anàlisi del projecte	32
5.1. Model General.....	32
5.2. Arxiu Òptic.....	33
5.2.1. Gestió d'Esquemes	34
5.2.2. Gestió de Plantilles	36
5.2.3. Gestió de Vistes	38
5.2.4. Gestió de Documents	40
5.3. Administració i Adquisició de documents	43
5.3.1. Gestió Arxiu Òptic	44
5.3.2. Adquisició de documents.....	47
5.4. Integració amb el Lirba	49
5.4.1. Record Management.....	50
5.4.1.1. Record Management Genèric	51
5.4.1.2. Record Management Específic.....	54

5.4.1.3.	Record Management Arxiu Òptic.....	56
5.4.2.	Repositori.....	58
5.4.2.1.	Visualització.....	59
5.4.2.2.	Configuració.....	61
5.4.3.	Factoria de serveis.....	63
5.4.3.1.	Factoria de serveis.....	64
5.4.3.2.	SoapParser.....	66
6.	Disseny.....	69
6.1.	Diagrames de seqüència.....	69
6.1.1.	Diagrames de seqüència de l'Arxiu Òptic.....	70
6.1.1.1.	La inicialització del servei remot.....	70
6.1.1.2.	La creació d'un document.....	71
6.1.1.3.	Modificar metainformació d'un document.....	72
6.1.2.	Diagrames de seqüència de l'administració i adquisició de documents.....	73
6.1.2.1.	L'adquisició d'una sèrie de documents.....	73
6.1.2.2.	Creació d'una plantilla.....	74
6.1.3.	Diagrames de seqüència de la integració amb el Lirba.....	75
6.1.3.1.	La inicialització d'un Servei Web.....	75
6.1.3.2.	La inicialització de l'arxiu òptic.....	76
6.1.3.3.	La crida d'una funcionalitat d'un Servei Web.....	77
6.2.	Utilització de Patrons.....	78
6.2.1.	Patrons GRASP.....	78
6.2.2.	Patrons GoF.....	78
6.2.2.1.	De creació.....	78
6.2.2.2.	De Estructura.....	78
6.2.2.3.	De Comportament.....	79
6.3.	Diagrama de Classes.....	80
6.3.1.	Arxiu Òptic.....	80
6.3.2.	Administració i adquisició de documents.....	81
6.3.3.	Integració amb Lirba.....	82
7.	Implementació.....	85
7.1.	Arxiu Òptic.....	85
7.1.1.	Servei Remot.....	85
7.1.2.	Carregador de classes.....	88
7.1.3.	Generador XML.....	91
7.2.	Administració i adquisició de documents.....	103
7.2.1.	Escaneig d'imatges.....	103
7.2.2.	Identificació de codis de barres.....	117
7.2.3.	Exemple: adquisició via escaner.....	121
7.3.	Integració Lirba.....	123
7.3.1.	Generació de missatges Soap.....	123
7.3.2.	Factoria de Serveis.....	130
7.3.3.	Exemple: incorporació d'un servei.....	134
7.3.4.	Accés remot a l'arxiu òptic.....	135
7.4.	Mòduls Implementats.....	138
7.4.1.	Mòduls Arxiu Òptic.....	138
7.4.2.	Mòduls Lirba.....	142
8.	Interfícies d'usuari.....	145
8.1.	Administració i adquisició de documents.....	145
8.1.1.	Adquisició de documents.....	145
8.1.2.	Editor d'esquemes.....	146
8.1.3.	Editor de plantilles.....	147

8.1.4. Llista elements.....	148
8.1.5. Definir Tipus documents.....	149
8.2. Integració amb el Lirba	150
9. Cas pràctic: Entrada d'assentaments comptables.	153
10. Conclusions	167
11. Bibliografia	171
Annex I: Instal.lació e Implantació	173
Arxiu Òptic.....	173
Client Arxiu Òptic	178
Lirba... ..	180

1. Introducció

1.1. Descripció del Projecte

En els darrers anys hem assistit a una important evolució tecnològica, tant en la societat com en les organitzacions.

Actualment, pren gran rellevància en el sí de les organitzacions, la gestió documental. La gestió documental la podem definir com: “el conjunt d'operacions i tècniques relatives a: la concepció, el desenvolupament, la implantació i l'avaluació dels sistemes administratius necessaris, des de la creació dels documents fins la seva destrucció o transferència a l'arxiu, per tal de garantir l'eficàcia i l'eficiència en la gestió administrativa”.

La incorporació d'un bon sistema de gestió documental redundarà en beneficis econòmics en les organitzacions, ja que facilita la identificació, la quantificació i la disposició dels documents; permetent economitzar en recursos materials i humans. Això ens porta a una millor organització i gestió de la informació, encaminada a optimitzar novament recursos humans i tecnològics.

Tenim així que PH Systems disposa fins aquests moments d'un conjunt d'aplicacions informàtiques que conformen una solució de gestió de negoci integrada pensada per diversos sectors. Però aquestes aplicacions no disposen d'una eina que pugui fer una gestió de la documentació d'una organització.

Així doncs aprofitant la faceta d'automatització de les TIC podem implementar una aplicació que permeti:

- Un arxiu central per a tots els documents
- Tenir la Informació disponible en tot moment des de qualsevol lloc
- Facilitat de recerca i ràpida recuperació dels documents
- Integració amb qualsevol flux de treball
- Màxima seguretat

Es pot també cercar una faceta d'augmentació en aquesta aplicació. Ja que els actius d'informació, són documents administratius, o sigui informació formal estructurada, es pot emprar un metamodel per definir l'estructura d'aquests documents i la informació que contenen (mitjançant un OCR). Aquest metamodel seria emprat per a la indexació dels documents i per tant conformaria el sistema de RI. En aquest cas doncs, els elements del metamodel serien els elements d'indexació i les aplicacions podrien satisfer les seves necessitats d'informació emprant els mateixos elements del metamodel segons les necessitats. La informació final estaria continguda dins de l'estructura del document.

1.2. Objectius

L'objectiu d'aquest projecte és millorar la gestió i/o la productivitat dels procediments informatitzats, que tenen com a element inicial en el seu flux de dades, els actius d'informació.

Això es pot subdividir en diferents objectius concrets:

- La utilització d'un Sistema d'informació que permeti accedir i recuperar els documents d'arxiu d'una organització, sense haver d'anar a l'arxiu físic.
- Facilitar un procediment automatitzat, àgil i versàtil de creació de documents electrònics a partir dels documents d'arxiu.
- La integració dels actius d'informació amb la gestió de negoci que proporciona un ERP com el Lirba.

A part d'aquests objectius, també n'hi ha uns de secundaris:

- Dotar de semàntica els documents electrònics corresponents a documents administratius.
- Establir lligams entre aquesta semàntica i un flux de treball del Lirba.
- Permetre al Lirba treballar amb més d'un gestor de documents electrònics.

1.3. Metodologia

Per desenvolupar el Sistema d'Informació del que tracta aquest projecte, s'utilitzarà una metodologia UML. També s'ha decidit utilitzar una eina CASE amb la fi de poder generar una documentació amb diagrames, exemples i definicions.

1.3.1. Anàlisi de Requeriments

En l'etapa de requeriments, es buscarà les necessitats que té l'usuari final alhora d'interactuar amb el SI, i en la manera que es presentarà la solució. El procediment a seguir consisteix en:

- Identificar les necessitats.
- Definir quins components podran ser reemplaçats per eines de tercers i/o l'integració amb aquests.

1.3.2. Anàlisi del SI

En aquesta etapa s'estudiaran els problemes que ha de resoldre el SI, plantejats en l'apartat de requeriments, i la forma com l'usuari interactuarà amb ell per obtenir la resposta adequada. El procediment a seguir consisteix en:

- Definir els Casos d'Ús detalladament.
- Estudiar els processos interns que ha de realitzar.

1.3.3. Disseny

En la fase de disseny estudiarem com resoldre els problemes. Es buscarà adequar l'anàlisi fet prèviament a les característiques i abast del software. S'estudiarà el flux de dades de cada funcionalitat que cal implementar. El procediment a seguir consisteix en:

- Desenvolupar un model de components flexible.
- Desenvolupar els processos de Control i Comunicació.

1.3.4. Implementació

Amb aquesta etapa s'establiran els processos estudiats i definits en les etapes anteriors per tal d'obtenir el SI final. El procediment a seguir consisteix en:

- Construir el gestor d'actius d'informació
- Construir la interfície d'adquisició de documents.
- Implementar la integració amb la capa de negoci.

1.4. Sobre aquest document

Aquest document és fruit de la planificació i elaboració d'un SI per la gestió dels documents administratius; una mancança en les solucions de software existents a PH Systems.

Com que aquest, és un projecte d'una empresa privada (agraeixo a en Robert i en Manel la seva confiança), queden exclosos d'aquest document detalls importants per l'empresa com la temporalització i les proves (i els seus resultats), ja que estan íntimament lligats a les condicions de facturació de la primera implantació a Casa Darnés.

Aquest projecte, s'ha dividit en diverses etapes, en les que han col·laborat altres persones de l'empresa per la que treballava. Entre elles, en Pere Galan cap de I+D e instigador inicial del projecte (que em va deixar fer-ho amb SCHEMA i XML sobre OCR). En Jordi Soler i en Jordi Prat com a responsables de producte i manteniment respectivament, revisant la integració dels diferents components amb el Lirba i els seus fluxos de treball (he d'agrair la llibertat que em van donar, i no només aquí, sinó també quan vaig crear el nucli del Lirba i la migració del NewGen cap aquest). En Jordi Naranjo i en Lluís Rodríguez (que va assumir les meves responsabilitats quan vaig plegar), ajudant en els primers contactes amb el client i en la finalització de codi quan apremiaven les dates d'entrega. Finalment en Pere de Transports Badosa, cap de sistemes d'aquesta empresa, que va col·laborar en tot el procés d'implantació (i tot i que va sofrir tots els errors, cada cop que hi anava, sempre va treure temps per a convidar-me a dinar).

El meu treball consistia en, l'anàlisi i el disseny de l'aplicació i pràcticament la totalitat de la implementació (que es va realitzar al llarg de més de 2 anys). Per això, presento aquesta memòria que consta de 12 capítols.

El Capítol 1, Introducció, es fa un descripció del projecte i es defineixen els objectius que es volen assolir.

El Capítol 2, Marc de treball, ens permet introduir aspectes dins l'àmbit d'influència d'aquest projecte; també sobre l'entorn de desenvolupament i producció.

El Capítol 3, Requeriments del projecte, s'especifiquen els requisits necessaris per centralitzar la gestió de documents administratius amb un SI.

El Capítol 4, Alternatives, s'expliquen quines opcions es van valorar i d'aquestes quines es van escollir finalment.

El Capítol 5, Anàlisi del projecte, detalla els requisits comentats en el capítol 4, i especifica els processos interns de forma clara i concisa.

El Capítol 6, Disseny de l'aplicació, detallarà quins procediments seguirà cada funcionalitat del SI.

El Capítol 7, Implementació, conté la implementació de les funcionalitats més importants del projecte; així com una descripció de tots els mòduls implementats.

El Capítol 8, Interfície d'Usuari, especifica la forma com s'ha construït la interfície i els detalls de tecnologies client per guanyar usabilitat i satisfacció de l'usuari que ha d'interactuar amb el SI.

El Capítol 9, Exemple: Entrada d'assentaments Comptables, tot el projecte té la finalitat de millorar la gestió dels fluxos de treball, mitjançant la incorporació de la gestió dels documents administratius, en aquest capítol és mostra el que realment pot fer en la millora de la gestió aquest projecte.

El Capítol 10, Conclusions, estudiem els resultats i la utilitat de l'eina implementada, valorant els punts forts i els dèbils d'aquesta. Així com futures ampliacions o millores a seguir.

El Capítol 11, Bibliografia, descriu la Bibliografia utilitzada per l'elaboració d'aquest projecte.

L'Annex A, Instal·lació e Implantació, aquí es detallen els principals fitxers de configuració que permeten una instal·lació a mida i flexible segons la funcionalitat a implantar i els productes preexistents.

2. Marc de treball

2.1. La informació a les organitzacions

La informació que s'utilitza en una organització pot adquirir formes molt variades. En aquest primer moment no ens interessa tant el suport de la informació (llibre, CD, disc d'ordinador, etc.) com el valor del contingut de cara a la presa de decisions en les activitats de gestió. Així, les unitats venudes d'un determinat producte és un tipus d'informació que té un valor diferent d'un rumor sobre el nomenament del nou director de recursos. Les dues dades són elements d'informació generats en l'organització que són útils per a la presa de decisions, però tenen un valor diferent i unes possibilitats d'anàlisi i de tractament totalment diferenciades.

La informació formal és la que està fonamentada i documentada, de la qual podem saber de manera certa el seu origen. Per tant ve arxivada en algun tipus de registre i obeeix algunes regles i procediments propis de l'organització.

La informació formal es pot presentar de manera estructurada o de manera no estructurada. La informació estructurada és aquella que té un esquema determinat que facilita d'obtenir-ne el significat. La informació estructurada té un tractament més senzill mitjançant tecnologies de la informació.

2.1.1. Les metadades

Les metadades són la via per a comunicar informació sobre un document o sobre els recursos que directament es relacionen amb la seva accessibilitat. Són, a més, una eina bàsica en l'organització, classificació i descripció de la informació, tasques pròpies de l'Arquitectura de la Informació.

Metadada és tota aquella informació descriptiva sobre el context, qualitat, condició o característiques d'un recurs, dada o objecte amb la finalitat de facilitar la seva recuperació, autenticació, avaluació, preservació i/o interoperabilitat.

Així, en definir un grup de metadades per a un objecte l'estem descrivint de manera que aquesta informació ajuda els usuaris a identificar si l'objecte els és útil i a localitzar-lo ràpidament.

Són exemples de metadades:

- L'encapçalament d'un fitxer multimèdia (imatge, vídeo o àudio).
- El resum d'un document.
- El catàleg d'una base de dades.
- Les paraules extretes d'un text.
- Les "pàgines grogues".

2.2. Diferències en la terminologia

Degut a diferències filosòfiques i semàntiques entre els models administratius anglòfons i llatins, es manté fins a dia d'avui certa tensió entre les nocions de gestió documental i l'arxivística, com si de coses molt diferents es tractessin. Per una part, en català no existeix una paraula del tot equivalent al concepte de “record” anglès, que cobreix tot allò que documenta quelcom. En català s'empra “document d'arxiu”, que té un significat més pobre. Quan en anglès es parla de Record Management, s'està fent referència a la gestió documental... de documents d'arxiu específicament. Però per altra banda en anglès la paraula “archive” significa invariablement arxiu històric, mentre que en català la paraula arxiu s'entén en varies facetes: arxiu de gestió, intermedi i finalment arxiu històric.

Degut a aquestes diferències es produeixen malentesos i certa tendència, en els països hispanoparlants i luxòfons almenys, de creure que l'arxivística només s'aplica als arxius històric, o que el “Records Management” es refereix a la gestió de qualsevol tipus de document. En realitat no és així: l'arxivística cobreix el “records management” i la gestió d'arxius històrics, mentres que la gestió documental es més àmplia que el “records management” ja que es pot aplicar a qualsevol tipus de document, no tant sols els administratius i els d'arxiu.

La gestió documental ha estat sempre una necessitat, un problema per a les organitzacions, representant despeses en locals i magatzems, infraestructures per a garantir l'estat de conservació; temps dedicat a l'organització recerca de documents, duplicacions, despeses de fotocopies, fax, etc.

La major part de les organitzacions necessiten accedir i consultar de forma freqüent la informació arxivada. En d'altres es la importància dels documents o el volum d'informació el que les estimula a buscar noves solucions innovadores que ofereixin avantatges i valor afegit sobre els sistemes tradicionals d'arxiu i emmagatzemament.

Un altre concepte de traducció ambigua el constitueix el “electronic records management” que tant pot significar un sistema de gestió de documents electrònics exclusivament, com un sistema electrònic de gestió de documents d'arxiu (en paper i/o electrònics), segons es posi èmfasi en “electronic records” o en “records management”.

El document d'arxiu, és a dir, aquells documents produïts o rebuts per una persona o una institució durant el curs de la gestió o l'activitat per a l'acompliment dels seus fins, i conservats com a prova o informació; per la qual cosa, qualsevol document d'arxiu, ha d'estar relacionat amb un productor o ha de pertànyer a un procés de producció del document, és a dir, ha de tenir una procedència, i ha de tenir una finalitat. En altres paraules, “els documents d'arxiu són el reflex de les funcions i activitats de l'home, producte i testimoni d'una gestió.”

Un document electrònic és un document que, essent analògic o digital, ha de ser interpretat i mostrat a l'ull humà mitjançant un reproductor. La Norma de descripció arxivística de Catalunya (NODAC) considera que els documents electrònics constitueixen un cas apart, i els defineix com a documents en els que la informació és manipulada, transmesa o processada informàticament i posseeix contingut, context i estructura per testificar l'activitat que l'ha generat.

A partir d'ara en aquesta memòria es considerarà, en general, que s'està parlant d'un sistema electrònic de gestió de documents electrònics, donant per entès que molts d'ells permeten també registrar en el sistema referències a documents en paper que s'hauran d'anar a buscar a un arxiu físic i no només al repositori de documents electrònics.

2.3. Gestió documental

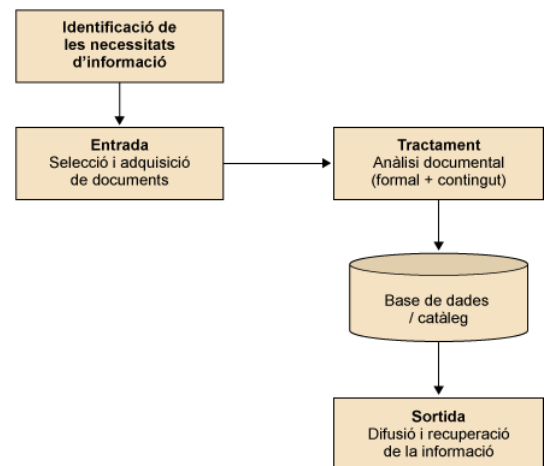
El propòsit de la gestió documental és la creació i el manteniment, durant el temps que sigui necessari, de documents capaços de donar suport a les activitats de l'organització i de provar que aquestes es duen a terme d'acord amb els requisits legals i normatius.

Un sistema de gestió de documents electrònics és, per tant, un sistema computaritzat, un conjunt de programes, emprats per a rastrejar i emmagatzemar documents electrònics i/o imatges digitals de documents originalment suportats en paper. El terme pot ser relacionat amb conceptes com sistemes d'administració de continguts (CMS) i és usualment vist com un sistema d'administració de contingut corporatiu i relacionat amb el terme Digital Assot Management.

Des d'un enfocament pràctic, per a aconseguir una gestió eficaç dels documents, caldria:

- establir polítiques i estratègies adaptades a les necessitats i expectatives de l'organització, tenint clar "per què es gestionen documents" i "on es vol arribar";
- implantar procediments i bones pràctiques de gestió documental, decidint "què s'ha de fer" i "com s'ha de fer";
- definir clarament les responsabilitats, determinant "qui ha de fer què".

La cadena documental o procés documental es defineix com un conjunt d'operacions successives d'entrada, tractament i sortida d'informació i de documents



- I. La primera fase és l'entrada, que comprèn la selecció, l'adquisició i la recepció dels fons.
- II. Respecte al tractament del document, se centra en l'anàlisi documental. El seu objectiu és representar els documents de manera que es puguin recuperar en una base de dades o en un catàleg. L'anàlisi que se'n fa és tant formal com de contingut. L'anàlisi formal descriu els atributs estructurals dels documents, per a representar-lo de manera única, sense ambigüitats, i s'hi estableixen punts d'accés bibliogràfics per autor i títol mitjançant una cerca. L'anàlisi de contingut identifica i representa de manera precisa la matèria a fi de crear punts d'accés per matèries o continguts dels documents per a recuperar-los posteriorment.
- III. L'última fase es refereix a la sortida dels documents. Està orientada a la difusió i recuperació de la informació, mitjançant instruments com els catàlegs locals o col·lectiu, bases de dades, dipòsits digitals institucionals, portals de recursos, o mitjançant serveis de difusió.

2.3.1. Record management

El "**Records Management**" és una disciplina que estudia el tractament adequat en les organitzacions de certs actius d'informació denominats "**Records**". Un Record és un actiu d'informació, digital o en paper, que reflecteix la història o memòria de l'organització respecte les seves activitats. Exemples d'aquests actius són: factures, comandes, documents comptables, etc.

Els actius d'informació es caracteritzen per:

- No haurien de ser alterables un cop creats.
- Existeix algun tipus d'exigència legal o corporativa perquè siguin conservats durant un període mínim de temps.
- Tenen valor per a l'organització.

El Records Management, terme que no té una traducció apropiada al català, ni tan sols un equivalent, està íntimament associat a la gestió documental, tot i que no és tant ampli com aquesta.

2.3.2. Sistemes de Recuperació de la Informació

La recuperació d'informació (RI a partir d'ara) és el procés de selecció d'informació d'un fons documental per mitjans automàtics o semiautomàtics, és a dir, amb l'ús parcial o intensiu d'ordinadors.

L'objectiu dels estudis sobre RI consisteix a trobar la forma de realitzar, de manera automàtica i amb la màxima eficiència possible, el grup de funcions o operacions següent:

- I. Identificar quins són els temes rellevants d'un document.
- II. Determinar i assignar els descriptors més adequats per a representar aquests temes.
- III. Determinar i assignar descriptors a les demandes d'informació formulades inicialment en llenguatge natural.
- IV. Construir llenguatges documentals, com ara tesaurus.
- V. Derivar i produir resums dels documents que, eventualment i en segons quines circumstàncies, puguin substituir la lectura del document complet.
- VI. Determinar quins són els documents més rellevants en relació amb una necessitat d'informació determinada i en quin grau ho són.

Tota la teoria moderna de l'RI es fonamenta en les tres idees nuclears següents:

- a. **La representació de la informació que contenen els diversos documents es realitza mitjançant l'assignació de diversos conjunts de termes d'indexació a cada document, i no tant per assignació dels documents a classes o subclasses d'un quadre de classificació.**
- b. **Les necessitats d'informació dels usuaris d'un sistema documental també es poden representar mitjançant conjunts de termes d'indexació.**
- c. **Els documents més rellevants amb relació a cada necessitat d'informació dels usuaris seran aquells que presentin un grau de semblança més gran respecte a la necessitat d'informació.**

En un sistema d'RI no es poden comparar directament documents i necessitats d'informació. En realitat, el que es compara són representacions de cadascuna de les dues entitats esmentades.

Formalment, per tant, en un sistema d'RI un document és un conjunt D , els elements del qual són termes d'indexació, segons aquest model general:

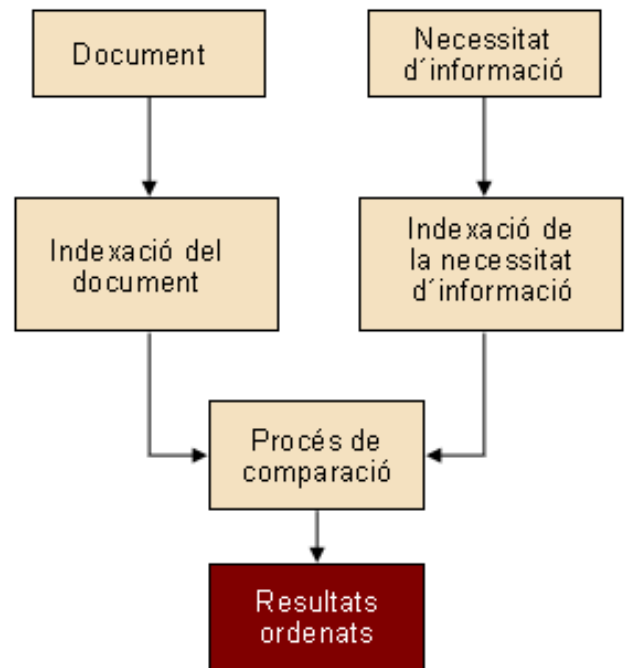
$$D_i = \{t_1, t_2, \dots, t_n\}$$

On t_1, t_2, \dots, t_n són paraules simples (per exemple, "economia") o compostes (per exemple, "economia política") que expressen les propietats semàntiques, és a dir, el contingut temàtic, del document D_i . Per exemple, suposem, per a simplificar, que el document D_i contingui 5 temes rellevants, t_1, t_2, t_3, t_4, t_5 , aleshores la representació de D_i seria la següent:

$$D_i = \{t_1, t_2, t_3, t_4, t_5\}$$

Segons el model general, una necessitat d'informació es representaria de la manera següent:

$$P_j = \{t_1, t_2, \dots, t_n\}$$



2.4. PH Systems

El Grup PH neix el desembre de 1990, després de l'absorció per part de Digital de la divisió d'informàtica de Philips Informàtica i Comunicacions, que va comportar la transformació de les seves delegacions provincials en empreses privades.

PH Systems és una empresa de base tecnològica formada per un equip de més de 30 professionals, tècnics qualificats amb àmplia experiència en el sector de les tecnologies de la informació i amb vocació de servei.

La estratègia de negoci que segueix és la de col·laborar estretament amb els clients per a poder satisfer al màxim les seves necessitats. I oferir valors afegits per arribar a ser un referent en el mercat com a empresa de solucions integrals.

PH Systems ofereix solucions informàtiques que permeten: optimitzar els processos empresarials, millorar la qualitat i l'accés a la informació, i reduir les despeses de gestió.

Es defineix com una empresa integradora de serveis, i compta amb divisions de negoci especialitzades al voltant dels següents serveis i solucions:

- Gestió Empresarial:- Implantació de solucions de gestió de negoci per la petita i mitjana empresa (ERP Lirba desenvolupat per PH Systems).
- Desenvolupament d'Aplicacions:- Serveis de consultoria, desenvolupament i disseny d'aplicacions totalment adaptades a les necessitats dels clients.
- Base de Dades:- Serveis d'auditoria de base de dades, optimització (tuning), migració a noves versions, estudis de polítiques de backup, alta disponibilitat i manteniment.
- Business Intelligence:- Gestió del rendiment corporatiu, reporting, quadres de comandament, Data Warehouse i anàlisi.
- Aprovisionament de Programari:- Aplicacions estàndard de Microsoft, antivirus, sistemes operatius, ...
- Aprovisionament de Maquinari:- Assessorament de solucions d'última generació, personalitzades i adaptades al 100% a les necessitats de les empreses: PC's de sobretaula, estacions de treball, portàtils, servidors, impressores, ...
- Manteniment de Maquinari i Programari:- Degut als constants canvis tecnològics i a vegades a la manca de recursos interns de les empreses, PH Systems ofereix diferents serveis de manteniment. Uns serveis que permeten a les empreses centrar-se en el seu negoci, deixant la informàtica en mans d'informàtics.

La seva activitat proporciona serveis i solucions a diferents sectors:

- Serveis e industrial
- Hoteler
- Distribució
- Càrnic
- Públic

Entre els seus clients destaquen tant empreses com institucions: Casa Darnés, el grup DISBESA, la Diputació de Girona, Transports Badosa, embotits La Selva, Tecalum, MercaGas, etc...

2.5. Utilització de les TIC

L'objecte del present projecte és assolir l'objectiu de millorar la gestió i/o la productivitat dels procediments informatitzats, que tenen com a element inicial en el seu flux de dades, els actius d'informació.

Les tecnologies de la informació tenen, d'una banda, una faceta d'automatització, és a dir, resulten molt eficients quan ens reemplacen en tasques rutinàries, en operacions que es poden reduir a operacions numèriques o lògiques. En aquest sentit, les aplicacions d'automatització han tingut un impacte important en la indústria, resultant en un augment de la productivitat.

D'altra banda, les tecnologies de la informació tenen també una faceta d'augmentació, és a dir, poden multiplicar les nostres capacitats intel·lectuals, ens poden ajudar a pensar, a escriure, a decidir i a actuar millor.

Un cas exemplificador seria: el fet que l'autor utilitzi un ordinador per a escriure aquest text no representa una aplicació augmentadora de les seves capacitats intel·lectuals; ho seria, per exemple, si la màquina fos capaç d'ordenar les idees disperses introduïdes per l'autor i les convertís directament en un text no solament llegible, sinó intel·ligent i tot.

La gran diferència entre les possibilitats automatitzadores i augmentadores de les tecnologies de la informació rau potser en el fet que per a automatitzar un procés n'hi ha prou d'entendre'l bé per a comprendre quines parts hi ha, com s'encadenen, etc., mentre que per a augmentar les nostres capacitats intel·lectuals hem d'entendre com processem la informació i generem coneixements.

3. Requeriments del projecte

Una mancança que té el Lirba, ja que no fa cap gestió dels documents administratius i com a actius d'informació que són, concretament informació formal i estructurada; podem amb les TIC realitzar una automatització i fins hi tot una augmentació del tractament de la informació.

No hem d'oblidar però que PH Systems ofereix solucions a les organitzacions i per tant aquestes organitzacions poden disposar d'altres SI preexistents.

Per tant de manera global s'haurà de procurar:

- Que els components es puguin configurar segons altres SI preexistents.
- Que és disposi de més d'una alternativa d'implantació i de funcionament de les principals funcionalitats.
- Que els components del sistema puguin ser distribuïts i això és pugui determinar en la implantació o en futures actualitzacions.
- Que hi hagi un baix acoplament entre els components per facilitar la creació de paquets i la seva implantació.
- Que les funcionalitats proveïdes per terceres aplicacions es puguin reemplaçar sense afectar el sistema, ni la seva funcionalitat.
- Que hi hagi un alt grau de reutilització.

Pel que fa a la necessitat de la gestió de documents; ja s'ha comentat anteriorment que amb les TIC podem implementar una aplicació que permeti:

- Un arxiu central per a tots els documents
- Tenir la Informació disponible en tot moment des de qualsevol lloc
- Facilitat de recerca i ràpida recuperació dels documents
- Integració amb qualsevol flux de treball
- Màxima seguretat

També s'ha de valorar la normativa de l'Agència Tributaria que permet l'eliminació de factures físiques després de realitzar un procés de digitalització certificada.

Els requisits a complir per part d'una empresa que desitgi implantar aquest sistema son:

- Disposar d'un escàner que permeti digitalitzar els documents en un format estàndard a una resolució mínima de 200ppp i sense pèrdua d'informació en la compressió de la imatge.
- Disposar d'un software que permeti recuperar aquesta imatge, introduir una sèrie de dades obligatòries i realitzar la certificació conforme es idèntic al original en paper.
- Disposar d'una firma electrònica reconeguda per l'AEAT i que el software de certificació pugui introduir en cada una de las imatges certificades.
- Garantir l'accés en línia a l'AEAT per a auditories

Així doncs tindrem tres aspectes fonamentals a tractar:

- Els documents administratius o actius d'informació.
- La adquisició d'aquests documents i la gestió del sistema.
- La integració del Lirba amb un Record Management.

Ara es passaran a detallar els requeriments de cadascun d'aquest components.

3.1. Arxiu Òptic

El Records Management és una disciplina que estudia el tractament adequat en les organitzacions a determinats actius d'informació anomenats Records, terme que no té traducció correcta al català (ni al castellà).

Un record es un actiu d'informació, digital o en paper, que reflecteix la història o memòria de l'organització respecte a les seves activitats. Exemples d'aquests actius són: factures, comandes, documents comptables, actes, correu electrònic, etc.

Aquests actius es caracteritzen per:

- No haurien de ser alterables un cop creats.
- Existeix algun tipus d'exigència legal o corporativa perquè siguin conservats durant un període mínim de temps.
- Tenen valor per a l'organització.

Aquest component per tant s'encarregarà de:

- Un arxiu central per a tots els documents
- Tenir la Informació disponible en tot moment des de qualsevol lloc
- Facilitat de recerca i ràpida recuperació dels documents

A més de:

- Disposar d'un software que permeti recuperar aquesta imatge, introduir una sèrie de dades obligatòries i realitzar la certificació conforme es idèntic al original en paper.
- Disposar d'una firma electrònica reconeguda per l'AEAT i que el software de certificació pugui introduir en cada una de las imatges certificades.

I de forma general:

- Màxima seguretat
- Garantir l'accés en línia a l'AEAT per a auditories

Per tant les parts que el composaran seran:

- Un arxiu central per a tots els documents
- Un sistema de RI
- Un magatzem Virtual

3.1.1. Arxiu central

Aquest component no ha de ser implementat necessàriament sinó que es pot utilitzar altres alternatives que ofereixi el mercat o bé una solució preexistent dins de l'organisme o ens vulgui implantar la solució. Aquesta capa però ha de ser accessible per diverses plataformes.

Per tant és necessària una capa que proporcioni aquests serveis de tal manera que un altre component pugui emprar-la per guardar-hi els documents. Així doncs es pot emprar una gestió documental, un gestor de continguts com el SharePoint o simplement el sistema de fitxers.

Per tant el requisits són:

- Un sistema d'emmagatzemament de documents electrònics ja existent

3.1.2. Sistema de RI

Aquest és el component principal, serà l'encarregat de donar les eines necessàries per poder seleccionar la informació continguda en l'arxiu central. Anteriorment ja s'ha explicat de manera general com funciona el procés de selecció, s'ha parlat de la indexació dels documents i de la indexació de les necessitats d'informació; la intersecció entre els conjunts d'aquestes dues indexacions donarà com a resultat la informació a mostrar.

Així doncs aquest component ha de proporcionar un llenguatge que ens permeti la indexació i la interrogació. Tot i que les tasques d'interrogació es poden delegar a una capa superior.

Una forma de classificar e identificar de forma eficient la informació és l'ús de les metadades, és a dir, informació relativa a les pròpies dades que faciliten la seva catalogació i a més proporcionen informació semàntica associada.

Tenim així que els documents administratius s'indexaran a partir de les seves metadades, com que aquests documents són estructurats caldrà un llenguatge d'esquema per poder descriure i especificar l'estructura d'aquests documents. Així mateix serà necessari un sistema que permeti generar la indexació d'un document coneguda la seva estructura i un altre que obtingui la informació continguda en el document i que s'associarà amb les metadades.

Per tant els requisits són:

- Un sistema de plantilles:- Una plantilla és una forma de dispositiu que sol proporcionar una separació entre la forma o estructura i el contingut. Es un mitjà o un artefacte que permet guiar, portar o construir un disseny o esquema predefinit.
- Un llenguatge d'esquemes:- S'utilitza per a descriure l'estructura i les restriccions dels continguts dels documents.
- Un OCR:- El Reconeixement Òptic de Caràcters (OCR), així com el reconeixement de text, en general són aplicacions dirigides a la digitalització de textos. Identifiquen automàticament símbols o caràcters que pertanyen a un determinat alfabet, a partir d'una imatge per a emmagatzemar-ho en forma de dades amb les que podrem interactuar mitjançant un programa d'edició de text o similar.
- Un sistema que generi les metadades i dades d'un document electrònic
- Accés remot a totes les funcionalitats e independència d'aquestes d'altres solucions informàtiques.

3.1.3. Magatzem Virtual

Dins de la flexibilitat que es pretén donar a aquest projecte, una opció disponible hauria de ser el poder o no eliminar els documents físics. En cas de que no és vulgui destruir el paper opcionalment es pot afegir un mòdul que interactuï amb el Sistema de RI i doni un atribut més al document que sigui la seva ubicació física. Aquest mòdul seguirà un sistema de classificació (ja sigui numèric o alfanumèric) que s'adaptarà al sistema d'arxiu físic de l'organització on s'implanti el projecte.

Per tant el requisits són:

- Un sistema que generi un índex de classificació seguint un arxiu físic.

3.2. Adquisició de documents i administració.

Una de les tasques més feixugues serà la de transformar els documents físics en documents electrònics, així com l'administració de la capa del sistema de RI.

3.2.1. Adquisició de documents

El projecte ha de ser prou flexible per adaptar-se a diferents organitzacions, per tant no s'ha de limitar a un tipus d'escàner concret ni a una forma d'escaneig específica. Podent-se usar aparells que permetin escanejar document a document o multitud d'ells. O també patir d'un recurs que proporcionï les imatges dels documents. Finalment s'ha de tenir en compte que un document administratiu pot estar compost per més d'un full de paper.

Així doncs l'adquisició de documents realitzarà dues tasques:

- I. Obtenció d'una imatge a partir d'un recurs (escàner, sistema de fitxers, etc...).
- II. Generació d'un document electrònic a partir de les diferents imatges que el conformen.

Una de les variables a tenir en compte és el rendiment i la flexibilitat per això les dues tasques anteriors tot i ser clarament diferenciades s'han de realitzar alhora, de forma desatesa sense afectar a les casuístiques de les diferents organitzacions on s'implanti.

Per tal de diferenciar els documents el sistema emprat serà el de codi de barres, per tant cada document físic haurà d'incorporar un codi de barres (si no l'incorpora ja, acords amb proveïdors etc...), aquest codi estarà en una etiqueta adhesiva que es pot generar amb informació del Magatzem Virtual si s'escau.

Per tant el requisits són:

- Un driver twain d'escaneig massiu configurable controlat per una capa superior
- Un driver sobre sistema de fitxers per obtenir imatges controlades per una capa superior
- Una eina de reconeixement de codis de barra
- Un generador de documents

3.2.2. Administració de l'arxiu òptic

Fins ara s'han anat definint diferents mòduls o components, que poden ser independents, opcionals i treballar de forma remota. Tot i així és necessita una aplicació que gestioni el correcte funcionament de l'Arxiu òptic.

Així doncs aquesta aplicació farà:

- La gestió de les plantilles.
- La gestió dels esquemes.
- L'adquisició dels documents i la generació d'etiquetes.
- Visualització dels documents.

De les tasques anteriors l'adquisició de documents ja s'encarrega un altre component de l'aplicació, pel que fa a la resta ara és passen a detallar.

3.2.2.1. Gestió de plantilles

Les plantilles relacionen un tipus de document administratiu amb una forma física concreta d'aquest, o sigui si tenim un tipus de document del tipus factura podem tenir diferents plantilles de factures, que és corresponguin amb formats físics diferents per exemple: la factura d'Endesa, la factura de Telefònica, la factura del proveïdor X, etc... El que fa exactament una plantilla és relacionar una regió física d'un

document amb un element de la seva estructura lògica. Les plantilles es gestionen des del sistema de RI, però cal una interfície d'usuari per poder-les crear i administrar.

Per tant el requisits són:

- Una interfície de creació de plantilles

3.2.2.2. Gestió d'esquemes

Un esquema descriu la estructura de la informació d'un document. L'arxiu òptic pot tenir tants esquemes com tipologies de documents administratius existeixin, i la seva gestió es realitza des del sistema de RI però cal una interfície d'usuari per poder-les crear i administrar.

Per tant el requisits són:

- Una interfície de creació d'esquemes

3.2.2.3. Visualització de documents.

Tot i que la visualització dels documents es pot fer des de l'Arxiu central, el Magatzem Virtual o bé des d'una aplicació que interactuï amb el Sistema RI, com el Lirba. L'Arxiu Òptic ha de ser una aplicació independent.

Per tant el requisits són:

- Una interfície de visualització de documents.

3.3. Integració amb el Lirba

La motivació d'aquest projecte és la de cobrir una mancança en les solucions informàtiques de gestió empresarial que té pels diferents sectors l'empresa PH Systems. Tot i que qualsevol solució ha de poder treballar independentment, al ser el Lirba l'eix central de PH Systems l'Arxiu Òptic s'ha de poder integrar perfectament amb aquest, però també el Lirba no s'ha de limitar a treballar només amb l'Arxiu Òptic sinó que ha de poder utilitzar solucions de tercers de forma integrada.

Per fer aquesta integració caldrà:

- Facilitat de recerca i ràpida recuperació dels documents
- Integració amb qualsevol flux de treball
- Màxima seguretat

En la solució que és proposa l'Arxiu Òptic (o algun altre producte ja existent en el mercat com Matrix) s'encarreguen de la faceta automatitzadora. En canvi la faceta d'augmentació és responsabilitat de l'aplicació de gestió empresarial ja sigui el Lirba o una altre.

A part de la necessitat de poder gestionar documents administratius en format físic, també hi ha la necessitat de poder afegir noves funcionalitats al Lirba de forma remota. Això s'ha de fer per requeriments d'altres projectes en SOAP.

Així doncs les tasques a realitzar en el Lirba són:

- Marc de treball per a Record Management.
- Marc de treball per a factoria de serveis.

Ara es passaran a detallar els requeriments d'aquest dos nous frameworks que s'integraran dins del Lirba.

3.3.1. Marc de treball per a Record Management

Aquest marc de treball ha de dotar al Lirba de la funcionalitat bàsica d'integrar en qualsevol flux de treball un o més documents electrònics. En el Lirba un flux de treball ve definit per una interfície (independentment del nombre de pantalles que necessiti); a més tot flux de treball genera una instància que és persisteix a la BD.

També però ha de permetre la faceta d'augmentació o sigui generar la indexació de la necessitat d'informació per recuperar documents electrònics i amb aquests generar una instància d'un flux de treball.

Així doncs el marc de treball tindrà dos components:

- Mòdul general
- Mòdul particular

Hi haurà un mòdul general únic pel Lirba, de mòduls particulars existiran tants com integracions a diferents solucions de Record Management es facin (actualment Arxiu Òptic i Matrix)

Els requeriments del mòdul general seran:

- Poder assignar tipus de documents a fluxos de treball
- Poder assignar un document electrònic a una instància d'un flux de treball
- Ser independent del Record Management associat.
- Generar una etiqueta identificativa des del Magatzem Virtual.

Els requeriments del mòdul particular de l'Arxiu Òptic seran:

- Poder assignar els elements d'un esquema als elements d'un flux de treball.
- Crear les necessitats d'informació d'una instància d'un flux de treball per buscar un document.
- Generar una instància d'un flux de treball a partir d'un document.
- Modificar les dades d'un document a partir d'una instància.
- Accés remot.

3.3.2. Marc de treball per la factoria de serveis

El Lirba no disposa d'un sistema per a poder accedir de forma remota a serveis que el dotin de noves funcionalitats, entre altres un d'aquests nous serveis serà l'Arxiu Òptic.

Donada l'alta flexibilitat i l'alt grau de configuració de que disposa el Lirba, aquest Marc de treball ha de permetre configurar i carregar nous serveis a partir d'una factoria sent responsable de cada mòdul (aplicació de Lirba) saber ,si és necessari, utilitzar o com funciona aquesta funcionalitat.

Un requisit d'aquest Marc de treball és que ha de treballar amb Soap.

Per tant el requisits són:

- Definició de serveis remots.
- Vinculació de serveis remots amb objectes del Lirba.
- Generació de missatges SOAP

4. Alternatives

Una alternativa que es va considerar abans de realitzar res, va ser la de només implementar la integració del Lirba a una gestió documental o bé a un gestor de continguts. Així doncs tenim:

1. Emprar una gestió documental
2. Emprar un gestor de continguts
3. Emprar un arxiu òptic com Docuware o Matrix
4. Construir un arxiu òptic propi

La primera i segona opció no s'ajusten als requeriments, si bé és cert que una gestió documental engloba el que seria l'arxiu corrent, té l'inconvenient de ser un sistema massa complex. Els gestors de continguts són sistemes per mantenir i actualitzar webs de manera fàcil.

DocuWare o Matrix que tenen l'avantatge de ser sistemes ja fets i contrastats, si ve complien perfectament la funció de gestionar l'arxiu corrent dels documents administratius externs, tenien l'inconvenient que no era tant trivial fer-ho amb els propis (els generats pel propi Lirba).

Construir un sistema propi, a part d'aconseguir un efecte automatitzador, té l'avantatge de que una integració amb el Lirba proporcionaria a part un efecte augmentador. A més de tenir la flexibilitat de dotar a cada implantació només les funcionalitats requerides per aquesta i per tant tenir la complexitat justa segons les necessitats d'implantació.

4.1. Arxiu Òptic

Un cop decidit implementar un Arxiu Òptic, es tenen diferents alternatives de construcció per aquest:

1. Com un mòdul del Lirba
2. Una aplicació independent del Lirba

La primera opció però, tenia l'inconvenient de que, qualsevol instal·lació que es volgués realitzar requeriria la instal·lació del Lirba. I aquesta opció no deslliura de realitzar la feina feixuga de fer la integració de l'Arxiu Òptic amb els diferents mòduls del Lirba. També hi ha el desavantatge, de que el Lirba esta implementat com una aplicació client servidor en VB6 i es volia començar a treballar en entorns SOA (un dels requeriments és que l'arxiu òptic sigui un component remot).

Pel que fa a les metadades es podia haver emprat:

1. Yaml
2. Jason
3. XML

Si ve l'XML és més laboriós de construir, i per contra Yaml i Jason s'integren més bé amb determinats llenguatges de programació; l'avantatge que va decantar cap a XML és que permet la utilització d'aquest en la majoria dels elements principals. Així doncs els esquemes és realitzaran en XML SCHEMA, les plantilles tot i ser un format propi (seran fitxers XML), les vistes seran plantilles XSLT i la informació dels documents estarà en fitxers XML.

En quan a la persistència de la informació, es podia optar per:

1. la utilització del sistema de fitxers
2. bases de dades relacionals
3. XML natives
4. Gestors de continguts
5. Un sistema Mixt

El principal avantatge d'un sistema mixt és la flexibilitat i adaptació a diferents implantacions i a sistemes ja preexistents. Així doncs com que es vol donar de la major flexibilitat possible al sistema, s'ha optat per realitzar un sistema de configuració que permeti a cada implantació decidir segons les necessitats de funcionalitat que es vol emprar:

- Una instal·lació que requereixi generar una indexació de les necessitats d'informació, i per tant una cerca complexa de documents pot emprar una bases de dades nativa en Xml per guardar aquests documents i emprar un llenguatge com Xpath per fer les consultes.
- Si en una implantació l'empresa ja disposa d'un gestor de continguts com SharePoint o Joomla, es pot emprar aquest com a arxiu central.

4.2. Adquisició i administració de documents.

Una de les tasques més feixugues és la de l'adquisició i l'administració de documents així com la de tot el sistema. Les alternatives que es van considerar van ser:

1. Realitzar l'administració de forma remota en un entorn web.
2. Eines d'escaneig en xarxa.
3. Un model centralitzat

En el primer cas tot i que l'Arxiu Òptic es un component remot hi havia la dificultat de realitzar l'adquisició de documents de forma remota. En l'escaneig per xarxa això implicava que els usuaris fessin la pròpia incorporació dels documents. Per a volums reduïts això es factible però per a grans volums és millor realitzar un escaneig de forma massiva.

Es va optar per un model on hi ha una recepció centralitzada dels documents i aquests es classifiquen, s'etiqueten i després s'escanegen de forma massiva. Això també serà òptim per aquells llocs que disposin d'un arxiu. En tot cas, per volums petits també hi ha l'opció d'incorporar els documents des de l'aplicatiu final, en aquest cas enlloc d'una classificació es farà una distribució dels documents a cada lloc de treball.

4.3. Integració amb el Lirba

Hi ha dos aspectes importants pel que fa la integració amb el Lirba, per un cantó tenim que s'ha de dotar al programa la funcionalitat del Record Management; i per altra banda tenim que aquesta funcionalitat es troba en un component remot, i el Lirba no pot interactuar amb components remots.

Per tal d'interactuar amb components remots, un dels requisits és la utilització de Soap, i la creació d'un marc de treball per interactuar amb Serveis Web.

En aquest darrer cas les alternatives eren:

1. Emprar directament la llibreria MSSOAP.
2. Generar missatges SOA com a text pla.
3. Construir una eina basada en MS Soap per la generació de missatges SOA de forma genèrica.

Les dues primeres alternatives tenen l'inconvenient, de que cada cop que es vulgui afegir un nou Servei Web, s'ha d'implementar codi per tal de realitzar la comunicació Soap.

L'alternativa escollida, que és construir una eina que generi els missatges Soa a partir d'un tipus, permet afegir qualsevol Servei Web a l'entorn de programació sense haver d'implementar la generació de missatges.

Si bé aquesta opció és més complexa, i sobrepassa aquest projecte, com que aquest marc de treball s'emprarà en altres projectes, aquests es veuran afavorits per aquesta implementació reduint-ne el cost total.

En quan a dotar de la funcionalitat del Record Management al lirba tenim les alternatives següents:

1. Vincular directament la funcionalitat al Lirba.
2. Realitzar una abstracció permetent la utilització de diferents aplicacions.

En el primer cas es té l'avantatge de conèixer ambdues aplicacions, però té l'inconvenient de que, per cada aplicació de Record Management diferent emprada, s'ha de fer una implementació que en vinculi la seva funcionalitat.

Com que, ja com a mínim existeixen dues aplicacions (la que es construeix, i Matrix) que seran utilitzades per diferents clients. I a més altres clients, que tenen el Lirba o el NewGen (versió anterior al Lirba), disposen de solucions pròpies o existents en el mercat, s'ha optat per assumir el major cost de realitzar l'abstracció però dotar d'una major flexibilitat al Lirba.

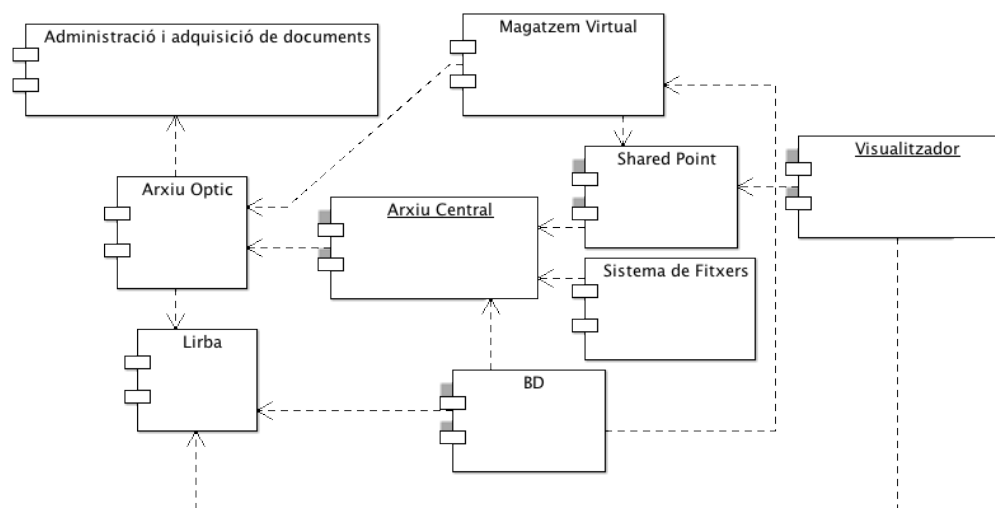
5. Anàlisi del projecte

En aquest capítol analitzarem i detallarem les funcionalitats del sistema des del punt de vista de l'actor que hi interactuarà. Per poder mostrar-ho correctament presentarem un conjunt de diagrames de casos d'ús segons cada mòdul del SI, i les fitxes corresponents per detallar cada funcionalitat.

Per entendre millor el funcionament del SI és necessari també que mostrem els diferents mòduls que componen el projecte (aquests mòduls poden ser entre si actors al interactuar entre ells)

5.1. Model General

El projecte està compost per tres components principals: l'Arxiu Òptic, l'aplicació d'administració i adquisició de documents i el Lirba. Aquests components principals poden interactuar també amb altres components externs.

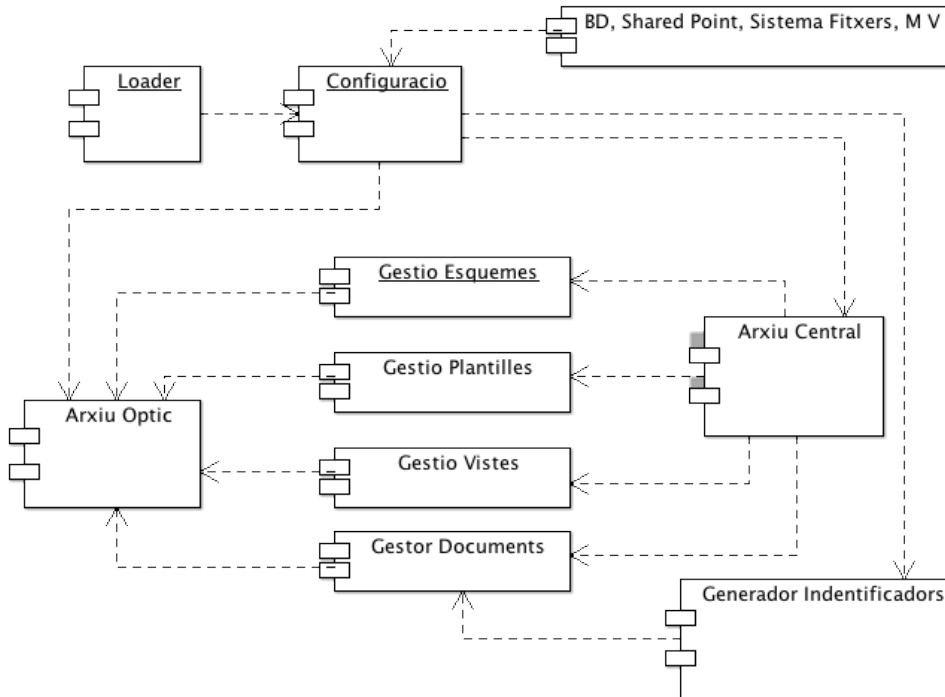


Els diferents components que hi ha són:

- **Arxiu Òptic:**- Aquest component s'encarrega de la gestió dels documents administratius, entre altres tasques té la funció d'indexació i classificació dels documents, així com la de generar la seva metainformació.
- **Adm. i Adq. de documents:**- Aquesta aplicació proporciona una interfície per administrar l'Arxiu Òptic, així com un mètode d'adquisició de documents en format físic
- **Lirba:**- Aquest paquet informàtic s'encarrega de la gestió d'una organització, a més ara integrarà dins dels seus fluxos de treball els documents administratius.
- **BD:**- Aquest mòdul es pot encarregar de la persistència de la informació.
- **Sistema de fitxers:**- Aquest mòdul es pot encarregar de la persistència de la informació.
- **SharedPoint:**- Aquest mòdul es pot encarregar de la persistència de la informació.
- **Magatzem Virtual:**- Aquest mòdul proporciona, en cas de no eliminar el paper físic una ubicació en l'arxiu per cada document electrònic introduït.
- **Mòdul de Visualització:**- La principal tasca de l'Arxiu Òptic és generar un document electrònic que contingui dades i metadades que representin un document físic, si bé també està disponible la representació visual del document físic, aquest mòdul permetrà generar una visualització d'un document a partir de les dades i les metadades.

5.2. Arxiu Òptic

Com s'ha comentat l'Arxiu Òptic s'encarrega de la gestió dels documents administratius, i la seva principal tasca consisteix en generar les dades i la metainformació a partir d'un document electrònic. Així doncs un document dins l'Arxiu Òptic tindrà un identificador, la seva representació electrònica, les seves metadades i les seves dades. L'únic requisit és disposar d'un identificador. L'arxiu Òptic ha de ser un component remot, no és responsabilitat del contenidor d'aquest component la seva inicialització, el propi component s'ha de poder inicialitzar quan se li demana alguna funcionalitat.

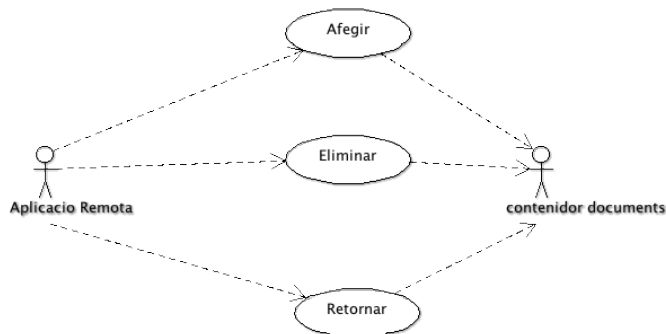


Els diferents components que hi ha són:

- Arxiu Òptic Aquest component remot és la interfície que permet a altres components externs interactuar amb l'Arxiu Òptic.
- Carregador Components:- Aquest component s'encarrega de trobar i habilitar d'altres components quan es vol obtenir alguna funcionalitat.
- Configuració:- S'encarrega de gestionar la configuració del R.I.
- Arxiu Central:- És un magatzem extern on es guarden tots els documents electrònics.
- Gestió d'Esquemes:- Aquest mòdul s'encarrega de la gestió dels esquemes.
- Gestió de Plantilles:- Aquest mòdul s'encarrega de la gestió de les plantilles
- Gestió de Vistes:- Aquest mòdul s'encarrega de la gestió de les vistes.
- Gestió de Documents:- Aquest mòdul s'encarrega de la gestió dels Documents.
- Generador d'identificadors:- Aquest mòdul s'encarrega de proporcionar un identificador únic o ubicació pel document.

5.2.1. Gestió d'Esquemes

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de gestió d'esquemes.



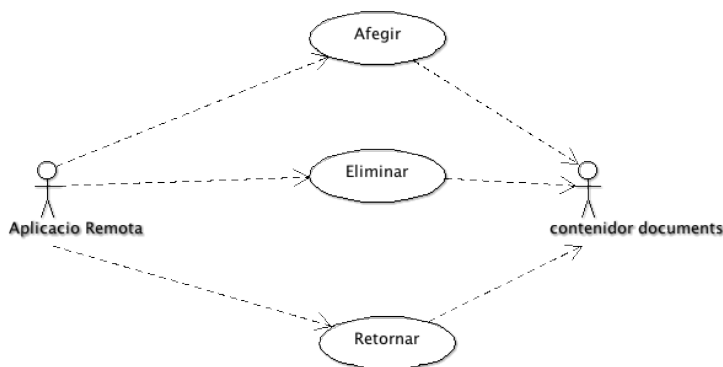
CAS D'ÚS:	Afegir Esquema
Descripció	Afegeix un esquema dins de l'arxiu central
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	1 Es comprova que s'ha inicialitzat 2 Es comprova que és un esquema vàlid 3 S'afegeix a l'arxiu central
Subfluxos	Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució. Si al validar les dades el sistema detecta que no és un esquema vàlid genera una excepció però permet continuar l'execució.
Fluxos Alternatius	
Postcondició	Hi ha un nou esquema disponible.
Requeriments no funcionals	
Comentaris	Les excepcions es poden recuperar a partir d'una petició de l'actor al sistema.

CAS D'ÚS:	Eliminar Esquema
Descripció	Elimina un esquema dins de l'arxiu central
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix i no té cap dependència 3 S'elimina de l'arxiu central
Subfluxos	Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució. Si no existeix l'esquema o aquest té dependències es genera una excepció però permet continuar l'execució.
Fluxos Alternatius	
Postcondició	S'ha eliminat un esquema de l'arxiu central
Requeriments no funcionals	
Comentaris	Les excepcions es poden recuperar a partir d'una petició de l'actor al sistema.

CAS D'ÚS:	Retornar Esquema
Descripció	Retorna un esquema o un element d'aquest (un URI)
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es retorna un esquema
Subfluxos	<p>Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.</p> <p>Si al validar les dades el sistema detecta que no existeix l'esquema o l'element demanat genera una excepció però permet continuar l'execució.</p>
Fluxos Alternatius	<p>Retorna elements</p> <ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es comprova que l'element existeix dins de l'esquema 4 Es retornen els subelements d'aquest element <p>Retorna Estructura</p> <ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es comprova que l'element existeix dins de l'esquema 4 Es retorna l'estructura d'aquest element.
Postcondició	
Requeriments no funcionals	
Comentaris	Sempre es retorna un objecte serialitzat, que donat el cas pot estar buit.

5.2.2. Gestió de Plantilles

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de gestió de plantilles.



CAS D'ÚS:	Afegir Plantilla
Descripció	Afegeix una plantilla dins de l'arxiu central
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que existeix l'esquema 3 Es comprova que és una plantilla vàlida 4 Es crea una relació entre la plantilla i l'esquema 5 S'afegeix a l'arxiu central
Subfluxos	<p>Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.</p> <p>Si al validar les dades el sistema detecta que no és una plantilla vàlida o que no existeix l'esquema genera una excepció però permet continuar l'execució.</p>
Fluxos Alternatius	
Postcondició	Hi ha un nova plantilla d'un determinat esquema disponible.
Requeriments no funcionals	
Comentaris	Les excepcions es poden recuperar a partir d'una petició de l'actor al sistema.

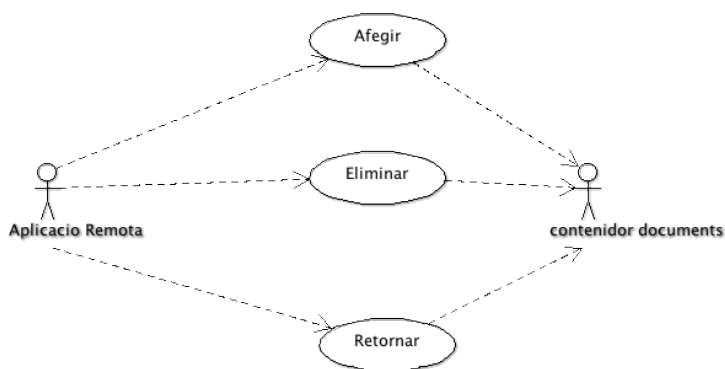
CAS D'ÚS:	Eliminar Plantilla
Descripció	Elimina una Plantilla dins de l'arxiu central
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que la plantilla existeix i no té cap dependència 3 S'elimina la relació amb l'esquema 4 S'elimina de l'arxiu central
Subfluxos	<p>Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.</p> <p>Si no existeix la plantilla o aquesta té dependències es genera una excepció però permet continuar l'execució.</p>
Fluxos Alternatius	

Postcondició	S'ha eliminat una plantilla d'un esquema de l'arxiu central i la seva relació amb l'esquema.
Requeriments no funcionals	
Comentaris	Les excepcions es poden recuperar a partir d'una petició de l'actor al sistema.

CAS D'ÚS:	Retornar Plantilla
Descripció	Retorna una o més plantilles o un element d'aquesta (una Area)
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	1 Es comprova que s'ha inicialitzat 2 Es comprova que la plantilla existeix 3 Es retorna una plantilla
Subfluxos	Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució. Si al validar les dades el sistema detecta que no existeix la plantilla o l'element demanat genera una excepció però permet continuar l'execució.
Fluxos Alternatius	Retorna Plantilles 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es retorna els identificadors de les plantilles d'un esquema donat Retorna Àrees 1 Es comprova que s'ha inicialitzat 2 Es comprova que la plantilla existeix 4 Es retornen les àrees d'aquesta plantilla
Postcondició	
Requeriments no funcionals	
Comentaris	Sempre es retorna un objecte serialitzat, que donat el cas pot estar buit. Una plantilla té com atribut l'esquema associat Una àrea té associat un URI de l'esquema i una regió del document físic

5.2.3. Gestió de Vistes

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de gestió de vistes.



CAS D'ÚS:	Afegir Vista
Descripció	Afegeix una vista dins de l'arxiu central
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	1 Es comprova que s'ha inicialitzat 2 Es comprova que existeix l'esquema 3 Es comprova que és una vista vàlida 4 Es crea una relació entre la vista i l'esquema 5 S'afegeix a l'arxiu central
Subfluxos	Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució. Si al validar les dades el sistema detecta que no és una vista vàlida o que no existeix l'esquema genera una excepció però permet continuar l'execució.
Fluxos Alternatius	
Postcondició	Hi ha un nova Vista d'un determinat esquema disponible.
Requeriments no funcionals	
Comentaris	Les excepcions es poden recuperar a partir d'una petició de l'actor al sistema.

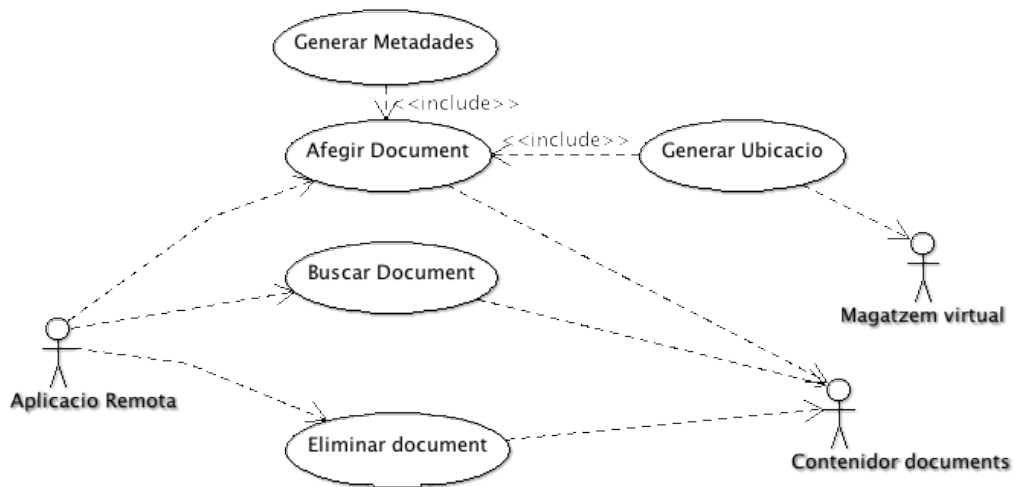
CAS D'ÚS:	Eliminar Vista
Descripció	Elimina una Vista dins de l'arxiu central
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	1 Es comprova que s'ha inicialitzat 2 Es comprova que la vista existeix 3 S'elimina la relació amb l'esquema 4 S'elimina de l'arxiu central
Subfluxos	Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució. Si no existeix la vista o aquesta té dependències es genera una excepció però permet continuar l'execució.
Fluxos Alternatius	

Postcondició	S'ha eliminat una vista d'un esquema de l'arxiu central i la seva relació amb aquest esquema.
Requeriments no funcionals	
Comentaris	Les excepcions es poden recuperar a partir d'una petició de l'actor al sistema.

CAS D'ÚS:	Retorna Vista (les vistes són plantilles de visualització)
Descripció	Retorna una o més vistes
Actors	Aplicació Remota
Precondició	Existeix una configuració correcta
Flux Principal	1 Es comprova que s'ha inicialitzat 2 Es comprova que la vista existeix 3 Es retorna una vista
Subfluxos	Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució. Si al validar les dades el sistema detecta que no existeix la vista genera una excepció però permet continuar l'execució.
Fluxos Alternatius	Retorna Vistes 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es retorna els identificadors de les vistes d'un esquema donat
Requeriments no funcionals	És una tasca de l'actor la generació d'una vista a partir de la plantilla de vista retornada i les dades i metadades d'un document que compleixi un esquema donat.
Comentaris	Sempre es retorna un objecte serialitzat, que donat el cas pot estar buit. Una plantilla té com atribut l'esquema associat Una àrea té associat un URI de l'esquema i una regió del document físic

5.2.4. Gestió de Documents

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de gestió de documents.



CAS D'ÚS:	Afegir Document
Descripció	Genera un document electrònic amb la seva metainformació
Actors	Aplicació Remota
Precondició	
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es comprova que la plantilla existeix 4 Si no té un identificador el genera 5 Si no té les metadades les genera 6 S'afegeix un nou document al arxiu central
Subfluxos	<p>Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.</p> <p>Si al validar les dades el sistema detecta que no existeix l'esquema o la plantilla genera una excepció però permet continuar l'execució.</p> <p>Si és necessari es signa digitalment el document resultant.</p> <p>En cas de que no existeixi el document a l'arxiu central es crea una relació amb la plantilla.</p>
Fluxos Alternatius	<p>Ja existeix l'identificador</p> <ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es comprova que la plantilla existeix 4 Si hi ha un document a l'arxiu central es reemplaça <p>Ja té les metadades</p> <ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es comprova que la plantilla existeix 4 Si hi ha un document a l'arxiu central es reemplaça
Postcondició	S'ha afegit o modificat un document dins l'arxiu central
Requeriments no funcionals	No cal generar un document amb tota la informació això dependrà de l'origen d'aquest document, i el tractament que en vulgui fer l'aplicació client.
Comentaris	Un document pot constar dels següents arxius electrònics:

	<p>-Arxiu amb el resultat del OCR -Arxiu signat de la imatge física -Arxiu amb el resultat del OCR estructurat segons l'esquema i amb les metadades.</p>
--	--

CAS D'ÚS:	Eliminar Document
Descripció	Esborra un document dins de l'arxiu central.
Actors	Aplicació Remota
Precondició	
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que existeixi l'identificador. 3 S'elimina la relació amb la plantilla. 4 S'elimina el document.
Subfluxos	<p>Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.</p> <p>Si al validar les dades el sistema detecta que no existeix l'esquema, la plantilla o l'identificador de document genera una excepció però permet continuar l'execució.</p>
Fluxos Alternatius	
Postcondició	S'ha eliminat un document.
Requeriments no funcionals	Aquesta capa no està pensada per substituir el paper físic, tot i que proporciona eines per fer-ho. La integritat de l'arxiu central ho ha de fer una capa superior a l'arxiu òptic.
Comentaris	

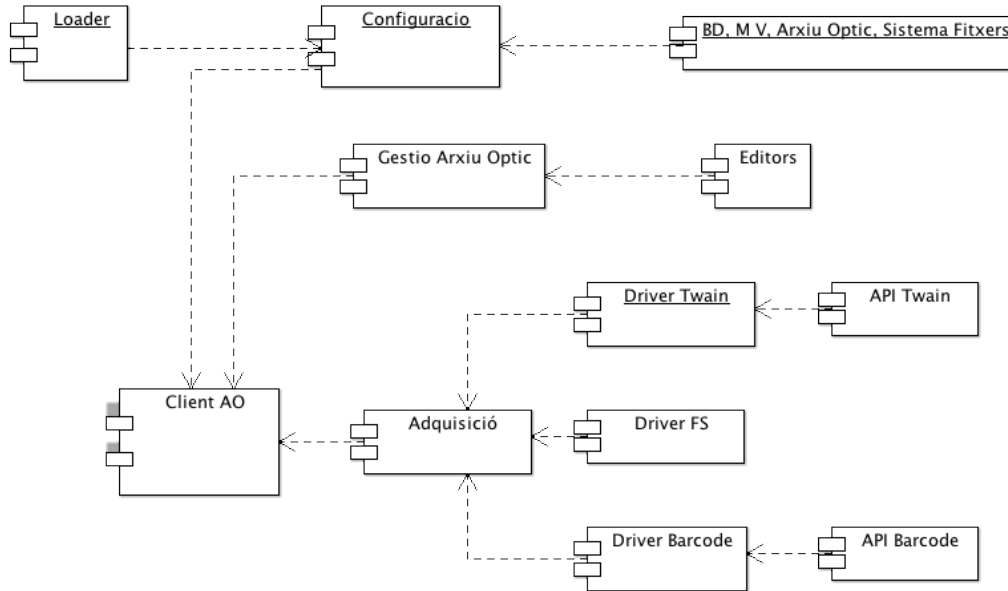
CAS D'ÚS:	Buscar document
Descripció	Retorna un o més documents
Actors	Aplicació Remota
Precondició	L'arxiu central ha de permetre un llenguatge de recerca optimitzat.
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que existeixi l'identificador. 3 Es retorna
Subfluxos	Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.
Fluxos Alternatius	<p>Via relació.</p> <ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Si s'ha passat un esquema o una plantilla es comprova la seva existència. 3 Es retorna una llista d'identificadors <p>Via Consulta.</p> <ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es passa la consulta a l'arxiu central 3 Es construeix una llista d'identificadors amb el resultat 4 Es retorna una llista d'identificadors
Postcondició	Es Retorna un document final o un document serialitzat que conté una llista d'identificadors.
Requeriments no funcionals	És la capa de l'arxiu central qui proporciona les eines de recerca, aquesta capa s'encarrega de la indexació generant la metainformació.
Comentaris	<p>Un document pot constar dels següents arxius electrònics:</p> <p>-Arxiu amb el resultat del OCR -Arxiu signat de la imatge física -Arxiu amb el resultat del OCR estructurat segons l'esquema i amb les metadades.</p>

CAS D'ÚS:	Generar Metadades
Descripció	Genera un document amb metainformació a partir d'una imatge
Actors	Aplicació Remota
Precondició	Es disposa d'un OCR
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que l'esquema existeix 3 Es comprova que la plantilla existeix 3 Es recorre jeràrquicament els elements de l'esquema. 4 Per cada element es genera la metadada associada si procedeix 5 Per l'element tractat es busca a la plantilla si hi ha una àrea definida 6 Per l'àrea definida es processa l'ORC i es genera la dada
Subfluxos	<p>Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.</p> <p>Si al validar les dades el sistema detecta que no existeix l'esquema o la plantilla genera una excepció però permet continuar l'execució.</p>
Fluxos Alternatius	
Postcondició	S'ha generat un document amb la metainformació i les dades
Requeriments no funcionals	
Comentaris	Sempre es genera metainformació tot i que pot no contenir res

CAS D'ÚS:	Generar identificador
Descripció	Generar un identificador únic pel document
Actors	Aplicació Remota
Precondició	
Flux Principal	<ol style="list-style-type: none"> 1 Es comprova que s'ha inicialitzat 2 Es comprova que existeix l'esquema 3 Es comprova que existeix la plantilla 4 Es demana al magatzem virtual que retorni un identificador 5 Es retorna l'identificador.
Subfluxos	<p>Si no s'ha inicialitzat es carrega la configuració a partir d'un recurs del contenidor i continua l'execució.</p> <p>Si al validar les dades el sistema detecta que no existeix l'esquema o la plantilla genera una excepció però permet continuar l'execució.</p>
Fluxos Alternatius	
Postcondició	
Requeriments no funcionals	No necessàriament s'ha de generar un identificador des de l'arxiu òptic per afegir un document, es pot generar un identificador des de tercers aplicacions que interactuïn amb el Magatzem Virtual.
Comentaris	

5.3. Administració i Adquisició de documents

Aquest mòdul com ja s'ha explicat serà l'encarregat d'administrar l'Arxiu Òptic i fer l'adquisició massiva de documents.

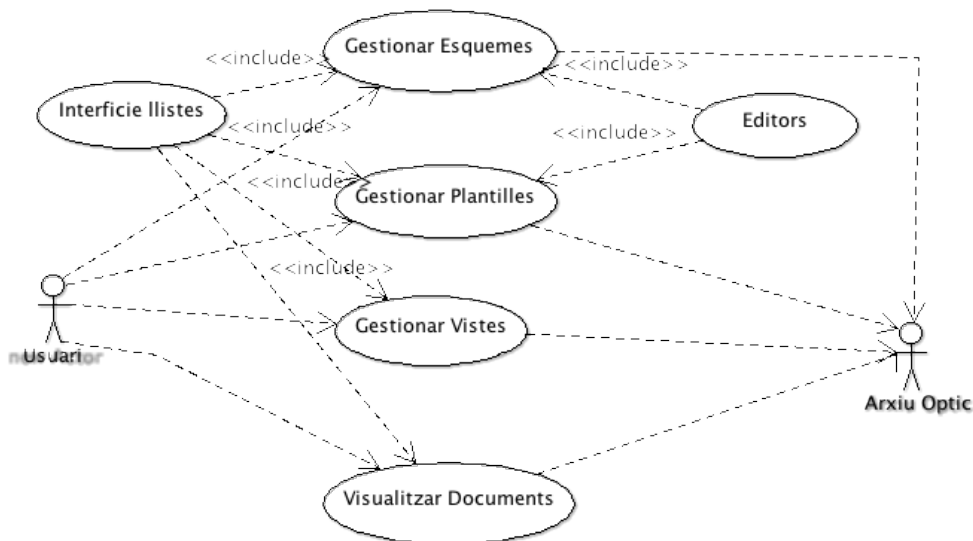


Els diferents components que hi ha són:

- Client AO:- Aquesta mòdul és l'aplicació que dóna accés a l'usuari a la funcionalitat del AO.
- Carregador Components:- Aquest component s'encarrega de trobar i habilitar d'altres components quan es vol obtenir alguna funcionalitat ja sigui remota o local.
- Configuració:- S'encarrega de gestionar la configuració.
- Gestió AO:- Interfícies d'usuari genèriques per fer la gestió d'elements dins l'AO
- Editors:- Editors específics de cada tipus d'element disponible en l'AO (que sigui editable).
- Adquisició:- Generador de documents electrònics a partir d'un recurs (driver) d'adquisició.
- Driver Barcode:- Mòdul encarregat de diferenciar diferents documents a partir d'un recurs d'adquisició, utilitzant codis de barra.
- Api Barcode:- Mòdul extern de reconeixement de codis de barres.
- Driver Twain:- Mòdul que genera un recurs d'adquisició a partir de l'API Twain.
- Driver FS:- Mòdul que genera un recurs d'adquisició a partir del sistema de Fitxers.

5.3.1. Gestió Arxiu Òptic

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de gestió d'arxiu Òptic.



CAS D'ÚS:	Gestionar Esquemes
Descripció	Introduir, modificar, crear o eliminar esquemes en l'Arxiu Òptic
Actors	Usuari
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	<ol style="list-style-type: none"> 1 El sistema retorna una llista amb els esquemes disponibles. 2 Aquesta llista es passa a l'interfície de llistes per ser mostrada. 3 El sistema passa a la interfície de llistes les accions disponibles així com l'enllaç (event) a cada acció per poder-se executar. 4 L'usuari escull una acció a realitzar en la interfície de llistes i proporciona les dades necessàries.
Subfluxos	<p>En cas de modificació o creació.</p> <ol style="list-style-type: none"> 1 El sistema busca l'editor d'esquemes i el presenta a l'usuari. 2 Es carrega l'esquema a modificar o un esquema en blanc. 3 El sistema estableix l'enllaç (event) a l'inclusió d'un esquema. <p>En cas d'introduir esquema existent en fitxer.</p> <ol style="list-style-type: none"> 1 El sistema proporciona un diàleg per buscar un fitxer i el presenta a l'usuari. 2 El sistema estableix l'enllaç (event) a l'inclusió d'un esquema.
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte, el SI mostra un avís indicant els camps erronis i no permet aplicar els canvis.
Postcondició	S'ha afegit, modificat o eliminat un esquema
Requeriments no funcionals	L'aplicació client només dona una interfície d'usuari per poder fer accions a l'arxiu òptic, a més afegeix alguna funcionalitat per simplificar i poder reutilitzar
Comentaris	

CAS D'ÚS:	Gestionar Plantilles
Descripció	Introduir, modificar, crear o eliminar plantilles en l'Arxiu Òptic
Actors	Usuari
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	<ol style="list-style-type: none"> 1 El sistema retorna una llista amb les plantilles disponibles. 2 Aquesta llista es passa la l'interfície de llistes per ser mostrada. 3 El sistema passa a la interfície de llistes les accions disponibles així com l'enllaç (event) a cada acció per poder-se executar. 4 L'usuari escull una acció a realitzar en la interfície de llistes i proporciona les dades necessàries.
Subfluxos	<p>En cas de modificació o creació.</p> <ol style="list-style-type: none"> 1 El sistema busca l'editor de plantilles i el presenta a l'usuari. 2 Es carrega la plantilla a modificar o una plantilla en blanc. 3 El sistema estableix l'enllaç (event) a l'inclusió d'una plantilla.
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte, el SI mostra un avís indicant els camps erronis i no permet aplicar els canvis.
Postcondició	S'ha afegit, modificat o eliminat una plantilla.
Requeriments no funcionals	L'aplicació client només dona una interfície d'usuari per poder fer accions a l'arxiu òptic, a més afegeix alguna funcionalitat per simplificar i poder reutilitzar.
Comentaris	

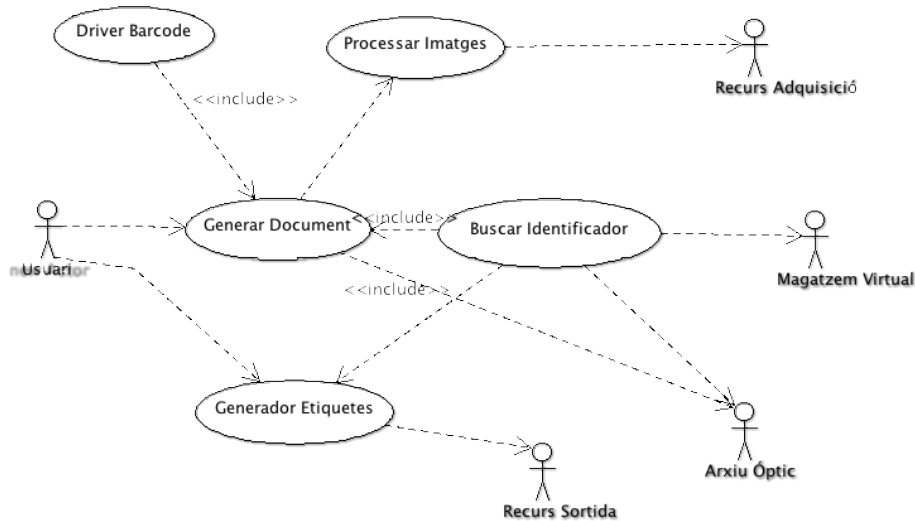
CAS D'ÚS:	Gestionar Vistes
Descripció	Introduir, eliminar esquemes en l'Arxiu Òptic
Actors	Usuari
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	<ol style="list-style-type: none"> 1 El sistema retorna una llista amb les vistes disponibles. 2 Aquesta llista es passa la l'interfície de llistes per ser mostrada. 3 El sistema passa a la interfície de llistes les accions disponibles així com l'enllaç (event) a cada acció per poder-se executar. 4 L'usuari escull una acció a realitzar en la interfície de llistes i proporciona les dades necessàries.
Subfluxos	<p>En cas d'introduir una vista existent en fitxer.</p> <ol style="list-style-type: none"> 1 El sistema proporciona un diàleg per buscar un fitxer i el presenta a l'usuari. 2 El sistema estableix l'enllaç (event) a l'inclusió d'una vista.
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha afegit o eliminat una vista.
Requeriments no funcionals	L'aplicació client només dona una interfície d'usuari per poder fer accions a l'arxiu òptic, a més afegeix alguna funcionalitat per simplificar i poder reutilitzar
Comentaris	Per poder modificar una vista s'ha d'eliminar i tornar a afegir-la.

CAS D'ÚS:	Visualitzar documents
Descripció	Mostrar o eliminar documents en l'Arxiu Òptic
Actors	Usuari
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	<p>1 El sistema retorna una llista amb els documents disponibles, segons la selecció realitzada.</p> <p>2 Aquesta llista es passa a la interfície de llistes per ser mostrada.</p> <p>3 El sistema passa a la interfície de llistes les accions disponibles així com l'enllaç (event) a cada acció per poder-se executar.</p> <p>4 L'usuari escull una acció a realitzar en la interfície de llistes i proporciona les dades necessàries.</p>
Subfluxos	<p>Mostrar documents.</p> <p>1 El sistema retorna una llista de les vistes disponibles segons plantilla associada.</p> <p>2 A més també proporciona la vista de metadades o document físic.</p>
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha mostrat o eliminat un document.
Requeriments no funcionals	L'aplicació client només dona una interfície d'usuari per poder fer accions a l'arxiu òptic, a més afegeix alguna funcionalitat per simplificar i poder reutilitzar
Comentaris	<p>Es pot demanar la visualització del document original (es mostrarà en pdf o conjunt d'imatges)</p> <p>També es pot mostrar un txt amb el resultat de l'ORC.</p> <p>Finalment es pot mostrar les metadades i la informació associada a aquestes metadades en una vista escollida.</p>

CAS D'ÚS:	Interfície de llistes
Descripció	Funcionalitat bàsica de mostrar elements en una llista
Actors	Usuari
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	<p>1 L'usuari demana una llista d'elements d'un tipus segons una selecció</p> <p>2 El mòdul de gestió corresponent vincula les accions disponibles sobre llistes a la gestió corresponent.</p> <p>3 L'usuari interactua sobre les accions.</p>
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha generat una interfície amb una llista d'elements.
Requeriments no funcionals	
Comentaris	<p>Les accions usals sobre llistes son:</p> <ul style="list-style-type: none"> Afegir (des d'arxiu). Eliminar Seleccionar i mostrar accions disponibles. Crear (des d'editor) Visualitzar

5.3.2. Adquisició de documents

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul d'adquisició de documents.



CAS D'ÚS:	Generar Document
Descripció	Afegeix un o més documents dins de l'arxiu òptic
Actors	Usuari
Precondició	Hi ha accés remot a l'Arxiu Òptic i al magatzem Virtual
Flux Principal	1 Demana imatges al processador d'imatges 2 Rep una imatge del processador d'imatges 3 Passa l'imatge al Barcode 4 Determina si hi ha un codi de barres vàlid 5 Amb totes les imatges anteriors al darrer codi de barres vàlid crea una copia electrònica del document físic. 6 Afegeix a l'Arxiu Òptic el nou document
Subfluxos	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Fluxos Alternatius	Si es vol es pot introduir només un document (sense codi de barres) en aquest cas : 1 Demana imatges al processador d'imatges 2 Rep una imatge del processador d'imatges 3 Amb totes les imatges crea una copia electrònica del document físic. 4 Busca un identificador vàlid 5 Afegeix a l'Arxiu Òptic el nou document
Postcondició	S'ha introduït un o més documents a l'arxiu òptic
Requeriments no funcionals	
Comentaris	

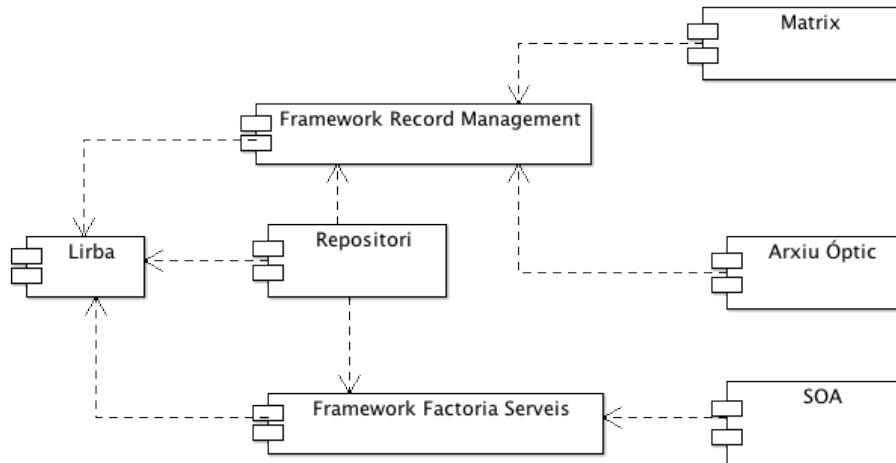
CAS D'ÚS:	Generador Etiquetes
Descripció	Genera un codi de barres amb un identificador vàlid per l'Arxiu Òptic i una ubicació física (si és el cas)
Actors	Usuari
Precondició	Hi ha accés remot a l'Arxiu Òptic i al magatzem Virtual
Flux Principal	1 L'usuari estableix per quina plantilla vol generar etiquetes i la quantitat que en vol. 2 L'usuari determina el recurs de sortida (per defecte impressora d'etiquetes) 3 El sistema redirigeix els identificadors generats cap al recurs de sortida
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'han generat les etiquetes demanades en el recurs de sortida.
Requeriments no funcionals	
Comentaris	

CAS D'ÚS:	Driver Barcode
Descripció	Encapsulació d'un reconeixedor de codi de barres a partir d'una imatge
Actors	
Precondició	
Flux Principal	1 El driver rep una imatge 2 El driver crea una iteració per recórrer els codis de barres de l'imatge 3 El driver proporciona un codi de barres de forma iterativa 4 El driver diu que ja no hi ha més imatges.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'han enviat a la capa superior els codis de barres detectats.
Requeriments no funcionals	Aquesta capa encapsula (en forma d'iteració) el funcionament d'una llibreria de mercat existent.
Comentaris	

CAS D'ÚS:	Processar imatges
Descripció	Retorna de forma iterativa les imatges a partir d'un recurs d'entrada
Actors	
Precondició	Disposar d'un recurs d'entrada
Flux Principal	1 El sistema rep una petició sobre un recurs 2 Es carrega el manegador del recurs 3 Es crea una iteració sobre el manegador 4 Es retorna una imatge 5 Es finalitza la iteració quan no hi ha més imatges a retornar
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Es retornen una o més imatges d'un document físic
Requeriments no funcionals	
Comentaris	

5.4. Integració amb el Lirba

El Lirba és el conjunt d'aplicacions proporcionats per PH Systems que conformen la solució integrada de la gestió d'una empresa. Aquest projecte vol afegir a aquesta gestió la gestió dels documents administratius, en aquest apartat es fa l'anàlisi de com fer la integració a un Record Management, tenint en compte que s'ha de poder utilitzar un d'existent en el mercat apart del propi (Arxiu Òptic).

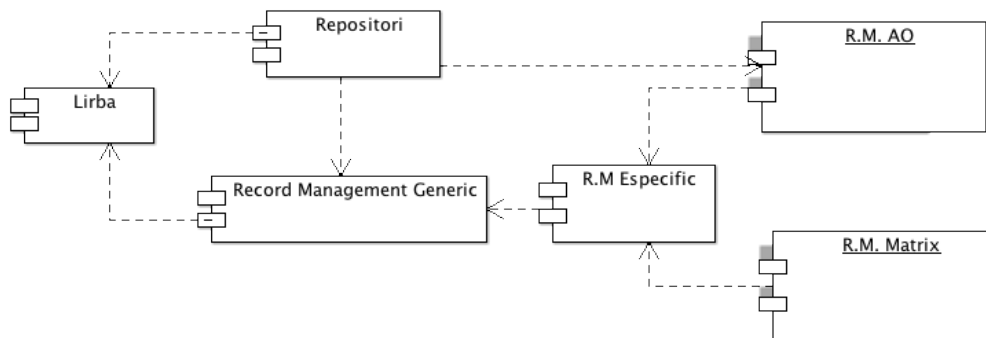


Els diferents components que hi ha són:

- Nucli Lirba:- Conjunt de mòduls que proporcionen la funcionalitat comuna a totes les aplicacions (entre altres conté el Framework d'accés a dades propietari)
- Framework Record Management:- Mòdul d'integració amb la gestió de documents administratius.
- Framework Factoria Serveis:- Mòdul que permet integrar funcionalitats de forma remota al Lirba.
- Repositori:- Mòdul que gestiona la configuració i les metadades dels fluxos de treball disponibles en el Lirba.

5.4.1. Record Management

El framework de Record Management és l'encarregat de fer l'integració del Lirba amb la gestió de documents administratius. Com que es pot fer emprar més d'un gestor, aquest framework divideix la seva funcionalitat en una part comuna que és independent del gestor utilitzat i una altre específica de cada gestor.

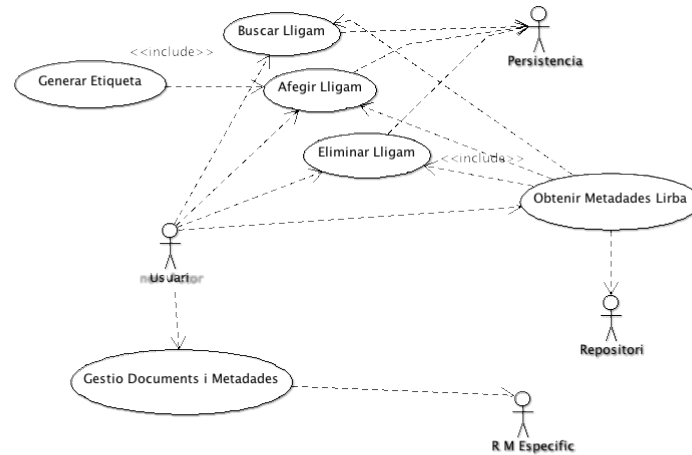


Els diferents components que hi ha són:

- Nucli Lirba:- Conjunt de mòduls que proporcionen la funcionalitat comuna a totes les aplicacions (entre altres conté el Framework d'accés a dades propietari)
- Record Management Genèric:- Mòdul (factoria) d'integració amb la gestió de documents administratius independent del gestor.
- Record Management Específic:- Funcionalitats comunes depenents del gestor.
- Record Management AO:- Mòdul específic amb les funcionalitats del AO.
- Record Management Matrix:- Mòdul específic amb les funcionalitats de Matrix.
- Repositori:- Mòdul que gestiona la configuració i les metadades dels fluxos de treball disponibles en el Lirba.

5.4.1.1. Record Management Genèric

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul Record Management Genèric.



CAS D'ÚS:	Buscar Lligam
Descripció	Retorna un identificador de document o bé un identificador d'un flux de treball
Actors	Usuari
Precondició	
Flux Principal	1 Rep un identificador de document 2 La capa de persistència retorna una estructura de valors (nom, valor) 3 Es retorna aquesta estructura
Subfluxos	
Fluxos Alternatius	1 Rep una estructura de valors (nom, valor) 2 Es busca l'identificador parella d'aquesta estructura 3 Es retorna l'identificador
Postcondició	Retorna un identificador
Requeriments no funcionals	Un identificador de flux de treball és un identificador compost.
Comentaris	Un lligam té tres valors: dos identificadors i la definició de l'estructura del flux de treball.

CAS D'ÚS:	Afegir lligam
Descripció	Afegeix un nou lligam entre un document i un flux de treball
Actors	Usuari
Precondició	
Flux Principal	1 Rep l'estructura del flux amb els seus valors i l'identificador de flux 2 Rep l'identificador de document 3 Recupera les metadades del flux 4 Genera una estructura que identifica el flux a partir de l'estructura amb les dades del flux i l'estructura amb les metadades. 3 Comprova que no existeixi el lligam 4 Genera en la capa persistent un lligam dels dos elements anteriors

	5 Genera una etiqueta d'aquest lligam si ho indiquen les metadades.
Subfluxos	
Fluxos Alternatius	Si es genera una etiqueta vol dir que es farà l'escaneig posterior a la incorporació del document per tant el document no existeix en l'arxiu òptic (es fa una reserva al generar un identificador). Si no es fa vol dir que s'està vinculant un document ja escanejat i per tant ja existent en l'arxiu òptic
Postcondició	S'ha generat un lligam
Requeriments no funcionals	
Comentaris	Aquesta capa permet a cada flux de treball, depenent de la informació definida pel tipus de document Lirba associat a la metainformació treballar de forma indistinta d'una manera o una altre.

CAS D'ÚS:	Eliminar un lligam
Descripció	Elimina un lligam entre un document i un flux de treball
Actors	Usuari
Precondició	
Flux Principal	1 Rep l'estructura de l'identificador de flux amb els seus valors 2 Rep l'identificador de document 3 Comprova que existeixi el lligam 4 Elimina en la capa persistent el lligam dels dos elements anteriors
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha eliminat un lligam
Requeriments no funcionals	
Comentaris	

CAS D'ÚS:	Obtenir metadades Lirba
Descripció	Obté la informació d'estructura d'un flux de treball
Actors	Usuari
Precondició	
Flux Principal	1 Es rep l'identificador d'un flux de treball 2 Es genera una estructura (nom, valor) que pot identificar una instància de qualsevol flux d'aquest tipus.
Subfluxos	
Fluxos Alternatius	1 Es rep una plantilla o un tipus de document Lirba 2 Es determina quin identificador de flux té associat 3 Es genera una estructura (nom, valor) que pot identificar una instància de qualsevol flux d'aquest tipus.
Postcondició	S'ha generat una estructura que identifica el flux de treball
Requeriments no funcionals	Aquesta estructura correspon als elements clau del flux
Comentaris	

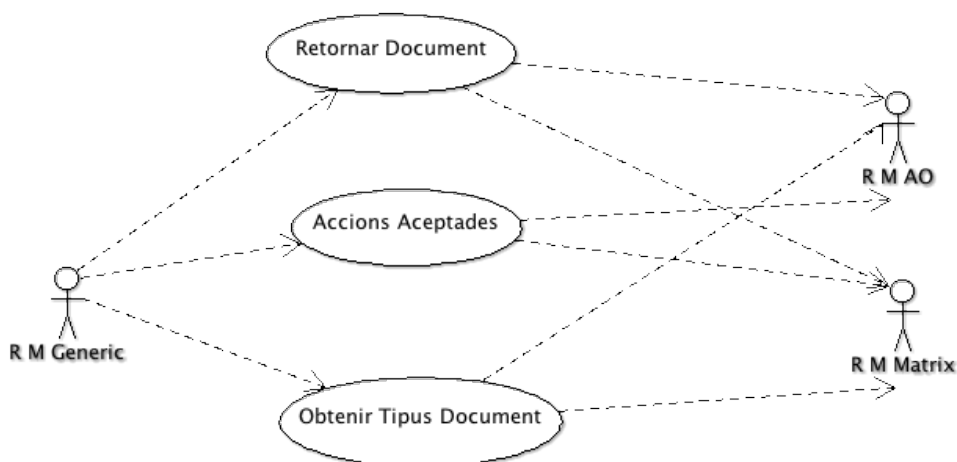
CAS D'ÚS:	Generar Etiqueta
Descripció	Genera una etiqueta el codi de barres
Actors	
Precondició	
Flux Principal	1 Rep una estructura amb les dades del flux. 2 Rep una estructura amb la metainformació de l'etiqueta 3 Rep l'identificador de document 4 Genera una estructura amb els valors de l'etiqueta segons metadades 5 Genera una etiqueta segons format determinat en les metadades.
Subfluxos	

Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha generat una etiqueta.
Requeriments no funcionals	
Comentaris	

CAS D'ÚS:	Gestió de documents i metadades
Descripció	Gestiona les metadades segons el Record Management associat.
Actors	Usuari
Precondició	Tenir definit quin record management s'està emprant
Flux Principal	1 Retorna una llista de tipus de document Lirba 2 Per cada tipus document es poden associar diferents esquemes 3 S'estableix com es genera l'identificador de flux i la etiqueta (si cal)
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha establert l'identificador de cada flux de treball segons un tipus de document de Lirba.
Requeriments no funcionals	Cada record management particular pot a més tenir la seva pròpia configuració
Comentaris	Els tipus de document, fan referència a un tipus de document administratiu i una funcionalitat concreta dins del Lirba, els esquemes només fan referència a un tipus de document administratiu.

5.4.1.2. Record Management Específic

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul Record Management Específic.



CAS D'ÚS:	Accions Aceptades
Descripció	Informa de les accions disponibles per un record management concret.
Actors	RM Genèric
Precondició	
Flux Principal	1 Rep una petició sobre una acció determinada 2 Determina si el RM associat pot realitzar aquella acció
Subfluxos	
Fluxos Alternatius	1 Rep una petició d'accions disponibles. 2 Retorna una llista de les accions disponibles
Postcondició	S'ha establert si una acció es pot portar a terme amb un RM determinat.
Requeriments no funcionals	
Comentaris	

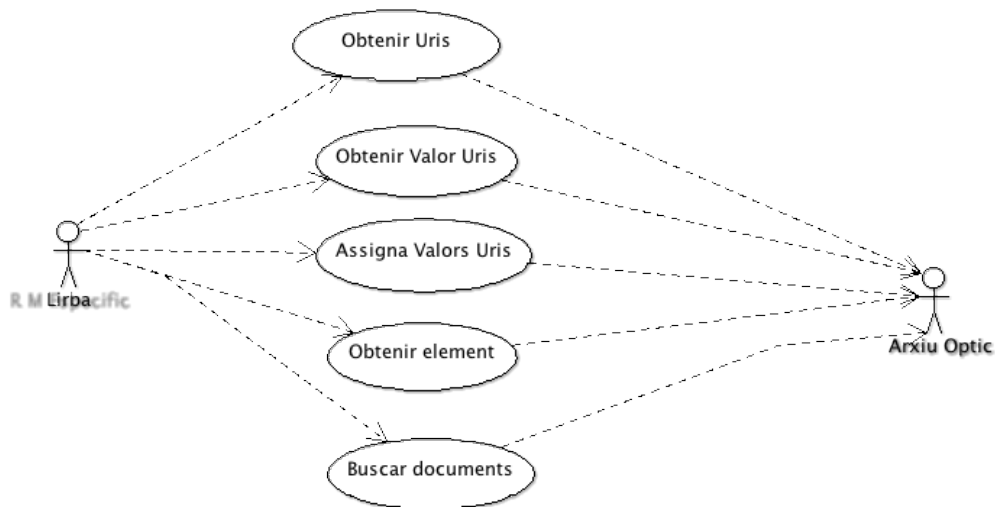
CAS D'ÚS:	Retornar Document
Descripció	Retorna un document electrònic
Actors	RM Genèric
Precondició	
Flux Principal	1 Es rep un identificador de document 2 Es fa una petició al controlador de RM 3 Es crea una estructura amb la informació retornada
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Es retorna una estructura de document electrònic en format Lirba.
Requeriments no funcionals	
Comentaris	

CAS D'ÚS:	Retornar Tipus document
Descripció	Llista els formats o tipus de documents disponibles
Actors	RM Genèric
Precondició	
Flux Principal	1 Es rep una petició d'informar quins tipus de document hi ha

	2 Es retorna una llista amb els tipus de document existents
Subfluxos	
Fluxos Alternatius	
Postcondició	Es determina quins tipus de documents hi ha disponibles per un determinat RM
Requeriments no funcionals	
Comentaris	

5.4.1.3. Record Management Arxiu Òptic

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul Record Management AO.



CAS D'ÚS:	Obtenir Uris
Descripció	Retorna metainformació sobre els esquemes
Actors	Lirba
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	1 Es rep una petició de metainformació d'un esquema 2 Es fa una petició al component remot 3 Es retorna una llista d'elements
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	El Lirba coneix l'estructura d'un esquema
Requeriments no funcionals	Com que els esquemes tenen una estructura en forma d'arbre els elements retornats tenen l'informació de la seva ubicació dins l'estructura.
Comentaris	

CAS D'ÚS:	Obtenir Valors Uris
Descripció	Retorna la informació d'una metadada d'un document concret
Actors	Lirba
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	1 Es rep una petició sobre un document en concret 2 En la petició hi ha una estructura d'elements (nom, valor) on el nom correspon a un úric. 3 Es fa una crida amb aquesta estructura al component remot 4 S'obté com a resposta la mateixa estructura però amb el camp valor informat.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Es sap el valor d'un document per una variable concreta
Requeriments no funcionals	
Comentaris	

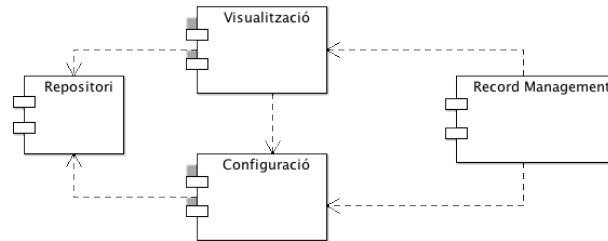
CAS D'ÚS:	Assigna Valors Uris
Descripció	Estableix la informació d'una metadada d'un document concret
Actors	Lirba
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	1 Es rep una petició sobre un document en concret 2 En la petició hi ha una estructura d'elements (nom, valor) on el nom correspon a un uri, i el valor a la informació de la metadada. 3 Es fa una crida amb aquesta estructura al component remot
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha modificat la informació semàntica d'un document
Requeriments no funcionals	
Comentaris	D'aquesta manera la informació semàntica es pot generar o bé a través del OCR a l'introduir un document o bé es pot anar refinant al passar per diversos fluxos de treball, cada flux afegiria una informació concreta. Tot això sense modificar la imatge o document electrònic original (només la part semàntica, dades i metainformació).

CAS D'ÚS:	Obtenir element
Descripció	Obté qualsevol tipus d'arxiu que hi ha en l'Arxiu òptic
Actors	Lirba
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	1 Es rep una petició d'un element (sabent el seu identificador) 2 Es determina quin tipus d'element s'està demanant 3 Es fa una petició al component remot 4 Es retorna el resultat de la petició remota
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha obtingut un element de l'arxiu òptic
Requeriments no funcionals	En els elements diferents a un document l'identificador és el nom
Comentaris	Els tipus d'element són: esquemes, plantilles, vistes i documents.

CAS D'ÚS:	Buscar documents
Descripció	Busca documents a l'arxiu òptic
Actors	Lirba
Precondició	Hi ha accés remot a l'Arxiu Òptic
Flux Principal	1 Es rep una petició de documents en un determinat estat i d'un tipus de document concret de Lirba. 2 Es tradueix aquesta petició a una petició que accepti el component remot 3 Es transforma el resultat de la crida remota en una llista d'identificadors de documents.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'obté una llista d'identificadors de documents
Requeriments no funcionals	
Comentaris	

5.4.2. Repositori.

El repositori és un marc de treball que permet a les aplicacions del Lirba obtenir informació sobre el propi Lirba, des d'informació de configuració fins als fluxos de treball disponibles o el comportament de cada interfície d'usuari. També proporciona funcionalitats específiques modificant i creant interfícies o bé el seu comportament, per tant per poder integrar correctament l'Arxiu Òptic al Lirba s'ha modificat aquest marc de treball.

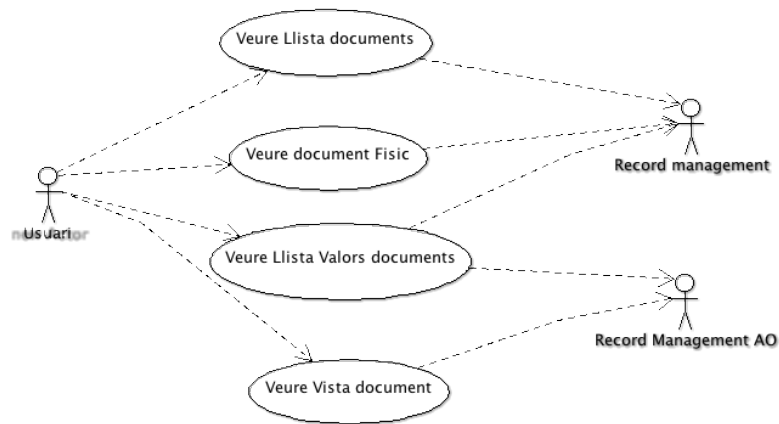


Els diferents components que hi ha són:

- **Visualització:**- Aquest mòdul és l'encarregat de donar la funcionalitat a nivell d'usuari a tot el Lirba.
- **Configuració:**- Aquest mòdul gestiona la part de repositori que vincula les metadades del Record management i els fluxos de treball del Lirba.

5.4.2.1. Visualització

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de Visualització.



CAS D'ÚS:	Veure Llista documents
Descripció	Interfície que ensenya una llista de documents
Actors	Usuari
Precondició	
Flux Principal	1 L'usuari fa una petició per visualitzar documents segons un estat i tipus de document Lirba. 2 Es demana al controlador genèric una llista de documents 3 Es mostra l'identificador de cada document seleccionat
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'obté una interfície amb una llista d'identificadors de documents
Requeriments no funcionals	No hi ha problema en no poder veure més dades, ja que al ser documents pendents de processar (estat) s'han de fer tots.
Comentaris	Aquestes funcions es criden des d'un flux de treball, per tant es sap el tipus de document Lirba i el tipus document associat al Record Management.

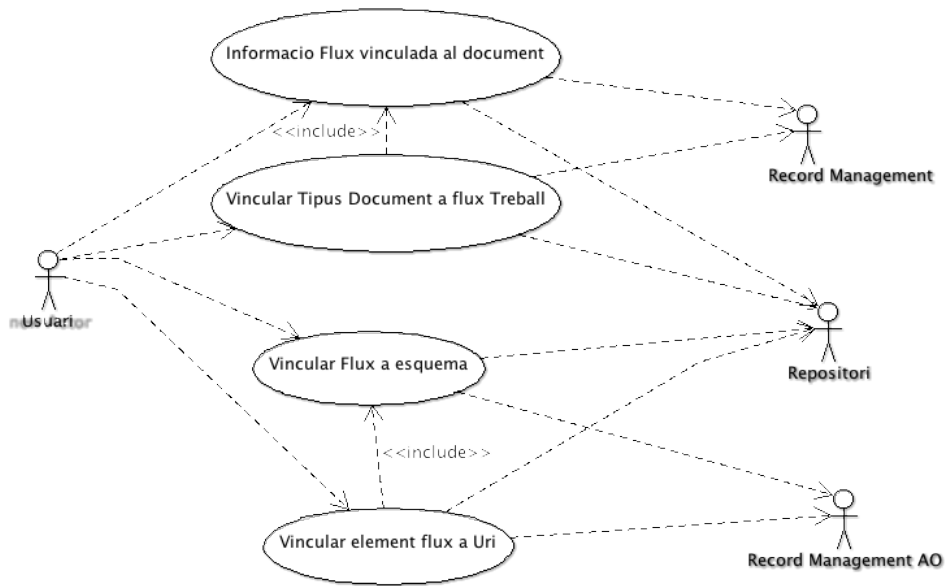
CAS D'ÚS:	Veure document físic
Descripció	Visualitza un document físic en el seu format electrònic
Actors	Usuari
Precondició	
Flux Principal	1 L'usuari es troba en una instància d'un flux de treball 2 Demana la visualització del document 3 Es mostra el PDF o imatge del document.
Subfluxos	
Fluxos Alternatius	També es pot demanar des de les llistes de documents.
Postcondició	Visualització d'un document
Requeriments no funcionals	No hi ha perquè haver document físic, documents interns generats per alguna de les aplicacions.
Comentaris	En aquest darrer cas o bé s'ha generat un pdf o bé es pot generar una visualització fen una crida al controlador específic

CAS D'ÚS:	Veure llista valors documents
Descripció	Es mostra un grid amb la informació d'un conjunt de documents
Actors	Usuari
Precondició	Tenir Arxiu Òptic
Flux Principal	<ol style="list-style-type: none"> 1 L'usuari fa una petició per visualitzar documents segons un estat i tipus de document Lirba. 2 Es demana al controlador específic una llista de documents. 3 Es determina segons els flux i tipus document Lirba quines metadades mostrar 3 Es demana per cada document la informació de les metadades a mostrar 4 Es crea un graella amb les dades obtingudes.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Visualització de les dades dels documents de forma semàntica.
Requeriments no funcionals	No ha de perquè haver-hi informació tot i tenir els documents incorporats.
Comentaris	

CAS D'ÚS:	Veure Vista document
Descripció	Es mostra la metainformació d'un document a partir d'una vista
Actors	Usuari
Precondició	Tenir Arxiu Òptic
Flux Principal	<ol style="list-style-type: none"> 1 L'usuari determina quina vista vol veure sobre un document 2 Es demana el document (les dades amb metadades) al component remot. 3 Es demana la (plantilla) vista al component remot 4 Es genera una visualització del document.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Es visualitza un document amb una vista determinada
Requeriments no funcionals	La generació de la vista no és un requeriment
Comentaris	Si la vista és un XSLT i les dades un XML es pot visualitzar com a pàgina web dins d'un browser encastrat en un formulari.

5.4.2.2. Configuració

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de Configuració.



CAS D'ÚS:	Vincular Tipus document a flux de treball
Descripció	Estableix una relació entre els documents administratius disponibles en els fluxos de treball del Lirba i els documents administratius definits en el Record Management instal·lat (ja sigui Arxiu Òptic, Matrix o un altre)
Actors	Usuari
Precondició	
Flux Principal	1 L'usuari selecciona un document Lirba 2 El Lirba proporciona via Record Management genèric els documents administratius establerts. 3 L'usuari en selecciona un i queden relacionats.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Es crea una relació entre Lirba i el Record management
Requeriments no funcionals	
Comentaris	

CAS D'ÚS:	Informació flux vinculada al document
Descripció	Defineix metainformació del document Lirba d'un flux de treball per un document del Record Management
Actors	Usuari
Precondició	
Flux Principal	1 Es mostren per un tipus de document Lirba tots els elements definits 2 L'usuari pot afegir o treure elements que pertanyin al flux 3 Aquests elements es poden marcar com a identificador i/o com a generadors d'etiqueta.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha definit la metainformació d'un document Lirba d'un flux.
Requeriments no funcionals	
Comentaris	Amb aquesta informació es fan els vincles i es creen les etiquetes

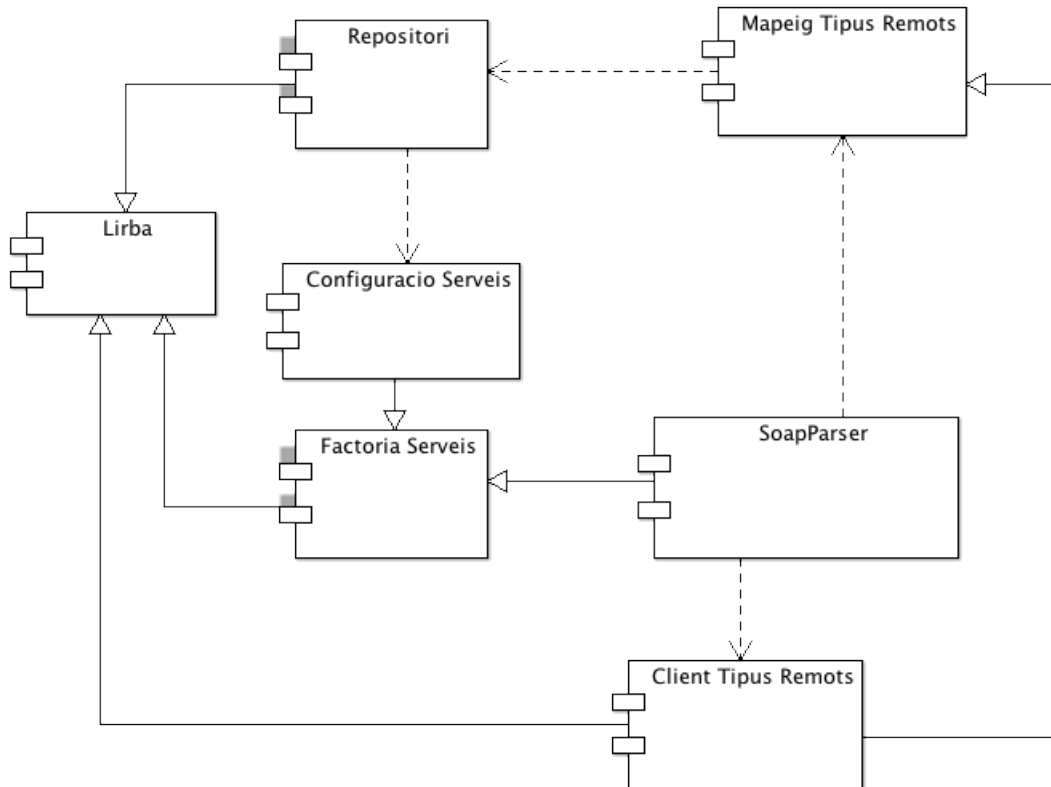
CAS D'ÚS:	Vincular Flux a Esquema
Descripció	En el cas de l'Arxiu òptic es pot vincular metainformació del Lirba amb Metainformació de l'Arxiu Òptic
Actors	Usuari
Precondició	
Flux Principal	1 L'usuari selecciona un document Lirba 2 El Lirba proporciona via Arxiu Òptic els documents administratius establerts. 3 L'usuari en selecciona un i queden relacionats.
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Es crea una relació entre Lirba i un esquema
Requeriments no funcionals	
Comentaris	

CAS D'ÚS:	Vincular element flux a Uri
Descripció	Defineix metainformació del document Lirba d'un flux de treball per un document del Arxiu Òptic
Actors	
Precondició	
Flux Principal	1 Es mostren per un tipus de document Lirba tots els elements definits 2 L'usuari pot afegir o treure elements que pertanyin al flux 3 Aquests elements es lliguen amb uris de l'esquema corresponent
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	Es crea una relació entre un element d'un flux Lirba i un Uri d'un esquema
Requeriments no funcionals	Aquest sistema permet fer traduccions flux a dades o dades a flux
Comentaris	O sigui es pot fer que una instància d'un flux informi el contingut d'un document enllaçat amb informació estructurada o bé a l'inversa es pot instanciar un flux amb les dades d'un document.

5.4.3. Factoria de serveis

La factoria de serveis és un marc de treball que permet a les aplicacions del Lirba obtenir serveis que siguin remots. Aquest mòdul és necessari per altres implantacions que s'han de fer però com que l'arxiu òptic és un component remot s'ha afegit a aquest projecte la creació d'aquest marc de treball.

L'objectiu d'aquest mòdul és desacoplar la creació de clients de serveis remots, de l'ús d'aquest servei en el propi Lirba. Així doncs el nucli del Lirba a partir de la configuració del repositori i d'aquest marc de treball s'encarregarà de carregar els serveis remots necessaris. Cada aplicació del Lirba només estarà acoplada a la interfície client del tipus remot, però no a com s'obté aquesta funcionalitat.

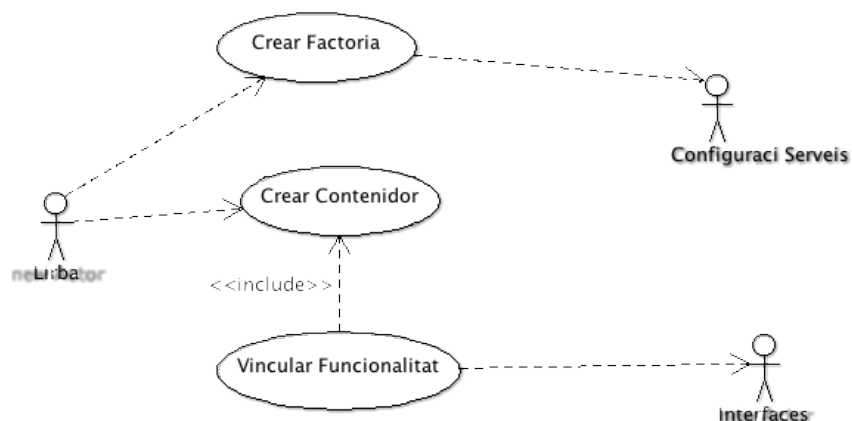


Els diferents components que hi ha són:

- **Repositori**:- Mòdul que gestiona la configuració i les metadades dels fluxos de treball disponibles en el Lirba.
- **Configuració Serveis**:- Aquest mòdul configura la factoria segons els serveis definits al repositori.
- **Factoria Serveis**:- Aquest mòdul dona la funcionalitat de serveis web al sistema
- **SoapParser**:- Aquest mòdul s'encarrega de generar els missatges SOAP
- **Client Tipus Remots**:- Aquest mòdul proporciona les interfases dels serveis remots
- **Mapeig Tipus Remots**:- Aquest mòdul proporciona la serialització

5.4.3.1. Factoria de serveis

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul de Factoria de serveis.



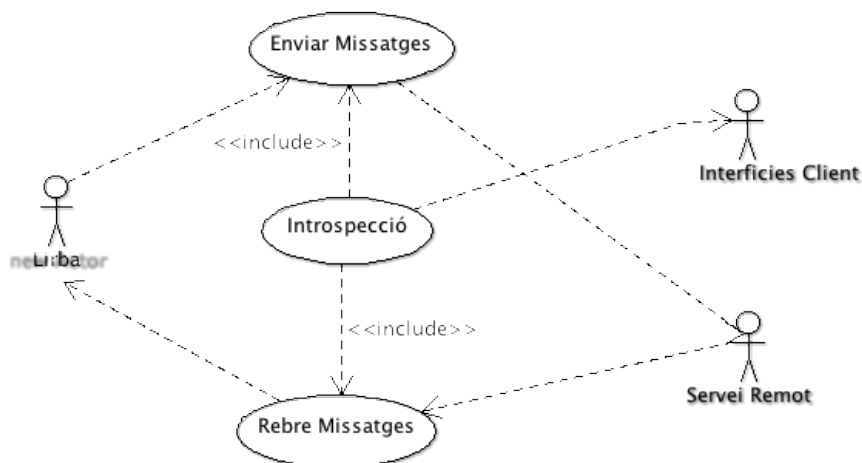
CAS D'ÚS:	Crear Factoria
Descripció	Per tal de desacoplar la creació de Serveis Web, primer s'ha de crear una factoria que està vinculada a una configuració.
Actors	Lirba
Precondició	
Flux Principal	1 Es demana al repositori la configuració 2 Es carrega la configuració en una estructura
Subfluxos	
Fluxos Alternatius	Si a l'enviar les dades, alguna d'aquestes no és correcte mostra un avís.
Postcondició	S'ha creat una factoria de Serveis Web
Requeriments no funcionals	
Comentaris	

CAS D'ÚS:	Crear contenidor
Descripció	Un contenidor és l'objecte que té la informació de la funcionalitat del servei Web
Actors	Lirba
Precondició	S'ha creat una factoria
Flux Principal	1 Es recorre tota l'estructura del contenidor 2 S'estableix per cada element de l'estructura una inicialització SOAP 3 Si l'element és un objecte es vincula la funcionalitat
Subfluxos	
Fluxos Alternatius	
Postcondició	S'ha creat el contenidor d'un servei Web
Requeriments no funcionals	
Comentaris	En aquest moments ja es disposa d'un objecte remot, via servei Web, l'objecte en si és buit, però es mostra com una interface definida en l'estructura del contenidor. Si es fa una crida aquest objecte en realitat s'està fent una crida al SoapParser.

CAS D'ÚS:	Vincular funcionalitat
Descripció	Per poder realitzar serialitzacions complexes cal uns objectes de mapeig
Actors	
Precondició	
Flux Principal	<p>1 Es determina quin objecte de serialització és</p> <p>2 Es determina quin objecte de mapeig té associat</p> <p>3 Es determina si l'objecte de mapeig implementa la interfície de mapeig.</p> <p>4 S'associa l'objecte de mapeig a l'objecte de serialització</p>
Subfluxos	Si l'objecte carregat no existeix o no implementa la interface de mapeig és produeix un error.
Fluxos Alternatius	
Postcondició	L'objecte soap sap serialitzar un objecte complex.
Requeriments no funcionals	
Comentaris	

5.4.3.2. SoapParser

A continuació mostrem el diagrama de casos d'ús i les fitxes pel mòdul SoapParser.



CAS D'ÚS:	Enviar Missatges
Descripció	Construeix un missatge SOAP a partir d'una crida a un objecte
Actors	Lirba
Precondició	Existeix un contenidor
Flux Principal	1 Es busca la introspecció de la crida realitzada 2 Es busca la informació vinculada a aquella introspecció 3 Es realitza la serialització dels paràmetres. 4 Si els paràmetres són complexos es crida en forma de pila recursiva els mapejadors associats. 5 Es crea un observador que processarà la resposta de manera asíncrona.
Subfluxos	
Fluxos Alternatius	
Postcondició	S'ha generat un missatge Soap que permet fer una crida remota
Requeriments no funcionals	
Comentaris	Cada cop que es fa una crida a l'objecte vuit (Servei Web) aquesta crida és redirecciona cap a l'enviament de missatges.

CAS D'ÚS:	Introspecció
Descripció	Donat un objecte determina la seva metainformació
Actors	Lirba
Precondició	
Flux Principal	1 Es determinen el membres públics de l'objecte 2 Es determinen les crides públiques de l'objecte 3 Per cada crida es determina quins objectes s'han de passar com a paràmetres i el seu tipus
Subfluxos	
Fluxos Alternatius	
Postcondició	S'ha descobert la metainformació d'un objecte
Requeriments no funcionals	El seu tipus pot ser: entrada, sortida, o ambdós

Comentaris	En determinats llenguatges aquesta funcionalitat la proporciona el mateix llenguatge
------------	--

CAS D'ÚS:	Rebre missatges
Descripció	Donat un missatge soap es genera una resposta
Actors	Servei Remot
Precondició	S'ha creat un observador
Flux Principal	<ol style="list-style-type: none"> 1 Es busca la introspecció de la crida realitzada 2 Es busca la informació vinculada a aquella introspecció 3 Es realitza la deserialització dels paràmetres. 4 Si els paràmetres són complexos es crida en forma de pila recursiva els mapejadors associats.
Subfluxos	
Fluxos Alternatius	
Postcondició	S'ha generat una resposta a un missatge SOAP
Requeriments no funcionals	Tot i que les crides remotes són asíncrones, les crides en el programa són síncrones
Comentaris	

6. Disseny

Un cop fet l'anàlisi on s'han mostrat els mòduls que compondran aquest projecte, en aquest capítol estudiarem com ho hem de fer.

Així doncs, definirem la forma com col·laboren els objectes per aconseguir els requeriments definits. Per aconseguir-ho farem servir els diagrames de seqüència que mostren el flux de missatges entre objectes, i el diagrama de classes d'aquests. A mesura que es defineixin aquests objectes, apareixeran situacions en les que serà necessari recórrer als patrons de disseny, per això també he afegit un apartat on s'expliquen els patrons utilitzats.

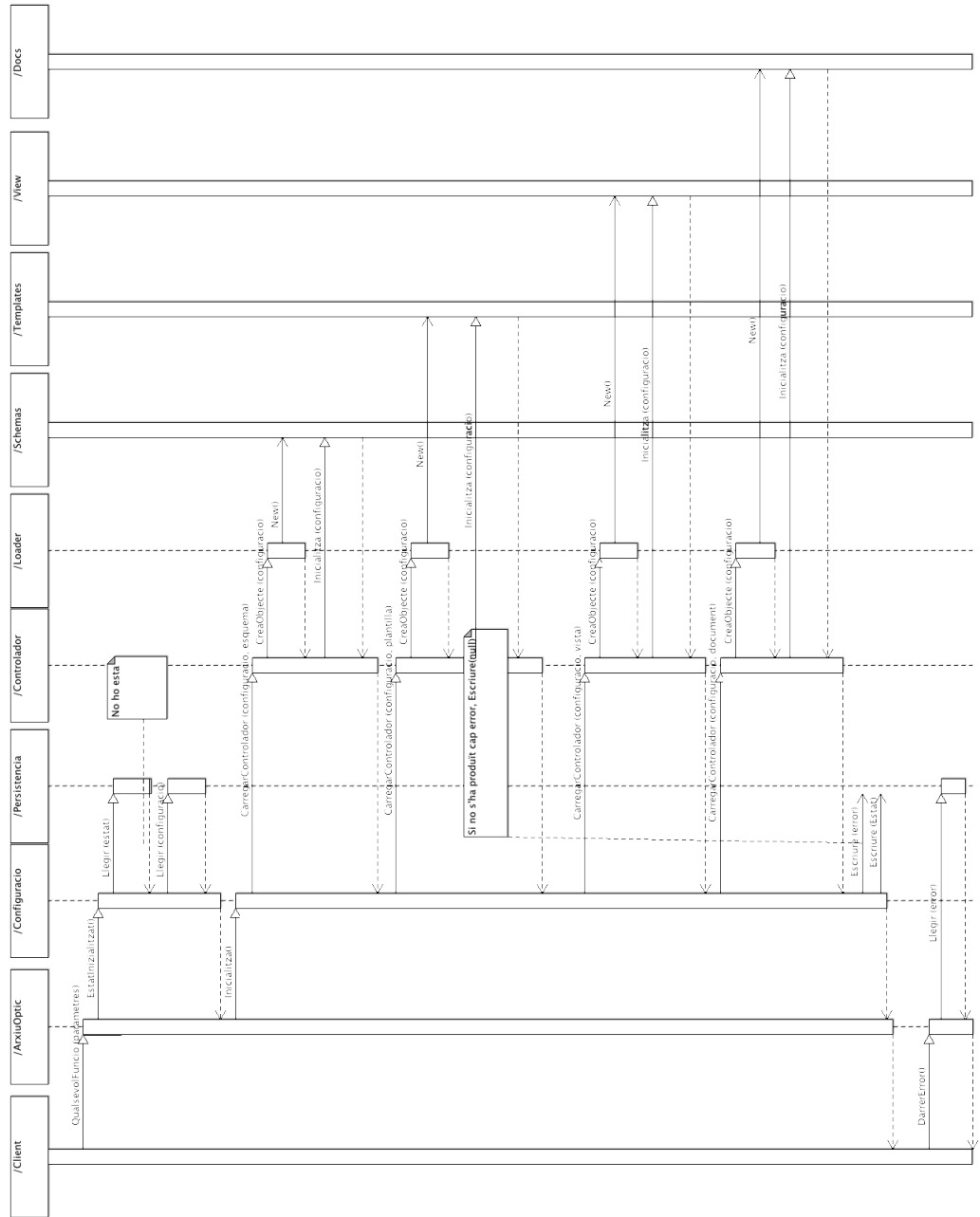
6.1. Diagrames de seqüència

En aquest apartat s'hi adjunten un conjunt de diagrames de seqüència base, repartits en els diferents mòduls del SI. Ens centrarem en els aspectes més significatius de cada mòdul així veurem, entre altres:

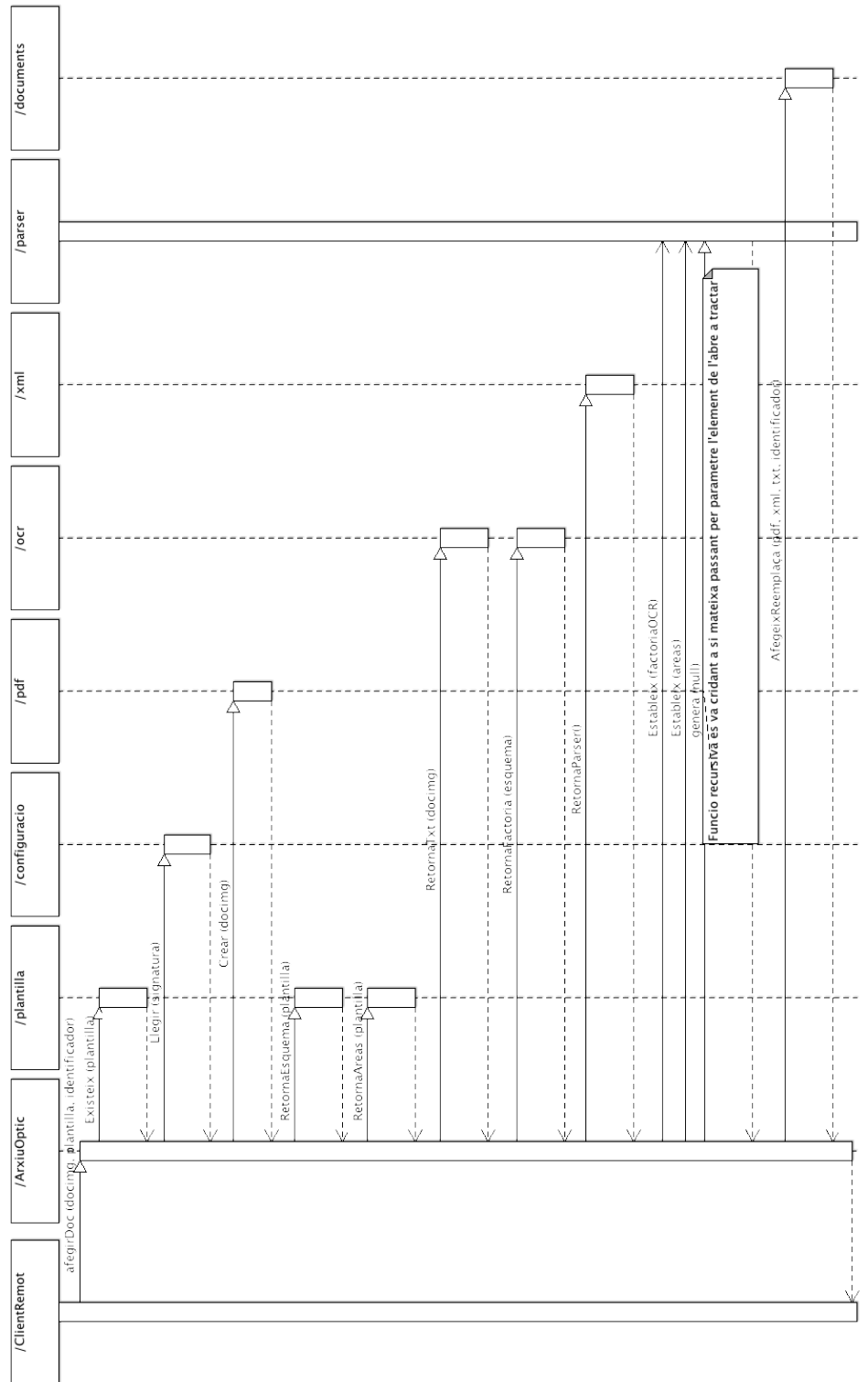
- La inicialització del servei remot
- La creació d'un document
- L'adquisició d'una sèrie de documents
- Creació d'una plantilla
- La inicialització d'un Servei Web
- La inicialització de l'arxiu òptic
- La crida d'una funcionalitat d'un Servei Web.

6.1.1. Diagrames de seqüència de l'Arxiu Òptic

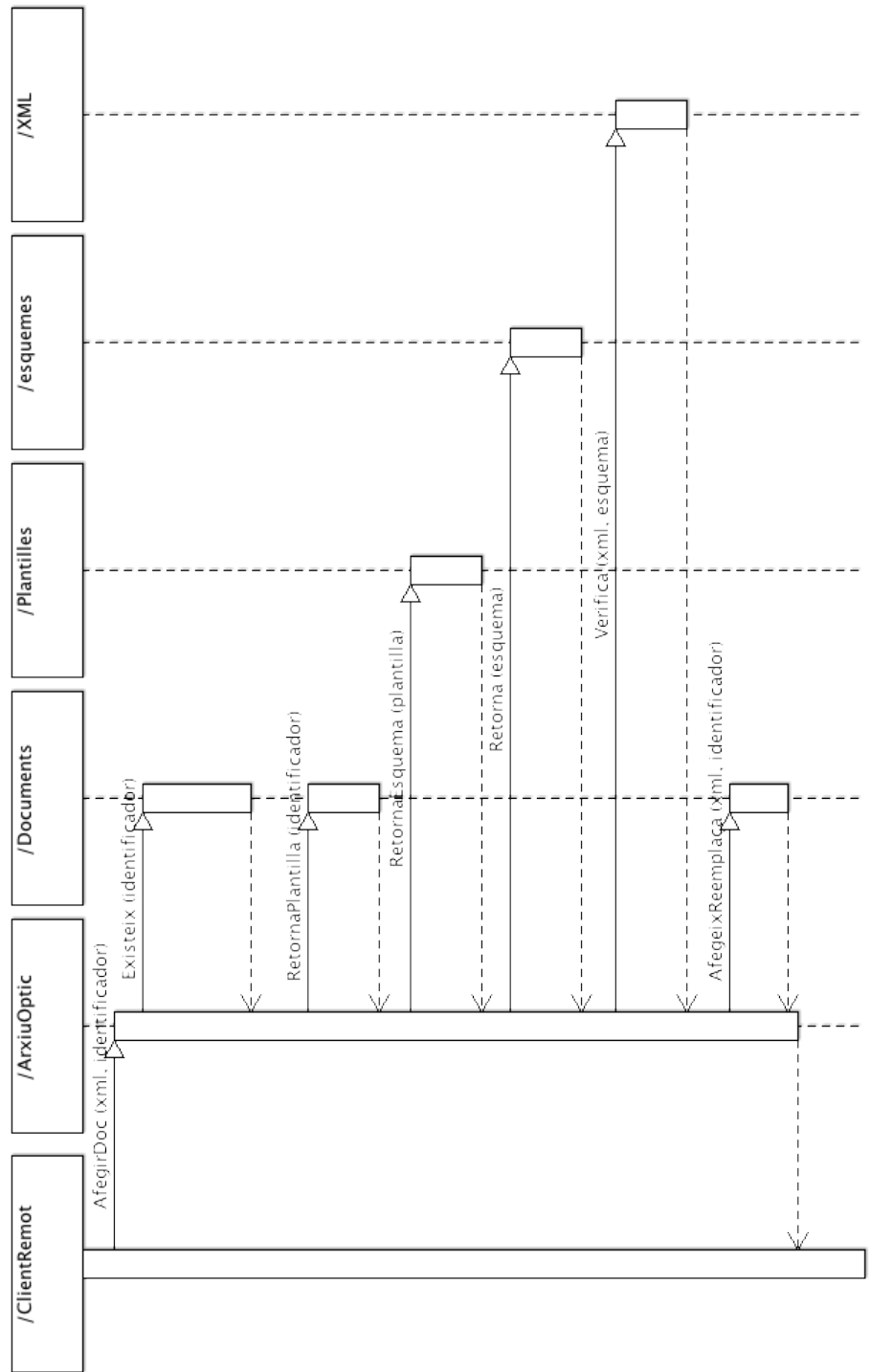
6.1.1.1. La inicialització del servei remot



6.1.1.2. La creació d'un document

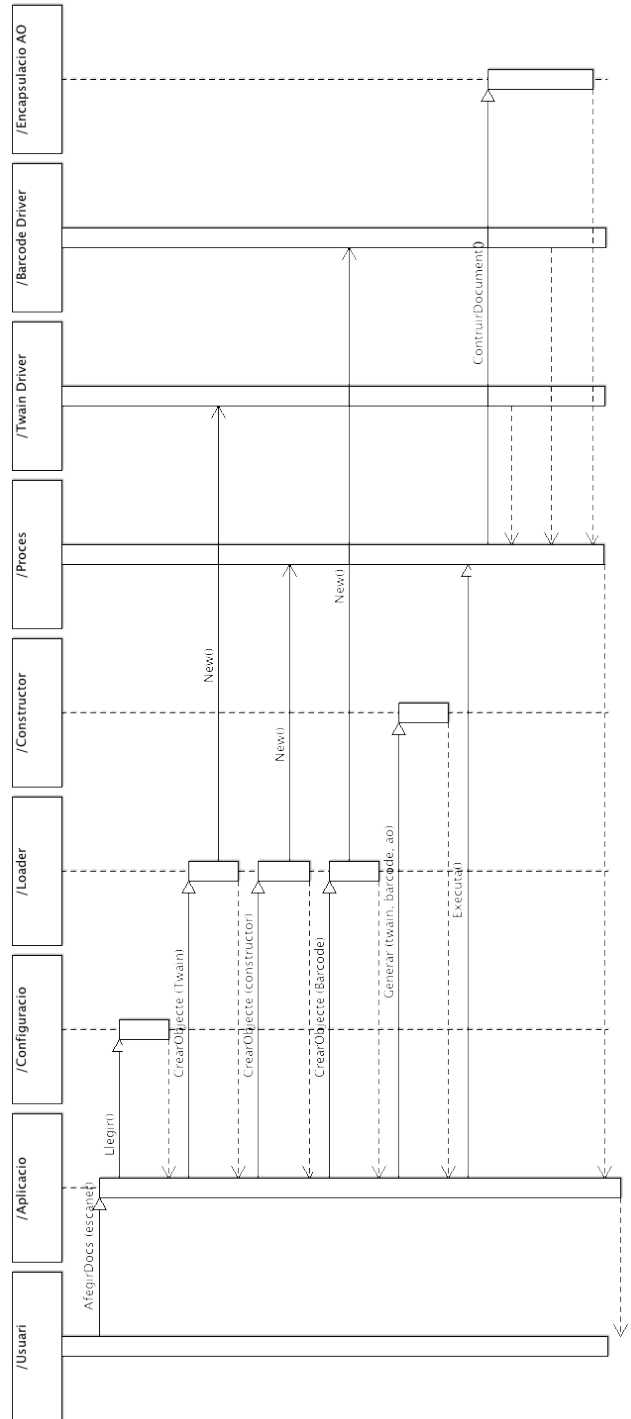


6.1.1.3. Modificar metainformació d'un document

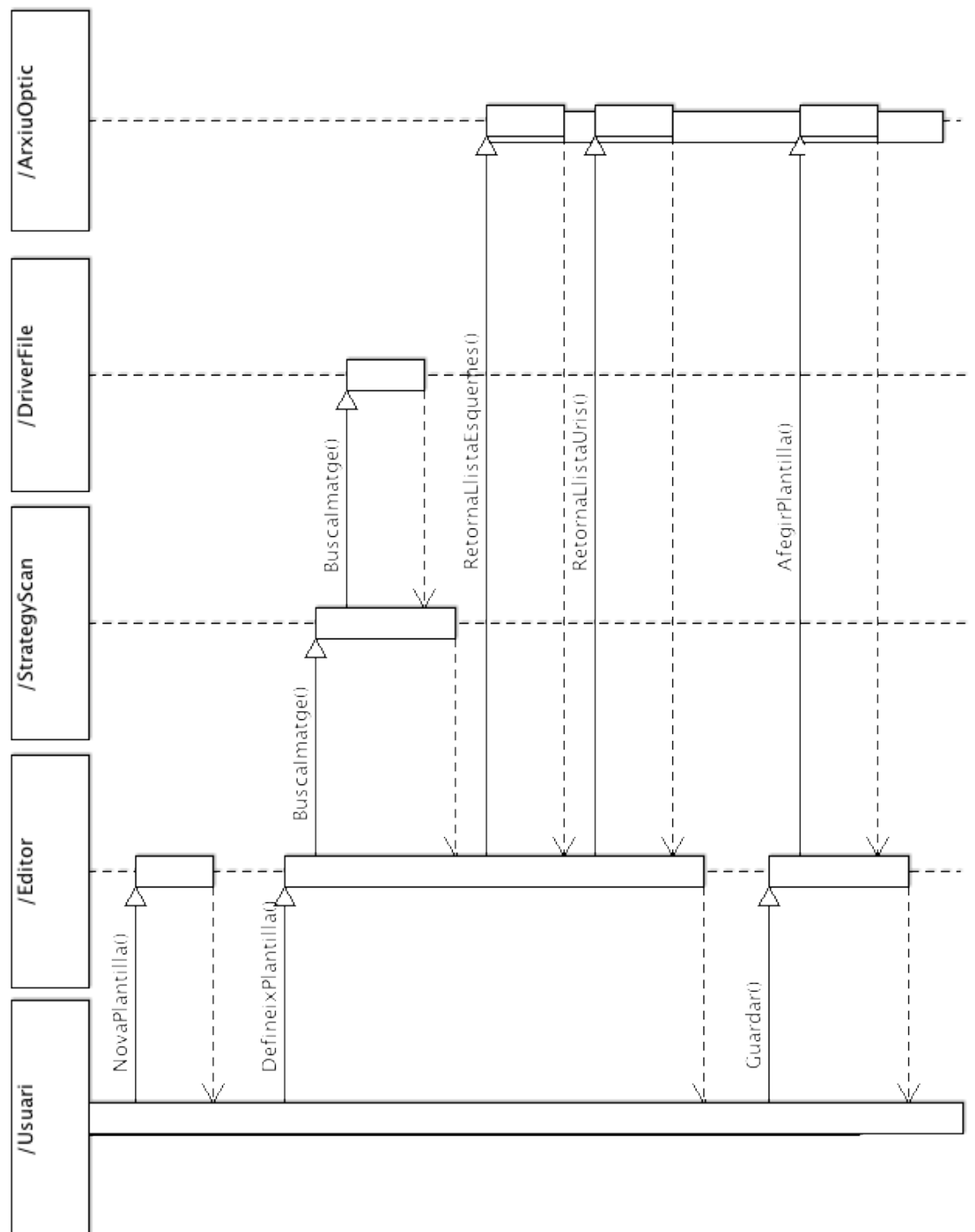


6.1.2. Diagrames de seqüència de l'administració i adquisició de documents

6.1.2.1. L'adquisició d'una sèrie de documents

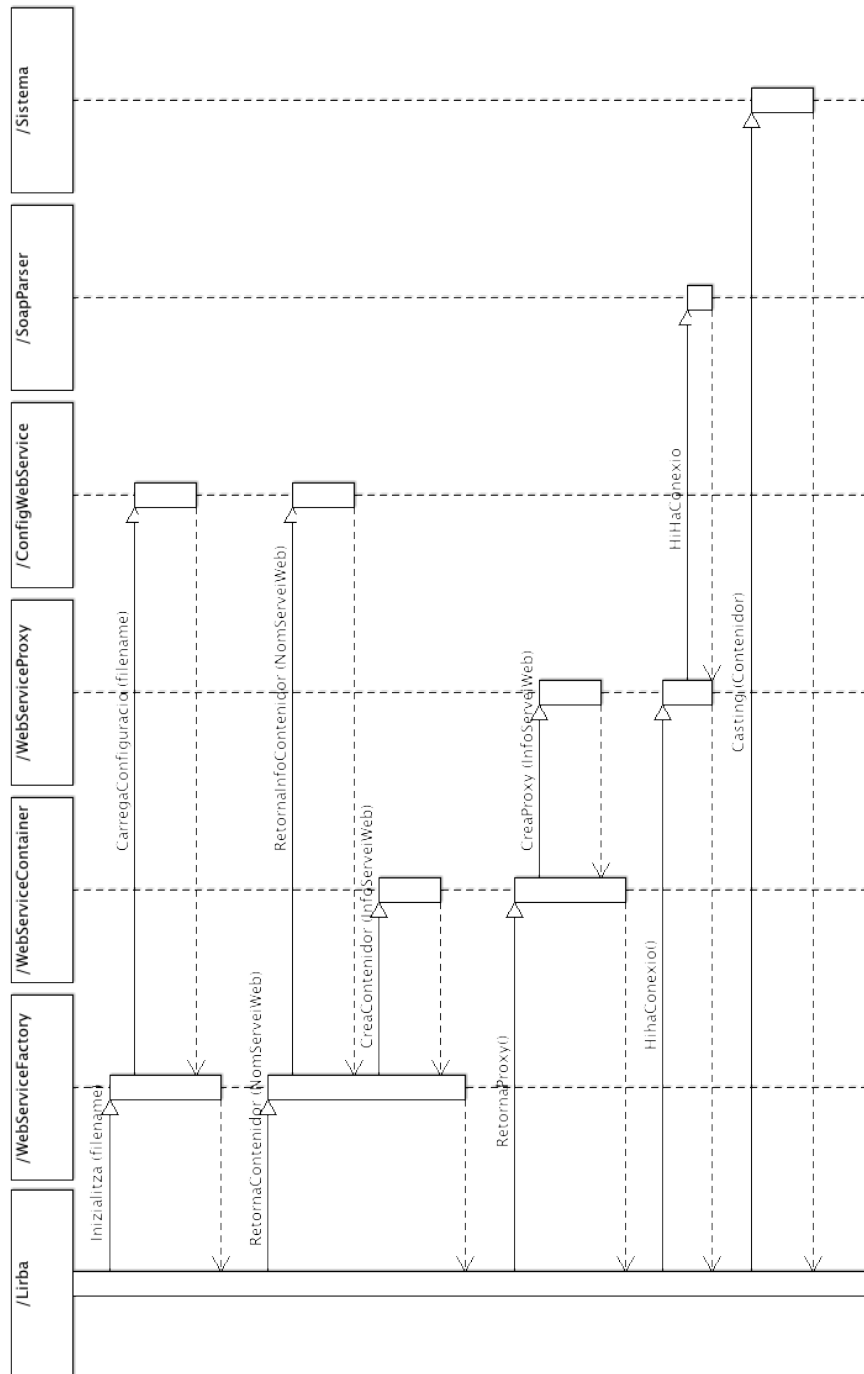


6.1.2.2. Creació d'una plantilla

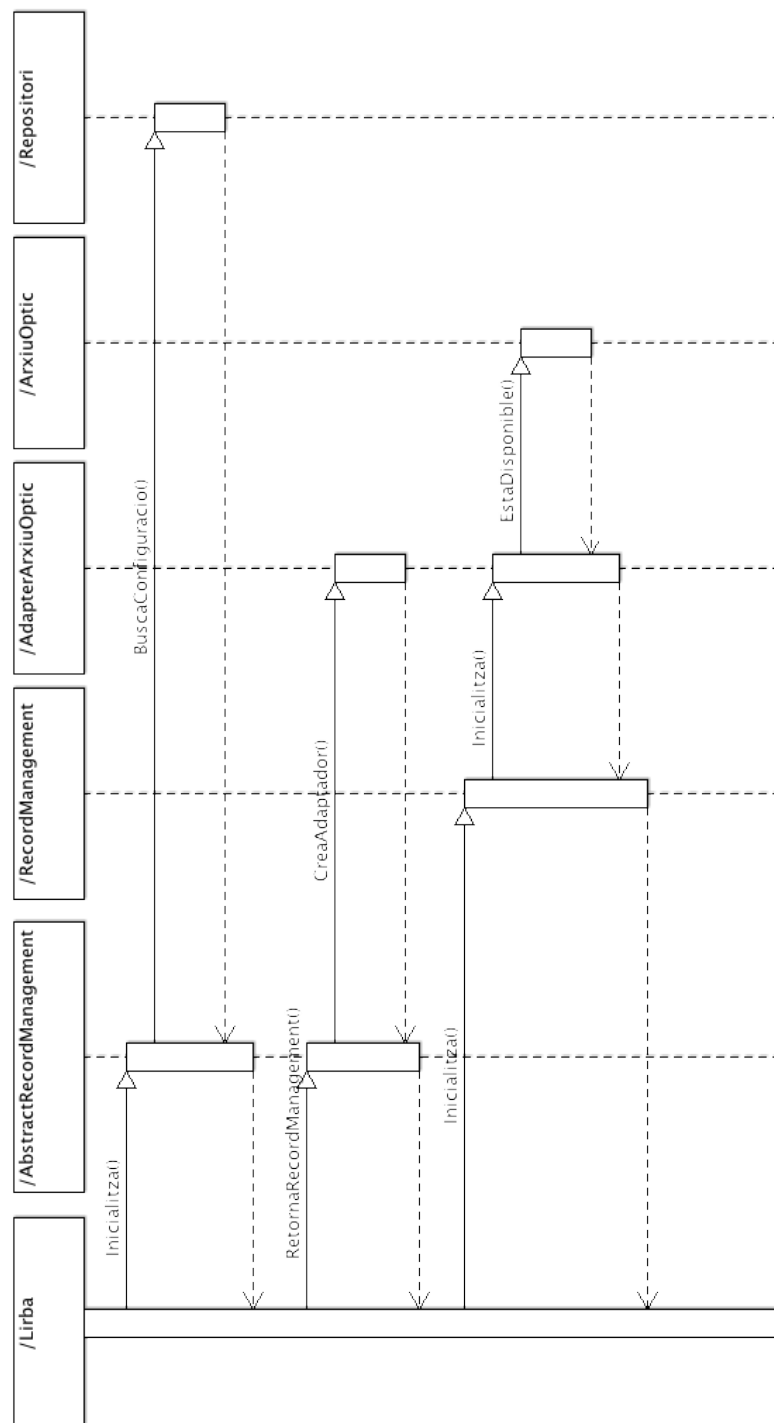


6.1.3. Diagrames de seqüència de la integració amb el Lirba

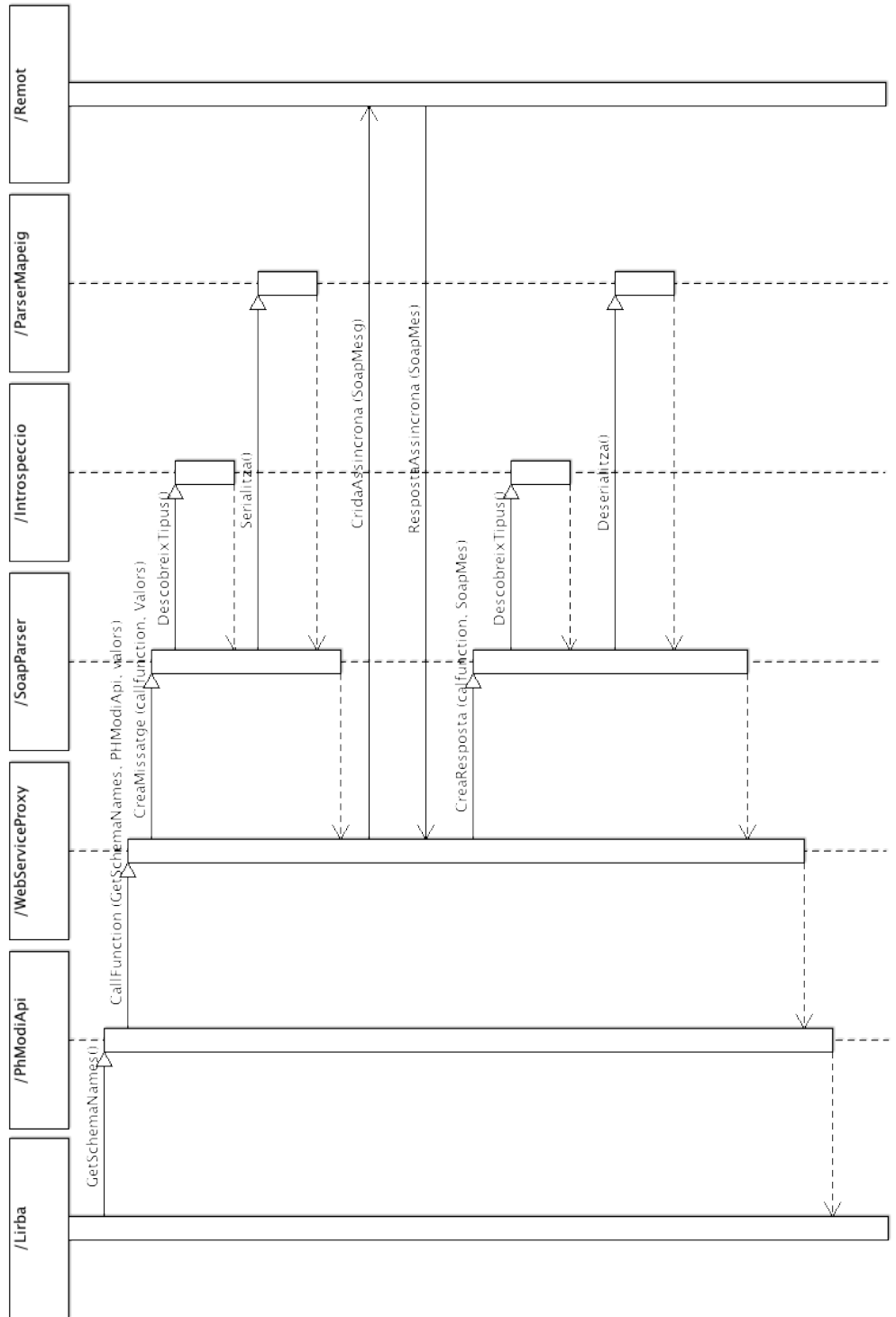
6.1.3.1. La inicialització d'un Servei Web



6.1.3.2. La inicialització de l'arxiu òptic



6.1.3.3. La crida d'una funcionalitat d'un Servei Web.



6.2. Utilització de Patrons

Per tal d'assegurar la reusabilitat en el disseny dels elements del sistema, així com evitar la reiteració a la recerca de solucions ja conegudes i emprades anteriorment s'han utilitzat diferents patrons, tant GRASP com GoF.

6.2.1. Patrons GRASP

En el cas dels patrons GRASP per assignació de responsabilitats, s'han emprat més com a bones pràctiques, tot i així a pres especial rellevància:

Fabricació Pura: Hi ha moltes funcionalitats, que en treballar amb un alt grau d'abstracció s'han pogut reutilitzar i agrupar en classes helper. Així mateix com que hi ha una part del sistema que s'ha d'implementar amb un llenguatge com el VB6, s'han emprat mòduls per donar les funcionalitats del sistema a les interfícies d'usuari de tal manera que els programadors que les implementin no s'hagin de preocupar d'aquestes.

Indirecció: Per tal de fer un desacoplament dels components del sistema, entre la funcionalitat i la creació i recuperació d'aquest, tenim l'abstract factory i el factory. De tal manera que la factoria abstracta s'encarrega de crear el sistema i les factories de donar les funcionalitats, la indirecció entre els dos elements es produeix a partir de la configuració.

6.2.2. Patrons GoF

6.2.2.1. De creació

Abstrac Factory: Aquest patró, que ens permet treballar amb diferents tipologies de famílies d'objectes, s'ha emprat profusament per poder desacoplar, principalment en els components principals de l'Arxiu Òptic. Així doncs, tots els components, s'han estructurat sobre col·leccions abstractes i d'aquesta manera, el sistema interactua amb un únic tipus d'element, independentment de la funcionalitat que tingui o la persistència que se li vulgui donar.

Factory: Aquest patró de creació, que ens permet amagar la subclasse que es genera, s'ha emprat sempre que es requeria obtenir un tipus concret d'objecte. Així per exemple, com amb la construcció del sistema s'empra una factoria abstracta, per poder treballar amb col·leccions, per implementar la funcionalitat és treballa amb una factoria concreta que retorna una família d'objectes concret, ja sigui documents, vistes, etc...

Builder: Aquest patró, que centralitza el procés de creació, s'ha emprat per la creació dels documents (un cop generats els documents electrònics que el componen) i la generació del XML.

Singleton: Aquest patró, ens assegura que només hi haurà una única instància d'un objecte. Això s'ha emprat per la creació del Servei Web.

6.2.2.2. De Estructura

Adapter: Aquest patró, que adapta un objecte ja existent a una funcionalitat no prevista, s'ha emprat entre altres, per adaptar la funcionalitat de productes de terceres companyies a funcionalitats del sistema, com els serveis Web del SharePoint a la funcionalitat de l'Arxiu Òptic o bé el reconeixement de codis de barres.

Bridge: Aquest patró, que desacopla una abstracció de la seva implementació, s'ha emprat entre altres per l'adquisició de documents, permetent realitzar aquesta a partir de diferents fons externes, com un escàner o bé el sistema de fitxers.

Composite: Aquest patró, que permet tractar objectes compostos, s'ha emprat per establir els documents dins de l'Arxiu Òptic, ja que cada documents està compost per diferents documents electrònics (xml,txt,img).

Decorator: Aquest patró, que afegeix dinàmicament funcionalitat a un objecte, s'ha emprat per la construcció dinàmica de Serveis Web.

Facade: Aquest patró, s'ha emprat en tots aquells components que és preveien que podrien ser remots.

Proxy: Aquest patró, que manté una representació d'un objecte s'ha emprat per la construcció dinàmica de Serveis Web.

6.2.2.3. De Comportament

Command: Aquest patró, que encapsula una operació de l'objecte, s'ha emprat per a realitzar el lligam entre la interacció de l'usuari amb les interfícies i la crida a les funcionalitats de l'Arxiu Òptic.

Interpreter: Aquest patró, s'empra per desacoplar el sistema de persistència emprat amb el sistema de Recuperació d'informació. Així doncs, segons el sistema de persistència d'una implantació les aplicacions client podran realitzar un o altre mètode de RI.

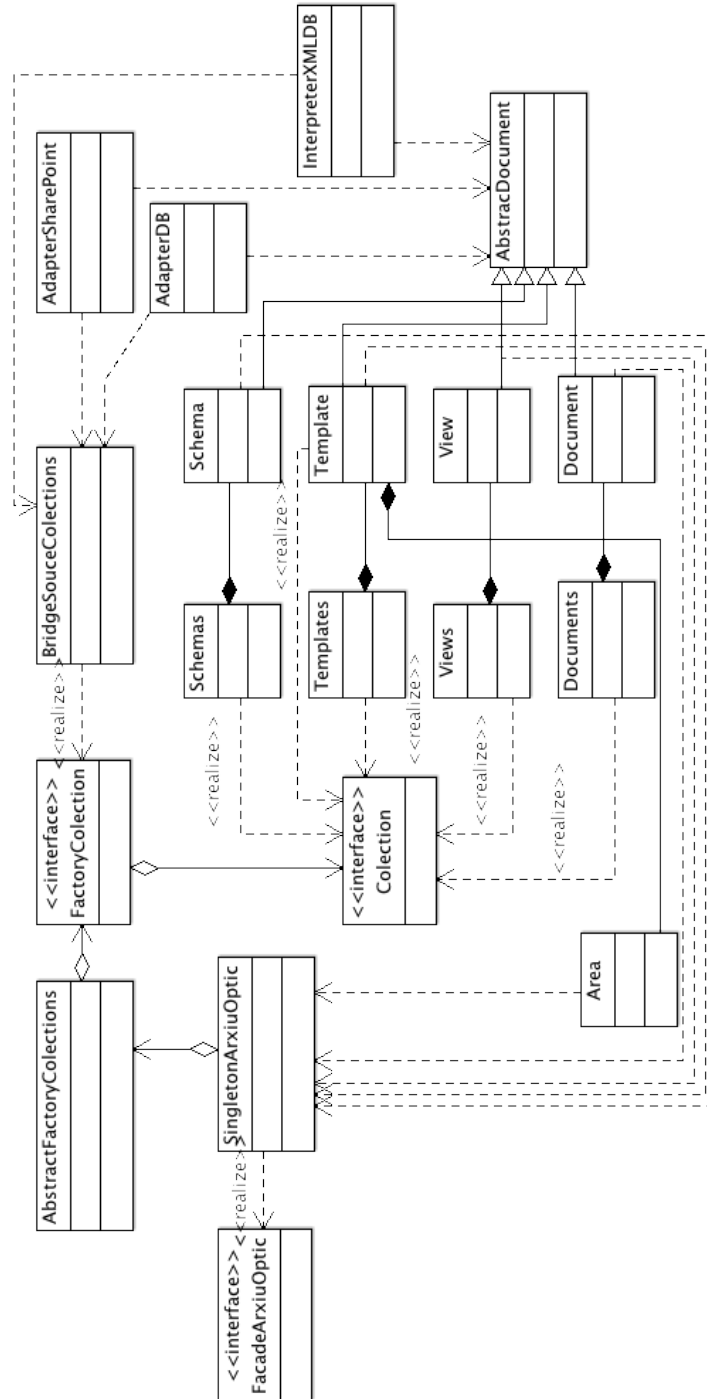
Iterator: Aquest patró, que permet realitzar recorreguts sobre objectes de forma independent de la seva implementació, s'ha emprat profusament, ja que tot els sistema treballa sobre col.leccions. També s'ha emprat en la creació del Driver Twain.

Observer: Aquest patró, que defineix una relació d'un a molts, permetent la notificació de canvis d'estat, s'ha emprat per a poder treballar asíncronament, o bé per treballar sincronament a partir de sistemes asíncrons. Juntament amb l'Strategy, l'iterator, l'adapter i el bridge conformen la funcionalitat d'adquisició d'imatges.

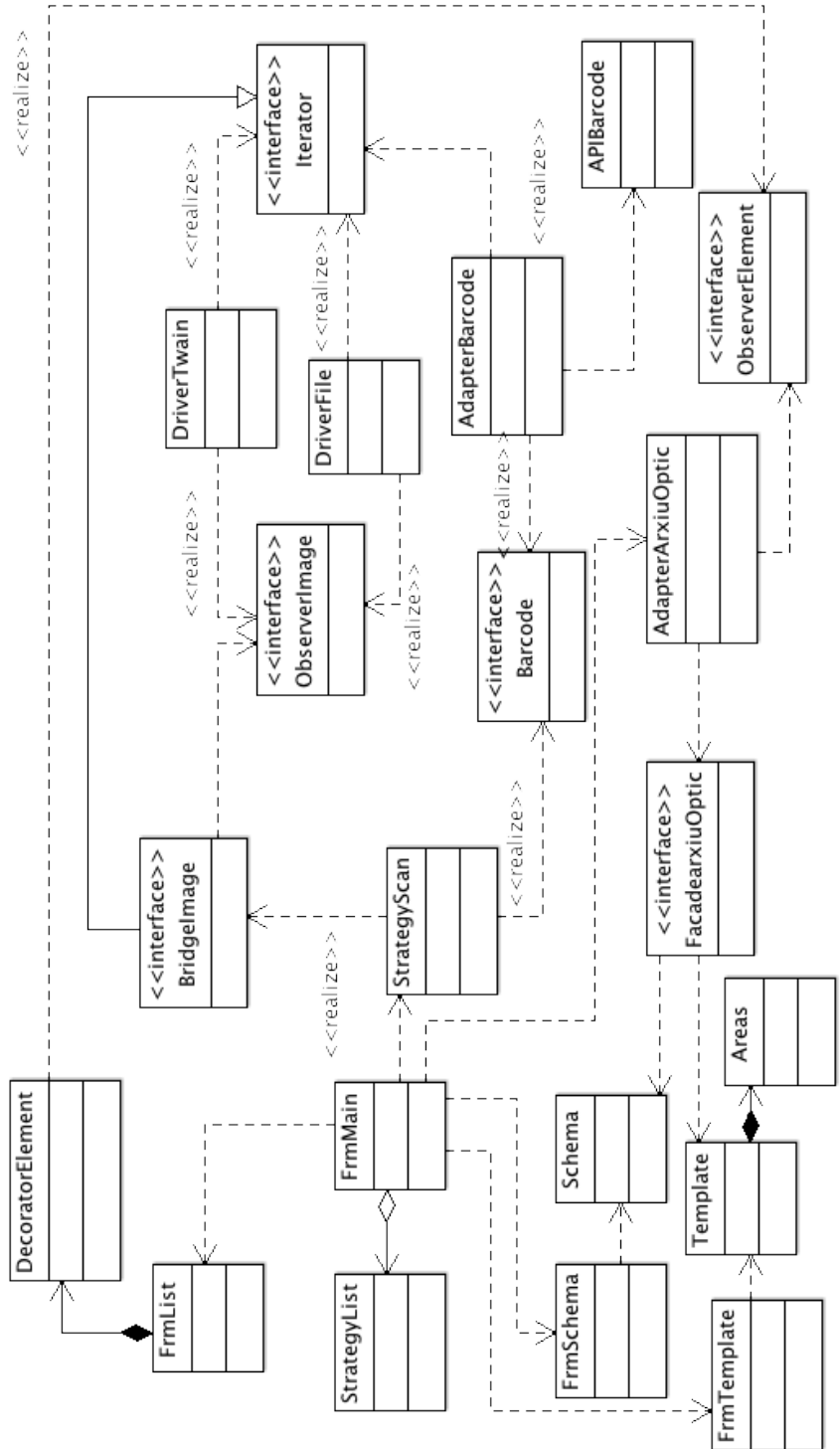
Strategy: Aquest patró, que permet la utilització de diferents mètodes per resoldre un problema, s'ha emprat, per resoldre la creació de documents, permetent l'adquisició d'imatges des de diferents tipus de fons origen.

6.3. Diagrama de Classes

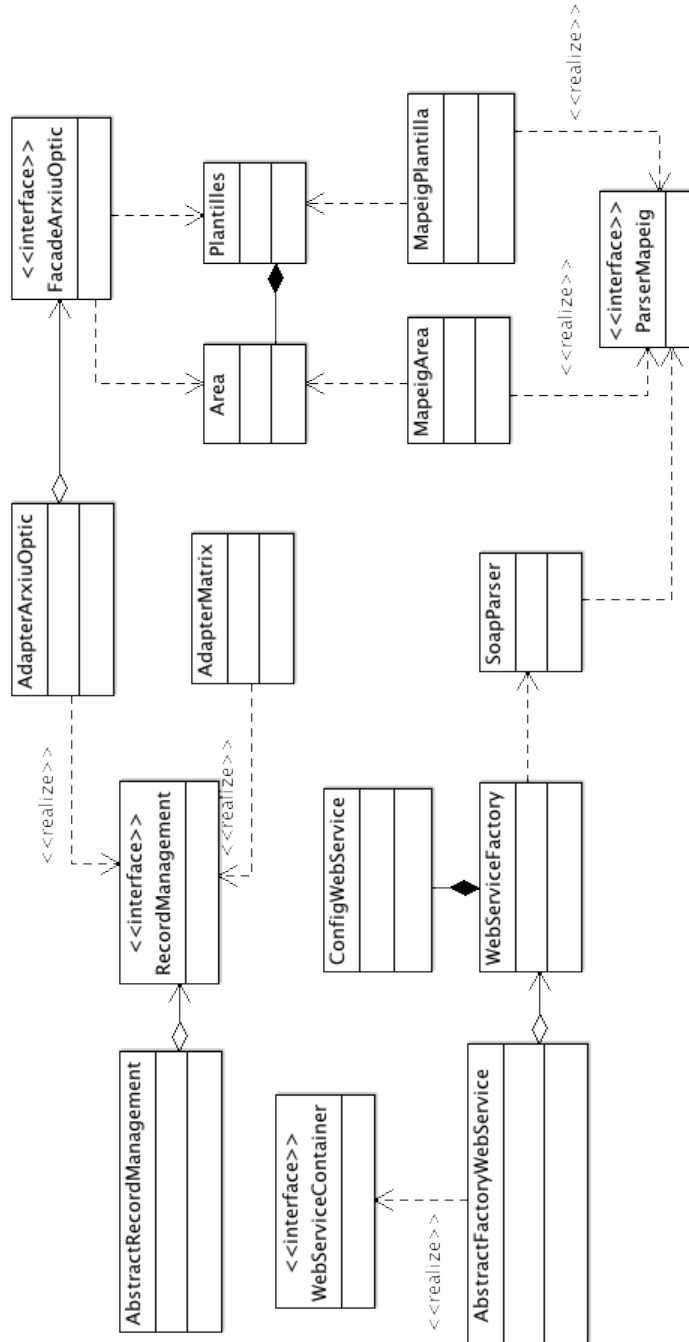
6.3.1. Arxiu Òptic



6.3.2. Administració i adquisició de documents



6.3.3. Integració amb Lirba



7. Implementació

Després d'analitzar les especificacions del SI i de dissenyar tant la seva estructura com el funcionament de les seves especificacions, es va realitzar la seva implementació. A caràcter general la implementació es va realitzar tota amb la plataforma de Microsoft. S'han emprat diferents llenguatges, com el C#, el VB i el VB6.

En aquest capítol doncs, presentaré l'estructura principal de les implementacions del SI i mostraré alguns exemples.

7.1. Arxiu Òptic

7.1.1. Servei Remot

Per realitzar el servei remot s'ha emprat Net remoting, així doncs la classe PhModiAPI hereda d'un objecte base que ofereix la funcionalitat remota e implementa una interface que és la funcionalitat que han de veure les aplicacions remotes (la interface fade facade).

```

Public NotInheritable Class PhModiApi
    Inherits System.MarshalByRefObject
    Implements PhSystems.PHModi.Remoting.IPhModiApi

    Private almacen As PhSystems.PHModi.Remoting.IAlmacen
    Private initialized As Boolean
    Public Shared LastError As Exception

    .....

    Private Function GetElementLocation(ByVal ElementName As String, ByVal TypeElement As PhSystems.PHModi.Remoting.TypeElement) As String

        Select Case TypeElement
            Case PhSystems.PHModi.Remoting.TypeElement.document
                If InStr(ElementName, "\") = 0 Then
                    Return Me.GetDocumentName(ElementName)
                Else
                    Return ElementName
                End If
            Case Else
                Return ElementName
        End Select

    End Function

#Region "IPhModiApi"

    .....

    Public Function ExistElement(ByVal ElementName As String, ByVal TypeElement As PhSystems.PHModi.Remoting.TypeElement) As Boolean
        Implements PhSystems.PHModi.Remoting.IPhModiApi.ExistElement

        If Not Me.IsAvaiable Then Return False
        ElementName = Me.GetElementLocation(ElementName, TypeElement)
        Select Case TypeElement
            Case PhSystems.PHModi.Remoting.TypeElement.document
                Return PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.Documents.Exist(ElementName)
            Case PhSystems.PHModi.Remoting.TypeElement.schema
                Return PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.Schemas.Existeix(ElementName)
            Case PhSystems.PHModi.Remoting.TypeElement.template
                Return PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.templates.Existeix(ElementName)
            Case PhSystems.PHModi.Remoting.TypeElement.xslt 'PFC
                Return PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.XSLT.Existeix(ElementName)
        End Select

    End Function

    Public Function GetLastException() As String Implements PhSystems.PHModi.Remoting.IPhModiApi.GetLastException
        If Me.LastError Is Nothing Then Exit Function
        Return Me.LastError.ToString
    End Function

    Public Function GetElementTemplate(ByVal TemplateName As String) As String Implements
        PhSystems.PHModi.Remoting.IPhModiApi.GetElementTemplate

        Try

```

```

        If Not Me.IsAvailable Then Return ""
        If PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.templates Is Nothing Then Return ""
        TemplateName = Me.GetElementLocation(TemplateName, PhSystems.PHModi.Remoting.TypeElement.template)
        If Not PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.templates.Existeix(TemplateName) Then
Return ""
        Dim template As PhSystems.PHModi.Serialization.OCR.Template =
PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.templates.GetTemplate(TemplateName)
        If template Is Nothing Then Return ""
        If template.Element Is Nothing Then Return ""
        Return template.Element.Name()
        Catch ex As Exception
        Me.RemotingError(ex)
        End Try

    End Function
    Public Function GetElementsSchema(ByVal SchemaName As String) As String() Implements
PhSystems.PHModi.Remoting.IPHModiApi.GetElementsSchema

    Try
        If Not Me.IsAvailable Then Return New String() {}
        SchemaName = Me.GetElementLocation(SchemaName, PhSystems.PHModi.Remoting.TypeElement.schema)
        Return PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.Schemas.GetElementsSchema(SchemaName)
        Catch ex As Exception
        Me.RemotingError(ex)
        End Try

    End Function
    Public Function GetDocumentsNamesTemplates(ByVal TemplateName As String) As String() Implements
PhSystems.PHModi.Remoting.IPHModiApi.GetDocumentsNamesTemplates

    Try
        If Not Me.IsAvailable Then Return New String() {}
        TemplateName = Me.GetElementLocation(TemplateName, PhSystems.PHModi.Remoting.TypeElement.template)
        Return
PhSystems.PHModi.Serialization.Remoting.Configuration.singleton.Documents.GetDocumentsNamesTemplate(TemplateName)
        Catch ex As Exception
        Me.RemotingError(ex)
        End Try

    End Function

.....
#End Region

Public Sub New()

    MyBase.New()
    On Error Resume Next
    PhSystems.Generic.Base.Logs.setLog(PhSystems.Generic.Base.ClassLoader.TypeFullName(Me.GetType))
    PhSystems.Generic.Base.Logs.WriteEntry("Init", EventLogEntryType.Information)

End Sub

Protected Overrides Sub Finalize()

    MyBase.Finalize()
    Me.Close()

End Sub
Private Sub RemotingError(ByVal ex As Exception)

    Me.Close()
    Me.LastError = ex
    PhSystems.Generic.Base.Logs.WriteEntry(ex.Message & Chr(13) & ex.StackTrace, EventLogEntryType.Error)

End Sub
Private Sub RemotingInitialize()

    PhSystems.PHModi.Serialization.Remoting.PHModiApiConfig.open()
    If Not PhSystems.PHModi.Serialization.Remoting.PHModiApiConfig.singleton Is Nothing Then Me.initialized =
PhSystems.PHModi.Serialization.Remoting.PHModiApiConfig.singleton.Initialized()

End Sub
Private Sub Close()

    PhSystems.PHModi.Serialization.Remoting.PHModiApiConfig.close()

End Sub

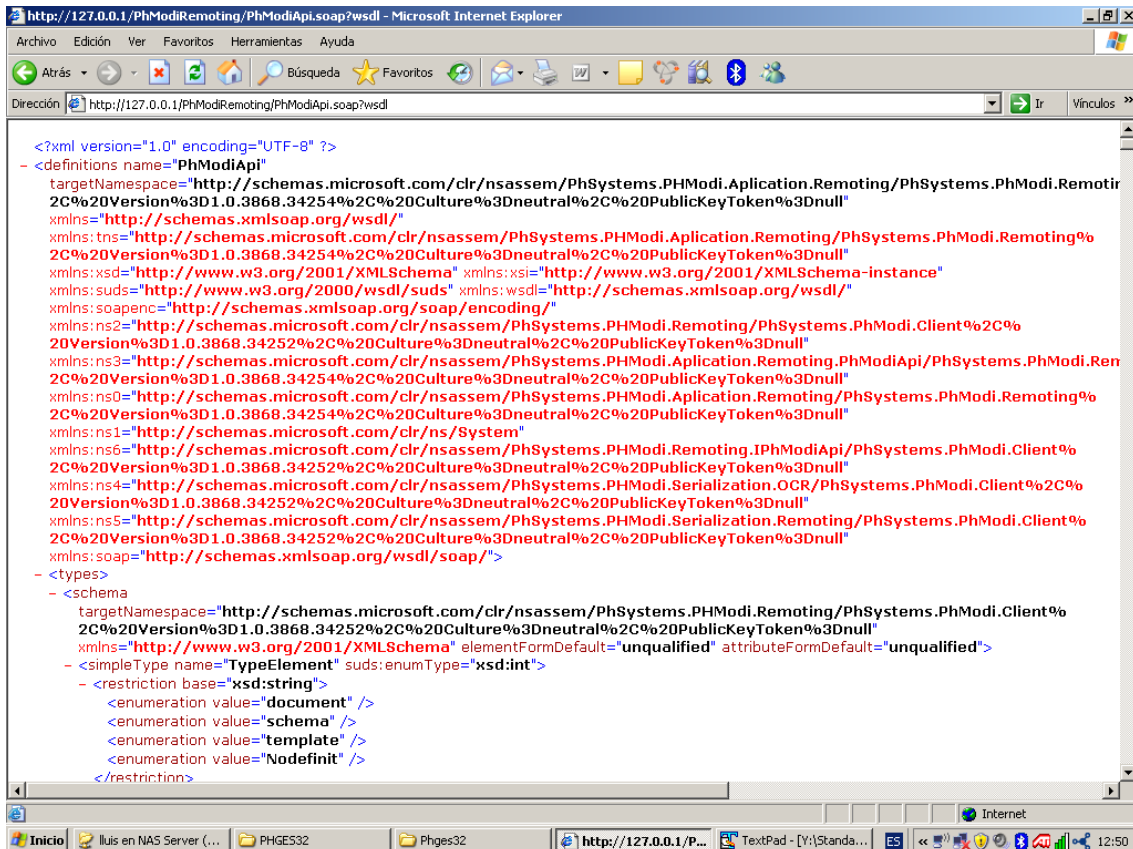
End Class

```

Un cop implementada la funcionalitat i compilada en una DLL, s'ha de publicar aquest servei remot, per fer-ho s'utilitza el ISS com a contenidor, i es configura perquè a una petició dins del SITE creï un objecte singleton sobre la classe especificada. Com que s'estableix que el protocol de transport sigui SOAP, el que s'està fent en realitat és publicar un servei web.

```
<configuration>
  <system.runtime.remoting>
    <application>
      <service>
        <wellknown mode="Singleton"
        type="PhSystems.PHModi.Application.Remoting.PhModiApi,PhSystems.PhModi.Remoting,
        Culture=neutral, PublicKeyToken=null"
        objectUri="PhModiApi.soap"/>
        <wellknown mode="Singleton"
        type="PhSystems.PHModi.Application.Remoting.ALmacen,PhSystems.PhModi.ALmacen,
        Culture=neutral, PublicKeyToken=null"
        objectUri="PhALmacenApi.soap"/>
        <wellknown mode="Singleton"
        type="PhSystems.PHModi.Application.Remoting.PhTypeDocEx,PhSystems.PhModi.ALmacen,
        Culture=neutral, PublicKeyToken=null"
        objectUri="PhTypeDocApi.soap"/>
      </service>
    </application>
    <customErrors mode="off"/>
    <debug loadTypes="true" />
  </system.runtime.remoting>
<!--
  <httpRuntime
  executionTimeout="110"
  maxRequestLength="4096"
  requestLengthDiskThreshold="80"
  useFullyQualifiedRedirectUrl="false"
  minFreeThreads="8"
  minLocalRequestFreeThreads="4"
  appRequestQueueLimit="5000"
  enableKernelOutputCache="true"
  enableVersionHeader="true"
  requireRootedSaveAsPath="true"
  enable="true"
  shutdownTimeout="90"
  delayNotificationTimeout="5"
  waitChangeNotification="0"
  maxWaitChangeNotification="0"
  enableHeaderChecking="true"
  sendCacheControlHeader="true"
  apartmentThreading="false" />
-->
</configuration>
```

Per tant si fem una crida des de qualsevol navegador podrem accedir a la seva definició (arxiu wsdl)



7.1.2. Carregador de classes.

Una característica important és la flexibilitat del sistema; així doncs moltes funcionalitats s'han implementat com a interfaces, que fan de facade, permetent d'aquesta manera carregar la implementació real des de llibreries externes al sistema, que poden ser components remots.

Aquí el codi que permet carregar classes via configuració.

```
<Serializable()> Public Class ClassLoader
    <XmlAttributeAttribute("TypeClassName"), System.ComponentModel.DefaultValueAttribute("")> Public TypeClassName As String
    <XmlAttributeAttribute("Arguments"), IsNullable:=True, Type:=GetType(ClassLoader)> Public Arguments As ClassLoader()

    Public Sub New()
        Me.TypeClassName = ""
        Me.Arguments = Nothing
    End Sub
    Public Sub New(ByVal TypeClassName As String, ByVal Arguments As Object())
        Me.TypeClassName = TypeClassName
        Me.Arguments = Arguments
    End Sub
    Public Function ClassLoad() As Object
        ForName(Me.TypeClassName, GetArgs)
    End Function
    Private Function GetArgs() As Object()
        If Me.Arguments Is Nothing Then Return Nothing
        Dim i As Integer
        Dim ret(Me.Arguments.Length - 1) As Object
        For i = 0 To Me.Arguments.Length - 1
            ret(i) = Me.Arguments(i).ClassLoad
        Next
        Return ret
    End Function
    Private Shared Sub CallByNameIndirectionSenderEventHandler(ByRef item As Object, ByRef sender As Object, ByRef EventArgs
As EventArgs)
        Try
            'nomes queda per tractar el cas de que sigui una propietat "indizada"
            CallByName(sender, Split(item, "=")(0), CallType.Set, New Object() {Split(item, "=")(1)})
            'If Split(item, "=")(1) Like "*" Then
            '    sender.GetType.GetProperty(Split(item, "=")(0)).GetMethod.Invoke(sender, New Object()
{ForString(sender.GetType.GetProperty(Split(item, "=")(0)).DeclaringType.GetConstructor(System.Type.EmptyTypes).Invoke(New
Object() {}), Split(item, "=")(1))})
            'Else
            '    sender.GetType.GetProperty(Split(item, "=")(0)).GetMethod.Invoke(sender, New Object()
{System.Convert.ChangeType(Split(item, "=")(1), sender.GetType.GetProperty(Split(item,
"="(0)).GetMethod.GetParameters(0).ParameterType)})
            'End If
            Catch ex As Exception
        End Try
    End Sub
    Private Shared Function GetObject(ByVal TypeName As String, ByVal type As System.Type) As Object
        Return Activator.GetObject(type, TypeName)
    End Function
    Private Shared Function IsUrl(ByVal url As String) As Boolean
        Return Mid(UCase(url), 1, Len("HTTP://")) = "HTTP://" Or Mid(UCase(url), 1, Len("HTTPS://")) = "HTTPS://"
    End Function
    Public Shared Sub ForString(ByRef [object] As Object, ByVal value As String)
        If value Like "*" Then PhSystems.Generic.Base.Iterate.IndirectionEvent(Split(value.Substring(1, value.Length - 2),
",").GetEnumerator, [object], Nothing, AddressOf CallByNameIndirectionSenderEventHandler)
    End Sub
    Public Shared Function ForName(ByVal TypeName As String, ByVal type As System.Type) As Object
        Try
            If IsUrl(TypeName) Then Return GetObject(TypeName, type)
            If Not getWellKnownClientTypeEntry(TypeName) Is Nothing Then Return
GetObject(getWellKnownClientTypeEntry(TypeName).ObjectUrl, type)
            Return ForName(TypeName)
        Catch ex As Exception
            If System.Environment.UserInteractive Then MsgBox(ex.Message)
        End Try
    End Function
End Class
```



```

Public Shared Function getWellKnownClientTypeEntry(ByVal Type As Type) As
System.Runtime.Remoting.WellKnownClientTypeEntry
    Return getWellKnownClientTypeEntry(Type.Assembly.FullName & ", " & Type.FullName)
End Function
Public Shared Function getWellKnownClientTypeEntry(ByVal Type As String) As
System.Runtime.Remoting.WellKnownClientTypeEntry
    If InStr(Type, ",") = 0 Then Exit Function
    Dim enumerator As IEnumerator =
System.Runtime.Remoting.RemotingConfiguration.GetRegisteredWellKnownClientTypes.GetEnumerator()
    While enumerator.MoveNext
        If CType(enumerator.Current, System.Runtime.Remoting.WellKnownClientTypeEntry).AssemblyName & ", " &
CType(enumerator.Current, System.Runtime.Remoting.WellKnownClientTypeEntry).TypeName = Split(Type, ",")(1) & ", " & Split(Type,
",")(0) Then Return enumerator.Current
    End While
End Function
Public Shared Function getWellKnownClientTypeEntry(ByVal assemblyName As String, ByVal TypeName As String) As
System.Runtime.Remoting.WellKnownClientTypeEntry
    Return getWellKnownClientTypeEntry(assemblyName & ", " & TypeName)
End Function
Public Shared Function ForName(ByVal Type As String, Optional ByVal args() As Object = Nothing) As Object
    If InStr(Type, ",") = 0 Then Exit Function
    Return ForName(Mid(Type, Split(Type, ",")(0).Length + 2), Split(Type, ",")(0), args)
End Function
Public Shared Function ForName(ByVal assemblyName As String, ByVal TypeName As String, Optional ByVal args() As Object =
Nothing) As Object
    Try
        Return AppDomain.CurrentDomain.CreateInstanceAndUnwrap(assemblyName, TypeName, True,
Reflection.BindingFlags.Default, Nothing, args, Nothing, Nothing, Nothing)
    Catch ex As Exception
        If System.Environment.UserInteractive Then MsgBox(ex.Message)
    End Try
End Function
Public Shared Function IsSubclassOf(ByVal subject As Type, ByVal Subclass As Type) As Boolean
    Return IsSubclassOf(subject, Subclass.FullName)
End Function
Public Shared Function IsSubclassOf(ByVal subject As Type, ByVal fullname As String) As Boolean
    If subject.FullName = fullname Or subject.FullName = GetType(Object).FullName Then Return subject.FullName =
fullname
    Return IsSubclassOf(subject.BaseType, fullname)
End Function
Public Shared Function TypeValue(ByVal typestring As String) As System.Type
    If Not InStr(typestring, ",") = 0 Then Return System.Reflection.Assembly.LoadWithPartialName(Split(typestring,
",")(1)).GetType(Split(typestring, ",")(0))
    Return System.Type.GetType(typestring)
End Function
Public Shared Function TypeValue(ByVal myTypeCode As TypeCode) As System.Type
    Return TypeValue(CType(If(Not InStr(myTypeCode.ToString, ".") = 0, myTypeCode.ToString(), "System." &
myTypeCode.ToString()), String))
End Function
Public Shared Function TypeFullName(ByVal type As Type) As String
    Return type.FullName & ", " & type.Assembly.FullName
End Function
End Class

```

Els fitxers de configuració no són més que classes serialitzables en XML.

```

<?xml version="1.0"?>
<PhModiApiConfig xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
DataResourceName="PhSystems.PHModi.Remoting.IPHModiApi.config"
DataResourceClassName="PhSystems.Base.Resource.XmlSerializerAdapterResource,PhSystems.Base, Version=1.0.2761.28181,
Culture=neutral, PublicKeyToken=null" xmlns="http://phsystems.es/PHModi/PHModiApiConfig.xsd">
  <ConfigRules />
  <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2761.28181, Culture=neutral,
PublicKeyToken=null" />
</PhModiApiConfig>

```

Un altre exemple seria:

```
<?xml version="1.0"?>
<ArrayOfParam xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Param Name="IBuilderDBExecute" Value="PhSystems.Data.Simple.BuilderDBExecute,Phsystems.Data, Version=1.0.2628.15835, Culture=neutral, PublicKeyToken=null" />
  <Param Name=".IDbConnection" Value="System.Data.OracleClient.OracleConnection,System.Data.OracleClient, Version=1.0.5000.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  <Param Name=".Connection" Value="user id=Lrboptic;data source=arxiuoptic.bmark;password=Lrboptic" />
  <Param Name=".Behavior" Value="Default" />
  <Param Name=".DbType" Value="String" />
  <Param Name=".Direction" Value="1" />
  <Param Name=".SourceVersion" Value="Default" />
  <Param Name=".Name.SourceColumn" Value="Name" />
  <Param Name=".ParentName.SourceColumn" Value="ParentName" />
  <Param Name=".ResourceName.SourceColumn" Value="ResourceName" />
  <Param Name=".DynamicProperties" Value="OracleType" />
  <Param Name=".OracleType" Value="VarChar" />
  <Param Name="Get.CommandText" Value="SELECT Name,ParentName FROM AOLINK WHERE ParentName=:ParentName and ResourceName=:ResourceName" />
  <Param Name="GetALL.CommandText" Value="SELECT Name,ParentName FROM AOLINK WHERE ResourceName=:ResourceName" />
  <Param Name="Drop.CommandText" Value="DELETE FROM AOLINK WHERE ParentName=:ParentName AND ResourceName=:ResourceName" />
  <Param Name="Put.CommandText" Value="INSERT INTO AOLINK (ResourceName,ParentName,Name) VALUES(:ResourceName,:ParentName,:Name)" />
  <Param Name="Exist.CommandText" Value="SELECT COUNT(*) FROM AOLINK WHERE ParentName=:ParentName AND Name=:Name AND ResourceName=:ResourceName" />
  <Param Name="GetInverse.CommandText" Value="SELECT Name,ParentName FROM AOLINK WHERE Name=:Name and ResourceName=:ResourceName" />
  <Param Name="DropOne.CommandText" Value="DELETE FROM AOLINK WHERE ParentName=:ParentName AND Name=:Name AND ResourceName=:ResourceName" />
  <Param Name="Count.CommandText" Value="SELECT COUNT(*) FROM AOLINK WHERE ParentName=:ParentName AND ResourceName=:ResourceName" />
  <Param Name="CountALL.CommandText" Value="SELECT COUNT(*) FROM AOLINK WHERE ResourceName=:ResourceName" />
  <Param Name="GetPattern.CommandText" Value="SELECT Name,ParentName FROM AOLINK WHERE Name LIKE :Pattern and ResourceName=:ResourceName" />
  <Param Name="GetPattern.Pattern.SourceColumn" Value="Name" />
</ArrayOfParam>
```

Una detall molt interessant és, que la càrrega de classes és independent de si aquestes són remotes o no, això s'aconsegueix carregant un fitxer amb la configuració dels tipus remots, així doncs quan s'inicialitza qualsevol component el primer que es fa és carregar el fitxer serialitzat i convertir-lo en un objecte de configuració aquest objecte de configuració amb el classLoader s'encarregar d'anar composant tots els subelements que conformen el component. El classLoader carrega per tipus els objectes, tipus que estan definits en els fitxers XML (que es transformaran en propietats d'objectes de configuració), aquests tipus però poden estar en una llibreria local o bé ser una interface local sobre un tipus remot.

El fitxer de configuració remota és el següent:

```
<configuration>
  <system.runtime.remoting>
    <application>
      <lifetime>
        LeaseTime="20M"
        sponsorshipTimeout="8M"
        renewOnCallTime="8M"
        LeaseManagerPollTime="40S"
      />
      <client url="http://localhost/PhModiRemoting/">
        <wellknown
          type="PhSystems.PHModi.Remoting.IPHModiApi,PhSystems.PhModi.Client"
          objectUri="PhModiApi.soap" url="http://localhost/PhModiRemoting/PhModiApi.soap"/>
        </client>
      <client url="http://localhost/PhModiRemoting/">
        <wellknown
          type="PhSystems.PHModi.Remoting.IPHTypeDocApiEx,PhSystems.PhModi.Client"
          objectUri="PhTypeDocApi.soap" url="http://localhost/PhModiRemoting/PhTypeDocApi.soap"/>
        </client>
      <client url="http://localhost/PhModiRemoting/">
        <wellknown
          type="PhSystems.PHModi.Remoting.IALmacen,PhSystems.PhModi.Client"
          objectUri="PhALmacenApi.soap" url="http://localhost/PhModiRemoting/PhALmacenApi.soap"/>
        </client>
      <channels>
        <channel ref="http" useDefaultCredentials="true"/>
      </channels>
    </application>
  </system.runtime.remoting>
</configuration>
```

Així doncs, això ens permet carregar els components a partir de facades, independentment de si són remots o no, i la implementació concreta que facin (DB,XML, etc.); tot via configuració.

7.1.3. Generador XML

Un objectiu del projecte és dotar de semàntica els documents electrònics, això s'aconsegueix definint un esquema per cada tipus de document electrònic i generant un document XML amb les dades i metadades de les imatges dels documents administratius. Això es pot realitzar de dues formes diferents:

- a partir d'un flux de treball sabent la correspondència entre els elements de negoci i els Uris de l'esquema.
- o bé a partir d'una plantilla on hi ha definides àrees assignades a un URI de l'esquema

En tot cas al sistema genera a partir d'un esquema i opcionalment d'uns Uris amb valor (via OCR o no) un XML que compleix aquell esquema.

Namespace PhSystems.XML.Schema.Parser

```

Friend Delegate Function MoveNextParticle(ByRef Particle As System.Xml.Schema.XmlSchemaParticle, ByVal index As Integer) As Boolean
Public Module helper

    Public Function createXmlElement(ByVal QualifiedName As System.Xml.XmlQualifiedName) As System.Xml.XmlElement
        Return (New System.Xml.XmlDocument).CreateElement(QualifiedName.Name, QualifiedName.Namespace)
    End Function
    Public Function GetObject(ByVal Schema As System.Xml.Schema.XmlSchema, ByVal element As System.Xml.Schema.XmlSchemaElement) As System.Xml.Schema.XmlSchemaElement
        If Not element.RefName Is Nothing Then If Not element.RefName.IsEmpty Then Return Schema.Elements(element.RefName)
        Return element
    End Function
    Public Function GetObject(ByVal Schema As System.Xml.Schema.XmlSchema, ByVal attribute As System.Xml.Schema.XmlSchemaAttribute) As System.Xml.Schema.XmlSchemaAttribute
        If Not attribute.RefName Is Nothing Then If Not attribute.RefName.IsEmpty Then Return Schema.Attributes(attribute.RefName)
        Return attribute
    End Function
    Public Function GetQualifiedName(ByRef Attribute As System.Xml.Schema.XmlSchemaAttribute) As System.Xml.XmlQualifiedName
        If Attribute Is Nothing Then Return New System.Xml.XmlQualifiedName("", "")
        If Not Attribute.Name Is Nothing Then Return Attribute.QualifiedName
        Return Attribute.RefName
    End Function
    Public Function GetQualifiedName(ByRef element As System.Xml.Schema.XmlSchemaElement) As System.Xml.XmlQualifiedName
        If element Is Nothing Then Return New System.Xml.XmlQualifiedName("", "")
        If Not element.Name Is Nothing Then Return element.QualifiedName
        Return element.RefName
    End Function
    Public Function GetObjectType(ByRef _object As Object) As Type
        If _object.GetType.IsPublic Then Return _object.GetType
        Return System.Type.GetType(_object.valuetype.fullname)
    End Function
    Public Overloads Function GetDefaultValue(ByRef Element As System.Xml.Schema.XmlSchemaElement) As String
        If Not Element.FixedValue Is Nothing Then Return Element.FixedValue
        Return Element.DefaultValue
    End Function
    Public Overloads Function GetDefaultValue(ByRef Attribute As System.Xml.Schema.XmlSchemaAttribute) As String
        If Not Attribute.FixedValue Is Nothing Then Return Attribute.FixedValue
        Return Attribute.DefaultValue
    End Function
    Public Function OwnerDocument(ByVal node As System.Xml.XmlNode) As System.Xml.XmlDocument
        If TypeOf node Is System.Xml.XmlDocument Then Return node
        Return node.OwnerDocument
    End Function
    Public Function GetElementSchema(ByVal Schema As System.Xml.Schema.XmlSchema, ByVal elementName As String) As System.Xml.XmlQualifiedName
        If Schema Is Nothing Then Return New System.Xml.XmlQualifiedName
        Dim enumerator As IEnumerator = Schema.Elements.GetEnumerator
        If GetElementSchema(enumerator, elementName) Then Return CType(CType(enumerator.Current, System.Collections.DictionaryEntry).Value, System.Xml.Schema.XmlSchemaElement).QualifiedName
        enumerator = Schema.Items.GetEnumerator
        If GetElementSchema(enumerator, elementName) Then Return CType(enumerator.Current, System.Xml.Schema.XmlSchemaElement).QualifiedName
    
```

```

    If elementName = "" And Not Schema.Elements.Count = 0 Then Return
    CType(CType(PhSystems.Generic.Base.Iterate.Frist(Schema.Elements.GetEnumerator),
    System.Xml.Schema.XmlSchemaElement).QualifiedNames, System.Collections.DictionaryEntry).Value,
    Return New System.Xml.XmlQualifiedName(elementName)

End Function

Private Function GetElementSchema(ByVal enumerator As IEnumerator, ByVal elementName As String) As Boolean

    Dim bol = enumerator.MoveNext
    If bol Then
        Dim objCurrent As Object = enumerator.Current
        If TypeOf objCurrent Is System.Collections.DictionaryEntry Then objCurrent = CType(objCurrent,
    System.Collections.DictionaryEntry).Value()
        bol = TypeOf objCurrent Is System.Xml.Schema.XmlSchemaElement
        If bol Then bol = CType(objCurrent, System.Xml.Schema.XmlSchemaElement).Name = elementName
        If Not bol Then bol = GetElementSchema(enumerator, elementName)
    End If
    Return bol

End Function

End Module

Public Interface ISchemaParse
    Inherits ISchemaParseCreate, ISchemaParseHelper

    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Documentation As System.Xml.Schema.XmlSchemaDocumentation)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Datatype As System.Xml.Schema.XmlSchemaDatatype)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ObjectEnumerator As System.Xml.Schema.XmlSchemaObjectEnumerator)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ObjectCollection As System.Xml.Schema.XmlSchemaObjectCollection)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Collection As System.Xml.Schema.XmlSchemaCollection)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Particle As System.Xml.Schema.XmlSchemaParticle)
    System.Xml.Schema.XmlSchemaCollectionEnumerator
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef sObject As System.Xml.Schema.XmlSchemaObject)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Content As System.Xml.Schema.XmlSchemaContent)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ContentModel As System.Xml.Schema.XmlSchemaContentModel)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Facet As System.Xml.Schema.XmlSchemaFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef NumericFacet As System.Xml.Schema.XmlSchemaNumericFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef IdentityConstraint As System.Xml.Schema.XmlSchemaIdentityConstraint)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef GroupBase As System.Xml.Schema.XmlSchemaGroupBase)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef TypeContent As System.Xml.Schema.XmlSchemaSimpleTypeContent)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Type As System.Xml.Schema.XmlSchemaType)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef External As System.Xml.Schema.XmlSchemaExternal)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef XmlSchema As System.Xml.Schema.XmlSchema, ByVal element As
    System.Xml.XmlQualifiedName)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Annotated As System.Xml.Schema.XmlSchemaAnnotated)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AnyAttribute As System.Xml.Schema.XmlSchemaAnyAttribute)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Attribute As System.Xml.Schema.XmlSchemaAttribute)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AttributeGroup As System.Xml.Schema.XmlSchemaAttributeGroup)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AttributeGroupRef As System.Xml.Schema.XmlSchemaAttributeGroupRef)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexContentExtension As
    System.Xml.Schema.XmlSchemaComplexContentExtension)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexContentRestriction As
    System.Xml.Schema.XmlSchemaComplexContentRestriction)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleContentExtension As
    System.Xml.Schema.XmlSchemaSimpleContentExtension)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleContentRestriction As
    System.Xml.Schema.XmlSchemaSimpleContentRestriction)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexContent As System.Xml.Schema.XmlSchemaComplexContent)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleContent As System.Xml.Schema.XmlSchemaSimpleContent)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef EnumerationFacet As System.Xml.Schema.XmlSchemaEnumerationFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MaxExclusiveFacet As System.Xml.Schema.XmlSchemaMaxExclusiveFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MaxInclusiveFacet As System.Xml.Schema.XmlSchemaMaxInclusiveFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MinExclusiveFacet As System.Xml.Schema.XmlSchemaMinExclusiveFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MinInclusiveFacet As System.Xml.Schema.XmlSchemaMinInclusiveFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef FractionDigitsFacet As
    System.Xml.Schema.XmlSchemaFractionDigitsFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef LengthFacet As System.Xml.Schema.XmlSchemaLengthFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MaxLengthFacet As System.Xml.Schema.XmlSchemaMaxLengthFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MinLengthFacet As System.Xml.Schema.XmlSchemaMinLengthFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef TotalDigitsFacet As System.Xml.Schema.XmlSchemaTotalDigitsFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef PatternFacet As System.Xml.Schema.XmlSchemaPatternFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef WhiteSpaceFacet As System.Xml.Schema.XmlSchemaWhiteSpaceFacet)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Group As System.Xml.Schema.XmlSchemaGroup)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Key As System.Xml.Schema.XmlSchemaKey)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Keyref As System.Xml.Schema.XmlSchemaKeyref)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Unique As System.Xml.Schema.XmlSchemaUnique)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Notation As System.Xml.Schema.XmlSchemaNotation)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Any As System.Xml.Schema.XmlSchemaAny)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Element As System.Xml.Schema.XmlSchemaElement)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ALL As System.Xml.Schema.XmlSchemaALL)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Choice As System.Xml.Schema.XmlSchemaChoice)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Sequence As System.Xml.Schema.XmlSchemaSequence)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef GroupRef As System.Xml.Schema.XmlSchemaGroupRef)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleTypeList As System.Xml.Schema.XmlSchemaSimpleTypeList)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleTypeRestriction As
    System.Xml.Schema.XmlSchemaSimpleTypeRestriction)
    System.Xml.Schema.XmlSchemaSimpleTypeRestriction)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleTypeUnion As System.Xml.Schema.XmlSchemaSimpleTypeUnion)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexType As System.Xml.Schema.XmlSchemaComplexType)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleType As System.Xml.Schema.XmlSchemaSimpleType)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef XPath As System.Xml.Schema.XmlSchemaXPath)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Annotation As System.Xml.Schema.XmlSchemaAnnotation)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AppInfo As System.Xml.Schema.XmlSchemaAppInfo)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Import As System.Xml.Schema.XmlSchemaImport)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Include As System.Xml.Schema.XmlSchemaInclude)
    Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Redefine As System.Xml.Schema.XmlSchemaRedefine)

```

```

End Interface
Public Interface ISchemaParseCreate

    Function onCreateElement(ByRef node As System.Xml.XmlNode, ByRef element As System.xml.schema.XmlSchemaElement) As Boolean
    Function onCreateAttribute(ByRef node As System.Xml.XmlNode, ByRef attribute As System.xml.schema.XmlSchemaAttribute) As Boolean
    Sub onAssingDefaultValue(ByRef node As System.Xml.XmlNode, ByVal sObject As System.xml.schema.XmlSchemaObject)

End Interface
Public Interface ISchemaParseHelper

    Function onSelectChoice(ByRef choice As System.xml.schema.XmlSchemaChoice) As Integer
    Function onMoveNextParticle(ByRef particle As System.xml.schema.XmlSchemaParticle, ByVal index As Integer) As Boolean

End Interface

Public MustInherit Class SchemaParserAbstract
    Implements ISchemaParse

    Protected stakIterating As stak
    Protected node As System.Xml.XmlNode
    Protected schema As System.Xml.Schema.XmlSchema
    Protected NotInheritable Class stak
        Public stak As stak
        Private Enumerator As IEnumerator
        Public Sub New(ByVal stak As stak, ByVal Enumerator As IEnumerator)

            Me.stak = stak
            Me.Enumerator = Enumerator

        End Sub

        Private Function IIRetXPathRefIteration(ByVal defaultValue As String, ByVal values As ArrayList, ByVal inmersio As Integer) As ArrayList

            Select Case inmersio
                Case 0
                    values.Add(defaultValue)
                    Return values
                Case Is > 0
                    Return IIRetXPathRefIteration(PhSystems.Generic.Funcions.Cadenes.BinariSplit(defaultValue, "\SchemaItem", False), values, inmersio - 1)
            End Select

        End Function

        Private Function IRetXPathRefIteration(ByVal defaultValue As String, ByVal values As ArrayList, ByVal inmersio As Integer) As ArrayList

            If Enumerator Is Nothing Then Return Me.IIRetXPathRefIteration(defaultValue, values, 0)
            If Not TypeOf Enumerator.Current Is parsingIteratingObjet And Me.stak Is Nothing Then Return Me.IIRetXPathRefIteration(defaultValue, values, 0)
            If Not TypeOf Enumerator.Current Is parsingIteratingObjet Then Return Me.stak.IIRetXPathRefIteration(defaultValue, values, inmersio + 1)
            Return Me.IIRetXPathRefIteration(defaultValue, Me.stak.IRetXPathRefIteration(defaultValue, values, inmersio + 1), inmersio)

        End Function

        Public Function RetXPathRefIteration(ByVal defaultValue As String) As String()

            Return Me.IRetXPathRefIteration(defaultValue, New ArrayList, 0).ToArray(defaultValue.GetType)

        End Function

        Public Function RetIteracio(ByVal recursiveStak As Boolean) As Integer

            If Enumerator Is Nothing Then Return 1
            If Not TypeOf Enumerator.Current Is parsingIteratingObjet And (Me.stak Is Nothing Or Not recursiveStak) Then Return 1
            If Not TypeOf Enumerator.Current Is parsingIteratingObjet And recursiveStak Then Return Me.stak.RetIteracio(recursiveStak)
            Return CType(Enumerator.Current, parsingIteratingObjet).iteracio

        End Function

        Public Function GetTypeEnumerator() As Type
            If Enumerator Is Nothing Then Return Nothing
            Return CType(Enumerator, Object).GetType
        End Function

    End Class

    Public Sub New()

        Me.stakIterating = New stak(Me.stakIterating, Nothing)

    End Sub

#Region "SchemaParserAbstractiteracio"

    Private NotInheritable Class parsingIteratingObjet

        Private _XSchemaParticle As System.Xml.Schema.XmlSchemaParticle
        Private _iteracio As Integer

        Public Sub New(ByRef XSchemaParticle As System.Xml.Schema.XmlSchemaParticle, ByVal iteracio As Integer)
            _XSchemaParticle = XSchemaParticle
            _iteracio = iteracio
        End Sub
    End Class

```

```

End Sub
Public Function XmlSchemaParticle() As System.Xml.Schema.XmlSchemaParticle
    Return _XSchemaParticle
End Function
Public Function iteracio() As Integer
    Return _iteracio
End Function

End Class
Private NotInheritable Class parsingIteratingEnumerator
    Implements IEnumerator

    Private index As Integer
    Private _HasMoveNext As MoveNextParticle
    Private fLag As Boolean
    Friend Particle As System.Xml.Schema.XmlSchemaParticle

    Public Sub New(ByRef Particle As System.Xml.Schema.XmlSchemaParticle, ByRef HasMoveNext As MoveNextParticle)
        Me.Particle = Particle
        Me._HasMoveNext = HasMoveNext
        index = 0
    End Sub
    Public ReadOnly Property Current() As Object Implements System.Collections.IEnumerator.Current
        Get
            If fLag Then Return New parsingIteratingObjet(Me.Particle, index)
        End Get
    End Property
    Public Function MoveNext() As Boolean Implements System.Collections.IEnumerator.MoveNext

        If _HasMoveNext Is Nothing Then Return False
        fLag = _HasMoveNext(Me.Particle, index)
        If fLag Then index = index + 1
        Return fLag

    End Function
    Public Sub Reset() Implements System.Collections.IEnumerator.Reset
        index = 0
    End Sub

End Class
Private NotInheritable Class parsingIteratingEnumerable
    Implements IEnumerable

    Private _xobject As System.xml.schema.XmlSchemaObject
    Private onHasMoveNext As MoveNextParticle

    Private Sub New(ByRef xobject As System.xml.schema.XmlSchemaObject, ByRef HasMoveNext As MoveNextParticle)
        _xobject = xobject
        onHasMoveNext = HasMoveNext
    End Sub
    Public Function HasMoveNext(ByRef Particle As System.xml.schema.XmlSchemaParticle, ByVal index As Integer) As
Boolean

        If Not index < Particle.MinOccurs And Not onHasMoveNext Is Nothing Then Return onHasMoveNext(Particle, index)
        Return (Particle.MinOccurs > index) And (index <= Particle.MaxOccurs)

    End Function
    Public Shared Function Factory(ByRef Particle As System.Xml.Schema.XmlSchemaParticle, ByRef HasMoveNext As
MoveNextParticle) As Object

        If Particle.MaxOccurs = Particle.MinOccurs And Particle.MinOccurs = 1 Then Return Particle
        Return New parsingIteratingEnumerable(Particle, HasMoveNext).GetEnumerator

    End Function
    Public Shared Function Factory(ByRef subject As Object, ByRef HasMoveNext As MoveNextParticle) As Object

        If subject Is Nothing Then Return Nothing
        If PhSystems.Generic.Base.ClassLoader.IsSubClassOf(subject.GetType,
GetType(System.Xml.Schema.XmlSchemaParticle)) Then Return Factory(CType(subject, System.Xml.Schema.XmlSchemaParticle),
HasMoveNext)
        If TypeOf subject.GetType Is PhSystems.XML.Schema.parser.SchemaParserAbstract.parsingIteratingObjet Then Return
CType(subject, parsingIteratingObjet).XmlSchemaParticle
        Return subject

    End Function
    Public Shared Function GetIteracio(ByRef xobject As Object) As Integer

        If TypeOf xobject Is parsingIteratingObjet Then Return CType(xobject, parsingIteratingObjet).iteracio

    End Function
    Public Function GetEnumerator() As System.Collections.IEnumerator Implements
System.Collections.IEnumerable.GetEnumerator
        Return New parsingIteratingEnumerator(_xobject, AddressOf HasMoveNext)
    End Function

End Class
Private Shared Function FactoryIteration(ByRef subject As Object, ByRef HasMoveNext As MoveNextParticle) As Object

    Return parsingIteratingEnumerable.Factory(subject, HasMoveNext)

End Function

#End Region

#Region "La mara dels ous de la iteracio"

Private Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByVal IterationObject As IEnumerator)

```

```

If IterationObject Is Nothing Then Exit Sub
Me.stakIterating = New stak(Me.stakIterating, IterationObject)
While IterationObject.MoveNext
    If TypeOf IterationObject.Current Is parsingIteratingObjet Then
        CType(Me, SchemaParserAbstract).parsing(node, CType(IterationObject.Current, parsingIteratingObjet))
    ElseIf TypeOf IterationObject.Current Is System.Xml.Schema.XmlSchemaParticle Then
        CType(Me, ISchemaParse).parsing(node, CType(IterationObject.Current, System.Xml.Schema.XmlSchemaParticle))
    Else
        CType(Me, ISchemaParse).parsing(node, IterationObject.Current)
    End If
End While
Me.stakIterating = Me.stakIterating.stak

End Sub
Private Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByVal IterationObject As parsingIteratingObjet)

    If IterationObject Is Nothing Then Exit Sub
    'demandar informacio sobre la iteracio
    If TypeOf IterationObject.XmlSchemaParticle Is System.Xml.Schema.XmlSchemaAny Then CType(Me,
ISchemaParse).parsing(node, CType(IterationObject.XmlSchemaParticle, System.Xml.Schema.XmlSchemaAny)) : Exit Sub
    If TypeOf IterationObject.XmlSchemaParticle Is System.Xml.Schema.XmlSchemaElement Then CType(Me,
ISchemaParse).parsing(node, CType(IterationObject.XmlSchemaParticle, System.Xml.Schema.XmlSchemaElement)) : Exit Sub
    If TypeOf IterationObject.XmlSchemaParticle Is System.Xml.Schema.XmlSchemaAll Then CType(Me,
ISchemaParse).parsing(node, CType(IterationObject.XmlSchemaParticle, System.Xml.Schema.XmlSchemaAll)) : Exit Sub
    If TypeOf IterationObject.XmlSchemaParticle Is System.Xml.Schema.XmlSchemaChoice Then CType(Me,
ISchemaParse).parsing(node, CType(IterationObject.XmlSchemaParticle, System.Xml.Schema.XmlSchemaChoice)) : Exit Sub
    If TypeOf IterationObject.XmlSchemaParticle Is System.Xml.Schema.XmlSchemaSequence Then CType(Me,
ISchemaParse).parsing(node, CType(IterationObject.XmlSchemaParticle, System.Xml.Schema.XmlSchemaSequence)) : Exit Sub
    If TypeOf IterationObject.XmlSchemaParticle Is System.Xml.Schema.XmlSchemaGroupRef Then CType(Me,
ISchemaParse).parsing(node, CType(IterationObject.XmlSchemaParticle, System.Xml.Schema.XmlSchemaGroupRef)) : Exit Sub

End Sub
Private Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByVal IterationObject As parsingIteratingEnumerator)

    parsing(node, CType(IterationObject, IEnumerator))

End Sub

#End Region

#Region "parsing"

Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef XmlSchema As
System.Xml.Schema.XmlSchema, ByVal eElement As System.Xml.XmlQualifiedName) Implements ISchemaParse.parsing

    Me.node = node
    Me.schema = XmlSchema
    If node Is Nothing Then Exit Sub
    If XmlSchema Is Nothing Then Exit Sub
    If XmlSchema.Elements.Item(eElement) Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, CType(XmlSchema.Elements.Item(eElement), System.Xml.Schema.XmlSchemaElement))

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ALL As System.Xml.Schema.XmlSchemaAll)
Implements ISchemaParse.parsing

    Me.node = node
    If node Is Nothing Then Exit Sub
    If ALL Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, ALL.Annotation)
    CType(Me, ISchemaParse).parsing(node, ALL.Items)

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Annotated As
System.Xml.Schema.XmlSchemaAnnotated) Implements ISchemaParse.parsing

    Me.node = node
    If Annotated Is Nothing Then Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaAnyAttribute Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaAnyAttribute)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaAttribute Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaAttribute)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaAttributeGroup Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaAttributeGroup)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaAttributeGroupRef Then CType(Me,
SchemaParserAbstract).parsing(node, CType(Annotated, System.Xml.Schema.XmlSchemaAttributeGroupRef)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaContent Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaContent)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaContentModel Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaContentModel)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaFacet)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaGroup Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaGroup)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaIdentityConstraint Then CType(Me,
SchemaParserAbstract).parsing(node, CType(Annotated, System.Xml.Schema.XmlSchemaIdentityConstraint)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaNotation Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaNotation)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaParticle Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaParticle)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaSimpleTypeContent Then CType(Me,
SchemaParserAbstract).parsing(node, CType(Annotated, System.Xml.Schema.XmlSchemaSimpleTypeContent)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaType Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaType)) : Exit Sub
    If TypeOf Annotated Is System.Xml.Schema.XmlSchemaXPath Then CType(Me, SchemaParserAbstract).parsing(node,
CType(Annotated, System.Xml.Schema.XmlSchemaXPath)) : Exit Sub

```

```

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Annotation As
System.Xml.Schema.XmlSchemaAnnotation) Implements ISchemaParse.parsing

        Me.node = node
        If Annotation Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, Annotation.Items)

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Any As System.Xml.Schema.XmlSchemaAny)
Implements ISchemaParse.parsing

        Me.node = node
        If Any Is Nothing Then Exit Sub
        MsgBox("falta implementar XmlSchemaAny")

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AnyAttribute As
System.Xml.Schema.XmlSchemaAnyAttribute) Implements ISchemaParse.parsing

        Me.node = node
        If AnyAttribute Is Nothing Then Exit Sub
        MsgBox("falta implementar XmlSchemaAnyAttribute")

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AppInfo As
System.Xml.Schema.XmlSchemaAppInfo) Implements ISchemaParse.parsing

        Me.node = node
        If AppInfo Is Nothing Then Exit Sub
        MsgBox("falta implementar XmlSchemaAppInfo")

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Attribute As
System.Xml.Schema.XmlSchemaAttribute) Implements ISchemaParse.parsing

        Me.node = node
        Dim RealAttribute As System.Xml.Schema.XmlSchemaAttribute = GetObject(Me.schema, Attribute)
        If RealAttribute Is Nothing Then Exit Sub
        If OwnerDocument(node) Is Nothing Or GetQualifiedName(RealAttribute).Name = "" Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, RealAttribute.Annotation)
        Me.onCreateAttribute(node, RealAttribute)

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AttributeGroup As
System.Xml.Schema.XmlSchemaAttributeGroup) Implements ISchemaParse.parsing

        Me.node = node
        If AttributeGroup Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, AttributeGroup.Annotation())
        CType(Me, ISchemaParse).parsing(node, AttributeGroup.Attributes)

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef AttributeGroupRef As
System.Xml.Schema.XmlSchemaAttributeGroupRef) Implements ISchemaParse.parsing

        Me.node = node
        If AttributeGroupRef Is Nothing Or Me.schema Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, AttributeGroupRef.Annotation)
        CType(Me, SchemaParserAbstract).parsing(node, Me.schema.AttributeGroups(AttributeGroupRef.RefName))

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Choice As
System.Xml.Schema.XmlSchemaChoice) Implements ISchemaParse.parsing

        Dim mychoice As Integer = Me.onSelectChoice(Choice)
        If Choice Is Nothing Then Exit Sub
        If mychoice > Choice.Items.Count Or 0 > mychoice Then Exit Sub
        Dim [object] As Object = Me.FactoryIteration(Choice.Items(mychoice), AddressOf Me.onMoveNextParticle)
        If TypeOf [object] Is IEnumerable Then CType(Me, SchemaParserAbstract).parsing(node, CType([object], IEnumerable))
        If TypeOf [object] Is System.Xml.Schema.XmlSchemaObject Then CType(Me, SchemaParserAbstract).parsing(node,
CType([object], System.Xml.Schema.XmlSchemaObject))

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ObjectEnumerator As
System.Xml.Schema.XmlSchemaObjectEnumerator) Implements ISchemaParse.parsing

        Me.node = node
        CType(Me, SchemaParserAbstract).parsing(node, CType(ObjectEnumerator, IEnumerable))

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ObjectCollection As
System.Xml.Schema.XmlSchemaObjectCollection) Implements ISchemaParse.parsing

        Me.node = node
        If ObjectCollection Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, ObjectCollection.GetEnumerator())

    End Sub
    Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Collection As
System.Xml.Schema.XmlSchemaCollection) Implements ISchemaParse.parsing

        Me.node = node
        If Collection Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, Collection.GetEnumerator())

    End Sub

```



```

Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef CollectionEnumerator As
System.Xml.Schema.XmlSchemaCollectionEnumerator) Implements ISchemaParse.parsing

    Me.node = node
    CType(Me, SchemaParserAbstract).parsing(node, CType(CollectionEnumerator, IEnumerator))

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexContent As
System.Xml.Schema.XmlSchemaComplexContent) Implements ISchemaParse.parsing

    Me.node = node
    If ComplexContent Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, ComplexContent.Annotation)
    CType(Me, ISchemaParse).parsing(node, ComplexContent.Content)

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexContentExtension As
System.Xml.Schema.XmlSchemaComplexContentExtension) Implements ISchemaParse.parsing

    Me.node = node
    If ComplexContentExtension Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, ComplexContentExtension.Annotation)
    CType(Me, ISchemaParse).parsing(node, ComplexContentExtension.Attributes)

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexContentRestriction As
System.Xml.Schema.XmlSchemaComplexContentRestriction) Implements ISchemaParse.parsing

    Me.node = node
    CType(Me, ISchemaParse).parsing(node, ComplexContentRestriction.Annotation)
    CType(Me, ISchemaParse).parsing(node, ComplexContentRestriction.Particle)
    CType(Me, ISchemaParse).parsing(node, ComplexContentRestriction.Attributes)

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ComplexType As
System.Xml.Schema.XmlSchemaComplexType) Implements ISchemaParse.parsing

    Me.node = node
    If ComplexType Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, ComplexType.Annotation)
    CType(Me, ISchemaParse).parsing(node, ComplexType.Attributes)
    CType(Me, ISchemaParse).parsing(node, ComplexType.Particle)

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Content As
System.Xml.Schema.XmlSchemaContent) Implements ISchemaParse.parsing

    Me.node = node
    If TypeOf Content Is System.Xml.Schema.XmlSchemaComplexContentExtension Then CType(Me, ISchemaParse).parsing(node,
CType(Content, System.Xml.Schema.XmlSchemaComplexContentExtension))
    If TypeOf Content Is System.Xml.Schema.XmlSchemaComplexContentRestriction Then CType(Me, ISchemaParse).parsing(node,
CType(Content, System.Xml.Schema.XmlSchemaComplexContentRestriction))
    If TypeOf Content Is System.Xml.Schema.XmlSchemaSimpleContentExtension Then CType(Me, ISchemaParse).parsing(node,
CType(Content, System.Xml.Schema.XmlSchemaSimpleContentExtension))
    If TypeOf Content Is System.Xml.Schema.XmlSchemaSimpleContentRestriction Then CType(Me, ISchemaParse).parsing(node,
CType(Content, System.Xml.Schema.XmlSchemaSimpleContentRestriction))

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef ContentModel As
System.Xml.Schema.XmlSchemaContentModel) Implements ISchemaParse.parsing

    Me.node = node
    If TypeOf ContentModel Is System.Xml.Schema.XmlSchemaComplexContent Then CType(Me, ISchemaParse).parsing(node,
CType(ContentModel, System.Xml.Schema.XmlSchemaComplexContent))
    If TypeOf ContentModel Is System.Xml.Schema.XmlSchemaSimpleContent Then CType(Me, ISchemaParse).parsing(node,
CType(ContentModel, System.Xml.Schema.XmlSchemaSimpleContent))

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Element As
System.Xml.Schema.XmlSchemaElement) Implements ISchemaParse.parsing

    Me.node = node
    Dim RealElement As System.Xml.Schema.XmlSchemaElement = GetObject(Me.schema, Element)
    If GetQualifiedRealElementName(RealElement).Name = "" Then Exit Sub
    If Not onCreateElement(node, RealElement) Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, RealElement.Annotation)
    If RealElement.ElementType.GetType.IsPublic Then CType(Me, ISchemaParse).parsing(node,
CType(RealElement.ElementType, System.Xml.Schema.XmlSchemaType))
    CType(Me, ISchemaParse).parsing(node, RealElement.Constraints)

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Group As
System.Xml.Schema.XmlSchemaGroup) Implements ISchemaParse.parsing

    Me.node = node
    If Group Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, Group.Annotation)
    CType(Me, ISchemaParse).parsing(node, Group.Particle)

End Sub
Public Overridable OverLoads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef GroupBase As
System.Xml.Schema.XmlSchemaGroupBase) Implements ISchemaParse.parsing

    Me.node = node
    If GroupBase Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, CType(GroupBase, System.Xml.Schema.XmlSchemaParticle))

```

```

    If TypeOf GroupBase Is System.Xml.Schema.XmlSchemaAll Then CType(Me, ISchemaParse).parsing(node, CType(GroupBase,
System.Xml.Schema.XmlSchemaAll))
    If TypeOf GroupBase Is System.Xml.Schema.XmlSchemaChoice Then CType(Me, ISchemaParse).parsing(node, CType(GroupBase,
System.Xml.Schema.XmlSchemaChoice))
    If TypeOf GroupBase Is System.Xml.Schema.XmlSchemaSequence Then CType(Me, ISchemaParse).parsing(node,
CType(GroupBase, System.Xml.Schema.XmlSchemaSequence))

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef GroupRef As
System.Xml.Schema.XmlSchemaGroupRef) Implements ISchemaParse.parsing

        Me.node = node
        If GroupRef Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, GroupRef.Annotation)
        CType(Me, ISchemaParse).parsing(node, GroupRef.Particle)

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Particle As
System.Xml.Schema.XmlSchemaParticle) Implements ISchemaParse.parsing

        Me.node = node
        If Particle Is Nothing Then Exit Sub
        Dim _object As Object = Me.FactoryIteration(Particle, AddressOf Me.onMoveNextParticle)
        If _object Is Nothing Then Exit Sub
        If TypeOf _object Is IEnumerator Then CType(Me, SchemaParserAbstract).parsing(node, CType(_object, IEnumerator)) :
Exit Sub
        If TypeOf _object Is System.Xml.Schema.XmlSchemaAny Then CType(Me, SchemaParserAbstract).parsing(node,
CType(_object, System.Xml.Schema.XmlSchemaAny)) : Exit Sub
        If TypeOf _object Is System.Xml.Schema.XmlSchemaElement Then CType(Me, SchemaParserAbstract).parsing(node,
CType(_object, System.Xml.Schema.XmlSchemaElement)) : Exit Sub
        If PhSystems.Generic.Base.ClassLoader.IsSubclassOf(_object.GetType, GetType(System.Xml.Schema.XmlSchemaGroupBase))
Then CType(Me, SchemaParserAbstract).parsing(node, CType(_object, System.Xml.Schema.XmlSchemaGroupBase)) : Exit Sub
        If TypeOf _object Is System.Xml.Schema.XmlSchemaGroupRef Then CType(Me, SchemaParserAbstract).parsing(node,
CType(_object, System.Xml.Schema.XmlSchemaGroupRef)) : Exit Sub

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Sequence As
System.Xml.Schema.XmlSchemaSequence) Implements ISchemaParse.parsing

        Me.node = node
        If node Is Nothing Then Exit Sub
        If Sequence Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, Sequence.Annotation)
        CType(Me, ISchemaParse).parsing(node, Sequence.Items)

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleContent As
System.Xml.Schema.XmlSchemaSimpleContent) Implements ISchemaParse.parsing

        Me.node = node
        If SimpleContent Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, SimpleContent.Annotation)
        CType(Me, ISchemaParse).parsing(node, SimpleContent.Content)

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleContentExtension As
System.Xml.Schema.XmlSchemaSimpleContentExtension) Implements ISchemaParse.parsing

        Me.node = node
        If SimpleContentExtension Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, SimpleContentExtension.Annotation)
        CType(Me, ISchemaParse).parsing(node, SimpleContentExtension.Attributes)

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleContentRestriction As
System.Xml.Schema.XmlSchemaSimpleContentRestriction) Implements ISchemaParse.parsing

        Me.node = node
        If SimpleContentRestriction Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, SimpleContentRestriction.Annotation)
        CType(Me, ISchemaParse).parsing(node, SimpleContentRestriction.Attributes)
        CType(Me, ISchemaParse).parsing(node, SimpleContentRestriction.Facets)

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleType As
System.Xml.Schema.XmlSchemaSimpleType) Implements ISchemaParse.parsing

        Me.node = node
        If SimpleType Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, SimpleType.Annotation)
        CType(Me, SchemaParserAbstract).parsing(node, SimpleType.Datatype)

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Datatype As
System.Xml.Schema.XmlSchemaDatatype) Implements ISchemaParse.parsing

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Documentation As
System.Xml.Schema.XmlSchemaDocumentation) Implements ISchemaParse.parsing

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef TypeContent As
System.Xml.Schema.XmlSchemaSimpleTypeContent) Implements ISchemaParse.parsing

        Me.node = node
        If TypeContent Is Nothing Then Exit Sub
        CType(Me, ISchemaParse).parsing(node, TypeContent.Annotation)

```

```

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleTypeList As
System.Xml.Schema.XmlSchemaSimpleTypeList) Implements ISchemaParse.parsing

    Me.node = node
    If SimpleTypeList Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, SimpleTypeList.Annotation)
    CType(Me, ISchemaParse).parsing(node, SimpleTypeList.ItemType)

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleTypeRestriction As
System.Xml.Schema.XmlSchemaSimpleTypeRestriction) Implements ISchemaParse.parsing

    Me.node = node
    If SimpleTypeRestriction Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, SimpleTypeRestriction.Annotation)
    CType(Me, ISchemaParse).parsing(node, SimpleTypeRestriction.BaseType)
    CType(Me, ISchemaParse).parsing(node, SimpleTypeRestriction.Facets)

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef SimpleTypeUnion As
System.Xml.Schema.XmlSchemaSimpleTypeUnion) Implements ISchemaParse.parsing

    Me.node = node
    If SimpleTypeUnion Is Nothing Then Exit Sub
    CType(Me, ISchemaParse).parsing(node, SimpleTypeUnion.Annotation)
    CType(Me, ISchemaParse).parsing(node, SimpleTypeUnion.BaseTypes)

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Notation As
System.Xml.Schema.XmlSchemaNotation) Implements ISchemaParse.parsing

    Me.node = node

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Type As
System.Xml.Schema.XmlSchemaType) Implements ISchemaParse.parsing

    Me.node = node
    If TypeOf Type Is System.Xml.Schema.XmlSchemaComplexType Then CType(Me, ISchemaParse).parsing(node, CType(Type,
System.Xml.Schema.XmlSchemaComplexType))
    If TypeOf Type Is System.Xml.Schema.XmlSchemaSimpleType Then CType(Me, ISchemaParse).parsing(node, CType(Type,
System.Xml.Schema.XmlSchemaSimpleType))

End Sub
Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef sObject As
System.Xml.Schema.XmlSchemaObject) Implements ISchemaParse.parsing

    Me.node = node
    If TypeOf sObject Is System.Xml.Schema.XmlSchema Then CType(Me, SchemaParserAbstract).parsing(node, CType(sObject,
System.Xml.Schema.XmlSchema)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAnnotated Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAnnotated)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleTypeList Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaSimpleTypeList)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAnyAttribute Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAnyAttribute)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAttribute Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAttribute)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAttributeGroup Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAttributeGroup)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAttributeGroupRef Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAttributeGroupRef)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaContent Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaContent)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaComplexContentExtension Then CType(Me,
SchemaParserAbstract).parsing(node, CType(sObject, System.Xml.Schema.XmlSchemaComplexContentExtension)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaComplexContentRestriction Then CType(Me,
SchemaParserAbstract).parsing(node, CType(sObject, System.Xml.Schema.XmlSchemaComplexContentRestriction)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleContentExtension Then CType(Me,
SchemaParserAbstract).parsing(node, CType(sObject, System.Xml.Schema.XmlSchemaSimpleContentExtension)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleContentRestriction Then CType(Me,
SchemaParserAbstract).parsing(node, CType(sObject, System.Xml.Schema.XmlSchemaSimpleContentRestriction)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaContentModel Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaContentModel)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaComplexContent Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaComplexContent)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleContent Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaSimpleContent)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaEnumerationFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaEnumerationFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaMaxExclusiveFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaMaxExclusiveFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaMaxInclusiveFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaMaxInclusiveFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaMinExclusiveFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaMinExclusiveFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaMinInclusiveFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaMinInclusiveFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaNumericFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaNumericFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaFractionDigitsFacet Then CType(Me,
SchemaParserAbstract).parsing(node, CType(sObject, System.Xml.Schema.XmlSchemaFractionDigitsFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaLengthFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaLengthFacet)) : Exit Sub

```

```

    If TypeOf sObject Is System.Xml.Schema.XmlSchemaMaxLengthFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaMaxLengthFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaMinLengthFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaMinLengthFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaTotalDigitsFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaTotalDigitsFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaPatternFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaPatternFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaWhiteSpaceFacet Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaWhiteSpaceFacet)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaGroup Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaGroup)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaIdentityConstraint Then CType(Me,
SchemaParserAbstract).parsing(node, CType(sObject, System.Xml.Schema.XmlSchemaIdentityConstraint)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaKey Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaKey)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaKeyref Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaKeyref)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaUnique Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaUnique)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaNotation Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaNotation)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaParticle Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaParticle)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAny Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAny)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaElement Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaElement)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaGroupBase Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaGroupBase)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaALL Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaALL)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaChoice Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaChoice)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSequence Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaSequence)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaGroupRef Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaGroupRef)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleTypeContent Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaSimpleTypeContent)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleTypeList Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaSimpleTypeList)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleTypeRestriction Then CType(Me,
SchemaParserAbstract).parsing(node, CType(sObject, System.Xml.Schema.XmlSchemaSimpleTypeRestriction)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleTypeUnion Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaSimpleTypeUnion)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaType Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaType)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaComplexType Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaComplexType)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaSimpleType Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaSimpleType)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaXPath Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaXPath)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAnnotation Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAnnotation)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaAppInfo Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaAppInfo)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaDocumentation Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaDocumentation)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaExternal Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaExternal)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaImport Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaImport)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaInclude Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaInclude)) : Exit Sub
    If TypeOf sObject Is System.Xml.Schema.XmlSchemaRedefine Then CType(Me, SchemaParserAbstract).parsing(node,
CType(sObject, System.Xml.Schema.XmlSchemaRedefine)) : Exit Sub

```

End Sub

#Region "nosequefer"

```

Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef EnumerationFacet As
System.Xml.Schema.XmlSchemaEnumerationFacet) Implements ISchemaParse.parsing
    Me.node = node

```

End Sub

```

Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Facet As
System.Xml.Schema.XmlSchemaFacet) Implements ISchemaParse.parsing
    Me.node = node

```

End Sub

```

Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef FractionDigitsFacet As
System.Xml.Schema.XmlSchemaFractionDigitsFacet) Implements ISchemaParse.parsing
    Me.node = node

```

End Sub

```

Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef PatternFacet As
System.Xml.Schema.XmlSchemaPatternFacet) Implements ISchemaParse.parsing
    Me.node = node

```

End Sub

```

Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef IdentityConstraint As
System.Xml.Schema.XmlSchemaIdentityConstraint) Implements ISchemaParse.parsing
    Me.node = node

```

```

    End Sub
    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Key As System.Xml.Schema.XmlSchemaKey)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Keyref As System.Xml.Schema.XmlSchemaKeyref)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef NumericFacet As System.Xml.Schema.XmlSchemaNumericFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef LengthFacet As System.Xml.Schema.XmlSchemaLengthFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MaxExclusiveFacet As System.Xml.Schema.XmlSchemaMaxExclusiveFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MaxInclusiveFacet As System.Xml.Schema.XmlSchemaMaxInclusiveFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MaxLengthFacet As System.Xml.Schema.XmlSchemaMaxLengthFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MinExclusiveFacet As System.Xml.Schema.XmlSchemaMinExclusiveFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MinInclusiveFacet As System.Xml.Schema.XmlSchemaMinInclusiveFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef MinLengthFacet As System.Xml.Schema.XmlSchemaMinLengthFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef TotalDigitsFacet As System.Xml.Schema.XmlSchemaTotalDigitsFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Unique As System.Xml.Schema.XmlSchemaUnique)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef WhiteSpaceFacet As System.Xml.Schema.XmlSchemaWhiteSpaceFacet)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef XPath As System.Xml.Schema.XmlSchemaXPath)
    Implements ISchemaParse.parsing
        Me.node = node
    End Sub

#End Region

#Region "extern"

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Redefine As System.Xml.Schema.XmlSchemaRedefine)
    Implements ISchemaParse.parsing
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef External As System.Xml.Schema.XmlSchemaExternal)
    Implements ISchemaParse.parsing
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Import As System.Xml.Schema.XmlSchemaImport)
    Implements ISchemaParse.parsing
    End Sub

    Public Overridable Overloads Sub parsing(ByVal node As System.Xml.XmlNode, ByRef Include As System.Xml.Schema.XmlSchemaInclude)
    Implements ISchemaParse.parsing
    End Sub

#End Region

```

```

#End Region

#Region "ISchemaParseHelper"

    Protected MustOverride Function onMoveNextParticle(ByRef Particle As System.Xml.Schema.XmlSchemaParticle, ByVal index As Integer) As Boolean Implements ISchemaParseHelper.onMoveNextParticle
    Protected MustOverride Function onSelectChoice(ByRef Choice As System.Xml.Schema.XmlSchemaChoice) As Integer Implements ISchemaParseHelper.onSelectChoice

#End Region

#Region "ISchemaParsecreate"

    Protected MustOverride Sub onAssingDefaultValue(ByRef node As System.Xml.XmlNode, ByVal sObject As System.Xml.Schema.XmlSchemaObject) Implements ISchemaParseCreate.onAssingDefaultValue
    Protected MustOverride Function onCreateElement(ByRef node As System.Xml.XmlNode, ByRef element As System.Xml.Schema.XmlSchemaElement) As Boolean Implements ISchemaParseCreate.onCreateElement
    Protected MustOverride Function onCreateAttribute(ByRef node As System.Xml.XmlNode, ByRef attribute As System.Xml.Schema.XmlSchemaAttribute) As Boolean Implements ISchemaParseCreate.onCreateAttribute

#End Region

    End Class

End Namespace

```

7.2. Administració i adquisició de documents

7.2.1. Escaneig d'imatges

Per realitzar l'adquisició d'imatges, es va optar per crear una iteració, sobre una classe de generació de documents electrònics. Per tant la classe final que retorna les imatges també és una iteració, ja sigui sobre fitxers o sobre Twain. La implementació de la classe Twain està feta en C#, i és una encapsulació en objectes de la classe d'exemple que ve amb la documentació de l'especificació Twain.

```
using System;
using System.Collections;
using System.Runtime.InteropServices;
using System.Windows.Forms;

namespace PhSystems.Twain
{
    public interface ITwainCommand
    {
        void TransferReady(TwSourceManager sender);
        void CloseRequest(TwSourceManager sender);
        void CloseOk(TwSourceManager sender);
        void DeviceEvent(TwSourceManager sender);
        void Failure(TwSourceManager sender);
        void Cancel(TwSourceManager sender);
    }

    public interface ITwainForm:ITwainCommand
    {
        IntPtr hwndp();
    }

    public abstract class TwImage: IDisposable
    {
        internal IntPtr hbitmap;
        ~TwImage()
        {
            this.Dispose();
        }

        protected TwImage(IntPtr hbitmap)
        {
            this.hbitmap= hbitmap;
        }

        public System.Drawing.Image GetImage()
        {
            //IntPtr punterImatge=GlobalLock(hbitmap);
            //return GetImage(GetBitmapRectangle(punterImatge),punterImatge,GetPixelInfo( punterImatge));
            return DIBtoImage(GlobalLock(hbitmap));
        }

        public void Dispose()
        {
            if (hbitmap!=IntPtr.Zero)
                Marshal.FreeHGlobal(hbitmap);
        }

        private static System.Drawing.Image DIBtoImage(IntPtr dib_data)
        {
            int ncolors;
            int imagesize;
            int[] palette;
            byte[] imagebits;
            int bytesPerLine;

            TwBitmapInfoHeader bmi;
            System.Drawing.Bitmap bmp;
            System.Drawing.Imaging.BitmapData bits;
            System.Drawing.Imaging.ColorPalette pal;

            bmi = (TwBitmapInfoHeader)Marshal.PtrToStructure(dib_data, typeof(TwBitmapInfoHeader));
            bytesPerLine = (int)((((bmi.biWidth * bmi.biBitCount) + 31) & ~31) >> 3);
```

```

ncolors = (int)bmi.biClrUsed;
if (ncolors == 0)
    if (bmi.biBitCount < 24)
        ncolors = (int)(1 << bmi.biBitCount);
imagesize = (int)bmi.biSizeImage;
if (imagesize == 0)
    imagesize = bytesPerLine * bmi.biHeight;
switch(bmi.biBitCount)
{
    case 1:
        { // Monochrome
            bmp = new System.Drawing.Bitmap(bmi.biWidth, bmi.biHeight,
System.Drawing.Imaging.PixelFormat.Format1bppIndexed);
            bits = bmp.LockBits(new System.Drawing.Rectangle(0, 0, bmp.Width,
bmp.Height), System.Drawing.Imaging.ImageLockMode.WriteOnly, System.Drawing.Imaging.PixelFormat.Format1bppIndexed);
            palette = new int[2];
            break;
        }
    case 4:
        { // 4bpp
            bmp = new System.Drawing.Bitmap(bmi.biWidth, bmi.biHeight,
System.Drawing.Imaging.PixelFormat.Format4bppIndexed);
            bits = bmp.LockBits(new System.Drawing.Rectangle(0, 0, bmp.Width,
bmp.Height), System.Drawing.Imaging.ImageLockMode.WriteOnly, System.Drawing.Imaging.PixelFormat.Format4bppIndexed);
            palette = new int[16];
            break;
        }
    case 8:
        { // 8bpp
            bmp = new System.Drawing.Bitmap(bmi.biWidth, bmi.biHeight,
System.Drawing.Imaging.PixelFormat.Format8bppIndexed);
            bits = bmp.LockBits(new System.Drawing.Rectangle(0, 0, bmp.Width,
bmp.Height), System.Drawing.Imaging.ImageLockMode.WriteOnly, System.Drawing.Imaging.PixelFormat.Format8bppIndexed);
            palette = new int[256];
            break;
        }
    case 24:
    case 32:
        { // 32bpp
            bmp = new System.Drawing.Bitmap(bmi.biWidth, bmi.biHeight,
System.Drawing.Imaging.PixelFormat.Format32bppArgb);
            bits = bmp.LockBits(new System.Drawing.Rectangle(0, 0, bmp.Width,
bmp.Height), System.Drawing.Imaging.ImageLockMode.WriteOnly, System.Drawing.Imaging.PixelFormat.Format32bppArgb);
            palette = new int[0];
            break;
        }
    default:
        {
            throw new Exception("Unexpected number of bits:" +
bmi.biBitCount.ToString());
        }
}
if (bmi.biBitCount < 24)
{
    pal = bmp.Palette; // Managed palette
Marshal.Copy((IntPtr)((int)dib_data + Marshal.SizeOf(typeof(TwBitmapInfoHeader))),
palette, 0, palette.Length);
    for (int i = 0; i < ncolors; i++)
        pal.Entries[i] = System.Drawing.Color.FromArgb(palette[i]);
}
}

```



```

        imagebits = new byte[bytesPerLine];
        for (int y = 0; y < bmi.biHeight; y++)
        {
            // Copy from source to managed
            Marshal.Copy((IntPtr)((int)dib_data + Marshal.SizeOf(typeof(TwBitmapInfoHeader)) +
palette.Length * 4 + imagebits.Length * y), imagebits, 0, imagebits.Length);

            // Copy from managed to dest
            Marshal.Copy(imagebits, 0, (IntPtr)((int)bits.Scan0 + bits.Stride * (bmi.biHeight - 1
- y)), imagebits.Length);
        }

        /*imagebits = new byte[imagesize];
        // Copy from source to managed
        Marshal.Copy((IntPtr)((int)dib_data + Marshal.SizeOf(typeof(TwBitmapInfoHeader)) +
palette.Length * 4), imagebits, 0, imagebits.Length);

        // Copy from managed to dest
        Marshal.Copy(imagebits, 0, bits.Scan0, imagebits.Length);*/
        bmp.UnlockBits(bits);
        return bmp;
    }

    private static System.Drawing.Image GetImage(System.Drawing.Rectangle bmprect, IntPtr bminfo, IntPtr
pixdat)
    {
        System.Drawing.Image me = new System.Drawing.Bitmap(bmprect.Width, bmprect.Height);
        System.Drawing.Graphics g = System.Drawing.Graphics.FromImage(me);
        IntPtr hdc = g.GetHdc();
        SetDIBitsToDevice(hdc, 0, 0, bmprect.Width, bmprect.Height, 0, 0, 0, bmprect.Height, pixdat,
bminfo, 0);

        g.ReleaseHdc(hdc);
        return me;
    }

    private static System.Drawing.Rectangle GetBitmapRectangle( IntPtr bmpptr )
    {
        System.Drawing.Rectangle bmprect=new System.Drawing.Rectangle();
        TwBitmapInfoHeader bmi=new TwBitmapInfoHeader();
        Marshal.PtrToStructure(bmpptr,bmi);
        bmprect.X = 0;bmprect.Y = 0;
        bmprect.Width = bmi.biWidth;
        bmprect.Height = bmi.biHeight;
        return bmprect;
    }

    private static IntPtr GetPixelInfo( IntPtr bmpptr )
    {
        System.Drawing.Rectangle bmprect=new System.Drawing.Rectangle();
        TwBitmapInfoHeader bmi=new TwBitmapInfoHeader();
        Marshal.PtrToStructure(bmpptr,bmi);
        bmprect.X = 0;bmprect.Y = 0;
        bmprect.Width = bmi.biWidth;
        bmprect.Height = bmi.biHeight;
        if( bmi.biSizeImage == 0 )
            bmi.biSizeImage = (((bmi.biWidth * bmi.biBitCount) + 31) & ~31) >> 3) *
bmi.biHeight;

        int p = bmi.biClrUsed;
        if( (p == 0) && (bmi.biBitCount <= 8) )
            p = 1 << bmi.biBitCount;
        p = (p * 4) + bmi.biSize + (int) bmpptr;
        return (IntPtr) p;
    }

    [DllImport("kernel32.dll", ExactSpelling=true)]
    private static extern IntPtr GlobalLock( IntPtr handle );
    [DllImport("gdi32.dll", ExactSpelling=true)]
    private static extern int SetDIBitsToDevice( IntPtr hdc, int xdst, int ydst,int width, int height, int
xsrc, int ysrc, int start, int lines,IntPtr bitsptr, IntPtr bmpptr, int color );
}

public sealed class TwTransferPicture:TwImage
{
    internal TwRC rc;
    internal TwTransferPicture():base(IntPtr.Zero)
    {
    }
}

```

```

        this.rc=TwRC.Cancel;
    }
    public IntPtr Gethbitmap()
    {
        return this.hbitmap;
    }
}

public abstract class TwBase
{
    private TwIdentity appid;
    private TwIdentity srcds;
    private TwError GlobalStatus;

    private bool IsEnableDS;
    private bool IsOpenDSM;
    private bool IsOpenDS;
    private bool IsTranfering;

    public TwBase()
    {
        appid = new TwIdentity();
        appid.Id = IntPtr.Zero;
        appid.Version.MajorNum = 1;
        appid.Version.MinorNum = 1;
        appid.Version.Language = (short)TWCY.SPAIN;
        appid.Version.Country = (short)TWLG.CATALAN;
        appid.Version.Info = "LCR 1";
        appid.ProtocolMajor = TwProtocol.Major;
        appid.ProtocolMinor = TwProtocol.Minor;
        appid.SupportedGroups = (int)(TwDG.Image | TwDG.Control);
        appid.Manufacturer = "PhSystems";
        appid.ProductFamily = "Not Freeware";
        appid.ProductName = "PhTwain";
        srcds = new TwIdentity();
        srcds.Id = IntPtr.Zero;
    }

    private void SetGlobalStatus(TwRC rc, TwTriplet tr, Object target)
    {
        if (rc!=TwRC.Success && rc!=TwRC.XferDone)
        {
            GlobalStatus=new TwError(tr,TwCC.Success,target);
            if (DSMstatus(appid,srcds,TwDG.Control,TwDAT.Status,TwMSG.Get,GlobalStatus.Status)!=TwRC.Success)
                GlobalStatus=new TwError(tr,TwCC.OperationError,null);
        }
        if (GlobalStatus==null)
            GlobalStatus=new TwError(tr,TwCC.Success,target);
    }

    protected internal TwStatus GetGlobalStatus()
    {
        return GlobalStatus.Status;
    }

    protected internal TwRC DataSourceCapability(TwMSG MSG,TwCapability cap)
    {
        TwRC rc = DSscap(appid, srcds, TwDG.Control, TwDAT.Capability, MSG, cap);
        SetGlobalStatus(rc,new TwTriplet(TwDG.Control, TwDAT.Capability, MSG),cap);
        return rc;
    }

    protected internal TwRC DataSourceManagerParent(TwMSG MSG,IntPtr hwndp)
    {
        TwRC rc=TwRC.Success;
        if( !(TwMSG.CloseDSM==MSG && appid.Id == IntPtr.Zero))
            rc= DSMparent(appid, IntPtr.Zero, TwDG.Control, TwDAT.Parent, MSG, ref hwndp );
        if (TwMSG.CloseDSM==MSG)
            appid.Id = IntPtr.Zero;
        if (MSG==TwMSG.OpenDSM && rc==TwRC.Success)
            IsOpenDSM=true;
        if (MSG==TwMSG.CloseDSM && rc==TwRC.Success)
            IsOpenDSM=false;
        SetGlobalStatus(rc,new TwTriplet(TwDG.Control, TwDAT.Parent, MSG),null);
        return rc;
    }

    protected internal TwRC DataSourceManagerUserInfo(TwMSG MSG,IntPtr hwnd)

```

```

{
    TwUserInterface    guif = new TwUserInterface();
    if (MSG!=TwMSG.DisableDS)
        guif.Initialize(hwnd);
    TwRC rc = DSuserif( appid, srcds, TwDG.Control, TwDAT.UserInterface,MSG, guif );
    if (MSG==TwMSG.EnableDS && rc==TwRC.Success)
        IsEnableDS=true;
    if (MSG==TwMSG.DisableDS && rc==TwRC.Success)
        IsEnableDS=false;
    SetGlobalStatus(rc,new TwTripLet(TwDG.Control, TwDAT.UserInterface,MSG),NULL);
    return rc;
}

protected internal TwRC DataSourceManagerIdentity(TwMSG MSG)
{
    try
    {
        TwRC rc=DSMident(appid, IntPtr.Zero, TwDG.Control, TwDAT.Identity, MSG, srcds);
        if (MSG==TwMSG.OpenDS && rc==TwRC.Success)
            IsOpenDS=true;
        if (MSG==TwMSG.CloseDS && rc==TwRC.Success)
            IsOpenDS=false;
        SetGlobalStatus(rc,new TwTripLet(TwDG.Control, TwDAT.Identity, MSG),NULL);
        return rc;
    }
    catch {}
    return TwRC.Failure;
}

protected internal TwRC DataSourceManagerImageInfo(TwMSG operation, TwImageInfo ImageInfo)
{
    TwRC rc = DSiinf( appid, srcds, TwDG.Image, TwDAT.ImageInfo, operation, ImageInfo );
    SetGlobalStatus(rc,new TwTripLet(TwDG.Image, TwDAT.ImageInfo, operation),ImageInfo);
    return rc;
}

protected internal TwRC DataSourceManagerImageTransfer(ref IntPtr hbitmap)
{
    TwRC rc = DSixer( appid, srcds, TwDG.Image, TwDAT.ImageNativeXfer, TwMSG.Get, ref hbitmap );
    SetGlobalStatus(rc,new TwTripLet(TwDG.Image, TwDAT.ImageNativeXfer, TwMSG.Get),NULL);
    return rc;
}

protected internal TwRC DataSourceManagerPendingInfo(TwMSG MSG,ref TwPendingXfers pxfr)
{
    TwRC rc = DSpxfer( appid, srcds, TwDG.Control, TwDAT.PendingXfers, MSG, pxfr );
    if(MSG==TwMSG.Reset || MSG==TwMSG.EndXfer)
        this.IsTranfering= (MSG==TwMSG.EndXfer && pxfr.Count!=0 && (rc==TwRC.Success ||
rc==TwRC.XferDone));
    //
        this.IsTranfering!=(rc==TwRC.Success || rc==TwRC.XferDone)&& pxfr.Count!=0;
    SetGlobalStatus(rc,new TwTripLet(TwDG.Control, TwDAT.PendingXfers, MSG),pxfr);
    return rc;
}

protected internal TwRC DataSourceManagerEvent(ref TwEvent evtmsg)
{
    evtmsg.Message = 0;
    return DSevent(appid, srcds, TwDG.Control, TwDAT.Event, TwMSG.ProcessEvent, ref evtmsg);
}

protected internal bool IsTWTranfering()
{
    return IsTranfering;
}

protected internal bool IsTWEnablesDS()
{
    return IsEnableDS;
}

protected internal bool IsTWOpenedDSM()
{
    return IsOpenDSM;
}

protected internal bool IsTWOpenedDS()
{
    return IsOpenDS;
}

```

```

    }

    protected internal bool IsApplicationAvailable()
    {
        return appid.Id != IntPtr.Zero;
    }

    protected internal bool IsDataSourceAvailable()
    {
        return srcds.Id != IntPtr.Zero;
    }

    internal TwainCommand PassMessage(TwRC rc, short Message)
    {
        if( Message == (short) TwMSG.XFerReady)
        {
            this.IsTranfering=true;
            return TwainCommand.TransferReady;
        }
        if( Message == (short) TwMSG.CloseDSReq)
            return TwainCommand.CloseRequest;
        if( rc==TwRC.Cancel )
            return TwainCommand.Cancel;
        if( rc==TwRC.Failure)
            return TwainCommand.Failure;
        if( Message == (short) TwMSG.CloseDSOK || Message == (short) TwMSG.ClosedS)
            return TwainCommand.CloseOk;
        if( Message == (short) TwMSG.DeviceEvent )
            return TwainCommand.DeviceEvent;
        if( rc == TwRC.NotDSEvent )
            return TwainCommand.Not;
        return TwainCommand.Null;
    }

    public static int ScreenBitDepth
    {
        get
        {
            IntPtr screenDC = CreateDC( "DISPLAY", null, null, IntPtr.Zero );
            int bitDepth = GetDeviceCaps( screenDC, 12 );
            bitDepth *= GetDeviceCaps( screenDC, 14 );
            DeleteDC( screenDC );
            return bitDepth;
        }
    }

    public static bool CheckForTwain()
    {
        string sbWinDir = "";
        // 1. Look in WIN.INI for "[TWAIN]"
        bool bTWAINFound = false;
        if (System.IO.File.Exists(System.IO.Path.Combine(sbWinDir.ToString().Trim(), "win.ini")))
        {
            System.IO.StreamReader reader =
            System.IO.File.OpenText(System.IO.Path.Combine(sbWinDir.ToString().Trim(), "win.ini"));
            String strLine = reader.ReadLine();
            while (strLine != null && !bTWAINFound)
            {
                bTWAINFound = (strLine.IndexOf("[TWAIN]", 0, strLine.Length) >= 0);
                if (!bTWAINFound)
                {
                    strLine = reader.ReadLine();
                }
                // read the next line
            }
            reader.Close();
        }
        // 2. Look for TWAIN_32.DLL
        return (bTWAINFound
            System.IO.File.Exists(System.IO.Path.Combine(System.IO.Directory.GetParent(System.Reflection.Assembly.GetExecutingAssembly().Location).ToString(), "TWAIN_32.DLL")) || System.IO.File.Exists(System.IO.Path.Combine(sbWinDir.ToString().Trim(), "TWAIN_32.DLL")) || System.IO.File.Exists(System.IO.Path.Combine(System.Environment.SystemDirectory, "TWAIN_32.DLL")));
    }

    public TwIdentity GetIdentity()
    {
        return srcds;
    }

    public TwError GetLastError()
    {
        return this.GlobalStatus;
    }

```

```

        #region DSM entry point
        // ----- DSM entry point DAT_ variants:
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSMparent( [In, Out] TwIdentity origin, IntPtr zeroPtr, TwDG dg, TwDAT dat,
TwMSG msg, ref IntPtr
        refptr );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSMident( [In, Out] TwIdentity origin, IntPtr zeroPtr, TwDG dg, TwDAT dat, TwMSG
msg, [In, Out] TwIdentity
        idds );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSMstatus( [In, Out] TwIdentity origin, [In, Out] TwIdentity dest, TwDG dg,
TwDAT dat, TwMSG msg, [In, Out] TwStatus
        dsmstat );
        // ----- DSM entry point DAT_ variants to DS:
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSuserif( [In, Out] TwIdentity origin, [In, Out] TwIdentity dest, TwDG dg, TwDAT
dat, TwMSG msg, TwUserInterface
        guif );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSevent( [In, Out] TwIdentity origin, [In, Out] TwIdentity dest, TwDG dg, TwDAT
dat, TwMSG msg, ref TwEvent
        evt );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSstatus( [In, Out] TwIdentity origin, [In] TwIdentity dest, TwDG dg, TwDAT dat,
TwMSG msg, [In, Out]
        TwStatus dsmstat );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DScap( [In, Out] TwIdentity origin, [In] TwIdentity dest, TwDG dg, TwDAT dat,
TwMSG msg, [In, Out]
        TwCapability capa );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSiinf( [In, Out] TwIdentity origin, [In] TwIdentity dest, TwDG dg, TwDAT dat,
TwMSG msg, [In, Out]
        TwImageInfo imginf );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSixer( [In, Out] TwIdentity origin, [In] TwIdentity dest, TwDG dg, TwDAT dat,
TwMSG msg, ref IntPtr
        hbitmap );
        [DllImport("twain_32.dll", EntryPoint="#1")]
        private static extern TwRC DSpxfer( [In, Out] TwIdentity origin, [In] TwIdentity dest, TwDG dg, TwDAT dat,
TwMSG msg, [In, Out]
        TwPendingXfers pxfr );
        [DllImport("gdi32.dll", ExactSpelling=true)]
        private static extern int GetDeviceCaps( IntPtr hdc, int nIndex );
        [DllImport("gdi32.dll", CharSet=CharSet.Auto)]
        private static extern IntPtr CreateDC( string szdriver, string szdevice, string szoutput, IntPtr devmode );
        [DllImport("gdi32.dll", ExactSpelling=true)]
        private static extern bool DeleteDC( IntPtr hdc );
        #endregion
    }

    public abstract class TwBaseManager:TwBase,IMessageFilter
    {
        private WINMSG            winmsg;
        private TwEvent           evtmsg;
        private bool               isDisable;
        private System.Collections.ArrayList Capabilities;

        protected abstract void Finish();
        protected abstract IntPtr Gethwnd();
        protected abstract bool IsPendingXfers();
        //protected abstract TwPendingXfers PendingXfers();
        protected abstract bool FilterMessage(TwainCommand cmd);

        protected abstract bool TWTransferImage(ref IntPtr Image);
        protected abstract bool TWOpenTransferImage();
        protected abstract bool TWCloseTransferImage(bool IsCancel);
        protected abstract bool TWOpenDSM();
        protected abstract bool TWCloseDSM();
        protected abstract bool TWOpenDS();
        protected abstract bool TWCloseDS();
        protected abstract bool TWDisableDS();
        protected abstract bool TWGetIdentity();
        protected abstract bool TWEnableDS(bool ShowUI);
        protected abstract bool TWSetCapabilities(TwCapability[] Capabilities);

        public class ITwTransferPictureEnumerator:IEnumerator
        {
            private TwBaseManager Host;
            private TwTransferPicture image;
            internal ITwTransferPictureEnumerator(TwBaseManager Host)
            {
                this.Host=Host;
                this.Host.TWOpenTransferImage();
            }

            #region Miembros de IEnumerator
            public void Reset()
            {
                image=null;
                this.Host.TWCloseTransferImage(true);
            }

            public object Current
            {
                get
                {
                    return image;
                }
            }
        }
    }

```

```

    }
}

public bool MoveNext()
{
    image=null;
    if (this.Host.IsPendingXfers())
        image= (TwTransferPicture)this.TransferPicture();
    if (image!=null)
        return (image.rc==TwRC.Success);
    return false;
}

#endregion

private TwImage TransferPicture()
{
    TwTransferPicture rc=new TwTransferPicture();
    TwImageInfo iinf = new TwImageInfo();
    rc.rc=TwRC.Failure;
    if (this.Host.IsPendingXfers())
    {
        rc.rc=this.Host.DataSourceManagerImageInfo(TwMSG.Get, iinf);
        if(this.Host.onfilterMessage(rc.rc, TwMSG.Null))
            if (!this.Host.TWTransferImage(ref rc.hbitmap))
                rc.rc=TwRC.Failure;
        if (!this.Host.TWCloseTransferImage(!rc.rc==TwRC.Success //
rc.rc==TwRC.XferDone)))
            rc.rc=TwRC.Failure;
        return rc;
    }
    this.Host.onfilterMessage(rc.rc, TwMSG.Null);
    return rc;
}

public TwImage CurrentPicture()
{
    return (TwImage)this.Current;
}

public TwainCommand Error()
{
    if (image!=null)
        return ((TwBase)this.Host).PassMessage(image.rc, (short)TwMSG.Null);
    return ((TwBase)this.Host).PassMessage((TwRC.Failure), (short)TwMSG.Null);
}
}

protected TwBaseManager()
{
    evtmsg.EventPtr = Marshal.ALLOCGlobal(Marshal.SizeOf(winmsg));
    Capabilities=new System.Collections.ArrayList();
    Capabilities.Add(new TwCapability(TwCap.XFERCOUNT, (short)-1 ));
}

~TwBaseManager()
{
    this.Finish();
    if (evtmsg.EventPtr!=IntPtr.Zero)
        Marshal.FreeHGlobal(evtmsg.EventPtr);
    if (Capabilities!=null)
    {
        System.Collections.IEnumerator enumerator=Capabilities.GetEnumerator();
        while (enumerator.MoveNext())
            ((TwCapability)enumerator.Current).Dispose();
        this.Capabilities.Clear();
    }
}

bool IMessageFilter.PreFilterMessage(ref Message m )
{
    return this.FilterMessage(this.MessageToTwCommand(ref m));
}

private TwainCommand MessageToTwCommand(ref Message m)
{
    int pos = GetMessagePos();

```

```

winmsg.hwnd          = m.Hwnd;
winmsg.message       = m.Msg;
winmsg.wParam        = m.WParam;
winmsg.lParam        = m.LParam;
winmsg.time          = GetMessageTime();
winmsg.x              = (short) pos;
winmsg.y              = (short) (pos >> 16);

Marshal.StructureToPtr(winmsg, evtmsg.EventPtr, false);
TwRC rc=this.DataSourceManagerEvent(ref evtmsg);
return ((TwBase)this).PassMessage(rc, evtmsg.Message);
}

public bool IsAvailable()
{
    if (isDisable)
        return false;
    if (this.IsTWEnablesDS())
        if (!this.TWDisableDS())
            return onfilterMessage(TwRC.Failure, TwMSG.Null);
    if (this.IsTWOpenedDS())
        if (!this.TWCloseDS())
            return onfilterMessage(TwRC.Failure, TwMSG.Null);
    if (!this.IsTWOpenedDSM())
        if (InitApplication())
            return onfilterMessage(TwRC.Failure, TwMSG.Null);
    return true;
}

protected void Disable()
{
    isDisable=true;
}

protected bool onfilterMessage(TwRC rc, TwMSG MSG)
{
    //if (rc==TwRC.Success || rc==TwRC.XferDone)
    //    return true;
    return this.FilterMessage(((TwBase)this).PassMessage(rc, (short)MSG));
}

protected bool InitApplication()
{
    if (this.IsTWEnablesDS())
        if (!this.TWDisableDS())
            return false;
    if (this.IsTWOpenedDS())
        if (!this.TWCloseDS())
            return false;
    if (this.IsTWOpenedDSM())
        if (!this.TWCloseDSM())
            return false;
    if (!this.TWOpenDSM())
        return false;
    else
        if (!this.TWGetIdentity())
            this.TWCloseDSM();
    return this.IsTWOpenedDSM();
}

public void Select()
{
    if (this.IsAvailable())
    this.onfilterMessage(this.DataSourceManagerIdentity(TwMSG.UserSelect), PhSystems.Twain.TwMSG.UserSelect);
}

public bool AddCapability(TwCap cap, short sval)
{
    return this.AddCapability(new TwCapability(cap, sval));
}

public bool AddCapability(TwCapability cap)
{
    System.Collections.ArrayList delete=new System.Collections.ArrayList();
    IEnumerator Enumerator=Capabilitys.GetEnumerator();
    while (Enumerator.MoveNext())
        if (((TwCapability)Enumerator.Current).Cap==cap.Cap)
            delete.Add(Enumerator.Current);
    Enumerator=delete.GetEnumerator();
    while (Enumerator.MoveNext())
        Capabilitys.Remove(Enumerator.Current);
    //if (!this.AcceptCapability((TwCap)cap.Cap))
    //    return false;
    Capabilitys.Add(cap);
    return true;
}

public void Acquire(bool ShowUI)

```

```

    {
        if (this.IsAvailable())
            if (this.TWOpenDS())
                if (this.TWSetCapabilitis((TwCapability[])Capabilities.ToArray(typeof(TwCapability))))
                    this.TWEnableDS(ShowUI);
                else
                    this.TWCloseDS();
    }

    public bool AcceptCapability(TwCapability cap)
    {
        bool ret=false;
        if (!this.IsAvailable())
            return ret;
        bool open=!this.IsTWOpenedDS();
        if (this.TWOpenDS())
        {
            ret=this.onfilterMessage(this.DataSourceCapability(TwMSG.QuerySupport,
            cap),PhSystems.Twain.TwMSG.QuerySupport);
            if (open)
                this.TWCloseDS();
        }
        return ret;
    }

    public bool AcceptCapability(TwCap Capability)
    {
        return this.AcceptCapability(new TwCapability(Capability));
    }

    public TwCapability GetCapability(TwCap cap)
    {
        TwCapability tcap=new TwCapability(cap);
        bool ret=false;
        if (!this.IsAvailable())
            return null;
        bool open=!this.IsTWOpenedDS();
        if (this.TWOpenDS())
        {
            ret=this.onfilterMessage(this.DataSourceCapability(TwMSG.Get,
            tcap),PhSystems.Twain.TwMSG.Get);
            if (open)
                this.TWCloseDS();
        }
        if (ret)
            return tcap;
        return null;
    }

    public TwCap[] SupportedCapabilitis()
    {
        ArrayList llista =new ArrayList();
        //this.GetCapability(TwCap.SUPPORTEDCAPS).GetTwArray();
        //this.GetCapability(TwCap.IBITDEPTH).GetTwEnumeration();
        Twain.TwContainerCapability
        this.GetCapability(TwCap.SUPPORTEDCAPS).GetContainerCapabilityValue();
        if (Capability == null)
            return new TwCap[]{};
        if (!(Capability is TwArray))
            return new TwCap[]{};
        if (((TwArray)Capability).ItemList==null)
            return new TwCap[]{};
        for(int i=0;i<((TwArray)Capability).ItemList.Length;i++)
            llista.Add((TwCap)((TwArray)Capability).ItemList[i]);
        return (TwCap[])llista.ToArray(typeof(TwCap));
    }

    public ITwTransferPictureEnumerator TransferPictures()
    {
        return (new ITwTransferPictureEnumerator(this));
    }

    #region DLLImport
    [DllImport("kernel32.dll", ExactSpelling=true)]
    internal static extern IntPtr GlobalAlloc( int flags, int size );
    [DllImport("kernel32.dll", ExactSpelling=true)]
    internal static extern IntPtr GlobalLock( IntPtr handle );
    [DllImport("kernel32.dll", ExactSpelling=true)]
    internal static extern bool GlobalUnlock( IntPtr handle );
    [DllImport("kernel32.dll", ExactSpelling=true)]
    internal static extern IntPtr GlobalFree( IntPtr handle );
    [DllImport("user32.dll", ExactSpelling=true)]
    private static extern int GetMessagePos();
    [DllImport("user32.dll", ExactSpelling=true)]
    private static extern int GetMessageTime();
    [StructLayout(LayoutKind.Sequential, Pack=4)]

    internal struct WINMSG

```



```

    {
        public IntPtr          hwnd;
        public int            message;
        public IntPtr        wParam;
        public IntPtr        lParam;
        public int           time;
        public int           x;
        public int           y;
    }

    #endregion
}

public abstract class TwBaseSource:TwBaseManager
{
    protected enum TwOperationMessage
    {
        DataSource=0,
        DataSourceManager=1,
        DataSourceInterface=2,
        DataSourceTransferring=3
    }

    protected enum TwOperationType
    {
        Begining=0,
        Ending=1
    }

    private TwPendingXfers pxfr = new TwPendingXfers();
    protected abstract void state(TwOperationType type, TwOperationMessage Operation);

    protected override bool TWTransferImage(ref IntPtr Image)
    {
        if (this.IsTWTranfering())
            if(
                Image,PhSystems.Twain.TwMSG.Get))
                this.onfilterMessage(DataSourceManagerImageTransfer(ref
                    return true;
                return false;
            }

        protected override bool TWDisableDS()
        {
            if (!this.IsTWEnablesDS())
                return true;
            if
                (this.onfilterMessage(this.DataSourceManagerUserInfo(TwMSG.DisableDS,IntPtr.Zero),PhSystems.Twain.TwMSG.DisableDS))
                this.state(TwOperationType.Ending,TwOperationMessage.DataSourceInterface);
            return !this.IsTWEnablesDS();
        }

        protected override bool TWEnableDS(bool ShowUI)
        {
            if (this.IsTWEnablesDS())
                return true;
            if
                (this.onfilterMessage(DataSourceManagerUserInfo(TwMSG.EnableDS,ShowUI?this.Gethwnd():IntPtr.Zero),PhSystems.Twain.TwMSG.EnableDS
                ))
                this.state(TwOperationType.Begining,TwOperationMessage.DataSourceInterface);
            return this.IsTWEnablesDS();
        }

        protected override bool TWOpenDSM()
        {
            if (this.IsTWOpenedDSM())
                return true;
            if
                (this.onfilterMessage(this.DataSourceManagerParent(TwMSG.OpenDSM,this.Gethwnd()
                ),PhSystems.Twain.TwMSG.OpenDSM))
                this.state(TwOperationType.Begining,TwOperationMessage.DataSourceManager);
            return this.IsTWOpenedDSM();
        }

        protected override bool TWCloseDSM()
        {
            if (this.IsTWOpenedDSM())
                return true;
            if
                (this.onfilterMessage(this.DataSourceManagerParent(TwMSG.CloseDSM,this.Gethwnd()),PhSystems.Twain.TwMSG.CloseDSM))
                this.state(TwOperationType.Ending,TwOperationMessage.DataSourceManager);
            return !this.IsTWOpenedDSM();
        }

        protected override bool TWOpenDS()
        {
            if (this.IsTWOpenedDS())
                return true;
            if
                (this.onfilterMessage(this.DataSourceManagerIdentity(TwMSG.OpenDS),PhSystems.Twain.TwMSG.OpenDS))

```

```

        this.state(TwOperationType.Begining, TwOperationMessage.DataSource);
        return this.IsTWOpenedDS();
    }

    protected override bool TWCloseDS()
    {
        if (!this.IsTWOpenedDS())
            return true;
        if
(this.onfilterMessage(this.DataSourceManagerIdentity(TwMSG.CloseDS), PhSystems.Twain.TwMSG.CloseDS))
            this.state(TwOperationType.Ending, TwOperationMessage.DataSource);
        return !this.IsTWOpenedDS();
    }

    protected override bool TWGetIdentity()
    {
        return
this.onfilterMessage(this.DataSourceManagerIdentity(TwMSG.GetDefault), PhSystems.Twain.TwMSG.GetDefault);
    }

    protected override bool TWSetCapabilitis(TwCapability[] Capabilitis)
    {
        if (Capabilitis==null)
            return false;
        System.Collections.IEnumerator enumerator=Capabilitis.GetEnumerator();
        while (enumerator.MoveNext())
            if(!this.onfilterMessage(this.DataSourceCapability(TwMSG.Set, (TwCapability)enumerator.Current), PhSystems.Twain.TwMSG.S
et))
                return false;
        return true;
    }

    protected override bool TWCloseTransferImage(bool IsCancel)
    {
        if (this.IsTWTranfering())
            if(
this.onfilterMessage(DataSourceManagerPendingInfo(IsCancel?TwMSG.Reset:TwMSG.EndXfer, ref
this.pxfr), PhSystems.Twain.TwMSG.EndXfer))
                return true;
            return false;
    }

    protected override bool TWOpenTransferImage()
    {
        if (pxfr==null)
            return false;
        if (this.onfilterMessage(this.IsTWTranfering()?TwRC.Success:TwRC.Failure, TwMSG.Null))
            pxfr.Count=1;
        else
            pxfr.Count=0;
        return pxfr.Count==1;
    }

    /*
    protected override TwPendingXfers PendingXfers()
    {
        return pxfr;
    }
    */

    protected override bool IsPendingXfers()
    {
        if (pxfr==null)
            return false;
        return pxfr.Count!=0;
    }

    protected override void Finish()
    {
        if (this.IsTWEnablesDS())
            if (!this.TWDisableDS())
                return;
        if (this.IsTWOpenedDS())
            this.TWCloseDS();
        else
            return;
        if (this.IsTWOpenedDSM())
            this.TWCloseDSM();
        else
            return;
    }
}

public class TwSourceManager:TwBaseSource
{
    private bool msgfilter=false;
    private IntPtr hwnd=IntPtr.Zero;

```

```

private PostFilterMessage myPreFilterMessage=null;

public delegate void PostFilterMessage(TwainCommand TC);

protected TwSourceManager() {}

~TwSourceManager()
{
    this.myPreFilterMessage=null;
    this.hwnd=IntPtr.Zero;
}

public TwSourceManager(PostFilterMessage myFMessage,IntPtr hwnd)
{
    this.InitApplication(myFMessage,hwnd);
}

protected override IntPtr Gethwnd()
{
    return this.hwnd;
}

protected override bool FilterMessage(TwainCommand cmd)
{
    bool ret=false;
    switch( cmd )
    {
        default:
        {
            return ret;
        }
        case TwainCommand.Null:
        {
            return (!ret);
        }
        case TwainCommand.Not:
        {
            return ret;
        }
        case TwainCommand.CloseRequest:
        {
            //if (this.myPreFilterMessage!=null)
            //    this.myPreFilterMessage(cmd);
            if (this.TWDisableDS())
                ret=this.TWCloseDS();
            if (this.myPreFilterMessage!=null)
                this.myPreFilterMessage(cmd);
            return ret;
        }
        case TwainCommand.CloseOk:
        {
            //if (this.myPreFilterMessage!=null)
            //    this.myPreFilterMessage(cmd);
            if (this.TWDisableDS())
                ret=this.TWCloseDS();
            if (this.myPreFilterMessage!=null)
                this.myPreFilterMessage(cmd);
            return ret;
        }
        case TwainCommand.DeviceEvent:
        {
            if (this.myPreFilterMessage!=null)
                this.myPreFilterMessage(cmd);
            return (!ret);
        }
        case TwainCommand.TransferReady:
        {
            if (this.myPreFilterMessage!=null)
                this.myPreFilterMessage(cmd);
            else
                this.FilterMessage(TwainCommand.Failure);
            if (this.IsTWTranfering())
                this.TWCloseTransferImage(true);
            if (this.TWDisableDS())
                ret=this.TWCloseDS();
            return ret;
        }
        case TwainCommand.Failure:
        {
            if (this.myPreFilterMessage!=null)
                this.myPreFilterMessage(cmd);
            return (ret);
        }
        case TwainCommand.Cancel:
        {
            if (this.myPreFilterMessage!=null)
                this.myPreFilterMessage(cmd);
            return (ret);
        }
    }
}

protected override void state(TwOperationType type, TwOperationMessage Operation)
{
    if (Operation==TwOperationMessage.DataSourceInterface)

```

```

    }

    OperationInterface(type);
}

protected bool InitApplication(PostFilterMessage mfmsg, IntPtr hwnd)
{
    this.hwnd=hwnd;
    this.myPreFilterMessage=mfmsg;
    if (!(this.hwnd==IntPtr.Zero || this.myPreFilterMessage==null))
        return base.InitApplication();
    base.Disable();
    return false;
}

private void OperationInterface(TwOperationType type)
{
    if(msgfilter && type==TwOperationType.Ending)
        Application.RemoveMessageFilter(this);
    if(!msgfilter && type==TwOperationType.Begining)
        Application.AddMessageFilter(this);
    msgfilter=!msgfilter && type==TwOperationType.Ending) || (!msgfilter &&
type==TwOperationType.Begining);
}

public sealed class TwFormHelper:TwSourceManager
{
    private bool Available;
    private ITwainForm Form;
    public TwFormHelper(ITwainForm Form)
    {
        this.Form=Form;
        Available=base.InitApplication(new
TwSourceManager.PostFilterMessage(this.TwainCommandFilter), Form.hwndp());
    }

    ~TwFormHelper()
    {
        this.Form=null;
        if (this.IsAvailable())
            base.Finish();
    }

    public new bool IsAvailable()
    {
        if (base.IsAvailable())
            return (Available);
        return false;
    }

    private void TwainCommandFilter(TwainCommand TC)
    {
        if (this.Form==null)
            return;
        switch (TC)
        {
            case TwainCommand.CloseOk:
            {
                this.Form.CloseOk(this);
                break;
            }
            case TwainCommand.CloseRequest:
            {
                this.Form.CloseRequest(this);
                break;
            }
            case TwainCommand.TransferReady:
            {
                this.Form.TransferReady(this);
                break;
            }
            case TwainCommand.DeviceEvent:
            {
                this.Form.DeviceEvent(this);
                break;
            }
            case TwainCommand.Cancel:
            {
                this.Form.Cancel(this);
                break;
            }
            case TwainCommand.Failure:
            {
                this.Form.Failure(this);
                break;
            }
        }
    }
}

}
}
}
}
}

```

7.2.2. Identificació de codis de barres

Tal com s'ha dit els documents s'identifiquen a partir d'un codi de barres, cada cop que és llegeix un nou codi de barres, implica que s'està processant un document nou. Per tal de no estar lligats a un software concret de lectura de codis de barres es va fer el procés de generació de documents independent del lector de codis de barres.

Namespace *PhSystems.Image.BarCode*

```

Public Enum BarcodeType
    UnknownBarcode
    Industry2of5Barcode
    InterLeaved2of5Barcode
    IATA2of5Barcode
    DataLogic2of5Barcode
    Invert2of5Barcode
    BCDMatrixBarcode
    Matrix2of5Barcode
    Code32Barcode
    Code39Barcode
    Code39ExtendedBarcode
    CodabarBarcode
    Code93Barcode
    Code93ExtendedBarcode
    Code128Barcode
    EAN13Barcode
    EAN8Barcode
    UPCEBarcode
    UPCABarcode
    Add5Barcode
    Add2Barcode
    EAN128Barcode
    PatchCodeBarcode
    PostNetBarcode
    PDF417Barcode
    DataMatrixBarcode

End Enum

Public Enum ShowTextInfo
    None
    Value
    ValueAndChecksum
    All
End Enum

Public Interface IBarcode
    Inherits IComparable

    ReadOnly Property Area() As System.Drawing.Rectangle
    ReadOnly Property BarcodeType() As BarcodeType
    Property BarcodeValue() As String
    Function CreateImage(Optional ByVal ShowText As ShowTextInfo = ShowTextInfo.None) As System.Drawing.Image

End Interface

Public Interface IBarcodeImage
    Inherits IEnumerable, IDisposable

    Delegate Function TestBarcode(ByVal Barcode As IBarcode) As Boolean
    Function Image() As System.Drawing.Image
    Function ReadBarcode(ByVal TestBarcode As TestBarcode) As IBarcode
    Function WriteBarcode(ByVal Barcode As IBarcode, ByVal Area As System.Drawing.Rectangle) As Boolean

End Interface

Public Interface IBarcodeImageFactory
    Function CreateBarcodeImage(ByVal Image As System.Drawing.Image) As IBarcodeImage
    Function CreateBarcode(ByVal Area As System.Drawing.Rectangle, ByVal BarcodeType As BarcodeType, ByVal BarcodeValue As String) As IBarcode

End Interface

Public Interface IProccesImages
    Inherits IEnumerable

    Sub Initialize(ByVal BarcodeImageFactory As IBarcodeImageFactory)
    ReadOnly Property CurrentImage() As IBarcodeImage

End Interface

Public MustInherit Class AbstractBarcode
    Implements PhSystems.Image.BarCode.IBarcode

    Private myArea As System.Drawing.Rectangle
    Private myBarcodeType As PhSystems.Image.BarCode.BarCodeType
    Private myBarcodeValue As String
    Protected MustOverride Function CreateImage(Optional ByVal ShowText As ShowTextInfo = ShowTextInfo.None) As System.Drawing.Image Implements Image.BarCode.IBarcode.CreateImage
    
```

```

Public Sub New(ByVal Area As System.Drawing.Rectangle, ByVal BarcodeType As PhSystems.Image.BarCode.BarCodeType, ByVal
BarcodeValue As String)

    Me.myArea = Area
    Me.myBarcodeType = BarcodeType
    Me.myBarcodeValue = BarcodeValue

End Sub

Private ReadOnly Property Area() As System.Drawing.Rectangle Implements PhSystems.Image.BarCode.IBarcode.Area

    Get
        Return (Me.myArea)
    End Get

End Property

Private ReadOnly Property BarcodeType() As PhSystems.Image.BarCode.BarCodeType Implements
PhSystems.Image.BarCode.IBarcode.BarcodeType

    Get
        Return Me.myBarcodeType
    End Get

End Property

Private Property BarcodeValue() As String Implements PhSystems.Image.BarCode.IBarcode.BarcodeValue

    Get
        Return Me.myBarcodeValue
    End Get

    Set(ByVal Value As String)
        Me.myBarcodeValue = Value
    End Set

End Property

Public Function CompareTo(ByVal obj As Object) As Integer Implements System.IComparable.CompareTo

    If obj Is Nothing Then Return -1
    If TypeOf obj Is PhSystems.Image.BarCode.IBarcode Then Return Me.BarcodeValue.CompareTo(CType(obj,
PhSystems.Image.BarCode.IBarcode).BarcodeValue)
    Return -1

End Function

End Class

Public Class DocumentImage
Implements IComparable
Implements IDisposable

Private mBarcode As IBarcode
Private Images As ArrayList

Friend Sub New(ByVal Barcode As IBarcode)

    Me.mBarcode = Barcode
    Me.Images = New ArrayList

End Sub

Public Function Barcode() As IBarcode

    Return Me.mBarcode

End Function

Public Function Image(ByVal stream As System.IO.Stream) As System.Drawing.Image

    If PhSystems.Image.ImageConverter.AgruparTiff(Images.GetEnumerator, stream) Then Return
System.Drawing.Image.FromStream(stream)

End Function

Public Function Image2(ByVal stream As System.IO.Stream) As System.Drawing.Image

    If PhSystems.Image.ImageConverter.AgruparTiff2(Images.GetEnumerator, stream) Then Return
System.Drawing.Image.FromStream(stream)

End Function

Public Function SaveImage(ByVal path As String) As Boolean

    On Error Resume Next
    If Me.PageCount > 0 Then
        Dim stream As System.IO.Stream = System.IO.File.Open(path, IO.FileMode.Create)
        Dim image As System.Drawing.Image = Me.Image(stream)
        image.Dispose()
        stream.Close()
        Dim stream2 As System.IO.Stream = System.IO.File.Open(path & "2", IO.FileMode.Create)
        Dim image2 As System.Drawing.Image = Me.Image2(stream2)
        image2.Dispose()
        stream2.Close()
    End If

End Function

```

```

End Function
Public Function PageCount() As Integer

    Return Me.Images.Count()

End Function

Public Sub AddImage(ByVal Image As System.Drawing.Image)

    On Error Resume Next
    Me.Images.Add(Image)

End Sub

Public Function CompareTo(ByVal obj As Object) As Integer Implements System.IComparable.CompareTo

    If obj Is Nothing Then Return -1
    If TypeOf obj Is DocumentImage Then Return Me.Barcode.CompareTo(CType(obj, DocumentImage).Barcode)
    Return -1

End Function

Public Sub Dispose() Implements System.IDisposable.Dispose

    If Not Images Is Nothing Then
        Dim enumerator As IEnumerator = Images.GetEnumerator
        While enumerator.MoveNext
            If Not enumerator.Current Is Nothing Then If TypeOf enumerator.Current Is IDisposable Then
                CType(enumerator.Current, IDisposable).Dispose()
            End While
            Images.Clear()
        End If

    End Sub

End Class

Public Class ProccesDocuments
    Implements IEnumerator

    Private DocumentImage As DocumentImage
    Private NextDocumentImage As DocumentImage
    Private images As IProccesImages
    Private TestBarCode As PhSystems.Image.BarCode.IBarcodeImage.TestBarCode
    Private EventOnAddImage As OnAddImage

    Public Delegate Sub OnAddImage(ByVal DocumentImage As DocumentImage)

    Private Sub New(ByVal images As IProccesImages, ByVal TestBarCode As PhSystems.Image.BarCode.IBarcodeImage.TestBarCode,
        ByVal EventOnAddImage As OnAddImage, ByVal previous As DocumentImage)

        Me.images = images
        Me.TestBarCode = TestBarCode
        Me.NextDocumentImage = previous
        Me.EventOnAddImage = EventOnAddImage

    End Sub

    Public ReadOnly Property CurrentDocument() As DocumentImage

        Get

            Return Me.DocumentImage

        End Get

    End Property

    Private ReadOnly Property IEnumeratorCurrent() As Object Implements System.Collections.IEnumerator.Current

        Get

            Return Me.CurrentDocument

        End Get

    End Property

    Private Function MoveNext() As Boolean Implements System.Collections.IEnumerator.MoveNext

        Dim barcode As IBarcode
        Dim image As IBarcodeImage

        If Me.images.MoveNext Then
            image = Me.images.CurrentImage
            barcode = image.ReadBarcode(Me.TestBarCode)
            If barcode Is Nothing Then
                If Me.NextDocumentImage Is Nothing Then Me.NextDocumentImage = New DocumentImage(Nothing)
                Me.NextDocumentImage.AddImage(image.Image)
                If Not EventOnAddImage Is Nothing Then EventOnAddImage(Me.NextDocumentImage)
                Return Me.MoveNext
            Else
                Me.DocumentImage = Me.NextDocumentImage
                Me.NextDocumentImage = New DocumentImage(barcode)
                Me.NextDocumentImage.AddImage(image.Image)
                If Not EventOnAddImage Is Nothing Then EventOnAddImage(Me.NextDocumentImage)
                If Me.DocumentImage Is Nothing Then Return Me.MoveNext
            End If
        End If
    End Function

```

```
        End If
    Else
        Me.DocumentImage = Nothing
    End If
    If Me.DocumentImage Is Nothing Then
        Me.DocumentImage = Me.NextDocumentImage
        Me.NextDocumentImage = Nothing
    End If
    Return Not Me.DocumentImage Is Nothing
End Function

Private Sub Reset() Implements System.Collections.IEnumerator.Reset
    Me.images.Reset()
End Sub

Public Shared Function Proces(ByVal images As IProcesImages, ByVal TestBarCode As
PhSystems.Image.BarCode.IBarcodeImage.TestBarCode, ByVal EventOnAddImage As OnAddImage, ByVal previous As DocumentImage) As
IEnumerator
    Return New ProccesDocuments(images, TestBarCode, EventOnAddImage, previous)
End Function

End Class

End Namespace
```


7.2.3. Exemple: adquisició via escaner

Per tal d'adquirir documents via escàner n'hi ha prou amb fer un punt de menú amb el codi següent:

```
Private Sub acquire(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem1.Click

    Me.Visible = True
    If manager.IsAvailable() Then
        manager.Acquire(PhSystems.PhModi.WinClient.Configuration.Singleton.Twainconfig.ShowUIonAcquire)
        If Me.failure Then
            On Error Resume Next
            MsgBox("Error: " & manager.LastError.Operation.ToString, MsgBoxStyle.Exclamation, "Scanner")
            Me.Close()
        End If
    Else
        Me.Close()
    End If

End Sub
```

Si el que es vol és fer-ho des d'un directori es pot fer:

```
Private Sub acquireFile(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ImportDoc.Click
    Me.failure = False
    Me.FolderBrowserDialog1.RootFolder = System.Environment.SpecialFolder.MyComputer
    If System.IO.Directory.Exists(Me.path_temp) Then Me.FolderBrowserDialog1.SelectedPath =
    System.IO.Directory.GetParent(Me.path_temp).FullName
    Me.FolderBrowserDialog1.Description = "Directorio de importaci6n de imatges"
    Me.FolderBrowserDialog1.ShowDialog()
    If System.IO.Directory.Exists(Me.FolderBrowserDialog1.SelectedPath()) Then
        Dim current As PhSystems.Image.BarCode.DocumentImage = Me.previus
        Me.previus = Nothing
        PhSystems.Generic.Base.Iterate.IndirectionEvent(PhSystems.Image.BarCode.ProcesDocuments.Proces(PhSystems.PhModi.Image.ProcesIm
        ages.Proces(PhSystems.PhModi.Image.FileSystemImageEnumerator.GetInstance(Me.FolderBrowserDialog1.SelectedPath,
        PhSystems.PhModi.WinClient.Configuration.Singleton.PatternImage), PhSystems.PhModi.Image.ProcesImages.GetBarcodeImageFactory),
        AddressOf Me.myTestBarcode, AddressOf Me.OnAddImage, current), New PhSystems.Generic.Base.Iterate.IndirectionHandler(AddressOf
        AfegirDoc))
        AfegirDoc(Nothing)
        Dim Result As MsgBoxResult = MsgBox("Desea eliminar Las imatges importadas?", MsgBoxStyle.OKCancel, "Scanear
        Directorio")
        If Result = MsgBoxResult.OK Then PhSystems.Generic.Base.Iterate.IndirectionEvent(New
        PhSystems.Generic.Base.FileSystemEnumerator(Me.FolderBrowserDialog1.SelectedPath,
        PhSystems.PhModi.WinClient.Configuration.Singleton.PatternImage), New
        PhSystems.Generic.Base.Iterate.RemovingSubHandler(AddressOf System.IO.File.Delete))
        End If
        MsgBox("Importacion finalizada")
    End Sub
```

Per que tot funcioni només caldrà en el formulari afegir el següent:

```
Public manager As PhSystems.Twain.TwSourceManager
Private Sub PreFilterMessage(ByVal TC As PhSystems.Twain.TwainCommand)
    Select Case TC
        Case PhSystems.Twain.TwainCommand.TransferReady
            Me.failure = False
            Dim current As PhSystems.Image.BarCode.DocumentImage = Me.previus
            Me.previus = Nothing
            PhSystems.Generic.Base.Iterate.IndirectionEvent(PhSystems.Image.BarCode.ProcesDocuments.Proces(PhSystems.PhModi.Image.ProcesIm
            ages.Proces(PhSystems.PhModi.Image.TwImageEnumerator.GetInstance(manager.TransferPictures),
            PhSystems.PhModi.Image.ProcesImages.GetBarcodeImageFactory), AddressOf Me.myTestBarcode, AddressOf Me.OnAddImage, current), New
            PhSystems.Generic.Base.Iterate.IndirectionHandler(AddressOf AfegirDoc))
        Case PhSystems.Twain.TwainCommand.Failure
            Me.failure = True
        Case PhSystems.Twain.TwainCommand.CloseOk, PhSystems.Twain.TwainCommand.CloseRequest
            If Not Me.failure Then
                Dim Result As MsgBoxResult = MsgBox("Desea adquirir mas documentos?", MsgBoxStyle.OKCancel, "Scanear")
                If Result = MsgBoxResult.OK Then
                    acquire(Nothing, Nothing)
                Else
                    AfegirDoc(Nothing)
                End If
            End If
        Else
            Select Case manager.LastError.Operation
                Case PhSystems.Twain.TwOperation.CapabilitySet
                    MsgBox([Enum].GetName(GetType(PhSystems.Twain.TwCC), manager.LastError.Error) & " " &
                    [Enum].GetName(GetType(PhSystems.Twain.TwOperation), manager.LastError.Operation) & " " &
                    [Enum].GetName(GetType(PhSystems.Twain.TwCap), CType(manager.LastError.GetTarget, PhSystems.Twain.TwCapability).Cap),
                    MsgBoxStyle.Critical, "Error")
                Case Else
                    MsgBox([Enum].GetName(GetType(PhSystems.Twain.TwCC), manager.LastError.Error) & " " &
                    [Enum].GetName(GetType(PhSystems.Twain.TwOperation), manager.LastError.Operation), MsgBoxStyle.Critical, "Error")
            End Select
        End If
        Me.failure = False
        Case PhSystems.Twain.TwainCommand.Cancel
            Me.failure = False
    End Select
End Sub
```

La rutina anterior permet un escaneig full a full, en cas de que no es disposi d'un aparell d'escaneig massiu.

La vinculació amb l'arxiu òptic es fa mitjançant una classe helper:

```
Private Sub AfegirDoc(ByVal item As Object)
    helper.AsyncCallback(AddressOf helper.create(Me, item, adjunt_doc).AfegirDoc)
End Sub
```

Això es fa així ja que al ser una classe remota, es millor treballar asíncronament.

```
Public Shared Sub AfegirDoc(ByVal adqdoc As frm_adqdoc, ByRef BarcodeValue As String, ByVal PageCount As Integer, ByVal finalitzat As Boolean, ByVal asinc As Boolean)
    If asinc Then
        CType(New FinalitzarDocDelegate(AddressOf FinalitzarDoc), FinalitzarDocDelegate).BeginInvoke(adqdoc, BarcodeValue, Nothing, Nothing)
        CType(New AfegirDocDelegate(AddressOf AfegirDocument), AfegirDocDelegate).BeginInvoke(adqdoc, BarcodeValue, PageCount, finalitzat, Nothing, Nothing)
    Else
        FinalitzarDoc(adqdoc, BarcodeValue)
        AfegirDocument(adqdoc, BarcodeValue, PageCount, finalitzat)
    End If
End Sub

Private Shared Sub AfegirDocument(ByVal adqdoc As frm_adqdoc, ByRef BarcodeValue As String, ByVal PageCount As Integer, ByVal finalitzat As Boolean)
    Dim DocumentName As String = ""
    Dim SchemaName As String = ""
    Dim TemplateName As String = ""
    Dim State As String = "Error"

    If PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHmodiapi Is Nothing Then
        MsgBox("No se tiene acceso al servicio web", MsgBoxStyle.Critical, "Error")
        Exit Sub
    End If
    DocumentName = PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHmodiapi.GetDocumentName(BarcodeValue)
    If PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHmodiapi.ExistElement(DocumentName, Remoting.TypeElement.document) Then
        State = PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHmodiapi.GetDocumentNameState(DocumentName)
        TemplateName = PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHmodiapi.GetTemplateNameDocument(DocumentName)
        If Not PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHmodiapi.ExistElement(TemplateName, Remoting.TypeElement.schema) Then
            SchemaName = PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHmodiapi.GetSchemaNameTemplate(TemplateName)
        Else
            SchemaName = TemplateName
        End If
    End If
    Select Case PhSystems.PHModi.WinClient.Configuration.Singleton.Singleton.Mask.GetTypeBarcodeAction(BarcodeValue)
    Case PhSystems.PHModi.WinClient.Configuration.MaskBarcode.BarcodeAction.Escritura
        SchemaName = PhSystems.PHModi.WinClient.Configuration.Singleton.GetPhTypeDocApi().GetElementNameByTypeDoc(PhSystems.PHModi.WinClient.Configuration.Singleton.Mask.GetTypeDoc(BarcodeValue))
        State = ""
    End Select
    ReplaceExisteixDoc(adqdoc, New System.Windows.Forms.ListViewItem(New String() {BarcodeValue, SchemaName, TemplateName, PageCount, State, IIf(finalitzat, "Finalizado", "Escaneando")}, IIf(State = "Error", 2, IIf(PhSystems.PHModi.WinClient.Configuration.Singleton.Singleton.Mask.GetTypeBarcodeAction(BarcodeValue) = Configuration.MaskBarcode.BarcodeAction.Escritura, 3, 1))))
End Sub
```

7.3. Integració Lirba

7.3.1. Generació de missatges Soap

Per tal de comunicar-se amb l'arxiu òptic s'ha implementat un framework de serveis, de tal manera que establint una configuració, es poden carregar diferents serveis web. Al final tot passa per construir missatges soap, que és realitza des de la classe parser:

```

Option Explicit

Public Function GetSoapMappersNamesByWSDLOperation(WSDLOperation As MSSOAPLib30.IWSDLOperation) As String()

Dim EnumSoapMappers As MSSOAPLib30.IEnumSoapMappers
Dim SoapMapper As MSSOAPLib30.ISoapMapper
Dim fetched As Long
Dim ret() As String
Dim i As Integer

If WSDLOperation Is Nothing Then Exit Function
WSDLOperation.GetOperationParts EnumSoapMappers
If EnumSoapMappers Is Nothing Then Exit Function
Do
    EnumSoapMappers.Next 1, SoapMapper, fetched
    If fetched = 1 Then
        ReDim ret(i)
        ret(i) = SoapMapper.PartName
        i = i + 1
    End If
Loop While fetched = 1
GetSoapMappersNamesByWSDLOperation = ret

End Function

Public Function GetWSDLOperationNames(WSDLPort As MSSOAPLib30.IWSDLPort) As String()

Dim EnumWSDLOperations As MSSOAPLib30.IEnumWSDLOperations
Dim WSDLOperation As MSSOAPLib30.IWSDLOperation
Dim fetched As Long
Dim ret() As String
Dim i As Integer

If WSDLPort Is Nothing Then Exit Function
WSDLPort.GetSoapOperations EnumWSDLOperations
If EnumWSDLOperations Is Nothing Then Exit Function
Do
    EnumWSDLOperations.Next 1, WSDLOperation, fetched
    If fetched = 1 Then
        ReDim Preserve ret(i)
        ret(i) = Trim(WSDLOperation.Name)
        i = i + 1
    End If
Loop While fetched = 1
GetWSDLOperationNames = ret

End Function

Public Function NewParameterContainer(SoapTypeMapper As MSSOAPLib30.ISoapTypeMapper, Optional Encoding As String = "", Optional flags As Long = 0, Optional EncodingMode As enEncodingStyle = enDocumentLiteral, Optional var As Variant = Nothing) As ParameterContainer

Dim pc As New ParameterContainer

pc.Initialize SoapTypeMapper, Encoding, flags, EncodingMode, isObject(var), var
Set NewParameterContainer = pc

End Function

Public Function GetWSDLService(WSDLReader As MSSOAPLib30.WSDLReader30, ServiceName As String) As MSSOAPLib30.IWSDLService

Dim EnumWSDLService As MSSOAPLib30.IEnumWSDLService
Dim WSDLService As MSSOAPLib30.IWSDLService
Dim fetched As Long

On Error GoTo t

If WSDLReader Is Nothing Then Exit Function
WSDLReader.GetSoapServices EnumWSDLService
If EnumWSDLService Is Nothing Then Exit Function
Do
    EnumWSDLService.Next 1, WSDLService, fetched
    If fetched = 1 Then
        If WSDLService.Name = ServiceName Or ServiceName = "" Then
            Set GetWSDLService = WSDLService
            Exit Function
        End If
    End If
Loop While fetched = 1
t:
    
```

```

End Function
Public Function GetWSDLPort(WSDLReader As MSSOAPLib30.WSDLReader30, WSDLService As MSSOAPLib30.IWSDLService, PortName As String)
As MSSOAPLib30.IWSDLPort

Dim EnumWSDLPorts As MSSOAPLib30.IEnumWSDLPorts
Dim WSDLPort As MSSOAPLib30.IWSDLPort
Dim fetched As Long

On Error GoTo t
WSDLService.GetSoapPorts EnumWSDLPorts
If EnumWSDLPorts Is Nothing Then Exit Function
Do
    EnumWSDLPorts.Next 1, WSDLPort, fetched
    If fetched = 1 Then
        If WSDLPort.Name = PortName Or PortName = "" Then
            Set GetWSDLPort = WSDLPort
            Exit Function
        End If
    End If
Loop While fetched = 1
t:
End Function

Public Function GetWSDLOperation(WSDLPort As MSSOAPLib30.IWSDLPort, OperationToFind As String) As MSSOAPLib30.IWSDLOperation

Dim EnumWSDLOperations As MSSOAPLib30.IEnumWSDLOperations
Dim WSDLOperation As MSSOAPLib30.IWSDLOperation

On Error Resume Next
If WSDLPort Is Nothing Then Exit Function
WSDLPort.GetSoapOperations EnumWSDLOperations
If EnumWSDLOperations Is Nothing Then Exit Function
EnumWSDLOperations.Find OperationToFind, WSDLOperation
Set GetWSDLOperation = WSDLOperation

End Function

Public Function GetSoapMapperByName(ByRef WSDLOperation As MSSOAPLib30.IWSDLOperation, Name As String) As
MSSOAPLib30.ISoapMapper

End Function

Public Function GetSoapMapperReturn(ByRef WSDLOperation As MSSOAPLib30.IWSDLOperation) As MSSOAPLib30.ISoapMapper

Dim EnumSoapMappers As MSSOAPLib30.IEnumSoapMappers
Dim SoapMapper As MSSOAPLib30.ISoapMapper
Dim fetched As Long

If Not WSDLOperation Is Nothing Then
    WSDLOperation.GetOperationParts EnumSoapMappers
    If Not EnumSoapMappers Is Nothing Then
        Do
            EnumSoapMappers.Next 1, SoapMapper, fetched
            If fetched = 1 Then
                If Not SoapMapper.IsInput Then
                    Set GetSoapMapperReturn = SoapMapper
                    fetched = 0
                End If
            End If
        Loop While fetched = 1
    End If
End If

Set SoapMapper = Nothing
Set EnumSoapMappers = Nothing

End Function

Public Function GetObjectResultBySoapReader(SoapReader As MSSOAPLib30.SoapReader30, SoapTypeMapperFactory As
MSSOAPLib30.SoapTypeMapperFactory30, SoapMapper As MSSOAPLib30.ISoapMapper, XMLDOMNode As MSXML2.IXMLDOMNode,
OutputSoapTypeMapper As ParameterContainer, ErrorSoapTypeMapper As ParameterContainer) As ReturnValue

Dim i As Integer
Dim mat As Variant
Dim ret As ReturnValue
Dim XMLDOMNodeList As MSXML2.IXMLDOMNodeList

If Not SoapReader Is Nothing Then
    If Not SoapReader.dom.xml = "" Then
        If SoapReader.Fault Is Nothing Then
            If Not OutputSoapTypeMapper Is Nothing Then
                If Not SoapMapper Is Nothing And XMLDOMNode Is Nothing Then Set XMLDOMNode = GetNodeName(SoapReader,
SoapReader.dom, "/" & SoapMapper.PartName)
                If Not XMLDOMNode Is Nothing Then
                    If XMLDOMNode.baseName = "Array" And XMLDOMNode.namespaceURI = "http://schemas.xmlsoap.org/soap/encoding/"
Then
                        If Not OutputSoapTypeMapper Is Nothing Then
                            Set XMLDOMNodeList = XMLDOMNode.selectNodes("item")
                            mat = OutputSoapTypeMapper.CreateArrayType
                            For i = 0 To XMLDOMNodeList.Length
                                ReDim Preserve mat(i)
                                Set ret = GetObjectResultBySoapReader(SoapReader, SoapTypeMapperFactory, SoapMapper,
GetNode(SoapReader, XMLDOMNodeList.Item(i)), OutputSoapTypeMapper, ErrorSoapTypeMapper)
                                If ret.isObject Then
                                    Set mat(i) = ret.Value
                                Else
                                    mat(i) = ret.Value
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
End If

```

```

        Next
    End If
    Set GetObjectResultBySoapReader = NewReturnValue(mat)
Else
    If Not OutputSoapTypeMapper.SoapTypeMapper Is Nothing Then Set GetObjectResultBySoapReader =
NewReturnValue(OutputSoapTypeMapper.SoapTypeMapper.Read(SoapReader,
XMLDOMNode,
OutputSoapTypeMapper.EncodingMode, OutputSoapTypeMapper.flags))
    End If
Else
    Set GetObjectResultBySoapReader = NewReturnValue(Nothing)
    End If
Else
    Set GetObjectResultBySoapReader = NewReturnValue(Nothing)
    End If
Else
    If Not ErrorSoapTypeMapper Is Nothing Then
        If Not ErrorSoapTypeMapper.SoapTypeMapper Is Nothing Then
            ErrorSoapTypeMapper.SoapTypeMapper.Init SoapTypeMapperFactory, Nothing, Nothing, 0
            Set GetObjectResultBySoapReader = NewReturnValue(ErrorSoapTypeMapper.SoapTypeMapper.Read(SoapReader,
SoapReader.Fault, ErrorSoapTypeMapper.Encoding, ErrorSoapTypeMapper.EncodingMode, ErrorSoapTypeMapper.flags))
        Else
            Set GetObjectResultBySoapReader = NewReturnValue(SoapReader.Fault)
        End If
    Else
        Set GetObjectResultBySoapReader = NewReturnValue(SoapReader.Fault)
    End If
End If
Else
    Set GetObjectResultBySoapReader = NewReturnValue(XMLDOMNode)
End If
Else
    Set GetObjectResultBySoapReader = NewReturnValue(XMLDOMNode)
End If
Set SoapReader = Nothing
Set XMLDOMNode = Nothing

End Function

Public Sub GenerateMessageSoapByParameterContainer(SoapTypeMapperFactory As MSSOAPLib30.SoapTypeMapperFactory30, ElementType As
String, XmlNamespace As String, container As ParameterContainer, SoapSerializer As MSSOAPLib30.SoapSerializer30)

    On Error Resume Next
    If Not container Is Nothing Then
        If Not container.SoapTypeMapper Is Nothing Then
            SoapSerializer.StartElement ElementType, XmlNamespace
            container.SoapTypeMapper.Write SoapSerializer, container.Encoding, container.EncodingMode, container.flags,
container.var
            SoapSerializer.EndElement
        End If
    End If
End Sub

Public Sub GenerateMessageSoapByMemberInfo(WSDLOperation As MSSOAPLib30.IWSDLOperation, SoapTypeMapperFactory As
MSSOAPLib30.ISoapTypeMapperFactory, SoapSerializer As MSSOAPLib30.SoapSerializer30, Values As Variant)

Dim EnumSoapMappers As MSSOAPLib30.IEnumSoapMappers
Dim SoapMapper As MSSOAPLib30.ISoapMapper
Dim fetched As Long

On Error GoTo t
SoapSerializer.StartEnvelope
SoapSerializer.StartBody
SoapSerializer.StartElement WSDLOperation.InputMessage.MessageName, WSDLOperation.InputMessage.MessageNamespace
Set EnumSoapMappers = WSDLOperation.InputMessage.MessageParts

If Not EnumSoapMappers Is Nothing Then
    Do
        EnumSoapMappers.Next 1, SoapMapper, fetched
        If fetched = 1 And Not SoapMapper Is Nothing Then If SoapMapper.IsInput Then GenerateMessageSoapByParameterContainer
SoapTypeMapperFactory, IIf(IsSympleType(helper.GetElement(Values, SoapMapper.ParameterOrder)), SoapMapper.ElementName,
SoapMapper.ElementType), SoapMapper.XmlNamespace,
Me.NewParameterContainer(SoapTypeMapperFactory.GetTypeMapperByName(SoapMapper.ElementType, SoapMapper.XmlNamespace), , , ,
helper.GetElement(Values, SoapMapper.ParameterOrder)), SoapSerializer
        Loop While fetched = 1
    End If
    SoapSerializer.EndElement
    SoapSerializer.EndBody
    SoapSerializer.EndEnvelope
t:
    Set SoapMapper = Nothing
    Set EnumSoapMappers = Nothing

End Sub

Public Sub GenerateMessageSoap(WSDLReader As MSSOAPLib30.WSDLReader30, WSDLOperation As MSSOAPLib30.IWSDLOperation, parameters
As Collection, SoapSerializer As MSSOAPLib30.SoapSerializer30)

Dim EnumSoapMappers As MSSOAPLib30.IEnumSoapMappers
Dim SoapMapper As MSSOAPLib30.ISoapMapper
Dim container As ParameterContainer
Dim fetched As Long

On Error GoTo t
SoapSerializer.StartEnvelope
SoapSerializer.StartBody
SoapSerializer.StartElement WSDLOperation.InputMessage.MessageName, WSDLOperation.InputMessage.MessageNamespace

```

```

Set EnumSoapMappers = WSDLOperation.InputMessage.MessageParts

Do
    EnumSoapMappers.Next 1, SoapMapper, fetched
    If fetched = 1 And Not SoapMapper Is Nothing Then
        If SoapMapper.IsInput Then
            Set container = parameters(SoapMapper.PartName)
            GenerateMessageSoapByParameterContainer WSDLReader.TypeFactory, SoapMapper.ElementType, SoapMapper.XmlNamespace,
            container, SoapSerializer
        End If
    End If
Loop While fetched = 1
SoapSerializer.EndElement
SoapSerializer.EndBody
SoapSerializer.EndEnvelope
t:
Set SoapMapper = Nothing
Set EnumSoapMappers = Nothing
Set container = Nothing

End Sub

Public Function ResultOperation(WSDLOperation As MSSOAPLib30.IWSDLOperation, WSDLPort As MSSOAPLib30.IWSDLPort,
SoapTypeMapperFactory As MSSOAPLib30.ISoapTypeMapperFactory, Values As Variant) As MSSOAPLib30.SoapReader30

Dim SoapConnectorFactory As New MSSOAPLib30.SoapConnectorFactory30
Dim SoapSerializer As MSSOAPLib30.SoapSerializer30
Dim SoapConnector As MSSOAPLib30.ISoapConnector
Dim SoapReader As MSSOAPLib30.SoapReader30

On Error GoTo t
If Not WSDLOperation Is Nothing And Not WSDLPort Is Nothing And Not SoapTypeMapperFactory Is Nothing Then
    Set SoapSerializer = New SoapSerializer30
    Set SoapConnector = SoapConnectorFactory.CreatePortConnector(WSDLPort)
    SoapConnector.ConnectWSDL WSDLPort
    SoapConnector.BeginMessageWSDL WSDLOperation
    SoapSerializer.Init SoapConnector.InputStream
    GenerateMessageSoapByMemberInfo WSDLOperation, SoapTypeMapperFactory, SoapSerializer, Values
    SoapConnector.EndMessage
    Set SoapReader = New MSSOAPLib30.SoapReader30
    SoapReader.Load SoapConnector.OutputStream
    Set ResultOperation = SoapReader
End If
t:
Set SoapSerializer = Nothing
Set SoapConnector = Nothing

End Function

Public Function ExecuteOperation(WSDLOperation As MSSOAPLib30.IWSDLOperation, WSDLPort As MSSOAPLib30.IWSDLPort,
SoapTypeMapperFactory As MSSOAPLib30.ISoapTypeMapperFactory, Values As Variant) As ReturnValue

Dim SoapMapper As MSSOAPLib30.ISoapMapper

Set SoapMapper = GetSoapMapperReturn(WSDLOperation)
If Not SoapMapper Is Nothing Then
    Set ExecuteOperation = GetObjectResultBySoapReader(Me.ResultOperation(WSDLOperation, WSDLPort, SoapTypeMapperFactory,
Values), SoapTypeMapperFactory, SoapMapper, Nothing,
NewParameterContainer(SoapTypeMapperFactory.GetTypeMapperByName(SoapMapper.ElementType, SoapMapper.XmlNamespace)),
NewParameterContainer(GetTypeMapper(SoapTypeMapperFactory, "SoapError", Nothing)))
Else
    Set ExecuteOperation = NewReturnValue(Null)
End If
Set SoapMapper = Nothing

End Function

Public Function GetTypeMapper(pSoapTypeMapperFactory As MSSOAPLib30.SoapTypeMapperFactory30, TypeName As String, par_SchemaNode
As IXMLDOMNode) As MSSOAPLib30.ISoapTypeMapper

Set GetTypeMapper = helper.GetTypeMapper(pSoapTypeMapperFactory, TypeName, par_SchemaNode)

End Function

Public Function NewReturnValue(Value As Variant) As ReturnValue

Dim ret As New ReturnValue

On Error Resume Next
ret.isObject = isObject(Value)
If ret.isObject Then
    Set ret.Value = Nothing
    Set ret.Value = Value
Else
    ret.Value = Value
End If
Set NewReturnValue = ret

End Function

Public Function GetNodeSchema(ByVal Factory As MSSOAPLib30.ISoapTypeMapperFactory, XMLDOMNode As MSXML2.IXMLDOMNode) As
MSXML2.IXMLDOMNode

Dim dom As MSXML2.IXMLDOMDocument

On Error Resume Next
If Not Factory Is Nothing Then

```

```

    Select Case XMLDOMNode.baseName
        Case "eLement"
            If Not XMLDOMNode.Attributes.getNamedItem("type").nodeValue Like "xsd:*" Then Set GetNodeSchema =
Factory.GetDefinitionsNode(Split(XMLDOMNode.Attributes.getNamedItem("type").nodeValue, ":")(1),
XMLDOMNode.ownerDocument.selectSingleNode("//namespace::" & Split(XMLDOMNode.Attributes.getNamedItem("type").nodeValue,
":")(0)).nodeValue, False)
            If XMLDOMNode.Attributes.getNamedItem("type").nodeValue Like "xsd:*" Then
                Set dom = New MSXML2.DOMDocument40
                dom.LoadXml "<enXSD" & Split(XMLDOMNode.Attributes.getNamedItem("type").nodeValue, ":")(1) & "/>"
                Set GetNodeSchema = dom.childNodes(0)
                Set dom = Nothing
            End If
        End Select
    End If

End Function

Public Function GetNodeTypeSchema(XMLDOMNode As MSXML2.IXMLDOMNode, Name As String) As MSXML2.IXMLDOMNode

If Not XMLDOMNode Is Nothing Then Set GetNodeTypeSchema = XMLDOMNode.selectSingleNode(".*[@name='" & Name & "']")

End Function

Public Function GetNodeNameSchema(ByVal Factory As MSSOAPLib30.ISoapTypeMapperFactory, XMLDOMNode As MSXML2.IXMLDOMNode, Name As
String) As MSXML2.IXMLDOMNode

If Not XMLDOMNode Is Nothing Then Set GetNodeNameSchema = GetNodeSchema(Factory, GetNodeTypeSchema(XMLDOMNode, Name))

End Function

Public Function GetNode(ByVal SoapReader As MSSOAPLib30.ISoapReader, XMLDOMNode As MSXML2.IXMLDOMNode) As MSXML2.IXMLDOMNode

Set GetNode = helper.GetNode(SoapReader, XMLDOMNode)

End Function

Public Function GetNodeName(ByVal SoapReader As MSSOAPLib30.ISoapReader, XMLDOMNode As MSXML2.IXMLDOMNode, Name As String) As
MSXML2.IXMLDOMNode

Set GetNodeName = helper.GetNodeName(SoapReader, XMLDOMNode, Name)

End Function

Public Function GetResult(selection As Variant, SoapMapper As MSSOAPLib30.ISoapMapper, SoapTypeMapperFactory As
ISoapTypeMapperFactory) As SoapParser.ReturnValue

Dim SoapReader As New MSSOAPLib30.SoopReader30
Dim XMLDOMSelection As MSXML2.IXMLDOMSelection
Dim SoapTypeMapper As ISoapTypeMapper
Dim ret As SoapParser.ReturnValue
Dim mat As Variant
Dim i As Integer

If TypeName(selection) = "IXMLDOMSelection" Then
    Set XMLDOMSelection = selection
    SoapReader.Load XMLDOMSelection.context.ownerDocument
    Set GetResult = GetObjectResultBySoapReader(SoapReader, SoapTypeMapperFactory, SoapMapper, XMLDOMSelection.context,
NewParameterContainer(SoapTypeMapperFactory.GetTypeMapperByName(SoapMapper.ElementType, SoapMapper.XmlNamespace)),
NewParameterContainer(GetTypeMapper(SoapTypeMapperFactory, "SoapError", Nothing)))
ElseIf IsArray(selection) And TeElements(selection) Then
    mat = CreateArrayType(UCASE(TypeName(SoapMapper.ComValue)) = "OBJECT()")
    Set SoapTypeMapper = GetSoapTypeMapper(SoapTypeMapperFactory, SoapMapper.SchemaNode, TypeName(SoapMapper.ComValue) Like
"*(")")
    ReDim mat(LBound(selection) To UBound(selection))
    For i = LBound(selection) To UBound(selection)
        Set XMLDOMSelection = selection(i)
        SoapReader.Load XMLDOMSelection.context.ownerDocument
        If TypeName(selection(i)) = "IXMLDOMSelection" Then
            Set ret = GetObjectResultBySoapReader(SoapReader, SoapTypeMapperFactory, SoapMapper, XMLDOMSelection.context,
NewParameterContainer(SoapTypeMapper), NewParameterContainer(GetTypeMapper(SoapTypeMapperFactory, "SoapError", Nothing)))
        Else
            Set ret = NewReturnValue(selection(i))
        End If
        If ret.isObject Then
            Set mat(i) = ret.Value
        Else
            mat(i) = ret.Value
        End If
    Next
    Set GetResult = NewReturnValue(mat)
Else
    Set GetResult = NewReturnValue(selection)
End If
Set XMLDOMSelection = Nothing
Set SoapTypeMapper = Nothing
Set SoapReader = Nothing
Set ret = Nothing

End Function

Public Sub ResolverErrorHandler(str_Function As String, WebServiceError As SoapParser.SoopError, SoapClient As
MSSOAPLib30.SoopClient30)
helper.ResolverErrorHandler str_Function, WebServiceError, SoapClient
End Sub

Public Function HiHaErrorServer(ret As SoapParser.ReturnValue) As Boolean
HiHaErrorServer = helper.HiHaErrorServer(ret)

```

End Function

```
Public Function GetSoapTypeMapper(SoapTypeMapperFactory As MSSOAPLib30.ISoapTypeMapperFactory, XMLDOMNode As MSXML2.IXMLDOMNode,
Optional GetTypeItemArray As Boolean = False) As ISoapTypeMapper
```

```
Dim Value As Integer
```

```
If Not XMLDOMNode Is Nothing Then
  If Not GetTypeItemArray Then
    If XMLDOMNode.namespaceURI = "" Then
      If XMLDOMNode.parentNode Is XMLDOMNode.ownerDocument Then
        Value = helper.GetEnumValue(tli.TLIApplication.InterfaceInfoFromObject(SoapTypeMapperFactory).Parent,
"enXSDType", XMLDOMNode.baseName, -1)
      Else
        Value = helper.GetEnumValue(tli.TLIApplication.InterfaceInfoFromObject(SoapTypeMapperFactory).Parent,
"enXSDType", Me.GetNodeNameSchema(SoapTypeMapperFactory,
SoapTypeMapperFactory.GetDefinitionsNode(XMLDOMNode.parentNode.baseName, XMLDOMNode.parentNode.namespaceURI, False),
XMLDOMNode.baseName).baseName, -1)
      End If
      Set GetSoapTypeMapper = SoapTypeMapperFactory.GetMapper(Value, Nothing)
    ElseIf XMLDOMNode.namespaceURI = "http://www.w3.org/2001/XMLSchema" Then
      Set GetSoapTypeMapper = SoapTypeMapperFactory.GetTypeMapper(XMLDOMNode)
    Else
      Set GetSoapTypeMapper = SoapTypeMapperFactory.GetTypeMapperByName(XMLDOMNode.baseName, XMLDOMNode.namespaceURI)
    End If
  Else
    If Not XMLDOMNode.parentNode Is Nothing Then Set GetSoapTypeMapper = GetSoapTypeMapper(SoapTypeMapperFactory,
Me.GetNodeNameSchema(SoapTypeMapperFactory, XMLDOMNode.parentNode.parentNode,
(Split(Split(GetAttribute(XMLDOMNode.selectSingleNode("//*[ref='soapenc:arrayType']"), "wsdl:arrayType") & ":" & ":"(1) & "[",
"["(0))))))
    End If
  End If
End If
```

End Function

```
Function GetIsObject(SoapTypeMapperFactory As MSSOAPLib30.ISoapTypeMapperFactory, XMLDOMNode As MSXML2.IXMLDOMNode) As Boolean
```

```
GetIsObject = XMLDOMNode.namespaceURI = ""
```

End Function

```
Function GetElement(SoapTypeMapperFactory As MSSOAPLib30.ISoapTypeMapperFactory, ByVal par_soapreader As
MSSOAPLib30.ISoapReader, ByVal par_node As MSXML2.IXMLDOMNode, ByVal par_encoding As String, ByVal par_encodingMode As
MSSOAPLib30.enEncodingStyle, ByVal par_flags As Long) As SoapParser.ReturnValue
```

```
Dim SoapTypeMapper As ISoapTypeMapper
```

```
Dim ret As ReturnValue
```

```
Set SoapTypeMapper = GetSoapTypeMapper(SoapTypeMapperFactory, par_node)
If Not SoapTypeMapper Is Nothing Then
  Set ret = Me.NewReturnValue(SoapTypeMapper.Read(par_soapreader, par_node, par_encoding, par_encodingMode, par_flags))
  If ret.IsEmpty Then ret.InitializeEmptyValue Me.GetIsObject(SoapTypeMapperFactory, par_node)
Else
  Set ret = Me.NewReturnValue(Nothing)
End If
Set GetElement = ret
```

End Function

```
Public Function SoapTypeMapperRead(m_Factory As MSSOAPLib30.ISoapTypeMapperFactory, ByVal par_soapreader As
MSSOAPLib30.ISoapReader, ByVal par_node As MSXML2.IXMLDOMNode, ByVal par_encoding As String, ByVal par_encodingMode As
MSSOAPLib30.enEncodingStyle, ByVal par_flags As Long) As Collection
```

```
Dim llista As New Collection
```

```
Dim XMLDOMNode As MSXML2.IXMLDOMNode
```

```
On Error Resume Next
```

```
If Not par_soapreader Is Nothing And Not par_node Is Nothing Then
  For Each XMLDOMNode In par_node.selectNodes("/*")
    llista.Add Me.GetElement(m_Factory, par_soapreader, SoapParser.GetNode(par_soapreader, XMLDOMNode), par_encoding,
par_encodingMode, par_flags), UCase(XMLDOMNode.baseName)
  Next
End If
Set SoapTypeMapperRead = llista
```

End Function

```
Public Sub SoapTypeMapperWriteObject(obj As Variant, m_Factory As MSSOAPLib30.ISoapTypeMapperFactory, ByVal par_ISoapSerializer
As MSSOAPLib30.ISoapSerializer, ByVal schema_node As MSXML2.IXMLDOMNode, ByVal par_encoding As String, ByVal par_encodingMode As
MSSOAPLib30.enEncodingStyle, ByVal par_flags As Long)
```

```
Dim i As Integer
```

```
Dim node As MSXML2.IXMLDOMNode
```

```
Dim Interface As tli.InterfaceInfo
```

```
Dim ElementType As String, XmlNamespace As String
```

```
Set Interface = tli.TLIApplication.InterfaceInfoFromObject(obj)
If Not Interface Is Nothing And Not obj Is Nothing Then
  For i = 1 To Interface.Members.Count
    If Interface.Members(i).InvokeKind = INVOKE_PROPERTYGET Then
      Me.SoapTypeMapperWriteObjectMember m_Factory, par_ISoapSerializer, schema_node, par_encoding, par_encodingMode,
par_flags, Interface.Members(i).Name, CallByName(obj, Interface.Members(i).Name, INVOKE_PROPERTYGET)
    End If
  Next
End If
```

End Sub


```
Public Function SoapTypeMapperWriteObjectMember(m_Factory As MSSOAPLib30.ISoapTypeMapperFactory, ByVal par_ISoapSerializer As MSSOAPLib30.ISoapSerializer, ByVal schema_Node As MSXML2.IXMLDOMNode, ByVal par_encoding As String, ByVal par_encodingMode As MSSOAPLib30.enEncodingStyle, ByVal par_flags As Long, Name As String, Value As Variant)
```

```
Dim node As MSXML2.IXMLDOMNode
```

```
Dim ElementType As String, XmlNamespace As String
```

```
Set node = Me.GetTypeSchema(schema_Node, Name)
If Not node Is Nothing Then
    ElementType = Split(GetAttribute(node, "type") & ":", ":")(1)
    XmlNamespace = GetNameSpace(node.ownerDocument, Split(GetAttribute(node, "type") & ":", ":")(0))
    Set node = Me.GetTypeSchema(m_Factory, node)
    If Not node Is Nothing Then
        'If UCase(XmlNamespace) = UCase("http://www.w3.org/2001/XMLSchema") Then
        If helper.IsSimpleType(Value) Then
            Me.GenerateMessageSoapByParameterContainer m_Factory, Name, XmlNamespace, par_encoding, par_encodingMode, par_flags, Value)
        Else
            Me.GenerateMessageSoapByParameterContainer m_Factory, ElementType, XmlNamespace, par_encoding, par_encodingMode, par_flags, Value)
        End If
    End If
End If
```

```
End Function
```

```
Public Function SoapTypeMapperReadObject(objName As String, m_Factory As MSSOAPLib30.ISoapTypeMapperFactory, ByVal par_soapreader As MSSOAPLib30.ISoapReader, ByVal par_Node As MSXML2.IXMLDOMNode, ByVal par_encoding As String, ByVal par_encodingMode As MSSOAPLib30.enEncodingStyle, ByVal par_flags As Long) As Variant
```

```
Dim i As Integer
Dim obj As Variant
Dim Interface As tli.InterfaceInfo
Dim llista As Collection
```

```
On Error Resume Next
Set obj = CreateObject(objName)
If Not obj Is Nothing Then
    Set Interface = tli.TLIApplication.InterfaceInfoFromObject(obj)
    Set llista = Me.SoapTypeMapperRead(m_Factory, par_soapreader, par_Node, par_encoding, par_encodingMode, par_flags)
    If Not Interface Is Nothing And Not llista Is Nothing And Not obj Is Nothing Then
        For i = 1 To Interface.Members.Count
            If Interface.Members(i).InvokeKind = INVOKE_PROPERTYPUT Or Interface.Members(i).InvokeKind = INVOKE_PROPERTYPUTREF Then
                CallByName obj, Interface.Members(i).Name, Interface.Members(i).InvokeKind, GetValue(llista, UCase(Interface.Members(i).Name))
            Next
        End If
    End If
End If
```

```
Set SoapTypeMapperReadObject = obj
```

```
End Function
```

7.3.2. Factoria de Serveis

La factoria de serveis permet al Lirba carregar serveis web a partir d'un fitxer de configuració, un fitxer de configuració com el següent:

```
<?xml version="1.0"?>
<!DOCTYPE WebServices [
  <!ELEMENT WebServices (WebService*,WebServiceGlobalTypes)>
  <!ELEMENT WebService (WebServiceConnection,WebServiceContainer,WebServiceTypes)>
  <!ELEMENT WebServiceGlobalTypes (WebServiceType*)>
  <!ELEMENT WebServiceTypes (WebServiceType*)>
  <!ELEMENT WebServiceConnection EMPTY>
  <!ELEMENT WebServiceContainer EMPTY>
  <!ELEMENT WebServiceType EMPTY>
  <!ATTLIST WebService id CDATA #REQUIRED
    ref CDATA #IMPLIED>
  <!ATTLIST WebServiceConnection wsdlfile CDATA #IMPLIED
    wsmlfile CDATA #IMPLIED
    Port CDATA #IMPLIED
    Service CDATA #IMPLIED
    ref CDATA #IMPLIED>
  <!ATTLIST WebServiceContainer targetPROGID CDATA #IMPLIED
    iid CDATA #IMPLIED
    ref CDATA #IMPLIED>
  <!ATTLIST WebServiceTypes ref CDATA #IMPLIED
    OnError CDATA #IMPLIED>
  <!ATTLIST WebServiceType id CDATA #IMPLIED
    targetNamespace CDATA #IMPLIED
    targetPROGID CDATA #IMPLIED
    iid CDATA #IMPLIED
    ref CDATA #IMPLIED>
]>

<WebServices>
  <WebService id="Almacen">
    <WebServiceConnection
      wsdlfile="http://127.0.0.1/PhModiRemoting/Almacen.soap?wsdl" />
    <WebServiceContainer
      targetPROGID='PhTypeMapper.PhModiApiProxi'
      iid='{81732648-553C-4A9F-BD71-EB9FDB07D792}' />
    <WebServiceTypes OnError="ShowError"/>
  </WebService>
  <WebService id="PhModiApiLluis">
    <WebServiceConnection
      wsdlfile="http://127.0.0.1/PhModiRemoting/PhModiApi.soap?WSDL" />
    <WebServiceContainer
      ref='PhModiApi'/>
    <WebServiceTypes
      ref='PhModiApi'/>
  </WebService>
  <WebService id="PhModiApi">
    <WebServiceConnection
      wsdlfile="http://minerva:8000/phmodi/PhModiApi.soap?WSDL" />
    <WebServiceContainer
      targetPROGID='PhTypeMapper.PhModiApiProxi'
      iid='{81732648-553C-4A9F-BD71-EB9FDB07D792}' />
    <WebServiceTypes OnError="ShowError">
      <WebServiceType
        id='ParsingOcrResult'
        targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
        targetPROGID='PhTypeMapper.AbstractMapper' />
      <WebServiceType
        id='ComponentDesignator'
        targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
        targetPROGID='PhTypeMapper.ComponentDesignator' />
      <WebServiceType
        id='Area'
        targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
        targetPROGID='PhTypeMapper.Area' />
      <WebServiceType
        id='DownloadFile'
        targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
        targetPROGID='PhTypeMapper.DownloadFile' />
      <WebServiceType
        id='TempLateInfo'
        targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
        targetPROGID='PhTypeMapper.AbstractMapper' />
      <WebServiceType
        id='DocumentInfo'
        targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
        targetPROGID='PhTypeMapper.AbstractMapper' />
    </WebServiceTypes>
  </WebService>
</WebServiceGlobalTypes/>
</WebServices>
```

Bàsicament aquest fitxer, estableix els serveis i on es troben definits (normalment un arxiu wsdl), a part d'aixó també defineix les classes de mapeig; aquestes classes són les encarregades de fer la serialització dels tipus complexos d'un servei web. Un exemple seria la classe de mapeig d'àrees.

```

Option Explicit
Implements MSSOAPLib30.ISoapTypeMapper

Private m_Factory As MSSOAPLib30.ISoapTypeMapperFactory
Private m_Schema As MSXML2.IXMLDOMNode
Private m_WSMNode As MSXML2.IXMLDOMNode
Private m_xsdType As MSSOAPLib30.enXSDType

Private Sub Class_Initialize()

Set m_Factory = Nothing
Set m_Schema = Nothing
Set m_WSMNode = Nothing

End Sub

Private Sub Class_Terminate()

Set m_Factory = Nothing
Set m_Schema = Nothing
Set m_WSMNode = Nothing

End Sub

Private Function ISoapTypeMapper_Iid() As String

On Error Resume Next
ISoapTypeMapper_Iid = tli.TLIApplication.InterfaceInfoFromObject(Me).Guid

End Function

Private Sub ISoapTypeMapper_Init(ByVal par_Factory As MSSOAPLib30.ISoapTypeMapperFactory, ByVal par_Schema As
MSXML2.IXMLDOMNode, ByVal par_WSMNode As MSXML2.IXMLDOMNode, ByVal par_xsdType As MSSOAPLib30.enXSDType)

Set m_Factory = par_Factory
Set m_Schema = par_Schema
Set m_WSMNode = par_WSMNode
m_xsdType = par_xsdType

End Sub

Private Function ISoapTypeMapper_SchemaNode() As MSXML2.IXMLDOMNode

Set ISoapTypeMapper_SchemaNode = m_Schema

End Function

Private Function ISoapTypeMapper_VarType() As Long

ISoapTypeMapper_VarType = VBA.VbVarType.vbObject

End Function

Private Function ISoapTypeMapper_XsdType() As MSSOAPLib30.enXSDType

ISoapTypeMapper_XsdType = m_xsdType

End Function

Private Sub ISoapTypeMapper_Write(ByVal par_ISoapSerializer As MSSOAPLib30.ISoapSerializer, ByVal par_encoding As String, ByVal
par_encodingMode As MSSOAPLib30.enEncodingStyle, ByVal par_flags As Long, par_var As Variant)

Dim Area As PhModiClientVb6.Area
Dim SoapTypeMapper As ISoapTypeMapper

If TypeOf par_var Is PhModiClientVb6.Area Then
Set Area = par_var
If Not par_ISoapSerializer Is Nothing Then
par_ISoapSerializer.StartElement "AreaInfoNames"
Set SoapTypeMapper = SoapParser.GetTypeMapper(m_Factory, "enXSDarray", Nothing)
If Not SoapTypeMapper Is Nothing Then
SoapTypeMapper.Write par_ISoapSerializer, par_encoding, par_encodingMode, par_flags, Area.AreaInfoNames
End If
par_ISoapSerializer.EndElement
par_ISoapSerializer.StartElement "AreaInfo"
If Not SoapTypeMapper Is Nothing Then
SoapTypeMapper.Write par_ISoapSerializer, par_encoding, par_encodingMode, par_flags, Area.AreaInfo
End If
par_ISoapSerializer.EndElement
Set SoapTypeMapper = Nothing
End If
End If
Set Area = Nothing

End Sub

Private Function ISoapTypeMapper_Read(ByVal par_soapreader As MSSOAPLib30.ISoapReader, ByVal par_Node As MSXML2.IXMLDOMNode,
ByVal par_encoding As String, ByVal par_encodingMode As MSSOAPLib30.enEncodingStyle, ByVal par_flags As Long) As Variant

Dim Area As New PhModiClientVb6.Area
Dim SoapTypeMapper As ISoapTypeMapper

```

```

Dim XMLDOMNode As MSXML2.IXMLDOMNode
Dim AreaInfoNames() As String
Dim AreaInfo() As String

If Not par_soapreader Is Nothing Then
    Set SoapTypeMapper = SoapParser.GetTypeMapper(m_Factory, "enXSDarray", Nothing)
    If Not SoapTypeMapper Is Nothing Then
        'SoapTypeMapper.Init m_Factory, Nothing, Nothing, enXSDarray
        Set XMLDOMNode = SoapParser.GetNodeName(par_soapreader, par_Node, "AreaInfoNames")
        If Not XMLDOMNode Is Nothing Then AreaInfoNames = SoapTypeMapper.Read(par_soapreader, XMLDOMNode, par_encoding,
par_encodingMode, par_flags)
        Set XMLDOMNode = Nothing
    End If
    If Not SoapTypeMapper Is Nothing Then
        Set XMLDOMNode = SoapParser.GetNodeName(par_soapreader, par_Node, "AreaInfo")
        If Not XMLDOMNode Is Nothing Then AreaInfo = SoapTypeMapper.Read(par_soapreader, XMLDOMNode, par_encoding,
par_encodingMode, par_flags)
        Set XMLDOMNode = Nothing
    End If
    Area.SetValues AreaInfoNames, AreaInfo
    Set ISoapTypeMapper_Read = Area
    Set SoapTypeMapper = Nothing
End If
Set Area = Nothing

End Function

```

Finalment una altra tasca important és la reflexió. Per crear un nou servei s'ha de clonar una classe a partir de la seva interface, però a més s'ha d'evitar acoplar la factoria de serveis web, a un servei web en particular, així s'aconsegueix amb la reflexió o introspecció; dotant al sistema una forma d'averiguar les metadades dels objectes (en Java o .Net ho proporciona l'entorn de programació, en VB6 s'ha hagut d'implementar).

Option Explicit

Implements WebServices.IReflectionType

Private Interface As tli.InterfaceInfo

Private Function IReflectionType_GetMemberInfo(MemberName As String) As IReflectionType

If Not Interface Is Nothing Then Set IReflectionType_GetMemberInfo = Interface.GetMember(MemberName)

End Function

Private Function IReflectionType_GetParameterInfo(MemberName As String, ParameterName As String) As WebServices.IReflectionType

End Function

Private Function IReflectionType_GetParametersNames(MemberName As String) As String()

Dim Member As tli.MemberInfo

Dim ret() As String

Dim i As Integer

If Not Interface Is Nothing Then Set Member = Interface.GetMember(MemberName)

If Not Member Is Nothing Then

 ReDim Preserve ret(Member.Parameters.Count - 1)

 For i = 1 To Member.Parameters.Count

 ret(i - 1) = Member.Parameters(i).Name

 Next

End If

IReflectionType_GetParametersNames = ret

End Function

Private Function IReflectionType_GetMembersNames() As String()

Dim ret() As String

Dim i As Integer

If Not Interface Is Nothing Then

 For i = 1 To Interface.Members.Count

 ReDim Preserve ret(i - 1)

 ret(i - 1) = Interface.Members(i).Name

 Next

End If

IReflectionType_GetMembersNames = ret

End Function

Private Sub IReflectionType_Initialize(InterfaceInfo As Variant)

On Error Resume Next

If TypeOf InterfaceInfo Is tli.InterfaceInfo Then

 Set Interface = InterfaceInfo

Else

 Set Interface = tli.TLIApplication.InterfaceInfoFromObject(InterfaceInfo)

End If

End Sub

Private Function IReflectionType_Invoke(OperationName As String, ParamArray Values() As Variant) As Variant

```
End Function
Private Function IReflectionType_InvokeArray(OperationName As String, Values() As Variant) As Variant
End Function
Private Function IReflectionType_IsAvailable() As Boolean
IReflectionType_IsAvailable = Not Interface Is Nothing
End Function
Private Sub IReflectionType_Terminate()
Set Interface = Nothing
End Sub
```


7.3.4. Accés remot a l'arxiu òptic

Aquí tenim la implementació de la connexió a l'arxiu òptic així com funcions que encapsulen determinades funcionalitats. Les interfícies d'usuari treballen amb aquestes funcions.

```

Subí EstablirConexioArxiuOpticPh()
On Error GoTo EstablecerConexionAO

Dim WebServiceContainer As IWebServiceContainer
Dim PhModi As PhModiClientVb6.IPhModiApi
Dim obj As Variant

frm_conexiones.pan_avis.Panels(1).text = "Conectando con el Servicio Web de Archivo Optico."
If ConexionAO Then TancarConexioArxiuOptic
If CadenaConexioAO <> "" Then
    'Set WebServiceFactory = CreateObject("WebServiceConfig.WebServiceFactory")
    Set obj = CreateObject("WebServiceConfig.WebServiceFactory")
    If TypeOf obj Is WebServices.IWebServiceFactory Then
        Set WebServiceFactory = obj
        WebServiceFactory.Initialize RetPath(NewGenINI) & "WebServices.xml"
        If WebServiceFactory.IsAvailable Then
            Set WebServiceContainer = WebServiceFactory.WebServiceContainer(CadenaConexioAO)
            If Not WebServiceContainer Is Nothing Then
                If WebServiceContainer.WebServiceProxi.IsAvailable Then
                    If TypeOf WebServiceContainer Is PhModiClientVb6.IPhModiApi Then
                        Set PhModi = WebServiceContainer
                        If TypeOf WebServiceContainer Is PhModiClientVb6.IApiArxiuOptic Then 'Lcr AO New Version
                            Set PhModiApi = PhModi
                            ConexionAO = PhModi.IsAvailable
                            If Not ConexionAO Then
                                TancarConexioArxiuOptic
                                ngmsgbox "EL Servicio Web no esta disponible", MB_ICONINFORMATION, "ATENCI"
                            End If
                        Else
                            TancarConexioArxiuOptic
                            ngmsgbox "No se puede instanciar Servicio Web", MB_ICONINFORMATION, "ATENCI"
                        End If 'Lcr AO New Version
                    Else
                        TancarConexioArxiuOptic
                        ngmsgbox "No se puede crear Servicio Web", MB_ICONINFORMATION, "ATENCI"
                    End If
                Else
                    TancarConexioArxiuOptic
                    ngmsgbox "EL Contenedor del Servicio Web no esta disponible", MB_ICONINFORMATION, "ATENCI"
                End If
            Else
                TancarConexioArxiuOptic
                ngmsgbox "No se encuentra La configuraciùn del Servicio Web", MB_ICONINFORMATION, "ATENCI"
            End If
        Else
            TancarConexioArxiuOptic
            ngmsgbox "El Proxi del Servicio Web no esta disponible", MB_ICONINFORMATION, "ATENCI"
        End If
        If Not ConexionAO Then
            TancarConexioArxiuOptic
            ngmsgbox "Servicio Web no disponible", MB_ICONINFORMATION, "ATENCI"
        End If
    Else
        TancarConexioArxiuOptic
        ngmsgbox "No se puede crear el Proxi del Servicio Web", MB_ICONINFORMATION, "ATENCI"
    End If
Else
    ConexionAO = False
End If
frm_conexiones.pan_avis.Panels(1).text = ""
Exit Sub

EstablecerConexionAO:
    ConexionAO = False
    frm_conexiones.pan_avis.Panels(1).text = ""
    'ngmsgbox "No se ha podido establecer La conexiùn con La BD documental", MB_ICONINFORMATION, "ATENCI"
    ngmsgbox "No se ha podido establecer La conexiùn con La BD documental" & Chr(13) & Err.Description, MB_ICONINFORMATION,
    "ATENCI"

End Sub

Public Function GeneraEtiqueta(EntornArchiuOptic As PhModiClientVb6.IEntornAO) As PhModiClientVb6.IdocInfo

    Dim ao As PhModiClientVb6.IGenericAO

    If Not CalGenerarEtiqueta(EntornArchiuOptic) Then Exit Function
    Set ao = EntornArchiuOptic.FactoriaManegadorArxiuOptic
    If ao Is Nothing Then Exit Function
    Set GeneraEtiqueta = ao.CreaDocument(StatProces(EntornArchiuOptic, True))

End Function

Public Function BorraVinclesAo(frm As Form, idproces As String, HDBC As Integer, documentos() As LinkAo) As Boolean

Dim i As Integer
    
```

```

Dim ERROR As Boolean
Dim EntornArxiuOptic As PhModiClientVb6.IEntornAO

For i = LBound(documentos) To UBound(documentos)
    If Not documentos(i).IdAo = "" Then
        Set EntornArxiuOptic = FactoriaEntornArchiuOptic(frm, idproces, documentos(i).tipusdoc, HDDBC)
        ERROR = Not DestruirVincle(EntornArxiuOptic, documentos(i).IdAo, documentos(i).IdLirba)
        If AcceptaOperacioArxiuOptic(EntornArxiuOptic, AccioArxiuOptic.CanviEstat) And Not ERROR Then CanviStat
    EntornArxiuOptic, documentos(i).IdAo, StatProces(EntornArxiuOptic, False), StatProces(EntornArxiuOptic, True)
        If ERROR Then Exit Function
    End If
Next
BorraVinclesAo = Not ERROR

End Function

Public Function BuscaIdentificadorDocumentAO(EntornArchiuOptic As PhModiClientVb6.IEntornAO, vaLors() As IVariabLeValuada) As String

    If EntornArchiuOptic Is Nothing Then Exit Function
    If tamany(vaLors) = 0 Then vaLors = ValorsVariablesFormulari(EntornArchiuOptic, VariabLesTipusDocument(EntornArchiuOptic, NomesKey))
    BuscaIdentificadorDocumentAO = KeyAo(EntornArchiuOptic, vaLors)

End Function

Public Function BuscaDocuments(EntornArchiuOptic As PhModiClientVb6.IEntornAO, Estat As String) As String()

    If EntornArchiuOptic Is Nothing Then Exit Function
    Dim ao As PhModiClientVb6.IGenericAO
    Set ao = EntornArchiuOptic.FactoriaManegadorArxiuOptic
    If ao Is Nothing Then Exit Function
    BuscaDocuments = ao.BuscaDocuments(Estat)

End Function

Private Sub MostrarPDFWebBrowser(EntornArchiuOptic As PhModiClientVb6.IEntornAO, id_ao As String)

Dim doc As PhModiClientVb6.DownloadFile
Dim tmp As String, ruta As String
Dim i As Integer
Dim browser_abierto As Boolean
Dim ext As Integer
Dim fic As String, F As String
Dim pun As Integer
Dim ao As IGenericAO
Dim fsis As Object

Set fsis = CreateObject("Scripting.FileSystemObject")
'ParsingOCR(PhSystems.PHModi.WinClient.Configuration.Singleton.GetPHModiapi.NewDocument(Me.nom_plantilla.text, ""),
Me.GetImage)
If Not EntornArchiuOptic Is Nothing Then Set ao = EntornArchiuOptic.FactoriaManegadorArxiuOptic
If ao Is Nothing Then Exit Sub
Set doc = ao.BuscaDocument(id_ao)
If doc Is Nothing Then
    Exit Sub
End If
If Not doc.IsAvailable Then
    Exit Sub
End If
On Error Resume Next
If Dir(Leer_datgen("", "", "PATHTMP"), vbDirectory) = "" Or Len(Leer_datgen("", "", "PATHTMP")) = 0 Then
    If inis32.EsIniTemp Then
        If Dir(Leer_datgen("", "", "PATINIT"), vbDirectory) = "." Or Dir(Leer_datgen("", "", "PATHTMP"), vbDirectory) = ""
Then
            ruta = RetPath(NewGenINI) & "." & Leer_datgen("", "", "PATINIT") 'Temporal Inis
        Else
            ruta = Leer_datgen("", "", "PATINIT") & USUARIO & "/"
        End If
    Else
        ruta = apis32.GetTempPath()
    End If
Else
    ruta = Leer_datgen("", "", "PATHTMP")
End If
If Leer_datgen("", "", "AOTMP") <> "" Then
    ruta = Leer_datgen("", "", "AOTMP") & USUARIO & "/"
End If
tmp = ".t.pdf"
If Not doc.IsAvailable Then
    Exit Sub
End If
browser_abierto = False
For i = 0 To Forms.Count - 1
    If Forms(i).LINKTOPIC = "frm_browser" Then
        browser_abierto = True
    Exit For
End If
Next i
'Revisem existencia fitxers
pun = InStr(tmp, ".")
If pun > 0 Then tmp = Left$(tmp, pun - 1)
ext = -1
Do While True
    ext = ext + 1
    fic = tmp & format$(ext, "00") & ".pdf"
    If Dir$(ruta & fic) = "" Then
        If browser_abierto Then

```



```
        tmp = fic
    End If
    Exit Do
Else
    If Not browser_abierto Then Kill ruta & fic
End If
Loop
If Not browser_abierto Then tmp = "t00.pdf"
If Dir$(ruta & tmp) <> "" Then
    fsys.DeleteFile (ruta & tmp)
End If
If doc.SaveAs(ruta & tmp) Then
    'EncapsulacioPdf ruta & tmp, "", ""
'    ruta = "c:\tmp"
'    tmp = "\t00.pdf"
'    WaitOnProgram(shell "ImageToPdf.exe " & ruta & tmp & " " & ruta & tmp)
Dim frm As New frm_browser
frm.Lbl_parametro = ruta & tmp
frm.Show
Set frm = Nothing
End If
End Sub
```

7.4. Mòduls Implementats

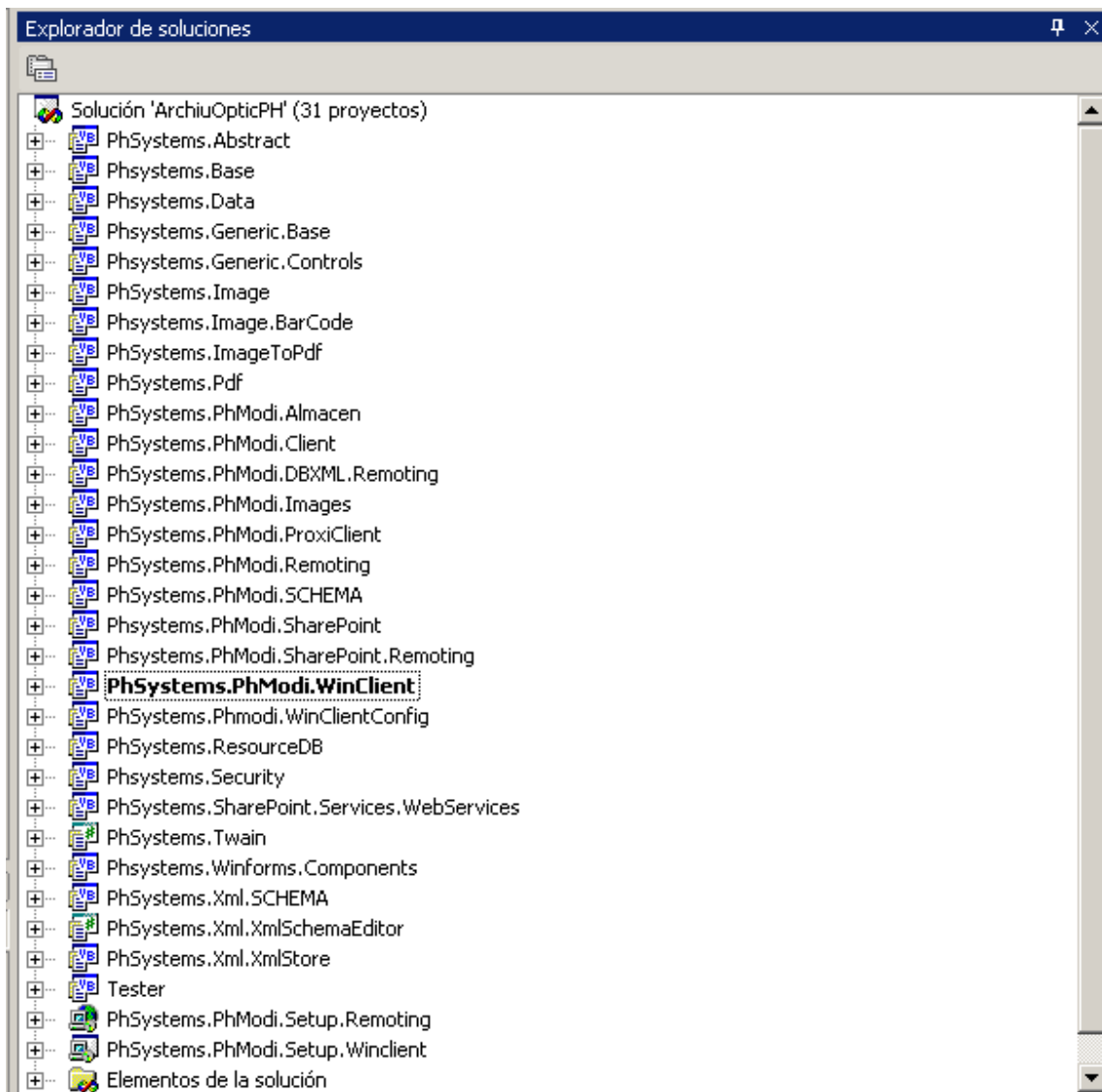
Ara es farà una breu descripció de tots els mòduls implementats en aquest projecte. Bàsicament la implementació es divideix en dos grans subprojectes:

- L'Arxiu Òptic
- El Lirba

7.4.1. Mòduls Arxiu Òptic

En aquest subprojecte s'implementen, tant l'arxiu òptic en si, com el client que s'emprarà per administrar l'arxiu òptic. Com es pot veure en l'imatge inferior, són 31 projectes fets en .Net.

La tipologia d'aquests projectes és diversa, hi ha: executables, llibreries, paquets d'instal·lació i proves. El llenguatge emprat majoritàriament ha estat el VB, tot i que també s'ha utilitzat el C#.



PhSystems.Abstract - Aquesta llibreria conté, totes les abstraccions e interfaces necessàries pels serveis remots, inclou entre altres: col·leccions, relacions, processos, persistència, recursos, transaccions, factories i contenidors. Els objectes aquí presents no són finals requereixen implementar mètodes específics a un cas concret que és realitza en altres llibreries

PhSystems.Base - Aquesta llibreria conté, factories sobre relacions, recursos i col·leccions. És una particularització de la llibreria abstracte que permet crear la base del component remot; fonamentalment treballa sobre serialització XML i adaptadors de recursos en sistema de fitxers.

PhSystems.Data - Aquesta llibreria conté, per una part una abstracció de la utilització de recursos en BD, i per altra banda una implementació simple d'aquesta abstracció. Això permet realitzar un recurs persistent en BD, configurable via una classe serializable en XML.

Un exemple seria:

```
<?xml version="1.0"?>
<ArrayOfParam xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Param Name="IBuilderDBExecute" Value="PhSystems.Data.Simple.BuilderDBExecute,Phsystems.Data, Version=1.0.2628.15835, Culture=neutral, PublicKeyToken=null" />
  <Param Name=".IDbConnection" Value="System.Data.OracleClient.OracleConnection,System.Data.OracleClient, Version=1.0.5000.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  <Param Name=".Connection" Value="user id=phges;data source=orcl;password=phges" />
  <Param Name=".Behavior" Value="Default" />
  <Param Name=".DbType" Value="String" />
  <Param Name=".Direction" Value="1" />
  <Param Name=".SourceVersion" Value="Default" />
  <Param Name=".DynamicProperties" Value="OracleType" />
  <Param Name=".OracleType" Value="VarChar" />
  <Param Name=".TypeDoc.SourceColumn" Value="Docume_cod" />
  <Param Name=".Uri.SourceColumn" Value="Uri" />
  <Param Name=".DocCod.SourceColumn" Value="Id_docA0" />
  <Param Name=".ElementName.SourceColumn" Value="Esq_cod" />
  <Param Name=".Value.SourceColumn" Value="Valor" />
  <Param Name=".Description.SourceColumn" Value="Descripcion" />
  <Param Name="ExistTypeDoc.CommandText" Value="select count(*) from DOCTYPEESQ where Docume_cod=:TypeDoc" />
  <Param Name="GetElementNameByTypeDoc.CommandText" Value="select Esq_cod from DOCTYPEESQ where Docume_cod=:TypeDoc" />
  <Param Name="PutElementNameByTypeDoc.CommandText" Value="Insert into DOCTYPEESQ (ESQ_COD,DOCUME_COD) values(:ElementName,:TypeDoc)" />
  <Param Name="CountElementNameByTypeDoc.CommandText" Value="Select count(*) From DOCTYPEESQ where DOCUME_COD=:TypeDoc" />
  <Param Name="UpdateElementNameByTypeDoc.CommandText" Value="Update DOCTYPEESQ set ESQ_COD=:ElementName where DOCUME_COD=:TypeDoc" />
  <Param Name="GetTypeDoc.CommandText" Value="select Docume_cod from DOCTYPEESQ" />
  <Param Name="PutUriTypeDoc.CommandText" Value="Insert into DOCTYPEURIDES (DOCUME_COD,DESCRIPCION,URI) values(:TypeDoc,:Description,:Uri)" />
  <Param Name="CountUriTypeDoc.CommandText" Value="select count(*) From DOCTYPEURIDES where DOCUME_COD=:TypeDoc and URI=:Uri" />
  <Param Name="UpdateUriTypeDoc.CommandText" Value="Update DOCTYPEURIDES set DESCRIPCION=:Description where DOCUME_COD=:TypeDoc and URI=:Uri" />
  <Param Name="GetDescriptionUriTypeDoc.CommandText" Value="select descripcion from DOCTYPEURIDES where Docume_cod=:TypeDoc and Uri=:Uri" />
  <Param Name="GetUriTypeDoc.CommandText" Value="select Uri from DOCTYPEURIDES where Docume_cod=:TypeDoc" />
  <Param Name="PutValueTypeDoc.CommandText" Value="Insert into DOCVALUES (ID_DOCA0,VALOR,URI) values(:DocCod,:uri,:Value)" />
  <Param Name="CountValueTypeDoc.CommandText" Value="select count(*) from DOCVALUES where ID_DOCA0=:DocCod and URI=:uri" />
  <Param Name="UpdateValueTypeDoc.CommandText" Value="update DOCVALUES set VALOR=:Value where ID_DOCA0=:DocCod and URI=:uri" />
  <Param Name="GetValueUriTypeDoc.CommandText" Value="select Valor from DOCVALUES where Id_docA0=:DocCod and Uri=:Uri" />
  <Param Name="DeleteUriTypeDoc.CommandText" Value="delete from DOCTYPEURIDES where DOCUME_COD=:typeDoc and URI=:uri" />
  <Param Name="DeleteValueTypeDoc.CommandText" Value="delete from DOCVALUES where ID_DOCA0=:DocCod and URI=:uri" />
  <Param Name="DeleteTypeDoc.CommandText" Value="delete from DOCTYPEESQ where DOCUME_COD=:typeDoc" />
  <Param Name="GetDocumentQuery.CommandText" Value="" />
</ArrayOfParam>
```

Així per exemple, podriem definir que la col·lecció d'elements XSLT agafes un recurs en BD implementat per un constructor simple configurat amb un fitxer (semblant a l'anterior) ubicat en un determinat lloc.

```
<XSLTCollection DataBaseAdapter="PhSystems.Base.Resource.SystemIOAdapterResource,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null">
  <Relation ResourceName="XSLTFactoryRelationsSchemas.Config"
  FactoryClassName="PhSystems.Base.Relations.DataBaseFactoryRelations,PhSystems.ResourceDB, Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null"
  AdapterResourceClassName="PhSystems.Base.Resource.DataBaseAdapterResource,PhSystems.ResourceDB, Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null">
    <Configuration Name="PhSystems.Data.Simple.BuilderParams.config"
    ObjectClassName="PhSystems.Data.Simple.BuilderParams,Phsystems.Data, Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null" />
  </Relation>
  <Key>
    <Configuration Name="" ObjectClassName="" />
  </Key>
  <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
  ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null" />
</XSLTCollection>
```

No s'està limitat a BD també es poden emprar d'altres mètodes com el sistema de fitxers:

```
<DefaultImagesCollection                               DataBaseAdapter="PhSystems.Base.Resource.SystemIOAdapterResource,PhSystems.Base,
Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null">
  <Relation>
    <Configuration Name="" ObjectClassName="" />
  </Relation>
  <Key>
    <Configuration Name="" ObjectClassName="" />
  </Key>
  <DataBaseConfig                                     Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
</DefaultImagesCollection>
```

PhSystems.Generic.Base.- Aquesta llibreria conté tot el que pugui necessitar una aplicació, el classLoader, envolcalls, iteradors, funcions de conversió, serielitzadors, enumeradors, etc...

PhSystems.Generic.Controls.- Aquesta llibreria conté funcionalitat reutilitzable sobre menús i botones.

PhSystems.Image.- Aquesta llibreria conté proveïdors i enumeradors sobre imatges, a part de funcions de conversió sobre aquestes.

PhSystems.Image.Barcode.- Aquesta llibreria proporciona les interfaces necessàries per poder processar imatges amb codis de barres. (Desacopla aquesta funcionalitat de la implementació feta).

PhSystems.ImageToPdf.- Aquest executable de comandes estil script permet a d'altres aplicacions, que no puguin fer-ho, generar imatges a partir de pdf.

PhSystems.Pdf.- Aquesta llibreria proporciona la funcionalitat bàsica per generar i treballar sobre pdf.

PhSystems.Generic.Controls.- Aquesta llibreria conté funcionalitat reutilitzable sobre menús i botones.

PhSystems.Image.- Aquesta llibreria conté proveïdors i enumeradors sobre imatges, a part de funcions de conversió sobre aquestes.

PhSystems.Image.Barcode.- Aquesta llibreria proporciona les interfaces necessàries per poder processar imatges amb codis de barres. (Desacopla aquesta funcionalitat de la implementació feta).

PhSystems.ImageToPdf.- Aquest executable de comandes estil script permet a d'altres aplicacions, que no puguin fer-ho, generar imatges a partir de pdf.

PhSystems.Pdf.- Aquesta llibreria proporciona la funcionalitat bàsica per generar i treballar sobre pdf.

PhSystems.PhModi.Almacen.- Aquesta llibreria implementa un magatzem virtual, sobre un arxiu físic, basat en lots i caixes.

PhSystems.PhModi.Client.- Aquesta llibreria conté totes les interfaces i objectes públics de l'arxiu òptic, d'aquesta manera una aplicació només requereix una referència a aquesta llibreria per treballar amb l'Arxiu Òptic. (Cal recordar que l'Arxiu Òptic no necessàriament ha de funcionar sobre Soap)

PhSystems.PhModi.DBXML.Remotig.- Aquesta llibreria implementa un adaptador de recursos en XmlDB

PhSystems.PhModi.Images.- Aquesta llibreria conté una implementació d'un iterador d'imatges sobre un enumerador Twain o FileSystem i una implementació del Barcode sobre una llibreria comercial anomenada AtalaSoft

PhSystems.PhModi.ProxiClient.- Aquesta llibreria és un proxy que proporciona funcionalitats reduïdes del sistema.

PhSystems.PhModi.Remoting.- Aquesta llibreria implementa els serveis remots disponibles així com tots els objectes de configuració i elements abstractes lligats a la funcionalitat del sistema.

PhSystems.PhModi.SCHEMA.- Aquesta llibreria conté la funcionalitat de l'arxiu òptic sobre esquemes i plantilles OCR.

PhSystems.PhModi.SharePoint.- Aquesta llibreria conté funcionalitat bàsica sobre SharePoint.

PhSystems.PhModi.SharePoint.Remoting.- Aquesta llibreria implementa un adaptador de recursos en SharePoint.

PhSystems.PhModi.WinClient.- Aquesta aplicació conté les interfícies d'usuari per l'adquisició i administració de documents.

PhSystems.PhModi.WinclientConfig.- Aquesta llibreria conté la funcionalitat bàsica de configuració de l'aplicació.

PhSystems.ResourceDB.- Aquesta llibreria implementa un adaptador de recursos en BD.

PhSystems.Security.- Aquesta llibreria conté la funcionalitat bàsica de polítiques de seguretat i autenticació.

PhSystems.SharePoint.Services.WebServices.- Aquesta llibreria conté una encapsulació dels Serveis Web del SharePoint.

PhSystems.Twain.- Implementació d'enumeradors e iteradors sobre Twain.

PhSystems.WinForms.Components.- Implementació bàsica de diferents components d'interfície reutilitzable.

PhSystems.Xml.SCHEMA.- Aquesta llibreria conté tota la funcionalitat sobre Schema i XML (en un apartat anterior d'aquest capítol ja s'ha mostrat el codi de generació de XML a partir d'un Schema)

PhSystems.Xml.XmlSchemaEditor.- Aquesta llibreria conté les interfícies d'usuari d'un editor d'esquemes Schema XML.

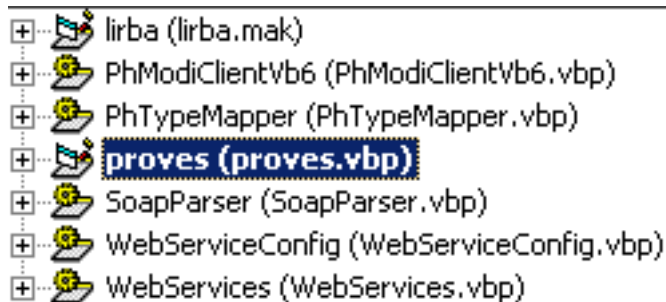
PhSystems.XmlStore.- Aquesta llibreria conté la base per generar una adaptador de recursos en una XMLDB.

PhSystems.PhModi.Setup.Remoting.- Aquesta aplicació realitza la instal·lació i configuració dels serveis remots en un ordinador amb IIS.

PhSystems.PhModi.Setup.WinClient Aquesta aplicació realitza la instal·lació i de l'aplicació client.

7.4.2. Mòduls Lirba

En aquest subprojecte s'implementen tant la integració de l'Arxiu Òptic, com els marcs de treball de la factoria de serveis i el Record Management.



lirba.- En aquesta aplicació hi ha les interfícies d'usuari d'administració del marc de treball del Record Management.

PhModiClientVb6.- En aquesta llibreria hi ha les interfaces i objectes públics de l'arxiu Òptic.

PhTypeMapper.- En aquesta llibreria hi ha els objectes de mapeig sobre diferents objectes públics.

SoapParser.- En aquesta llibreria hi ha la implementació de generació de missatges SOAP.

WebserviceConfig.- En aquesta llibreria hi ha la implementació d'una factoria de Serveis Web basada en serialització XML.

WebServices.- En aquesta llibreria hi ha la definició abstracta d'una factoria de Serveis Web.

Resta Aplicacions.- En total el Lirba Standard són cap a una quarantena d'aplicacions, cadascuna en un projecte Vb6. A més cada client té una copia de cada projecte que utilitza, sigui del Standard o no. En cadascun d'aquests projectes hi ha uns mòduls (i classes) estàndard que ha de tenir. Dins d'aquesta categoria entren el marc de treball de recursos, el de BD i les classes del Record Management (així com els mòduls que aïllen la seva funcionalitat, el Lirba és basa en crida a funcions)

8. Interfícies d'usuari

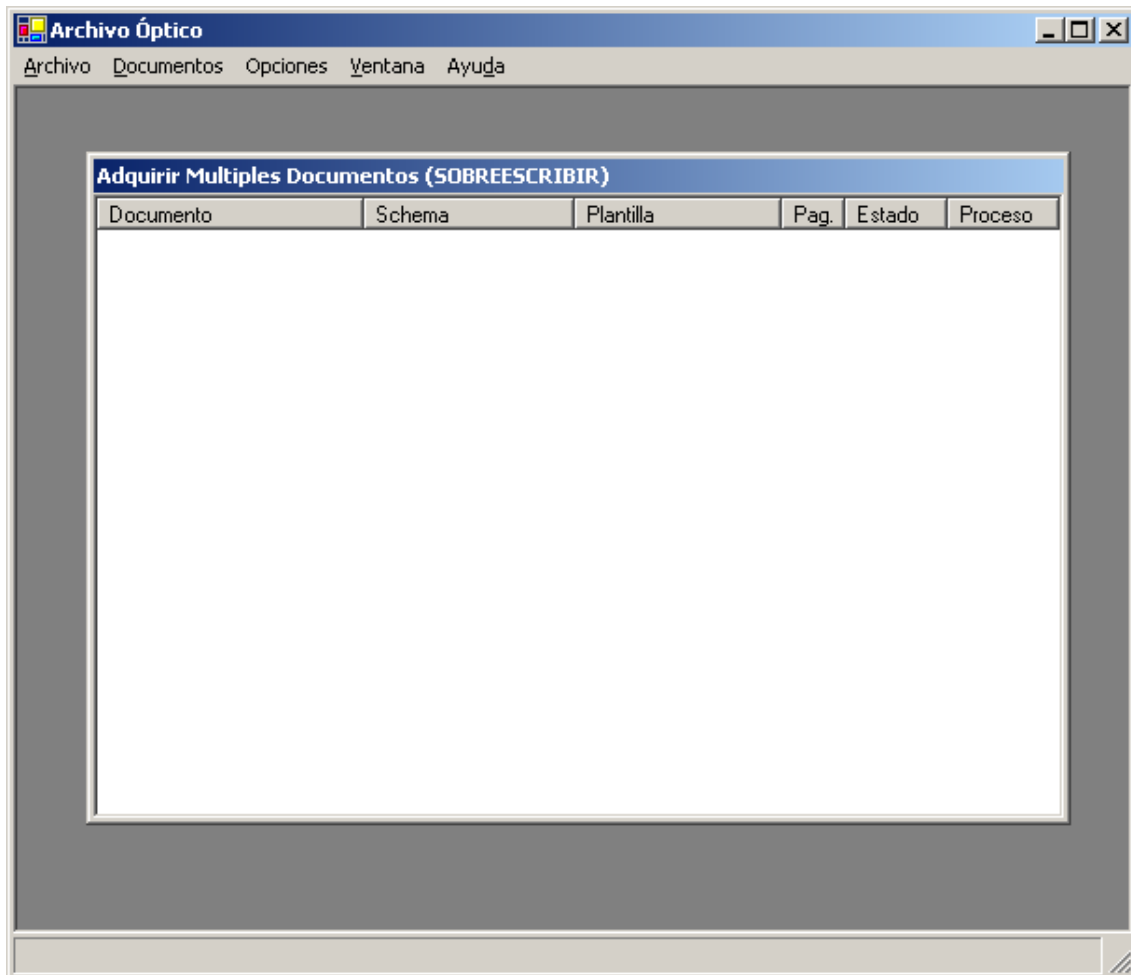
8.1. Administració i adquisició de documents.

8.1.1. Adquisició de documents.

La interfície d'usuari que permet l'adquisició de forma massiva de documents, és la que es pot observar en la figura de sota.

Aquesta interfície està basada en la interfície de llista d'elements, és molt simple i permet poques opcions: adquirir documents, escollir l'origen (escàner o directori) i configurar l'escàner.

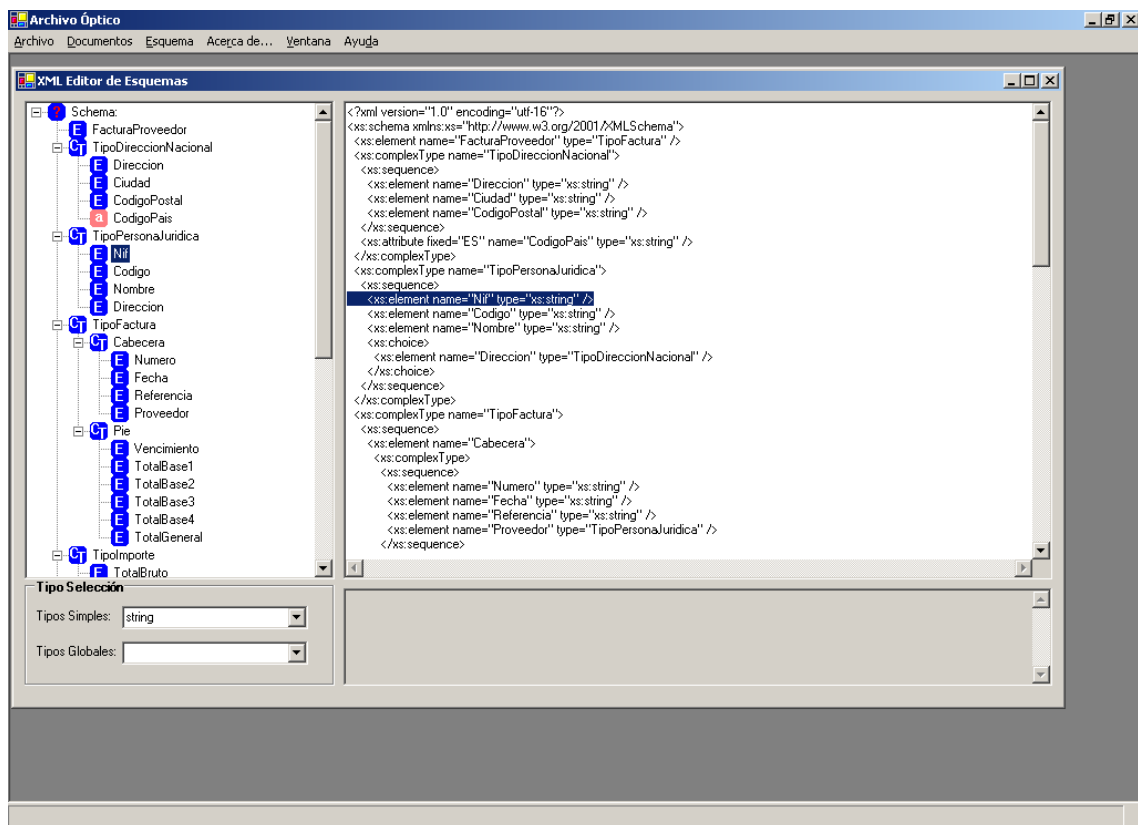
Un cop posat en marxa el procés d'adquisició d'imatges, a mesura que el sistema les va processant e identificant a partir del seu codi de barres, aquestes van apareixent a la llista. Alhora el sistema les va introduint a l'Arxiu Òptic.



8.1.2. Editor d'esquemes.

A part de l'adquisició de documents hi ha dues tasques més a realitzar, creació d'esquemes i de plantilles. Aquest editor, és una interfície d'usuari que permet crear i modificar un SCHEMA Xml, està dividit en 3 parts.

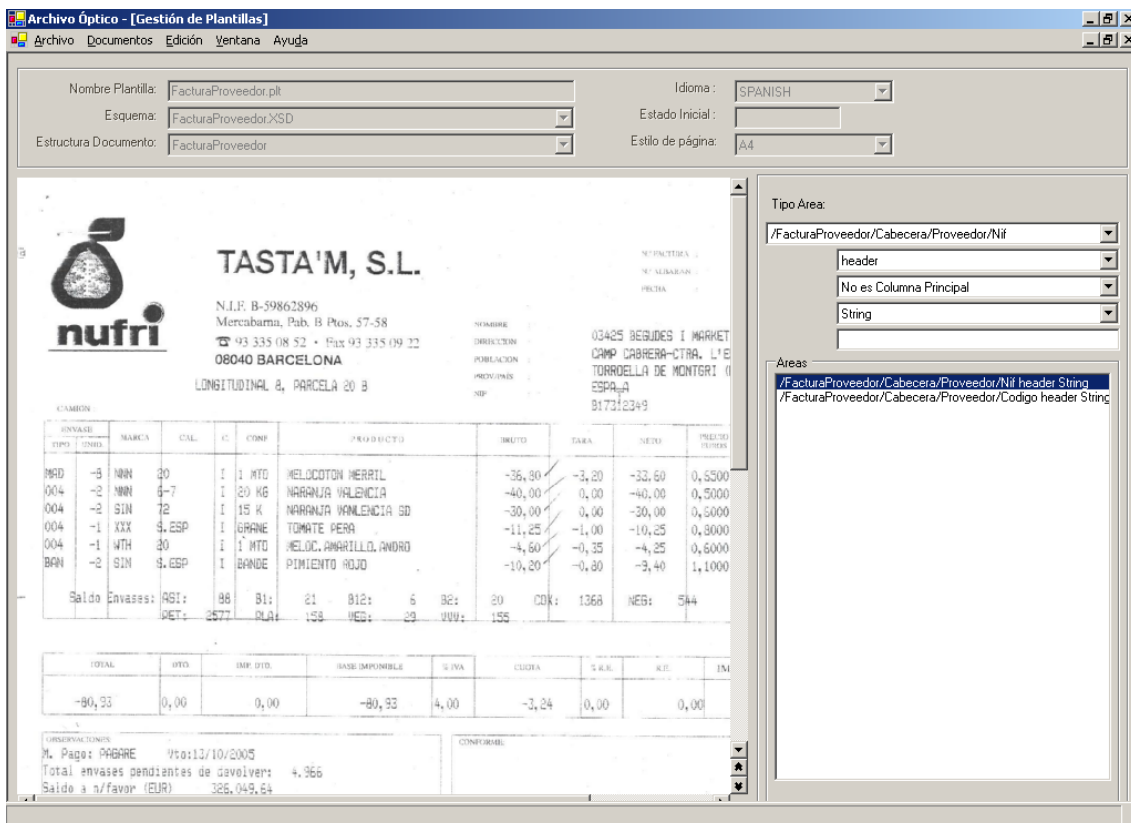
- Arbre d'elements: que permet recorre la seva estructura (i afegir nous elements), si es selecciona un element, les altres parts de la interfície reaccionen a aquest event.
- Editor de text: permet modificar l'esquema com un simple fitxer de text, tot i que és més recomenable anar afegint elements a l'arbre i modificar-lo en aquest editor.
- Informació: aquest element mostra informació sobre els tipus, i també sobre l'error de sintaxi si es valida l'esquema.



8.1.3. Editor de plantilles.

Per tal de poder crear plantilles, s'ha implementat aquesta interfície d'usuari. Està dividida en 3 elements.

- Informació plantilla: En aquest element, hi ha la informació de la plantilla, com l'esquema emprat, l'estil de paginació o l'idioma a utilitzar amb l'ORC
- Visualització document: Per tal de poder generar les àrees, en aquest element és visualitza un document de mostra. L'usuari pot seleccionar amb el ratolí àrees de píxels.
- Informació àrea: En aquest element és relaciona una àrea de píxels seleccionada amb un Uri disponible de l'esquema seleccionat.



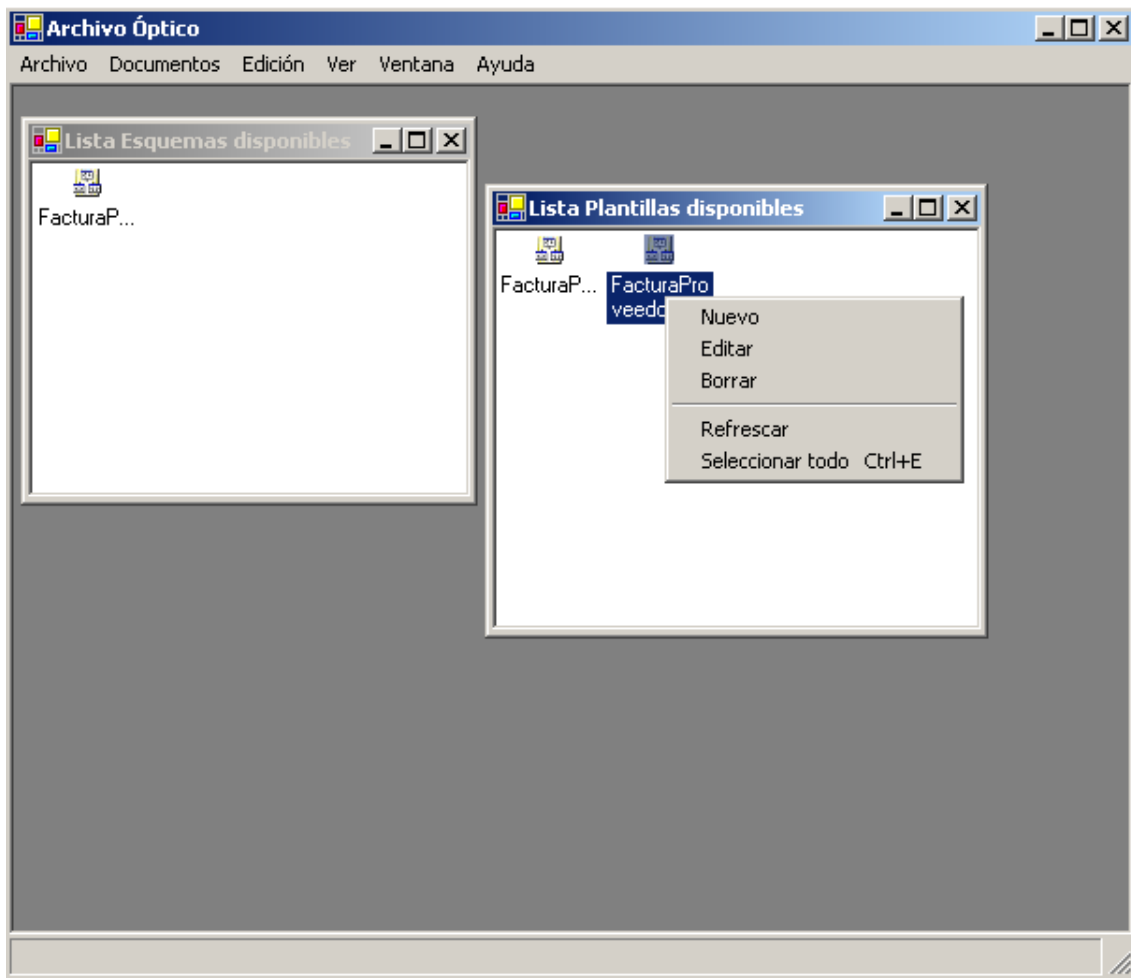
Amb aquest editor, es poden diferenciar diferents tipus d'àrea, corresponents a: la capçalera, el cos, o el peu. L'Arxiu Òptic és capaç de crear estructures repetitives, i treballar amb documents multipàgina; però té moltes limitacions en aquest sentit. Aquesta limitació és deu a l'ús de l'API Modi del MS Office, però en tot cas el sistema és escalable i es poden emprar en un futur altres Api's.

8.1.4. Lista elements.

Un dels components més important a nivell d'interfície d'usuari d'aquesta aplicació, són les llistes. Bàsicament aquesta aplicació treballa amb conjunts d'elements, ja siguin: documents electrònics, plantilles, esquemes, imatges, etc...

Per tant es va implementar una interfície d'usuari que fos una llista d'elements, quan des de les opcions de menú es crida a qualsevol procediment de gestió, aquest el que fa és proporcionar una llista amb els elements a gestionar i les opcions específiques de gestió segons el tipus d'element.

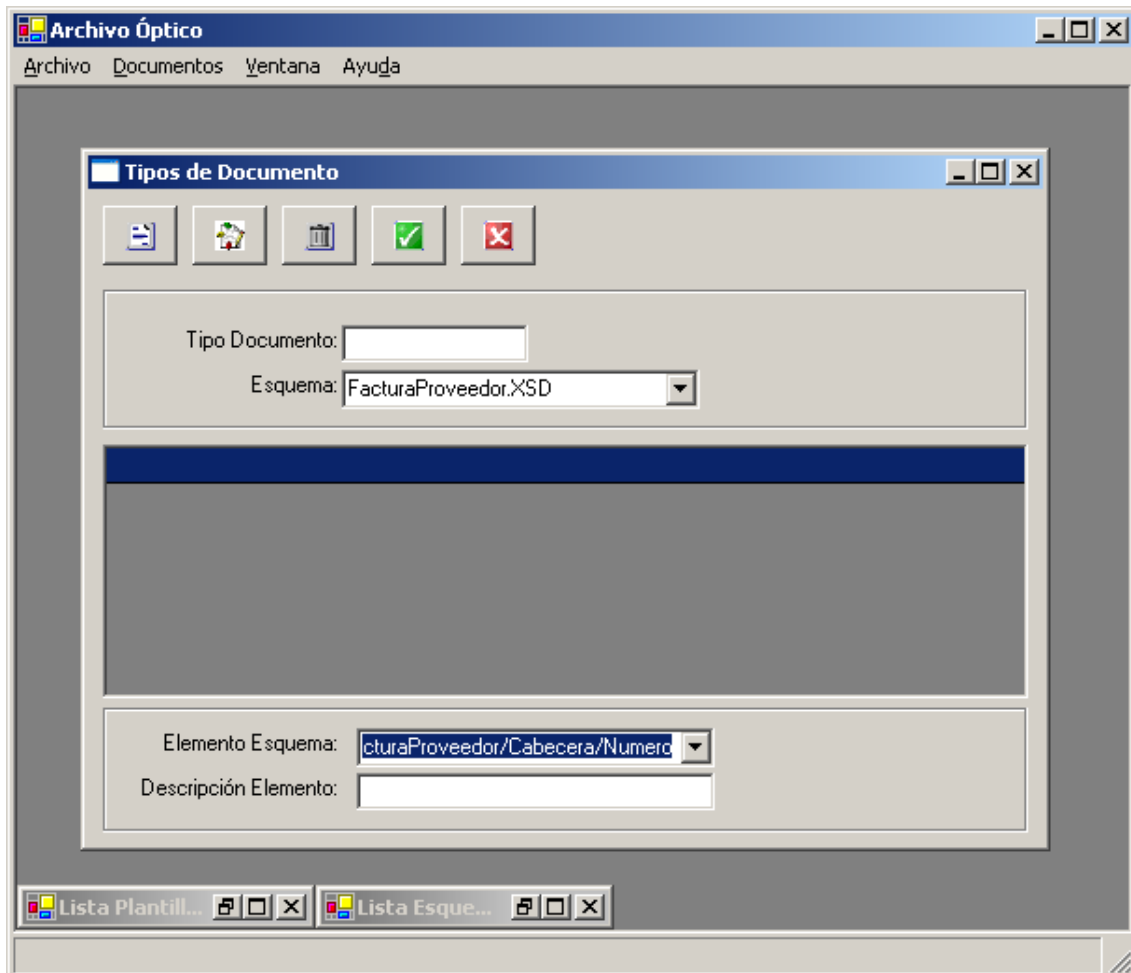
Com ja s'ha vist anteriorment no és responsabilitat d'aquesta interfície saber les accions que pot realitzar, sinó que el controlador informa a la llista de quines opcions ha de visualitzar.



8.1.5. Definir Tipus documents.

Per realitzar la integració amb aplicacions de gestió, cal una interfície d'usuari per tal de poder definir tipus de document i associar la informació dels flux de treball amb les dades d'un esquema.

Així doncs en aquesta interfície hi ha elements que no són propis de l'arxiu òptic, com que depenen de cada instal·lació aquesta pantalla és diferent per cada organisme (tot i que n'hi ha una de genèrica basada en fitxers de configuració).

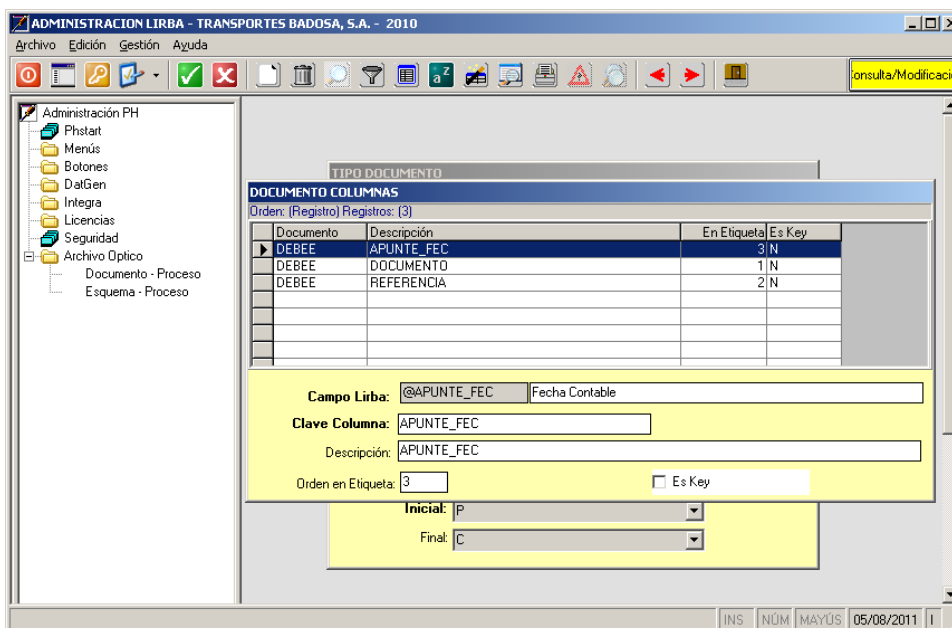
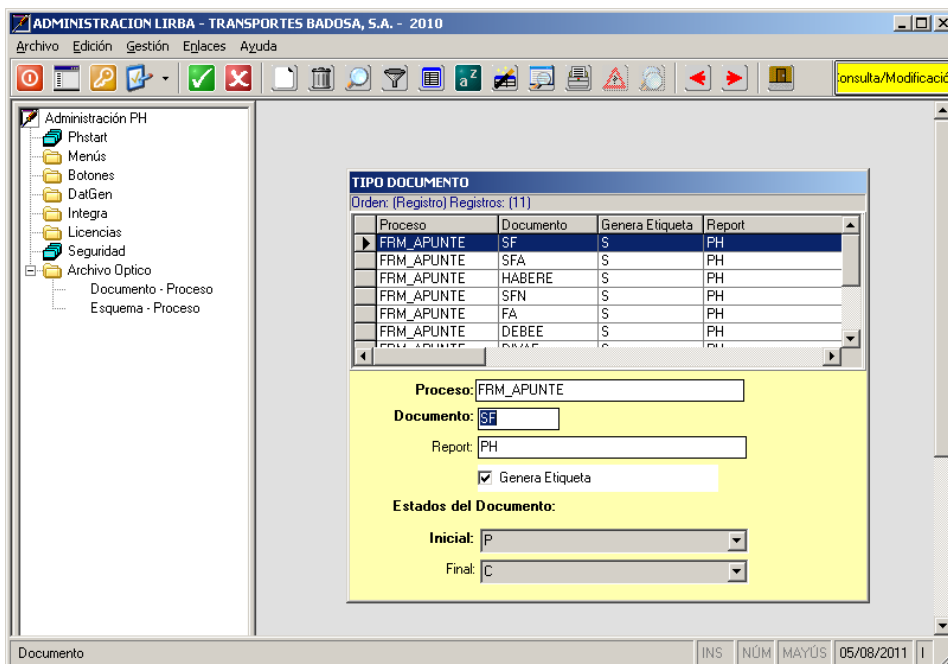


8.2. Integració amb el Lirba

Les interfícies d'usuari en el Lirba són totes de 4 tipus: mestre/detall, mestre, mestre/l·lista o l·lista. En aquest cas les funcionalitats s'han realitzat amb pantalles tipus mestre/l·lista. On a la part de dalt hi ha una l·lista amb els elements ja entrats i la part de sota conté les dades del manteniment.

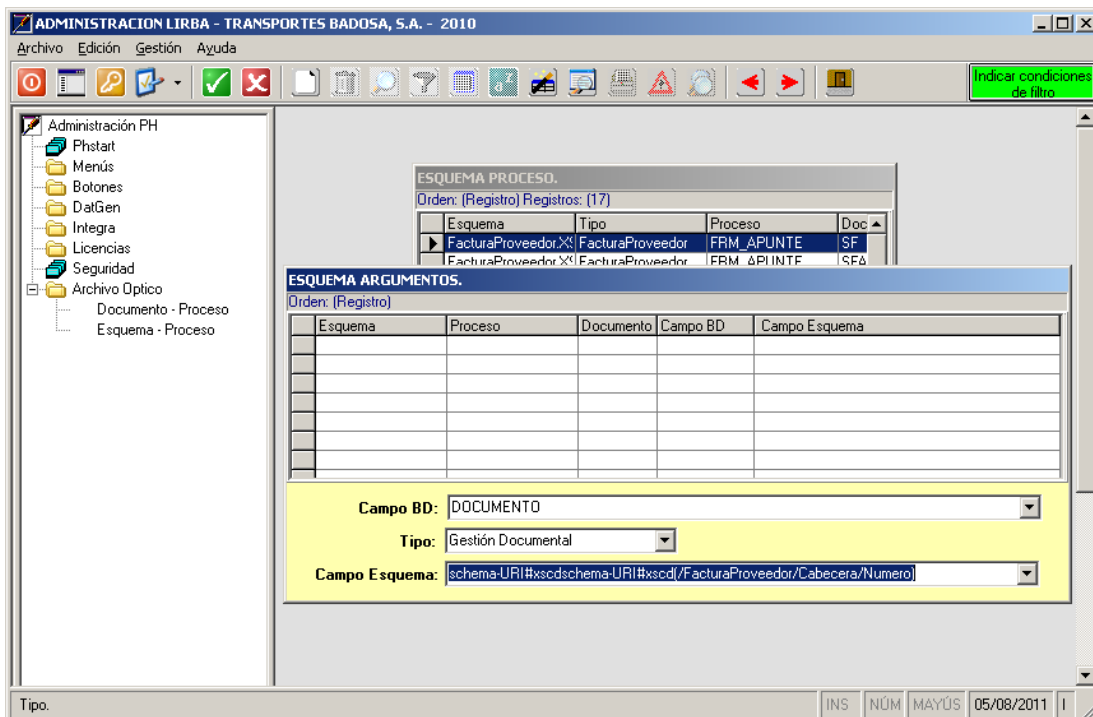
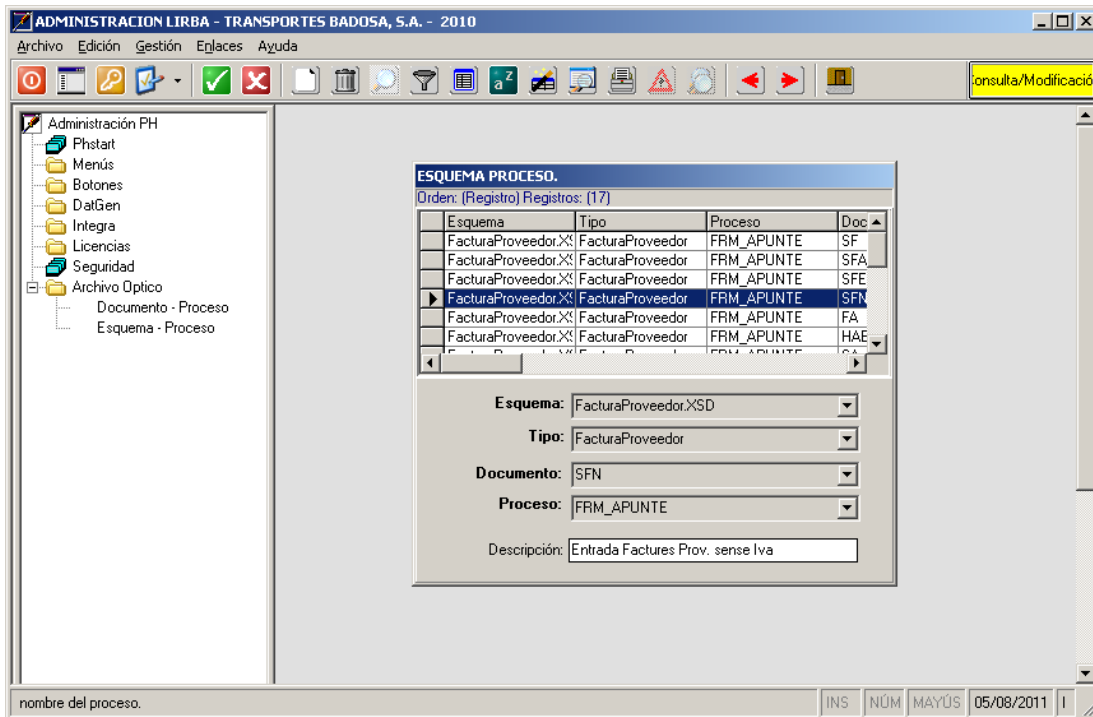
8.2.1. Record Management genèric

Per realitzar el manteniment del Record Management hi ha dues pantalles de manteniment. En una, hi ha el mestre, manteniment de tipus de documents per flux de treball (en aquest cas la definició del tipus de document "Su Factura" pel flux de comptabilitat). L'altre manteniment és el del detall, que defineix quins elements té aquest document (i com és comporten), de tots els elements disponibles d'aquest flux en el repositori del Lirba.



8.2.2. Record Management: Arxiu Òptic

Per realitzar el manteniment de l'Arxiu Òptic hi ha dues pantalles de manteniment. En una, es defineix la relació que hi ha entre els tipus de documents definits al Lirba i els esquemes definits a l'arxiu òptic (la relació entre els tipus de document i les plantilles és implícita a partir dels estats definits a cadascun). En l'altre manteniment es defineix una traducció Uri/element, aquesta traducció és bidireccional, permet al Lirba omplir camps de les interfícies però també alhora informar a l'arxiu Òptic dels valors d'aquests per regenerar el XML associat a un document.



9. Cas pràctic: Entrada d'assentaments comptables.

Lirba Financials està pensat per a la petita i mitjana empresa, agilitant i ajudant a resoldre la casuística comptable d'una manera ràpida e intuïtiva, sota un entorn senzill i amigable, alhora que robust. Lirba Financials destaca per la seva eficient entrada de seients, basada en documents, i un control exhaustiu de la gestió de cartera, sense oblidar el potent mòdul d'immobilitzat, ...

Lirba Financials va més enllà de la simple introducció de seients, permetent a l'usuari definir que forma i sota quin comportament deuen ser introduïts els mateixos. Per a facilitar el seu treball Lirba Financials li proporciona una sèrie d'eines com la definició d'un pla matriu, contrapartides automàtiques, seients ràpids, consolidació d'empreses comptables, possibilitat de mantenir més d'un exercici actiu, control de divises, centres de treball e infinitat d'opcions que satisfaran al més exigent dels nostres clients.

Per altra banda tenim que l'Arxiu Òptic ens permet la gestió dels actius d'informació, i que gràcies a la integració que s'ha realitzat entre l'Arxiu Òptic i el Lirba, podem generar una instància d'un flux de treball a partir d'un document electrònic.

Ara veurem com es realitzaria l'entrada d'una factura (com la de la figura inferior), dins el mòdul de comptabilitat a partir del document electrònic generat per l'Arxiu Òptic.

HEWLETT-PACKARD ESPAÑOLA, S.L.
C/ Vicente Aleixandre, 1
Parque Empresarial Las Rozas
28232 Las Rozas (Madrid)
Apartado de Correos 68


Facturado a:
Lluís Cabaco Roca
Garigas n18
17740 Vilafant
Spain

Vendido a:
Lluís Cabaco Roca
Garigas n18
17740 Vilafant
Spain

Embarcado a:
Lluís Cabaco Roca
Attn:Lluís Cabaco Roca
Garigas n18
17740 Vilafant
Spain

Internal Post: ES

Factura
Original



No. de documento: 3666446
Fecha del documento: 26.01.2011
Pagina: 1 / 2

Nuestra referencia: 70V234666001
Pedido Web: SCE091779

Su referencia: SCE091779
Fecha del pedido: 24.01.2011
No. de cliente: 700099504
No de contrato:
Contacto en el cliente:
Su NIF: 40444805H
Nuestro NIF: ESB28260933
Factura a la atención de:
Tel.: 901 810 158
e-mail: TIENDAHP.SOPORTE@HP.COM

Gracias, el importe de esta factura ha sido cargado en su tarjeta de crédito/débito o cuenta PayPal


Comentario:

Deal ID/Version 85767117

Posicion	Cantidad	Modelo	Descripcion	Precio unit.	Precio Tot.
0100	1	Q6000A	Cartucho de impresión negro HP Color Las <small>Nota de entrega: 804831636 / 25.01.2011 Pack-ID: ELE1B15239</small>	82,12 EUR	82,12
0200	1	Q6002A	Cartucho de impresión amarillo HP Color <small>Nota de entrega: 804831636 / 25.01.2011 Pack-ID: ELE1B15239</small>	89,75 EUR	89,75
Base imponible				EUR	171,87
IVA 18%				EUR	30,94
TOTAL				EUR	202,81

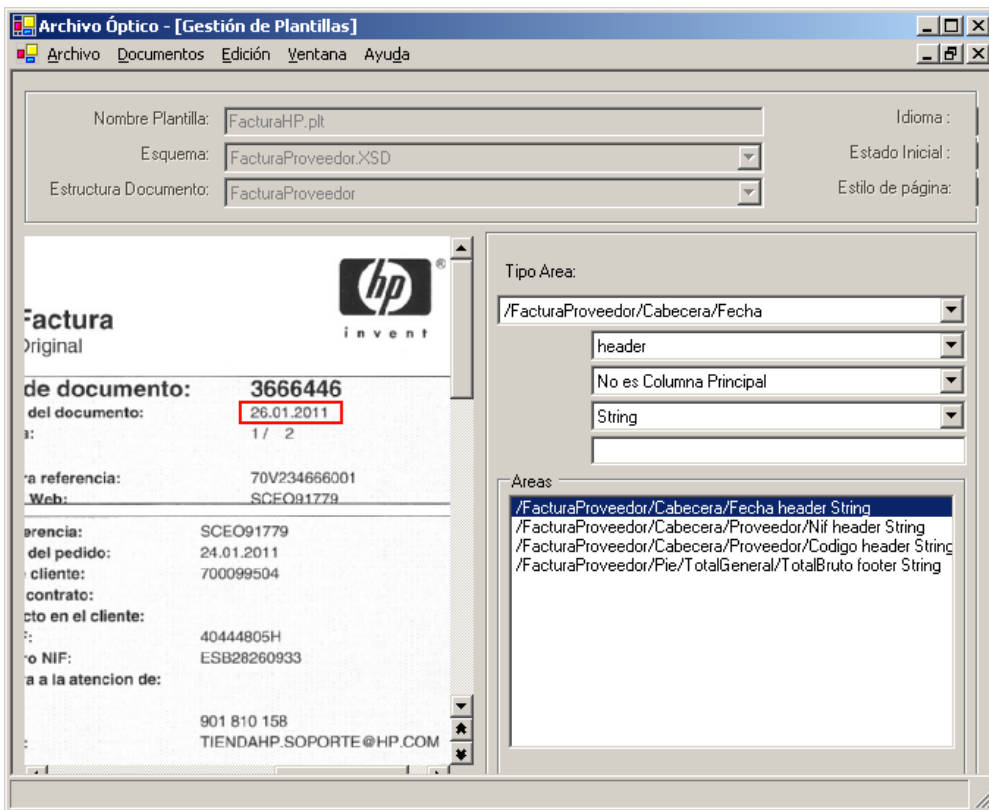
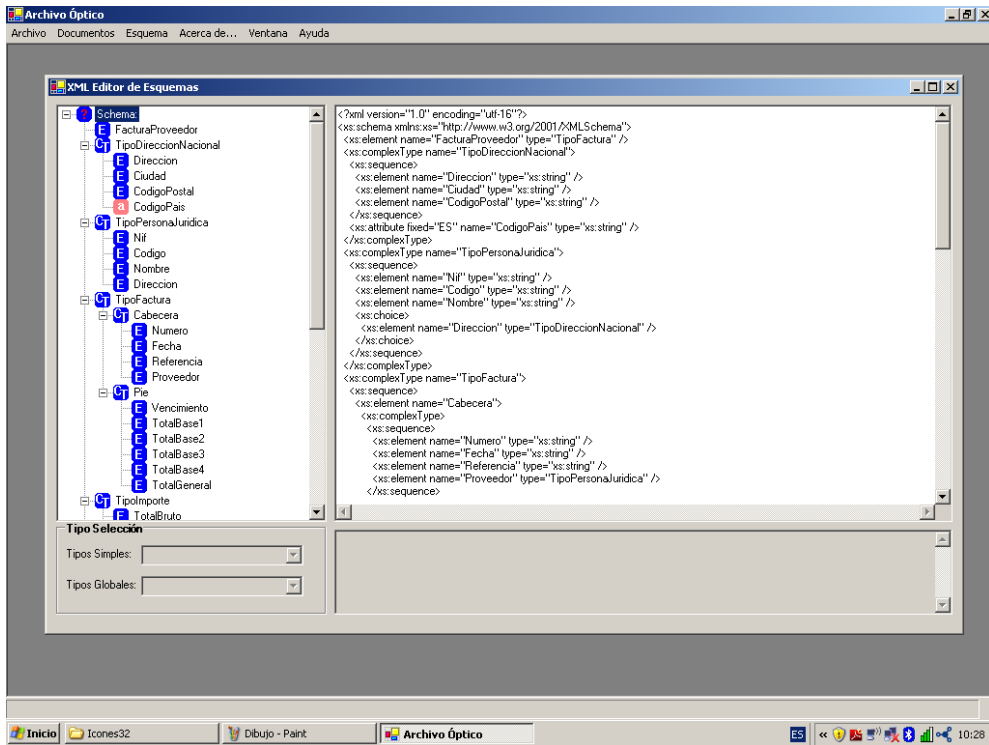
Incoterms: DDP Ship-to-destination

Inscrita en el Reg.Merc.de Madrid, 23-9-71, Tomo 2746 general, 2074 de la Sección 3.a del Libro de Sociedades, Folio 82, Hoja 18865
NIF B-28260933 Sociedad Unipersonal
DELEGACIONES: 28232 Las Rozas (Madrid) - C/ Vicente Aleixandre, 1 - Parque Empresarial Las Rozas - TEL. 902027020
41018 Sevilla - Inca Garcilaso s/n, 1ª planta (Ed. Expo) - TEL. 902027020
46009 Valencia - Pta.de América, 2 1ª Planta (Ed. Euro) - TEL. 902027020
48009 Bilbao - Itáñez de Bilbao, 28, 6º (Ed. MAPFRE) - TEL. 902027020
08190 Sant Cugat del Valles (Barcelona) - Avda.Graells 501 - TEL.902027020



La primera tasca a realitzar és la configuració de l'Arxiu Òptic per tal de poder generar els documents electrònics. S'hauran de realitzar les tasques següents:

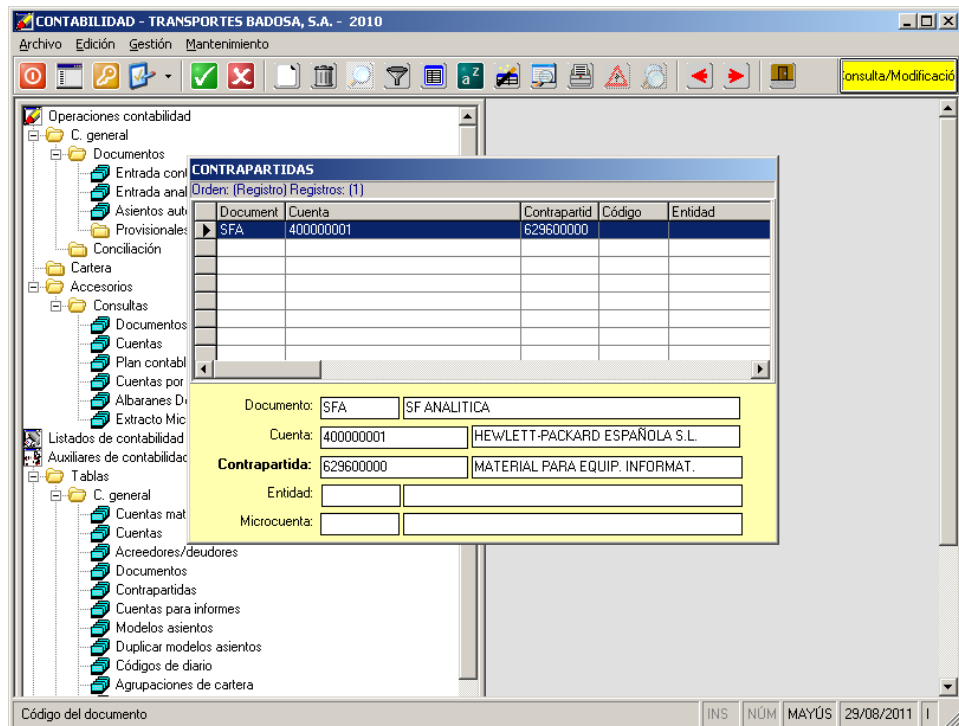
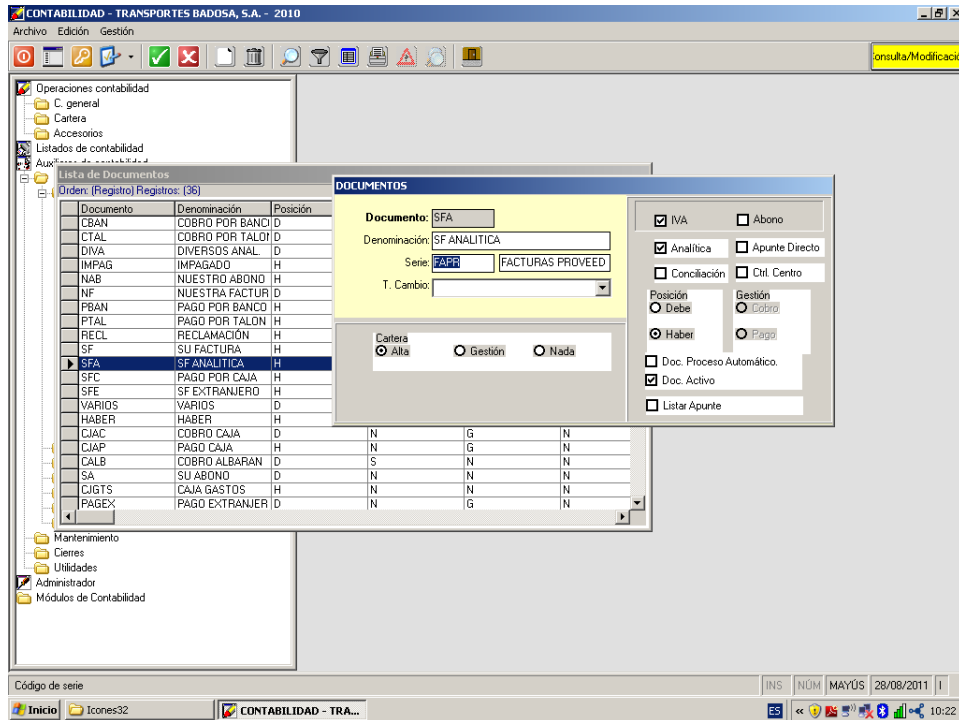
- Crear un esquema que defineixi la estructura i tipologia del document a incorporar.
- Crear una plantilla que defineixi la morfologia del document a incorporar.



L'esquema només serà necessari un cop, i definirà l'estructura de tots els documents possibles de la tipologia "factura". En canvi de plantilles, seràn necessàries la definició de tantes, com morfologies diferents adoptin els documents a escanejar.

El que s'està fent és indicar a l'Arxiu Òptic com ha de generar l'XML associat a un document escanejat, a partir del OCR.

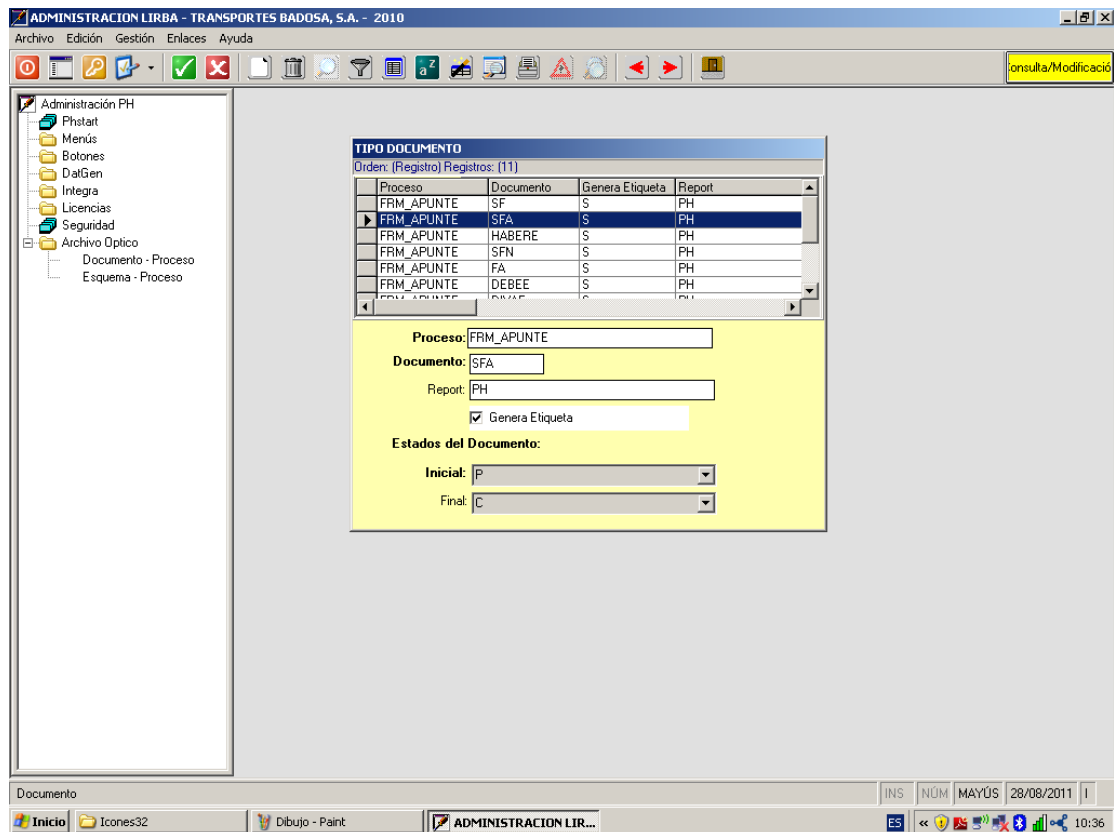
Un cop realitzades aquestes tasques caldrà configurar el Lirba per tal de que la comptabilitat accepti factures de proveïdors.



S'haurà de definir el tipus de document i el comportament d'aquest document dins de la comptabilitat, entre altres com afecta a la cartera, si porta iva, la serie que utilitza, etc... També es pot definir una contrapartida, per tal d'agilitzar la formulació de l'assentament comptable.

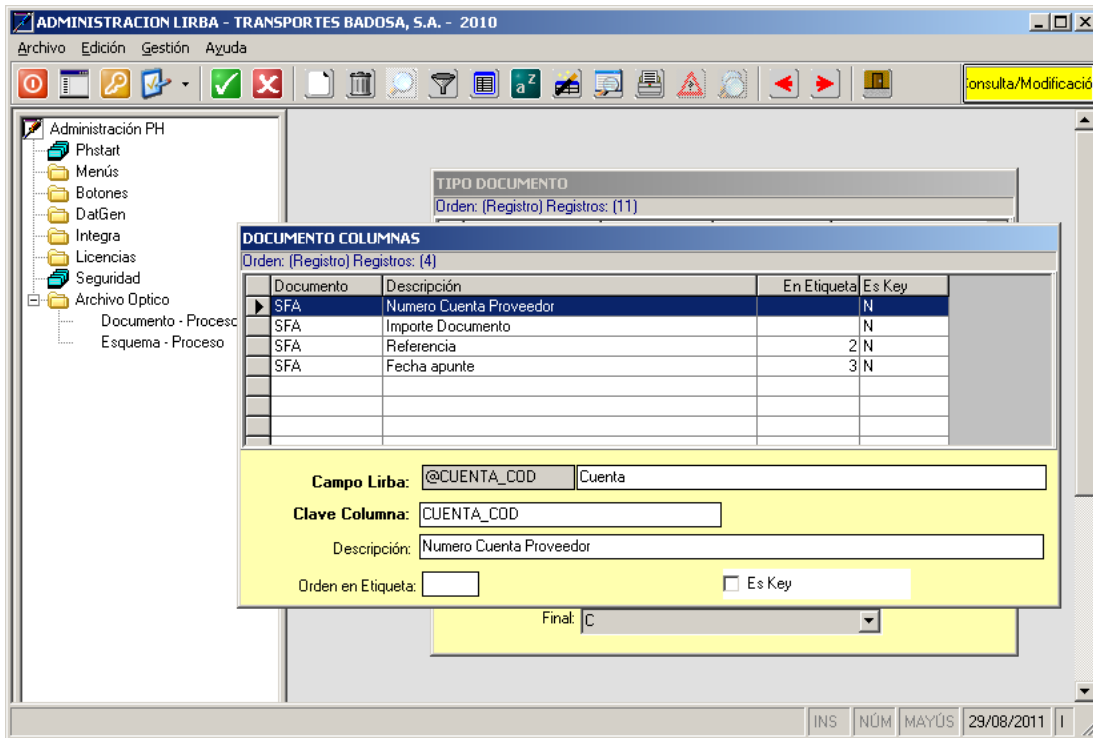
Fins aquí, de moment s'ha configurat el Lirba independentment de si hi ha un Arxiu Òptic o no, ara s'ha de realitzar el lligam entre el Lirba i un Record Management.

Aquesta vinculació s'ha diferenciat en dos parts, una genèrica que permet vincular el Lirba amb diferents aplicacions de tercers, i que només permet lligar un actiu d'informació a una intància d'un flux de treball, i una altre més específica de l'Arxi Òptic que permet vincular les metadades d'un flux de treball amb les metadades d'un actiu d'informació. Això ens permetrà, com veurem, no només vincular documents sinó generar apunts comptables a partir dels documents.

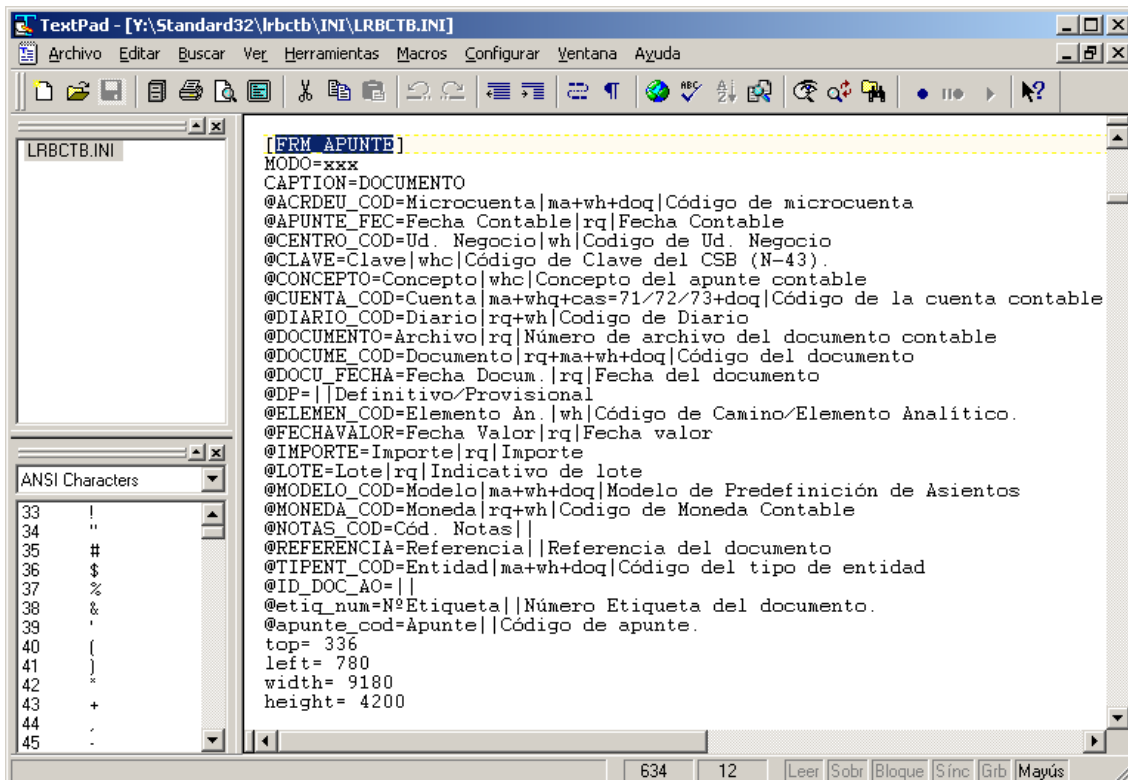


Primer s'han de definir quins fluxos de treball (procés) i tipologia de documents (hem de recordar que el lirba està encarat a document, ara no estem parlant de tipus de document electrònic), hi haurà disponibles. També s'indicarà en quin estat inicial es trobarà el document i a quin estat passarà en cas de realitzar-se amb èxit alguna operació (això permetrà l'encadenament de fluxos de treball). També es podrà indicar al realitzar aquella instància del flux de treball si cal generar un llistat (generalment un codi de barres).

Un cop definit el flux de treball, s'ha de determinar quines variables del flux, estaran disponibles pel Record Management.



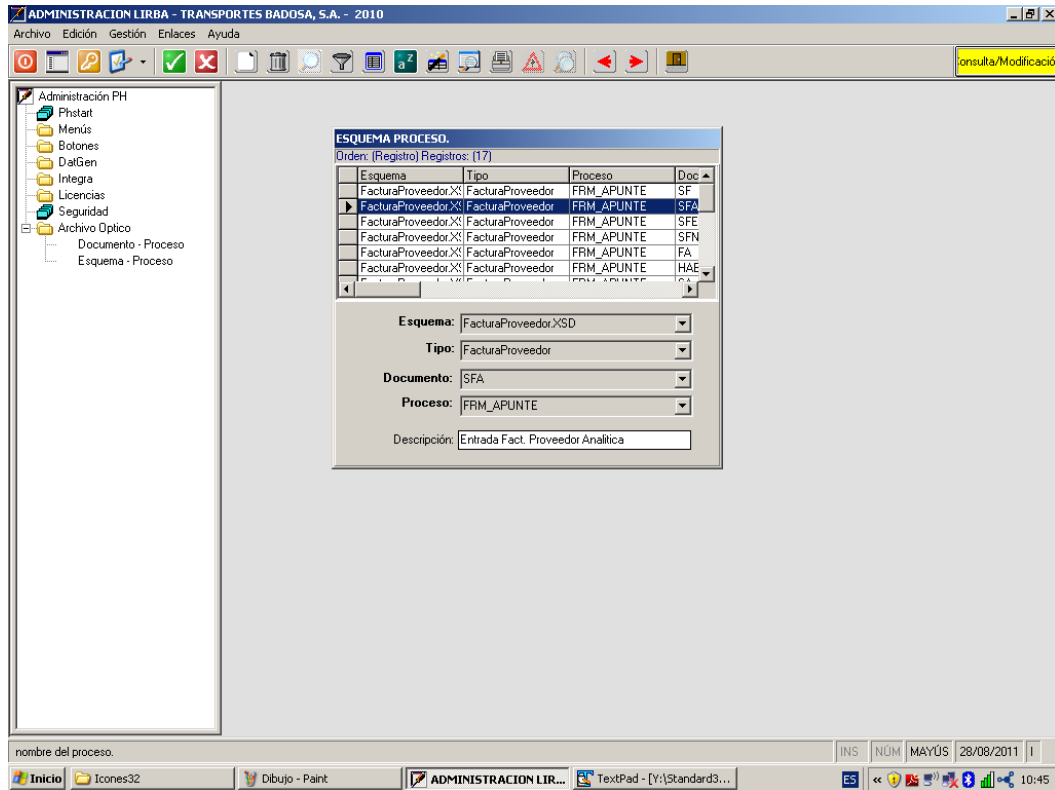
Aquestes variables són els elements del flux que estan definides en el repositori del Lirba, que ens indica el codi de la variable, la seva descripció, el seu comportament i fins i tot la seva persistència.



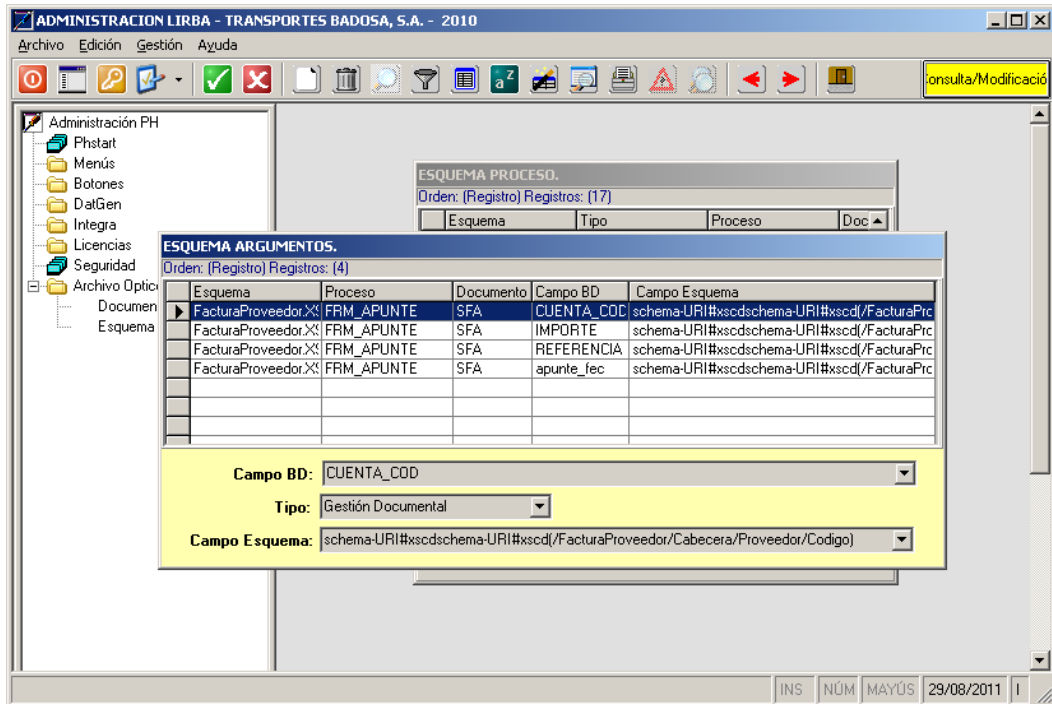
En aquest moment el Lirba ja podria vincular documents electrònics, seleccionat d'un programa extern a una instància d'un flux de treball, i l'usuari podria un cop vinculats els documents visualitzar des del propi Lirba el document origen del registre on es troba.

Fins aquí ja s'ha aconseguit fer una automatització del procés, ara passarem a vincular els metamodels del Lirba i l'Arxiu Òptic que ens permetran la generació automàtica d'instàncies d'un flux de treball (en aquest cas assentaments comptables) i per tant aconseguir una faceta augmentadora i una millora considerable en la gestió.

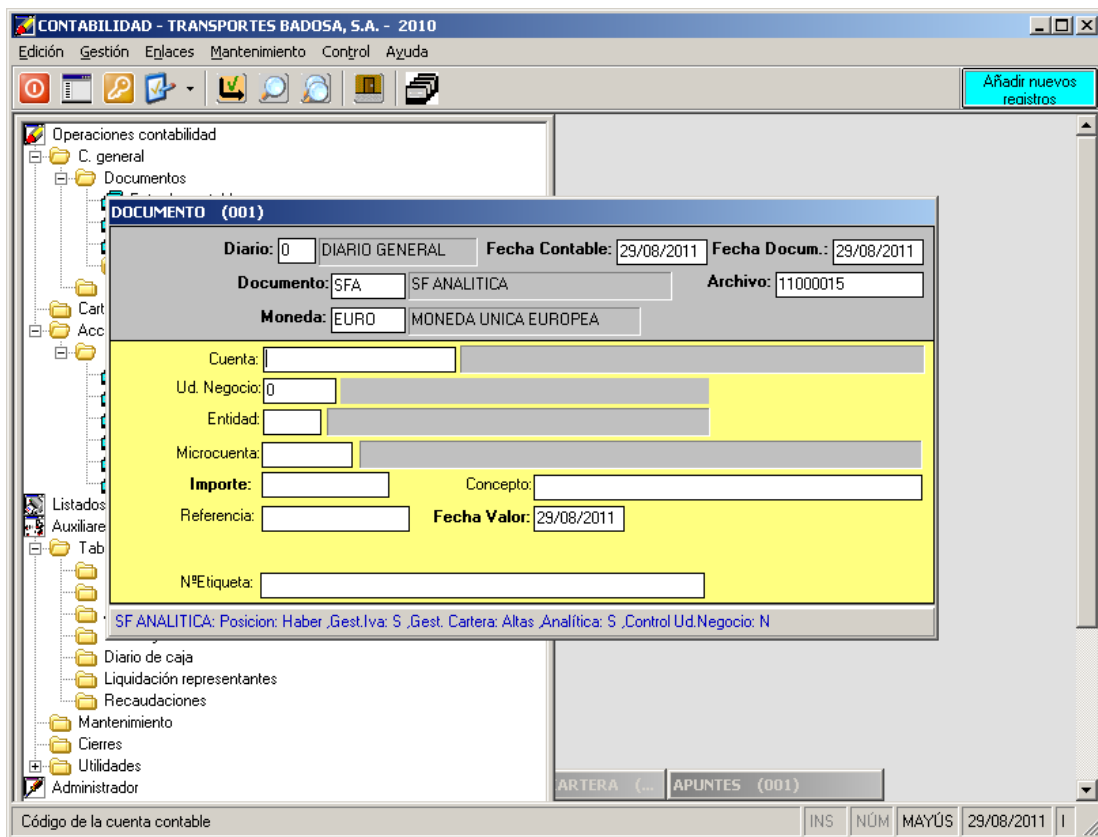
Primer de tot s'ha de vincular el flux definit anteriorment amb un esquema de l'Arxiu Òptic.



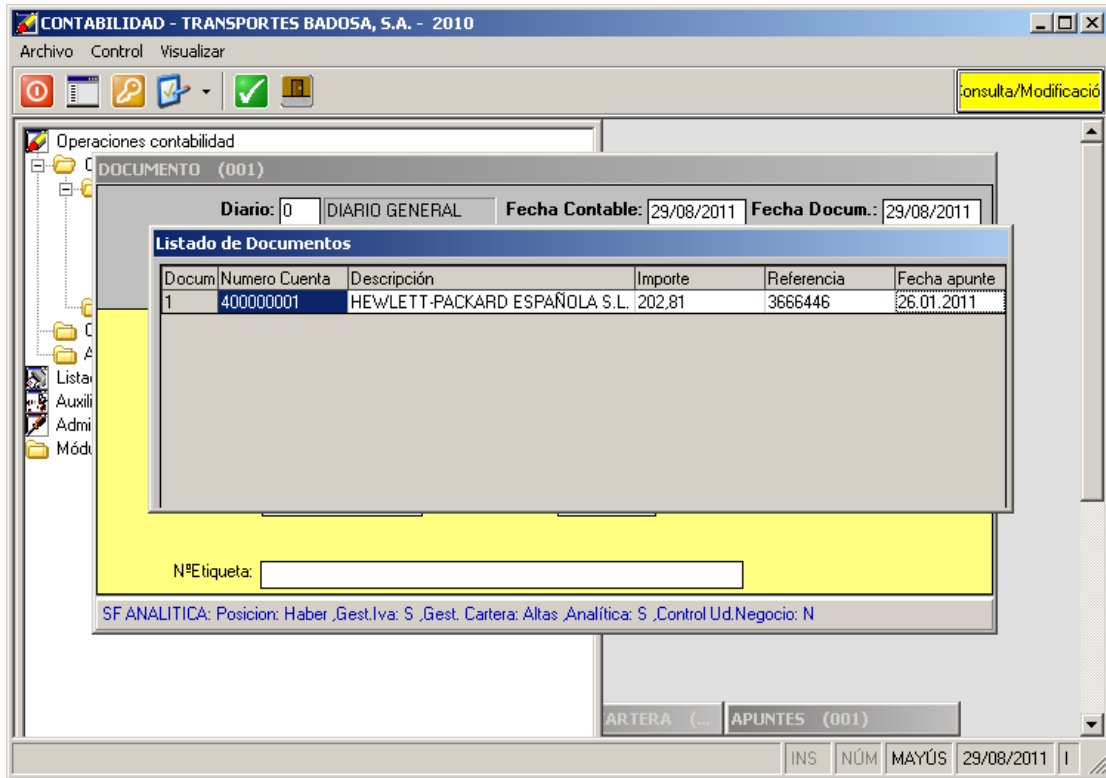
Ara ja tan sols restarà vincular les variables disponibles amb un Uri de l'esquema XML.



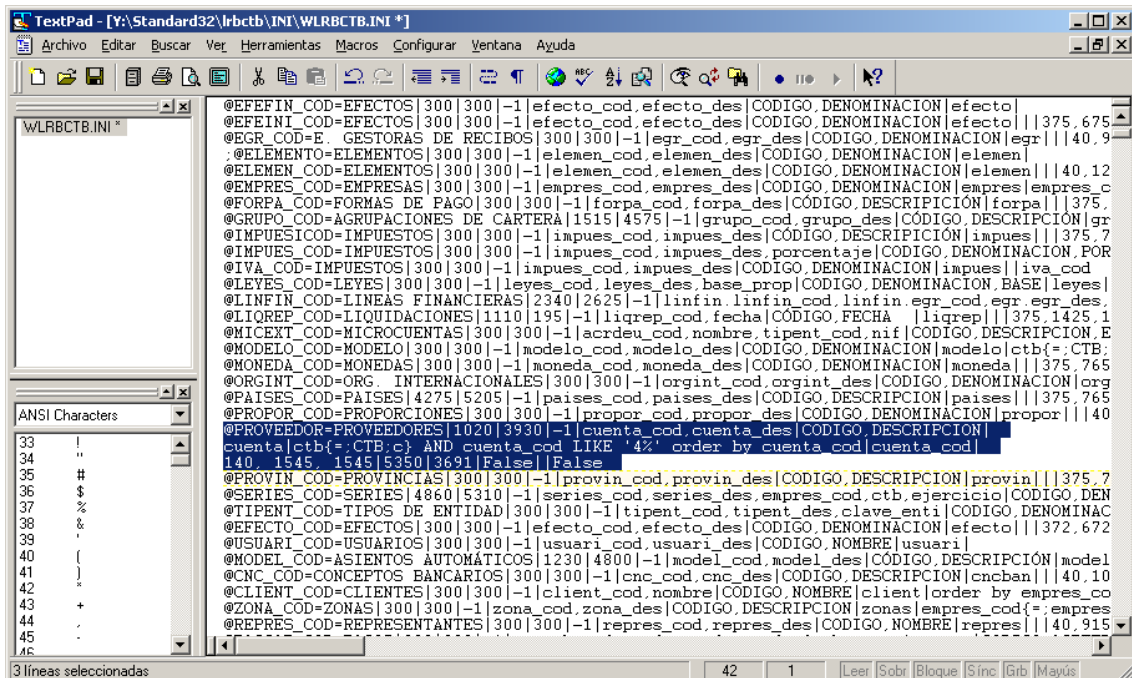
Ara ja es pot realitzar l'entrada assistida de documents a la comptabilitat. El primer que cal fer és entrar a la generació d'apunts comptables i determinar el tipus de document comptable (en aquest cas SFA).



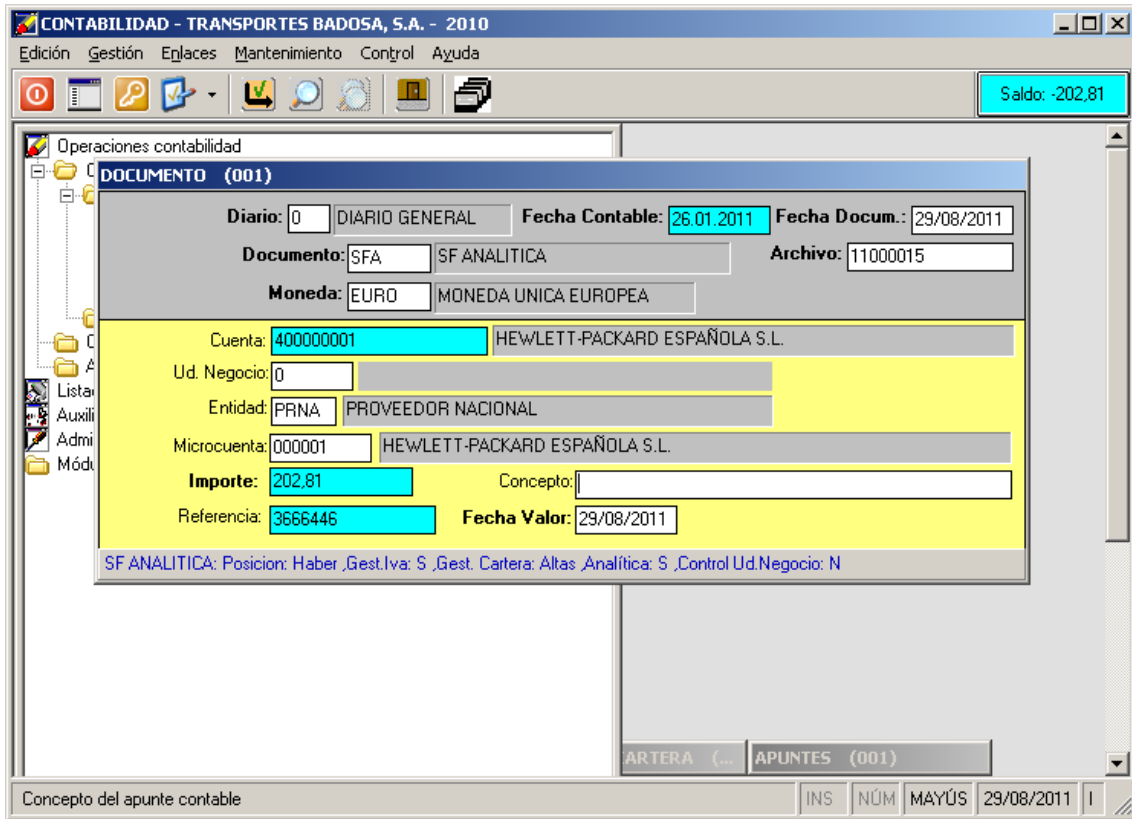
Ara l'aplicació ja es capaç, si li demanem, de mostrar-nos els documents pendents de processar pel tipus SFA.



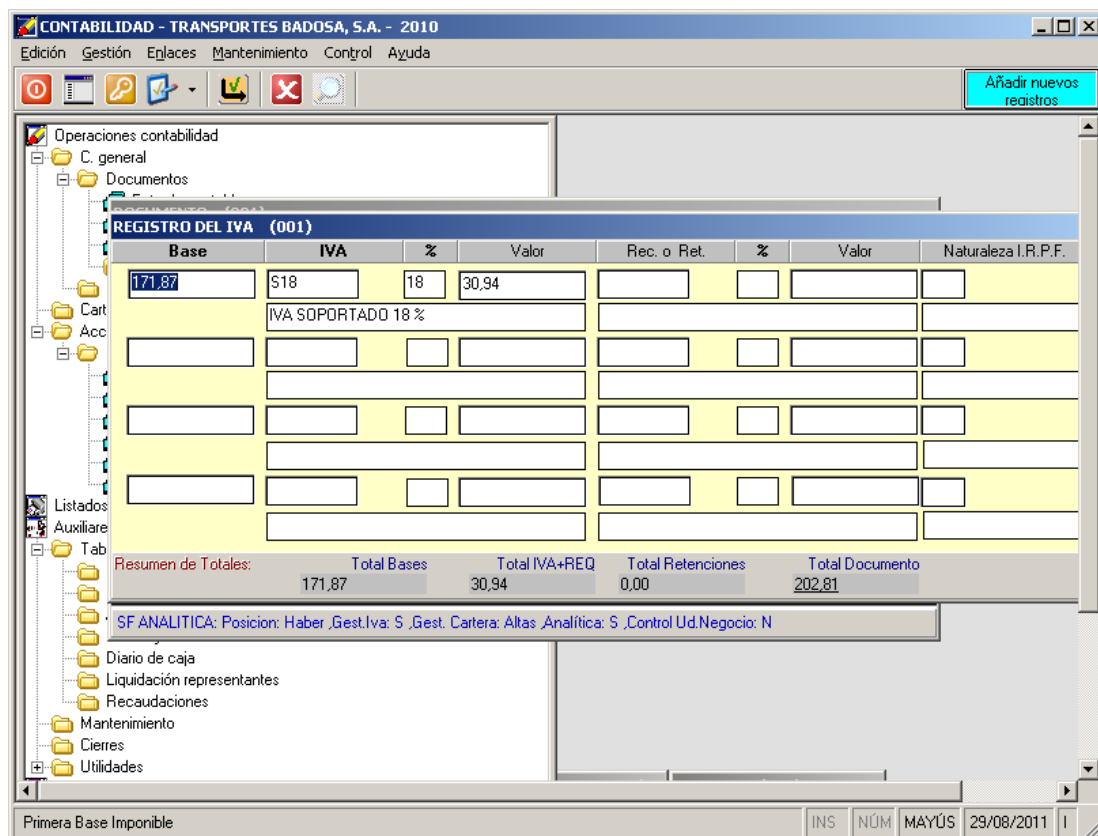
La pantalla que surt, és un grid format per les variables definides en el flux, per tant per cada flux serà diferent, aquestes variables s'omplen amb la informació continguda en els documents XML. A més de les variables definides, el sistema també afegeix les descripcions de les claus foranes, per fer-ho determina per cada variable si hi ha una estructura en el repositori que permeti fer el Lookup.



Un cop seleccionat un document dels disponibles, els valors de les variables s'introdueixen automàticament en l'assentament, a més el sistema determina altres valors (depenents de la porpia idiosincrasia del Lirba). En cas de que un valor no s'ajusti (error del OCR) el podem modificar.



L'apunt comptable es va generant a partir de la pròpia configuració del Lirba i de les dades obtingudes del XML. En aquest cas com s'havia definit IVA i alta de cartera, apareixen aquests elements, els valors dels quals ja estan introduïts només cal validar-los.



CONTABILIDAD - TRANSPORTES BADOSA, S.A. - 2010

Archivo Edición Gestión Enlaces Mantenimiento Control Ayuda

Operaciones contabilidad

C. general

Documentos

DOCUMENTOS

Orden: [Registro] Registros: (1)

Recibo	Efecto	C/P	Concepto	N.	Referencia
RBT	RBT	P		1	3666446

Vencimiento: 29/09/2011 Importe: 202,81 Cobro Pago

Efecto Ini: RBT RECIBO Situación: Pendiente Bloqueado

Efecto Fin: RBT RECIBO

Concepto:

Ud. Negocio: 0 Grupo:

Domiciliación:

Entidad:

Agencia:

Linea financiera: Referencia: 3666446

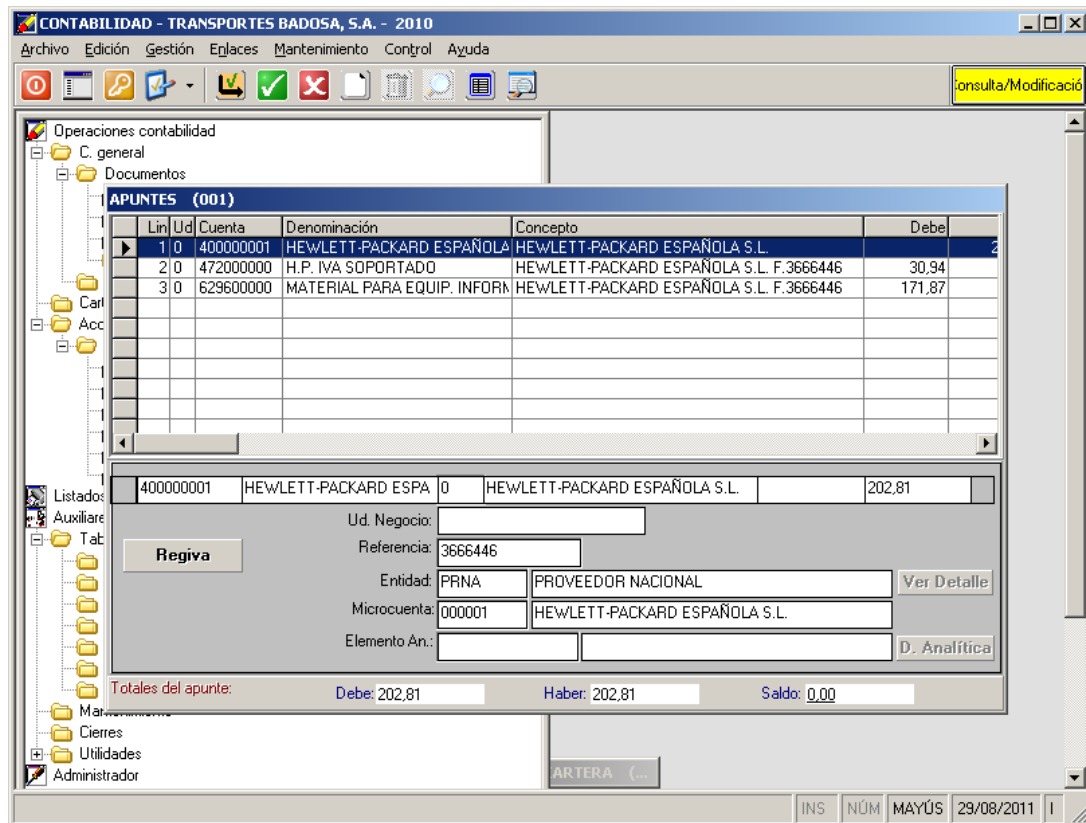
Total Cartera: 202,810

Fecha de vencimiento del recibo

INS NÚM MAYÚS 29/08/2011

Ara ja es pot finalitzar l'entrada de l'assentament, com que s'ha definit una contrapartida l'apunt s'ha realitzat amb molt poc esforç.

Si s'ha definit que es generés una etiqueta es generarà en aquest moment (això té sentit si no s'ha escanejat el document i per tant no s'ha seleccionat cap document), un cop finalitzat es traspassa la informació de les variables modificades si és el cas, o bé si no s'ha seleccionat cap document, cap a l'Arxiu Òptic.



Com s'ha observat hi ha per tant dos mètodes de treball:

- Per un cantò es pot partir de documents prèviament escanejats dels quals s'obté la informació del XML.
- O bé, es poden introduir els assentaments i generar etiquetes i la informació del XML, per posteriorment escanejar els documents.

En ambdós casos obtindrem un vincle entre un assentament comptable i el document físic que l'ha generat.

Això permetrà que es puguin visualitzar aquests documents des del propi Lirba, i que no es requereixi anar a l'arxiu físic quan es vulguin obtenir (per fer còpies o veure informació extra).

CONTABILIDAD - TRANSPORTES BADOYA, S.A. - 2010

Operaciones contabilidad

C. general

EXTRACTO DE CUENTAS CONTABLES

Orden: (Registro) Registros: (2)

Ejercicio	Fecha del	Documento	Referencia	Conc
2011	29/08/2011	11000014	3666446	SU P

Lista:

Auxilia:

T:

N:

C:

U:

Admi:

Mód:

Cuenta: 400000001 HEWLETT-PACKARD

Ud. Negocio:

Fecha inicial: 29/08/2011

Fecha final: 29/08/2011

Moneda: EURO MONEDA UNICA EU

En Divisas

Navegador

Buscar

59,8%

HEWLETT-PACKARD ESPAÑOLA, S.L. Factura Original

C/ Puerto Asislarán, 1
28222 Las Rozas (Madrid)
Aptdo de Correos 58

Factura
Original

No. de documento: 3666446
Fecha del documento: 29/08/2011
Paginas: 1 / 2

Nuestra referencia: 70V23466501
Pedido Web: 3CFC01779

Su referencia: 3CFC01779
Fecha del pedido: 24/01/2011
No. de cliente: 70009054
No. de contrato:

Contacto en el cliente:
Su NIF: 45444654
Nuestro NIF: E382820033
Factura a la atención de:

Tel.: 901 810 158
e-mail: TENDAS@SUPPORTE.HP.COM

Gracias, el importe de esta factura ha sido cargado en su tarjeta de crédito/débito o cuenta PayPal

Comentario:

Entregado a:
Lluís Cabaco Roca
Ofic. Lluís Cabaco Roca
Callejas n.º 8
17745 Villafra
Spain

Entregado a:
Lluís Cabaco Roca
Ofic. Lluís Cabaco Roca
Callejas n.º 8
17745 Villafra
Spain

Internal Post: ES Dual ID/Moviso: 85767117

Posicion	Cantidad	Modelo	Descripcion	Precio unil.	Precio Tot.
----------	----------	--------	-------------	--------------	-------------

Fecha inicial para selección del extracto

INS NÚM MAYÚS 29/08/2011 |

10. Conclusions

Al llarg d'aquest document s'ha explicat quins tipus d'aplicacions disposava PhSystems, i de les mancances detectades, principalment la gestió de la paperassa dins d'una organització. S'ha parlat de la gestió documental i la diferència amb el Record Management; i s'ha establert que l'objectiu d'aquest projecte era millorar la gestió i/o la productivitat dels procediments informatitzats, que tenen com a element inicial en el seu flux de dades, els actius d'informació. En definitiva l'objectiu era transformar una mancança, la falta de la gestió dels actius d'informació; en una fortlesa, la millora de la productivitat.

Per aconseguir aquest objectiu, s'havia subdividit en diferents objectius concrets, un cop realitzat el projecte podem dir:

- El conjunt d'eines realitzades permeten accedir i recuperar documents d'arxiu corrent (Record Management), a més s'ha seguit les directives que permeten, si es vol, l'eliminació dels documents físics.
- De les eines realitzades, destaca l'aplicació d'administració, que funciona com a Arxiu Òptic, permetent de forma automatitzada, àgil i versàtil la creació de documents electrònics a partir dels documents d'arxiu corrent.
- També s'ha realitzat la integració dels actius d'informació amb la gestió de negoci que proporciona un ERP com el Lirba; on s'ha tingut especial cura en proporcionar la funcionalitat sense acoblar-la a les eines realitzades, permetent la utilització de productes de terceres companyies (com és el cas de Matrix o DocuWare)

S'han realitzat, per tant aquests objectius, a més conforme s'han establert els requeriments i s'ha realitzat l'anàlisi, s'han pogut detectar noves oportunitats que s'han resolt satisfactòriament.

Apart dels objectius anteriors, que s'han acomplert, també n'hi havia uns de secundaris, la consecució dels quals (concretament la de dos d'ells), ha estat cabdal per al projecte:

- Amb la incorporació dels Esquemes XML i les plantilles, s'ha pogut resoldre la problemàtica de dotar de semàntica els documents electrònics corresponents a documents administratius.
- Aprofitant el coneixement del Nucli del Lirba i la poderosa eina que ens proporciona el repositori d'aquest, s'han pogut establir lligams entre semàntica dels documents d'arxiu corrent i els fluxos de treball del Lirba.

Per tant es pot afirmar, que s'han aconseguit tots els objectius secundaris i els diferents objectius concrets; però ens hem de preguntar si això ha permès aconseguir l'objectiu principal.

A l'inici d'aquest document, s'ha tractat l'ús de les TIC a les organitzacions; és aquest punt, el que ens determinarà més clarament si s'ha acomplert l'objectiu principal. Les TIC com s'han comentat tenen dues facetes: la faceta automatitzadora i la faceta augmentadora.

Crec sens dubte, que s'ha aconseguit realitzar -amb les eines generades-, una faceta automatitzadora; però això no és suficient per determinar una millora en la gestió i/o la productivitat dels procediments informatitzats. La clau, crec que està, en la faceta augmentadora.

Si determinem que l'objectiu, seguint l'exemple (explicat en el capítol 2) era passar d'un editor de text a una aplicació augmentadora de les seves capacitats intel·lectuals que permetés a un ordinador crear obres a partir de les idees expressades per l'autor; rotundament no.

Però crec que, a part d'automatitzar el procés -de tractament dels actius d'informació-, on n'hi hagut prou d'entendre'l bé per a comprendre quines parts hi ha, com s'encadenen, etc., també s'ha aconseguit

augmentar les nostres capacitats a partir d'entendre com processem la informació i generem coneixements; com s'ha pogut observar clarament en el capítol 9.

Seguint l'exemple i aplicant-lo a la comptabilitat; si bé no s'ha aconseguit generar apunts a partir dels documents administratius entrants (sense la intervenció humana); si que s'ha pogut (a partir d'entendre com processem els documents comptables) configurar el sistema per associar la informació dels documents electrònics -a partir de les seves metadades-, amb la generació del corresponent apunt comptable (amb la mínima intervenció humana).

Un cop vistes les fortaleeses de les eines que s'han generat (gràcies a les oportunitats detectades), hi ha una sèrie de febleses que s'haurien de millorar:

- Existeix una forta limitació en quan a la generació de les metadades, aquesta limitació està causada per:
 - La utilització de la llibreria Modi per la realització de la tasca d'OCR.
 - La utilització de regions basades en píxels per la creació de les plantilles.
- En aquest projecte s'ha deixat totalment de banda la gestió documental per centrar-nos en els documents d'arxiu corrent, i per tant no s'han tingut en compte documents del tipus: patents, informació tècnica, etc... Que si són necessaris en determinats departaments d'una organització.
- S'ha fet un gran esforç per a realitzar un sistema d'indexació flexible, i emprar un llenguatge basat en les metadades per tal de formular les necessitats d'informació; però s'ha deixat tota la feina del llenguatge d'interrogació a l'ús de terceres aplicacions, com seria el cas d'emprar una base de dades nativa en XML per a contenir les metadades dels documents electrònics generats.

Aquestes mancances o febleses, s'han intentat solventar a partir de l'acció de l'usuari o bé amb el coneixement procés, així per exemple en el cas de la comptabilitat:

- No és requereix una complexa formulació de les necessitats d'informació, ja que tots els documents que pertanyen a un tipus concret, si no s'han processat prèviament s'han de processar obligatòriament (s'han de comptabilitzar totes les factures).
- Si l'OCR no funciona correctament sempre l'usuari pot modificar la dada incorrecta i aquest canvi modificarà les metadades del document electrònic (afinant l'OCR).
- No és necessari una complexa detecció de regions o àrees, amb la capçalera i el peu d'una factura en sol haver-hi prou.

En qualsevol cas i tenint en compte la modularitat i extensibilitat del sistema desenvolupat s'ha de considerar:

- La utilització d'eines de tercers o pròpies que permetin una detecció de plantilles i de les regions o àrees d'aquestes de forma automàtica (i no basada en píxels), de tal manera que el sistema aprengui a partir de l'inclusió de nous documents.
- La millor integració amb gestors de continguts o gestions documentals pel tractament de documents extensos.

11. Bibliografia

1. Colaboradors de Wikipedia. *Gestión documental* [en línea]. Wikipedia, La enciclopedia libre, 2011. Disponible en: http://es.wikipedia.org/w/index.php?title=Gesti%C3%B3n_documental&oldid=48783979.
2. Checa Rubio, Ana María, Masip Masip, Pere. *Fonaments de cerca i recuperació de la informació*. Materials UOC.
3. Gallego Lorenzo, Josefa, Juncà Campdadrós, Manela. *Fons de la Informació*. Materials UOC.
4. Pérez-Monotoro Gutiérrez, Mario. *El coneixement en les organitzacions*. Materials UOC.
5. Colaboradors de Wikipedia. *Records management* [en línea]. Wikipedia, La enciclopedia libre, 2011. Disponible en: http://en.wikipedia.org/w/index.php?title=Records_management&oldid=436724211.
6. Appleman, Dan. *Como Dominar Todas Las API de WIN32*. Inforbooks, 2000.
7. Appleman, Dan. *Desarrollo de Componentes Com/ActiveX Con Visual*. Sams, 2000.
8. Hunter, David. *Iniciacion a XML*. Inforbooks, 2001.
9. Bellinaso, Marco, y Kevin Hoffman. *ASP.NET Website Programming: Problem - Design - Solution, Visual Basic .NET Edition*. Wrox, 2003.
10. Cauldwell, Patrick, Rajesh Chawla, Vivek Chopra, Gary Damschen, y Chris Dix. *Servicios Web XML/ XML Web Services*. Anaya Multimedia, 2002.
11. David, Jean-Luc, Tony Loton, Erik Gunvaldson, Christopher Bowen, y Noah Coad. *Visual Studio 2005 Team System*. Tra. Anaya Multimedia, 2007.
12. Davis, Jeff. *SOA: Open Source*. Anaya Multimedia, 2009.
13. Gosnell, Denise, Matthew Reynolds, y Bill Forgey. *Beginning Visual Basic .NET Databases*. 1o ed. Wrox Press, 2001.
14. Hunter, David. *Iniciacion a XML*. Inforbooks, 2001.
15. McClure, Wallace B., Scott Cate, Paul Glavich, y Craig Shoemaker. *Ajax con ASP.net / Ajax with ASP.net*. Grupo Anaya Comercial, 2007.
16. Rahmel, Dan. *Professional Joomla!* Wrox, 2007.
17. Robinson, Simon, Burt Harvey, Christian Nagel, Ollie Cornes, Karli Watson, Morgan Skinner, Jay Glynn, Zach Greenvoss, y Scott Allen. *Professional C#*. 2o ed., 2002.
18. Tisseghem, Patrick. *SharePoint Server 2007*. ANAYA MULTIMEDIA, 2008.
19. Cornella, Alfons. *Entendre el futur ...o res* [en línea]. UOC, 1999. Disponible en: http://www.uoc.edu/humfil/digithum/digithum2/catala/Art_Cornella/cornella_7.htm.
20. Weis, Margaret. *The Soulforge*. Wizards of the Coast, 1999.

Annex I: Instal.lació e Implantació

En aquest annex explicarem els requisits i els passos a seguir per iniciar el treball amb el SI. El dividirem en tres apartats, un per cada component principal: Arxiu Òptic, Client i Lirba.

Com ja s'ha vist anteriorment els dos primers disposen, dels seus corresponents projectes d'instal.lació; creats per tal efecte. El Lirba també disposa d'un projecte d'instal.lació, l'únic que s'ha requerit és afegir les noves llibreries creades, juntament amb els fitxers de configuració i regenerar el projecte. Anem a veure-ho tot amb una mica més de detall.

Arxiu Òptic

En el cas de l'Arxiu Òptic, l'instal.lador realitzat, ja s'encarrega de crear un directori virtual dins del IIS (o en el seu defecte el Personal Web Server). També realitza la instal.lació del .Net Framework 1.3 (si no hi és).

Per tant, l'única limitació que té, ve imposada per l'ús de la llibreria Modi del Ms Office. Així doncs, cal assegurar-se d'haver instal·lat totes les opcions d'imatges del Ms Office.

Tot i així, sense aquest component, el servei també seria funcional, a excepció de les plantilles i l'OCR; però això es pot salvar des de la part client si es vol, i per tant hi hauria l'opció de realitzar una instal.lació amb Mono.

Ara detallarem els diversos fitxers de configuració:

- En el directori Virtual
 - Configuració del Servei web: es defineix els serveis web disponibles, així com la seva tipologia.
 - Configuració del Loader de l'Arxiu Òptic: és un objecte serialitzat en XML que correspon a la configuració.
 - Configuració de l'adaptador de recursos de l'Arxiu Òptic
- En el directori de dades
 - Configuració de l'Arxiu Òptic
 - Altres

Veiem ara els fitxers:

Web.config

```
<configuration>
  <system.runtime.remoting>
    <application>
      <service>
        <wellknown mode="Singleton"
          type="PhSystems.PHModi.Aplicacion.Remoting.PHModiApi,PhSystems.PHModi.Remoting,
          Culture=neutral, PublicKeyToken=null" objectUri="PHModiApi.soap"/>
        <wellknown mode="Singleton"
          type="PhSystems.PHModi.Aplicacion.Remoting.ALmacen,PhSystems.PHModi.ALmacen,
          Culture=neutral, PublicKeyToken=null" objectUri="PHALmacenApi.soap"/>
        <wellknown mode="Singleton"
          type="PhSystems.PHModi.Aplicacion.Remoting.PHTypeDocEx,PhSystems.PHModi.ALmacen,
          Culture=neutral, PublicKeyToken=null" objectUri="PHTypeDocApi.soap"/>
      </service>
    </application>
    <customErrors mode="off"/>
    <debug loadTypes="true" />
  </system.runtime.remoting>
</configuration>
```

En aquest cas s'han definit 3 serveis web, en mode singleton, a partir de les corresponents classes remotes, sobre un fitxer d'intercanvi amb l'extensió .soap (si s'interroga des d'un browser la url formada per "http://[servidor]/[directori virtual]/[cache.soap]?wsdl" , s'obté la definició del servei web)

PhSystems.PHModi.Serialization.Remoting.PHModiApiConfig.Config

```
<?xml version="1.0"?>
<PHModiApiConfig xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  DataResourceName="PhSystems.PHModi.Remoting.IPHModiApi.config"
  DataResourceClassName="PhSystems.Base.Resource.XmlSerializerAdapterResource,PhSystems.Base, Version=1.0.2761.28181,
  Culture=neutral, PublicKeyToken=null" xmlns="http://phsystems.es/PHModi/PHModiApiConfig.xsd">
  <ConfigRules />
  <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
  ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2761.28181, Culture=neutral,
  PublicKeyToken=null" />
</PHModiApiConfig>
```

En aquest cas es defineix que l'Arxiu Òptic (*PhSystems.PHModi.Remoting.IPHModiApi.config*) és un recurs que s'ha de carregar amb una determinada classe d'una llibreria (*PhSystems.Base.Resource.XmlSerializerAdapterResource, PhSystems.Base*), que no s'ha d'aplicar cap regla (*<ConfigRules />*) i que s'ha d'emprar un determinat adaptador de recursos (*PhSystems.Base.Resource.AdapterResourceConfig.config*) que ve definit per una classe d'una llibreria (*PhSystems.Base.Resource.AdapterResourceConfig, PhSystems.Base*).

PhSystems.Base.Resource.AdapterResourceConfig.config

```
<?xml version="1.0" encoding="utf-8"?>
<AdapterResourceConfig xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BaseDirectory>C:\PFC\dades\tester</BaseDirectory>
</AdapterResourceConfig>
```

Com que prèviament s'ha definit un recurs amb serialització XML, l'adaptador d'aquest recurs té un fitxer de configuració amb el seu nom que indica el directori on es troben els recursos. Un adaptador sempre té el seu tipus de recurs associat, així doncs aquí tenim, un adaptador de recursos en directoris, que recupera recursos serialitzats en XML. No necessàriament l'adaptador hauria de ser en sistema de fitxers, i no necessàriament els recursos haurien de ser objectes serialitzats en XML.

Ara el sistema ja sap com carregar la classe de configuració del Servei Web i com inicialitzar aquest, això però dependrà del que digui la configuració d'aquest.

PhSystems.Base.Resource.AdapterResourceConfig.config

```
<?xml version="1.0"?>
<configuration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://phsystems.es/PHModi/template.xsd">
  <StateCollection DataBaseAdapter="PhSystems.PHModi.Serialization.Remoting.DocumentState,PhSystems.PHModi.Remoting,
Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null">
    <Relation>
      <Configuration Name="" ObjectClassName="" />
    </Relation>
    <Key>
      <Configuration Name="" ObjectClassName="" />
    </Key>
    <DataBaseConfig Name="" ObjectClassName="" />
  </StateCollection>
  <XSLTCollection DataBaseAdapter="PhSystems.Base.Resource.SystemIOAdapterResource,PhSystems.Base, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null">
    <Relation ResourceName="XSLTFactoryRelationsSchemas.Config"
FactoryClassName="PhSystems.Base.Relations.DataBaseFactoryRelations,PhSystems.ResourceDB, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null"
AdapterResourceClassName="PhSystems.Base.Resource.DataBaseAdapterResource,PhSystems.ResourceDB, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null">
      <Configuration Name="PhSystems.Data.Simple.BuilderParams.config"
ObjectClassName="PhSystems.Data.Simple.BuilderParams,Phsystems.Data, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
    </Relation>
    <Key>
      <Configuration Name="" ObjectClassName="" />
    </Key>
    <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
  </XSLTCollection>
  <SchemasCollection DataBaseAdapter="PhSystems.Base.Resource.SystemIOAdapterResource,PhSystems.Base, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null">
    <Relation ResourceName="TemplatesFactoryRelationsSchemas.Config"
FactoryClassName="PhSystems.Base.Relations.DataBaseFactoryRelations,PhSystems.ResourceDB, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null"
AdapterResourceClassName="PhSystems.Base.Resource.DataBaseAdapterResource,PhSystems.ResourceDB, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null">
      <Configuration Name="PhSystems.Data.Simple.BuilderParams.config"
ObjectClassName="PhSystems.Data.Simple.BuilderParams,Phsystems.Data, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
    </Relation>
    <Key>
      <Configuration Name="" ObjectClassName="" />
    </Key>
    <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
  </SchemasCollection>
  <TemplatesCollection DataBaseAdapter="PhSystems.Base.Resource.XmlSerializerAdapterResource,PhSystems.Base,
Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null">
    <Relation ResourceName="DocumentsFactoryRelationTemplates.Config"
FactoryClassName="PhSystems.Base.Relations.DataBaseFactoryRelations,PhSystems.ResourceDB, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null"
AdapterResourceClassName="PhSystems.Base.Resource.DataBaseAdapterResource,PhSystems.ResourceDB, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null">
      <Configuration Name="PhSystems.Data.Simple.BuilderParams.config"
ObjectClassName="PhSystems.Data.Simple.BuilderParams,Phsystems.Data, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
    </Relation>
    <Key>
      <Configuration Name="" ObjectClassName="" />
    </Key>
    <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
  </TemplatesCollection>
  <DocumentsCollection DataBaseAdapter="PhSystems.Base.Resource.SystemIOAdapterResource,PhSystems.Base, Version=1.0.2628.23235,
Culture=neutral, PublicKeyToken=null">
    <Relation>
      <Configuration Name="" ObjectClassName="" />
    </Relation>
    <Key>
      <Configuration Name="" ObjectClassName="" />
    </Key>
    <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
  </DocumentsCollection>
  <DefaultImagesCollection DataBaseAdapter="PhSystems.Base.Resource.SystemIOAdapterResource,PhSystems.Base,
Version=1.0.2628.23235, Culture=neutral, PublicKeyToken=null">
    <Relation>
      <Configuration Name="" ObjectClassName="" />
    </Relation>
    <Key>
      <Configuration Name="" ObjectClassName="" />
    </Key>
    <DataBaseConfig Name="PhSystems.Base.Resource.AdapterResourceConfig.config"
ObjectClassName="PhSystems.Base.Resource.AdapterResourceConfig,PhSystems.Base, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null" />
  </DefaultImagesCollection>
  <ReturnAsPDF>false</ReturnAsPDF>
  <SaveAsPDF>false</SaveAsPDF>
  <IOCRAreasResult>PhSystems.PHModi.OCR.OCRAreasResult,PhSystems.PHModi.Schema, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null</IOCRAreasResult>
</configuration>
```

```
<IOCRStringResult>PhSystems.PHModi.OCR.OCRstringResult,PhSystems.PHModi.Schema, Version=1.0.2628.23235, Culture=neutral,
PublicKeyToken=null</IOCRStringResult>
</Configuration>
```

En aquest cas es defineixen tots els components de l'Arxiu Òptic, és sempre el mateix, component, tipus de component, adaptador de recurs associat, configuració del adaptador de recurs, tipus del recurs.

Com ja s'ha vist anteriorment, un adaptador de recursos pot funcionar també sobre BD. Existeix un recurs, que implementa un accés simple a una BD, el fitxer de configuració del qual seria el següent:

PhSystems.Data.Simple.BuilderParams.config

```
<?xml version="1.0"?>
<ArrayOfParam xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Param Name="IBuilderDBExecute" Value="PhSystems.Data.Simple.BuilderDBExecute,Phsystems.Data, Version=1.0.2628.15835,
Culture=neutral, PublicKeyToken=null" />
  <Param Name=".IDbConnection" Value="System.Data.OracleClient.OracleConnection,System.Data.OracleClient, Version=1.0.5000.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  <Param Name=".Connection" Value="user id=phges;data source=orcl;password=phges" />
  <Param Name=".Behavior" Value="Default" />
  <Param Name=".DbType" Value="String" />
  <Param Name=".Direction" Value="1" />
  <Param Name=".SourceVersion" Value="Default" />
  <Param Name=".DynamicProperties" Value="OracleType" />
  <Param Name=".OracleType" Value="VarChar" />
  <Param Name=".TypeDoc.SourceColumn" Value="Docume_cod" />
  <Param Name=".Uri.SourceColumn" Value="Uri" />
  <Param Name=".DocCod.SourceColumn" Value="Id_docA0" />
  <Param Name=".ElementName.SourceColumn" Value="Esq_cod" />
  <Param Name=".Value.SourceColumn" Value="Valor" />
  <Param Name=".Description.SourceColumn" Value="Descripcio" />
  <Param Name="ExistTypeDoc.CommandText" Value="select count(*) from DOCTYPEESQ where Docume_cod=:TypeDoc" />
  <Param Name="GetElementNameByTypeDoc.CommandText" Value="select Esq_cod from DOCTYPEESQ where Docume_cod=:TypeDoc" />
  <Param Name="PutElementNameByTypeDoc.CommandText" Value="Insert into DOCTYPEESQ (ESQ_COD,DOCUME_COD)
values(:ElementName,:TypeDoc)" />
  <Param Name="CountElementNameByTypeDoc.CommandText" Value="Select count(*) From DOCTYPEESQ where DOCUME_COD=
:TypeDoc" />
  <Param Name="UpdateElementNameByTypeDoc.CommandText" Value="Update DOCTYPEESQ set ESQ_COD=:ElementName where
DOCUME_COD=:TypeDoc" />
  <Param Name="GetTypeDoc.CommandText" Value="select Docume_cod from DOCTYPEESQ" />
  <Param Name="PutUriTypeDoc.CommandText" Value="Insert into DOCTYPEURIDES (DOCUME_COD,DESCRIPCION,URI)
values(:TypeDoc,:Description,:Uri)" />
  <Param Name="CountUriTypeDoc.CommandText" Value="select count(*) From DOCTYPEURIDES where DOCUME_COD=:TypeDoc and
URI=:Uri" />
  <Param Name="UpdateUriTypeDoc.CommandText" Value="Update DOCTYPEURIDES set DESCRIPCION=:Description where
DOCUME_COD=:TypeDoc and URI=:Uri" />
  <Param Name="GetDescriptionUriTypeDoc.CommandText" Value="select descripcion from DOCTYPEURIDES where Docume_cod=:TypeDoc and
Uri=:Uri" />
  <Param Name="GetUriTypeDoc.CommandText" Value="select Uri from DOCTYPEURIDES where Docume_cod=:TypeDoc" />
  <Param Name="PutValueUriTypeDoc.CommandText" Value="Insert into DOCVALUES (ID_DOCA0,VALOR,URI) values(:DocCod,:uri,:Value)" />
  <Param Name="CountValueUriTypeDoc.CommandText" Value="select count(*) from DOCVALUES where ID_DOCA0=:DocCod and
URI=:uri" />
  <Param Name="UpdateValueUriTypeDoc.CommandText" Value="update DOCVALUES set VALOR=:Value where ID_DOCA0=:DocCod and
URI=:uri" />
  <Param Name="GetValueUriTypeDoc.CommandText" Value="select Valor from DOCVALUES where Id_docA0=:DocCod and Uri=:Uri" />
  <Param Name="DeleteUriTypeDoc.CommandText" Value="delete from DOCTYPEURIDES where DOCUME_COD=:typeDoc and URI=:uri" />
  <Param Name="DeleteValueUriTypeDoc.CommandText" Value="delete from DOCVALUES where ID_DOCA0=:DocCod and URI=:uri" />
  <Param Name="DeleteTypeDoc.CommandText" Value="delete from DOCTYPEESQ where DOCUME_COD=:typeDoc" />
  <Param Name="GetDocumentQuery.CommandText" Value="" />
</ArrayOfParam>
```

Aquest fitxer de configuració està pensat per contenir tota la informació de l'accés a la BD, així com la implementació de la classe de mapeig a aquesta. L'adaptador de BD seria una factoria abstracta que retornaria un objecte serialitzat **BuilderParams** però aquest objecte serviria alhora per generar via clonació un nou objecte persistent en base de dades, creat dinàmicament que implementaria la interface *IBuilderDBExecute* i que a més tindria en aquest cas funcions com *GetElementNameByTypeDoc* amb un paràmetre *:TypeDoc*. Tot això es faria en temps d'execució i es resoldria mitjançant reflexió sobre els objectes *CommandText* creats.

El més lògic seria que la capa de negoci disposés d'una interface i la classe clonada és retornés sobre aquella interface.

També poden existir altres fitxers de configuració depenen de les funcionalitats afegides o bé de les característiques de la implantació d'aquestes. Tenim per exemple una implantació de magatzem amb serialització XML:

Almacen.config

```
<?xml version="1.0"?>
<ArrayOfParam xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Param Name="Ubicacion" Value="C:\PFC\dades\A0_BADOSA\" />
  <Param Name="BuilderClassName" Value="PhSystems.XML.Data.BuilderXMLDataSet, Phsystems.Data, Version=1.0.2825.26463, Culture=neutral, PublicKeyToken=null" />
</ArrayOfParam>
```

Almacen.xsd

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="Almacen" targetNamespace="http://tempuri.org/Almacen.xsd" elementFormDefault="qualified"
  attributeFormDefault="qualified" xmlns="http://tempuri.org/Almacen.xsd" xmlns:mstns="http://tempuri.org/Almacen.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="Almacen" msdata:IsDataSet="true">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Almacen">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="tipo_cod" type="xs:string" />
              <xs:element name="ejercicio" type="xs:integer" />
              <xs:element name="estado" type="xs:string" />
              <xs:element name="caja_ult" type="xs:double" />
              <xs:element name="lote_ult" type="xs:double" />
              <xs:element name="doc_ult" type="xs:double" />
              <xs:element name="lotxcaj" type="xs:double" />
              <xs:element name="docxlot" type="xs:double" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
    <xs:key name="AlmacenKey1" msdata:PrimaryKey="true">
      <xs:selector xpath="./mstns:Almacen" />
      <xs:field xpath="mstns:tipo_cod" />
      <xs:field xpath="mstns:ejercicio" />
    </xs:key>
  </xs:element>
</xs:schema>
```

Almacen.xml

```
<?xml version="1.0" standalone="yes"?>
<Almacen xmlns="http://tempuri.org/Almacen.xsd">
  <Almacen>
    <tipo_cod>SF0</tipo_cod>
    <ejercicio>2007</ejercicio>
    <caja_ult>0</caja_ult>
    <lote_ult>0</lote_ult>
    <doc_ult>32</doc_ult>
    <lotxcaj>500</lotxcaj>
    <docxlot>200</docxlot>
  </Almacen>
  <Almacen>
    <tipo_cod>SAP</tipo_cod>
    <ejercicio>2007</ejercicio>
    <caja_ult>0</caja_ult>
    <lote_ult>0</lote_ult>
    <doc_ult>1</doc_ult>
    <lotxcaj>500</lotxcaj>
    <docxlot>200</docxlot>
  </Almacen>
  <Almacen>
    <tipo_cod>SFA</tipo_cod>
    <ejercicio>2009</ejercicio>
    <caja_ult>0</caja_ult>
    <lote_ult>0</lote_ult>
    <doc_ult>1</doc_ult>
    <lotxcaj>500</lotxcaj>
    <docxlot>200</docxlot>
  </Almacen>
</Almacen>
```

Client Arxiu Òptic

En el cas del client de l'Arxiu Òptic, l'instal·lador realitzat, fa una instal·lació per defecte vinculant el programa a un servei web, que s'ha d'informar durant la instal·lació.

Tot i així no s'està limitat a aquesta opció, com que es treballa sobre .Net Remoting, es pot modificar el fitxer de configuració remota. (Aquest fitxer també es pot configurar en el servidor)

Remoting.config

```
<configuration>
  <system.runtime.remoting>
    <application>
      <lifetime
        LeaseTime="20M"
        sponsorshipTimeout="8M"
        renewOnCallTime="8M"
        LeaseManagerPollTime="40S"
      />
      <client url="http://127.0.0.1/PhModiRemoting/">
        <wellknown
          type="PhSystems.PHModi.Remoting.IPhModiApi,PhSystems.PHModi.Client"
          objectUri="PhModiApi.soap" url="http://127.0.0.1/PhModiRemoting/PhModiApi.soap"/>
        </client>
      <client url="http://127.0.0.1/PhModiRemoting/">
        <wellknown
          type="PhSystems.PHModi.Remoting.IPhTypeDocApi,PhSystems.PHModi.Client"
          objectUri="PhTypeDocApi.soap" url="http://127.0.0.1/PhModiRemoting/PhTypeDocApi.soap"/>
        </client>
      <client url="http://127.0.0.1/PhModiRemoting/">
        <wellknown
          type="PhSystems.PHModi.Remoting.IAlmacen,PhSystems.PHModi.Client"
          objectUri="PhAlmacenApi.soap" url="http://127.0.0.1/PhModiRemoting/PhAlmacenApi.soap"/>
        </client>
      <channels>
        <channel ref="http" useDefaultCredentials="true"/>
      </channels>
    </application>
  </system.runtime.remoting>
</configuration>
```

En aquest fitxer s'estableix com s'ha d'instanciar un objecte de forma remota, no necessàriament ha d'anar sobre soap, en aquest cas es fa així per la facilitat del IIS de crear Serveis Web i la limitació imposada per poder realitzar la integració amb Lirba.

El funcionament és ben senzill, com ja s'ha explicat, sempre que es vol carregar un component o funcionalitat (ja sigui en el client o bé en el servidor), el que es fa és cridar una classe lloador que ja sigui, via una factoria, o factoria abstracta amb (o sense) fitxers de configuració obté el nom d'un tipus `Value="PhSystems.XML.BuilderXMLDataSet,Phsystems.Data, Version=1.0.2825.26463, Culture=neutral, PublicKeyToken=null"`. Aquesta cadena es passa a .Net Remoting que obté l'objecte i la factoria s'encarrega d'obtenir la seva configuració e inicialitzar-lo (si cal, i sap fer-ho).

Si el tipus no està configurat en el fitxer Remoting.config associat, llavors senzillament es retorna un tipus local.

A part del fitxer anterior també hi ha un fitxer de configuració, que fa de factoria dels serveis que proporciona el programa, i configura aspectes relacionats amb el Driver d'escaneig.

Configuration.config

```
<?xml version="1.0"?>
<Configuration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Timeout=""
RemotingConfigurationFile=".\\Remoting.Config"
IPTypeDocApiClassName="PhSystems.PHModi.Remoting.IPhTypeDocApi,PhSystems.PhModi.Client, Version=1.0.2642.13535,
Culture=neutral, PublicKeyToken=null" IPHModiApiClassName="PhSystems.PHModi.Remoting.IPHModiApi,PhSystems.PhModi.Client,
Version=1.0.2642.13535, Culture=neutral, PublicKeyToken=null"
IALmacenApiClassName="PhSystems.PHModi.Remoting.IALmacen,PhSystems.PhModi.Client, Version=1.0.2642.13535, Culture=neutral,
PublicKeyToken=null" BustiaEntradaImatges="c:\\scanner\\imatges" BustiaEntradaTemporal="c:\\scanner\\tmp">
  <Twainconfig>
    <TwainAttribute Value="-1" />
    <TwainAttribute Name="IPIXELTYPE" Value="BW" />
    <TwainAttribute Name="IXRESOLUTION" Value="150" />
    <TwainAttribute Name="IYRESOLUTION" Value="150" />
  </Twainconfig>
</Configuration>
```

Lirba.

Pel que fa a la integració amb el lirba, existeixen diferents fitxers de configuració, en realitat aquests fitxers estan en una BD, però per realitzar el manteniment és més senzill efectuar-ho com si es tractessin d'incís (quan s'introdueixen a la BD, cada entrada té un usuari que l'ha generat; per defecte és el system però si és un valor configurable al sobreescrivre aquest valor i es guarda amb l'usuari loguinat –també es genera una copia com a default-). Quan s'afegeix o és modifica una funcionalitat, s'han de modificar els INI's i tornar-los a distribuir (l'aplicació detecta els canvis i els carrega amb System; una entrada pot tenir més d'un usuari però té preferència el system)

- Configuració general
 - Configuració dels mòduls instal·lats
 - Configuració Serveis Web
- Configuració de cada mòdul
 - Configuració de processos
 - Configuració de botoneres
 - Configuració de consultes
 - Configuració de llistes
 - Configuració de manteniments
 - Configuració de taules

Anem a veure alguns dels detalls dels fitxers.

Ngstart.ini

```
[DICCIONARIO]
source=c:\phges32\phges32\std32.udl
;sourceini=c:\phges32\phges32\std32.udl
sourcetmp=tmp32
Logoninfotmp=DLG=2;adUseTransaction;NLS_DATE_FORMAT=dd/mm/yyyy
urlao=PhModiApiLluis
ImageHeight=24
ImageWidth=24
idLocal =
idservidor =
Logoninfo=
Logoninfoini=
titulo=
run=
pathini=
reportpath=
pathLocal=
IdInst=STD
nocaps = -1
sourceini=
```

```
[LIRBA]
titulo=ADMINISTRACI" N PH
run=Y:\STANDARD32\bin32\Lirba.exe
pathini=Y:\STANDARD32\Lirba\ini
reportpath=Y:\STANDARD32\Lirba\Report
;LogonInfo=uid=ph;pwd=ph;
;source=Y:\STANDARD32\phges32\std32.udl
;LogonInfotmp=
;sourcetmp=Y:\STANDARD32\phges32\tmp32.udl
sourcetmp=
newalm=NEWALM
gesaux=GESAux
gescmp=GESCOMP
geslis=GESLIS
```

```

gesmst=GESMST
produc=PRODUC
gesvnt=GESVNT
newcon=NEWCON
auxili=AUXILI
jetlis=JETLIS
newadm=NEWADM
newutl=NEWUTL
newadm=NEWADM
inmovi=INMOVI
windowstate=2
mdileft=150
mditop=300
mdiwidth=9360
mdiheight=6810
guardarconfig=False
fmrcolor=12632064
bmrcolor=12632256
fmbcolor=8421376
bmbcolor=8421504
vistaprevia=-1
fontbold=0
fontitalic=0
fontsize=9
fontname=Arial
fmrbcolor=8454143
fmu1color=16777088
fmu2color=12566463
blicolor=32768
seguroaltas=0
capslock=-1
fontunderline=0
fontstrikethru=0
forecolor=-2147483640
    
```

Existeix una secció pel diccionari i una secció per cada mòdul instal·lat, bàsicament hi ha informació de connexions, mòduls accessibles (de mòdul a mòdul, hi ha funcionalitats que es criden a un mòdul diferent del que s'està executant; i no ha de perquè ser el mateix pot haver-hi excés diferents), etc...

WebServices.xml

```

<?xml version="1.0"?>
<!DOCTYPE WebServices [
  <!ELEMENT WebServices (WebService*,WebServiceGlobalTypes)>
  <!ELEMENT WebService (WebServiceConnection,WebServiceContainer,WebServiceTypes)>
  <!ELEMENT WebServiceGlobalTypes (WebServiceType*)>
  <!ELEMENT WebServiceTypes (WebServiceType*)>
  <!ELEMENT WebServiceConnection EMPTY>
  <!ELEMENT WebServiceContainer EMPTY>
  <!ELEMENT WebServiceType EMPTY>
  <!ATTLIST WebService id CDATA #REQUIRED
    ref CDATA #IMPLIED>
  <!ATTLIST WebServiceConnection wsdlfile CDATA #IMPLIED
    wsmfile CDATA #IMPLIED
    Port CDATA #IMPLIED
    Service CDATA #IMPLIED
    ref CDATA #IMPLIED>
  <!ATTLIST WebServiceContainer targetPROGID CDATA #IMPLIED
    iid CDATA #IMPLIED
    ref CDATA #IMPLIED>
  <!ATTLIST WebServiceTypes ref CDATA #IMPLIED
    OnError CDATA #IMPLIED>
  <!ATTLIST WebServiceType id CDATA #IMPLIED
    targetNamespace CDATA #IMPLIED
    targetPROGID CDATA #IMPLIED
    iid CDATA #IMPLIED
    ref CDATA #IMPLIED>
]
<WebServices>
  <WebService id="Almacen">
    <WebServiceConnection
      wsdlfile="http://127.0.0.1/PhModiRemoting/Almacen.soap?wsdl" />
    <WebServiceContainer
      targetPROGID='PhTypeMapper.PhModiApiProxi'
      iid='{81732648-553C-4A9F-BD71-EB9FDB07D792}' />
    <WebServiceTypes OnError="ShowError" />
  </WebService>
  <WebService id="PhModiApiLluis">
    <WebServiceConnection
      wsdlfile="http://127.0.0.1/PhModiRemoting/PhModiApi.soap?WSDL" />
    <WebServiceContainer ref='PhModiApi' />
    <WebServiceTypes ref='PhModiApi' />
  </WebService>
  <WebService id="PhModiApi">
    <WebServiceConnection
      wsdlfile="http://minerva:8000/phmodi/PhModiApi.soap?WSDL" />
    <WebServiceContainer
      targetPROGID='PhTypeMapper.PhModiApiProxi'
    
```

```

        id='{81732648-553C-4A9F-BD71-EB9FDB07D792}' />
<WebServiceTypes OnError="ShowError">
  <WebServiceType
    id='ParsingOcrResult'
    targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
    targetPROGID='PhTypeMapper.AbstractMapper' />
  <WebServiceType
    id='ComponentDesignator'
    targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
    targetPROGID='PhTypeMapper.ComponentDesignator' />
  <WebServiceType
    id='Area'
    targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
    targetPROGID='PhTypeMapper.Area' />
  <WebServiceType
    id='DownloadFile'
    targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
    targetPROGID='PhTypeMapper.DownloadFile' />
  <WebServiceType
    id='TemplateInfo'
    targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
    targetPROGID='PhTypeMapper.AbstractMapper' />
  <WebServiceType
    id='DocumentInfo'
    targetNamespace='http://schemas.microsoft.com/cLr/nsassem/PhSystems.PHModi.Serialization.Remoting/PhSystems.PHModi.Cli
ent%2C%20Version%3D1.0.3868.34252%2C%20Culture%3Dneutral%2C%20PublicKeyToken%3Dnull'
    targetPROGID='PhTypeMapper.AbstractMapper' />
</WebServiceTypes>
</WebService>
</WebServiceGlobalTypes/>
</WebServices>

```

Bàsicament és una serialització d'una factoria abstracta, que conté contenidors de Serveis Web, cada contenidor té la seva connexió, la interface pública que s'ha de clonar a partir del Servei Web definit; i els tipus de mapeig per poder fer la serialització Soap sobre tipus complexos. Cada element té el seu nom i es pot reutilitzar fent-li una referència.

Lirba.ini

```

[FRM_ACCESO]
@DEPART_COD=Departamento|whc|Código de Departamento
CAPTION=ACCESO
LEFT=2925
MOD0=mu2
TOP=1530

[FRM_ADMCUE]
@ACTIVO=Activa||Indica si La cuenta est. activa
@ANALITICA=Analítica||Indica si La cuenta tiene analítica
@CTB=|ne|
@CUENTA_COD=Cuenta|efg+rq|Código de Cuenta
@CUENTA_DES=Denominación|rq|Denominación de La Cuenta
@CUENTOT=Cuenta Total.|ne|Cuenta sobre La que se Totaliza
@FRA_CLAVE=Clave de Entidad|efa+efg|
@LASTUPD=||
@NIVEL=Nivel|ne|Nivel de La Cuenta.
@PASES=Pases|efg|Indicar si admite pases contables
@TIPENT_COD=Tipo Entidad|wh+efg+cas=0|Tipo de Entidad
@USUARIO=||
ABRIENCONSULTA=N
CAPTION=CUENTAS
HEIGHT=2940
LEFT=3795
LISTAUTOMATICA=N
MOD0=mr
TABLA=cuenta
TOP=375
WIDTH=6150

[FRM_AGENCI]
@AGENCI_COD=Agencia|efg+rqe|Código de La Agencia.
@AGENCI_DES=Denominación|rq|Denominación de La Agencia.
@CODPOSTAL=Código Postal||Código Postal de La Agencia.
@DIAS_CAREN=Dias Carencia||Dias de Carencia para La Agencia.
@DIAS_RIESG=Dias Riesgo||Dias de Riesgo para La Agencia.
@DIRECCION=Dirección||Dirección de La Agencia.
@DIREXT_COD=dirext_cod||
@EGR_COD=egr_cod||
@FAX=Fax||Nf de Fax de La Agencia.
@LASTUPD=||
@NOMEXT_COD=nomext_cod||
@NOTAS_COD=notas_cod||

```

```
@PAISES_COD=País|ma+wh|Código del País de La Agencia.
@POBLACION=PoblaciÓn||PoblaciÓn de La Agencia.
@PROVIN_COD=Provincia|ma+wh+cen=09d108|Código de La Provincia de La Agencia.
@TELEFONO=TelÉfono|Nj de TelÉfono de La Agencia.
@TELEX=TeLex||Nj de TeLex de La Agencia.
@USUARIO=||
CAPTION=AGENCIAS
LINEAS=1
MOD0=mr
TABLA=agenci
```

Cada procés que té una interacció amb l'usuari, té definit una secció (també hi ha l'opció de posar una secció diccionari per valors reutilitzables). No necessàriament, ha de ser un formulari, sinó que pot ser un conjunt de formularis (un exemple d'això seria el procés de comptabilització).

Cada secció conté els elements del procés (@nom_element), i cada element conté el seu origen de BD, el seu comportament (efg+rqe) i la seva descripció.

Quan l'usuari a partir d'una opció de menú, o bé de l'arbre de menús accedeix a un procés/flux de treball el sistema recupera la seva secció e inicialitza tots els elements.

BLirba.ini

```
[DICCIONARIO_DESACTIVAR_MENUS]
MAXVBLET=8
MAXMETHOD=0
VBLET01=@mnu_examinar@enabled@false
VBLET02=@mnu_mantener@enabled@false
VBLET03=@mnu_global@enabled@false
VBLET04=@mnu_listados@enabled@false
VBLET05=@mnu_filtro@enabled@false
VBLET06=@mnu_borrar@enabled@false
VBLET07=@mnu_lista@enabled@false
VBLET08=@mnu_ordenar@enabled@false

[DICCIONARIO_DESACTIVAR]
MAXVBLET=3
MAXMETHOD=0
VBLET01=@mnu_examinar@enabled@false
VBLET02=@mnu_mantener@enabled@false
VBLET03=@mnu_global@enabled@false

[DICCIONARIO_ACTIVAR_GESTION]
MAXVBLET=5
MAXMETHOD=0
VBLET01=@mnu_listados@enabled@true
VBLET02=@mnu_filtro@enabled@true
VBLET03=@mnu_lista@enabled@true
VBLET04=@mnu_borrar@enabled@true
VBLET05=@mnu_ordenar@enabled@true

[DICCIONARIO_ACTIVAR_QWH]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_examinar@enabled@true

[DICCIONARIO_ACTIVAR_QWHI]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_examinar@enabled@true

[#MR]
STATUS2=YELLOW|Consulta/ModificaciÓn|
MAXITEM=14
ITEM01=Validar|mnu_validar_click|0|0|true|true|Validar La informaciÓn de pantalla
ITEM02=Deshacer|mnu_deshacer_click|0|0|true|true|Deshacer Los cambios de pantalla
ITEM03=spc1|3|0|true|true|
ITEM04=A0adir|mnu_alta_click|1|0|true|true|Control de modalidad de altas
ITEM05=Eliminar|mnu_borrar_click|0|0|true|true|Borrar registro
ITEM06=spc2|3|0|true|true|
ITEM07=Consultar|mnu_examinar_click|0|0|true|true|Localizador de cÓdigos
ITEM08=Filtro|mnu_filtro_click|0|0|true|true|Introducir condiciones de filtro
ITEM09=Lista|mnu_lista_click|0|0|true|true|Activar La Lista
ITEM10=Listados|mnu_listados_click|0|0|true|true|Listados...
ITEM11=CambiarGlobal|mnu_global_click|0|0|true|true|ModificaciÓn Global
ITEM12=Mantener|mnu_mantener_click|0|0|true|true|Mantenimiento de Los campos
ITEM13=spc3|3|0|true|true|
ITEM14=Salir|mnu_salir_click|0|0|true|true|Salir
PROPIETATS=1|0|0
STATUS1=CYAN|A0adir nuevos registros|
STATUS3=GREEN|Indicar condiciones de filtro|
PREMENUMAME=frm_menuspr
```

```

PREMENUFRMNAME=frm_menuspr

[ #MB ]
ITEM10=Personalizar|mnu_setlista_click|0|0|true|true|Personalizar Las columnas de La Lista
MAXITEM=19
ITEM01=Validar|mnu_validar_click|0|0|true|true|Validar La informaci3n de pantalla
ITEM02=Deshacer|mnu_deshacer_click|0|0|true|true|Deshacer Los cambios de pantalla
ITEM03=spc1|4|0|true|true|
ITEM04=Aadir|mnu_alta_click|1|0|true|true|Control de modalidad de altas
ITEM05=Eliminar|mnu_borrar_click|0|0|true|true|Borrar registro
ITEM06=Consultar|mnu_examinar_click|0|0|true|true|Localizador de c3digos
ITEM07=Filtro|mnu_filtro_click|0|0|true|true|Introducir condiciones de filtro
ITEM08=Lista|mnu_lista_click|0|0|true|true|Activar La Lista
ITEM09=Ordenar|mnu_ordenar_click|0|0|true|true|Ordenar La Lista
ITEM11=Imprimir|mnu_printlista_click|0|0|true|true|Imprimir La Lista
ITEM12=Listados|mnu_listados_click|0|0|true|true|Listados...
ITEM13=CambiarGlobal|mnu_global_click|0|0|true|true|Modificaci3n Global
ITEM14=Mantener|mnu_mantener_click|0|0|true|true|Mantenimiento de Los campos
ITEM15=spc2|3|0|true|true|
ITEM16=ANTERIOR|frm_conexiones@GridPrevious|0|0|true|true|Siguiete registro
ITEM17=SIGUIENTE|frm_conexiones@GridNext|0|0|true|true|Registro anterior
ITEM18=spc3|3|0|true|true|
ITEM19=Salir|mnu_salir_click|0|0|true|true|Salir
PROPIETATS=1|0|0|
STATUS1=CYAN|Aadir nuevos registros|
STATUS2=YELLOW|Consulta/Modificaci3n|
STATUS3=GREEN|Indicar condiciones de filtro|
PREMUNAME=frm_menuspr
PREMENUFRMNAME=frm_menuspr

[ DICCIONARIO_DESACTIVARGLOBAL ]
MAXVBLET=0
MAXMETHOD=1
VBMETHOD01=frm_conexiones@desactivarbotones(5)

[ DICCIONARIO_SOLOSALIR ]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_salir@enabled@true

[ DICCIONARIO_ACTIVAR_AF ]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_asistente@enabled@true

[ DICCIONARIO_DESACTIVAR_AF ]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_asistente@enabled@false

[ DICCIONARIO_ACTIVAR_WH ]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_examinar@enabled@true

[ DICCIONARIO_DESACTIVAR_QBE ]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_qbe@enabled@false

[ DICCIONARIO_ACTIVAR_QBE ]
MAXVBLET=2
MAXMETHOD=0
VBLET01=@mnu_qbe@enabled@true
VBLET02=@mnu_alta@Enabled@True

[ DICCIONARIO_DESACTIVAR_ALTA ]
MAXVBLET=1
MAXMETHOD=0
VBLET01=@mnu_alta@enabled@false

[ DICCIONARIO_ACTIVAR_ALTA ]
MAXVBLET=2
MAXMETHOD=0
VBLET01=@mnu_alta@enabled@true
VBLET02=@mnu_lista@Enabled@True

```

Aquest fitxer conté totes les botoneres definides en el mòdul. Les botoneres estan en seccions amb el nom del procés (com a nom de secció) o bé el procés té definida una botonera estandard (ex: #MB). Apart d'aquestes seccions, n'hi ha d'altres que fan referència al comportament, aquests comportaments van lligats al comportament definit a cada element i al mode (estat) en que es troba el procés.

CLirba.ini

```
[FRM_CTLDEP]
CON01=Ubicaci3n del Recurso en Los Equipos| 2250| 2085| 8835| 3000
CON01COL=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON01EXPR=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON01REFERUSR=1,2,3,4
CON01TABLES=ctlins,ctlctl
CON01WHERE=ctlins.ins_cod{=@dep_dep;c} AND ctlins.ctl_id{=@ctl_id;c}
CON01WIDTHUSR=40,1950,1650,0,4260
CON02=Referencias Cruzadas del Recurso en Los Modulos| 2250| 2085| 8835| 3000
CON02COL=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON02EXPR=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON02REFERUSR=1,2,3,4
CON02TABLES=ctlins,ctlctl
CON02WHERE=ctlins.ins_cod{=@dep_dep;c} AND ctlins.ctl_id{=@ctl_id;c}
CON02WIDTHUSR=40,1950,1650,0,4260
LISTACON=con01,Recurso Por Equipos|con02,Recurso Por Modulos

[FRM_CTLINS]
CON02COL=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON02EXPR=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON02REFERUSR=1,2,3,4
CON02TABLES=ctlins,ctlctl
CON02WHERE=ctlins.ins_cod{=@dep_dep;c} AND ctlins.ctl_id{=@ctl_id;c}
CON02WIDTHUSR=40,1950,1650,0,4260
LISTACON=con01,Recurso Por Equipos|con02,Recurso Por Modulos
CON01=Referencias Cruzadas del Recurso en Los Equipos| 2250| 2085| 8835| 3000
CON01COL=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON01EXPR=ctlins.ins_phstart, ctlins.ins_ver, ctlins.ins_ubi, ctlins.ins_obs
CON01REFERUSR=1,2,3,4
CON01TABLES=ctlins,ctlctl
CON01WHERE=ctlins.ins_cod{=@dep_dep;c} AND ctlins.ctl_id{=@ctl_id;c}
CON01WIDTHUSR=40,1950,1650,0,4260
CON02=Referencias Cruzadas del Recurso en Los Modulos| 2250| 2085| 8835| 3000
```

Aquest fitxer defineix les consultes, m3s apropiat seria dir vistes, que un proc3s pot mostrar. Aquestes vistes es mostren en una interfície creada dinàmicament. Poden ser aniuades, contenir paràmetres del proc3s (o altres processos actius), o bé demanar-los a l'usuari.

WLirba.ini

```
[DICCIONARIO]
@DEPART_COD=DEPARTAMENTOS|1665|3135|-
1|usudepar.depart_cod,depart.depart_des|C"DIGO,DESCRIPCIN|usudepar,depart|usudepar.depart_cod=depart.depart_cod||40,765,2985|5140|4606|True|False
@CTB=EMPRESAS CONTABLES|300|300|-1|ctb,ctb_des|CODIGO,DESCRIPCION|ctb|ctb<>'*'|375,765,3000|5140|4606
@EMPRES_COD=EMPRESAS|675|1215|-1|empres_cod,empres_des|CODIGO,DESCRIPCION|empres|empres_cod<>'*'|375,765,3000|5140|4606
@PAISES_COD=PAISES|4275|5205|-1|paises_cod,paises_des|CODIGO,DESCRIPCION|paises||375,765,3000|5140|4606
@PROVIN_COD=PROVINCIAS|300|300|-1|provin_cod,provin_des|CODIGO,DESCRIPCION|provin||375,765,3000|5140|4606
@USUARI_COD=USUARIOS|300|300|-1|usuari_cod,usuari_des|CODIGO,NOMBRE|usuari|
@MONEDA_COD=MONEDAS|300|300|-1|moneda_cod,moneda_des|CODIGO,DENOMINACION|moneda||375,765,3000|5140|4606
@EGR_COD=E. GESTORAS DE RECIBOS|4155|5730|-1|egr_cod,egr_des|CODIGO,DENOMINACION|egr||375,765,3000|5140|4606
@AGENCI_COD=AGENCIAS|300|300|-1|agenci_cod,agenci_des|CODIGO,DIRECCION|agenci||375,765,3000|5140|4606
@TIPENT_COD=TIPOS DE ENTIDAD|300|300|-1|tipent_cod,tipent_des,clave_enti|CODIGO,DENOMINACION,CLAVE ENT.|tipent||375,765,3000,1065|5140|5690
@CUENTA_COD=CUENTAS|300|300|-1|cuenta_cod,cuenta_des|CODIGO,DENOMINACION|cuenta||cuenta_cod|375,1875,3000|5250|5970
@FORPA_COD=FORMAS DE PAGO|300|300|-1|forpa_cod,forpa_des|C"DIGO,DESCRIPCIN|forpa||375,765,3000|5140|4606
@IMPUESCICOD=IMPUESTOS|300|300|-1|impues_cod,impues_des|C"DIGO,DESCRIPCIN|impues||375,765,3000|5140|4606

[FRM_ACCESO]
@DEPART_COD=DEPARTAMENTOS|1665|3135|-
1|usudepar.depart_cod,depart.depart_des|C"DIGO,DESCRIPCIN|usudepar,depart|usudepar.depart_cod=depart.depart_cod||40,765,2985|5140|4606|False|False

[FRM_USUARIO]
LISTAEXPR=usuari_cod,usuari_des,segur_cod,empres_cod,depart_cod
LISTACOL=Usuario,Nombre,C3d. Seguridad,Empresa,Departamento
LISTAOCULTOS=1
LISTACAMPOSOCULTOS=usuari_cod
LISTACLAVES=c
LISTAREFERUSR=1,2,3,4,5||
LISTAWIDTHUSR=40,,1830,2640,1335,810,4455
LISTAEXTENDRIGHTCOLUMN=0

[FRM_CTB]
LISTAEXPR=ctb,ctb_des,iniejerfis,ejeractual,inibloqueo,finbloqueo
LISTACOL=Ctb,Nombre,I.Ejerc.,Ejerc.Act.,I.Bloqueo,F.Bloqueo
LISTAOCULTOS=1
LISTACAMPOSOCULTOS=ctb
LISTACLAVES=c
LISTAREFERUSR=1,2,3,4,5,6
LISTAWIDTHUSR=40,,780,3525,660,975,1095,4515,0,0
LISTAEXTENDRIGHTCOLUMN=0
```

Aquest fitxer defineix les llistes i els Lookups. Un lookup és una clau forana que genera una llista per poder seleccionar un element. La llista és el mateix que el lookup però sobre la clau principal, permet moure's per la selecció de registres/instàncies d'un determinat flux (al seleccionar un element de la llista el flux s'inicialitza amb la instància del registre seleccionat).

Els lookups solen tenir un botó en la botonera per tots ells, però que s'activa al tenir l'enfoc de l'usuari en un de concret (i s'inicialitza amb les dades d'aquest fitxer).

La llista (una per cada flux), pot estar dins de la mateixa interfície del flux o bé en una interfície a part.

MLirba.ini

```
[DICCIONARIO]
@ACRDEU_COD=frm_acrdeu/Acreedor/Deudor
@ACTIVI_COD=frm_activi/Actividad
@AGENCI_COD=frm_agenci/Agencia
@ARANCE_COD=frm_arance/Arancel
@ARTICU_COD=frm_articu/Articulo
@EGR_COD=frm_egr/E.G.R.
@FAMART_COD=frm_famest/Familia Articulo
@FAMCLI_COD=frm_famest/Familia Cliente
@FAMEST_COD=frm_famest/Familia
@IDIOMA_COD=frm_idioma/Idioma
@LINFIN_COD=|Linea financiera
@PROVEE_COD=frm_acrdeu/Proveedor
@TARIFA_COD=frm_tarifa/Tarifa
@TIPENT_COD=frm_tipent/Tipo Entidad
@TRANSP_COD=frm_transp/Transportista
@ZONAS_COD=frm_zonas/Zona
```

Aquest fitxer defineix els manteniments, tot lookup sol tenir associat un manteniment, per tant aquí es defineix quin procés/flux de treball fa el manteniment del registre d'aquella clau forana. No necessàriament el procés ha d'estar en el mateix executable (el nom del mòdul és una indirecció al mòdul definit en la secció del ngstart del mòdul actual).

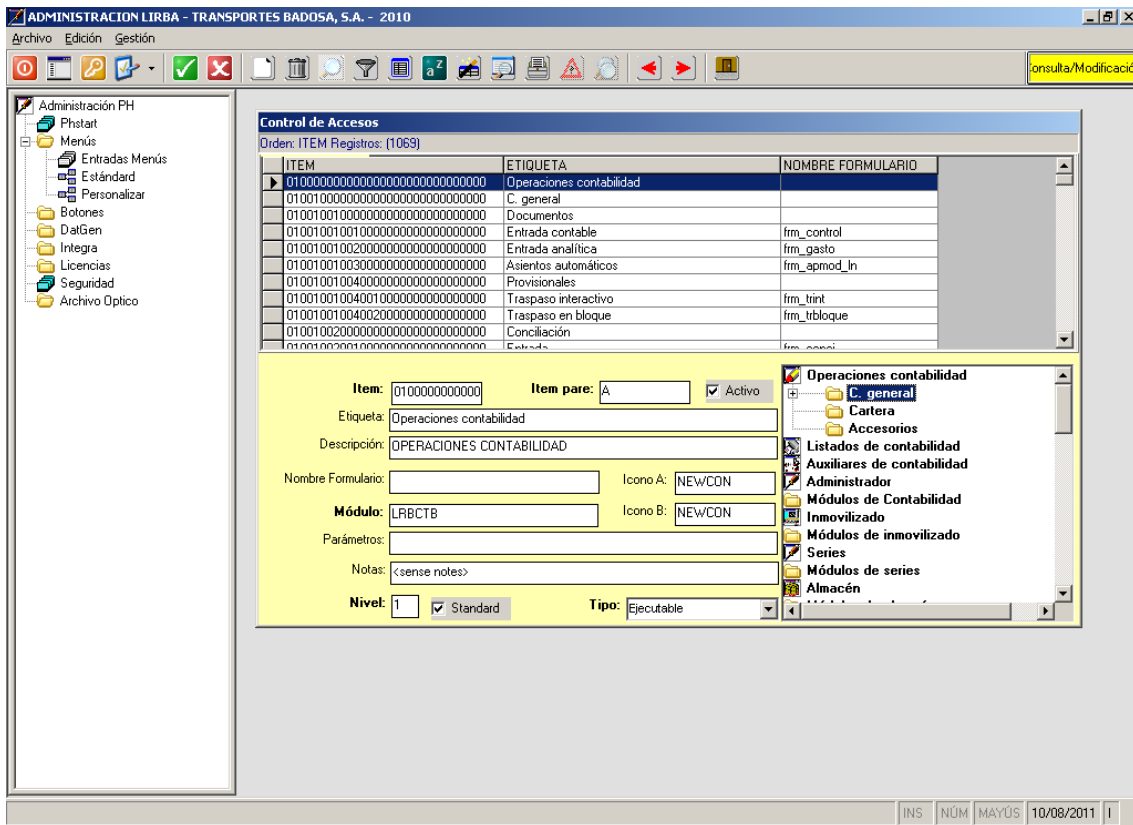
TLirba.ini

```
[TEMPORAL1]
CREATEI=create table temp (tmp_cod numeric(6,0),codi char(4),descrip char(10))
DROP=drop table temp
```

Aquest fitxer defineix les taules temporals, el nom és únic per tot el mòdul i els processos les poden reutilitzar. Una aplicació típica és per la realització d'informes en Crystal Reports.

A cada implantació s'haurà d'afegir als fitxers anteriors els valors necessaris per fer el manteniment i les crides de l'arxiu òptic en els processos/fluxos dels diferents mòduls (això es pot fer afegint un boto a les botoneres que és vulgui i fer un element lookup que sigui el vincle de la instància del flux amb el document associat). Per comoditat s'ha fet una implantació estàndard i només cal copiar aquesta on calgui.

Apart dels fitxers anteriors també s'haurà de definir els punts de menú, veure imatge.



I la seguretat associada, veure imatge.

