



LibreECM: Aplicació de gestió de recursos documentals

Jordi Pérez Bayerri
Grau Enginyeria Informàtica
Àrea de treball final

Consultor: Albert Grau Perisé
Professor: Santi Caballe Llobet

14 de juny de 2016



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/)

Table of Contents

1	Introducció.....	3
2	Descripció del projecte.....	5
2.1	Justificació del projecte.....	5
2.2	Objectius del projecte.....	7
2.3	Estudi de mercat.....	8
3	Metodologia.....	12
4	Anàlisis dels requisits.....	13
5	Especificació.....	27
6	Disseny.....	31
6.1	Arquitectura.....	31
7	Implementació.....	33
7.1	Creació i compilació de projecte OpenCMIS.....	33
7.2	Creació de components.....	34
8	Conclusions i estat de l'art.....	39
8.1	Estat de l'art.....	39
9	Glossari de termes.....	43
10	Bibliografia.....	44
11	Manual d'usuari.....	45
11.1	Compilació.....	45
11.2	Execució.....	46

1 Introducció

El present projecte, realitza la implementació d'un servei gestor de continguts sota el paraigües de [Enterprise Content Management](#). que Wikipedia defineix com:

«Enterprise Content Management (ECM) is the strategies, methods and tools used to capture, manage, store, preserve, and deliver content and documents related to organizational processes. ECM covers the management of information within the entire scope of an enterprise whether that information is in the form of a paper document, an electronic file, a database print stream, or even an email.»^[1]

Per tant, un ECM ha de donar servei d'emmagatzemament de continguts ja sigui en format fitxer o en un format que permeti emmagatzemar informació, com seria una taula d'una base de dades.

Així, els serveis que presenta son:

- Gestió de continguts
- Cerca de continguts de forma jeràrquica
- Descàrrega de documents
- Introducció de documents.
- Gestió d'usuaris

En general, el sistema ha de cobrir els requeriments d'una empresa generalista, des del punt de vista de administració, gestió i informació.

Per tant, per complir aquest requisits, l'eina ha de proveir al menys:

- Serveis d'emmagatzemament de continguts.
- Serveis de cerca de continguts.
- Serveis d'entrada de continguts.

Altres serveis complementaris desitjables que ofereixen les solucions enmarcades en ECM poden ser:

- Diferents serveis d'emmagatzematge: local, al núvol privat o en un servei de núvol.

- Diferents sistemes d'entrada d'informació: Web, bases de dades, sistemes propietaris, DataWarsehouse...
- Diferents sistemes de sortida de la informació: Serveis de gestió de continguts web, serveis de creació de documents, eines de presa de decisió (BPM), correu electrònic, etc.
- Cerques i administració col·laboratives.
- Gestió i actualització de bens digitals
- Gestió de fluxos de treball
- Gestió automatitzada de digitalitzacions de documents físics: gestió d'escàners, digitalització i classificació de documents (OCR), etc.

La implementació de tots aquest serveis, supera les capacitats temporals d'aquest projecte, per la qual cosa, m'he centrat en la gestió de continguts.

2 Descripció del projecte

2.1 Justificació del projecte

Actualment, qualsevol empresa gestiona multitud d'informació provinent de moltes fonts. Molts cops, aquesta informació es rep en formats diferents, incompatibles entre ells:

- Pdf, Word, PowerPoint o mails amb ofertes comercials
- Contractes en Word o PDF.
- Informacions en Excel.
- Contabilitat en una base de dades.
- Arxius de clients en formats binaris

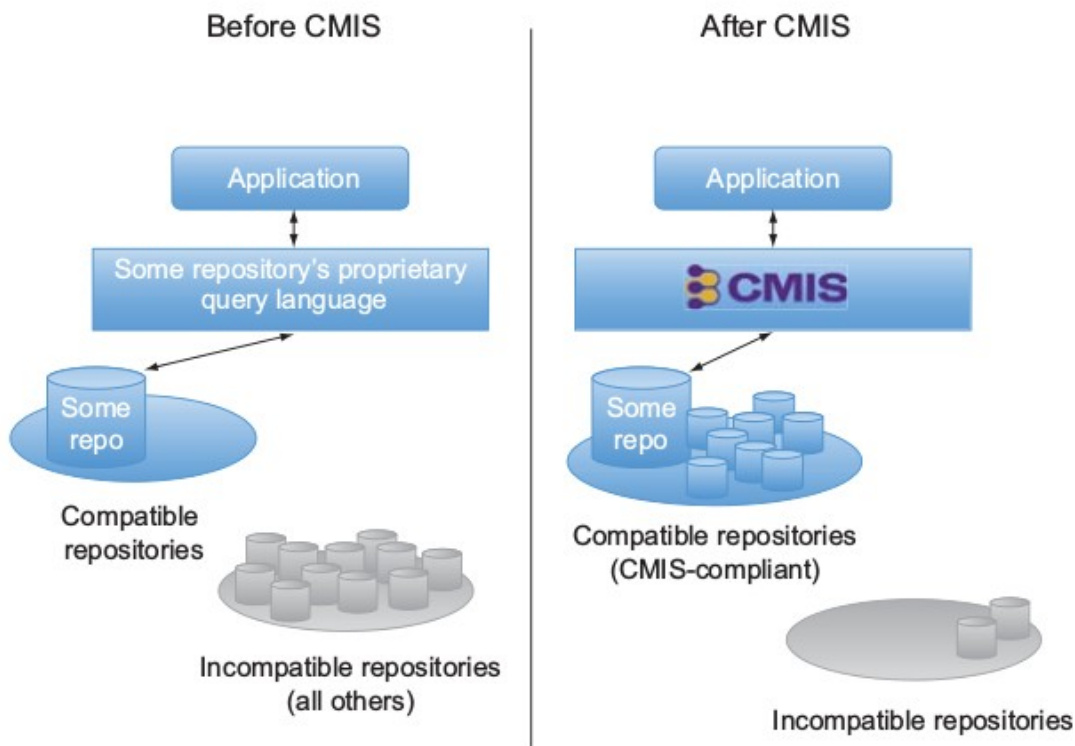
L'organització, control, seguiment, accessibilitat i gestió de tota aquesta informació, es molt complex, i requereix de múltiples sistemes d'accés, amb API diferents, que fan de la presa de decisions una tasca més complexa.

Totes aquestes fonts d'informació, fan que hi hagi multitud de sistemes d'accés a la informació. I com més gran és l'empresa, més complex es el problema.

A finals dels anys 90, es va començar una tendència cap a la estandardització de protocols, com per exemple és SQL, que permet accedir a multitud de bases de dades des d'un sistema pràcticament unificat.

Seguint aquest esquema, l'any 2010, el consorci OASIS (EMC, IBM, Microsoft, Alfresco, OpenText, Oracle, i SAP entre altres), va publicar un estàndard de comunicació entre serveis d'informació, concretament, entre serveis de continguts, anomenat Content Management Interoperability Services (CMIS) versió 1.0. Posteriorment, al 2010, es va publicar una ampliació de l'estàndard, la versió 1.1.

La idea és, a semblança del que fa SQL, crear un sistema unificat i general d'accés a multitud de serveis de continguts, ja siguin arxius, bases de dades, informació binària, etc.



Il·lustració 1: De: CMIS and Apache Chemistry in Action

Amb el protocol definit, i una implementació com es Chemistry, sota el paraigües del projecte Apache, es necessari crear tot de sistemes que siguin capaços de llegir multitud d'òrgens i fer-los accessibles per qualsevol client de CMIS.

Això és el que pretén fer LibreECM, ser un sistema pont entre un dipòsit de continguts i els clients, mitjançant la interfície OpenCMIS, la implementació en Java de Chemistry.

2.2 Objectius del projecte

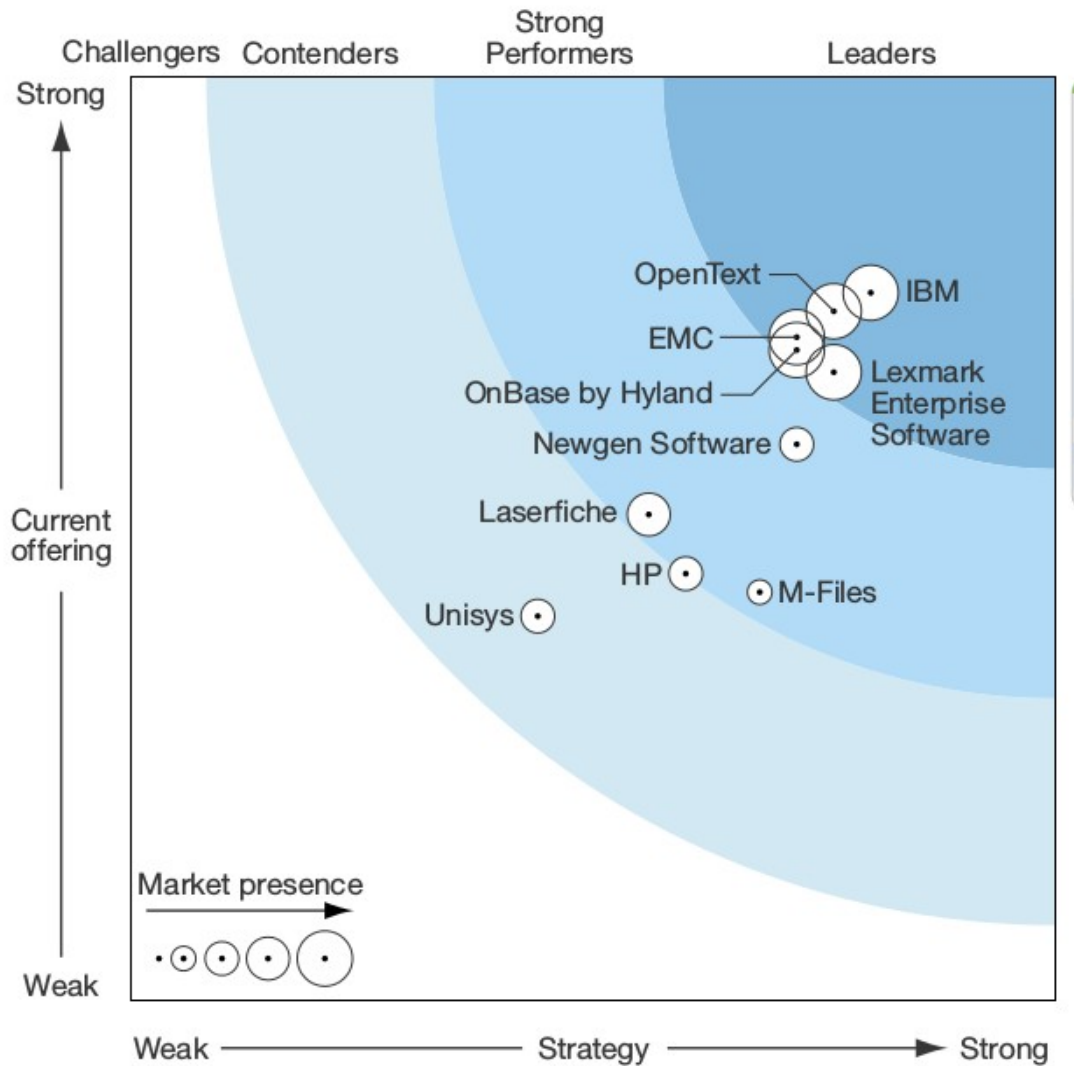
El present projecte preten oferir una eina ECM per la gestió de continguts, amb les següents característiques:

- Modularitat: el projecte ha de oferir diversos components de fàcil gestió i administració que permetin ampliar o modificar característiques sense haver de refer tot el sistema. Aquests mòduls han de ser el màxim d'independents entre ells, per possibilitar la interoperabilitat amb múltiples implementacions dels mateixos. Així, es proveeixen els següents components:
 - LibreECMMain: component principal del sistema, és el punt de connexió entre tots els mòduls i l'encarregat de gestionar la lògica de negoci i comprovar la seguretat.
 - LibreECMInput: component encarregat de llegir els continguts des de la seva persistència. Es per tant l'encarregat de llegir els arxius, bases de dades, aplicacions, formats binaris, XML, etc. on pugui residir la informació.
 - LibreECMDDBB: encarregat de la persistència de les dades internes del sistema, bàsicament ubicació i propietats dels dipòsits de continguts, i la informació sobre els usuaris i els seus drets.
 - LibreECMOpenCMIS: és una de les implementacions per accedir al sistema des de la part client. En aquest cas, s'implementa gracies a OpenCMIS un protocol estandarditzat per accedir a dipòsits de continguts.
 - LibreECMConfigurator: eina de configuració web del sistema, que permet administrar orogens de dipòsits i administrar usuaris.
 - LibreECMEvent: eina que gestiona el esdeveniments desencadenats al realitzar qualsevol tasca. Aquests esdeveniments es defineixen gracies a els objectes Polítiques (Policy).
- Accés estandaritzat: per donar la major facilitat d'accés al sistema, s'empra CMIS com a estàndard d'accés per la part client. Això permet que el sistema sigui directament accessible per multitud de sistemes existents. També es preveu donar accés estandaritzat a la gestió d'events, com seria Java SEDA o vincular la solució a un gestor d'events com ara Mule ESB

2.3 Estudi de mercat

L'àmbit en que es mou el nostre projecte hi ha molta oferta al mercat.

Segons un estudi de Forrester, del 3er quart del 2015, les principals solucions del sector son:



- [Enterprise content manager](#) (IBM): conjunt de serveis proveïts per IBM com sota el paraigües d'aquesta denominació, cada secció està formada al seu torn, de nous grups de paquets de solucions independents, però interoperables:
 - [Content management and imaging](#): Grup d'utilitats per automatitzar la digitalització i gestionar continguts, proveïnt d'accés des de qualsevol dispositiu
 - Unifica en un punt documents de múltiples orígens: físics, web, documents de qualsevol format, derivats d'altres eines.

- Accessible des de qualsevol dispositiu.
- Gestió col·laborativa del document.
- Vinculable a un núvol privat.
- Classifica el contingut en temps real
- Capacitats de generar informes de resposta al actualitzar un document o informació.
- [Advanced Case Management](#): Utilitats que gestionen continguts i els incorporen dins dels processos de l'empresa. Automatització de la gestió de continguts.
 - Mitjançant interfícies i plantilles, produeixen resultats analítics, cerques avançades i informes.
 - Solucion al núvol i multi-dispositius.
 - Sistema d'auditoria automàtic: per rendiment, legal, seguretat, etc.
- [Information Lifecycle Governance](#): Eines d'anàlisis de la informació i aplicació de polítiques gestionables en resposta al contingut.
 - Gestiona i manté informació al llarg de tot el seu cicle de vida
 - Actualitza informes i actes al llarg del cicle de vida de la informació, comparant-la amb altres documents, però també amb versions anteriors, per evaluar-ne l'evolució.
- [OpenText](#) (OpenText):
 - Gestiona els fluxes d'informació des de la captura fins l'arxivament i posada a disposició.
 - Aplica polítiques de resposta adaptables a qualsevol tipus de contingut, independentment de l'origen, tant intern com extern.
 - Adapta la resposta a les necessitats de format, posant a disposició eines de vinculació amb les principals utilitats de gestió de SAP, Oracle o Microsoft.
 - Multi-dispositiu i multiplataforma.
 - Accés ràpid, segur i optimitzable en funció dels processos.
 - Capa d'eines extensible amb una API per adaptar a qualsevol nova situació.

- [Eines Lexmark](#) (Lexmark): Lexmark ha comprat diverses empreses del sector ([Kofax](#), ReadSoft, Saperion) i està en procés d'unificació i de consolidació dels serveis que cadascuna aporta.
 - Automatització de processos de transformació de la informació, independentment de l'origen d'aquesta: paper, documents, generat per aplicacions o del web.
 - Seguretat granular avançada
 - Flexibilitat organitzacional.
 - Cerques avançades.
 - Llibreria de serveis i APIs.
- [Documentum](#) (EMC Corporation)
 - Gestió de documents i el seu cicle de vida
 - Gestió i administració de la captura i digitalització de documents físics.
 - Sistemes de comunicació i informació de canvis en documentació.
 - Automatització de processos per la presa de decisions.
- [OnBase](#) (Hyland): Unifica serveis de documentació amb BPM en una sola utilitat.
 - Operativa multi-dispositiu, amb emmagatzematge local o al núvol.
 - Plataforma segura.
 - Captura i administra continguts de múltiples orígens: físic, documents, informes, BPM, etc.
 - Automatització avançada de processos en resposta a aconteixements.
 - Integració en aplicacions de tercers via mòduls.
 - Obtenció de resultats i anàlisis de manera plana i simple, sense necessitat de programació o consultes complexes.

Altres alternatives les remarco a continuació:

- [DocuShare](#) (Xerox):
 - Automatització de gestió de la informació
 - Gestió de contractes

- Polítiques d'accés, gestió de processos i publicació avançades.
- Gestió d'informació personal
- [Oracle Enterprise Manager](#) (Oracle): Mitjançant la utilitat Cloud Manager, posa a disposició de l'usuari, en un sol punt, accés sincronitzat a diversos serveis de núvol.
- [Microsoft Sharepoint](#) (Microsoft): Utilitat que serveix com a punt d'accés a la informació, ja sigui en format fitxer, com en qualsevol format. Proveeix dels següents serveis:
 - Intranet, extranet
 - Administració de continguts i de documents: Manté, emmagatzema i fa seguiment de documentació i informació empresarial. S'integra amb els productes Office, permetent edició síncrona dels continguts. Manté control de versions.
 - Permet edició col·laborativa de documents.
 - Núvol personal.
 - Xarxa social empresarial privada.
 - Cerques empresarials, Business intelligence i administració de fluxes de treball
 - Administració de continguts web
- [Alfresco](#) (Alfresco Software): solució de codi lliure
 - Referència comercial en quan a la implantació de Chemistry
 - Repositori de documents
 - Interfície de treball col·laboratiu.
 - Integració en Office i Google Docs
 - Núvol híbrid (local i extern)
 - Extensible amb mòduls
- [Nuxeo](#) (Nuxeo): solució de codi lliure
 - Repositori de documents
 - Arquitectura extensible amb mòduls
 - Integració al núvol (local i extern)
 - API completa d'accés al servei.

3 Metodologia

La metodologia emprada en el present projecte es basa en la clàssica de cascada. S'ha implementat els components més bàsics i independents i s'ha anat ampliant la funcionalitat segons l'estructura següent:

Les fases han estat:

- Anàlisi de requisits
- Especificació
- Disseny
- Implementació, i dins d'aquesta fase, l'ordre d'implementació de components:
 1. Estudi de CMIS. S'ha estudiat el protocol definit per CMIS, els objectes i com implementar un servidor
 2. Disseny dels objectes bàsics: Repository, Users. Codificació del component LibreECMCommonAPI, un component que defineix les classes que utilitzaran la resta de components. A més, gestiona l'arxiu de configuració en XML que la resta de components utilitzarà. També defineix algunes eines comunes.
 3. Codificació del component LibreECMBBDD. Disseny de les classes JPA, son classes equivalents a les de LibreECMCommonAPI però incorporant anotacions per la persistència.
 4. Codificació del component LibreECMMain. Es el punt d'accés al sistema, que coordina la resta de components. Hauria de ser el component que implementés la seguretat i fes les comprovacions de lògica de negoci.
 5. Codificació del component LibreECMInput. Aquest component gestiona els continguts dels dipòsits. Inicialment es defineixen tres tipus de dipòsit, FSRepository, DBRepository i ZippedFSRepository. Només s'implementa FSRepository, que emmagatzema els continguts en arxius.
 6. Codificació del component LibreECMOpenCMIS. Aquest component implementa l'accés dels client pel protocol CMIS. Utilitza, per tant les llibreries Chemistry OpenCMIS, i bàsicament fa una conversió dels objectes del format de LibreECM a OpenCMIS.
- Verificació i correcció d'errors.

4 Anàlisi dels requisits

CMIS defineix una sèrie de serveis que s'han d'implementar per tenir un sistema compatible amb CMIS v1.1.

- Serveis de dipòsits
 - Obtindre dipòsits
 - Obtindre informació de dipòsits
 - Obtindre tipus d'objectes
 - Obtindre tipus heredats d'objectes
 - Obtindre definicions de tipus
- Serveis de navegació
 - Obtindre objectes fills.
 - Obtindre objectes continguts a carpeta.
 - Obtindre arbre de carpetes.
 - Obtindre carpeta pare.
 - Obtindre objectes pares.
- Serveis d'objectes.
 - Crear document
 - Crear document a partir de fluxe de dades.
 - Crear carpeta
 - Crear relació
 - Crear política
 - Obtindre accions permesses
 - Obtindre objecte
 - Obtindre propietats
 - Obtindre objecte per ruta
 - Obtindre fluxe de dades d'un objectes
 - Obtindre representacions (thumbnails)
 - Actualitzar propietats

- Moure objecte
- Esborrar objecte
- Esborrar arbre de carpetes
- Assignar fluxe de dades
- Esborrar fluxe de dades
- Serveis de versionament
 - Confirmació versió (check-out)
 - Cancel·lar versió
 - Iniciar versió (check-in)
 - Obtindre darrera versió d'objecte
 - Obtindre propietats de darrera versió d'objecte
 - Obtindre totes les versions
- Serveis de carpetes
 - Afegir objecte a carpeta
 - Eliminar objecte de carpeta
- Serveis de descoberta
 - Petició (query)
 - Obtindre canvis en objecte
- Serveis de relacions
 - Obtindre relacions d'objecte
- Serveis de polítiques
 - Aplicar política
 - Eliminar política
 - Obtindre polítiques aplicades
- Serveis de control d'accés (ACL)
 - Obtindre controls (ACL)
 - Aplicar controls

A més, dels serveis descrits en el protocol CMIS, s'han definit altres serveis complementaris per LibreECM:

- Serveis de dipòsits
 - Afegir dipòsit
 - Eliminar dipòsit
 - Definir dipòsit.
- Serveis de configuració
 - Definir port de component
 - Definir URL de component
 - Definir carpeta temporal
- Serveis d'usuaris
 - Afegir usuari
 - Eliminar usuari
 - Modificar rol usuari
 - Afegir rol
 - Eliminar rol
- Serveis de tipus
 - Afegir tipus de contingut
 - Eliminar tipus de contingut

Seguint aquests serveis, s'han definit els següents casos d'un, realitzats per dos rols d'actors que intervenen al sistema:

- Administradors: persones que configuren el sistema
- Usuaris: demandants de serveis

Cadascú d'ells tindrà una interfície d'accés diferenciada, la primera per la configuració i manteniment del sistema i la segona però seran serveis que empraran el nostre sistema.

Es poden identificar els següents casos d'ús de l'aplicació:

- a) Administrador que gestiona usuaris ([Veure Representació](#)):

- i. Alta, modificació o esborrat d'usuaris.

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Alta d'usuari
Alies:	AddUser
Actors:	Administrador
Funció:	Registrar un nom d'usuari i un codi d'accés al sistema
Descripció:	Un administrador registra el nom i un codi d'accés d'un usuari nou. El sistema verifica que no existeixi anteriorment. L'administrador registra a quins dipòsits es donarà accés, i a quin nivell (total o lectura).
Referències d'extensió:	
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Modificació d'usuari
Alies:	UpdateUser
Actors:	Administrador
Funció:	Modifica els drets o el codi d'accés o el rol d'un usuari existent
Descripció:	Un administrador modifica la configuració d'un usuari existent, actualitzant el seu codi d'accés o modificant el nivell d'accés dels dipòsits als que té accés. També pot modificar els drets d'administració.
Referències d'extensió:	
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS

Nom:	Eliminació d'usuari
Alies:	DeleteUser
Actors:	Administrador
Funció:	S'elimina un usuari del sistema
Descripció:	Un administrador elimina un usuari del sistema
Referències d'extensió:	
Referències d'inclusió:	

b) Administrador que modifica comportament del sistema ([Representació](#)):

i. Llistar, consultar, navegar, afegir, modificar o esborrar dipòsits.

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Llistar dipòsits
Alies:	DiscoverRepositories
Actors:	Administrador, Usuari
Funció:	Obtenir una llista dels dipòsits disponibles
Descripció:	L'usuari fa una consulta i obté un llistat amb els dipòsits disponibles. Per cada dipòsit obté el seu identificador ID.
Referències d'extensió:	
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Consulta dipòsit
Alies:	AskRepository
Actors:	Administrador, Usuari

Funció:	Consulta les capacitats d'un dipòsit
Descripció:	Un usuari obté les dades de capacitats i propietats d'un dipòsit, i un diccionari de dades (Data Dictionary) dels tipus suportats.
Referències d'extensió:	ReadRepositoryRights
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Navegar dipòsit
Alies:	NavigateRepository
Actors:	Usuari
Funció:	Dona accés a la carpeta arrel d'un dipòsit
Descripció:	Un usuari obté accés a la carpeta arrel d'un dipòsit. Es comproven les credencials de l'usuari.
Referències d'extensió:	ReadRepositoryRights
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Afegir dipòsit
Alies:	AddRepository
Actors:	Administrador
Funció:	Afegeix un nou dipòsit al sistema
Descripció:	Un administrador afegeix un nou dipòsit als existents. També afegeix els tipus de continguts personalitzats, definits en el dipòsit, al sistema. Els nous tipus només seran accessibles al nou dipòsit.
Referències	AddContentType

d'extensió:	
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Esborrar dipòsit
Alies:	DeleteRepository
Actors:	Administrador
Funció:	Un administrador esborra un dipòsit de la llista de disponibles.
Descripció:	Un administrador modifica la configuració d'un usuari existent, actualitzant el seu codi d'accés o modificant el nivell d'accés dels dipòsits als que té accés. S'eliminen els tipus de continguts personalitzats que pugui tenir el dipòsit.
Referències d'extensió:	DeleteContentType
Referències d'inclusió:	

ii. Gestionar els ports d'accés al sistema

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Modificar port web
Alies:	UpdateWebPort
Actors:	Administrador
Funció:	Un administrador modifica els ports d'accés a l'administració web.
Descripció:	Un administrador modifica la configuració del servidor Facelet que serveix un JSP per configurar el sistema. El port per defecte es 8080
Referències d'extensió:	
Referències	

d'inclusió:	
--------------------	--

iii. Llegir i vincular usuaris i dipòsits i els seus drets.

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Modificar drets d'un usuari a un dipòsit
Alies:	UpdateRepositoryRights
Actors:	Administrador
Funció:	Un administrador modifica els drets d'un usuari respecte a un dipòsit
Descripció:	Un administrador modifica els drets d'un usuari concret respecte a un dipòsit concret (ACL). Els drets poden ser lectura, escriptura, total o definits pel dipòsit. En aquest darrer cas es dona accés total i es la gestió interna del dipòsit la que gestiona els drets. La modificació té conseqüències sobre l'estat actual dels continguts.
Referències d'extensió:	
Referències d'inclusió:	UpdateContent

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Llegir drets d'un usuari a un dipòsit
Alies:	ReadRepositoryRights
Actors:	Administrador
Funció:	Un administrador obté els drets d'un usuari respecte a un dipòsit
Descripció:	Un administrador obté els drets d'un usuari concret respecte a un dipòsit concret (ACL). Els drets poden ser lectura, escriptura, total o definits pel dipòsit.
Referències d'extensió:	

Referències d'inclusió:	
--------------------------------	--

iv. Gestió dels mime-types associats a continguts.

DESCRIPCIÓ DE CAS D'ÚS	
Nom:e	Modificar mime-types
Alies:	UpdateMimeType
Actors:	Administrador
Funció:	Un administrador modifica el llistat de Mime-types
Descripció:	Un administrador modifica el llistat de Mime-types. Els arxius amb un mime-type modificat o eliminat, el mantenen.
Referències d'extensió:	
Referències d'inclusió:	UpdateRendition

v. Gestió dels intèrprets dels continguts.

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Afegir intèrpret
Alies:	AddRendition
Actors:	Administrador
Funció:	Un administrador afegeix els intèrprets d'un mime-type
Descripció:	Un administrador modifica la configuració d'un mime-type, afegint un intèrpret. Els intèrprets son cadenes de text que el component de Vista ha de gestionar per mostrar o editar l'arxiu.
Referències d'extensió:	
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Modificar intèrpret
Alies:	UpdateRendition
Actors:	Administrador
Funció:	Un administrador modifica els intèrprets d'un mime-type
Descripció:	Un administrador modifica la configuració d'un mime-type, afegint, modificant o eliminant intèrprets. Els intèrprets son cadenes de text que el component de Vista ha de gestionar per mostrar o editar l'arxiu.
Referències d'extensió:	
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Esborrar intèrprets
Alies:	DeleteRendition
Actors:	Administrador
Funció:	Un administrador elimina un intèrpret d'un mime-type
Descripció:	Un administrador modifica la configuració d'un mime-type, eliminant un intèrpret. Els intèrprets son cadenes de text que el component de Vista ha de gestionar per mostrar o editar l'arxiu.
Referències d'extensió:	
Referències d'inclusió:	

vi. Crear, modificar o esborrar tipus de continguts.

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Crear tipus de contingut

Alies:	AddContentType
Actors:	Administrador
Funció:	Un administrador afegeix un nou tipus de contingut.
Descripció:	Un administrador afegeix al sistema un nou tipus de contingut que ha de heretar d'un d'existent.
Referències d'extensió:	ReadRepositoryRights
Referències d'inclusió:	AddRendition

DESCRIPCIÓ DE CAS D'ÚS

Nom:	Modificar tipus de contingut
Alies:	UpdateContentType
Actors:	Administrador
Funció:	Un administrador modifica un nou tipus de contingut.
Descripció:	Un administrador modifica al sistema un tipus de contingut ja existent. Implica modificar tots els tipus de continguts hereus.
Referències d'extensió:	UpdateContentType, UpdateContent, ReadRepositoryRights, ModifyRendition
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS

Nom:	Eliminar tipus de contingut
Alies:	DeleteContentType
Actors:	Administrador
Funció:	Un administrador elimina un tipus de contingut.
Descripció:	Un administrador elimina del sistema un tipus de contingut

	existent. Tots els continguts del tipus son convertits al tipus pare. Els tipus de continguts que hereten d'aquest, passen a heretar del tipus pare. Els tipus originals no son eliminables.
Referencies d'extensió:	UpdateContentType, UpdateContent, ReadRepositoryRights, DeleteRendition
Referencies d'inclusió:	

vii. Definició dels esdeveniments que desencadenen accions

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Modificar esdeveniments
Alies:	UpdateEvents
Actors:	Administrador
Funció:	Un administrador modifica els esdeveniments
Descripció:	Un administrador modifica la configuració dels esdeveniments que s'executaran a cada modificació d'un contingut. Els esdeveniments son objectes que executen altres processos. Hi ha un objecte esdeveniment pare.
Referencies d'extensió:	ReadRepositoryRights
Referencies d'inclusió:	UpdateContent

c) Usuaris que accedeixen a dipòsits. ([Representació](#))

i. Lectura, inserció, modificació o esborrat de continguts.

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Lectura del contingut
Alies:	ReadContent
Actors:	Usuari
Funció:	Un usuari obté una copia d'un contingut

Descripció:	Un usuari fa una petició d'un contingut, el sistema comprova si té els drets necessaris, i en funció d'això envia una còpia de l'arxiu o denega l'accés.
Referències d'extensió:	ReadRepositoryRights
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Modificació de contingut
Alies:	UpdateContent
Actors:	Usuari
Funció:	Un usuari modifica un contingut
Descripció:	Un usuari fa una petició de modificació d'un contingut, el sistema comprova si té els drets necessaris, i en funció d'això crea una còpia de l'arxiu o denega l'accés. Si s'accepta la modificació, es crea una nova versió de l'arxiu, es marca l'anterior com a no darrera versió, i es desa al dipòsit.
Referències d'extensió:	ReadRepositoryRights
Referències d'inclusió:	

DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Modificació de carpeta d'un contingut
Alies:	UpdateContentFolder
Actors:	Usuari
Funció:	Un usuari modifica la carpeta (o carpetes) d'un contingut
Descripció:	Un usuari fa una petició de modificació (afegir o eliminar) la carpeta (o carpetes) d'un contingut, el sistema comprova si té els drets necessaris, i en funció d'això realitza la modificació, i es

	desa al dipòsit.
Referències d'extensió:	ReadRepositoryRights
Referències d'inclusió:	

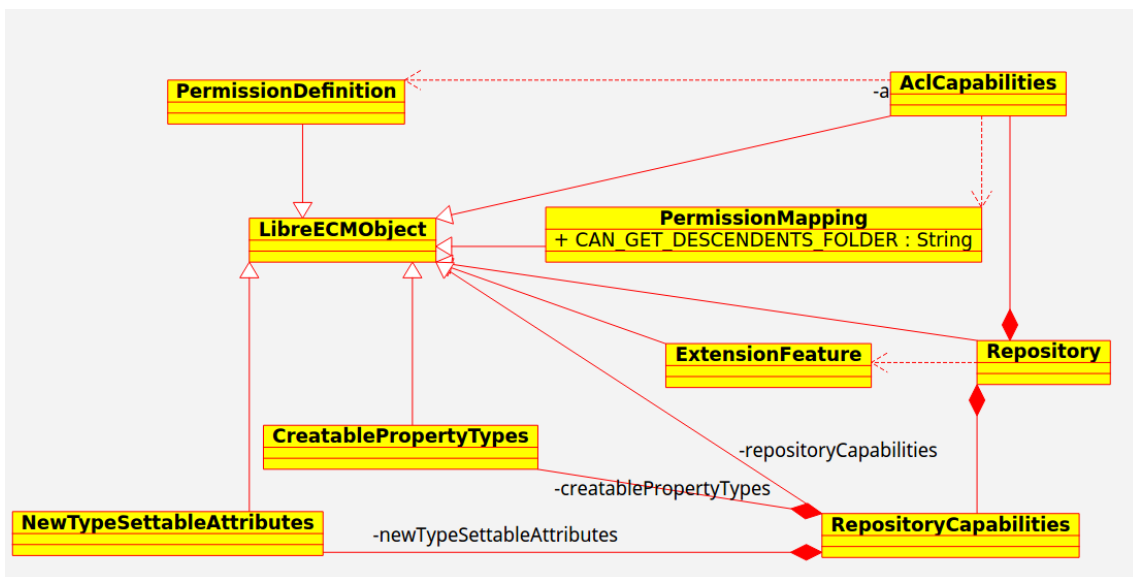
DESCRIPCIÓ DE CAS D'ÚS	
Nom:	Lectura de versió del document
Alies:	ReadDocumentVersion
Actors:	Usuari
Funció:	Un usuari obté una còpia d'un document que ja no es actual
Descripció:	Un usuari fa una petició d'un document que ja no es la darrera versió, el sistema comprova si té els drets necessaris, si existeix la versió demanada, i en funció d'això envia una còpia de l'arxiu o denega l'accés.
Referències d'extensió:	ReadRepositoryRights
Referències d'inclusió:	

5 Especificació

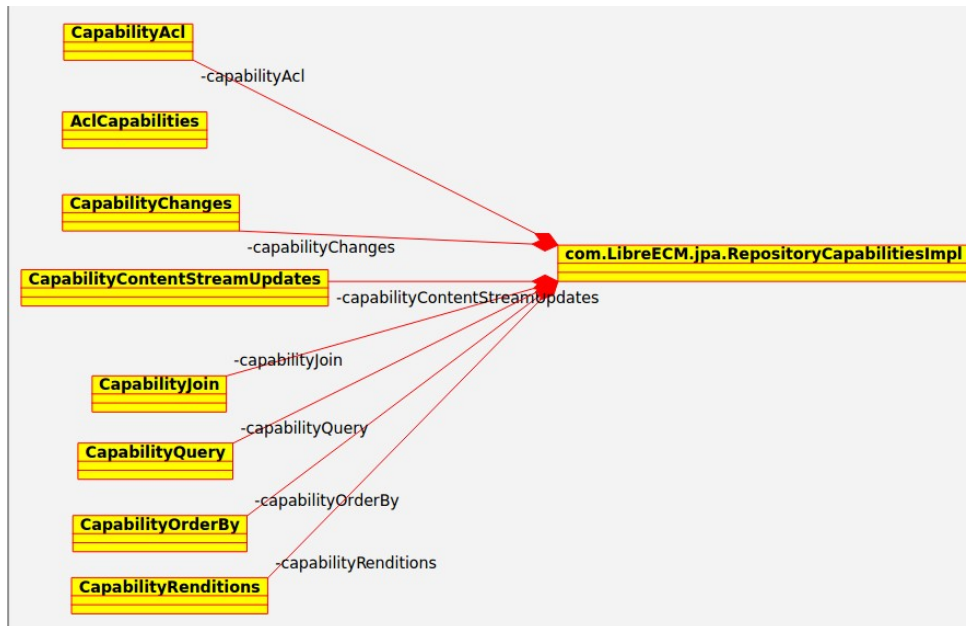
L'especificació del sistema es la següent, definint els principals objectes que intervenen en el present projecte:

Per la part dels dipòsits, l'estructura de classes dels dipòsits es la següent, composant-se un dipòsit de:

- Unes capacitats d'accés ACL, que cada capacitat, va associada a una definició de permisos
- Unes capacitats del dipòsit, que defineixen com implementa un dipòsit concret els objectes, quin objectes son permesos, quin tipus, i quines propietats té cadascun.

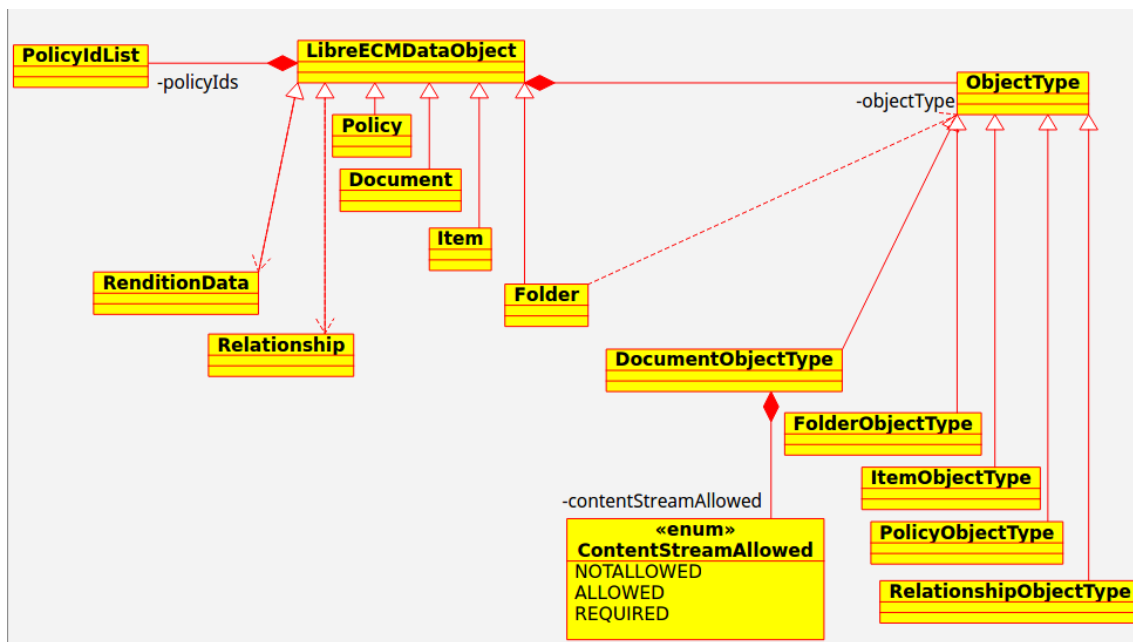


Les capacitats potencials de cada dipòsit son:



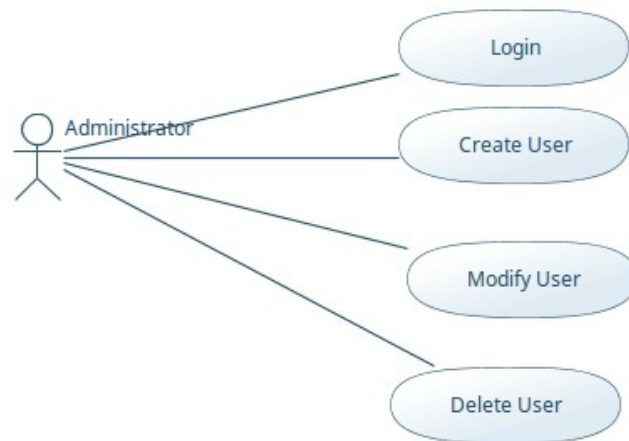
Per la part dels objectes, aquesta es l'estructura, tota derivada de LibreECMDataObject:

- Hereten Item, Document, Folder, Relationship, Policy i Rendition
- Es defineix un ObjectType que generarà herència sobre un tipus per cada objecte bàsic.

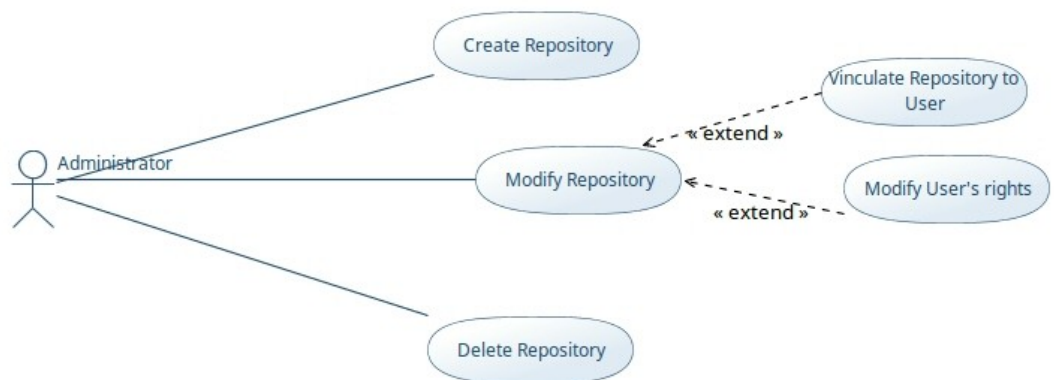


Així, els cassos d'ús a partir d'aquestes estructures son:

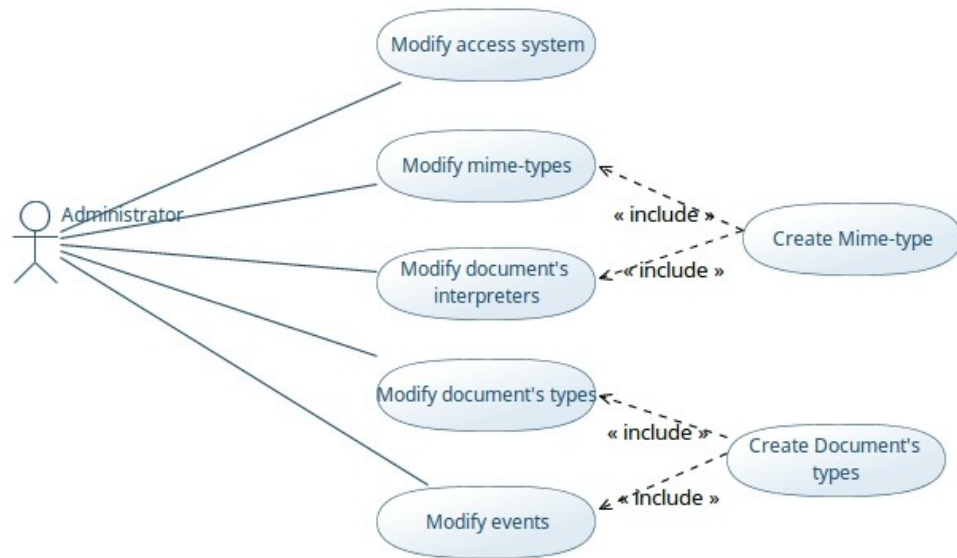
i. Alta, modificació o esborrat d'usuaris.



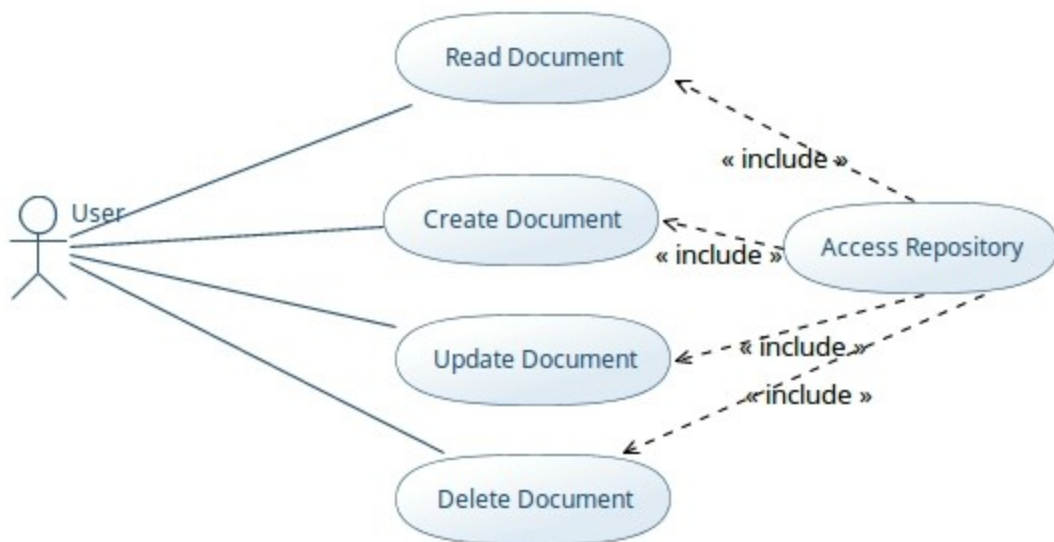
ii. Afegir, modificar o esborrar sistemes d'emmagatzemament, vincular usuaris i dipòsits i els seus drets, i modificar els drets dels usuaris sobre els dipòsits



iii. Gestionar els ports d'accés al sistema, gestió dels mime-types associats a continguts, gestió dels intèrprets dels continguts, modificar els tipus de continguts.



iv. Lectura, inserció, modificació o esborrat de continguts.



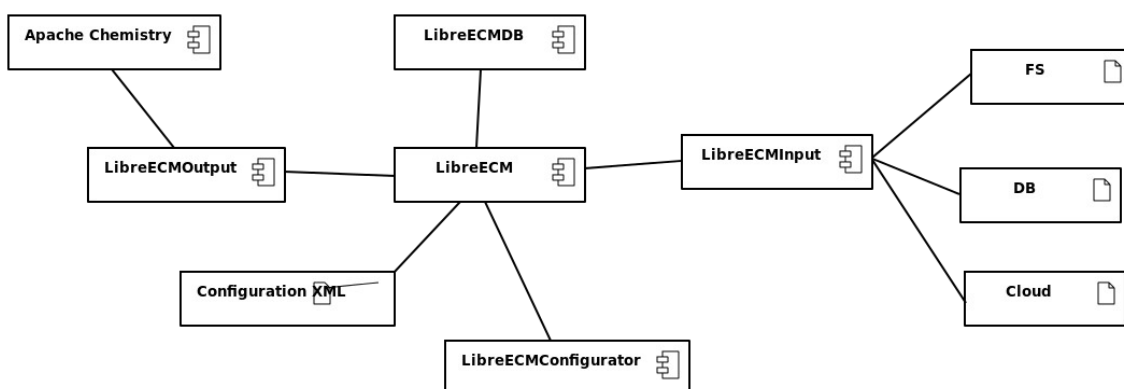
6 Disseny

6.1 Arquitectura

L'arquitectura emprada en un sistema de components. Cada component s'encarrega d'una tasca de definida, de manera que qualsevol canvi en els requisits del sistema, actualització o modificació, només influeix en el component concret, o té poca repercussió.

Les tasques assignades a cada component son les següents:

- LibreECMCommonAPI: definició dels principals objectes comuns a tota la implementació. A més defineix eines complementàries també d'ús a tot el sistema. Per acabar, gestiona l'arxiu de configuració XML del qual llegeixen tots els components.
- LibreECM BBDD: gestiona la persistència de les dades dels dipòsits (saber quins dipòsits es troben configurats i el seu estat) i dels usuaris i els seus rols.
- LibreECM Main: Hauria de ser el component que implementés la seguretat i fes les comprovacions de lògica de negoci.
- LibreECM Input. Aquest component gestiona el format de persistència dels continguts dels dipòsits.
- LibreECM OpenCMIS. Aquest component implementa l'accés dels client pel protocol CMIS. Utilitza, per tant les llibreries Chemistry OpenCMIS, i bàsicament fa una conversió dels objectes del format de LibreECM a OpenCMIS.



Les tecnologies emprades per aquests components son les següents:

- Tot el sistema empra Maven com a eina de gestió de les dependències de les llibreries, la compilació i el desplegament.
- Framework Spring. Es una àmplia col·lecció de llibreries i utilitats que donen molts tipus de serveis. Concretament s'han emprat:
 - Spring-boot: implementa un sistema de desplegament que permet llençar un component com si es tractés d'una aplicació Java ordinària. **Aquest es l'únic cas** on el port definit en la lectura de configuració (XML) s'aplica, ja que en un desplegament en Tomcat o Wildfly, aquest s'ignora i s'utilitza el port del contenidor.
 - Spring-web: per la comunicació entre components s'empren Servlet, que son consumits per altres components. La tecnologia d'intercanvi es Restfull amb format JSON.
 - Spring-security: encara que no implementat, està preparat el sistema per una senzilla gestió de la seguretat gracies a Spring-security.
 - Spring-JPA: per la gestió de la persistència.
- El servidor d'aplicacions web testejat es Tomcat en la versió 8, i no anteriors, ja que s'empra Servlet en versió 3, que aquest suporta. No hi hauria d'haver problema per emprar altres productes.
- Java 8 s'empra en diversos punts de la codificació, encara que no estrictament necessaris (sempre hi ha maneres de fer el mateix amb un codi més compatible), la simplificació i rapidesa que aquesta màquina virtual aporta en la codificació, fa que sigui el requisit mínim.
- PostgreSQL s'empra com a sistema predefinit de persistència.

7 Implementació

7.1 Creació i compilació de projecte OpenCMIS

Es necessari emprar Maven (versió 3) per compilar i instal·lar el projecte.

OpenCMIS.

S'han emprat els manuals de <http://chemistry.apache.org/java/how-to/how-to-build.html>

Primer utilitzem una aplicació SVN per descarregar la darrera versió de OpenCMIS, per exemple, en sistemes Linux emprem la comanda:

```
mkdir OpenCMIS; cd OpenCMIS
```

```
svn export https://svn.apache.org/repos/asf/chemistry/opencmis/trunk
```

```
cd trunk
```

```
mvn install
```

```
cd ..
```

Creem la configuració inicial:

```
mvn archetype:generate -DgroupId=org.libreecm -DartifactId=libreECMSrv  
-DpackageName=local.libreecm -Dversion=0.1
```

S'ha de filtrar per webapp o maven per obtenir un projecte maven-archetype-webapp

Amb el projecte ja creat, eliminem la carpeta src/main/webapp per poder crear una carpeta amb un servlet.

```
cd libreECMSrv
```

Creem el projecte per importar-lo a Eclipse:

```
mvn eclipse:eclipse
```

L'importem a Eclipse com a projecte ja existent.

7.2 Creació de components

Els components es basen en el Framework Spring i es poden desplegar tant en un contenidor com executar-se directament per Tomcat integrat gracies a Spring-boot.

Els components empraran la següent tecnologia de Spring:

- Spring Boot Starter Web: configura fàcilment serveis Web, amb intercanvis d'objectes POJO en format JSON.
- Spring Boot Starter Actuator: ens proveirà d'accés per controlar el sistema des de ports addicional i fins i tot per telnet.
- Spring Security: configura els accessos al sistema mitjançant ACL, l'accés segur al web de configuració, el transport de la informació i el sistema d'identificació d'usuaris.

Per crear cada component, creem un nou projecte important les bases de Spring, seguint les instruccions de <http://spring.io/guides/gs/rest-service/>:

Descarreguem el codi base d'un projecte REST:

git clone <https://github.com/spring-guides/gs-rest-service.git>

Apliquem el sistema Spring Boot Starter Actuator, que hereta de:

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.3.3.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
```

Previament, hem d'haver compilat i importat el jar de LibreECMCommonAPI (mvn install).

Les llibreries genèriques per als components son:

```
<dependency>
  <groupId>org.LibreECM</groupId>
  <artifactId>LibreECMCommonAPI</artifactId>
  <version>0.0.1-SNAPSHOT</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring.version}</version>
</dependency>
```

Afegim a l'arxiu pom.xml, dins la secció <dependencies>, per a que maven descarregui les llibreries: security, session, starter-web, :

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.session</groupId>
  <artifactId>spring-session</artifactId>
</dependency>
```

Per la configuració llegida o escrita en fitxers XML que fa cada component i per la gestió dels errors i excepcions, necessitem:

```
<!-- Jackson JSON Mapper -->
<dependency>
  <groupId>org.codehaus.jackson</groupId>
  <artifactId>jackson-mapper-asl</artifactId>
  <version>1.9.10</version>
</dependency>

<!-- spring oxm -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-oxm</artifactId>
  <version>${spring.version}</version>
</dependency>

<!-- Uses Castor for XML -->
<dependency>
  <groupId>org.codehaus.castor</groupId>
  <artifactId>castor</artifactId>
  <version>1.2</version>
</dependency>

<!-- Castor need this -->
<dependency>
  <groupId>xerces</groupId>
  <artifactId>xercesImpl</artifactId>
  <version>2.8.1</version>
</dependency>

<!-- Spring hateoas -->
<dependency>
  <groupId>org.springframework.hateoas</groupId>
  <artifactId>spring-hateoas</artifactId>
  <version>0.7.0.RELEASE</version>
</dependency>
```

I dins de <repositories> afegim d'on obtenir les llibreries de seguretat:

```
<repository>
  <id>spring-milestones</id>
  <name>Spring Milestones</name>
  <url>https://repo.spring.io/libs-milestone</url>
  <snapshots>
    <enabled>false</enabled>
  </snapshots>
</repository>
```

En el cas del component de BBDD, hem d'afegir el driver cap a la base de dades, el podem afegir al arxiu pom.xml, només pels components que accedeixin a BBDD (LibreECMDB):

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
  <groupId>postgresql</groupId>
  <artifactId>postgresql</artifactId>
</dependency>
```

Afegim a l'arxiu application.properties a la carpeta src/main/resources/

```
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.generate-ddl=true
spring.datasource.url=jdbc:postgresql://localhost:5432/LibreECMDB
spring.datasource.username=postgres
spring.datasource.password=postgres
```

Fet això creem a servidor una base de dades LibreECMDB

Creem l'estructura de carpetes:

```
mkdir -p src/main/java/org/LibreECM
```

Aquí ja podem crear les nostres classes, segons els tipus següents:

- Clases POJO: son classes simples Java, que seran les que s'intercanviaran el servidor i el consumidor de recursos, amb format JSON. Es composaran de camps, constructors, i accessors.
- Controladors de recursos: son classes que gestionen les peticions HTTP. Tenen la anotació `@RestController`. Poden respondre a diversos mapatges gracies a la anotació `@RequestMapping("/LibreECM")` seguides d'una classe que retorna un objecte POJO. Com a paràmetres de la funció, es passen els paràmetres que suporta la crida HTTP:

```
public Greeting greeting (@RequestParam(value="name",
defaultValue="World") String name) {...}
```

 - `@RestController` es un addiment de les anotacions `@Controller` i `@ResponseBody` juntes.
 - Si fos necessari establir un tipus de crida GET, PUT o POST, es pot fer `@RequestMapping(method=GET)`
 - Els paràmetres de la crida HTTP son opcionals per defecte, això es pot variar afegint `required=true` al `@RequestParam`
- Classe d'aplicació: la classe on s'inicia l'acció. Una classe amb la anotació `@SpringBootApplication` podrà ser executada directament pel

servidor Tomcat sense cap gestor de desplegament (Tomcat/WildFly).

```
public static void main(String[] args)
{ SpringApplication.run(Application.class, args); }
```

- @SpringBootApplication afegeix en una sola les anotacions @Configuration, @EnableAutoConfiguration i @ComponentScan.
- Els consumidors, son classes que es connecten amb els controladors i utilitzen els serveis que aquests donen.
- No es necessari definir cap web.xml

Ja podem executar a Tomcat el servei, gracies a Maven: mvn spring-boot:run

Per poder desplegar gracies a Mavel el servei des del contenidor WildFly, necessitem crear un arxiu war:

- Hem d'afegir o modificar l'arxiu pom.xml amb:
 - Empaquetat com a war: <packaging>war</packaging>
 - Hem de marcar les dependències del projecte amb Spring com a provided: <scope>provided</scope>. Així el arxiu war tindrà afegides les llibreries de les que depèn el projecte.
 - Hem d'afegir com a plugin (segons <https://www.mkyong.com/maven/maven-deploy-web-application-to-wildfly/>):

```
<plugin>
  <groupId>org.wildfly.plugins</groupId>
  <artifactId>wildfly-maven-plugin</artifactId>
  <version>1.1.0.Alpha5</version>
  <configuration>
    <hostname>127.0.0.1</hostname>
    <port>9990</port>
    <username>george</username>
    <name>LibreECM.war</name>
  </configuration>
</plugin>
```

- Hem de variar la classe on trobem el mètode main:
 - La classe ha d'extendre SpringBootServletInitializer: public class Application extends SpringBootServletInitializer

- Hi ha d'haver un mètode configure sobreescrit configure: `@Override`
`protected SpringApplicationBuilder`
`configure(SpringApplicationBuilder application) { return`
`application.sources(Application.class); }`
- Creem el arxiu war amb la instrucció `wildfly:deploy`, i ja es desplega el servei al contenidor.

8 Conclusions i estat de l'art

El present projecte no ha pogut acabar de ser implementat.

L'elecció dels sistemes Maven i Spring, han implicat una descoberta per mi, però han implicat una necessitat d'aprenentatge molt forta, ja que son eines que no havia emprat mai.

Els principals punts de bloqueig que han suposat una càrrega insalvable d'inversió de temps, han estat:

- La voluntat de no re-utilitzar les classes que ofereix OpenCMIS, més enllà del component LibreECMOpenCMIS, encara que s'empen de manera residual a LibreECMDDBB, ha implicat la gran part de cost de temps invertit en desenvolupar tot un sistema complet de dipòsits i objectes.
- La conversió i l'alliberació de bugs en la conversió a objectes OpenCMIS, ha estat tota una odissea, ja que errors en la definició de les classes internes, i la necessitat d'una comprensió profunda del funcionament de CMIS.
- La comunicació entre components, la conversió correcta dels objectes en JSON gracies a la llibreria Jackson, i la lectura dels fluxes de dades, ja sigui de manera completa o per segments.

L'avaluació però a nivell d'aprenentatge ha estat brutal, ja que ara la meva capacitat de comprensió de l'estructura de comunicació RestFull, de les conversions d'objectes amb JSON, la utilització del framework Spring (una gran eina), son aprenentatges molt grans.

8.1 Estat de l'art

Les capacitats actuals de LibreECM son bàsicament de lectura:

- Lectura de documents d'un dipòsit de tipus FSRepository (emmagatzemats en arxius en una carpeta accessible).
- Es pot accedir amb els bindings Atom, Web Services o Browser (URLs).
- Com a client es pot emprar Workbench, que es l'eina de testing oferta per OpenCMIS. Els intents d'accés per LibreOffice han fracassat (bug).
- Es pot navegar per les carpetes.
- Es llegeixen correctament els continguts de carpetes.
- S'han testat els tipus Document, Folder, Item correctament.

- Es pot llegir un flux de dades (continguts dels Document) ja siguin menors de 4mb (es llegeixen completament) o superiors (es llegeixen en fragments de 4mb cadascun i s'emmagatzemen en una carpeta temporal)

Finalment, aquest es l'estat en que s'entreguen cadascun dels serveis definits:

- Serveis de dipòsits
 - Afegir dipòsit (100% mitjançant arxiu de configuració)
 - Eliminar dipòsit (0%)
 - Obtindre dipòsits (100%)
 - Obtindre informació de dipòsits (100%)
 - Obtindre tipus d'objectes (100%)
 - Obtindre tipus heredats d'objectes (100%)
 - Obtindre definicions de tipus (100%)
- Serveis de navegació
 - Obtindre objectes fills. (100%)
 - Obtindre objectes continguts a carpeta (100%).
 - Obtindre arbre de carpetes. (100%)
 - Obtindre carpeta pare. (100%)
 - Obtindre objectes pares. (100%)
- Serveis d'objectes.
 - Crear document (0%)
 - Crear document a partir de fluxe de dades. (0%)
 - Crear carpeta (0%)
 - Crear relació (0%)
 - Crear política (0%)
 - Obtindre accions permesses (75%)
 - Obtindre objecte (100%)
 - Obtindre propietats (100%)
 - Obtindre objecte per ruta (0%)
 - Obtindre fluxe de dades d'un objecte (100%)

- Obtindre representacions (thumbnails) (0%)
- Actualitzar propietats (0%)
- Moure objecte (0%)
- Esborrar objecte (0%)
- Esborrar arbre de carpetes (0%)
- Assignar fluxe de dades (0%)
- Esborrar fluxe de dades (0%)
- Serveis de versionament
 - Confirmació versió (check-out) (0%)
 - Cancelar versió (0%)
 - Iniciar versió (check-in) (0%)
 - Obtindre darrera versió d'objecte (0%)
 - Obtindre propietats de darrera versió d'objecte (0%)
 - Obtindre totes les versions (0%)
- Serveis de carpetes
 - Afegir objecte a carpeta (0%)
 - Eliminar objecte de carpeta (0%)
- Serveis de descoberta
 - Petició (query) (0%)
 - Obtindre canvis en objecte (0%)
- Serveis de relacions
 - Obtindre relacions d'objecte (100%)
- Serveis de polítiques
 - Aplicar política (0%)
 - Eliminar política (0%)
 - Obtindre polítiques aplicades (100%)
- Serveis de control d'accés (ACL)
 - Obtindre controls (ACL) (100%)

- Aplicar controls (0%)
- Serveis de configuració
 - Definir port de component (100%)
 - Definir URL de component (100%)
 - Definir carpeta temporal (100%)
- Serveis d'usuaris
 - Afegir usuari (0%)
 - Eliminar usuari (0%)
 - Modificar rol usuari (0%)
 - Afegir rol (0%)
 - Eliminar rol (0%)
- Serveis de tipus
 - Afegir tipus de contingut (0%)
 - Eliminar tipus de contingut (0%)

9 Glossari de termes

El glossari reflexa els principals conceptes aplicables al present projecte. Es defineix el terme en català i entre parèntesi l'equivalen en anglès tal com s'empra al codi font.

- Contingut (LibreECMDataObject): conté informació i meta-informació d'interès de ser emmagatzemada. Es la unitat general bàsica a partir de la qual hereten tots els tipus de informació a emmagatzemar.
- Dipòsit (Repository): ubicació on s'emmagatzema el contingut. Es poden definir diversos tipus i diverses realitzacions del mateix. En la present versió només s'implementa la persistència en format fitxer.
- Document (Document): format bàsic de contingut, associat a un fluxe de dades. Els documents son el tipus més simple de contingut amb informació.
- Item (Item): format bàsic de contingut, sense fluxe de dades associat, per tant, només defineixen meta-dades.
- Carpeta (Folder): format bàsic de jerarquia de contingut. Una carpeta pot contindre qualsevol tipus de contingut que ella mateixa defineixi. La utilitat de les carpetes es permetre una organització lògica de tipus jeràrquic que simplifiqui la cerca i permeti certa organització.
- Relació (Relationship): format bàsic de vincle entre continguts. Una relació vincula dos continguts de manera unidireccional.
- Política (Policy): format bàsic de llançament d'esdeveniments com a resposta a accions. Cada acció que un usuari realitza, pot llençar un esdeveniment definible a qualsevol nivell que realitza tasques concretes.
- Intèrpret de continguts (Rendition): format de previsualització d'una carpeta. Es compona d'un fluxe de dades que permet diferenciar carpetes. Por estar referit al contingut de la mateixa o definir-se un tipus concret per una carpeta concreta.

10 Bibliografia

Bibliografia

- CMIS and Apache Chemistry In Action
 - Florian Müller, Jay Brown, Jeff Potts
 - Manning Publications
 - 2013
 - ISBN 9781617291159
- Alfresco CMIS
 - Martin Bergljung
 - Pack Publising
 - 2014
 - ISBN 9781782163527
- Spring in Action
 - Craig Walls
 - Manning Publications
 - 2011
 - ISBN 9781935182351

11 Manual d'usuari

11.1 Compilació

Per la compilació es necessari tenir instal·lat Maven i JDK en la versió 8.

La compilació es necessari executar-la primer en el component LibreECMCommonAPI, ja que es tracta d'una llibreria comú a la resta de components:

Maven resoldrà la resta de dependències:

```
mvn clean package install
```

I afegir al dipòsit de Maven:

```
mvn install:install-file -Dfile=LibreECMCommonAPI-0.1.0.jar  
-DgroupId=org.Librecm -DartifactId=LibreECMCommonApi  
-Dversion=0.1.0 -Dpackaging=jar
```

La resta de components poder ser compilats i empaquetats:

```
mvn clean package
```

Per l'execució es necessari Apache Tomcat en la versió 8.

11.2 Execució

FITXA DEL TREBALL FINAL

Títol del treball:	<i>LibreECM: Aplicació de gestió de recursos documentals</i>
Nom de l'autor:	<i>Jordi Pérez Bayerri</i>
Nom del consultor/a:	<i>Albert Grau Perisé</i>
Nom del PRA:	<i>Santi Caballe Llobet</i>
Data de lliurament (mm/aaaa):	<i>06/2016</i>
Titulació o programa:	<i>Grau en Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Java EE</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Dipòsit, document, ECM</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>L'aplicació LibreECM és un gestor de continguts sota el paraigües ECM (Enterprise Content Manager).</p> <p>Es tracta d'un administrador de continguts en format fitxer o de tipus lògic.</p> <p>Existeix un o més dipòsits que contenen la informació, que l'aplicació gestiona. Aquesta s'encarrega de la seva administració, així com del control d'accés a aquests.</p> <p>El projecte es compon de 4 components:</p> <ul style="list-style-type: none">• Main unifica la comunicació dels components en un sol punt, i serveix com a punt de referència per l'accés al sistema.• DDBB manté la persistència d'usuaris i dipòsits.• Input interactua amb els continguts, ja sigui en format fitxer, base de dades, text, etc.• OpenCMIS és una interfície sota l'estàndard CMIS de OASIS, per l'accés a dipòsits de continguts. Per això s'utilitza la solució OpenCMIS de Chemistry, sota el paraigües d'Apache. <p>La interacció entre els components es realitza mitjançant RestFull (JSON), gracies al framework Spring.</p> <p>Els serveis que dona LibreECM son:</p> <ul style="list-style-type: none">• Serveis de dipòsits.• Serveis de navegació: llegir carpetes, accedir a carpetes.	

- Serveis d'objectes: meta-propietats, lectura i descàrrega.
- Serveis de descoberta.

El resultat és una aplicació que permet accedir a la informació continguda en algun tipus de persistència remotament. Es fa mitjançant un estàndard definit i consolidat, que permet interactuar amb multitud de serveis ja existents.

Abstract (in English, 250 words or less):

LibreECM is a content manager belonging to ECM (Enterprise Content Manager) systems.

It's a content manager for file systems or for logic systems.

Exists one or more repositories containing information, that is managed by the application. LibreECM job is to administrate this, and to do control checking.

The project is composed by 4 components:

- Main unifies communication between components at one point, and serves as a reference point for system access.

- DDBB persists information about users and repositories.
- Input interacts with contents, with any format, as could be files, data bases, text, etc.
- OpenCMIS is an interface under the CMIS of OASIS protocol, for access repositories contents. By this reason, OpenCMIS from Apache project is used.

Interaction between components uses RestFull (JSON) technology, thanks to Spring framework.

Services provided by LibreECM are:

- Repository services
- Navigation services: read folder, folder access.
- Object services: meta-properties, read and download.
- Discovery services.

As a result, the application could access any information at some remote persistence. It's done with a defined and consolidated standard, that allows interaction with many services now existent.



