DOCUMENTACIÓ
PRIMERA ENTREGA

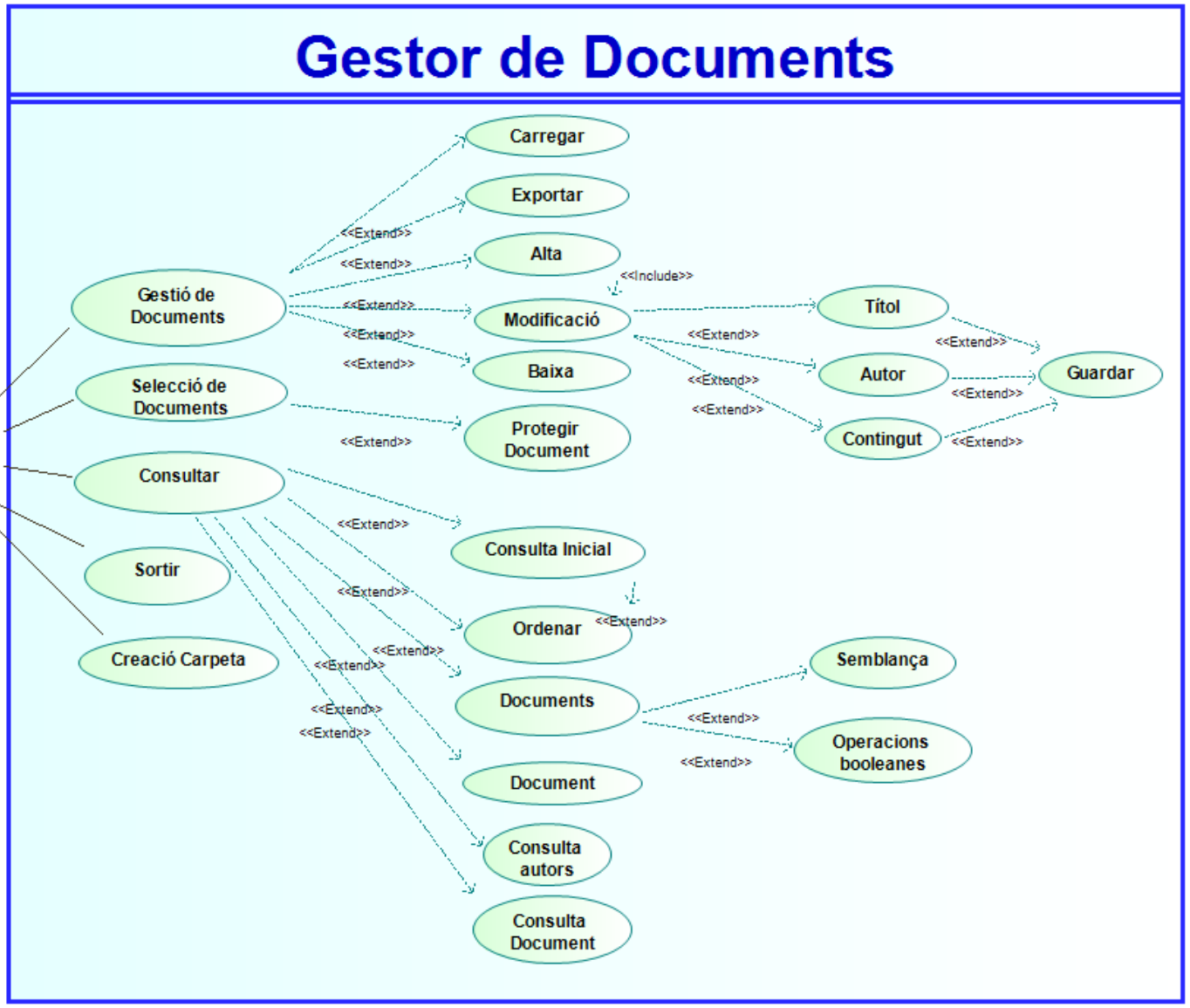
Júlia Alice Amenós Dien
Víctor Mena Doz
Marc Navarro Acosta
Jordi Soley Masats

ÍNDEX

ESTRUCTURA DEL DOCUMENT

Aquest document està destinat a presentar i argumentar les eleccions realitzades pel que fa al disseny i estructura del nostre projecte. S'inclou el diagrama dels Casos d'Ús i una descripció de les estructures de dades i algorismes que implementen la lògica de la capa domini.

DIAGRAMA CASOS D'ÚS


Gestor de Documents
Usuari
Gestió de Documents
Selecció de Documents
Consultar
Sortir
Creació Carpeta
Carregar
Exportar
Alta
Modificació
Baixa
Protegir Document
Consulta Inicial
Ordenar
Documents
Document
Consulta autors
Consulta Document
Títol
Autor
Contingut
Guardar
Semblança
Operacions booleanes
<<Extend>>
<<Include>>

EXPLICACIÓ CASOS D'ÚS

CASOS D'ÚS DEL NOSTRE SISTEMA

CASOS D'ÚS

- Carregar document (XML o text pla)
  - **Actors Primaris:** Usuari
  - **Precondició:** No existeix cap document amb el mateix identificador *(títol, autor)*.
  - **Activació**: L'usuari prem un botó per registrar un document.
  - **Escenari principal d'èxit:**
    1. L'usuari especifica el tipus de format.
    2. L'usuari tria el document a carregar.
    3. El sistema requereix confirmació de l'operació
    4. El sistema el carrega i li retorna un feedback.
  - **Escenaris alternatius**:
    - L'usuari busca el document que vol carregar però s'equivoca al escollir el document i a la confirmació cancel·la l'operació.

- Exportar document (XML o text pla)
  - **Actors Primaris:** Usuari
  - **Precondició:** Existeix el document que es vol exportar.
  - **Activació:** L'Usuari prem un botó per exportar un document.
  - **Escenari principal d'èxit:**
    1. L'usuari tria el document a exportar
    2. El sistema requereix confirmació de l'operació
    3. El sistema exporta el document i li retorna un feedback
  - **Escenaris alternatius:**
    - Hi ha hagut un error d'exportació del document.

- Alta Document
  - **Actors Primaris:** Usuari
  - **Precondició:** El document no existeix al sistema.
  - **Activació:** L'usuari prem un botó per registrar un document.
  - **Escenari principal d'èxit:**
    1. S'obre una vista on es pot redactar un nou document.
    2. L'usuari redacta el document.
    3. L'usuari prem el botó de guardar el document.
    4. Apareix un avís per demanar confirmació (especificació del títol).
    5. Prem botó "Acceptar".
  - **Escenaris alternatius:**
    - L'usuari decideix no guardar el document.
    - L'usuari no ha indicat un dels tres camps requerits.
    - L'usuari decideix en la confirmació seguir editant el document.

- Baixa Document
  - **Actors Primaris:** Usuari
  - **Precondició:** El document identificat per *(títol, autor)* existeix al sistema.
  - **Activació:** L'usuari prem un botó per eliminar el document escollit.
  - **Escenari principal d'èxit:**
    1. Apareix un avís per assegurar si l'usuari vol eliminar.
    2. L'usuari accepta i el document s'elimina.
  - **Escenaris alternatius:**
    - L'usuari decideix cancel·lar l'eliminació i el document queda intacte.

- Modificar Document
  - **Actors Primaris:** Usuari
  - **Precondició**: El document està escollit per l'usuari a través d'un dels tipus de consulta.
  - **Activació**: L'usuari prem un botó per modificar algun atribut del document (podria ser títol, autor o el propi contingut).
  - **Escenari principal d'èxit:**
    1. S'obre una vista que permet a l'usuari fer modificacions sobre el document.
    2. L'Usuari fa els canvis adients i els confirma.
    3. El document queda correctament modificat.
  - **Escenaris alternatius:**
    - L'Usuari modifica el document però no ho confirma i al document no se li aplica cap canvi.
    - L'Usuari intenta canviar l'autor i/o títol coincidint amb els atributs d'un document ja existent.

- Consulta
  - **Actors Primaris:** Usuari
  - **Precondició:** -
  - **Activació:** L'usuari prem un botó per llistar totes les possibles consultes
  - **Escenari principal d'èxit:**
    1. Apareixen per pantalla totes les cinc consultes de documents possibles.
  - **Escenaris alternatius:**
    - L'usuari decideix cancel·lar la consulta.

- Consulta títols autor
  - **Actors Primaris**: Usuari
  - **Precondició**: -
  - **Activació**: L'Usuari introdueix un Autor i prem un botó per buscar els seus Documents.
  - **Escenari principal d'èxit:**
    1. Apareix un llistat de Documents que pertanyen a l'autor sol·licitat.
  - **Escenaris alternatius:**
    - La cerca no obté resultats i apareix una vista buida amb l'opció de tornar al menú. (return missatge d'error).

- Consulta autors prefix
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació**: L'Usuari introdueix un prefix i prem un botó per veure els Autors que comencen amb aquell prefix.
  - **Escenari principal d'èxit:**
    1. Es mostren tots els Autors que comencen pel prefix sol·licitat.
  - **Escenaris alternatius:**
    - El resultat de la cerca és buit perquè no hi ha cap Autor que comenci pel prefix sol·licitat i apareix una vista buida amb possibilitat de tornar al menú.

- Consulta document concret
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació:** L'Usuari introdueix un títol i un autor i prem un botó per veure aquell concret Document.
  - **Escenari principal d'èxit:**
    1. Es mostra el document en un llistat que indica els seus atributs identificadors (títol i autor).
  - **Escenaris alternatius:**
    - L'Usuari introdueix un títol i un autor que no tenen cap coincidència en el sistema i es mostra una vista d'error.

- Consulta documents inicial
  - **Actors Primaris:** Usuari
  - **Precondició:** -
  - **Activació:** L'Usuari obre el programa.
  - **Escenari principal d'èxit:**
    1. Es mostren tots els documents i subcarpetes de la carpeta pare de l'aplicació.

- Llistar documents per semblança
  - **Actors Primaris:** Usuari
  - **Precondició:** L'Usuari ha seleccionat un document.
  - **Activació:** En la vista del Document seleccionat prem un botó que activa la funció de buscar semblants.
  - **Escenari principal d'èxit:**
    1. Indica un enter positiu *k* i ho envia al sistema.
    2. El sistema fa una cerca i obté els *k* documents més semblants.
    3. S'obre una vista que mostra un llistat amb els *k* documents que el sistema ha seleccionat.
  - **Escenaris alternatius:**
    - L'Usuari posa un enter negatiu per *k*.
    - L'Usuari posa una *k* més gran que el nombre de documents que hi ha a la nostra base.

- Consulta documents per operacions booleanes
  - **Actors Primaris:** Usuari
  - **Precondició:** -
  - **Activació:** L'Usuari introdueix unes expressions booleanes en una barra de cerca.
  - **Escenari principal d'èxit:**
    1. L'expressió booleana té sentit lògic.
    2. Es fa una cerca filtrant en funció de les condicions de la búsqueda.
    3. Apareixen tots els documents que compleixen l'operació booleana.
  - **Escenaris alternatius:**
    - L'expressió booleana no és correcta i es llença una vista d'error.
    - La consulta no té resultats i es mostra una vista informant-ho i amb possibilitat de repetir-la.

- Consulta documents per query
  - **Actors Primaris**: Usuari
  - **Precondició:** -
  - **Activació**: L'Usuari indica en una barra de cerca p paraules i un enter *k* i posteriorment prem un botó d'acceptar.
  - **Escenari principal d'èxit:**
    1. El sistema fa una cerca i obté els *k* documents més relacionats amb les paraules que ha rebut.
    2. S'obre una vista que conté els *k* resultats.
  - **Escenaris alternatius:**
    - Es troben menys documents dels k demanats (podria inclús no trobar-ne cap).
    - L'enter *k* introduït per l'usuari és negatiu.

- Selecció documents
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació**: Usuari indica que vol seleccionar documents.
  - **Escenari principal d'èxit:**
    1. Un o més documents indicat per l'usuari queden seleccionats.

- Modificar contingut
  - **Actors Primaris**: Usuari
  - **Precondició**: -
  - **Activació**: Usuari indica que vol seleccionar documents.
  - **Escenari principal d'èxit:**
    1. L'usuari pot modificar el contingut del document.
    2. L'usuari guarda els canvis realitzats.
  - **Escenaris alternatius:**
    - L'usuari decideix no guardar els canvis.

- Modificar títol
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació**: Usuari indica que vol modificar el títol del document.
  - **Escenari principal d'èxit:**
    1. El sistema canvia el títol del document i el desa.
  - **Escenaris alternatius:**
    - Ja existeix un document amb el mateix autor i títol que s'intenten posar.

- Modificar autor
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació**: Usuari indica que vol modificar l'autor del document.
  - **Escenari principal d'èxit:**
    1. El sistema canvia l'autor del document.
    2. En el cas de que l'autor no existís, llavors s'hauria de crear una nova instància d'autor.
  - **Escenaris alternatius:**
    - Ja existeix un document amb el mateix títol i l'autor que s'intenta posar.

- Guardar
  - **Actors Primaris:** Usuari
  - **Precondició:** -
  - **Activació:** Usuari indica que vol guardar el document seleccionat (s'ha de trobar obert).
  - **Escenari principal d'èxit:**
    1. Sistema guarda el document en l'estat actual.

- Protegir document
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació**: Usuari indica que vol protegir el document seleccionat.
  - **Escenari principal d'èxit:**
    1. Usuari proporciona una contrasenya i el document queda bloquejat.

- Sortir
  - **Actors Primaris**: Usuari
  - **Precondició**: -
  - **Activació**: Usuari indica que vol sortir de l'aplicació.
  - **Escenari principal d'èxit**:
    1. El sistema tanca l'aplicació.

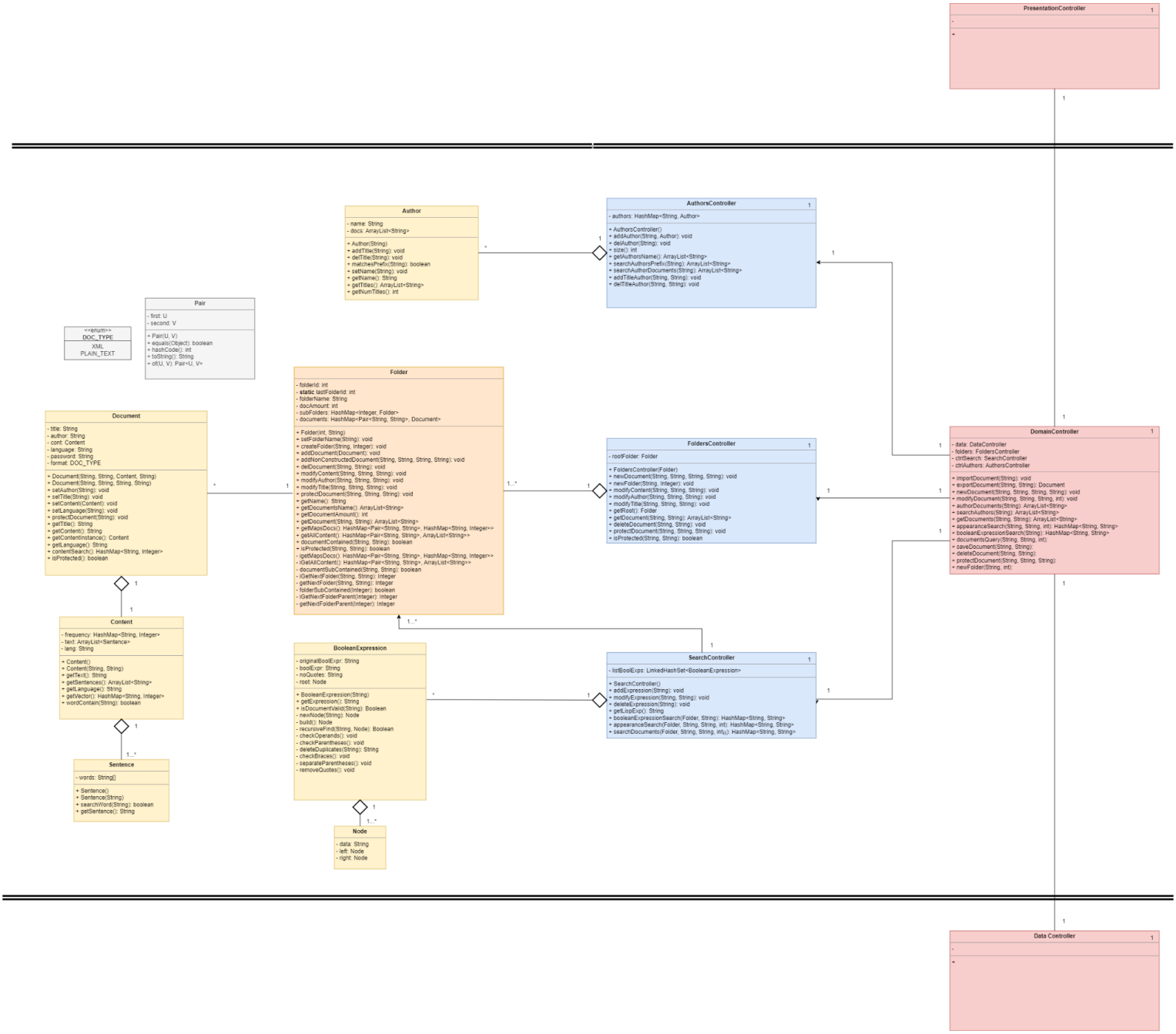
- Creació carpetes
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació:** Usuari indica que vol crear una carpeta.
  - **Escenari principal d'èxit:**
    1. Es crea una nova carpeta amb el nom indicat per l'usuari.

- Gestió de documents
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació**: L'usuari indica que vol gestionar els documents
  - **Escenari principal d'èxit:**
    1. El sistema llista totes les gestions possibles.

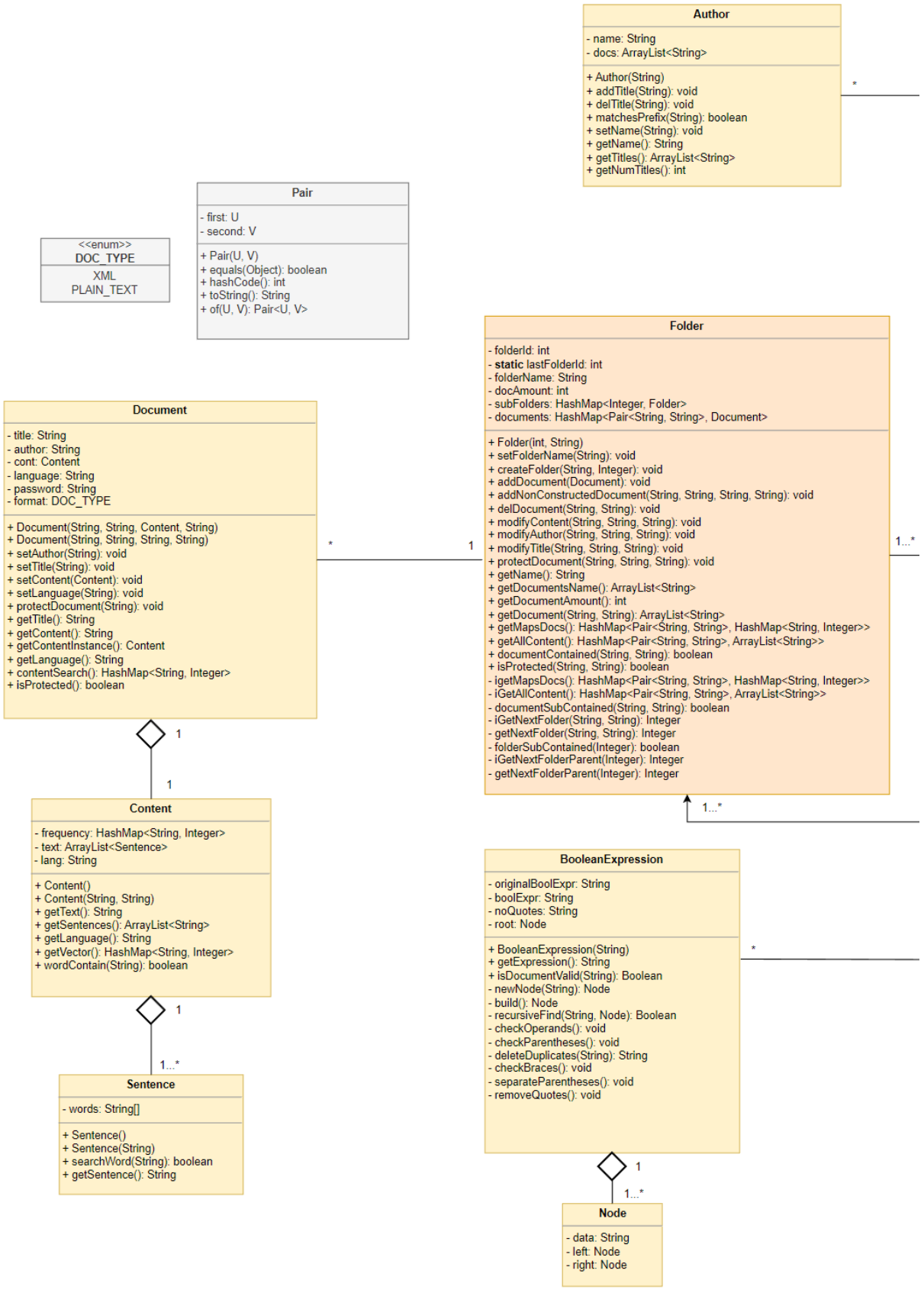
- Ordenar
  - **Actors Primaris:** Usuari
  - **Precondició**: -
  - **Activació**: L'usuari indica que canviar l'ordre de llistat dels documents.
  - **Escenari principal d'èxit:**
    1. Els documents queden reordenats segons la selecció de l'usuari.

DIAGRAMA UML

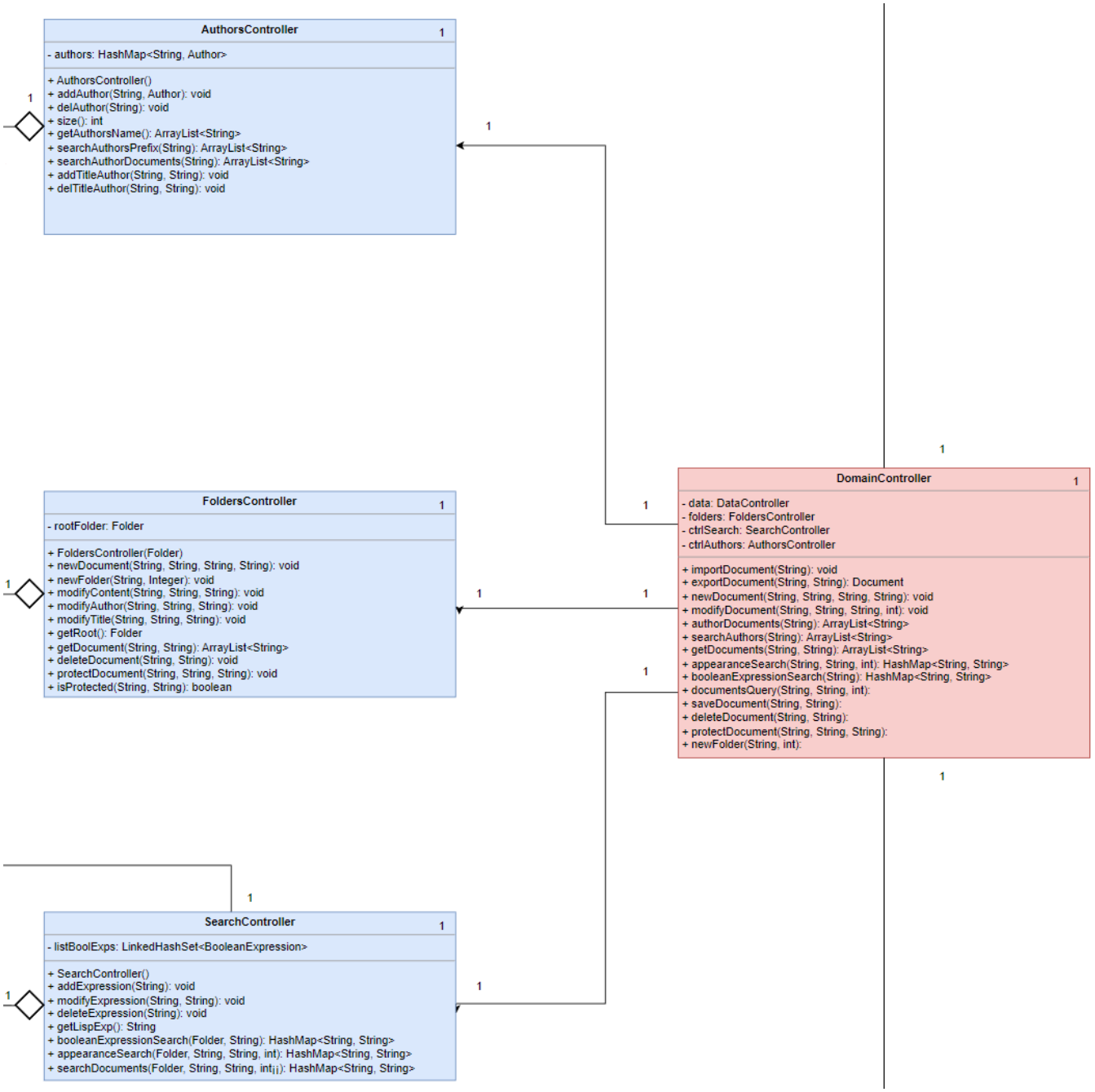
UML GLOBAL SCHEMA



UML DOMAIN CLASSES



UML DOMAIN CONTROLLERS



EXPLICACIÓ DEL DIAGRAMA UML

Controladors

En la nostra arquitectura en tres capes, els Controladors són qui es relacionen entre les diferents capes. Per aquesta raó el nostre diagrama UML té 3 Controladors principals (marcats en roig), el *DomainController, DataController* i *PresentationController*.

Per tal de gestionar els Objectes que desenvolupen la lògica del nostre Domini, el propi *DomainController* inicialitza les instàncies dels seus subControladors i els hi delega tasques:

- *SearchController*, encarregat de les cerques de Documents.
- *AuthorsController*, encarregat de la gestió dels Autors dels Documents del nostre Sistema.
- *FoldersController*, encarregat de la lògica d'organització dels Conjunts de Documents mitjançant carpetes. És el responsable d'inicialitzar la carpeta **root** que és la base del nostre sistema de Documents.

Clases

Les classes són els Objectes que gestionen la lògica del Domini del nostre Sistema.
Els conjunts de documents estan gestionats per l'Objecte carpeta(**rootFolder**), que és inicialitzat en el moment d'inici del Sistema per *FoldersController.*

La classe *Folder* gestiona les funcions dels Documents per a que tinguin coherència en el seu conjunt; és a dir, si modifiquem un Document hem de fer-ho a través de carpeta per tal que la modificació sigui efectiva sobre el Conjunt.

La classe *Document* conté operacions per fer gestions d'un Document concret, conté informació sobre el seu títol, el seu autor i una instància de la classe *Content* que representa el seu contingut.

El contingut d'un Document és representat per la classe *Content* que disposa de una classe *Sentence* que representa cada frase del contingut per fer anàlisi requerits per funcionalitats de cerques posteriors.

Els autors del Sistema es representen mitjançant la classe Autor, emmagatzemant totes les instàncies en el controlador que les crea, *AuthorsController*.

Per satisfer les funcionalitats d'algunes de les cerques utilitzem la classe *BooleanExpression* que analitza i emmagatzema operacions entrades per l'usuari per buscar Documents concrets.

DESCRIPCIÓ ESTRUCTURES DE DADES I ALGORISMES PRINCIPALS

1. Unitats bàsiques

1.1. Author

Un Author és una classe formada per un nom (String), i una llista dels títols dels seus documents (ArrayList<String>).

Les funcionalitats que té són, poder canviar-li en nom (tot i que no s'utilitzarà en aquest sistema), afegir i eliminar-li els títols dels seus corresponents documents i comprovar si el nom d'un autor comença per un prefix donat.

1.2. Content

La classe Content sorgeix de la necessitat d'haver d'emmagatzemar el contingut d'un document de forma eficient i útil després per fer les cerques necessàries. Per aquests motius Content conté una llista de Sentence (ArrayList<Sentence>) per a guardar tot el contingut eficientment, ocupant el mínim espai possible i permetent una fàcil manipulació. A més a més, també inclou un HashMap que s'utilitzarà per realitzar les búsquedes de documents més semblants. Aquesta estructura guarda les paraules més rellevants del text de document i el nombre de vegades que apareix cadascuna. Per últim, es guarda un string que indica el llenguatge del text del document guardat a content.

Les funcionalitats d'aquesta classe són sobretot funcions per obtenir informació: esbrinar l'idioma del text, rebre una estructura amb totes les sentence que componen el content, agafar el mapa de paraules rellevants dels text i per últim obtenir l'string montat de tot el text. L'única que no és així és per a trobar un conjunt de paraules dins del text.

1.3. Sentence

Una Sentence està implementada de tal forma que és un conjunt d'strings que formen una oració del text del document.

És una classe molt senzilla que únicament fa dues coses a part de la creadora: buscar un grups de paraules i dir si aquestes es troben dins del conjunt d'strings o passar la seva instància, però construida com a un sòl string.

1.4. BooleanExpression

Per a permetre la cerca d'expressions booleanes en els documents, hem implementat la classe *BooleanExpression*. Donat un *String*, primer de tot comprova que l'expressió sigui vàlida: es verifica el tancament correcte de parèntesis, claus i cometes, així com les posicions dels operands. A més a més, hem aprofitat per depurar els conjunts de paraules i convertir-los en una concatenació de paraules amb l'operador lògic '&'. Hem decidit que els mots que tenen significat gramatical per l'expressió, com "!", "()", "&", "{", volem que també

es puguin cercar. Per tal d'evitar confondre'ls amb els operadors, especificarem en l'interfície que s'incloguin entre cometes.

En segon lloc, s'ha representat l'expressió resultant anterior en un arbre binari, on els nodes son operadors, paraules o seqüències de paraules. Aquest té en compte l'ordre de precedència dels operadors: parèntesis, operador AND (&) i operador OR (|) (de més a menys prioritat). La negació (!) s'ha concatenat amb la paraula. Per aquesta estructura hem necessitat crear una *nested class Node* que permeti implementar l'arbre.

2. Conjunts

Les classes detallades a continuació són estructures complexes que utilitzen algunes de les unitats simples descrites anteriorment.

2.1. Document

Es considera un Document una estructura que disposa un text, que anomenem *Contingut* i és representat per la classe **Content**.
Un document té un Autor i un Títol associats que l'identifiquen únicament en el nostre Sistema.

Per aquesta raó, s'han organitzat les seves estructures internes d'acord a la dupla *<Títol, Autor>* que identifiquen un document concret.

Un Document està escrit amb un llenguatge que pot ser Català, Castellà o Anglès.

2.2. Folder

Una Carpeta un subconjunt de Documents que organitza tots els Documents del nostre Sistema.
S'identifica per un enter únic per cada carpeta i en el qual l'arrel té un 0.

Una carpeta pot disposar de Subcarpetes que comptaran amb un subconjunt de Documents i formaran un arbre a partir d'una carpeta arrel que contindrà recursivament tots els Documents del nostre Sistema.

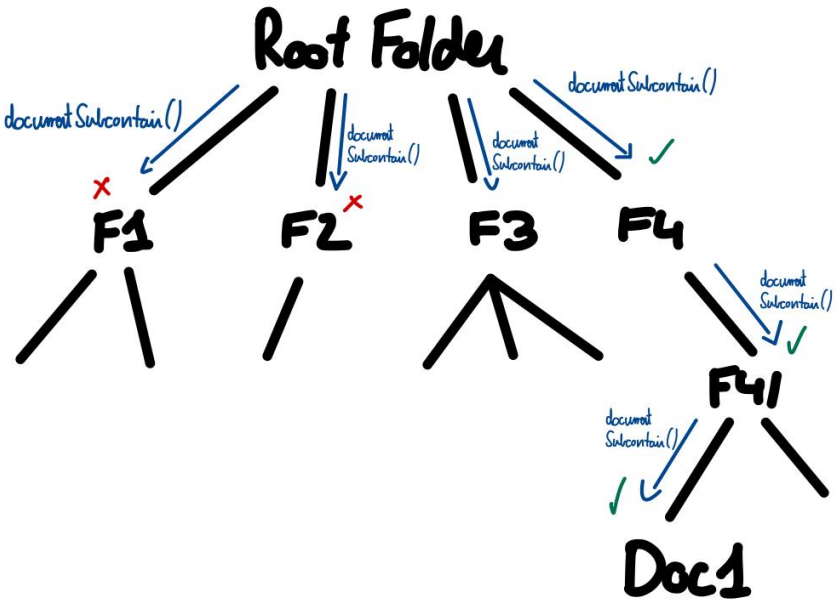
Una Carpeta conté diverses estructures:

- Un HashMap que conté tots els Documents de la Carpeta, amb una clau *<Títol, Autor>* i una instancia Document com a valor.
- Un HashMap que emmagatzema les Subcarpetes amb un enter que les identifica i una instancia Folder com a valor.

3. Algorismes

3.1. Cerca d'un Document

La cerca d'un Document es fa en la classe *Folder* examinant l'arbre i aplicant una cerca recursiva.
Les Carpetes tenen un identificador únic i un conjunt de Documents presents en cada carpeta.
En l'algorisme implementat per la búsqueda d'un Document concret, es comprova si el Document hi és a l'arrel.
Si està, el retorna directament i sinó, comença una cerca recursiva en les seves Subcarpetes fins que el troba i retorna l'enter corresponent a l'identificador de la següent carpeta que ha de visitar. Aquesta funcionalitat es veu al dibuix com *documentSubcontained()*.



3.2. Cerca per expressió booleana

La cerca per expressió booleana ha de retornar tots els documents que contenen una frase que la satisfà. La idea principal consisteix en consultar el contingut de tots els documents, separat per frases, per verificar si alguna d'elles la compleix. En el moment en que se'n troba una, acceptem el document i podem aturar de comprovar la resta de frases del document i passem al següent.

Aquest algoritme empra tres controladors: *domain, search i folder*. La primera classe estableix la connexió entre les dues altres. *Folder controller* obté tots els documents de cada carpeta, mentre que *Search controller* verifica si aquests compleixen l'expressió booleana especificada per l'usuari.

Per validar una frase donada una expressió, hem assumit que es busca si el conjunt de caràcters hi apareix en algun moment, ja sigui a principi, fi o meitat de mot. És a dir, tal i com ho hem implementat, en el següent exemple ens sortiria que la frase és vàlida:

- Expressió booleana: pro & anys
- Frase: "Vaig començar a programar als deu anys"

Finalment, per verificar les frases dels continguts, a partir de l'arrel de l'arbre binari anem comprovant recursivament si la frase conté o no els nodes fulla de l'expressió booleana. Pels nodes interns, que seran sempre un dels dos operands lògics, si es tracta d'una AND s'han de satisfer ambdós fills, mentre que pel cas de la OR, només cal que un sigui vàlid.

3.3. Cerca per semblança

La cerca per semblança consisteix en, donat un document, obtenir una llista dels documents més semblants a aquest. Per implementar-la hem hagut d'aplicar els conceptes d'espais vectorials i tf-idf.

El tf-idf és la freqüència que té una paraula en un document i la rellevància que dona en una col·lecció de documents. En poques paraules, la freqüència d'una paraula és el nombre de vegades que apareix en un document.

Per altra banda, els espais vectorial són models algebraics per a representar documents de text de manera que siguin vectors d'identificadors i així poder filtrar informació i comparar documents.

Aplicant aquests conceptes hem decidit utilitzar un mapa per cada document que guardi les paraules importants que contingui els seus respectius text. Llavors aquesta funció l'arriba una estructura que emmagatzema les keys que identifiquen als documents i els seus maps.

El que es fa a aquesta cerca és agafar el document del que volem semblants i fer un bucle per comparar-ho amb cada document de la base de dades. Per fer aquesta comparació es crea un mapa auxiliar per al document que es compara amb document que volem, de tal forma que tinguem un mapa amb la mateixa mida i contingut que el document del que volem semblants. És a dir:

Mapa del document del que volem semblants:

```
{(futbol, 1), (Messi, 1), (Barcelona, 3), (Pique, 2)}
```

Mapa del document comparat en una iteració:

```
{(Messi, 4), (Antonella, 7), (Barcelona, 1)}
```

Mapa auxiliar creat per al document comparat en una iteració:

```
{(futbol, 0), (Messi, 4), (Barcelona, 1), (Pique, 0)}
```

D'aquesta forma podrem aplicar el concepte de similaritat de cosinus per fer la comparació. Per aplicar tot això haurem d'obtenir la normal de cada mapa dels dos documents i fer un producte vectorial per cada posició fins obtenir aquest valor que ens permetrà mesurar la semblança de dos documents.

Una vegada tinguem un llistat amb totes les semblances de tots el documents amb el seleccionat muntarem un map resultat que emmagatzemi les keys dels k documents més semblants per retornar-los.

3.4. Cerca per rellevància donat un conjunt de paraules

La cerca per rellevància ha de retornar tots els documents que compleixen la premissa de tenir incloses al seu text les paraules que es donen com a paràmetre explícit. Aquests documents es retornaran en un HashMap que contindrà las keys dels documents: títol i nom de l'autor.

Per poder fer aquesta cerca el més eficient possible amb les classes que he decidit implementar nosaltres, hem decidit agafar tots el continguts de tots els documents de la nostra base de dades i fer una cerca massiva de les paraules donades a buscar. Així que he fet un bucle que agafa cada document i li busca aquest conjunt de paraules crean un mapa auxiliar que porti el recompte de l'aparició d'aquestes paraules dins de cada document.

Una vegada s'ha obtingut el mapa resultant amb els documents que incloguin totes o algunes de les paraules donades, s'organitzen de tal forma que el document més rellevant és el primer que es ficarà al mapa resultat i així successivament fins traduir d'un mapa a l'altre.

3.5. Cerca d'autors donat un prefix

La cerca d'autors donat un prefix és, donat un prefix, obtenir una llista dels noms dels autors que comencen per aquest prefix. L'implementació es basa en a partir del HashMap<String, Author> de tots els autors del sistema, recórrer els valors (les instàncies de Author) i per cada instància amb una funció es comprova si el nom d'aquest comença pel prefix donat.

Per comprovar si el nom de cada autor comença pel prefix, es recorren els 2 strings caràcter a caràcter.

En el cas que l'autor comenci pel prefix, s'afegeix a un ArrayList de Strings que un cop s'hagin comprovat tots els autors, es retornarà com a resultat. Aquest, contindrà tots els noms dels autors que comencin pel prefix.

4. Controladors

4.1. Controlador de Cerques

Aquest controlador permet gestionar les cerques que estan relacionades amb el contingut dels documents, obtinguts mitjançant el controlador de domini.

D'una banda, és l'encarregat de tractar l'historial d'expressions booleanes de l'usuari. Es podran consultar les expressions realitzades anteriorment, així com modificar-les, borrar-les o afegir-ne de noves.

D'altra banda, realitza la cerca per expressió booleana, per semblança i de documents rellevants. L'algorisme de cada una d'elles ha estat descrita en detall en els apartats anteriors.

4.2. Controlador d'Autors

El controlador d'autors gestiona totes les instàncies de la classe Author del sistema.

Les funcionalitats que té són crear i eliminar autors, calcular el nombre d'autors del sistema, buscar autors que comencen per un prefix donat, buscar els títols dels documents d'un autor en concret i afegir i eliminar títols de documents d'un autor.

Les funcions d'afegir i eliminar títols d'un autor estan implementades de tal forma que quan s'intenta afegir un títol a un autor que no existeix, aquest es crea. Aquest cas es dóna quan s'assigna un document a un autor que encara no en tenia cap, això vol dir que l'hem de donar d'alta al sistema. A més quan a un autor se li elimina l'únic document que tenia, aquest s'elimina del sistema.

4.3. Controlador de Carpetes

El controlador de carpetes és el qui inicialitza en un primer moment la instància de Carpeta (anomenada rootFolder), contra la qual es fan totes les funcions principals.

El Controlador Domini delega les funcions que requereixen documents ja establerts en el Sistema a aquest controlador, que opera contra la carpeta RootFolder.

Totes les operacions relacionades amb subCarpetes o consultar la inclusió de certs Documents en una carpeta s'utilitzen algorismes recursius com el descrit en l'apartat Búsqueda d'un Document.

4.4. Controlador Domini

El controlador de domini és la peça clau de tot el funcionament del nostre disseny de tres capes. És l'encarregat de comunicar-se amb la capa de persistencia i la de presentació a través dels seus respectius controladors.

En quant a la capa domini, aquest controlador inicialitza la instància d'altres tres: el controlador d'autors, el controlador de carpetes i el controlador de cerques.

Conté tota la lògica del Sistema per realitzar tots els casos d'ús especificats més amunt d'aquest document.