

Gestió de documents

Versió del lliurament 1.0

Català Montaner, Marc

marc.catala.montaner@est.fib.upc.edu

Fors Munar, Adrià

adria.fors@est.fib.upc.edu

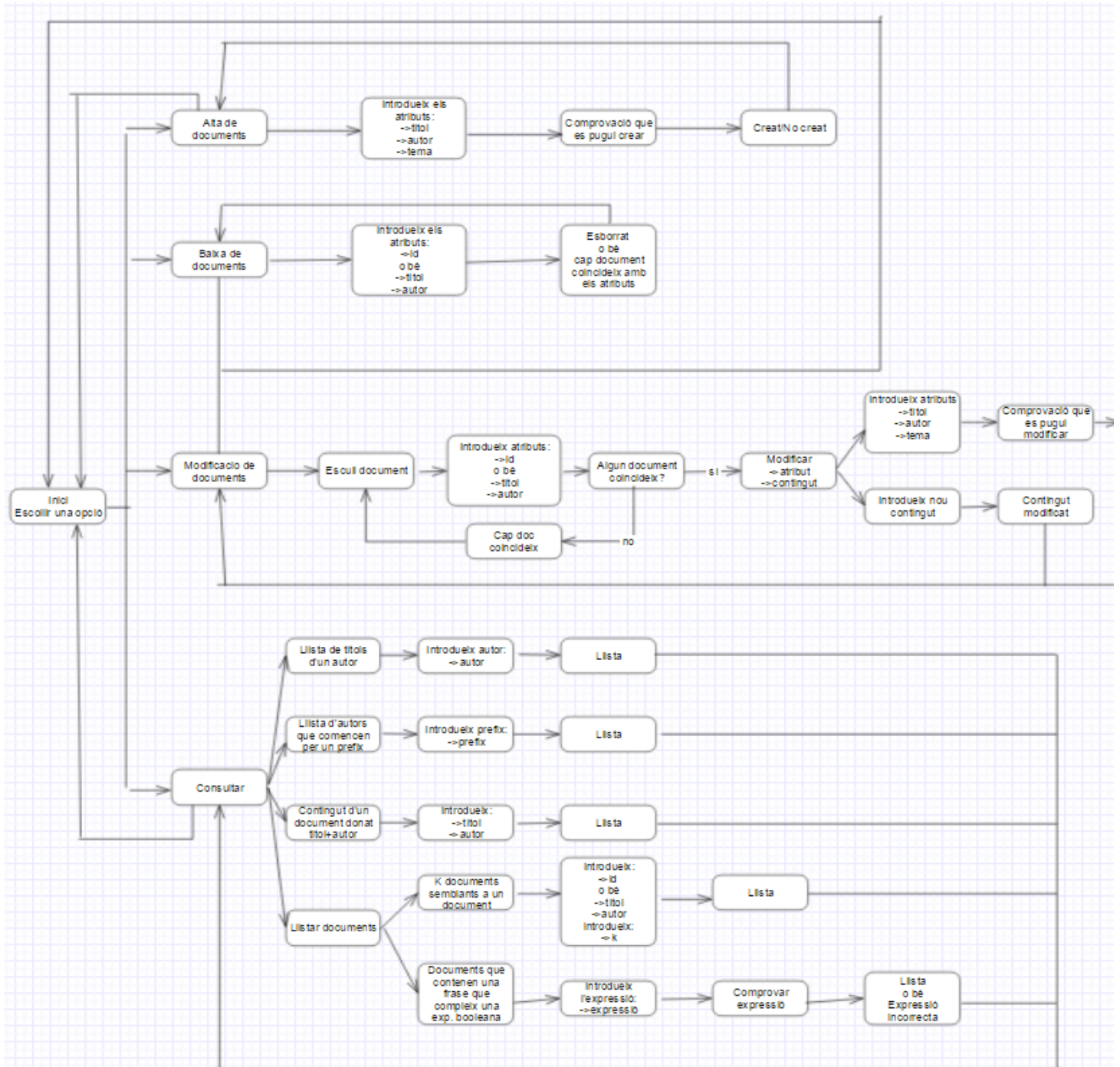
Rodríguez Gutiérrez, Sergio

sergio.rodriguez.gutierrez@est.fib.upc.edu

Índex

- Diagrama de casos d'úspàg.3
- Explicació diagrama de casos d'úspàg.4
- Diagrama estàtic complet del model conceptual de dadespàg.6
- Classes Implementadespàg.7
- Descripció estructures de dades i algorismespàg.8

Diagrama de casos d'ús



Explicació diagrama de casos

d'ús

Per començar, el sistema demana a l'usuari que triï entre donar d'alta documents, donar-los de baixa, modificar-los o consultar-los.

En el cas que l'usuari introdueixi donar d'alta documents, el sistema li demanarà que introdueixi els atributs títol, autor i tema del document que l'usuari vol crear. A continuació el sistema farà una comprovació per tal de que es pugui crear el document. En cas afirmatiu, el document serà creat i tornarà a accedir a donar d'alta documents. En cas contrari, si la implementació falla, no es crearà el document i es retornarà a donar d'alta documents.

En el cas que l'usuari introdueixi donar de baixa documents, el sistema li demanarà que introdueixi l'identificador o bé el títol i autor del document que l'usuari vol donar de baixa. Tant si troba el document i s'elimina com si no es troba cap document que coincideixi amb els atributs introduïts per l'usuari prèviament, es retornarà cap a donar de baixa els documents.

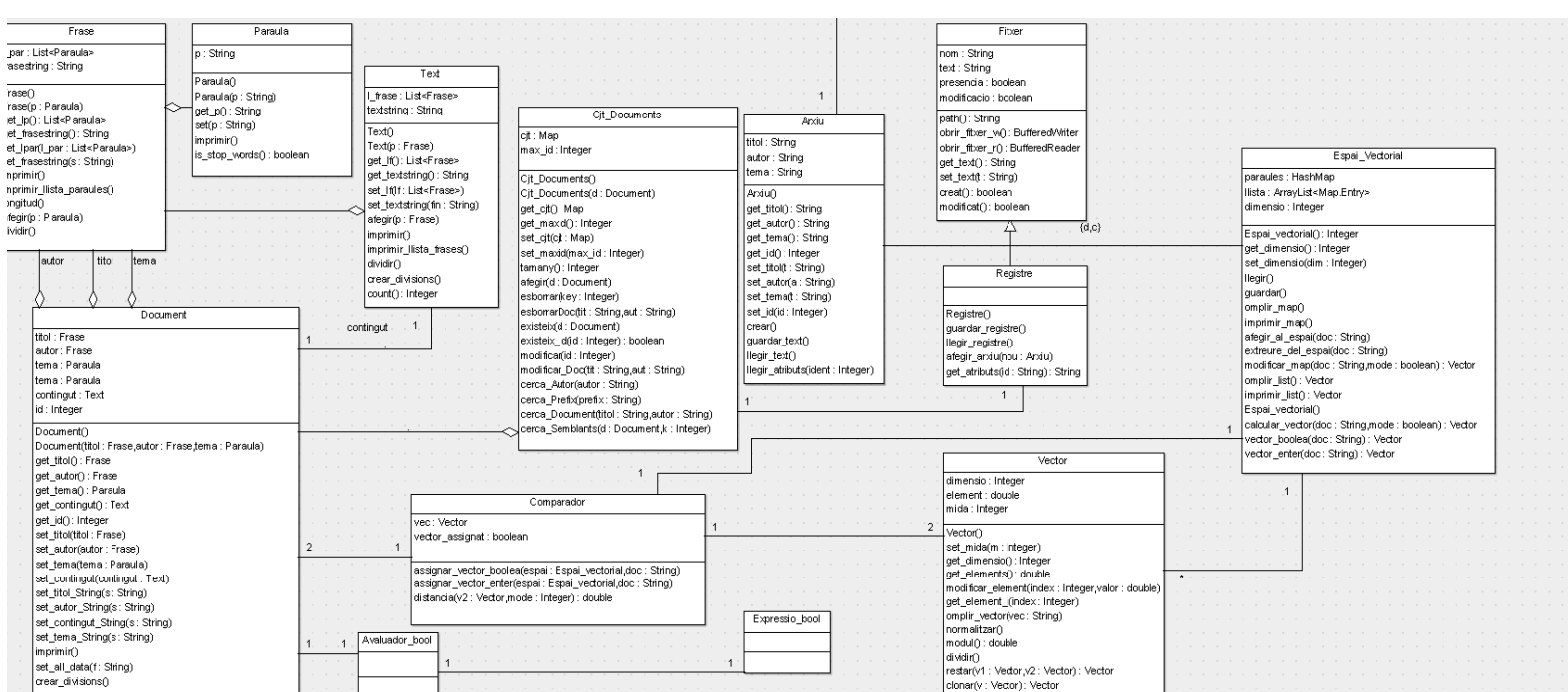
En el cas que l'usuari introdueixi modificar documents, el sistema li demanarà que introdueixi l'identificador o bé el títol i autor del document que l'usuari vol modificar. Si el document no existeix et torna a demanar que introdueixis els atributs anteriors i, en cas que si existeixi, el sistema demana a l'usuari que triï entre modificar atributs o modificar el contingut. En el primer cas, l'usuari ha d'introduir un nou títol, un nou autor i un nou tema. El sistema fa la comprovació de que tot sigui correcte i, finalment, es retorna a la modificació de documents. En el segon cas, l'usuari ha d'introduir el nou contingut. Un cop modificat, es retornarà cap a la modificació de documents.

En el cas que l'usuari introdueixi consultar documents, el sistema li demanarà que esculli una opció. La primera, et mostra una llista de tots els títols d'un autor que l'usuari ha introduït prèviament. En la segona opció, el sistema demana a l'usuari que entri un prefix i el sistema li retornarà una llista de tots els autors que comencin per el prefix anterior. En la tercera opció, el sistema demana a l'usuari que introdueixi el títol i l'autor i el sistema retornarà el

contingut d'aquell document. En la quarta i última opció, l'usuari podrà escollir entre dos opcions més. La primera, donat un natural k i un document que introduirà l'usuari indicant l'identificador d'aquell document o bé el títol i autor, el sistema retornarà una llista amb els k documents més semblants al document indicat per l'usuari. L'altre opció, consisteix en una expressió booleana que retorna tots els documents que satisfan la expressió. Un cop escollida la opció per part de l'usuari, haurà d'introduir l'expressió booleana que vol que sigui avaluada. Finalment, el sistema retornarà la llista de documents que satisfan l'expressió. Tant aquesta opció, com les tres consultes anteriors, es retornarà cap a escollir consulta . En cas de que l'expressió sigui incorrecta, simplement es retornarà cap a la consulta.

Tant per si s'accedeix a donar d'alta documents, donar-los de baixa, modificar-los o consultar-los, l'usuari sempre té l'opció de poder tornar enrere per canviar a una altre opció.

Diagrama estàtic complet del model conceptual de dades



Classes Implementades

La realització de tasques va ser distribuïda amb la intenció d'una repartició equitativa de la feina, tot i que depenen de la classe a implementar, s'ha necessitat crear altres classes o subclasses i no a quedat tant repartit com volíem. La distribució és la següent:

Català Montaner, Marc:

- Classe Paraula
- Classe Frase
- Classe Document
- Drivers corresponents
- Diagrames i pdf

Fors Munar, Adrià:

- Classe Arxiu
- Classe Comparador
- Classe Espai_vectorial
- Classe Fitxer
- Classe Registre
- Classe Vector
- Drivers corresponents

Rodríguez Gutiérrez, Sergio:

- Classe Text
- Classe Cjt_Documents
- Drivers corresponents

Descripció estructures de dades i algorismes

En aquesta primera entrega, hem utilitzat les següents estructures de dades i algorismes:

Estructures de dades:

A la classe Frase hem utilitzat una `List<Paraula>` per guardar per separat totes les paraules de la frase

A la classe Text hem utilitzat el mateix mètode creant una `List<Frase>` per dividir tot el text en frases per, posteriorment, dividir-les en paraules, per tal de poder contar i treure les stop words.

A la classe Cjt_Documents a l'hora de fer la cerca a partir d'un prefix hem utilitzat un `Set<String>` autors per tal d'anar guardant tots els autors que comencen per aquell prefix i, aconseguir que no es mostrin autors repetits gràcies al Set. També hem fet servir un `Map<Integer, Document>`, que s'ha utilitzat per tenir cada document amb un identificador diferent i així poder buscar documents a través d'un identificador.

A la classe Espai_vectorial utilitzem un `HashMap<String,Integer>` per contabilitzar el nombre de repeticions de cada paraula. També utilitzem un `ArrayList<Map.Entry<String, Integer>>` per ordenar les paraules amb les seves respectives repeticions.

A la classe vector utilitzem un Array de doubles que es necessari per guardar tots els termes de la dimensió.

Algorismes:

- **Distància vectorial entre dos documents (Espai vectorial de booleans)**

Donat dos documents i els seus respectius vectors fets a partir d'un string que conté el contingut del document, es fa una resta entre els dos vectors de booleans i, s'obté un vector que, fent el mòdul, obtindrà la distància entre els dos vectors. En cas de que un dels vectors no tingués cap contingut, s'inicialitzaria amb tots els booleans a false.

- **Cosinus entre els vectors de dos documents (Espai vectorial de booleans)**

Donat dos documents i els seus respectius vectors fets a partir d'un string que conté el contingut del document, es fa el producte escalar entre els dos vectors i es guarda el resultat. Aquest, serà dividit entre la multiplicació dels mòduls del vector i sortirà el resultat.

- **Distància vectorial entre dos documents (Espai vectorial d'enters)**

Donat dos documents i els seus respectius vectors fets a partir de d'un string que conté el contingut del document, en cada posició dels dos vectors es fa una divisió entre el nombre de cops que la paraula ha aparegut i el nombre total de paraules d'aquell document. Un cop es formen els dos vectors es resten i es fa el mòdul entre ells per tal de determinar la distància.