

# DESCRIPCIÓ DEL DIAGRAMA DE CLASSES

## Classe Document

Aquesta classe és molt senzilla, i simplement té dos atributs que representen l'autor i el títol del document, i 4 mètodes, els quals són els respectius *getters* i *setters*.

## Classe Controlador de Documents

Aquesta classe ens permet la gestió de documents i fer algunes consultes sobre aquests. Conté dos atributs:

- a. Documents: és una llista de Documents
- b. TítolsPerAutor: és un TreeMap amb clau String i valor Llista de String.

Aquestes dues estructures de dades estan explicades en profunditat en el document [ESTRUCTURA DE DADES I ALGORISMES](#)

Els mètodes més importants de la classe són els següents:

- El mètode [public void crearDocument\(String titol, String autor\)](#): Permet l'alta de documents.
- El mètode [public void eliminarDocument\(String titol, String autor\)](#): Permet la baixa (eliminació) de documents.
- El mètode [public boolean existeixDocument\(String titol, String autor\)](#): Retorna true si existeix un document, altrament false.
- El mètode [public void modificarAutor\(String anticAutor, String nouAutor, String titol\)](#): Permet la modificació de l'autor d'un document.
- El mètode [public void modificarTitol\(String autor, String anticTitol, String nouTitol\)](#): Permet la modificació del títol d'un document.
- El mètode [public List<String> llistarTítolsAutor\(String autor\)](#): Llista tots els títols d'un autor.

- El mètode *public List<String> llistarAutorsPrefix(String prefix)*: Llista tots els autors que contenen el prefix introduït. Si el prefix és buit llista tots els autors.

## Classe Conjunt Continguts

Aquesta classe ens permet la gestió dels continguts dels documents.

Els atributs de la classe Conjunt Continguts són:

- *List<HashMap<String, Integer>> freqContingut*: Representa la freqüència de les paraules per cada contingut.
- *List<String> Contingut*: Conté cadascun dels continguts del sistema.
- *Set<String> stopWords*: Conté el conjunt de paraules stopWords.

Els mètodes més importants de la classe són els següents:

- El mètodes *public void afegirContingut(String contingut)* i *public void afegirContingutPath(String path)*: Permet afegir un nou contingut mitjançant un path a un fitxer o directament amb el contingut.
- El mètodes *public void modificarContingut(int id, String contingut)* i *public void modificarContingutPath(int id, String path)*: Permet modificar el contingut que té la posició id en l'estructura de dades mitjançant un path a un fitxer o directament amb el contingut.
- El mètode *public int[] kRellevants(String[] paraules, int k, int mode)*: Obté els índexs dels k documents més rellevants amb un conjunt de paraules. El mode serveix per indicar quins dels dos modes utilitzar en l'assignació de pesos explicats en el document *ESTRUCTURA DE DADES I ALGORISMES*.
- El mètode *public void eliminarContingut(int id)*: Permet la baixa (eliminació) d'un contingut.

- El mètode `public String[] obtenirParaulesContingut(int id)`: Permet obtenir totes les paraules del contingut en la posició id en el nostre estructura de dades sense stopWords.
- Els mètodes `public String getContingut(int id)` i `public List<String> getConjuntContinguts()`: Permet obtenir o bé tots els continguts del sistema o un contingut en concret.

## Classe Expressió

Aquesta classe fa referència a les consultes de les expressions booleanes donats pels usuaris. Les expressions booleanes estan formades pels operadors & | i ! i conjunt de paraules (delimitats per {}), seqüències de paraules (delimitats per ""), o paraules soltes com a operands. Per exemple donada una expressió {p1 p2 p3} & ("hola adeu" | pep) & !joan, una frase satisfà aquesta expressió si conte les 3 paraules p1 p2 p3, i conte la seqüència "hola adeu" o la paraula Pep i no conté la paraula Joan.

Per poder interpretar aquestes expressions, utilitzarem una estructura de dades Binary Tree que ens servirà a l'hora d'avaluar l'expressió amb les frases dels documents. Per consegüent, a la classe Expressió té un atribut Binary Tree i un booleà per determinar si l'expressió que ens dona l'usuari està ben formalitzar o no.

Els atributs de la classe Expressió són:

- BinaryTree theTree: Representa l'expressió però en l'estructura de binaryTree
- Boolean es\_correccte: Representa si l'expressió donat per l'usuari estigui ben formalitzat o no.

## Classe Controlador d'Expressions

Aquesta classe ens permet la gestió de les expressions booleanes, és a dir, permet donar alta, baixa i modificacions del conjunt d'expressions que els guardarem en una estructura de hashMap on la clau l'expressió donat per l'usuari i el valor és la classe Expressió.

Els atributs de la classe Controlador de Expressions són:

- HashMap<String,Expressio> expressions: Representa el conjunt d'expressions.

Les principals funcionalitats d'aquesta classe són:

- el mètode public List ConsultaExpressioBooleana(String ex,List cont): Aquest mètode et retorna tots els índexs dels documents que satisfà l'expressió booleana en alguna de les frases dels seus continguts.
- el mètode public void anadir\_expressio(String ex): Permet l'alta d'expressions booleanes.
- el mètode public void deleteExpressio(String key): Permet la baixa (eliminació) de funcions booleanes.
- el mètode public void setExpressio(String key, String nova\_ex): Permet la modificació de funcions booleanes.

## **Classe Controlador de Domini**

La classe Controlador de Domini ens permet gestionar totes les funcionalitats del programa. Com atributs, té els controladors de classes que, en conjunt, duen a terme les diferents funcionalitats esmentades. A partir d'aquest, el controlador de domini va rebre les diferents peticions i les delega al seu controlador corresponent.

## **Altres classes**

El controlador de descàrregues i el controlador de formats no s'havien d'implementar per aquesta entrega, per tant, la seva explicació queda per les següents entregues.