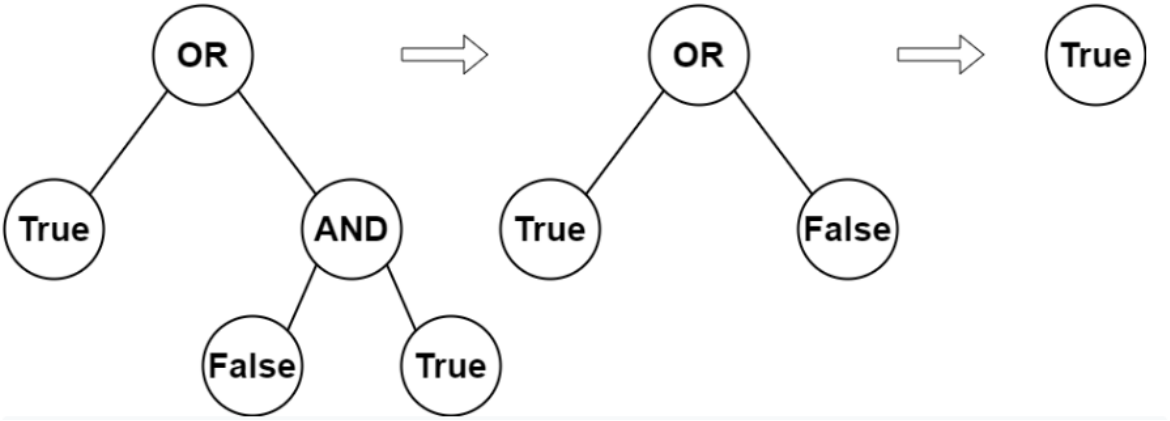
ESTRUCTURA DE DADES I ALGORISMES

Expressió

A la classe d'expressió es fa servir una estructura de dades de BinaryTree per poder-los avaluar les frases de cada contingut. Hem fet servir BinaryTree perquè resulta una forma més fàcil per poder comprovar si una frase satisfà el binaryTree que tenim. Per cada node contindrà la informació de la paraula i un booleà per avaluar si és true o false. Per exemple:



Si tenim una expressió del dibuix anterior on els fulls dels nodes són les paraules o seqüència de paraules que hem de comprovar, en el cas que el fill esquerre del pare OR és True llavors no ens fa falta comprovar el fill dret perquè no ens interessa.
L'especialitat del BinaryTree és que d'alguna forma ens divideix una expressió en subexpressions, és a dir, separa una expressió en 2 expressions fins que no es pot més que és el cas quan ens trobem en les fulles que han de ser si o si paraules a comprovar.

TF-IDF

El tf-idf és una mesura que expressa com és de rellevant unes paraules per a un document en un conjunt.
Aquesta mesura s'utilitza sovint com a factor de ponderació per la recuperació de la informació i en la mineria de text.
El valor tf-idf augmenta proporcionalment amb el nombre de vegades que apareix una paraula en el document, però això es compensa per la freqüència de la paraula en el conjunt de documents. Això serveix per gestionar el fet que algunes paraules són generalment més comunes que altres.

Hem utilitzat dos mètodes per l'assignació de pesos en el tf-idf que són uns dels més comuns.

Mètode 1:

- tf: és la freqüència amb la que apareix una paraula en el document en concret.
- idf: log(N/nt), on N és el nombre de documents i nt el nombre de documents que contenen la paraula t.

Mètode 2:

- tf: log(1+freqt), on la freqt és el nombre de cops que apareix la paraula t en el document
- idf: 1, en aquest mètode idf és unitari.

Documents i continguts

Hem decidit gestionar els documents i els seus continguts per separat, ja que després de mantenir diverses discussions amb l'equip i amb el professorat, vam arribar a la conclusió que era millor tenir-ho separat des d'un bon començament per facilitar les operacions i la implementació de la capa de persistencia de les següents entregues. Així doncs, un document estarà constituït per un títol i un autor, i el contingut d'aquest document serà emmagatzemat en una altra estructura de dades.

Començarem explicant les estructures de dades relacionades amb els documents. Hi ha dues estructures de dades que estan relacionades amb els documents. La primera és l'estructura de dades *documents* en el Controlador de Documents. Aquesta estructura és una llista de documents que conté tots els documents donats d'alta des del començament de l'execució del programa. Cal recalcar que si un document és donat de baixa s'elimina de la llista. Hem escollit una llista perquè d'aquesta manera podem utilitzar la posició dels documents en aquesta llista per indexar-los, i fer ús d'aquests índex en altres estructures de dades. La segona estructura de dades relacionada amb els documents és *titolsPerAutor.* Aquesta estructura és un TreeMap on la clau és un String i el valor és una llista de String. El TreeMap conté tantes claus com autors diferents hi ha entre tots els documents, i cada llista conté tants String com títols hi ha amb autor igual a la clau. Hem creat aquest TreeMap perquè ens permet fer les consultes de títols d'un autor i d'autors que compleixen un prefix amb un cost molt baix. Això és així perquè mantenim el TreeMap ordenat de forma creixent alfabèticament.

Per altra banda, les estructures de dades relacionades amb el contingut són tres,
la primera és l'encarregada d'emmagatzemar els continguts, i les altres dues tenen com a únic propòsit facilitar el tf-idf. Totes es troben com a atributs en la classe ConjuntContinguts.
La primera estructura s'anomena *Contingut*, i és una llista de String. Aquesta llista té tants elements com elements conté *documents*. A cada posició de la llista hi ha el String que conté el contingut del document que hi ha en la mateixa posició en *documents.* Cal recalcar que *documents* i *Contingut* sempre tindran el mateix nombre d'elements, ja que si es dóna un document d'alta el seu contingut també

s'afegirà a l'estructura, i si se'n dóna un de baixa el seu contingut s'esborrarà de l'estructura.
La segona estructura s'anomena ***freqContingut***.. Aquesta estructura es una Llista de HashMap. A cada posició de la llista hi ha el HashMap que fa referència al contingut que hi ha emmagatzemat en la mateixa posició en ***Contingut***. Tots els HashMap tenen com a clau un String i com a valor un enter. Hi ha tantes claus com paraules diferents conté el contingut de ***Contingut*** amb l'índex del HashMap. L'enter representa la freqüència de la paraula en el contingut de ***Contingut*** amb l'índex del HashMap.
La tercera i última estructura de dades és un Set de String. S'anomena ***stopWords***, i conté tants String com stop words existeixen en els fitxers proporcionats per l'assignatura en el racó.

Funcionalitats principal del programa

El programa desenvolupat és un gestor de documents. Aquest permet fer ús de les funcions bàsiques que conformen qualsevol gestor de documents. A continuació, expliquem les funcionalitats que presenta.
Les funcionalitats bàsiques i més importants són l'alta i la baixa de documents en el programa. Una alta és la creació d'un document, el qual posseeix un autor, un títol i un contingut. Per altra banda, una baixa és l'eliminació d'un document existent en el programa.
A més, el programa també permet dur a terme modificacions sobre els documents. Aquestes modificacions es poden dur a terme sobre l'autor, el títol o el contingut del document.
Una altra funcionalitat característica dels gestors de documents i, per tant, present en el nostre gestor també, és la possibilitat de poder dur a terme diferents tipus de consultes sobre els documents existents en el gestor. Aquestes consultes són les següents:

- Consultar els títols d'un cert autor.
- Consultar els autors que contenen un prefix, a decidir per l'usuari.
- Consultar un cert número, a decidir per l'usuari, de documents semblants a un document base, també a decidir per l'usuari.

- Consultar el conjunt de documents que compleixen una expressió booleana ben formada a partir dels caràcters **&**, **|**, **i !**.
- Consultar un cert número, a decidir per l'usuari, de documents més rellevants donades un conjunt de paraules, també a decidir per l'usuari.

Per últim, ens agradaria afegir que el nom que hem escollit per l'extensió del format propi del programa és jamp.