DESCRIPCIÓN DE LAS CLASES

1- Clase Autor

Descripción:

Representa un autor.

Atributos:

- name (String): identificador del autor.

Relaciones:

- Relación de asociación con la clase “Autores”: relaciona la clase “Autor” con todos los autores creados.
- Relación de asociación con la clase “Título”: relaciona la clase “Autor” con el Título.

Métodos:

- Funcionalidades básicas (constructoras, getters y setters).

2- Clase Autores

Descripción:

Representa un conjunto de autores.

Atributos:

- currentIdx (Integer): cantidad actual de documento.
- autores (TreeMap<Autor, HashMap<Título, Integer>>): el conjunto de autores con sus documentos.

Relaciones:

- Relación de asociación con la clase “Autor”: relaciona la clase “Autores” con el Autor para obtener todos los autores del sistema.
- Relación de asociación con la clase “ControladorDocumento”: relaciona la clase “Autores” con el ControladorDocumento para instanciar sus funcionalidades.

Métodos:

- Funcionalidades básicas (constructoras y getters).

Públicos:

- getOrderedAutores(): devuelve un conjunto de autores ordenados.
- getDocumentIdx(DocumentHeader header): determina la posición del documento cuyo DocumentHeader header.
- has(Autor aut), has(String aut): determina la existencia del Autor aut o la existencia del autor cuyo nombre aut, respectivamente.
- add(Autor aut), add(String aut): añade un Autor aut o un autor cuyo nombre aut al conjunto de Autores, respectivamente.
- addTitleToAutor(DocumentHeader header): añade un título al conjunto de Títulos de un autor.
- remove(Autor aut), remove(String aut): elimina el Autor aut o el autor cuyo nombre aut del conjunto de Autores, respectivamente.
- removeTitle(String aut, String tit): borra el documento cuyo nombre de autor aut y título tit.

3- Clase Título

Descripción:

Representa un título.

Atributos:

- name (String): identificador del título.

Relaciones:

- Relación de asociación con la clase “Autor”: relaciona la clase “Título” con Autor para generar un Documento.

Métodos:

- Funcionalidades básicas (constructoras, getters y setters).

4- Clase Documento

Descripción:

Representa un documento.

Atributos:

- autor (Autor): el autor del documento.
- titulo (Titulo): el título del documento.
- contenido (String): el contenido del documento.

Relaciones:

- Clase asociativa entre las clases “Autor” y “Título”: la clase “Documento” hereda sus atributos.
- Relación de asociación con la clase “Documentos”: relaciona la clase “Documento” con todos los documentos creados.

Métodos:

- Funcionalidades básicas (constructoras, getters y setters).

Públicos:

- stringToArrayList(): convierte el contenido de un String a un ArrayList.
- existeString(String conjuntoPalabras): verifica la existencia del conjuntoPalabras en el contenido del documento.

5- Clase Documentos

Descripción:

Representa un conjunto de documentos.

Atributos:

- documentos (ArrayList<Documento>): el conjunto de documentos.
- frecResult (ArrayList<ArrayList<InfoModificado>>): donde se guarda la información de la modificación de los documentos, con sus similitudes respecto otros documentos.
- tf (ArrayList<HashMap<String,Double>>): donde se guarda para cada palabra, la frecuencia de aparición en un determinado documento.
- contidf (HashMap(String,Double)): donde se guarda para cada palabra, la frecuencia de aparición en el conjunto de documentos.

Relaciones:

- Relación de asociación con la clase “ControladorDocumento”: relaciona la clase “ControladorDocumento” con Documentos para que este gestiones sus funcionalidades
- Relación de asociación con la clase “Documento”: relaciona la clase “Documentos” con documentos formados en este conjunto.
- Relación de asociación con la clase “StopWords”: relaciona la clase “Documentos” con StopWords para obtener un conjunto de palabras que no se pueden usar en la generación de similitudes.
- Relación de asociación con la clase “InfoModificado”: relaciona la clase “Documentos” con la información de la modificación de cada documento.

Métodos:

- Funcionalidades básicas (Constructoras, getters, setters).

Privados:

- existeEnContidf(String p): verifica la existencia de la palabra p como key en el contidf.
- tf(ArrayList<String> doc,String p): calcula la frecuencia de la palabra p en el Documento doc.
- inicializarTF(Documento doc): inicializa el tf del Documento que pasa por parámetro.
- actualizarIDF(Documento doc): actualiza el contidf cada vez que haya una modificación del contenido del Documento doc o cuando se ha añadido el Documento doc.
- eliminarDocIDF(Documento doc): actualiza el contidfcuando se ha eliminado el Documento doc.
- modificarTF(int idx): actualiza el tf del documento con el índice idx cuando haya una modificación en el contenido de este documento.
- intersect(HashMap<String,Double> doc1,HashMap<String,Double> doc2): calcula la similitud entre doc1 y doc2 a partir de sus TFs e IDFs.

Públicos:

- add(Documento doc): añade el Documento doc al conjunto de documentos.
- remove(int idx): elimina el documento con el índice idx del conjunto de documentos.
- getContent(int idx): devuelve el contenido del documento con índice idx.
- getDocumento(int idx): devuelve el Documento que tiene el índice idx.
- tieneString(int idx, String texto): verifica si el documento con índice idx contiene el texto que pasa como parámetro.
- tienePalabra(int idx, String p): verifica si el documento con índice idx contiene la palabra p.
- generaSimilitudEntreDocs(int docIndice, int docSim): genera el “ángulo” entre la similitud entre los contenidos de dos documentos.
- modifyContent(int idx, String contenido): reemplaza el contenido del documento con índice idx por el contenido que pasa por parámetro y actualiza el tf, idf y frecResult después de esta modificación.

6- Clase InfoModificado

Descripción:

Representa la información de la modificación del documento.

Atributos:

- frecuencia (Double): la similitud entre dos documentos.
- modif (Boolean): indica si el contenido ha sido modificado, por defecto es false.

Relaciones:

- Relación de asociación con la clase “Documentos”: relaciona la clase “InfoModificado” con todos los documentos creados.

Métodos:

- Funcionalidades básicas (constructora).

7- Clase StopWords

Descripción:

Representa un conjunto de palabras frecuentes que no se usan para la búsqueda de similitud.

Atributos:

- stopwords (set<String>): un conjunto de strings.

Relaciones:

- Relación de asociación con la clase “Documentos”: relaciona la clase “StopWords” con el Documentos para indicar el conjunto de palabras que no se puedan usar en la computación de similitudes.

Métodos:

- Funcionalidades básicas (constructora).

8- Clase Expresión

Descripción:

Representa una expresión booleana.

Atributos:

- expresion (String): identificador de la expresión booleana.

Relaciones:

- Relación de asociación con la clase “ControladorExpresiones”: relaciona la clase “Expresión” con el ControladorExpresiones para instanciar sus funcionalidades.

Métodos:

- Funcionalidades básicas (constructoras, getters y setters).

Privados:

- areBracketsBalanced(String expr): comprobación del balanceamiento de los símbolos de la expresión.

9- Clase Persistencia

Descripción:

Representa una estructura de recuperación de los documentos y de las expresiones booleanas cuando se cierra el programa.

Atributos:

No tiene ningún atributo.

Relaciones:

- Relación de asociación con la clase “ControladorDominio”: relaciona la clase “Persistencia” con el ControladorDominio para instanciar sus funcionalidades de recuperar los datos.

Métodos:

Privados:

- getDirectory(): determina el directorio donde se guarda los datos a recuperar, si no existe se crea uno de nuevo.
- createFile(String autor, String titulo, String contenido, File directory): crea un nuevo fichero con el nombre: autor y título, y guarda el contenido en este fichero.
- persistDocumentos(Documento[] documentos, File directory): guarda los documentos al directorio directory.
- persistExpresiones(HashMap<String, Expresion> expresiones, File directory): guarda las expresiones con sus alias correspondientes al directorio directory.

Públicos:

- persist(Documento[] documentos, HashMap<String, Expresion> expresiones): guarda los documentos

y las expresiones (con sus alias) a un directorio, si no existe se crea uno.
- recoverExpresiones(): recuperar todas las expresiones con sus alias correspondientes guardados en un directorio.
- recoverDocumentos(): recuperar todos los documentos guardados en un directorio.

10- Clase BúsquedaPorSimilitud

Descripción:

Representa la búsqueda de documentos más similares a un documento determinado.

Atributos:

No tiene ningún atributo.

Relaciones:

- Relación de asociación con la clase "ControladorBúsqueda": relaciona la clase "BúsquedasPorSimilitud" con el ControladorBúsqueda para instanciar sus funcionalidades.

Métodos:

Públicos:

- buscar(DocumentHeader header, int K, Libreria libreria): devuelve una lista de k documentos más parecidos al documento cuyo cabecera DocumentHeader header.

11- Clase BúsquedaPorExpresión

Descripción:

Representa la búsqueda de documentos por una expresión booleana.

Atributos:

- libreria (Libreria): el conjunto de documentos existente en la base de datos.

Relaciones:

- Relación de asociación con la clase "ControladorBúsqueda": relaciona la clase "BúsquedaPorExpresión" con el ControladorBúsqueda para instanciar sus funcionalidades.

Métodos:

Privados:

- parse(String expr): determina un BinTree de parse de expresión expr.
- getWords(String words): determina un conjunto de palabras de expresión separado por " ".
- findNext(String expr, char c): encontrar la posición que cierre los símbolos: ' (', ' { ', ' " '.
- findNextOrEnd(String expr, char c): encontrar la posición que cierre los símbolos: ' (', ' { ', ' " '.

Públicos:

- buscar(String expresion, Libreria libreria): determina una lista de documentos que tiene la expresión booleana expresion.
- buscarRec(BinaryTree<ParseNode> bTree, TreeMap<Autor, HashSet<Titulo>> indice): determina el conjunto de documentos que cumplen la condición de la expresión.

12- Clase BúsquedaPorPrefijo

Descripción:

Representa la búsqueda de autores por un prefijo determinado.

Atributos:

No tiene ningún atributo.

Relaciones:

- Relación de asociación con la clase "ControladorBúsqueda": relaciona la clase "BúsquedasPorPrefijo" con el ControladorBúsqueda para instanciar sus funcionalidades.

Métodos:

Públicos:

- buscar(TreeSet<Autor> autores, String prefix): devuelve una lista de autores con el prefijo prefix.
- computeNextPrefix(String prefix): devuelve la siguiente letra de la palabra prefix.

13- Clase ControladorExpresiones

Descripción:

Representa una estructura para gestionar las expresiones booleanas.

Atributos:

- expresiones (HashMap<String, Expresion>): un conjunto de expresiones booleanas con sus alias.

Relaciones:

- Relación de asociación con la clase "Expresion": relaciona la clase "ControladorExpresiones" con la

Expresion para gestionar la expresión booleana con su alia.
- Relación de asociación con la clase “ControladorDominio”: relaciona la clase “ControladorExpresiones” con el ControladorDominio para instanciar sus funcionalidades al ControladorDominio.

Métodos:

- Funcionalidades básicas (constructoras y getters).

Públicos:

- add(String alias, String expresion): añadir una expresión expresion con alia alias.
- updateExpresion(String alias, String expr): cambia la expresión de la alia alias por expr, con la precondición que esta existe.
- removeExpresion(String alias): elimina la alia alias, con la precondición que esta existe.
- getAsString(String alias): devuelva la expresión de la alia alias.
- getExpresiones(): devuelve la lista completa de expresiones con sus alias correspondientes existentes.

14- Clase ControladorDominio

Descripción:

Representa una estructura que se encarga de instanciar la resta de controladores de la capa de dominio y proporcionar una interfaz.

Atributos:

- libreria(Libreria): un conjunto de documentos creados.
- cExpresiones(ControladorExpresiones): el controlador de gestionar las expresiones.

Relaciones:

- Relación de asociación con la clase “ControladorBúsqueda”: relaciona la clase “ControladorDominio” con el ControladorBúsqueda para gestionar sus funcionalidades.
- Relación de asociación con la clase de “ControladorDocumento”: relaciona la clase “ControladorDominio” con el ControladorDocumento” para gestionar sus funcionalidades.
- Relación de asociación con la clase “ControladorExpresiones”: relaciona la clase “ControladorDominio” con el ControladorExpresiones para gestionar sus funcionalidades.
- Relación de asociación con la clase “Persistencia”: relaciona la clase “ControladorDominio” con la Persistencia para gestionar sus funcionalidades.

Métodos:

- Funcionalidad básica (constructora).

Públicos:

- createDocumento(File doc): cargar el fichero doc a la base de datos.
- createDocumento(String aut, String tit, String cont): crea un nuevo documento cuyo autor aut, título tit y contenido cont.
- modifyDocumento(String aut, String tit, String cont): modifica el contenido del document cuyo autor aut y título tit por cont.
- removeDocumento(String aut, String tit): elimina el documento cuyo autor aut y título tit.
- exportDocumento(Documento doc, File path, String name): exporta documento doc con el formato name.
- getTitles(String aut): devuelve una lista de títulos del autor aut.
- obtenerAutoresPrefijo(String pre): devuelve una lista de autores con el prefijo pre.
- getContent(String aut, String tit): devuelve el contenido del documento cuyo autor aut y título tit.
- busquedaPorSimilitud(String aut, String tit, int k): lista los k documentos más parecidos al documento cuyo autor aut y título tit.
- busquedaPorExpresion(String alias): lista los documentos que cumplan la expresión alias.
- addExpresion(String alias, String expr): añade una nueva expresión expr con la alia alias.
-updateExpresion(String alias, String expr): cambia la expresión de la alia alias por expr, con la precondición de que exista la dicha alia.
-removeExpresion(String alias): elimina la alia alias, si existe la alia.
-getExpresion(String alias): devuelve la expresión cuyo alia alias.
-persist(): guarda el estado de la base de datos una vez cerrado el programa.

15- Clase ControladorDocumento(Librería)

Descripción:

Representa una estructura que se encarga de gestionar el conjunto de documentos y autores que hay en el sistema.

Atributos:

- autores(Autores): el conjunto de autores creados.
- documentos(Documentos): el conjunto de documentos creados.

Relaciones:

- Relación de asociación con la clase “Documentos”: relaciona la clase “ControladorDocumento” con el Documentos para gestionar sus funcionalidades.
- Relación de asociación con la clase “Autores”: relaciona la clase “ControladorDocumento” con el Autores para gestionar sus funcionalidades.
- Relación de asociación con la clase “ControladorDominio”: relaciona la clase “ControladorDocumento” con el ControladorDominio que este gestione sus funcionalidades.

Métodos:

- Funcionalidades básicas (constructora y getters).

Públicos:

- createDocumento(String aut, String tit, String cont): crea el documento cuyo autor aut, título tit y contenido cont.
- removeDocumento(String aut, String tit): elimina el documento cuyo autor aut y título tit.
- modifyDocumento(String aut, String tit, String cont): modifica el contenido del documento cuyo autor aut y título tit por el contenido cont.
- tienePalabra(DocumentHeader docHeader, String word): indica la existencia de la palabra word en el documento cuyo DocumentHeader docHeader.
- tieneString(DocumentHeader docHeader, String toMatch): indica la existencia de un string toMatch en el documento cuyo DocumentHeader docHeader.
- computeSimilarity(DocumentHeader header, DocumentHeader toCompare): calcula la similaridad del contenido entre dos documentos, uno cuyo DocumentHeader header y otro, toCompare.

16- Clase ControladorBúsqueda

Descripción:

Representa una estructura que se encarga de gestionar todas las clases de búsquedas de documentos.

Atributos:

No tiene ningún atributo.

Relaciones:

- Relación de asociación con la clase “ControladorDominio”: relaciona la clase “ControladorBúsqueda” con el ControladorDominio para poder instanciar sus funcionalidades.
- Relación de asociación con la clase “BúsquedaPorSimilitud”: relaciona la clase “ControladorBúsqueda” con la BúsquedaPorSimilitud para gestionar los documentos cuyo más similitudes
- Relación de asociación con la clase “BúsquedaPorExpresion”: relaciona la clase “ControladorBúsqueda” con la BúsquedaPorExpresion para gestionar los documentos que corresponden la expresión.
- Relación de asociación con la clase “BúsquedaPorPrefijo”: relaciona la clase “ControladorBúsqueda” con la BúsquedaPorPrefijo para gestionar los documentos con el mismo prefijo.

Métodos:

Públicos:

- buscarPorPrefijo(TreeSet<Autor> autores, String prefix): devuelve una lista de autores con el mismo prefijo prefix.
- buscarPorExpresion(String expr, Libreria libreria): devuelve una lista de DocumentHeader(Autor y Título) que tienen la misma expresión expr.
- buscarPorSimilitud(DocumentHeader header, int K, Libreria libreria): devueleve una lista de K DocumentHeader que són más similares al DocumentHeader header.