

# **GESTOR DE DOCUMENTS**

## **Descripció Diagrama de Classes**

**Projectes de Programació**

Quadrimestre Tardor 2022-23

**Grup 13.4**

Dante de Prado Rojo

Pol Salvador Nogués

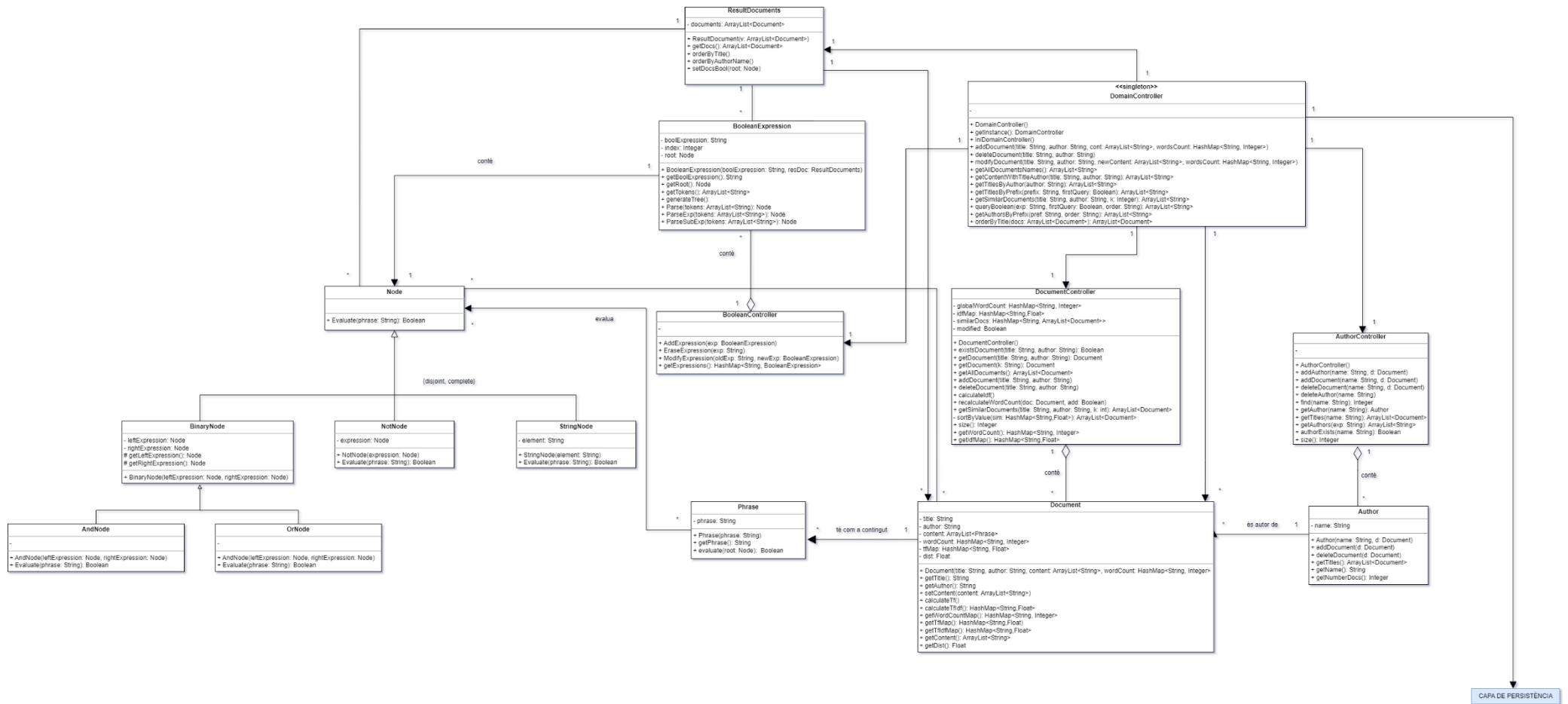
Sergio Sanz Martínez

Youcef Trabsa Biskri



# 1. Diagrama del model conceptual

## 1.1 Disseny del diagrama del model conceptual



**Nom de la classe:** Document

**Breu descripció de la classe:** Enregistra l'informació d'un document que ha sigut carregat o creat a l'aplicació.

**Cardinalitat:** Un per cada document.

**Descripció dels atributs:**

*title* — Títol del document (final).

*author* — Autor del document (final).

*content* — ArrayList de frases del document.

*wordCount* — HashMap amb les paraules i el nombre de vegades que apareixen en un document (final).

*tfMap* — HashMap amb les paraules i el tf de cadascuna.

*disc* — Float amb la distància del vector *tfidf*

**Descripció de les relacions:**

- Relació d'associació amb la classe "Author": indica qui és l'autor del document.
- Relació d'associació amb la classe "Phrase": indica quin és el conjunt de frases que conté el contingut d'un document.
- Relació d'agregació amb la classe "DocumentController": indica que el document forma part del conjunt de documents del sistema
- Relació d'associació amb la classe "Node", requerida per poder fer comprovacions d'expressions booleanes a les frases del Document.
- Relació d'associació amb la classe "DomainController": indica quines funcionalitats es poden dur a terme amb la classe Document, com per exemple crea instàncies d'aquesta i realitzar diverses funcions.
- Relació d'associació amb la classe "ResultDocuments", requerida per guardar el resultat de la cerca.

**Nom de la classe:** DocumentController

**Breu descripció de la classe:** Té el conjunt d'instàncies de "Document"

**Cardinalitat:** Una.

**Descripció dels atributs:**

*globalWordCount* — Mapa que contabilitza el número de repeticions d'una paraula en tots els documents del sistema.

*idfMap* — Frecuencia inversa de totes les paraules de tots el documents.

*similarDocs* — Emmagatzema els documents més semblants a un document segons el seu cosine similarity.

*modified* — Booleà per registrar si s'han afegit/esborrat documents

**Descripció de les relacions:**

- Relació d'associació amb la classe "Document": indica quins són els documents del sistema.
- Relació d'associació amb la classe "DomainController": indica quines funcionalitats es poden dur a terme amb la classe "DocumentController".

**Nom de la classe:** Author

**Breu descripció de la classe:** Enregistra l'informació d'un autor al nostre sistema.

**Cardinalitat:** Un per cada autor.

**Descripció dels atributs:**

*name* — nom de l'autor (final).

**Descripció de les relacions:**

- Relació d'associació amb la classe "Document": indica quin són els seus documents.
- Relació d'agregació amb la classe "AuthorController": indica que l'autor forma part del conjunt d'autors de l'aplicació.

**Nom de la classe:** AuthorController

**Breu descripció de la classe:** Té el conjunt d'instàncies de la classe "Author".

**Cardinalitat:** Una.

**Descripció dels atributs:****Descripció de les relacions:**

- Relació d'associació amb la classe "DomainController": el DomainController crea una instància d' AuthorController i s'utilitza en les funcionalitats necessàries del sistema.
- Relació d'agregació amb la classe "Author": indica quins són els autors del sistema.

**Nom de la classe:** Phrase

**Breu descripció de la classe:** Enregistra les frases que conté el contingut d'un document.

**Cardinalitat:** Una per cada frase de cada document.

**Descripció dels atributs:**

*phrase* — una string que conté una frase del contingut d'un document. (final)

**Descripció de les relacions:**

- Relació d'associació amb la classe "Document": la classe Document crea una instància de Phrase per cada frase del seu contingut.
- Relació d'associació amb la classe "Node": s'utilitza un node *root* a la classe Phrase per saber si l'atribut *phrase* compleix l'expressió booleana continguda al arbre amb node arrel *root*.

**Nom de la classe:** BooleanController

**Breu descripció de la classe:** Té el conjunt d'instàncies de la classe BooleanExpression.

**Cardinalitat:** Una.

**Descripció dels atributs:**

- *untouchedExpressions* — llista de les expressions booleanes existents sense les modificacions fetes a la classe DomainController.

**Descripció de les relacions:**

- Relació d'associació amb la classe "DomainController": el DomainController crea una instància de BooleanController i s'utilitza en les funcionalitats necessàries del sistema.
- Relació d'agregació amb la classe "BooleanExpression": indica el conjunt de BooleanExpression que formen part del sistema.

**Nom de la classe:** BooleanExpression

**Breu descripció de la classe:** Instància d'una consulta de tipus booleà.

**Cardinalitat:** Una per cada consulta booleana.

**Descripció dels atributs:**

*boolExpression* — expressió de la consulta booleana (final).

*index* — variable que s'utilitza per recórrer *tokens* i crear l'arbre. S'ha creat com atribut ja que requerim de saber el seu valor actualitzat en tot moment mentre canviem de classes.

*root* — node arrel de l'arbre que conté la boolean expression.

**Descripció de les relacions:**

- Relació d'associació amb la classe "Node": BooleanExpression fa servir les funcionalitats de Node per poder crear un arbre on estarà continguda la expressió booleana.
- Relació d'agregació amb la classe "BooleanController": la classe BooleanController conté totes les instàncies de BooleanExpression.
- Relació d'associació amb la classe "ResultDocuments", requerida per eliminar els documents que no compleixen el resultat de la cerca.

**Nom de la classe:** ResultDocuments

**Breu descripció de la classe:** Enregistra el resultat de les diverses consultes que es poden duir a terme en el sistema i realitzar ordenacions sobre aquestes.

**Cardinalitat:** Una.

**Descripció dels atributs:**  
*documents* — ArrayList de documents que indica quins són els documents que satisfan una consulta.

**Descripció de les relacions:**

- Relació d'associació amb la classe "DomainController": el DomainController crea una instància de ResultDocuments i s'utilitza en les funcionalitats necessàries del sistema.
- Relació d'associació amb la classe "Document": indica la informació d'un document dels resultants a una consulta.
- Relació d'associació amb la classe "Node": ResultDocuments comproba si algun document del seu array té alguna frase que compleix l'expressió booleana continguda en Node.
- Relació d'associació amb la classe "BooleanExpression": BooleanExpression li fa arribar a ResultDocument el Node per a que elimini els documents que no compleixin l'expressió booleana.

**Nom de la classe:** DomainController

**Breu descripció de la classe:** S'utilitza per tal que funcionin totes les classes i poder provar les seves funcionalitats.

**Cardinalitat:** Una.

**Descripció dels atributs:**

**Descripció de les relacions:**

- Relació d'associació amb la classe "BooleanController": el DomainController crea una instància de BooleanController i accedeix a ella quan tractem expressions booleanes.
- Relació d'associació amb la classe "DocumentController": el DomainController crea una instància de DocumentController i accedeix a ella quan tractem documents.
- Relació d'associació amb la classe "AuthorController": el DomainController crea una instància de AuthorController i accedeix a ella quan tractem autors.
- Relació d'associació amb la classe "ResultDocuments": el DomainController crea una instància de ResultDocuments i accedeix a aquesta per obtenir els resultats de les consultes, ordenar-les i poder fer consultes sobre consultes.

**Nom de la classe:** Node

**Breu descripció de la classe:** Classe abstracta que representa un node de l'arbre.

**Cardinalitat:** Una per cada expressió booleana.

**Descripció dels atributs:**

**Descripció de les relacions:**

- Relació d'associació amb la classe "BooleanExpression": la BooleanExpression conté un node que es l'arrel de l'arbre binari que conté l'expressió booleana.
- Relació d'associació amb la classe "Phrase": Phrase utilitza el node arrel d'un arbre pero comprovar si el seu atribut frase compleix l'expressió booleana continguda a l'arbre.

- Relació d'associació amb la classe "Document": Document envia un node arrel a totes les seves frases per a que puguin comprovar si compleixen l'expressió booleana continguda a l'arbre.
- Relació d'associació amb la classe "ResultDocument": ResultDocuments envia el node arrel a tots els seus documents per saber si algun compleix l'expressió booleana continguda a l'arbre.