

Gestor de Documents

Identificador de l'equip: 32.3

Joan Caballero Castro: joan.caballero@estudiantat.upc.edu

Jeremy Comino Raigón: jeremy.comino@estudiantat.upc.edu

Marc Gonzalez Vidal: marc.gonzalez.v@estudiantat.upc.edu

Oriol Miró López-Feliu: oriol.miro.lopez-feliu@estudiantat.upc.edu

Versió del lliurament: 2.0

2ª entrega PROP

Índex

1. Diagrama de Classes i Controlador de la Capa de Presentació	1
1.1 Disseny del diagrama del model conceptual	1
1.2 Explicació de les Classes de Presentació	3
1.2.1 VistaPrincipal	3
1.2.2 VistaEditor	5
1.2.3 PanelAltaDocument	6
1.2.4 PanelImportarDocument	7
1.2.5 PanelObrirDocument	8
1.2.6 PanelExportarDocument	8
1.2.7 PanelBaixaDocument	9
1.2.8 PanelEditarTitolAutorDocument	9
1.2.9 PanelAfegirExpressio	10
1.2.10 PanelModificarExpressio	11
1.2.11 PanelEliminarExpressio	11
1.2.12 PanelConsultarExpressio	12
1.2.13 PanelCercarDocAutor	12
1.2.14 PanelCercarDocContingut	13
1.2.15 PanelCercarDocSemblants	14
1.2.16 PanelCercarDocExp	14
1.2.17 PanelCercarDocRellevants	15
1.2.18 PanelCercarAutors	16
1.2.19 PanelDocuments	17
1.2.20 PanelAutors	18
1.2.21 CtrlPresentacio	18
2. Diagrama de Classes i Controlador de la Capa de Domini	22
2.1. Disseny del diagrama de model conceptual	22
2.2 Explicació de les Classes de Domini	24
2.2.1 Document	24
2.2.2 CtrlDomini	25
2.2.3 CtrlDocument	27
2.2.4 CtrlIndex	28
2.2.5 CtrlOrdenacioResultats	30
2.2.6 IndexAutors	31
2.2.7 IndexEspaiVectorial	32
2.2.8 IndexExpressionsBooleans	33
2.2.9 IndexTitol	34
2.2.10 ExpressioBooleana	35
2.2.11 NodeAutors	36
2.2.12 NodeExpressio	36
2.2.13 Vector_EV	37
2.2.14 Vector_TFIDF	38

2.2.15 Vector_BM25	38
2.2.16 Resultat	39
2.2.17 ResultatAutor	40
3. Diagrama de Classes i Controlador de la Capa de Persistència	41
3.1 Disseny del diagrama del model conceptual	41
3.2 Explicació de les Classes de Persistència	43
3.2.1 CtrlPersistencia	43
3.2.2 GestorDocument	44
3.2.3 CtrlConfiguracio	45
3.2.4 Container	45
3.2.5 TransformadorArxius	46
3.2.6 TransformadorTextPla	46
3.2.7 TransformadorXML	46
3.2.8 TransformadorPDF	46
3.2.9 Transformador JSON	46
4. Diagrama de Classes i Controlador de la Capa Transversal	48
4.1 Disseny del diagrama del model conceptual	48
4.2 Explicació de les Classes de la Capa Transversal	50
4.2.1 Tupla	50
5.1 Estructures de dades	51
5.1 Capa de Domini	51
5.1.1 Document	51
5.1.2 IndexAutors	52
5.1.3 IndexTitol	55
5.1.4 IndexEspaiVectorial	55
5.1.5 IndexExpressioBooleana	55
5.1.6 ExpressioBooleana	57
5.1.7 Vector_EV	58
5.1.8 Vector_TFIDF	58
5.1.9 Vector_BM25	58
5.2 Capa de Persistència	59
5.2.1 GestorDocuments	59
5.2.2 CtrlConfiguracio	59
5.2.3 TransformadorTextPla	60
5.2.4 TransformadorXML	60
5.2.5 TransformadorPDF	60
5.2.6 TransformadorJSON	60

1. Diagrama de Classes i Controlador de la Capa de Presentació

1.1 Disseny del diagrama del model conceptual

A continuació mostrem el Diagrama de Classes de la Capa de Presentació. Recomanem baixar el diagrama que es troba ubicat en el directori DOCS en format svg (DiagramaPresentacio.svg) o pdf (DiagramaPresentacio.pdf) per a poder veure-ho amb millor resolució.

1.2 Explicació de les Classes de Presentació

1.2.1 VistaPrincipal

Aquesta és la vista que es mostra en obrir el nostre gestor i és l'encarregada d'implementar totes les funcionalitats del gestor menys la funcionalitat d'editar el contingut dels documents, que en aquest cas s'encarrega una altra vista.

En ella trobem una barra de menús amb totes les funcionalitats dividides en desplegable segons si estan relacionades amb documents, expressions o cerques.

Per utilitzar una funcionalitat haurem de prémer sobre el seu botó corresponent en la barra de menús i apareixerà en la nostra finestra un panell amb tot el contingut necessari per executar la funcionalitat.

- Atributs
 - JMenuBar menuBar: Barra de menús principal.
 - JMenu mDocuments: Menú en la barra de menús principal que agrupa totes les funcionalitats relacionades amb els documents.
 - JMenuItem bAltaDocument: Ítem del menú de documents per obrir la funcionalitat de donar d'alta documents.
 - JMenuItem bImportarDocument: Ítem del menú de documents per obrir la funcionalitat d'importar documents.
 - JMenuItem bObrirDocument: Ítem del menú de documents per obrir la funcionalitat d'obrir documents.
 - JMenuItem bExportarDocument: Ítem del menú de documents per obrir la funcionalitat d'exportar documents.
 - JMenuItem bBaixaDocument: Ítem del menú de documents per obrir la funcionalitat de donar de baixa documents.
 - JMenuItem bEditarTitolAutorDocument: Ítem del menú de documents per obrir la funcionalitat d'editar el títol i l'autor dels documents.
 - JMenu mExpressions: Menú en la barra de menús principal que agrupa totes les funcionalitats relacionades amb les expressions.
 - JMenuItem bAfegirExpressio: Ítem del menú d'expressions per obrir la funcionalitat d'afegir una expressió.
 - JMenuItem bModificarExpressio: Ítem del menú d'expressions per obrir la funcionalitat de modificar una expressió.
 - JMenuItem bEliminarExpressio: Ítem del menú d'expressions per obrir la funcionalitat d'eliminar una expressió.

- JMenuitem bConsultarExpressio: Ítem del menú d'expressions per obrir la funcionalitat de consultar una expressió.
- JMenu mCercarD: Menú en la barra de menús principal que agrupa totes les funcionalitats relacionades amb la cerca de documents.
- JMenuitem bCercarDocAutor: Ítem del menú de cerca de documents per obrir la funcionalitat de cercar documents donat el nom d'un autor.
- JMenuitem bCercarDocContingut: Ítem del menú de cerca de documents per obrir la funcionalitat de consultar el contingut d'un document donat el seu títol i autor.
- JMenuitem bCercarDocSemblants: Ítem del menú de cerca de documents per obrir la funcionalitat de cercar documents semblants a un document seleccionat.
- JMenuitem bCercarDocExp: Ítem del menú de cerca de documents per obrir la funcionalitat de cercar documents donada una expressió booleana.
- JMenuitem bCercarDocRellevants: Ítem del menú de cerca de documents per obrir la funcionalitat de cercar els documents més rellevants a una sèrie de paraules.
- JMenuitem bCercarAutors: Ítem de la barra de menús principal per obrir la funcionalitat de cercar autors per prefix.
- JPanel pPrincipal: Panell principal que anirà canviant el seu contingut pel contingut d'altres panells segons la funcionalitat que hagi escollit l'usuari.
- JPanel pAltaDocument: Panell encarregat de la funcionalitat de donar d'alta documents.
- JPanel pImportarDocument: Panell encarregat de la funcionalitat d'importar documents.
- JPanel pObrirDocument: Panell encarregat de la funcionalitat d'obrir documents.
- JPanel pExportarDocument: Panell encarregat de la funcionalitat d'exportar documents.
- JPanel pBaixaDocument: Panell encarregat de la funcionalitat de donar de baixa documents.
- JPanel pEditarTitolAutorDocument: Panell encarregat de la funcionalitat d'editar el títol i l'autor de documents.
- JPanel pAfegirExpressio: Panell encarregat de la funcionalitat d'afegir expressions booleanes.
- JPanel pModificarExpressio: Panell encarregat de la funcionalitat de modificar expressions booleanes.

- JPanel pEliminarExpressio: Panell encarregat de la funcionalitat d'eliminar expressions booleans.
- JPanel pConsultarExpressio: Panell encarregat de la funcionalitat de consultar el contingut de les expressions booleans donat el seu nom.
- JPanel pCercarDocAutor: Panell encarregat de la funcionalitat de cercar els documents escrits per un autor.
- JPanel pCercarDocContingut: Panell encarregat de la funcionalitat de consultar el contingut d'un document donat el seu títol i autor.
- JPanel pCercarDocSemblants: Panell encarregat de la funcionalitat de cercar els documents més semblants a un altre document.
- JPanel pCercarDocExp: Panell encarregat de la funcionalitat de cercar els documents que compleixen una expressió booleana.
- JPanel pCercarDocRellevants: Panell encarregat de la funcionalitat de cercar els documents més rellevants a una sèrie de paraules.
- JPanel pCercarAutors: Panell encarregat de la funcionalitat de cercar els autors que comencen per un cert prefix.
- Mètodes
 - VistaPrincipal(): Constructora de la vista principal del gestor.

1.2.2 VistaEditor

Aquesta és la vista que es mostra a l'obrir un document, i permet realitzar diverses operacions sobre ell. Disposarem d'una secció on es podrà editar el contingut del document, i a dalt hi haurà una barra on es podran activar altres funcionalitats, dividides en si estan relacionades amb els documents, amb l'edició del document actual o amb el text.

Des del primer menú es podrà obrir un altre document, guardar l'actual o importar l'actual, des del segon es podrà editar l'autor o títol i des del tercer es podrà retallar, copiar o enganxar text.

Per utilitzar una funcionalitat haurem de prémer sobre el seu botó corresponent en la barra de menús i apareixerà en la nostra finestra un panell amb tot el contingut necessari per executar la funcionalitat.

- Atributs
 - JTextArea text: Area on es modificarà el text.
 - JFrame f: JFrame que engloba tot.
 - String autorDocumentActual: String que guarda el nom de l'autor del document que estem editant actualment.

- String titolDocumentActual: String que guarda el títol del document que estem editant actualment.
- String pathDocumentActual: String que guarda el path del document que estem editant actualment.
- JMenuBar mb: Barra que conté tots els menus.
- JMenu mFitxer: Menu per a miObrir, miGuardar i miExportar.
- JMenuItem miObrir: Menu item per obrir un altre document
- JMenuItem miGuardar: Menu item per guardar a memòria l'estat actual del document.
- JMenuItem miExportar: Menu item per exportar el document.
- JMenu mEdicio: Menu per a miEditarAutor i miEditarTitol.
- JMenuItem miEditarAutor: Menu item per editar l'autor del document actual.
- JMenuItem miEditarTitol: Menu item per editar el títol del document actual.
- JMenu mText: Menu per a miRetallar, miCopiar i miEnganxar.
- JMenuItem miRetallar: Menu item per retallar part del text.
- JMenuItem miCopiar: Menu item per copiar part del text.
- JMenuItem miEnganxar: Menu item per enganxar part del text.
- Mètode
 - VistaEditor(): Constructora del panell de donar d'alta documents.

1.2.3 PanelAltaDocument

Aquest panell és l'encarregat d'implementar la funcionalitat d'afegir un nou document. Es mostren dos camps de text per introduir el títol i l'autor del document i un botó que obrirà la finestra de selecció del path. Una vegada se seleccioni el path aquest es mostrarà en text perquè l'usuari pugui veure la seva selecció.

Per donar d'alta el document l'usuari haurà de prémer el botó de confirmar, que en cas de ser tot correcte es mostrarà un missatge indicant que el document s'ha afegit correctament i en cas d'error es mostrarà un pop-up a l'usuari informant de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar lTitol i tfTitol.
 - JPanel p2: Panell per agrupar lAutor i tfAutor.
 - JPanel p3: Panell per agrupar bChooser i tfPath.
 - JLabel lTitol: Text per informar de la funcionalitat del camp de text
 - JTextField tfTitol: Camp de text per introduir el títol.

- JLabel lAutor: Text per informar de la funcionalitat del camp de text
- JTextField tfAutor: Camp de text per introduir l'autor.
- JButton bChooser: Botó que obre la finestra de selecció del path.
- JFileChooser chooser: Finestra per introduir el path del document.
- JTextField tfPath: Text que mostra el path seleccionat.
- JLabel informar: Text que informa que s'ha donat d'alta un document correctament.
- JButton bAcceptar: Botó per confirmar l'addició del document.
- Mètode
 - PanelAltaDocument(): Constructora del panell de donar d'alta documents.

1.2.4 PanellImportarDocument

Aquest panell és l'encarregat d'implementar la funcionalitat d'importar un document. Es mostra un botó que obrirà una finestra per seleccionar el path del document a important, i a la dreta del botó es mostra el path que l'usuari ha seleccionat. Només s'acceptaran documents PDF, TXT, XML i JSON. També es mostra un botó per confirmar que es vol importar el document escollit.

Si el document s'ha importat correctament, es mostrarà un missatge de text indicant que s'ha pogut importar el document, i en cas d'error es mostrarà un pop-up informant l'usuari de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar bChooser i tfPath.
 - JButton bChooser: Botó que obre la finestra de selecció del path.
 - JFileChooser chooser: Finestra per introduir el path del document.
 - JTextField tfPath: Text que mostra el path seleccionat.
 - JLabel informar: Text que informa que s'ha importat el document seleccionat correctament.
 - JButton bAcceptar: Botó per confirmar l'eliminació del document.
- Mètode
 - PanellImportarDocument(): Constructora del panell d'importar documents.

1.2.5 PanelObrirDocument

Aquest panell és l'encarregat d'implementar la funcionalitat d'obrir un document. Es mostra un botó que obrirà una finestra per seleccionar el path del document a obrir, i a la dreta del botó es mostra el path que l'usuari ha seleccionat. També es mostra un botó per confirmar que es vol obrir el document escollit.

Seguidament s'obre la vista del editor de text, per poder actuar sobre el document escollit.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar bChooser i tfPath.
 - JButton bChooser: Botó que obre la finestra de selecció del path.
 - JFileChooser chooser: Finestra per introduir el path del document.
 - JTextField tfPath: Text que mostra el path seleccionat.
 - JButton bAcceptar: Botó per confirmar l'eliminació del document.
- Mètode
 - PanelObrirDocument(): Constructora del panell de d'obrir documents.

1.2.6 PanelExportarDocument

Aquest panell és l'encarregat d'implementar la funcionalitat d'exportar un document. Es mostra un botó que obrirà una finestra per seleccionar el path del document a important, i a la dreta del botó es mostra el path que l'usuari ha seleccionat. A sota es mostrarà un selector per escollir el format al qual es vol exportar el document. També es mostra un botó per confirmar que es vol exportar el document escollit.

Si el document s'ha exportat correctament, es mostrarà un missatge de text indicant que s'ha pogut exportar el document, i en cas d'error es mostrarà un pop-up informant l'usuari de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar bChooser i tfPath.
 - JPanel p2: Panell per agrupar IFormat i cbFormat
 - JButton bChooser: Botó que obre la finestra de selecció del path.
 - JFileChooser chooser: Finestra per introduir el path del document.
 - JTextField tfPath: Text que mostra el path seleccionat.
 - IFormat: Label per explicar la funcionalitat del ComboBox.

- JComboBox cbFormat: ComboBox usada per escollir el format al que s'exportarà el document.
- JLabel informar: Text que informa que s'ha exportat el document seleccionat correctament.
- JButton bAcceptar: Botó per confirmar l'eliminació del document.
- Mètode
 - PanelExportarDocument(): Constructora del panell d'exportar documents.

1.2.7 PanelBaixaDocument

Aquest panell és l'encarregat d'implementar la funcionalitat de donar de baixa un document. Es mostra un botó que obrirà una finestra per seleccionar el path del document a eliminar, i a la dreta d'aquest botó es mostra el path que l'usuari ha seleccionat. També es mostra un botó que l'usuari haurà de prémer per confirmar l'eliminació del document escollit.

Si el document s'ha donat de baixa correctament, es mostrarà un missatge de text indicant que s'ha pogut eliminar el document, i en cas d'error es mostrarà un pop-up informant l'usuari de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar bChooser i tfPath.
 - JButton bChooser: Botó que obre la finestra de selecció del path.
 - JFileChooser chooser: Finestra per introduir el path del document.
 - JTextField tfPath: Text que mostra el path seleccionat.
 - JLabel informar: Text que informa que s'ha eliminat el document seleccionat correctament.
 - JButton bAcceptar: Botó per confirmar l'eliminació del document.
- Mètode
 - PanelBaixaDocument(): Constructora del panell d'eliminar documents.

1.2.8 PanelEditarTitolAutorDocument

Aquest panell és l'encarregat d'implementar la funcionalitat editar el nom o autor d'un document. Es mostra un selector per escollir el document a editar, i una vegada se seleccioni el path aquest es mostrarà en text perquè l'usuari pugui veure la seva selecció.

Seguidament, es mostraran dos labels indicant l'actual títol i autor del document escollit. També hi ha un selector per escollir que es vol editar, i al costat d'aquest camp de text per introduir el nou títol/autor

Per confirmar l'edició el document l'usuari haurà de prémer el botó de confirmar, que en cas de ser tot correcte es mostrarà un missatge indicant que el document s'ha afegit correctament i en cas d'error es mostrarà un pop-up a l'usuari informant de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar bChooser i tfPath.
 - JPanel p2: Panel per agrupar cbFuncionalitat i tfEdicio
 - JButton bChooser: Botó que obre la finestra de selecció del path.
 - JFileChooser chooser: Finestra per introduir el path del document.
 - JTextField tfPath: Text que mostra el path seleccionat.
 - JLabel lInfo: Label que informa del títol i autor actuals del document.
 - JComboBox cbFuncionalitat: ComboBox per escollir si es vol edita títol o autor.
 - JTextField tfdEdicio: TextField per introduir el nou títol o autor
 - JLabel informar: Text que informa que s'ha eliminat el document seleccionat correctament.
 - JButton bAcceptar: Botó per confirmar obrir el document.
- Mètode
 - PanelEditarDocument(): Constructora del panell d'editar documents.

1.2.9 PanelAfegirExpressio

Aquest panell és l'encarregat d'implementar la funcionalitat d'afegir una nova expressió. Es mostren dos camps de text per introduir el nom i el contingut de la nova expressió i un botó que s'haurà de prémer per confirmar l'addició de l'expressió.

Si l'expressió s'ha creat correctament, es mostrarà un missatge de text indicant que s'ha pogut crear l'expressió, i en cas que s'hagi produït un error apareixerà un pop-up informant de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar lNom i tfNom.
 - JPanel p2: Panell per agrupar lContingut i tfContingut.
 - JLabel lNom: Text per informar de la funcionalitat del camp de text.
 - JTextField tfNom: Camp de text per introduir el nom.
 - JLabel lContingut: Text per informar de la funcionalitat del camp de text.

- JTextField tfContingut: Camp de text per introduir el contingut.
- JLabel informar: Text que informa que s'ha creat una expressió correctament.
- JButton bAcceptar: Botó per confirmar l'addició de l'expressió.
- Mètode
 - PanelAfegirExpressio(): Constructora del panell d'afegir expressions.

1.2.10 PanelModificarExpressio

Aquest panell és l'encarregat d'implementar la funcionalitat de modificar una expressió. Es mostra un selector per escollir l'expressió a modificar, seguidament d'un text on es mostra el contingut de l'expressió seleccionada, un camp de text per posar el nou contingut i un botó per confirmar els canvis.

Si l'expressió s'ha modificat correctament, es mostrarà un missatge de text indicant que s'ha pogut modificar l'expressió, i en cas que s'hagi produït un error apareixerà un pop-up informant de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar INom i cbContingut.
 - JPanel p2: Panell per agrupar INouContingut i tfNouContingut.
 - JLabel INom: Text per informar de la funcionalitat selector.
 - JComboBox cbContingut: Selector per escollir l'expressió.
 - JLabel IContingut: Text per informar del actual contingut de l'expressió.
 - JLabel INouContingut: Text per informar de la funcionalitat del camp de text.
 - JTextField tfNouContingut: Camp de text per introduir el nou contingut.
 - JLabel informar: Text que informa que s'ha modificat l'expressió correctament.
 - JButton bAcceptar: Botó per confirmar la modificació de l'expressió.
- Mètode
 - PanelModificarExpressio(): Constructora del panell de modificar expressions.

1.2.11 PanelEliminarExpressio

Aquest panell és l'encarregat d'implementar la funcionalitat d'eliminar una expressió. Es mostra un selector per escollir l'expressió a eliminar.

Si l'expressió s'ha eliminat correctament, es mostrarà un missatge de text indicant que s'ha pogut eliminar l'expressió.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar INom i cbContingut.
 - JLabel INom: Text per informar de la funcionalitat selector.
 - JComboBox cbContingut: Selector per escollir l'expressió.
 - JLabel informar: Text que informa que s'ha eliminat l'expressió correctament.
 - JButton bAcceptar: Botó per confirmar l'eliminació de l'expressió.
- Mètode
 - PanelEliminarExpressio(): Constructora del panell d'eliminar expressions.

1.2.12 PanelConsultarExpressio

Aquest panell és l'encarregat d'implementar la funcionalitat de consultar el contingut d'una expressió. Es mostra un selector per escollir l'expressió a consultar, seguidament d'un text on es mostra el contingut de l'expressió seleccionada.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar INom i cbContingut.
 - JLabel INom: Text per informar de la funcionalitat selector.
 - JComboBox cbContingut: Selector per escollir l'expressió.
 - IMostrarContingut: Label per mostrar el contingut de l'expressio seleccionada.
- Mètode
 - PanelConsultarExpressio(): Constructora del panell de consultar expressions.

1.2.13 PanelCercarDocAutor

Aquest panell és l'encarregat d'implementar la funcionalitat de cercar els documents escrits per un autor. Es mostra un ComboBox amb un desplegable on es mostren tots els autors existents en el que l'usuari en podrà triar un i un botó que l'usuari haurà de prémer per realitzar la cerca dels documents escrits per l'autor seleccionat.

Separada d'aquesta zona que anomenem la zona d'input tenim a sota la zona de resultats, que és on es mostren els resultats de la consulta mitjançant la classe PanelDocuments, en la qual es mostrarà per cada document el seu títol i autor i un ComboBox desplegable per seleccionar el mètode d'ordenació dels resultats.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel p1: Panell per agrupar lAutor i cb.
 - JLabel lAutor: Text per informar de la funcionalitat del ComboBox.
 - JComboBox<String> cb: ComboBox on es mostren tots els noms dels autors.
 - JButton bAcceptar: Botó per realitzar la consulta.
 - JSeparator separator: Línia que separa la zona d'input de la de resultats.
 - JPanel pDocuments: Panell on es mostraran els resultats de la consulta.
 - JPanel pAux: Panell auxiliar que estarà dins de pDocuments per poder mostrar els resultats de les consultes dinàmicament.
- Mètode
 - PanelCercarDocAutor(): Constructora del panell de cercar documents donat el nom d'un autor.

1.2.14 PanelCercarDocContingut

Aquest panell és l'encarregat d'implementar la funcionalitat de llistar el contingut d'un document donat el seu títol i el nom del seu autor. Es mostra un camp de text perquè l'usuari introdueixi el títol del document i un altre camp de text perquè introdueixi el nom de l'autor. Per realitzar la cerca haurà de prémer un botó situat a sota d'aquests dos camps. Una vegada realitzada la cerca es mostrarà el contingut del document en un àrea de text si s'ha pogut trobar el document introduït, o en cas que no s'hagi trobat el document es mostrarà un pop-up informant a l'usuari de l'error.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar lTitol i tfTitol.
 - JPanel p2: Panell per agrupar lAutor i tfAutor.
 - JLabel lTitol: Text per informar de la funcionalitat del camp de text.
 - JTextField tfTitol: Camp de text per introduir el títol.
 - JLabel lAutor: Text per informar de la funcionalitat del camp de text.
 - JTextField tfAutor: Camp de text per introduir l'autor.
 - JButton bCercar: Botó per realitzar la cerca.
 - JTextArea: Àrea de text on es mostrarà el contingut del document.
- Mètode
 - PanelCercarDocContingut(): Constructora del panell de cercar el contingut d'un document donat el seu títol i autor.

1.2.15 PanelCercarDocSemblants

Aquest panell és l'encarregat d'implementar la funcionalitat de cercar els K documents semblants a un document donat. Es mostra un camp de números perquè l'usuari introdueixi el nombre de documents que vol que la consulta li retorni, i un botó que obrirà una finestra per seleccionar el path del document amb el qual es contrastarà. Seguidament, es podrà escollir quin sistema de semblança es vol utilitzar, Tf-idf o BM25.

Per realitzar la cerca l'usuari haurà de prémer un botó situat a sota d'aquests camps.

Separada d'aquesta zona que anomenem la zona d'input tenim a sota la zona de resultats, que és on es mostren els resultats de la consulta mitjançant la classe PanelDocuments, en la qual es mostrarà per cada document el seu títol i autor i un ComboBox desplegable per seleccionar el mètode d'ordenació dels resultats.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel p1: Panell per agrupar IK i sK.
 - JPanel p2: Panell per agrupar IMetode i cbMetode.
 - JLabel IK: Text per informar de la funcionalitat del camp de números.
 - JSpinner sK: Camp de números per introduir el nombre de documents a cercar.
 - JLabel IMetode: Text per informar de la funcionalitat del chooser de mètode.
 - JComboBox cbMetode: ComboBox per escollir el metode de semblança a utilitzar.
 - JButton bChooser: Botó que obre la finestra de selecció del path.
 - JFileChooser chooser: Finestra per introduir el path del document.
 - JButton bCercar: Botó per realitzar la consulta.
 - JSeparator separator: Línia que separa la zona d'input de la de resultats.
 - JPanel pDocuments: Panell on es mostraran els resultats de la consulta.
 - JPanel pAux: Panell auxiliar que estarà dins de pDocuments per poder mostrar els resultats de les consultes dinàmicament.
- Mètode
 - PanelCercarDocSemblants(): Constructora del panell de cercar documents rellevants a un document.

1.2.16 PanelCercarDocExp

Aquest panell és l'encarregat d'implementar la funcionalitat de cercar els documents que compleixin una expressió booleana. Es mostra un ComboBox per escollir el tipus de cerca: per una expressió guardada anteriorment o per una introduïda manualment.

En el cas de la primera, es mostra una segona ComboBox per escollir l'expressió, i en el cas de la segona, un TextField per poder introduir l'expressió.

Seguidament, hi haurà un botó que l'usuari haurà de prémer per realitzar la cerca dels documents que compleixin l'expressió booleana seleccionada. En cas que fos introduïda manualment, i hi hagués algun error, es mostraria un pop-up informant de l'error.

Separada d'aquesta zona que anomenem la zona d'input tenim a sota la zona de resultats, que és on es mostren els resultats de la consulta mitjançant la classe PanelDocuments, en la qual es mostrarà per cada document el seu títol i autor i un ComboBox desplegable per seleccionar el mètode d'ordenació dels resultats.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel p1: Panell per agrupar cbFuncionalitat, cbExpressions i tfExpressio.
 - JComboBox cbFuncionalitat: ComboBox on es mostren les dues formes de realitzar aquesta consulta
 - JComboBox cbExpressions: ComboBox on es mostren les expressions booleanes guardades.
 - JComboBox tfExpressio: TextField que serveix per introduir manualment l'expressió.
 - JButton bAcceptar: Botó per realitzar la consulta.
 - JSeparator separator: Línia que separa la zona d'input de la de resultats.
 - JPanel pDocuments: Panell on es mostraran els resultats de la consulta.
 - JPanel pAux: Panell auxiliar que estarà dins de pDocuments per poder mostrar els resultats de les consultes dinàmicament.
- Mètode
 - PanelCercarDocExpressio(): Constructora del panell de cercar documents per expressió booleana.

1.2.17 PanelCercarDocRellevants

Aquest panell és l'encarregat d'implementar la funcionalitat de cercar els documents més rellevants a una sèrie de paraules. Es mostra un camp de números perquè l'usuari introdueixi el nombre de documents que vol que la consulta li retorni i un camp de text per introduir les paraules que es volen consultar. Per realitzar la cerca l'usuari haurà de prémer un botó situat a sota d'aquests camps.

Separada d'aquesta zona que anomenem la zona d'input tenim a sota la zona de resultats, que és on es mostren els resultats de la consulta mitjançant la classe PanelDocuments, en

la qual es mostrarà per cada document el seu títol i autor i un ComboBox desplegable per seleccionar el mètode d'ordenació dels resultats.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel p1: Panell per agrupar IK i sK.
 - JPanel p2: Panell per agrupar IParaules i tfParaules.
 - JLabel IK: Text per informar de la funcionalitat del camp de números.
 - JSpinner sK: Camp de números per introduir el nombre de documents a cercar.
 - JLabel IParaules: Text per informar de la funcionalitat del camp de text.
 - JTextField tfParaules: Camp de text per introduir les paraules a cercar.
 - JButton bCercar: Botó per realitzar la consulta.
 - JSeparator separator: Línia que separa la zona d'input de la de resultats.
 - JPanel pDocuments: Panell on es mostraran els resultats de la consulta.
 - JPanel pAux: Panell auxiliar que estarà dins de pDocuments per poder mostrar els resultats de les consultes dinàmicament.
- Mètode
 - PanelCercaDocRellevants(): Constructora del panell de cercar documents rellevants a una sèrie de paraules.

1.2.18 PanelCercarAutors

Aquest panell és l'encarregat d'implementar la funcionalitat de cercar els autors que comencen per un prefix. Es mostra una TextField perquè l'usuari introdueixi el prefix que vol, i un botó que l'usuari haurà de prémer per realitzar la cerca dels autors que comencin pel prefix indicat.

Separada d'aquesta zona que anomenem la zona d'input tenim a sota la zona de resultats, que és on es mostren els resultats de la consulta mitjançant la classe PanelAutors, en la qual es el seu nom i imatge, i un ComboBox desplegable per seleccionar el mètode d'ordenació dels resultats.

- Atributs
 - JLabel titolPanel: Títol del panell.
 - JPanel p1: Panell per agrupar IAutor i tfPrefix.
 - JLabel IAutor: Text per informar de la funcionalitat del tfPrefix.
 - JTextField tfPrefix: TextField Per introduir el prefix dels autors.
 - JButton bAcceptar: Botó per realitzar la consulta.
 - JSeparator separator: Línia que separa la zona d'input de la de resultats.

- JPanel pAutors: Panell on es mostraran els resultats de la consulta.
- JPanel pAux: Panell auxiliar que estarà dins de pAutors per poder mostrar els resultats de les consultes dinàmicament.
- Mètode
 - PanelCercarAutorsPrefix(): Constructora del panell de cercar autors donat un prefix.

1.2.19 PanelDocuments

Aquest panell és l'encarregat de mostrar per pantalla els resultats de les consultes de documents i l'opció d'ordenar els resultats segons uns certs criteris. Es mostra un ComboBox desplegable on l'usuari podrà escollir quin mètode d'ordenació utilitzar.

Per cada document es mostra una icona juntament amb la seva informació com pot ser el títol i el nom del seu autor principalment. Segons el mètode d'ordenació escollit es poden mostrar altres informacions com per exemple la data de creació del document si hem decidit ordenar els resultats segons la seva data de creació.

- Atributs
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar cbOrdenarDocuments.
 - JPanel p2: Panell on es mostraran els resultats de la consulta. Cada document estarà compost per un JLabel pel títol, un JLabel per l'autor, un JLabel extra que es mostrarà en determinades ocasions i una icona.
 - String[] metodesOrdenacioDocuments: Array amb els mètodes d'ordenació de documents que implementa el nostre gestor (alfabèticament per títol i autor, alfabèticament per autor i títol, per mida i per data de creació).
 - JComboBox cbOrdenarDocuments: ComboBox amb els mètodes d'ordenació de metodesOrdenacioDocuments.
- Mètodes
 - PanelDocuments(): Constructora del panell de mostrar els resultats de les consultes de documents.
 - createImageIcon(String path, String desc): Funció que retorna la imatge guardada a *path* amb descripció *desc*.

1.2.20 PanelAutors

Aquest panell és l'encarregat de mostrar per pantalla els resultats de les consultes d'autors i l'opció d'ordenar els resultats segons uns certs criteris. Es mostra un ComboBox desplegable on l'usuari podrà escollir quin mètode d'ordenació utilitzar.

Per cada autor es mostra una icona juntament amb el seu nom i el nombre de títols escrits.

- Atributs
 - JPanel pPrincipal: Panell principal que agrupa tots els altres panells.
 - JPanel p1: Panell per agrupar cbOrdenarAutors.
 - JPanel p2: Panell on es mostraran els resultats de la consulta. Cada autor estarà compost per un JLabel pel nom, un JLabel pel nombre de títols i una icona.
 - String[] metodesOrdenacioAutors: Array amb els mètodes d'ordenació d'autors que implementa el nostre gestor (alfabèticament per nom d'autor i decreixentment per nombre de títols escrits).
 - JComboBox cbOrdenarAutors: ComboBox amb els mètodes d'ordenació de metodesOrdenacioAutors.
- Mètodes
 - PanelAutors(): Constructora del panell de mostrar els resultats de les consultes d'autors.
 - createmagelcon(String path, String desc): Funció que retorna la imatge guardada a *path* amb descripció *desc*.

1.2.21 CtrlPresentacio

El Controlador de Presentació s'encarrega de fer de comunicador entre les vistes i panells de la Capa de Presentació. També és el que comunica la Capa de Presentació amb la Capa de Domini.

- Atributs
 - CtrlDomini cd: Instància del Controlador de Domini.
- Mètodes
 - iniciarPresentacio(): Mostra per pantalla la vista principal.
 - vistaEditor(): Mostra per pantalla la vista de l'edició del document actualment obert.
 - panelAltaDocument(): Retorna el PanelAltaDocument que s'encarrega de la funcionalitat de donar d'alta documents.
 - panellImportarDocument(): Retorna el PanellImportarDocument encarregat de la funcionalitat d'importar documents.

- `panelObrirDocument()`: Retorna el `PanelObrirDocument` encarregat de la funcionalitat d'obrir documents.
- `panelExportarDocument()`: Retorna el `PanelExportarDocument` encarregat de la funcionalitat d'exportar documents.
- `panelBaixaDocument()`: Retorna el `PanelBaixaDocument` encarregat de la funcionalitat de donar de baixa documents.
- `panelEditarTitolAutorDocument()`: Retorna el `PanelEditarTitolAutorDocument` encarregat de la funcionalitat d'editar el títol i l'autor de documents.
- `panelAfegirExpressio()`: Retorna el `PanelAfegirExpressio` encarregat de la funcionalitat d'afegir expressions booleanes.
- `panelModificarExpressio()`: Retorna el `PanelModificarExpressio` encarregat de la funcionalitat de modificar expressions booleanes.
- `panelEliminarExpressio()`: Retorna el `PanelEliminarExpressio` encarregat de la funcionalitat d'eliminar expressions booleanes.
- `panelConsultarExpressio()`: Retorna el `PanelConsultarExpressio` encarregat de la funcionalitat de consultar el contingut de les expressions booleanes donat el seu nom.
- `panelCercarDocAutor()`: Retorna el `PanelCercarDocAutor` encarregat de la funcionalitat de cercar els documents escrits per un autor.
- `panelCercarDocContingut()`: Retorna el `PanelCercarDocContingut` encarregat de la funcionalitat de consultar el contingut d'un document donat el seu títol i autor.
- `panelCercarDocSemblants()`: Retorna el `PanelCercarDocSemblants` encarregat de la funcionalitat de cercar els documents més semblants a un altre document.
- `panelCercarDocExp()`: Retorna el `PanelCercarDocExp` encarregat de la funcionalitat de cercar els documents que compleixen una expressió booleana.
- `panelCercarDocRellevants()`: Retorna el `PanelCercarDocRellevants` encarregat de la funcionalitat de cercar els documents més rellevants a una sèrie de paraules.
- `panelCercarAutors()`: Retorna el `PanelCercarAutors` encarregat de la funcionalitat de cercar els autors que comencen per un cert prefix.
- `panelDocuments(arr : ArrayList<ArrayList<String>>)`: Retorna el `PanelDocuments` encarregat de la funcionalitat de mostrar els resultats de les consultes de documents, emmagatzemat en *arr*, i de la seva ordenació.

- `panelAutors(arr : ArrayList<ArrayList<String>>)`: Retorna el `PanelAutors` encarregat de la funcionalitat de mostrar els resultats de les consultes d'autors, emmagatzemat en *arr*, i de la seva ordenació.
- `altaDocument(titol : String, autor : String, path : String)`:
- `importarDocument(pOrigen : String, pDesti : String)`:
- `obrirDocument(path : String)`: Obre el document amb el path donat i retorna la seva informació.
- `tancarDocument()`: Tanca el document actualment obert.
- `guardarDocument()`: Guarda el document actualment obert.
- `getDocumentObert()`: Obté la informació del document actualment obert.
- `exportarDocument(pOrigen : String, pDesti : String)`: Exporta el document amb path *pOrigen* i fa la transformació al format seleccionat a *pDesti*.
- `baixaDocument(path : String)`: Dona de baixa al document amb el path donat.
- `editarTitol(nouTitol : String, titolAnterior : String, autor : String)`: Canvia el títol del document (*titolAnterior*, *autor*) a *nouTitol*.
- `editarAutor(autorAnterior : String, nouAutor : String, titol : String)`: Canvia l'autor del document (*titol*, *autorAnterior*) a *nouAutor*.
- `editarContingut(nouContingut : String)`: Canvia el contingut del document actualment obert per *nouContingut*.
- `afegirExpressio(nom : String, exp : String)`: Afegeix una nova expressió amb nom *nom* i contingut *exp*.
- `modificarExpressio(nom : String, novaExp : String)`: Canvia el contingut de l'expressió *nom* per *novaExp*.
- `eliminarExpressio(nom : String)`: Elimina l'expressió amb nom *nom*.
- `consultarExpressio(nom : String)`: Consulta el contingut de l'expressió amb nom *nom*.
- `llistarExpressions()`: Retorna una llista amb el nom de totes les expressions.
- `llistarTitolsAutors(autor : String)`: Retorna una llista amb la informació de tots els documents escrits per l'autor *autor*.
- `llistarContingutDocumentPerAutorTitol(autor : String, titol : String)`: Retorna el contingut del document identificat per (*titol*, *autor*).
- `llistarKDocumentsSemblantsD(k : int, path : String, metode : int)`: Retorna una llista amb la informació dels *k* documents més semblants al document amb path *path* utilitzant el mètode seleccionat: TFIDF (0) o BM25 (1).
- `llistarTitolsPerExpressioGuardada(nom : String)`: Retorna una llista amb la informació dels documents que compleixen l'expressió booleana identificada pel nom *nom*.

- `l·listarTitolsPerExpressioNoGuardada(exp : String)`: Retorna una l·lista amb la informaci3 dels documents que compleixen l'expressi3 booleana amb contingut *exp*.
- `l·listarKDocumentsRellevantsPParaules(k : int, conj_paraules : String[], metode : int)`: Retorna una l·lista amb els *k* documents m3s rellevants al conjunt de paraules *conj_paraules* utilitzant el m3tode seleccionat TFIDF (0) o BM25 (1).
- `l·listarAutorsPerPrefix(prefix : String)`: Retorna una l·lista amb la informaci3 dels autors que tenen com a prefix *prefix*.
- `tancarAplicacio()`: Guarda tota la informaci3 per preservar-la en diferents sessions i tanca la vista principal.

2. Diagrama de Classes i Controlador de la Capa de Domini

2.1. Disseny del diagrama de model conceptual

A continuació mostrem el diagrama del model conceptual. Recomanem baixar el diagrama que es troba ubicat en el directori DOCS en format svg (DiagramaDomini.svg) o pdf (DiagramaDomini.pdf) per a poder veure-ho amb millor resolució.



2.2 Explicació de les Classes de Domini

2.2.1 Document

La classe Document és la classe que conté la representació de tot el que necessita un document dins del nostre projecte per poder dur a terme totes les funcionalitats descrites a l'enunciat del projecte.

- Atributs:
 - `titol`: indica el títol del document..
 - `autor`: indica l'autor del document.
 - `localització`: guarda el path del document.
 - `dataCreacio`: guarda la data en la que es va crear el document.
 - `dataDarreraModificacio`: guarda la darrera data en la que es va modificar el document.
 - `paraules`: map per guardar les posicions en les quals apareix una paraula.
 - `contingut`: array que guarda el contingut del document un cop s'ha fet el filtrat per tenir a cada posició una paraula (eliminem els espais).
 - `contingut_raw`: guardem el contingut sense cap tipus de filtre.
- Mètodes:
 - `Document(titol: String, autor: String, localitzacio: String, dataCreacio: LocalDate, dataDarreraModificacio: LocalDate, contingut: Map<String, HashSet<Integer>>, contingutRaw: String)`: creadora privada que serveix per fer clonacions d'una instància.
 - `Document(path: String, aut: String, tit: String)`: creadora pública que ens retorna una instància d'un document amb contingut buit.
 - `allmatches(s: String)`: retorna un arraylist amb les paraules de l'string, és a dir, lleva els espais.
 - `guardarPosicionsParaules(cont: ArrayList<String>)`: s'encarrega de calcular totes les posicions d'una paraula i guardar-ho a l'atribut `paraules`.
 - `canviarTitelDoc(nouTitel: String)`: canvia l'atribut títol del Document.
 - `canviarAutorDoc(nouAutor: String)`: canvia l'atribut autor del Document.
 - `modificarContingut(contin: String)`: modifica el contingut d'un Document.
 - `getTitel()`: retorna el títol.
 - `getAutor()`: retorna l'autor.
 - `getLocalitzacio()`: retorna la localització.
 - `getDataCreacio()`: retorna la data de creació.
 - `getDataDarreraModificacio()`: retorna la data de darrera modificació.

- `getContingut()`: retorna el contingut en forma de array, sense espais.
- `getContingutRaw()`: retorna el contingut en forma de string, amb espais.
- `conteParaula(p: String)`: retorna true en cas que el document contingui aquesta paraula, altrament retorna false.
- `getPosicionsParaula(paraula: String)`: retorna un hashset amb totes les posicions de les aparicions de la paraula.
- `clonar()`: servei per obtenir una nova instància que te exactament els mateixos atributs que la instància `this`.

2.2.2 CtrlDomini

El controlador de Domini s'encarrega d'instanciar la resta de controladors de la capa de domini i proporciona una interfície més pròxima l'usuari dels mètodes de la resta de controladors com el controlador d'índex o de Document.

- Atributs:
 - `ctrlOrdenació`: controlador encarregat d'ordenar els resultats
 - `ctrlIndex`: controlador encarregat de gestionar tot lo relacionat amb les consultes i modificació d'índex
 - `ctrlDocument`: controlador encarregat de gestionar tot l'apartat de Document.
- Mètodes:

El controlador de Domini fa delegacions a la resta de controladors. Però explicarem la funcionalitat que ha de fer en conjunt.

- `CtrlDomini()`: Instacia la resta de controladors
- `AltaDocument(autor : String, titol : String, path : String)`: Dona d'alta un document amb el valors donats
- `BaixaDocument(path : String)`: Borra el document amb el path donat
- `ObrirDocument(path : String)`: Obre el Document amb el path donat i retorna el contingut d'aquest
- `getDocumentObert(path: String)`: Dona una array de String que conté informació a mostrar a l'usuari del Document
- `TancarDocument()`: Tanca el Document Obert.
- `EditarTitol(nouTitol : String, titolAnterior : String, autor : String)`: Edita el Titol d'un Document i es fan les corresponents modificacions en els índexos
- `EditarAutor(autorAnterior : String, nouAutor : String, titol : String)`: Edita l'autor d'un document i es fan les corresponents modificacions en els índexos
- `EditarContingut(nouContingut : String)`: Edita el contingut del Document Obert i es fan les corresponents modificacions en els índexos

- LlistarAutorsPerPrefix(prefix : String) : Donat un prefix retorna informació com el nom i número de títols de tots els autors amb aquest prefix.
- LlistarKDocumentsSemblantsD(k : int, path : String, metode : int): Retorna informació com l'autor, títol, mida, etc. Dels K documents més semblants a un Document donat
- LlistarKDocumentsRellevantsPParaules(k:int,conj_paraules:TreeSet<String> , metode : int): Retorna informació com l'autor, títol, mida, etc. Dels K documents més semblants a un conjunt de paraules donat.
- LlistarTitolsAutors(autor : String): Llista tots els títols d'un autor
- LlistarTitolsPerExpressioGuardada(nom : String):Retorna informació com l'autor, títol, mida, etc. Dels Documents que compleixen el criteris de l'expressió guardada amb el nom donat
- LlistarTitolsPerExpressioNoGuardada(exp : String):Retorna informació com l'autor, títol, mida, etc. Dels Documents que compleixen el criteris de l'expressió donada
- LlistarContingutDocumentPerAutorTitol(autor : String, títol : String): Retorna el contingut del document que té com a autor i títol els paràmetres donats.
- AfegirExpressio(nom : String, exp : String): Afegeix una expressió booleana al gestor
- EliminarExpressio(nom : String): Elimina l'expressió guardada amb el nom donat.
- ModificarExpressio(nom : String, novaExp : String): Modifica l'expressió amb el nom donat amb la nova expressió donada
- ConsultarExpressio(nom : String) : Retorna l'expressió guardada amb aquell nom.
- ImportarDocument(pOrigen: String, pDesti: String): Importa el document amb path igual a pOrigen a un Document que pot entendre el Gestor ubicat a pDesti
- ExportarDocument(pathOriginal : String, pathDestí : String): Exporta el Document amb path igual a pathOriginal del gestor i fa la transformació a pathDestí.
- TancarAplicacio(): Fa la preservació de l'estat actual de l'aplicació abans de tancar l'aplicació.

2.2.3 CtrlDocument

El controlador de document s'encarrega de tota la gestió de les funcionalitats on s'ha de fer alguna manipulació amb els documents, això inclou la donada d'alta de Documents, Baixa de Documents i canviar el contingut d'aquest. També es comunica amb el controlador d'índex per poder introduir de manera correcta totes aquestes modificacions als índexs.

- Atributs:
 - DocumentObert: Representació d'un Document Obert
 - DocumentModificat: Representació del Document Obert que emmagatzema totes les modificacions
 - CapaPers: instància per fer la comunicació amb la capa de Persistència.
 - indexos: Instància del controlador d'Índex
- Mètodes:
 - CtrlDocument(index : CtrlIndex, capa : CtrlPersistencia): Instància el controlador i té com a atributs els paràmetres donats.
 - allmatches(s : String) : Separa la String donada en una array que representa cada paraula de la String
 - AltaDocument(autor : String, títol : String, path : String): Dona d'alta un document amb els valors donats
 - BaixaDocument(path : String): Borra el document amb el path donat
 - ObrirDocument(path : String): Obre el Document amb el path donat i retorna el contingut d'aquest
 - getDocumentObert(path: String): Dona una array de String que conté informació a mostrar a l'usuari del Document
 - TancarDocument(): Tanca el Document Obert i també elimina el Document Modificat si s'ha guardat
 - TancarDocumentForçat(): Tanca el Document Obert i també elimina el Document Modificat independentment de si s'ha guardat o no.
 - EditarTítol(nouTítol : String, títolAnterior : String, autor : String): Edita el Títol d'un Document i demana les corresponents modificacions a la resta de controladors
 - EditarAutor(autorAnterior : String, nouAutor : String, títol : String): Edita l'autor d'un document i demana les corresponents modificacions a la resta de controladors
 - EditarContingut(nouContingut : String): Edita el contingut del Document Obert

- GuardarDocument(): Sobreescriu el document Obert amb el Document Modificat i demana a la resta de controladors les corresponents modificacions.
- ImportarDocument(pOrigen: String, pDesti: String): Demana a la capa de persistència d'importar el document amb path igual a pOrigen a un Document que pot entendre el Gestor ubicat a pDesti
- ExportarDocument(pathOriginal : String, pathDestí : String): Demana a la capa de persistència d'exportar el Document amb path igual a pathOriginal del gestor i fa la transformació a pathDestí.

2.2.4 CtrlIndex

El controlador d'Index s'encarrega de totes les funcionalitats relacionades amb la consulta i modificació dels índexs.

- Atributs:
 - titols_autors: índex que emmagatzema els títols d'un autor
 - espaiVectorial: índex que fa la gestió de l'espai vectorial
 - prefixos_autors: índex per fer consultes sobre el nom d'autors
 - expressionsBooleans: índex que fa la gestió de totes les expressions booleans.
 - capaPers: instància per fer la comunicació amb la persistència
- Mètodes:
 - CtrlIndex(capa : CtrlPersistencia): Fa la instanciació dels índexs i té com a atribut el paràmetre donat.
 - getDocumentsAmbPatro(list : HashSet<String>, frase : ArrayList<String>): Retorna els paths de tots els documents que conté el paràmetre list que contenen la frase proporcionada.
 - allmatches(s : String) : Separa la String donada en una array que representa cada paraula de la String
 - getInterseccion(array : ArrayList<String>): Dona un conjunt de paths dels documents que contenen tots els elements de l'array.
 - getResultatDocument(d : Document): Es retorna una instància de ResultatDocument que conté tota l'informació a mostrar del Document d.
 - getResultatAutor(autor : String, titols : TreeMap<String, String>): Retorna una instància de ResultatAutr que conté tota l'informació a mostrar donat un autor i els seus documents.

- LlistarKDocumentsSemblants(k : int, path : String, metode : int): Retorna informació com l'autor, títol, mida, etc. Dels K documents més semblants a un Document donat
- LlistarTítolsAutors(autor : String): Llista tots els títols d'un autor
- LlistarDocumentsExpressioBooleana(nom : String):Retorna informació com l'autor, títol, mida, etc. Dels Documents que compleixen el criteris de l'expressió guardada amb el nom donat
- LlistarDocumentsExpressioBooleana(exp : String):Retorna informació com l'autor, títol, mida, etc. Dels Documents que compleixen el criteris de l'expressió donada
- LlistarContingutDocumentPerAutorTitol(autor : String, títol : String): Retorna el contingut del document que té com a autor i títol els paràmetres donats.
- AfegirExpressio(nom : String, exp : String): Afegeix una expressió booleana al gestor
- altaDocument(path : String): Afegeix el document guardat al path donat a tots els índexos
- baixaDocument(autor : String, títol : String, path : String): Elimina del índexos el Document donat
- existeixDocument(títol : String, autor : String): Dona un booleà per veure si existeix un document amb títol i autor donats
- modificarTitol(nouTitol : String, titolAnterior : String, autor : String): Fa la corresponent modificació del Document amb claus titolAnterior i autor amb el nou títol donat en els índexos.
- modificarAutor(autorAnterior : String, nouAutor : String, títol : String): Fa la corresponent modificació del Document amb claus autorAnterior i títol amb el nou nom d'autor donat en els índexos.
- modificarContingut(path : String, nouContingut : ArrayList<String>): Modifica en el índex espacial el Document amb guardat amb el parametre path i se li assigna aquest nouContingut en l'espai vectorial
- getTotsDocuments() : retorna el path de tots el Documents existents en el Gestor
- getDocumentsParaula(p : String): retorna el path de tots el Documents existents en el Gestor que contenen la paraula p.
- getDocumentsFrase(f : String): retorna el path de tots el Documents existents en el Gestor que contenen la frase f.
- EliminarExpressio(nom : String): Demana a l'índex corresponent l'eliminació de l'expressió guardada amb el nom donat.

- `ModificarExpressio(nom : String, novaExp : String)`: Demana a l'índex corresponent la modificació l'expressió amb el nom donat amb la nova expressió donada
- `ConsultarExpressio(nom : String)` : Demana a l'índex corresponent l'expressió guardada amb aquell nom i la retorna.

2.2.5 CtrlOrdenacioResultats

El controlador `CtrlOrdenacioResultats` és l'encarregat d'ordenar els resultats de les consultes de documents i d'autors segons uns certs criteris.

- Atributs
 - `ArrayList<ResultatAutor> rAutor`: `ArrayList` que guarda l'últim resultat de consulta d'autors, amb l'ordenació per defecte que retorna la consulta.
 - `ArrayList<ResultatDocument> rDocument`: `ArrayList` que guarda l'últim resultat de consulta de documents, amb l'ordenació per defecte que retorna la consulta.
- Mètodes
 - `CtrlOrdenacioResultats()`: Constructora per defecte.
 - `setResultatAutor(arr : ArrayList<ResultatAutor>)`: Estableix `rAutor` amb el valor de *arr*.
 - `setResultatDocument(arr : ArrayList<ResultatDocument>)`: Estableix `rDocument` amb el valor de *arr*.
 - `getResultatAutor()`: Retorna l'atribut `rAutor`.
 - `getResultatDocument()`: Retorna l'atribut `rDocument`.
 - `OrdenarAlfabeticamentPerAutor(arr : ArrayList<ResultatAutor>)`: Retorna *arr* amb els seus elements ordenats creixentment per ordre alfabètic.
 - `OrdenarPerNTitolsAutor(arr : ArrayList<ResultatAutor>)`: Retorna *arr* amb els seus elements ordenats decreixentment per número de títols escrits.
 - `OrdenarAlfabeticamentPerAutorTitol(arr : ArrayList<ResultatDocument>)`: Retorna *arr* amb els seus elements ordenats principalment per nom d'autor i secundàriament per títol.
 - `OrdenarAlfabeticamentPerTitolAutor(arr : ArrayList<ResultatDocument>)`: Retorna *arr* amb els seus elements ordenats principalment per títol i secundàriament per nom d'autor.
 - `OrdenarPerMidaDocument(arr : ArrayList<ResultatDocument>)`: Retorna *arr* amb els seus elements ordenats per mida.

- OrdenarPerDataCreacio(arr : ArrayList<ResultatDocument>): Retorna *arr* amb els seus elements ordenats segons la seva data de creació.

2.2.6 IndexAutors

La classe IndexAutors s'encarrega de gestionar les cerques d'autors per prefix. Emmagatzema els noms dels autors en una estructura de dades enfocada en la cerca per prefixos (TST), de tal manera que consultar autors és eficient tant en temps com espai. Internament, l'estructura de dades funciona com un arbre, i, per tant, parlarem dels seus elements com a nodes i del seu element principal com a node arrel.

- Atributs
 - NodeAutors root: Node arrel de l'arbre.
- Mètodes
 - IndexAutors(): Constructora per defecte. Crea un arbre buit amb root = null.
 - afegirAutor(autor : String): Afegeix *autor* a l'arbre.
 - eliminarAutor(autor : String): Elimina *autor* de l'arbre.
 - modificarAutor(autor1 : String, autor2 : String): Canvia *autor1* per *autor2* en l'arbre.
 - cercarAutors(prefix : String): Retorna una ArrayList amb els noms dels autors que tenen com a prefix el paràmetre *prefix*.
 - afegirAutorRec(n : NodeAutors, s : String, i : int): Mètode privat per afegir a l'autor *s* en l'arbre. *n* és el node actual que estem visitant i *i* la posició del caràcter de *s* actual.
 - eliminarAutorRec(n : NodeAutors, s : String, i : int): Mètode privat per eliminar a l'autor *s* de l'arbre. *n* és el node actual que estem visitant i *i* la posició del caràcter de *s* actual.
 - cercarAutorsRec(n : NodeAutors, s : String, i : int): Mètode privat per cercar als autors que tenen a *s* com a prefix en l'arbre. *n* és el node actual que estem visitant i *i* la posició del caràcter de *s* actual.
 - cercaExhaustiva(n : NodeAutors, s : String, arr : ArrayList<String>): Mètode privat per cercar als autors que tenen a *s* com a prefix en l'arbre. *n* és el node actual que estem visitant i *arr* és una ArrayList on anem guardant tots els autors que ens trobem que tinguin a *s* com a prefix.

2.2.7 IndexEspaiVectorial

Aquesta classe s'encarrega de representar l'espai vectorial al qual se li faran totes les consultes relacionades amb temes de similitud de documents i obtenir els documents que contenen una certa paraula. Utilitzar el model de l'espai vectorial es permet fer aquestes consultes d'una forma molt eficient.

- Atributs:
 - index: index que emmagatzema els vectors en les diferents representacions donat un path.
 - documents_contenen_paraula: map per tenir precalculat tots els documents que contenen una certa paraula.
 - documents_mes_sembants_a_document: map per tenir precalculada la similaritat dels documents entre si.
- Mètodes:
 - IndexEspaiVectorial(): constructora de la classe, retorna una instància de la classe IndexEspaiVectorial.
 - recalcularVectors(): serveix per actualitzar les coordenades de tots els vectors de l'index.
 - recalcularDocumentsSemblants(): serveix per recalcular i mantenir actualitzat l'atribut documents_mes_sembants_a_document, és a dir, les similaritats entre documents.
 - obtenirCosinus(v1: Vector_EV, v2: Vector_EV): calcula el cosinus que formen els dos vectors.
 - compararDocuments(path1: String, path2: String): retorna un double indicant el cosinus entre els dos documents, el cosinus ens indica la similaritat, a més, aquesta funció aprofita que els vectors son sparser.
 - crearDocument(): mètode per crear tots els vectors necessaris a l'hora d'afegir un nou document
 - extensioCorrecta(path: String): comprova que l'extensió d'un path sigui l'escollida per al projecte, en aquest cas “.ojmj”.
 - sortByValue(unsortMap: Map<String,Double>) retorna un map amb el contingut del map passat per paràmetre però ordenat decreixentment per valor.
 - afegirDocument(path: String, paraules: ArrayList<String>): serveix per afegir un Document i el seu contingut a l'índex.
 - eliminarDocument(path: String): serveix per eliminar un Document de la representació de l'espai vectorial.

- `modificarDocument(path: String, paraules: ArrayList<String>)`: modifica el contingut d'un document i els seus respectius vectors.
- `llistarKDocumentsSemblants(k: int, path: String, op: int)` retorna una llista amb els k documents similars al document indentificar per path.
- `llistarKDocumentsRellevants(k: int, paraules: Set<String>, op: int)` retorna una llista amb els k documents més rellevants per les paraules.
- `getAllDocuments()`: retorna una llista amb tots els paths documents que hi ha a l'índex.
- `getDocumentsContenenParauls(P: String)`: retorna una llista amb els paths de tots els documents que contenen la paraula P.

2.2.8 IndexExpressionsBooleanes

Aquesta classe té la funció de preservar totes les expressions booleanes, per poder fer les consultes per aquestes. Es guarden associades a un nom, i es permet fer la cerca d'una expressió guardada o no guardada.

- Atributs:
 - `Map<String, ExpressioBooleana> conjuntExp`: Estructura de dades que relaciona cada expressió booleana amb el seu nom.
- Mètodes:
 - `IndexExpressionsBooleanes()`: Creadora buida.
 - `IndexExpressionsBooleanes(index: CtrlIndex)`: Creadora amb parametre, per posar-lo de manera estatica dins les expressions booleanes.
 - `afegirExpressio(nom: String, exp: String)`: Afegeix una nova expressió booleana al conjunt d'expressions, amb nom: nom i contingut: exp.
 - `esborrarExpressio(nom: String)`: Esborra l'expressió amb nom: nom del conjunt d'expressions booleanes.
 - `modificarExpressio(nom: String, exp: String)` Modifica l'expressió amb nom: nom, canviant el seu contingut a exp.
 - `cercarExpressioPerNom(nom: String)`: Retorna la llista de documents que compleixen la expressió booleana guardada al conjunt d'expressions amb el nom: nom.
 - `cercaExpressio(exp: String)`: Retorna la llista de documents que compleixen la expressió booleana exp.
 - `consultaExpressio(nom: String)` Retorna el contingut de l'expressió booleana guardada a la colecció amb el nom: nom.

- `llistarExpressions()`: Retorna una llista amb els noms de totes les expressions guardades al conjunt d'expressions booleanes.
- `existeixNom(nom: String)` Retorna un boolea que indica si al conjunt d'expressions booleanes existeix alguna amb el nom: `nom`.
- `setControlador(index: CtrlIndex)`: Posa `index` com atribut estatic de les expressions booleanes.

2.2.9 IndexTitol

Aquesta classe té la funció de preservar tots els autors i títols d'aquest per poder fer les consultes dels títols d'autors. A més a més, per cada autor i títol s'adjunta la seva direcció de l'ordinador, per poder mostrar-ho l'usuari en diferents consultes.

- Atributs: `TreeMap<String, TreeMap<String, String>>` `Index`
- Mètodes:
 - `IndexTitol()`: Instancia tot l'índex
 - `afegirTitol(autor : String, titol : String, path : String)`: afegeix un nou Document a l'índex
 - `getTítols(autor : String)`: Mostra els títols d'un autor donat el seu nom
 - `getTitol (autor : String, titol : String)`: Dona el path corresponent a l'autor i títol donat.
 - `modificarAutor(autorAnterior : String, nouAutor : String, titol : String)`: Modifica el nom de l'autor d'un Document donat.
 - `modificarTitol(nouTitol : String, titolAnterior : String, autor : String)`: Modifica el títol d'un Document donat.
 - `eliminarAutor(autor : String)`: elimina l'autor donat de l'índex
 - `eliminarTitol(autor : String, titol : String)`: elimina el Document amb títol i autor com claus de l'índex.
 - `existeixAutor(autor : String)`: Retorna un booleà conforme existeix l'autor o no
 - `existeixDocument(titol : String, autor : String)`: Retorna un booleà conforme existeix el Document donat a l'índex.

2.2.10 ExpressioBooleana

ExpressioBooleana és la classe que s'encarrega de gestionar les consultes per expressió booleana. Per poder realitzar aquesta funcionalitat de manera eficient hem decidit usar un arbre binari com a estructura de dades.

- Atributs:
 - CtrlIndex cIndex: Controlador d'indexos, necessari per fer la consulta de documents que segueixin l'expressió booleana.
 - String stringExpression: Representació en string de l'expressió booleana definida a la mateixa classe.
 - Nodeexpressió root: Arrel de la representació en arbre binari de l'expressió booleana.
- Mètodes:
 - ExpressioBooleana(): Creadora buida.
 - ExpressioBooleana(exp: String): Creadora, inicialitza la classe actual per tal de representar l'expressió booleana exp.
 - crearExpressio(exp: String): Inicialitza la classe actual per tal de representar l'expressió booleana exp.
 - buildTree(father: NodeExpressio, s: String): Construeix per sota del node d'arbre binari father l'expressió booleana definida per s, de manera recursiva.
 - findHighestOperation(s: String): Retorna la posició del operador que té màxima preferència dins de l'expressió booleana definida a s.
 - convertCorchsToOps(s: String): Funció auxiliar, converteix una expressió booleana la qual es només formada per corxetes en l'equivalent a nivell de resultat amb operadors and, per tal de poder representar-se en l'arbre binari. Un exemple seria convertir "{a b c}" en "a & b & c".
 - getDocuments(): Retorna una llista amb tots els documents que compleixin l'expressió booleana definida a la classe actual.
 - getDocumentsRecursiu(n: NodeExpressio): Retorna tots els documents que compleixin l'expressió booleana definida per la representació en arbre binari que té com arrel n, de manera recursiva.
 - consultaExpressio(): Retorna la representació en string de l'expressió booleana definida a la classe.
 - setCtrlIndex(): Setter del controlador d'index de les expressions booleanes, per tal de poder accedir als documents.

2.2.11 NodeAutors

NodeAutors és una classe auxiliar de la classe IndexAutors. En ella s'emmagatzema tota la informació necessària per efectuar la cerca d'autors per prefix.

- Atributs
 - char data: Caràcter que representa el node.
 - boolean finalParaula: Booleà que indica si existeix un autor amb nom igual al camí des de l'arrel d'IndexAutors fins aquest node.
 - NodeAutors left: Node del fill esquerre del node actual.
 - NodeAutors mid: Node del fill del mig del node actual.
 - NodeAutors right: Node del fill dret del node actual.
- Mètodes
 - NodeAutors(): Constructora per defecte.
 - newNode(c : char): Crea un node amb data *c*, finalParaula *fals* i tres fills *nulls*.
 - getData(): Obté l'atribut data del node.
 - getFinalParaula(): Obté l'atribut finalParaula del node.
 - getLeft(): Obté el fill esquerre del node.
 - getMid(): Obté el fill del mig del node.
 - getRight(): Obté el fill dret del node.
 - setData(d : char): Estableix l'atribut data amb valor *d*.
 - setFinalParaula(b : boolean): Estableix l'atribut finalParaula amb valor *b*.
 - setLeft(n : NodeAutors): Estableix l'atribut left amb valor *n*.
 - setMid(n : NodeAutors): Estableix l'atribut mid amb valor *n*.
 - setRight(n : NodeAutors): Estableix l'atribut right amb valor *n*.

2.2.12 NodeExpressio

NodeExpressio és una classe auxiliar per la classe ExpressioBooleana. En ella s'emmagatzema la informació necessària per efectuar la cerca per expressions booleana.

- Atributs
 - NodeExpressio es: Fill esquerre del node actual.
 - NodeExpressio de: Fill dret del node actual.
 - String valor: En el cas de ser un node fulla, valor del predicat.
 - char op: Operació que dur a terme el node actual del arbre.
- Mètodes
 - NodeExpressio(): Creadora buida.
 - NodeExpressio(s: String): Creadora de node fulla, sent *s* el predicat definit al valor.

- setFillDret(d: NodeExpressio): Setter del fill dret.
- setFillEsquerre(e: NodeExpressio): Setter del fill esquerre.
- setValor(s: String): Setter del valor.
- setOp(c: char) Setter de l'operació definida al node actual.
- getFillDret(): Getter del fill dret.
- getFillEsquerre(): Getter del fill esquerre.
- getValor(): Getter del valor.
- getOp(): Getter de l'operació definida al node actual.

2.2.13 Vector_EV

És una classe abstracta que ens permet representar un document amb el model espai vectorial. Aquesta classe, com és abstracta, té per mètodes el mínim nombre de funcions que ha de tenir qualsevol estratègia de representació dels vectors, que es vulgui implementar.

- Atributs:
 - coordenades: map que guarda parelles on donada una paraula la relacionem amb la seva coordenada.
 - norma: emmagatzema la norma del vector.
 - stopWords: paraules que no hem de considerar a l'hora de representar l'espai vectorial.
- Mètodes:
 - Vector_EV(): constructora de la classe, instància un objecte de la classe Vector_EV.
 - calcular_norma(): calcula i actualitza la norma del vector.
 - calcular_coordenades(paraula: String): retorna la coordenada que li pertoca a una paraula.
 - afegirParaula(paraula: String): afegeix la paraula al vector que representa el document.
 - modificarCoordenada(paraula: String, coord: double): modifica la coordenada d'una paraula.
 - recalcularTotesCoordenades(): mètode per actualitzar totes les coordenades de les paraules i recalcular la norma.
 - norma(): retorna la norma del vector.

2.2.14 Vector_TFIDF

Aquesta classe hereda de Vector_EV, i implementa tots els mètodes necessaris per implementar l'estratègia de pesos del TFIDF (representació obligatòria del projecte).

- Atributs:
 - aparicions_paraula: map que guarda el número d'aparicions d'una paraula.
 - paraules_totals_document: guarda el número de paraules que té el document.
 - n_documents: guarda el número total de documents que hi ha
 - n_documents_contenen_paraula: map que guarda el número de documents que contenen una certa paraula.
- Mètodes:
 - Vector_TFIDF(num: int): constructora de la classe, instància un objecte de la classe Vector_TFIDF.
 - calcular_coordenades(paraula: String): retorna la coordenada que li pertoca a una paraula.
 - afegirParaula(paraula: String): afegeix la paraula al vector que representa el document.
 - conteParaula(paraula: String): retorna un boolea indicant si el vector conte la paraula.
 - recalcularTotesCoordenades(): metode par actualitzar totes les coordenades de les paraules i recalcular la normal.
 - destructora():actualitza els atributs estàtics necessaris pel correcte funcionament dels vectors abans de que l'instancia quedi desreferenciada completament.

2.2.15 Vector_BM25

Aquesta classe hereda de Vector_EV, i implementa tots els mètodes necessaris per implementar l'estratègia de pesos del BM25 (representació opcional del projecte).

- Atributs:
 - aparicions_paraula: map que guarda el número d'aparicions d'una paraula.
 - paraules_totals_document: guarda el número de paraules que té el document.
 - n_documents: guarda el número total de documents que hi ha
 - n_documents_contenen_paraula: map que guarda el número de documents que contenen una certa paraula.

- paraules_totals_n_documents: guarda el número de paraules totals que hi ha entre tots els documents.
- k i b: constants necessàries pels càlculs del BM25.
- Mètodes:
 - Vector_BM25(num: int): constructora de la classe, instància un objecte de la classe Vector_BM25.
 - calcular_coordenades(paraula: String): retorna la coordenada que li pertoca a una paraula.
 - afegirParaula(paraula: String): afegeix la paraula al vector que representa el document.
 - conteParaula(paraula: String): retorna un boolea indicant si el vector conte la paraula.
 - recalculerTotesCoordenades(): metode per actualitzar totes les coordenades de les paraules i recalculer la normal.
 - destructora():actualitza els atributs estàtics necessaris pel correcte funcionament dels vectors abans de que l'instancia quedi desreferenciada completament.

2.2.16 Resultat

Resultat és una classe que s'utilitza per a poder fer l'ordenació dels resultats de consultes de documents i autors segons uns certs criteris. Té dues subclasses: ResultatAutor i ResultatDocument, on cadascuna d'elles conté la informació necessària per a poder realitzar les seves respectives consultes.

- Atributs
 - ArrayList<String> arr: ArrayList on guardarem la informació d'un resultat de consulta, ja sigui la informació d'un document o d'un autor.
- Mètodes
 - Resultat(): Constructora per defecte.
 - getResultat(): Retorna l'atribut arr.
 - setResultat(c : ArrayList<String>): Estableix l'atribut arr amb el valor de c.

2.2.17 ResultatAutor

Subclasse de la superclasse Resultat per emmagatzemar la informació dels resultats de les consultes d'autors. Emmagatzema en l'atribut arr de Resultat el nom de l'autor i el nombre de títols que aquest ha escrit.

- Mètodes
 - ResultatAutor(): Constructora per defecte.
 - getAutor(): Retorna el nom de l'autor.
 - getNTitols(): Retorna el nombre de títols que ha escrit l'autor.
 - setAutor(s : String): Setter del nom de l'autor.
 - setNTitols(x : int): Setter del nombre de títols que ha escrit l'autor.

2.2.18 ResultatDocument

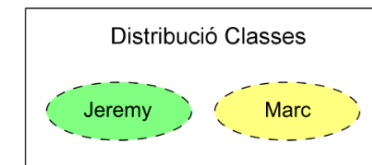
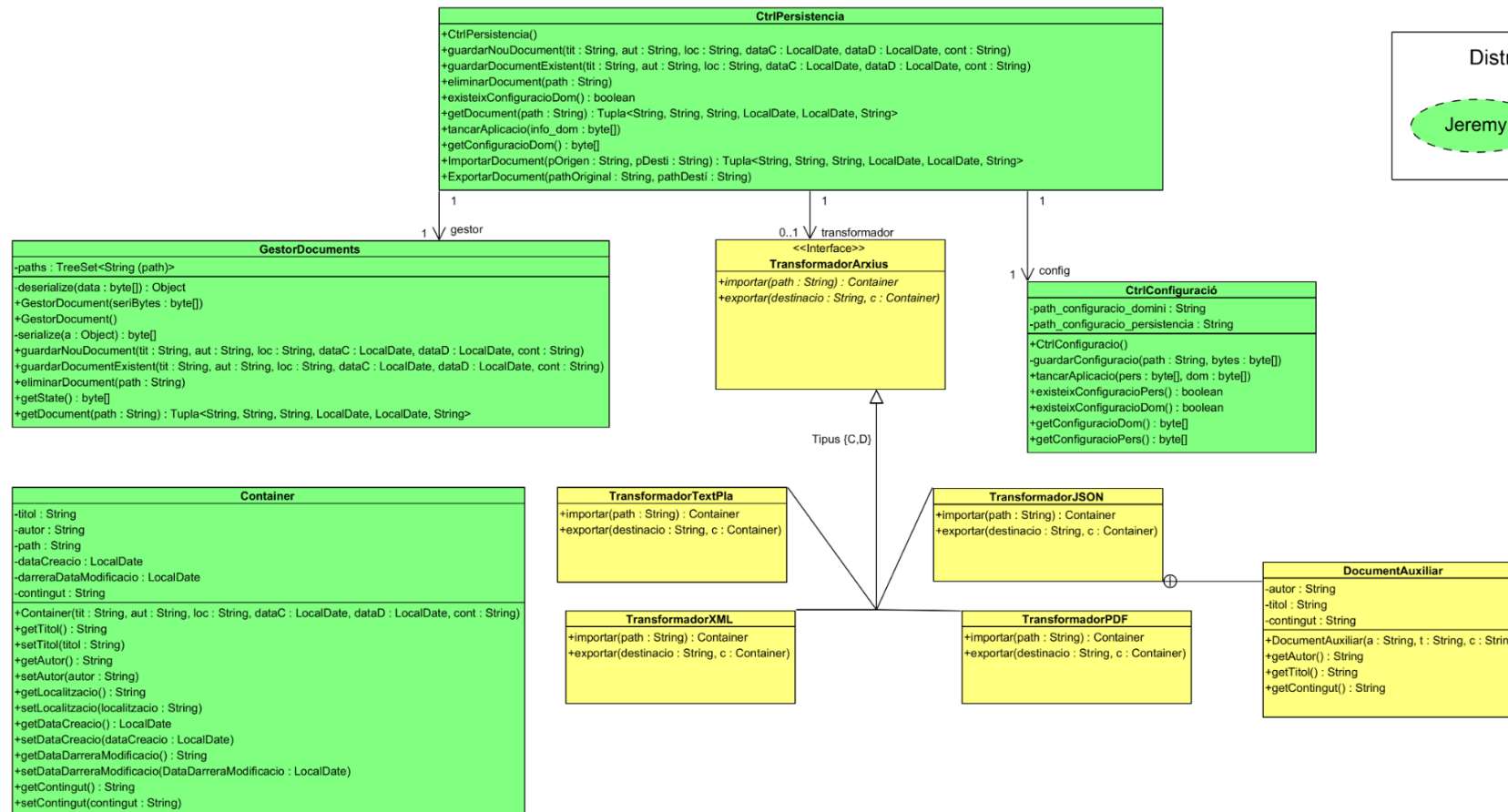
Subclasse de la superclasse Resultat per emmagatzemar la informació dels resultats de les consultes de documents. Emmagatzema en l'atribut arr de Resultat el títol del document, el nom del seu autor, el path, la mida i la seva data de creació.

- Mètodes
 - ResultatDocument(): Constructora per defecte.
 - getAutor(): Retorna el nom de l'autor del document.
 - getTitol(): Retorna el títol del document.
 - getPath(): Retorna el path del document.
 - getMida(): Retorna la mida del document.
 - getDataCreacio(): Retorna la data de creació del document.
 - setAutor(s : String): Setter del nom de l'autor del document.
 - setTitol(s : String): Setter del títol del document.
 - setPath(s : String): Setter del path del document.
 - setMida(m : int): Setter de la mida del document.
 - setDataCreacio(data : LocalDate): Setter de la data de creació del document.

3. Diagrama de Classes i Controlador de la Capa de Persistència

3.1 Disseny del diagrama del model conceptual

A continuació mostrem el Diagrama de Classes de la Capa de Persistència. Recomanem baixar el diagrama que es troba ubicat en el directori DOCS en format svg (DiagramaPersistencia.svg) o pdf (DiagramaPersistencia.pdf) per a poder veure-ho amb millor resolució.



3.2 Explicació de les Classes de Persistència

3.2.1 CtrlPersistencia

Aquesta classe és l'encarregada d'instanciar tota la resta de controladors d'aquesta capa. A més a més, dona una interfície més pròxima a la capa de Domini per poder fer totes les funcions de preservació dels documents i de tota la configuració del Domini en un moment concret.

- Atributs:
 - gestor: Gestor de Documents encarregat de la preservació i modificació de tots els documents del gestor.
 - config: Controlador de Configuració encarregat de la preservació de i donar la configuració de tota l'aplicació.
 - transformador: Transformador d'arxius encarregat de l'importació i exportació de documents.
- Mètodes:
 - CtrlPersistencia(): Instancia la resta de controladors.
 - guardarNouDocument(tit : String, aut : String, loc : String, dataC : LocalDate, dataD : LocalDate, cont : String): Es demana al gestor de documents que faci el guardat d'un nou document amb els parametres donats.
 - guardarDocumentExistent(tit : String, aut : String, loc : String, dataC : LocalDate, dataD : LocalDate, cont : String): Es demana al gestor de documents que faci el guardat d'un document existents amb els parametres donats.
 - eliminarDocument(path : String): Demana al gestor de Documents que es faci la eliminació del document amb el path donat
 - existeixConfiguracioDom(): Retorna si existeix una configuració del Domini guardada pel controlador de configuració
 - getDocument(path : String): Demana al gestorDocuments una tupla que conté tota la informació del document guardat en el path donat i la retorna.
 - tancarAplicacio(info_dom : byte[]): Demana al Controlador de configuració fer la preservació de la informació de domini donada i també del gestor de Documents.
 - getConfiguracioDom() retorna una array de bytes que correspon a la configuració del domini.

- ImportarDocument(pOrigen: String, pDesti: String): Demana al transformador l'importació del document amb path igual a pOrigen a un Document que pot entendre el Gestor ubicat a pDesti
- ExportarDocument(pathOriginal : String, pathDestí : String): Demana al transformador d'arxius l'exportació el Document amb path igual a pathOriginal del gestor i fa la transformació a pathDestí.

3.2.2 GestorDocument

Aquesta classe és l'encarregada de totes les funcionalitats de gestió i preservació dels Documents que es creen en el gestor.

- Atributs:
 - paths: TreeSet<String>: Set que conté tots els paths dels documents presents al gestor
- Mètodes:
 - deserialize(data : byte[]): Donat un array de bytes que codifica un objecte, fa la transformació a objecte
 - GestorDocument(seriBytes : byte[]): Instancia un GestorDocuments que té una configuració propia.
 - GestorDocument(): Instancia un GestorDocuments amb un treeSet buit.
 - serialize(a : Object): Donat un objecte serialitzable, dona la seva representació en una array de bytes
 - guardarNouDocument(tit : String, aut : String, loc : String, dataC : LocalDate, dataD : LocalDate, cont : String): Fa el guardat a disc d'un nou document amb els parametres donats.
 - guardarDocumentExistent(tit : String, aut : String, loc : String, dataC : LocalDate, dataD : LocalDate, cont : String): Fa el guardat a disc d'un document existents amb els parametres donats.
 - eliminarDocument(path : String): Elimina el document amb el path donat del disc.
 - getState(): Retorna la representació en array de bytes de l'atribut paths
 - getDocument(path : String): Retorna una tupla que conté tota la informació del document guardat en el path donat.

3.2.3 CtrlConfiguracio

Aquesta classe és l'encarregada de la gestió i preservació de tota la configuració de l'aplicació (tant de la capa de Domini, com la pròpia capa de Persistència).

- Atributs:
 - path_configuracio_domini: String que representa el path relatiu on es guardarà la configuració del domini
 - path_configuracio_pers: String que representa el path relatiu on es guardarà la configuració de la persistència
- Mètodes:
 - CtrlConfiguracio(): Instancia un controlador de Configuració
 - guardarConfiguracio(path : String, bytes : byte[]): Guarda el bytes donats com a parametres en un fitxer ubicat en el path.
 - tancarAplicacio(pers : byte[], dom : byte[]): Guarda pers i dom en fitxers ubicats en path_configuracio_domini i path_configuracio_pers respectivament.
 - existeixConfiguracioPers() : Dona un booleà indicant si existeix o no un configuració de la persistència.
 - existeixConfiguracioDom(): Dona un booleà indicant si existeix o no un configuració de la capa de Domini.
 - getConfiguracioDom(): Retorna un array de bytes que correspon a la configuració del domini.
 - getConfiguracioPers(): Retorna un array de bytes que correspon a la configuració del domini.

3.2.4 Container

Aquesta classe té dos objectius. El primer objectiu és donar una classe que ens permeti poder fer la serialització de tota la informació relacionada amb els documents. El segon objectiu és el pas d'informació entre el GestorDocuments i TransformadorArxius a l'hora de fer les importacions i exportacions de Documents.

- Atributs:
 - titol: String que representa el Títol d'un Document
 - autor: String que representa l'autor d'un Document
 - path: String que representa el path on es guarda el Document
 - dataCreació: LocalDate que representa la data de creació del document.
 - darreraDataModificacio: LocalDate que representa la data de darrera data de modificació del document.

- contigut: String que representa el contingut del document:
- Mètodes:
 - Container(tit : String, aut : String, loc : String, dataC : LocalDate, dataD : LocalDate, cont : String): Crea un objecte que té com a atributs, els parametres donats.
 - Com que aquesta classe es un container, únicament té getters i setters dels atributs.

3.2.5 TransformadorArxius

És una interfície que ens permetrà implementar qualsevol tipus d'exportació o importació. Per fer-ho simplement haurem de fer una classe que hereti d'aquesta interfície i l'implementi.

- Mètodes:
 - importar(path: String): retorna un Container amb les dades de l'arxiu a importar.
 - exportar(destinacio: String, c: Container): crea un arxiu del tipus que sigui l'exportació amb les dades del Container.

3.2.6 TransformadorTextPla

Implementa la interfície TransformadorArxius per tal de poder importar y exportar arxius .ojmj a .txt. Per fer-ho s'ha utilitzat la llibreria nativa de Java, que ens permet fer crides al sistema operatiu (java.io) per escriure i llegir arxius.

3.2.7 TransformadorXML

Implementa la interfície TransformadorArxius per tal de poder importar y exportar arxius .ojmj a .xml. Per fer-ho s'han utilitzat les llibreries natives de Java javax.xml y org.w3c.dom.

3.2.8 TransformadorPDF

Implementa la interfície TransformadorArxius per tal de poder importar y exportar arxius .ojmj a .pdf. Per fer-ho s'ha utilitzat la llibreria externa iTextPDF5.

3.2.9 Transformador JSON

Implementa la interfície TransformadorArxius per tal de poder importar y exportar arxius .ojmj a .json. Per fer-ho s'ha utilitzat la llibreria externa Gson. Per implementar-la hem hagut

d'utilitzar una classe auxiliar niada per poder permetre el pas d'informació a la llibreria, ja que aquesta necessita un objecte que contingui la informació.

4. Diagrama de Classes i Controlador de la Capa Transversal

4.1 Disseny del diagrama del model conceptual

A continuació mostrem el Diagrama de Classes de la Capa Transversal. Recomanem baixar el diagrama que es troba ubicat en el directori DOCS en format svg (DiagramaTransversal.svg) o pdf (DiagramaTransversal.pdf) per a poder veure-ho amb millor resolució.

Tupla
-first : T1 -second : T2 -third : T3 -fourth : T4 -fifth : T5 -sixth : T6
+Tupla() +Tupla(t1 : T1, t2 : T2, t3 : T3, t4 : T4, t5 : T5, t6 : T6) +setFirst(first : T1) +getFirst() : T1 +setSecond(second : T2) +getSecond() : T2 +setThird(third : T3) +getThird() : T3 +setFourth(fourth : T4) +getFourth() : T4 +setFifth(fifth : T5) +getFifth() : T5 +setSixth(sixth : T6) +getSixth() : T6

Distribució de les Classes

Jeremy

4.2 Explicació de les Classes de la Capa Transversal

4.2.1 Tupla

Aquesta classe serveix per fer el pas de tota la informació necessària que conforma un Document entre la capa de Persistència i la capa de Domini. Utilitza tipus genèrics amb l'objectiu de ser una estructura que en un futur es pugui estendre a més funcions que requereixin emmagatzemar altres tipus d'atributs.

- Atributs:
 - First: Variable genèrica (T1)
 - Second: Variable genèrica (T2)
 - Third: Variable genèrica (T3)
 - Fourth: Variable genèrica (T4)
 - Fifth: Variable genèrica (T5)
 - Sixth: Variable genèrica (T6)
- Mètodes:
 - Tupla(): Crea una tupla buïda
 - Tupla(t1 : T1, t2 : T2, t3 : T3, t4 : T4, t5 : T5, t6 : T6) : Crea una tupla que té com a atributs els parametres introduïts.
 - Com aquesta classe es per fer el pass d'informació de persistència a Domini, sols conté getters i setters del atributs.

5.1 Estructures de dades

5.1 Capa de Domini

5.1.1 Document

El document és l'estructura que té tota la informació relacionada amb un Document. Aquesta classe té principalment 3 atributs importants:

- L'autor del Document
- El títol del Document
- El contingut del Document

En el nostre cas, el principal problema que vam tenir era trobar una estructura de dades que fos relativament assequible de modificar i que no ocupi una gran quantitat de memòria. Tot això per guardar el contingut del Document.

Per tant, vam pensar en aquestes 3 alternatives:

- **Path del Document associat en el sistema operatiu**

Aquesta tècnica consisteix en l'operació directament amb el document del sistema operatiu que conté tota la informació associada a un document del gestor. D'aquesta manera, en el gestor no teníem un document que té un pes molt gran en memòria. Únicament teníem en memòria la seva direcció en el sistema operatiu (path).

El principal problema d'aquesta tècnica era que trenca el principi de 3 capes, on cada capa és independent de l'altre, en aquest cas la capa de Domini sap com s'està guardant en el sistema els documents. A més a més, tractar amb el contingut és difícil, ja que no és una estructura dinàmica. És a dir, si volguéssim introduir una nova paraula entremig del nostre contingut actual, primer hauríem de moure tot el contingut posterior a aquesta nova paraula i després introduir aquesta nova paraula. Per tant, aquest mètode no és el millor, encara que el cost en memòria sigui de les millors opcions.

- **String que conté tot el contingut.**

El principal problema d'aquesta estructura és un dels mencionats anteriorment. Aquest és la modificació quan hi ha paraules a afegir o eliminar. Per tant, vam optar que en comptes de modificar una string. Sempre que l'usuari vulgui fer una modificació del contingut, sempre ens doni una string amb el contingut total.

- **ArrayList de Strings amb un atribut amb el contingut cru (raw)**

Aquesta és l'estructura que hem utilitzat com a representació del contingut d'un document. Usarem també com a tècnica de modificació del contingut, la mencionada

anteriorment. La raó és degut al fet que d'aquesta manera, el tractament del contingut per part de l'índex de l'espai vectorial es pot fer de manera òptima.

5.1.2 IndexAutors

IndexAutors és la classe que s'encarrega de gestionar les consultes d'autors per prefix. Per poder realitzar aquesta funcionalitat de manera eficient hem decidit utilitzar un TST com a estructura de dades.

Un TST és una estructura de dades en forma d'arbre ternari de cerca. La idea bàsica del TST és muntar un arbre de manera que cada node representa un prefix dels seus fills. D'aquesta manera es poden emmagatzemar de manera eficient paraules (com strings).

Característiques dels TST

- Permeten operacions eficients per prefixos, a diferència d'altres estructures de dades.
- Són flexibles, permetent afegir i esborrar dades guardades.
- Poden emmagatzemar de forma molt eficient en memòria i temps paraules o seqüències que comparteixen prefixos.

Funcionament

Ja que en el projecte hem implementat un arbre de cerca ternari (o TST per a abreviar), explicarem el comportament d'aquest.

L'arbre estarà compost per diferents nodes. Cadascun amb els següents atributs:

- La dada a emmagatzemar (per exemple, un caràcter de la paraula).
- Un punter a cadascun dels seus 3 nodes fills (denominats left, mid i right).
- Un booleà que indica si el node marca el final d'una paraula.

La manera de guardar paraules dins de l'arbre serà mitjançant els nodes centrals o mid.

Així doncs, només estarem adquirint informació addicional sobre la seqüència quan avancem pel fill central en el recorregut de l'arbre. Si en canvi avancem pels fills left i right no haurem afegit informació sinó que haurem passat a fer la cerca en un dels subàrbres dels fills.

TST vs. altres alternatives

Anteriorment em mencionat que em decidit utilitzar com a estructura de dades un TST perquè era eficient en memòria i espai. En aquest apartat posarem en comparació el TST

amb altres estructures de dades i veurem que, en efecte, la nostra decisió de triar el TST ha estat la correcta comparant-la amb les altres alternatives.

Alternativa 1: Trie

La idea general d'un trie és tenir un arbre on cada node té exactament tants fills com lletres té l'abecedari de l'alfabet. Per tant la cerca en un trie consisteix en anar recorrent l'arbre seguint el camí dels nodes que formen la paraula. Per cada node accedirem al node fill que correspon amb la següent lletra de la paraula, i aquest procediment es repeteix tantes vegades com lletres tingui la paraula (k). Cost en temps: $O(k)$.

El problema dels trie és que per cada node hem de tenir tants fills com lletres tingui l'abecedari, la qual cosa malgasta molta memòria ja que probablement molts d'aquests nodes mai s'utilitzaran i només estan desperdiciant memòria.

Alternativa 2: TST

El TST es comporta d'una manera molt similar al trie, amb la principal diferència que cada node tindrà com a màxim tres fills. Això fa que el gran problema en memòria que tenien els tries desapareixi amb la implementació del TST.

En canvi hi ha una petita pèrdua en temps a l'hora de cercar en un TST, ja que ara no ens és suficient visitar $O(k)$ nodes per trobar la paraula, sinó que és molt probable que n'haguem de visitar unes de més. Això és degut a que ara, a l'igual que abans, hem de visitar $O(k)$ vegades nodes mid de l'arbre per avançar en la paraula, però també ens serà necessari avançar per nodes left i right en certes ocasions, i això fa que el cost en temps s'incrementi amb el nombre de vegades que haguem de visitar aquests 2 nodes: $O(\log(n))$ on n és el nombre d'autors guardats en el TST.

Per tant el cost en cas mitjà d'un TST és $O(\log(n) + k)$ a diferència de $O(k)$ en el cas del trie.

Conclusió

Encara que tenim una petita pèrdua en temps a l'utilitzar el TST (suposant que no tenim una gran quantitat d'autors en el nostre gestor), ens és molt més preferible utilitzar el TST sobre el trie a causa de la seva gran millora en espai. És per aquest motiu que em decidit utilitzar el TST.

Cost

Creació

Crear un TST solament implica crear un node arrel buit, amb cost $O(1)$.

Cerca

Quan busquem en un arbre ternari podem diferenciar dues operacions: Primer, recorrem k (on k és la grandària en caràcters de la nostra paraula o seqüència) vegades els fills centrals de nodes de l'arbre per a formar la paraula en qüestió. Segon, fem cerques recursives en els subàrbres esquerre i dret de cada node (per a trobar el prefix que necessitem) amb cost logarítmic en funció del nombre d'autors guardats n .

Per tant, el cost d'aquesta operació serà $O(\log(n) + k)$ en arbres degudament equilibrats.

Inserció

El procés d'inserció és molt semblant al de cerca, ja que simplement es fa un recorregut de l'arbre com s'ha explicat anteriorment i s'assumeix que la paraula ja es troba dins d'aquest. Si tots els nodes ja existeixen de manera que es pugui fer la paraula, es marca l'últim com a final de paraula. En cas contrari, es creen nous nodes a mesura que es vagin necessitant (amb cost $O(1)$) fins a formar l'estructura de nodes requerida per a representar la paraula.

Per tant, el cost serà $O(\log(n) + k)$ en cas mitjà i $O(n + k)$ en cas pitjor.

Esborrat

De la mateixa forma que la inserció, l'esborrat també depèn fortament de l'operació de cerca, perquè esborrar un element de l'arbre suposa buscar si està i la seva posició i després processar qualsevol d'aquests 4 possibles casos:

- La paraula no es troba en l'arbre: No es fan canvis en aquest. Cost $O(1)$
- La paraula es troba en l'arbre i és prefix d'altres paraules: Simplement es posa a fals l'atribut que marca final de paraula. Cost $O(1)$
- La paraula es troba en l'arbre i té altres paraules com a prefix: S'eliminen recursivament els nodes que formen la paraula, començant pel final fins a arribar a un node final del prefix o esgotar el subàrbre que la conté. Cost $O(k)$
- La paraula es troba en l'arbre i no és prefix ni té prefixos: S'eliminen recursivament tots els seus nodes, començant pel final fins a esgotar el subàrbre que la conté. Cost $O(k)$.

Com que els casos a processar tenen cost menor al de cerca i es fan seqüencialment després d'executar-la, el cost final és el de fer una cerca: $O(\log(n) + k)$ en cas mitjà i $O(n + k)$ en cas pitjor.

5.1.3 IndexTitol

IndexTitol és la classe que s'encarrega de gestionar les consultes de contingut donat un autor i un títol i donar tots els títols d'un autor. Per tant, hem optat per utilitzar un `TreeMap<String, TreeMap<String, String>>` com que d'aquesta manera ens assegurem que totes les consultes siguin **$O(\log(n))$** sent n el nombre d'autors que hi ha en l'índex.

5.1.4 IndexEspaiVectorial

IndexEspaiVectorial és la classe que ens permet representar documents dins d'un espai vectorial i fer diferents operacions que ens són interessants per saber per exemple, d'una forma relativament eficient, el grau de coincidència d'un document a un altre o d'una query als documents. Els atributs d'aquesta classe són `índex`, que és un `map<String, ArrayList<Vector_EV>>` que ens permet obtenir en un temps $O(\log(n))$ els vectors en les diferents representacions d'un document. També guardem a un `map<String, ArrayList<String>>` els documents que contenen una paraula, per poder fer les cerques amb temps $O(\log(n))$. També tenim precalculat a l'atribut `documents més semblants a document` els documents més semblants a un document com el seu nom indica. D'aquesta manera els podem obtenir en $O(\log(n))$ també.

Hem decidit utilitzar aquestes estructures de dades ja que són molt eficients, a més perquè hem preferit tenir-ho tot precalculat que calcular-ho sota demanda.

5.1.5 IndexExpressioBooleana

IndexExpressioBooleana és la classe que ens permet guardar les representacions de les expressions booleanes. Per tal de fer això, disposem d'un índex, utilitzant l'estructura de dades `HashMap`.

Funcionament

Aquest índex relaciona el nom de les expressions booleanes amb la seva representació, és a dir, un string amb una instància de la classe `ExpressioBooleana`.

Justificació de l'elecció de l'ED

Hem decidit implementar l'índex amb aquesta ED, ja que suposa una complexitat temporal molt petita per totes les operacions que necessitem efectuar.

Complexitat

Les operacions que es requereixen del l'índex són: afegir una expressió, modificar-la, esborrar-la, consultar-la o cercar per expressió (ja sigui donant aquesta directament o donant el nom d'una guardada).

Afegir

Afegir en un HashMap té un cost mitjà de $O(1)$, pese a que en el pitjor cas es $O(k)$, sigui k la quantitat d'elements del HashMap. També s'ha de tenir en compte el temps per crear la representació de l'expressió booleana, estudiat al apartat ExpressioBooleana, que es $O(n)$, sigui n la longitud de la string que representa l'expressió booleana, i per tant el cost d'afegir es $O(n)$ en mitjà i $O(n*k)$ en pitjor cas.

Modificar

Modificar suposa el mateix cost que afegir, deut al funcionament intern de les classes, i per tant en temps mitjà es $O(n)$ i en en el pitjor cas $O(n*k)$.

Esborrar

Esborrar suposa un cost de $O(1)$ en temps mitjà, i $O(k)$ en el pitjor cas.

Consultar

Consultar suposa un cost de $O(1)$ en temps mitjà, i $O(k)$ en el pitjor cas.

Cercar

En la cerca, es requereix distingir dos casos: cerca donant l'expressió i cerca donant el nom d'una guardada. En ambdós casos caldrà una cerca, amb la complexitat documentada a ExpressioBooleana sent $O(n*m)$, prenent m com la quantitat de documents que hi han.

- Donant l'expressió: Hem de primer crear l'expressió (temps $O(n)$) i després cercar-la ($O(n*m)$). Per tant el cost total seria $O(n^2*m)$.
- Donant el nom: Hem de primer buscar l'expressió (consultar-la, $O(1)$ en temps mitjà, $O(k)$ en pitjor cas) i després cercar-la ($O(n*m)$). Per tant, el cost total serà $O(n*m)$ en cas mitjà i $O(k*m*n)$ en pitjor cas.

5.1.6 ExpressioBooleana

ExpressioBooleana és la classe que s'encarrega de gestionar les consultes per expressió booleana. Per poder realitzar aquesta funcionalitat de manera eficient hem decidit utilitzar un Arbre Binari com a estructura de dades. Ens guardem l'arrel de dit arbre a la classe com atribut.

Funcionament

L'arbre estarà compost per diferents nodes. Cadascun amb els següents atributs:

- Un punter a cadascun dels seus 2 nodes fills (esquerre i dret).
- Una string que, en cas de que el node actual sigui fulla, conté un predicat.
- Un char que indica de quin tipus és el node:
 - 0: Node fulla i valor es el predicat.
 - “: Node fulla i valor es una frase a buscar
 - !: Node operació '!', només el fill dret té valor.
 - & o |: Node operació '&' o '|', ambdòs fills tenen valor.

Llavors a l'hora de crear l'arbre, busquem l'operació amb màxima precedència i actuem d'acord amb quina és aquesta, donant valor al tipus de node i si cal, donat valor a l'atribut valor i/o cridant recursivament a la funció pel sub-arbre dret i esquerre.

Justificació d'elecció de la ED

Hem escollit aquesta estructura de dades com que és una representació completa, i permet realitzar fàcilment les consultes sub-partint el problema.

Complexitat

Com que l'ED per representar una constant, l'expressió booleana definida per la classe, només tenen sentit les operacions de creació i cerca, ja que les altres es gestionen des de l'índex, sobre les instàncies d'aquesta classe.

Creació

Segui n el mida de la string que representa l'expressió booleana, el cost de transformar-la en arbre es $O(n)$

Cerca

Quan busquem tots els documents que compleixen expressió booleana definida per la classe, hem de fer com a molt $O(n)$ crides a subarbres. Tot i això, l'operació més costosa és la intersecció. Hem utilitzat un HashSet per anar "pujant" recursivament els resultats per l'arbre, i el cost d'intersecció amb HashSet és $O(m)$, sigui m la quantitat de documents que hi ha. Per tant, el cost de cerca és $O(n*m)$, però en la majoria de casos serà molt millor.

5.1.7 Vector_EV

El Vector_EV conté el mínim nombre de mètodes que necessitem per representar un document en el model d'espai vectorial. Cada herència d'aquesta classe implementa la seva estratègia de pesos. Utilitzant aquesta classe abstracta ens és molt fàcil afegir noves estratègies de pesos al nostre projecte. Només caldria fer una herència i implementar els diferents mètodes que necessiti. Els atributs que conté són un `map<String,Double>` on hi haurà les coordenades de cada paraula i podrem consultar la coordenada d'una paraula en temps $O(\log n)$. També té la norma del Vector i finalment unes stopWords que són paraules que s'han d'ignorar.

5.1.8 Vector_TFIDF

El Vector_TFIDF és una herència de Vector_EV implementa una estratègia de pesos utilitzant el model del TFIDF. Per fer-ho, d'atributs es guarda les aparicions d'una paraula dins d'un `map<String, Integer>` i el nombre de paraules totals que hi ha al document, aquesta part ens serveix per calcular el TF. Per calcular l'IDF tenen 2 atributs static, compartits per totes les classes, un que és el nombre de documents (l'haurem d'incrementar en 1 cada vegada que afegim un nou document i també el nombre de documents que contenen una paraula, aquesta part és per calcular l'idf. Hem decidit fer-ho amb els static, ja que vam preferir que la responsabilitat de calcular les coordenades d'un vector sigui el mateix vector i no l'espai vectorial.

5.1.9 Vector_BM25

El Vector_BM25 és una herència de Vector_EV implementa una estratègia de pesos utilitzant el model del BM25. Com podem veure comparteix tots els atributs amb Vector_TFIDF i a més afegim les paraules totals que hi ha als n documents i dos paràmetres que s'usaran per calcular les coordenades (k i b).

5.2 Capa de Persistència

5.2.1 GestorDocuments

Com el nostre model utilitza path per la identificació dels Documents, necessitàvem que els Documents es guardaren sempre en el disc i no en RAM. Per aquest motiu vam pensar en aquestes dues alternatives per fer el pas de l'objecte Document a un fitxer en disc.

Serialització propia

Com a primera opció que vam pensar, va ser fer nosaltres una serialització pròpia. És a dir, fer operació per fer la traducció dels Documents. Aquesta alternativa utilitzaria la de `FileWriter` per fer aquesta traducció. La principal desavantatge és que aquest fitxer no seria diferent d'un document de text pla i no volíem fer una barreja entre els documents exportats i els documents que són part del sistema. A més a més, si en un futur es volgués fer modificacions en el document, també s'hauria de modificar aquesta operació.

Serialització de Java

Java ja té implementades operacions per fer el pas directe d'objectes a un format de fitxer en disc. Per tant, ens facilita de forma considerable la implementació del GestorDocuments. Ja que podem aquesta traducció de manera ràpida gràcies a les classes `ObjectOutputStream` i `ObjectInputStream`. L'únic inconvenient que vam veure és que a causa que Java requereix que l'objecte que es vol serialitzar necessiti fer una herència de la interfície `Serializable` i especificar un `serialVersionUID` per fer correctament aquestes traduccions ens veuríem obligats a fer una estructura de dades a forma de contenidor per així fer tot el procés de manera senzilla.

Tenint aquestes dues opcions, vam escollir la serialització de JAVA a causa de la garantia que tot aquest sistema i no volíem tornar a implementar una funcionalitat que ja està feta pels mateixos creadors de JAVA. A més a més, amb l'estructura container necessària, veiem també que ens ajudaria en el pas d'informació entre el GestorDocuments i TransformadorArxius.

5.2.2 CtrlConfiguracio

A causa de que aquesta classe s'encarrega de fer tot el pass de la configuració de l'aplicació a un format permanent en disc. Vam basarnos en el model que hem utilitzat en el GestorDocuments per fer això. L'únic canvi és que en comptes de donar-nos una instància d'un objectes (en aquest poden ser instàncies de tots els indexos i del GestorDocuments),

s'ens proporcionaria arrays de bytes que representen la codificació que tenen els indexos i el GestorDocuments, d'aquesta manera mantenim el propòsit de l'arquitectura en tres capes a la vegada que simplifica el process. L'únic inconvenient és el mateix d'abans, hem fet herència de la interfície *Serializable* i hem especificat un *SerialUID* en cada index i classe necessaria dels index.

5.2.3 TransformadorTextPla

Com Java ja ens ofereix la llibreria java.io per fer crides d'escriptura i lectura al sistema operatiu en cap moment ens vam plantejar utilitzar una llibreria externa per implmentar el pas d'informació del .ojmj a un arxiu .txt.

5.2.4 TransformadorXML

En aquest cas primer ens vam plantejar fer-ho també amb la llibreria java.io i de fet l'exportació no es gaire complicada fer-la d'aquesta forma, però sí que és veritat que la lectura d'aquest tipus d'arxius es complicava molt fent-ho d'aquesta forma, tot i que no hagués segut impossible. Investigant una mica més vam trobar que amb les llibreries org.w3c.dom i javax.xml ho podem fer. A més aquestes llibreries són natives de java així que amb aquestes no cal afegir llibreries externes.

5.2.5 TransformadorPDF

Aquí sí que ens vam plantejar ràpidament utilitzar alguna llibreria ja que pdf es un format molt complex com per fer-ho a mà. Com vam veure que java no tenia cap llibreria nativa per implementar la importació i exportació de pdf vam buscar llibreries externes i ens vam decidir per iText5. També vam trobar la iText7 i tot i tenir més funcionalitats vam decidir triar per la iText5, ja que per a l'ús que nosaltres volíem fer era molt més senzilla.

5.2.6 TransformadorJSON

Per als arxius JSON ens va passar com al XML, primer vam pensar de fer-ho amb el java.io i es podia fer l'exportació amb bastanta facilitat, el problema era la importació. Per tant, al final vam decidir utilitzar una llibreria i com que java no en té per json vam optar per la llibreria externa creada per Google Gson.

