

PROP

Compressor d'arxiu

Diagrama de Classes

Identificador de Grup: grup-6.4

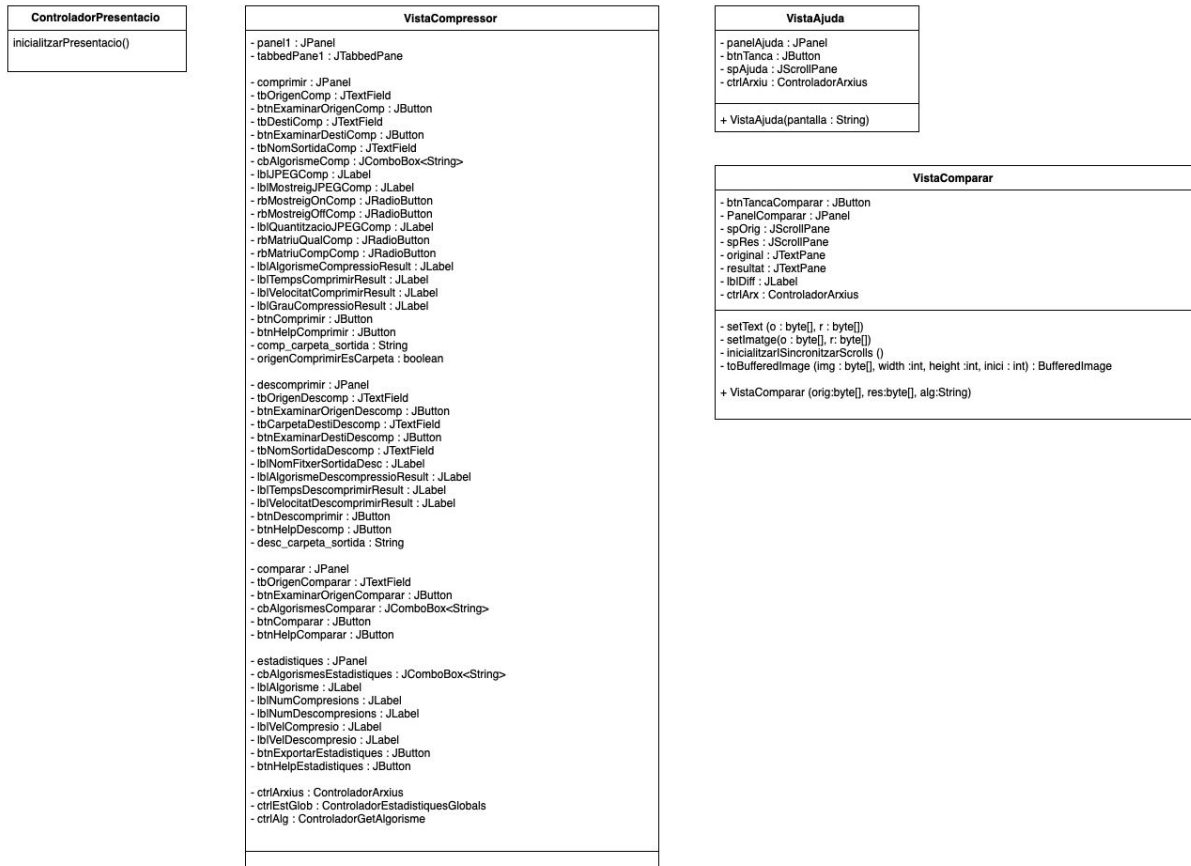
Joan Bosch Pons
Joan Salvador Magrans
Clara Sánchez Flaquer
Daniela Zhao

ÍNDEX

Capa de Presentació	3
Diagrama de Classes:	3
VistaCompressor	3
Comprimir	4
Descomprimir	6
Comparar	7
Estadistiques Globals	7
VistaComparar	8
VistaAjuda	9
Controladors	10
ControladorPresentacio	10
Capa de Domini	11
Diagrama de classes	11
Fitxer	11
ResultatAlgorisme	13
EstadistiquesLocals	14
EstadistiquesGlobals	15
Algorisme	16
LZ78	17
LZSS	18
LZW	18
JPEG	19
HuffmanJPEG	21
Node	22
Controladors	23
ControladorEstadistiquesGlobals	23
ControladorArxius	23
ControladorAlgorisme	26
Capa de Persistència	27
Diagrama de classes	27
Controladors	27
ControladorDades	27

Capa de Presentació

Diagrama de Classes:



<https://imgur.com/a/2UmWrwF>

- **VistaCompressor**

La classe VistaCompressor implementa la vista principal del programa. Està dividida en quatre pestanyes.

Conté els següents atributs:

- panel1: JPanel
Finestra principal.
- tabbedPane1: JTabbedPane

La vista de pestanyes comentada anteriorment. Conté les quatre pestanyes.

- ctrlArxius : ControladorArxius
Instància de ControladorArxius per tal de fer les crides a comprimir i descomprimir per fitxers o carpetes.
- ctrlEstGlob : ControladorEstadistiquesGlobals
Instància de ControladorEstadistiquesGlobals per obtenir les estadístiques globals de cada un dels algorismes.
- ctrlAlg : ControladorAlgorisme
Instància de ControladorAlgorisme, per poder obtenir els noms dels algorismes a triar en els desplegable.

- **Comprimir**

Aquesta pestanya permet comprimir fitxers individuals o carpetes i mostra les estadístiques de la compressió.

Conté els següents atributs:

- comprimir : JPanel
La pestanya de comprimir.
- tbOrigenComp : JTextField
Quadre de text que mostra el path del fitxer o carpeta origen seleccionat.
- btnExaminarOrigenComp : JButton
Botó que mostra un selector de fitxers o carpetes per a l'origen.
- tbDestiComp : JTextField
Quadre de text que mostra el path de la carpeta destí seleccionada.
- btnExaminarDestiComp : JButton
Botó que mostra un selector carpetes per al destí.
- tbNomSortidaComp : JTextField
Quadre de text on escriure el nom del fitxer de destí.
- cbAlgorismeComp : JComboBox<String>
Desplegable per triar l'algorisme de compressió.
- lblJPEGComp : JLabel
Text d'explicació de les opcions de JPEG. Es mostra quan l'algorisme seleccionat és JPEG.

- lblMostreigJPEGComp : JLabel
Text d'explicació de les opcions de JPEG. Es mostra quan l'algorisme seleccionat és JPEG.
- rbMostreigOnComp : JRadioButton
Botó per triar la configuració de mostreig. Es mostra quan l'algorisme seleccionat és JPEG.
- rbMostreigOffComp : JRadioButton
Botó per triar la configuració de mostreig. Es mostra quan l'algorisme seleccionat és JPEG.
- lblQuantitzacioJPEGComp : JLabel
Text d'explicació de les opcions de JPEG. Es mostra quan l'algorisme seleccionat és JPEG.
- rbMatriuQualComp : JRadioButton
Botó per triar la configuració de quantització. Es mostra quan l'algorisme seleccionat és JPEG.
- rbMatriuCompComp : JRadioButton
Botó per triar la configuració de quantització. Es mostra quan l'algorisme seleccionat és JPEG.
- lblAlgorismeCompressioResult : JLabel
Text que mostrarà l'algorisme de compressió utilitzat.
- lblTempsComprimirResult : JLabel
Text que mostrarà el temps de compressió.
- lblVelocitatComprimirResult : JLabel
Text que mostrarà la velocitat de compressió.
- lblGrauCompressioResult : JLabel
Text que mostrarà el grau de compressió.
- btnComprimir : JButton
Botó per iniciar la compressió.
- btnHelpComprimir : JButton
Botó per mostrar la vista d'ajuda.
- comp_carpeta_sortida : String
Emmagatzema el path de la carpeta destí de la compressió.

- origenComprimirEsCarpeta : boolean
Cert si l'origen és una carpeta. S'utilitza per saber si s'haurà de cridar a compressió de fitxers o carpetes.

No hi ha cap mètode associat a aquesta pestanya.

- **Descomprimir**

Aquesta pestanya permet descomprimir arxius compatibles comprimits amb el mateix aplicatiu.

- descomprimir : JPanel
La pestanya de descomprimir.
- tbOrigenDescomp : JTextField
Quadre de text on es mostra el path del fitxer a descomprimir.
- btnExaminarOrigenDescomp : JButton
Botó que mostra el selector de fitxers per a descomprimir.
- btnExaminarDestiDescomp : JButton
Botó que mostra el selector de carpetes i directoris per guardar el resultat de la descompressió.
- tbNomSortidaDescomp : JTextField
Quadre de text on es mostra/indica el nom del fitxer resultant de la descompressió.
- lblNomFitxerSortidaDesc : JLabel
Etiqueta que conté "Nom Fitxer Sortida"
- lblAlgorismeDescompressioResult : JLabel
Etiqueta que conté "Algorisme utilitzat"
- lblVelocitatDescomprimirResult : JLabel
Etiqueta que conté "Velocitat de descompressió"
- btnDescomprimir : JButton
Botó que serveix per iniciar el procés de descompressió.
- btnHelpDescomprimir : JButton
Botó que serveix per obrir la finestra d'ajuda de descompressió.

- desc_carpeta_sortida : String
Atribut que conté el directori complet de l'arxiu que s'està descomprimint.

- **Comparar**

Aquesta pestanya permet comparar dos fitxers.

Conté els següents components:

- comparar : JPanel
La pestanya de comparar.
- tbOrigenComparar : JTextField
Quadre de text on es mostra el path del fitxer a comparar.
- btnExaminarOrigenComparar : JButton
Botó que mostra un selector de fitxers o carpetes per a l'origen.
- cbAlgorismeComparar : JComboBox<String>
Desplegable per triar l'algorisme de compressió/descompressió.
- btnComparar : JButton
Botó per iniciar el procés de comparació de fitxers.
- btnHelpComparar : JButton
Botó per mostrar la vista d'ajuda.

- **Estadístiques Globals**

Aquesta pestanya permet visualitzar i exportar les estadístiques globals d'un algorisme seleccionat.

Conté els següents atributs:

- estadistiques : JPanel
La pestanya d'estadístiques globals.
- btnHelpEstadistiques : JButton
Botó per mostrar la vista d'ajuda.
- cbAlgorismesEstadistiques : JComboBox<String>
Desplegable on es mostren els algorismes.

- btnExportarEstadistiques : JButton
Botó per exportar les estadístiques mostrades per pantalla.
- lblAlgorisme : JLabel
Etiqueta on es mostra l'algorisme seleccionat.
- lblVelCompresio : JLabel
Etiqueta on es mostra la velocitat de compressió.
- lblVelDescompresio : JLabel
Etiqueta on es mostra la velocitat de descompressió.
- lblNumCompresions : JLabel
Etiqueta on es mostra en número de compressions.
- lblNumDescompresions : JLabel
Etiqueta on es mostra en número de descompressions.

- **VistaComparar**

La classe VistaComparar implementa la finestra per comparar dos fitxers de l'aplicació.

Conté els següents atributs:

- PanelComparar : JPanel
La finestra de comparar.
- spOrig : JScrollPane
És un component de Swing que gestiona una regió de la finestra, se li pot afegir *scrolls* verticals i horitzontals. Se li assigna un component *JTextPane* amb el contingut del fitxer original a comparar.
- spRes : JScrollPane
És un component de Swing que gestiona una regió de la finestra, se li pot afegir *scrolls* verticals i horitzontals. Se li assigna un component *JTextPane* amb el contingut del fitxer resultat a comparar.
- btnTancaComparar : JButton
Botó per poder tancar la finestra per comparar fitxers.

- lblDiff : JLabel

Indica a l'usuari si el contingut dels dos fitxers són iguals o difereixen.

Conté els següents mètodes:

- setText (byte[], byte[]) :
Assigna els textos corresponents a cada *panel*.
- setImatge(byte[], byte[]) :
Assigna les imatges corresponents a cada *panel*.
- inicialitzarISincronitzarScrolls () :
Configura els *scrollbars* per a que apareixin quan sigui necessari i sincronitzar-los perquè es mostri el mateix en els dos *panels*.
- toBufferedImage (byte[], int, int, int) : BufferedImage
Conversió de byteArray a BufferedImage per poder mostrar la imatge.
- VistaComparar (byte[], byte[], String) :
Creadora on es li passa el contingut original, el contingut comprimit-descomprimit i l'algorisme utilitzat per controlar si ha sigut una imatge o un text.

- **VistaAjuda**

La classe VistaAjuda implementa la finestra d'ajuda de l'aplicació.

Conté els següents atributs:

- ctrlArxiu : ControladorArxiu
Objecte controlador de la capa de domini.
- spAjuda : JScrollPane
És un component de Swing que gestiona una regió de la finestra, se li pot afegir *scrolls* verticals i horitzontals.
- panelAjuda : JPanel
És un component de Swing que guarda i organitza un grup de components.
- btnTanca : JButton
És un component de Swing per poder tancar la finestra per comparar fitxers.

Conté els següents mètodes:

- VistaAjuda(String)

Creadora on se li passa per paràmetre el nom de la finestra per mostrar el contingut corresponent.

Controladors

- **ControladorPresentacio**

La classe ControladorPresentació implementa l'intermediari entre la capa de Presentació i la funció *Main* de l'aplicació.

No conté cap atribut.

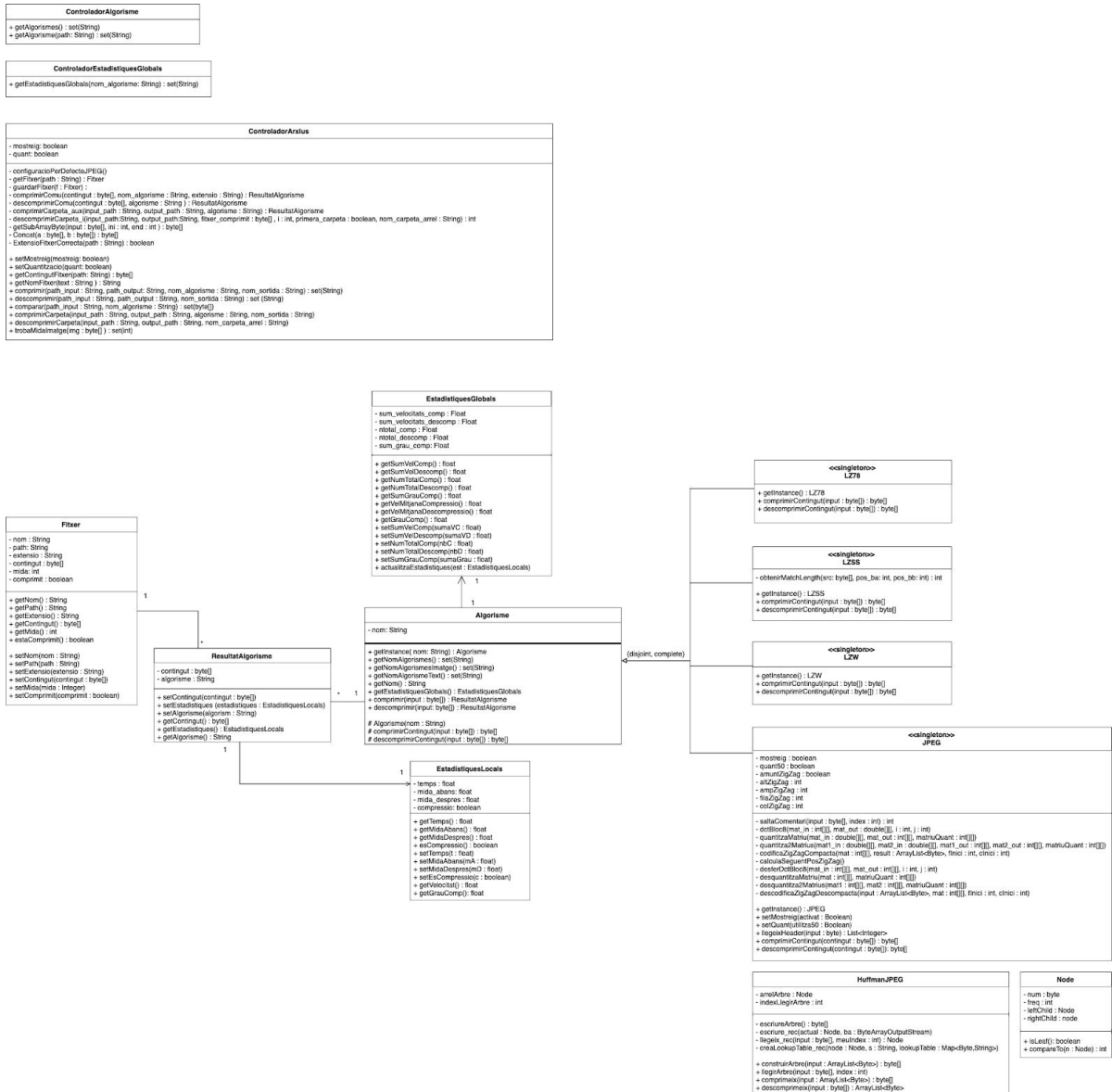
Conté els següents mètodes:

- inicialitzarPresentacio()

Crea una nova finestra principal de l'aplicació.

Capa de Domini

Diagrama de classes



- **Fitxer**

La classe Fitxer implementa getters i setters pels atributs privats que conté la classe.

Conté els següents atributs:

- nom : String

Indica el nom del fitxer corresponent.

<https://imqur.com/a/myOlzO4>

- path : String
Indica el path on es troba el fitxer corresponent.
- extensio : String
Indica l'extensió del fitxer corresponent.
- contingut : byte[]
Indica el contingut del fitxer corresponent el bytes.
- mida : int
Indica la mida que ocupa el fitxer corresponent.
- comprimit : boolean
Indica si el fitxer corresponent està comprimit o no.

Conté els següents mètodes:

- getNom() : String
Retorna el nom corresponent al fitxer.
- getPath() : String
Retorna el path corresponent al fitxer.
- getExtensio() : String
Retorna l'extensió corresponent al fitxer.
- getContingut() : byte[]
Retorna el contingut del fitxer en bytes.
- getMida() : Integer
Retorna la mida del fitxer corresponent.
- estaComprimir() : boolean
Retorna si el fitxer corresponent està comprimit o no.
- setNom(String) :
Assigna el paràmetre passat al nom del fitxer corresponent.
- setPath(String) :
Assigna el paràmetre passat al path del fitxer corresponent.
- setExtensio(String) :
Assigna el paràmetre passat a l'extensió del fitxer corresponent.
- setContingut(byte[]) :
Assigna el paràmetre passat al contingut del fitxer corresponent.

- setMida(String) :
Assigna el paràmetre passat a la mida del fitxer corresponent.
- setComprimit(String) :
Assigna el paràmetre passat a comprimit del fitxer corresponent.

- **ResultatAlgorisme**

La classe ResultatAlgorisme és concebuda com un DTO (Data Transfer Object), és a dir, com una forma senzilla per poder comunicar dades del resultat de compressió/descompressió entre les classes de domini.

Conté els següents atributs:

- contingut : byte[]
Conté el resultat (contingut) de la compressió o descompressió d'un fitxer o carpeta.
- algorisme : String
Conté l'algorisme que s'ha utilitzat durant el procés de compressió o descompressió.

Conté els següents mètodes:

- setContingut(byte[])
Assigna el contingut passat com a paràmetre al contingut de la classe.
- setEstadistiques(EstadistiquesLocals)
Assigna les estadístiques passades com a paràmetre com a estadístiques locals de la classe.
- setAlgorisme(String)
Assigna a la variable algorisme de la classe el valor que es passa com a paràmetre.
- getContingut() : byte[]
Obté el contingut de la variable contingut.
- getEstadistiques() : EstadistiquesLocals
Obté la instància d'estadístiques locals de la classe.

- getAlgorisme() : String
Obté l'algorisme assignat a la classe.

- **EstadistiquesLocals**

La classe EstadistiquesLocals implementa les estadístiques generades per una compressió/descompressió.

Conté els següents atributs:

- temps : float
Indica el temps que s'ha trigat en la compressió/descompressió.
- mida_abans: float
Indica la mida original del fitxer.
- mida_despres : float
Indica la mida del fitxer comprimit/descomprimit.
- compressio: boolean
Indica si ha sigut una compressió (*true*) o una descompressió (*false*).

Conté els següents mètodes:

- getTemps() : float
Obté el temps.
- getMidaAbans() : float
Obté la mida original del fitxer.
- getMidaDespres() : float
Obté la mida del fitxer comprimit/descomprimit.
- esCompressio() : boolean
Obté cert si és una compressió, altrament fals.
- setTemps(float)
Assigna el paràmetre d'entrada al temps.
- setMidaAbans(float)
Assigna el paràmetre d'entrada a la mida original del fitxer.

- setMidaDespres(float)
Assigna el paràmetre d'entrada a la mida del fitxer comprimit/descomprimit.
- setEsCompressio(boolean)
Assigna el paràmetre d'entrada al booleà compressio.
- getVelocitat() : float
Obté la velocitat de compressió/descompressió.
- getGrauComp(): float
Obté el grau de compressió.

- **EstadistiquesGlobals**

La classe EstadistiquesGlobals implementa les estadístiques d'un algorisme.

Conté els següents atributs:

- sum_velocitats_comp : Float
Conté la suma de les velocitats de compressions que s'han executat.
- sum_velocitats_descomp : Float
Conté la suma de les velocitats de descompressions que s'han executat.
- ntotal_comp : Float
Conté el número total de compressions que s'han executat.
- ntotal_descomp : Float
Conté el número total de descompressions que s'han executat.
- sum_grau_comp: Float
Conté la suma de graus de compressions que s'han executat.

Conté els següents mètodes:

- getSumVelComp() : float
Obté la suma de velocitats de compressions.
- getSumVelDescomp() : float
Obté la suma de velocitats de descompressions.
- getNumTotalComp() : float
Obté el número total de compressions.

- `getNumTotalDescomp()` : float
Obté el número total de descompressions.
- `getVelMitjanaCompressio()` : float
Obté la velocitat mitjana de compressions.
- `getVelMitjanaDescompressio()` : float
Obté la velocitat mitjana de descompressions.
- `getMitjGrauComp()` : float
Obté el grau mitjà de compressions.
- `setSumVelComp(float)`
Assigna la suma de velocitats de compressions.
- `setSumVelDescomp(float)`
Assigna la suma de velocitats de descompressions.
- `setNumTotalComp(float)`
Assigna el número total de compressions.
- `setNumTotalDescomp(float)`
Assigna el número total de descompressions.
- `setSumGrauComp(float)`
Assigna la suma de graus de compressions.
- `actualitzaEstadistiques(EstadistiquesLocals)`
Actualitza els atributs de la classe amb les dades del paràmetre d'entrada.

- **Algorisme**

La classe algorisme és la superclasse que controla les funcionalitats de comprimir i descomprimir del sistema. També gestiona la generació de les estadístiques locals i l'actualització de les estadístiques globals.

Conté els següents atributs:

- `nom` : String (protected)
Conté el nom de l'algorisme al que fa referència la instància.

Conté els següents mètodes:

- `getInstance(String) : Algorisme`
La funció `getInstance`, retorna una instància de l'algorisme indicat al paràmetre d'entrada. És una funció específica per poder implementar el patró Singleton al nostre projecte.
- `getNomAlgorismes() : Set(String)`
Obté el conjunt total d'algorismes disponibles al sistema.
- `getNomAlgorismesImatges() : Set(String)`
Obté el conjunt total d'algorismes de compressió d'arxius d'imatge disponibles al sistema.
- `getNomAlgorismesText() : Set(String)`
Obté el conjunt total d'algorismes de compressió d'arxius de text disponibles al sistema.
- `getNom() : String`
Obté el nom de l'algorisme de la instància que s'executa la comanda.
- `getEstadistiquesGlobals() : EstadistiquesGlobals`
Obté la instància d'Estadístiques Globals de l'algorisme.
- `comprimir(byte[]) : ResultatAlgorisme`
Funció pública per comprimir el contingut que se li passa per paràmetre. És l'encarregada de la generació de les estadístiques locals de compressió i de l'actualització de les estadístiques globals de compressió de l'algorisme.
- `descomprimir(byte[]) : ResultatAlgorisme`
Funció pública per descomprimir el contingut que se li passa per paràmetre. És l'encarregada de la generació de les estadístiques locals de descompressió i de l'actualització de les estadístiques globals de descompressió de l'algorisme.
- `comprimirContingut(byte[]) : byte[]`
Funció abstracta que implementen les subclasses de `Algorisme` per comprimir la seqüència de bytes.
- `descomprimirContingut(byte[]) : byte[]`
Funció abstracta que implementen les subclasses de `Algorisme` per descomprimir la seqüència de bytes.

- **LZ78**

La classe LZ78 és una subclasse d'Algorisme, que implementa la compressió i descompressió de text amb l'algorisme del mateix nom.

No conté cap atribut.

Conté els següents mètodes:

- `comprimirContingut(byte[]) : byte[]`
Comprimeix el contingut que obté en el paràmetre d'entrada i retorna el resultat.
- `descomprimirContingut(byte[]) : byte[]`
Descomprimeix el contingut que obté en el paràmetre d'entrada i retorna el resultat.

- **LZSS**

La classe LZSS és una subclasse d'Algorisme, que implementa la compressió i descompressió de text amb l'algorisme del mateix nom.

No conté cap atribut.

Conté els següents mètodes:

- `obtenirMatchLength(byte[], int, int) : int`
Calcula la llargada de coincidència començant desde les posicions que rep.
- `getInstance() : LZSS`
Obté la instància de LZSS.
- `comprimirContingut(byte[]) : byte[]`
Comprimeix el text contingut en el paràmetre d'entrada i retorna el resultat.
- `descomprimirContingut(byte[]): byte[]`
Descomprimeix el text contingut en el paràmetre d'entrada i retorna el resultat.

- **LZW**

La classe LZW és una subclasse d'Algorisme, que implementa la compressió i descompressió de text amb l'algorisme del mateix nom.

No conté cap atribut.

Conté els següents mètodes:

- `comprimirContingut(byte[]) : byte[]`
Comprimeix el text contingut en el paràmetre d'entrada i retorna el resultat.
- `descomprimirContingut(byte[]) : byte[]`
Descomprimeix el text contingut en el paràmetre d'entrada i retorna el resultat.

- **JPEG**

La classe JPEG és una subclasse d'Algorisme, que implementa la compressió i descompressió d'imatges amb l'algorisme del mateix nom.

Conté els següents atributs:

- `jpeg: JPEG`
Emmagatzema la instància de JPEG.
- `mostreig: boolean`
Conté la configuració de mostreig per la compressió/descompressió. Cert vol dir mostreig 4:2:0 i fals vol dir 4:4:4.
- `quant50: boolean`
Conté la configuració de quantització per la compressió/descompressió. Cert vol dir que es farà servir MATQ50 i fals que s'utilitzarà MATQ.
- `amuntZigZag: boolean`
Per fer zig-zag. Cert vol dir que s'està anant cap amunt i fals, cap avall.
- `altZigZag: int`
Indica la posició final d'alçada del bloc 8x8 on s'està fent el zig-zag.

- ampZigZag: int
Indica la posició final d'amplada del bloc 8x8 on s'està fent el zig-zag.
- filaZigZag: int
Indica la posició actual d'alçada del bloc 8x8 on s'està fent el zig-zag.
- colZigZag: int
Indica la posició actual d'amplada del bloc 8x8 on s'està fent el zig-zag.

Conté els següents mètodes:

- getInstance(): JPEG
Obté una instància de JPEG.
- setMostreig(boolean)
Desa el valor passat com a paràmetre a l'atribut 'mostreig'.
- setQuant(boolean)
Desa el valor passat com a paràmetre a l'atribut 'quant50'.
- comprimirContingut(byte[]): byte[]
Comprimeix la imatge continguda en el paràmetre d'entrada i retorna el resultat.
- llegeixHeader(byte[]): List<Integer>
Llegeix la capçalera ppm de la imatge continguda en l'array de bytes.
- saltaComentari(byte[], int): int
Salta un comentari de la capçalera. Mètode privat cridada per llegeixHeader.
- dctBloc8(int[][], double[][], int, int)
Aplica la transformació discreta del cosinus a un bloc 8x8 d'una matriu corresponent a un dels components del color.
- quantitzaMatriu(double[][], int[], int[])
Aplica la quantització a una matriu.
- quantitza2Matrius(double[][], double[][], int[], int[], int[])
Aplica la quantització a dues matrius. Pensat per fer Cb i Cr alhora.
- codificaZigZagCompacta(int[], ArrayList<Byte>, int, int)
Aplica el zig-zag a un bloc 8x8 de la matriu passada com a paràmetre i desa el resultat a l'ArrayList. També elimina seqüències de molts zeros.

- `calculaSegurntPosZigZag()`
Calcula la següent posició de la matriu quan es recorre en zig-zag. Aquest mètode el criden `codificaZigZagCompacta` i `descodificaZigZagDescompacta`.
- `descomprimirContingut(byte[]): byte[]`
Descomprimeix el fitxer comprimit que conté el paràmetre d'entrada i retorna el ppm descomprimit que s'ha obtingut.
- `desferDctBloc8(int[], int[], int, int)`
Desfà la transformació discreta del cosinus a un bloc 8x8 d'una matriu.
- `desquantitzaMatriu(int[], int[])`
Desfà la quantització aplicada anteriorment a una matriu.
- `desquantitza2Matrius(int[], int[], int[])`
Desfà la quantització a dues matrius alhora. Pensat per fer-ho amb Cb i Cr a la vegada.
- `descodificaZigZagDescompacta(ArrayList<Byte>, int[], int, int)`
Desfà el zig-zag de l'ArrayList de forma que reconstrueix un bloc 8x8 de la matriu. També recupera les seqüències llargues de zeros que s'han eliminat al codificar amb zig-zag.

● **HuffmanJPEG**

La classe HuffmanJPEG s'encarrega de la part de Huffman de l'algorisme JPEG. Degut a la seva complexitat hem cregut convenient posar-ho en una casse separada.

Conté els següents atributs:

- `arrelArbre: Node`
És el node arrel de l'arbre utilitzat per obtenir els codis que fa servir l'algorisme per comprimir o descomprimir.
- `indexLlegirArbre: int`
Posició de l'array que s'està llegint un arbre a `llegirArbre`. També s'utilitza per saber l'inici del contingut a descomprimir.

Conté els següents mètodes:

- `construirArbre(ArrayList<Byte>) : byte[]`
Construeix un arbre per generar els codis pels bytes continguts en el paràmetre d'entrada i el retorna en format `byte[]`.
- `escriureArbre() : byte[]`
Mètode privat que s'encarrega d'escriure l'arbre generat a `construirArbre`.
- `escriure_rec(Node, ByteArrayOutputStream)`
Mètode recursiu d'`escriureArbre`.
- `llegirArbre(byte[], int)`
Llegeix l'arbre contingut al `byte[]` a partir de la posició indicada per l'enter.
- `llegeix_rec(byte[], int) : Node`
Mètode recursiu de `llegirArbre` que va construint els nodes de l'arbre.
- `comprimeix(ArrayList<Byte>) : byte[]`
Comprimeix amb Huffman l'`ArrayList` passat com a paràmetre amb l'arbre que s'haurà construït prèviament.
- `creaLookupTable_rec(Node, String, Map<Byte,String>)`
Mètode privat utilitzat per `comprimeix` per generar el Map amb els codis de cada byte.
- `descomprimeix(byte[]) : ArrayList<Byte>`
Descomprimeix amb Huffman el `byte[]` que es passa com a paràmetre. Cal haver cridat abans a `llegirArbre`, ja que és aquest mètode el que s'encarrega de determinar en quin punt del `byte[]` comença el contingut a descomprimir.

- **Node**

La classe `Node` representa un node d'un arbre. Només l'utilitza `HuffmanJPEG` i per això està implementada com una classe privada dins seu.

Conté els següents atributs:

- `num: byte`
Número que representa el node.
- `freq: int`
Freqüència del número.

- leftChild: Node
Fill esquerre.
- rightChild: Node
Fill dret.

Conté els següents mètodes:

- isLeaf(): boolean
Retorna cert si el node és fulla, és a dir, no té ni fill dret ni esquerre.
- compareTo(Node): int
Mètode per comparar dos nodes per tal d'ordenar-los. El resultat és la comparació de les freqüències i si són iguals és la comparació del número.

Controladors

- **ControladorEstadistiquesGlobals**

S'encarrega de proporcionar a la capa de presentació els valors de les estadístiques globals.

No conté cap atribut.

Conté els següents mètodes:

- getEstadistiquesGlobals(String) : Set(String)
Retorna les estadístiques que té l'algorisme.

- **ControladorArxius**

La classe ControladorArxius s'encarrega de proporcionar a la capa de presentació les funcions necessàries per poder comprimir i descomprimir un fitxer o una carpeta.

Conté els següents atributs:

- mostreig : boolean
Conté la configuració de mostreig per la compressió/descompressió. Cert vol dir mostreig 4:2:0 i fals vol dir 4:4:4.

- quant : boolean

Conté la configuració de quantització per la compressió/descompressió. Cert vol dir que es farà servir MATQ50 i fals que s'utilitzarà MATQ.

Conté els següents mètodes:

- setMostreig(most : boolean) :
Assigna el valor passat per paràmetre a l'atribut mostreig de la classe.
- setQuantitzacio(q : boolean) :
Assigna el valor passat per paràmetre a l'atribut quant de la classe.
- configuracioPerDefecteJPEG() :
Es configuren els atributs per defecte de JPEG, és a dir, mostreig a fals i quant a true.
- getFitxer(String) : Fitxer
Retorna un tipus Fitxer del corresponent path introduït com a paràmetre.
- guardarFitxer(Fitxer) :
Crida una funció de la capa de dades on crea i guarda el Fitxer passat per paràmetre al sistema de fitxers.
- getContingutFitxer(String) : byte[]
Retorna el contingut del fitxer corresponent al paràmetre passat en un array de bytes.
- comprimirComu(byte[], String) : ResultatAlgorisme
Comprimeix el contingut d'un fitxer passat per paràmetre amb l'algorisme corresponent o el mode automàtic passat per paràmetre també.
- comprimir(String, String, String, String) : set(String)
Mètode que crida a la funció privada comprimirComu i retorna un set de String on conté les estadístiques locals, el contingut i l'algorisme. També crea el fitxer corresponent a la compressió.
- descomprimirComu(byte[], String) : ResultatAlgorisme
Descomprimeix el contingut d'un fitxer passat per paràmetre amb l'algorisme corresponent passat per paràmetre també.

- `descomprimir(String, String, String, String) : set(String)`
Mètode que crida a la funció privada `descomprimirComu` i retorna un set de `String` on conté les estadístiques locals, el contingut i l'algorisme. També crea el fitxer corresponent a la descompressió.
- `comparar(String, String) : set(byte[])`
Funció per executar la funcionalitat principal de comparar. S'obté el contingut del fitxer que es troba a la ruta passada per paràmetre i es comprimeix i descomprimeix amb l'algorisme també passat com a paràmetre. Es retorna el contingut inicial del fitxer i el contingut final després de passar-lo pel procés de compressió i descompressió.
- `Concat(byte[], byte[]) : byte[]`
Funció que retorna una array de bytes resultant de concatenar la primera amb la segona (amb aquest ordre).
- `comprimirCarpeta_aux(String, String, String, String) : ResultatAlgorisme`
Comprimeix el contingut d'una carpeta a través dels paràmetres passats. Retorna un `ResultatAlgorisme` que conté el contingut, l'algorisme utilitzat i les estadístiques locals de la compressió.
- `comprimirCarpeta(String, String, String, String) :`
Crea un fitxer de tipus `Fitxer` i crida a la funció `comprimirCarpeta_aux` amb els paràmetres corresponents.
- `descomprimirCarpeta(String, String, String, String) :`
Crida una funció recursiva anomenada `descomprimirCarpeta_i` que descomprimeix un fitxer amb extensió `.comp` a una carpeta amb els seus fitxers i carpetes corresponents.
- `getSubArrayByte(byte[], int, int) : byte[]`
Funció que retorna el sub-vector contingut entre els enters passats com a paràmetre del vector de bytes passat a la funció.
- `descomprimirCarpeta_i(String, String, byte[], int, boolean, String) : int`
Mètode recursiu que descomprimeix un fitxer a una carpeta recursivament, on va creant els corresponents fitxers i capretes.
- `ExtensioFitxerCorrecta(String) : boolean`
Comprova que l'extensió d'un fitxer sigui `.txt` o `.ppm`.

- `trobarMidesImatge(byte[]) : set(Integer)`
Crida una funció de JPEG, retorna l'amplada, l'altura de la imatge i el valor màxim de colors i la posició on comença la informació de la imatge.
- `getNomFitxer(String) : String`
Retorna el nom del fitxer corresponent al path passat per paràmetre.

- **ControladorAlgorisme**

La classe `ControladorAlgorisme` s'encarrega de proporcionar a la capa de presentació els noms dels algorismes de compressió disponibles.

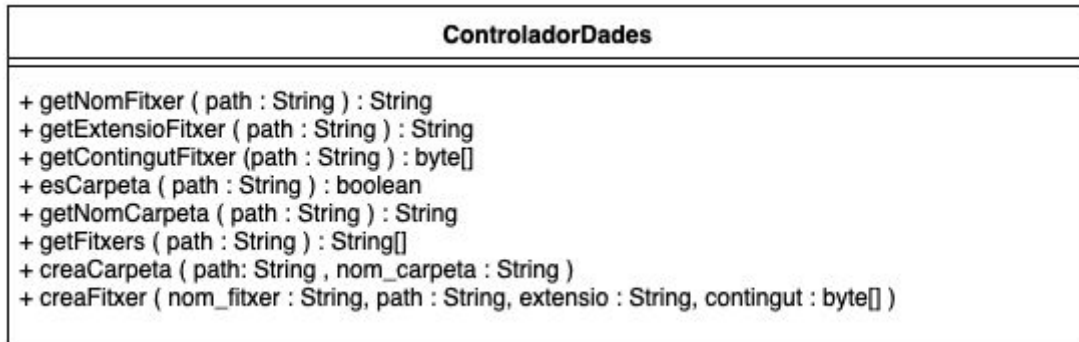
No conté cap atribut.

Conté els següents mètodes:

- `getAlgorismes(): List<String>`
Retorna els noms de tots els algorismes disponibles.
- `getAlgorismes(String): List<String>`
Retorna els noms dels algorismes que permeten comprimir el fitxer corresponent al path passat com a paràmetre.

Capa de Persistència

Diagrama de classes



Controladors

- **ControladorDades**

La classe ControladorDades s'encarrega de la comunicació amb el sistema de fitxers. Les funcions principals són la lectura i escriptura de fitxers o carpetes i la consulta d'informació relativa a aquests fitxers o carpetes.

No conté cap atribut.

Conté els següents mètodes:

- getNomFitxer(String): String
Retorna el nom del fitxer sense extensió corresponent al path passat com a paràmetre.
- getExtensioFitxer(String): String
Retorna l'extensió del fitxer corresponent al path passat com a paràmetre.
- getContingutFitxer(String): byte[]
Retorna el contingut del fitxer corresponent al path passat com a paràmetre en un array de bytes.

- esCarpeta(String): boolean
Si el path passat com a paràmetre correspon a una carpeta retorna cert, altrament retorna fals.
- getNomCarpeta(String): String
Retorna el nom de la carpeta corresponent al path passat com a paràmetre.
- getFitxers(String): String[]
Retorna els noms dels fitxers i carpetes que conté la carpeta que correspon al path passat com a paràmetre.
- creaCarpeta(String, String)
Crea una carpeta al directori indicat pel path del primer paràmetre amb el nom que s'indica al segon paràmetre.
- creaFitxer(String, String, String, byte[])
Crea o sobreescriu (si ja existeix) un fitxer amb el nom indicat al primer paràmetre i l'extensió indicada al tercer a la carpeta corresponent al path del segon paràmetre i hi escriu el contingut, que es troba a l'array de bytes.