**Compressor d'arxius**

Sergio Pascual Garcia (sergio.pascual)

Sheng Liu (sheng.liu)

Wei Xiao Xia (wei.xiao.xia)

**Index**

Diagrama de casos d’ús 2

Descripció dels casos d’ús 3

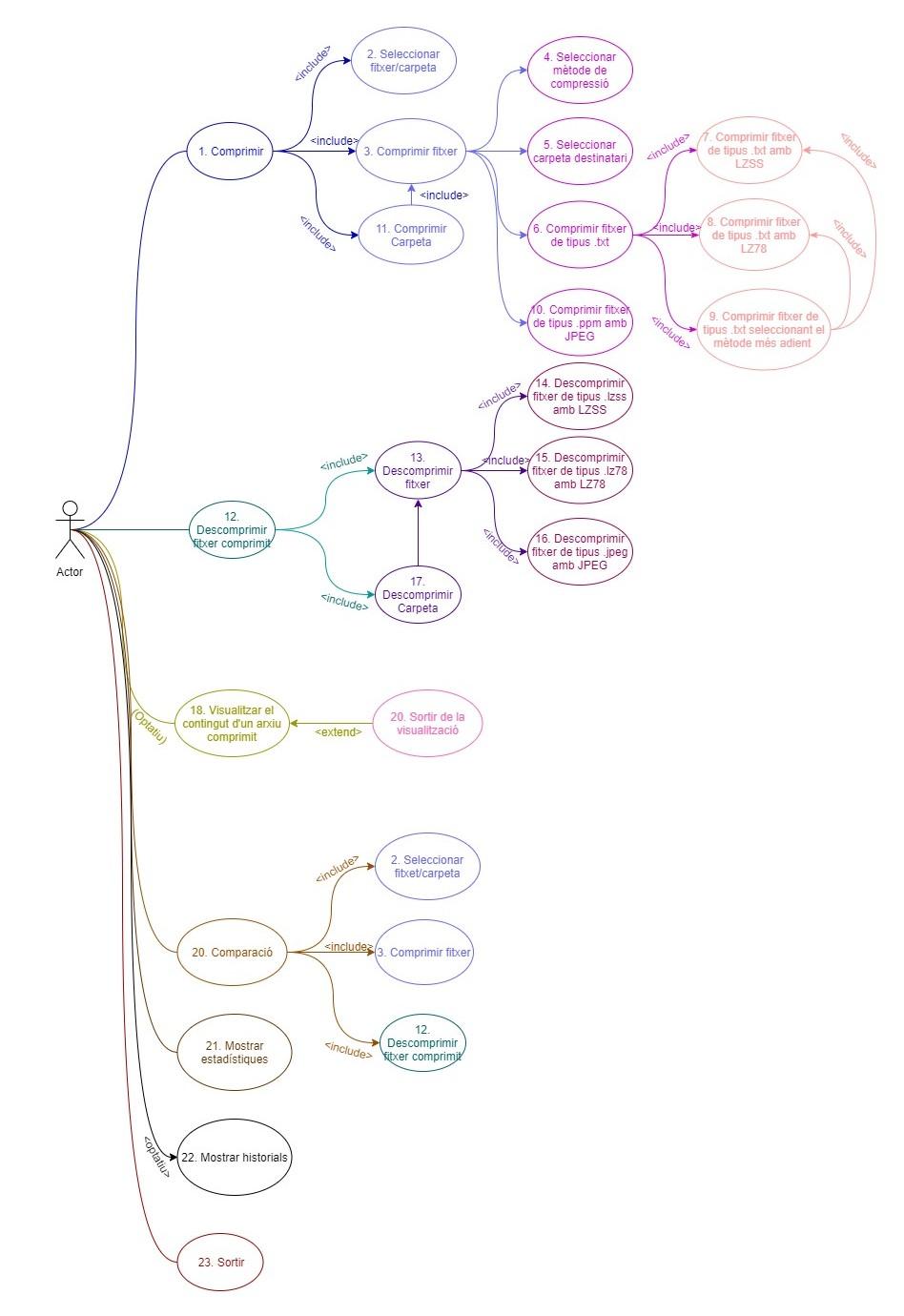
Diagrama de classes de disseny de Domini 23

Descripció de les classes de Domini 24

Descripció dels algorismes 27

Organització de l’equip de treball 33

Diagrama de casos d’ús



Descripció dels casos d’ús

|  |
| --- |
| **1. Comprimir fitxer** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet reduir la mida del fitxer indicat per l’usuari. |
| **Precondicions** |
| El fitxer seleccionat per l’usuari ha de ser un fitxer amb extensió .txt o .ppm. |
| **Escenari principal** |
| 1. L’Usuari indica al programa el path del fitxer que desitja comprimir.   Seleccionar fitxer/carpeta.   1. El programa verifica el tipus de fitxer. 2. El programa identifica el fitxer i desitja saber amb quin mètode es vol comprimir el fitxer i la carpeta destinatària a on es vol que es guardi l’arxiu comprimit. Seleccionar Mètode de Compressió i carpeta destinatari. 3. El Programa comprimeix el fitxer.    * Comprimir fitxer de tipus .txt o    * Comprimir fitxer de tipus .ppm amb JPEG 4. El programa empaqueta el contingut comprimit en un arxiu comprimit i el guarda a la carpeta destinatària. |
| **Errors i alternatius** |
| 1a. El fitxer indicat per l’Usuari no existeix.  1a1. El programa notifica a l’Usuari que el fitxer amb el path  indicat no existeix.  2a. El fitxer indicat no és de tipus .txt o .ppm.  2a1. El programa notifica a l’Usuari que només permet comprimir  fitxers del tipus .txt o .ppm. |

|  |
| --- |
| **2. Seleccionar mètode de compressió i carpeta destinatària** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet que l’usuari pugui seleccionar l’algorisme que s’utilitzarà per comprimir el fitxer de tipus .txt . L’usuari pot triar entre:   * L’algorisme LZ78 * L’algorisme LZSS * Deixar que el programa esculli el mètode més adient   També permet que l’Usuari pugui seleccionar la carpeta destinatària on desitja guardar l’arxiu comprimit.  El path per defecte serà la mateixa carpeta on es troba el fitxer/carpeta indicat per l’usuari. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. El programa solicita el mètode de compressió. 2. L’Usuari indica el mètode de compressió. 3. El programa solicita el path de la carpeta destinataria. 4. L’Usuari indica al programa el path de la carpeta destinataria. |
| **Errors i alternatius** |
| 4a. Carpeta no existeix.  4a.1. El programa notifica a l’Usuari que la carpeta destinatària  no existeix. |

|  |
| --- |
| **3. Comprimir fitxer de tipus .txt** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa redueix, sense pèrdua d’informació, la mida del fitxer .txt seleccionat per l’Usuari. S’utiliza el mètode triat per l’Usuari. |
| **Precondicions** |
| El fitxer seleccionat ha de ser de tipus .txt. |
| **Escenari principal** |
| 1. El programa consulta la preferència de l’Usuari. 2. El programa comprimeix el fitxer .txt seleccionat seguint la   preferència de l’Usuari.  -Comprimir fitxer de tipus .txt amb LZSS  -Comprimir fitxer de tipus .txt amb LZ78  -Comprimir fitxer de tipus .txt seleccionant el mètode més adient   1. El programa obté el fitxer comprimit. |
| **Errors** |
| 1. El fitxer de tipus .txt es vuit. |

|  |
| --- |
| **4. Comprimir fitxer de tipus .txt amb LZSS** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa comprimeix, fent servir algorisme de compressió LZSS i sense pèrdua d’informació, la mida del fitxer .txt seleccionat. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. El programa comprimeix el fitxer .txt fent servir LZSS i mostrant el % del fitxer ja comprimit en forma de progress bar per la finestra. 2. Una vegada acabada la compressió, mostra el % de compressió i el temps trigat 3. El programa obté el fitxer comprimit |
| **Errors** |
| 1. El fitxer de tipus .txt seleccionat es vuit. |

|  |
| --- |
| **5. Comprimir fitxer de tipus .txt amb LZ78** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa comprimeix, fent servir algorisme de compressió LZ78 i sense pèrdua d’informació, la mida del fitxer .txt seleccionat. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. El programa comprimeix el fitxer .txt seleccionat fent servir LZ78 i mostrant el % del fitxer ja comprimit en forma de progress bar per la finestra. 2. Una vegada acabada la compressió, mostra el % de compressió i el temps trigat 3. El programa obté el fitxer comprimit |
| **Errors** |
| 1. El fitxer de tipus .txt seleccionat es vuit. |

|  |
| --- |
| **6. Comprimir fitxer de tipus .txt seleccionant el mètode més adient** |
| **Actor** |
| Programa |
| **Descripció** |
| El Programa comprimeix, amb el mètode més adient i sense pèrdua d’informació, la mida del fitxer .txt seleccionat. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. El programa escull aleatòriament quin dels dos següents mètodes es farà servir.  * Comprimir fitxer de tipus .txt amb LZSS * Comprimir fitxer de tipus .txt amb LZ78  1. El programa comprimeix el fitxer .txt amb el mètode seleccionat. 2. El programa obté el fitxer comprimit. |
| **Errors** |
|  |

|  |
| --- |
| **7. Comprimir fitxer de tipus .ppm amb JPEG** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa comprimeix, amb l'algoritme de compressió JPEG i amb pèrdua d’informació, la mida del fitxer .ppm seleccionat. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. El programa comprimeix el fitxer .ppm seleccionat fent servir JPEG i mostrant el % del fitxer ja comprimit en forma de progress bar per la finestra. 2. Una vegada acabada la compressió, mostra el % de compressió i el temps trigat. 3. El programa obté el fitxer comprimit. |
| **Errors** |
|  |

|  |
| --- |
| **8. Comprimir Carpeta** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet reduir la mida de la carpeta indicada per l’usuari mantenent la estructura interna. |
| **Precondicions** |
| La carpeta seleccionada per l’usuari només conté carpetes o fitxers de tipus .ppm o .txt. |
| **Escenari principal** |
| 1. L’Usuari indica al programa el Path de la carpeta que desitja comprimir. Seleccionar fitxer/carpeta. 2. El programa desitja saber amb quin mètode es comprimeix la carpeta seleccionada i la carpeta destinatària a on es guarda l’arxiu comprimit. Seleccionar Mètode de Compressió i carpeta destinatari. 3. El programa farà una crida recursiva intentant comprimir primer totes les subcarpetes de la carpeta seleccionada. La compressió d’una carpeta consisteix en comprimir tots els fitxers continguts en aquesta carpeta. Comprimir fitxer de tipus .txt o Comprimir fitxer de tipus .ppm amb JPEG   4.El programa empaqueta la carpeta comprimida en un arxiu comprimit i la guarda a la carpeta destinatària. |
| **Errors** |
|  |

|  |
| --- |
| **9. Descomprimir arxiu comprimit** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet descomprimir un arxiu comprimit indicat per l’Usuari, recuperant el contingut que contenia. |
| **Precondicions** |
| L’arxiu comprimit ha de ser prèviament empaquetat per aquest mateix programa. |
| **Escenari principal** |
| 1. L’Usuari indica al programa que desitja descomprimir l’arxiu comprimit seleccionat. 2. El programa verifica l’arxiu comprimit. 3. El programa identifica a l’arxiu comprimit i desitja saber la carpeta destinatari a on es guarda l’arxiu descomprimit. Seleccionar carpeta destinatari. 4. El programa analiza i descomprimeix el contingut de l’arxiu comprimit. Descomprimir fitxer o Descomprimir carpeta. 5. El programa obté l’arxiu descomprimit i ho guarda a la carpeta destinatari. |
| **Errors** |
| 2a. L’arxiu comprimit no és empaquetat per aquest mateix programa  2a.1 El programa notifica a l’Usuari que no pot descomprimir aquest arxiu i cancel·la el procés de descompressió. |

|  |
| --- |
| **10. Descomprimir fitxer** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa permet recuperar el contingut d’un fitxer comprimit. |
| **Precondicions** |
| El fitxer seleccionat ha de ser prèviament comprimit per aquest mateix programa. |
| **Escenari principal** |
| 1. El programa analiza el fitxer comprimit per tal de començar el procés de descompressió.    * Descomprimir fitxer .txt comprimit amb LZSS   o   * + Descomprimir fitxer .txt comprimit amb LZ78   o   * + Descomprimir fitxer .ppm comprimit amb JPEG  1. El programa obté el fitxer descomprimit. |
| **Errors** |
|  |

|  |
| --- |
| **11. Descomprimir fitxer de tipus .lzss amb LZSS** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa recupera el contingut del fitxer .lzss fent servir l’algoritme LZSS. |
| **Precondicions** |
| (correctesa del fitxer input assegurat per Descomprimir fitxer) |
| **Escenari principal** |
| 1. El programa recupera el contingut del fitxer .lzss seleccionat fent servir l’algoritme LZSS i mostrant el % de la informació ja recuperada en forma de progress bar per la finestra. 2. El programa obté el fitxer .txt original sense pèrdua d’informació. |
| **Errors** |
|  |

|  |
| --- |
| **12. Descomprimir fitxer de tipus .lz78 amb LZ78** |
| **Actor** |
| Programa |
| **Descripció** |
| El Programa recupera el contingut del fitxer .lz78 fent servir l’algoritme LZ78. |
| **Precondicions** |
| (correctesa del fitxer input assegurat per Descomprimir fitxer) |
| **Escenari principal** |
| 1. El programa recupera el contingut del fitxer .lz78 seleccionat fent servir l’algoritme LZ78 i mostrant el % de la informació ja recuperada en forma de progress bar per la finestra. 2. El programa obté el fitxer .txt original sense pèrdua d’informació. |
| **Errors** |
|  |

|  |
| --- |
| **13. Descomprimir fitxer de tipus .jpeg amb JPEG** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa recupera el contingut del fitxer .jpeg fent servir l’algoritme JPEG. |
| **Precondicions** |
| (correctesa del fitxer input assegurat per Descomprimir fitxer) |
| **Escenari principal** |
| 1. El programa recupera el contingut del fitxer .jpeg seleccionat fent servir l’algoritme JPEG i mostrant el % de la informació ja recuperada en forma de progress bar per la finestra. 2. El programa obté el fitxer .ppm original amb una certa pèrdua d’informació. |
| **Errors** |
|  |

|  |
| --- |
| **14. Descomprimir Carpeta** |
| **Actor** |
| Programa |
| **Descripció** |
| El programa permet recuperar el contigut d’una carpeta comprimida. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. L’Usuari indica al programa el path de la carpeta que desitja descomprimir. Seleccionar fitxer/carpeta. 2. El programa farà una crida recursiva intentant descomprimir primer totes les subcarpetes de la carpeta seleccionada. La descompressió d’una carpeta consisteix en descomprimir tots els fitxers continguts en aquesta carpeta i les carpetes filles. Descomprimir fitxer 3. El programa guarda la carpeta descomprimida en el path indicat. |
| **Errors** |
|  |

|  |
| --- |
| **15. Visualitzar el contingut d’un arxiu comprimit** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet que l’Usuari pugui visualitzar el contingut d’un arxiu comprimit sense haver-lo de descomprimir prèviament. |
| **Precondicions** |
| L’arxiu comprimit ha de ser prèviament empaquetat pel mateix programa. |
| **Escenari principal** |
| 1. L’Usuari indica al programa que desitja visualitzar el contingut de l’arxiu comprimit seleccionat. 2. El programa verifica l’arxiu comprimit. 3. El programa identifica l’arxiu comprimit i mostra el contingut de l’arxiu comprimit en una finestra. |
| **Errors** |
| 2a. L’arxiu comprimit no és empaquetat per aquesta mateix programa 2a.1 El Programa notifica a l’Usuari que no pot visualitzar aquest arxiu i cancel·la el procés de visualització. |

|  |
| --- |
| **16. Sortir de la visualització** |
| **Actor** |
| Usuari |
| **Descripció** |
| El Programa permet que l’Usuari pot tancar la finestra de visualització. |
| **Precondicions** |
| L’Usuari està en la finestra de visualització. |
| **Escenari principal** |
| 1. L’Usuari solicita el tancament de la finestra de visualització. 2. El programa tanca la finestra de visualització. |
| **Errors** |
|  |

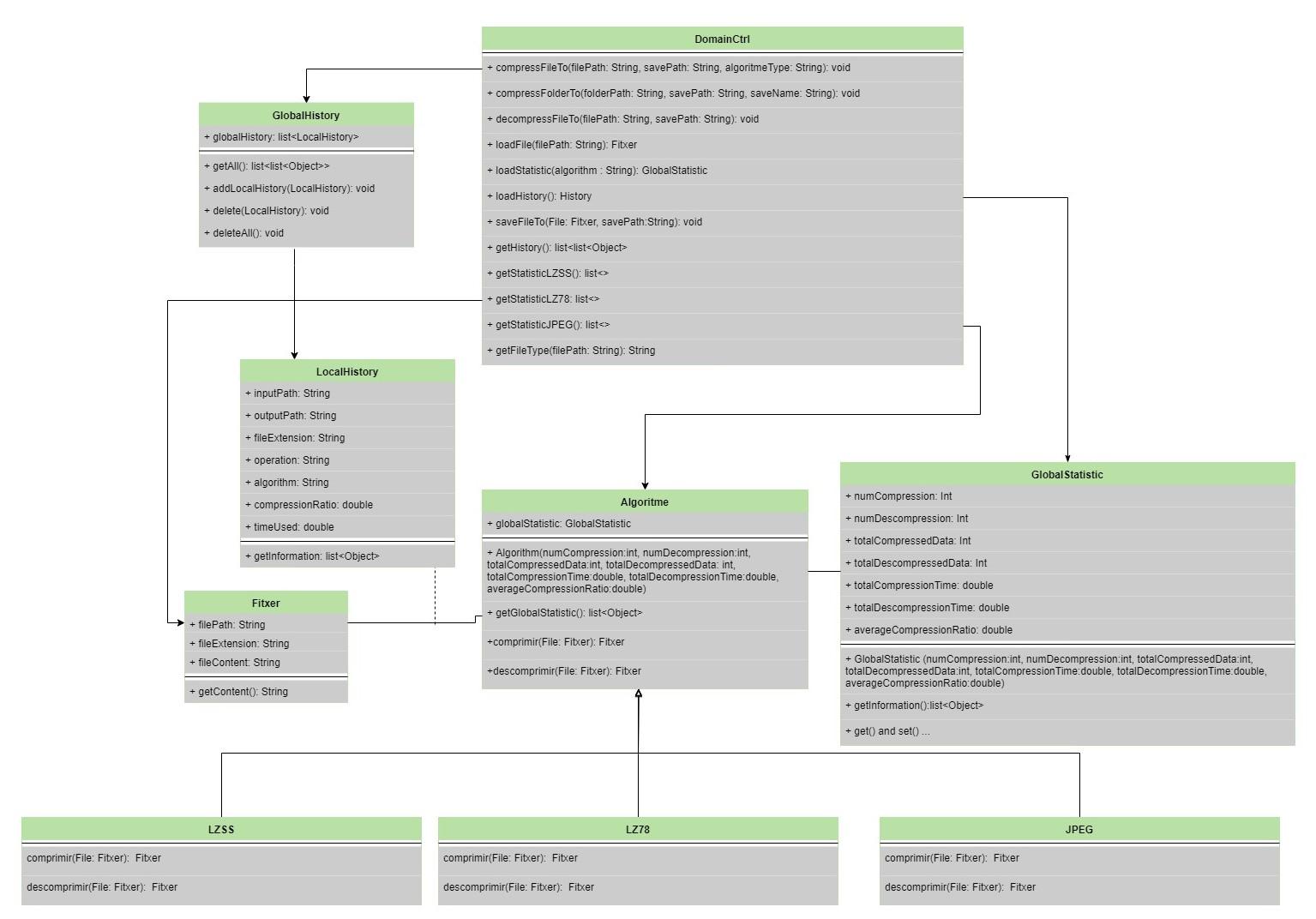
|  |
| --- |
| **17. Comparació** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet que l’Usuari pugui visualitzar un fitxer i la seva corresponent descompressió després d’aplicar el procés de compressió i descompressió amb un cert algoritme escollit manualment. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. L’Usuari indica al programa el path del fitxer del que vol veure la comparació. 2. El Programa realitza el procés de compressió i descompressió. Comprimir fitxer, Descomprimir fitxer amb el fitxer seleccionat. 3. El Programa mostra el contingut del fitxer original i el contingut del fitxer després d’haver aplicat el procés de compressió-descompressió. |
| **Errors** |
| 1a. El fitxer indicat per l’Usuari no existeix.  1a1. El programa notifica a l’Usuari que el fitxer amb el path  indicat no existeix.  2a. El fitxer indicat no és de tipus .txt o .ppm.  2a1. El programa notifica a l’Usuari que només permet comprimir  fitxers del tipus .txt o .ppm. |

|  |
| --- |
| **18. Mostrar estadistiques** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet que l’Usuari pugui visualitzar les estadístiques de l’algoritme que seleccioni prèviament. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. L’Usuari indica al programa l’algoritme del que vol veure les estadístiques globals. 2. El Programa mostra les estadístiques de l’algoritme seleccionat. Mostra la mida original total de tots els fitxers que ha comprimit, a més de la mitja del % de compressió de tots els fitxers comprimits i el temps mitjà que triga en comprimir. |
| **Errors** |
|  |

|  |
| --- |
| **19. Mostrar historials** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet que l’Usuari pot visualitzar un fitxer i la seva corresponent descompressió després d’aplicar el procés de compressió i descompressió amb un cert algorisme escollit manualment. |
| **Precondicions** |
|  |
| **Escenari principal** |
| 1. L’Usuari indica al programa que vol veure l’historial de compressió i descompressió fins a la data. 2. El Programa mostra tots els fitxers i carpetes comprimides o descomprimides fins al moment. De cadascun mostra el path origen, el path destí, el tipus de fitxer, l’algoritme aplicat, si s’ha comprimit o descomprimit, el ratio de compressió en el cas de si s’ha comprimit i el temps trigat en fer el procés. |
| **Errors** |
|  |

|  |
| --- |
| **20. Sortir** |
| **Actor** |
| Usuari |
| **Descripció** |
| El programa permet que l’Usuari pugui tancar aquest. |
| **Precondicions** |
| el programa s’està executant. |
| **Escenari principal** |
| 1. L’Usuari solicita el tancament del programa. 2. El programa termina. |
| **Errors** |
|  |

Diagrama de clases de disseny de Domini



Descripció de les classes de Domini

Classe 1: Fitxer

És una classe per guardar les propietats bàsiques del fitxer que s’ha de comprimir o descomprimir.

Atributs:

* contingut(String): el contingut del fitxer
* extensió(String): el tipus del fitxer
* ruta d’accés(String): path del fitxer

Funcions:

* getContent( ): String

Classe 2: GlobalStatistic

És una classe per guardar les estadístiques de compressió dels diferents algoritmes.

Atributs:

* numCompression(int): Número de compressions
* numDescompression(int): Número de descompressions
* totalCompressedData(int): Total de bytes comprimits
* totalDecompressedData(int): Total de bytes descomprimits
* totalCompressionTime(double): Total de temps trigat en comprimir
* totalDescompressionTime(double): Total de temps trigat en descomprimir
* averageCompressionRatio(double): Ratio de compressió total

Funcions:

* getInformation( ): list<Object>
* getNumCompression( ): int
* setNumCompression( ): Void
* getNumDecompression( ): int
* setNumDecompression( ): Void
* getTotalCompressedData( ): int
* setTotalCompressedData( ): Void
* getTotalDecompressedData( ): int
* setTotalDecompressedData( ): Void
* getTotalCompressionTime( ): double
* setTotalCompressionTime( ): Void
* getTotalDescompressionTime( ): double
* setTotalDescompressionTime( ): Void
* getAverageCompressionRatio( ): double
* setAverageCompressionRatio( ): Void

Classe 3: Algoritme

És una classe de la qual deleguen els diferents algorismes de compressió.

Atributs:

* globalStatistic: GlobalStatistic

Funcions:

* getGlobalStatistic( ): list<Object>
* comprimir(File): Fitxer
* descomprimir(File): Fitxer

Classe 4: LZSS

És una classe que conté l’algoritme LZSS tant per comprimir com per descomprimir.

Funcions:

* comprimir(File): Fitxer
* descomprimir(File): Fitxer

Classe 5: LZ78

És una classe que conté l’algoritme LZ78 tant per comprimir com per descomprimir.

Funcions:

* comprimir(File): Fitxer
* descomprimir(File): Fitxer

Classe 6: JPEG

És una classe que conté l’algoritme JPEG tant per comprimir com per descomprimir.

Funcions:

* comprimir(File): Fitxer
* descomprimir(File): Fitxer

Classe 7: GlobalHistory

És una classe per guardar tots els historials locals com a entrades per tal de fer un historial global.

Atributs:

* globalHistory: list<LocalHistory>

Funcions:

* getAll( ): list<list<Object>>
* addLocalHistory(LocalHistory): Void
* delete(LocalHistory): Void
* deleteAll( ): Void

Classe 8: LocalHistory

És una classe per guardar els diferents processos de comprimir o descomprimir fets pel programa.

Atributs:

* inputPath: String
* outputPath: String
* fileExtension: String
* operation: String
* algorithm: String
* compressionRatio: double
* timeUsed: double

Funcions:

* getInformation( ): list<Object>

Descripció dels algorismes

LZ78

**Funcionament de l'algorisme:**

LZ78 és un algorisme de compressió de dades sense pèrdues. LZ78 aprofita una estructura de dades basada en diccionari està format per índex i string per comprimir les nostres dades. L’algoritme va llegint el contingut del fitxer caràcter a caràcter, comprovant si el caràcter és al diccionari. Si no hi és, afegeix aquest caràcter al String del diccionari i el seu índex és igual la posició on guardar el caràcter. Si hi és, llegeix un altre caràcter i mira si aquesta nova combinació existeix al diccionari. Si continua ser existent al diccionari, llavors afegeix un altre més fins que no està al diccionari i guarda aquest conjunt de caràcters al diccionari.

Cada pas LZ78 enviarà un parell (i, a) a l’output, on ‘i’ és un integer indica índex del String al diccionari i ‘a’ és un char. Si el caràcter hem llegit està al diccionari ‘i’ = 0 i ‘a’ = el caràcter. En canvi, si no hi és, ‘i’ = índex del conjunt dels caràcters més llargs que es pot trobar al diccionari i ‘a’ = el següent caràcter després del String trobat. Un cop finalitzada la compressió, obtindrem un conjunt dels parells(i,a).

Per a descomprimir, llegim l’output del compressió i creiem un altre diccionari buit. Si ‘i’ = 0, significa que ‘a’ no està al diccionari, la paraula que hem comprimit és igual ‘a’ i guardem ‘a’ al diccionari. Si ‘i’ != 0, és a dir, la paraula que hem comprimit és igual el String de l’índex i més caràcter ‘a’.

**Pseudocodi:**

**begin**

initialize a dictionary by empty

initialize String *F* by empty character

initialize int *i*

**while** (*i* < *input*.length) **do**

**begin**

*X =* **read**(*input*)

**if** (*F*+*X* is in the dictionary) **then**

*F* = *F*+*X*

**else**

**begin**

**output**(index of *F* in the dictionary,*X*)

encode *F+X* to the dictionary

initialize phrase *F* by empty character

++*i*

**end**

**end**

**end**

DECODING

**begin**

init dictionary empty

init int *j*

**while** (*j* < *input*.length) **do**

**begin**

read pair of index and character (*i*,*X*) from *input*

**if** (*i* is 0) **then**

**output**(*X*)

put new String *X* into dictionary

**else**

**output**(String of index i + *X*)

put new String String of index i + *X* into dictionary

++*j*

**end**

**end**

**Estructures de dades utilitzades:**

Com a estructura de dades principal, LZ78 utilitza un Hashmap com a diccionari per guardar les dades, perquè Hashmap format per un conjunt dels parells (key, value) que és molt adequat per posat l’índex i el String del diccionari i que és bastant eficient per fer una cercació.

I utilitzem StringBuilder com a output del compressió i descompressió, perquè cada pas de LZ78 obtenim nous resultats i hem d’afegir aquests resultats al output. Si output és un String, per fer aquest afegir ha de reserva un nou espai i copia output a aquest nou espai, per tant, tarda molt temps i ocupa molt espai. En canvi, StringBuilder no fa això que és més eficient.

**Posibles millores:**

Per tal de fer que la relació de compressió sigui més gran, intercanvio ‘i’ com a char. Abans ‘i’ és un integer que ocupa 4 bytes i ara només ocupa 2 byte.

Com que Hashmap només té el mètode .get(Key), per tant costa molt per veure si un String existeix al diccionari o no. Primer, hem de obtener el key i després comparar el String amb value del key. Per optimitzar això, en cas de fer comprimir he posat el String com a Key i l’índex com a value. Així, per fer la comparació podem get el String(key) directament.

LZSS

**Funcionament de l’algorisme:**

L’algorisme de compressió LZSS utilitza una finestra, que es va omplint a mesura que es va llegint el contingut del fitxer de text, fins a una mida de 32 KBytes.

L’algoritme va llegint el contingut del fitxer caràcter a caràcter, comprovant si el caràcter és a la finestra. Si hi és, llavors llegeix un altre i els posa en un String match i comprova que aquest sigui a la finestra. Si no es a la finestra, si la mida del match és menor que 4, escriu un flag a 0 i el primer caràcter al fitxer output, ocupant 2 chars. Si és més gran que 4, escriu un flag a 1, la posició del primer caràcter a la finestra i la mida del match, ocupant 3 chars.

La eficiencia d’aquest algorisme està en que hi hagi conjunts repetits de més de 3 caràcters, per tal de codificar-los en solament 3 chars.

Per a descomprimir, llegim caràcter a caràcter. Si aquest és un 0, llavors vol dir que el següent char és un literal. Si, en canvi, es un 1, llavors vol dir que els següents 2 chars son un conjunt posició i mida. En el cas de ser un literal, el llegim normalment. En el cas de ser un codi, el llegim i obtenim els caràcters de la finestra desde la posició indicada i amb la mida indicada en aquest. En tot dos casos, escribim els caracters tant en la finestra com en el fitxer output.

**Pseudocodi:**

Comprimir:

While(not EOF){

Busquem una referencia a la Finestra.

Si trobem un carácter igual{

Mirem el següent a veure si coincideix

Si coincideix{

Seguim mirant a la finestra per trobat el match més gran

}

Si no hi ha un match més gran{

Si la mida és més gran que 3{

Escrivim un flag a 1

Codifiquem la posició i el número de caràcters que

es repeteixen i l’escrivim

Fiquem tot el match a la finestra

}

Si la mida és més petita que 3{

Escrivim un flag a 0

Escrivim el primer caràcter trobat

El fiquem a la finestra

}

}

}

}

Descomprimir:

While(not EOF){

Si el flag llegit es 0{

Llegim el següent caràcter, l’escrivim i el fiquem a la finestra

}

Si el flag llegit es 1{

Llegim la posició i la mida i escrivim els caràcters de la finestra

desde la posició llegida fins la mida llegida

Fiquem tot el conjunt de caràcters en la finestra

}

}

**Estructures de dades utilitzades:**

Com a estructura de dades principal, el LZSS utilitza un StringBuilder com a FIFO. Com a alternativa, hi havia utilitzar una ArrayList de caràcters, però finalment es va utilitzar un StringBuilder per aprofitar un mètode natiu, *IndexOf(String):Integer*, que busca si un String es a dins d’un altre, retornant l’índex on es troba o -1 si no el troba. D’aquesta manera es simplifica la codificació, i s’aprofita un mètode ja optimitzat de l’estructura de dades.

**Posibles millores:**

On l’algorisme flaqueja es en els fitxers de text molt grans, on es pot veure que l'eficiència comença a reduir-se. Analitzant el problema, hem trobat que el problema es troba a l’hora de llegir el fitxer. Per tant, una posible millora seria utilitzar un FileReader per anar llegint el fitxer de text i d’aquesta manera reduir el temps que triga en comprimir i descomprimir.

Un altre problema que s’observa es que per a cada literal, s’ha de posar un flag a 0 davant, fet que provoca que hi hagin chars innecessaris. Com a solució, es proposa canviar la manera de codificar els literals, de manera que per a cada conjunt seguit de literals, es codifiqui amb un únic flag a 0 en comptes de un per cada literal.

JPEG

**Funcionament de l’algorisme:**

**Estructures de dades utilitzades:**

**Posibles millores:**

Organització de l’equip de treball

El treball ens l’hem organitzat en funció a la càrrega de treball que comportava el nostre algoritme.

El Sergio Pascual s’ha encarregat de fer l’algoritme LZSS. A més, s’ha encarregat de fer les classes Fitxer i Algoritme de la capa de domini i la clase IO de la capa de dades. També ha participat en fer la clase DomainCtrl, introduint l’ús de les seves classes a aquesta. A més, ha fet gran part de la documentació actual del projecte (Descripció de les classes de Domini, Descripció de l’algorisme LZSS i estructuració i creació d’aquest document) i la descripció dels casos d’ús referents a Comprimir. Per acabar, també s’ha encarregat de fer els drivers de les seves classes de Domini, a més del de la clase DomainCtrl, i els jocs de prova per provar-les.

La Sheng Liu s’ha encarregat de fer l’algoritme LZ78. Ha estat la que més classes de Domini ha codificat, entre les que estan la classe LZ78, GlobalStatistic, LocalHistory i GlobalHistori. A més, ha fet la classe DataCtrl i ha participat en fer la classe DomainCtrl. Pel que fa a la documentació, s’ha encarregat de fer la descripció dels casos d’ús referents a les estadístiques i a l’historial. També s’ha encarregat de fer els drivers de les classes de domini que ha implementat, menys les dues referents a l’Historial, i els seus jocs de prova.

El Wei Xiao s’ha encarregat de l'algoritme JPEG. Ha estat el que més càrrega de treball ha tingut degut a la complexitat del seu algoritme, i el que menys classes de domini ha codificat. S’ha encarregat de fer la classe JPEG i les classes auxiliars d’aquesta (BitReader, BitWriter, HuffmanTree i NodePtr). A més ha participat en fer la classe DomainCtrl. Pel que fa a la documentació, s’ha encarregat de fer la descripció dels casos d’ús referents a Descomprimir. També ha fet els drivers i els jocs de prova de totes les classes que ha implementat.