

Documentació UML

Casos d'ús

Comprimir un arxiu: L'usuari pot escollir a través de l'explorador d'arxius l'arxiu desitjat per a comprimir. Després tindrà la possibilitat d'escollir un algoritme de compressió dels disponibles, algoritme LZ78, algoritme LZW, algoritme LZSS, algoritme JPEG, i en el cas que l'algoritme escollit sigui JPEG es podrà escollir el nivell de compressió. Seguidament podrà escollir el directori del disc on es guarda l'arxiu comprimit (per defecte es guarda al mateix directori de l'arxiu sense comprimir), a més al final de la compressió se li mostraran per pantalla les estadístiques locals de la compressió actual, que són temps de compressió, grau de compressió, velocitat de compressió, mida de l'arxiu inicial, mida de l'arxiu final.

Comprimir carpeta/conjunts de fitxers: L'usuari pot escollir a través de l'explorador d'arxius un conjunt d'arxius o una carpeta. Després l'algoritme de compressió se selecciona automàticament per a cada arxiu. Seguidament podrà escollir el directori del disc on es guarda l'arxiu comprimit (per defecte es guarda al mateix directori de l'arxiu sense comprimir), a més al final de la compressió se li mostraran per pantalla les estadístiques locals de la compressió actual, que són temps de compressió, grau de compressió, velocitat de compressió, mida de l'arxiu inicial, mida de l'arxiu final.

Descomprimir: L'usuari escull l'arxiu o carpeta a descomprimir a través de l'explorador d'arxius i es descomprimeix fent servir l'algoritme de compressió adequat, a més al final de la descompressió se li mostraran per pantalla les estadístiques locals de la descompressió actual, que són temps de descompressió, grau de descompressió, velocitat de descompressió, mida de l'arxiu inicial, mida de l'arxiu final.

Generació estadístiques globals: L'usuari podrà accedir a les estadístiques de totes les compressions fetes per l'aplicació a partir d'un bot. Les estadístiques globals a les quals tindrà accés seran les estadístiques locals en mitjana, el nombre total d'arxius, el nombre de compressions per cada algoritme específic.

Algoritme LZ78: Algoritme lossless per comprimir i descomprimir textos. Aquest algoritme està especialitzat per l'ús de textos, encara que pot comprimir imatges.

Algoritme LZW: Algoritme lossless Algoritme lossless per comprimir i descomprimir textos. Aquest algoritme està especialitzat per l'ús de textos, encara que pot comprimir imatges.

Algoritme LZSS: Algoritme lossless per comprimir i descomprimir textos. Aquest algoritme està especialitzat per l'ús de textos, encara que pot comprimir imatges.

Algoritme JPEG: Algoritme lossy per comprimir i descomprimir imatges. Aquest algoritme està especialitzat per l'ús d'imatges, encara que pot comprimir textos.

Automàtic: El sistema automàticament escull l'algoritme més eficient per cada arxiu introduït i el tornarà comprimit en el cas que s'hagi de comprimir o descomprimit en el cas que s'hagi de descomprimir.

Tancar aplicació: L'usuari decideix tancar l'aplicació.

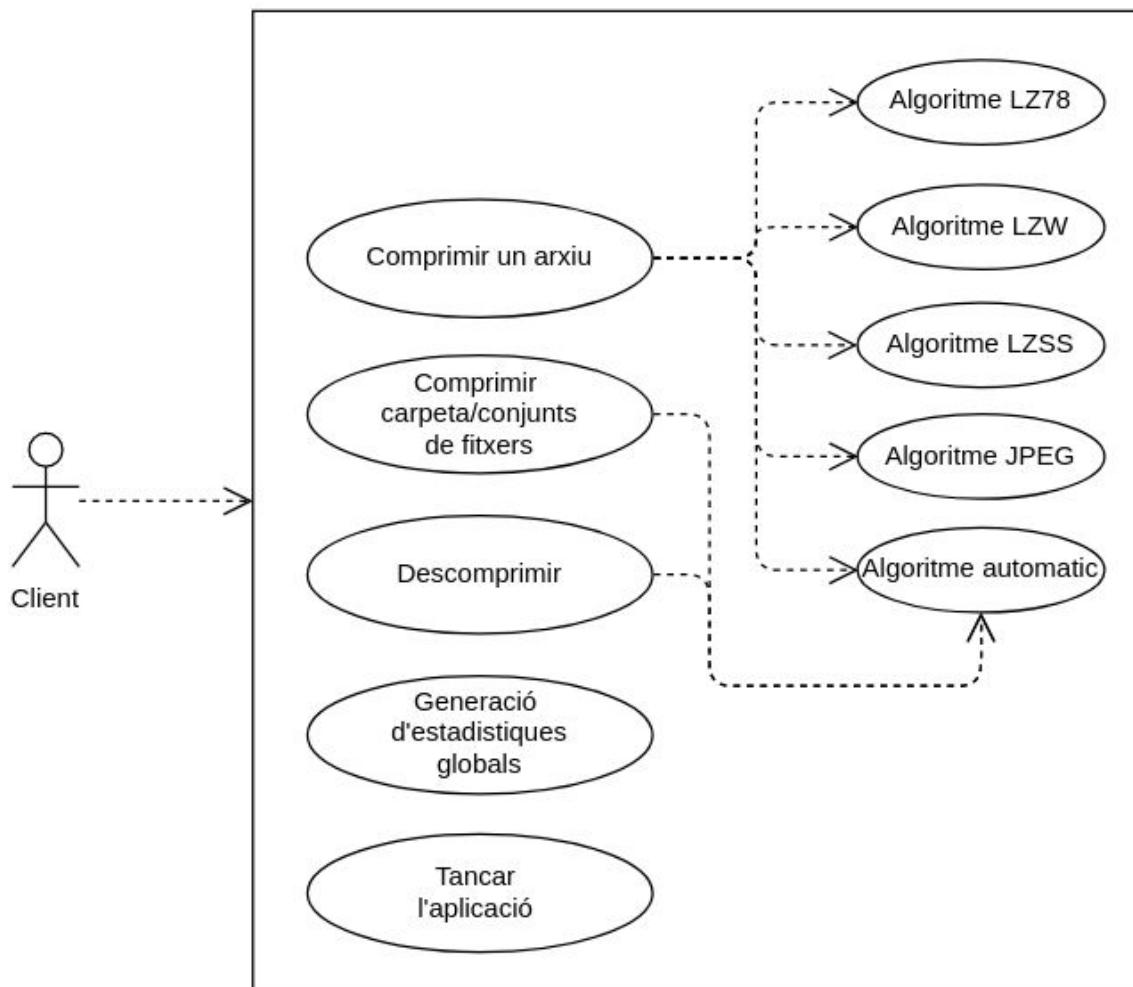


Diagrama de classes

Arxiu: La classe arxiu es la classe encarregada de representar un arxiu, les instàncies d'aquest classe s'identifiquen pel nom de l'arxiu i el seu directori. Aquestes tenen un contingut representat en forma de string.

Estadística: La classe estadística es la classe encarregada de gestionar les estadístiques sobre la compressió i descompressió d'arxius.

Estadística global: Las estadístiques globals son aquelles que representen una col·lecció de totes les compressions i descompressions que s'han fet al llarg de l'execució del programa.

Estadística local: Les estadístiques locals son aquelles que estan associades a un arxiu i un algoritme i les quals mostren una avaluació de la compressió feta.

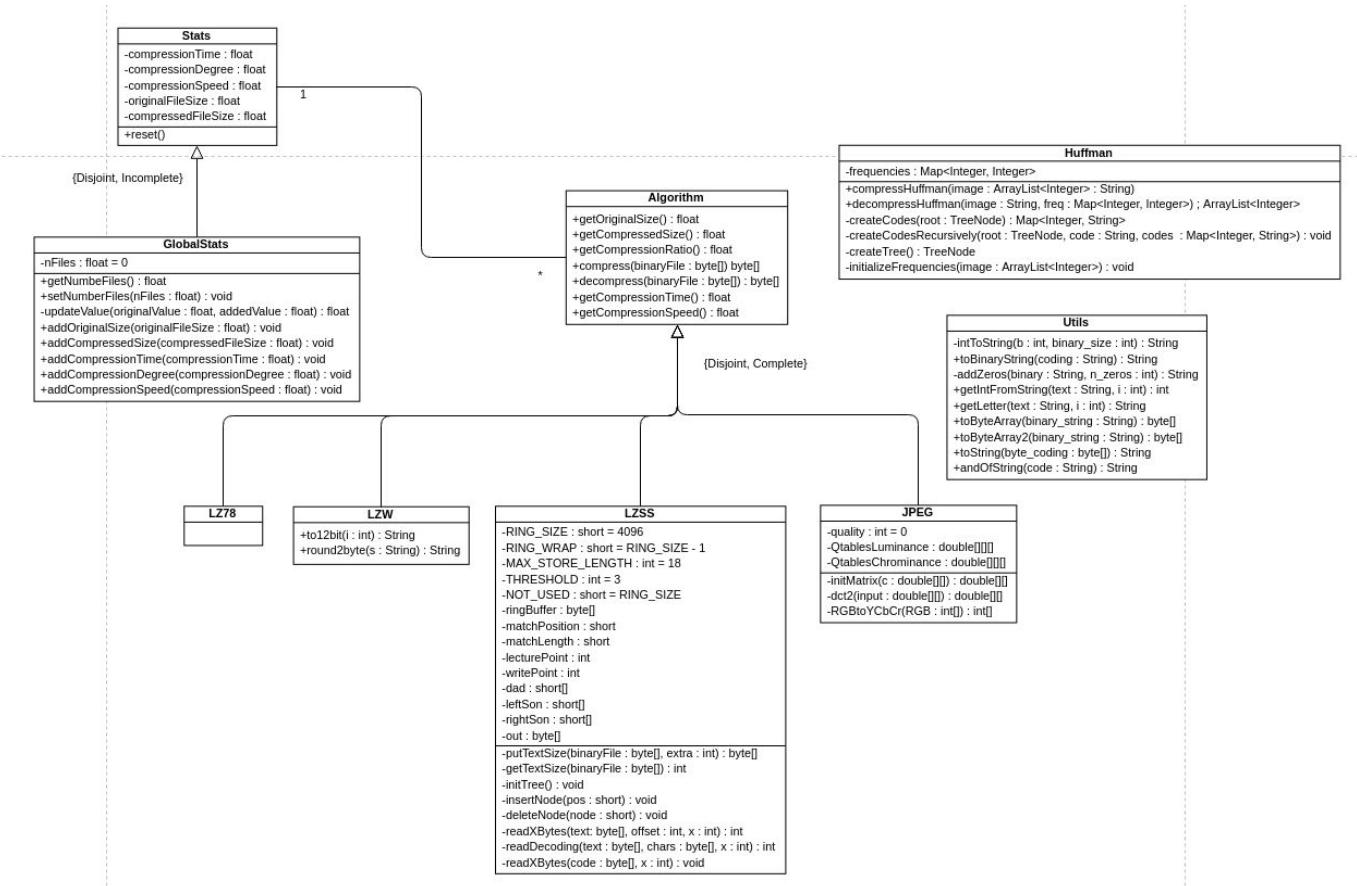
Algoritme: La classe algoritme és la representant dels diferents algorismes de compressió i es l'encarregada de comprimir o descomprimir un arxiu específic.

LZW: Representa un tipus d'algoritme que s'utilitza per comprimir arxius.

LZ78: Representa un tipus d'algoritme que s'utilitza per comprimir arxius.

LZSS: Representa un tipus d'algoritme que s'utilitza per comprimir arxius.

JPEG: Representa un tipus d'algoritme que s'utilitza per comprimir arxius.



Relació classes implementades

	Adrián Álvarez Martínez	Jaume Bernaus Serra	Pol Monroig Company	David Plana Santos
LZ78	-	-	Sí	-
LZW	-	Sí	-	-
LZSS	-	-	-	Sí
JPEG	Sí	-	-	-
Utils	Sí	-	Sí	-
Stats			Sí	
GlobalStats	-	-	Sí	-
Algorithm				
Huffman	Sí	-	-	-
DataCtrl				
DomainCtrl				
PresentationCtrl				
DriverLZ78	-	-	Sí	-
DriverLZW	-	Sí	-	-
DriverLZSS	-	-	-	Sí
DriverJPEG	Sí	-	-	-

Descripció algorismes implementats

LZ78: La idea de tots els algoritmes Lempel-Ziv és que si un text no és uniformement aleatori llavors una paraula que ja hem vist és més probable que aparegui que una que no hem vist mai. Específicament LZ78 funciona construint un diccionari de paraules que s'anomenen frases, una frase

és una concatenació de caràcters de mida més gran o igual a un. L'algoritme comença agafant la frase més petita del text que no haguem vist abans. Com que inicialment el diccionari està buit, la primera frase serà el primer caràcter. Seguidament agafem la següent frase que no hagèm vist, de tal manera que si ja hem vist una frase el que fem és afegir el següent caràcter, fins que aquesta no estigui present al diccionari. Per cada frase w que no hagèm vist la podem subdividir en dues subfrases tal que $w=xy$ i $|y|=1$. Això significa que si $|w|=1$, una frase format per un sol caràcter, tenim que x serà la paraula buida, que per precondició és una frase que sempre hem vist. Seguidament per cada paraula w que guardem al diccionari guardarem el índex de la posició de la paraula x . Quan acabem, la compressió representarà una sèrie d'índexs i caràcters. Per descomprimir simplement haurem de reconstruir el diccionari i les frases de manera recursiva. S'ha utilitzat un diccionari perquè hem de poder buscar i guardar frases úniques.

LZW:

Com la majoria dels mètodes de compressió l'algoritme LZW identifica les cadenes de caràcters repetides per tal de crear una taula d'equivalències i pode's assignar codis assignar codis més breus. El mètode LZW en una sola passada pel text pot crear el diccionari i codificar el text. A no és necessari passar aquest diccionari junt amb la codificació ja que el diccionari es reconstrueix a mesura que els codis es van descomprimint.

Això passa degut a que el diccionari comença inicialitzat amb 256 entrades. Es van afegint codificacions per cada parella de caràcters consecutius i més endavant de n caràcters, tot hi que aquesta codificació només s'acabara utilitzant si és torna a trobar aquests grups de caràcters en el text i d'aquesta forma s'aconseguiria la compressió.

Les estructures de dades per implementar l'algorisme han sigut un Map pel diccionari, ja que per cada entrada s'ha de guardar la cadena de caràcters, el seu codi i s'ha de poder buscar si el codi ja existeix. Un string per tal de guardar els codis en binari del output.

LZSS:

Aquest algoritme consisteix en emmagatzemar els 4096 caràcters més recents del text a descomprimir, i a mesura que es llegeixen els caràcters buscarem coincidències de la seqüència de caràcters nous amb els ja llegits, si coincideixen com a mínim 3 caràcters nous, i com a màxim 18, amb els ja llegits codificarem aquesta cadena de caràcters com a dos bytes que indiquen la posició en la que es troba la coincidència amb el buffer on tenim emmagatzemats els ja llegits i indicaran també la longitud de caràcters que s'han repetit, i d'aquesta manera disminuir la mida del fitxer, per tal de poder descomprimir després el fitxer tindrem un byte cada certa mida, els bits del qual valdran 1 o 0, que cada bit indica si es llegirà un caràcter que es pot escriure directament, o si haurem de llegir dos bytes per a obtenir la seqüència codificada explicada anteriorment, i d'allí obtenir els caràcters que hi havien escrits originalment.

L'estructura de dades d'aquest algoritme està tota feta amb vectors de bytes, 3 vectors que representen un arbre binari, que intentarà ser complet, el qual indicarà les diferents seqüències de caràcters començades per un determinat caràcter dintre de 256 possibilitats, els possibles en ASCII. Hi ha un vector que consisteix en un buffer que emmagatzema els 4096 caràcters més recents, i un altre vector on es guardarà tota la codificació del codi que retorna la funció, al comprimir o al descomprimir.

JPEG: