Changelog de PROPresor respecto a 1a entrega

En este documento se presentan los cambios realizados en el código y documentación respecto a la primera entrega, sin tener en cuenta todos los añadidos realizados en la segunda entrega.

Println en controladores

Como añadido, se han eliminado todos los println que se hacían en capas ajenas a la de presentación, dando paso a la versión definitiva del programa, al cual no muestra ninguno, ya que todo se muestra a través de diálogo.

DatosProceso

Se ha añadido un nuevo dato para mostrar la velocidad del proceso en compresión y descompresión.

GestorEstadísticas

Se ha añadido un nuevo método que devolverá la velocidad media de un algoritmo y proceso concreto.

CtrlEstadística

Se ha creado una nueva clase DatosEstadistica donde se guardan todos los datos que se producen al generar estadísticas y pasarlo como parámetro con tal de visualizar este contenido. Posteriormente, el CtrlPresentacion obtendrà esta información.

JPEG

El algoritmo ha sufrido dos importantes cambios relativos a la primera entrega. Por un lado, se ha usado la librería estándar IntStream para paralelizar bucles del programa, y por otro, se ha refactorizado el código para que haya un análisis descendente del código y que sea fácilmente legible, delegando responsabilidades a funciones privadas específicas de cada apartado del proceso del algoritmo.

Paralelización de bucles

Se han paralelizado los bucles proceso de los bloques 8x8 de las imágenes y los de cálculo de DCT, los cuales eran los principales causantes del alto coste en complejidad computacional. Se ha conseguido mejorar la eficiencia del código en un 700%. Esto es, el código se ejecuta aproximadamente unas 8 veces más rápido.

Estas mejoras de paralelización se pueden encontrar fácilmente al ver las invocaciones de la clase IntStream, las cuales se paralelizan invocando a su función parallel().

Refactor del código y análisis descendente

Se han refactorizado las funciones de comprimir y descomprimir para que cada parte del algoritmo tenga su propia función, asignando responsabilidades a funciones privadas, y facilitando el análisis descendente del código además de su visibilidad.

LZW

El algoritmo se ha comentado para explicar mejor el funcionamiento.

LZ78

Se ha modificado ligeramente la explicación del motivo de uso del árbol Trie en el algoritmo.

Diagrama de casos de uso

Se han independizado los casos de uso de entre sí ya que las interacciones explicadas no permitían realmente la navegación entre una a otra.
Se han fusionado los casos de uso de comprimir carpetas ya que el algoritmo de texto siempre es seleccionado automáticamente.
Se han modificado las explicaciones para indicar que el usuario es capaz de decidir el nombre del archivo/carpeta resultante.

Diagrama de clases

Se ha corregido el nombre de la clase Algoritmo.
Se ha cambiado el package de la clase Algoritmo a Enumeration y de la excepción FormatoErroneoException a Exception para indicar su independencia entre capas.