

Nombres: Andrea Bayona y David Calle

TAD VOLÚMEN

1. Conjunto mínimo de datos:

Se encuentran:

- imágenes, colección de Imagen2D, una lista que tiene en su interior las imágenes creadas.
- tamaño, entero, cantidad de imágenes que tiene la colección.

2. Operaciones:

- **Volumen():** Construye un objeto volumen sin imágenes.
- **Volumen(nombreBase,total):** inicializa la colección de imágenes usando como criterios el nombre base y la cantidad.
- **getTamaño():** retorna el tamaño que tiene la lista de las imágenes.
- **setTamaño(int tam):** cambia el valor de tamaño por tam.
- **getImágenes():** Retorna la colección de imágenes.
- **proyeccionXPromedio():** Retorna una imagen que representa la proyección en x de la colección de imágenes usando el criterio promedio.
- **proyeccionYPromedio():** Retorna una imagen que representa la proyección en y de la colección de imágenes usando el criterio promedio.
- **proyeccionZPromedio():** Retorna una imagen que representa la proyección en z de la colección de imágenes usando el criterio promedio.
- **proyeccionXMaximo():** Retorna una imagen que representa la proyección en x de la colección de imágenes usando el criterio máximo.
- **proyeccionYMaximo():** Retorna una imagen que representa la proyección en y de la colección de imágenes usando el criterio máximo.
- **proyeccionZMaximo():** Retorna una imagen que representa la proyección en z de la colección de imágenes usando el criterio máximo.
- **proyeccionXMinimo():** Retorna una imagen que representa la proyección en x de la colección de imágenes usando el criterio mínimo.
- **proyeccionYMinimo():** Retorna una imagen que representa la proyección en y de la colección de imágenes usando el criterio mínimo.
- **proyeccionZMinimo():** Retorna una imagen que representa la proyección en z de la colección de imágenes usando el criterio mínimo.
- **proyeccionXMediana():** Retorna una imagen que representa la proyección en x de la colección de imágenes usando el criterio mediana.

- **proyeccionYMediana():** Retorna una imagen que representa la proyección en y de la colección de imágenes usando el criterio mediana.
- **proyeccionZMediana():** Retorna una imagen que representa la proyección en z de la colección de imágenes usando el criterio mediana.

TAD IMAGEN2D

1. Conjunto mínimo de datos:

- imagen, colección de colección de enteros, matriz de números en la cual cada número representa un pixel.
- formato, cadena, representa el código en el que viene dada la imagen.
- fila,entero:, representa el número de filas que tiene la lista de listas
- column,entero, representa el número de columnas que tiene la lista de listas

2.Operaciones:

- **Imagen2D():** Inicializa una imagen sin valores.
- **Imagen2D(pNombre):**Inicializa la imagen que se encuentra en un archivo con nombre pNombre.
- **getFila():** Retorna la cantidad total de filas dentro de las imágenes.
- **setFila(tam):**cambia el valor de fila por tam.
- **getColumna():** Devuelve la cantidad total de columnas dentro de las imágenes.
- **setColumna(tam):**Permite darle el valor tam al atributo columna dentro de la clase Imagen.
- **getImagen():** retorna la matriz que representa la imagen.
- **setImagen(nuevo):**Permite cambiar la dirección por la que entra como parámetro a la cual apunta el apuntador dentro de imagen.
- **getFormato():** Devuelve el formato en el que se encuentra la imagen.
- **setFormato(nuevo):** Permite darle el valor de nuevo al formato que tiene la imagen,. el formato queda como nuevo.
- **imprimirImagen():** Permite imprimir la matriz que representa la imagen.
- **exportarImagen(nom_arch):** exporta la imagen en formato pgm.
- **buscarIntensidad(intensidad, lista):** Devuelve la posición en la que se encuentra la intensidad a buscar.
- **cargarHuffman(nombreArchivo):**Carga un archivo Huffman con nombre nombreArchivo y devuelve si se pudo o no cargar.
- **exportarHuffman(nom_arch):** Exporta el Huffman recibiendo como parámetro el nombre del archivo en el que este se encuentra.

- **calcularListaIntensidades():** Calcula las intensidades y las guarda en un lista, la cual retorna.
- **cargarArchivo(nombre):** Carga el archivo con el nombre que recibe como parámetro y retorna verdadero o falso dependiendo de si logró cargarlo

TAD NODO

1.Conjunto Mínimo de Datos:

- contenido, tipo plantilla(T), representa la información que está almacenada en el nodo.
- hijoDerecho, apuntador a Nodo, representa al Nodo hijo derecho del nodo.
- hijoIzquierdo, apuntador a nodo, representa al Nodo hijo izquierdo del nodo.

Operaciones:

- **getContenido(),** retorna el valor del contenido del nodo.
- **setContenido(pContenido),** cambia el valor del contenido, por pContenido.
- **getHijoIzquierdo(),** retorna el apuntador al Nodo hijo izquierdo del Nodo.
- **getHijoDerecho(),** retorna el apuntador al nodo hijo derecho del Nodo.
- **setHijoIzquierdo(nuevo),** cambia el valor del apuntador del hijo izquierdo por nuevo.
- **bool esHoja(),** retorna si el nodo no tiene hijos.

TAD ÁRBOL

1.Conjunto Mínimo de Datos:

cabeza, apuntador a nodo, contiene un apuntador a la cabeza del árbol.

2.Operaciones:

- **Arbol(),** crea un árbol cuya cabeza es vacía
- **getCabeza(),** retorna la cabeza del árbol
- **setCabeza(c),** cambia la cabeza por c.
- **eliminarArbol(),** borra cada uno de los nodos del árbol.
- **operator(in):** retorna si el contenido de la cabeza del árbol es mayor al contenido de la cabeza del árbol in.

TAD INTENSIDAD

1.Conjunto Mínimo de Datos:

- Valor, entero que representa el valor de píxel más grande de la imagen, es decir la intensidad máxima de la imagen.
- Intensidad, entero que representa el número de veces que se repite un valor.

2.Operaciones:

- **Intensidad(valor,frecuencia):** Inicializa la intensidad con su respectivo valor y frecuencia.
- **getValor():** Devuelve el valor de la intensidad.
- **setValor(pValor):** Cambia el valor de la intensidad por pValor.
- **getFrecuencia():** Devuelve el valor de la intensidad.
- **setFrecuencia(pFrecuencia):** Cambia el valor de la intensidad por pFrecuencia.
- **operator(in):** determina si la intensidad in es menor que la frecuencia de la intensidad actual.

Otras funciones Importantes:

- **reverse(s):** Crea una nueva cadena con la palabra invertida.
- **itoa(n,s):** Permite convertir un entero en carácter, recibiendo el entero que debe transformar y el carácter donde guardará el dato convertido
- **preOrden(nodo):** Imprime el valor de la frecuencia y la intensidad en pre-orden
- **hoffman(lista):** Crea un árbol de huffman a partir de la lista de intensidad lista.
- **codificarHoffman(nodo,valor,cadena):** retorna la representación del valor valor del árbol de huffman.
- **decodificarHoffman(nodo,lectura):** Guarda en un árbol la representación de un arbol de huffman guardada en un archivo.
- **decodificarValor(nodo,codificacion):** decodifica la cadena codificación en el arbol.
- **codificarValor(nodo,valor,retorno):** retorna la codificación de valor en un arbol de huffman.

class Class Model

