

Manual Técnico

Byron Estuardo Solís González

201906588

Primer Semestre – Abril 2022

INTRODUCCIÓN

Se desarrolló una aplicación en el lenguaje de programación Java, la cual le permite generar árboles e imágenes a partir de datos se le ingresen, además presenta varias opciones para personalizar y visualizar dichas imágenes.

Para el manejo de los datos se desarrollaron cuatro estructuras de datos, árbol binario, árbol binario AVL, matriz dispersa, y lista doblemente enlazada; Estas al complementarse con la herramienta de creación de grafos Graphviz le ofrecen al usuario una forma amigable de visualizar sus datos.

Sumado a esto, la aplicación cuenta con interfaz grafica desarrollada con la librería de Java Swing.

DATOS TECNICOS

- Lenguaje utilizado para la aplicación : Java
- IDE utilizado: NetBeas IDE 8.2
- Lenguaje utilizado para reportes: HTML
- Formato de texto utilizado para el ingreso de datos: JSON
- Librería utilizada para la lectura de archivos JSON : json-simple-1.1
- Herramienta utilizada para la creación de grafos: Graphviz

FLUJO DE TRABAJO

1. Ingreso de datos

La aplicación se alimenta de los datos que el usuario le proporciona a través de archivos de entrada con estructura JSON, estos contienen los datos de las capas, imágenes y álbumes de un usuario.

Para la lectura de estos se utilizó la librería de Google Json Simple, a continuación se presenta la estructura de los archivos de entrada.

a. Capas

```
[
  {
    "id_capa": 0,
    "pixeles": [
      {
        "fila": 25,
        "columna": 31,
        "color": "#FFCC33"
      },
      {
        "fila": 38,
        "columna": 31,
        "color": "#FFCC40"
      }
    ]
  },
  {
    "id_capa": 1,
    "pixeles": [
      {
        "fila": 21,
        "columna": 37,
        "color": "#FFCC27"
      }
    ]
  }
]
```

b. Imágenes

```
[
  {
    "id":3,
    "capas":[0,1,2,3,4]
  },
  {
    "id":1,
    "capas":[3,4]
  }, {
    "id":2,
    "capas":[0,2,4]
  },
]
```

c. Álbumes

```
[
  {
    "nombre_album":"Album 1",
    "imgs":[1,3,5] ← id de las imagenes
  },
  {
    "nombre_album":"Album 2",
    "imgs":[2]
  },
  {
    "nombre_album":"Album 3",
    "imgs":[]
  }
]
```

2. Funcionalidades:

a. Generación de imágenes

- Por recorrido limitado: En esta opción se indicará el número de capas a utilizar y el tipo de recorrido que se desea para generar la imagen: Preorden, inorden, postorden, dicho recorrido se realiza sobre el árbol de capas.

- Por Árbol de Imágenes: Se ingresará el id de la imagen el cual deberá buscarse y según el árbol de capas que posea se deberá generar la imagen utilizando un recorrido en amplitud.
 - Por capa: Se ingresará el id de una o varias capas a graficar y se buscarán en el árbol binario correspondiente para luego generar su imagen apilando dichas capas.
- b. Visualización del estado de las estructuras: Dentro de la interfaz gráfica del programa se debe incluir un módulo de visualización de estructuras, se debe mostrar al cliente el conjunto de estructuras disponibles, al seleccionarse una de ellas se generará un reporte gráfico (imagen de la estructura), la cual se mostrará en la pantalla. Las estructuras que es posible visualizar son las siguientes
- Árbol de imágenes
 - Árbol de capas
 - Listado de álbumes
 - Capa aislada
 - Ver imagen y árbol de capas

ESTRUCTURAS DE DATOS

1. **Matriz Dispersa:** Esta estructura se utiliza para guardar la información de una capa, es decir las posiciones y colores de los píxeles utilizados para generar imágenes.

Esta cuenta con una serie de nodos enlazados (se un objeto único para todas las posiciones de la matriz), que se rigen a través de una fila y columna de nodos, ambas ordenadas.

Métodos importantes:

- Combinar matrices: Une 2 matrices dispersas en una sola, tomando como base una de ellas, a la que le son sobrepuestos los nodos de la otra.
- Generar DTO: Genera un archivo plano con estructura DTO de la matriz que se le pase como parámetro
- Generar imagen: Convierte archivo DTO en una imagen de tipo JPG.

2. **Árbol binario de búsqueda (ABB):** Esta estructura se utilizará para guardar todas las capas con las cuales se pueden generar distintas imágenes.

Contiene todas las capas un usuario, guardándolas a medida que son leídas del archivo de entrada.

Métodos importantes:

- Recorridos: Se cuenta con 3 métodos recursivos para el recorrido del grafo, PreOrden, Inorden y PostOrden, estos tienen 2 variantes, la primera devuelve una cadena de texto con los ids de las capas recorridas y la segunda devuelve una matriz resultando de la unión (de par en par) de las matrices de las capas recorridas.
- Buscar: Recibe un id, y si se encuentra la capa a la que pertenece se le retorna, de lo contrario retorne Null.
- Generar DTO: Genera un archivo plano con estructura DTO del ABB que se le pase como parámetro
- Generar imagen: Convierte archivo DTO en una imagen de tipo JPG.

3. **Árbol AVL:** Esta estructura se utilizará para guardar cada una de las imágenes de un cliente

Este contiene todos los métodos del Árbol binario de búsqueda, además de 4 métodos de rotación para mantener el árbol equilibrado, estos son ejecutados luego de una inserción al árbol, se le manda como primer parámetro el nodo insertado, verifica si el subárbol del que este es raíz está equilibrado, si no lo está, hace las rotaciones necesarias, si lo está, pasa al padre y repite el proceso hasta llegar a la raíz

Métodos importantes:

- Rotación simple derecha
- Rotación simple izquierda
- Rotación doble izquierda
- Rotación doble derecha
- Generar DTO: Genera un archivo plano con estructura DTO del ABB que se le pase como parámetro
- Generar imagen: Convierte archivo DTO en una imagen de

4. **Lista doblemente enlazada:** Esta estructura se utilizará para guardar el listado de álbumes por cada cliente, a su vez cada álbum posee una sub lista de imágenes de tipo de lista simple con las imágenes de dicho álbum.

- Generar DTO: Genera un archivo plano con estructura DTO del ABB que se le pase como parámetro
- Generar imagen: Convierte archivo DTO en una imagen de

Conclusiones

La implementación de estructuras de datos no lineales a cierto tipo de proceso puede agilizar enormemente la eficiencia de estos, tanto en tiempo de procesamiento como en espacio en memoria, vale totalmente la pena aprender el funcionamiento de estas y las áreas adecuadas para su implementación.

Se debe incluir en la estimación del tiempo de programación, una buena parte de tiempo para corregir errores y solucionar imprevistos, ya que si no puede que el proyecto no se termine a tiempo.

.