

Universidad de San Carlos de Guatemala
División de Ciencias de la Ingeniería
Ingeniería en Ciencias y Sistemas
Estructura de datos 2021
Catedrático: Ing. Oliver Ernesto Sierra Pac
Auxiliar: Juan Pablo Valiente González



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Generador de imágenes por capas

Proyecto 1

Contenido

Objetivos

General

Específicos

Descripción

Capa

Imagen

Usuarios

Carga Masiva

Carga masiva de Capas

Carga masiva de imágenes

Carga masiva de usuarios

Funcionalidad

Generación de imágenes

Por recorrido limitado

Por lista de imágenes

Por capa

Por usuario

CRUD

Usuarios

Imágenes

Estado de la memoria

Ver lista de imágenes

[Ver árbol de capas](#)

[Ver capa](#)

[Ver imagen y árbol de capas](#)

[Ver árbol de usuarios](#)

[Observaciones](#)

[Entregables](#)

[Fecha y modo de entrega:](#)

Objetivos

General

- Que el estudiante aplique los conocimientos adquiridos sobre estructuras de datos, en aplicaciones que permitan manipular la información de forma óptima para el ordenador y permite mostrarla de forma sencilla.

Específicos

- Utilizar estructuras de datos como matrices y árboles para almacenar la información en tiempo de ejecución.
- Generar reportes que representen el estado de la memoria de forma gráfica.
- Generar reportes que permitan ver la información almacenada de forma gráfica y de fácil interpretación.

Descripción

El sistema consiste en un generador de imágenes por capas, la aplicación contará con un conjunto de capas cargadas previamente y almacenadas en memoria para ser utilizadas, estas capas contendrán imágenes hechas con pixeles, cada capa contendrá elementos simples o compuestos, al colocar una capa sobre otra estas irán formando una imagen más completa.

El sistema es capaz de generar imagen seleccionando las capas deseadas.

Capa

Una capa contiene la definición de elementos simples por ejemplo una piedra, un muro o elementos compuestos como una casa, un árbol. Las capas se almacenarán utilizando matrices dispersas dado que no existe una dimensión fija para las capas, en cada nodo se almacenará el color del píxel en Hexadecimal.

Por ejemplo

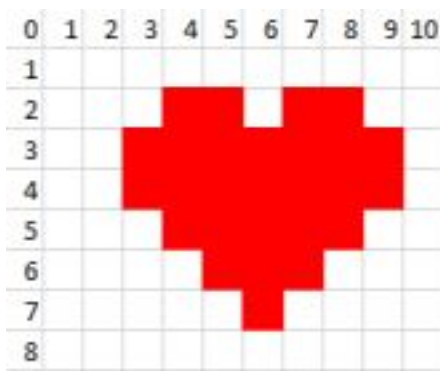


Ilustración 1

Una imagen como la anterior, si se generara una imagen únicamente de esta capa el resultado debería ser:



En las áreas donde no existe color se mostrará blanco al generar la imagen. Las capas son únicas por lo cual una capa puede aparecer en cualquier imagen solo una vez estas tendrán un id que las identificará.

Las capas deben almacenarse en un árbol binario de búsqueda (ABB)

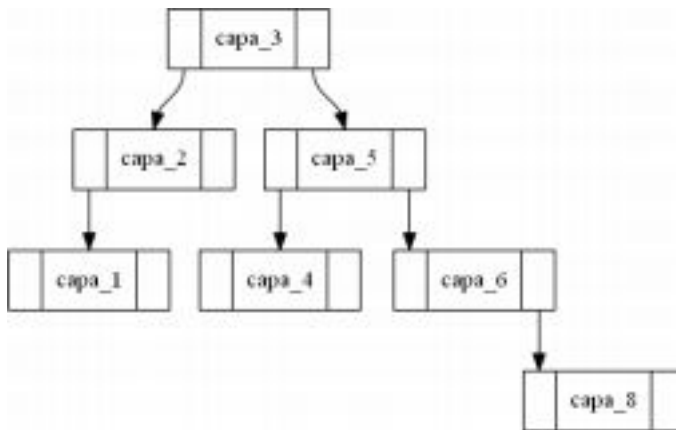


Ilustración 4

Imagen

Una imagen es el resultado de superponer una o más capas, las imágenes se pueden generar de diferentes maneras que serán definidas más adelante.

La aplicación contará con un conjunto de imágenes predefinidas, cada imagen a la vez tiene una lista de capas que conforman la imagen, se almacenará en una lista circular doblemente enlazada ordenada según el id de la imagen.

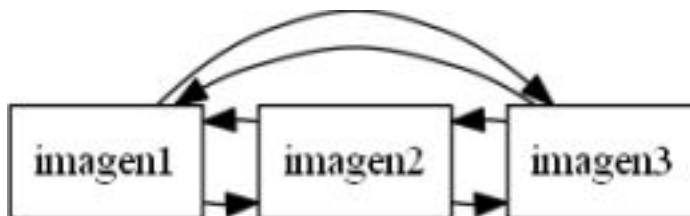


Ilustración 5

Cada imagen a su vez tendrá una lista de capas, el orden en el que se insertan las capas indica la forma en que se superpondrán las capas.

Por ejemplo, viendo la ilustración 6, en la imagen2 la capa4 se colocará primero y sobre esta se debe colocar la capa1

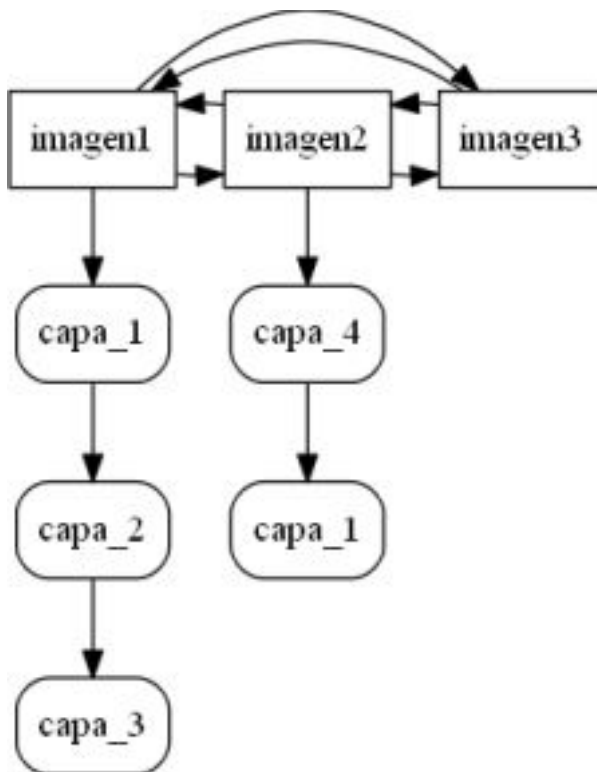


Ilustración 6

La lista de capas de cada imagen debe apuntar a la capa respectiva en el árbol, esto será un apuntador, no una copia de la información.

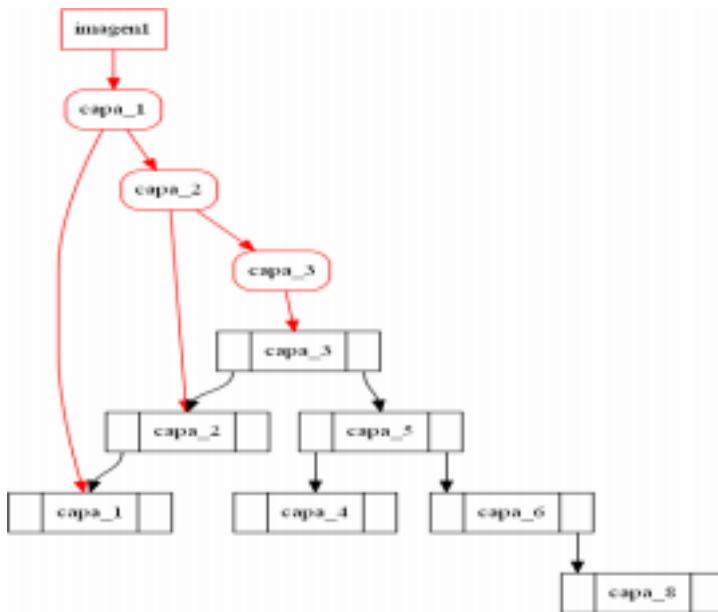


Ilustración 7

Usuarios

Dentro de la aplicación se manejan usuarios, los usuarios son capaces de agregar imágenes, los usuarios se almacenarán en un árbol binario.

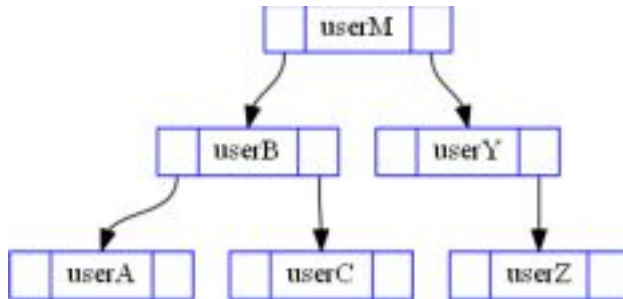


Ilustración 8

Los usuarios pueden tener una lista de imágenes que han registrado. La lista de imágenes de los usuarios puede ser de enlace simple o doble quedando a su elección.

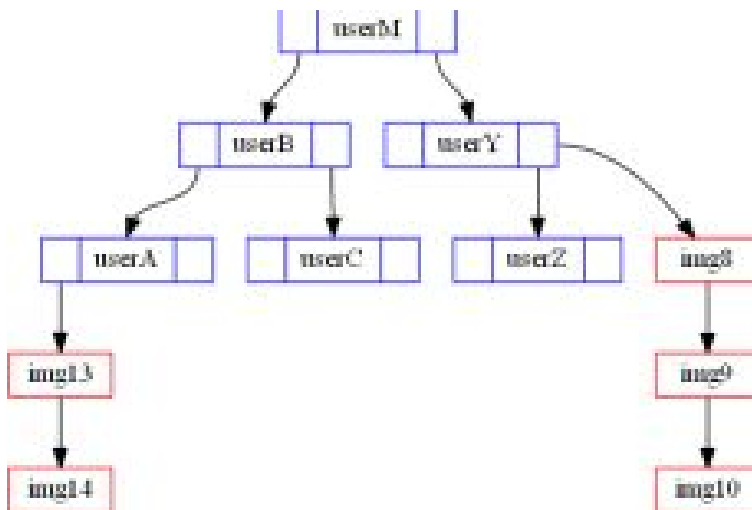


Ilustración 9

Carga Masiva

Para poder hacer más rápida la aplicación cada una de las secciones de la aplicación contará con una carga masiva desde un archivo, estos serán los siguientes

Carga masiva de Capas

Existirá un archivo de entrada que permitirá la carga masiva de capas, siguiendo la sintaxis:

```

id {
  fila,columna,color;
  ...
  fila columna, color;
}

id {
  fila,columna,color;
  ...
  fila columna, color;
}

```

Ver ejemplo adjunto de archivo de capas. La extensión del archivo es *.cap.

Carga masiva de imágenes

La sintaxis del archivo de carga para imágenes será:

```
id{ idcapa, ... ,idcapa }
```

```

1{2,3,4}
2{4,1}
3{}
100{4,2,3,1}
50{1}

```

El archivo que contiene la definición de las imágenes tendrá la extensión *.im. En el caso de las imágenes que no poseen capas, estas solo generaran un pixel negro al momento de generarse la imagen.

Carga masiva de usuarios

Ejemplo de archivo de carga para usuarios contará primero con el nombre del usuario seguido de 2 puntos y una lista de imágenes separadas por coma, finalizando con un punto y coma

```

userM;;
userB;;
userA:13,14;
userY:8,9,10;

```

El orden de la carga de archivos será el siguiente:

1. capas
2. imágenes
3. usuarios

Funcionalidad

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.

Por ejemplo, en el caso de que el número de capas a utilizar sea 3 y fuera la inserción sería 1,2,3

- Si el tipo fuera inorden se deberán superponer las capas
1,2,3
- Si el tipo fuera preorden se deberán superponer las capas
3,2,1
- Si el tipo fuera postorden se deberán superponer las capas
1,2,3

Por lista de imágenes

Se ingresará el id de la imagen el cual deberá buscarse en la lista circular de imágenes y según la lista de capas que esta tenga se deberá generar la imagen.

Por capa

Se ingresará el id de la capa a graficar y se buscará en el árbol ABB.

Por usuario

Debe permitir buscar un usuario y seleccionar la imagen a graficar de su lista de imágenes.

CRUD

Usuarios

La aplicación debe ser capaz de agregar nuevos usuarios, eliminar usuarios y modificar usuarios.

Imágenes

Agrega una nueva imagen seleccionando el usuario a quien se agregara tomando en cuenta que los id de imagen es único y de existir el id no permitirá agregarla, elimina imágenes primero seleccionando el usuario y luego el id de la imagen, actualizando la lista circular y la lista de imágenes del usuario a quien pertenecía la imagen.

Graficar el estado de la memoria

Ver lista de imágenes

Deberá mostrar la lista circular doble y la lista de capas de cada imagen (ilustración 6).

Ver árbol de capas

Deberá mostrar el árbol de capas (ilustración 4).

Ver capa

Se indicará el número de capa a mostrar (ilustración 2).

Ver imagen y árbol de capas

Se debe seleccionar una imagen, se mostrará la lista de capas y el árbol de capas, mostrando el enlace de la lista de capas hacia el árbol ABB de capas. (ilustración 7).

Ver árbol de usuarios

Mostrará el ABB de usuarios (ilustración 8).

Observaciones

- Lenguaje de programación a utilizar: Java
- Sistema Operativo: Libre
- IDE: Libre.
- La aplicación que no genere gráficas/imágenes no podrá ser calificada, se debe utilizar Graphviz.
- Todas las estructuras deben de ser realizadas por el estudiante, sin el uso de librerías específicas de ningún IDE o Framework.
- Durante la calificación se harán preguntas para validar que el estudiante realizó la práctica, de no responder correctamente anulara la nota obtenida en la o las secciones en la que aplique tal concepto
- La aplicación será compilada y ejecutada al momento de la calificación.
- **REQUISITO MÍNIMO PARA CALIFICACIÓN: Carga masiva de datos.**

Entregables

- Se debe entregar un manual de usuario con capturas de la aplicación explicando como funciona
- Se debe entregar un manual técnico explicando la organización del código de la aplicación (diagramas, etc)
- Se debe entregar el código fuente y ejecutable de la aplicación
- Todos los puntos anteriores se entregan por Github

Fecha y modo de entrega:

- [Tentativa] Lunes 5 de abril antes de las 23:59 horas vía Classroom
- Si no se cumple con la hora de entrega, se tendrá una penalización de 10% cada 15 minutos incumplidos empezando por los 2 minutos de retraso.
- Si no se entrega en Classroom, no se calificará.
- Las copias tendrán nota de 0 puntos y serán reportadas al catedrático y a coordinación de Sistemas.