

---

## DOCUMENTACIÓN

---

### 1. 201800937 – José Abraham Solórzano Herrera

#### Resumen

El programa consiste en analizar un archivo de entrada con el lenguaje marcado, logrando mostrar el contenido de una matriz por medio de una imagen, utilizando listas ortogonales. Cada imagen tendrá la opción de realizar operaciones entre una imagen o dos imágenes; las operaciones entre una imagen, pueden hacer una rotación horizontalmente y verticalmente, generar una transpuesta, limpiar zona, agregar una línea horizontal y vertical, agregar un rectángulo, agregar un triángulo rectángulo; sin embargo, las operaciones sobre dos imágenes, pueden realizar la operación: Unión, une todos los elementos entre las dos imágenes. Intersección, muestra todos los elementos que tengan en común entre las dos imágenes. Diferencia, esta operación genera como resultado una imagen con todos los elementos de la primera imagen que no pertenezcan a la segunda imagen. Diferencia simétrica, esta operación genera como resultado una imagen con todos los elementos de la primera imagen que no pertenezcan la segunda imagen y todos los elementos de la segunda imagen que no pertenezcan a la primera imagen.

#### Palabras Claves

1. Operación
2. TDA
3. Lista
4. Patrón

#### Abstract

The program consists of analyzing an input file with the marked language, managing to display the content of a matrix by means of an image, using orthogonal lists. Each image will have the option of performing operations between one image or two images; operations between an image, can rotate horizontally and vertically, generate a transpose, clean zone, add a horizontal and vertical line, add a rectangle, add a right triangle; however, operations on two images can perform the operation: Union, joins all the elements between the two images. Intersection, shows all the elements that have in common between the two images. Difference, this operation results in an image with all the elements of the first image that do not belong to the second image. Symmetric difference, this operation results in an image with all the elements of the first image that do not belong to the second image and all the elements of the second image that do not belong to the first image.

#### Keywords

1. Operation
2. ADD
3. List
4. Pattern

## Introducción

El problema consiste en desarrollar una solución, usando una lista ortogonal, el problema debe ser presentado por medio de una interfaz gráfica, el software utiliza la librería de tkinter que provee varias soluciones para una interfaz gráfica; sin embargo, cada matriz debe ser mostrada por medio de una imagen, la cual se desarrolló por medio de la librería de Graphviz, proporciona varias opciones para poder realizar la creación de la matriz; además, para poder realizar la lista ortogonal se utilizó estructuras de datos para poder implementarla. Las operaciones para una imagen permiten, realizar una rotación horizontal y vertical, hacer una transpuesta, limpiar una zona de una imagen, agregar línea horizontal y vertical a una imagen, agregar rectángulo y triángulo rectángulo. Las operaciones sobre dos imágenes, pueden realizar una unión, diferencia, intersección y una diferencia simétrica.

## Desarrollo

### Matriz

En matemática, una matriz es un arreglo bidimensional de números. Dado que puede definirse tanto la suma como el producto de matrices, en mayor generalidad se dice que son elementos de un anillo

### Estructura de Datos

Las estructuras de datos nos permiten manipular cualquier tipo de dato, además de almacenar dinámicamente; se pueden representar de diversas formas, como enlazadas o contiguas.

Las estructuras contiguamente asignadas se componen por bloques de memoria única, incluye arrays, matrices, heaps, y hash tables; en cambio una estructura enlazada está compuesta por fragmentos de memoria unidos por punteros; Los contenedores son estructuras que permiten almacenar y recuperar datos en un orden determinado sin importar su contenido.

### a. Estructura Enlazada

Las estructuras enlazadas son dadas por los punteros, son los encargados de apuntar a una dirección de memoria donde se encuentra ubicado un valor.



Figura I. Lista Enlazada

Fuente: Elaboración propia.

### b. Lista Circular

Las lista circular es muy parecida a una estructura simple enlazada, la diferencia que tiene con la lista simple, es que no tiene fin; En las listas circulares, nunca se llega a una posición en la que ya no sea posible desplazarse. Cuando se llegue al último elemento, el desplazamiento volverá a comenzar desde el primer elemento.

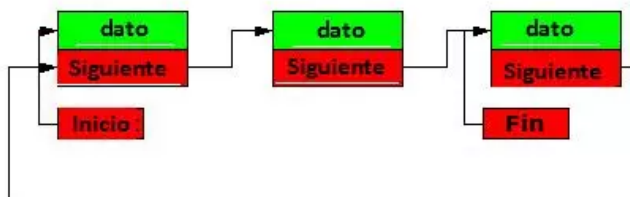


Figura II. Lista Circular

Fuente: Elaboración propia.

### c. Lista Doblemente Enlazada

Una lista doble es aquella en la que sus nodos se encuentran encadenados por dos ligas, es decir, cada nodo apunta al siguiente nodo de la lista, así como al nodo que le antecede en la lista. Una lista doble puede ser implementada como lineal o circular. En una lista lineal, la liga siguiente del último nodo y la liga anterior del primer nodo apuntan hacia un valor nulo, mientras que en una lista circular la liga siguiente del último nodo apunta hacia el primer nodo de la lista y la liga anterior del primer nodo apunta hacia el último nodo de la lista.

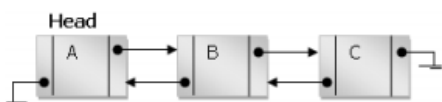


Figura II. Lista Doblemente Enlazada  
Fuente: Elaboración propia.

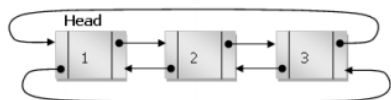


Figura II. Lista Doblemente Circular  
Fuente: Elaboración propia.

### d. Lista Ortogonal

Una lista ortogonal es aquella en la que sus nodos se encuentran encadenados por cuatro ligas, es decir, cada nodo se encuentra doblemente ligado en forma horizontal, y cada nodo se encuentra doblemente ligado en forma vertical. Una lista ortogonal puede ser implementada como lineal o circular. Este tipo de listas se puede utilizar para representar matrices

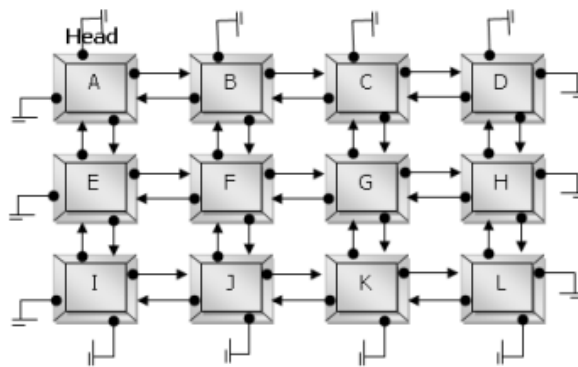


Figura II. Lista Ortogonal  
Fuente: Elaboración propia.

En una lista se pueden efectuar operaciones por medio de algoritmos que se deben desarrollar de acuerdo con el tipo de lista. Algunas de las operaciones básicas que se pueden efectuar sobre una lista son:

**1) Recorrido:** Esta operación consiste en visitar todos los nodos que forman parte de una lista. Para recorrer todos los nodos de la lista es necesario posicionarse en el primer nodo de la lista y después avanzar hacia el nodo que apunta la liga siguiente y así sucesivamente hasta encontrar el fin de la lista.

**2) Inserción.** Esta operación consiste en agregar un nuevo nodo a una lista. La ubicación del nuevo nodo puede ser al inicio, al final o en cualquier posición dentro de la lista.

**3) Borrado.** Esta operación consiste en eliminar un nodo de la lista y redireccionar las ligas de los nodos antecesor y sucesor para el caso de un nodo que se encuentre en una posición intermedia. El borrado también se puede aplicar tanto al primer nodo de la lista como al último nodo de la lista.

**4) Búsqueda.** Esta operación consiste en recorrer todos los nodos de la lista desde el primer nodo para ir comparando el valor de cada nodo con el valor que se está buscando

hasta encontrar el nodo con el valor indicado o encontrar el fin de la lista.

### **Recursividad**

La recursividad es un concepto fundamental en matemáticas y en computación, es una alternativa diferente para implementar estructuras de repetición (ciclos), los módulos se hacen llamadas recursivas. Se puede usar en toda situación en la cual la solución pueda ser expresada como una secuencia de movimientos, pasos o transformaciones gobernadas por un conjunto de reglas no ambiguas. La recursividad consiste en que una función o un método sea llamado a sí mismo.

Caso recursivo: una solución que involucra volver a utilizar la función original, con parámetros que se acercan más al caso base. Los pasos que sigue el caso recursivo son los siguientes:

1. El procedimiento se llama a sí mismo
2. El problema se resuelve, tratando el mismo problema pero de tamaño menor
3. La manera en la cual el tamaño del problema disminuye asegura que el caso base eventualmente se alcanzará.

### **Lenguaje de Programación**

En el programa se utilizó el lenguaje de programación python para poder desarrollar la solución al problema, Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad;

además, Python es gratuito, incluso para propósitos empresariales, en los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.

### **Conclusiones**

El lenguaje de programación python nos permite crear con sencillez y velocidad los programas, además de proporcionar un óptimo código a comparación de otros lenguajes como Java o C, entre otros.

Las listas son estructuras de datos que son dinámicas, esto significa que adquieren espacio y liberan espacio a medida que se necesita. Son muy versátiles, pueden definirse estructuras más complejas a partir de las listas, como por ejemplo arreglos de listas, son eficaces para diseñar colas de prioridad, pilas y colas sin prioridad, y en general cualquier estructura cuyo acceso a sus elementos se realice de manera secuencial.

### **Referencias bibliográficas:**

[https://github.com/bram814/IPC2\\_Proyecto2\\_201800937.git](https://github.com/bram814/IPC2_Proyecto2_201800937.git)

David Budgen(2015).Software Design..

## Anexos

