# Análisis, Desarrollo y Evolución de Algoritmos en Informática 2.

## a. Análisis del problema y consideraciones para la alternativa de solución propuesta.

Para abordar el problema, dedicamos tiempo a una lectura exhaustiva del archivo desafío 1. Nuestro objetivo fue comprender a fondo el problema antes de proceder con la elaboración de cualquier diagrama de flujo o estrategia de solución. Enfocamos nuestra atención en comprender las complejidades del problema y evaluar las posibles alternativas de solución de manera integral.

Una de las principales estrategias que adoptamos fue dividir todo el programa en funciones. Esto nos permitió realizar un trabajo colaborativo más efectivo y organizado. Además, nos enfocamos en comprender cómo utilizar Git y GitHub de manera adecuada antes de comenzar con la implementación de las funciones.

Hasta el momento, hemos creado una función para cada tarea específica. Por ejemplo, una función para crear la matriz, otra para rotar la matriz y otra para determinar la posición que se está buscando en la matriz rotada, basándonos en su fila y columna. Estamos avanzando a un ritmo sólido y hemos logrado comprender muy bien el problema en su totalidad.

## b. Esquema donde describa las tareas que usted definió en el desarrollo de los algoritmos.

Las tares que definimos en cada algoritmo ósea en cada función fue recordar muy bien lo que se vio en clase acerca de arreglos, memoria dinámica, punteros y demás temas que son indispensables para el desarrollo de esta práctica, además complementándolo muy bien con lo que sea ha estudiado por nuestra propia cuenta

1. Definir la función imprimir matriz ();

* Me recibe como parámetro un entero el cual va a ser el tamaño de la matriz
* De pendiendo del valor que se reciba como parámetro se llena la matriz teniendo en cuanta todas las condiciones que se nos han propuesto
* Después de imprimir la matriz se hace la debida liberación de la memoria
* La función no retorna ningún valor

1. Definir funciones rotarmatriz1, rotarmatriz2, rotarmatriz3 ();

* Cada una recibe un entero como parámetro para determinar el tamaño de la matriz a rotar.
* Implementan algoritmos para rotar la matriz en 90, 180 y 270 grados en sentido antihorario respectivamente
* Imprimen la matriz rotada y liberan la memoria utilizada.

1. Definir función buscar\_posicion ();

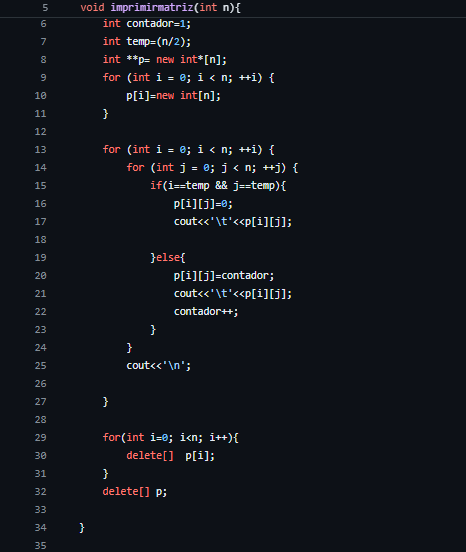
* Esta función me recibe como parámetro un puntero que tiene la dirección de memoria del primer valor de la clave la cual esta en un arreglo dinámico y el otro parámetro es un entero el cual nos determina si la siguiente matriz es mayor igual o de menor tamaño.
* Haciendo las debidas operaciones para que también apunte a la segunda dirección de memoria la cual tiene almacenada el valor de la columna
* Y después se pone en marcha un algoritmo el cual nos determina cual es el valor de la fila y columna de la siguiente matriz
* Por último, nos retorna un puntero al arreglo dinámico (cabe recalcar que se debe liberar la memoria una vez utilizada)

1. Por ultimo se la hace su debida implementación a cada función en el main con sus debidas verificaciones y condiciones

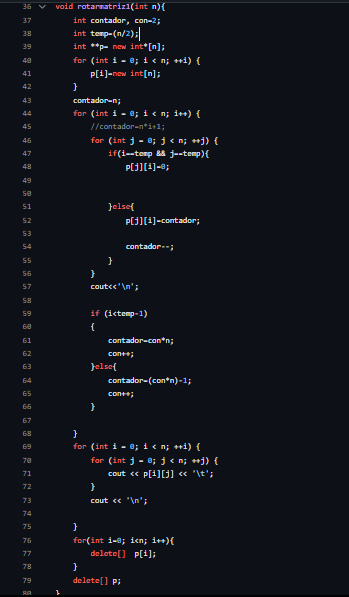
El informe se ira actualizando conforme se vaya avanzando en el main.

c. Algoritmos implementados.

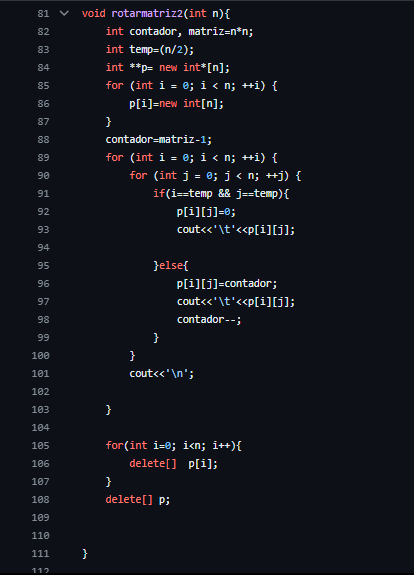
1. La función imprimirmatriz crea una matriz cuadrada de tamaño n e imprime sus elementos en forma de matriz, con un patrón especial que coloca el valor 0 en el centro de la matriz y luego llena los demás elementos de la matriz con números consecutivos empezando desde 1 y aumentando en orden de izquierda a derecha y de arriba hacia abajo.



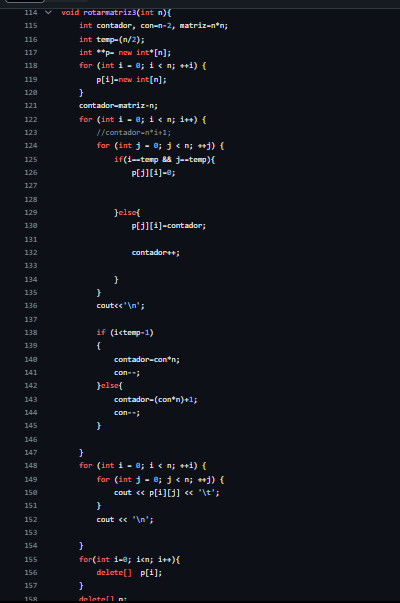
2. Esta función realiza una rotación de 90 grados en sentido antihorario de una matriz cuadrada y luego la imprime en pantalla. La lógica de rotación implica asignar valores especiales a ciertos elementos de la matriz para lograr la rotación adecuada.



3. Esta función realiza una rotación de 180 grados en sentido antihorario de una matriz cuadrada y luego la imprime en pantalla. La lógica de rotación implica asignar valores especiales a ciertos elementos de la matriz para lograr la rotación adecuada.



4. Esta función realiza una rotación de 270 grados en sentido antihorario de una matriz cuadrada y luego la imprime en pantalla. La lógica de rotación implica asignar valores especiales a ciertos elementos de la matriz para lograr la rotación adecuada.



5. La función buscar\_posicion recibe como parámetros un puntero clave que apunta a un arreglo de enteros con la posición de un elemento en una matriz (fila y columna), y un entero siguiente\_matriz que indica si la siguiente matriz es mayor, igual o de menor tamaño. Luego, la función calcula y devuelve un puntero que apunta a un arreglo de enteros con la posición del elemento en la siguiente matriz.

