# INFORME PARCIAL 1

*INFORMÁTICA II*

**Mateo Echavarria Perez**

**Marcos Restrepo Molina**

UNIVERSIDAD DE ANTIOQUIA

2024

# 1. ANÁLISIS DEL PROBLEMA Y CONSIDERACIONES INICIALES:

Para dar una implementación adecuada, es necesario definir algunas consideraciones:

1.1 DEFINICIÓN DE TÉRMINOS:

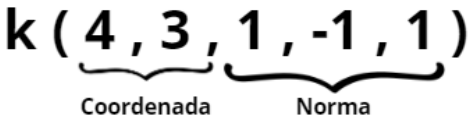



Figura 1

- **Coordenada:** se constituye de dos componentes, el primero de ellos corresponde a la fila y el siguiente a la columna, como se observa en la Figura 1. Se toma como referencia las coordenadas de la primera matriz, y a partir de esa se compara con las siguientes una vez alineadas por el centro.
- **Norma:** cada uno de sus elementos puede ser 1,0,-1. donde cada valor indica mayor, igual o menor respectivamente. Cada uno de los elementos de esta norma nos da la relación entre la matriz actual y la siguiente en la coordenada que se está analizando, así:

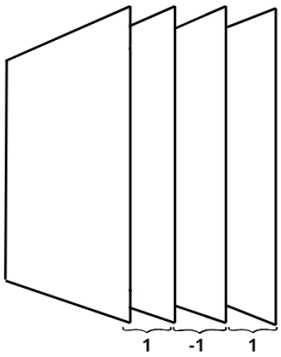



Figura 2

los elementos de la norma no tienen límite alguno, y el número de matrices resultantes en la cerradura X siempre es el número de normas sumado uno, por ejemplo, en la Figura 2, la cual podría representar la regla k(4 ,3 ,1 ,-1 ,1) se han creado 4 matrices para solo 3 normas.

1.2 TAMAÑO MÍNIMO DE LAS MATRICES:

El tamaño de las matrices debe ser compatible con las coordenadas de la regla k, ya que estas deben existir dentro de la matriz sin salir del rango máximo. Es por ello que cualquier matriz que se quiera generar como opción para la cerradura X debe tener un tamaño representado por un número IMPAR y MAYOR al componente más grande de la coordenada (fila o columna) que fue otorgada en la regla k. Por ejemplo, si se ingresa k(6,7,1,0) el tamaño de cada matriz debe ser un número IMPAR y MAYOR a 7.

1.3 CAMBIO DE COORDENADAS EN MATRICES ALINEADAS Y CON DIFERENTE TAMAÑO:

Una vez se alinea una matriz A con otra matriz B tomando como referencia su centro, como se muestra en la Figura 3, se deduce que para matrices de igual tamaño las coordenadas del valor situado detrás son iguales. Sin embargo, para matrices de diferentes tamaños las coordenadas de cualquier valor cambian sumando una unidad en filas y columnas por cada salto en tamaño (números impares de diferencia).

Por ejemplo, si se observa la Figura 4, se tienen dos matrices alineadas A y B de tamaño 3x3 y 5x5 respectivamente, si se toma la coordenada (0,0) en la matriz A situada delante, se observa que la coordenada del valor que se sitúa justo detrás en la matriz B es (1,1), es decir, se sumó una unidad tanto en filas como en columnas, ya que el salto en tamaño fue de sólo un número impar.

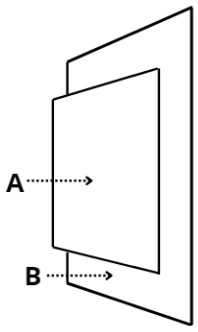



Figura 3

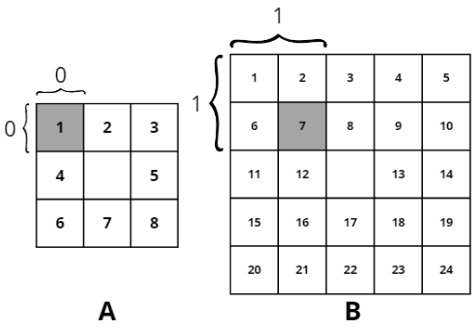



Figura 4

la suma de una unidad se da por cada salto en tamaño, es decir, si se tiene dos matrices A y B de tamaño 3x3 y 7x7 respectivamente, el salto en tamaño es de 2 numeros impares de diferencia, por ello, para cualquier coordenada en A se suma 2 unidades tanto en filas como en columnas para hallar el valor situado detrás en la matriz B. Y así sucesivamente para cualquier par de matrices de diferente tamaño.

## 1.4 RELACIÓN ENTRE TAMAÑO DE MATRIZ Y SUS VALORES:

Si se analiza un punto en dos matrices, una más grande que otra, alineadas por el centro como se muestra en la Figura 3, sin ninguna rotación (ambas en posición neutra), donde A es la matriz (3x3) delantera y B la matriz (5x5) situada detrás, se observa como cualquier valor de A es siempre menor al valor que se sitúa justo detrás en la matriz B, como el ejemplo que se muestra en la Figura 4. Este ejercicio se puede extrapolar a matrices de mayor tamaño siempre y cuando no se roten.

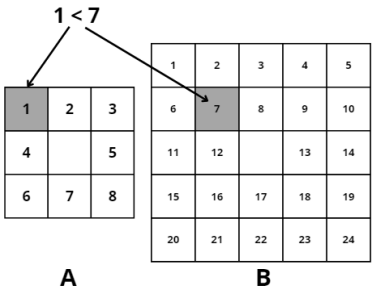
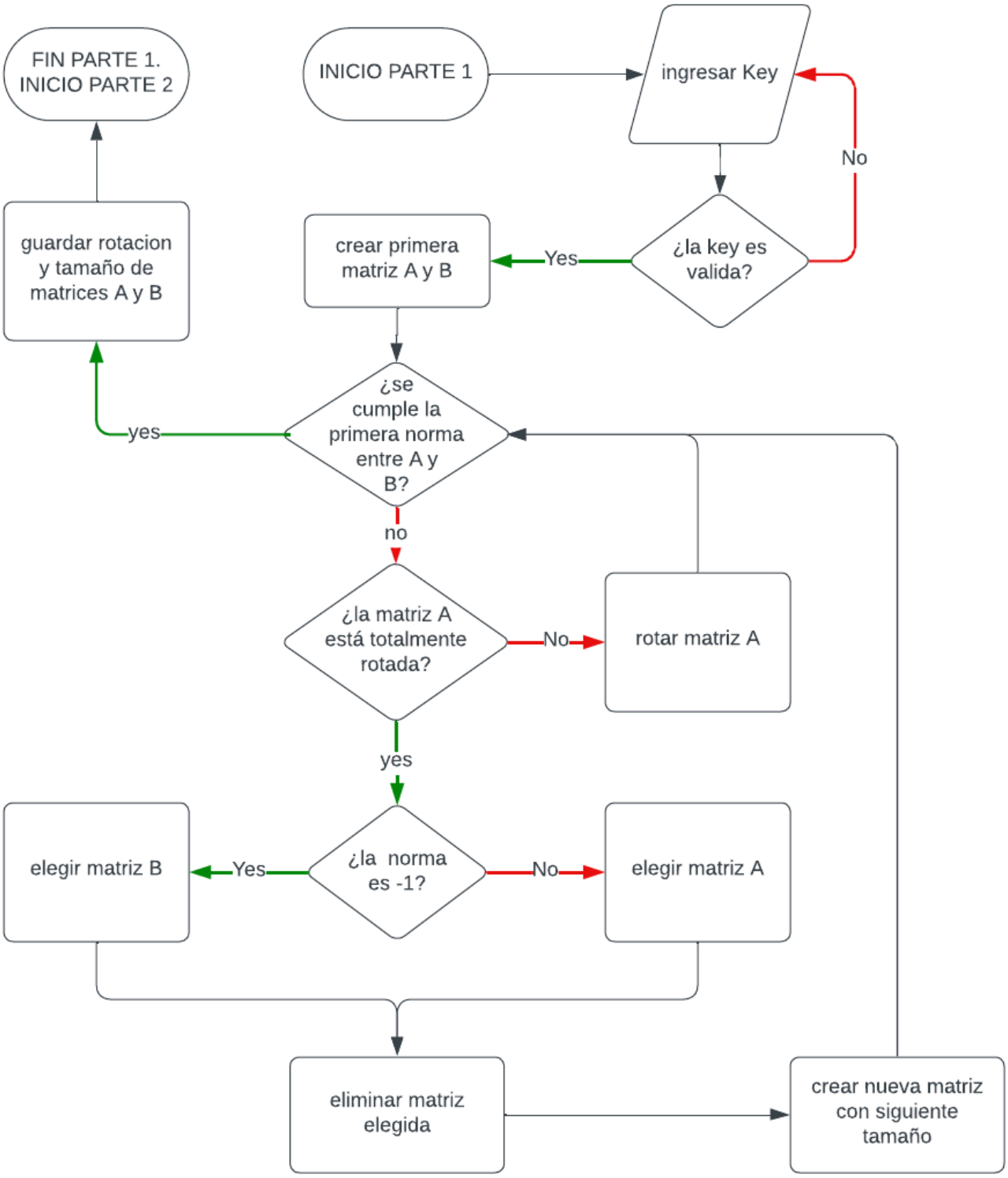



Figura 5

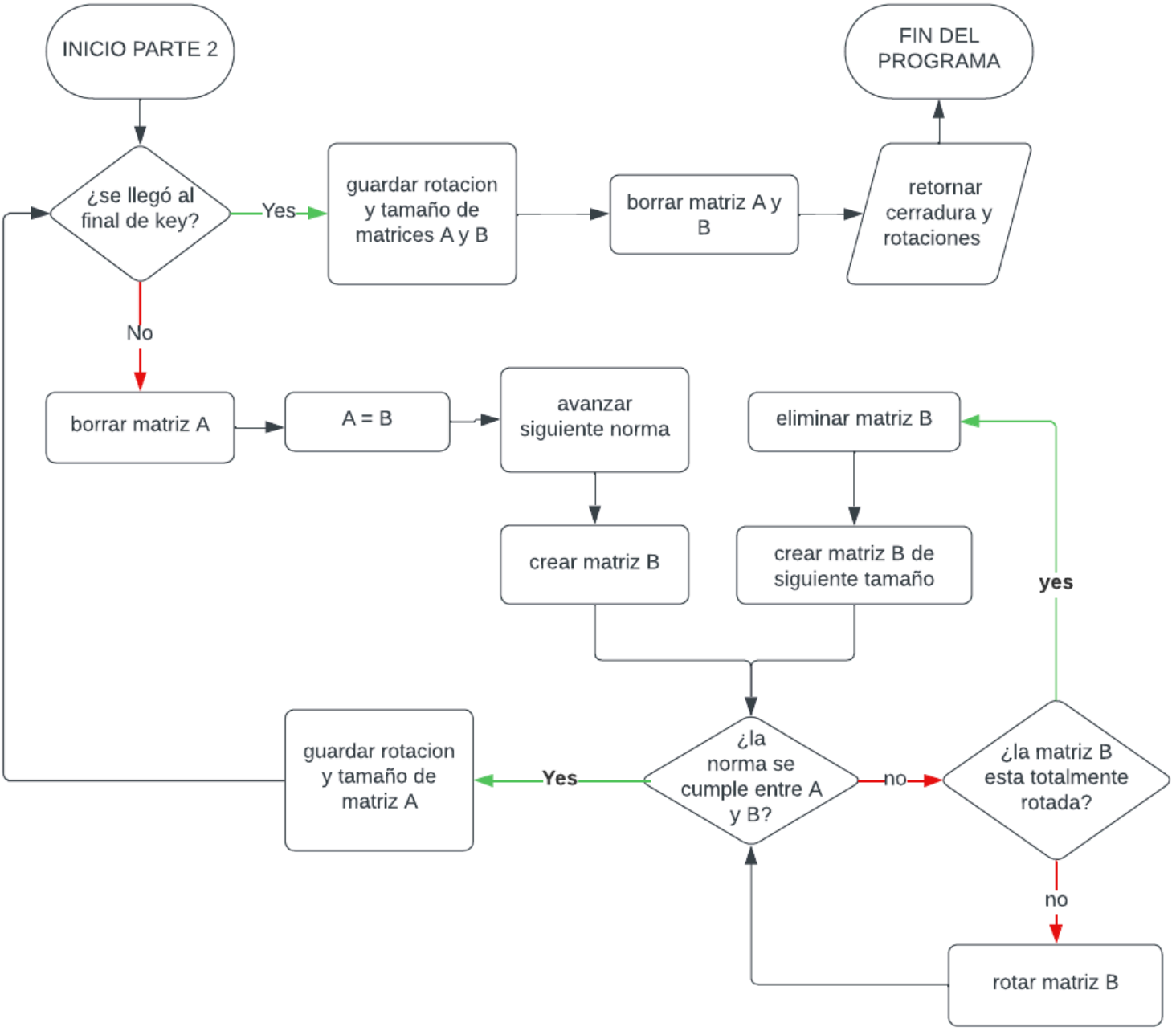
Esto se puede utilizar a nuestro favor, ya que si se tiene dos matrices del mismo tamaño, sin más opciones de rotación, se puede hacer más grande la matriz que más sea conveniente dependiendo de si se necesita un número más grande o pequeño.

# 2. DIAGRAMA DE FLUJO

**Parte 1:**



**Parte 2:**


INICIO PARTE 2
¿se llegó al final de key?
Yes
guardar rotacion y tamaño de matrices A y B
borrar matriz A y B
retornar cerradura y rotaciones
FIN DEL PROGRAMA
No
borrar matriz A
A = B
avanzar siguiente norma
crear matriz B
eliminar matriz B
crear matriz B de siguiente tamaño
yes
¿la norma se cumple entre A y B?
Yes
no
guardar rotacion y tamaño de matriz A
¿la matriz B esta totalmente rotada?
no
rotar matriz B

# 3. EXPLICACIÓN DEL DIAGRAMA DE FLUJO

El diagrama de flujo propuesto se divide en 2 partes, una vez se llega al final de la parte 1 se inicia automáticamente la parte 2. Se sustenta de la siguiente manera:

**Parte 1:**

La clave k (nombrada como “key”) es un arreglo de valores enteros el cual debe ser validado previamente para evitar errores en el código. Una vez hecho esto, se procede a crear las dos primeras matrices teniendo en cuenta el tamaño mínimo visto en el punto 1.2 y se comparan entre sí con la primera norma de la regla k. dicha comparación se da entre valores alineados uno detrás de otro teniendo en cuenta lo descrito en el punto 1.3.

Siempre que dicha norma no se cumpla se procede a rotar la matriz A. En caso de llegar a su última rotación se elige una de las dos matrices para redimensionar, la forma en la que se elige esta matriz depende de si la primera norma es 1 o -1.

Una vez que se comprueba que dicha matriz está totalmente rotada, se procede a aumentar su tamaño, es allí donde toma sentido elegir la matriz A o B, ya que como se vio anteriormente existe una relación entre los valores alineados en dos matrices de diferentes tamaños como se demostró en el punto 1.4 .

En cualquier caso, si la norma es -1 se elige la matriz trasera y se aumenta su tamaño, de lo contrario se hace con la matriz delantera, esto con el fin de que se cumpla la norma.

Este proceso se repite hasta comprobar únicamente la primera norma. Una vez se logra esto se guardan las rotaciones y tamaño de las matrices A y B, y se procede con la siguiente parte.

**Parte 2:**

Se verifica si se llegó al final de k, mientras sea falso se continúa con el programa. Lo primero que se hace es borrar la matriz A (su dimensión y rotaciones ya se guardaron anteriormente) con el fin de optimizar el uso de memoria, y al identificador A se le asigna lo que hay en la matriz B (sin hacer copia). Posteriormente, se crea otra matriz con el tamaño mínimo y se le asigna el identificador B. Por último, se avanza a la siguiente norma y se valida.

Este proceso se repite de forma similar que en la parte 1, sin embargo, aquí solo se rota o redimensiona la matriz B y usamos la A como referencia sin hacerle modificaciones. Por

ello, no hay validaciones de 1 o -1 como en la anterior parte, ya que no es necesario elegir matriz.

Una vez se cumpla la norma se guarda la matriz A y mientras no se llegue al final de key se reinicia el proceso. Cuando se llega al final de key se guardan las rotaciones y tamaños de ambas matrices, se borra su contenido, se retorna X y sus rotaciones, y finaliza el programa.

## 4. ALGORITMOS IMPLEMENTADOS

La implementación se basa en los siguientes algoritmos fundamentales:

- Crear estructura.
- Rotar estructura.
- Verificar valor en punto dado (validar norma de la regla k).
- Borrar matriz (liberar espacio en memoria).
- Guardar tamaños en X (cerradura) y en R (rotaciones).

## 5. PROBLEMAS DE DESARROLLO

- Al dividir el programa en varios módulos se pierde un poco la linealidad, lo que nos podría traer dificultades al momento de corregir errores o incluso entender el código después de mucho tiempo.
- Al momento de comenzar la implementación, nos percatamos de algunas consideraciones que pasamos por alto, sin embargo, las logramos implementar sin mayores inconvenientes.

## 6. EVOLUCIÓN DE LA SOLUCIÓN

- Creamos repositorio en Github.
- Se crean los módulos con algoritmos más básicos: rotar matriz, crear matriz, comparar valores. etc.

## 7. CONSIDERACIONES PARA TENER EN CUENTA EN LA IMPLEMENTACIÓN

- En la función para rotar matriz se debe recibir como parámetros el tamaño de la matriz y su puntero correspondiente. Dentro de esta función, se debe crear un nuevo puntero doble donde se copiarán los elementos de la primera matriz de forma que se pueda rotar. al final de la función se debe borrar la matriz original y retornar la nueva.
- En la función para crear matriz se debe recibir como parámetro el tamaño de la matriz, (se debe crear con el uso de memoria dinámica) y por último se retorna un doble puntero que apunte a la nueva matriz.
- Se debe crear una función para comparar valores entre matrices, en la cual se reciba como parámetros dos estructuras, sus respectivos tamaños, las coordenadas, y la norma. se debe verificar el tamaño de las matrices para tener en cuenta como cambian sus coordenadas (punto 1.3), y su rango(punto 1.2). Por último, se debe retornar un valor booleano si la norma se cumple.
- crear dos vectores X y R. En X se guardarán los tamaños de cada matriz, y en el vector R las rotaciones para cada una.
- Se debe validar la key ingresada por el usuario. Teniendo en cuenta los siguientes aspectos: sintaxis, coordenadas positivas, dos coordenadas, al menos una norma y regla de mayor, menor o igual solo con números permitidos (1,0,-1).
- El centro de cada matriz se rellena con un 0. Se debe validar que la coordenada analizada de cada matriz no sea su centro.
- Al saber que el usuario puede ingresar un número reglas, se deben almacenar adecuadamente para no caer en un caso de pérdida de memoria.