**Informe parcial 1**

**Informática II**

*Cristian Murillo*

*Karen Valentina Gonzalez Rodriguez*

**UNIVERSIDAD DE ANTIOQUIA**

**2024**

**Informe parcial 1 Informática II**

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

**¿Qué es lo que nos pide el problema?** **y ¿Qué límites tenemos?**

El problema planteado implica la creación de un sistema de cerraduras X que pueden tener diferentes configuraciones de estructuras M, rotaciones y reglas de apertura. Para abordar este problema, se decidió implementar una solución modular en C + +, que permita la creación dinámica de estructuras de datos, rotaciones de matrices, configuración de cerraduras y validación de reglas de apertura.

* Nos dan una clave de apertura y en base a ello debemos generar una configuración que cumpla con los requisitos de la clave k
* A la hora de generar x los números, tienen que ser impares y las matrices cuadradas
* Para la hora de crear matrices (rotación) para la configuración x y la regla k se necesita memoria dinámica
* Necesitamos validar las entradas del problema
* Además el programa debe ser eficiente y tener buen uso de la memoria dinámica
* No podemos usar librerías ni cosas que no sean por defecto

**Definiciones importantes para la resolución del problema:**

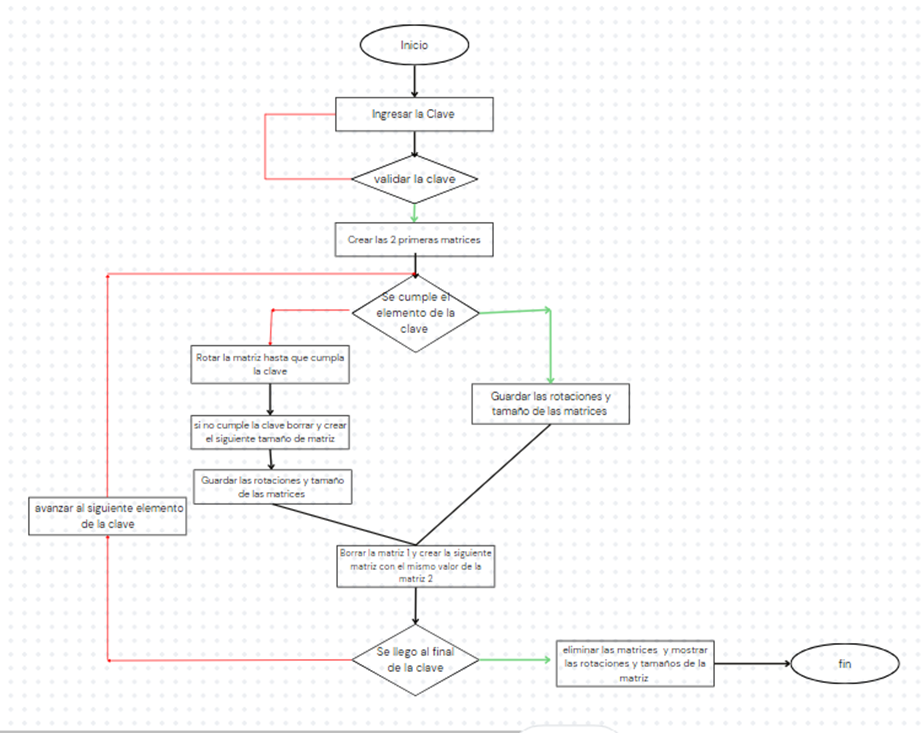
**1. Coordenada:** Se compone de dos componentes, al primero se le denomina la fila, y el segundo es la columna de la matriz que vamos a utilizar,

**2. Norma:** Es la relación entre la matriz actual y la la coordenada que se está analizando, los elementos de la norma no tiene límite. El número de matrices creadas en la cerradura X siempre será el número de normas sumándole 1 en secuencia.

**3. Tamaño Mínimo de la matriz:** Para garantizar que las coordenadas de la regla k estén dentro del rango máximo y se ajusten adecuadamente dentro de las matrices, es esencial que el tamaño de las matrices sea compatible con estas coordenadas. Por lo tanto, al generar una matriz como opción para la cerradura X, se requiere que su tamaño sea un número impar y mayor que el valor más grande entre la coordenada de fila y columna proporcionada en la regla k. Por ejemplo, si consideramos la regla k(6,7,1,0), el tamaño de cada matriz debe ser mayor que 7 y también debe ser un número impar.

**4. Cambio de coordenadas de matrices alineadas y de diferente tamaño:** El cambio de coordenadas en matrices alineadas y con diferentes tamaños se produce después de alinear una matriz A con otra matriz B utilizando su centro como referencia, tal como se muestra en la figura 3. Si las matrices tienen el mismo tamaño, las coordenadas del valor situado detrás son las mismas. Sin embargo, cuando las matrices tienen tamaños diferentes, las coordenadas de cualquier valor cambian añadiendo una unidad tanto en filas como en columnas por cada diferencia en tamaño, que corresponde a números impares.

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



La clave ingresada por el usuario es un arreglo numérico de valores enteros dado por el usuario en el cual todos los elementos a excepción de los 2 primeros serán 1 0 -1 a partir de hay se crean 2 matrices y se comparan entre sí teniendo como referencia el valor de la clave.

Si no se cumple el valor se rotará la matriz 1 y en caso de rotarla completamente y no cumplir el valor de la clave se borrará la matriz 2 y se creará una matriz con el tamaño directamente siguiente a la matriz según el número de la clave si es 1 se irá a la matriz directamente mayor y si es – 1 se irá a la matriz directamente menor.

Una vez se compruebe el valor de la clave se borrara la matriz 1 y se creará una nueva matriz con el mismo valor de la matriz 2 después se verificará si se llegó al final del arreglo y en caso de no ser así se repetirá el proceso de rotación y comprobación hasta el final del arreglo una vez se finalice el arreglo se eliminaran las matrices y se mostrarán las rotaciones y el tamaño de cada ciclo

**c. Algoritmos implementados.**

La implementación se apoya en los siguientes algoritmos clave:

* Generar una estructura.
* Girar la estructura.
* Comprobar el valor en una ubicación dada (validar la regla k).
* Eliminar la matriz (liberar memoria).
* Almacenar tamaños en X (cerradura) y en R (rotaciones).

**d. Problemas de desarrollo que afrontó.**

Al trabajar en nuestro proyecto llegamos a tener los siguientes problemas:

* Se pueden llegar a presentar errores en la validación
* Problemas con el uso de la memoria (no es tan fluido y eficiente como nos gustaría)
* Cuando dividimos el programa para el trabajo personal, llegamos a tener algunos errores de comunicación, pero luego de algunas reuniones virtuales logramos ponernos en sintonía.

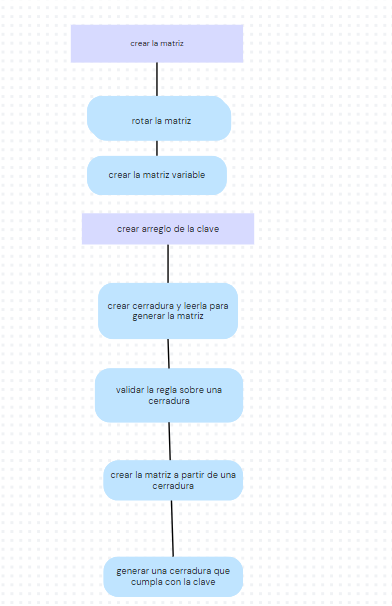
**e. Evolución de la solución y consideraciones para tener en cuenta en la implementación.**

1. **Evolución de nuestra solución:**

* Nos hemos reunido a trabajar a distancia, creamos un repositorio en Git Hub

1. **Condecoraciones para la implementación:**

* La función encargada de rotar una matriz debe recibir como argumentos el tamaño de la matriz y un puntero que apunte a ella. Dentro de esta función, se debe crear un nuevo puntero doble donde se copiarán los elementos de la matriz original para permitir su rotación. Al finalizar la función, se debe liberar la memoria ocupada por la matriz original y devolver el nuevo puntero.
* En la función de creación de matriz, se debe recibir el tamaño deseado como parámetro (la matriz se crea utilizando memoria dinámica) y finalmente se devuelve un puntero doble que apunta a la nueva matriz.
* Se requerirá también una función para comparar valores entre matrices. Esta función deberá recibir como parámetros dos estructuras que contengan las matrices a comparar, sus respectivos tamaños, las coordenadas a comparar y la norma a aplicar. Se debe verificar que los tamaños de las matrices sean compatibles con el cambio en las coordenadas según el punto 1.3, así como su rango según el punto 1.2. Finalmente, la función debe devolver un valor booleano que indique si se cumple la norma especificada.
* Para gestionar los tamaños y rotaciones de las matrices, se deben crear dos vectores: uno llamado X para almacenar los tamaños de cada matriz, y otro llamado R para guardar las rotaciones correspondientes a cada una.
* La validación de la clave ingresada por el usuario debe considerar aspectos como la sintaxis correcta, coordenadas positivas, el uso de dos coordenadas, al menos una norma y la aplicación de una regla de mayor, menor o igual, solo con números permitidos (1, 0, -1).
* Dado que el usuario puede ingresar múltiples reglas, es importante almacenarlas de manera adecuada para evitar pérdidas de memoria.

****