Universidad de Antioquia



Informe desarrollo del desafío 1

Escalante Fonseca Juan David

Murillo Hernández Marcos Jardel

Informática 2

Aníbal José Guerra Soler

Augusto Enrique Salazar Jiménez

Medellín 29/03/2024

**Índice**

1. Análisis del problema y consideraciones para la alternativa.
2. Esquema de tareas para el desarrollo de los algoritmos.
3. Algoritmos implementados.
4. Problemas de desarrollo que se afrontaron.
5. Evolución de la solución y consideraciones para tener en cuenta en la implementación.
6. **Análisis del problema y consideraciones para la alternativa.**

Para el análisis del problema, se hizo un estudio detallado del problema, la necesidad y los requerimientos para cumplir a detalle con el desafío.  
Para la empresa informa2 se requiere diseñar un sistema de cerraduras denominadas X, que consisten en la alineación de una estructura de datos M, la cual puede girar y cambiar de tamaño, a estos cambios de posición los llamaremos “Rotaciones”, las cuales pueden ser 90°,180°,270°. Respectivamente en sentido antihorario.

Para encontrar la combinación de cerraduras correcta, se parte de una regla K, la cual valida ciertas condiciones entre las estructuras alineadas. Para abrir la cerradura es necesario rotar (en caso tal de ser necesario) cada una de las estructuras para que las celdas se alineen de manera que la validación de K sea verdadera.

A continuación, se dará una descripción mas detallada de cada uno de los componentes involucrados:

* **Estructura de datos M:** Es un arreglo de datos bidimensional, el cual varía su tamaño, de forma impar (3x3, 5x5, 7x7, …, NxN), dicha estructura puede rotar, como se menciona previamente, teniendo varios estados.
* Estrucura Fija.
* Estado 1: Rotación 90 grados.
* Estado 2: Rotación 180 grados.
* Estado 3: Rotación 270 grados.
* **Cerradura X:** Consiste en la alineación de varias estructuras M, una tras otra usando la celda del centro como referencia, puesto que la celda del centro estará vacía (marcada con cero), cabe resaltar que no hay restricción en la cantidad de estructuras ni en sus tamaños, los cuales pueden varias como se mencionó previamente.
* **Regla de validación K:** Para abrir la cerradura X, se debe validar la regla K. esta regla considera el valor de una celda especifica, su posición y su ubicación dentro de las diferentes estructuras alineadas. La regla K especifica comparaciones entre los valores de ciertas celdas de las estructuras alineadas y tiene la siguiente forma:

Donde **K** tiene varios parámetros, los cuales son los siguientes:

* **X:** Posición de la fila de la primera matriz.
* **Y:** Posición de la columna de la primera Matriz
* **V:** Validación, la cual tiene tres posibles casos.
  + **1:** Que el valor de la matriz siguiente sea mayor.
  + **-1:** Que el valor de la matriz siguiente sea menor.
  + **0:** Que el valor de la matriz sea igual.

Para el desafío, los valores de K, serían nuestros parámetros de entrada y X los valores de salida.

Para abordar este problema, lo primero es diseñar los diferentes módulos requeridos con el fin de cumplir con cada uno de los requisitos.

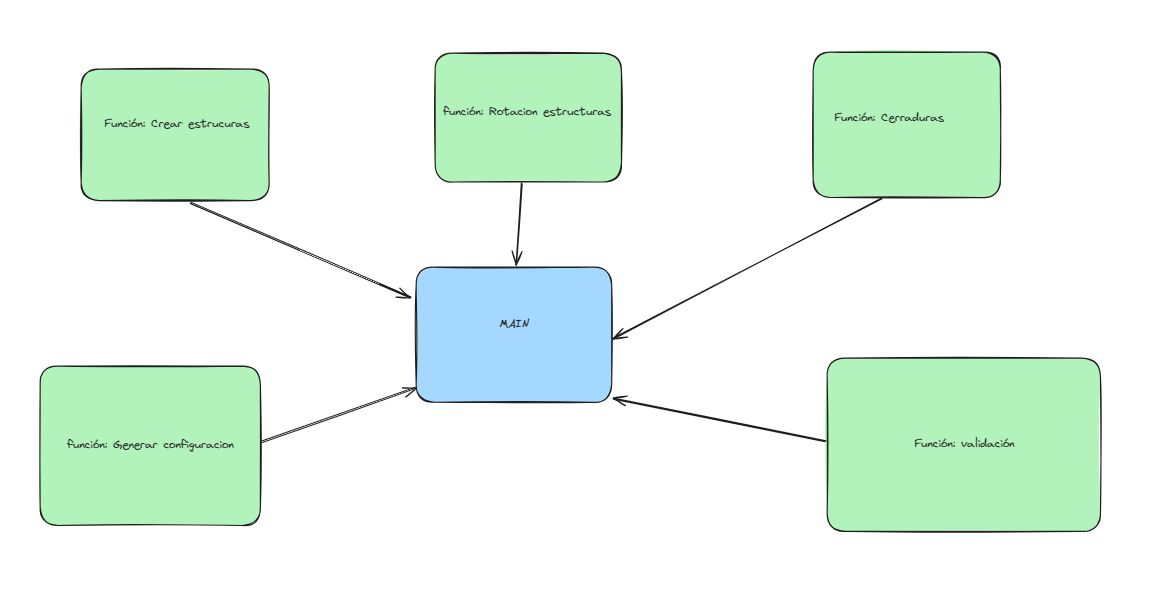
1. **Esquema de tareas para el desarrollo de los algoritmos.**

Lo primero es desarrollar los numerales del 1 al 4, para partir de estos para desarrollar el 5 el cual es una combinación de los cuatro anteriores, haciendo unos ajustes adicionales.

Para ello, a continuación, daremos una descripción de cada uno de los puntos.

* **Modulo para crear estructuras de datos de tamaño variable:** Se creará una función la cual nos permita generar estructuras de datos de tamaño variable, siguiendo las características descritas en las consideraciones iniciales. Esta tendrá la capacidad de cambiar de tamaño y realizar operaciones básicas en la estructura de datos. Es indispensable el uso de memoria dinámica puesto que el tamaño de la estructura es variable.
* **Funciones para realizar las rotaciones de las estructuras:** Se deberán crear las funciones necesarias que permitan la rotación de las estructuras de datos según se muestra en el archivo del desafío y como se describen previamente y se tendrán 4 estados.
* **Módulo para configurar cerraduras variables:** Desarrollar un módulo que permita configurar cerraduras de manera práctica, con la posibilidad de alinear diferentes estructuras de datos de tamaño variable según lo mencionado previamente.
* **Funciones para validar una regla de apertura sobre una cerradura:** Implementar funciones que validen si una regla de apertura dada cumple con las condiciones y configuraciones que especifica la cerradura. Esto implicaría comparar valores de las celdas según las especificaciones de K.
* **Modulo para generar configuraciones de cerradura a partir de una regla:** Desarrollar un módulo que tome una regla de apertura como entrada y genera al menos una combinación para que la cerradura cumpla con dicha regla. Aquí debemos validar diferentes combinaciones de tamaños y rotaciones de estructuras de datos hasta encontrar una que satisfaga la regla.

**Este ultimo punto involucra los 4 puntos anteriores, por lo que quedaría un programa con la siguiente estructura:**



1. **Algoritmos implementados.**

En la función principal se presenta un menú que permite al usuario seleccionar entre varias opciones de operaciones.

**Operaciones del menú:**

* Creación de estructuras de tamaño variable (generarMatriz e imprimirMatriz).
* Rotación de estructuras de datos (rotarMatrizAntiohorario90, rotarMatrizAntiohorario180, rotarMatrizAntiohorario270).
* Configuración de cerraduras de datos.
* Validación de regla K.
* Generación de configuración para generar regla X.

**Funciones para operaciones sobre matrices:**

* **generarMatriz:** Esta función crea una matriz cuadrada de tamaño n, donde el centro de la matriz contiene un 0, y los demás elementos son valores incrementales.
* **imprimirMatriz:** Se imprime la matriz en consola
* **liberarMemoria:** Se libera la memoria reservada previamente.
* **rotarMatrizAntihorario(90, 180, 270):** Rotan la matriz en 90, 180 y 270 grados en sentido antihorario respectivamente.
* **validarReglaK:** Se valida que los datos ingresados sean los correspondientes a las condiciones dadas en el pacial
* **generarX:** Genera un patrón de valores según una serie de reglas y condiciones.
* **actualizarMatrices:** Se actualiza el tamaño de las matrices para así poder encontrar los valores de X.

1. **Problemas de desarrollo que se afrontaron.**

* **Control de versiones en git:** Se tuvieron dificultades en la colaboración con git, desincronización entre las ramas, lo que nos resultó en conflictos que tuvimos que darles solución revisando el código en conjunto.
* **Problemas con fugas de memoria:** A la hora de hacer debugs en ciertas porciones de código identificamos valores inconsistentes en ciertas posiciones de las matrices, se identificó que no se estaba liberando la memoria de forma correcta.
* **Rotado de matriz 270 antiohorario:** Tuvimos un inconveniente a la hora de generar la matriz que rota 270 grados, esto resultó en posición incorrecta de las filas y columnas, para la solución se depuró el código y se hizo una corrección en la lógica con la que se genera dicha matriz.
* **Problemas con tipos de datos no correspondientes:** Se detectó a la hora de ingresar un tipo de dato no correspondiente se genera una inconsistencia en la ejecución.
* **Problemas de validaciones.**
* **Problemas con actualización de matrices:** Se tuvieron dificultades a la hora de aumentar el tamaño de las matrices, lo que nos generaba resultados incorrectos para la cerradura, para ello se hizo una depuración del código para analizar donde no se estaba incrementando el tamaño.

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

Para afrontar el desafío empezamos haciendo una lluvia de ideas y consideramos empezar realizando las siguientes funciones para ir avanzando y así afrontar los problemas que se presentaran para abordarlos uno a uno para luego implementarlas y lograr el objetivo que es generar la cerradura x:

* **Generar matriz de tamaño n impar:** para esta función las consideraciones que se tuvieron en cuenta fue que solo podía generar un matriz de tamaño impar y esta a su vez en el centro debía llevar un cero.
* **Generar matriz de tamaño n impar y rotarla 90,180,270:** para esta función las consideraciones que se tuvieron en cuenta fue que al generar la matriz de cierto tamaño n impar esta se rotara en sentido antihorario.
* **Configurar cerraduras x:** se tuvo en cuenta que fuera de tamaño impar y que solo se almacenará el tamaño de esta matriz.
* **Validar regla k:** para esta se tuvo en cuenta la cantidad de datos que ingresaba el usuario, que los dos primeros datos que ingreso el usuario fueran positivos y que los siguientes datos solo pudieran ser 1, -1 o 0.

Con estas funciones se garantiza la funcionalidad que pedía el problema para implementarlas para generar x.

* **Generar clave x:** en esta parte nos aseguramos de que los datos que ingresaba el usuario fuera el que estaba seleccionando de la matriz para hacer las comparaciones y a su vez ir validando si estas cumplían con los datos ingresados por el usuario y si no hacia esto ir aumentando la matriz al siguiente valor impar y volver a coger el dato que se encontraba en la posición pero como es una matriz más grande se necesitaba sumarle 1 a la fila y columna para que así estuvieran centradas las matrices independientemente del tamaño.

También otra de las consideraciones para este punto fue que se debía ir liberando la memoria ya que se necesitaba para poder continuar haciendo las comparaciones necesarias y a su vez que no haya fugas de memoria.