**DESAFIO INFORMÁTICA II**

**JUAN DIEGO TORO FRANCO**

**LUIS ANGEL OSORIO ALANDETE**

**INGENIERÍA EN TELECOMUNICACIONES Y**

**ELECTRONICA**

**UNIVERSIDAD DE ANTIOQUIA**

**FACULTAD DE INGENIERÍA**

**PRIMER SEMESTRE 2024**

**2024**

**DESAFIO INFORMÁTICA II**

**JUAN DIEGO TORO FRANCO**

**LUIS ANGEL OSORIO ALANDETE**

**INGENIERÍA EN TELECOMUNICACIONES Y**

**ELECTRONICA**

**INFORME DE SOLUCIÓN AL DESAFIO**

**PLANTEADO**

**ANIBAL GUERRA**

**AUGUSTO SALAZAR MARICON**

**DOCENTES**

**UNIVERSIDAD DE ANTIOQUIA**

**FACULTAD DE INGENIERÍA**

**PRIMER SEMESTRE 2024**

**2024**

**TABLA DE CONTENIDO**:

1. CONTEXTO DEL PROBLEMA…………………………………………………………………………………………… 1
2. OBJETIVOS DEL INFORME……………………………………………………………………………………………….2
3. METODOLOGÍA USADA…………..………………………………………………………………………………………2
   1. ALGORITMOS Y TÉCNICAS EMPLEADAS………..………………………………………………………..…3
   2. HERRAMIENTAS USADAS ………………………………………………………………………………………….4
4. ESQUEMA DE LA SOLUCIÓN ………………………………………………………………………………………….. 5
5. CÓDIGOS Y EJEMPLOS DE EJECUCIÓN …………………………………………………………………………….6
6. CONCLUSIONES Y POSIBLES ERRORES …………………………………………………………………………….9
7. REFERENCIAS Y BIBLIOGRAFÍA ……………………………………………………………………………………….10
8. **CONTEXTO DEL PROBLEMA**

Una empresa tiene la necesidad de que a partir de una estructura de datos dada referida como x se haga un análisis a partir de una estructura más simple referida como k la cual tiene como objetivo la comparación de datos específicos en la estructura de datos modificando esta para cumplir ciertas condiciones establecidas. Esta estructura de datos es concretamente un conjunto de matrices las cuales poseen restricciones como las siguientes:

* Dimensión impar 3x3, 5x5, etc.
* El dato del centro siempre será 0

La estructura de datos k será la que determina a partir de cierta interacción con la estructura x cuando esta última deberá modificar su orden o dimensión, esto con el objetivo de que se cumpla una especificación determinada por la estructura k que posee las siguientes restricciones:

* Dimensión 1xN estricto
* Los dos primeros datos hacen referencia a una fila y una columna específicamente
* Los 3 datos siguientes solo pueden ser -1, 1 y 0
* -1 se refiere a número menor, 1 número mayor y 0 se refiere a mismo número

El problema consiste en hacer comprobaciones que hagan que la estructura x cumpla con las especificaciones dadas por la estructura k.

El principal objetivo es el análisis continuo de los datos pertenecientes a la estructura en sus posibles modificaciones, ver el comportamiento de los datos de la estructura x al usar la estructura k como detonante para la modificación de una estructura dada.

**La solución del problema debe ser implementada en lenguaje c++ por medio de un algoritmo de consola.**

1. **OBJETIVOS DEL INFORME**

Hacer seguimiento de la solución que se implementará al problema utilizando estrategias para el correcto funcionamiento de el algoritmo realizado. Principalmente se toma como una plantilla para el tipo de solución que se implementará planeando, de forma específica, el cómo se llevará a cabo la solución.

En este informe se llevará registro de:

* Planeación de solución.
* Metodología usada.
* Pruebas de escritorio.
* Explicación del algoritmo y ejemplos de funcionamiento
* Posibles fallos y mejoras a realizar
* Conclusiones del desarrollo.

1. **METODOLOGÍA USADA**

La mejor forma de abordar este tipo de problema en donde hay tantas consideraciones, complejidad y posibles fallos, es aplicar la famosa idea de “divide y vencerás” es decir, desarrollar el algoritmo de forma modular de tal forma que este problema a resolver sea más sencillo a abordar.

* 1. **Algoritmos y técnicas empleadas**

En los algoritmos que desarrollamos usamos técnicas como memoria dinámica, punteros y funciones de tal forma que:

La memoria dinámica se utilizó para el cambio de tamaños en arreglos en tiempo de ejecución y esto nos permitió flexibilidad y evitar posibles fallos de desbordamiento por haber declarado las matrices de manera estática.

Los punteros nos ayudaron en el pasar matrices como parámetros en funciones y retornarlas ya que de forma nativa en lenguaje c++ las funciones no pueden devolver datos de tipo array.

Las funciones se utilizaron principalmente para dividir el código y sea más legible y sencillo de realizar.

* 1. **Herramientas usadas**

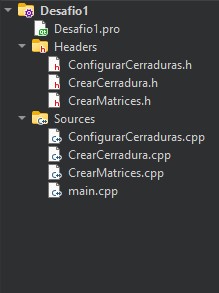
Usamos el intérprete QT como entorno de desarrollo, Git como software de control de versiones y de forma paralela utilizamos GitHub como repositorio de la solución desarrollada, estas dos últimas con el propósito de facilitar el trabajo colaborativo y la actualización constante de versiones para el algoritmo.

Para la documentación y un amplio entendimiento de como implementar una lógica más completa usamos la plataforma StackOverflow tomando como referencia códigos subidos por otros usuarios de funcionalidades especificas necesitadas para la complementación de las soluciones planteadas al problema.

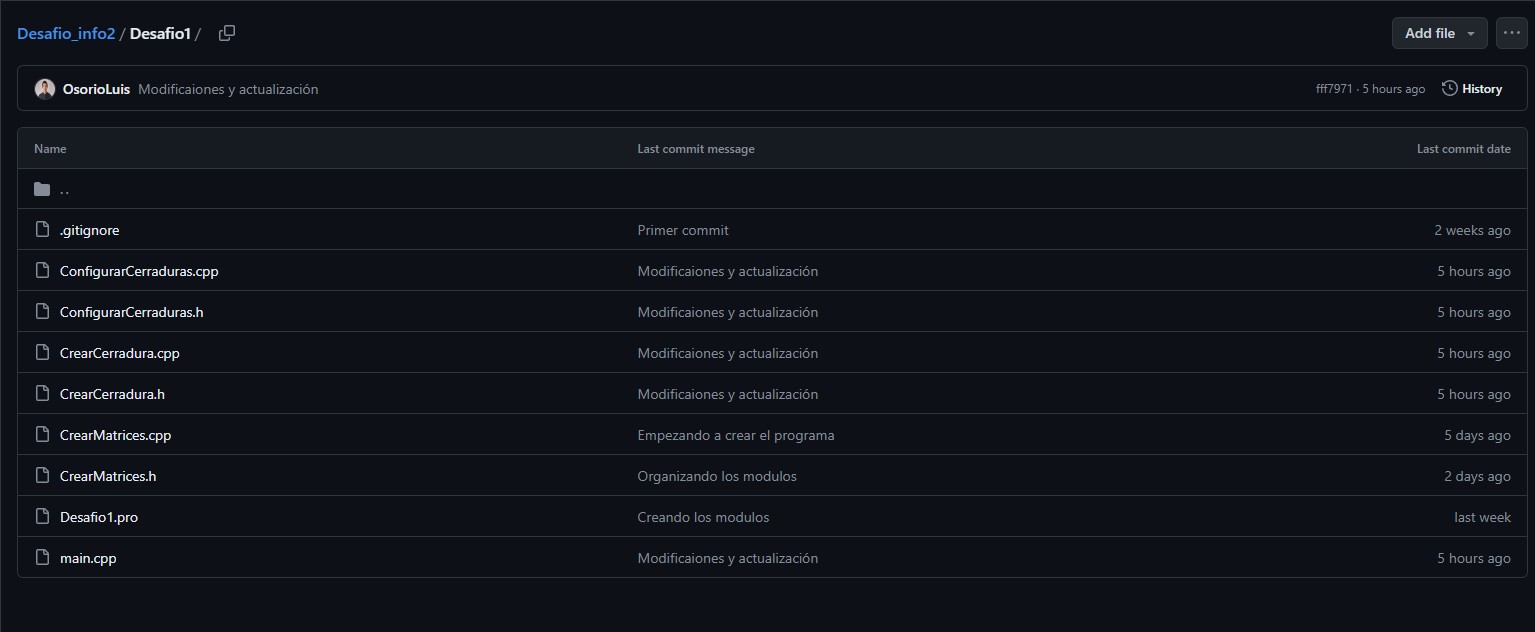
1. **ESQUEMA DE SOLUCIÓN**

Luego de hacer un análisis del flujo de ejecución se determinó que al usar las comprobaciones iniciales se harían por medio de funciones las cuales bucles que controlan la ejecución y construcción de los datos necesarios también serán encargados del llamado de funciones.

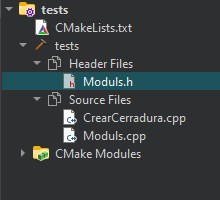
Para la separación del proyecto se realizó de la siguiente manera:



* Se definen las cabeceras las cuales contienen las plantillas de las funciones a usar
* Se definen en la carpeta source los archivos que contienen las funciones
* En el archivo main se tiene toda la ejecución principal del proyecto
* No hubo uso de librerías externas

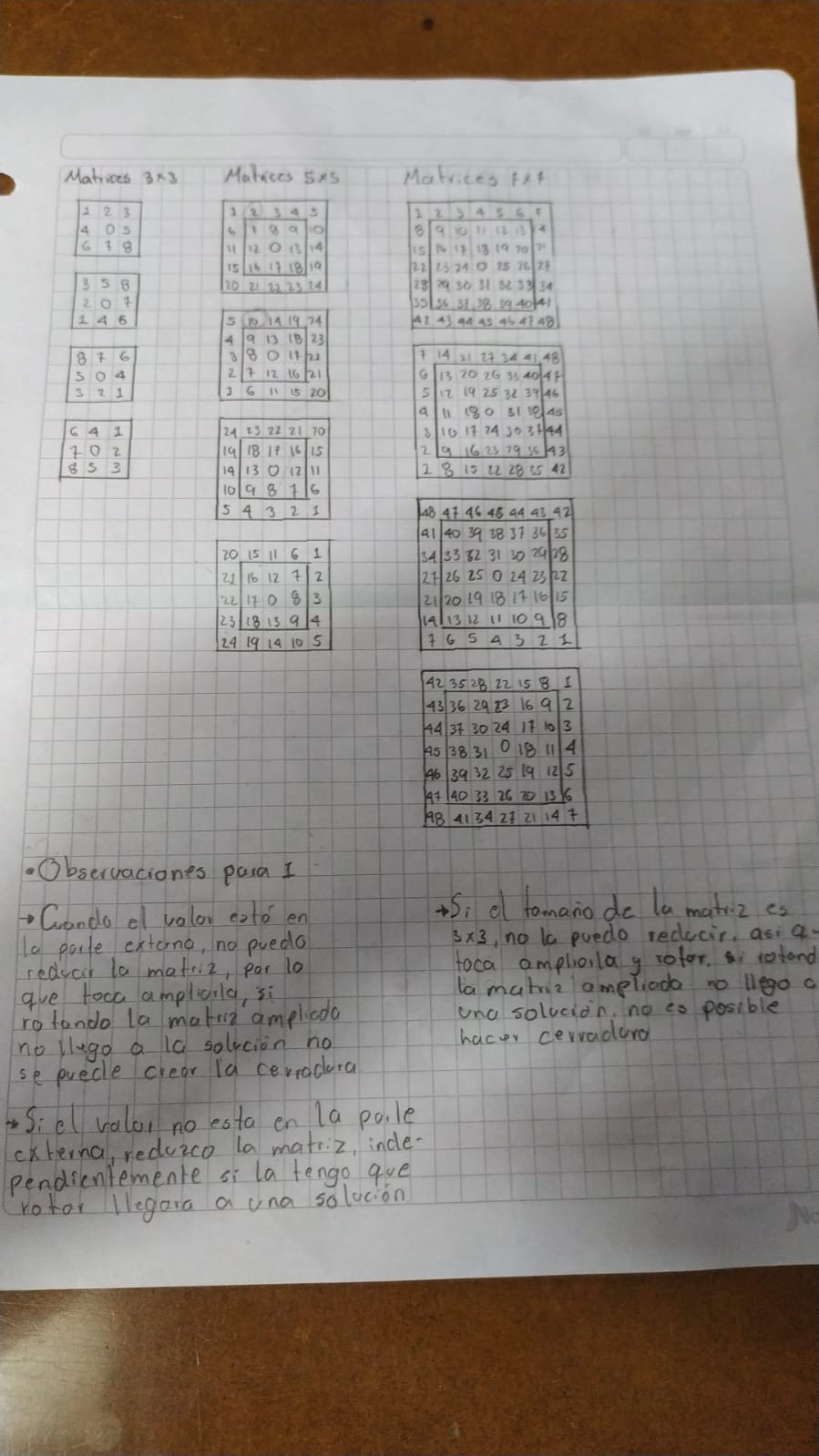
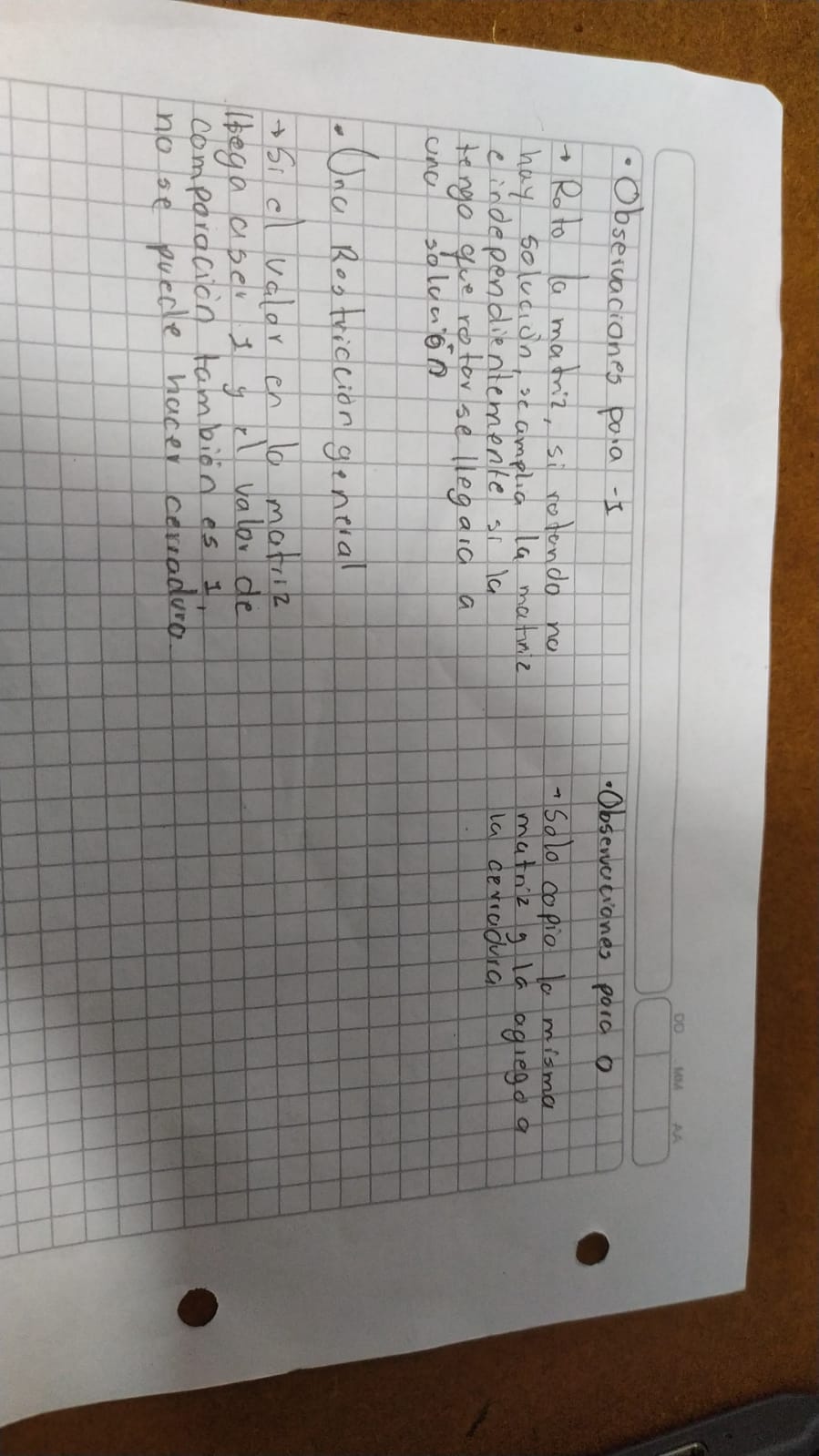


Se usó la herramienta GitHub para almacenar el proyecto y trabajar de manera cómoda, se crearon dos ramas adicionales las cuales contienen los códigos de cada desarrollador para luego actualizar en la plataforma GitHub al hacer cambios significativos, esto también nos permitió el seguimiento de la aplicación y su correcta actualización

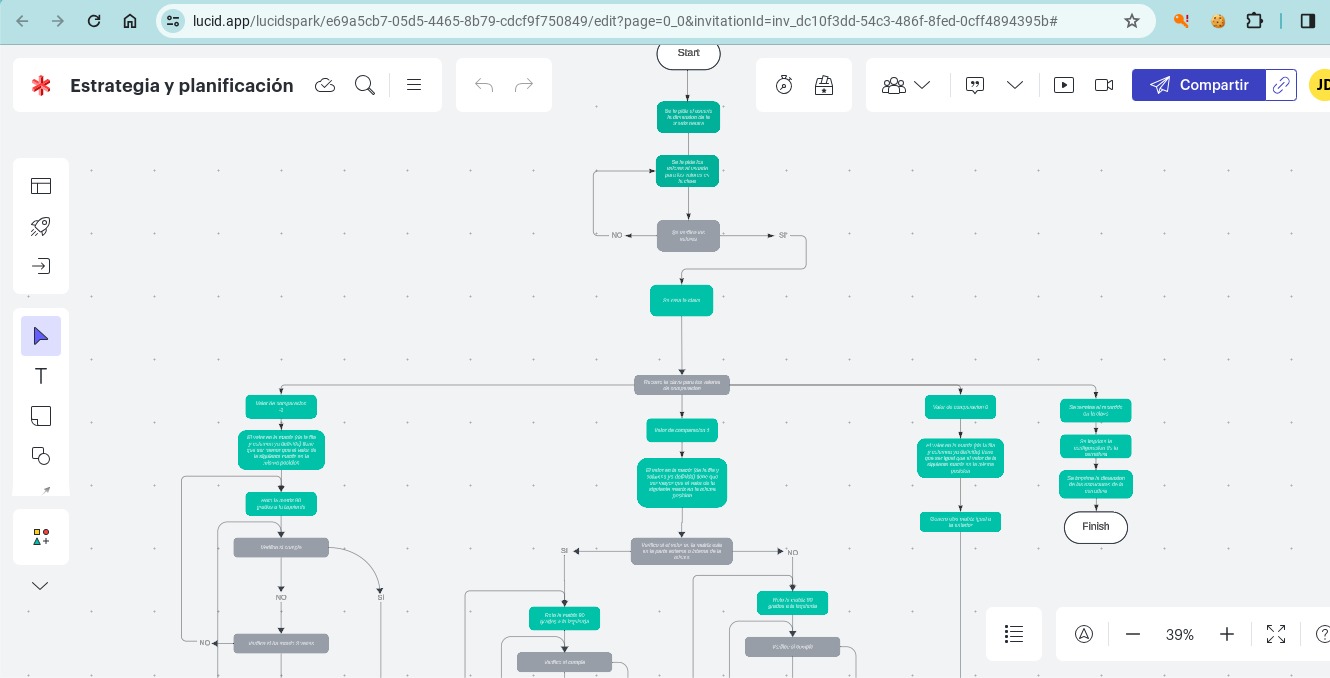


-También se creó un proyecto adicional el cual es el encargado de hacer pruebas de cada función de forma individual y también conjunta, esto con el fin de hacer un seguimiento individual del funcionamiento de la solución propuesta.

Gracias a esto se encontraron diferentes errores durante el proceso que fueron identificados de manera sencilla por este proyecto test adicional.



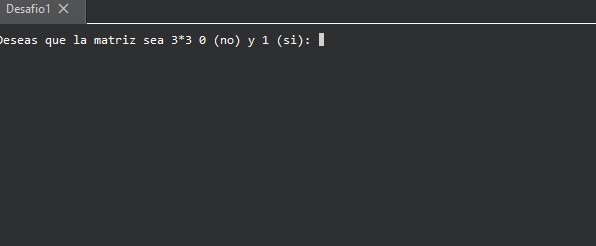
También se hicieron diferentes pruebas a mano donde se analizaba el problema a profundidad para encontrar soluciones más sencillas y eficientes, este proceso fue el más importante ya que nos permitió tener en cuenta excepciones específicas que luego fueron corregidas en el proyecto, aun así hubo de cara a ejecución computacional se mostraron errores que se nos escaparon a nuestro análisis previo.



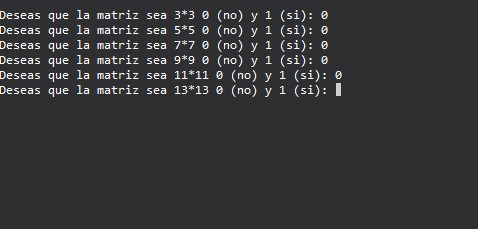
No hubo soluciones descartadas, pero si funcionalidades del proyecto fueron modificadas, el diagrama de flujo anteriormente adjuntado fue de ayuda para analizar el funcionamiento de la solución planteada y gracias a esto y a las estrategias usadas se pudo solucionar el problema de forma satisfactoria.

1. **CODIGOS Y EJEMPLOS DE EJECUCUCÍON**

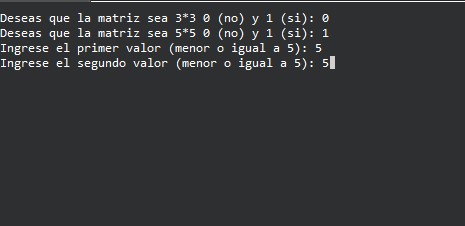
A continuación, se adjuntas imágenes relacionadas con la ejecución del código y sus respectivas explicaciones



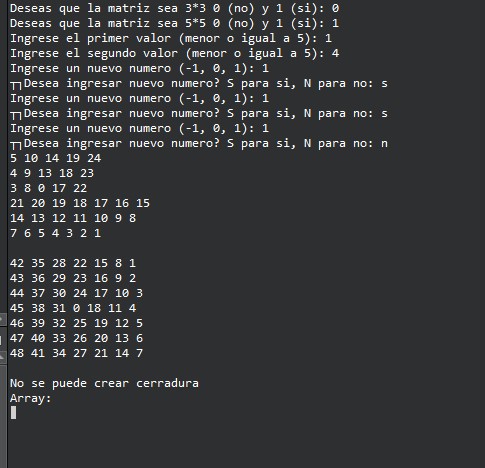
Se le pide al usuario la dimensión la cual desea la matriz, esta se le pide por medio de 0 y 1, donde en el caso de que sea 1 está será la dimensión a usar, este pedido de datos no tiene en cuenta cualquier otra excepción posible



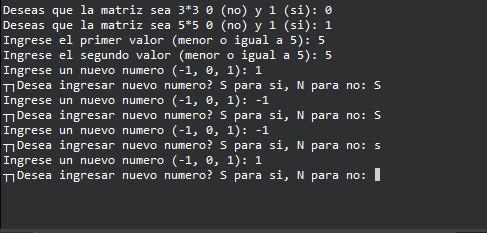
El usuario se le pide cuantas veces sean necesarias y este se encarga de construir la dimensión de la matriz deseada por el usuario

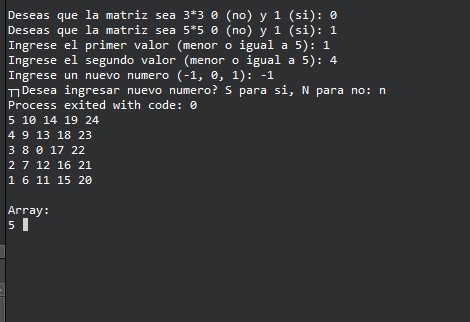


Luego se le piden los datos de la llave, estos tienen la restricción de que los primeros dos valores serán las coordenadas y en caso de que los datos ingresados no sean de la dimensión de la matriz elegida por el usuario le seguirá pidiendo los datos hasta que los cumpla



Por ejemplo, aquí tenemos un caso en el que la matriz y la llave ingresada no es cerradura, se roto y luego se amplió para verificar si es posible, pero al final se imprime un mensaje y se muestra un array vacío





En este caso si se cumple la cerradura, pero como solo se pidió un dato de comparación se agrego al array la dimensión de la matriz que cumplía.

1. **CONCLUSIONES Y POSIBLES ERRORES**

El desarrollo del código fue posible gracias también a un esquema real de cómo funciona de manera específica la solución al problema planteado, se tomó como maqueta diferentes matrices de diferentes dimensiones y luego estás comparadas entre ellas mismas estando alineadas con el centro fijo para cada una, el funcionamiento de cerradura es una forma de definir este problema y la llave es aquella que determina que se cumple las condiciones que esta misma impone para cumplir una cerradura ajustada a esa llave

Este problema fue un reto bastante desafiante, pero gracias a esto nos permitió mejorar nuestras estrategias para afrontar un problema y brindar una solución que satisfaga lo requerido, el trabajo cooperativo fue esencial y también el uso de herramientas de códigos ya hechos por otras personas como es StackOverflow y otras documentaciones oficiales y no oficiales, el proyecto fue realizado enteramente por nuestros medios, también se usó inteligencia artificial para optimizar las funciones realizadas por nosotros. Todas estas herramientas nos mostraron que no necesariamente un problema a resolver se debe hacer a ciegas, si no que podemos ayudarnos e incluso llegar a una solución más completa.

De forma individual nuestras capacidades de análisis, de estructuración de código y nuestras habilidades como programadores mejoraron de forma grata y gracias a los problemas que se nos presentaron por el camino nos hizo darnos cuenta de una falencia que como grupo teníamos y esta era la semántica de nuestro código, gran parte de los problemas se presentaron por esto y también por no ser específicos en los datos que usábamos, gracias a esto corregimos estas falencias como desarrolladores.

En cuanto al apartado de posibles errores se tienen como consideraciones los siguientes puntos:

* Al digitar datos que se escapen de rango para la correcta ejecución de nuestra solución, esto específicamente se tuvo en cuenta a la hora de realizar la llave, pero es posible que esta reciba datos que no cumplan las restricciones que esta debe tener y causaría errores en la ejecución
* Cuando se define una llave sin límite, esto puede ocasionar que el código llegue hasta cierta parte y no recorre todo lo que debe recorrer
* Al realizar pruebas con diferentes ejemplos nos dimos cuenta de que hay excepciones para algunos casos generales, es decir, se descubrió un patrón para la gran mayoría de casos, pero esto puede no abarcar casos muy específicos los cuales no pueden ser solucionados por nuestra propuesta general para el problema.
* Ingresar datos que no corresponden, no se tomaron en cuenta cuando al usuario se le piden datos numéricos y este ingresa datos de otro tipo
* Posible baja eficiencia.

La identificación de estos problemas que lastimosamente no pudieron solucionarse a tiempo fue necesaria para la entrega y las consideraciones a tener en cuenta.

**REFERENCIAS Y BIBLIOGRAFÍA**

<https://stackoverflow.com/questions/618511/a-proper-way-to-create-a-matrix-in-c>

<https://codereview.stackexchange.com/questions/274190/matrix-implementation>

para la creación de matrices y manejo de datos en ella

<https://medium.com/@alegomeznieto/memoria-dinámica-en-c-c5ceb6c47514>

<https://www.programarya.com/Cursos/C++/Estructuras-de-Datos/Punteros>

<https://es.stackoverflow.com/questions/602873/acceder-a-puntero-triple>

<https://devdocs.io/qt/>

Documentación para el uso de cierta información necesaria

<https://www.freecodecamp.org/espanol/news/guia-para-principiantes-de-git-y-github/>

<https://docs.github.com/es/get-started/getting-started-with-git/set-up-git>

<https://docs.github.com/es/get-started/start-your-journey/git-and-github-learning-resources>

para el uso del control de versiones