

## Informe Desafío1

Jorge Enrique Rueda Urrea - 1.020.488.162  
Edison Fredy Serrano Álvarez - 1.055.273.822  
Ingeniería de Telecomunicaciones

Fecha: Abril 2024



UNIVERSIDAD  
DE ANTIOQUIA

Facultad de Ingeniería

**Keywords:** Estructuras, funciones, cerradura, datos, arreglos, memoria, punteros

### 1 RESUMEN INTRODUCTORIO:

Este proyecto implica el desarrollo de un programa en C++ que simula la apertura de una "cerradura". La arquitectura del sistema está concebida para gestionar la "cerradura", que se compone de matrices de diferentes dimensiones. Nuestro objetivo es realizar una serie de operaciones sobre estas matrices para lograr abrir la "cerradura". Para esto, se requiere girar cada "matriz" de manera independiente para validar la regla de apertura. Nuestro programa hace uso de arreglos y punteros para representar y operar sobre las matrices, además de implementar diversas funciones para modularizar el código y dividir el programa principal en subprogramas. Esta combinación de elementos conforma la estructura final de nuestro proyecto.

### 2 ARQUITECTURA DEL CODIGO:

Para la implementación de nuestra solución haremos uso de las siguientes funcionalidades del lenguaje C++:

- Punteros y Memoria Dinámica
- Funciones y Modularidad
- Iteración y Control de Flujo

### 3 ANÁLISIS GENERAL

Nuestro código implementa un sólido manejo de errores y validación de entradas para garantizar la correcta ejecución de las operaciones. Además, es crucial tener en cuenta la eficiencia y el uso de recursos, ya que el uso de memoria dinámica y la manipulación de matrices pueden requerir un cuidadoso manejo para evitar fugas de memoria o errores de acceso. En términos de complejidad algorítmica, la variabilidad de nuestro código se presenta en función de las operaciones realizadas, tales como la creación de matrices, comparaciones y ajustes de tamaño, donde el número de iteraciones y operaciones puede aumentar significativamente con tamaños grandes de matrices.

### 4 ESTRUCTURA DE DATOS M:

- Se nos proporciona una estructura de datos llamada M que puede girar y cambiar de tamaño. La estructura M se representa como una matriz de enteros.
- La matriz M tiene la particularidad de poder rotar, es decir, cambiar su disposición de manera que los elementos se muevan de una manera específica. Por ejemplo, girar la matriz en sentido horario.
- También puede cambiar sus dimensiones, es decir, puede tener diferentes números de filas y columnas, siempre y cuando la cantidad de filas sea impar e igual al número de columnas

### 5 CERRADURAS X

- La empresa de seguridad Informa2 utiliza la estructura M para diseñar cerraduras, que se denominarán X.
- Estas cerraduras X están compuestas por varias estructuras M alineadas una tras otra, utilizando la celda del centro de cada estructura como referencia.
- No hay restricciones en cuanto a la cantidad de estructuras M que componen una cerradura X, y las estructuras pueden tener diferentes tamaños.

### 6 SISTEMA DE APERTURA DE LA CERRADURA X:

- El sistema de apertura de una cerradura X funciona a partir de la validación de una regla K.
- La regla K considera el valor de una celda específica, su posición y la ubicación dentro de las diferentes estructuras alineadas
- Por ejemplo, se nos da una posible entrada K que consiste en una serie de valores que indican condiciones para las celdas dentro de las estructuras alineadas.
- Para abrir la cerradura X, es necesario rotar cada una de las estructuras de forma independiente, alineando las celdas de tal manera que la validación de la regla K sea verdadera.

### 7 IMPLEMENTACION EN C++:

- Para implementar este ejercicio en C++, representamos las estructuras M y la cerradura X utilizando arreglos dinámicos y punteros.
- Creamos funciones para crear, rotar y liberar la memoria de las estructuras M.
- Implementamos una función para validar la regla K en una cerradura X, la cual recorre las estructuras alineadas y verificará si se cumplen las condiciones especificadas en la regla. Esto implica acceder a las celdas específicas de las diferentes estructuras alineadas y comparar sus valores según lo especificado en la regla.
- Implementamos una función para generar configuraciones de cerraduras a partir de una regla, esta función va a generar al menos una configuración de cerradura que cumpla con la regla especificada por el usuario.
- Finalmente, en el programa principal, creamos las estructuras M según las especificaciones de la cerradura X, definiremos una regla K, rotamos las estructuras según sea necesario y validamos la apertura de la cerradura X.

## 8 ANALISIS VARIOS:

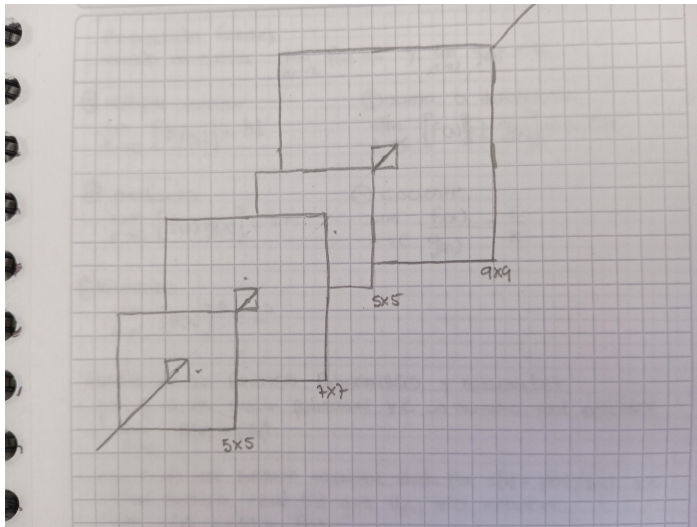


Figure 1: Consideraciones iniciales de como deben ir las matrices alineadas por el centro.

## 9 FUNCIONES USADAS:

Para nuestra solución implementamos las siguientes funciones, cuyo contenido se verá reflejado en el programa principal:

- imprimirMatriz: Función para imprimir nuestras matrices
- crearMatriz: Función para crear la matriz según un tamaño
- obtenerContenidoCelda: Función para obtener el contenido de una celda con el propósito de comparar
- compararMatrices: Función para comparar dos matrices dadas
- liberarMatriz: Función para liberar memoria
- crearArregloDeMatrices: Función para crear un arreglo de matrices que será básicamente la cerradura
- igualarTamanos: Función para igualar tamaños de matrices con el fin de poder compararlas
- imprimirCerradura: Función para imprimir la cerradura generada anteriormente
- esDigito: Función para validar si tenemos un dígito
- esNumeroEnteroPositivo: Función para validar si un número ingresado es positivo

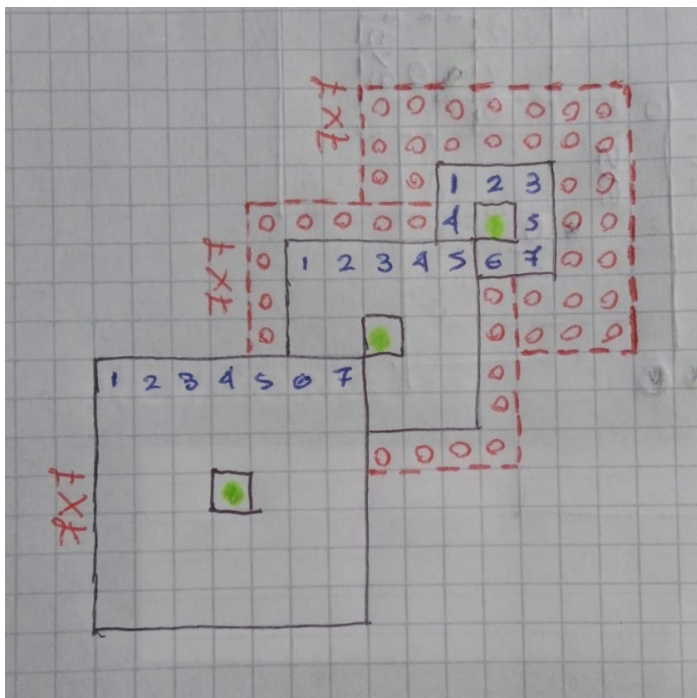


Figure 2: Analisis de como deben ser las matrices para poder comparar posiciones correctamente.