

Informe Desarrollo Parcial I

Manuel Esteban Orjuela Montealegre

Universidad De Antioquia

Facultad de Ingeniería

Departamento de Electrónica y Telecomunicaciones

Informática II

Semestre 2024 – 1

Análisis del Problema y Consideraciones para la Alternativa de la Solución Propuesta

Estructura de Datos.

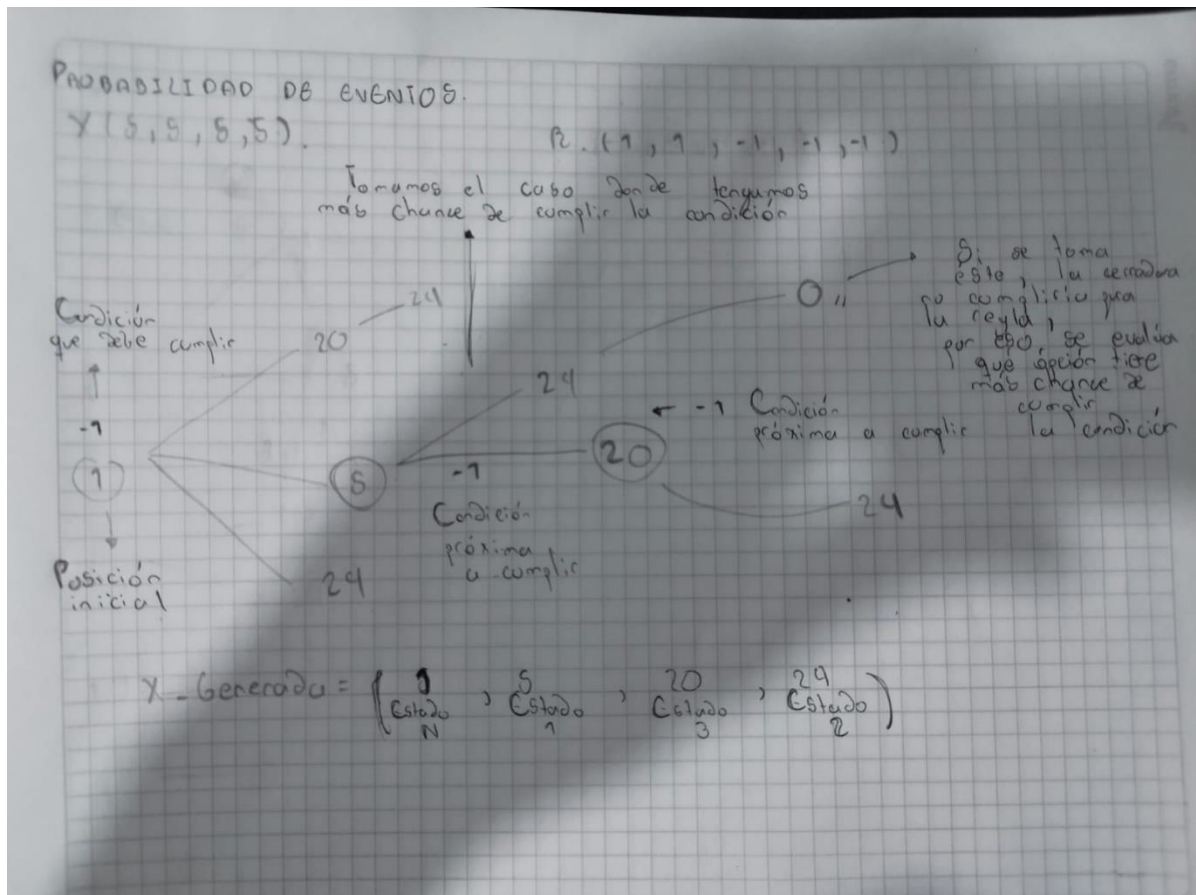
Para la solución planteada se tomaron en cuenta las siguientes consideraciones para el resguardo de datos:

- . Utilizar una estructura dinámica triple que almacene los grupos de matrices de acuerdo a una cerradura.
- . Utilizar tres arreglos dinámicos encargados de almacenar la información correspondiente a las dimensiones de la matriz, las rotaciones de la matriz y la regla.

Selección caso óptimo.

De acuerdo a un análisis estadístico desarrollado, partí de la siguiente premisa:

- . Existen múltiples momentos en las que una matriz en una posición específica cumpla con la condición establecida por la regla. En consecuencia, es posible que, al seleccionar una opción, no se opte por la más 'eficiente' debido a distintos factores como la aleatoriedad o ser la primera opción encontrada. Por lo tanto, se realizó el siguiente análisis para determinar el caso más favorable, con el objetivo de elegir la opción que maximice las posibilidades de que una regla se cumpla en una matriz.



Cada vez que estemos evaluando una cerradura, tomamos las dos estructuras en comparación y almacenamos en otra estructura dinámica triple los casos donde se cumple la condición para luego elegir el más óptimo.

Generar una Cerradura a partir de una llave.

Para el módulo de generar una cerradura a partir de una llave aproveché el algoritmo anteriormente encontrado y por cuestiones de tiempo, limité los parámetros de solución en lo siguiente:

- . La cerradura generada tendrá máximo 6 elementos.
- . Las dimensiones de las estructuras dentro de la cerradura solo tendrán máximo 13 dimensiones. {3,5,7,9,11,13}.

La estrategia empleada para encontrar las cerraduras implicó manipular las dimensiones de las matrices de manera aleatoria dentro de un arreglo unidimensional independiente. Este enfoque permitió cumplir con la condición establecida por la regla. La solución hallada consistió en determinar las dimensiones con las que trabajaré, identificar un índice mínimo (el cual representa la posición que todas las matrices dentro de la cerradura deben satisfacer, determinada por la información proporcionada en la regla), y utilizar un generador de números aleatorios para definir las dimensiones de la cerradura. Posteriormente, se evaluó la regla sobre la cerradura resultante.

Tareas Definidas para el Desarrollo de Algoritmos

La implementación desarrollada fue dividida en 8 módulos para facilitar el desarrollo.

Modulo 1. Creación de estructuras bidimensionales de tamaño variable.

Modulo 2. Rotación de estructuras bidimensionales de tamaño variable.

Modulo 3. Creación de estructura tridimensional de tamaño variable.

Modulo 4. Liberación de memoria de estructuras dinámicas

Modulo 5. Validación de una regla sobre una cerradura.

Modulo 6. Validaciones Generales.

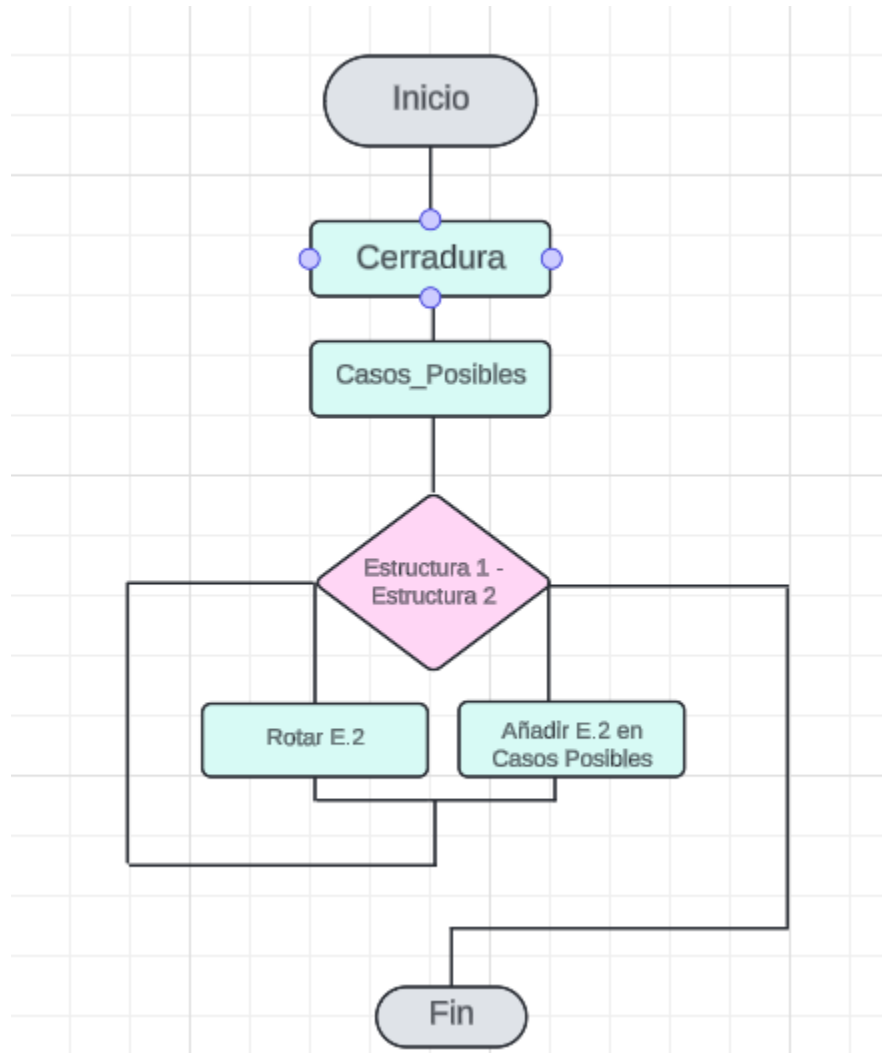
Modulo 7. Generación de estructuras tridimensionales y unidimensionales que cumplen las condiciones de la regla.

Modulo 8. Generación de una estructura tridimensional a partir de una regla.

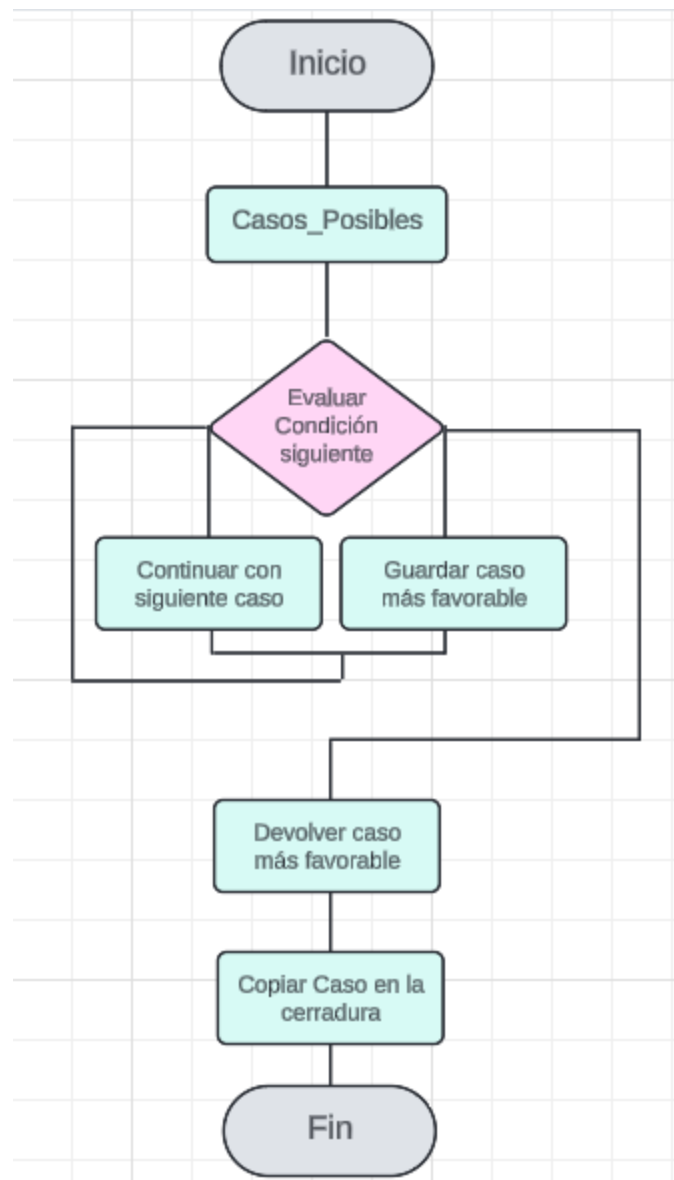
Algoritmos Implementados

En este apartado se describirán los algoritmos fundamentales de mi implementación.

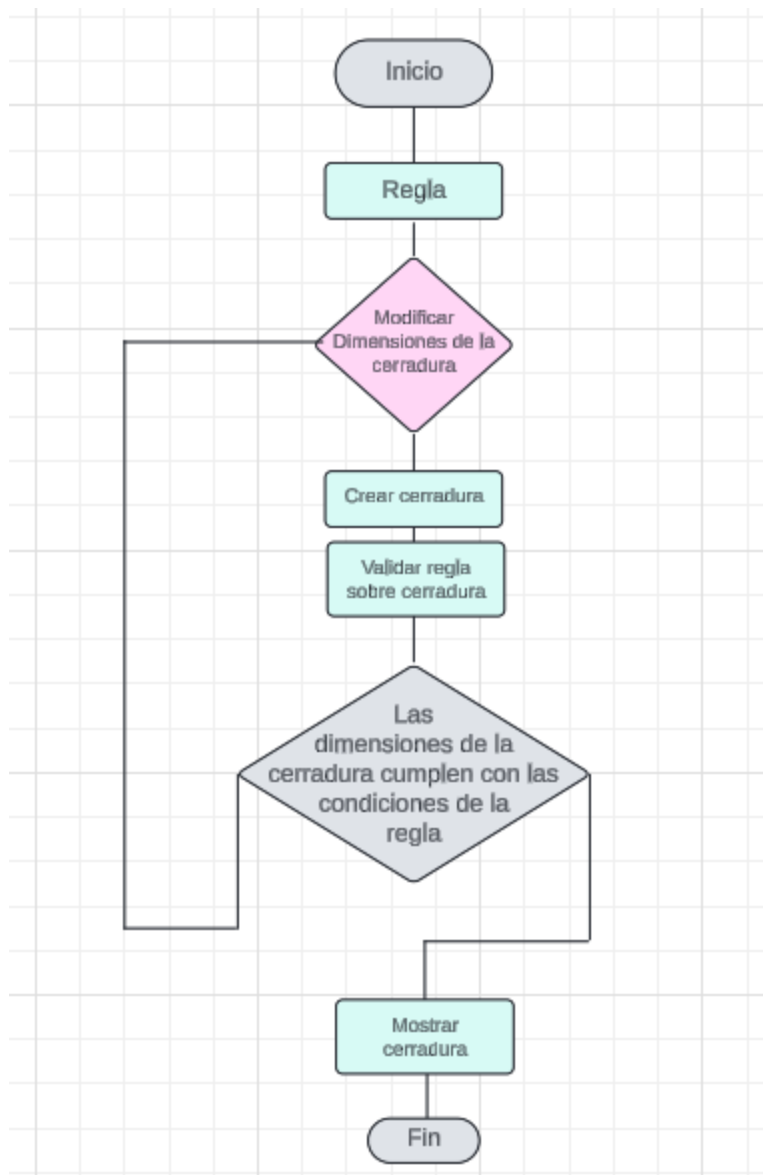
Búsqueda de casos posibles donde las estructuras de una cerradura cumplan la condición.



Selección caso posible más favorable.



Generar una cerradura a partir de una regla.



Problemas en el desarrollo y evolución de la Solución

Momento 1. Creación estructura de datos.

Solución propuesta: En mi primera implementación codifiqué tareas básicas para la manipulación de arreglos bidimensionales tal como crear, eliminar, rotar, copiar e inicializar.

Además de eso, hice mis primeras pruebas para las estructuras de datos para la cerradura, regla, dimensión de matrices y las rotaciones junto con sus respectivas liberaciones de memoria.

Momento 2. Primera implementación de validar regla sobre cerradura.

Solución propuesta: Evaluar cada par de estructuras internas de la cerradura y observar si la condición de la regla se cumple realizando rotaciones.

Problema hallado: El programa tomaba la primera opción que se presentaba por lo que existían reglas que se cumplían sobre una cerradura si se rotaba la primera matriz de la cerradura (En este momento solo evaluaba dos estructuras por la facilidad de análisis), pero se abortaba la ejecución al encontrar una opción disponible.

Momento 3. Rotar las primeras dos matrices para así obtener todas las combinaciones iniciales.

Solución propuesta: Se establecerá un puntero triple para almacenar todas las posibles combinaciones identificadas. Luego, mediante un proceso iterativo, se recorrerán cada par de estructuras de los casos. Durante este recorrido, se copiarán las opciones encontradas en las dos primeras matrices de las cerraduras, y posteriormente se evaluarán.

Problema hallado: Ahora, el proceso es doblemente iterativo y más largo, por lo que, menudo selecciona la primera opción disponible en la cerradura, pasando por alto los casos más favorables donde la validación de la llave podría tener lugar en una cerradura cumpliendo la condición. Este enfoque puede resultar en una solución que no aprovecha al máximo las posibilidades disponibles.

Momento 4. Analizar y elegir el caso más favorable.

Solución propuesta: Almacenamos todas las opciones disponibles en otra estructura tridimensional y mediante un análisis explicado anteriormente, elegimos la opción más óptima.

Problema encontrado: Fugas de memoria en el proceso de copia de la estructura de las posibilidades. Problema: Heap block at 000001F93F494360 modified at 000001F93F494388.

Momento 5. Decisiones y optimización.

Solución propuesta: Asignar como fija 'Estado neutro' a la primera matriz de la cerradura para así evitar problemas de fuga de memoria con la creación de las rotaciones entre las dos primeras estructuras. Para el tema de la fuga de memoria, asignar al puntero triple dos espacios adicionales cuando se encuentre una opción, un lugar para guardar la matriz encontrada y otro para un espacio nulo.

Momento 6. Generar Cerradura.

Solución propuesta: Limitar parámetros como cantidad de estructuras y un límite de dimensión para facilitar el trabajo. Mediante un numero aleatorio, destinar las dimensiones de la cerradura y evaluar la regla sobre la cerradura que se crea.