

# Informe evolución y planteamiento del Parcial 1

## Informática II

Juan Pablo González Blandón, Juan José Balvin Torres.

*Estudiantes de la facultad de ingeniería de la Universidad de Antioquia*

**Resumen—** En este informe podrán encontrar avances, consideraciones, análisis y demás aspectos relacionados con el Parcial 1 de Informática II de la Universidad de Antioquia

Cabe recalcar que son ítems moldeables con el tiempo y en constante evolución, más sin embargo la base de la solución será la misma.

### I. INTRODUCCIÓN

En La empresa Informa2 tiene la necesidad de que dada una Regla K, se genere una configuración de X que la satisfaga. La salida del programa deber ser la configuración X, tamaño de cada una de las dimensiones y las rotaciones que se deben hacer en cada una de sus estructuras para abrir la correspondiente cerradura.

Para más información sobre el proyecto a realizar dar click en el siguiente enlace:

[https://udearroba.udea.edu.co/internos/pluginfile.php/2364452/mod\\_assign/introattachment/0/Desafio%201.pdf?forcedownload=1](https://udearroba.udea.edu.co/internos/pluginfile.php/2364452/mod_assign/introattachment/0/Desafio%201.pdf?forcedownload=1)

### II. ANÁLISIS DEL PROBLEMA Y CONSIDERACIONES PARA LA ALTERNATIVA DE SOLUCIÓN PROPUESTA.

Dada una contraseña K se desea hallar una cerradura X que la satisfaga, para esto debemos tener en cuenta que no se sabe la cantidad de estructuras de datos (M) alineadas, que consiste en un conjunto de matrices de cuadradas de tamaño variables, más sin embargo se pueden deducir mediante la contraseña K ingresada por el usuario.

Se sabe que la contraseña K tiene la siguiente estructura: K (m, n, cond 1, cond 2,...,cond i)

Siendo m la fila y n la columna de la posición de M, de allí podemos deducir entonces que la cerradura X debe contener como mínimo estructuras M con la siguiente condición:

Si  $m > n \rightarrow M$  es de orden  $m \times m$ , Si  $n > m \rightarrow M$  es de orden  $n \times n$ .

Ahora bien, la cerradura X debe contener  $i+1$  estructuras M debido a que depende de las condiciones se hacen comparación de par en par (de a dos estructuras).

Teniendo en cuenta todo lo anterior para llegar a conocer la

cerradura X de la forma: X (estructura 1, estructura 2, estructura  $i+1$ ), se necesita llegar a verificar que las condiciones de k se cumplan, para esto se propone la siguiente solución, analizar todas las combinaciones posibles teniendo en cuenta el tamaño de cada matriz y cada uno de sus estados, cabe recalcar que este procedimiento puede llegar a ser muy ineficiente, por lo que se esta buscando una solución diferente mediante patrones que pueda reducir las iteraciones necesarias para llegar a cumplir con el objetivo.

### III. ESQUEMA DE ALGORITMOS

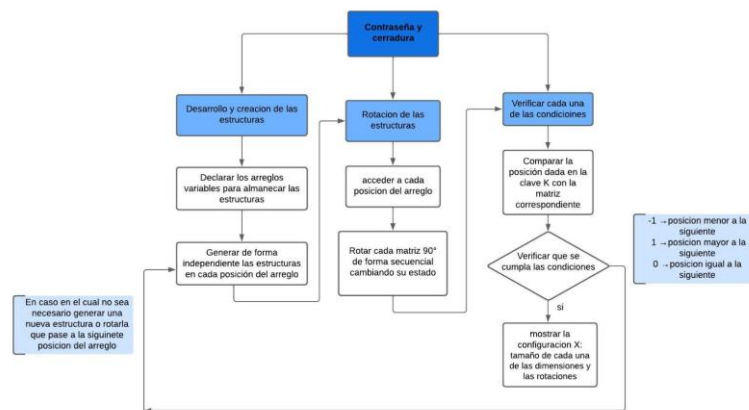


Fig. 1. En esta figura se puede visualizar el esquema empleado para definir las tareas de los algoritmos, este esquema lo podrá encontrar en el repositorio para visualizarlo de una mejor forma.

### IV. ALGORITMOS IMPLEMENTADOS

Los algoritmos implementados para la solución de la problemática planteada en la introducción son:

#### *Algoritmos que pueden encontrar en el repositorio*

- Algoritmo encargado de la generación de las matrices correspondientes a la clave K por el usuario
- Algoritmo que permita realizar la rotación de dichas matrices
- Pequeño programa inicial para pedir al usuario la clave K y verificar que esta misma este correcta

\* Facultad de ingeniería de la Universidad de Antioquia

### *Algoritmos que están en proceso de implementación*

- Algoritmo encargado para realizar las rotaciones u aumentos de orden necesarios a las estructuras contenidas en la cerradura
- Algoritmo que valide cuando la cerradura es la pertinente para la clave y cuando lo logre, que imprima esta misma, la dimensión de las matrices y los estados de estas mismas

### V. PROBLEMAS EXPERIMENTADOS

Durante el desarrollo de los algoritmos anteriores concretamente en el de la generación de las matrices aunque sabías que debías de tener un espacio en blanco en la mitad de la estructura, no habíamos aterrizado bien que implicaciones tenía como por ejemplo: que al ser un espacio en la mitad, la matriz se debía rellenar con caracteres ya que no se puede combinar tipos en un arreglo, e incluso usando caracteres los números con dos dígitos o mas ocupaban dicha cantidad de caracteres, por ende se opto por poner un cero en la mitad.

Debido a la poca familiarización con el sistema de versionamiento de GitHub tuvimos ciertas fallas, pero con algo de investigación pudimos solventar.

Otro problema que es bastante importante de mejorar es la comunicación con el colaborador y realizar las tareas pensando en que en un futuro van a trabajar en conjunto con las del equipo de trabajo, por ende, algunos algoritmos puede que no encajen de forma cooperativa y sea necesario modificarlos.

### VI. EVOLUCIÓN DE LA SOLUCIÓN Y CONSIDERACIONES A TENER EN CUENTA EN LA IMPLEMENTACIÓN

Como ya se mencionó anteriormente la consideración más fundamental en la implementación es que las funciones que se realicen de forma individual, puedan ser conectadas de forma satisfactoria al realizar los pull request.

En un principio no se entendía bien el problema y se veía algo complicado, pero, a medida que se fue analizando y llevando una lluvia de ideas y aportes del equipo de trabajo se llegó a una solución general y bastante satisfactoria de como abordar la problemática. La evolución y el seguimiento del proyecto en código fuente se puede ser visualizada en el repositorio.