

## Proyecto #4

Emanuelle Jiménez Sancho.  
2017136727.

Instituto Tecnológico de Costa Rica

Lenguajes de programación

Maria Mora

12/11/19

<b>Descripción del problema.</b>	<b>3</b>
<b>Diseño del sistema y solución.</b>	<b>3</b>
<b>Objetivos alcanzados</b>	<b>4</b>
<b>Objetivos no alcanzados</b>	<b>4</b>
<b>Instrucciones uso de la aplicación.</b>	<b>4</b>
<b>Conclusión personal.</b>	<b>6</b>

## Descripción del problema.

El problema a tratar es realizar funciones contra matrices que pueden muy grandes, esto hace muy ineficiente tanto los algoritmos aplicados hacia la matriz, como el espacio en la memoria de dichos datos. Es por eso que se utilizan matrices dispersas, que le permiten agregar únicamente los valores insertados a la matriz, ahorrando así el tiempo de recorrido de la matriz, como la reducción del espacio de almacenamiento del mismo.

## Diseño del sistema y solución.

Como bien se mencionó anteriormente, la solución a este problema es utilizar matrices dispersas, que le permitan al sistema ahorrar espacio y tiempo de procesamiento.

Van a existir dos tipos de estructuras en el programa, una lista general, que me permite conocer TODOS los valores de la matriz, esta es una lista con la fila, columna y el valor en dicho campo. De igual forma existe una estructura de matriz, que me ayuda a saber que tan grande es la matriz y un puntero a la lista.

Consideraciones a tomar en cuenta:

- A la hora de cargar la matriz, esta debe tener exactamente el mismo tamaño que los datos del csv.
- Cuando se carga una matriz, esta no puede insertar más elementos en ella.

## Objetivos alcanzados

Todos los objetivos fueron alcanzados

## Objetivos no alcanzados

No hubieron objetivos que no fueron alcanzados.

## Instrucciones uso de la aplicación.

El programa tiene una secuencia de esta forma:

- Primero le pide al usuario el tamaño de la matriz.
- Seguidamente le pide al usuario si quiere meter valores a la matriz, como mínimo tiene que haber un valor.
- Luego le despliega el menú, que es el siguiente:

0.Crear matriz

1.Insertar elementos en la matriz:

2.Sumar columnas:

3.Sumar filas:

4.Calcular transpuesta:

5.Imprimir matriz:

6.Guardar matriz:

7.Cargar matriz:

8.Multiplicar matrices:

9.Sumar matrices:

- En este caso el sistema le pide al usuario lo que desea hacer
- Se debe crear la matriz primero antes de realizar alguna función.
- Se deben insertar los datos uno por uno en la segunda opción.
- En caso de que sea sumar columnas, entonces el sistema le pide cuál columna quiere sumar.
- En caso de que sea sumar filas, entonces el sistema le pide cuál fila quiere sumar.
- En caso de que quiera calcular la transpuesta, entonces el sistema simplemente imprime una matriz con ceros incluidos y con los datos transpuestos.
- En caso de que desee guardar la matriz, este lo almacena en el **mismo** directorio del programa.
- En caso de cargar la matriz, este lo buscará dentro del **mismo** directorio.
- En caso de imprimir la matriz, este lo imprime.
- En caso de multiplicar o sumar, el sistema le pide que inserte 2 matrices nuevas y sus valores.

## Conclusión personal.

Este proyecto me enseñó a utilizar memoria dinámica en c, algo nunca antes visto en tareas o proyectos de la U. Me ayudó de igual forma a reforzar los conceptos de C y del paradigma imperativo, estos conceptos los veremos próximamente en compiladores, es por eso que agradezco tener dicho proyecto como conclusión del curso.