

GUÍA N°7  
*Algoritmos con matrices*

**Ejercicio 1**

Escribir matemáticamente algoritmos para resolver los siguientes problemas.

- Calcular la suma de la diagonal principal de una matriz cuadrada.
- Calcular la suma de la diagonal secundaria de una matriz cuadrada.
- Construir un arreglo lineal donde cada elemento del mismo es la suma de cada fila de un arreglo bidimensional rectangular.
- Dada una matriz A de  $n \times m$  y un vector b de m elementos, calcular A.b.
- Dadas dos matrices de  $n \times m$ , construir la matriz suma.
- Dadas dos matrices, construir si es posible la matriz producto.
- Construir la matriz transpuesta.
- Calcular la suma de todos los elementos de la matriz.
- Calcular la cantidad de elementos iguales a cero de la matriz.
- Dada una matriz cuadrada A y un escalar c, construir  $B = c.A$ .

**Ejercicio 2**

Dada una matriz cuadrada, desarrollar algoritmos para:

- Verificar si es simétrica.
- Verificar si es diagonalmente dominante.

Una matriz es diagonalmente dominante si

$$|a_{i,i}| > \sum_{j=1, j \neq i}^n |a_{i,j}|, \forall i = \{1, \dots, n\}$$

**Ejercicio 3**

Escribir matemáticamente algoritmos para la manipulación de matrices especiales. Para minimizar el tiempo de ejecución, los algoritmos a diseñar deben evitar realizar operaciones cuyos resultados se conozcan a priori (por ejemplo: evita una multiplicación por 0 o por 1).

- Sean A y B matrices cuadradas de  $n \times n$  triangulares superiores y x un vector de tamaño n. Diseñar algoritmos para calcular:
  - A.x
  - A+B
  - A.B
- Repetir a) pero con matrices de Hessenberg inferior.
- Repetir a) con matrices tridiagonales.
- Repetir a) con matrices de Hessenberg superior.
- Sea A una matriz triangular inferior y B una triangular superior diseñar un algoritmo para calcular A.B.

**Ejercicio 4**

Escribir matemáticamente el algoritmo para ortogonalizar una base de un espacio n-dimensional (método Gram Smith). Implementarlo en Java.