

UNIVERSIDAD  
**AUSTRAL**



**INGENIERÍA**

---

# Álgebra III

## TP III

Algoritmos con matrices

## Ejercicios

---

Cada equipo deberá entregar los ejercicios que le correspondan. Los ejercicios se pueden encontrar en la "Guía N 7 Algoritmos con matrices" que está en el campus en la Unidad 5.

## Entrega

---

La fecha límite de entrega es el día **Martes 31 de Mayo**. El alumno deberá entregar en el campus el archivo: "**Guide7Solution.java**" con los ejercicios correspondientes; éste y los test que serán usados para corregir se pueden encontrar en el repositorio de github: <https://github.com/aolmedo87/algebra3>.

## Aclaraciones para el ejercicio 3

---

En el ejercicio 3 se pide: "*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)*". Para poder testearlo, es necesario conocer la cantidad de accesos a las matrices/vectores. Para esto están las clases "MatrixMatrixOperation" y "MatrixVectorOperation". Deberán utilizar dichas clases como contenedores de las matrices al momento de programar los algoritmos. Estas actúan como un contenedor para: los 2 operandos, el resultado y un contador de accesos a la matrix. La manera de acceder a las matrices es a través de los métodos `getValue()` que retornan de a un elemento.

Aclaración: **En el caso de no acceder a los valores de la matriz a través del método `getValue()` o no utilizar las clases utilitarias para realizar las operaciones, se tomará el punto como desaprobado.**

Ejemplo de uso (tomado de la clase Example en el código)

Supongamos que un ejercicio pidiera un algoritmo para optimizar la multiplicación de una matriz con ceros en la diagonal con un vector

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} (0) \cdot (2) + (1) \cdot (3) + (1) \cdot (4) \\ (1) \cdot (2) + (0) \cdot (3) + (1) \cdot (4) \\ (1) \cdot (2) + (1) \cdot (3) + (0) \cdot (4) \end{bmatrix}$$

Podemos ver que los accesos totales son 18. Este número no es óptimo, ya que no tiene en cuenta que la matriz tiene ceros en la diagonal. Sacando esos accesos, nos quedan 12.

¿Cómo se puede optimizar? Si nos fijamos, cada vez que “i” es igual que “j” (los índices de los for loop anidados) hay un cero en la matriz. Al agregar la condición que i sea distinto de j, se saltan los que son cero

```
for (int i = 0; i < rows; i++) {  
    for (int j = 0; j < columns; j++) {  
        If i == j the matrix has a zero, so we skip it, making less calculations  
        if (i != j)  
            result[i] += op.multiplyPositions(i, j, j);  
    }  
}
```

¿Por qué las matrices de los tests tienen 9 en vez de 0?

En los tests del ejercicio 3 encontrarán que en vez de tener 0 en estas posiciones van a tener 9 como valor. Esto es otra manera de comprobar que no se acceda a esa posición, ya que no se deberían usar. Por ejemplo, en el ejercicio 3a la consigna indica que las matrices son cuadradas triangulares superiores. Esto implica que por debajo de la diagonal de la matriz todos los valores son 0. Por ende, en los tests, utilizaremos matrices que en vez de 0 contienen 9 en estas posiciones para asegurarnos que no se acceda a esa posición.

## Corrección

---

Se evaluarán 2 aspectos:

1. **JUNIT tests:** Todos los test de “Guide7Test.java” que corresponden a los ejercicios de la entrega deben pasar. Si el test falla, el ejercicio se da por desaprobado (0 puntos).
2. **Análisis del código:** Una vez pasados los tests se leerá que el código sea correcto, es decir que sea claro de leer (nombre de variables y métodos auxiliares) y no tenga operaciones innecesarias. Esto no penaliza mucho, a menos que haya un error muy grave.
3. **Análisis de la expresión matemática:** Se revisará la expresión matemática de cada ejercicio.

## Nota

---

La tabla a continuación muestra la cantidad de puntos por ejercicio:

	Ej 1	Ej 2	Ej 3	Ej 4
Puntos	1.5	3	1.5	4

La nota final es la suma de los puntos totales, siguiendo la siguiente fórmula de interpolación.

$$nota = \left( \frac{p-6}{4} \right) \cdot 10 + \left( \frac{p-10}{-4} \right) \cdot 4$$

## Consultas

---

Para: aclaraciones, consultas, dudas, sugerencias, y demás mandar un mail a [agustin.olmedo@ing.austral.edu.ar](mailto:agustin.olmedo@ing.austral.edu.ar).