

## PROYECTO DE MATRICES

### (Configuración para Autograding)

**INSTRUCCIONES:** Configura una tarea con `Autograding` para que los alumnos puedan resolver una tarea de manejo básico de matrices en base a los siguientes requisitos:

1.- Se debería crear una clase **Matriz** para manejar matrices de números enteros. Las variables de instancia de esta clase deberían incluir: (1) Número de renglones (2) Número de columnas (3) Elementos de la matriz.

2.- La Matriz debería construirse a partir de la lectura de un archivo **mX.mat**. Este sería un archivo de texto que incluiría información de las dimensiones de la matriz y sus elementos.

**Ej.-** Para una matriz de 2x2 el archivo **mX.mat** tendría la siguiente estructura:

```
2x2
1 3
-1 0
```

**3.-La corrida del programa** debería mostrar en consola (pantalla):

```
Leyendo matriz A...
Leyendo matriz B...
```

```
Sumando matrices:
```

```
1 3
-1 0
+
```

```
4 5
-1 0
=
```

```
5 8
-2 0
```

```
Multiplicando matrices:
```

```
1 3
-1 0
+
```

```
4 5
-1 0
=
```

```
1 5
-4 -5
```

3.- Los métodos de la clase **Matriz** los define el programador acorde a lo necesario para satisfacer los requisitos (**Ej.-** `suma()`, `multiplica()`, `getRenglones()`, `getColumnas()`, `esIgualA()`, etc).

## CONFIGURACIÓN PARA AUTOGRADING

### I.- CREA EL REPOSITORIO PLANTILLA BASE PARA LA TAREA (20% DE LA EVIDENCIA)

- a) A partir del repositorio plantilla para ejercicios de `Autograding` diseña el repositorio solución para implementar esta tarea. Agrega los dos archivos **m1.mat** y **m2.mat** para que el alumno pueda probar su tarea de acuerdo a los requisitos.
- b) Toma 1 o 2 capturas de pantalla de lo que has hecho y agrégalo a la evidencia para acreditación del curso.

### II.- CREA EL REPOSITORIO PLANTILLA COMPLETO DE LA TAREA (50% DE LA EVIDENCIA)

- a) Crea las siguientes pruebas para calificar tu proyecto:
  - 1) **Prueba 1:** que valida que se ha **programado correctamente la suma de matrices**. Se configurará como prueba Java en `GitHub Classroom`. (Valor: 25 puntos).
  - 2) **Prueba 2:** que valida que si desean multiplicar matrices, **se cumple con la condición de multiplicación** (las columnas de la primer matriz deben coincidir con los renglones de la segunda matriz). Se configurará como prueba Java en `GitHub Classroom`. (Valor: 25 puntos)
  - 3) **Prueba 3:** que valida que se ha **programado correctamente la multiplicación de matrices**. Se configurará como prueba Java en `GitHub Classroom`. (Valor: 25 puntos)
  - 4) **Prueba 4:** que valida que **la salida del programa del alumno es de acuerdo a la corrida indicada en los requisitos**. Se configurará como prueba Input/Output en `GitHub Classroom`. (Valor: 25 puntos)
- b) A partir del repositorio anterior, obtén una copia **\*.tmp** para preparar el repositorio que distribuirás como código inicial en `GitHub Classroom` para las tareas de los alumnos.
- c) Determina el nivel de dificultad de tu tarea. Acorde a tu decisión, elimina lo que consideres conveniente en el código inicial de tu repositorio **\*.tmp**. No elimines las pruebas.
- d) Toma 1, 2 o 3 capturas de pantalla de lo que has hecho y agrégalo a la evidencia para acreditación del curso.

### III.- CONFIGURA EL EJERCICIO PARA GITHUB CLASSROOM: (30% DE LA EVIDENCIA)

- a) Dentro de la clase que tienes disponible por el curso en `GitHub Classroom`, crea la tarea **MATRICES** y configura todo lo necesario para su habilitación como **Autograding**.
- b) Prueba la liga que te ofrece `GitHub Classroom` y resuelve el ejercicio mostrando el resultado obtenido en las pruebas locales y en el **Autograding** de `GitHub`, deberían coincidir.
- c) Toma 1 o 2 capturas de pantalla de lo que has hecho y agrégalo a la evidencia para acreditación del curso.

### IV.- ENTREGA DE EVIDENCIA EN TEAMS.

- a) Suba tu evidencia en el apartado de tareas del curso en Microsoft Teams.