**Compañero 1**

* **Coordina con el compañero 2:**
  + **Asegúrate de que la estructura de tus matrices sea compatible con sus funciones de multiplicación.**

**Objetivo:**

* Pedir al usuario el **número de filas y columnas** (debe validar que sean números enteros positivos).
* Solicitar **cada elemento de la matriz** uno por uno (debe asegurarse de que sean números válidos, no letras ni símbolos).

**Suma de matrices:**

1. **Suma de matrices:**
   * Crear el código para sumar las matrices A y B.
     + Verifique que ambas matrices sean del mismo tamaño.
     + Retorne una nueva matriz con la suma de cada elemento
   * Manejar error como las que las dimensiones coincidan y se podría mostrar un mensaje del error

**IMPORTANTE:**

* No usar input() o print().
* Asegurarse de que las entras solo acepten números
* Comentar el código para entenderlo mejor explicando la función que cumple esa parte del código.

**El nombre de las funciones:**

* Para este modulo se usara el nombre de modulo **matricesSuma.py.**
* **Función –** crearMatriz(fila, columnas)
* **Función-** sumarMatrices(A,B)

**Compañero 2**

* **Tus funciones serán usadas por:**
  + El menú principal para mostrar resultados al usuario
  + El compañero 4 para resolver sistemas de ecuaciones
  + El compañero 3 para algunos cálculos de determinantes

**Coordina con el compañero 1:**

* Asegúrate que la estructura de matrices sea compatible
* Verifica cómo manejan los errores para ser consistentes

**Objetivo**

Crear todas las herramientas necesarias para que el programa pueda realizar *multiplicaciones con matrices*.

**1- Multiplicación por Escalar**

**Qué debes hacer:**

* + Crear una función que tome una matriz y un número para multiplicarlos
  + Multiplicar *cada elemento* de la matriz por ese número
  + Devolver la nueva matriz resultante

**Validaciones importantes:**

* + Asegurarte que el escalar sea un número válido (puede ser entero o decimal, positivo o negativo)
  + Verificar que la matriz no esté vacía

2-**Multiplicación entre Matrices**

* **Qué debes hacer:**
  + Crear una función que pueda multiplicar dos matrices (A × B)
  + Seguir las reglas matemáticas de multiplicación de matrices
  + Verificar que las dimensiones sean compatibles (el número de columnas de A debe ser igual al número de filas de B)
* **Manejo de errores:**
  + Si las matrices no son compatibles para multiplicación, mostrar un mensaje de error

**Operación Combinada (αA + βB)**

* **Qué debes hacer:**
  + Crear una función que combine las multiplicaciones por escalar con la suma de matrices
  + Primero multiplicar cada matriz por su escalar (α por A, β por B)
  + Luego sumar los resultados
* **IMPORTANTE:**
  + No interactuar directamente con el usuario:
  + Tus funciones solo reciben datos y devuelven resultados
  + El menú principal se encargará de mostrar mensajes al usuario
  + Comentar el código para entenderlo mejor explicando la función que cumple esa parte del código.
* **El nombre de las funciones:**
  + modulo matricesMultiplicacion.py

**Compañero 3**

**Tu trabajo es crucial para que el Compañero 4 pueda resolver sistemas de ecuaciones.**

***Qué debes hacer exactamente:***

* Crearás una función que reciba una matriz cuadrada.
* Implementarás dos métodos distintos:  
  *Para matrices 2×2:* Usarás la fórmula rápida  
  *Para matrices 3×3:* Aplicarás la Regla de Sarrus
* Validarás que:
  + La matriz sea cuadrada (2×2, 3×3, etc.)
  + Todos los elementos sean números
  + La matriz no esté vacía

**2. Cálculo de Matriz Inversa**

*Proceso detallado:*

1. Primero calcularás el determinante (usando tu propia función)
2. Si el determinante es cero → mostrar error "Matriz no invertible"
3. Para matrices 2×2 aplicarás:
   * Intercambiarás a y d
   * Cambiarás el signo de b y c
   * Dividirás cada elemento por el determinante

**Manejo de Errores**

*Casos que debes anticipar:*

1. Matriz rectangular (no cuadrada):
   * Error: "La matriz debe ser cuadrada para calcular su determinante"
2. Matriz con texto o valores no numéricos:
   * Error: "Todos los elementos deben ser valores numéricos"
3. Matriz singular (determinante cero):
   * Error: "Matriz no invertible - determinante es cero"

**Integración con el Proyecto**

*Cómo conectará tu código:*

* El Compañero 4 usará tu función de inversa para resolver sistemas de ecuaciones
* El menú principal mostrará tus resultados con formato bonito
* Todos usarán tu mismo sistema de reporte de errores
* **IMPORTANTE:**
  + Usa variables temporales para hacer tus cálculos más legibles
  + Considera usar round() para redondear decimales
  + Comentar el código para entenderlo mejor explicando la función que cumple esa parte del código.

**Compañero 4**

**Resolución de Sistemas de Ecuaciones Lineales (Ax = B)**

**Qué debes crear:**

* Una función resolverSistema(A, B) que:
  1. Reciba:
     + A: Matriz de coeficientes (cuadrada)
     + B: Vector de términos independientes
  2. **Usará funciones de los compañeros:**
     + La inversa de A (del Compañero 3)
     + Multiplicación de matrices (del Compañero 2)
  3. Devolverá:
     + Vector solución x **o**
     + Error si el sistema no tiene solución única

**Validaciones críticas:**  
✔️ Verificar que A sea cuadrada  
✔️ Comprobar que las dimensiones de A y B sean compatibles  
✔️ Manejar el caso cuando A no sea invertible (determinante = 0)

**Operaciones con Vectores**

**Funciones a implementar:**

1. **Suma de vectores:**
   * Validar que tengan la misma longitud
   * Retornar vector resultante
   * Ejemplo: [1, 2] + [3, 4] = [4, 6]
2. **Producto punto:**
   * Calcular la suma de productos de elementos correspondientes
   * Ejemplo: [1, 2] · [3, 4] = (1×3) + (2×4) = 11

**Manejo de errores:**  
❌ Vectores de diferentes longitudes → Error claro