ÁLGEBRA LINEAL MA0322

Especificaciones del Proyecto Final

Prof. Randy Wynta Banton

---

1. Entrega y formato

- **Fecha de entrega y presentación:** 28 de noviembre.
  (La fecha final de entrega y/o defensa podrá variar según disposición del docente).
- El proyecto es **grupal**, pero la defensa es **individual**.
- **Peso:** 65 % proyecto – 35 % defensa oral.
- **Incentivo por innovación:** hasta un **+7 %** adicional sobre la nota del proyecto.
- **Lenguaje:** libre elección.
- **Entrega:**
  1. Archivo ejecutable funcional.
  2. Código fuente completo.
  3. Manual de usuario breve.
  4. Instrucciones de ejecución (dependencias, compilación, etc.).
- **Formato sugerido del archivo:** `MA0322_Proyecto_<Grupo>_v1.zip`.
- Se evaluará tanto el **código** como la **interfaz** y el **manual**.

2. Restricciones generales

- **Prohibido** usar librerías o funciones que realicen directamente los métodos requeridos (determinantes, Gauss-Jordan, detección de triángulos, etc.).
- Se permiten únicamente funciones trigonométricas básicas (`sin`, `cos`, `tan`, `atan`) y operaciones aritméticas elementales.
- Todos los métodos deben ser implementados desde cero.
- Validaciones obligatorias para toda entrada de datos.
- **Fraude académico:** el uso de código externo no citado o copiado implica nota **0** y sanciones adicionales.
- La interfaz debe ser **intuitiva, creativa y funcional**.

3. Estructura general del proyecto

El proyecto debe ser un **único programa** compuesto por tres módulos:

Parte I – Triángulos (lados, ángulos y clasificación)

Objetivo: permitir al usuario definir tres puntos en $\mathbb{R}^2$ mediante clics, determinar si forman un triángulo y, en caso afirmativo, calcular sus lados, ángulos y clasificarlo.

Requisitos:

1) Interfaz gráfica que permita seleccionar tres puntos (clics) y mostrar sus coordenadas (x, y).
2) Detectar si los puntos son colineales:
   - Si **no** forman triángulo, justificar teóricamente (mostrar el cálculo del área nula o producto vectorial) y graficar la figura.
   - Si **sí** forman triángulo:
     - Calcular vectores, longitudes y ángulos mediante operaciones vectoriales.
     - Mostrar paso a paso todos los cálculos.
     - Clasificar el triángulo por lados (equilátero, isósceles, escaleno) y por ángulos (acutángulo, rectángulo, obtusángulo). Debe justificar la clasificación.
3) La representación gráfica debe mostrar la figura con sus puntos y medidas.

Criterios: detección correcta (20 %), cálculos y justificación (35 %), representación gráfica (15 %), validaciones (10 %), manual y claridad (20 %).

Parte II – Intersección de planos (Gauss–Jordan)

Objetivo: calcular la recta de intersección de dos planos mediante el método de Gauss–Jordan o, si no se intersectan, la distancia entre ellos.

Requisitos:

1) Dos campos de entrada para las ecuaciones de los planos en x, y, z (en cualquier orden).
2) Validar formato de entrada.
3) Si los planos se intersecan:
   - Aplicar Gauss–Jordan y mostrar la eliminación paso a paso.
   - Obtener la ecuación paramétrica de la recta de intersección.
4) Si no se intersecan:
   - Calcular la distancia entre ellos con la fórmula de punto a plano.
   - Justificar cada paso y mostrar el cálculo completo.
5) Identificar planos coincidentes cuando corresponda.

Criterios: interpretación correcta de ecuaciones (15 %), Gauss–Jordan (35 %), distancia/justificación (30 %), interfaz (10 %), manual (10 %).

Parte III – Determinantes de matrices 3×3 y 4×4

Objetivo: calcular determinantes mostrando el procedimiento paso a paso.

Requisitos:

1) Permitir elegir entre matriz 3×3 o 4×4.
2) Ingreso ordenado de los elementos.
3) Para 3×3, permitir elegir el método (Sarrus, cofactores, reducción, etc.).
4) Para 4×4:
   - Elegir si los subdeterminantes 3×3 se calculan por Sarrus o cofactores.
   - Permitir elegir la fila o columna de desarrollo.
5) Mostrar todo el procedimiento paso a paso.
6) Implementar los métodos desde cero (no usar funciones prehechas).

Criterios: implementación (40 %), detalle de procedimiento (30 %), interfaz (15 %), validaciones y documentación (15 %).

4. Validaciones obligatorias

- Detección de campos vacíos, caracteres no numéricos o formatos incorrectos.
- Tolerancias numéricas claras (definir en el manual, ej. $\varepsilon = 10^{-6}$).
- Manejo de casos degenerados (singulares, colineales, coincidentes).
- Control de errores (evitar cierres inesperados).

5. Manual y defensa

El **manual de usuario** debe incluir:

a) Formato esperado de entradas (con ejemplos).

b) Pasos de ejecución y ejemplos de uso.

c) Explicación breve de las funciones principales.

Durante la defensa cada integrante debe:

a) Demostrar el uso del programa (una prueba por parte).

b) Explicar la lógica de implementación y validaciones.

c) Responder preguntas técnicas.

6. Rúbrica general

- Implementación y resultados: 40 %
- Documentación y manual: 10 %
- Procedimientos matemáticos: 20 %
- Interfaz y usabilidad: 15 %
- Validaciones y manejo de errores: 10 %
- Innovación (bonus): 0–7 %

7. Consideraciones finales

- El proyecto debe estar unificado (una sola aplicación).
- Código legible, bien comentado y estructurado.
- Cualquier uso indebido de librerías conllevará nota 0.
- Se trabajará en los grupos oficialmente establecidos en clase.
- Consultas y aclaraciones deberán realizarse oportunamente.

"El rigor matemático y la claridad visual serán los pilares de este proyecto."

Prof. Randy Wynta Banton