

Título del Informe: MI HUMILDE DESPEDIDA DE LAS MATEMÁTICAS DE MI CARRERA

Autor: Josué Romero Jarava

Curso: Métodos Numéricos – NRC: 75397

Fecha de última entrega: 04/06/2025

1. Resumen

El presente informe describe el desarrollo de una aplicación educativa orientada a resolver problemas clásicos de métodos numéricos, con énfasis en sistemas de ecuaciones lineales, interpolación, derivación, integración y ecuaciones diferenciales ordinarias (EDO). La herramienta fue desarrollada en Python, con una arquitectura modular y funciones que integran algoritmos numéricos tradicionales y generación de explicaciones automáticas por medio de inteligencia artificial (IA). Está dirigida a estudiantes y docentes del área de ingeniería, brindando un entorno amigable para experimentar con datos personalizados, comprender conceptos abstractos y reforzar habilidades lógico-matemáticas. Esta solución no solo facilita el aprendizaje autónomo, sino que además democratiza el acceso a una enseñanza más interactiva y significativa. La exportación de resultados, el historial de sesiones y las explicaciones paso a paso hacen de esta aplicación una alternativa eficaz tanto para uso académico como personal.

2. Introducción

Las matemáticas aplicadas constituyen un eje fundamental en la formación de ingenieros, especialmente cuando se abordan problemas del mundo real que requieren soluciones precisas, eficientes y comprensibles. El presente proyecto surge como una respuesta creativa y funcional a la necesidad de integrar herramientas computacionales con el aprendizaje teórico-práctico de los métodos numéricos. La implementación de técnicas como la eliminación de Gauss, el método iterativo de Jacobi, la interpolación de Lagrange, la derivación por diferencias finitas, la integración por cuadratura y la resolución de EDO con `solve_ivp`, ofrece a los estudiantes un laboratorio digital donde pueden experimentar, equivocarse y aprender. Además, la integración de inteligencia artificial mediante la API de

DeepSeek permite que cada paso computacional sea acompañado de una breve explicación matemática, favoreciendo no solo la comprensión, sino la reflexión crítica sobre lo que ocurre en cada etapa del proceso. Esta propuesta pedagógica se alinea con las exigencias contemporáneas de la educación en ingeniería: precisión, claridad, automatización y comprensión conceptual.

3. Requerimientos del sistema

Lenguaje de programación: Python 3

Librerías: numpy, scipy, matplotlib, requests, fpdf y openai

Plataforma: Programa de entrada y salida estándar

Requisitos: Python, Pip e Internet

Configuraciones especiales: El programa necesita de una API_KEY distinta porque es de uso limitado ya que fue generada desde openrouter.ai

4. Descripción de funcionalidades

4.1. Sistemas de Ecuaciones Lineales

- Método de Gauss: Transformación escalonada para resolución directa del sistema $Ax = b$.
- Método de Jacobi: Método iterativo con verificación de convergencia.
- Entrada de datos dinámica: matrices, vectores, tolerancia y número de iteraciones.
- Visualización paso a paso y explicación matemática generada por IA.

4.2. Derivación e Integración Numérica

- Cálculo de derivadas por diferencia central.
- Integración por método de cuadratura usando SciPy.
- Funciones introducidas como expresiones matemáticas.

4.3. Ecuaciones Diferenciales Ordinarias (EDO)

- Solución de EDO de primer orden mediante solve_ivp.
- Condiciones iniciales y funciones dinámicas ingresadas por el usuario.

- Exportación de soluciones y trayectorias.

4.4. Interpolación

Interpolación de Lagrange basada en puntos (x, y) .

Cálculo del polinomio resultante y evaluación visual.

5. Diseño e implementación

5.1. Arquitectura del sistema

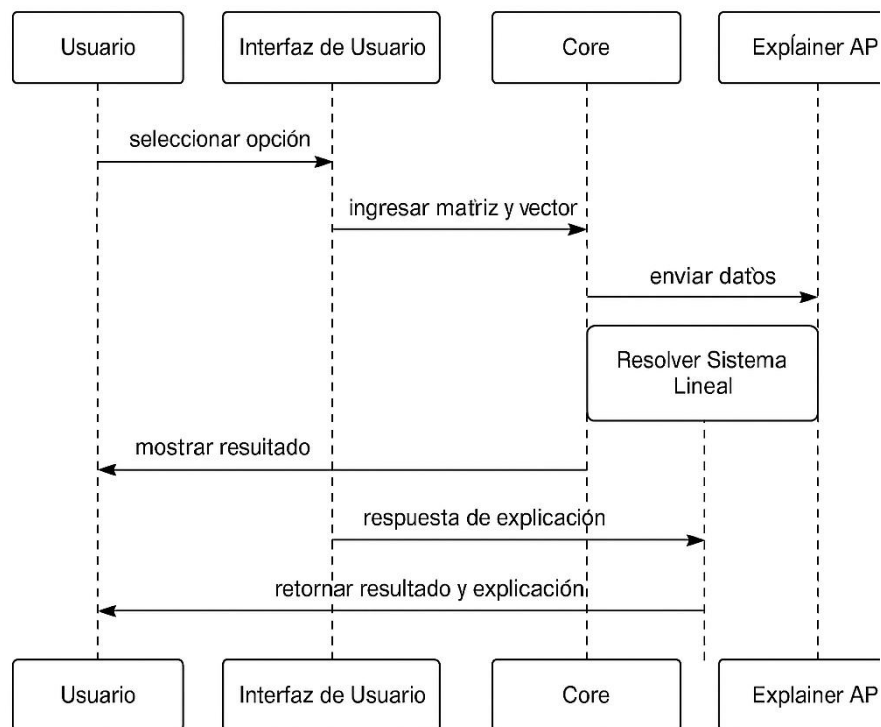
Modularización por carpetas: core, utils, database, interface

Interfaz basada en consola (CLI)

Explicación de IA gestionada con llamada a API externa

5.2. Diagramas del sistema

Diagrama de actividades: Representa el flujo desde la selección del método hasta la exportación.



gam: Sequence diagram

Diagrama de casos de uso: Incluye al usuario y las acciones permitidas como: ingresar datos, resolver, ver historial, exportar.

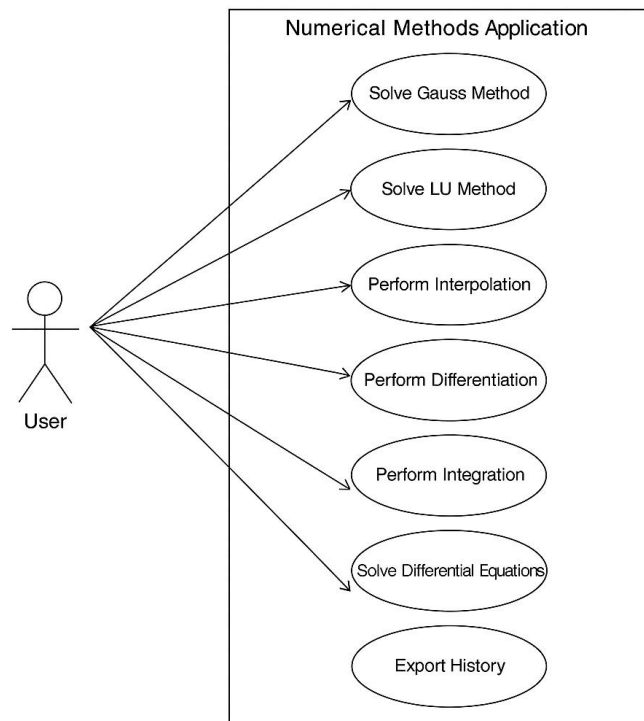


Diagrama de secuencia: Muestra la secuencia de llamadas entre módulos (cli → core → utils/db).

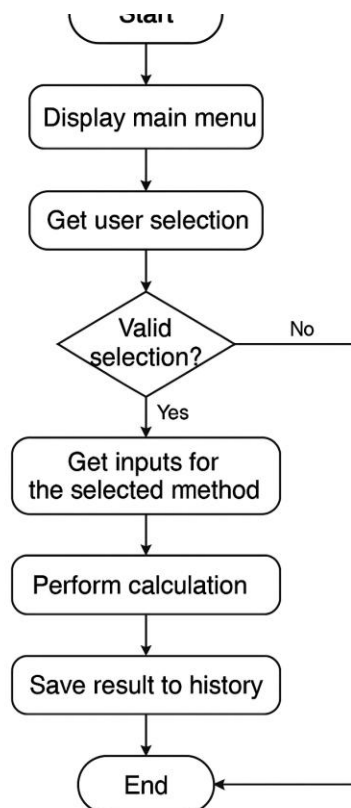
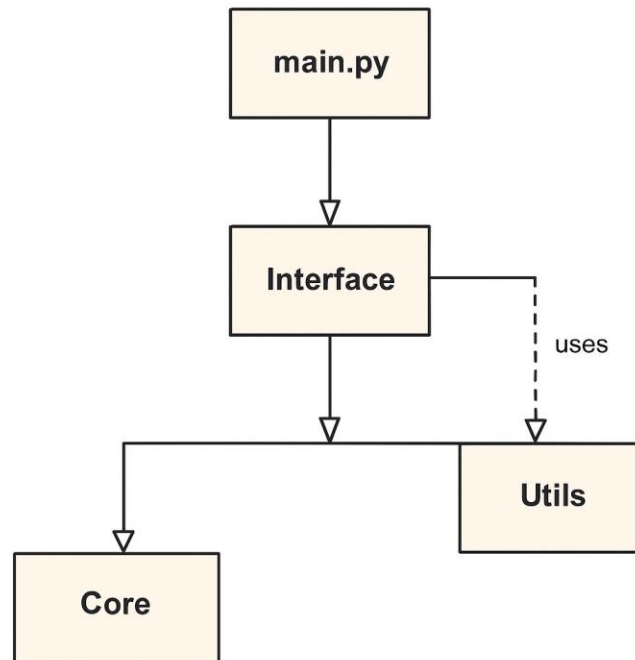


Diagrama de componentes: Representa la arquitectura física: CLI + Lógica + BD + PDF/Exportación + API IA.



5.3. Flujo del sistema

Menú principal muestra opciones (Gauss, Jacobi, etc.)

Usuario ingresa los datos por consola.

Módulo core procesa datos y ejecuta método.

Explicación es generada con IA.

Resultado es mostrado y guardado en base de datos.

Usuario puede exportar el historial a PDF.

6. Exportación e historial

Historial de sesiones almacenado en SQLite.

Cada entrada guarda: método, entrada, resultado, explicación y timestamp.

Exportación a PDF en carpeta /docs/exporte_metodo.pdf con formato claro y profesional.

7. Retos enfrentados y soluciones

Entrada dinámica segura: Se resolvió usando `ast.literal_eval` y validación por tipos.

Llamadas a IA seguras: Se implementó manejo de errores por si la API no responde.

Modularidad: Se dividió en archivos funcionales para facilitar mantenimiento y escalabilidad.

Exportación: Se usó `fpdf` para crear informes PDF legibles directamente desde los datos de SQLite.

8. Conclusiones

Este proyecto representa mucho más que una práctica de programación: es un homenaje reflexivo a la relación entre las matemáticas, la computación y el pensamiento crítico. Lo que inicialmente fue una asignatura desafiante se convirtió en una oportunidad para integrar el conocimiento técnico con la creatividad digital. La experiencia no solo me permitió reafirmar la importancia de la precisión algorítmica, sino que también me enseñó a transformar complejidad en claridad a través del diseño de software. Al decir adiós a las matemáticas formales de mi carrera, me llevo no solo fórmulas, sino habilidades duraderas: resolver problemas, estructurar ideas, automatizar procesos y comunicar resultados de forma efectiva. Esta despedida no es un final, sino un punto de partida para aplicar el pensamiento matemático a nuevas áreas de la ingeniería y la vida profesional.

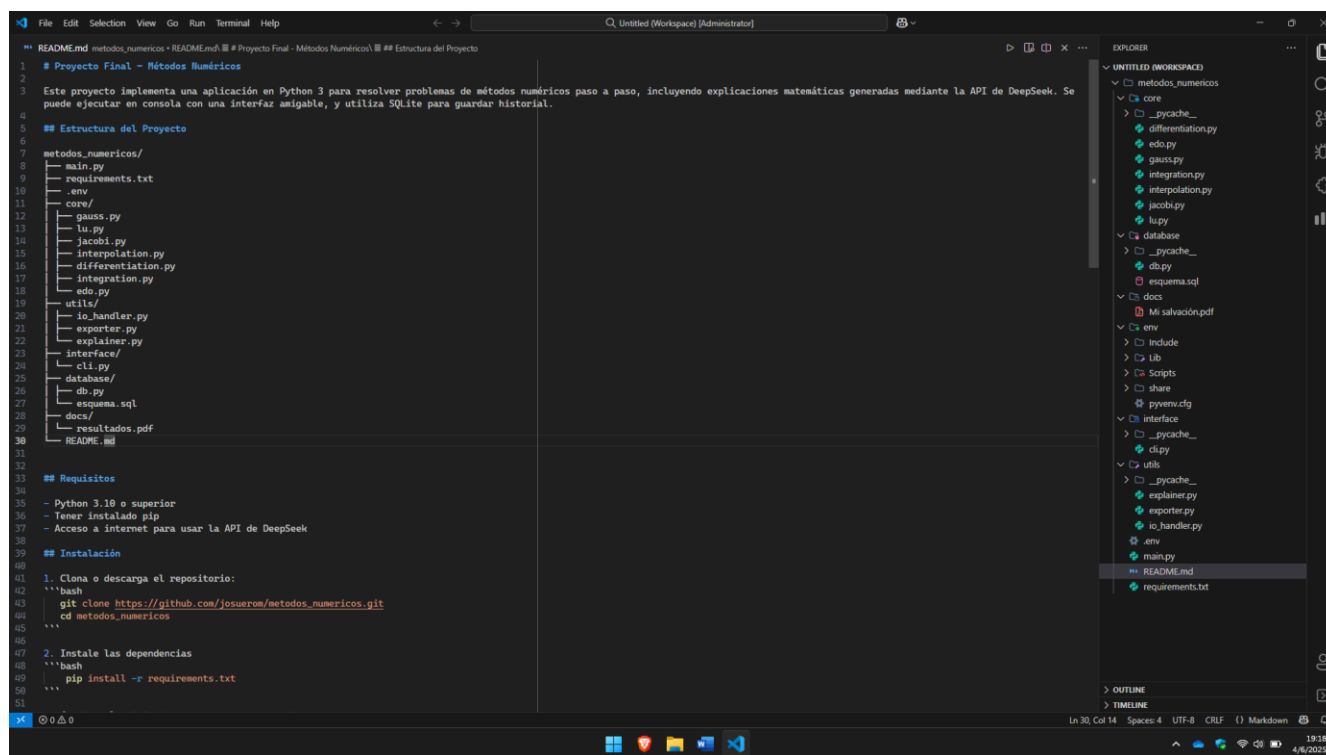
El desarrollo de esta aplicación representa una fusión efectiva entre el pensamiento matemático riguroso y las habilidades prácticas de ingeniería de software. Al implementar métodos numéricos fundamentales como Gauss, LU, Jacobi, interpolación, integración y resolución de EDOs, se consolidó no solo el conocimiento teórico, sino también la capacidad de traducir algoritmos matemáticos en soluciones computacionales funcionales y reutilizables. La incorporación de inteligencia artificial para la explicación paso a paso constituye un hito pedagógico que democratiza el aprendizaje, permitiendo que incluso los conceptos más abstractos se tornen comprensibles mediante lenguaje natural. Este proyecto final no solo demuestra dominio técnico, sino también criterio en el diseño modular, manejo de errores, validación de entradas y generación de reportes útiles. En esencia, este trabajo es un testimonio del rol que juega la programación en la transformación del aprendizaje

tradicional en experiencias más autónomas, accesibles y significativas para las futuras generaciones de ingenieros.

9. Referencias

- [1] Chapra, S. C., & Canale, R. P. (2015). Métodos numéricos para ingenieros (7.^a ed.). McGraw-Hill Education.
- [2] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van Mulbregt, P. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- [3] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- [4] OpenRouter. (2025). API Gateway for LLMs. <https://openrouter.ai/>
- [5] The Python Software Foundation. (2024). Python Language Reference Manual (version 3.12). <https://docs.python.org/3/>

10. Anexos



Captura 1. Muestra completa de toda la estructura del proyecto

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

C:\Workspace\metodos_numericos>pip install numpy scipy matplotlib requests fpdf openai
Collecting numpy
  Using cached numpy-2.2.6-cp313-cp313-win_amd64.whl.metadata (60 kB)
Collecting scipy
  Using cached scipy-1.15.3-cp313-cp313-win_amd64.whl.metadata (60 kB)
Collecting matplotlib
  Using cached matplotlib-3.10.3-cp313-cp313-win_amd64.whl.metadata (11 kB)
Collecting requests
  Using cached requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting fpdf
  Using cached fpdf-1.7.2-py2.py3-none-any.whl
Collecting openai
  Downloading openai-1.84.0-py3-none-any.whl.metadata (25 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Using cached contourpy-1.3.2-cp313-cp313-win_amd64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Using cached fonttools-4.58.1-cp313-cp313-win_amd64.whl.metadata (108 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Using cached kiwisolver-1.4.8-cp313-cp313-win_amd64.whl.metadata (6.3 kB)
Collecting packaging>=20.0 (from matplotlib)
  Using cached packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
x 0 0 0
```

Captura 2. Instalación de dependencias para la correcta ejecución del programa

```
File Edit Selection View Go Run Terminal Help

esquema.sql metodos_numericos • database\esquema.sql

1 CREATE TABLE IF NOT EXISTS historial (
2     id INTEGER PRIMARY KEY AUTOINCREMENT,
3     metodo TEXT,
4     entrada TEXT,
5     resultado TEXT,
6     explicacion TEXT,
7     timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
8 );
```

Captura 3. Esquema de la base de datos


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Versión 10.0.26100.4202]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Workspace\metodos_numericos>python main.py
==== MATRIX CALC MÉTODOS NUMÉRICOS ====
1. Gauss
2. Lu
3. Jacobi
4. Interpolación
5. Derivación
6. Integración
7. EDO
8. Exportar historial
0. Salir
>> 1
Matriz. Ej: [[2,1],[5,7]]: [[2,1],[5,7]]
Términos. Ej: 11 13: 11 13
[
Asistente matemático: Explica matemáticamente el paso a paso completo de la eliminación Gaussiana para: (array([[2. , 1. ],
[0. , 4.5]]), array([ 11. , -14.5]), 0, 1, np.float64(2.5))
]
Resultado: [ 7.11111111 -3.22222222]

¿Desea continuar probando este programa? s
==== MATRIX CALC MÉTODOS NUMÉRICOS ====
1. Gauss
2. Lu
3. Jacobi
4. Interpolación
5. Derivación
6. Integración
7. EDO
8. Exportar historial
0. Salir
>>
```

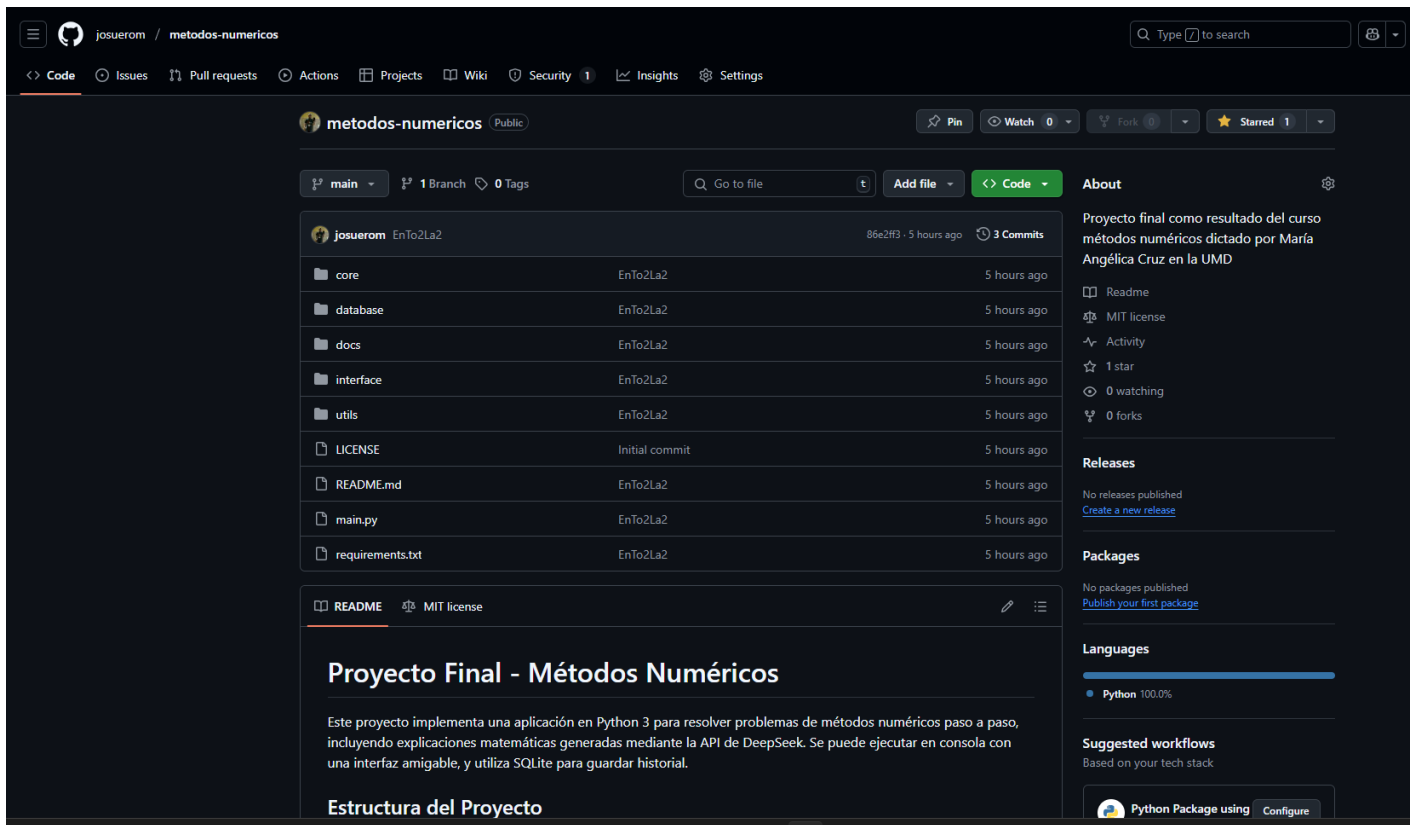
Captura 4. Muestra de la primera fase frente al usuario final

```
differentiation.py core
1 from utils.explainer import explain_step
2 from database.db import save_result
3
4 def numerical_derivative(f_expr, x, h=1e-5):
5     try:
6         f = eval(f"lambda x: {f_expr}")
7     except Exception as e:
8         raise ValueError(f"Error al procesar la función: {e}")
9
10    df = (f(x + h) - f(x - h)) / (2 * h)
11    explanation = explain_step(x, h, df, 'derivative')
12    save_result("Derivación", f"f(x)={f_expr}, x={x}", df, explanation)
13    return df, explanation

integration.py metodos_numericos * core/integration.py
1 from scipy.integrate import quad
2 from utils.explainer import explain_step
3 from database.db import save_result
4
5 def numerical_integration(f_expr, a, b):
6     try:
7         f = eval(f"lambda x: {f_expr}")
8     except Exception as e:
9         raise ValueError(f"Error al procesar la función: {e}")
10
11    result, _ = quad(f, a, b)
12    explanation = explain_step(a, b, result, 'integration')
13    save_result("Integración", f"f(x)={f_expr}, a={a}, b={b}", result, explanation)
14    return result, explanation

gauss.py core
1 import numpy as np
2 from utils.explainer import explain_step
3 from database.db import save_result
4
5 def gauss_elimination(A, b):
6     n = len(b)
7     steps = []
8     A = A.astype(float)
9     b = b.astype(float)
10
11    for i in range(n):
12        if A[i][i] == 0.0:
13            raise ValueError("Pivot cero encontrado.")
14        for j in range(i+1, n):
15            ratio = A[j][i]/A[i][i]
16            A[j, :] -= ratio * A[i, :]
17            b[j] -= ratio * b[i]
18            steps.append(explain_step(A, b, i, j, ratio))
19
20    x = np.zeros(n)
21    x[-1] = b[-1]/A[-1][-1]
22
23    for i in range(n-2, -1, -1):
24        x[i] = (b[i] - np.dot(A[i, i+1:], x[i+1:]))/A[i][i]
```

Captura 5. Código fuente de los principales módulos



Captura 6. Publicación de repositorio y documentación del proyecto para mayores detalles

Enlace: <https://github.com/josuerom/metodos-numericos>

“A mi amado Padre Celestial le dedico especialmente este proyecto, aunque use IA uso también de él.”

Que Dios la bendiga profesora María Angélica Cruz Guerrero, no vaya a olvidar al costeño que varias veces le dirigió la palabra.