

**UNIVERSIDAD NACIONAL DEL ALTIPLANO
PUNO**

**FACULTAD DE INGENIERÍA ESTADÍSTICA E
INFORMÁTICA**

**ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA
E INFORMÁTICA**



Método Gauss-Jordan con Tkinter
CURSO: MÉTODOS OPTIMIZACIÓN

DOCENTE:
ING. Fred Torres Cruz

PRESENTADO POR:
Edilfonso Muñoz Ancori

SEMESTRE: V NIV

PUNO-PERÚ
2025

Transformación de Ejemplos a Matrices Aumentadas

1. Modelo de Regresión Lineal

Sistema de ecuaciones:

$$\begin{aligned}2w_1 + 3w_2 - w_3 &= 5, \\ -w_1 + 2w_2 + 4w_3 &= 6, \\ 3w_1 - w_2 + 2w_3 &= 7.\end{aligned}$$

Matriz aumentada:

$$\left[\begin{array}{ccc|c} 2 & 3 & -1 & 5 \\ -1 & 2 & 4 & 6 \\ 3 & -1 & 2 & 7 \end{array} \right]$$

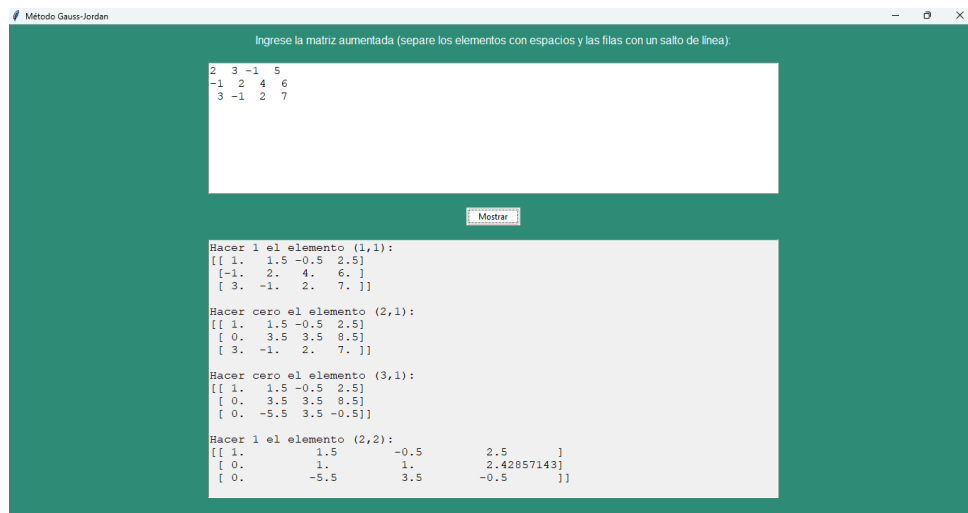


Figura 1: Solucion de la ecuacion en Python con Tkinter que implementa el método de Gauss-Jordan

2. Calibración de Hiperparámetros

Sistema de ecuaciones:

$$\begin{aligned}x + 2y + 3z &= 12, \\ 2x - y + z &= 4, \\ -x + 2y - 2z &= 0.\end{aligned}$$

Matriz aumentada:

$$\begin{bmatrix} 1 & 2 & 3 & 12 \\ 2 & -1 & 1 & 4 \\ -1 & 2 & -2 & 0 \end{bmatrix}$$

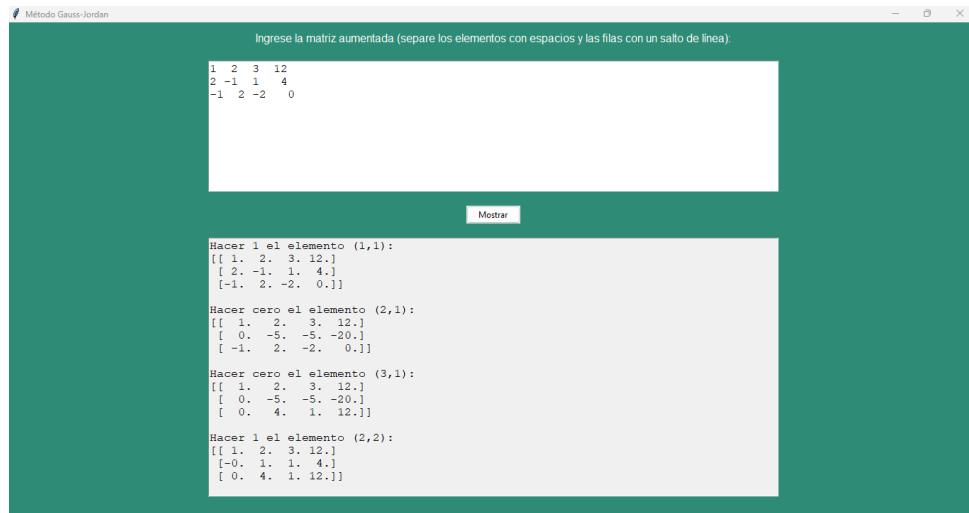


Figura 2: Solucion de la ecuacion en Python con Tkinter que implementa el método de Gauss-Jordan

3. Asignación Óptima de Recursos

Sistema de ecuaciones:

$$\begin{aligned} a + b + c &= 6, \\ 2a - b + 3c &= 13, \\ -a + 2b - c &= 2. \end{aligned}$$

Matriz aumentada:

$$\begin{bmatrix} 1 & 1 & 1 & 6 \\ 2 & -1 & 3 & 13 \\ -1 & 2 & -1 & 2 \end{bmatrix}$$

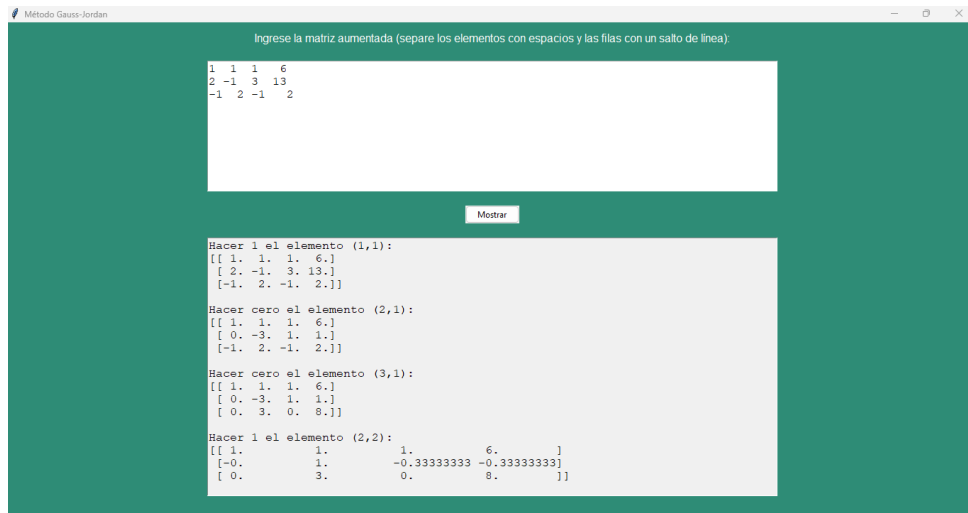


Figura 3: Solucion de la ecuacion en Python con Tkinter que implementa el método de Gauss-Jordan

4. Optimización de Parámetros de un Bosque Aleatorio

Sistema de ecuaciones:

$$p + 2q + 3r = 10,$$

$$2p - q + 4r = 12,$$

$$3p + 3q - r = 6.$$

Matriz aumentada:

$$\begin{bmatrix} 1 & 2 & 3 & 10 \\ 2 & -1 & 4 & 12 \\ 3 & 3 & -1 & 6 \end{bmatrix}$$

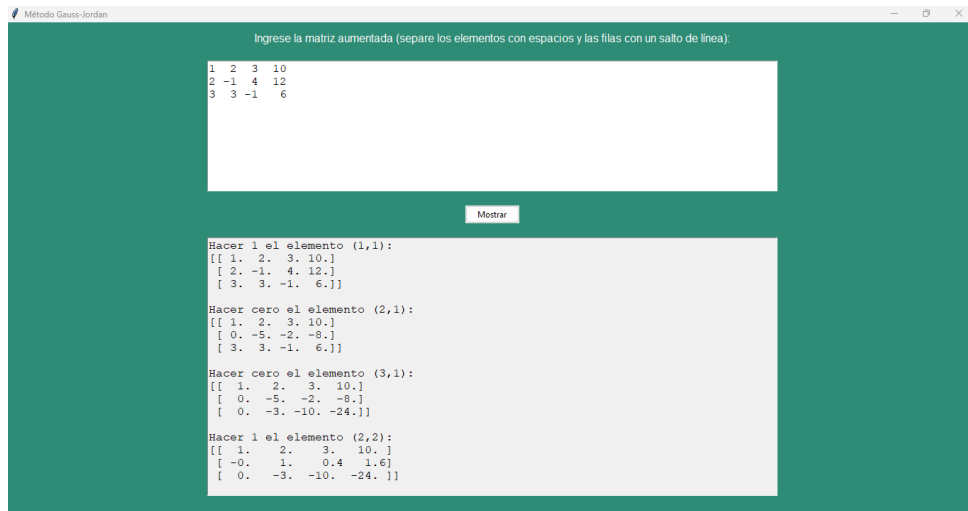


Figura 4: Solucion de la ecuacion en Python con Tkinter que implementa el método de Gauss-Jordan

5. Estimación de Demanda de Inventario

Sistema de ecuaciones:

$$\begin{aligned} u + v + 2w &= 9, \\ 2u - 3v + 4w &= 5, \\ u - 2v + w &= 1. \end{aligned}$$

Matriz aumentada:

$$\begin{bmatrix} 1 & 1 & 2 & 9 \\ 2 & -3 & 4 & 5 \\ 1 & -2 & 1 & 1 \end{bmatrix}$$

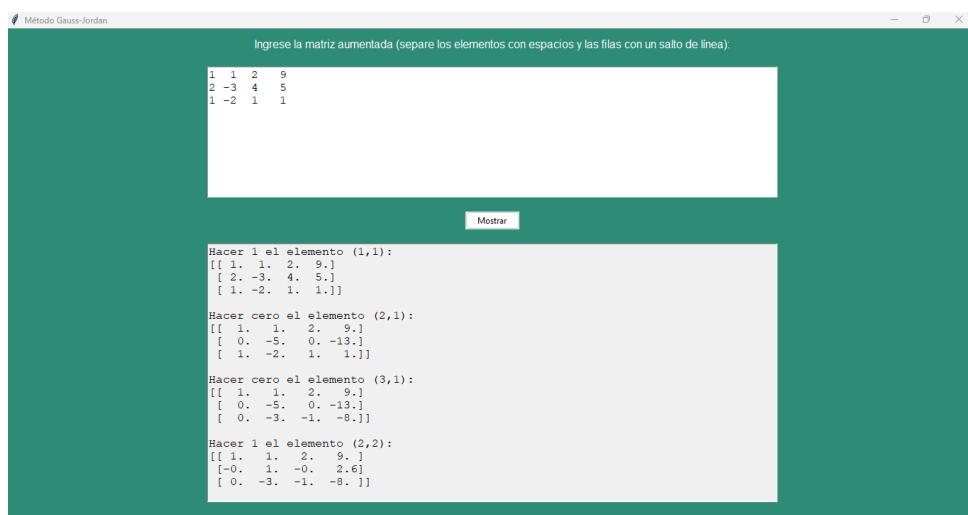


Figura 5: Solucion de la ecuacion en Python con Tkinter que implementa el método de Gauss-Jordan

Código Python para el Método Gauss-Jordan con Tkinter

El siguiente código implementa una interfaz gráfica en Python utilizando Tkinter para resolver matrices aumentadas con el método de Gauss-Jordan:

Listing 1: Código Python con Tkinter

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 import numpy as np
4
5 def gauss_jordan():
6     try:
7         # Leer la matriz desde la entrada
8         matriz = input_text.get("1.0", tk.END).strip()
9         matriz = [list(map(float, row.split())) for row in matriz.
10                    splitlines()]
11         matriz = np.array(matriz)
12
13         if matriz.shape[0] + 1 != matriz.shape[1]:
14             messagebox.showerror("Error", "La matriz debe ser una matriz
15                                     aumentada con n filas y n+1 columnas.")
16             return
17
18         pasos = []
19         n = len(matriz)
20
21         for i in range(n):
22             # Hacer 1 el elemento diagonal
23             diag_element = matriz[i, i]
24             if diag_element == 0:
25                 messagebox.showerror("Error", "No se puede resolver:
26                                         pivote cero encontrado.")
27                 return
28             matriz[i] = matriz[i] / diag_element
29             pasos.append(f"Hacer 1 el elemento ({i+1},{i+1}):\n{matriz}"
30                         )
31
32             # Hacer 0 los elementos por encima y por debajo del pivote
33             for j in range(n):
34                 if i != j:
35                     factor = matriz[j, i]
36                     matriz[j] = matriz[j] - factor * matriz[i]
37                     pasos.append(f"Hacer cero el elemento ({j+1},{i+1})
38                                 :\n{matriz}")
39
40             soluciones = matriz[:, -1]
41             pasos.append(f"Soluciones: {soluciones}")
42
43             output_text.delete("1.0", tk.END)
44             for paso in pasos:
45                 output_text.insert(tk.END, paso + "\n\n")
46     except Exception as e:
47         messagebox.showerror("Error", f"Ocurri un error: {e}")
48
49 # Crear ventana principal
50 root = tk.Tk()
```

```

46 root.title("M todo Gauss-Jordan")
47 root.geometry("800x600")
48
49 # Estilo de colores
50 root.configure(bg="#2e8c76")
51
52 # Etiqueta de instrucciones
53 instructions = tk.Label(root, text="Ingrese la matriz aumentada (separe
    los elementos con espacios y las filas con un salto de l nea):", bg=
    "#2e8c76", fg="white", font=("Arial", 12))
54 instructions.pack(pady=10)
55
56 # Cuadro de texto para ingresar la matriz
57 input_text = tk.Text(root, height=10, width=80, font=("Courier New", 12)
    )
58 input_text.pack(pady=10)
59
60 # Bot n para ejecutar el m todo
61 calculate_button = ttk.Button(root, text="Mostrar", command=gauss_jordan
    )
62 calculate_button.pack(pady=10)
63
64 # Cuadro de texto para mostrar resultados
65 output_text = tk.Text(root, height=20, width=80, font=("Courier New",
    12), state="normal", bg="#f0f0f0")
66 output_text.pack(pady=10)
67
68 # Iniciar la aplicaci n
69 root.mainloop()

```

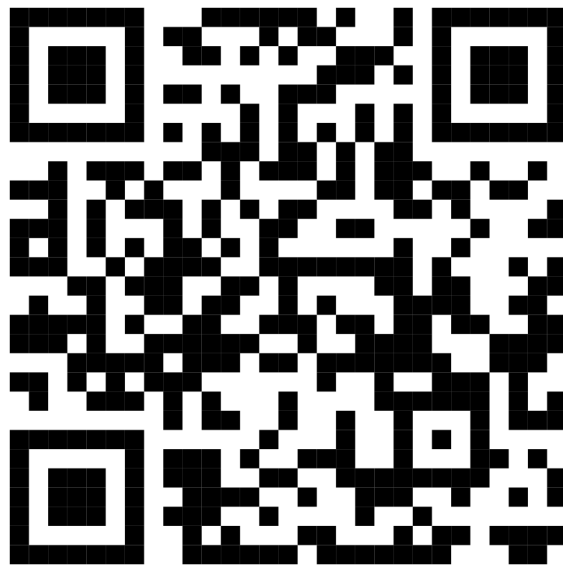


Figura 6: Codigo QR video de YuTube