UNIVERSIDAD AUTÓNOMA DE CHIAPAS.

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN, CAMPUS I.

LICENCIATURA EN INGENIERÍA EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE.

OCTAVO SEMESTRE, GRUPO: “M”

MATERIA: GRAFICACION.

DOCENTE: MTRO. SANDOVAL ZUÑIGA LUIS MANUEL.

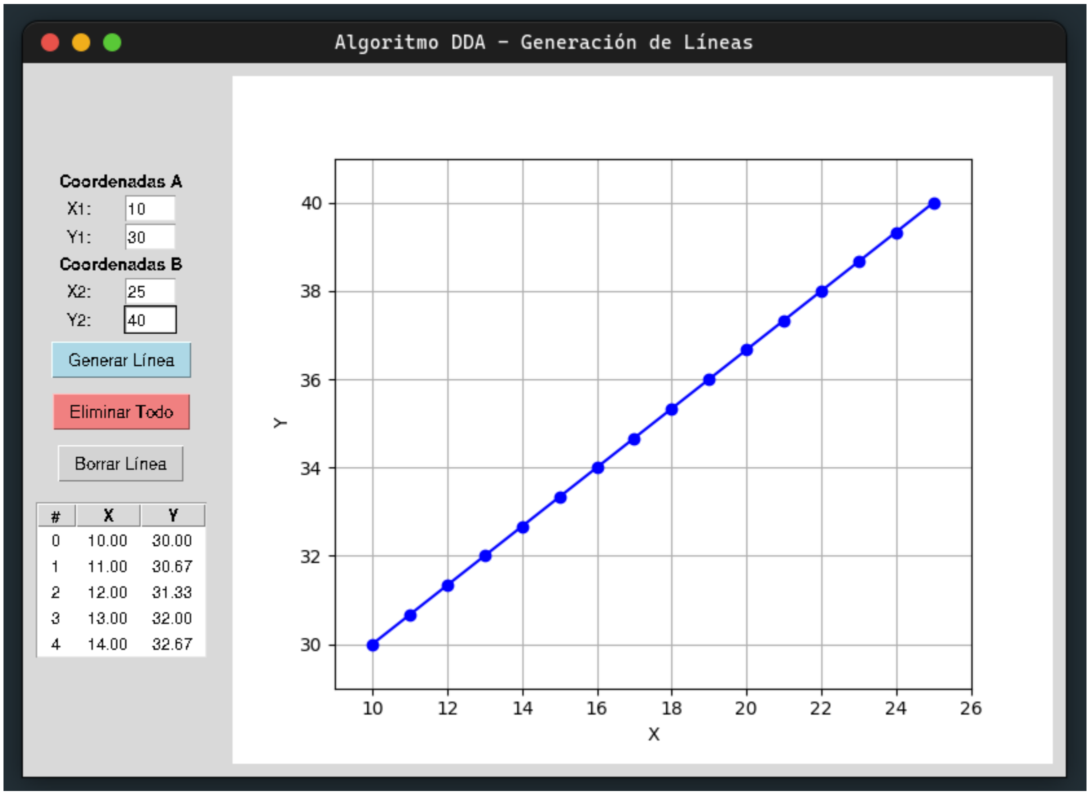
ALUMNOS:

- CARLOS DANIEL AMORES HERNANDEZ – A210367
- CRISTOBAL DE JESUS CORONEL CHAMBE – A210016
- JESUS ADRIAN CRUZ LEON – A210395

“1er. DOCUMENTO PROGRAMA DE LA LINEA DDA”

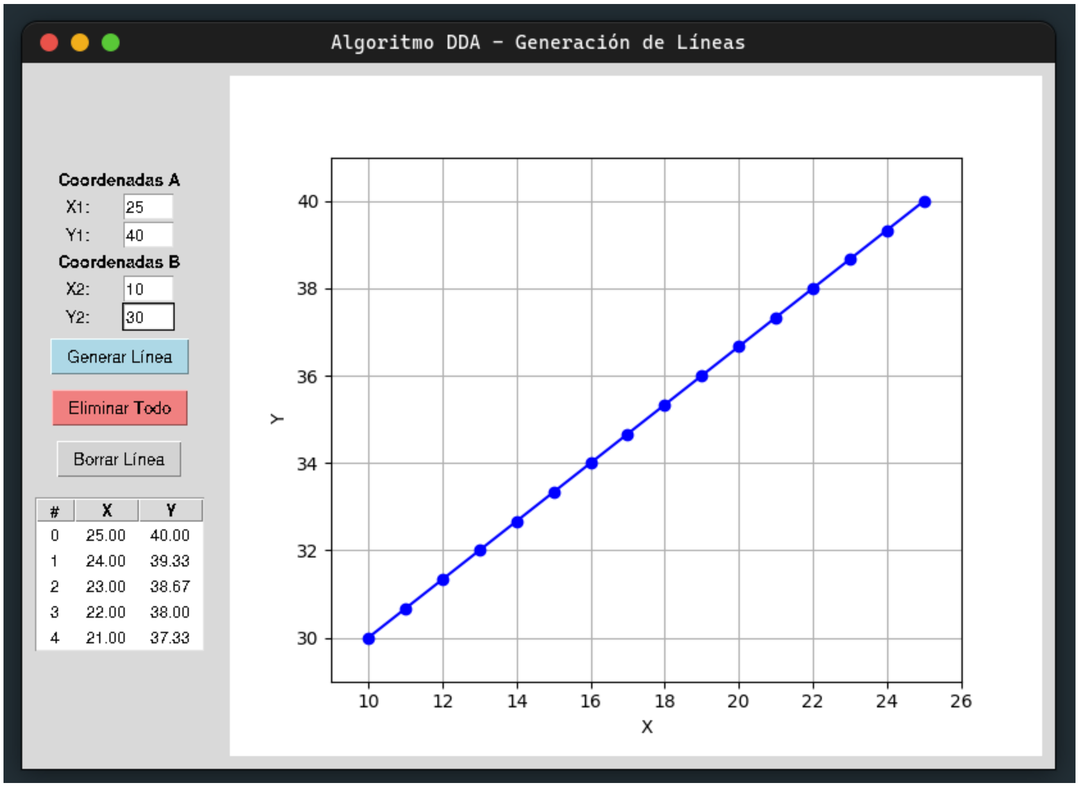
FECHA DE ENTREGA: 20 DE FEBRERO DEL 2025.

CAPTURA DE PANTALLA DEL PROGRAMA



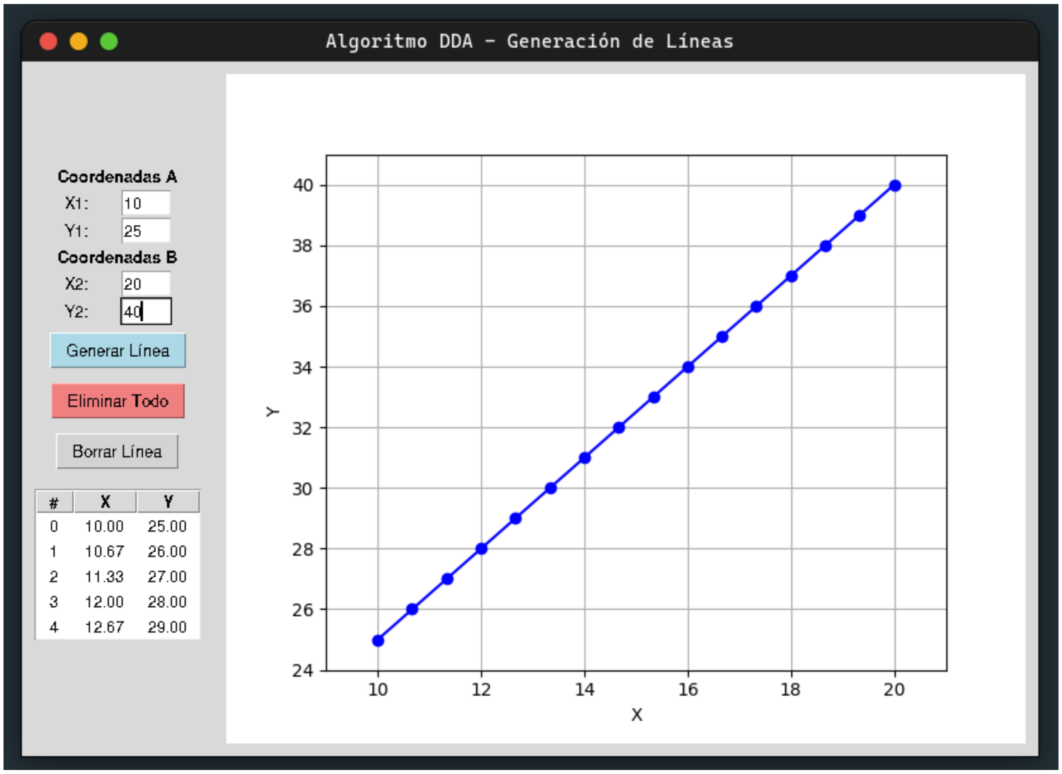
Coordenadas A
X1: 10
Y1: 30
Coordenadas B
X2: 25
Y2: 40

#	X	Y
0	10.00	30.00
1	11.00	30.67
2	12.00	31.33
3	13.00	32.00
4	14.00	32.67



Coordenadas A
X1: 25
Y1: 40
Coordenadas B
X2: 10
Y2: 30

#	X	Y
0	25.00	40.00
1	24.00	39.33
2	23.00	38.67
3	22.00	38.00
4	21.00	37.33



Coordenadas A

X1: 10

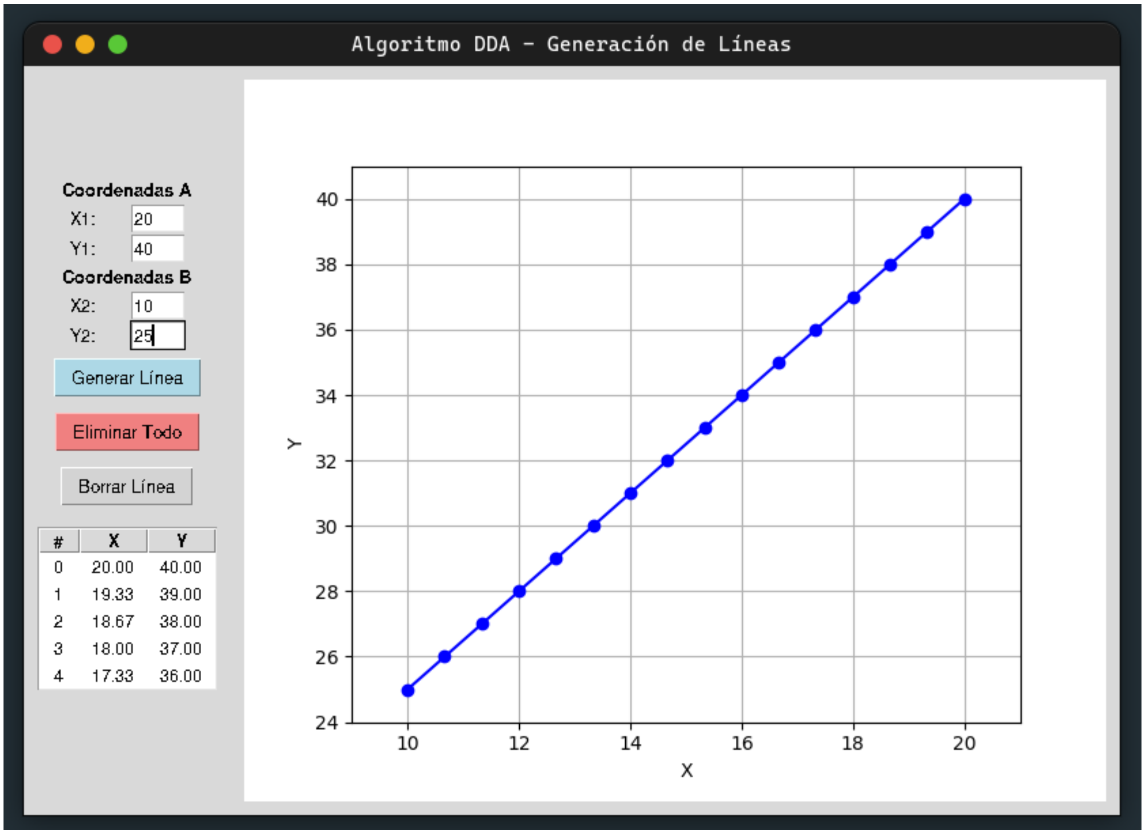
Y1: 25

Coordenadas B

X2: 20

Y2: 40

#	X	Y
0	10.00	25.00
1	10.67	26.00
2	11.33	27.00
3	12.00	28.00
4	12.67	29.00



Coordenadas A

X1: 20

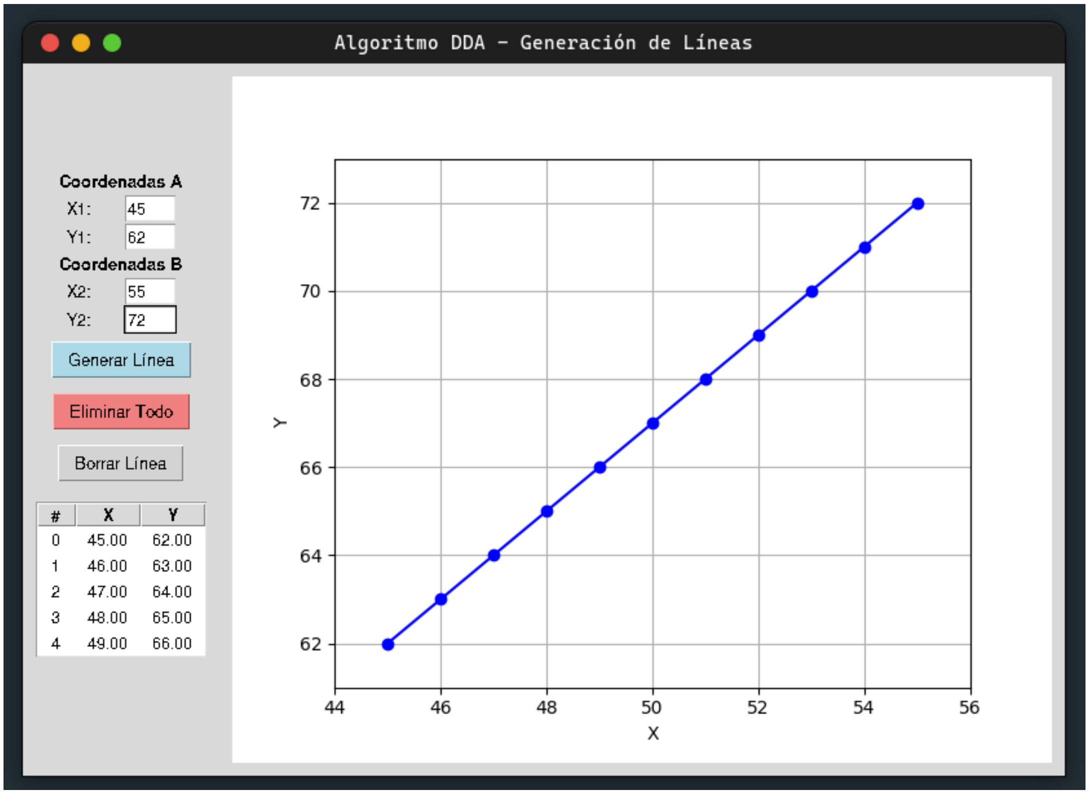
Y1: 40

Coordenadas B

X2: 10

Y2: 25

#	X	Y
0	20.00	40.00
1	19.33	39.00
2	18.67	38.00
3	18.00	37.00
4	17.33	36.00



Coordenadas A

X1: 45

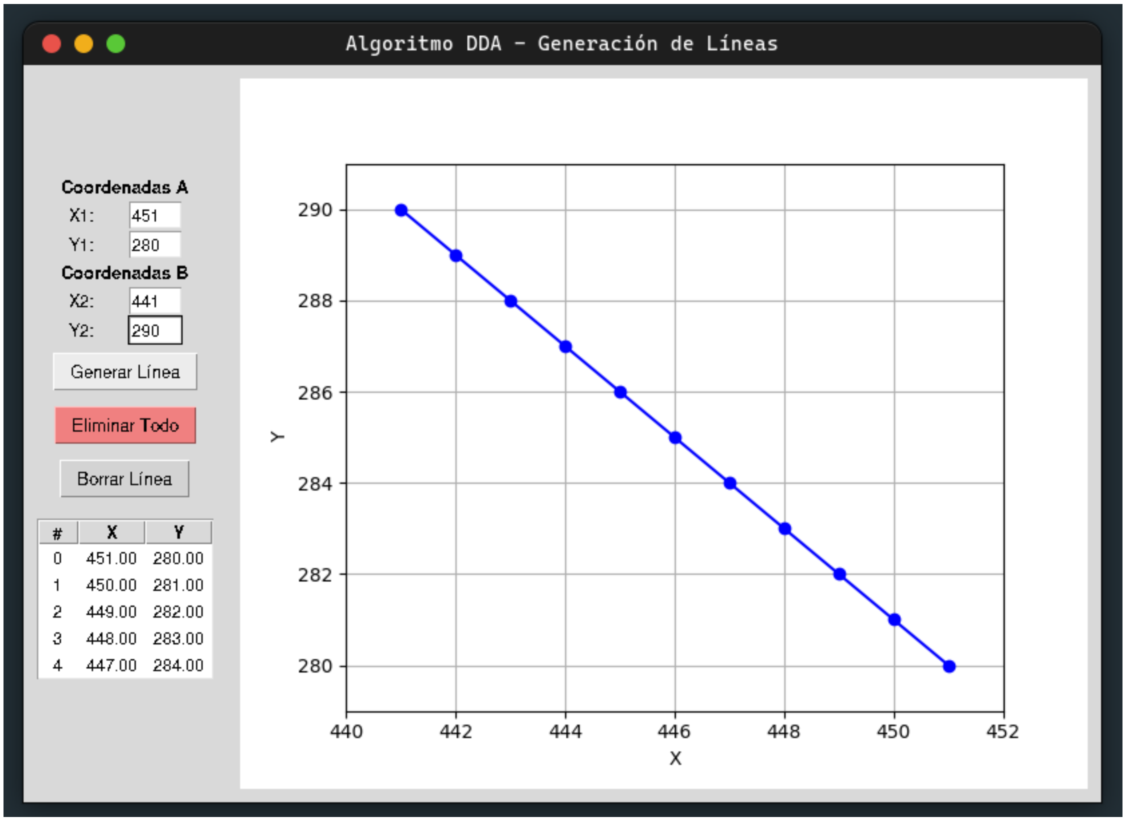
Y1: 62

Coordenadas B

X2: 55

Y2: 72

#	X	Y
0	45.00	62.00
1	46.00	63.00
2	47.00	64.00
3	48.00	65.00
4	49.00	66.00



Coordenadas A

X1: 451

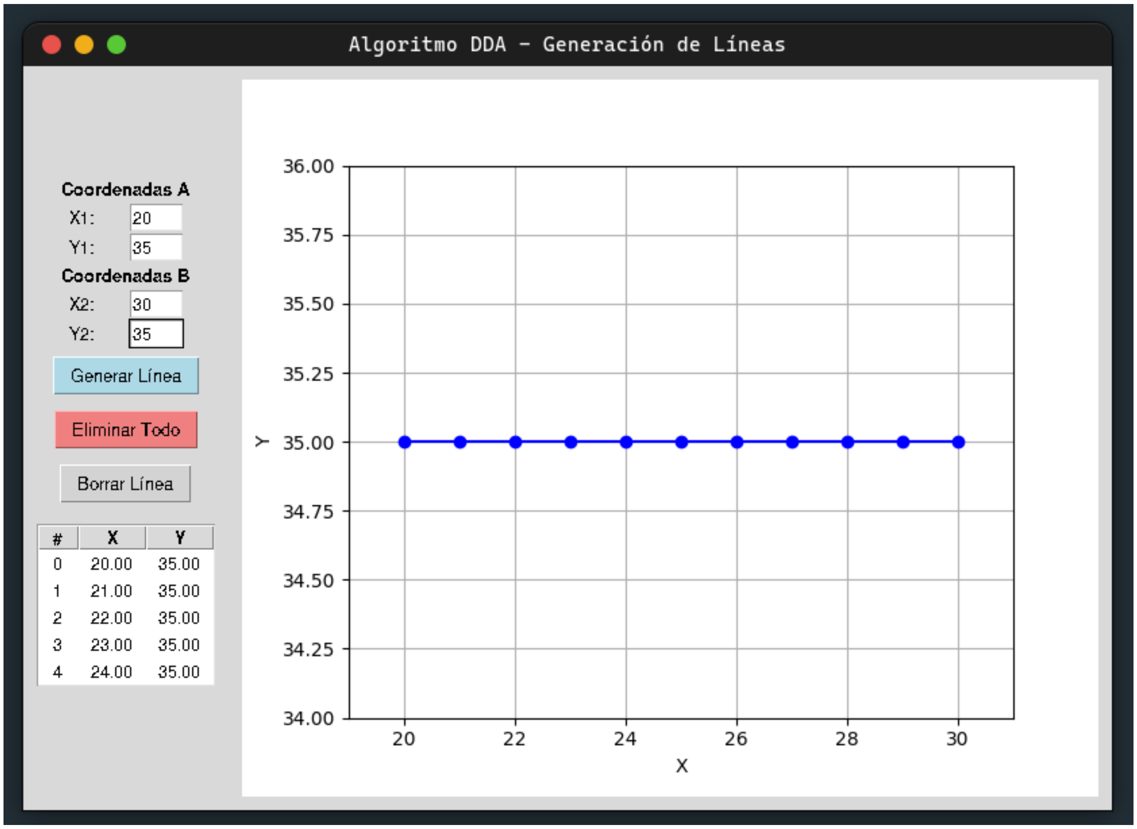
Y1: 280

Coordenadas B

X2: 441

Y2: 290

#	X	Y
0	451.00	280.00
1	450.00	281.00
2	449.00	282.00
3	448.00	283.00
4	447.00	284.00



Coordenadas A

X1: 20

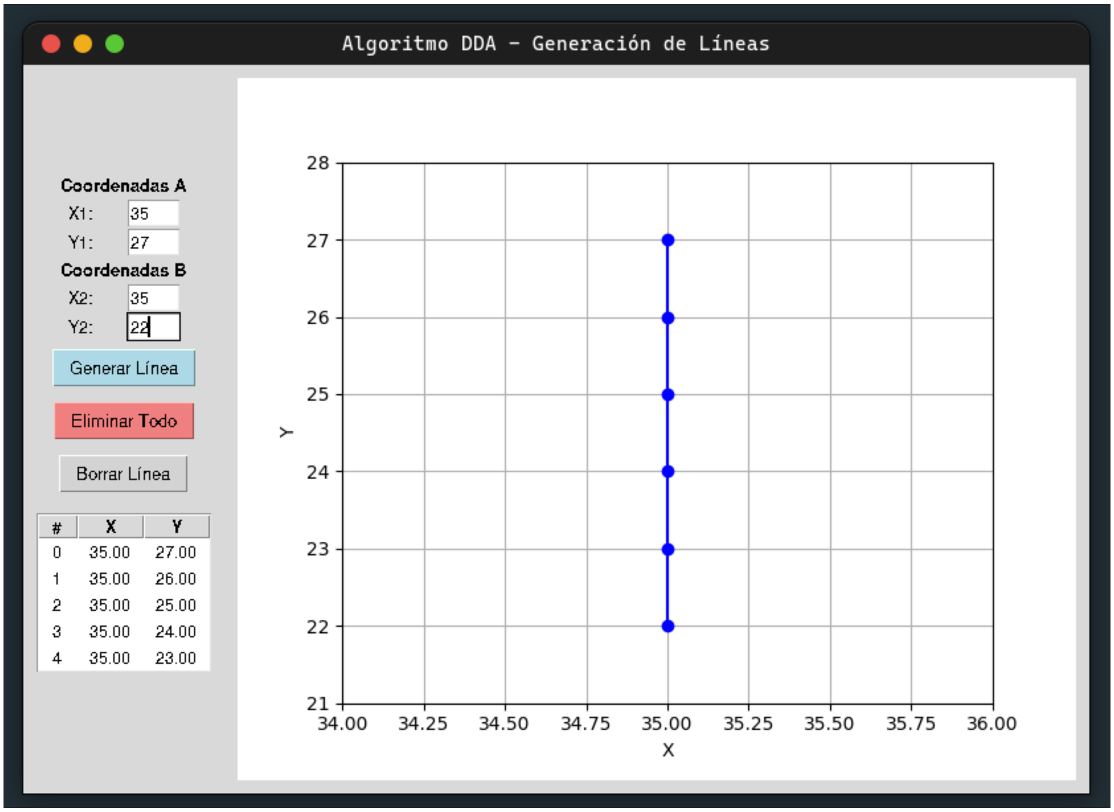
Y1: 35

Coordenadas B

X2: 30

Y2: 35

#	X	Y
0	20.00	35.00
1	21.00	35.00
2	22.00	35.00
3	23.00	35.00
4	24.00	35.00



Coordenadas A

X1: 35

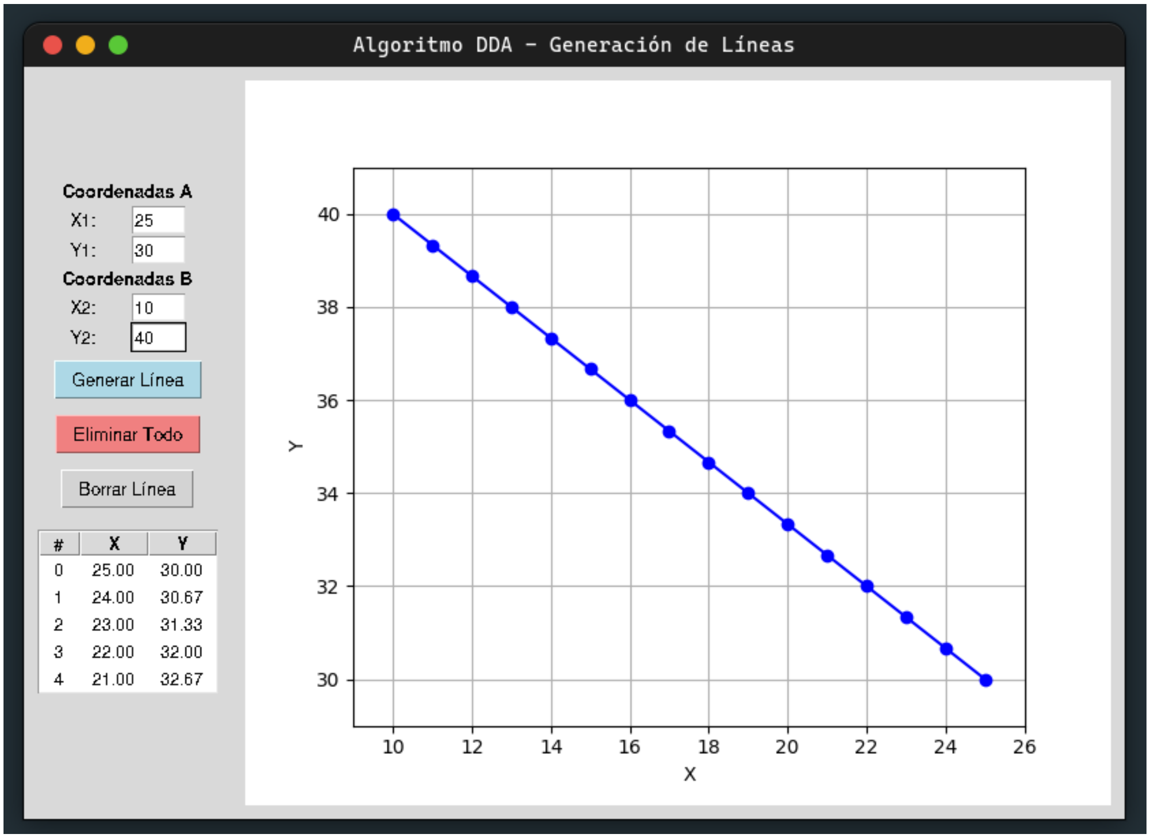
Y1: 27

Coordenadas B

X2: 35

Y2: 22

#	X	Y
0	35.00	27.00
1	35.00	26.00
2	35.00	25.00
3	35.00	24.00
4	35.00	23.00



Algoritmo DDA - Generación de Líneas

Coordenadas A

X1: 25

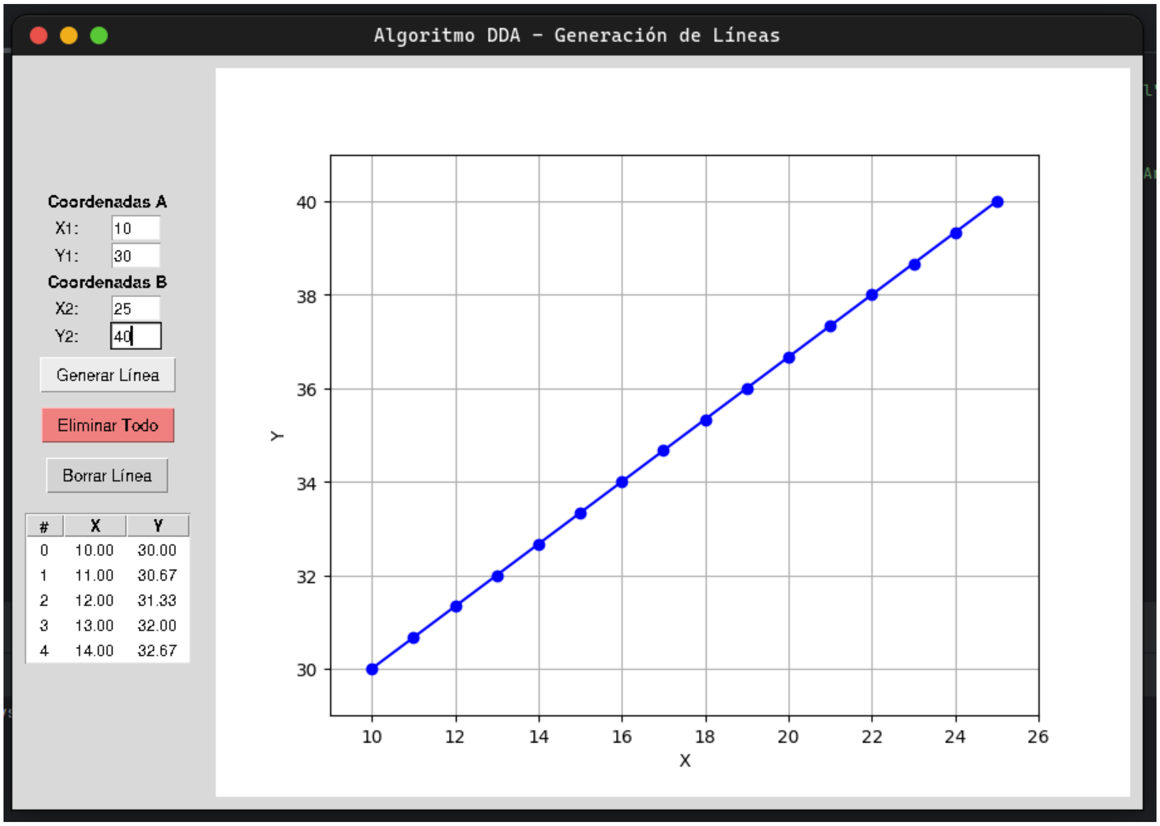
Y1: 30

Coordenadas B

X2: 10

Y2: 40

#	X	Y
0	25.00	30.00
1	24.00	30.67
2	23.00	31.33
3	22.00	32.00
4	21.00	32.67



Algoritmo DDA - Generación de Líneas

Coordenadas A

X1: 10

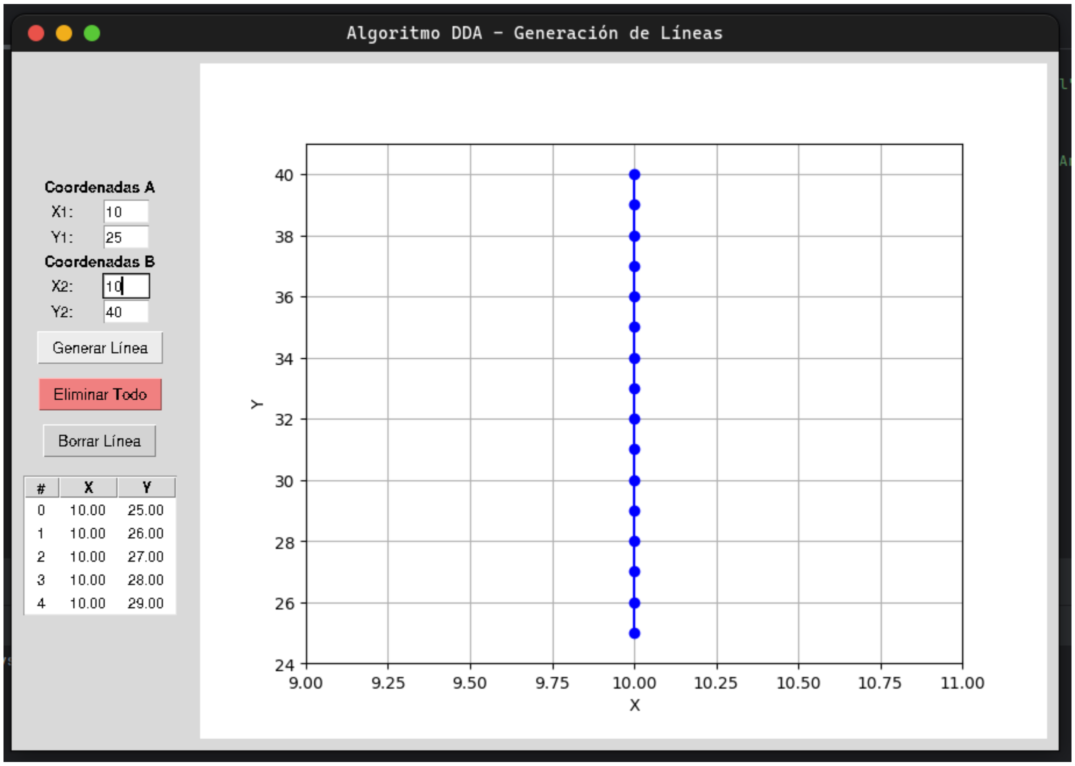
Y1: 30

Coordenadas B

X2: 25

Y2: 40

#	X	Y
0	10.00	30.00
1	11.00	30.67
2	12.00	31.33
3	13.00	32.00
4	14.00	32.67



Coordenadas A

X1: 10

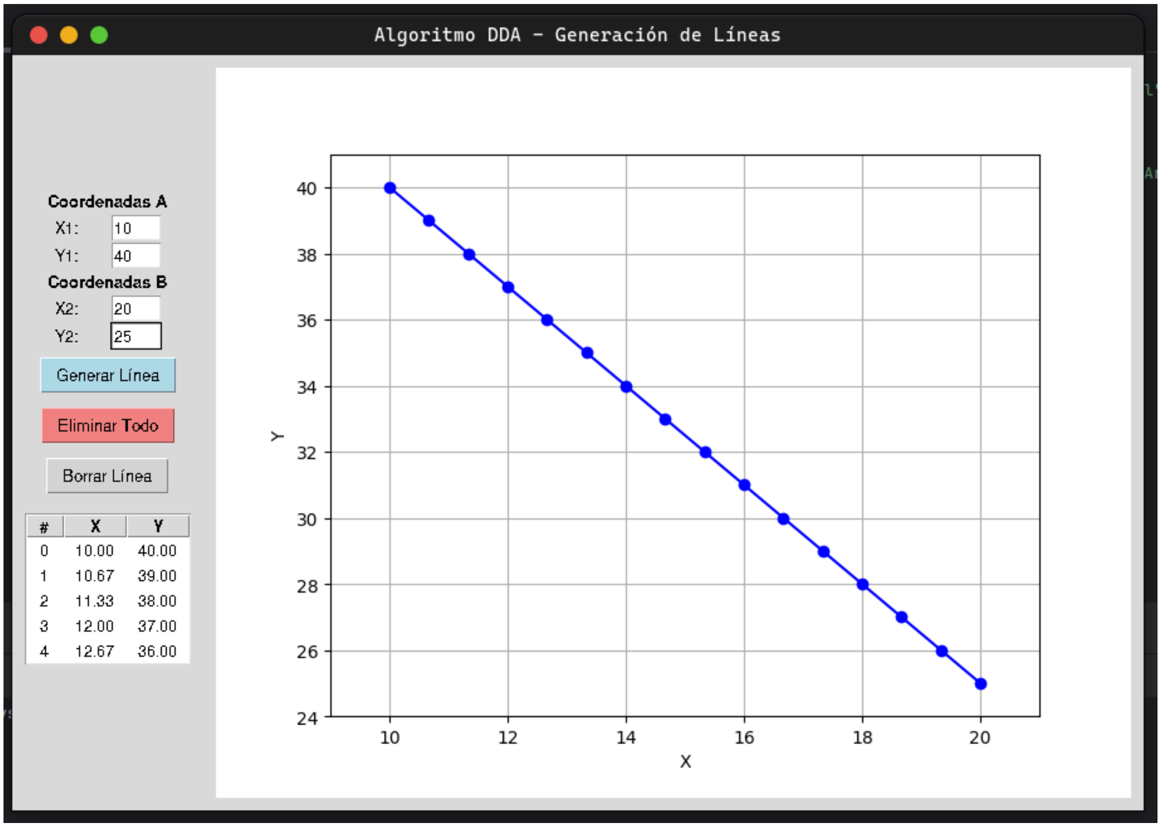
Y1: 25

Coordenadas B

X2: 10

Y2: 40

#	X	Y
0	10.00	25.00
1	10.00	26.00
2	10.00	27.00
3	10.00	28.00
4	10.00	29.00



Coordenadas A

X1: 10

Y1: 40

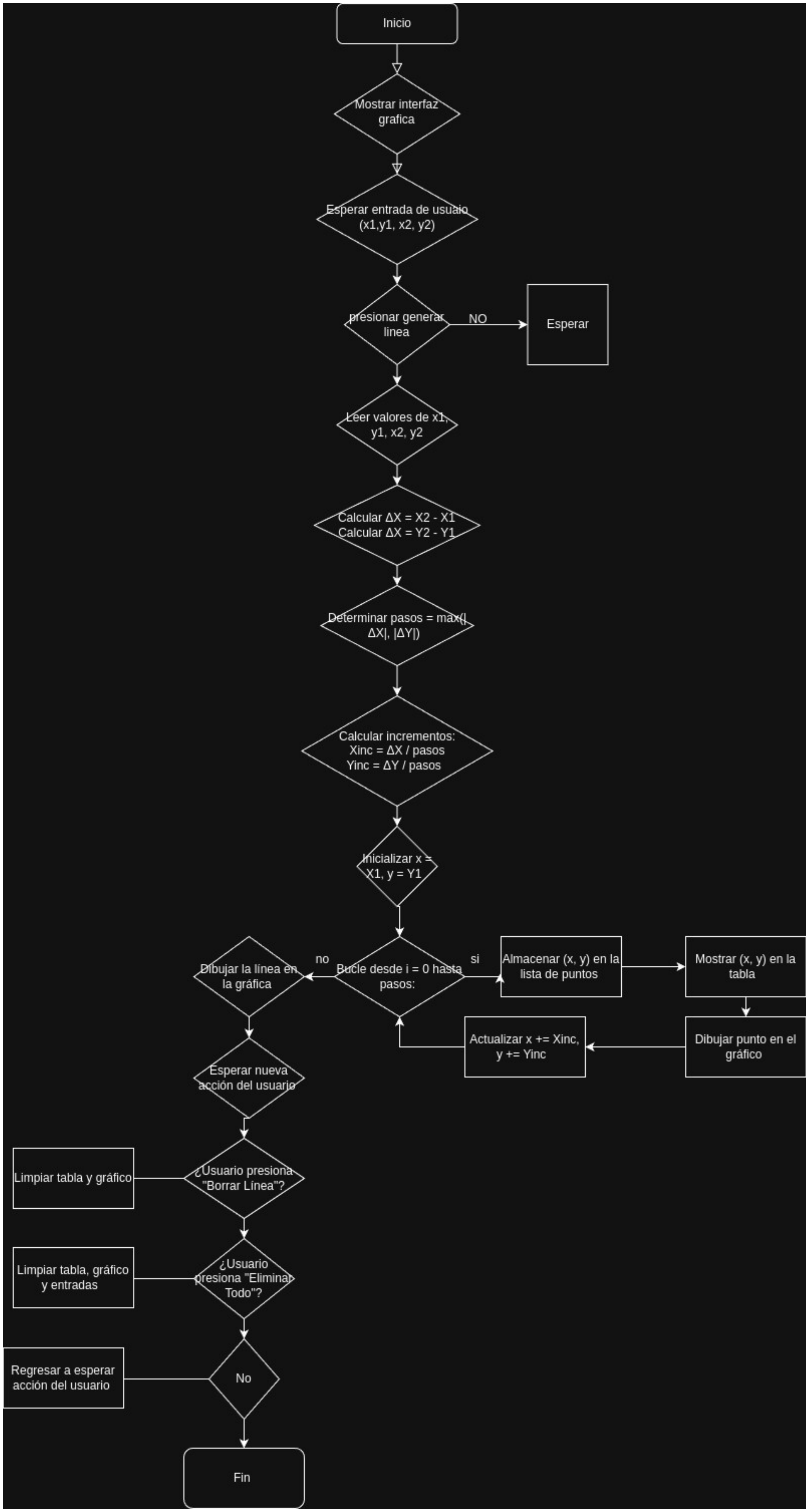
Coordenadas B

X2: 20

Y2: 25

#	X	Y
0	10.00	40.00
1	10.67	39.00
2	11.33	38.00
3	12.00	37.00
4	12.67	36.00

DIAGRAMA DE FLIJO


Inicio
Mostrar interfaz grafica
Esperar entrada de usuaio (x1,y1, x2, y2)
presionar generar linea
NO
Esperar
Leer valores de x1, y1, x2, y2
Calcular ΔX = X2 - X1
Calcular ΔX = Y2 - Y1
Determinar pasos = max(|ΔX|, |ΔY|)
Calcular incrementos:
Xinc = ΔX / pasos
Yinc = ΔY / pasos
Inicializar x = X1, y = Y1
Bucle desde i = 0 hasta pasos:
si
Almacenar (x, y) en la lista de puntos
Mostrar (x, y) en la tabla
Dibujar punto en el gráfico
Actualizar x += Xinc, y += Yinc
no
Dibujar la línea en la gráfica
Esperar nueva acción del usuario
¿Usuario presiona "Borrar Línea"?
Limpiar tabla y gráfico
¿Usuario presiona "Eliminar Todo"?
Limpiar tabla, gráfico y entradas
No
Regresar a esperar acción del usuario
Fin

PROGRAMA FUENTE

```
import tkinter as tk
from tkinter import ttk
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg


def dda_algorithm(x1, y1, x2, y2):
    points = []
    dx = x2 - x1
    dy = y2 - y1
    steps = max(abs(dx), abs(dy))

    Xinc = dx / steps
    Yinc = dy / steps

    x, y = x1, y1
    for i in range(int(steps) + 1):
        points.append((x, y))  # Guardamos valores con decimales
        x += Xinc
        y += Yinc
    return points


def plot_line():
    x1 = float(entry_x1.get())
    y1 = float(entry_y1.get())
    x2 = float(entry_x2.get())
    y2 = float(entry_y2.get())

    points = dda_algorithm(x1, y1, x2, y2)

    # Limpiar tabla
    for row in tree.get_children():
        tree.delete(row)

    # Llenar tabla con valores decimales
    for i, (x, y) in enumerate(points):
        tree.insert("", "end", values=(i, f"{x:.2f}", f"{y:.2f}"))  # Formato con 2 decimales

    # Graficar línea
    ax.clear()
    ax.plot([p[0] for p in points], [p[1] for p in points], marker="o", color="b", linestyle="-")
    ax.set_xlim(min(x1, x2) - 1, max(x1, x2) + 1)
    ax.set_ylim(min(y1, y2) - 1, max(y1, y2) + 1)
    ax.set_xlabel("X")
    ax.set_ylabel("Y")
    ax.grid(True)
    canvas.draw()


def clear_all():
    entry_x1.delete(0, tk.END)
    entry_y1.delete(0, tk.END)
    entry_x2.delete(0, tk.END)
    entry_y2.delete(0, tk.END)
    for row in tree.get_children():
        tree.delete(row)

    ax.clear()
    canvas.draw()


def clear_line():
    for row in tree.get_children():
        tree.delete(row)
    ax.clear()
    canvas.draw()


# Crear ventana principal
root = tk.Tk()
root.title("Algoritmo DDA - Generación de Líneas")
root.geometry("900x600")

# Sección principal
frame_main = tk.Frame(root)
frame_main.pack(side=tk.LEFT, padx=10, pady=10)

# Sección de entrada de coordenadas
frame_input = tk.Frame(frame_main)
frame_input.pack()

label_a = tk.Label(frame_input, text="Coordenadas A", font=("Arial", 10, "bold"))
label_a.grid(row=0, column=0, columnspan=2)

tk.Label(frame_input, text="X1:").grid(row=1, column=0)
tk.Label(frame_input, text="Y1:").grid(row=2, column=0)
entry_x1 = tk.Entry(frame_input, width=5)
entry_y1 = tk.Entry(frame_input, width=5)
entry_x1.grid(row=1, column=1)
entry_y1.grid(row=2, column=1)

label_b = tk.Label(frame_input, text="Coordenadas B", font=("Arial", 10, "bold"))
label_b.grid(row=3, column=0, columnspan=2)

tk.Label(frame_input, text="X2:").grid(row=4, column=0)
tk.Label(frame_input, text="Y2:").grid(row=5, column=0)
entry_x2 = tk.Entry(frame_input, width=5)
entry_y2 = tk.Entry(frame_input, width=5)
entry_x2.grid(row=4, column=1)
entry_y2.grid(row=5, column=1)

btn_generate = tk.Button(frame_input, text="Generar Línea", command=plot_line, bg="lightblue", font=("Arial", 10))
btn_generate.grid(row=6, column=0, columnspan=2, pady=5)

btn_clear = tk.Button(frame_input, text="Eliminar Todo", command=clear_all, bg="lightcoral", font=("Arial", 10))
btn_clear.grid(row=7, column=0, columnspan=2, pady=5)

btn_clear_line = tk.Button(frame_input, text="Borrar Línea", command=clear_line, bg="lightgray", font=("Arial", 10))
```

```
btn_clear_line.grid(row=8, column=0, columnspan=2, pady=5)

# Tabla de valores
frame_table = tk.Frame(frame_main)
frame_table.pack(pady=10)

tree = ttk.Treeview(frame_table, columns=("#", "X", "Y"), show="headings", height=5)
tree.column("#", width=30, anchor="center")
tree.column("X", width=50, anchor="center")
tree.column("Y", width=50, anchor="center")
tree.heading("#", text="#")
tree.heading("X", text="X")
tree.heading("Y", text="Y")
tree.pack()

# Sección de gráfica
frame_plot = tk.Frame(root)
frame_plot.pack(side=tk.RIGHT, padx=10, pady=10, expand=True, fill=tk.BOTH)

fig, ax = plt.subplots(figsize=(7, 7))
canvas = FigureCanvasTkAgg(fig, master=frame_plot)
canvas.get_tk_widget().pack(expand=True, fill=tk.BOTH)

root.mainloop()
```

DIAGRAMA DE GANT

Tareas y Responsabilidades

Tarea	Responsable	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7
Planificación y análisis	Todos	🟦						
Diseño de interfaz en Tkinter	Daniel		🟦	🟦				
Implementación del algoritmo DDA	Cristóbal			🟦	🟦			
Integración con la interfaz	Adrián				🟦	🟦		
Pruebas y corrección de errores	Cristobal					🟦	🟦	
Optimización del código	Adrián					🟦	🟦	
Documentación y entrega	Daniel						🟦	🟦