



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA
Escuela Profesional de Ingeniería de Sistemas

PROYECTO DE UNIDAD

CURSO: DISEÑO Y MODELAMIENTO VIRTUAL

DOCENTE: Hugo Manuel Barraza Vizcarra

Cortez Mamani, Julio Samuel (2023077283)

Tacna – Perú
2025



INDICE

- I. INTRODUCCIÓN**
- II. INFORMACIÓN GENERAL**
- III. RESUMEN Y PALABRAS CLAVES**
- IV. MARCO TEÓRICO**
- V. DISEÑO DEL SISTEMA**
- VI. ESPECIFICACIÓN DEL MENU**
- VII. METODOLOGIAS**
- VIII. RESULTADOS**
- IX. CONCLUSIONES**



I. INTRODUCCIÓN

La rasterización de figuras geométricas constituye un aspecto esencial dentro de los sistemas gráficos por computadora. En este proyecto se desarrolla un entorno CAD 2D capaz de representar figuras básicas mediante la implementación de algoritmos clásicos de trazado. El objetivo principal es analizar y aplicar estos algoritmos a nivel de píxel, evitando el uso de primitivas gráficas avanzadas como GL_LINES o las utilidades de GLU, con el fin de comprender en detalle el proceso de construcción de las imágenes en la etapa más elemental del rasterizado.

II. INFORMACIÓN GENERAL

Objetivo General:

Desarrollar una aplicación CAD 2D en C++ utilizando FreeGLUT/OpenGL, que permita dibujar rectas, círculos y elipses a partir de la implementación de algoritmos clásicos de rasterización. El sistema incluirá un menú interactivo y soporte de interacción mediante mouse y teclado.

Objetivos Específicos:

- Preparar y compilar proyectos en C++ empleando la librería FreeGLUT/OpenGL en diferentes entornos.
- Implementar el trazado de rectas usando el método directo y el algoritmo DDA.
- Desarrollar el algoritmo del punto medio para el dibujo de círculos.
- Aplicar el algoritmo del punto medio en la construcción de elipses (considerando las dos regiones).
- Diseñar un menú contextual que facilite la selección de figuras, colores y herramientas de dibujo.
- Comprobar la correcta ejecución de los algoritmos mediante casos de prueba con diferentes condiciones y parámetros.

III. RESUMEN Y PALABRAS CLAVES

Resumen:



El proyecto consiste en el diseño de un software CAD 2D desarrollado en C++ con el uso de la librería FreeGLUT/OpenGL. En él se implementan algoritmos clásicos de rasterización para la representación de rectas (método directo y DDA), círculos (punto medio) y elipses (punto medio). El programa incorpora un menú interactivo, manejo de eventos mediante mouse y utilidades básicas como la limpieza del lienzo. La validez de los algoritmos se comprueba a través de casos de prueba que abarcan diferentes pendientes, cuadrantes y dimensiones de radios o semiejes.

Palabras Claves:

CAD 2D, rasterización, FreeGLUT, OpenGL, DDA, punto medio, rectas, círculos, elipses.

IV. MARCO TEÓRICO

• RECTA – Método Directo

Este método parte de la ecuación general de la recta $y = mx + b$. Para su correcta implementación es necesario contemplar distintos casos: pendientes con valores menores a uno ($|m| < 1$), pendientes pronunciadas ($|m| > 1$), así como situaciones especiales de rectas verticales y horizontales.

• RECTA – Método DDA

El algoritmo DDA realiza el trazado de la recta a partir de incrementos fraccionarios en los ejes x o y, redondeando al entero más cercano en cada paso. Esto permite obtener un resultado más uniforme y con menor complejidad computacional respecto al método directo.

• CÍRCULO – Método Punto Medio

Para dibujar circunferencias se emplea el parámetro de decisión p, que indica cuál es el siguiente píxel que se debe activar. El algoritmo aprovecha la simetría en los ocho octantes del círculo, reduciendo así el número de cálculos necesarios.

• ELIPSE - Método Punto Medio

En el caso de las elipses, el algoritmo se divide en dos regiones de acuerdo con la pendiente de la curva:

- Región 1: cuando la pendiente es menor que uno.
- Región 2: cuando la pendiente es mayor o igual a uno.

El cambio entre regiones se determina con condiciones de transición que aseguran la continuidad del trazado y la correcta simetría en los cuatro cuadrantes de la figura.



V. Diseño del Sistema

Arquitectura:

- Interfaz de Usuario (GLUT): captura de eventos, menús y visualización.
- Módulo de Figuras: implementa algoritmos de recta, círculo y elipse.
- Módulo de Utilidades: limpieza de lienzo, coordenadas y ejes.

Módulos:

- **Utilidades graficas:**

- putPixel()
- drawAxes()
- Conversión de coordenadas de mouse → coordenadas OpenGL (lo haces dentro de mouse(), pero podrías abstraerlo en utils).

- **Rectas:**

- drawLineDirect()
- drawLineDDA()
- Todo el manejo de clics (primer punto y segundo punto) está en mouse(). Eso lo podrías encapsular después en una función propia del módulo de líneas, aunque no es obligatorio.

- **Círculos:**

- drawCircleMidpoint().
- Con el mouse() controlas el flujo para capturar centro y radio. Podría ir en este módulo también si quieres separar responsabilidades

- **Elipsis:**

- drawEllipseMidpoint().
- Al igual que en círculos, el manejo de clics lo tienes en mouse(). Está bien, pero podrías abstraerlo si buscas una separación más clara.

- **Interfaz:**

- initMenu()
- menuHandler()
- Conecta con mouse(), display() y cambia la herramienta/estilo.

- **Principal:**



- `init()` con configuración OpenGL.
- `display()` → llama a `renderShape()`.
- `mouse()` → maneja entrada para figuras.
- `main()` → configura ventana, registra callbacks, inicia bucle principal.

VI. ESPECIFICACIÓN DEL MENU

El menú implementado utilizando la función `glutCreateMenu()` de FreeGLUT, permitiendo la selección de figuras, colores, grosor de trazo y utilidades. Se especifican sus opciones:

- **Dibujo:**

- Recta (Directo) → Activa el algoritmo de la ecuación $y=mx+by = mx + by=mx+b$.
- Recta (DDA) → Activa el algoritmo incremental DDA.
- Círculo (Punto Medio) → Dibuja un círculo aplicando el algoritmo del punto medio.
- Elipse (Punto Medio) → Dibuja una elipse con el algoritmo del punto medio.

- **Color:**

- Negro
- Rojo
- Verde
- Azul

- **Grosor:**

- 1 px
- 2 px
- 3 px
- 5 px

- **Vista:**

- Mostrar/Ocultar cuadrícula
- Mostrar/Ocultar ejes
- Mostrar coordenadas del puntero

- **Herramientas:**

- Limpiar lienzo
- Borrar todas figura

VII. METODOLOGIAS

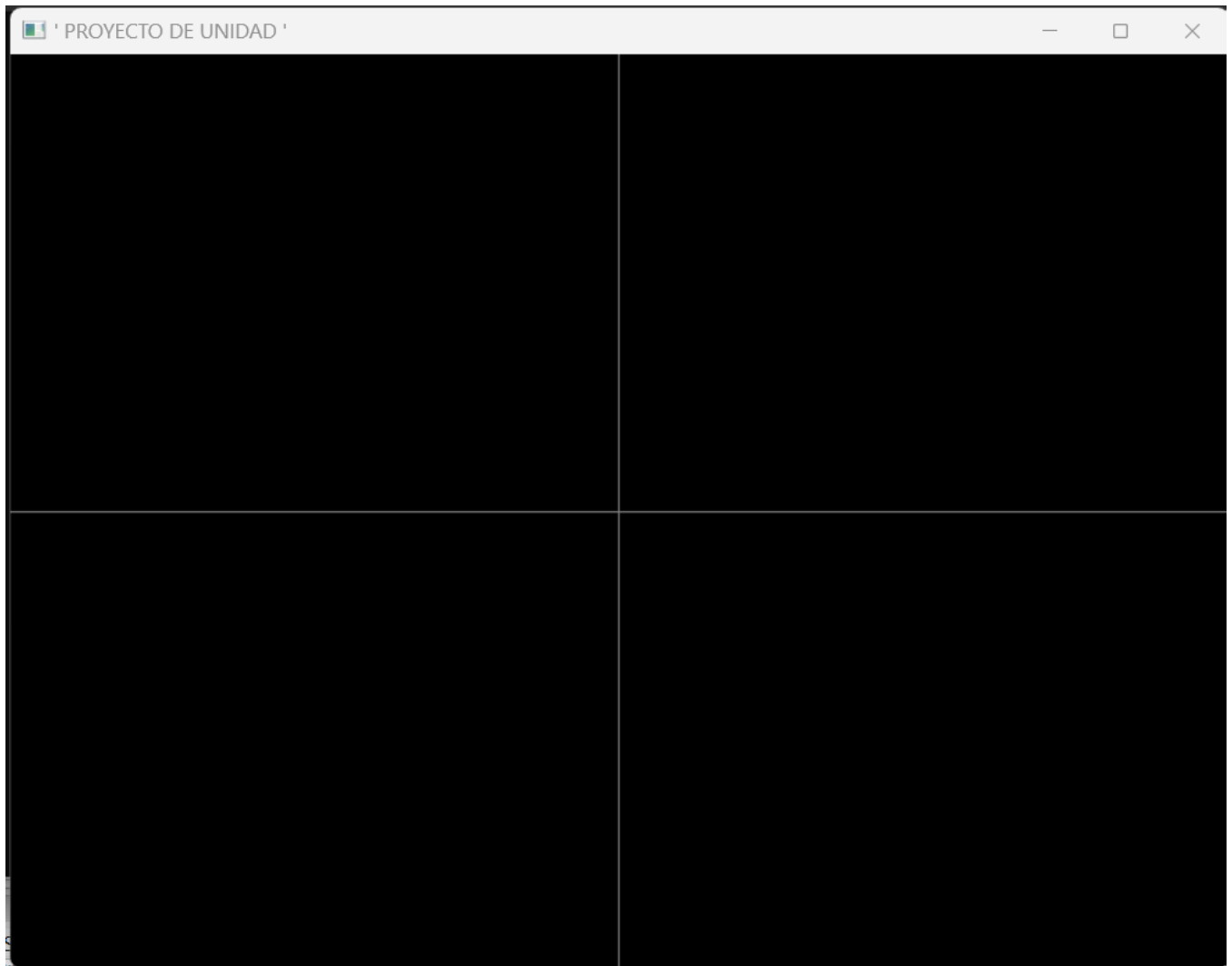


PROYECTO DE UNIDAD

- Lenguaje de programación: C++
- Entorno: Code::Blocks 20.03 con FreeGLUT y OpenGL
- Estructura del sistema: módulos de figuras, utilidades y menús.
- Diseño: uso de eventos de mouse para seleccionar puntos y vectores para almacenar múltiples figuras.

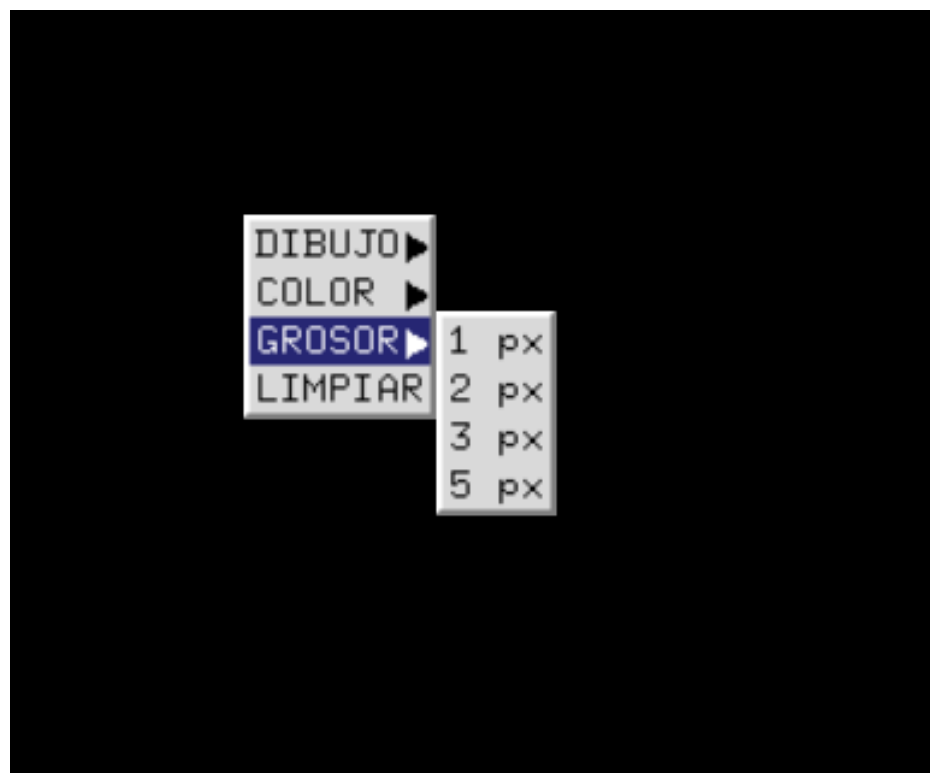
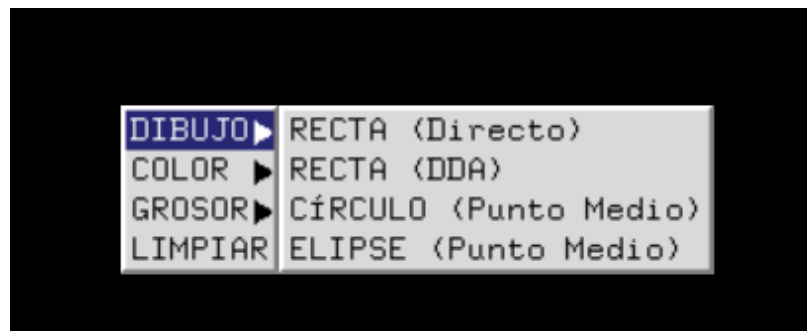
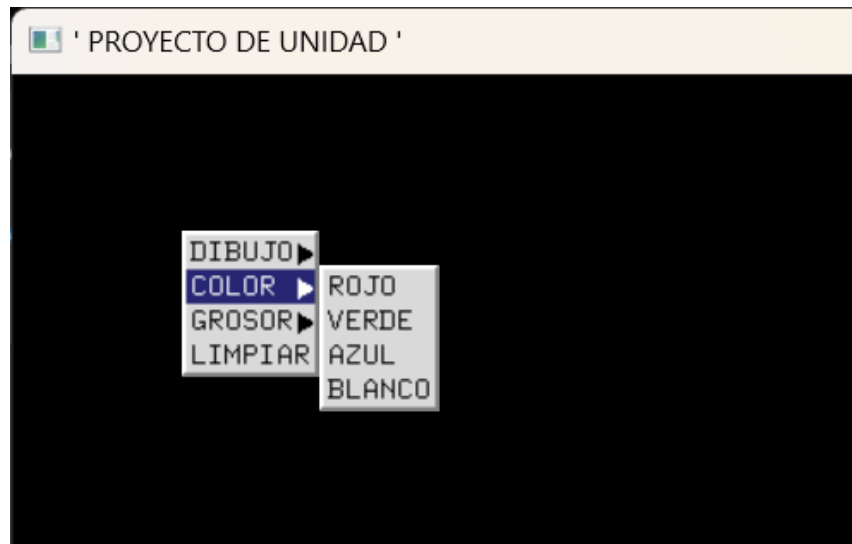
VIII.RESULTADOS

Ejecución correcta del programa.



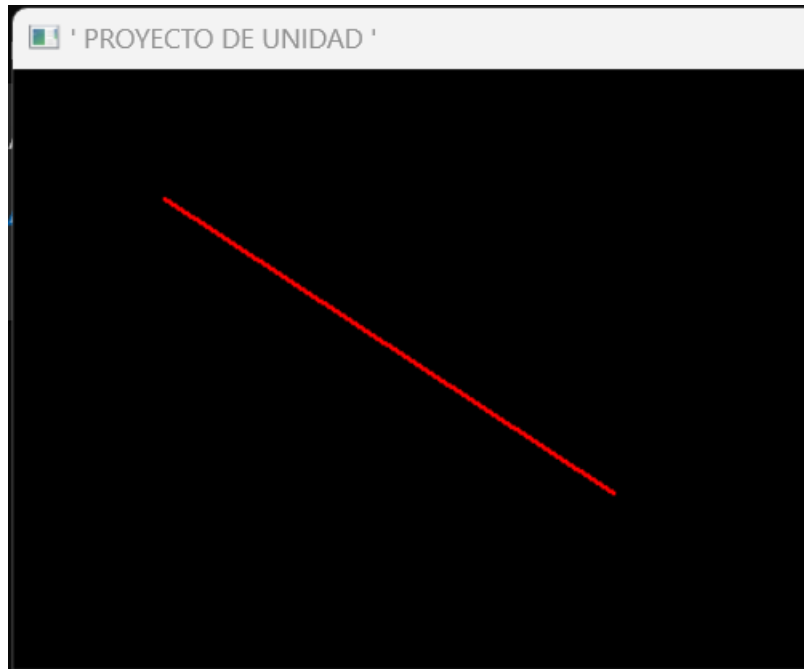


Correcto funcionamiento del Menú

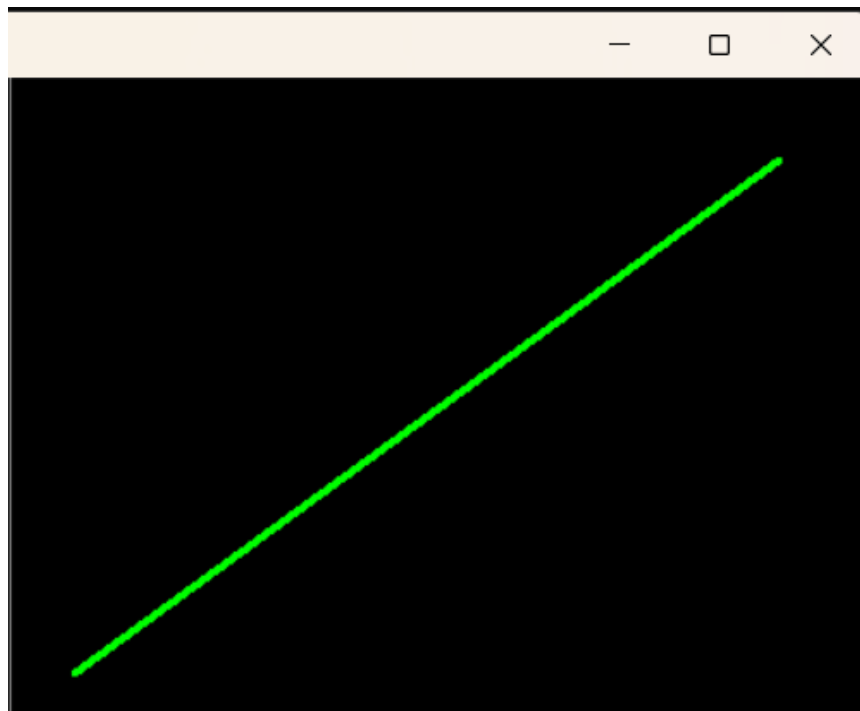




Correcto funcionamiento de la RECTA (Directa)

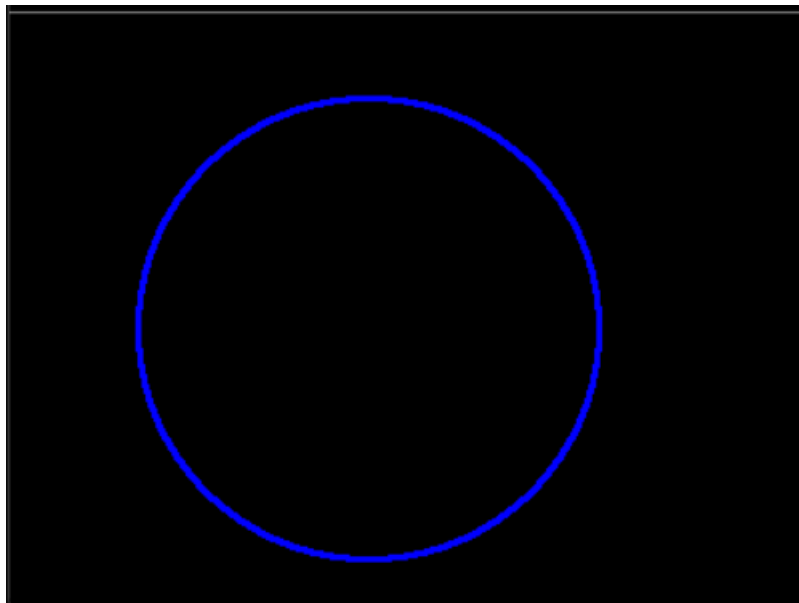


Correcto funcionamiento de la RECTA (Incremental)

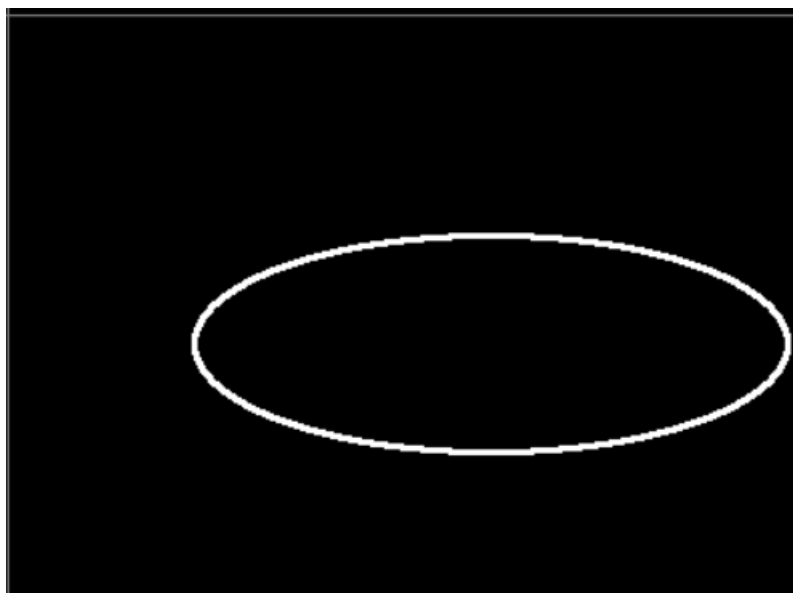


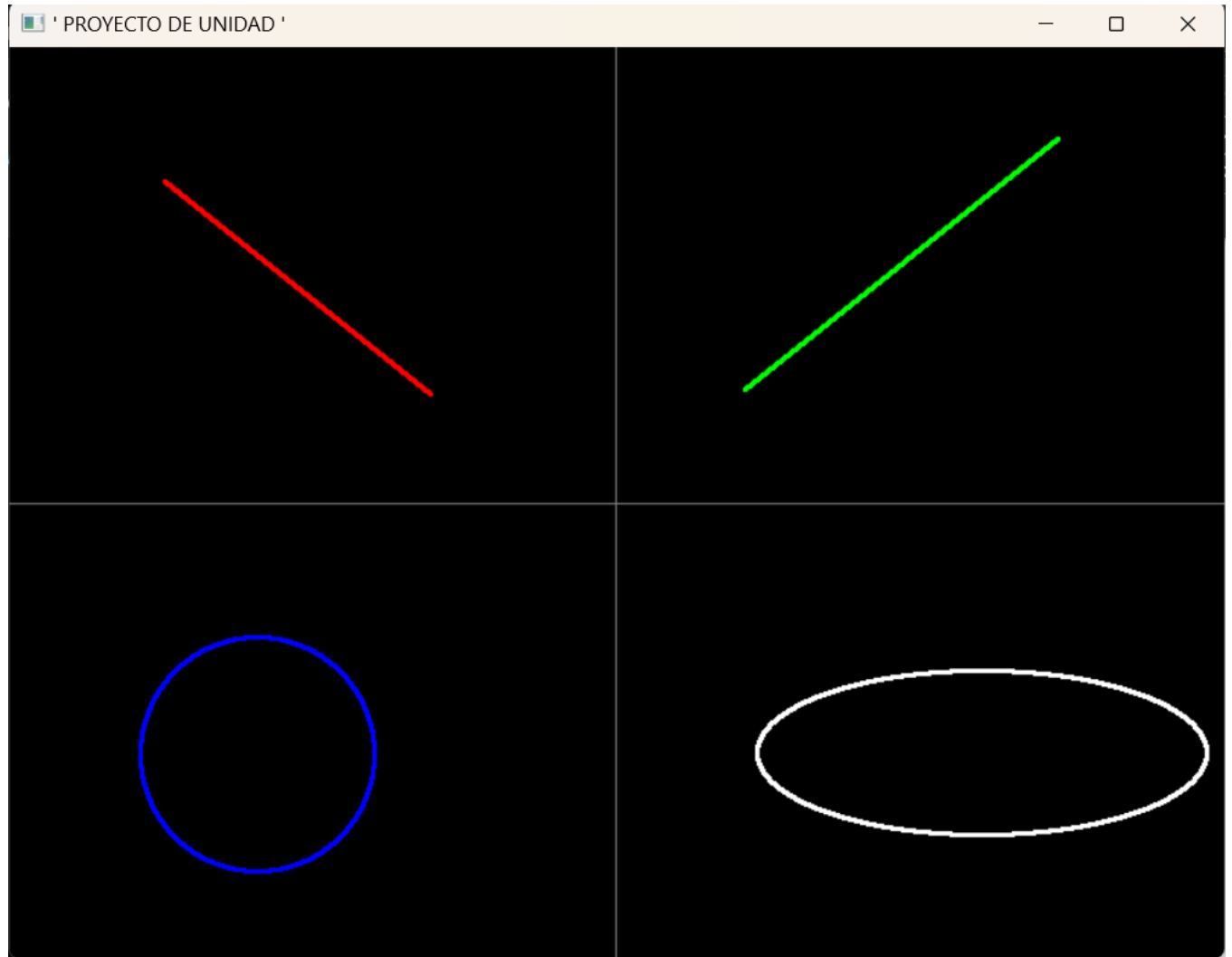


Correcto funcionamiento de la CIRCULO (Punto Medio)



Correcto funcionamiento de la ELIPSE (Punto Medio)







IX. CONCLUSIONES

- Se implementaron correctamente los algoritmos clásicos de rasterización en C++.
- Se comprobó la simetría de círculos y elipses en todos los cuadrantes.
- La integración de menús y mouse permite una experiencia interactiva y cercana a un software CAD real.
- El uso de vectores hizo posible dibujar múltiples figuras de manera simultánea.