UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

INFORME DE DMV (PROYECTO)

Curso: Diseño y Modelamiento Virtual

Docente: Ing. HUGO MANUEL BARRAZA VIZCARRA

Jimenez Laura, Carlos Raí (2023078693)

Tacna – Perú
2025

Indice

Resumen

En este informe se presenta el desarrollo de un software CAD 2D básico implementado en C++ utilizando la librería FreeGLUT/OpenGL. El sistema permite el trazado de líneas mediante los algoritmos Directo y DDA, así como la rasterización de círculos y elipses mediante el método del Punto Medio. Se integró un menú de opciones, controles por teclado y soporte de interacción con el mouse, además de una funcionalidad de exportación de imágenes en formato PPM. Los resultados muestran la correcta implementación de los algoritmos y la validación con diferentes casos de prueba.

Palabras clave: CAD 2D, rasterización, OpenGL, FreeGLUT, DDA, Punto Medio.

Introducción y Objetivos

Introducción

El estudio de los algoritmos de rasterización es fundamental en la computación gráfica, ya que permite comprender cómo se transforman las primitivas geométricas en representaciones digitales en pantalla. Este proyecto busca afianzar dichos conocimientos mediante la implementación de un software CAD 2D con herramientas básicas de dibujo.

Objetivos

- **Objetivo general:** Implementar un software CAD 2D en C++ con FreeGLUT/OpenGL que permita el trazado de líneas, círculos y elipses mediante algoritmos clásicos de rasterización.

- **Objetivos específicos:**

    - Implementar los algoritmos Directo, ADD/DDA y Punto Medio.

    - Desarrollar un sistema de menús y atajos de teclado.

    - Permitir la interacción mediante mouse.

    - Implementar exportación de imágenes a formato PPM.

    - Validar los algoritmos mediante casos de prueba.

Marco Teórico

- **Método Directo:** Utiliza la ecuación de la recta y=mx+by = mx + by=mx+b. Es simple, pero presenta problemas con redondeo y eficiencia cuando la pendiente es grande.

- **Método ADD/DDA (Digital Differential Analyzer):** Calcula los puntos de la línea de forma incremental, reduciendo errores de redondeo. Es más estable que el directo.

- **Método del Punto Medio para Círculos:** Se basa en un parámetro de decisión que aprovecha la simetría de los octantes para trazar solo una parte del círculo y reflejar los puntos.

- **Método del Punto Medio para Elipses:** Divide el trazado en dos regiones (pendiente < -1 y > -1) y usa un parámetro de decisión para determinar el siguiente píxel.

Diseño del Sistema

Arquitectura

- **Módulo de Interfaz:** gestiona la ventana OpenGL, menús y eventos de teclado/mouse.

- **Módulo de Rasterización:** contiene las implementaciones de los algoritmos (Directo, DDA, Punto Medio).

- **Módulo de Utilidades:** exportación a PNG, cuadrícula, ejes, grosor y color.

Diagramas de Flujo (por algoritmo)

1. **Línea Directa**: entrada de puntos → cálculo pendiente → iteración en X o Y → pintar píxeles.

2. **DDA**: entrada de puntos → determinar pasos → incremento de X e Y → pintar píxeles.

3. **Círculo Punto Medio**: entrada centro y radio → cálculo parámetro inicial → octantes → pintar píxeles.

4. **Círculo Incremental**: entrada centro y radio → calcular seno y coseno de ángulo fijo → generar coordenadas con incrementos → pintar píxeles.

5. **Elipse Punto Medio**: entrada centro y semiejes → región 1 → región 2 → pintar píxeles.

(Aquí van los diagramas de flujo en imágenes o hechos en Draw.io/PowerPoint)

Especificación del Menú y Atajos

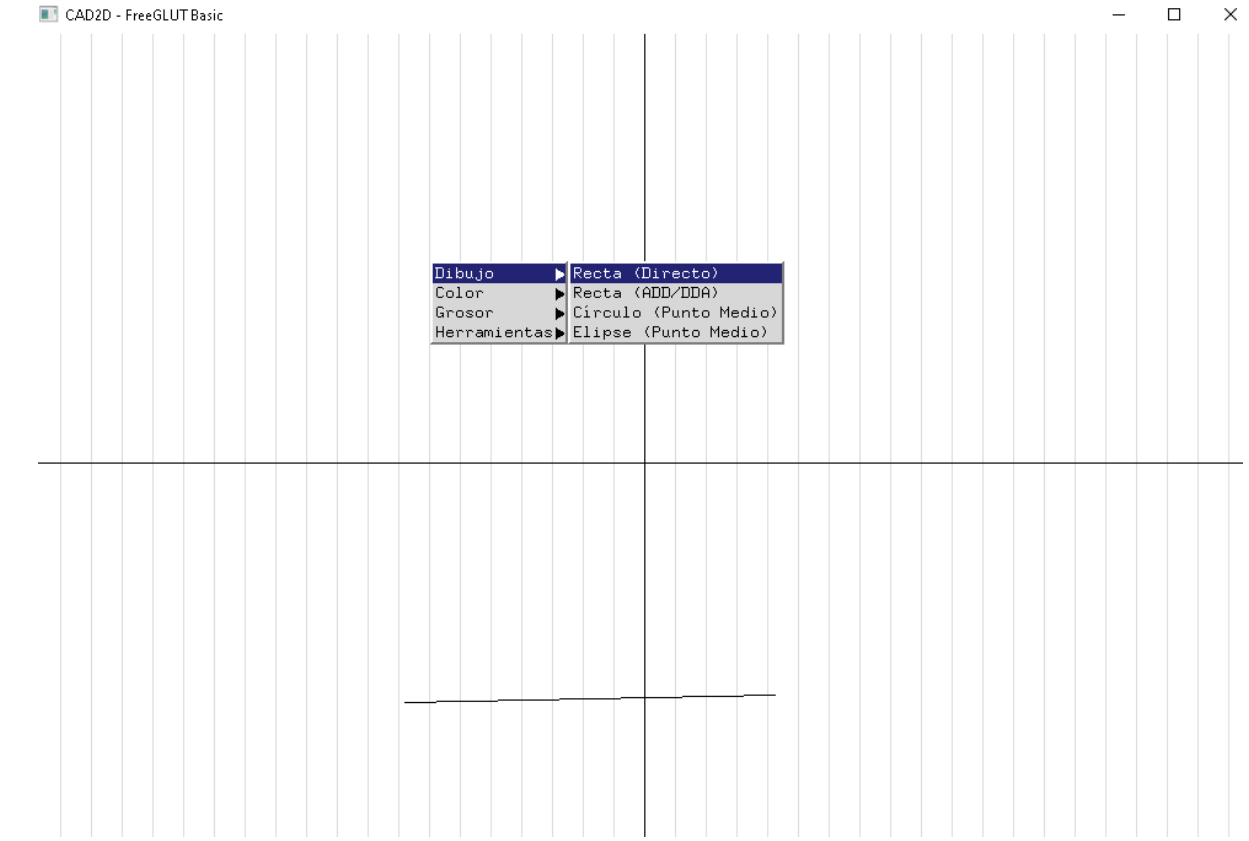
- **Menú principal (botón derecho):**
    - Dibujo → Línea Directa, Línea DDA, Círculo, Elipse.
    - Color → Negro, Rojo, Verde, Azul.
    - Grosor → 1 px, 2 px, 3 px, 5 px.
    - Vista → cuadrícula, ejes, coordenadas del puntero.
    - Herramientas → limpiar lienzo, borrar figura, exportar imagen.
    - Ayuda → atajos, acerca de.
- **Atajos de teclado:**
    - `G`: activar/desactivar cuadrícula.
    - `E`: activar/desactivar ejes.
    - `C`: limpiar lienzo.
    - `S`: exportar imagen.
    - `Z`: deshacer.
    - `Y`: rehacer.
    - `Esc`: salir.

Casos de Prueba y Resultados

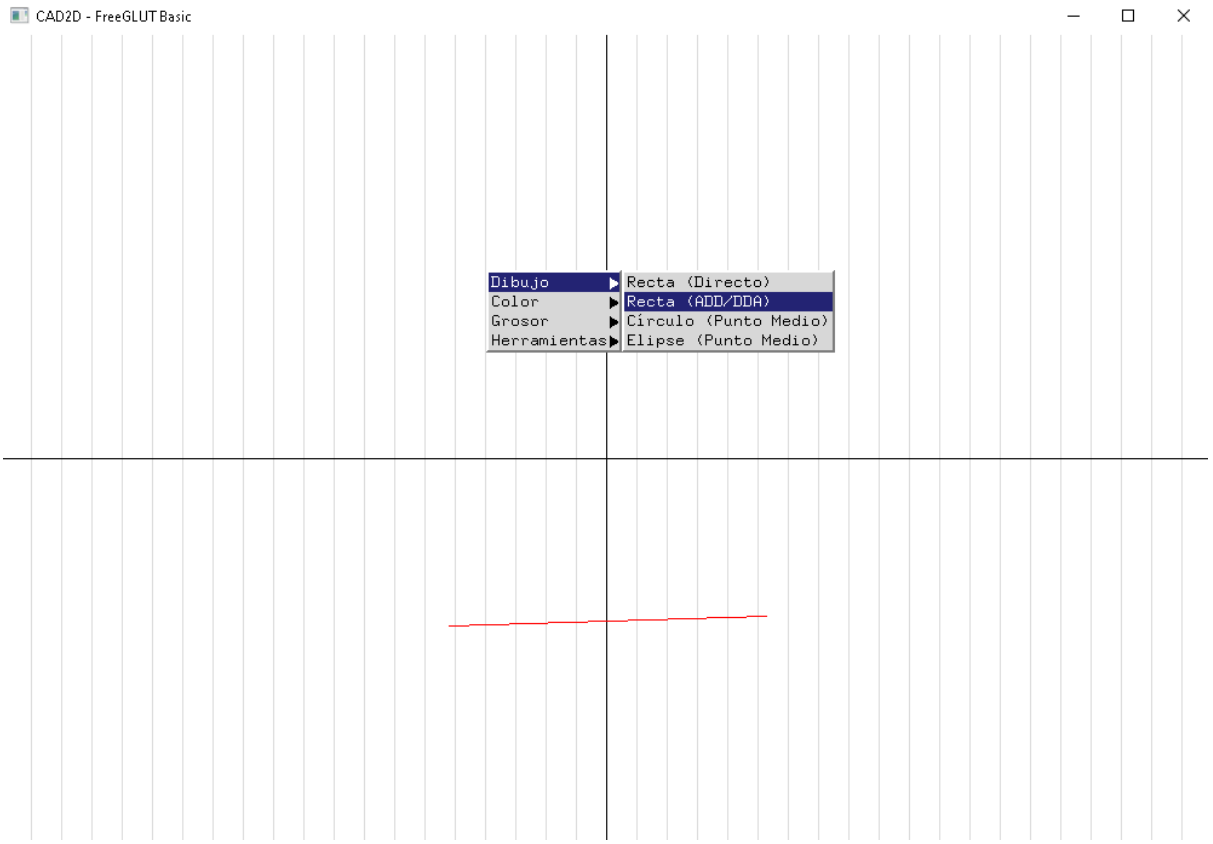
Para validar el correcto funcionamiento del software se realizaron varios casos de prueba, considerando diferentes configuraciones de figuras, posiciones y parámetros.

1. **Rectas (método directo y DDA)**
    - Se probaron pendientes positivas, negativas, horizontales y verticales.
    - El método directo presentó ligeras imprecisiones cuando la pendiente era mayor a 1 o menor a -1, debido al redondeo en los cálculos.
    - El algoritmo DDA mostró un trazado más uniforme, con menos saltos en la representación de píxeles.

Método Directo:

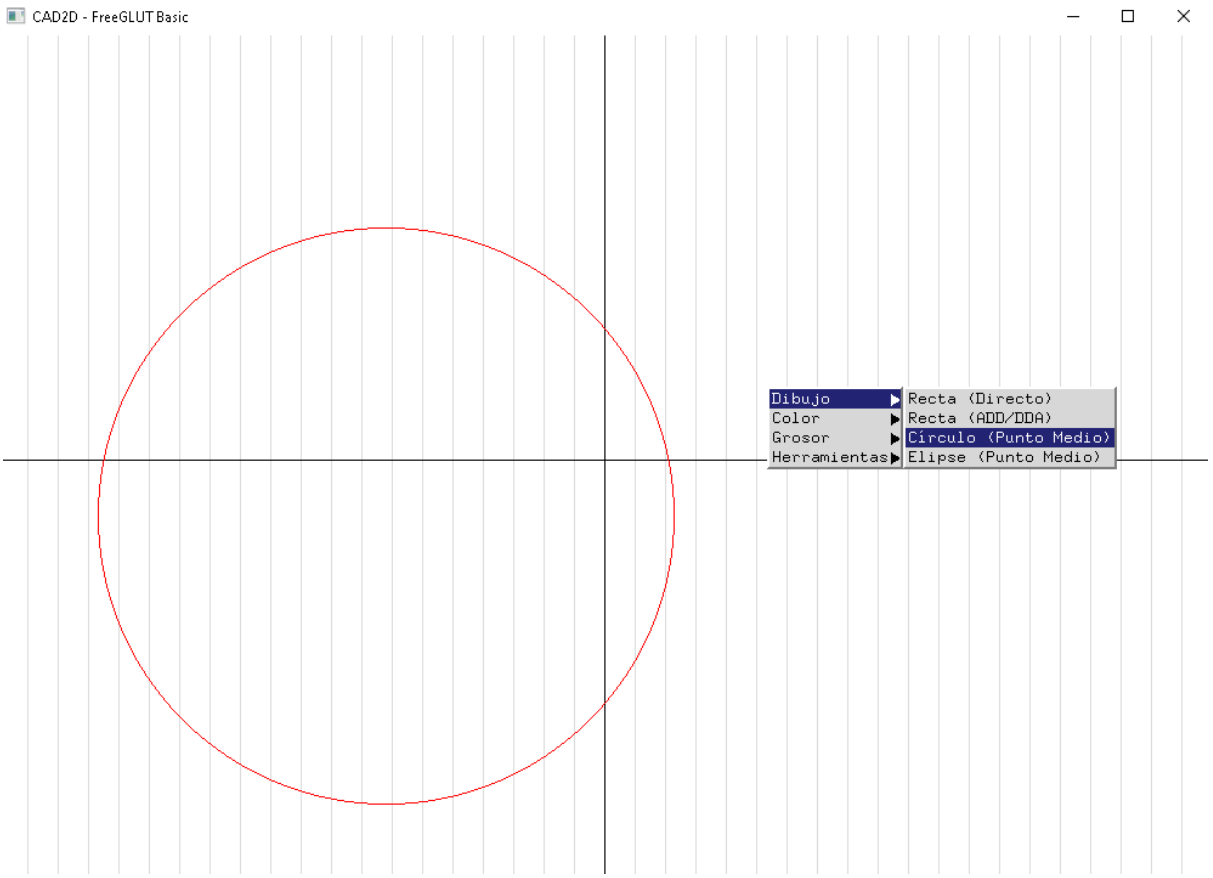


Método DDA:



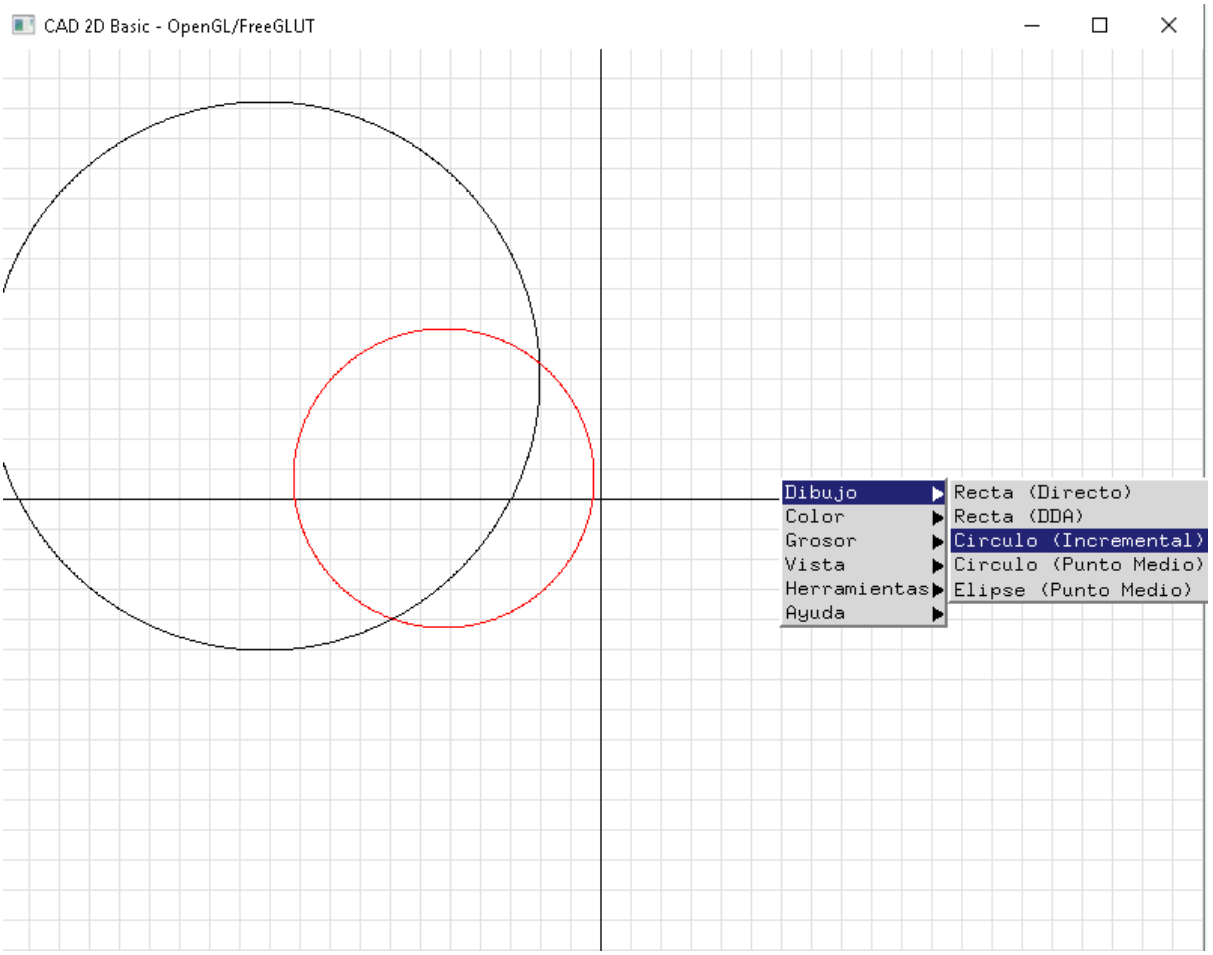
Círculos (punto medio)

- Se evaluaron radios pequeños (ej. 20 px) y grandes (ej. 200 px).
- El algoritmo mantuvo la simetría en los 8 octantes, confirmando su estabilidad en cualquier tamaño.
- En radios muy grandes, se observó que el tiempo de rasterización aumentaba, pero dentro de lo esperado.



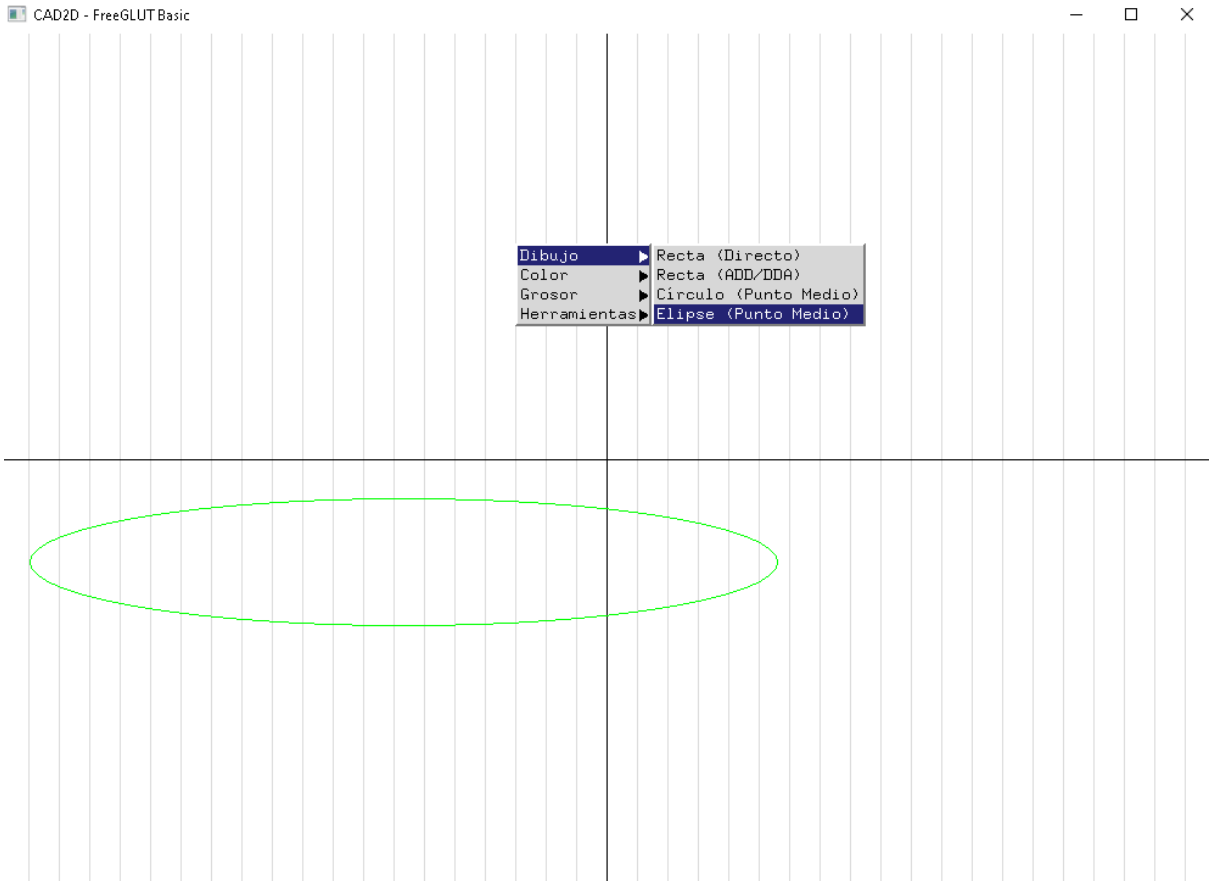
Círculo Incremental

- Se probaron radios pequeños y grandes, verificando que el uso de seno y coseno con incrementos angulares permitió generar los puntos del círculo de manera continua.
- El método demostró buena precisión en radios pequeños y medianos, aunque en radios muy grandes puede acumular ligeros errores de redondeo por el cálculo trigonométrico.
- La simetría se mantuvo estable y el resultado visual fue correcto en todos los octantes.



2. **Elipses (punto medio)**
    - Se probaron elipses con distintos radios: circulares (rx = ry), alargadas en X (rx > ry) y alargadas en Y (ry > rx).
    - La división en región 1 y 2 se ejecutó correctamente, evitando distorsiones.
    - El resultado visual fue satisfactorio en todos los cuadrantes.

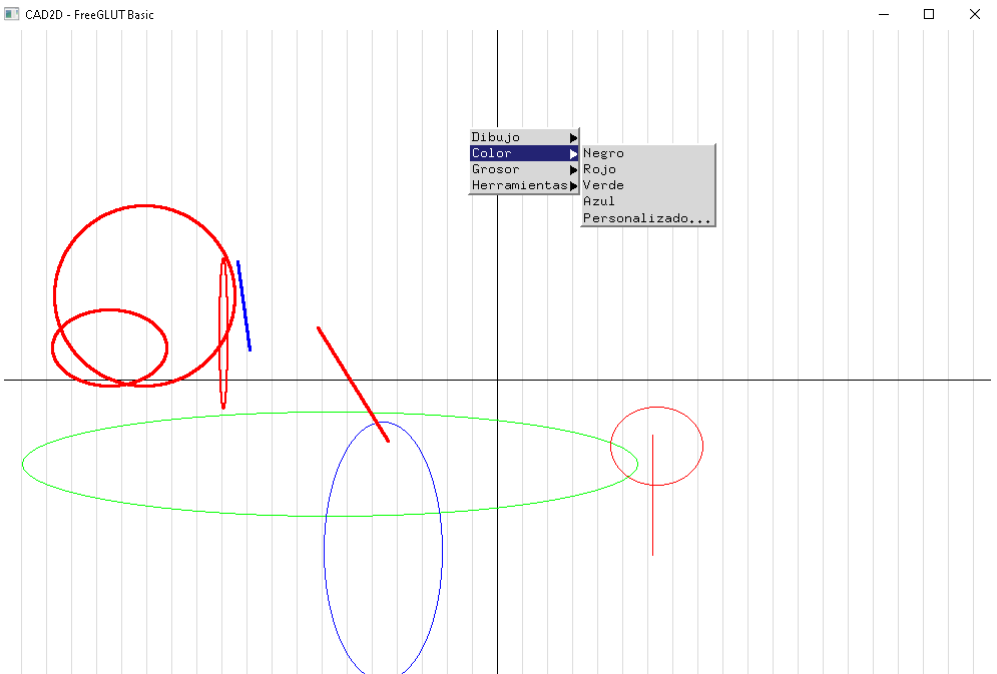
Elipse punto medio:



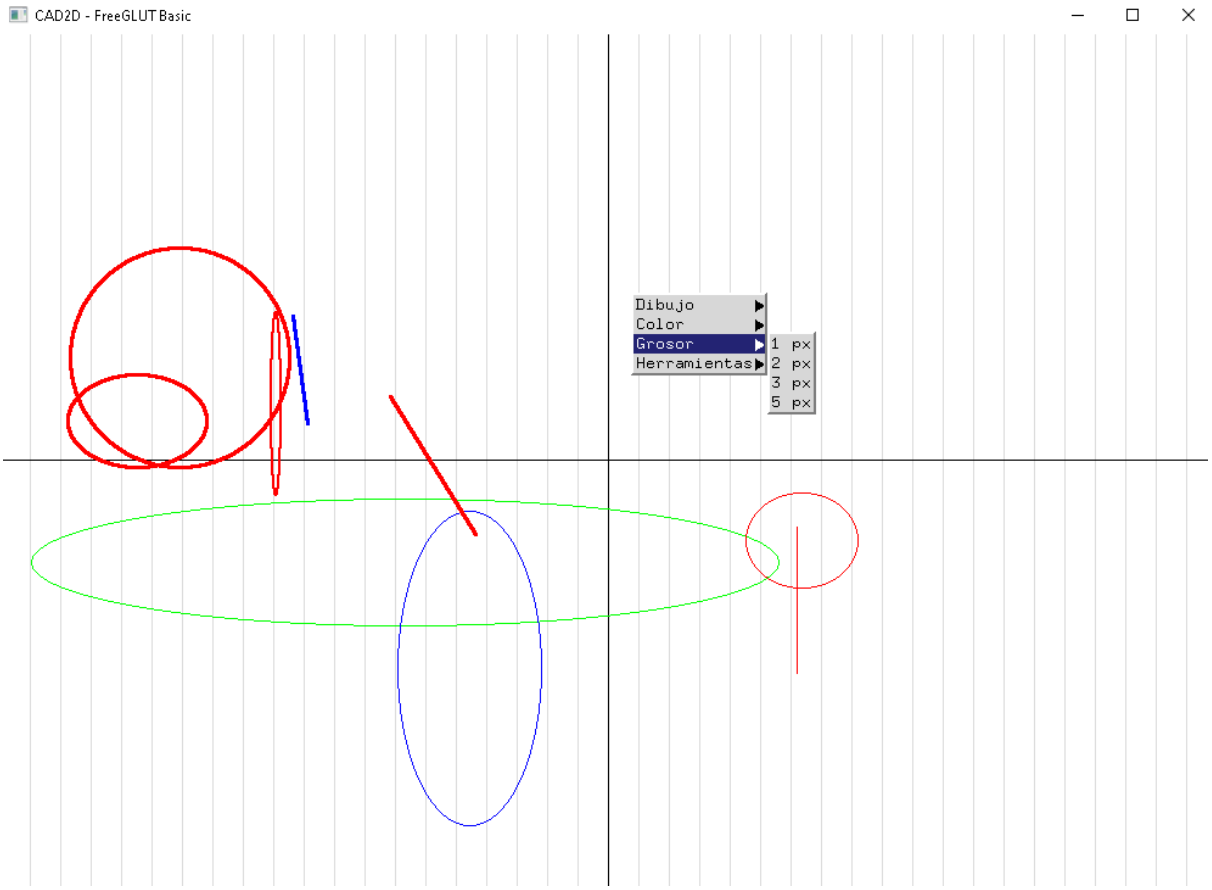
3. **Interacción y menú**

   - El menú popup permitió cambiar entre algoritmos, colores y grosor sin necesidad de reiniciar la aplicación.

Colores:

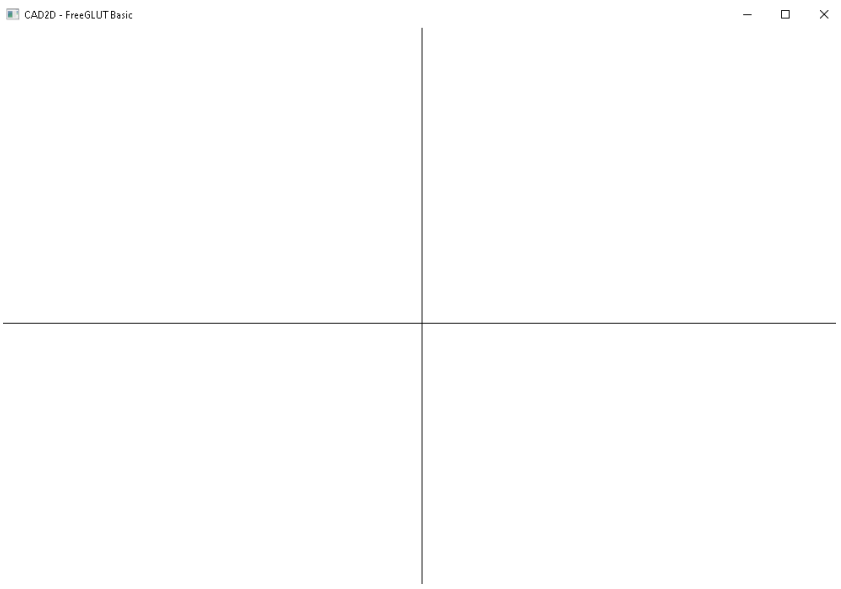


Grosor:

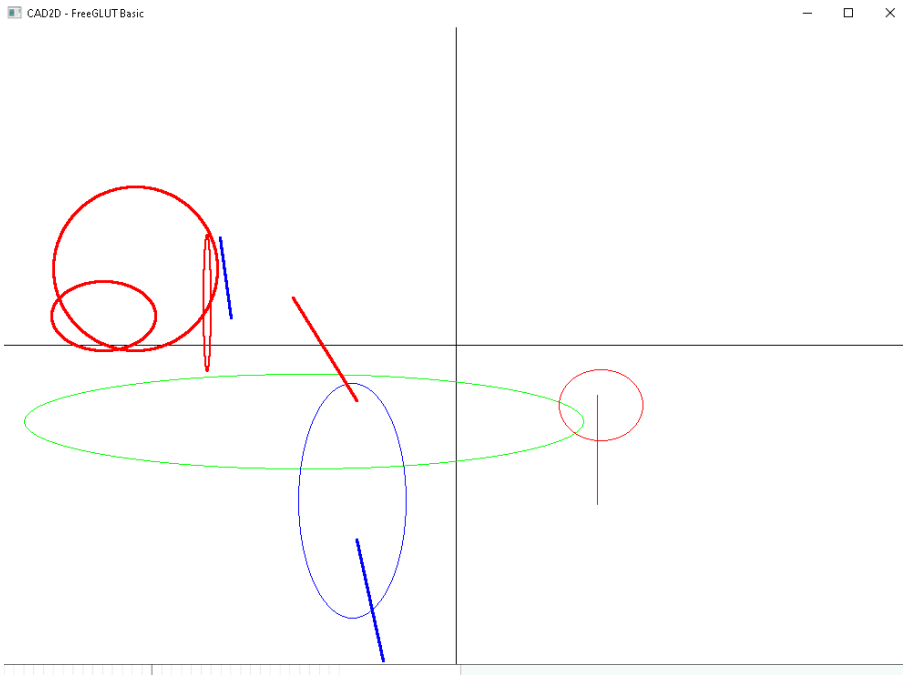


- Los atajos de teclado (ej. `C` para limpiar, `G` para rejilla, `S` para exportar,`Z/Y` (`undo/redo`)) facilitaron la validación rápida.

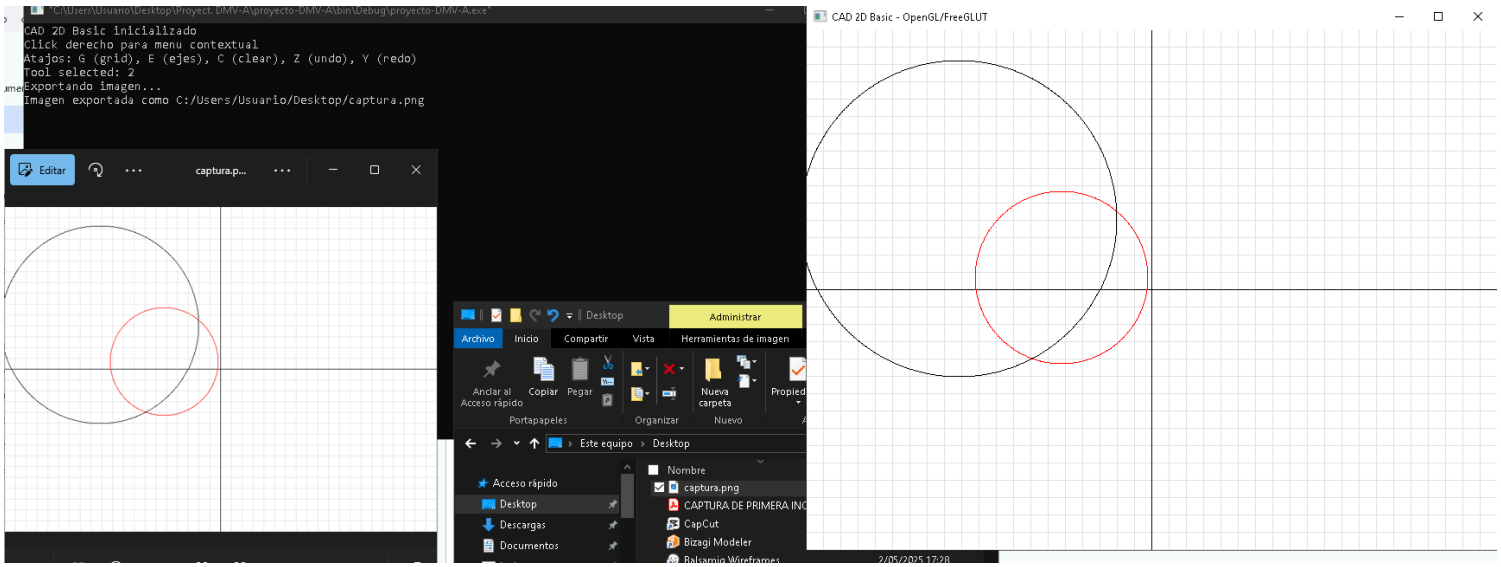
  C: Limpiar:



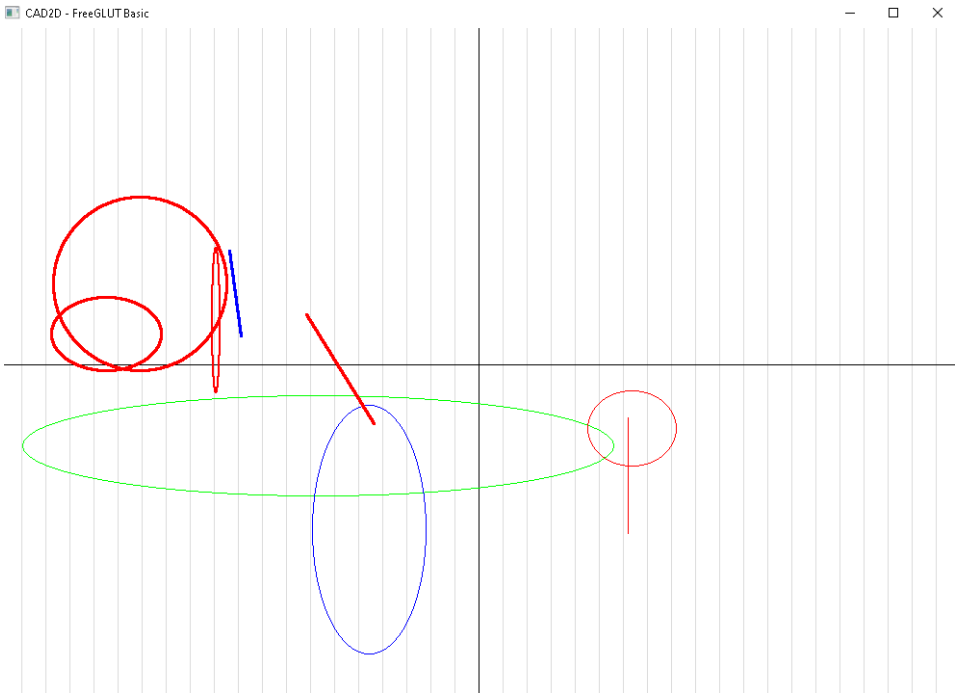
G: Rejilla:



S: Exportar



Z: Undo



Y: Redo

4. **Exportación de resultados**

    - Las pruebas de exportación generaron archivos en formato PPM. Si bien es un formato básico, permitió corroborar que los píxeles rasterizados coincidían con lo observado en pantalla.

En general, los casos de prueba confirmaron que los algoritmos cumplen con la teoría estudiada y que el software es funcional para ilustrar la lógica de rasterización.

(Aquí colocar capturas de pantalla del programa con cuadrícula y zoom)

Conclusiones y Trabajo Futuro

- El proyecto permitió implementar y analizar distintos algoritmos de rasterización, como línea directa, DDA y punto medio (para círculos y elipses). Se comprobó que el método directo es menos eficiente, mientras que DDA y punto medio ofrecen mejores resultados en precisión y rendimiento. El uso del menú y atajos facilitó la prueba práctica, mostrando claramente las diferencias entre algoritmos. En general, se cumplió con el objetivo de comprender y aplicar técnicas básicas de gráficos por computadora.

- Como trabajo futuro, se recomienda:
    - Ampliar la exportación de imágenes a formatos más comunes (PNG, JPG).
    - Incluir una interfaz gráfica más intuitiva.
    - Extender los algoritmos a aplicaciones en 3D.
    - Medir tiempos de ejecución para un análisis más completo.

Referencias

- Angel, E., & Shreiner, D. (2014). *Interactive Computer Graphics with WebGL*. Addison-Wesley.
- Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1996). *Computer Graphics: Principles and Practice*. Addison-Wesley.
- Shreiner, D., Sellers, G., Kessenich, J., & Licea-Kane, B. (2013). *OpenGL Programming Guide* (8th ed.). Addison-Wesley.

- FreeGLUT. (2023). *FreeGLUT Documentation*. Recuperado de http://freeglut.sourceforge.net/