UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

Informe de Proyecto de Unidad I
"CAD 2D BASICO"

Curso: Modelamiento de Procesos

Docente: Hugo Manuel Barraza Vizcarra

Quispe Checa, Pablo Sebastian (2023078432)

Tacna – Perú
2025

ÍNDICE

Informe de Proyecto de Unidad I

“CAD 2D BASICO”

I. Resumen

Este documento presenta el desarrollo de un software CAD 2D básico implementado en C++ utilizando FreeGLUT/OpenGL. El sistema incorpora algoritmos clásicos de rasterización para el trazado de líneas (método directo y DDA), círculos y elipses (método del punto medio). La aplicación cuenta con una interfaz gráfica interactiva que permite la selección de herramientas mediante menús contextuales y atajos de teclado, facilitando la creación de figuras geométricas básicas con diferentes colores y grosores. Los algoritmos implementados siguen los principios fundamentales de la computación gráfica, evitando el uso de primitivas de hardware para garantizar el aprendizaje de los conceptos teóricos.

Palabras clave: CAD 2D, rasterización, OpenGL, algoritmos gráficos, método del punto medio, DDA, computación gráfica.

II. Introducción

Los Sistemas de Diseño Asistido por Computadora (CAD, por sus siglas en inglés Computer-Aided Design) representan una de las aplicaciones más significativas de la computación gráfica en el ámbito de la ingeniería y el diseño técnico. Estos sistemas han revolucionado la manera en que se conciben, desarrollan y analizan proyectos en diversas disciplinas, desde la arquitectura y la ingeniería civil hasta el diseño mecánico y la electrónica.

El presente proyecto se enmarca en el estudio y desarrollo de los fundamentos que subyacen a estas poderosas herramientas, específicamente en el contexto del diseño bidimensional. La rasterización de primitivas gráficas constituye la base fundamental sobre la cual se construyen todos los sistemas gráficos modernos.

Comprender los algoritmos clásicos de trazado de líneas, círculos y elipses no solo tiene valor histórico, sino que proporciona insights profundos sobre los desafíos computacionales inherentes al renderizado digital. Aunque las tarjetas gráficas modernas implementan estos algoritmos a nivel hardware con una eficiencia extraordinaria, el conocimiento de sus principios matemáticos y computacionales sigue siendo esencial para cualquier profesional en el campo de la computación gráfica.

El desarrollo de este software CAD 2D básico utilizando FreeGLUT/OpenGL con C++ permite explorar de manera práctica los desafíos inherentes a la implementación de algoritmos de rasterización. OpenGL, como API gráfica multiplataforma, ofrece el entorno ideal para este propósito, proporcionando el control necesario sobre el pipeline gráfico mientras se mantiene un nivel de abstracción que facilita la implementación de los algoritmos fundamentales.

III. Objetivos

Objetivo General:

Desarrollar un software CAD 2D básico en C++ utilizando FreeGLUT/OpenGL que implemente algoritmos clásicos de rasterización para líneas, círculos y elipses, integrando una interfaz gráfica interactiva con capacidades de visualización y herramientas de edición básicas.

Objetivos Específicos:

A. 1. Implementación de Algoritmos de Rasterización
B. 2. Desarrollo de la Interfaz Gráfica
C. 3. Gestión de Interacción y Eventos
D. 4. Herramientas de Visualización y Utilidades
E. 5. Validación y Pruebas

IV. Marco Teórico

2.1 Algoritmo Directo para Líneas

El método directo utiliza la ecuación de la recta y = mx + b para calcular los puntos intermedios. Se manejan casos especiales para pendientes suaves (|m| < 1) y pronunciadas (|m| > 1), así como líneas verticales y horizontales.

2.2 Algoritmo DDA (Digital Differential Analyzer)

El algoritmo DDA es incremental y calcula los puntos de la línea mediante diferencias finitas. Es más eficiente que el método directo al evitar el cálculo repetitivo de la ecuación de la recta.

2.3 Algoritmo del Punto Medio para Círculos

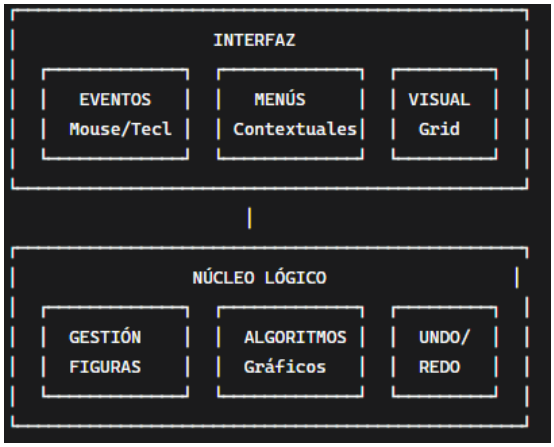
Basado en la ecuación del círculo $x^2 + y^2 = r^2$, este algoritmo utiliza un parámetro de decisión para seleccionar el siguiente pixel, aprovechando la simetría de los 8 octantes para reducir cálculos.

2.4 Algoritmo del Punto Medio para Elipses

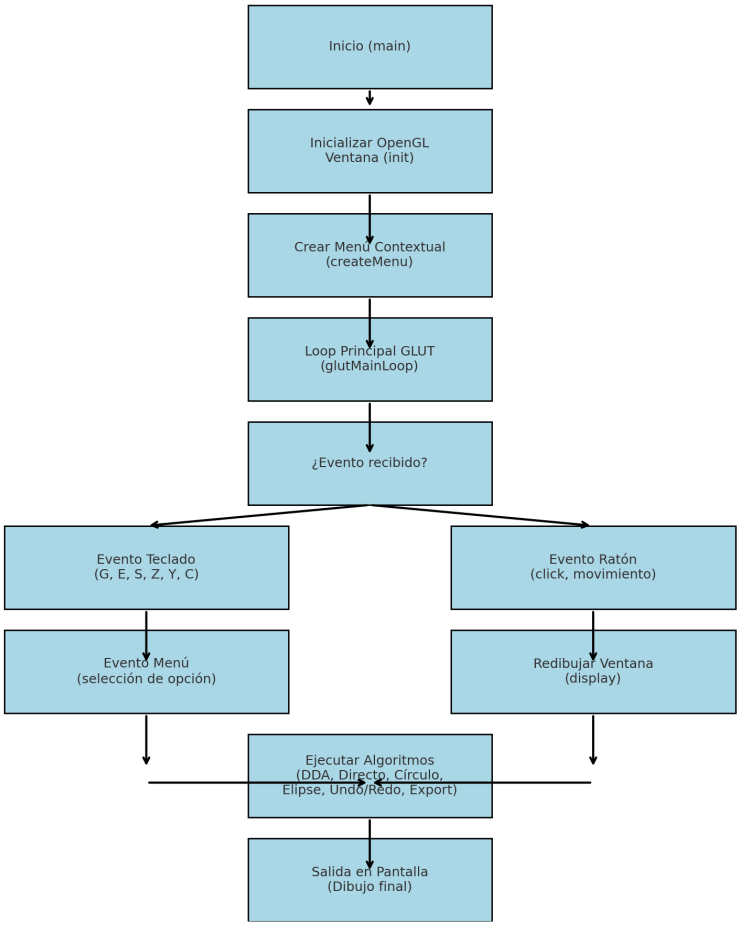
Extensión del algoritmo para círculos, maneja dos regiones diferentes con criterios de cambio específicos basados en la ecuación de la elipse $(x^2/a^2) + (y^2/b^2) = 1$.

V. Diseño de Sistema

➔ **Arquitectura del Sistema**



➔ **Diagrama de Flujo**



VI. Especificación del menú y atajos

Estructura del Menú Principal (Click Derecho)

1. Dibujo Recta (Directo)

- [Algoritmo: Línea Directa] Recta (DDA)
- [Algoritmo: DDA] Circulo (Punto Medio)
- [Algoritmo: Punto Medio] Elipse (Punto Medio) - [Algoritmo: Elipse]

2. Color

Negro - Color predefinido #000000

Rojo - Color predefinido #FF0000

Verde - Color predefinido #00FF00

Azul - Color predefinido #0000FF

Personalizado - Permite ingresar valores RGB personalizados

3. Grosor

- 1 px - Grosor de línea delgado
- 2 px - Grosor de línea medio
- 3 px - Grosor de línea grueso
- 5 px - Grosor de línea extra grueso

4. Vista

- Cuadricula - Alternar visualización de cuadrícula (Atajo: G)
- Ejes - Alternar visualización de ejes coordenados (Atajo: E)
- Coordenadas - Mostrar coordenadas actuales del cursor

5. Herramientas

1. Limpiar - Borrar todo el lienzo (Atajo: C)
2. Deshacer - Revertir la última acción (Atajo: Z)
3. Exportar - Guardar imagen como archivo PPM (Atajo: S)

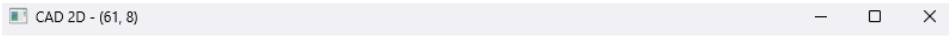
6. Formatos de Exportación PPM

- Formato de imagen portable de mapa de pixeles
- El archivo se guarda como "export.ppm" en el directorio actual
- Resolución: 800x600 píxeles (ajustable)

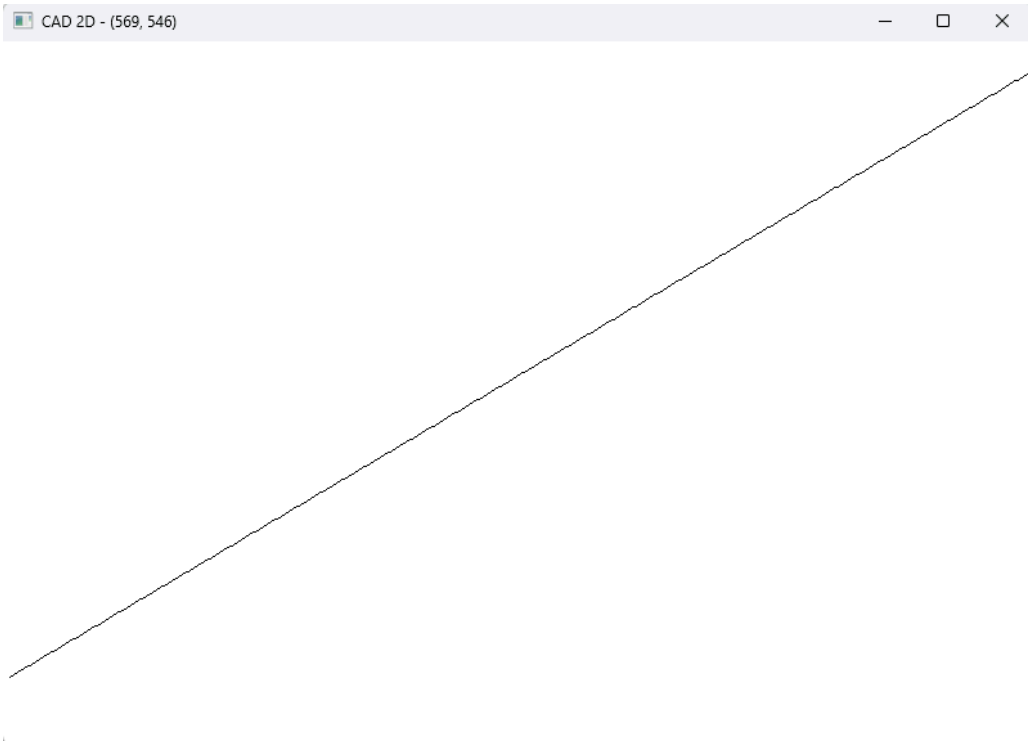
VII. Casos de Prueba y Resultados

5.1 Pruebas de Líneas

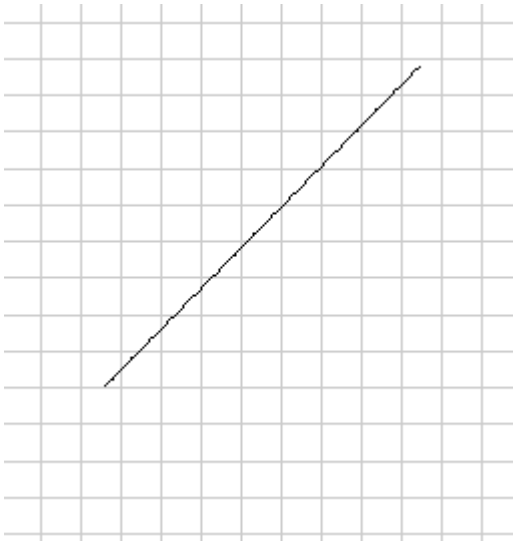
- Caso 1: Línea horizontal (pendiente = 0)



- Caso 2: Línea vertical (pendiente infinita)

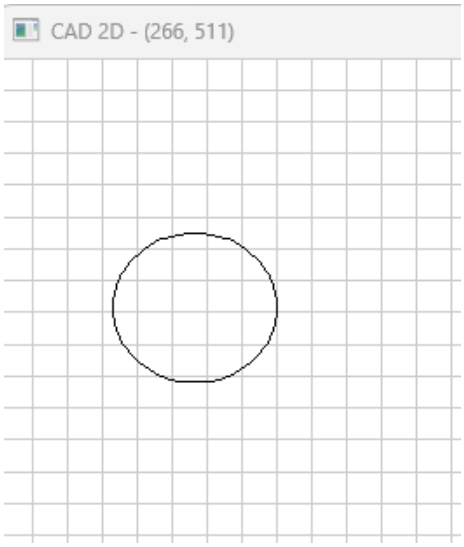


- Caso 3: Línea con pendiente 1 (45°)

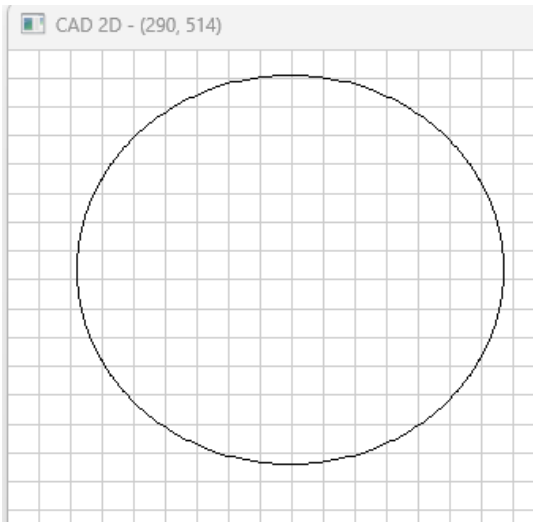


5.2 Pruebas de Círculos

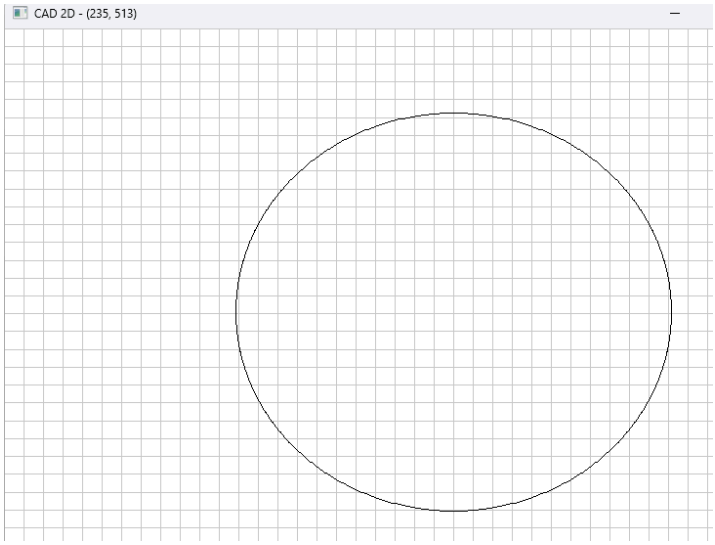
- Caso 1: Círculo pequeño (radio = 10 px)



- Caso 2: Círculo mediano (radio = 50 px)

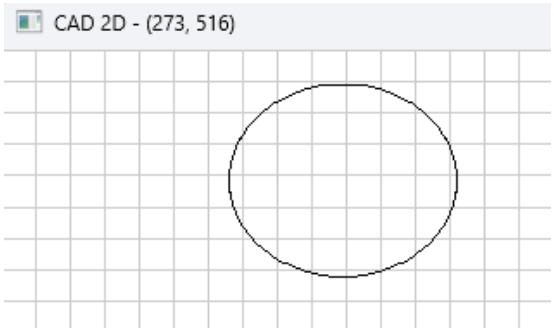


- Caso 3: Círculo grande (radio = 100 px)

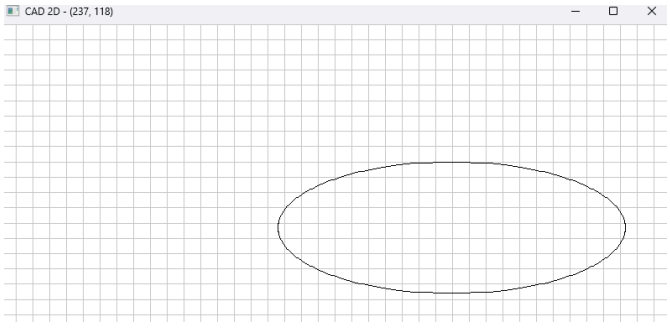


Pruebas de Elipses

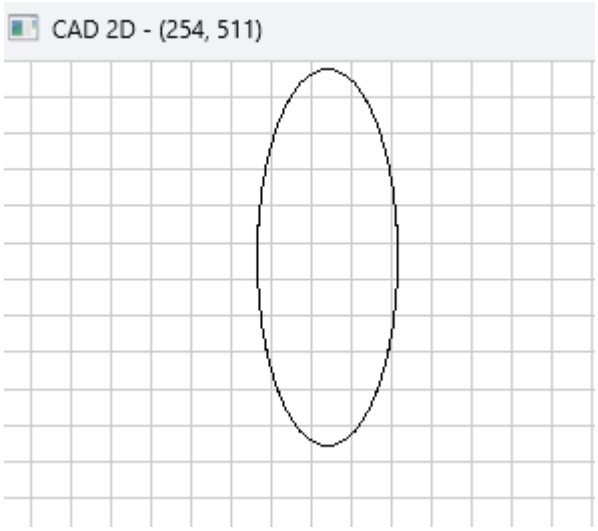
- Caso 1: Elipse circular (rx = ry)



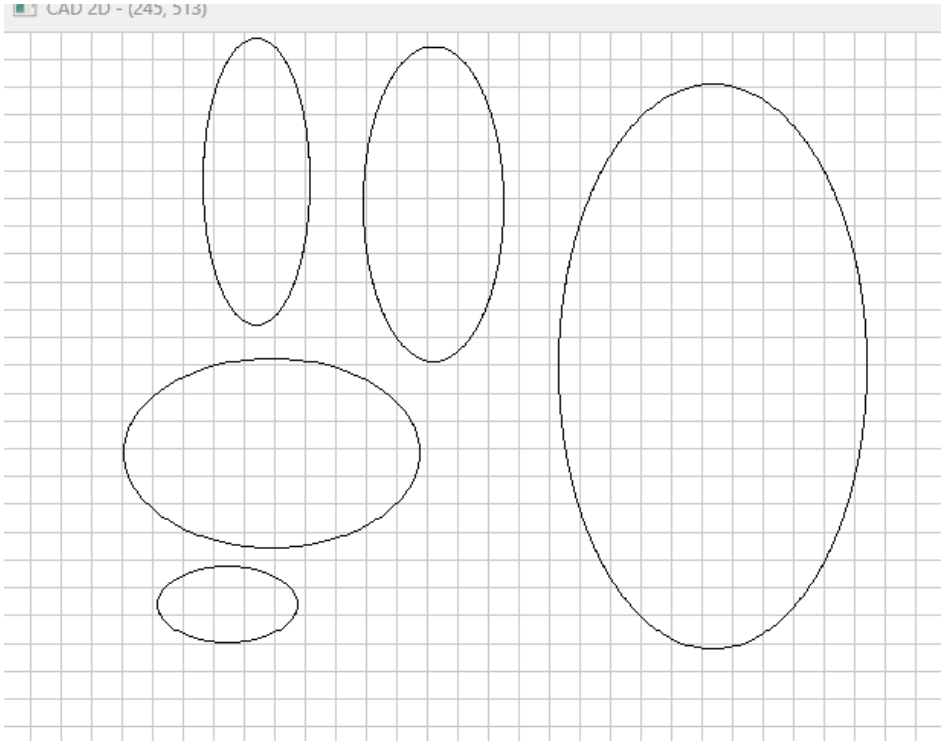
- Caso 2: Elipse horizontal (rx > ry)



- Caso 3: Elipse vertical (rx < ry)



- Caso 4: Elipse con diferentes relaciones de aspecto



VIII. CONCLUSIONES Y TRABAJO FUTURO

Conclusiones

- Se implementaron exitosamente los cuatro algoritmos de rasterización solicitados
- La interfaz gráfica resultó intuitiva y responsive
- El sistema maneja adecuadamente los eventos de usuario Los algoritmos producen resultados visualmente correctos
- El proyecto cumple con todos los requisitos funcionales

Trabajo Futuro

- Implementar curvas de Bézier y splines
- Agregar herramientas de edición (mover, rotar, escalar) Implementar capas y grupos de objetos Mejorar el sistema de exportación (PNG, SVG)
- Optimizar el rendimiento para figuras complejas
- Agregar patrones de relleno y texturas

BIBLIOGRAFÍA

- American Psychological Association. (2020). Publication manual of the American Psychological Association (7th ed.).
- Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1996).
- Computer Graphics: Principles and Practice. Addison-Wesley. Hearn, D., & Baker, M. P. (2004).
- Computer Graphics with OpenGL.
- Pearson Education. OpenGL Architecture Review Board. (2023). OpenGL Programming Guide. Addison-Wesley.