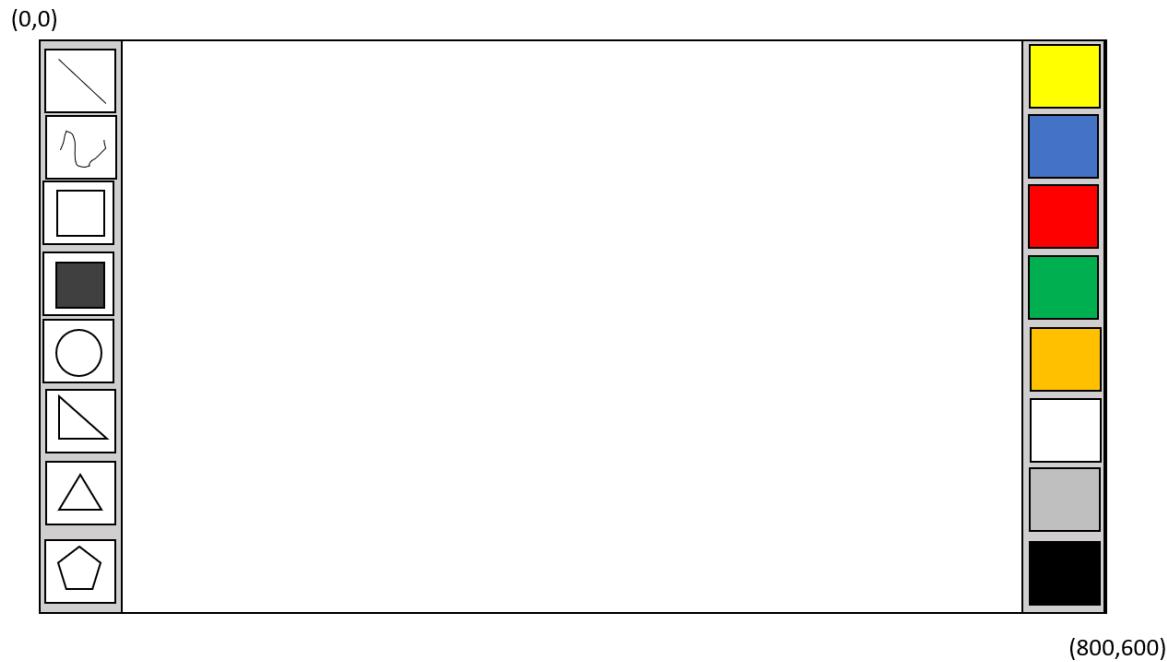


Proyecto - Graficador Interactivo en Python

Resultado de aprendizaje:	<ul style="list-style-type: none">• RA5. Identificar e implementar algoritmos que permiten el despliegue de modelos gráficos• RA6. Emplear, modelar e implementar primitivas de graficación• RA7. Identificar, comprender e implementar los principios de la animación utilizando herramientas computacionales• RA8. Identificar, modelar e implementar modelos de gestión de memoria adecuados para la correcta implementación de animaciones
Valor de calificación:	Entre 0.0 y 5.0
Tiempo estimado de desarrollo:	12 h
Dinámica de la actividad:	<ul style="list-style-type: none">• Trabajo individual o en pareja• Implementación de un graficador en 2D usando Pygame• Entrega del código fuente funcional y video demostrativo
Producto a entregar:	Aplicación funcional con menú interactivo, archivo .py, documentación y video de demostración
Documentos de Apoyo:	<ul style="list-style-type: none">• Documento guía: "<i>Algoritmos de Gráficos y Figuras Geométricas con Pygame</i>"• Documentación oficial de Pygame: https://www.pygame.org/docs/• Ejemplos y plantillas del repositorio de clase
Herramientas:	Python, Pygame
Instrumento de evaluación	Rubrica de evaluación para talleres
Fecha de apertura:	
Fecha de cierre:	

Descripción del Proyecto:



El proyecto consiste en el desarrollo de un graficador interactivo en 2D, orientado al aprendizaje visual de algoritmos gráficos fundamentales. El sistema debe incluir:

- **Ventana principal:** área de dibujo (lienzo blanco) y panel de control (barra lateral o superior) donde se ubiquen las opciones de figuras.
- **Opciones de usuario:** selección de figuras geométricas mediante teclado o botones, visualización inmediata del trazo, asignación de colores por tipo de figura y herramientas para limpiar o reiniciar el lienzo.
- **Figuras soportadas:**
 - **Líneas:** DDA y Bresenham
 - **Circunferencias:** Bresenham
 - **Curvas:** Bézier cúbica (con 4 puntos de control)
 - **Formas cerradas:** triángulos, rectángulos, polígonos y elipses

El énfasis estará en la implementación manual de los algoritmos, sin utilizar funciones gráficas predefinidas de pygame como `pygame.draw.line()`, asegurando así una comprensión profunda de los procedimientos de rasterización.

Objetivos:

Objetivo General

Diseñar e implementar una aplicación gráfica interactiva en Python, utilizando la biblioteca pygame, que permita a los estudiantes visualizar y manipular algoritmos clásicos de generación de figuras geométricas en dos dimensiones.

Objetivos Específicos

- Comprender e implementar algoritmos gráficos clásicos como DDA y Bresenham
- Modelar figuras 2D como líneas, circunferencias, elipses, polígonos y curvas Bézier
- Desarrollar una interfaz con interacción básica para la selección de figuras y colores
- Promover la estructuración del código por módulos con enfoque en buenas practices
- Entregar un proyecto funcional con documentación y demostración

Metodología:

El proyecto se desarrollará por fases: diseño de interfaz, implementación de algoritmos de rasterización, pruebas individuales y construcción del sistema final. Se promoverá el uso de clases y módulos separados (interfaz vs lógica de figuras) para mejorar el mantenimiento del código.

Recursos Necesarios:

- Python con biblioteca Pygame instalada
- Editor de código (VS Code, PyCharm, etc.)
- Acceso a recursos visuales de apoyo y ejemplos guía

Evaluación:

La evaluación se basará en la implementación precisa de algoritmos, calidad visual, modularidad del código, interfaz gráfica, documentación y claridad del video de demostración.

Requisitos del proyecto:

- Ventana principal con área de dibujo y panel de control
- Selección de figuras (línea, circunferencia, elipse, polígonos, curvas Bézier)
- Implementación manual de DDA, Bresenham (línea y circunferencia), curvas Bézier
- Visualización inmediata y color configurable
- Botones para limpiar lienzo y seleccionar figuras
- Modularización del código: lógica de dibujo separada de la interfaz
- Presentación en video funcional

Actividades y Entregables

- **Diseño del sistema:** bosquejo de la interfaz y funcionalidades del graficador.
- **Desarrollo algorítmico:** implementación de algoritmos gráficos en un módulo externo.
- **Construcción de la interfaz:** desarrollo visual con pygame y gestión de eventos de usuario.
- **Entrega del software:** código fuente completo, documentado y funcional.
- **Demostación del proyecto:** video corto mostrando la interacción con el graficador.

Rúbrica de Evaluación

Criterio	RAA Evaluado	Excelente (4.5 – 5.0)	Bueno (3.5 – 4.4)	Aceptable (2.5 – 3.4)	Insuficiente (0.0 – 2.4)
Implementación de algoritmos gráficos	RAA1, RAA2	Todos los algoritmos implementados manualmente y funcionales	La mayoría funcionan con detalles menores	Algunos algoritmos funcionan con errores	No funcionan o se usaron funciones gráficas predefinidas
Interfaz gráfica funcional	RAA3	Fluida, clara, permite selección de figuras y manipulación del lienzo	Interfaz funcional con elementos básicos	Interfaz básica o con fallas en interacción	Interfaz inoperativa o no implementada
Modularidad y estructura del código	RAA5	Código modular, bien comentado, separación entre lógica y GUI	Código con estructura clara, pero con redundancias	Código funcional pero con mezcla de responsabilidades	Código desorganizado o difícil de mantener
Documentación técnica del código	RAA5	Comentarios por función, manual de uso y guía	Comentarios básicos y guía parcial	Comentarios mínimos o documentación deficiente	Sin documentación técnica

		técnica incluidos			
Calidad de las figuras generadas	RAA2	Figuras bien definidas, renderizadas con precisión y en tiempo real	Figuras funcionales con detalles menores	Figuras con errores en forma o color	Figuras no se visualizan correctamente
Presentación del proyecto (video)	RAA6	Video explicativo claro, completo y bien estructurado	Video comprendible con detalles de edición	Video funcional pero poco claro o incompleto	Sin video o video no comprendible

Ponderación sugerida por criterio

Criterio **Peso (%)**

Implementación de algoritmos 25%

Interfaz gráfica 20%

Modularidad del código 15%

Documentación técnica 10%

Calidad visual de las figuras 15%

Presentación final del proyecto 15%

Total **100%**

Elaborado por:
FRANCISCO ALEJANDRO MEDINA