

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

GRAFICACIÓN

TAREA 1: GRAFICA DE UNA FUNCIÓN POLAR



Docente:
Prof. Iván Olmos Pineda

Alumno:
Jesús Huerta Aguilar

Matricula:
202041509

NRC: 10592
Sección: 001

CUARTO SEMESTRE

Puebla, Pue.

Fecha de entrega: 22/08/2022

INTRODUCCIÓN

Las funciones polares se basan en un sistema coordenado distinto; las coordenadas polares. Las coordenadas cartesianas se basan en dos componentes, una distancia horizontal y otra vertical. Las coordenadas polares se basan en un ángulo y una distancia llamada radio.

En este trabajo se mostrara la graficación de una función polar por medio del uso del IDE Code:Blocks, además de aplicar lo aprendido en clase para hacer variaciones tanto en forma como de color de la misma grafica.

CONCEPTOS DESARROLLADOS

La forma de determinar un punto del plano se describe mediante dos números: la distancia del punto al extremo de la semirrecta, llamado polo, y el ángulo que forma el eje polar (que es horizontal) con el segmento que une el punto con el polo, este ángulo debe medirse en sentido opuesto a las manecillas del reloj.

Este sistema de coordenadas puede resultar más conveniente en el cálculo de las ecuaciones de movimiento de varios sistemas mecánicos. Muchas veces tenemos objetos que se mueven en círculos y el uso de coordenadas polares puede simplificar las ecuaciones usadas.

ANÁLISIS EMPIRICO

Implementamos las librerías para poder graficar y operar las funciones trigonométricas en diferentes apartados.

```
#ifndef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>
#include <cmath>
GLfloat sizep;
int i;
```

Función para crear el espacio virtual.

Establecemos el color de la ventana de visualización donde los tres primeros parámetros corresponden al RGB y el cuarto parámetro corresponde al valor alfa que permite el efecto de transparencia (0 = totalmente transparente y 1 = totalmente opaco)

Establecemos los parámetros de proyección ortogonal y se visualizara una proyección ortogonal de dimensiones 400x400.

```
void init(void){

    glClearColor(0.0, 0.0, 0.0, 1.0);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-200,200,-200,200);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Función para dibujar la gráfica.

Definimos las variables a usar y después borramos la ventana de visualización. El primer punto debe de estar en 0° pero para calcular el diferencial tomamos la resolución la cual es de 720 y se divide sobre 360° ($360/720 = 0.5$) y este será el incremento que tomara el ángulo en cada paso del ciclo for, además, dentro del ciclo se ejecutarán las operaciones necesarias para pasar los valores polares a cartesianos y además de asignar los colores por punto a imprimir.

```
void dibujaGrafica(){
    float x,y,rpolar,angulo,r,g,b;
    glClear(GL_COLOR_BUFFER_BIT);
    angulo = 0;
    glPointSize(1);

    for (i=0;i<720;i++){
        rpolar = 5*cos(4*angulo);
        x = (rpolar*cos(angulo))*(25);
        y = (rpolar*sin(angulo))*(25);

        if(i <= 240){
            r = 1.0;
            g = 0.0;
            b = 0.0;
        }
        else if((i > 240) && (i <= 480)){
            r = 0.0;
            g = 1.0;
            b = 0.0;
        }
        else if((i > 480) && (i <= 720)){
            r = 0.0;
            g = 0.0;
            b = 1.0;
        }
        glColor3f(r, g, b);
        glBegin(GL_POINTS);
            glVertex2i(x,y);
        glEnd();
        angulo = angulo + 0.5;
    }
    glFlush();
}
```

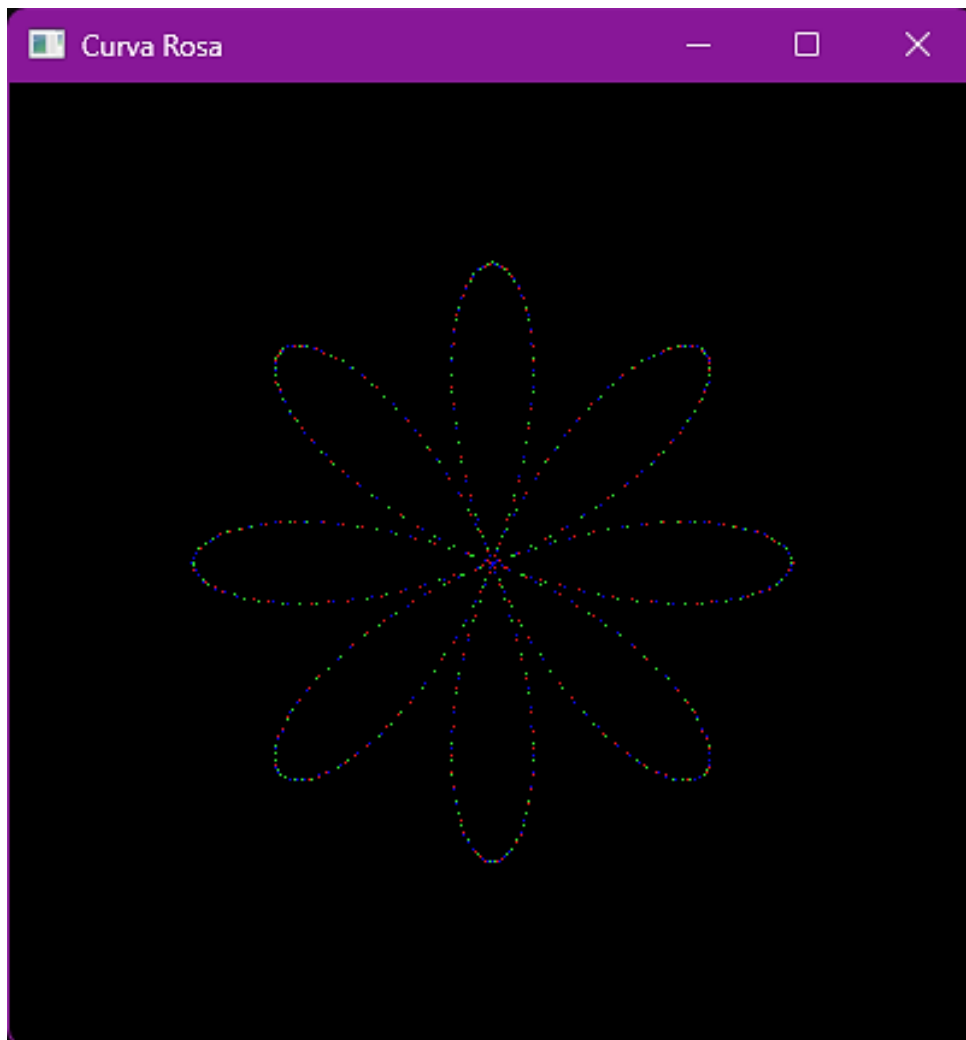
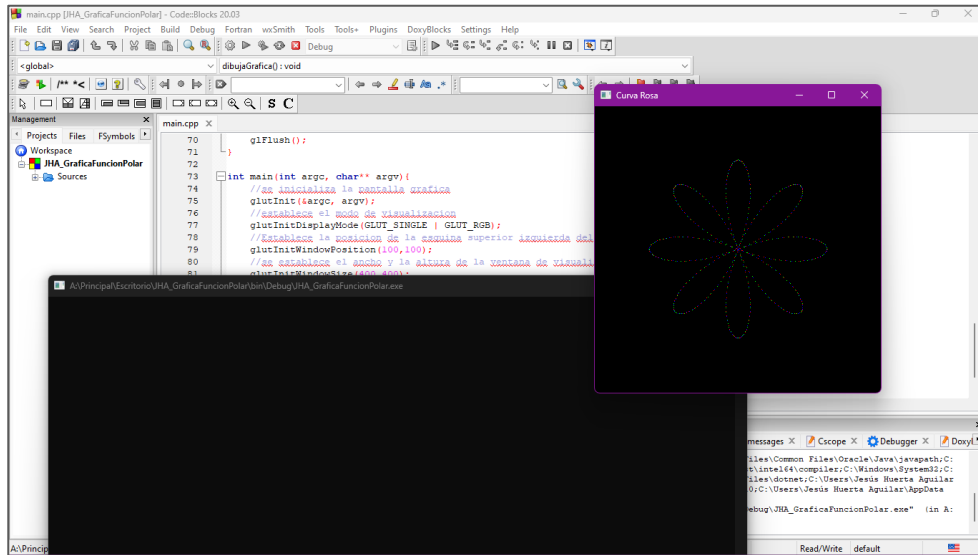
Función main.

Llamamos a las funciones e instrucciones necesarias para inicializar la pantalla grafica y establecer el modo de visualización, establecemos la posición en la esquina superior izquierda y el ancho y altura de la ventana de visualización además de crearla, ejecutamos la función de iniciación de parámetros y enviamos los gráficos a pantalla.

```
int main(int argc, char** argv){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(400,400);
    glutCreateWindow("Curva Rosa");
    init();
    glutDisplayFunc(dibujaGrafica);
    glutMainLoop();

    return 0;
}
```

EJECUCIONES



CONCLUSIONES

Durante la elaboración de este trabajo, se comprendió el uso de las instrucciones de OpenGL para la impresión de una grafica polar, poniendo a prueba los conocimientos obtenidos en las clases del profesor además de conocimiento general de transformaciones y trigonometría para la obtención de los puntos en el plano cartesiano, además de probar el cambio de colores en los puntos a imprimir para lograr un efecto arcoíris en el mismo, muchos aprendizajes de esta practica pueden ser trasladados a futuros proyectos para así, facilitar nuestro proceso de aprendizaje y el proceso de elaboración de estos trabajos.

Dada una ecuación en coordenadas polares de la forma $r = f(\theta)$ se puede obtener la gráfica, generando una tabla de valores para ciertos valores del ángulo y obtener su respectivo valor de r . Se forma la coordenada (r, θ) , se ubica en el sistema polar y finalmente se traza la gráfica siguiendo los puntos.

BIBLIOGRAFIA

- Olmos Pineda, I. (2022, 18 agosto). Reunión en «General». sharepoint. Recuperado 21 de agosto de 2022, de https://correobuap.sharepoint.com/sites/Section_202235-CCOS261-10592-001/_layouts/15/stream.aspx?id=%2Fsites%2FSection_202235-CCOS261-10592-001%2FShared%20Documents%2FGeneral%2FRecordings%2FReunión%20en%20_General_-20220818_070743-Grabación%20de%20la%20reunión%20Emp4
- Universidad Nacional Autónoma de México. (s. f.). Curvas Notables en Coordenadas Polares. Pagina del Colegio de Matemáticas. Recuperado 21 de agosto de 2022, de http://prepa8.unam.mx/academia/colegios/matematicas/paginacolmate/applets/matematicas_V/Applets_Geogebra//coordpolares.html