

```

#include<GL/glut.h>
#include<stdlib.h>
#include<math.h>

typedef struct {
    float dominioMin, dominioMax;
    float rangoMin, rangoMax;
} Funcion;

Funcion funciones[3];
float mundo;
int opcion;
int ejes;
float anchoVentana;
float altoVentana;

void redimensiona (int ancho, int alto);
void dibuja (void);
void inicializar (void);
void teclado (unsigned char key, int x, int y);
void tecladoEspecial (int key, int x, int y);
float evalua (float x);

int main(int argc, char **argv) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Graficador en 2D");
    inicializar();

    glutDisplayFunc(dibuja);
    glutReshapeFunc(redimensiona);
    glutKeyboardFunc(teclado);
    glutSpecialFunc(tecladoEspecial);
    glutMainLoop();

    return 0;
}

void redimensiona (int ancho, int alto) {

    float aspectRadio;
    float maxX, maxY;

    if (alto == 0) {
        alto = 1;
    }

    glViewport(0, 0, ancho, alto);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    aspectRadio = (float) ancho / (float) alto;

    if (ancho <= alto) {
        maxX = mundo;
        maxY = mundo / aspectRadio;
    } else {
        maxX = mundo * aspectRadio;
        maxY = mundo;
    }

    glOrtho(-maxX, maxX, -maxY, maxY, -mundo, mundo);
    altoVentana = alto;
    anchoVentana = ancho;
}

void dibuja (void) {
    float delta;
    float x;
    int i;

    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0f, 1.0f, 0.0f);

    delta = (funciones[opcion].dominioMax - funciones[opcion].dominioMin) / 100.0f;

    glBegin(GL_LINE_STRIP);

```

```

for (i = 0; i <= 100; i++)
{
    x = funciones[opcion].dominioMin + (i * delta);
    glVertex3f(x, evalua(x), 0.0);
}
glEnd();

if (ejes)
{
    glBegin(GL_LINES);
    glColor3f(1.0, 0.0, 0.0);
    glVertex3f(-mundo, 0.0, 0.0);
    glVertex3f(mundo, 0.0, 0.0);
    glColor3f(0.0, 1.0, 0.0);
    glVertex3f(0.0, -mundo, 0.0);
    glVertex3f(0.0, mundo, 0.0);
    glEnd();
}

glFlush();
}

void inicializar (void) {
    glClearColor(0.0, 0.0, 0.0, 0.0);
    opcion = 0;
    mundo = 4.0;
    ejes = 0;

    //Iniciamos la primer funcion sen(x)
    funciones[0].dominioMin = -4.0;
    funciones[0].dominioMax = 4.0;
    funciones[0].rangoMax = 1.5;
    funciones[0].rangoMin = -1.5;

    //Iniciamos la segunda funcion x^2
    funciones[1].dominioMin = -10.0;
    funciones[1].dominioMax = 10.0;
    funciones[1].rangoMax = 10.0;
    funciones[1].rangoMin = -10.0;

    //Iniciamos la segunda funcion cos(x)
    funciones[2].dominioMin = -10.0;
    funciones[2].dominioMax = 10.0;
    funciones[2].rangoMax = 1.0;
    funciones[2].rangoMin = -1.0;
}

float evalua (float x) {
    float y;

    switch (opcion)
    {
        case 0: y = sin(x); break;
        case 1: y = x * x; break;
        case 2: y = cos(x); break;
        default: break;
    }

    return y;
}

void teclado (unsigned char key, int x, int y) {
    switch (key)
    {
        case 'e':
        case 'E':
            ejes = !ejes;
            break;

        case 27:
            exit(0);
            break;

        default: break;
    }
    glutPostRedisplay();
}

void tecladoEspecial (int key, int x, int y) {

```

```
float mayor;
switch(key) {
    case GLUT_KEY_F1:
        opcion = 0;
        break;

    case GLUT_KEY_F2:
        opcion = 1;
        break;

    case GLUT_KEY_F3:
        opcion = 2;
        break;
}

mayor = fabs(funciones[opcion].dominioMin);

if (mayor < fabs(funciones[opcion].dominioMax)) {
    mayor = fabs(funciones[opcion].dominioMax);
}

if (mayor < fabs(funciones[opcion].rangoMin)) {
    mayor = fabs(funciones[opcion].rangoMin);
}

if (mayor < fabs(funciones[opcion].rangoMax)) {
    mayor = fabs(funciones[opcion].rangoMax);
}

mundo = mayor;
redimensiona(anchoVentana, altoVentana);
glutPostRedisplay();
}
```