

```

// PRESENTADO POR JOAN SEBASTIAN TIBAKUIRA COD 1202060
#include <iostream>
#include <GL\glut.h> //INCLUYE LA LIBRERIA OPEN GRAPHICS LIBRARY

using namespace std;
void Display() {
    glClear(GL_COLOR_BUFFER_BIT); //BORRAR BUFFER
    glFlush(); //VACIA EL BUFFER
}
void Pixel(int x, int y) { //RECIBE LAS COORDENADAS DEL USUARIO
    glBegin(GL_POINTS); // VERTICES COMO PUNTOS
    glColor3f(1, 0, 0); //COLOR RGB DEL PIXEL
    glVertex2f(x, y); //PERMITE VISUALIZAR EL PIXEL
    glEnd(); //FINALIZA EL PROGRAMA
    glFlush();
    glPointSize(1); //ADECUA EL TAMAÑO DEL PIXEL
}
void DDisplay() {
    glClearColor(255, 255, 255, 0);
    glColor3f(255, 255, 000);
    gluOrtho2D(-400, 400, -400, 400);
}
void Grafica() {
    int x, y, a, R1, R2;

    float A = 0, P = 0;
    cout << "DIGITE EL RANGO EN EL QUE LA QUIERE DIBUJAR (DIGITE UN NUMERO ENTRE -400
Y 400)" << endl;
    cin >> R1;
    cout << "DIGITE EL RANGO EN EL QUE LA QUIERE DIBUJAR (DIGITE UN NUMERO ENTRE -400
Y 400)" << endl;
    cin >> R2;
    cout << "DIGITE LA AMPLITUD DE LA FUNCION" << endl;
    cin >> A;
    cout << "DIGITE EL PERIODO DE LA FUNCION" << endl;
    cin >> P;
    for (float x = R1; x < R2; x++) {
        x = x - 0.9;
        y = A * sin(x / P);
        Pixel(x, y);
    }
}
void Plano() {
    for (int x = -400; x < 400; x++) {
        Pixel(0, x);
    }
    for (int y = -400; y < 400; y++) {
        Pixel(y, 0);
    }
}
void Mouse(int b, int e, int x, int y) {
    if ((e == GLUT_DOWN) && (b == GLUT_RIGHT_BUTTON)) {
        Plano();
        Grafica();
        Pixel(0, 200);
    }
    if ((e == GLUT_DOWN) && (b == GLUT_LEFT_BUTTON)) {

```

```

        exit(0);
    }
}
int main(int argc, char** argv) {
    glutInit(&argc, argv); // INICIALIZACION DE PROGRAMAS GLUT
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); // INDICA LA UTILIZACION DE DOBLE
    BUFFER Y RGB
    glutInitWindowPosition(50,50); // UBICACION DE LA VENTANA
    glutInitWindowSize(800,800); // ASIGNA EL TAMAÑO DE LA VENTANA
    glutCreateWindow("PIXEL"); // CREACION DE VENTANA Y ASIGNACION DE NOMBRE
    glutMouseFunc(Mouse);
    glutDisplayFunc(Display); // REGISTRA LA FUNCION DEL REDIBUJADO
    DDdisplay();
    glutMainLoop(); // REALIZA UN BUCLE INFINITO
    system("pause");
}

```