

## Código 19: Superficie de Bezier

```
#include <GL/glut.h>

GLfloat ptosctl[4][4][3] = {
    {{-8, 0, 8}, {-2, -7.5, 8.0}, {2, -7.5, 8}, {8, 0, 8}},
    {{-9, 0, 2}, {-2, -7.5, 2.0}, {2, -7.5, 2}, {9, 0, 2}},
    {{-9, 0, -2}, {-2, 4.5, -2.0}, {2, 4.5, -2}, {9, 0, -2}},
    {{-8, 0, -8}, {-2, 4.5, -8.0}, {2, 4.5, -8}, {8, 0, -8}}};

void inicializa(void)
{
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glMap2f(GL_MAP2_VERTEX_3, 0, 1, 3, 4, 0, 1, 12, 4, &ptosctl[0][0][0]);
    glEnable(GL_MAP2_VERTEX_3);
}

void dibuja(void)
{
    int i,j;

    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    glPushMatrix();
    glRotatef(30.0, 1.0, 0.0, 0.0);
    // Superficie de Bezier
    glLineWidth(2);
    glMapGrid2f(20,0,1,10,0,1);
    glEvalMesh2(GL_LINE,0,20,0,10);
    // Puntos de control
    glPointSize(7.0);
    glColor3f(0.0, 0.0, 1.0);
    glBegin(GL_POINTS);
        for (i = 0; i < 4; i++)
            for (j = 0; j < 4; j++)
                glVertex3fv(&ptosctl[i][j][0]);
    glEnd();
    glPopMatrix();
    glFlush();
}

void reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-12.0, 12.0, -12.0, 12.0, -12.0, 12.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(10,10);
    glutCreateWindow ("Superficie de Bezier");
    inicializa();
    glutReshapeFunc(reshape);
    glutDisplayFunc(dibuja);
    glutMainLoop();
    return 0;
}
```

