

Código 15: ColorMaterial (frag.)

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

GLboolean wire = GL_FALSE;

/* ----- Rotacion y Zoom ----- */
void mover (void)
{
    ...
}

/* ----- Funciones con Mouse y Teclas ----- */
void keyboard( unsigned char key, int x, int y) {
    switch(key) {
        case 27:  exit(0);
                 break;
        case 'i': zoom += 1;    //zoom in
                 break;
        case 'o': zoom -= 1;    //zoom out
                 break;
        case '1': wire = GL_TRUE;
                 break;
        case '2': wire = GL_FALSE;
                 glEnable(GL_LIGHTING);
                 glEnable(GL_LIGHT0);
                 glEnable(GL_COLOR_MATERIAL);
                 break;
        case '3':
                 glEnable(GL_LIGHTING);
                 glEnable(GL_LIGHT0);
                 glDisable(GL_COLOR_MATERIAL);
                 break;
        case '4':
                 glEnable(GL_LIGHTING);
                 glEnable(GL_LIGHT0);
                 glEnable(GL_COLOR_MATERIAL);
        default:break;
    }
    glutPostRedisplay();
}

void flechas(int key, int x, int y) {
    ...
}

//----- Dibuja -----
void lampara(void){
    glTranslatef(0, -42, -150);
    mover();
    glPushMatrix(); //Base
        glScalef(1, 0.2, 1);
        glColor3f(0, 0.5, .0);
        if (wire) glutWireSphere(20, 30, 30);
        else glutSolidSphere(20, 30, 30);
    glPopMatrix(); //Fin Base

    glPushMatrix(); //Soporte
        glTranslatef(0, 10, 0);
        glScalef(3, 10, 3);
        glColor3f(0, 0, 1);
        if (wire) glutWireCube(2);
        else glutSolidCube(2);
    glPopMatrix(); //Fin Soporte
    ...
}

glPopMatrix();
}

void Dibuja(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glPushMatrix();
        lampara();
    glPopMatrix();

    glutSwapBuffers();
}
```

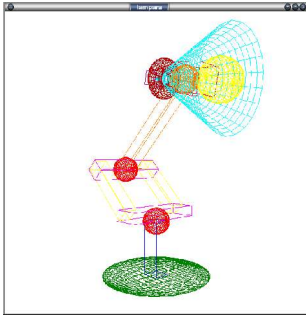
```

/* ----- Reshape ----- */
void reshape(int width, int height)
{
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45, (float)width/height, 50, 500);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

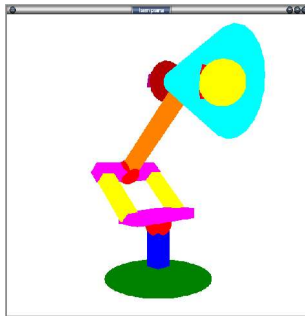
    glClearColor(1.0,1.0,1.0,0.0);
    glEnable(GL_DEPTH_TEST);
}

//----- main -----
int main(int argc, char** argv)
{
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);
    glutInitWindowPosition(50, 50);
    glutInitWindowSize(600,600);
    glutInit(&argc, argv);
    glutCreateWindow("lampara");
    glutDisplayFunc(Dibuja);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutSpecialFunc(flechas);
    glutMainLoop();
    return 0;
}

```



Wireframe



Sólido



Iluminado

