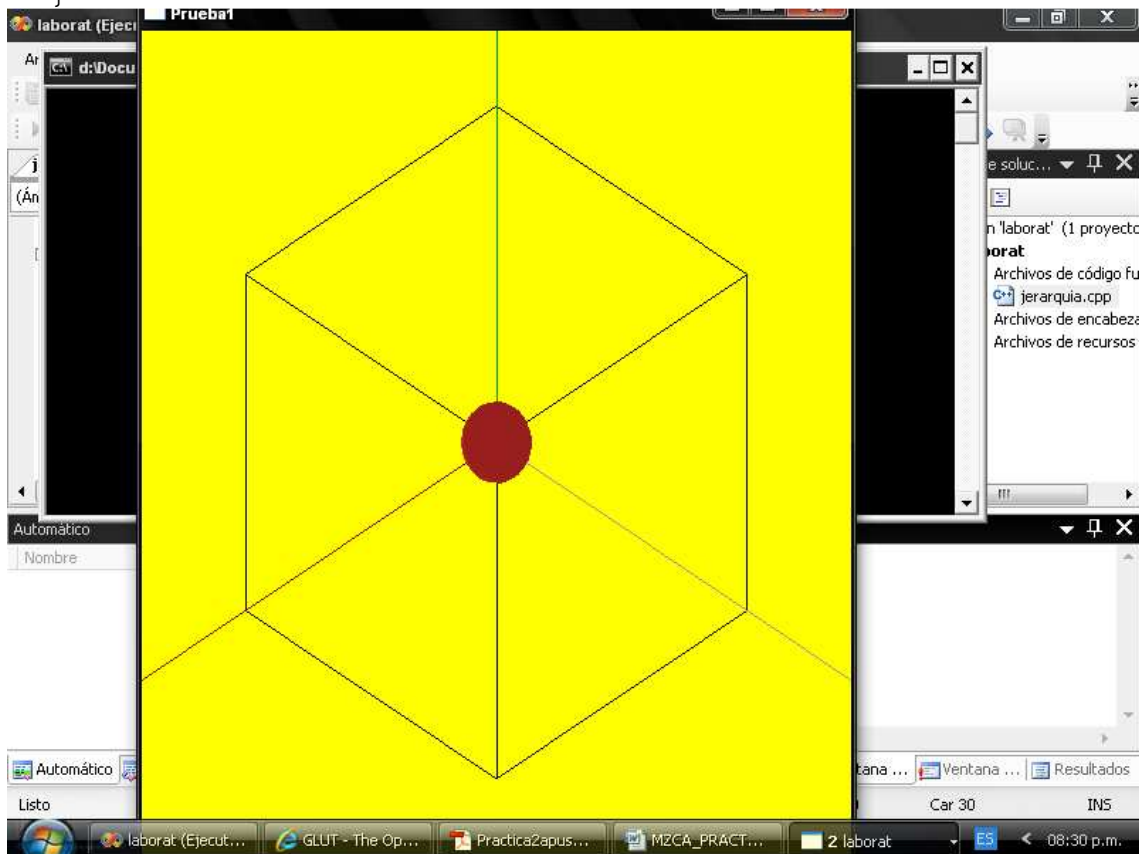


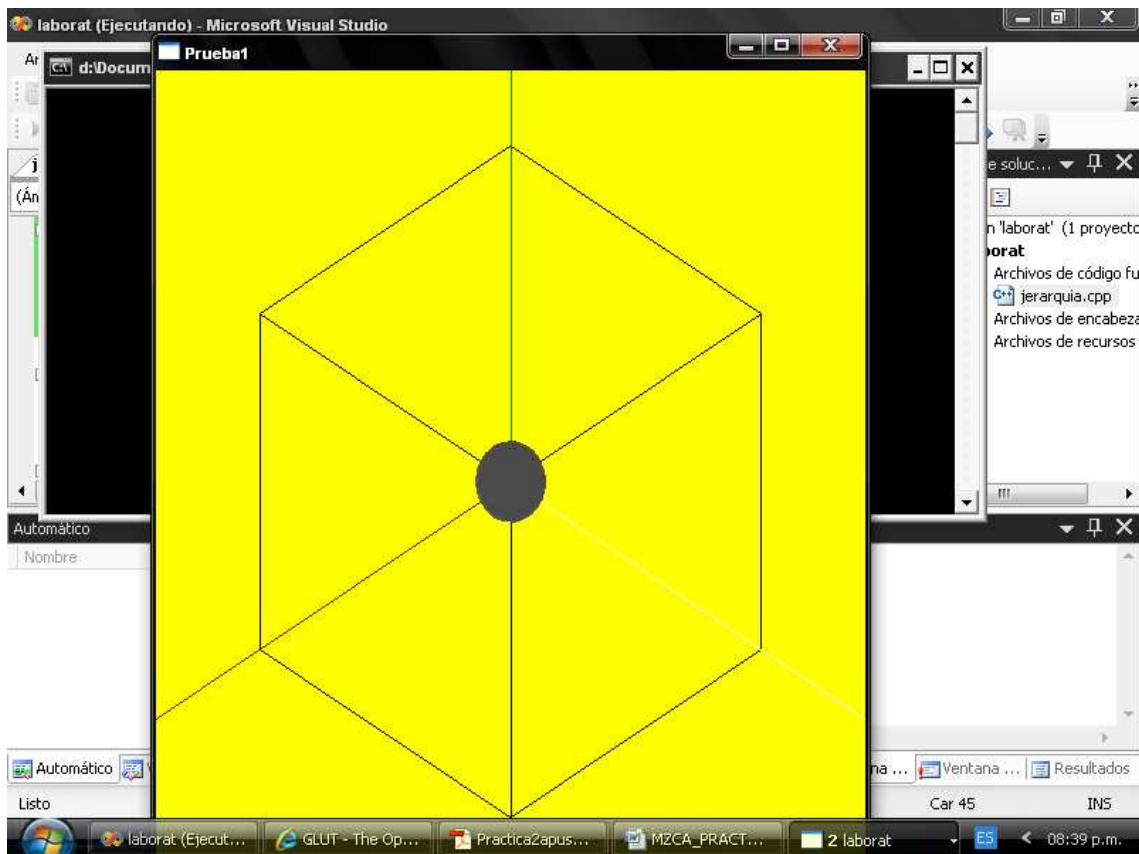
1.- Dibujar esfera:

```
void jerarquia(void)
{
    glutSolidSphere(0.5f,15,15);
}
```



2.- Cambiar el color:

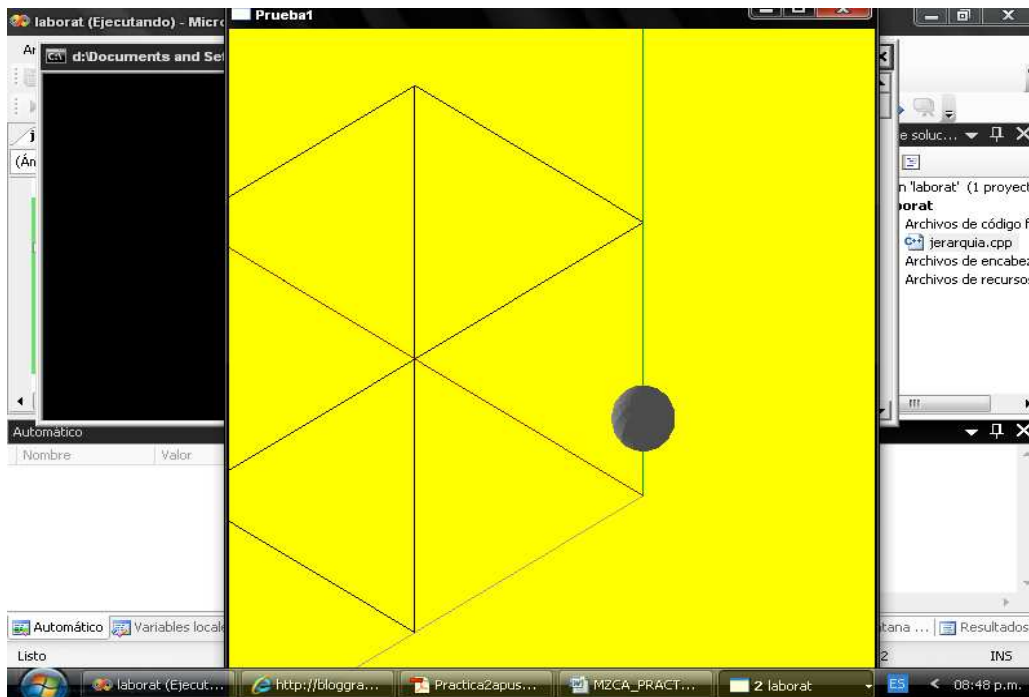
```
void jerarquia(void)
{
    glMaterialfv(GL_FRONT,GL_AMBIENT_AND_DIFFUSE,gris);
    glutSolidSphere(0.5f,15,15);
}
```



3.- Transformaciones de inicio:

- a) Traslade la esfera 2 unidades en z
- b) Rote 45 grados en Y

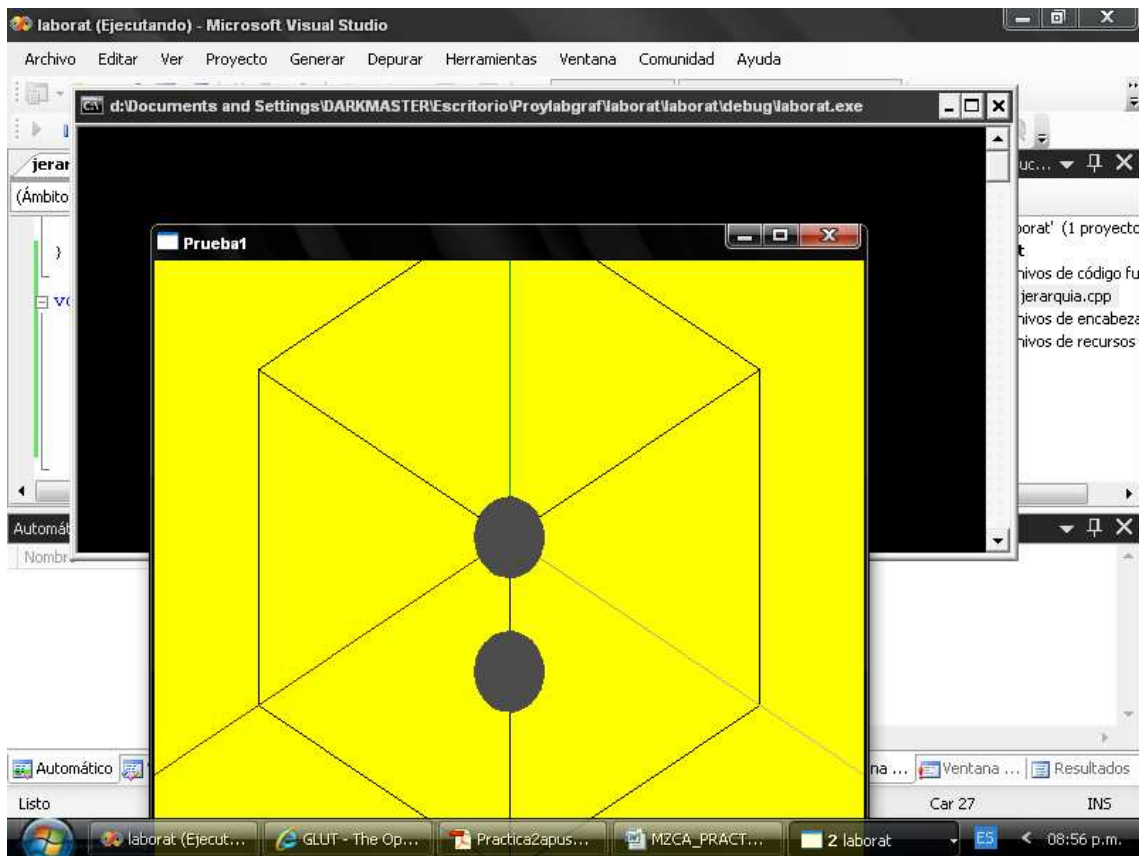
```
void jerarquia(void)
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glutSolidSphere(0.5f, 15, 15);
    glTranslatef(0, 0, 2);
    glRotatef(45, 0, 1, 0);
}
```



Modelado jerárquico 1

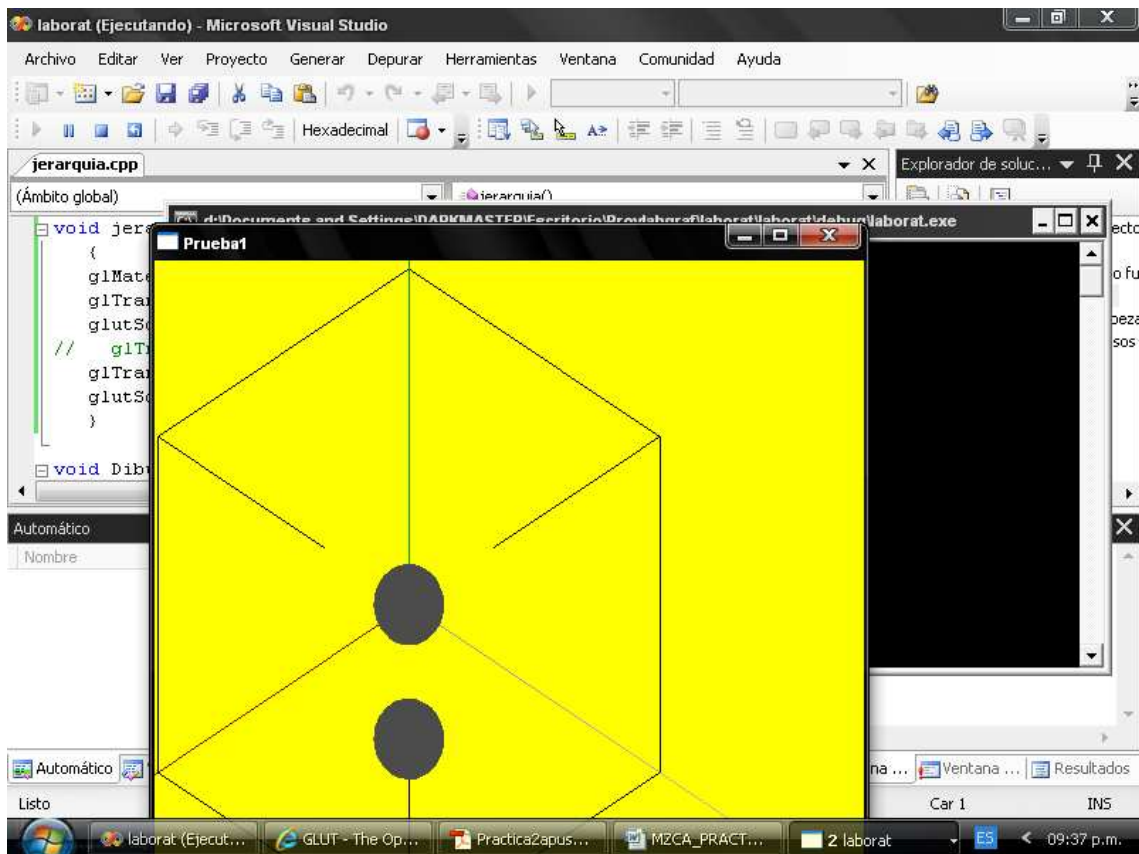
- a) Preparar dos esferas, dibujar esfera 1 en origen de coordenadas (0,0,0)
- b) Dibuje esfera 2 en posicion2 unidades mas desplazadas en y.

```
void jerarquia(void)
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glutSolidSphere(0.5f, 15, 15);
    glTranslatef(0, 2, 0);
    glutSolidSphere(0.5f, 15, 15);
}
```



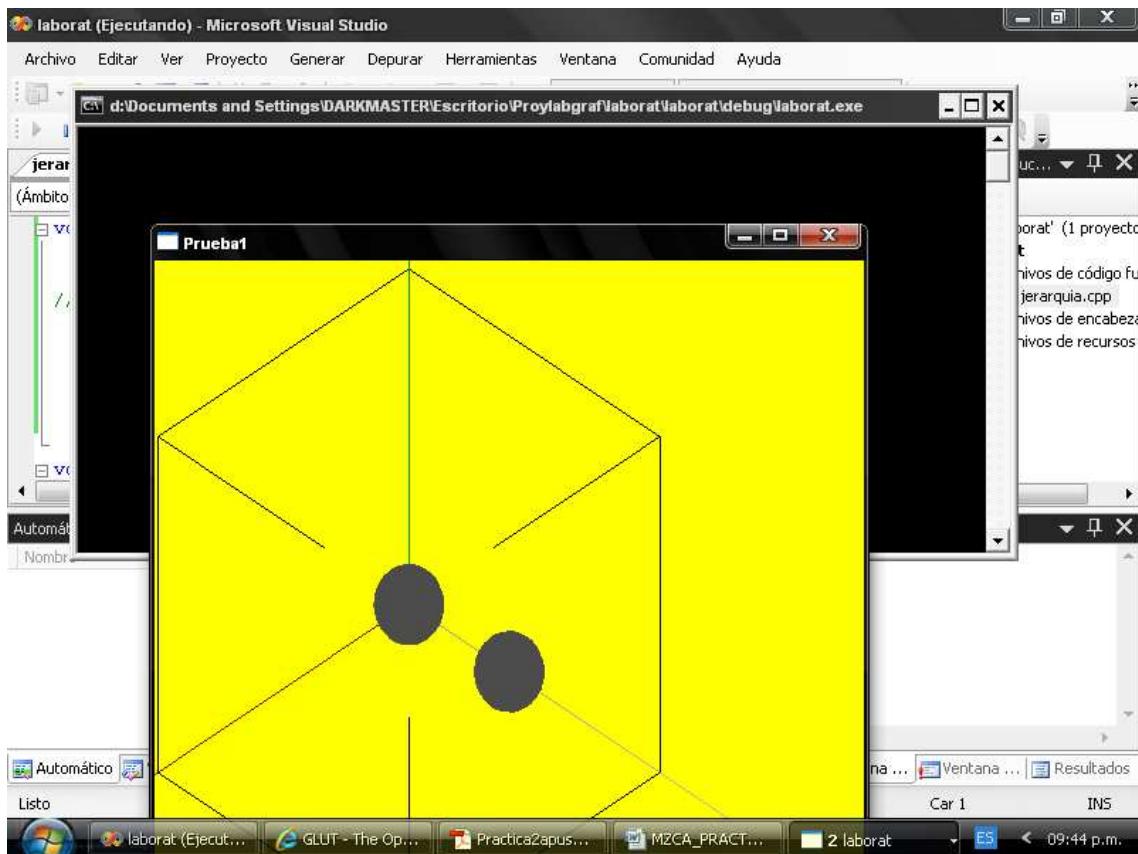
5.- Pila general

a) Traslade la esfera 1 dos unidades en X. Observe y anote que pasa, regrésela a su posición



b) Traslade la esfera 2 dos unidades en x, observe y anote que pasa, regrésela a su posición:

```
void jerarquia(void)
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    // glTranslatef(2,0,0);
    glutSolidSphere(0.5f, 15, 15);
    glTranslatef(2,0,0);
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f, 15, 15);
}
```

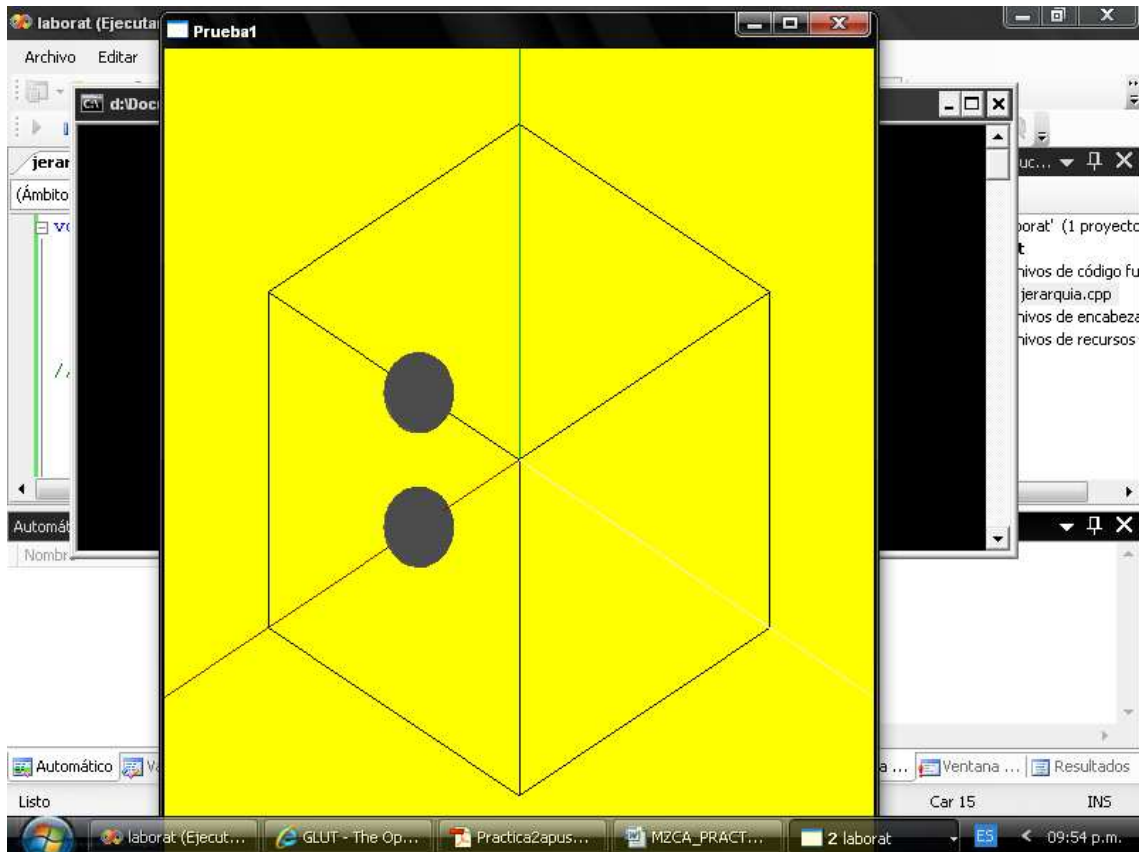


El objeto que se mueve y mueve a otros con el se llamara "Padre o Raiz" y los demás objetos que se mueven y no jalan a otros objetos se llaman "Hijos u hojas". En este caso, ¿ la esfera 1 es padre o hijo? , ¿y la esfera 2?: la esfera 1: padre o raíz, esfera 2: hijo u hoja.

6.- Pila push-pop. Manejo de pilas

a) General: coloque los nuevos comandos push y pop entre las transformaciones y la esfera

```
void jerarquia(void)
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glPushMatrix();
    glTranslatef(2,0,0);
    glutSolidSphere(0.5f, 15, 15);
    // glTranslatef(2,0,0);
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f, 15, 15);
    glPopMatrix();
}
```

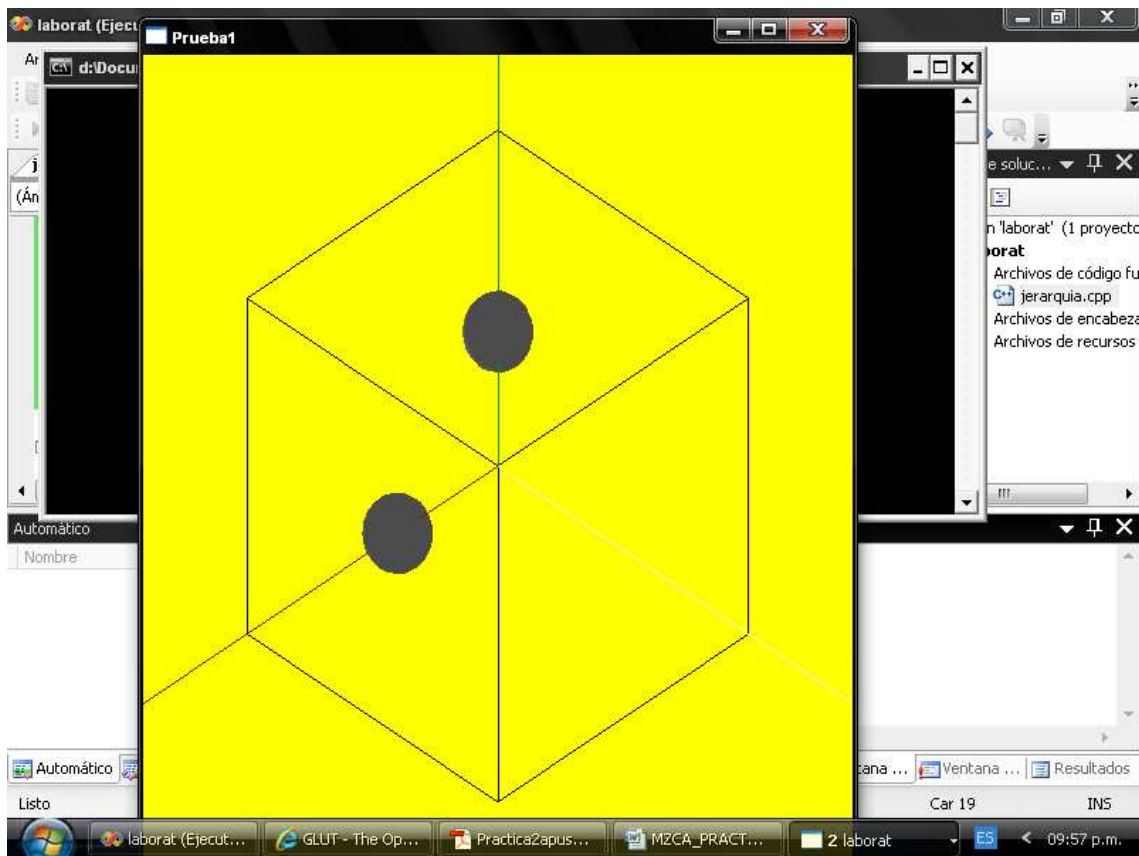


¿Hubo cambio o no?

En mi caso si, ya que al utilizar estas funciones la imagen se dibuja correctamente

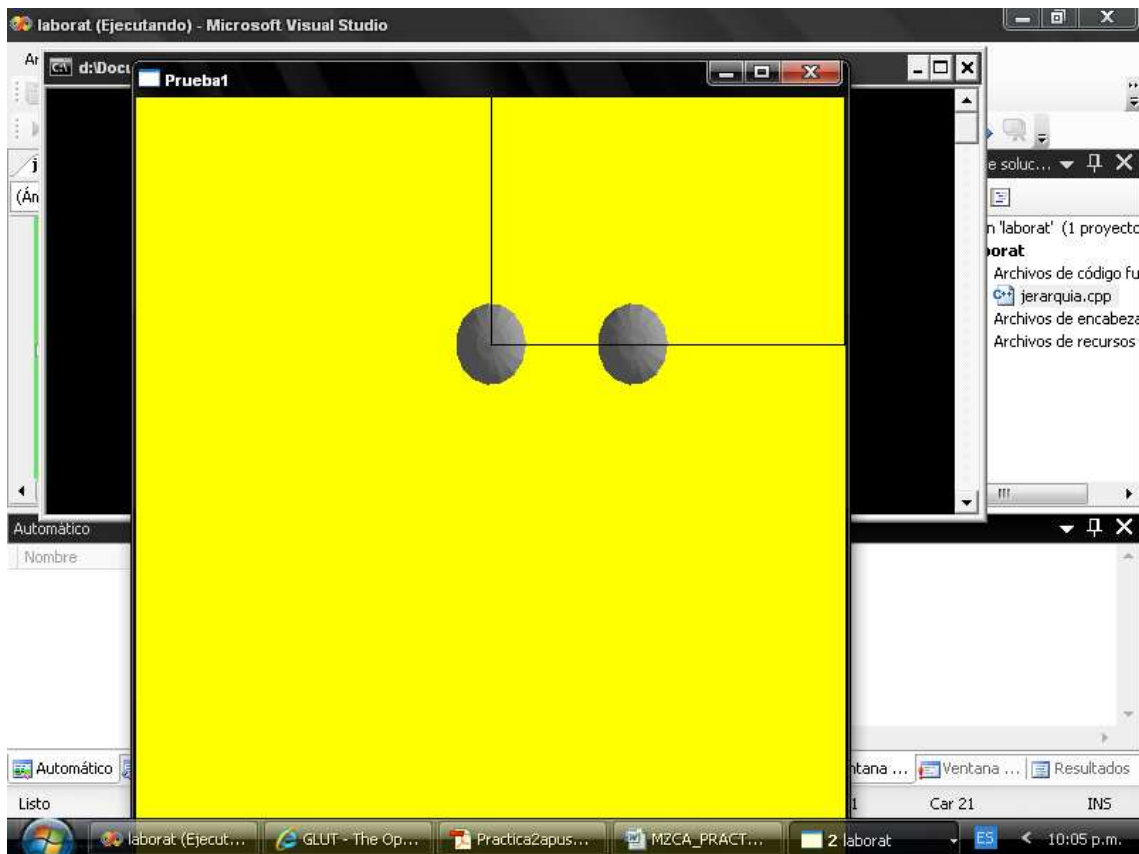
b) Particular: coloque el par Push y Pop para cada esfera:

```
void jerarquia(void)
{
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, gris);
    glPushMatrix();
    glTranslatef(2,0,0);
    glutSolidSphere(0.5f, 15, 15);
    glPopMatrix();
    glPushMatrix();
    // glTranslatef(2,0,0);
    glTranslatef(0,2,0);
    glutSolidSphere(0.5f, 15, 15);
    glPopMatrix();
}
```



¿Hubo cambio o no?

Si, la esfera 1 deja de influir en la esfera 2



Conclusiones:

En esta práctica fue posible apreciar el funcionamiento de las funciones `pushmatrix` y `popmatrix` y como afectan al resultado final, también aprendimos a rotar y a trasladar objetos mediante el uso de funciones predefinidas de OpenGL.