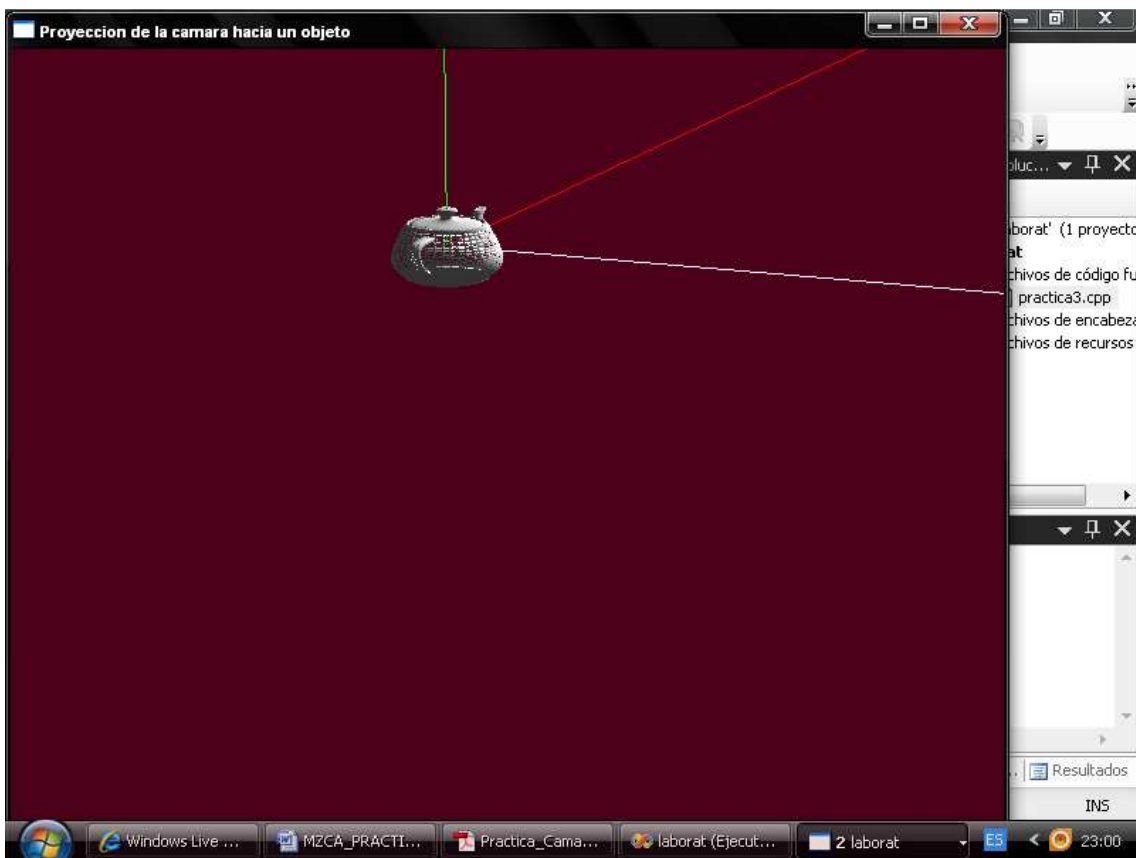


### Cámara

- a) En Word e impresiones de pantalla con explicaciones, presente una vista de la tetera, viendo desde 14 unidades atrás, desde donde sea necesario.

Para obtener esta vista simplemente movemos la cámara 14 unidades hacia atrás en dirección x para poder ver la tetera desde atrás:

```
gluLookAt(-14,7,7,3,0,0,0,1,0);
```



- b) Presente una vista, a unas 5 unidades lejos de la tetera, viendo desde 14 unidades atrás, desde donde sea necesario

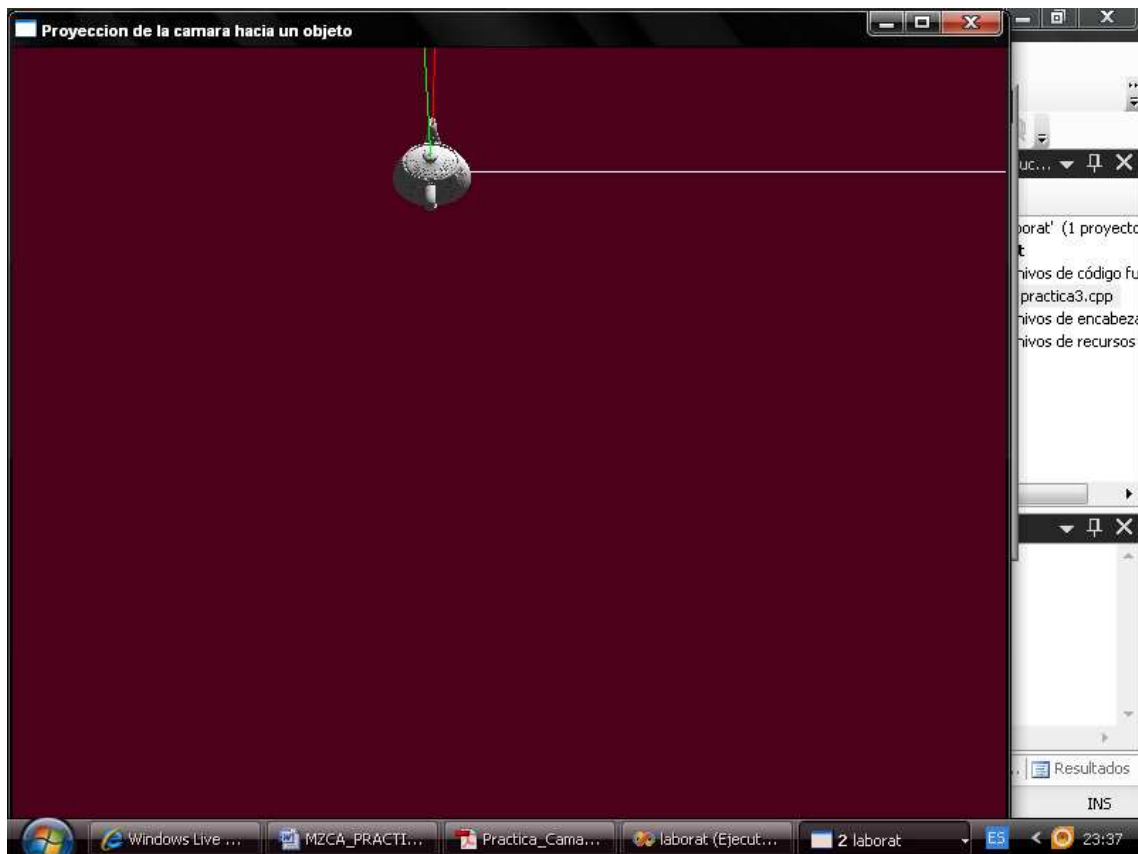
```
gluLookAt(-14,7,7,5,5,0,0,1,0);
```



c) Con esta misma vista anterior, coloque un punto de vista desde arriba

Para poder obtener esta vista simplemente colocamos la cámara por encima de la tetera, es decir, modificamos el valor de  $y$ .

```
gluLookAt(-14,25,0,0,0,0,0,0,1,0);
```



## Luces

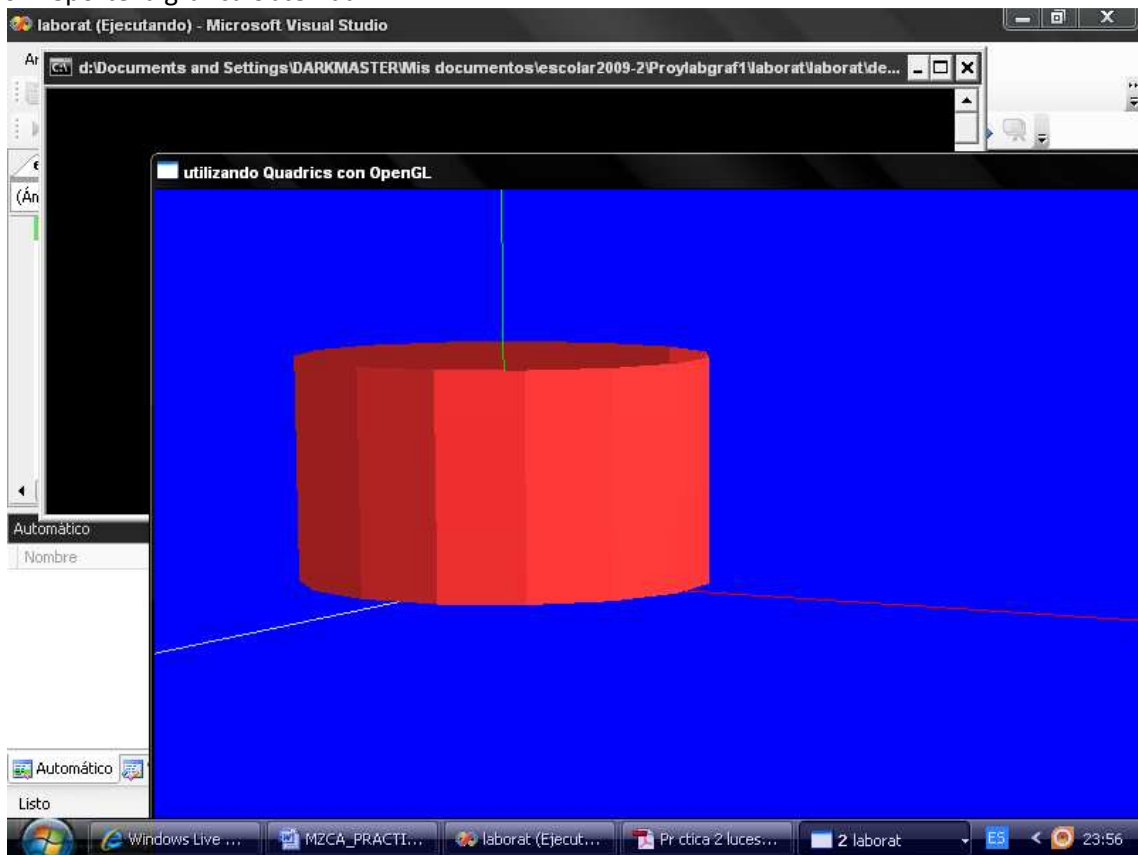
### Ejercicio 1:

- 1.- Vamos a crear un nuevo proyecto dentro de la carpeta "practicas\_gl" Asi que presione en la secuencia menú "File->New Project" o simplemente presione simultáneamente las teclas "ctrl+shift+N", el nuevo proyecto se llamara "ejer04".
- 2.- Localice el archivo "ejemplo\_luces\_1.cpp" y cópielo de su origen a la carpeta "...\\practicas\_gl\\ejer04"
- 3.- Presione la secuencia menú "Project->Add Existing\_item" o simplemente presione la combinación de teclas "Shift+Alt+A".
- 4.- El programa consta de una sección de
  - a. Variables globales para definir luces y materiales
  - b. Subrutina "luces()" que habilita las ecuaciones de luz de oOpenGL
  - c. Subrutina "EjesReferencia()" que dibuja los eje4s del sistema de referencia.
  - d. Subrutina "Cilindro()" que dibuja un cilindro (es posible aún no entienda el código".
  - e. Subrutina "RenderScene()" que ordena invoca a las subrutinas anteriores para dibujar la escena.

f. Subrutina "main()" que prepara la ventana Windows para OpenGL.

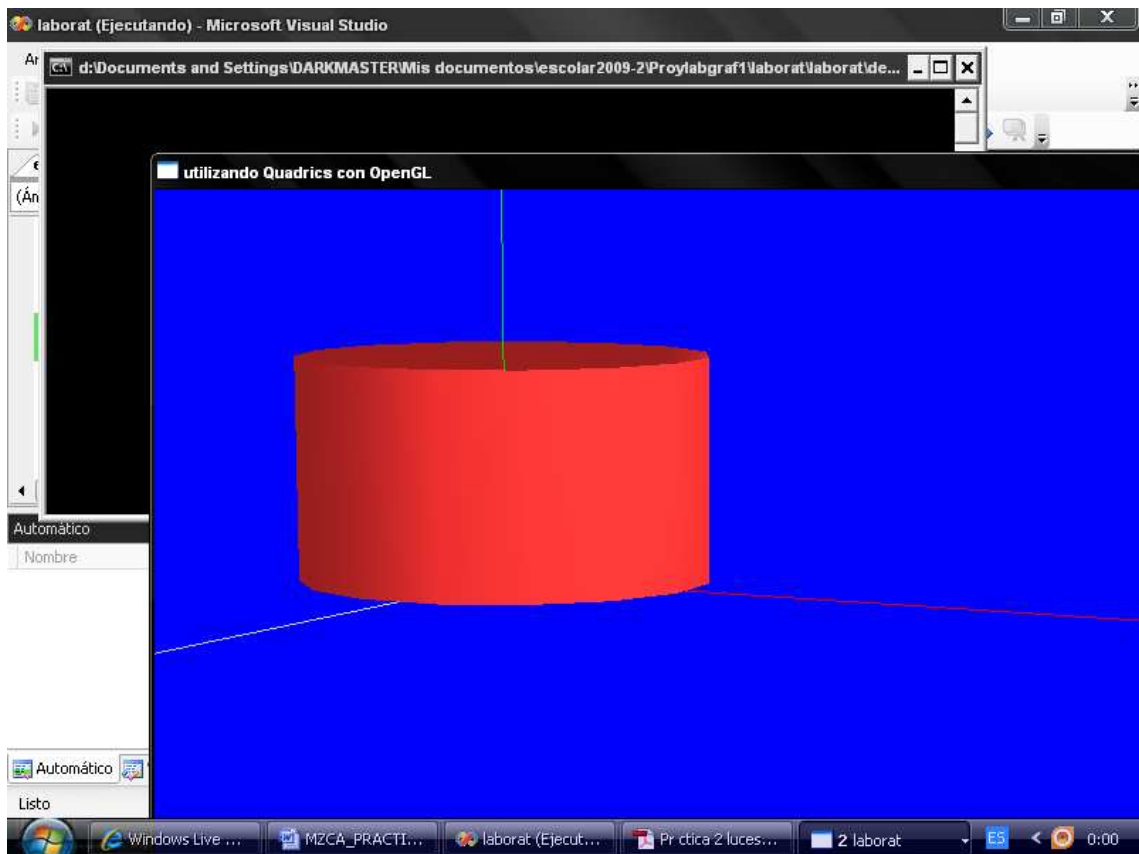
5.- Construya y ejecute

6.-Reporte la grafica obtenida:



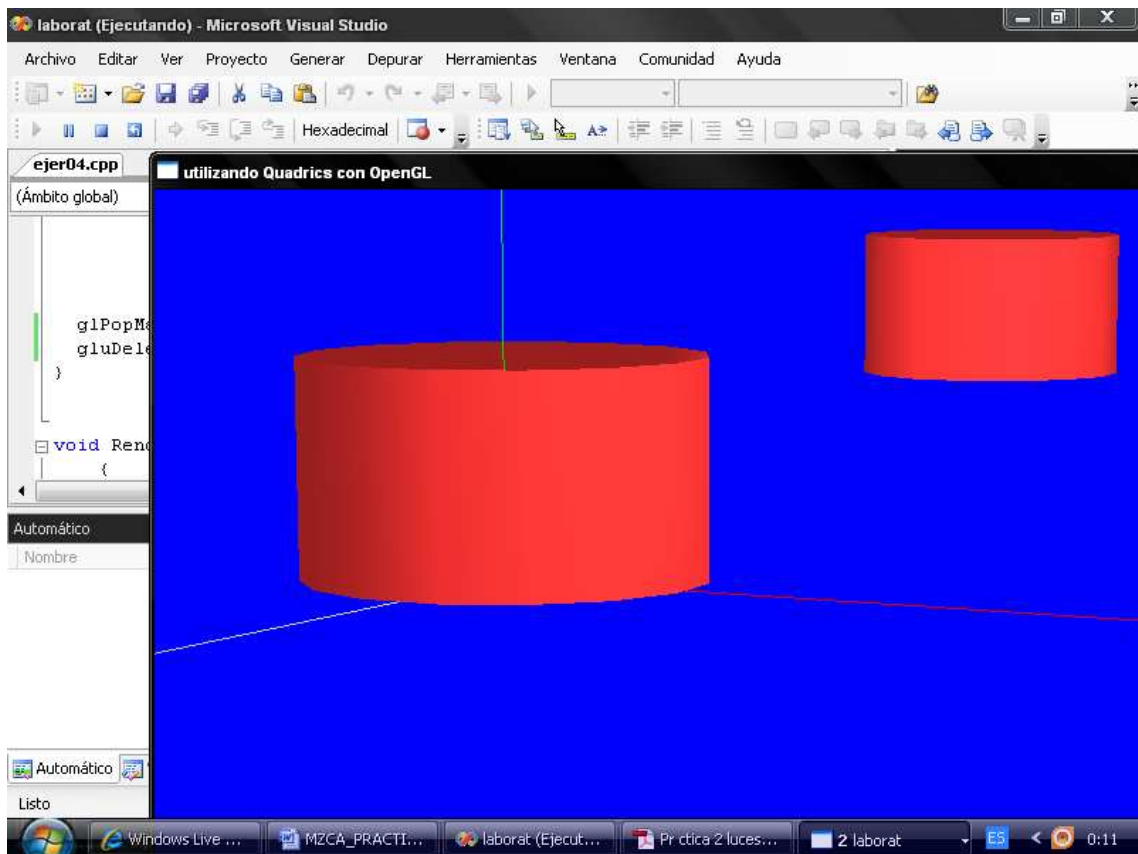
7.- Localice dentro de la rutina "RenderScene()" la línea "glShadeModel(GL\_FLAT)" y cambie "GL\_FLAT" por "GL\_SMOOTH". Construya y ejecute.

8.- Reporte la figura obtenida



## Ejercicio 2

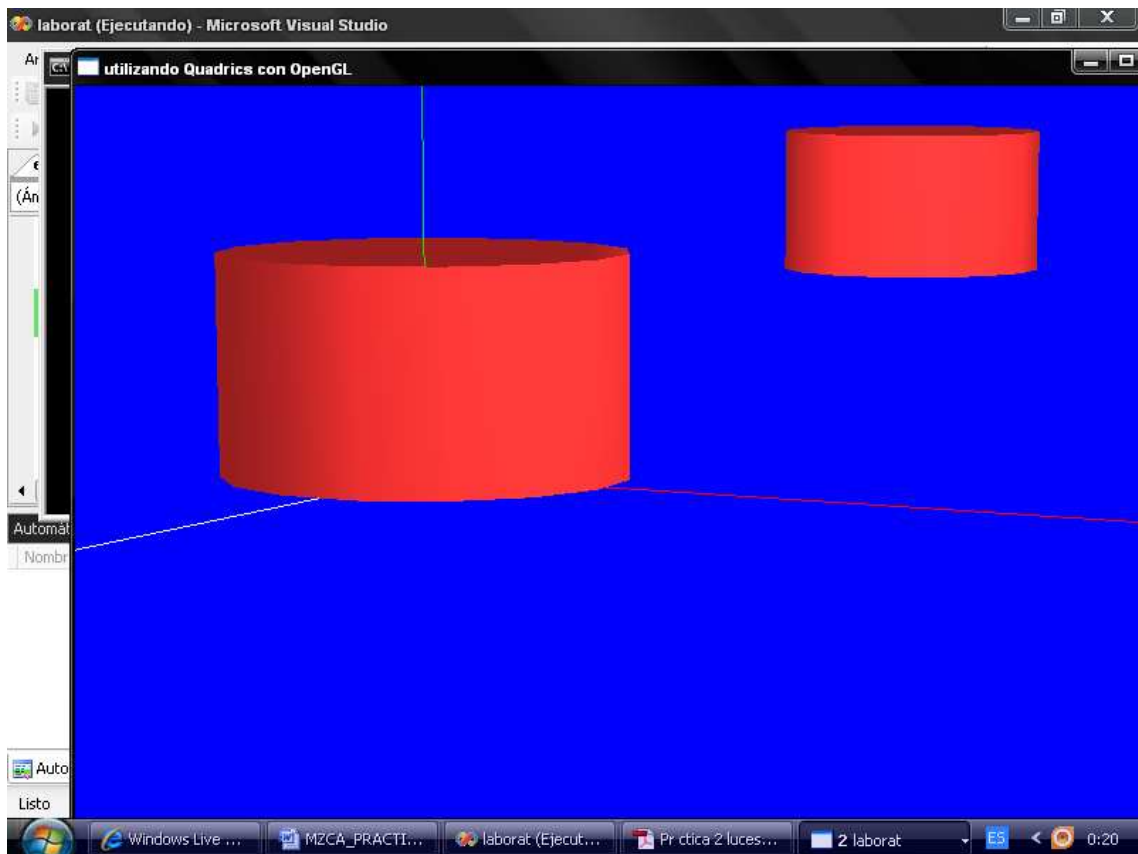
9.- Dentro de la subrutina "Cilindro()" hay código en comentarios que dibuja un segundo cilindro de iguales dimensiones al primero pero alejado, retire los comentarios, construya y ejecute el programa.



10.- Dentro de la subrutina "Luces()" agregue código para que el programa tome en cuenta la distancia en su modelo de iluminación

```
void luces(void)
{
//Cargando las ecuaciones de luz
    glEnable (GL_LIGHTING);
    glEnable (GL_LIGHT0);
    glEnable (GL_LINEAR_ATTENUATION);

    glLightfv(GL_LIGHT0, GL_AMBIENT, light_Ambient );
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_Diffuse );
    glLightfv(GL_LIGHT0, GL_POSITION, light_Position );
}
```

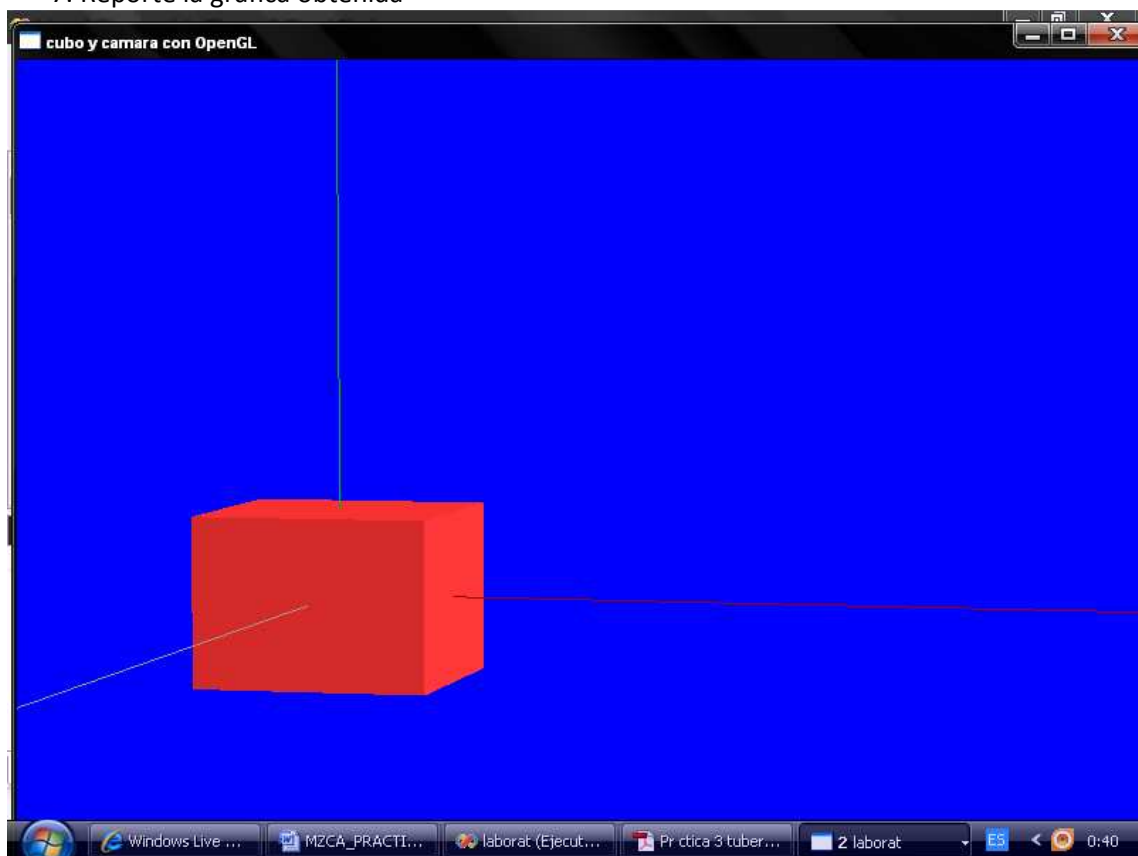


## Listas

### Ejercicio 1

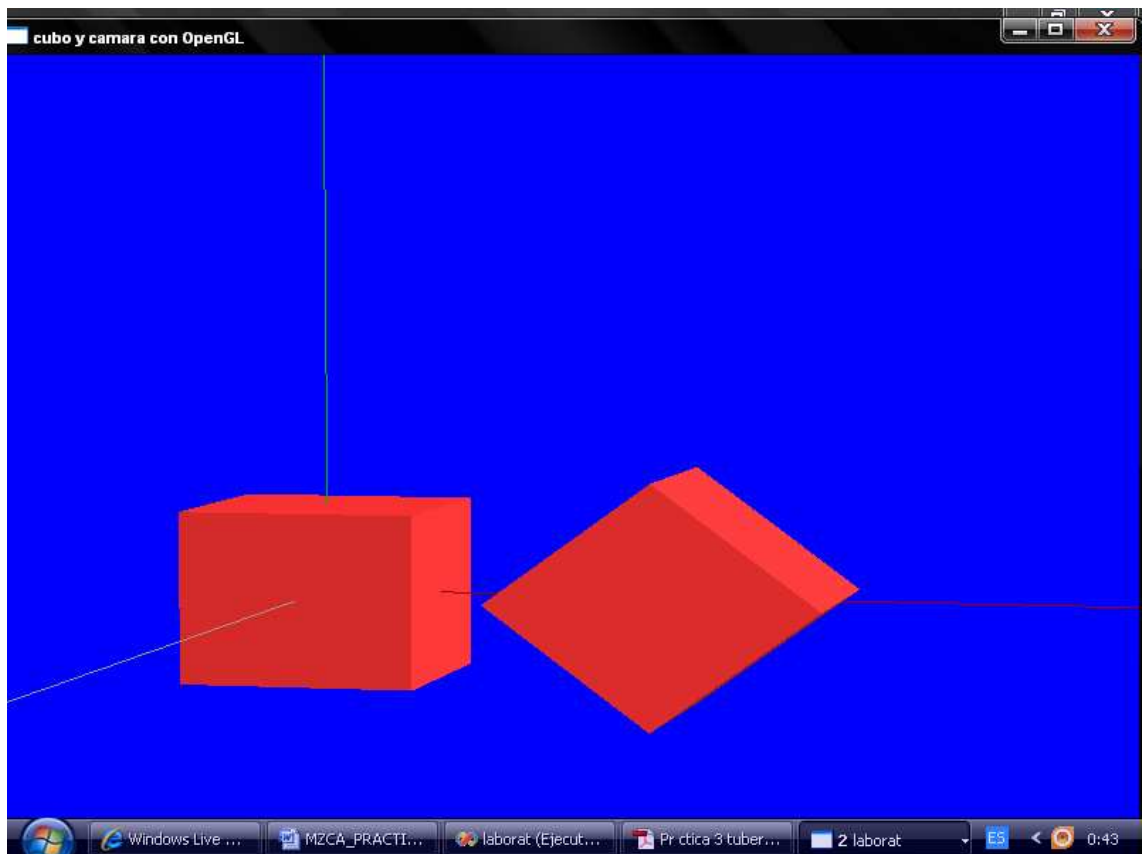
1. Vamos a crear un nuevo proyecto dentro de la carpeta "prácticas\_gl". Así que presione en secuencia menú "File->New Project" o simplemente presione simultáneamente las teclas "ctrl+shift+N". El nuevo proyecto se llamará "ejer05".
2. Localice el archivo el archivo "ejemplo\_listas\_01.cpp" y cópielo de su origen a la carpeta "...\\prácticas\_gl\\ejer05".
3. Presione la secuencia menú "Project->Add Existing\_Item" o simplemente presione la combinación de teclas "Shift+Alt+A".
4. El programa consta de las siguientes partes
  - a. Sección de variables globales.
  - b. Subrutina "luces()" para habilitar la ecuación de iluminación de OpenGL.
  - c. Subrutina "EjesReferencia()" para dibujar los ejes del sistema de referencia de OpenGL.habilitar
  - d. Subrutina "cubo()" Par a dibujar un cubo centrado en el origen de coordenadas.

- e. Subrutina "RenderScene()" Para invocar las subrutinas anteriores.
  - f. Subrutina "main()" Para crear una ventana Windows para us o con OpenGL.
5. A su vez, la subrutina "RenderScene" tiene tres secciones en desorden correspondientes a los ejercicios de estas práctica. Las secciones son:
- a. Ejercicio 3
  - b. Ejercicio 1
  - c. Ejercicio 2
6. Construya y ejecute.
7. Reporte la gráfica obtenida



## Ejercicio 2

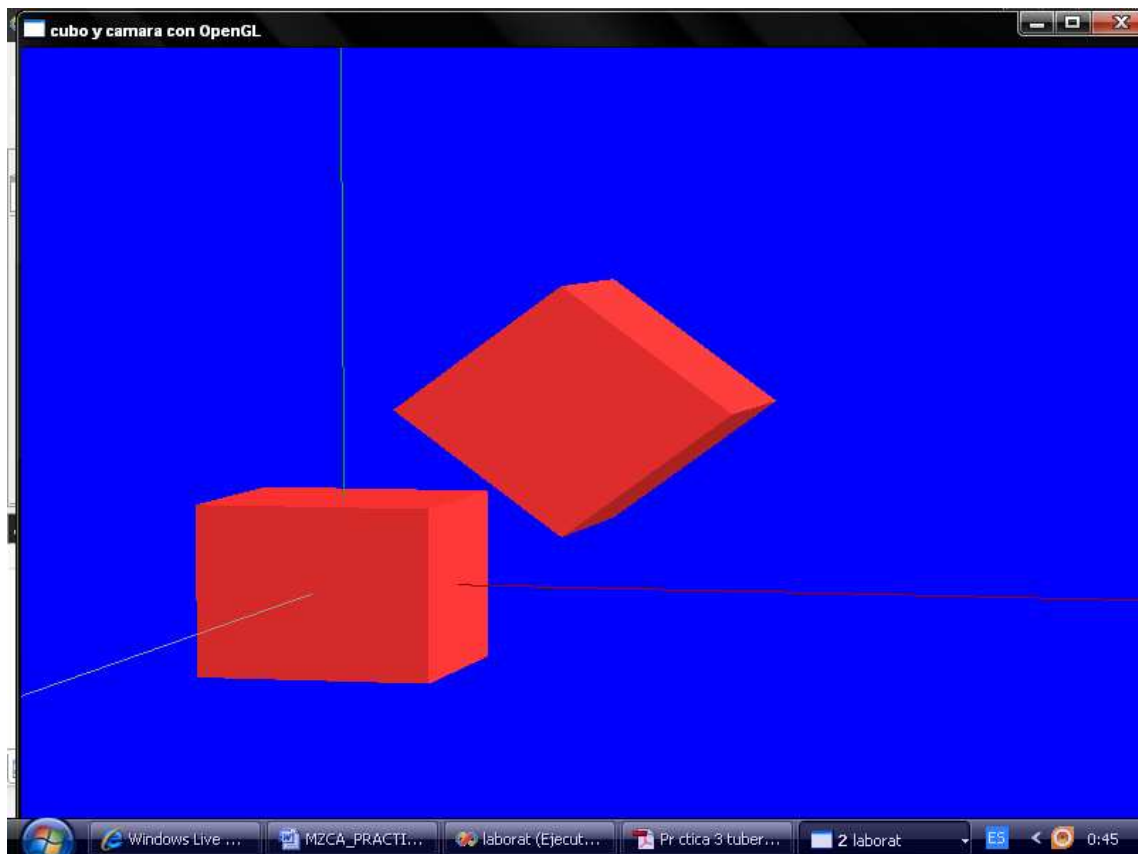
- 8. Dentro de la subrutina "RenderScene()" hay código entre comentarios titulado "Ejercicio\_2".
- 9. Retire los comentarios, construya y ejecute.
- 10. Reporte la gráfica obtenida.



11. Localice las líneas de código siguiente, cámbielas de orden, construya y ejecute (es posible que note distorsión del cubo que se dibuja debido a la relación de aspecto de su monitor).

- a. `glTranslatef(3,0,0);`
- b. `glRotatef(45,0.0,0.0,1.0);`

12. Reporte la gráfica observada

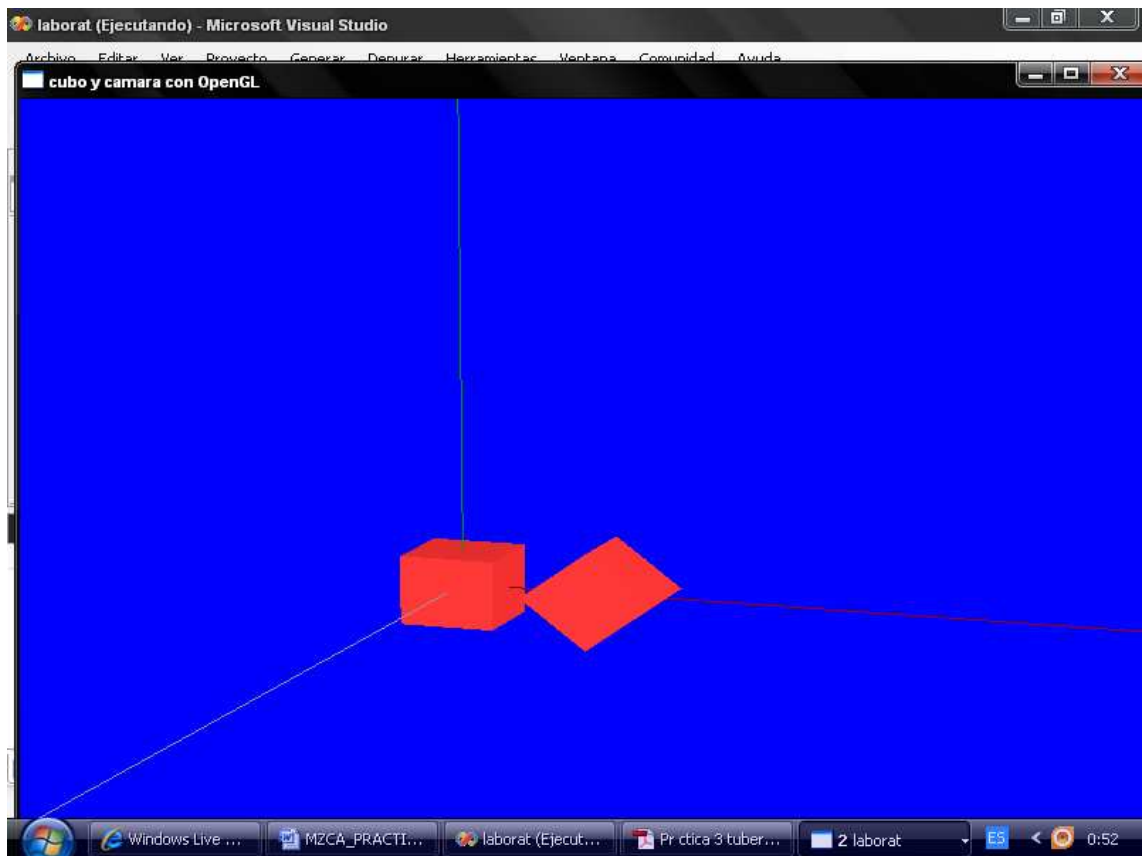


13. Elabore y reporte algún principio que permite explicar lo que ocurre con el orden en que se definen las transformaciones.

En el primer caso, primero trasladamos la figura hasta el origen y posteriormente la rotamos, pero en el segundo caso, primero se rota la figura y después se le traslada, el problema es que si se realiza una rotación desde un punto que no sea el origen se obtendrá un resultado que quizá no sea el esperado.

### Ejercicio 3

14. Dentro de la subrutina "RenderScene()" se hay código entre comentarios titulado "Ejercicio\_3".
15. Retire los comentarios, construya y ejecute.
16. Reporte la gráfica obtenida.



17. Dentro de los argumentos de la línea "gluLookAt()", cambie las coordenadas del "eye", del "At" y del "Up".

18. Reporte la gráfica obtenida.

```
gluLookAt(15,20,50,1,1,-5,1,2,3);
```

