



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e  
INTERACCIÓN HUMANO COMPUTADORA



## **EJERCICIOS DE CLASE N° 4**

**NOMBRE COMPLETO: HERNANDEZ SOLIS BRANDON**

**N° de Cuenta: 318263113**

**GRUPO DE LABORATORIO: 2**

**GRUPO DE TEORÍA: 6**

**SEMESTRE 2025-1**

**FECHA DE ENTREGA LÍMITE: 03/09/2024**

**CALIFICACIÓN: \_\_\_\_\_**

## EJERCICIOS DE SESIÓN:

1. Actividades realizadas. Una descripción de los ejercicios y capturas de pantalla de bloques de código generados y de ejecución del programa

Terminar de Construir la grúa con:  
-Cuerpo de la grúa (prisma rectangular).  
-brazo de 3 partes, 4 articulaciones, 1 canas

Sección de código de la base y llantas:

```
//AQUÍ SE DIBUJA LA CABINA, LA BASE, LAS 4 LLANTAS

//Posicion Base
model = glm::translate(model, glm::vec3(0.0f, 0.0f, -8.0f));

modelaux = model;

//Base Rectangular

glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
model = glm::scale(model, glm::vec3(8.0f, 4.0f, 6.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular

model = modelaux;

//Posicion Cilindro 1
model = glm::translate(model, glm::vec3(4.0f, -2.0f, 3.0f));

modelaux = model;

//Cilindro 1
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
color = glm::vec3(0.0f, 0.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[2]->RenderMeshGeometry(); //dibuja cubo, pirámide triangular, pirámide base cuadrangular
```

Sección de código de los Brazos y Articulaciones:

```

// SE EMPIEZA EL DIBUJO DEL BRAZO
model = glm::translate(model, glm::vec3(-4.0f, 2.0f, 3.0f));

//Posicion Articulacion 0
model = glm::translate(model, glm::vec3(0.0f, 2.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(0.0f, 0.0f, 1.0f));

modelaux = model;

//Articulacion 0
model = glm::scale(model, glm::vec3(0.8f, 0.8f, 0.8f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //Color
sp.render();

model = modelaux;

//Posicion Brazo 1
model = glm::translate(model, glm::vec3(-2.0f, 2.0f, 0.0f));
model = glm::rotate(model, glm::radians(135.0f), glm::vec3(0.0f, 0.0f, 1.0f));

modelaux = model;

model = glm::scale(model, glm::vec3(5.0f, 1.0f, 1.0f));

//Posicion Brazo 1
model = glm::translate(model, glm::vec3(-2.0f, 2.0f, 0.0f));
model = glm::rotate(model, glm::radians(135.0f), glm::vec3(0.0f, 0.0f, 1.0f));

modelaux = model;

model = glm::scale(model, glm::vec3(5.0f, 1.0f, 1.0f));

glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniformMatrix4fv(uniformView, 1, GL_FALSE, glm::value_ptr(camera.calculateViewMatrix()));
color = glm::vec3(1.0f, 0.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo, pirámide triangular, pirámide base cuadrangular

//para descartar la escala que no quiero heredar se carga la información de la matrix auxiliar

model = modelaux;

//Posicion Articulacion 1
model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion2()), glm::vec3(0.0f, 0.0f, 1.0f));

modelaux = model;

//Articulacion 1
model = glm::scale(model, glm::vec3(0.8f, 0.8f, 0.8f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //Color
sp.render();

```

```

//Posicion Brazo 3
model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));

modelaux = model;

//Brazo 3
model = glm::scale(model, glm::vec3(1.0f, 5.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular

model = modelaux;

//Posicion Articulacion 4

model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));
model = glm::rotate(model, glm::radians(45.0f), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion4()), glm::vec3(1.0f, 0.0f, 0.0f));

modelaux = model;

//Articulacion 4
model = glm::scale(model, glm::vec3(0.8f, 0.8f, 0.8f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //Color
sp.render();

model = modelaux;

```

Sección de código de la canasta:

```

//Posicion Canasta
model = glm::translate(model, glm::vec3(0.0f, -2.5f, 0.0f));

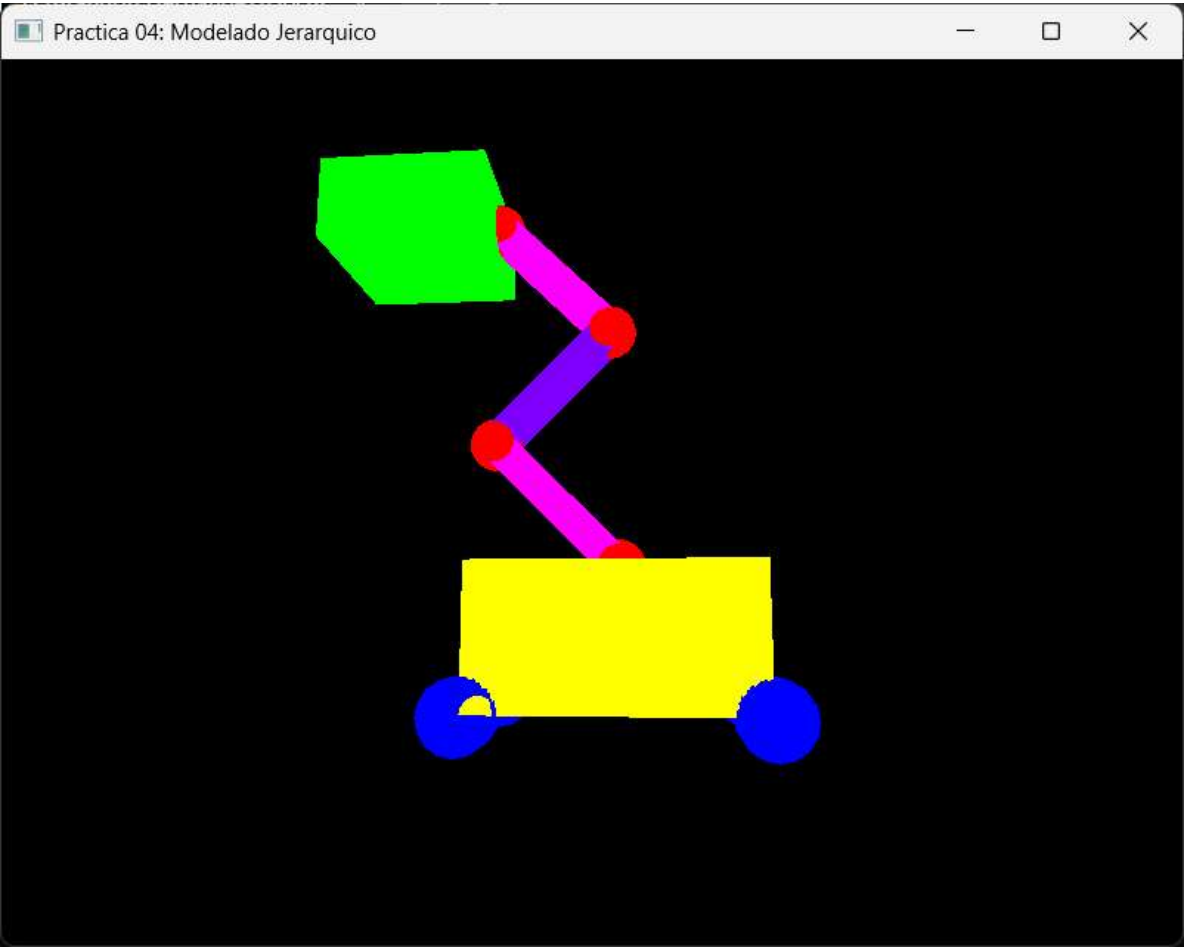
modelaux = model;

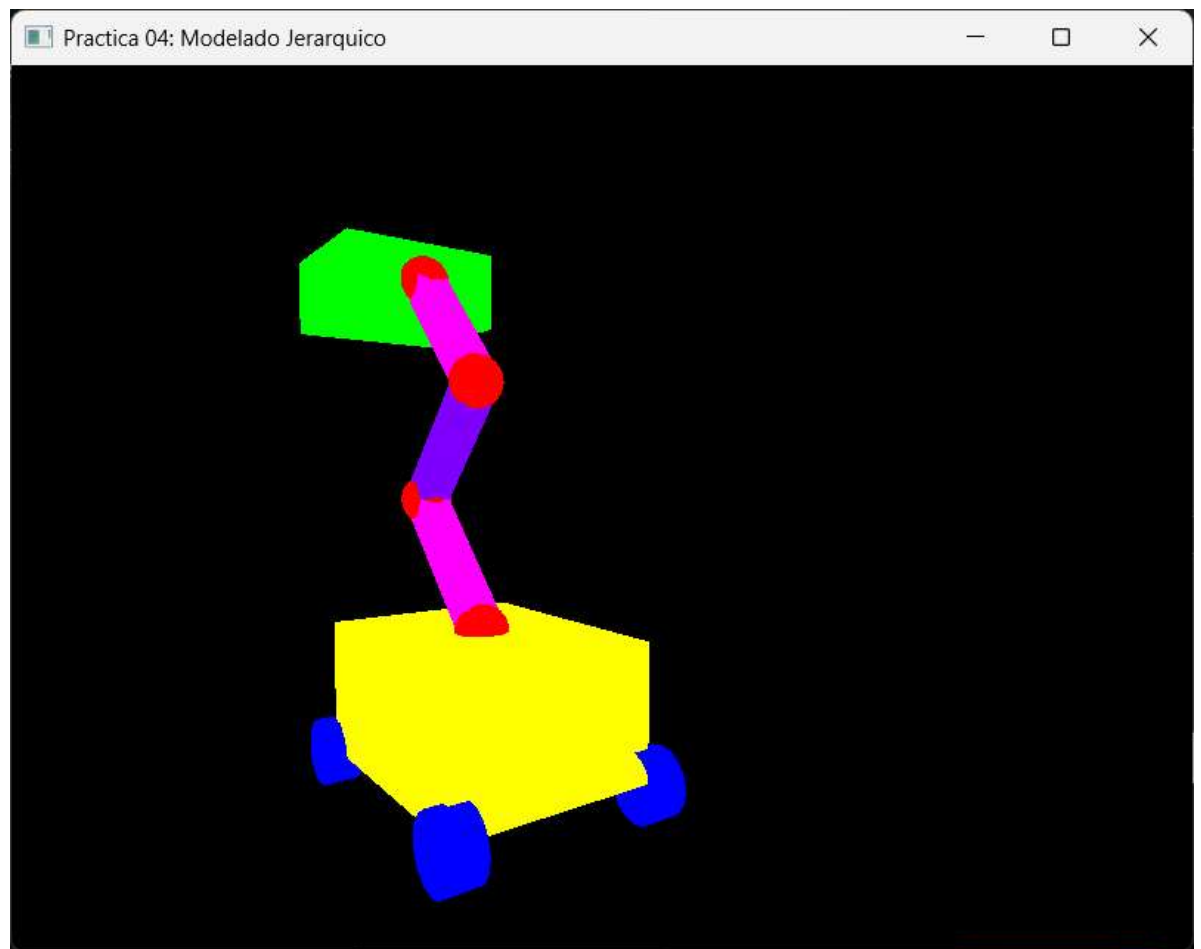
//Canasta
model = glm::scale(model, glm::vec3(2.5f, 5.0f, 5.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(0.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color)); //para cambiar el color del objetos
meshList[0]->RenderMesh(); //dibuja cubo y pirámide triangular

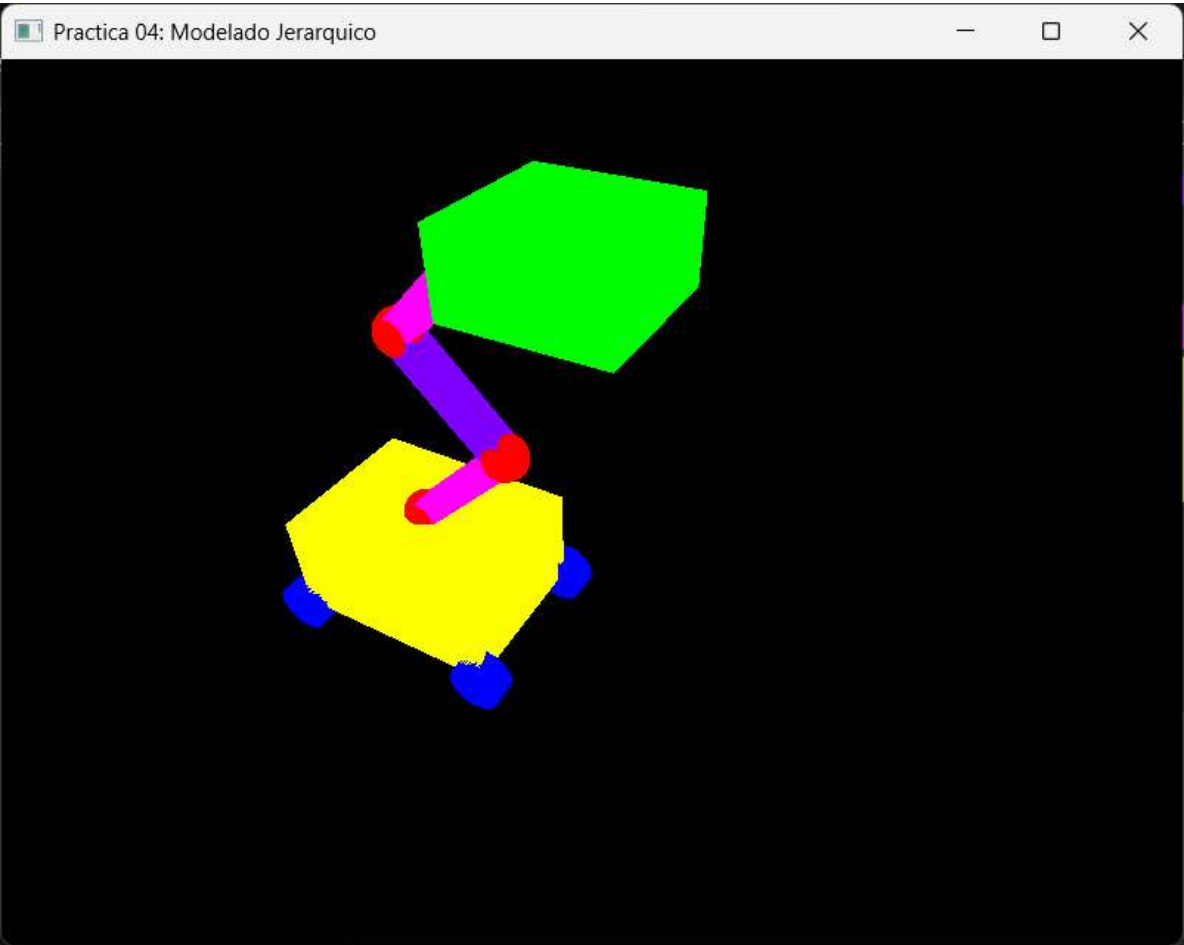
model = modelaux;

```

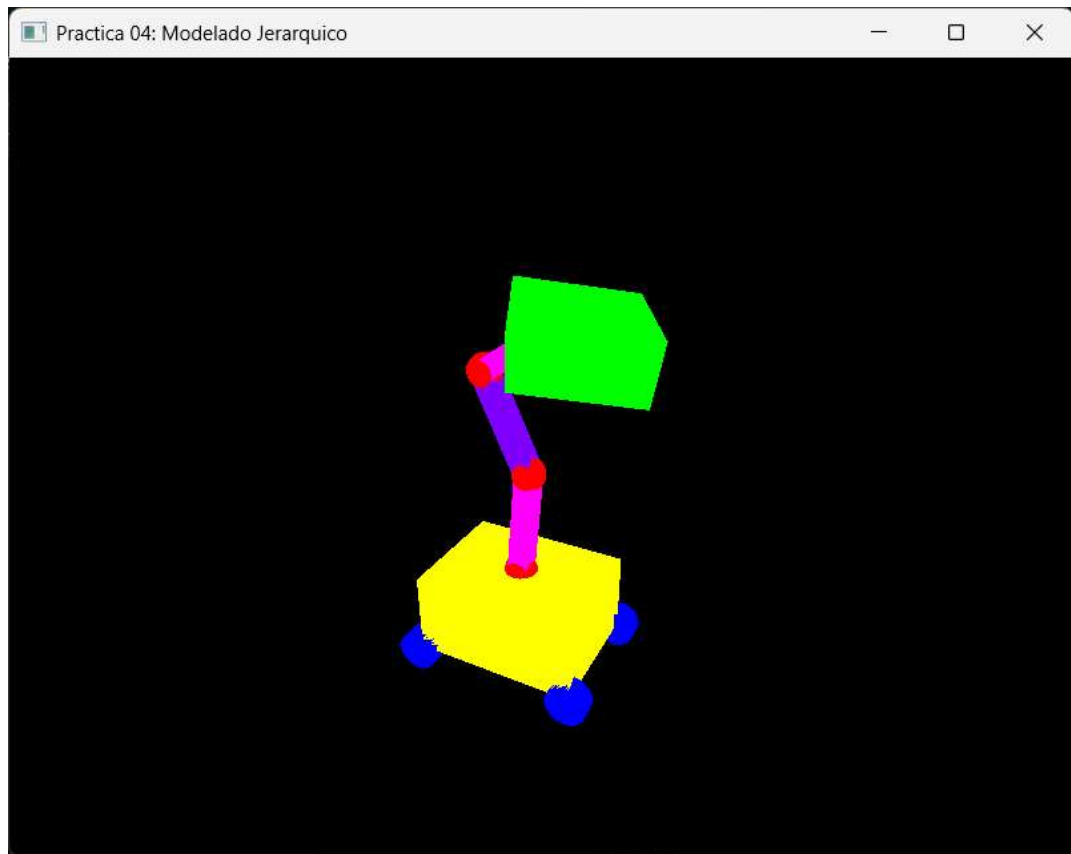
Modelo renderizado:











Se logro el ejercicio, pensaba ponerle que giraran las llantas, pero no estaba seguro porque no se notaba así que mejor lo quite. La canasta gira hacia los dos lados dependiendo de la tecla que se presione.

2. Problemas presentados. Listar si surgieron problemas a la hora de ejecutar el código

No hubo problemas.

3. Conclusión:

- a. Los ejercicios de la clase: Complejidad, explicación
- b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias.

Que practica tan divertida, me gusto andar moviendo las figuras y rotándolas y que se unan entre sí, la mejor parte fue como podían girar y podíamos decidir la dirección de giro y desde donde giraría. En general me gustó mucho y aprendí demasiado a hacer figuras, esto me hubiera servido mucho para la práctica anterior pero no tenía idea.