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 Nº 4

NOMBRE COMPLETO: Gomez Enríquez Agustin

Nº de Cuenta: 317031405

GRUPO DE LABORATORIO: 3

GRUPO DE TEORÍA: 5

SEMESTRE 2026-1

FECHA DE ENTREGA LÍMITE: 09 – septiembre - 2025

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**

```
316         // Esfera para articulaciones
317         sp.init();
318         sp.load();
319
320         // Matrices de modelo
321         glm::mat4 model(1.0f), modelaux(1.0f);
322
323         // Color inicial
324         glm::vec3 color(0.0f);
325
326         // ÁNGULO DE LA CANASTA
327         float anguloCanastaDeg = 0.0f;          // grados
328         const float giroSpeedDeg = 10.0f;       // grados/seg
```

En esta ocasión vamos a trabajar solamente en el main, de manera que trabajaremos con esferas, cilindros y un cubo. Después de definir nuestros parámetros como matrices y colores, decidí colocar unas variables para ajustar los grados y el movimiento que tendrá la canasta.

```
338         // Entrada de usuario (teclado/mouse)
339         glfwPollEvents();
340         camera.keyControl(mainWindow.getsKeys(), deltaTime);
341         camera.mouseControl(mainWindow.getXChange(), mainWindow.getYChange());
342
343         // Control de CANASTA: J (izq) / K (der)
344         {
345             const bool pressJ = mainWindow.getsKeys()[GLFW_KEY_J];
346             const bool pressK = mainWindow.getsKeys()[GLFW_KEY_K];
347             if (pressJ) anguloCanastaDeg -= giroSpeedDeg * deltaTime;
348             if (pressK) anguloCanastaDeg += giroSpeedDeg * deltaTime;
349         }
```

Después vamos a construir el control del movimiento de la canasta para que al presionar la tecla j o k se pueda mover en un solo sentido.

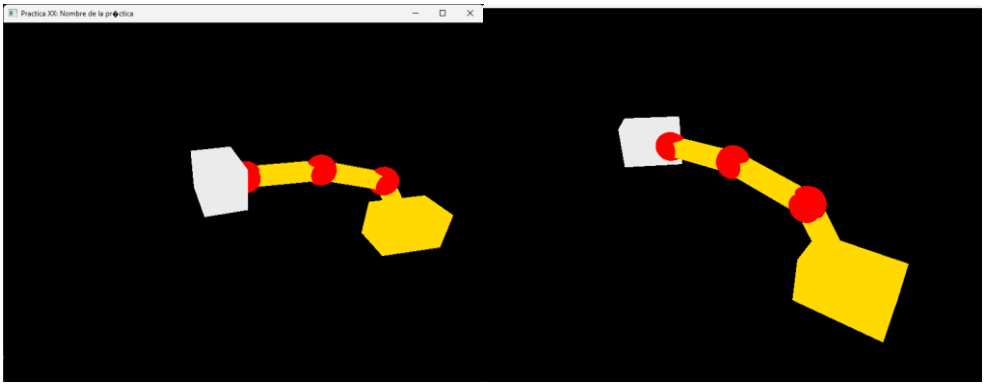
```
366         // BASE (caja de la máquina)
367         model = glm::mat4(1.0f);
368         model = glm::translate(model, glm::vec3(0.0f, 5.5f, -4.0f));
369         modelaux = model;
370         model = glm::scale(model, glm::vec3(5.0f, 3.0f, 3.0f));
371         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
372         color = glm::vec3(1.0f, 0.85f, 0.0f);
373         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
374         meshList[0]->RenderMesh();
375
376         // ARTICULACIÓN 1
377         model = modelaux;
378         model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(0.0f, 0.0f, 1.0f));
379         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
380         color = glm::vec3(1.0f, 0.0f, 0.0f);
381         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
382         sp.render();
383
384         // PRIMER BRAZO
385         model = glm::rotate(model, glm::radians(135.0f), glm::vec3(0.0f, 0.0f, 1.0f));
386         model = glm::translate(model, glm::vec3(2.5f, 0.0f, 0.0f));
387         modelaux = model;
388         {
389             glm::mat4 m = model;
390             m = glm::scale(m, glm::vec3(5.0f, 1.0f, 1.0f));
391             glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(m));
392             color = glm::vec3(1.0f, 0.85f, 0.0f);
393             glUniform3fv(uniformColor, 1, glm::value_ptr(color));
394             meshList[0]->RenderMesh();
395         }
```

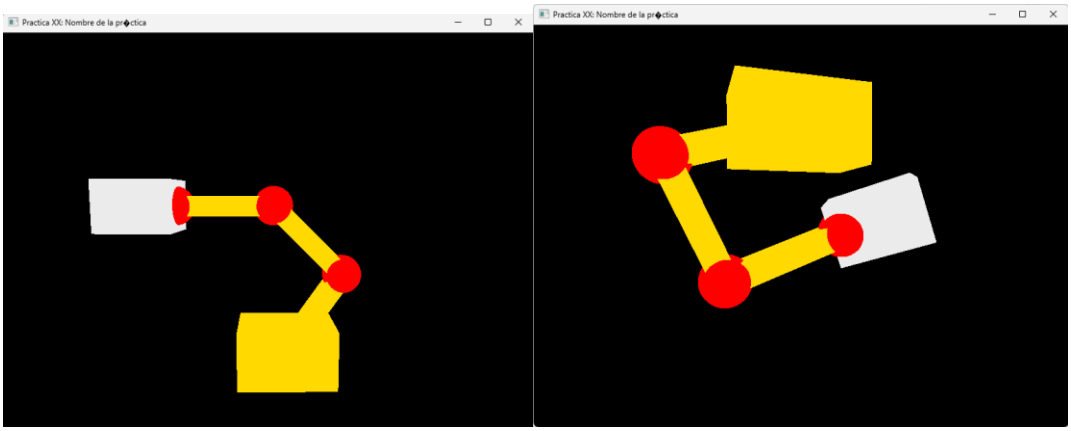
Durante clase se trabajaron los modelos base que son una caja como base, que internamente tiene una articulación (una pequeña esfera), que será nuestro punto de referencia para colocar un cilindro, este es nuestro brazo que se conectara con otra articulación, así hasta completar le jerarquía de extensiones. Debemos tomar en cuenta que cada brazo tendrá un movimiento individual pero afecta al siguiente bloque de brazos.

```
448         // CANASTA
449         {
450             glm::mat4 m = model_art4;
451             m = glm::translate(m, glm::vec3(1.7f, 0.0f, 0.0f));         // pequeño voladizo delante de la ultima articulación
452
453             // Giro "a izquierda/derecha" sobre el eje vertical Y
454             m = glm::rotate(m, glm::radians(anguloCanastaDeg), glm::vec3(0.0f, 1.0f, 0.0f));
455
456             // Tamaño de la canasta
457             m = glm::scale(m, glm::vec3(3.8f, 2.6f, 2.6f));
458
459             glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(m));
460             color = glm::vec3(0.92f, 0.92f, 0.92f);
461             glUniform3fv(uniformColor, 1, glm::value_ptr(color));
462             meshList[0]->RenderMesh();
463         }
464         glUseProgram(0);
465         mainWindow.swapBuffers();
466     }
```

Por último, vamos a crear nuestra canasta que será una caja con movimiento en el eje Y (de rotacion), pero tomando como eje un fragmento delante de la última articulación, esto para asegurar que la caja no va a invadir por completo a la articulación y tratar de hacerlo lo mas apegado a un caso real.

Ejecución:





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

Durante laboratorio no pude trabajar con la versión completa, me marcaba errores, la versión que estuvimos construyendo en clase fue la que usé al final, solamente dándole un orden al código para comprender como se une cada bloque. No tuve problemas con el ejercicio, pero noté que tardaba en cargar la imagen en la ventana y luego me tardaba en encontrar la imagen aun con la posición inicial de la cámara.

3. **Conclusión:**

Fue una actividad relativamente fácil, que nos permitió conocer tanto la jerarquía de figuras (Donde se llegan a encimar las figuras, pero respetan un orden de capas), además de esto logre comprender el como cambiando el punto de origen para cada articulación podemos establecer parámetros para los objetos que se relacionen a este, fue por esto por lo que podíamos hacer que los brazos mas alejados se pudieran girar si los brazos cercanos a la base se movían. Finalmente, con la actividad extra solo fue comprender el movimiento de los ejes para que en esta ocasión lo pudiéramos trabajar en el eje Y y no en el eje Z como los brazos.