



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



REPORTE DE PRÁCTICA N° 05

NOMBRE COMPLETO: Roberto Aburto López

N° de Cuenta: 319131996

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 06

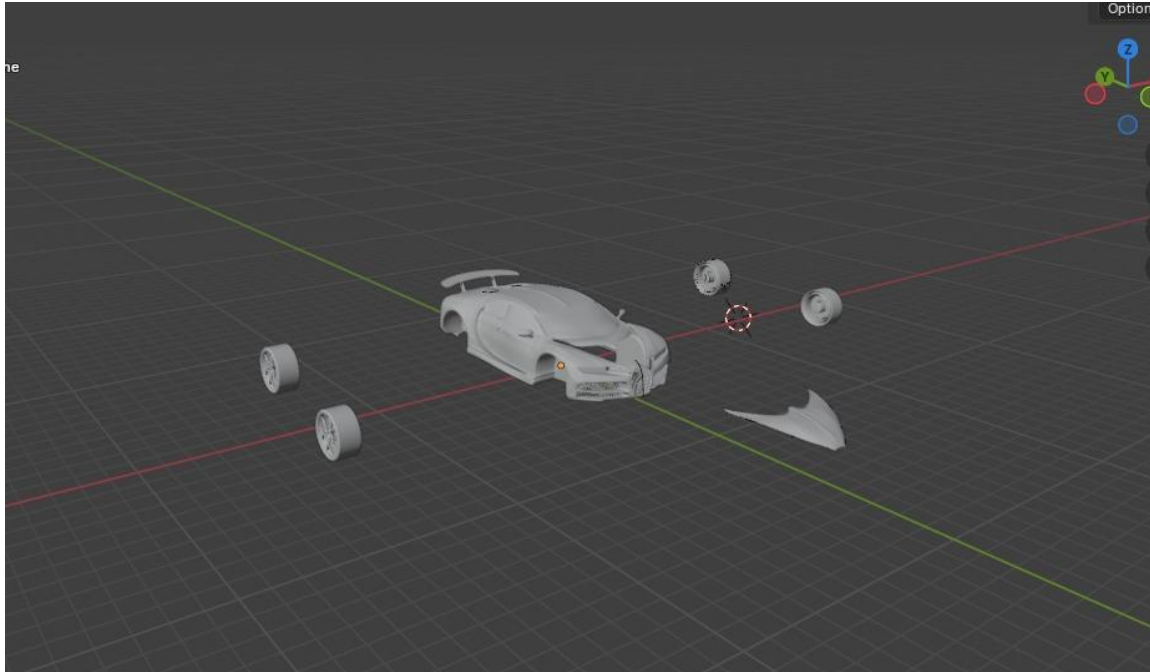
SEMESTRE 2026-1

FECHA DE ENTREGA LÍMITE: 27 de septiembre de 2025

CALIFICACIÓN: _____

1.- Importar su modelo de coche propio dentro del escenario a una escala adecuada.

Para esta actividad primero tuve que eliminar algunos elementos extra que tenía el archivo .obj del modelo de coche que descargué. Además, el modelo estaba construido con cuadrados, así que lo convertí a triángulos y separé las llantas y el cofre. Luego exporté cada parte como un archivo .obj para armar el modelo jerárquico. Un problema que se presentó fue que el programa no leía los archivos .fbx; no supe la razón, pero al final opté por convertir todos los modelos a .obj y así se solucionó.



Se importaron los modelos previamente exportados desde Blender en formato .obj, correspondiendo a las diferentes partes del coche. Cada uno fue cargado de manera individual en el programa para poder manipularlos por separado: el cuerpo principal del coche, el cofre, las llantas delanteras (izquierda y derecha) y las llantas traseras (izquierda y derecha).

```
// Coche
Model coche;
Model cofre;
Model ruedaDelDer;
Model ruedaDelIzq;
Model ruedaTrasIzq;
Model ruedaTrasDer;
```

```
// ----- Coche -----  
  
Model coche = Model();  
coche.LoadModel("Models/car.obj");  
  
Model ruedaDelDer = Model();  
ruedaDelDer.LoadModel("Models/ruedaDelDer.obj");  
  
Model ruedaDelIzq = Model();  
ruedaDelIzq.LoadModel("Models/ruedaDelIzq.obj");  
  
Model ruedaTrasDer = Model();  
ruedaTrasDer.LoadModel("Models/ruedaDelDer.obj");  
  
Model ruedaTrasIzq = Model();  
ruedaTrasIzq.LoadModel("Models/ruedaDelIzq.obj");  
  
Model cofre = Model();  
cofre.LoadModel("Models/cofre.obj");
```

3.- Importar el cofre del coche, acomodarlo jerárquicamente y agregar la rotación para poder abrir y cerrar.

[illegible]

```

////// ----- RUEDA FRONTAL IZQUIERDA -----
-----

model = glm::translate(model, glm::vec3(-5.0f, 0.0, 0.0));
modelaux2 = model;

model = glm::rotate(model, glm::radians(mainWindow.getarticulacion12()), glm::vec3(1.0f, 0.0f, 0.0f));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
ruedaDelIzq.RenderModel();

model = modelaux2;

////// ----- RUEDA TRASERA IZQUIERDA -----
-----

model = glm::translate(model, glm::vec3(0.0f, 0.0, 7.8));
modelaux2 = model;

model = glm::rotate(model, glm::radians(mainWindow.getarticulacion12()), glm::vec3(1.0f, 0.0f, 0.0f));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
ruedaTrasIzq.RenderModel();

model = modelaux2;

////// ----- RUEDA TRASERA DERECHA -----
-----

model = glm::translate(model, glm::vec3(5.0f, 0.0, 0.0));
modelaux2 = model;

model = glm::rotate(model, glm::radians(mainWindow.getarticulacion11()), glm::vec3(1.0f, 0.0f, 0.0f));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
ruedaTrasIzq.RenderModel();

model = modelaux2;

```

```

//// ----- COFRE -----

model = glm::translate(model, glm::vec3(-2.5f, 1.7, -7.0));
modelaux2 = model;

model = glm::rotate(model, glm::radians(mainWindow.getarticulacion10()), glm::vec3(1.0f, 0.0f, 0.0f));
color = glm::vec3(0.0f, 1.0f, 0.7f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
cofre.RenderModel();

model = modelaux2;

```

4.- Agregar traslación con teclado para que pueda avanzar y retroceder de forma independiente

Tuve que crear nuevas variables para controlar la traslación hacia adelante y hacia atrás. Es importante mencionar que, junto con el movimiento de traslación, las llantas también se animan dependiendo de la dirección en la que se desplace el coche.

```

Window::Window(GLint windowHeight, GLint windowWidth)
{
    ...

    carX = 20.0f;
    carZ = 0.0f;
    carYaw = 0.0f;

    // Frente del coche: +Z
    fwdX = 0;
    fwdZ = 1;
}

```

Movimiento del carro

```
// Coche
const float step = 0.5f;

if ((key == GLFW_KEY_UP) && (action == GLFW_PRESS || action == GLFW_REPEAT)) {
    // Avanza hacia adelante en el eje Z
    theWindow->carZ -= step;

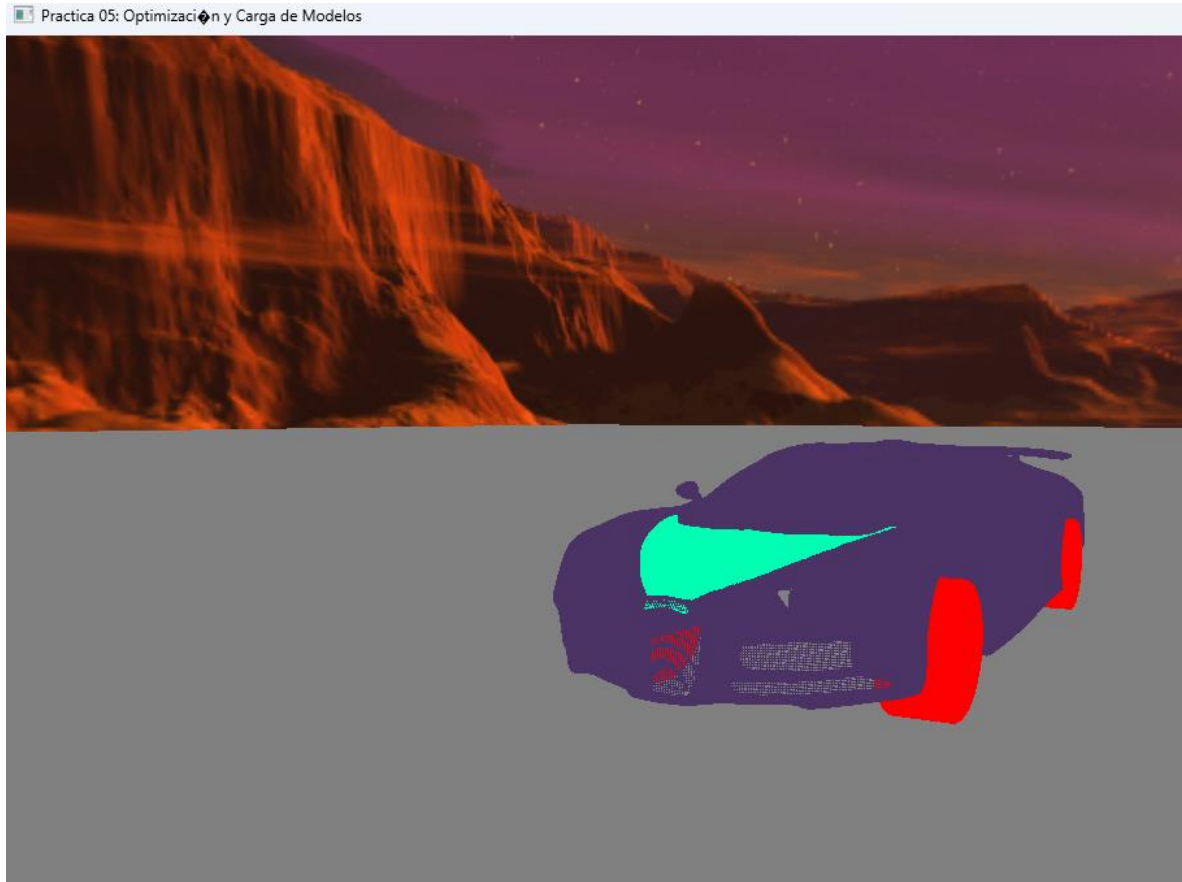
    // Animación de ruedas opcional
    theWindow->articulacion11 -= 10.0f;
    theWindow->articulacion12 -= 10.0f;
}

if ((key == GLFW_KEY_DOWN) && (action == GLFW_PRESS || action == GLFW_REPEAT)) {
    // Retrocede hacia atrás en el eje Z
    theWindow->carZ += step;

    theWindow->articulacion12 += 10.0f;
    theWindow->articulacion11 += 10.0f;
}

// Cofre
if (key == GLFW_KEY_M)
{
    if (theWindow->articulacion10 > 40)
    {
    }
    else{
        theWindow->articulacion10 += 10.0;
    }
}
```

Resultados



Video

<https://drive.google.com/file/d/1Srg57GlzAtVks0jJuw9wsqql9IUIKrBd/view?usp=sharing>

Conclusión

Esta práctica me resultó interesante porque aprendí a utilizar Blender, exportar modelos y cargarlos en mi programa. También entendí cómo dividir el modelo en partes para manipularlas por separado. Además, reforcé el manejo de jerarquías y la implementación de límites en los movimientos.