


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 Nº 06

**NOMBRE COMPLETO:** Tapia Ledesma Angel Hazel

**Nº de Cuenta:** 320070358

**GRUPO DE LABORATORIO:** 02

**GRUPO DE TEORÍA:** 02

**SEMESTRE 2026-1**

**FECHA DE ENTREGA LÍMITE:** 12-09-25

**CALIFICACIÓN: _____**

# Actividades realizadas

**Ejercicio 1: Crear un dado de 8 caras y texturizarlo por medio de código.**
**Ejercicio 2: Importar el modelo de su coche con sus 4 llantas acomodadas y tener texturizadas las 4 llantas (diferenciar caucho y rin)**
**Ejercicio 3: Texturizar la cara del personaje de la imagen tipo cars en el espejo (ojos) y detalles en cofre y parrilla de su propio modelo de coche**

# Descripción de las actividades

Para el caso de la primera actividad, lo que se hizo fue elegir una imagen de internet de un dado de 8 caras, posteriormente se trabajó con un programa de edición de imagen para recortar lo sobrante y solo dejar lo necesario para la textura, de igual modo que ajustar las dimensiones a las potencias de 2, en este caso 512 de ancho por 1024 de altura. Finalmente exportamos la imagen como formato PNG, aunque no ocupamos el canal alpha en este caso. Una vez con la imagen, la movimos a la carpeta de texturas, después ya trabajamos con Open GL.

En Open GL tuvimos que crear la figura geométrica perteneciente al dado de 8 caras, para esto generamos la función CrearDadoOchoCaras(), a la cual es similar a la del dado que teníamos, solo que aquí modificamos las coordenadas de los vértices, los índices y recalculamos cuantos vértices e índices había para crear la malla y de esta manera, fue como se generó la figura geométrica.

Para el texturizado, lo que se hizo fue importar la textura a Open GL, después se aplicó a la figura, sin embargo, aún no estaba ajustada, para ajustarla lo que se hizo fue poner las coordenadas del “area” de texturizado que queríamos de nuestra imagen a cada cara. En este caso, cómo la imagen era en pixeles y se necesitaba en escala de 0 a 1, para transformar de pixeles a esta escala, lo que se hizo fue localizar con la herramienta de edición las coordenadas del pixel de cada vértice, y luego dividir esta coordenada entre la anchura para el eje x y la altura para el eje Y, de este modo “transformamos” los pixeles a las coordenadas de Open GL.

Finalmente, lo que hicimos fue en el main incluir la llamada a la función CrearDadoOchoCaras() para usarlo en el main, después llamamos a la textura poniendo su ruta en la variable que almacena la textura y después, aplicamos las transformaciones necesarias, en este caso ninguna, para finalmente renderizar la figura.

Para la actividad 2 y 3 lo que se hizo fue primero trabajar en blender con el modelo del carro que se tenía, lo primero fue con las llantas, se eligió una llanta, después con su UV map, lo que se hizo fue exportar el UV Map y en una aplicación de edición de imagen se

trabajó sobre este UV Map, añadiendo las texturas tanto de la llanta como para el rin en el sitio correspondiente según donde marcaba el UV map, una vez que se hizo esto. Se ocultó la capa del UV Map para exportar solo la imagen con las texturas acomodadas. Posteriormente, lo que se hizo fue en blender fue seleccionar otra vez la llanta y en su textura, elegimos imagen e importamos la imagen generada en la aplicación de edición de imagen, después se fue al UV map para ajustar detalles menores del texturizado. A continuación, exportamos el modelo en formato .obj, el cual después se abrió en open GL, creando la variable que almacena el modelo. Finalmente aplicamos transformaciones necesarias y la renderización del modelo para las 4 llantas.

Para el caso del auto, en cuanto al cofre, parilla y el parabrisas, lo que se hizo fue el mismo proceso que para las llantas, se seleccionó el auto, se generó su UV Map y se almacenó, para en la aplicación de edición de imagen posicionar las texturas en la zona correspondiente, e importar la imagen, así como después trabajar con esa imagen en blender , adaptar los detalles del UV Map y exportar el modelo.

Finalmente, se abrió el modelo en open GL y de igual forma, se aplicaron las transformaciones correspondientes para después renderizarlo.

## Código generado:

### Actividad 1:

```
Texture dado8CarasTexture;
void CrearDadoOchoCaras()
{
    unsigned int cubo8caras_indices[] = {
        // Top
        // front
        0, 1, 2,
        // Right
        3, 4, 5,
        // back
        6, 7, 8,
        // left
        9, 10, 11,
        // Bottom
        // front
        12, 13, 14,
        // right
        15, 16, 17,
        // back
        18, 19, 20,
        // left
        21, 22, 23
    };
    //Ejercicio 1: reemplazar con sus dados de 6 caras texturizados, agregar normales
// average normals
    GLfloat cubo8caras_vertices[] = {
        // Arriba
        // front
        //x     y       z       S       T           NX      NY      NZ
        0.0f, 0.5f,  0.0f,      0.50f,  0.25f,      0.0f,   0.0f,   -1.0f,  //0
        -0.5f, 0.0f,  0.5f,     0.75f,  0.5f,       0.0f,   0.0f,   -1.0f,  //1
        0.5f,  0.0f,  0.5f,     0.25f,  0.5f,       0.0f,   0.0f,   -1.0f,  //2
        // right
        //x     y       z       S       T
        0.0f,  0.5f,  0.0f,     0.50f,  0.25f,      -1.0f,  0.0f,   0.0f,   //3
        0.5f,  0.0f,  0.5f,     0.25f,  0.5f,       -1.0f,  0.0f,   0.0f,   //4
        0.5f,  0.0f, -0.5f,     0.007f, 0.25f,      -1.0f,  0.0f,   0.0f,   //5
        // back
        0.0f,  0.5f, 0.0f,      0.50f,  0.25f,      0.0f,   0.0f,   1.0f,   //6
        -0.5f, 0.0f, -0.5f,     0.25f,  0.05f,      0.0f,   0.0f,   1.0f,   //7
        0.5f,  0.0f, -0.5f,     0.05f,  0.25f,      0.0f,   0.0f,   1.0f,   //8

        // left
        //x     y       z       S       T
        0.0f,  0.5f,  0.0f,     0.50f,  0.25f,      1.0f,   0.0f,   0.0f,   //9
        -0.5f,  0.0f,  0.5f,    0.75f,  0.5f,       1.0f,   0.0f,   0.0f,   //10
        -0.5f,  0.0f, -0.5f,    0.98f,  0.26f,      1.0f,   0.0f,   0.0f,   //11
```

```
        // bottom
        // front
        // front
        //x     y       z       S       T           NX      NY      NZ
        0.0f, -0.5f,  0.0f,     0.50f,  0.75f,      0.0f,   0.0f,   -1.0f,  //12
        -0.5f, 0.0f,  0.5f,     0.66f,  0.50f,      0.0f,   0.0f,   -1.0f,  //13
        0.5f,  0.0f,  0.5f,     0.33f,  0.50f,      0.0f,   0.0f,   -1.0f,  //14
        // right
        //x     y       z       S       T
        0.0f,  -0.5f, 0.0f,     0.50f,  0.75f,      -1.0f,  0.0f,   0.0f,   //15
        0.5f,  0.0f,  0.5f,     0.25f,  0.5f,       -1.0f,  0.0f,   0.0f,   //16
        0.5f,  0.0f, -0.5f,     0.01f,  0.75f,      -1.0f,  0.0f,   0.0f,   //17
        // back
        0.0f,  -0.5f, 0.0f,     0.50f,  0.75f,      0.0f,   0.0f,   1.0f,   //18
        -0.5f, 0.0f, -0.5f,     0.25f,  1.0f,       0.0f,   0.0f,   1.0f,   //19
        0.5f,  0.0f, -0.5f,     0.01f,  0.75f,      0.0f,   0.0f,   1.0f,   //20

        // left
        //x     y       z       S       T
        0.0f,  -0.5f, 0.0f,     0.50f,  0.75f,      1.0f,   0.0f,   0.0f,   //21
        -0.5f,  0.0f,  0.5f,    0.75f,  0.5f,       1.0f,   0.0f,   0.0f,   //22
        -0.5f,  0.0f, -0.5f,    1.0f,   0.75f,      1.0f,   0.0f,   0.0f,   //23

    };

    Mesh* dadoOchoCaras = new Mesh();
    dadoOchoCaras->CreateMesh(cubo8caras_vertices, cubo8caras_indices, 24*8, 24);
    meshList.push_back(dadoOchoCaras);
}
```

```
CrearDadoOchoCaras();
dado8CarasTexture=Texture("Textures/dado8caras.png");
dado8CarasTexture.LoadTextureA();
```

```
//Ejercicio 1: Crear un dado de 8 caras y texturizarlo por medio de código
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-1.5f, 4.5f, -2.0f));
model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, -1.0f, 0.0f));
model = glm::scale(model, glm::vec3(2.0f, 2.0f, 2.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
dado8CarasTexture.UseTexture();
meshList[5]->RenderMesh();
```

## Actividad 2 y 3

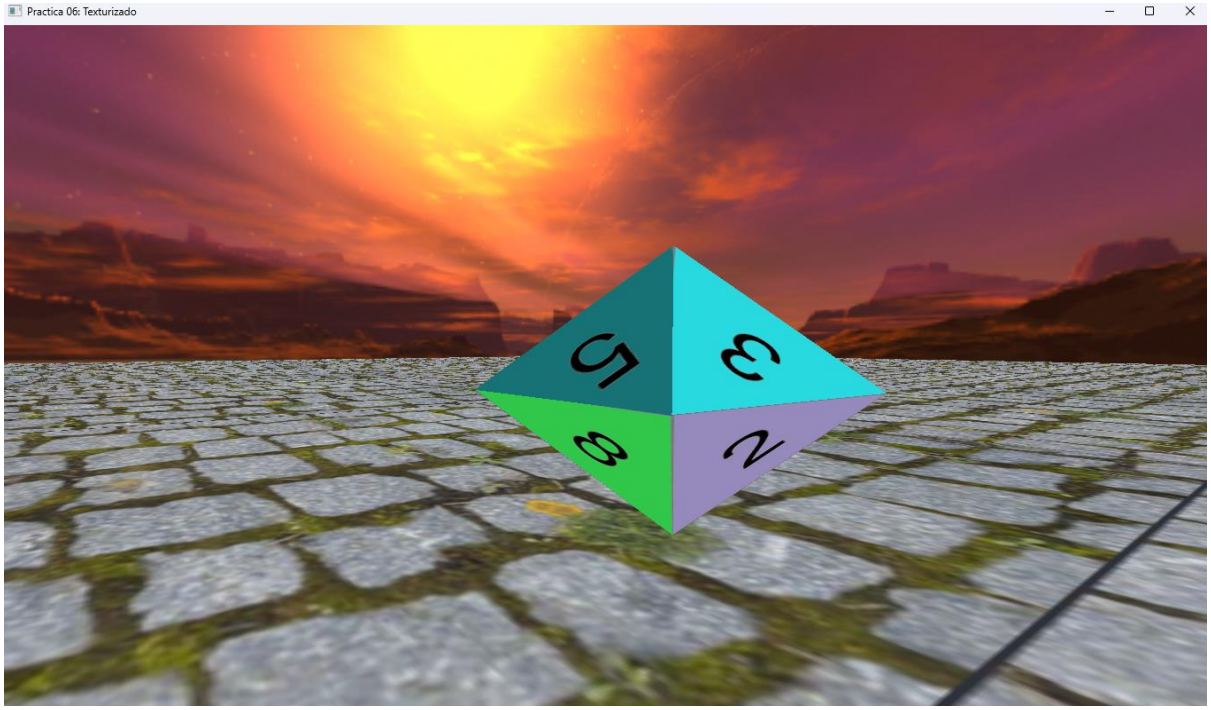
```
Model Porsche2014Carro_M;
Model Porsche2014Llanta_M;
```

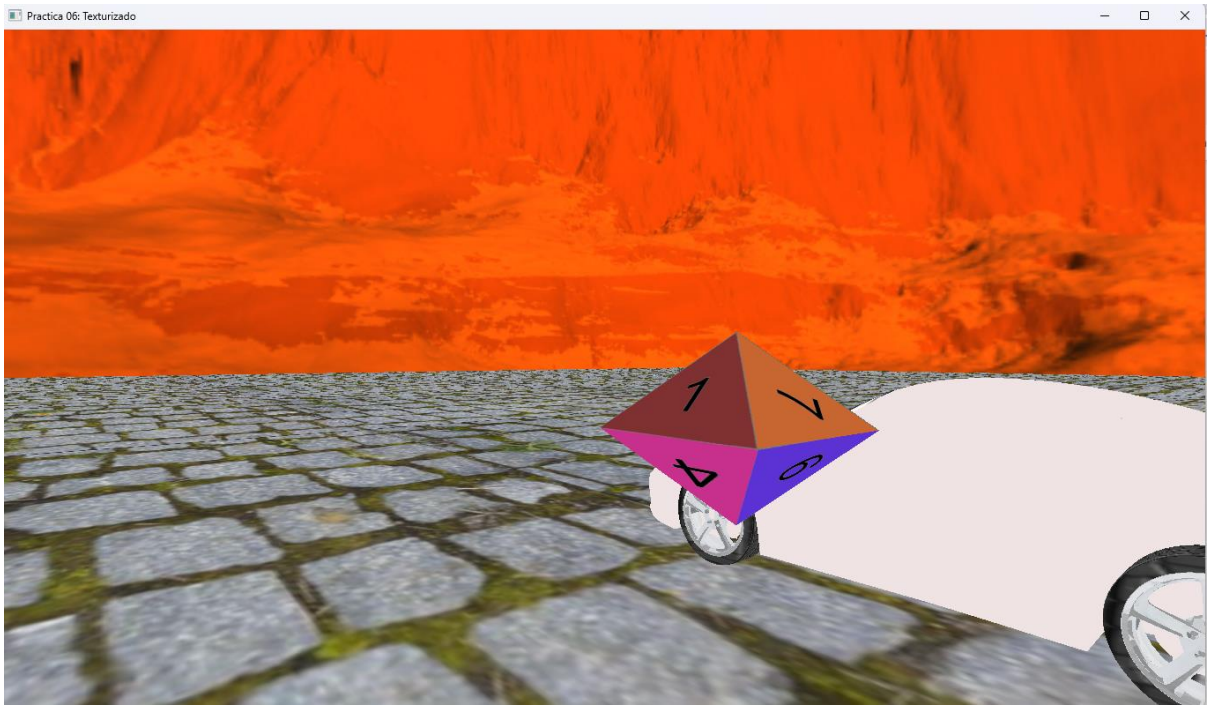
```
  Porsche2014Llanta_M = Model();
  Porsche2014Llanta_M.LoadModel("Models/porsche2014_llanta.obj");
  Porsche2014Carro_M = Model();
  Porsche2014Carro_M.LoadModel("Models/porsche2014_carro.obj");
```

```
Ejercicio 2 y 3: Importar el modelo de su coche con sus 4 llantas acomodadas
y tener texturizadas las 4 llantas (diferenciar caucho y rin)  y
texturizar el logo de la Facultad de ingeniería en el cofre de su propio modelo de coche
Agregar los ojos del auto a nuestro modelo y los detalles de la parrilla
*/

// Instancia de mi Porsche 2014
// Porsche 2014 carro
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, -1.1f, -6.0f));
model = glm::scale(model, glm::vec3(5.0f, 5.0f, 5.0f));
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Porsche2014Carro_M.RenderModel();

// llanta trasera derecha
model = modelaux;
model = glm::translate(model, glm::vec3(0.77f, 0.18f, 1.3f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Porsche2014Llanta_M.RenderModel();

//llanta trasera izquierda
model = modelaux;
model = glm::translate(model, glm::vec3(-0.77f, 0.18f, 1.3f));
model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Porsche2014Llanta_M.RenderModel();

//llanta delantera derecha
model = modelaux;
model = glm::translate(model, glm::vec3(0.74f, 0.17f, -1.17f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Porsche2014Llanta_M.RenderModel();

// llanta delantera derecha
model = modelaux;
model = glm::translate(model, glm::vec3(-0.74f, 0.17f, -1.17f));
model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Porsche2014Llanta_M.RenderModel();
```

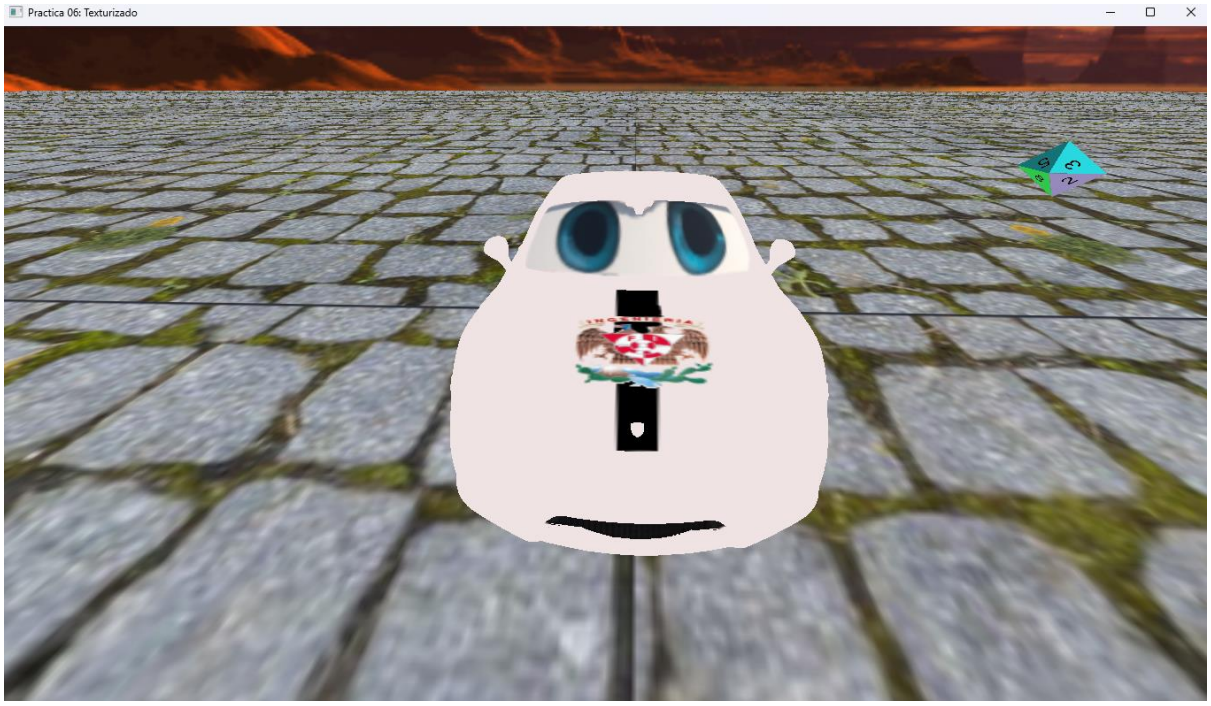
## Ejecución

## Actividad 1



**Actividad 2 y 3**





## Problemas presentados

En la práctica surgieron 2 problemas principales, el primero fue en la parte que inicialmente se pensó que era Open GL, sin embargo, al final me di cuenta de que era en el modelo.

Lo que pasó fue que se exportaban los modelos aparentemente bien, sin embargo, al cargar el modelo en open GL, el auto si cargaba la textura de la parrilla y el cofre, pero no la de los ojos, y se revisó el UV Map, la imagen de la textura y todo era

correcto, sin embargo, al revisar el modelo, me di cuenta de que en el modelo, había un “fondo” del para brizas, con la misma forma, y estaban exactamente en la misma posición, por lo que al exportar el modelo y renderizarlo, se traslapaban y solo se veía sin textura, para resolver esto, trasladé el parabrisas texturizado un poco más arriba, para que  no se “traslape” con la otra figura, y de esta manera funcionó.

El 2do error fue en la parte de open GL, ya que, en las llantas, se exportó el modelo de manera correcta, con las texturas, sin embargo, en open GL no se veían texturizadas, lo que pasó fue que las llantas tenían color negro, y al parecer, para que se vean las texturas hay que eliminar los colores, así que para solucionar esto, eliminé las líneas para ponerle color al objeto, de esta manera se mostró correctamente la textura.

**Conclusión:**

La práctica fue bastante interesante y útil, ayudó a saber mucho más sobre texturizado, en este caso con el primer ejercicio al texturizado a través de comandos con ayuda de Open GL, y en el 2do y 3er caso, con ayuda de blender. Otra cosa a la que nos ayudó fue a repasar los temas anteriores, cómo la carga de modelos, generar figuras a través de vértices e índices en OpenGL y las transformaciones, así que me pareció que fue bastante buena.

La complejidad me pareció moderada o incluso alta, más que nada porque el tiempo que hay que dedicarle, en el caso de la primera actividad a la creación del dado de 8 caras, con los vértice e índices, y también en cuanto a posicionar la textura con las coordenadas UV. En cuanto a la 2da parte, lo complicado fue el texturizado de blender, más que nada aprender sobre la herramienta, ya que por ejemplo en mi modelo, texturizaba un elemento, pero todos los elementos del auto con ese mismo material tomaban la textura y no era lo que se buscaba, así que tardé bastante encontrando una manera.

En cuanto a los comentarios, en esta ocasión, considero que se debería explicar más en el laboratorio a cerca del texturizado en blender con un modelo más complejo, ya que el cubo era bastante sencillo y no había problemas, sin embargo, con modelos más complejos con más polígonos o formas irregulares o con modelos con varias piezas, se complica trabajar el texturizado, entonces creo que sería algo útil ver un ejemplo así en el laboratorio.

# Bibliografía


- Gandalf3. (2014). “How to properly unwrap my mesh?”. Recuperado el 11 de octubre de 2025 de: https://blender.stackexchange.com/questions/6755/how-to-properly-unwrap-my-mesh