



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° 6**

**NOMBRE COMPLETO:** Araiza Valdés Diego Antonio

**N° de Cuenta:** 423032833

**GRUPO DE LABORATORIO:** 02

**GRUPO DE TEORÍA:** 06

**SEMESTRE:** 2026-1

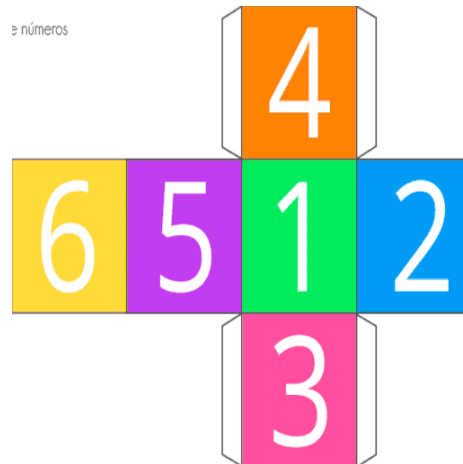
**FECHA DE ENTREGA LÍMITE:** 07/10/2026

**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

- 1.1.Actividad 1: Texturizar su cubo con la imagen dado\_animales o dado\_numeros ya optimizada por ustedes



*Primero, dentro de la aplicación Gimp, editamos y ajustamos la imagen para que no considerara las pestañas del cubo ni tuviera imágenes, cambiamos el tamaño a 512x512 px., activamos el canal Alpha y damos una transparencia del 22% al número 6. Lo exportamos como .png.*

```
brickTexture = Texture("Textures/brick.png");
brickTexture.LoadTextureA();
dirtTexture = Texture("Textures/dirt.png");
dirtTexture.LoadTextureA();
plainTexture = Texture("Textures/plain.png");
plainTexture.LoadTextureA();
pisoTexture = Texture("Textures/piso.tga");
pisoTexture.LoadTextureA();
dadoTexture = Texture("Textures/DadoNumTra.png");
dadoTexture.LoadTextureA();
logofiTexture = Texture("Textures/escudo_fi_color.tga");
logofiTexture.LoadTextureA();
```

*Luego incluimos la imagen a la carpeta Textures y cambiamos el nombre en el código.*

```

// front
//x      y      z      S      T      NX      NY      NZ
-0.5f, -0.5f, 0.5f, 0.26f, 0.34f, 0.0f, 0.0f, -1.0f, //0
0.5f, -0.5f, 0.5f, 0.49f, 0.34f, 0.0f, 0.0f, -1.0f, //1
0.5f, 0.5f, 0.5f, 0.49f, 0.66f, 0.0f, 0.0f, -1.0f, //2
-0.5f, 0.5f, 0.5f, 0.26f, 0.66f, 0.0f, 0.0f, -1.0f, //3
// right
//x      y      z      S      T
0.5f, -0.5f, 0.5f, 0.51f, 0.34f, -1.0f, 0.0f, 0.0f,
0.5f, -0.5f, -0.5f, 0.74f, 0.34f, -1.0f, 0.0f, 0.0f,
0.5f, 0.5f, -0.5f, 0.74f, 0.65f, -1.0f, 0.0f, 0.0f,
0.5f, 0.5f, 0.5f, 0.51f, 0.65f, -1.0f, 0.0f, 0.0f,
// back
-0.5f, -0.5f, -0.5f, 0.99f, 0.34f, 0.0f, 0.0f, 1.0f,
0.5f, -0.5f, -0.5f, 0.76f, 0.34f, 0.0f, 0.0f, 1.0f,
0.5f, 0.5f, -0.5f, 0.76f, 0.65f, 0.0f, 0.0f, 1.0f,
-0.5f, 0.5f, -0.5f, 0.99f, 0.65f, 0.0f, 0.0f, 1.0f,

// left
//x      y      z      S      T
-0.5f, -0.5f, -0.5f, 0.0f, 0.34f, 1.0f, 0.0f, 0.0f,
-0.5f, -0.5f, 0.5f, 0.24f, 0.34f, 1.0f, 0.0f, 0.0f,
-0.5f, 0.5f, 0.5f, 0.24f, 0.65f, 1.0f, 0.0f, 0.0f,
-0.5f, 0.5f, -0.5f, 0.0f, 0.65f, 1.0f, 0.0f, 0.0f,

// bottom
//x      y      z      S      T
-0.5f, -0.5f, 0.5f, 0.51f, 0.0f, 0.0f, 1.0f, 0.0f,
0.5f, -0.5f, 0.5f, 0.51f, 0.33f, 0.0f, 1.0f, 0.0f,
0.5f, -0.5f, -0.5f, 0.74f, 0.33f, 0.0f, 1.0f, 0.0f,
-0.5f, -0.5f, -0.5f, 0.74f, 0.0f, 0.0f, 1.0f, 0.0f,

//UP
//x      y      z      S      T
-0.5f, 0.5f, 0.5f, 0.51f, 0.99f, 0.0f, -1.0f, 0.0f,
0.5f, 0.5f, 0.5f, 0.51f, 0.67f, 0.0f, -1.0f, 0.0f,
0.5f, 0.5f, -0.5f, 0.74f, 0.67f, 0.0f, -1.0f, 0.0f,
-0.5f, 0.5f, -0.5f, 0.74f, 0.99f, 0.0f, -1.0f, 0.0f,

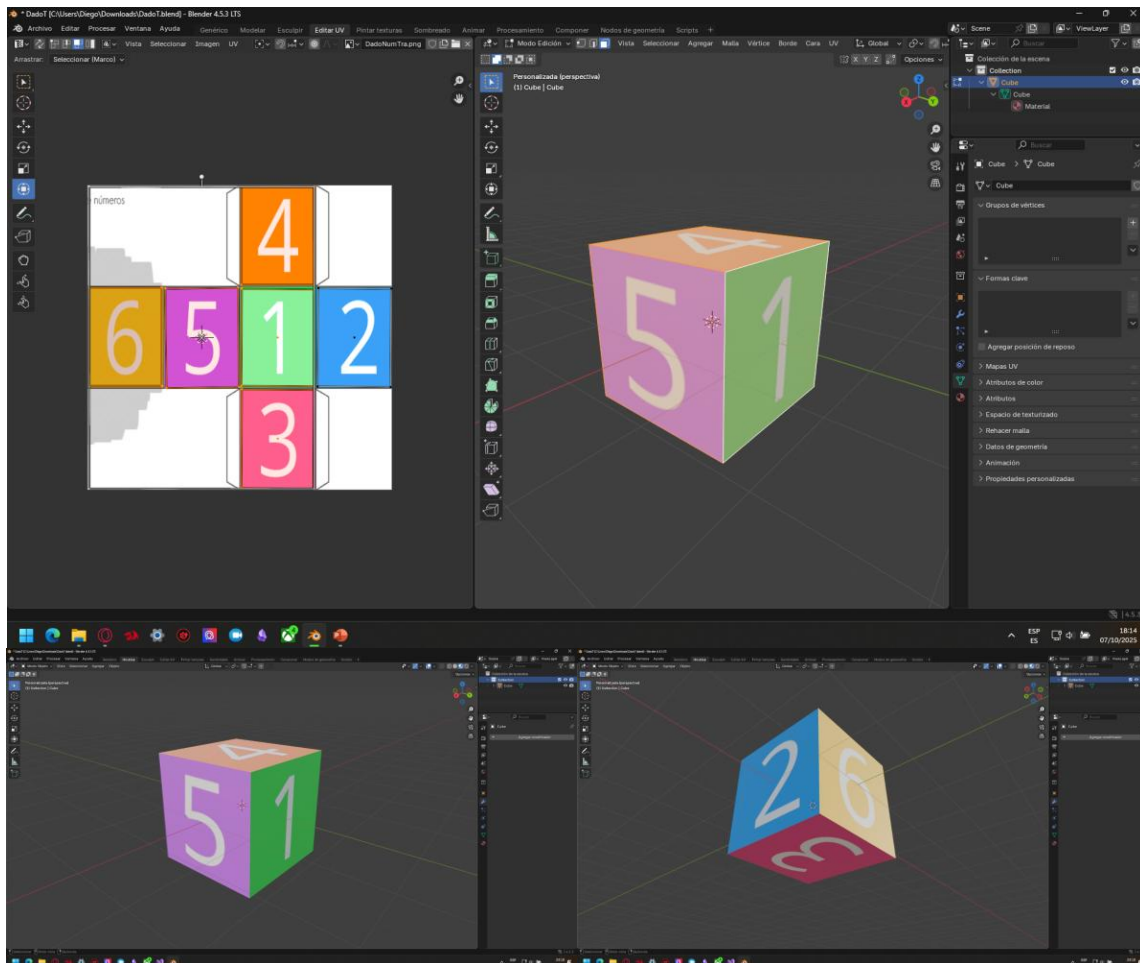
```

*Posteriormente modificamos los valores de S y T para todas las caras excepto la frontal para acomodar los demás valores numéricos*

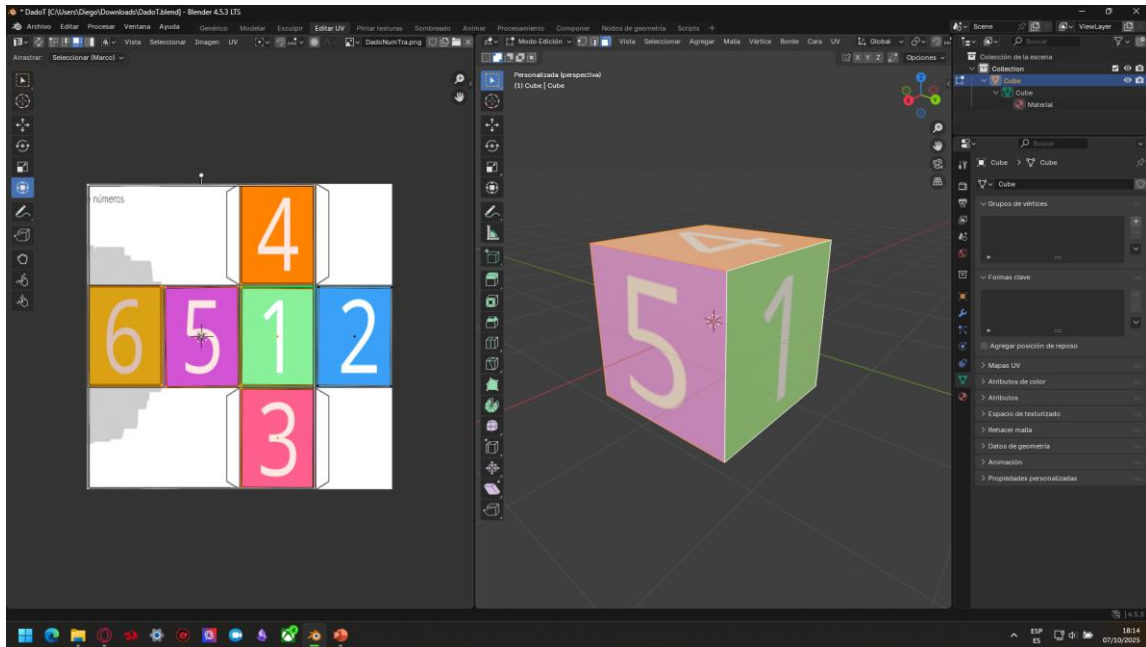
```
//Dado de Opengl
//Ejercicio 1: Texturizar su cubo con la imagen dado_animales ya optimizada por ustedes
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-1.5f, 4.5f, -2.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
dadoTexture.UseTexture();
meshList[4]->RenderMesh();
```

*Finalmente declaramos y trasladamos el cubo, además habilitamos la transparencia y le asignamos la textura de la imagen que importamos anteriormente.*

1.2.Actividad 2: Importar el cubo texturizado en el programa de modelado con la imagen dado\_animales o dado numeros ya optimizada por ustedes







Primero, con la imagen creada anteriormente, le damos textura al cubo, acomodando cada una de sus caras individualmente y lo exportamos como .obj

```
Model Kitt_M;
Model Llanta_M;
Model Dado_M;
```

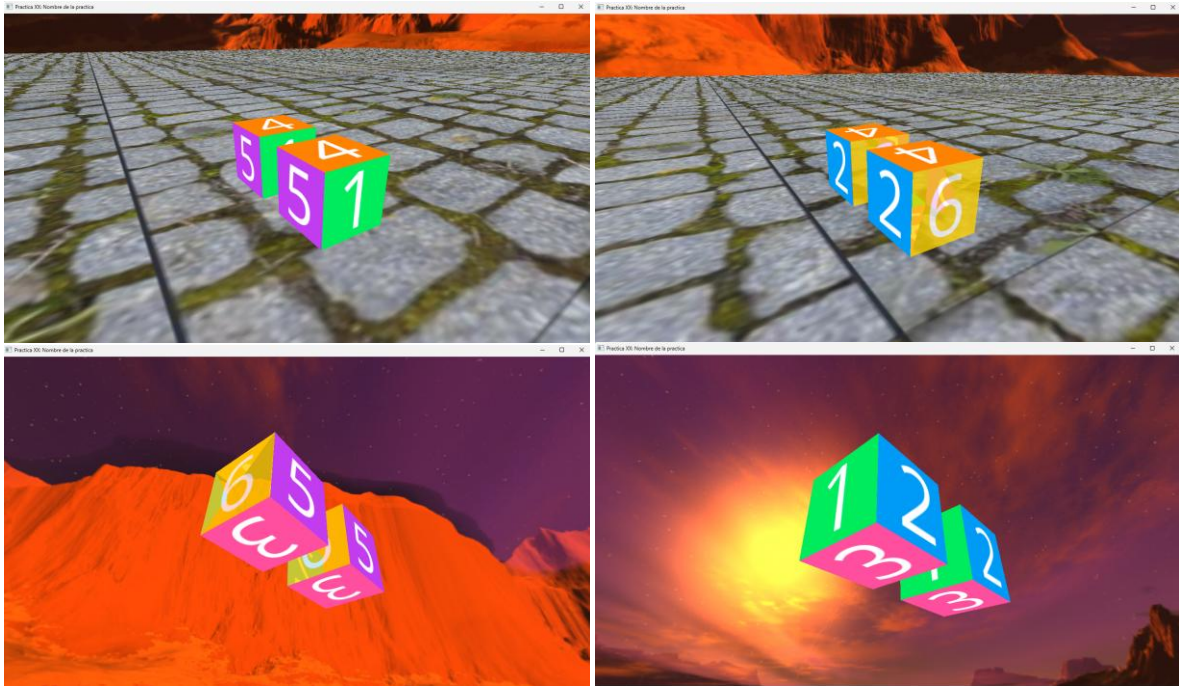
```
Kitt_M = Model();
Kitt_M.LoadModel("Models/kitt_optimizado.obj");
Llanta_M = Model();
Llanta_M.LoadModel("Models/llanta_optimizada.obj");
Dado_M = Model();
Dado_M.LoadModel("Models/Dado.obj");
```

Posteriormente declaramos e inicializamos el modelo con el nombre con el que lo exportamos anteriormente.

```
//Ejercicio 2: Importar el cubo texturizado en el programa de modelado con
//la imagen dado_animales ya optimizada por ustedes
//Dado importado
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-3.0f, 4.5f, -2.0f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Dado_M.RenderModel();
```

*Finalmente declaramos, trasladamos, rotamos y escalamos el cubo para luego dibujarlo.*

### 1.3. Evidencias:



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

R: No hubo ningún problema.

3. Conclusión:

- 3.1. Los ejercicios de la clase: Complejidad, explicación

El ejercicio fue bastante adecuado, no hubo mayor complicación para su realización. Solo se tuvo que instalar Gimp.

- 3.2. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias.

Respecto a la explicación, fue bastante buena, entendí a detalle como trabajar con texturas en Blender, y las preguntas que tuve fueron aclaradas al momento. Además, entendí el funcionamiento básico de Gimp. Todo esto me permitió realizar sin problema el ejercicio en mi casa. Lo único que no recordé fue como realizar la exportación, pero los videos fueron de mucha ayuda.