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 01

NOMBRE COMPLETO: Quintos Delgadillo Axel Alejandro

Nº de Cuenta: 319312544

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 06

SEMESTRE 2026-1

FECHA DE ENTREGA LÍMITE: 19/08/2025

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.**

Cambio de color de fondo:

- Se implementó la función changeColor() que alterna el color de fondo de la ventana entre rojo, verde y azul de manera cíclica.
- El cambio de color se realiza con un periodo de tiempo adecuado para que sea perceptible por el ojo humano.

```
void changeColor() {
    double time = glfwGetTime();
    int color = ((int)time) % 3;

    if (color == 0) {
        red = 1.0f; green = 0.0f; blue = 0.0f;
    }
    else if (color == 1) {
        red = 0.0f; green = 1.0f; blue = 0.0f;
    }
    else if (color == 2) {
        red = 0.0f; green = 0.0f; blue = 1.0f;
    }
}
```

La función changeColor() cambia de manera cíclica el color de fondo de la ventana entre rojo, verde y azul. Para ello, obtiene el tiempo transcurrido desde que se inició la ventana con glfwGetTime(), lo convierte a entero y calcula el residuo al dividirlo entre 3, generando así un valor que se repite continuamente (0, 1, 2). Dependiendo de este valor, se asignan los valores correspondientes de rojo, verde o azul a las variables red, green y blue, logrando que el fondo cambie de color automáticamente con un periodo perceptible al ojo humano.

Dibujo de figuras geométricas:

- Se dibujaron de forma simultánea un cuadrado y un rombo en la ventana de OpenGL.

```
GLfloat vertices[] = {
    0.5f,0.5f, 0.5f,
    0.0f,0.0f,0.5f,
    1.0f,0.0f,0.5f,
    0.5f,-0.5f, 0.5f,
    1.0f,0.0f,0.5f,
    0.0f,0.0f,0.5f,

    -1.0f,-0.5f, -0.5f,
    0.0f,-0.5f,-0.5f,
    0.0f,0.5f,-0.5f,
    -1.0f,-0.5f, -0.5f,
    0.0f,0.5f,-0.5f,
    -1.0f,0.5f,-0.5f,
};
```

El arreglo vértices contiene las coordenadas 3D de los vértices que se usan para dibujar las figuras en la ventana de OpenGL. Cada grupo de tres valores representa las coordenadas x, y, z de un vértice. Los primeros seis vértices corresponden a dos triángulos que forman un cuadrado, mientras que los siguientes seis forman un rombo, también mediante dos triángulos. Los valores positivos y negativos de las coordenadas permiten posicionar las figuras en diferentes lugares dentro del espacio de la ventana, y estos vértices se utilizan luego en un VBO para que OpenGL pueda renderizarlos correctamente.

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

Solo tuve problemas al momento de generar el repositorio ya que nunca lo había ocupado, respecto al programa lo estoy entendiendo adecuadamente.

3. **Conclusión.**
    a. Los ejercicios de la clase: Complejidad, explicación

    Los ejercicios de la clase resultaron de complejidad moderada, ya que implicaron comprender cómo funciona el pipeline de OpenGL, la interacción entre CPU y GPU mediante VBO y VAO, y la escritura de shaders en GLSL. Implementar el cambio de color cíclico y dibujar figuras geométricas simultáneamente permitió reforzar conceptos clave de gráficos por computadora, como el manejo de vértices, coordenadas 3D y renderizado en tiempo real. Además, estas prácticas ayudan a familiarizarse con la gestión de buffers y el ciclo de renderizado de una ventana interactiva.

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

Considero que las explicaciones podrían haberse dado más lentamente y con mayor detalle en algunos puntos, especialmente al tratar los shaders y la carga de vértices, ya que son conceptos nuevos para muchos estudiantes. Sería útil incluir ejemplos visuales de cómo se construyen las figuras y cómo se ve el cambio de color en cada frame. Además, un repaso sobre la diferencia entre el color de fondo (glClearColor) y el color de los objetos en el fragment shader ayudaría a comprender mejor los resultados visuales del programa. En general, la sesión fue clara y práctica, y permitió experimentar con los conceptos de gráficos en tiempo real de forma directa y visual.