



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° 1

NOMBRE COMPLETO: Mino Guzmán Yara Amairani

N° de Cuenta: 422017028

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 11 febrero 2025

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

1. Actividades realizadas

- Actividad 1: Cambiar el color de fondo de la pantalla entre rojo, verde y azul de forma cíclica y solamente mostrando esos 3 colores con un periodo de lapso adecuado para el ojo humano.

```
// Función para actualizar el color de fondo cada 1 segundo
void updateBackgroundColor() {
    switch (colorIndex) {
        case 0: glClearColor(1.0f, 0.0f, 0.0f, 1.0f); break; // Rojo
        case 1: glClearColor(0.0f, 1.0f, 0.0f, 1.0f); break; // Verde
        case 2: glClearColor(0.0f, 0.0f, 1.0f, 1.0f); break; // Azul
    }
    colorIndex = (colorIndex + 1) % 3; // Pasar al siguiente color
}
```

Esta función cambia el color de fondo cíclicamente entre rojo, verde y azul, según el índice `colorIndex` y, cada que la función es llamada, se ajusta el color de fondo utilizando `glClearColor`.

```
int colorIndex = 0; //para alternar entre colores

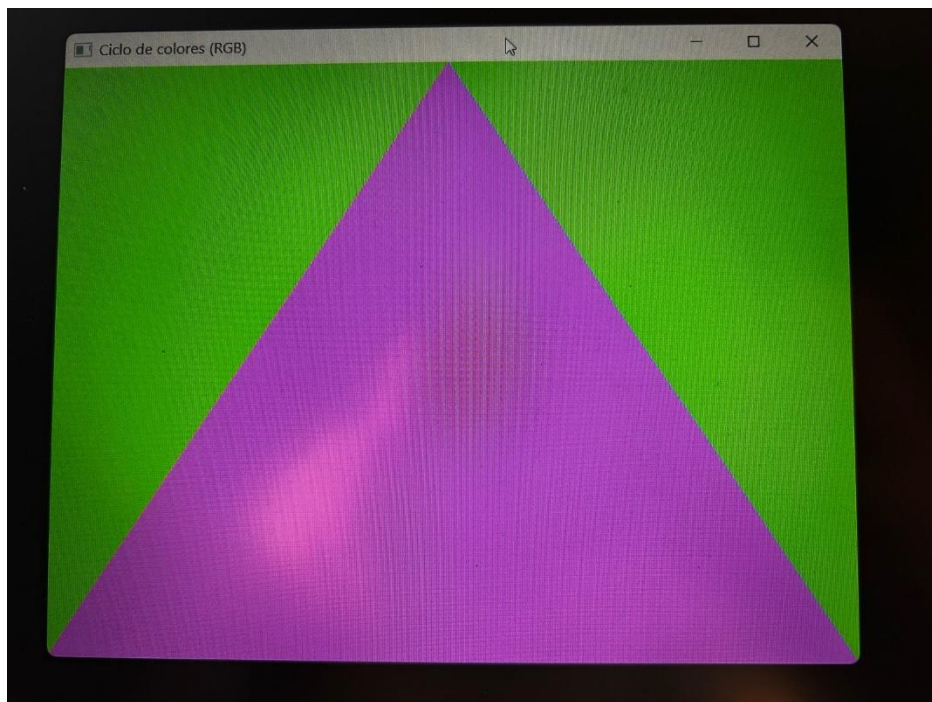
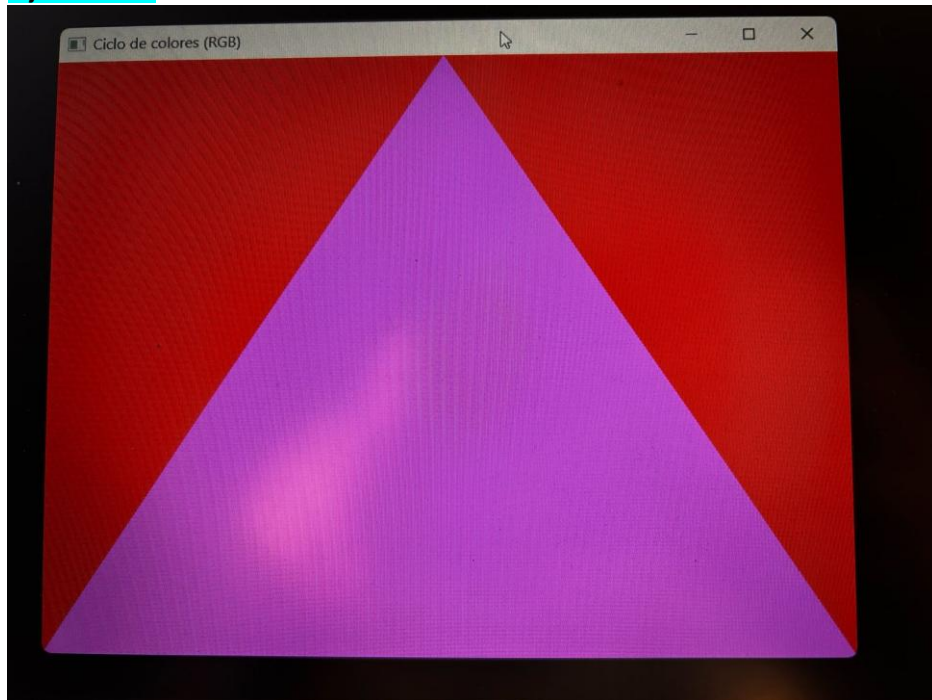
while (!glfwWindowShouldClose(mainWindow))
{
    glfwPollEvents();

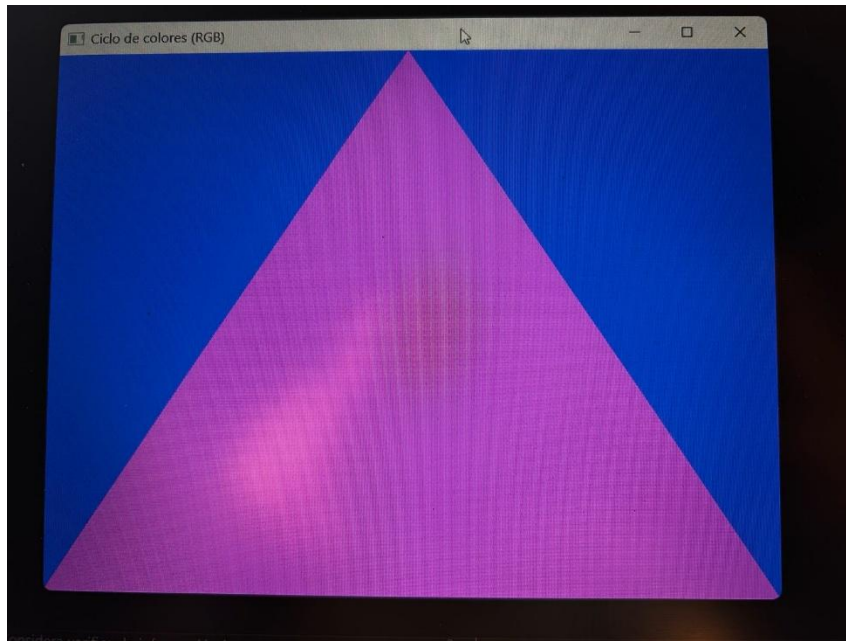
    // Verificar si ha pasado 1 segundo para cambiar de color
    double currentTime = glfwGetTime();
    if (currentTime - lastTime >= 1.0) {
        updateBackgroundColor();
        lastTime = currentTime;
    }

    glClear(GL_COLOR_BUFFER_BIT);
}
```

Se utiliza este ciclo while para que cambie el color de fondo cada que haya pasado 1 segundo.

Ejecución:





- Dibujar de forma simultánea en la ventana 1 cuadrado y 1 rombo separados.

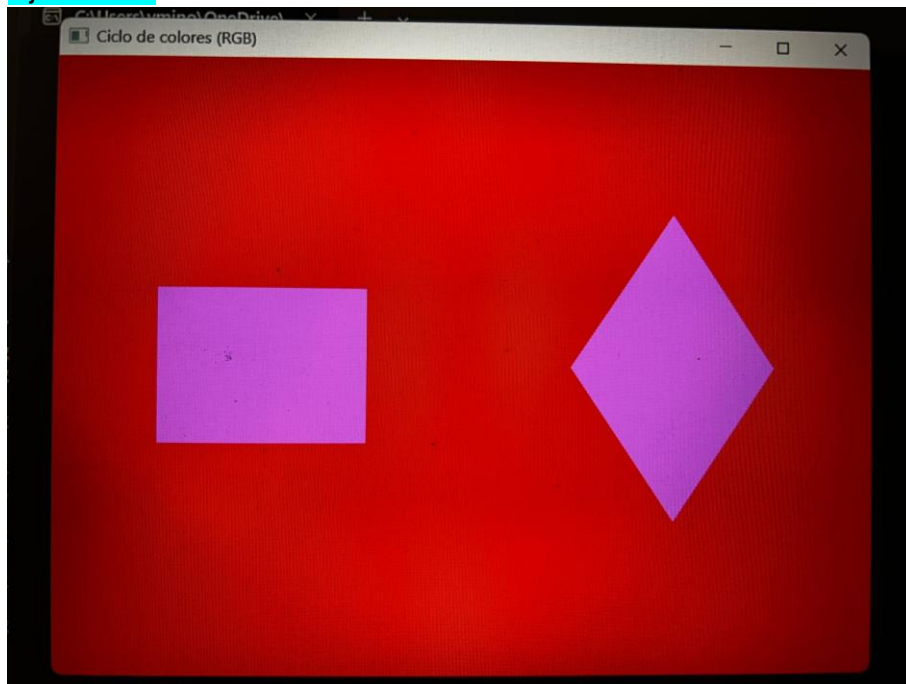
```
void CrearTriangulo()
{
    GLfloat vertices[] = {
        //Primer triangulo
        -0.75f, 0.25f, 0.0f,
        -0.25f, 0.25f, 0.0f,
        -0.75f, -0.25f, 0.0f,
        //Segundo triangulo
        -0.25f, 0.25f, 0.0f,
        -0.25f, -0.25f, 0.0f,
        -0.75f, -0.25f, 0.0f,
        //Tercer triangulo
        0.25f, 0.0f, 0.0f,
        0.5f, 0.5f, 0.0f,
        0.5f, -0.5f, 0.0f,
        //Cuarto triangulo
        0.75f, 0.0f, 0.0f,
        0.5f, 0.5f, 0.0f,
        0.5f, -0.5f, 0.0f
    };
}
```

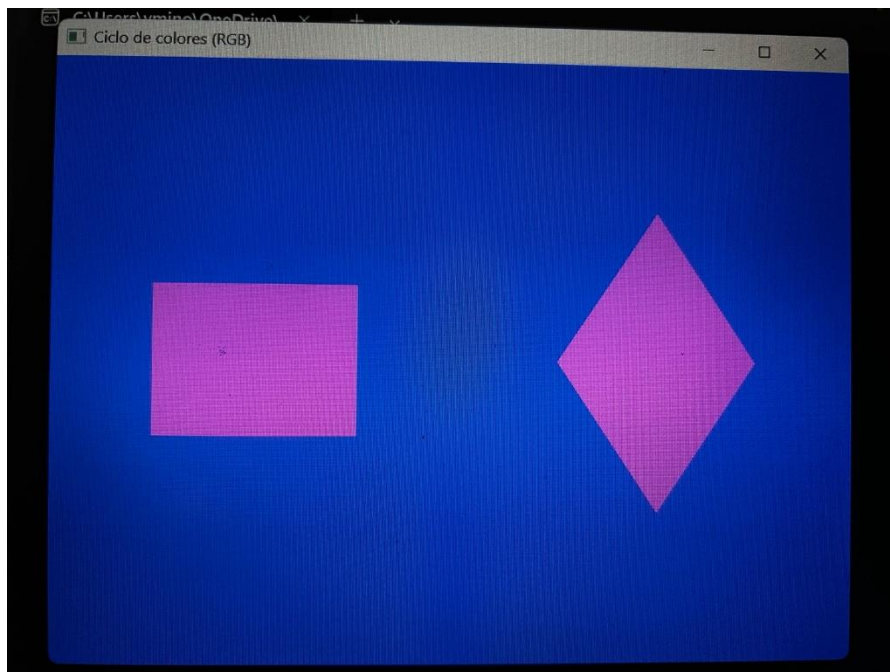
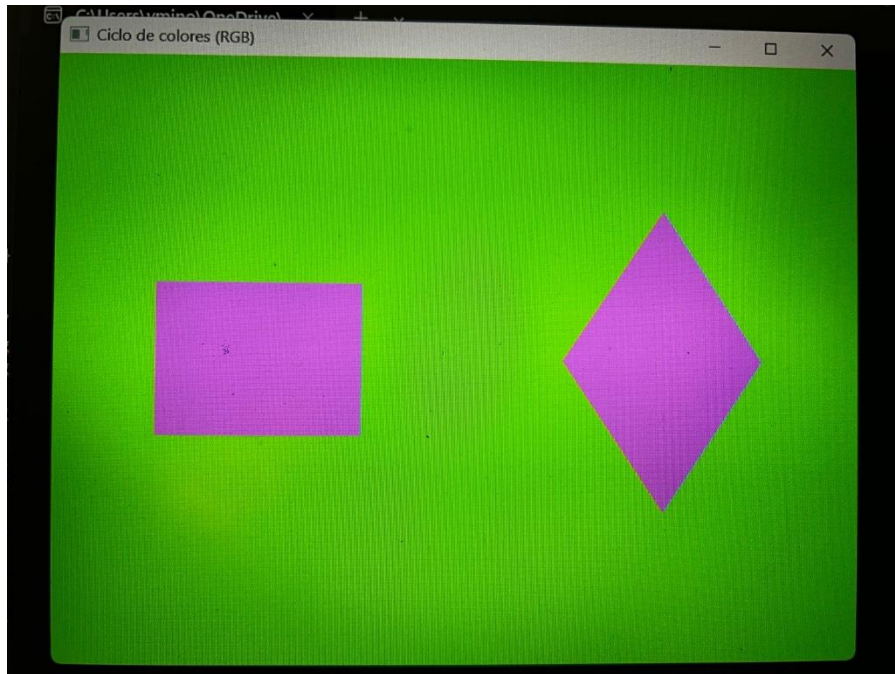
Se crearon 4 triángulos, el cuadrado utilizó 2 triángulos y el rombo utilizó 2 triángulos, las coordenadas del cuadrado fueron para que el cuadrado se mostrara del lado izquierdo y las coordenadas del rombo fueron para que se mostrara del lado derecho.

```
glUseProgram(shader);  
glBindVertexArray(VAO);  
glDrawArrays(GL_TRIANGLES, 0, 12);  
glBindVertexArray(0);  
glUseProgram(0);
```

En `glDrawArrays`, se puso el número 12 ya que debido a que un triángulo tiene 3 vértices y se utilizaron 4 triángulos, se tienen que mostrar 12 vértices, si no cambiamos ese número y sigue en 3, únicamente se mostrará un triángulo.

Ejecución:





2. Problemas presentados.

Me costó trabajo entender ciertas palabras reservadas que hay en C++, ya que no todas las que existen en C, existen en C++ y eso provocó que me tardara ya que tenía que estar investigando sobre palabras reservadas o algunas funciones.

3. Conclusión:

- a. Los ejercicios de la clase no estuvieron complicados, pero debido a que nunca había programado en C++, hizo que se volviera complicado para mí.
- b. Faltó explicar a detalle la práctica ya que siento que el tiempo fue poco, y también se podría explicar más a detalle lo que cambia de un lenguaje de programación a otro (C y C++). Los ejercicios hicieron que comprendiera mejor de cierta manera el uso de VisualStudio y que me familiarizara un poco más con la Computación Gráfica.