



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° 01**

**NOMBRE COMPLETO:** Nava Alberto Vanessa

**N° de Cuenta:** 318263522

**GRUPO DE LABORATORIO:** 02

**GRUPO DE TEORÍA:** 04

**SEMESTRE 2024-2**

**FECHA DE ENTREGA LÍMITE:** 17 de febrero de 2024

**CALIFICACIÓN:** \_\_\_\_\_

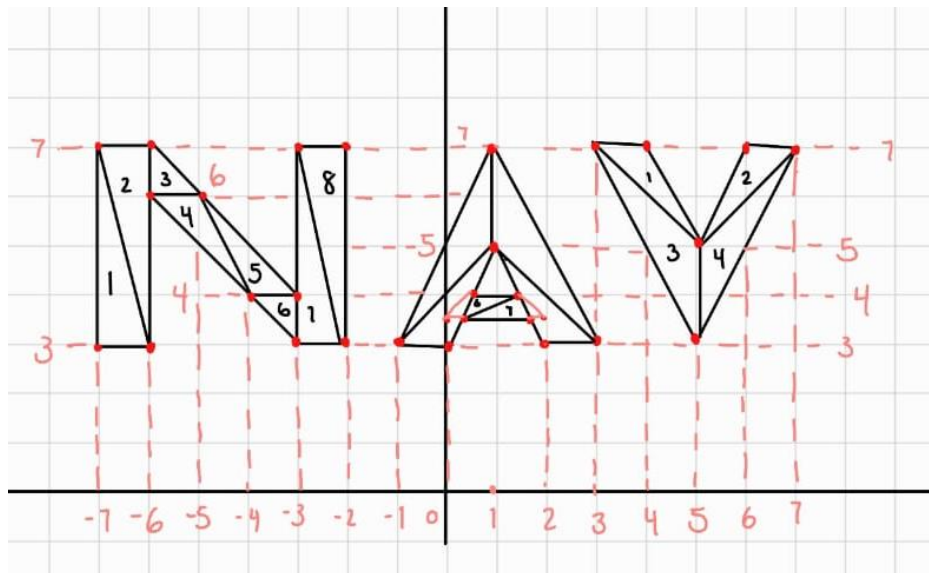
## Ejercicios

1. Cambio de color de fondo. Cambiar el color del fondo de la ventana de forma random en las gamas de colores RGB cada 3 segundos.
2. Letras del nombre. Letras iniciales de mi nombre (3 letras diferentes) dibujadas con triángulos del mismo color y acomodadas en la ventana que va en un rango de (-1, -1.0) hasta (1, 1.0).

## Desarrollo

Para la correcta solución de los ejercicios planteados por el profesor, primero modifiqué el código "segundo\_main.cpp" en el cual primero se modificaron los vértices de los triángulos por dibujar, para así poder dibujar las letras.

Para facilitarme la tarea de saber cada vértice de las letras por dibujar, realicé un plano en donde dibujé las letras y fui obteniendo las coordenadas de cada vértice:



De esta manera fue más sencillo obtener y establecer los vértices. El total de los vértices utilizados fue de 54 definidos en la función `glDrawArrays()`.

Para la letra "N" se utilizaron 24 vértices, para la letra "A" se utilizaron 18 vértices y para la letra "V" se utilizaron 12 vértices.

Seguido de esto, para poder modificar los colores del fondo se agregaron tres nuevas variables de tipo flotantes (rojo, verde y azul) que nos van a permitir almacenar un valor random que nos de la función `rand() % 3` (%3 le indica al programa que sean números entre 0 y 2) para así poder definir los números de cada color y según esto cada color será generado aleatoriamente.

También se agregó la biblioteca `<Windows.h>` para poder detener la pantalla durante dos segundos con la función `Sleep()`.

```

//Loop mientras no se cierra la ventana
while (!glfwWindowShouldClose(mainWindow))
{
    //Recibir eventos del usuario
    glfwPollEvents();

    rojo = rand() % 3;
    verde = rand() % 3;
    azul = rand() % 3;

    //Limpiar la ventana
    glClearColor(rojo, azul, verde, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);
    //Espera de 2 segundos para el cambio de color.
    Sleep(2000);
    glUseProgram(shader);

    glBindVertexArray(VAO);
    glDrawArrays(GL_TRIANGLES, 0, 54);
    glBindVertexArray(0);

    glUseProgram(0);

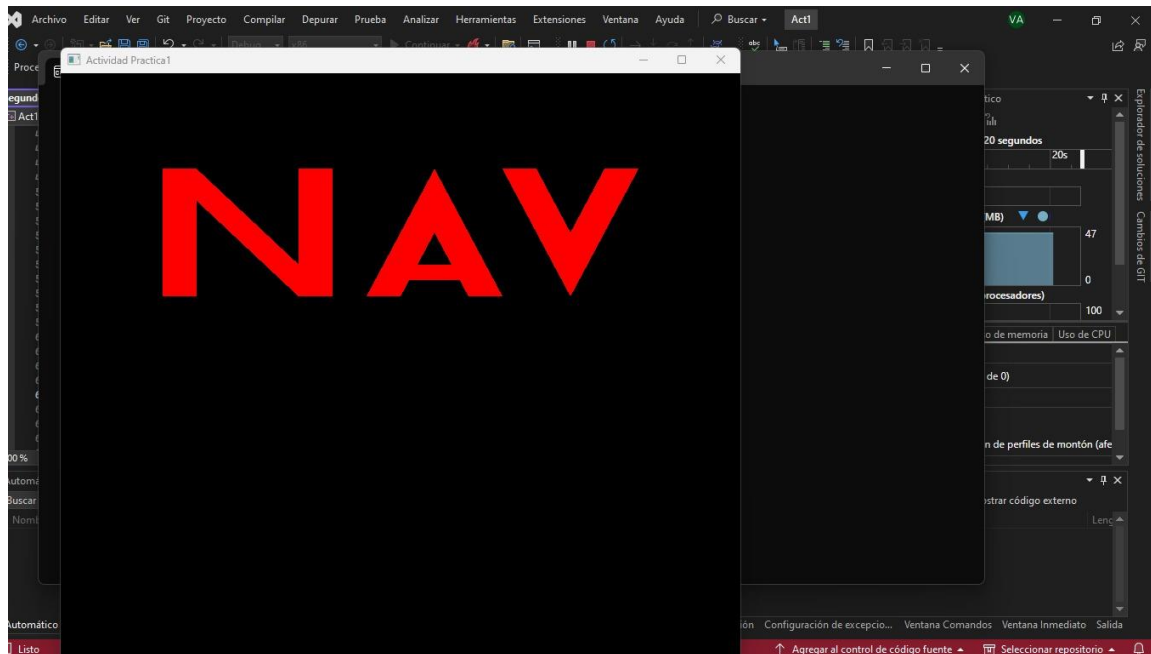
    glfwSwapBuffers(mainWindow);
}

```

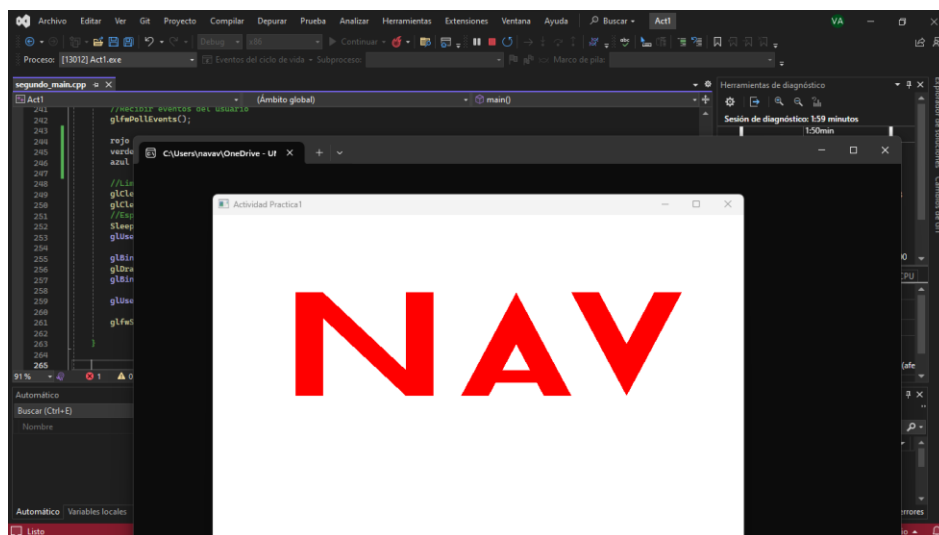
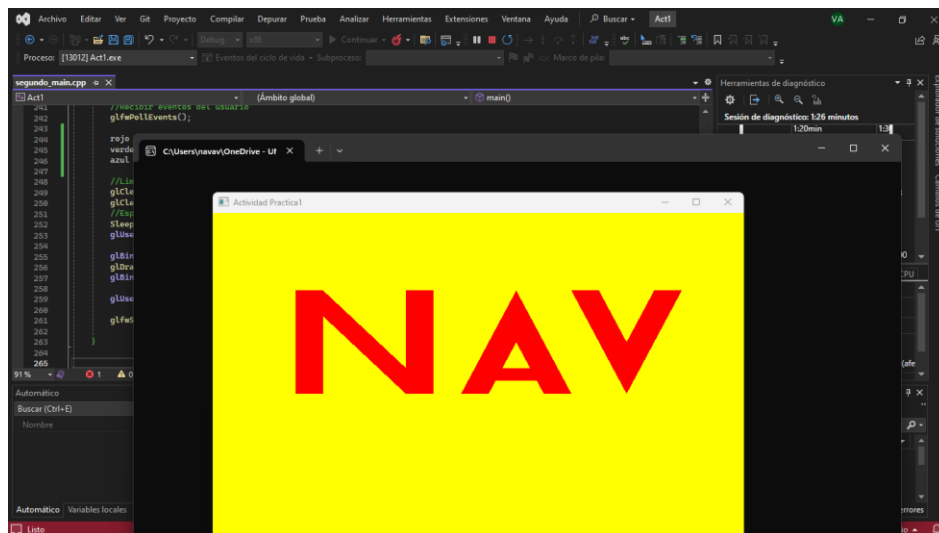
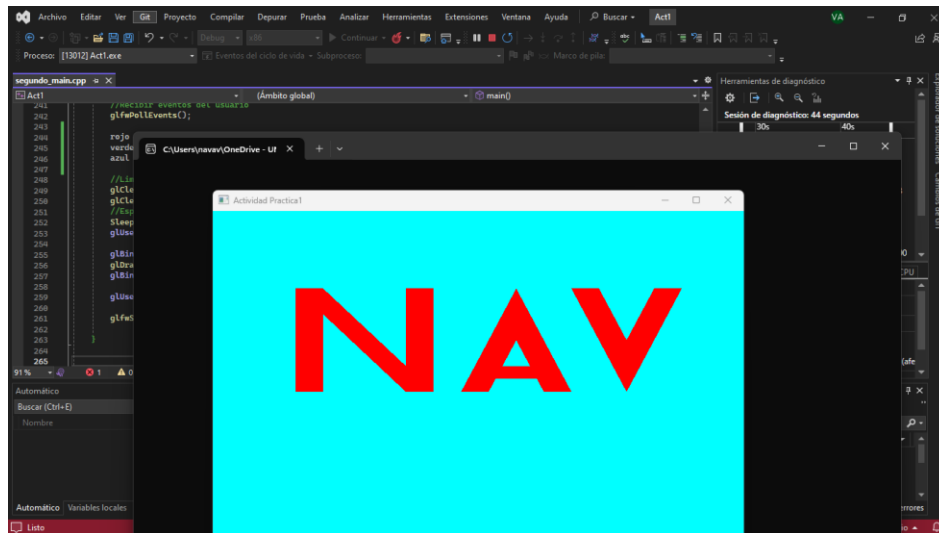
## Resultados

Los resultados sí fueron los esperados.

El resultado de dibujar las letras fue el siguiente:



El resultado de cambiar el color de fondo fue el siguiente:



### Obstáculos que se presentaron

Hubo un par de obstáculos que se presentaron, sin embargo, se logró solucionar el problema de estos sin mayor dificultad.

El primer obstáculo presentado fue del como hacer aleatorio el fondo. Al inicio había planteado una sentencia if, pero después me di cuenta de que estaba complicando el código más de lo que era, al final me di cuenta de que ni siquiera era necesario esa secuencia, por lo que lo terminé borrando, llegando a la versión actual del programa.

El segundo obstáculo fue cuando estaba dibujando las letras, porque sinceramente no comprendía de donde salían los vértices hasta que estuve investigando y comprendí que se puede comparar con un plano, lo que facilitó demasiado el dibujar las letras de manera sencilla.

### Conclusiones

Se lograron realizar los ejercicios propuestos por el profesor sin dificultad, fue una práctica sin mucha complejidad donde pude aprender más acerca de cómo es el funcionamiento básico de OpenGL junto con C++. En el laboratorio la explicación fue un poco rápida y llego un momento en el que deje de entender, sin embargo y por suerte no fue difícil de comprender después de preguntar las dudas que se me generaron.

### Bibliografía

Microsoft. (2023). *Rand*. Recuperado el 15 de febrero de 2024, de: <https://learn.microsoft.com/es-es/cpp/c-runtime-library/reference/rand?view=msvc-170>

Joey de Vires. (2013). *Hello Triangle*. Learning OpenGL. Recuperado el 15 de febrero de 2024, de: <https://learnopengl.com/Getting-started/Hello-Triangle>