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 01

NOMBRE COMPLETO: Aguilar Pérez José Ramón

Nº de Cuenta: 317515048

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 15/febrero/2025

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejercicios realizados.

- **Ventana cambia el color de fondo de forma random tomando rango de colores RGB y con una periodicidad de 2 segundos.**

Para la realización de este ejercicio, se agregaron las librerías *Windows.h* y *time.h*, ya que estas permiten utilizar las funciones de tiempo de espera y aleatoriedad, respectivamente.

```
#include <Windows.h> //Tiempo de espera entre camhio de color
#include <time.h> //randomizar colores
```

Dentro del main del programa, se agrega la función *srand()*, esto con el fin de asegurar que cada vez que se ejecuta el programa, se obtengan resultados diferentes.

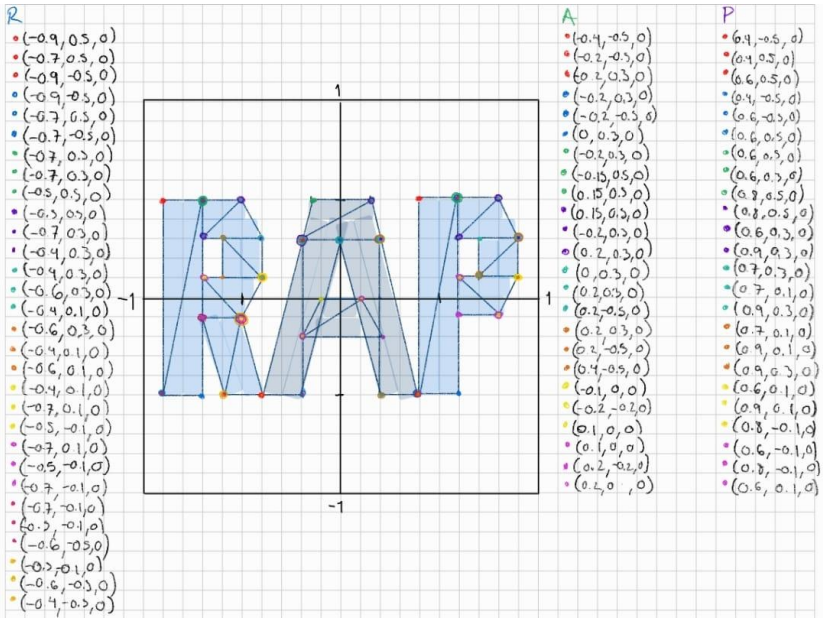
```
int main()
{
    srand(time(NULL)); //Asegura que por cada ejecucion salgan colores distintos
```

Para cambiar el fondo a colores aleatorios, se crean tres variables que representan los valores RGB del fondo, y se le asignan valores aleatorios entre los posibles 255 que pueden adoptar en este formato. Finalmente, se agrega la función *sleep()* para que el cambio de color se de cada 2 segundos.

```
//Variables para el fondo RGB
float rojo, verde, azul;

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

    //Asignacion de valores aleatorios para el RGB del fondo
    rojo = (float)(rand() % 256) / 255.0f;
    verde = (float)(rand() % 256) / 255.0f;
    azul = (float)(rand() % 256) / 255.0f;

    //Limpiar la ventana
    glClearColor(rojo, verde, azul, 1.0f);
    Sleep(2000); //Tarda 2 segundos en cambiar de un color a otro
    glClear(GL_COLOR_BUFFER_BIT);
```

- **3 letras iniciales de sus nombres creadas a partir de triángulos, todas las letras son del mismo color. Los dos ejercicios se muestran de forma simultánea y están en el mismo main.**

Para este ejercicio, se elaboró un boceto inicial con los vértices necesarios para dibujar las letras R, A y P. Fueron necesarios 78 vértices en total.



Se escribieron esto vértices en el programa, además de que estableció que el color de las letras sería un turquesa oscuro.

```
//R
-0.9f,0.5f,0.0f,
-0.7f,0.5f,0.0f,
-0.9f,-0.5f,0.0f,
-0.9f,-0.5f,0.0f,
-0.7f,0.5f,0.0f,
-0.7f,-0.5f,0.0f,
-0.7f,0.5f,0.0f,
-0.7f,0.3f,0.0f,
-0.5f,0.5f,0.0f,
-0.5f,0.5f,0.0f,
-0.7f,0.3f,0.0f,
-0.4f,0.3f,0.0f,
-0.6f,0.3f,0.0f,
-0.4f,0.1f,0.0f,
-0.4f,0.3f,0.0f,
-0.6f,0.1f,0.0f,
-0.4f,0.1f,0.0f,
-0.6f,0.3f,0.0f,
-0.7f,0.1f,0.0f,
-0.4f,0.1f,0.0f,
-0.5f,-0.1f,0.0f,
-0.7f,0.1f,0.0f,
-0.7f,-0.1f,0.0f,
-0.5f,-0.1f,0.0f,
-0.6f,-0.5f,0.0f,
-0.7f,-0.1f,0.0f,
-0.5f,-0.1f,0.0f,
-0.6f,-0.5f,0.0f,
-0.4f,-0.5f,0.0f,
-0.5f,-0.1f,0.0f,
```

```
//A
-0.4f,-0.5f,0.0f,
-0.2f,-0.5f,0.0f,
-0.2f,0.3f,0.0f,
-0.2f,0.3f,0.0f,
-0.2f,-0.5f,0.0f,
0.0f,0.3f,0.0f,
-0.2f,0.3f,0.0f,
-0.15f,0.5f,0.0f,
0.15f,0.5f,0.0f,
-0.2f,0.3f,0.0f,
0.2f,0.3f,0.0f,
0.15f,0.5f,0.0f,
0.0f,0.3f,0.0f,
0.2f,0.3f,0.0f,
0.2f,-0.5f,0.0f,
0.4f,-0.5f,0.0f,
0.2f,0.3f,0.0f,
0.2f,-0.5f,0.0f,
-0.1f,0.0f,0.0f,
-0.2f,-0.2f,0.0f,
0.1f,0.0f,0.0f,
0.2f,-0.2f,0.0f,
0.1f,0.0f,0.0f,
-0.2f,-0.2f,0.0f,
```

```
//P
0.4f,-0.5f,0.0f,
0.4f,0.5f,0.0f,
0.6f,0.5f,0.0f,
0.6f,-0.5f,0.0f,
0.4f,-0.5f,0.0f,
0.6f,0.5f,0.0f,
0.6f,0.3f,0.0f,
0.8f,0.5f,0.0f,
0.6f,0.5f,0.0f,
0.6f,0.3f,0.0f,
0.8f,0.5f,0.0f,
0.9f,0.3f,0.0f,
0.7f,0.3f,0.0f,
0.7f,0.1f,0.0f,
0.9f,0.3f,0.0f,
0.9f,0.1f,0.0f,
0.7f,0.1f,0.0f,
0.9f,0.3f,0.0f,
0.6f,0.1f,0.0f,
0.8f,-0.1f,0.0f,
0.9f,0.1f,0.0f,
0.6f,-0.1f,0.0f,
0.8f,-0.1f,0.0f,
0.6f,0.1f,0.0f,
```

```
void main()                                                 \n\
{                                                           \n\
    color = vec4(0.0f/255.0f,180.0f/255.0f,180.0f/255.0f,1.0f);
}";
```

Finalmente, se modificó el valor en la función *glDrawArrays()* para que los 78 vértices sean tomados en cuenta.

```
glBindVertexArray(VAO);
glDrawArrays(GL_TRIANGLES, 0, 78);
glBindVertexArray(0);
```

2.- Problemas presentados

Al momento de hacer los cambios de color, tanto para el fondo y las letras, se estaban empleando números mayores a 1, lo que provocaba que en la salida se obtuvieran puros blancos.

Esto se debe a que OpenGL solo maneja valores entre 0 y 1 para el RGB, por lo que la solución para esto fue dividir los distintos parámetros entre 255, para que así estuvieran en el rango de trabajo de OpenGL y así obtener los colores deseados.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

Los ejercicios presentados para esta práctica fueron interesantes de realizar, ya que permitían reutilizar conceptos de C++ vistos en cursos anteriores como lo es randomizar o “dormir” un proceso. Y gracias al ejercicio de clase, la realización de esta práctica se llevó sin mayores complicaciones.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica

La explicación de como generar las figuras fue precisa y concreta, así como la explicación de las funciones necesarias para la modificación correcta del programa.

c. Conclusión

Esta práctica me permitió conocer algunas de las funciones básicas de este nuevo ambiente gráfico de trabajo, además de que me ayudó a poner en práctica conocimientos que tenía un poco oxidados sobre C/C++. Además, los ejercicios realizados lograron que comprendiera mejor cómo funcionan los colores RGB en el ambiente de RGB, así como la importancia de los vértices para la creación de figuras.

1. Bibliografía en formato APA

- Abellán, J. (s/f). Obtención de Números Aleatorios en C. Chuidiang.org. Recuperado el 14 de febrero de 2025, de https://old.chuidiang.org/clinux/funciones/rand.php

- OpenGL Programming/Basics/Color. (s/f). Wikibooks.org. Recuperado el 14 de febrero de 2025, de https://en.wikibooks.org/wiki/OpenGL_Programming/Basics/Color

- Sleep function in C++. (2022, octubre 7). GeeksforGeeks. https://www.geeksforgeeks.org/sleep-function-in-cpp/