



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: Camacho Ignacio Violeta

Nº de Cuenta: 319061345

GRUPO DE LABORATORIO: 13

GRUPO DE TEORÍA: 06

SEMESTRE 2026-2

FECHA DE ENTREGA LÍMITE: 19-02-2026

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejecución de los ejercicios que se dejaron, comentar cada uno y capturas de pantalla de bloques de código generados y de ejecución del programa.

Ejercicio 1: Ventana cambia el color de fondo de forma random tomando rango de colores RGB y con una periodicidad de 2 segundos. (Verificar que al ejecutar el programa varias veces el orden de los colores si lo vean aleatorio y no siempre los mismos)

Ejercicio 2: las 3 letras iniciales de sus nombres creadas a partir de triángulos, acomodadas en forma diagonal de abajo hacia arriba, todas las letras son del mismo color.

Los dos ejercicios se muestran de forma simultánea y están en el mismo main

```
 2      #include <string.h>
 3      #include <stdlib.h>
 4      #include <time.h>
 5
 6
 7
 8
 9
10
11     // Variables para cambio de color en rgb
12     float lastTime = 0.0f;
13     float r = 0.0f, g = 0.0f, b = 0.0f;
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164     // Semilla diferente cada ejecución para evitar ciclos d reloj
165     srand((unsigned int)time(NULL));
166
167     CrearTriangulo();
168     CompileShaders();
169
170     while (!glfwWindowShouldClose(mainWindow))
171     {
172         glfwPollEvents();
173
174         float currentTime = glfwGetTime();
175
176         if (currentTime - lastTime >= 2.0f) //diferencia de dos segundos
177         {
178             r = (float)rand() / RAND_MAX;
179             g = (float)rand() / RAND_MAX;
180             b = (float)rand() / RAND_MAX;
181
182             lastTime = currentTime;
183         }
184
185         glClearColor(r, g, b, 1.0f);
186         glClear(GL_COLOR_BUFFER_BIT);
187     }
```

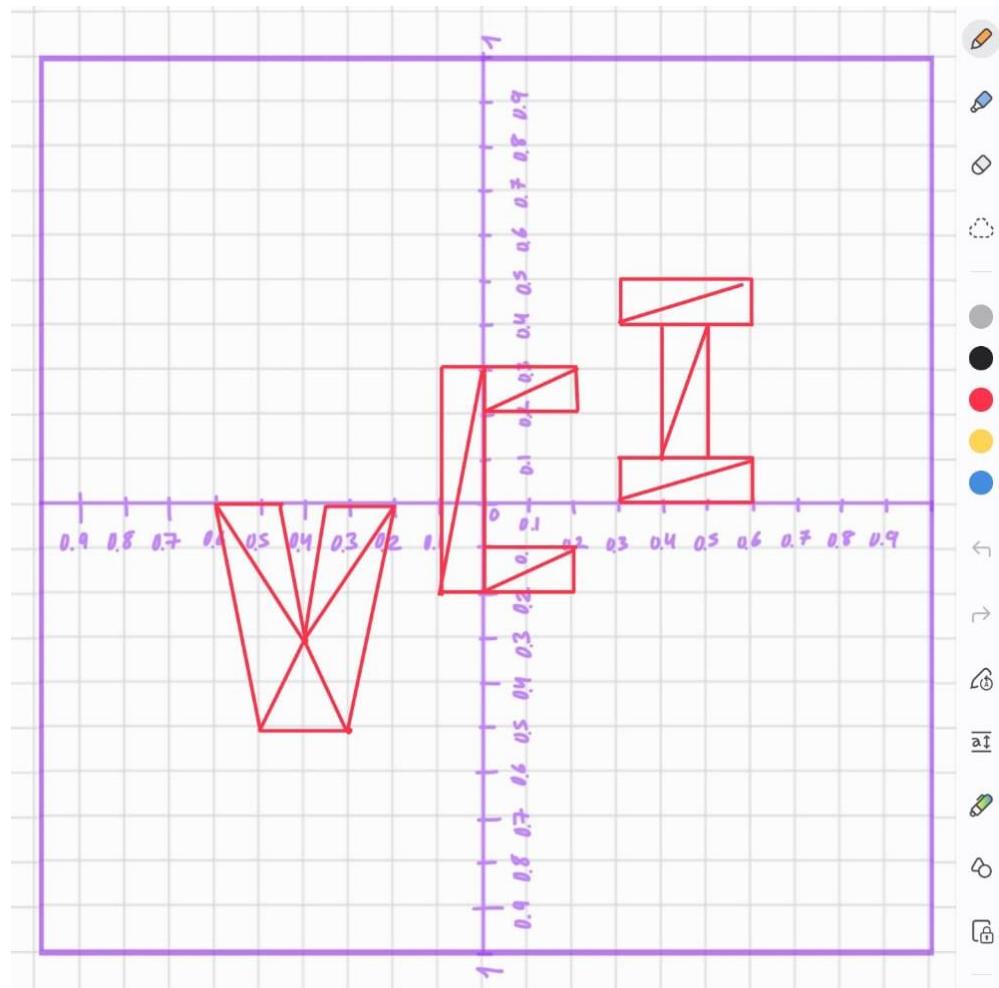
Para la primera parte del ejercicio se incluyeron 2 librerías nuevas para poder hacer uso de la función rand a la que luego se le modificaría la semilla a null para asegurar que realmente generara valores al azar, y time.h para obtener el tiempo actual del sistema y así poder obtener una diferencia de un cierto número

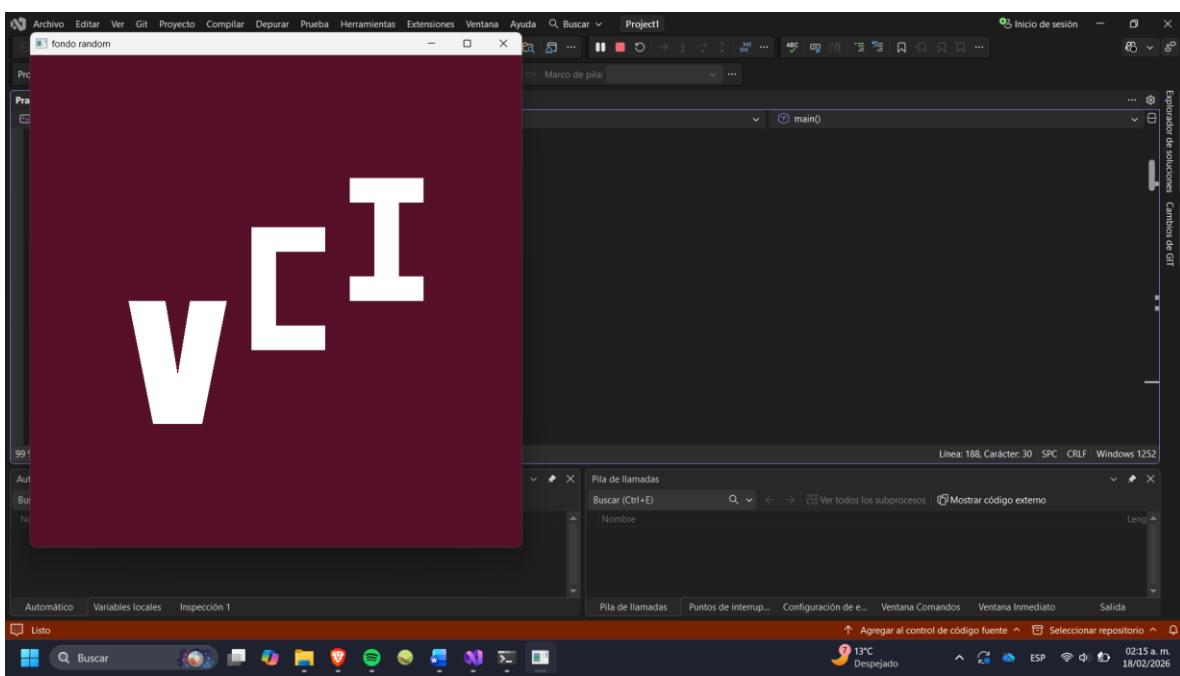
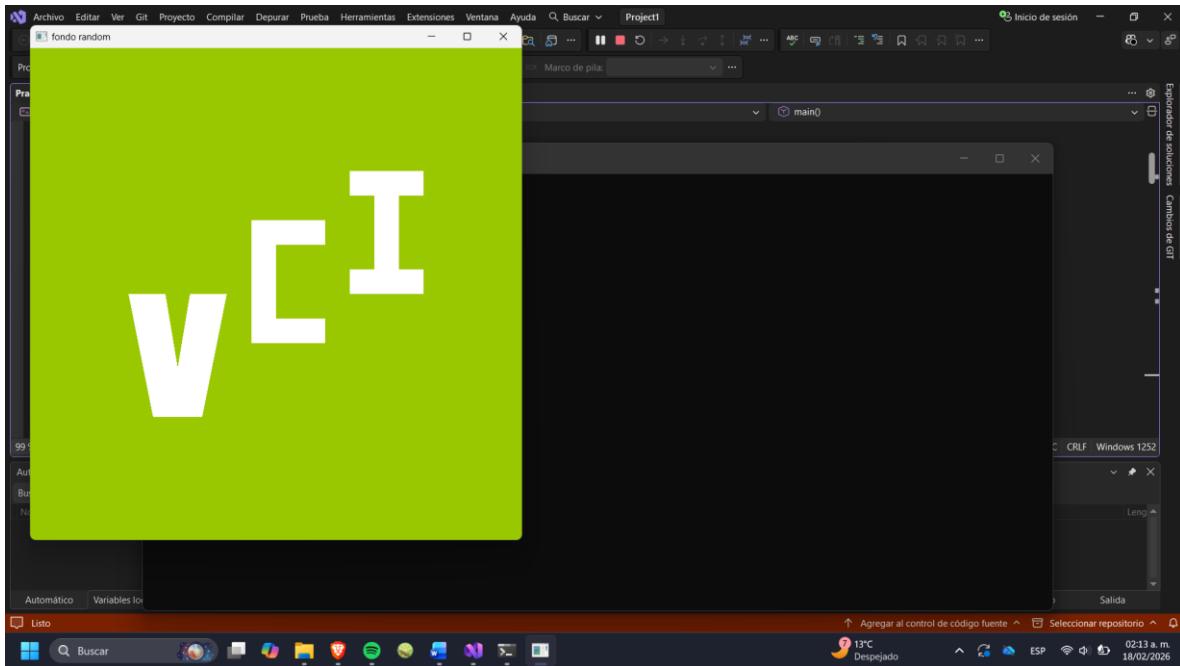
determinado de segundos (en este caso 2) que nos permite crear una pausa en la ejecución del código, en conjunto estos nos ayudan a generar valores random cada vez que se ejecuta el programa y crear los intervalos entre cada color, así como definimos las nuevas variables para almacenar los valores rgb de cada color y el controlador de tiempo.

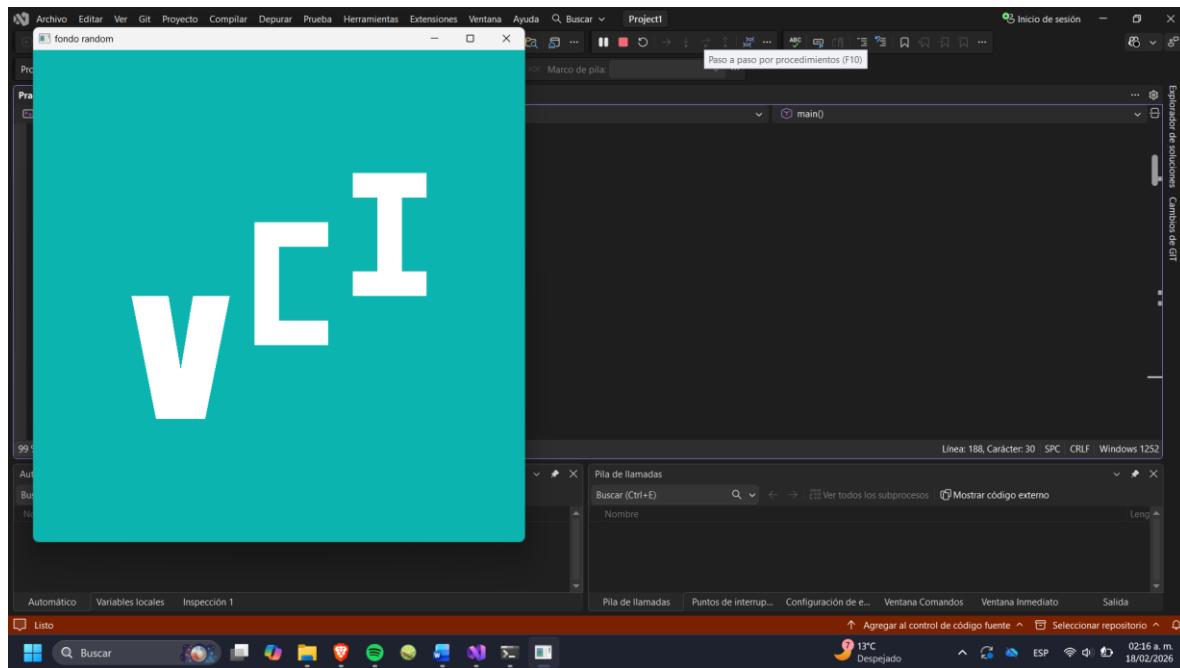
```
34     void CrearTriangulo() {
35         GLfloat vertices[] = {
36             //V
37             -0.6f, 0.0f, 0.0f,
38             -0.45f, 0.0f, 0.0f,
39             -0.4f, -0.3f, 0.0f,
40
41             -0.35f, 0.0f, 0.0f,
42             -0.2f, 0.0f, 0.0f,
43             -0.4f, -0.3f, 0.0f,
44
45             -0.6f, 0.0f, 0.0f,
46             -0.5f, -0.5f, 0.0f,
47             -0.4f, -0.3f, 0.0f,
48
49             -0.2f, 0.0f, 0.0f,
50             -0.3f, -0.5f, 0.0f,
51             -0.4f, -0.3f, 0.0f,
52
53             -0.5f, -0.5f, 0.0f,
54             -0.3f, -0.5f, 0.0f,
55             -0.4f, -0.3f, 0.0f,
56
57             //C
58             -0.1f, 0.3f, 0.0f,
59             0.0f, 0.3f, 0.0f,
60             -0.1f, -0.2f, 0.0f,
61
62             0.0f, -0.2f, 0.0f,
63             0.0f, 0.3f, 0.0f,
64             -0.1f, -0.2f, 0.0f,
65
66             //
```

| | | |
|----|-------------------------|--|
| 34 | void CrearTriangulo() { | |
| 35 | GLfloat vertices[] = { | |
| 36 | //V | |
| 37 | -0.6f, 0.0f, 0.0f, | |
| 38 | -0.45f, 0.0f, 0.0f, | |
| 39 | -0.4f, -0.3f, 0.0f, | |
| 40 | | |
| 41 | -0.35f, 0.0f, 0.0f, | |
| 42 | -0.2f, 0.0f, 0.0f, | |
| 43 | -0.4f, -0.3f, 0.0f, | |
| 44 | | |
| 45 | -0.6f, 0.0f, 0.0f, | |
| 46 | -0.5f, -0.5f, 0.0f, | |
| 47 | -0.4f, -0.3f, 0.0f, | |
| 48 | | |
| 49 | -0.2f, 0.0f, 0.0f, | |
| 50 | -0.3f, -0.5f, 0.0f, | |
| 51 | -0.4f, -0.3f, 0.0f, | |
| 52 | | |
| 53 | -0.5f, -0.5f, 0.0f, | |
| 54 | -0.3f, -0.5f, 0.0f, | |
| 55 | -0.4f, -0.3f, 0.0f, | |
| 56 | | |
| 57 | //C | |
| 58 | -0.1f, 0.3f, 0.0f, | |
| 59 | 0.0f, 0.3f, 0.0f, | |
| 60 | -0.1f, -0.2f, 0.0f, | |
| 61 | | |
| 62 | 0.0f, -0.2f, 0.0f, | |
| 63 | 0.0f, 0.3f, 0.0f, | |
| 64 | -0.1f, -0.2f, 0.0f, | |
| 65 | | |
| 66 | // | |
| 67 | // | |
| 68 | 0.0f, 0.3f, 0.0f, | |
| 69 | 0.2f, 0.3f, 0.0f, | |
| 70 | 0.0f, 0.2f, 0.0f, | |
| 71 | | |
| 72 | 0.0f, 0.2f, 0.0f, | |
| 73 | 0.2f, 0.3f, 0.0f, | |
| 74 | | |
| 75 | // | |
| 76 | 0.0f, -0.1f, 0.0f, | |
| 77 | 0.2f, -0.1f, 0.0f, | |
| 78 | 0.0f, -0.2f, 0.0f, | |
| 79 | | |
| 80 | 0.0f, -0.2f, 0.0f, | |
| 81 | 0.2f, -0.1f, 0.0f, | |
| 82 | | |

Para esta sección primero se hizo la planeación de como se iban a crear las letras y posteriormente se codificaron los triángulos de acuerdo con las coordenadas planeadas







2.- Liste los problemas que tuvo a la hora de hacer estos ejercicios y si los resolvió explicar cómo fue, en caso de error adjuntar captura de pantalla

No hubo errores esta vez, inclusive debido a la práctica ya mas desarrollada del manejo de vértices fue incluso más rápida su codificación, aunque la fase de planeación si fue más tardada por el hecho de haber tenido que idear un forma de utilizar el menor número de triángulos posibles y su escala de tamaño, así como para la randomización del tiempo hubo que buscar en internet cómo alterar a semilla para que fueran números realmente al azar y los colores no se repitieran en cada ejecución.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

Considero que la complejidad de los ejercicios fue razonable porque implicó el manejo de vértices que ya habíamos aprendido en clase y como cambiar el color de fondo de pantalla, lo único que presentó un nivel de dificultad fue la planeación de como construiríamos y posicionariíamos las letras, así como la randomización de los colores para el cambio

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

Personalmente siento el ritmo de trabajo algo acelerado ya que para muchos es la primera vez conviviendo con este tipo de ambientes gráficos y pese haber investigado previamente su funcionamiento no tenemos la práctica para visualizar tan fácilmente lo que se va a construir como otras personas que ya hayan desarrollado proyectos similares antes, mi única sugerencia, se podría complementar la práctica con ejemplos visuales adicionales

c. **Conclusión**

De manera personal la práctica me permitió tener un nivel mucho más alto de comprensión sobre el funcionamiento de OpenGL para la manipulación de gráficos ya que es la primera vez que tengo contacto con este tipo de ambiente gráfico, me ayudó a comprender como manejar elementos de tiempo y modificar propiedades visuales como color y vértices.

4.- Bibliografía en formato APA

- ISO/IEC. (s. f.). rand function — C standard library.
<https://en.cppreference.com/w/c/numeric/random/rand>
- GeeksforGeeks. (s. f.). rand() and srand() in C/C++.
<https://www.geeksforgeeks.org/rand-and-srand-in-ccpp/>
- TutorialsPoint. (s. f.). C library function - rand().
https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm