



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: CARBAJAL REYES IRVIN JAIR

N° de Cuenta: 422042084

GRUPO DE LABORATORIO: 11

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 19 DE FEBRERO DE 2025

CALIFICACIÓN: _____

Actividades

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

Para esta primer actividad, fue necesario el uso de las siguientes librerías:

```
5  #include <chrono>
6  #include <thread>
7  #include <random>
```

chrono y *thread* nos ayudan a implementar los cambios del color de fondo cada dos segundos, mientras que *random* nos ayuda a generar los números flotantes de manera aleatoria.

Para el cambio de color cada dos segundos fue necesario usar una variable para registrar el tiempo de inicio de ejecución *startTime* y otra variable para obtener el tiempo actual *currentTime*, ya que al hacer la diferencia entre estas dos variables, se obtiene el tiempo efectivo transcurrido, que buscamos que sean dos segundos.

```
225 // Guardar tiempo de inicio
226 auto startTime = std::chrono::high_resolution_clock::now();
```

Es importante colocar el *startTime* fuera del ciclo que transcurre mientras la ventana no se cierra, para evitar conflictos con el tiempo efectivo. El *currentTime* debe ir dentro de este *while*.

```
243 // Calcular el tiempo transcurrido
244 auto currentTime = std::chrono::high_resolution_clock::now();
245 std::chrono::duration<double> tiempo = currentTime - startTime;
```

Luego, usamos un condicional para cuando llegue al tiempo establecido, en nuestro caso dos segundos.

```
247 if (tiempo.count() >= 2.0) {
```

Para generar los colores de manera aleatoria, usamos las siguientes funciones:

```
228 std::random_device rd;
229 std::mt19937 gen(rd);
230
231 std::uniform_real_distribution<> dis(0.1, 255.0);
```

Así podremos generar números aleatorios de tipo flotante, determinando un rango entre 0.1 y 255.0.

Luego, creamos variables que representan cada color RGB para almacenar los valores aleatorios generados.

```
233 float r = dis(gen);
234 float g = dis(gen);
235 float v = dis(gen);
```

Inicializamos cada variable con un número aleatorio tipo flotante.

Al iniciar la ventana se le aplicará el color generado aleatoriamente al inicializar las variables

```
246 glClearColor(r / 255.0, g / 255.0, v / 255.0, 1.0f);
```

Una vez que hayan transcurrido los dos segundos, se generan nuevos números flotantes de manera aleatoria y se reinicia el timer.

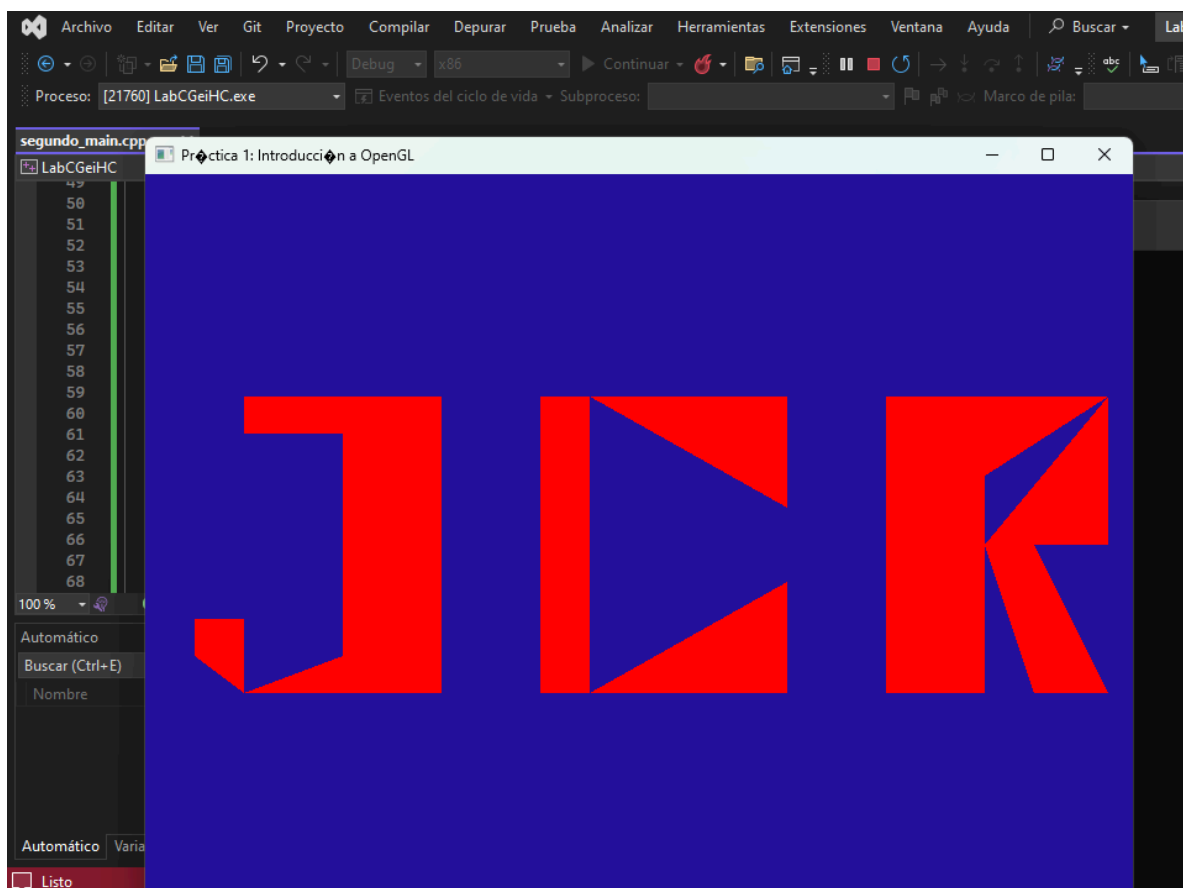
```
247 if (tiempo.count() >= 2.0) {
248     // Cambiar el color cada 2 segundos
249     r = dis(gen);
250     g = dis(gen);
251     v = dis(gen);
252     glClearColor(r / 255.0, g / 255.0, v / 255.0, 1.0f);
253     startTime = currentTime;
254 }
```

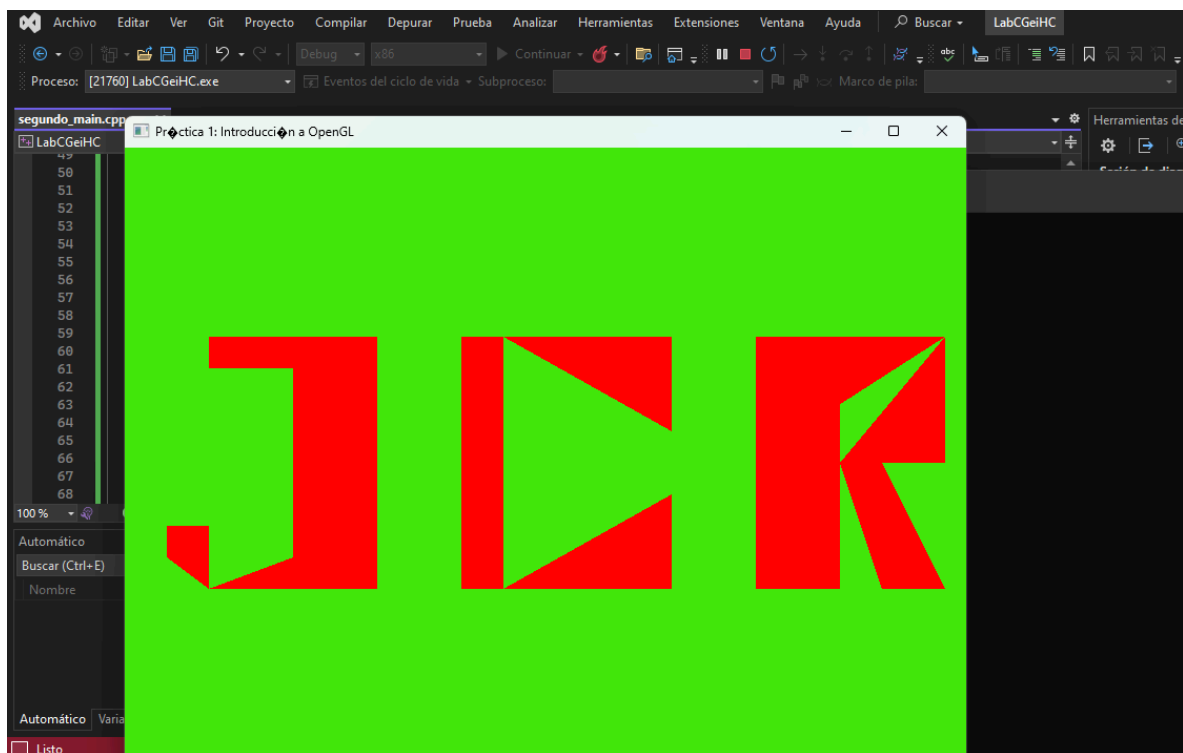
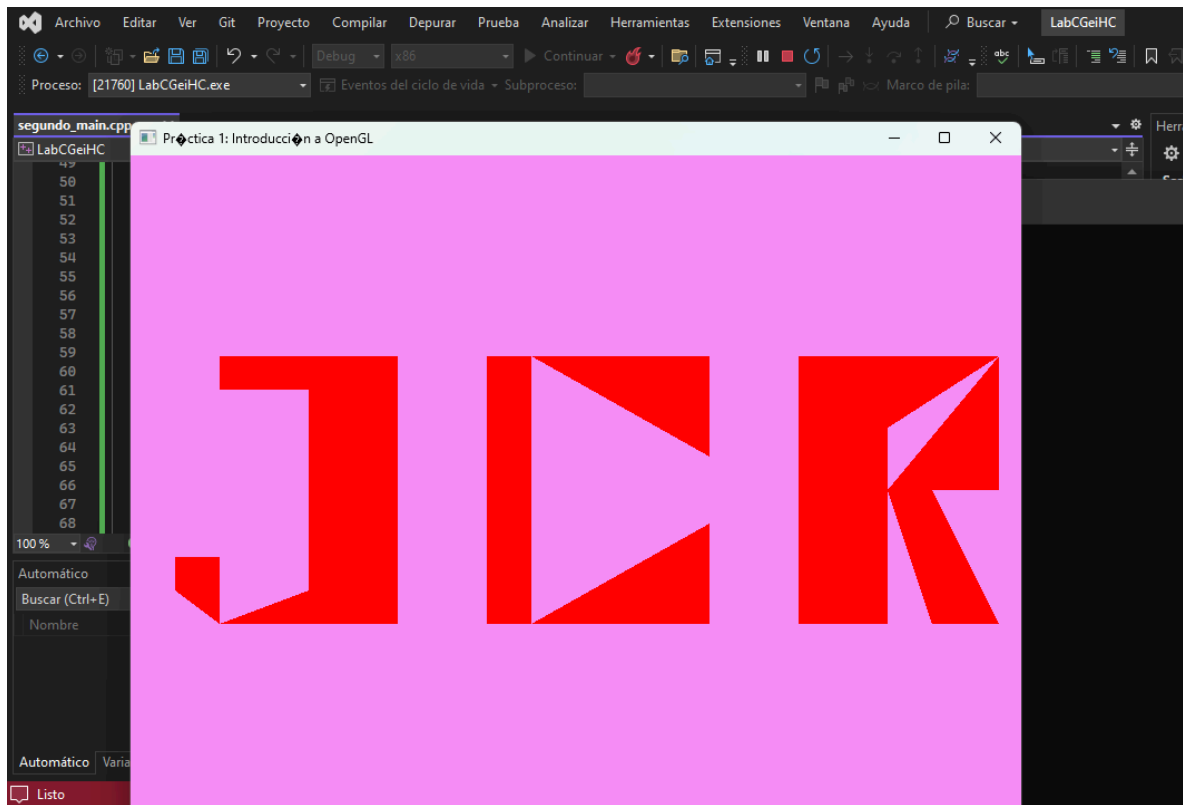
De este manera, tendremos colores dentro del rango RGB diferentes cada dos segundos.

2. Tres letras iniciales de sus nombres creadas a partir de triángulos, todas las letras son del mismo color.

Para la realización de este ejercicio primero se hizo un bosquejo del diseño de las iniciales por medio de triángulos.

Ejecución del programa:





Errores presentados

Los errores que se presentaron fueron solucionados en el ejercicio de clase, ya que se colocó el inicio del timer dentro del ciclo de la ventana abierta, por lo que se presentaron problemas en los tiempos para cambiar el color de fondo, sin embargo una vez iniciado el timer fuera del *while* se pudieron medir los dos segundos sin problema, agregando a su vez valores aleatorios a las variables correspondientes a cada color para que estas cambien cada dos segundos.

Conclusión

La práctica se desarrolló de manera satisfactoria, ya que la explicación del profesor durante la hora de clase nos permitió entender los conceptos clave para la creación de triángulos a partir de vértices y como modificar tanto el color de los mismos como el color de fondo, solo bastó con recordar algunos conceptos básicos del lenguaje C++ para realizar lo solicitado, como fue el uso de librerías de *chrono*, *threads* y *random* para implementar timers y generar números aleatorios, lo que nos permitió interactuar con los distintos elementos básicos que se vieron al inicio de la práctica.

Bibliografía

- OpenGL básico. (n.d.-b).
<http://www.opengl-tutorial.org/es/beginners-tutorials/>