



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: Jiménez Elizalde Josué

N° de Cuenta: 320334489

GRUPO DE LABORATORIO: 13

GRUPO DE TEORÍA: 06

SEMESTRE 2026-1

FECHA DE ENTREGA LÍMITE: 28/08/2025

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.

Como primer ejercicio se tiene que cambiar de forma aleatoria el color de fondo de la página cada tres segundos, similar a lo que se hizo en el ejercicio en clase, lo único que se cambio fue la forma de aparición de los colores.

Código y descripción de desarrollo:

```
//Utilizamos time y la inicialización del random
srand(time(NULL));
time_t last = time(NULL);
```

De inicios se tiene la inicialización del random, pero este tiene la característica que se ingresa como argumento “time(NULL)” el cual garantiza que sea aleatorio el numero que salga, ya que al iniciar el programa se va a ingresar otro número distinto. Por último, se tiene el tiempo ultimo que se obtiene al iniciar el programa.

```
time_t now = time(NULL);
if (now - last >= 2) { // han pasado 2 segundos
    float newR, newG, newB;
    //Generar un color suficientemente distinto al actual
    do {
        newR = rand() / (float)RAND_MAX; //0.0 - 1.0
        newG = rand() / (float)RAND_MAX;
        newB = rand() / (float)RAND_MAX;
        //suma de diferencias en canales para garantizar cambio notable
    } while (fabs(newR - rojo) + fabs(newG - verde) + fabs(newB - azul) < 0.35f);

    rojo = newR;
    verde = newG;
    azul = newB;

    last = now;
}
```

Con este sencillo bloque de código, se tiene la lógica del cambio de color, primero se obtiene el tiempo actual a la hora que salga la pantalla, por lo que la validación que se tiene dice el tiempo que se tiene que esperar para que vuelva a entrar, una vez que se respete esos segundos, entra al bloque del condicional. Se declaran

los nuevos colores que se ingresan a un bloque do-while para que sea lo suficientemente diferentes al color original.

Por último, se asigna esos valores a las variables originales y el último tiempo se convierte en el actual, para poder esperar otros segundos.

Ejecución del programa:



No se muestra el tiempo de espera, pero en las imágenes se tiene los colores distintos, el color base es el color verde y los siguientes se ven modificados por todo el bloque anterior.

2.- 3 letras iniciales de sus nombres creadas a partir de triángulos, todas las letras son del mismo color.

Código y descripción de desarrollo:

```
//Barra vertical (derecha de la J) - llega hasta y = 0.05
-0.70f, 0.90f, 0.0f,
-0.65f, 0.90f, 0.0f,
-0.70f, 0.10f, 0.0f,

-0.65f, 0.90f, 0.0f,
-0.65f, 0.10f, 0.0f,
-0.70f, 0.10f, 0.0f,

//Base/gancho de la J (horizontal, conecta con la vertical)
-0.95f, 0.15f, 0.0f,
-0.65f, 0.15f, 0.0f,
-0.95f, 0.10f, 0.0f,

-0.65f, 0.15f, 0.0f,
-0.65f, 0.10f, 0.0f,
-0.95f, 0.10f, 0.0f,

-0.90f, 0.20f, 0.0f,
-0.90f, 0.10f, 0.0f,
-0.95f, 0.10f, 0.0f,

-0.95f, 0.20f, 0.0f,
-0.90f, 0.20f, 0.0f,
-0.95f, 0.10f, 0.0f,

//----- 0 -----
//Parte superior
-0.60f, 0.9f, 0.0f,
-0.30f, 0.9f, 0.0f,
-0.60f, 0.85f, 0.0f,

-0.30f, 0.9f, 0.0f,
-0.30f, 0.85f, 0.0f,
-0.60f, 0.85f, 0.0f,

//Parte inferior
-0.60f, 0.15f, 0.0f,
-0.30f, 0.15f, 0.0f,
-0.60f, 0.10f, 0.0f,

-0.30f, 0.15f, 0.0f,
-0.30f, 0.10f, 0.0f,
-0.60f, 0.10f, 0.0f,

//Lateral izquierdo
-0.60f, 0.9f, 0.0f,
-0.55f, 0.9f, 0.0f,
-0.60f, 0.10f, 0.0f,

-0.55f, 0.9f, 0.0f,
-0.55f, 0.10f, 0.0f,
```

```

-0.60f, 0.10f, 0.0f,

//Lateral derecho
-0.35f, 0.9f, 0.0f,
-0.30f, 0.9f, 0.0f,
-0.35f, 0.10f, 0.0f,

-0.30f, 0.9f, 0.0f,
-0.30f, 0.10f, 0.0f,
-0.35f, 0.10f, 0.0f,

//----- E -----
//Palito vertical
-0.20f, 0.9f, 0.0f,
-0.15f, 0.9f, 0.0f,
-0.20f, 0.1f, 0.0f,

-0.15f, 0.9f, 0.0f,
-0.15f, 0.1f, 0.0f,
-0.20f, 0.1f, 0.0f,

//Línea superior
-0.20f, 0.9f, 0.0f,
0.05f, 0.9f, 0.0f,
-0.20f, 0.85f, 0.0f,

0.05f, 0.9f, 0.0f,
0.05f, 0.85f, 0.0f,
-0.20f, 0.85f, 0.0f,

//Línea del medio
-0.20f, 0.55f, 0.0f,
0.00f, 0.55f, 0.0f,
-0.20f, 0.50f, 0.0f,

0.00f, 0.55f, 0.0f,
0.00f, 0.50f, 0.0f,
-0.20f, 0.50f, 0.0f,

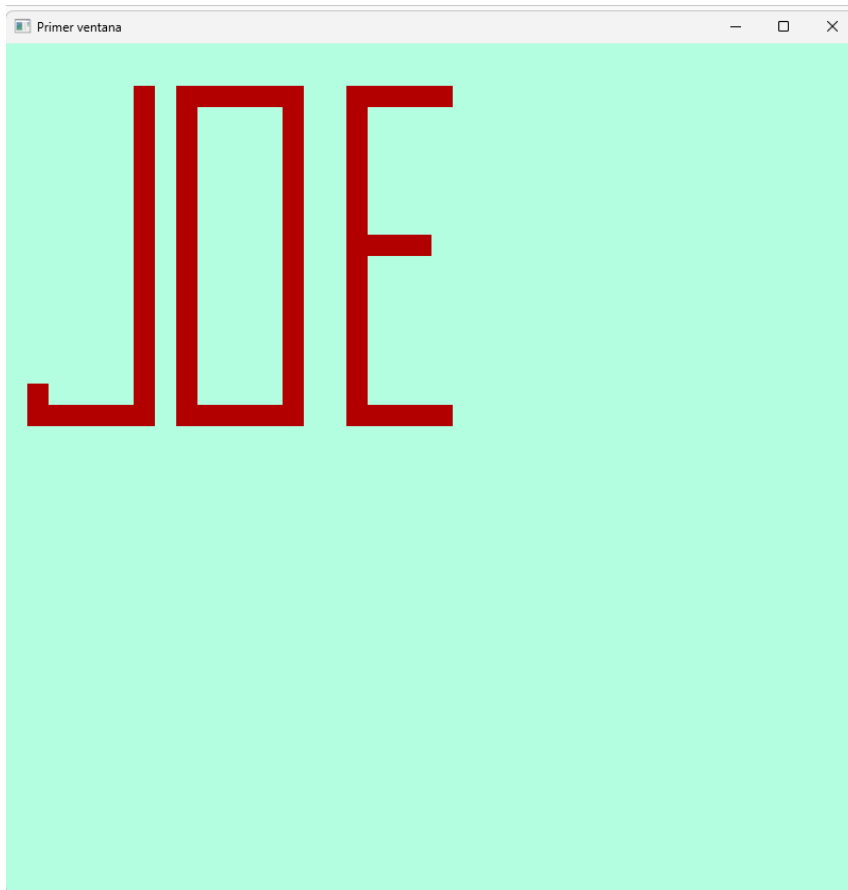
//Línea inferior
-0.20f, 0.15f, 0.0f,
0.05f, 0.15f, 0.0f,
-0.20f, 0.10f, 0.0f,

0.05f, 0.15f, 0.0f,
0.05f, 0.10f, 0.0f,
-0.20f, 0.10f, 0.0f,

```

Para la generación de las letras con triángulos, solamente se añadieron estas partes en el código las cuales representan cada punto de los triángulos, separados y documentados dependiendo de cada letra para una mayor claridad.

Ejecución del programa:



Como se puede observar están las tres letras, la primera es una J, la segunda es una O y la última es la letra E las cuales son algunas de mis iniciales.

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 se tuvo problemas a la hora de resolver los ejercicios, solo las complicaciones de entender el posicionamiento de cada uno de los puntos, con lo de los colores se investigó la forma que se pueda hacer de forma aleatoria y saber como implementarlo.

3.- Conclusión:

En clase se analizo cada parte del código OpenGL paso por paso para su comprensión, aunque por la complejidad de OpenGL algunos temas quedaron superficialmente y no se obtuvo más detalles de ellos, pero la generación de distintas figuras quedo claro en la elaboración de este reporte y ejercicios.

4.- Bibliografía en formato APA

IBM i. (2025, 8 abril). Recuperado 24 de agosto de 2025, de <https://www.ibm.com/docs/es/i/7.6.0?topic=functions-srand-set-seed-rand-function>

Rancel, M. R. (s. f.). *Generar números o secuencias aleatorios en C. Intervalos. srand y rand. Time null. RAND_MAX. (CU00525F).* aprenderaprogramar.com.
https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=917:generar-numeros-o-secuencias-aleatorios-en-c-intervalos-srand-y-rand-time-null-randmax-cu00525f&catid=82&Itemid=210