UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO COMPUTADORA

PREVIO Nº 1

NOMBRE COMPLETO: Gómez Enríquez Agustín

Nº de Cuenta: 317031405

GRUPO DE LABORATORIO: 3

GRUPO DE TEORÍA: 5

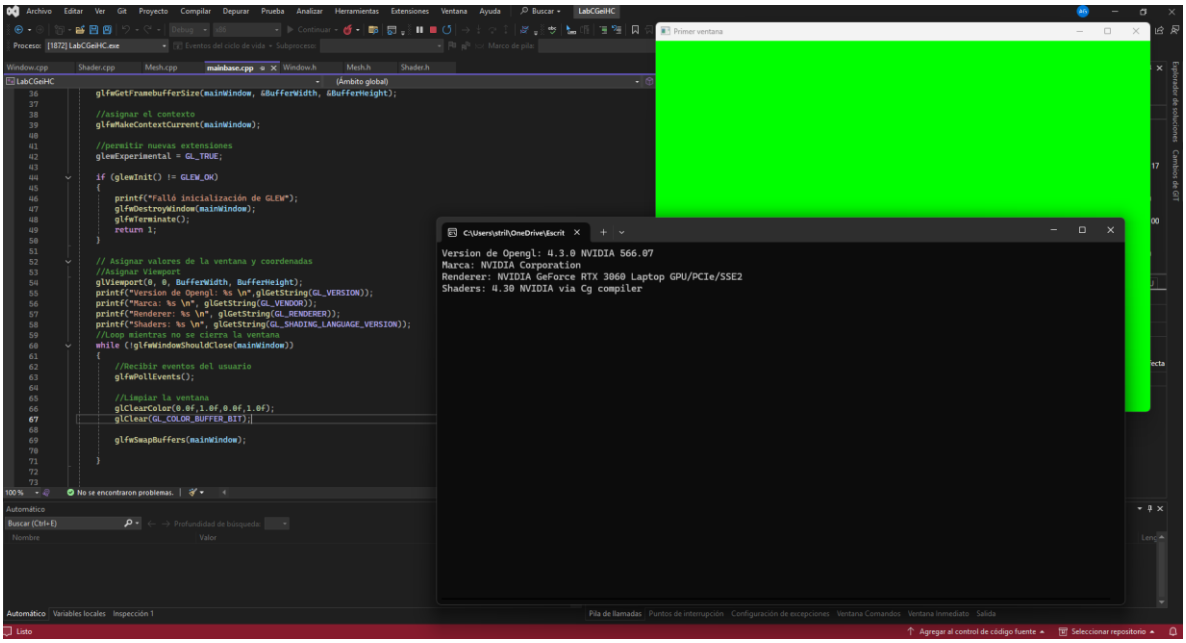
SEMESTRE 2026 - 1

FECHA DE ENTREGA LÍMITE: 24 de agosto del 2025

CALIFICACIÓN: _____

Respuestas previo 1:

1.-. Captura de pantalla como la del manual de configuración en la cual se muestra la ventana de fondo verde y la información de la consola con los datos de Hardware de su equipo de cómputo (no es necesario que se imprima para la entrega del previo a mano)



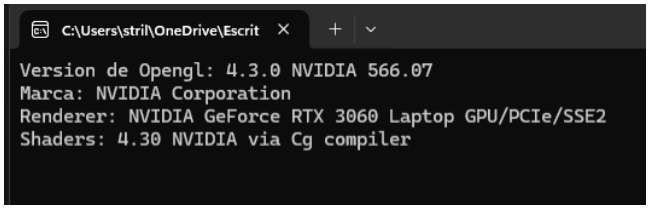
Texto:

Version de Opengl: 4.3.0 NVIDIA 566.07

Marca: NVIDIA Corporation

Renderer: NVIDIA GeForce RTX 3060 Laptop GPU/PCIe/SSE2

Shaders: 4.30 NVIDIA via Cg compiler



2. ¿Que es un VAO?

R. Bajo el contexto de Open GL, definimos a VAO (Vertex Array Object), que es un objeto encargado de guardar la configuración de los datos contenidos en los vertices desde los buffers. Almacena datos como: Posicion, Normales, coordenadas, color, (Etc)

3. Que es un VBO?

R. Vertex Buffer object es un buffer en la memoria de la GPU donde se guardan los datos de los vertices.

4. Que parametros recibe el comando glVertex Attrib Pointer?

R: Interpreta los datos de un atributo de vertice dentro de un bufer de vertices y recibe los siguientes Parametros: index, size, type, normalized, stride y Pointer.

- Size (GLint): Número de componentes por atributo de vertice (1-4)
- Type (GLenum): Tipo de dato de cada componente del atributo.
- Normalized (GL Boolean): Valor Booleano que indica si los datos fijos deben ser normalizados o no.
- Stride (GLSizei): Desplazamiento en bytes entre atributos de vertice consecutivo.
- Pointer (const void): Puntero a desplazamiento del primer componente del primer atributo vertice

5. ¿Que información maneja Vertex Shader?

R: Maneja y transforma los datos de los vertices de una malla 3D para calcular su posición final en pantalla, definir sus atributos (como color y texturas) y prepararlos para ser etapas posteriores como el pipeline grafico. Recibe información de cada vertice como coordenadas (posición), texturas (UV), colores y vectores normales.

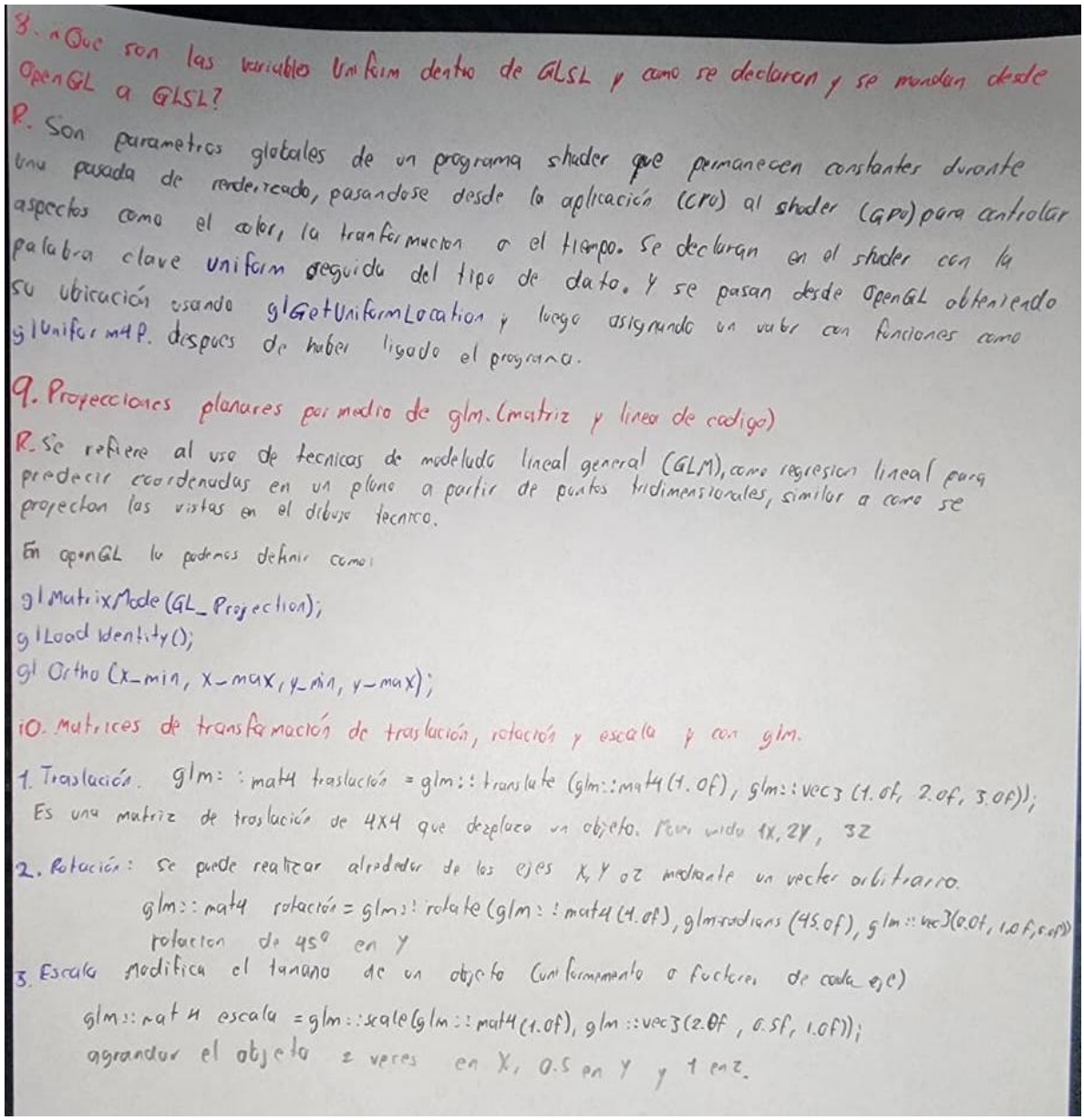
6. ¿Que información maneja Fragment Shader?

R: Maneja información sobre la geometria y propiedades de cada pixel a traves de datos interpolados desde la etapa de vertices, como su posición en pantalla y las texturas asignadas. Su función principal es calcular el color y el valor de profundidad de ese pixel, basandose en la iluminación, la configuración del material y las texturas.

7. ¿Que parametros recibe el comando glDrawArrays?

R: Recibe tres parametros:

- Mode (GLenum): Determina el tipo de primitiva geometrica que va a construir como lineas, puntos y triangulos.
- first (GLint): Es el indice del primer elemento en los arrays de vertices para construir primitivas
- Count (GLsizei): Especifica el numero de elementos de los arrays que se utilizaran para crear primitivas.



Referencias

- *Computer graphics stack exchange*. (n.d.). Computer Graphics Stack Exchange. Retrieved Agosto 25, 2025, from https://computergraphics.stackexchange.com/questions/10332/understanding-vao-and-vbo
- de la Computación e Inteligencia Artificial, C. (n.d.). *Dibujando en el espacio*. Uhu.Es. Retrieved Agosto 25, 2025, from https://www.uhu.es/francisco.moreno/gii_rv/docs/Tema_5.pdf
- *Función glDrawArrays*. (n.d.). Microsoft.com. Retrieved Agosto 25, 2025, from https://learn.microsoft.com/es-es/windows/win32/opengl/gldrawarrays

- *Shader basics - vertex shader*. (n.d.). Shader-tutorial.dev. Retrieved Agosto 25, 2025, from https://shader-tutorial.dev/basics/vertex-shader/
- Unity Technologies. (n.d.). *Ejemplos del Vertex y Fragment Shader*. Unity3d.com. Retrieved Agosto 25, 2025, from https://docs.unity3d.com/es/2018.4/Manual/SL-VertexFragmentShaderExamples.html
- *Vertex Buffer Objects*. (n.d.). Antongerdelan.net. Retrieved Agosto 25, 2025, from https://antongerdelan.net/opengl/vertexbuffers.html
- (N.d.). Umich.Mx. Retrieved Agosto 25, 2025, from https://lc.fie.umich.mx/~rochoa/Materias/LABORATORIOS/GRAFICACION/GRAF_02.pdf