



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



Manual Técnico

○ 317344376

REQUISITOS

Para poder ejecutar el programa se requiere contar, en el equipo de cómputo, con las siguientes características:

- Hardware
 - CPU de 2 Hilos como mínimo, frecuencia mínima de 2.9[GHz]
 - 4 – 8 GB RAM como mínimo (la velocidad de esta es indiferente)
 - Espacio en disco mínimo de 175 [MB]
 - Tarjeta Gráfica dedicada indiferente (también funciona con gráficos integrados)
- Software
 - Se requiere de un sistema operativo Windows 10 u 11 (versión indistinta)
- Archivos adicionales
 - Los que contiene el comprimido
 - assimp-vc140-mt.dll
 - glew32.dll
 - glfw3.dll
 - ProyectoFinal.pdb
 - Carpetas
 - Models
 - Shaders
 - Textures

Para llevar a cabo este proyecto se utiliza lo siguiente:

- Visual Studio 2017 en adelante
- Librería Gráfica: OpenGL versión 3.1 en adelante
- Librería de Interfaz: GLFW

Software de modelado utilizado:

- Maya
- 3Ds
- Blender

Páginas WEB importantes:

- Para descarga de modelos 3D
 - <https://open3dmodel.com/es/3d-models/obj>
 - <https://free3d.com/es/>
 - <https://www.turbosquid.com/es/Search/3D-Models/free/obj>
- Para creación de repositorios
 - <https://github.com/>

Modelos cargados

- kunai
- Bijudama
- Shuriken
- Bijus
- Bandana
- Katana Sasuke Uchiha
- Trébol carmesí de 4 hojas
- Kubikiribōchō
- Naruto
- Kama
- Personaje Tobi
- Restaurante

Por otro lado se contarán con animaciones no ligadas a esta serie televisiva tales como:

- La estrella de la muerte (Star Wars)
- La snitch dorada (Harry Potter)
- Garganta (Interestelar)

MODIFICACIÓN

Si se requiere hacer ajustes referentes a la ejecución del programa, ya sea agregando modelos, animaciones, luces u otras cosas, se deberá trabajar desde nivel código. Para ello deberá clonar el repositorio en su Visual Studio siguiendo los siguientes pasos:

1. Vincular Visual Studio y GitHub

- a. Deberá abrir primero su aplicación Visual Studio e iniciar con “Crear un Proyecto” (Figura 1.1)

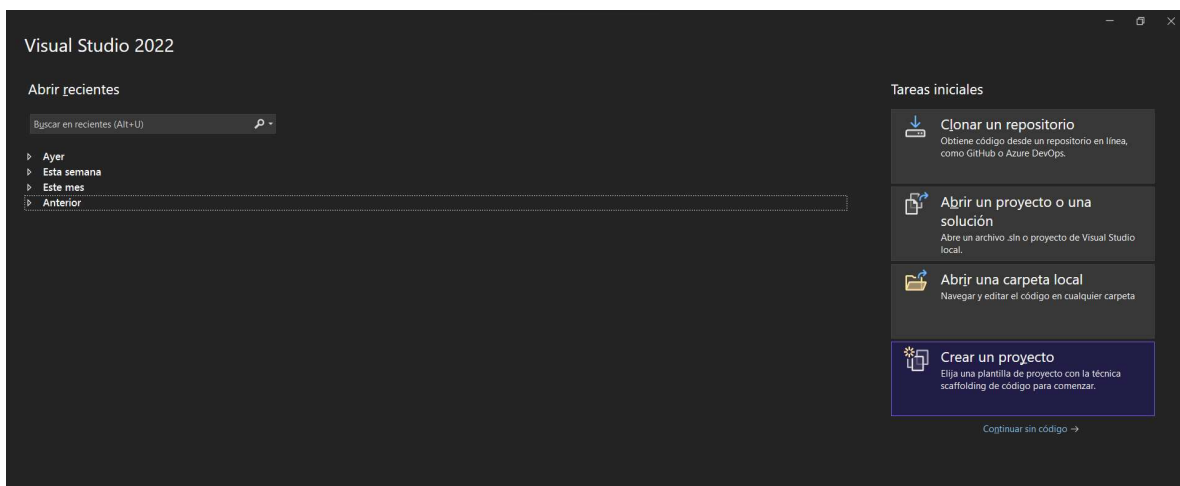


Figura 1.1 Crear un Proyecto

- b. Posteriormente seleccionará “Proyecto Vacío”, para ello en la sección de “Lenguajes” deberá estar seleccionado C++

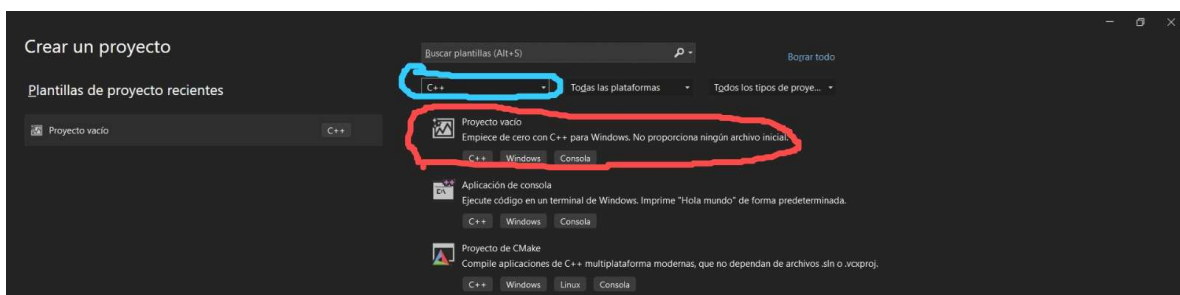


Figura 1.2 Proyecto Vacío

- c. Seleccione un nombre para su proyecto y una carpeta de destino, posteriormente de click en el botón “crear”

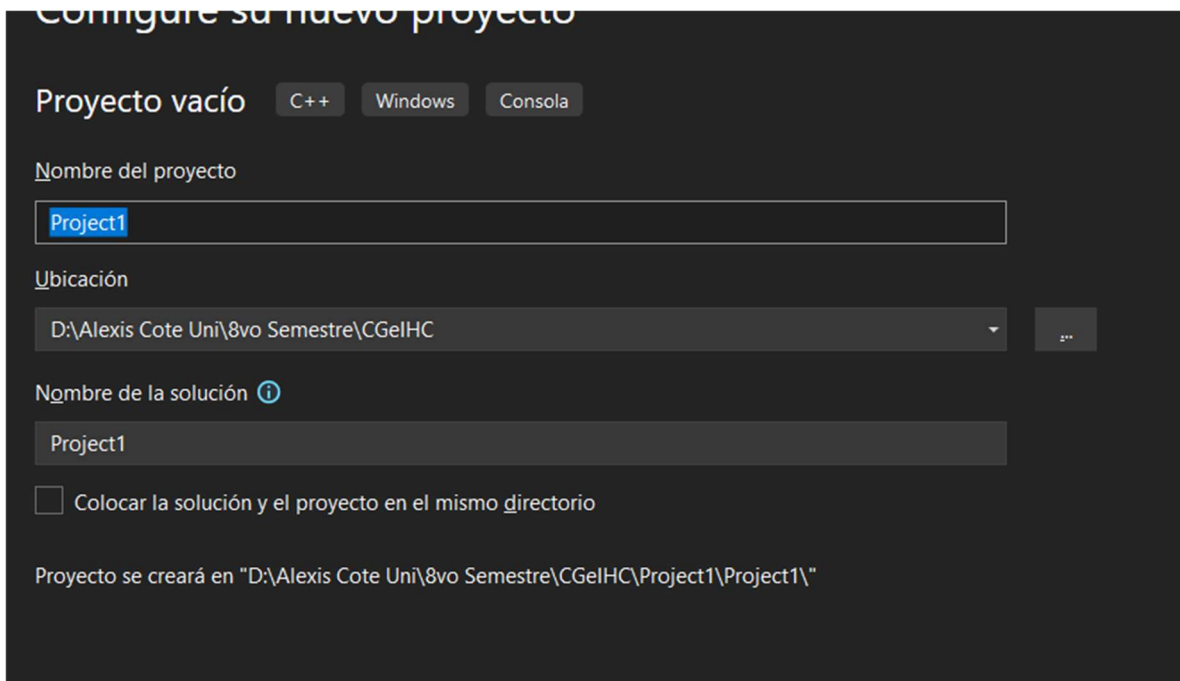


Figura 1.3 Nombre y ubicación del proyecto

- d. Busque la opción dónde diga “Agregar al control de código fuente”, selecciónela y de click en Git

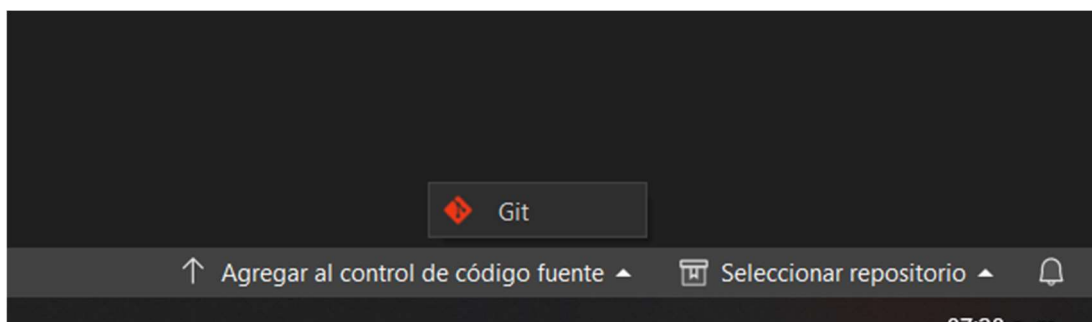


Figura 1.4 Git

- e. Le aparecerá una ventana como la de la Figura 1.5, en esa ventana usted llenará los datos referentes a GitHub (si es primera vez este lo mandará a la web para que confirme el inicio de sesión y la vinculación

de la cuenta) seleccionando cuenta, propietario, nombre del repositorio y descripción de este último.

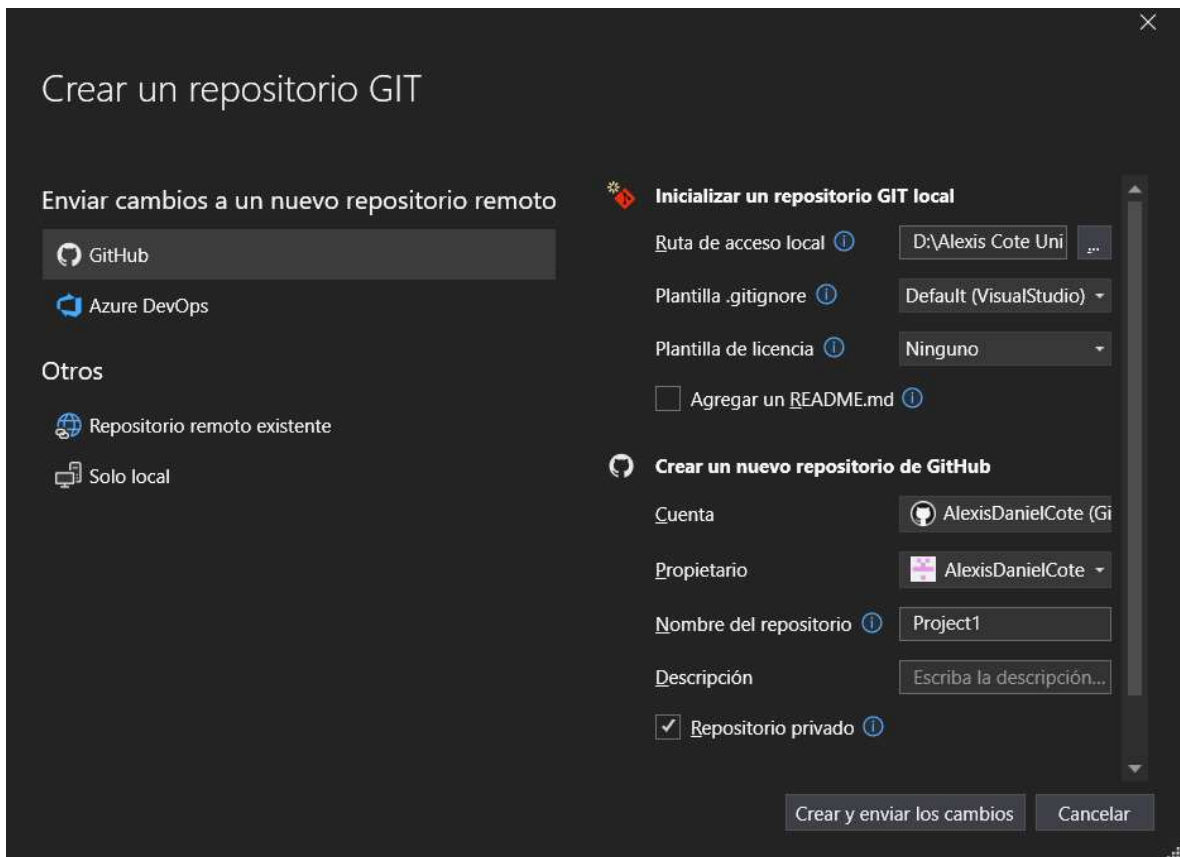


Figura 1.5 Creación repositorio.

- f. Por último, seleccione "Crear y enviar los cambios"
2. Clonar repositorio
 - a. Abra el Visual Studio y seleccione dónde diga "Clonar Repositorio"

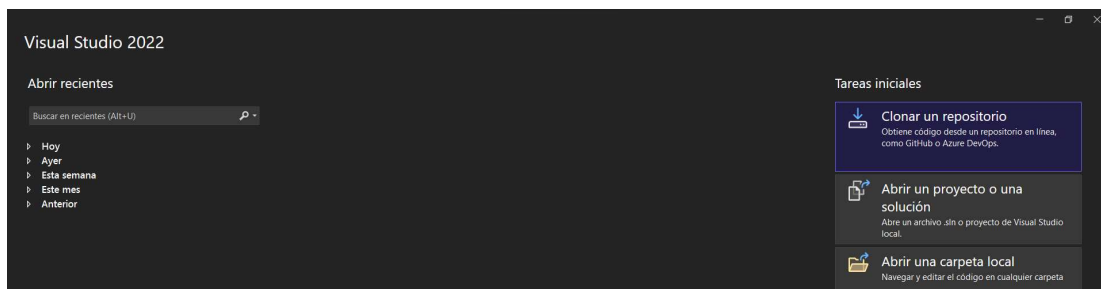


Figura 1.6 Clonar un repositorio

- b. Indique la liga del repositorio a clonar, en este caso será: <https://github.com/AlexisDanielCote/ProyectoFinal.git> y también seleccione/indique la carpeta dónde se guardará el repositorio en su máquina virtual. Finalmente de click en el botón “clonar”.

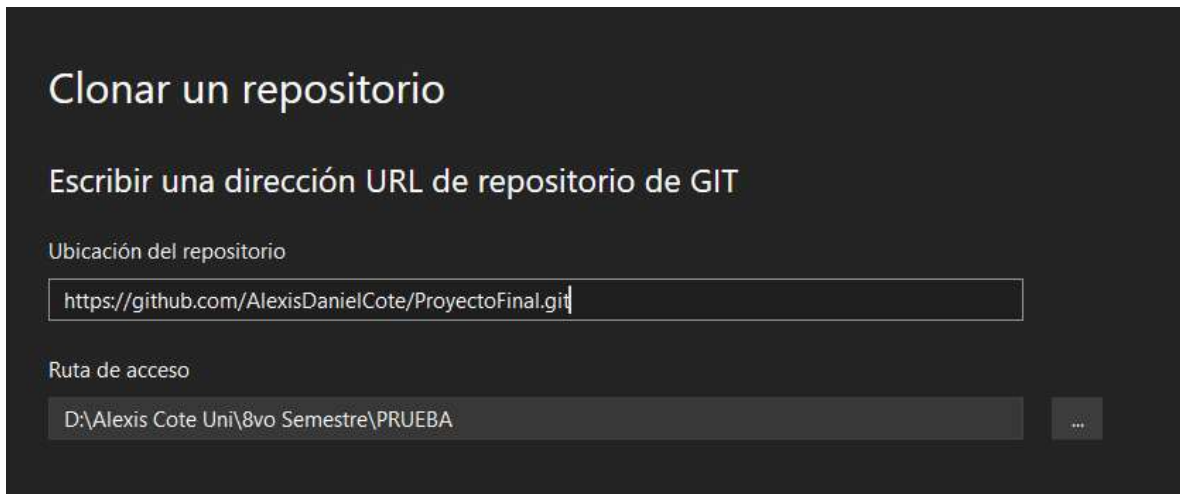


Figura 1.7 Liga y carpeta para clonar

- c. Espere en lo que termina de clonarse el repositorio (puede tardar algunos minutos según la velocidad de su internet).

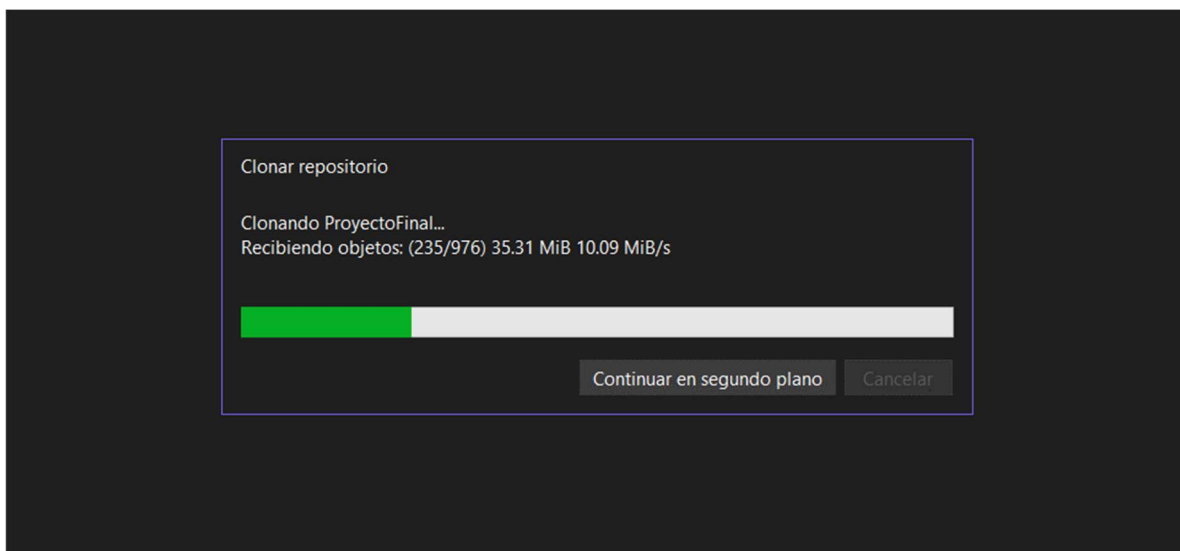


Figura 1.8 Descarga del repositorio

- d. Al finalizar abra el apartado de “Explorador de soluciones” y, desplegando las subsecciones de “Archivos de encabezado” y “Archivos de origen” deberá visualizar los documentos.

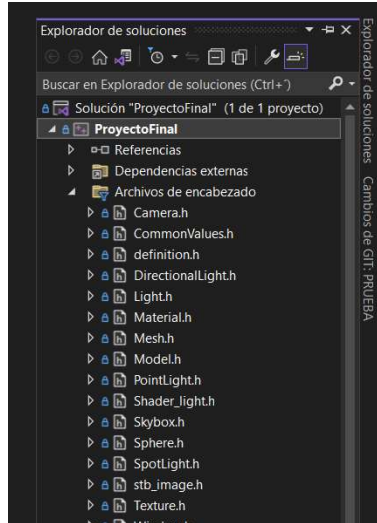


Figura 1.9 Archivos

- e. Para poder depurar/ejecutar el programa debemos realizar modificaciones en el compilador, para ello en la parte superior debemos ubicar la sección de “Depurador local de Windows”, del lado izquierdo tendrá dos opciones “Debug” y “x64”, desplegamos el de “x64” y lo cambiamos por la opción “x86”.

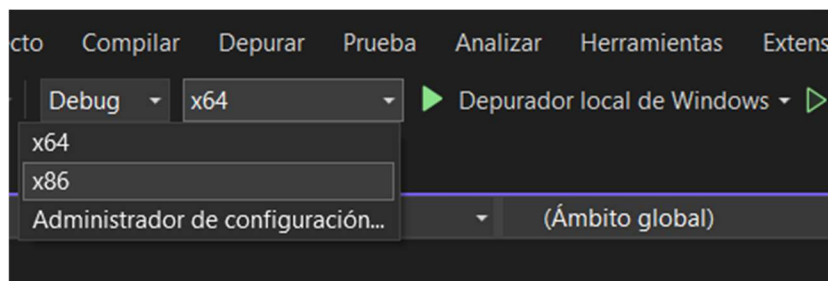


Figura 1.10 Modificación del compilador.

- f. Tras cada modificación en el código, texturas, modelos o documentos deberá realizar un comit y push en el repositorio, para ello en la

sección de “Cambios de GIT” se desplegará en el apartado de cambios todas las modificaciones realizadas en los archivos, ya sea dentro del mismo Visual Studio o mediante el gestor de documentos.

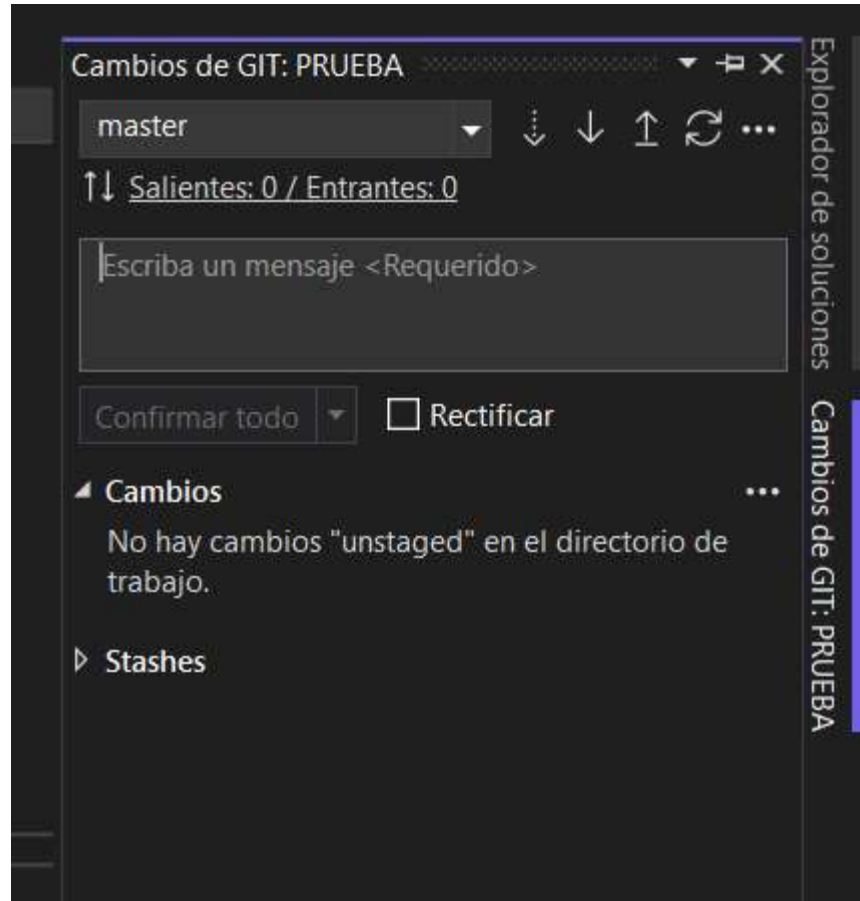


Figura 1.11 Cambios de GIT

- g. En la Figura 1.11 dónde dice “Escriba un mensaje <Requerido>” deberá de proporcionar una breve descripción del cambio realizado en el proyecto, posteriormente deberá dar click en el botón “Confirmar todo” para finalizar con la selección del botón “Push”.

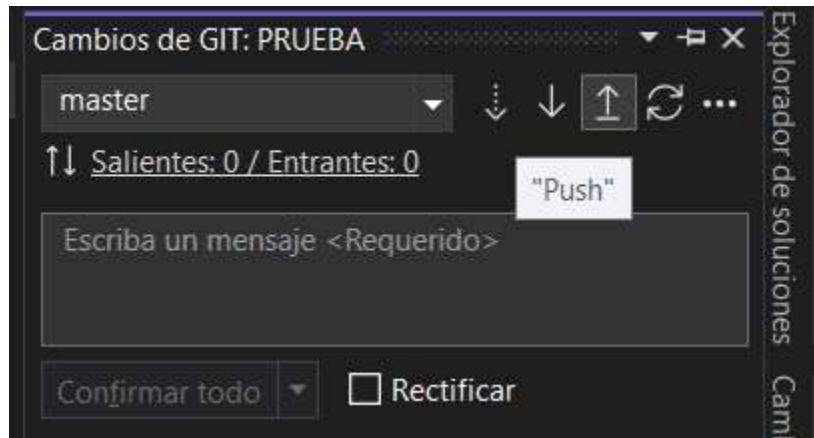


Figura 1.12 Push

- h. Si realiza cambios desde la web en el repositorio o si alguien más realizó cambios desde Visual Studio, deberá actualizar el proyecto antes de realizar modificaciones, comits y push. Para ello deberá seleccionar el botón "Sincronizar".

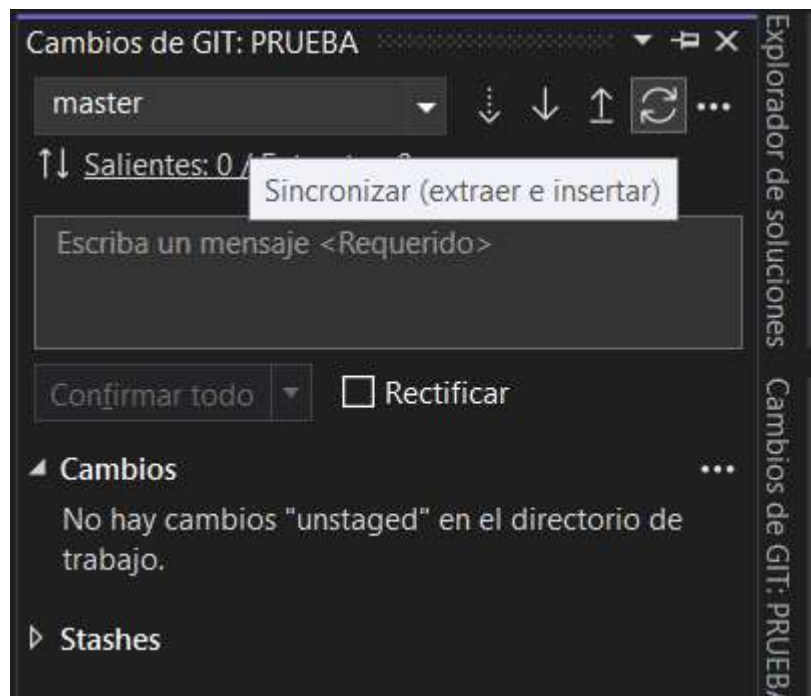


Figura 1.13 Sincronizar

EJECUCIÓN

Para poder ejecutar el programa usted deberá descargar todos los archivos que se encuentren dentro de la carpeta release del repositorio, o si lo prefiere descargar todo el repositorio en un ZIP (Figura 2.2), extraer la información y trabajar con el ejecutable que se encuentra en la carpeta mencionada previamente.

<https://github.com/AlexisDanielCote/ProyectoFinal.git>

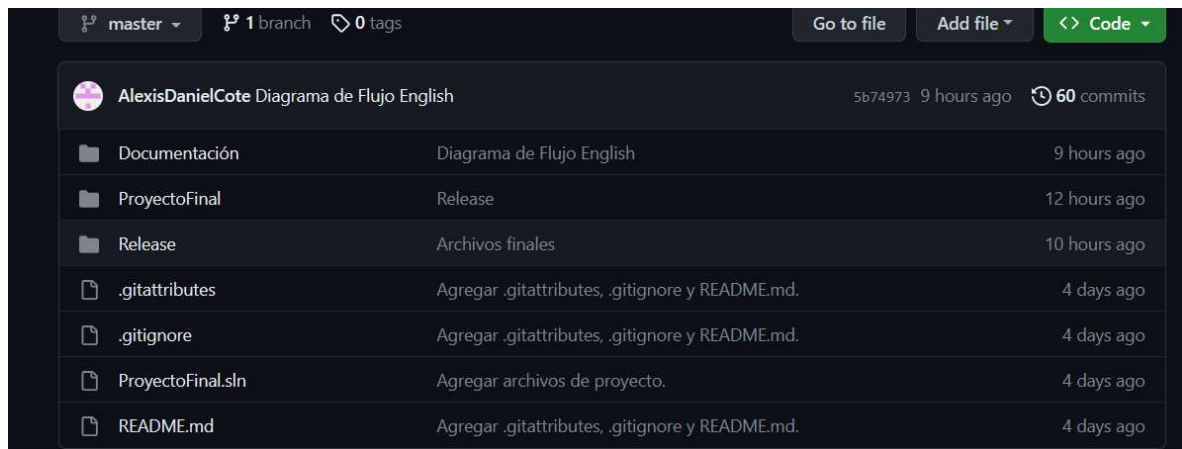


Figura 2.1 Repositorio, carpeta release

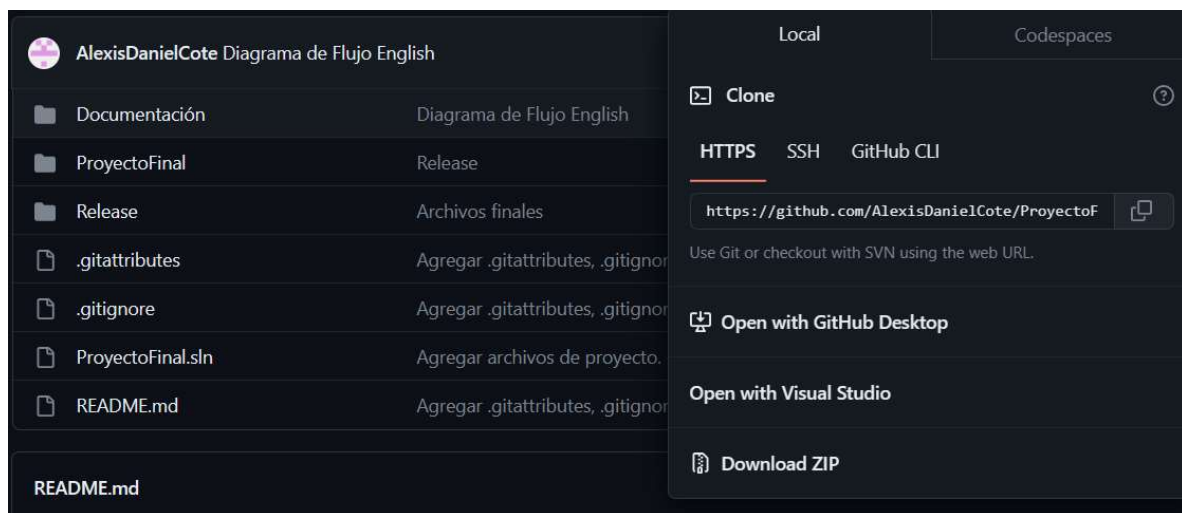


Figura 2.2 Descarga del repositorio

Si descarga solamente la carpeta, ésta deberá de verse de la siguiente manera en su gestor de documentos.

Nombre	Fecha de modificación	Tipo	Tamaño
Models	05/06/2023 08:06 a. m.	Carpeta de archivos	
shaders	05/06/2023 08:06 a. m.	Carpeta de archivos	
Textures	05/06/2023 08:06 a. m.	Carpeta de archivos	
assimp-vc140-mt.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	4,025 KB
glew32.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	381 KB
glfw3.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	70 KB
ProyectoFinal	04/06/2023 05:44 p. m.	Aplicación	232 KB
ProyectoFinal.pdb	04/06/2023 05:44 p. m.	Program Debug D...	2,012 KB

Figura 2.3 Carpeta release en Gestor de Documentos

Si descarga todo el repositorio en un ZIP deberá descomprimir el zip en una carpeta de su agrado; dará click derecho sobre el archivo a descomprimir, posteriormente dará click izquierdo dónde dice “Extraer todo...” y al final sólo seleccionará la carpeta dónde se extraerá y click en el botón extraer.



Figura 2.4 Extracción de Archivos

Esperamos a que se termine de extraer los documentos y al finalizar debería de, entrando a la carpeta, visualizar algo parecido a la siguiente imagen.

Este equipo > 2do disco (D:) > Alexis Cote Uni > 8vo Semestre > DESCARGAS > ProyectoFinal-master >

Nombre	Fecha de modificación	Tipo	Tamaño
Documentación	05/06/2023 08:18 a. m.	Carpeta de archivos	
ProyectoFinal	05/06/2023 08:18 a. m.	Carpeta de archivos	
Release	05/06/2023 08:18 a. m.	Carpeta de archivos	
.gitattributes	05/06/2023 08:18 a. m.	Archivo GITATTRIB...	3 KB
.gitignore	05/06/2023 08:18 a. m.	Archivo GITIGNORE	7 KB
ProyectoFinal.sln	05/06/2023 08:18 a. m.	Visual Studio Solut...	2 KB
README.md	05/06/2023 08:18 a. m.	Archivo MD	1 KB

Figura 2.5 Archivos extraídos

Concluido lo anterior deberá buscar, en la carpeta Release, el archivo que tenga la extensión **.exe** o bien diga que es una **Aplicación**.

Nombre	Fecha de modificación	Tipo	Tamaño
ProyectoFinal	04/06/2023 05:44 p. m.	Aplicación	232 KB
ProyectoFinal.pdb	04/06/2023 05:44 p. m.	Program Debug D...	2,012 KB

Figura 2.6 Archivo ejecutable.

Dé doble click izquierdo sobre el archivo y espere a la ejecución, al inicio le aparecerá una ventana negra seguida de una en blanco, no se alarme es normal, sólo espere a que en la ventana en blanco aparezca la animación tal como se muestra en la Figura 2.8.

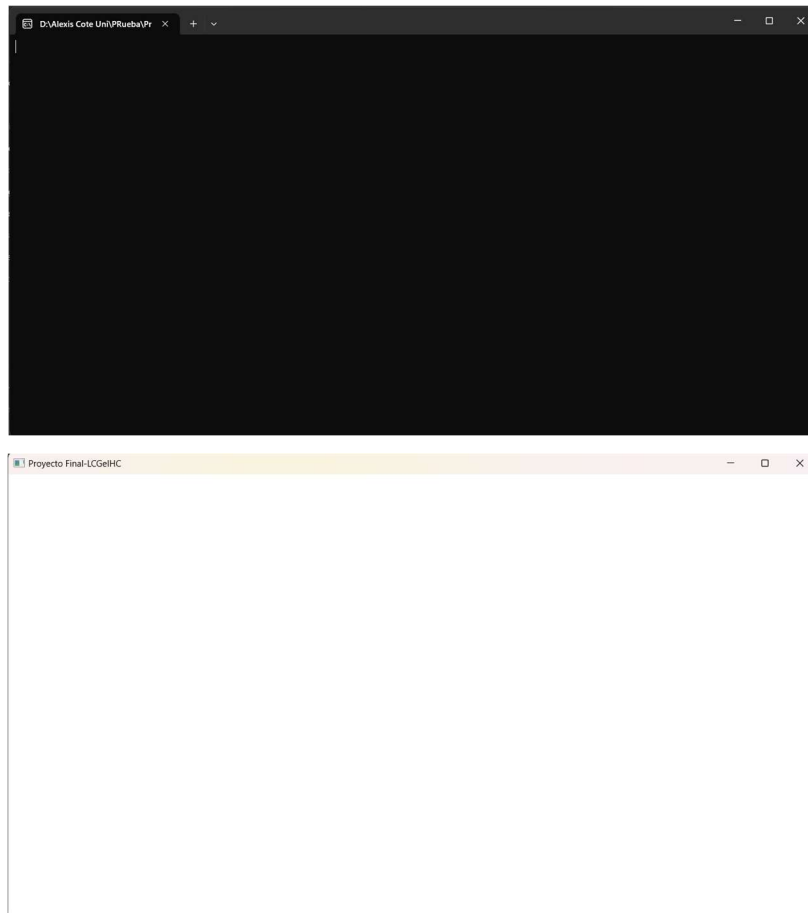


Figura 2.7 Primeras ventanas de le ejecución



Figura 2.8 Programa cargado y en ejecución

Se recomienda que desactive su antivirus cuando desee realizar la ejecución del programa debido a que, como es una aplicación de extensión **.exe** sin certificado alguno el antivirus lo puede bloquear o incluso eliminar ya que lo consideraría una amenaza; en caso de que lo elimine deberá desactivar su antivirus y realizar la extracción de los archivos nuevamente.

PROYECTO

Dentro de este proyecto usted encontrará varios documentos y carpetas sumamente esenciales para la ejecución y manipulación de nuestra aplicación.

Se cuentan con 16 archivos de extensión **.h**:

- Camera.h
- CommonValues.h
- definition.h
- DirectionalLight.h
- Light.h
- Material.h
- Mesh.h
- Model.h
- PointLight.h
- Shader_light.h
- Skybox.h
- Sphere.h
- SpotLight.h
- stb_image.h
- Texture.h
- Window.h

Por otra parte, también tenemos un total de 14 archivos de extensión **.cpp**:

- Camera.cpp
- DirectionalLight.cpp
- Light.cpp
- Material.cpp
- Mesh_tn.cpp
- Model.cpp
- PointLight.cpp
- ProyectoFinal.cpp
- Shader_light.cpp
- Skybox.cpp
- Sphere.cpp
- SpotLight.cpp
- Texture.cpp
- Window.cpp

Los archivos **.h** son los encargados de utilizar bibliotecas de OpenGL y a su vez de declarar variables que serán constantes, estructuras o clases a utilizar en su respectivo código cpp, por otro lado, los archivos **.cpp** son los encargados de definir

funciones, objetos y a la vez utilizar su respectiva biblioteca y variables que se le pasen a cada función/objeto.

- Camera.h y Camera.cpp como lo indica su nombre se emplea para poder crear y manipular una cámara sintética la cual nos permite movernos en el entorno de nuestra animación
- definition.h y CommonValues.h lo que hace es definir variables, estructuras que se utilizarán en otros códigos como referencia para declaración de vectores o creación de los mismos.
- DirectionalLight.h y DirectionalLight.cpp son los encargados de la declaración de variables, funciones, estructuras y manipulación de luces direccionales empleadas en nuestro programa.
- Light.h y Light.cpp se encarga de la creación de la luz del entorno la cual nos permite observar nuestro programa como si fuera de noche o de día.
- Material.h y Material.cpp nos permite darle características a nuestros materiales de tal forma que las componentes difusas, especular e intensidad den la característica de material opaco o brillante.
- Mesh.h y Mesh.cpp crea los mesh y permite manipularlos.
- Model.h y Model.cpp nos permite hacer la lectura de los archivos obj y sus respectivos mtl para hacer la carga de los modelos en el programa además de que nos permite hacer la modificación de los mismos referentes a rotaciones, traslaciones y escalas.
- PointLight.h y PointLight.cpp nos permite crear las luces puntuales que serán asignadas a objetos en nuestro programa.
- Shader_Light.h y Shader_Light.cpp se encarga de la creación de los shaders.
- Skybox.h y Skybox.cpp lo que realiza es la creación de las funciones, variables y manipulación de los mismos para poder crear nuestro entorno a base de imágenes dándole la característica de que este jamás será alcanzado por el usuario debido a la perspectiva que irá creando conforme se mueve la cámara.
- Sphere.h y Sphere.cpp se utilizan para la creación de esferas a partir de primitivas (triángulos).
- SpotLight.h y SpotLight.cpp crea nuestras SpotLight's.
- stb_image.h nos permite hacer la manipulación de imágenes a nivel OpenGL.
- Texture.h y Texture.cpp nos permite hacer la manipulación de imágenes de forma que creamos texturas en el programa estáticas o dinámicas.
- Window.h y Window.cpp se encargan de la creación de nuestra ventana dónde se cargará toda nuestra animación, también nos permite realizar la interacción con esta mediante el teclado y el mouse.
- ProyectoFinal.cpp es dónde se concentra prácticamente todas las declaraciones de nuestros objetos a trabajar, texturas, y animaciones.
 - Como primer paso lo que hacemos es incluir todas las bibliotecas a utilizar, tanto las bibliotecas personales que son los archivos .h

enlistados previamente y otras bibliotecas de OpenGL y del mismo compilador.

- Posteriormente desde la línea 34 hasta la 71 tendremos la declaración de nuestras variables a utilizar para nuestras animaciones tanto de objetos como de luces, estas en su mayoría son de tipo flotante pero tenemos algunas de tipo booleano.
- Del renglón 74 al 76 tenemos la declaración de nuestra ventana de trabajo.
- Luego tenemos la declaración de una esfera a utilizar para animación.
- Encontraremos la declaración de nuestra cámara sintética.
- Del renglón 81 al 84 tenemos la declaración de texturas.
- 86 y 87 tenemos la declaración de nuestros SkyBox.
- Del renglón 89 al 120 tenemos la declaración de nuestros modelos a utilizar para exportar nuestros objetos.
- Posteriormente tendremos la declaración de nuestros materiales, GLfloat, DirectionalLight's, PointLight's, SpotLight's entre otras variables.
- Del renglón 146 al 172 tendremos la declaración de nuestras normales y sus asignaciones para poder tener nuestros respectivos sombreados.
- De la 174 a la 257 tenemos la declaración de vértices y lados de objetos mesh.
- De la 258 a la 263 tenemos la creación de nuestros shaders.
- De la 265 hasta la línea 2106 tendremos nuestra función main la cual se encarga de la creación de todo nuestro programa, se encuentran declaraciones de variables que nos permiten generar animaciones, también se encuentra la carga de los modelos y texturas, como el renderizado de estos, junto con sus respectivas traslaciones, posiciones, escalado y rotaciones.

TECLAS PREDEFINIDAS

El programa cuenta con animaciones que se activan por teclado.

- Esc: Se encarga de terminar con la ejecución del programa cerrando la ventana.
- Mouse: Indicamos la dirección a la cuál estará apuntando nuestra cámara sintética.
- W: Desplaza la cámara sintética de manera positiva (hacia delante).
- S: Desplaza la cámara sintética de manera negativa (hacia atrás).
- A: Desplaza la cámara sintética de forma lateral a la izquierda.
- D: Desplaza la cámara sintética de forma lateral a la derecha.
- Flecha Up: Desplaza nuestro Avatar de forma positiva sobre el eje Z.
- Flecha Down: Desplaza nuestro Avatar de forma negativa sobre el eje Z.
- Flecha Right: Desplaza nuestro Avatar de forma positiva sobre el eje X.
- Flecha Left: Desplaza nuestro Avatar de forma negativa sobre el eje X.
- F1: Rota a nuestro avatar en el eje Y a nuestro avatar de forma antihoraria.
- F2: Rota a nuestro avatar en el eje Y a nuestro avatar de forma horaria.
- F3: Permite que el avatar agache la cabeza.
- F4: Permite que el avatar levante la cabeza.
- F5: Permite que el avatar rote su cabeza de forma antihoraria.
- F6: Permite que el avatar rote su cabeza de forma horaria.
- P: Activa la animación de los platos de fideo colocándolos y sirviéndolos.
- O: Activa la animación de los platos de fideo quitándolos de las mesas.

FALLAS

Algunas de las posibles fallas son:

- Al momento de ejecutarse se cierre de forma automática e instantánea y aparezca una notificación de su antivirus. Esto puede deberse a que el Antivirus detecta al programa como un virus, para resolverlo desactive su protección durante 15 minutos y vuelva a ejecutar.



Figura 5.1 Problemas de Antivirus

Otro motivo puede deberse a falta de archivos, asegúrese que cuenta con las carpetas de Models y Textures.

Models	05/06/2023 08:06 a. m.	Carpeta de archivos	
shaders	05/06/2023 08:06 a. m.	Carpeta de archivos	
Textures	05/06/2023 08:06 a. m.	Carpeta de archivos	
assimp-vc140-mt.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	4,025 KB
glew32.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	381 KB
glfw3.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	70 KB
ProyectoFinal	04/06/2023 05:44 p. m.	Aplicación	232 KB
ProyectoFinal.pdb	04/06/2023 05:44 p. m.	Program Debug D...	2,012 KB

Figura 5.2 Carpetas Models y Textures

- Aparezca un Error de sistema tal como el de las siguientes figuras provocando que el programa no se ejecute.

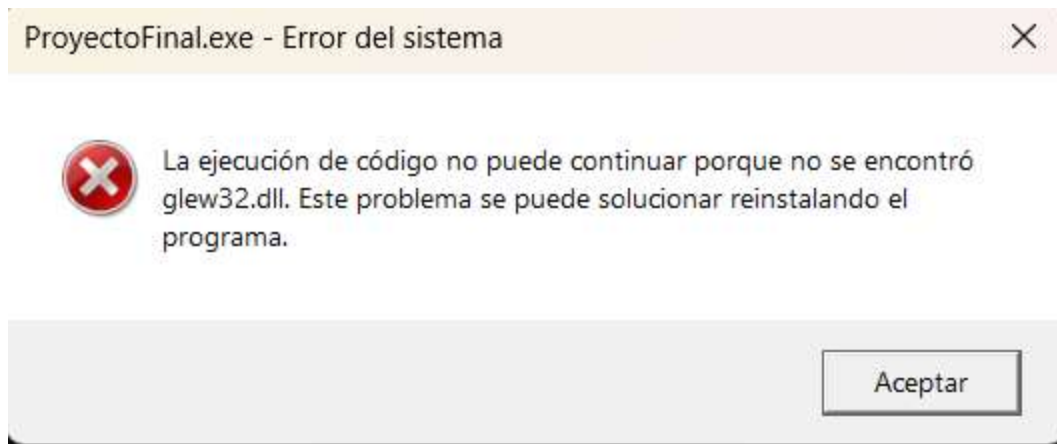


Figura 5.3 Error de Sistema 1

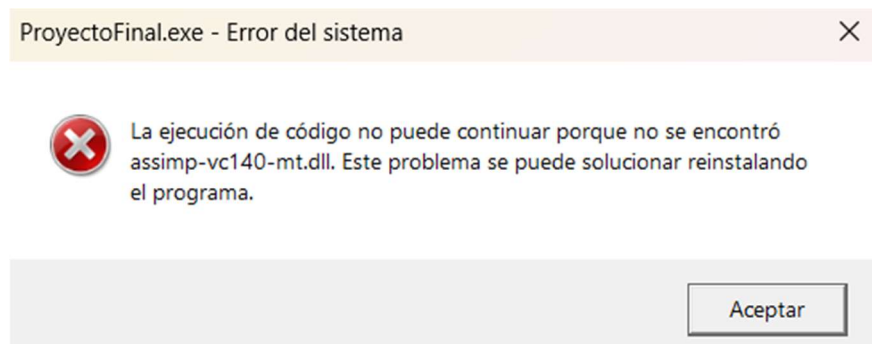


Figura 5.4 Error de Sistema 2

Esto se debe a que hacen falta los archivos **dll**, para solucionarlo asegúrese que dentro de la carpeta de la aplicación se encuentren los siguientes 3 archivos **dll**.

Nombre	Fecha de modificación	Tipo	Tamaño
Models	05/06/2023 08:06 a. m.	Carpeta de archivos	
shaders	05/06/2023 08:06 a. m.	Carpeta de archivos	
Textures	05/06/2023 08:06 a. m.	Carpeta de archivos	
assimp-vc140-mt.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	4,025 KB
glew32.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	381 KB
glfw3.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	70 KB
ProyectoFinal	04/06/2023 05:44 p. m.	Aplicación	232 KB
ProyectoFinal.pdb	04/06/2023 05:44 p. m.	Program Debug D...	2,012 KB

Figura 5.5 Archivos dll necesarios.

- Si al momento de ejecutar la aplicación, en la consola aparece algo como lo siguiente.

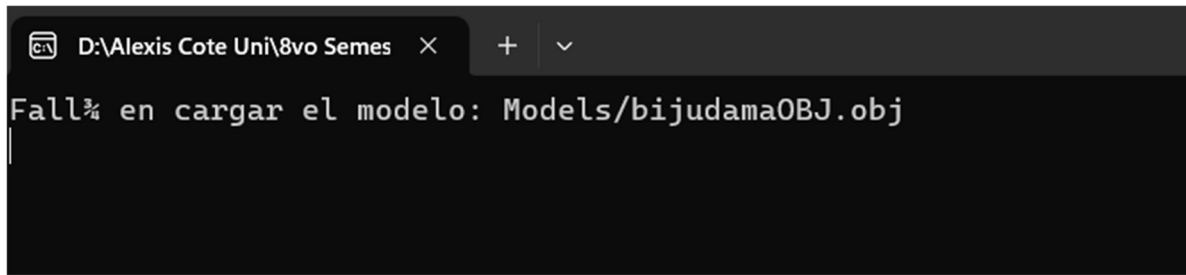


Figura 5.6 Falta de Modelo.

Puede que esto se deba a que no se descargó el modelo correspondiente, ingrese al repositorio y busque el archivo correspondiente para descargarlo. Este problema también puede presentarse con la carga de texturas, la solución es la misma, buscar el archivo faltante en el repositorio y descargarlo.

- Si la aplicación al ejecutarse no carga la animación, sólo aparece la pantalla en blanco.



Figura 5.7 Ventana en blanco

Y aparece en la ventana de la consola algo como lo siguiente.


```
D:\Alexis Cote Uni\8vo Semes x + -
Failed to read shaders/shader_light.vert! File doesn't exist.Failed to read shaders/shader_light.frag! File doesn't exist.
Error compiling the 35633 shader: '
'
Error compiling the 35632 shader: '
'
Error linking program: 'Link called without any attached shader objects.
'
Fall% en cargar el modelo: Models/GloboTerraqueo.obj
Failed to read shaders/skybox.vert! File doesn't exist.Failed to read shaders/skybox.frag! File doesn't exist.Error compiling the 35633 shader: '
'
Error compiling the 35632 shader: '
'
Error linking program: 'Link called without any attached shader objects.
'
Failed to read shaders/skybox.vert! File doesn't exist.Failed to read shaders/skybox.frag! File doesn't exist.Error compiling the 35633 shader: '
'
Error compiling the 35632 shader: '
'
Error linking program: 'Link called without any attached shader objects.
'
```

Figura 5.8 Errores

Asegúrese que se encuentra la carpeta de Shaders, de lo contrario descárguela del repositorio.

Models	05/06/2023 10:50 a. m.	Carpeta de archivos	
shaders	05/06/2023 08:06 a. m.	Carpeta de archivos	
Textures	05/06/2023 08:06 a. m.	Carpeta de archivos	
assimp-vc140-mt.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	4,025 KB
glew32.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	381 KB
glfw3.dll	04/06/2023 09:43 p. m.	Extensión de la ap...	70 KB
ProyectoFinal	04/06/2023 05:44 p. m.	Aplicación	232 KB
ProyectoFinal.pdb	04/06/2023 05:44 p. m.	Program Debug D...	2,012 KB

Figura 5.9 Carpeta Shaders