

---

**Universidad Nacional Autónoma De México**  
**Facultad de Ingeniería**  
*Computación Gráfica e Interacción*



**Prof.: Ing. Carlos Aldair Román Balbuena**  
***Proyecto Final***  
**Residencia**



**Hernández Calderón Fernando**  
***Grupo: 4, Semestre: 2021-2***  
***Fecha de Entrega: martes, 10 de agosto de 2021***

---

# Objetivos

- Diseñar una residencia donde se aplicarán los conceptos vistos en teoría.
- Diseñar un mundo virtual en el que se aplicará todo lo visto en la materia de computación gráfica utilizando códigos de uso abierto para la creación, cargado y localización de modelos en el mundo, simulando una zona residencial añadiendo la ambientación de objetos para una habitación dentro de alguno de estos modelos.
- Realizar animaciones para poder recrear de una mejor manera el espacio virtual.

## Introducción

Una API o Interfaz de Programación de Aplicaciones (en inglés, Application Programming Interface) es un conjunto de códigos que se pueden emplear para que varias aplicaciones se comuniquen entre ellas. Es algo que realiza una tarea similar a la interfaz de usuario a la hora de fomentar la interacción entre persona y programa, solo que, aplicado única y exclusivamente dentro del entorno del software. Sirve para conectar códigos o funciones entre plataformas diferentes, para que se puedan aprovechar los servicios de una web en otra.

### ¿Qué es OpenGL?

Se considera principalmente como una API que nos proporciona un gran conjunto de funciones que podemos usar para manipular gráficos e imágenes 3D y 2D. Sin embargo, por sí solo no es simplemente una API, sino una especificación, desarrollada y mantenida por el Grupo Khronos. También se usa para el desarrollo de videojuegos, donde compite con Direct3D en la plataforma de Microsoft.

Los que desarrollan las bibliotecas OpenGL suelen ser los fabricantes de tarjetas gráficas. Cada GPU admite versiones específicas de esta API, que son las versiones de OpenGL desarrolladas específicamente para esa tarjeta gráfica. Por ejemplo, cuando se utiliza un sistema de una compañía como Apple, mantienen la biblioteca OpenGL y, bajo Linux, existe una combinación de versiones de proveedores gráficos y adaptaciones para estas bibliotecas. Esto puede significar que cada vez que OpenGL muestra un comportamiento extraño que no debería, es muy probable que sea culpa de los fabricantes de tarjetas gráficas. Cada vez que hay un error en la implementación, generalmente se resuelve actualizando los controladores de la tarjeta gráfica. Esos controladores incluyen las versiones más recientes que admite su GPU.

Los principales objetivos de OpenGL son:

- Reducir la complejidad de la interfaz con las diferentes tarjetas gráficas, presentando al programador una API única y uniforme.
- Ocultar las diferentes capacidades de las diversas plataformas de hardware, requiriendo que todas las implementaciones soporten el conjunto completo de características de OpenGL (utilizando emulación por software si fuese necesario).

La operación básica de OpenGL es aceptar acciones primitivas tales como puntos, líneas y polígonos, y convertirlas en píxeles. Este proceso es realizado por una pipeline gráfica conocida como la Máquina de estados de OpenGL. La mayor parte de los comandos de OpenGL emiten operaciones primitivas a la segmentación de la gráfica. Hasta la aparición de la versión 2.0, cada etapa de la segmentación se ejecutaba en una función establecida, resultando poco configurable.

## Cotización del proyecto

### P1 Presupuesto Horas trabajadas

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Total	
1 Análisis Factibilidad, Planes y Requisitos	9	0	0	0	0	9	9%
2 Diseño	10	0	0	0	0	10	10%
3 Programación	0	18	21	0	0	39	39%
4 Documentación	0	7	7	0	0	14	14%
5 Pruebas individuales	0	1	20	0	0	21	21%
6 Integración y pruebas	0	0	2	0	0	2	2%
7 Go Life	0	0	0	1	0	1	1%
8 Revisión proyecto	0	0	0	4	0	4	4%
<b>Horas</b>	<b>19</b>	<b>26</b>	<b>50</b>	<b>5</b>	<b>0</b>	<b>100</b>	<b>100%</b>
	19%	26%	50%	5%	0%	100%	

### Hora de programacion

\$417

### P2 Presupuesto en Pesos sobre horas trabajadas

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Total	
1 Análisis Factibilidad, Planes y Requisitos	3,753	0	0	0	0	3,753	9%
2 Diseño	4,170	0	0	0	0	4,170	10%
3 Programación	0	7,506	8,757	0	0	16,263	39%
4 Documentación	0	2,919	2,919	0	0	5,838	14%
5 Pruebas individuales	0	417	8,340	0	0	8,757	21%
6 Integración y pruebas	0	0	834	0	0	834	2%
7 Go Life	0	0	0	417	0	417	1%
8 Revisión proyecto	0	0	0	1,668	0	1,668	4%
<b>Total Horas programación</b>	<b>7,923</b>	<b>10,842</b>	<b>20,850</b>	<b>2,085</b>	<b>0</b>	<b>41,700</b>	<b>100%</b>
	19%	26%	50%	5%	0%	100%	

### P3 Presupuesto en Insumos (costos)

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Total	
9 Infraestructura (SW)	5,000	0	0	0	0	5,000	24%
10 Costos Fijos (Luz, Agua, servicios,renta)	3,500	3,500	3,500	3,500	0	14,000	68%
11 Transportes	0	0	500	500	500	1,500	7%
<b>Total Costos (insumos)</b>	<b>8,500</b>	<b>3,500</b>	<b>4,000</b>	<b>4,000</b>	<b>500</b>	<b>20,500</b>	<b>100%</b>
	41%	17%	20%	20%	2%	100%	

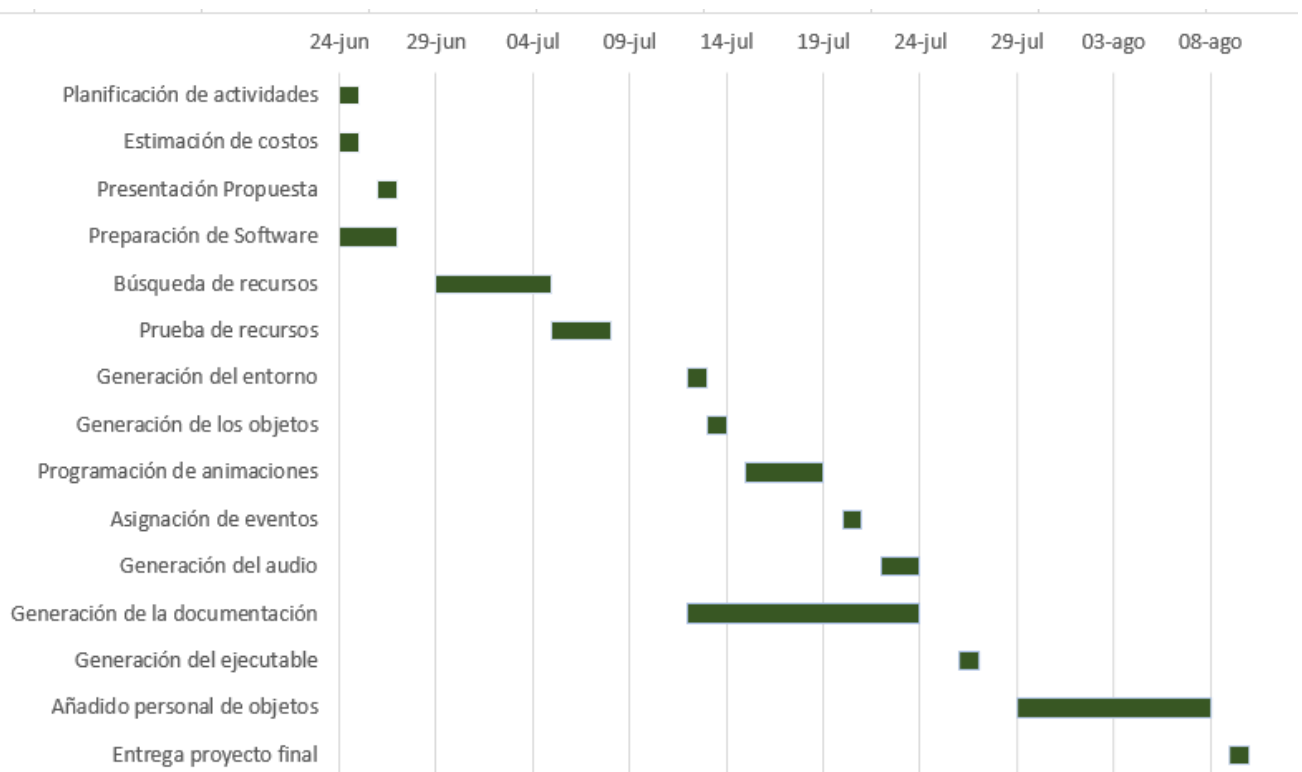
### P4 Presupuesto Total

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Total	
1 Análisis Factibilidad, Planes y Requisitos	3,753	0	0	0	0	3,753	
2 Diseño	4,170	0	0	0	0	4,170	
3 Programación	0	7,506	8,757	0	0	16,263	
4 Documentación	0	2,919	2,919	0	0	5,838	
5 Pruebas individuales	0	417	8,340	0	0	8,757	
6 Integración y pruebas	0	0	834	0	0	834	
7 Go Life	0	0	0	417	0	417	
8 Revisión proyecto	0	0	0	0	0	0	
9 Infraestructura (SW)	5,000	0	0	0	0	5,000	
10 Costos Fijos (Luz, Agua, servicios,renta)	3,500	3,500	3,500	3,500	0	14,000	
11 Transportes	0	0	0	500	500	1,000	
<b>Total Presupuesto</b>	<b>16,423</b>	<b>14,342</b>	<b>24,350</b>	<b>4,417</b>	<b>500</b>	<b>60,032</b>	
	27%	24%	41%	7%	1%	100%	

# Diagrama de actividades

Nombre actividad	Fecha Inicio	Duración en días	Fecha Fin
Planificación de actividades	24-jun	1	25-jun
Estimación de costos	24-jun	1	25-jun
Presentación Propuesta	26-jun	1	27-jun
Preparación de Software	24-jun	3	27-jun
Búsqueda de recursos	29-jun	6	03-jul
Prueba de recursos	05-jul	3	07-jul
Generación del entorno	12-jul	1	13-jul
Generación de los objetos	13-jul	1	14-jul
Programación de animaciones	15-jul	4	19-jul
Asignación de eventos	20-jul	1	21-jul
Generación del audio	22-jul	2	24-jul
Generación de la documentación	12-jul	12	24-jul
Generación del ejecutable	26-jul	1	27-jul
Añadido personal de objetos	29-jul	10	08-ago
Entrega proyecto final	09-ago	1	10-ago

## Gráfica



# Directorio de variables

Variables de manipulación y control	
<b>movAuto_x</b>	Variable para mover el auto en el eje x
<b>movAuto_z</b>	Variable para mover el auto en el eje z
<b>orienta</b>	Variable para rotar el auto en el eje y
<b>giroLlanta</b>	Variable para rotar las llantas del auto en x,y
<b>giroPuerta</b>	Variable para rotar la puerta del departamento 6 en el eje y
<b>movPersonaje</b>	Variable para trasladar los personajes en el eje z
<b>movPuerta</b>	Variable para rotar puerta del carro en x,y
<b>movCajon1 y movCajon2</b>	Variable para mover los cajones del escritorio
<b>puertaMicro</b>	Variable para abrir la puerta del microondas
<b>animacion</b>	Bandera para activar el set de animaciones 1
<b>animacionDos</b>	Bandera para activar el set de animaciones 2
<b>carroPflag</b>	Bandera para controlar el movimiento de la puerta del carro
<b>PuertaFlag</b>	Bandera para controlar el movimiento de la puerta del departamento 6
<b>cajon1Flag</b>	Bandera para controlar el movimiento del cajon1
<b>cajon2Flag</b>	Bandera para controlar el movimiento del cajon2
<b>microFlag</b>	Bandera para controlar el movimiento de la puerta del microondas
Funciones	
<b>animate</b>	Función para crear las animaciones de los objetos
<b>GetResolution</b>	Obtener la resolución optima de la tarjeta grafica
<b>Framebuffer</b>	Obtiene el tamaño del framebuffer
<b>Mouse_callback</b>	Monitorea el movimiento del mouse
<b>Scroll_callback</b>	Monitorea el movimiento del scroll del mouse
<b>my_input</b>	Para crear eventos al oprimir teclas

## Desarrollo

Para la configuración del proyecto consúltase el manual de usuario, una vez instalado el proyecto estas son las variables que se tienen a disposición para la modificación de los movimientos de los modelos, por otra parte, el proyecto se desarrolla ya en el interior del departamento 6 que es donde están todos los objetos necesarios para cumplir con los objetivos del proyecto.

El desarrollo del cargado de los modelos fue utilizando dos paginas principalmente, que son TurboSquid y CGTrader para obtener los archivos OBJ de los cuales modificaríamos para tener un escenario uniforme



**TURBOSQUID**  
by shutterstock

**cgtrader**

Abonado a esto se utilizo la herramienta de 3DS MAX para el aplicado de texturas cuando se presentaba el problema que los archivos descargados no tenían el archivo mtl o las texturas no estaban señaladas dentro de este mismo. O simplemente para modificarlos borrando otros objetos que hacían que se tuviera una carga más lenta



Algunas de las texturas como la del piso fue realizada utilizando el software nativo de Windows llamado Paint y otras a partir de texturas ya existentes fueron modificadas utilizando el software de GIMP para así crear otros colores de la misma textura como es el caso de los colores de los carros.

Desgraciadamente los objetos que fueron mencionados en el documento de referencia no pudieron ser añadidos al proyecto final por diversos problemas de texturizado y de errores en los vectores normales del archivo OBJ del modelo. Esto provoco que se tuviera que modificar un poco la apariencia final de nuestro escenario, siempre intentando respetar el aspecto general evitando que chocara el estilo de los edificios entre sí.

El proceso de ubicado tanto de los edificios como de los muebles, personajes y demás objetos se realizo lo mas estéticamente posible para no tener el fallo de objetos volando, también se cuidó que no chocara un mueble con otro y que cada cosa estuviera acorde al estilo del proyecto combinando varias texturas entre los modelos.

Finalmente destacando que puede presentarse algunos errores que se detallaran en el manual de usuario.

## Capturas de los objetos

Objetos dentro del departamento 6 designado para crear la habitación



Imagen 1. Mueble televisión (tele)



Imagen 2. Sillón con mesa (sofa)





Imagen 3. Cama (bed)

Imagen 4. escritorio con cajones (cajones y desk)

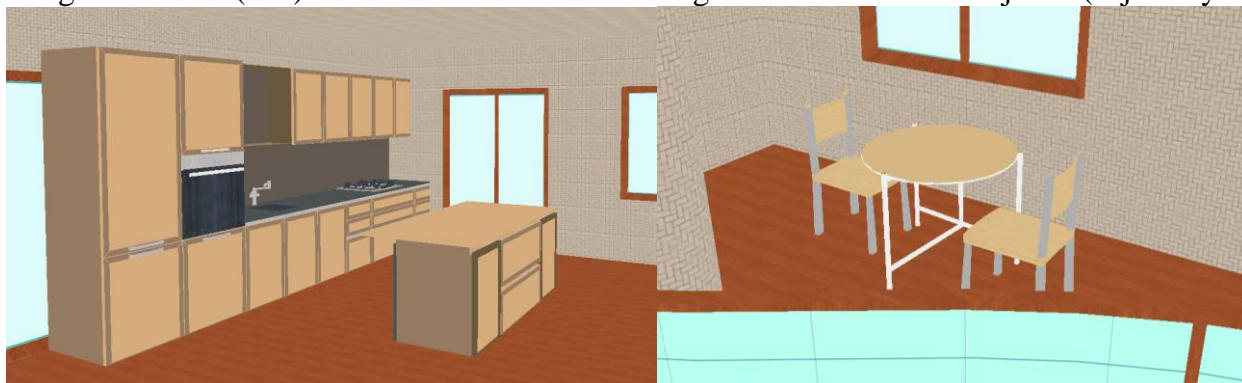


Imagen 5. Cocina (cocina)

Imagen 6. Mesita con sillas (chair y etla)

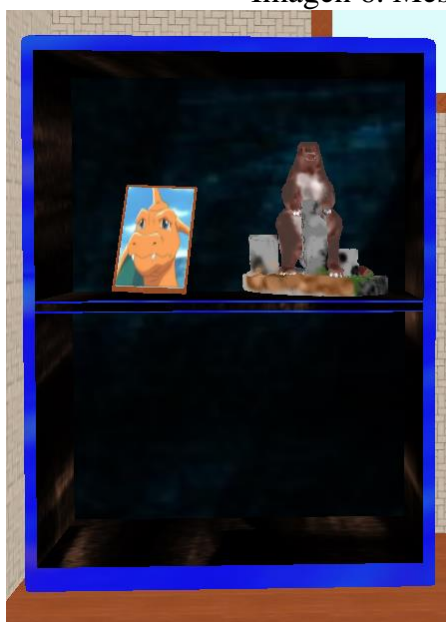


Imagen 7. Mueble con figuras (retrato, godzilla y rack)

## Objetos animados



Imagen 8. Carro que avanza



Imagen 9. Carro que abre la puerta



Imagen 10. Puerta que abre

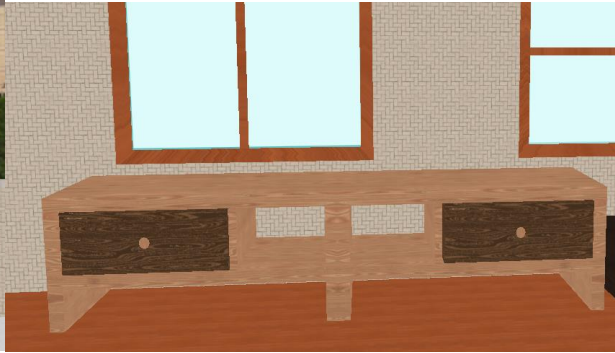


Imagen 11. Cajones que abren



Imagen 12. Microondas que abre la puerta

## Imagen propuesta inicial





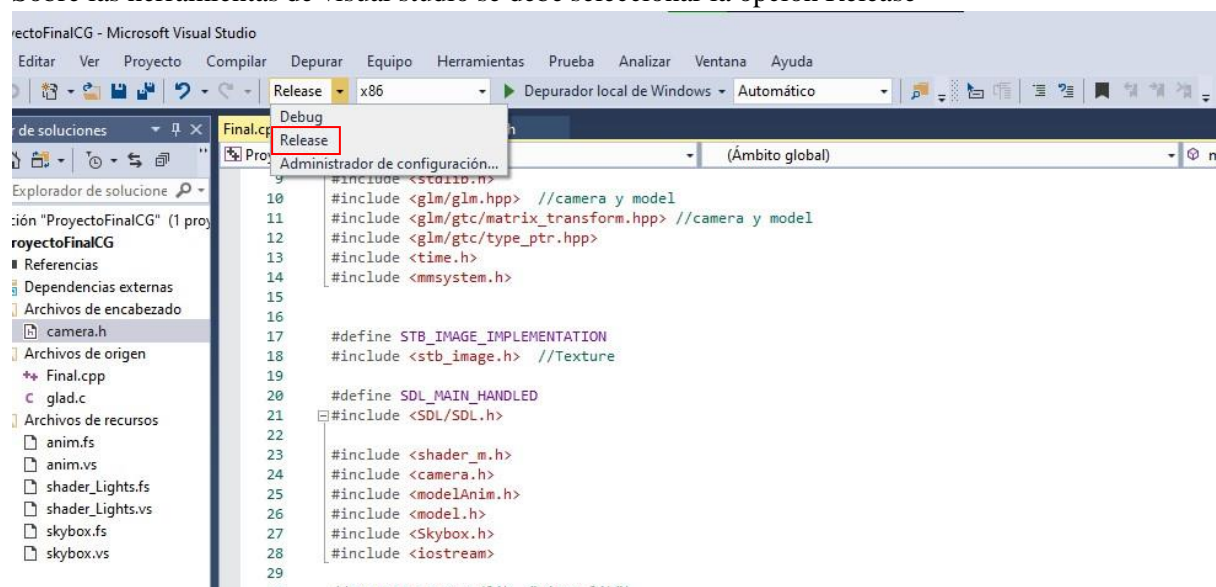
# Escenario exterior



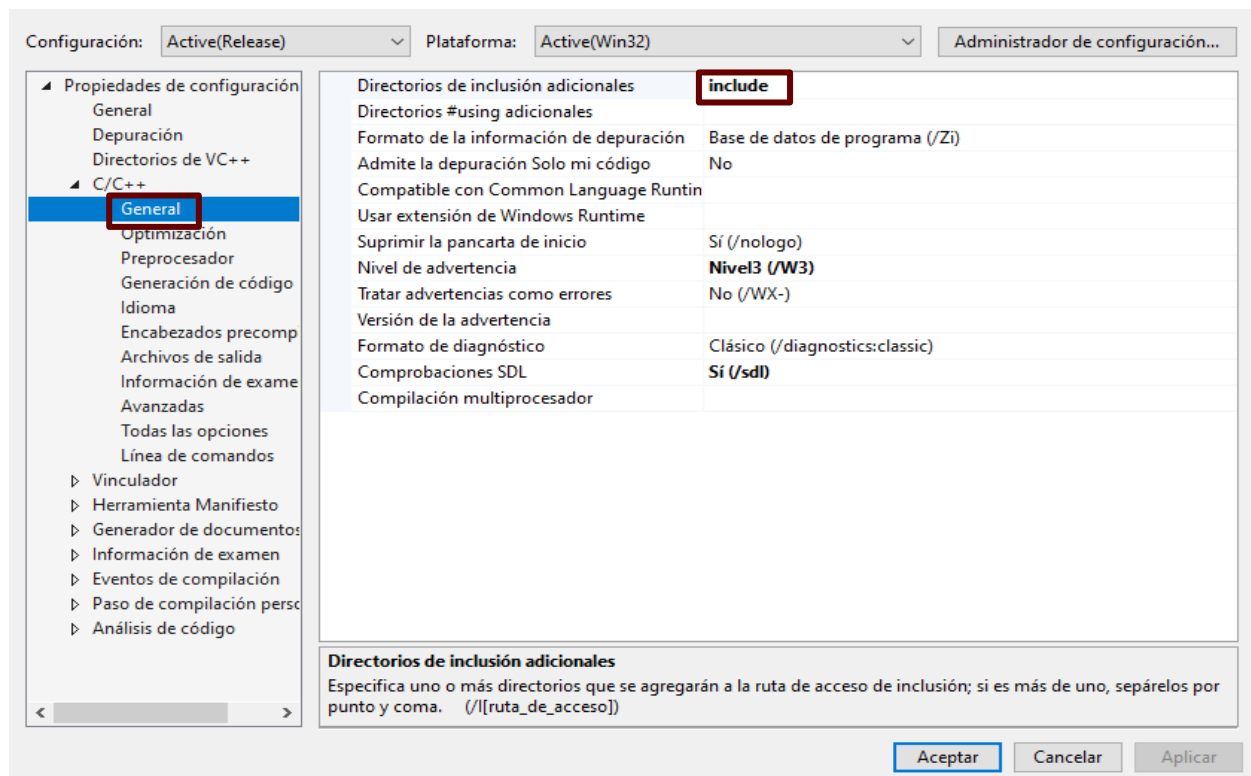
Véase el Manual de Usuario para aplicar las posibles modificaciones que requiera el proyecto y el funcionamiento de las animaciones.

## Creación del ejecutable

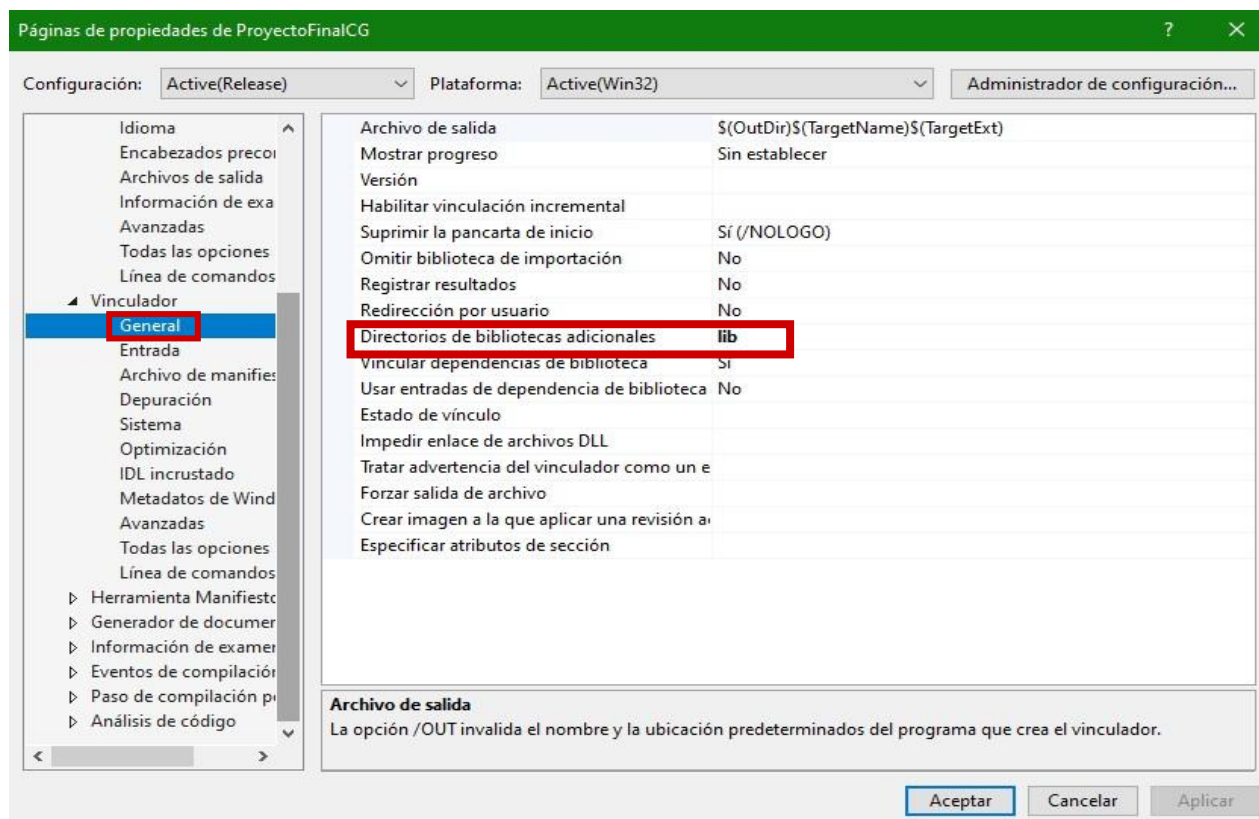
Sobre las herramientas de visual studio se debe seleccionar la opción Release



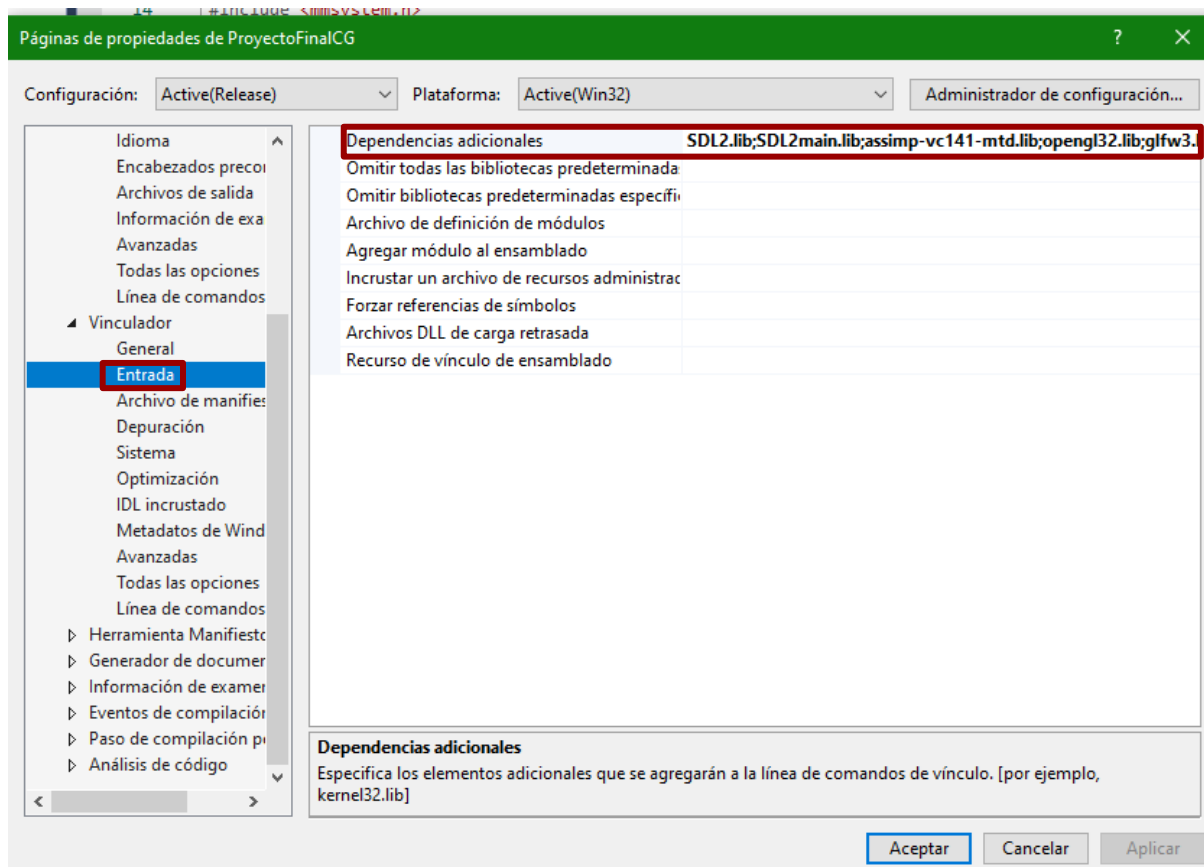
Una vez seleccionada esta opción se debe llenar la configuración del proyecto de la siguiente manera C/C++ → General → Directorios de inclusión adicional → include



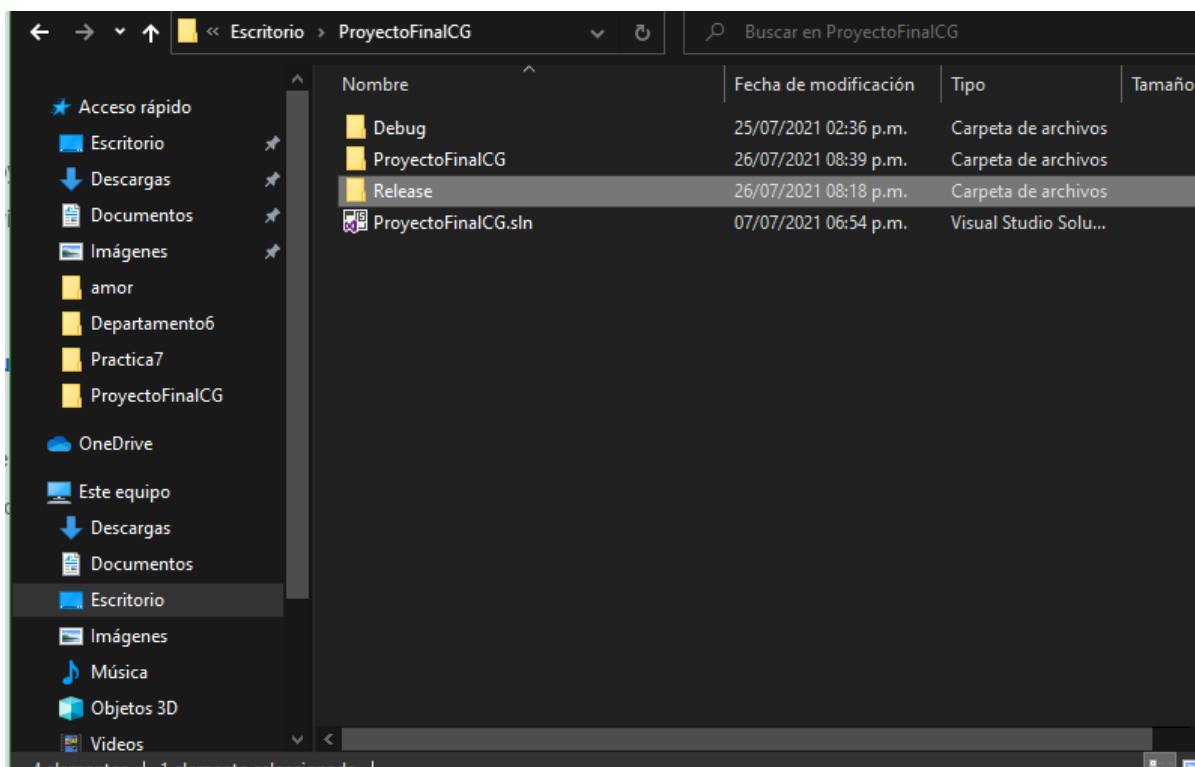
Por consiguiente, vamos a la pestaña de Vinculador → General → Directorios de bibliotecas adicionales → lib

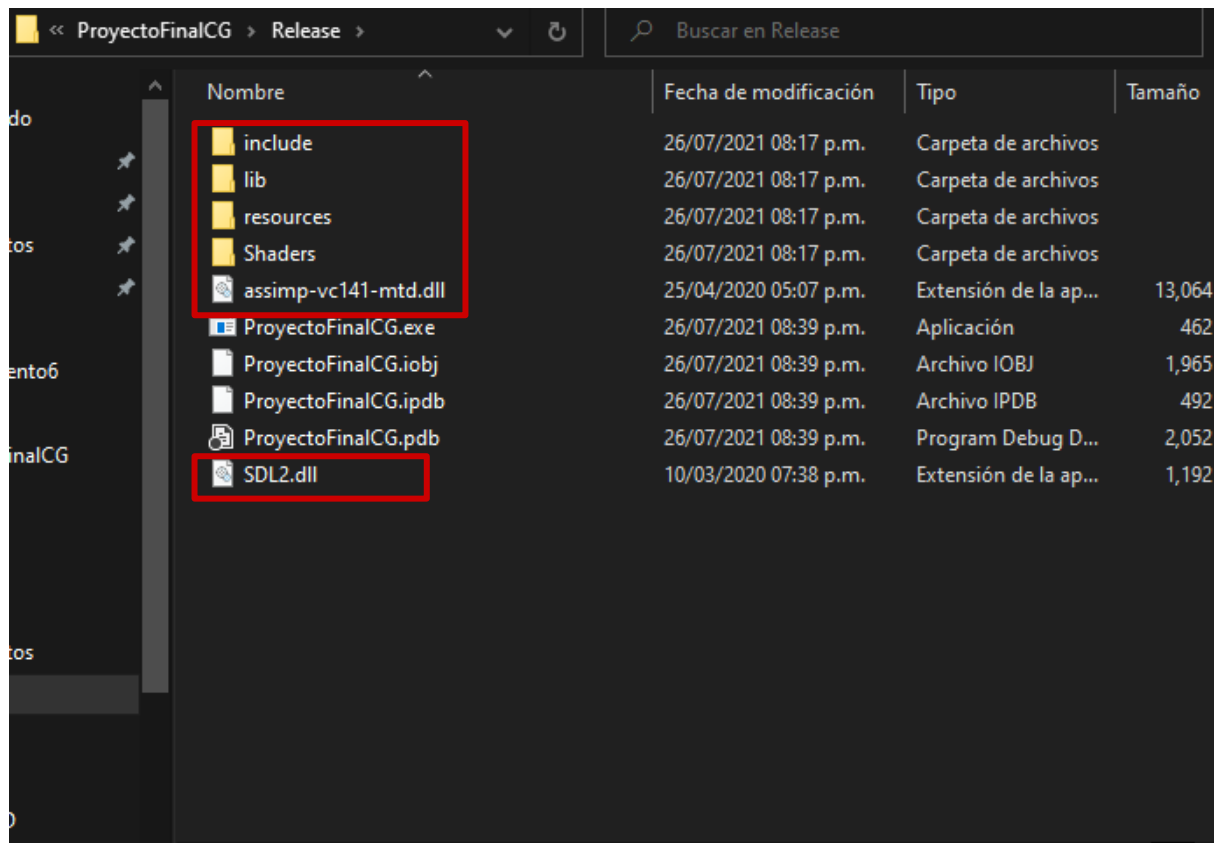


Por último nos vamos en la misma pestaña Vinculador en la parte de Entrada → Archivo desalida, la siguiente configuración **SDL2.lib;SDL2main.lib;assimp-vc141-mtd.lib;opengl32.lib;glfw3.lib;**



Esto haría que al compilar y lanzar el proyecto se cree la carpeta Release e indicara que le hace falta archivos los cuales son los modelos, Shaders, dll y bibliotecas necesarias





Ya haciendo esto estará listo para ejecutarse

## Modelos del Proyecto

Departamento 6:

<https://www.turbosquid.com/3d-models/apartment-building-6-3d-model-1435007>



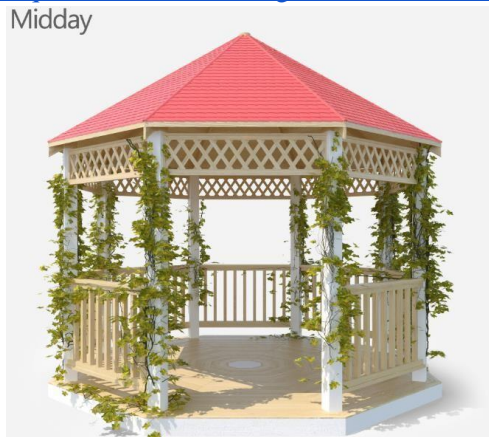


Consulado: <https://www.turbosquid.com/3d-models/house-government-russian-federation-3d-model-1463692>



Kiosco: <https://www.turbosquid.com/3d-models/gazebo-wood-3d-max/863050>

Midday



BancaParque: <https://www.turbosquid.com/3d-models/3d-table-1637017>





Edificio 4: <https://www.turbosquid.com/3d-models/apartment-house-building-3d-model-1650425>



Edificio 1: <https://www.turbosquid.com/3d-models/3d-apartment-building-architectural-model-1647087>



Sillas: <https://www.cgtrader.com/free-3d-models/furniture/chair/basel-chair-by-jasper-morrisson>



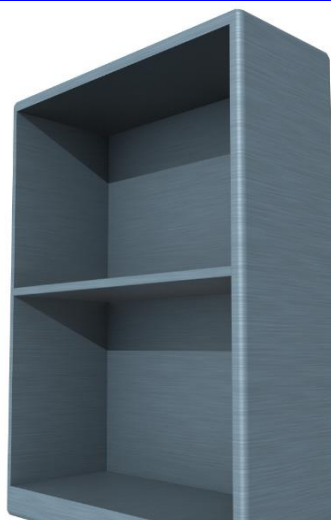
Rata: <https://www.turbosquid.com/3d-models/3d-model-mouse-1728319>



Etla: <https://www.turbosquid.com/3d-models/etla-by-cosmorelax-coffee-tables-d60d80-3d-1720568>



Rack: <https://www.turbosquid.com/3d-models/rack-3d-model-1726489>



Bed: <https://www.turbosquid.com/3d-models/bed-smania-caesar-3d-model-1537972>



Cocina: <https://www.turbosquid.com/3d-models/3d-gd-arredamenti-madison-kitchen-1521415>



Escritorio: <https://www.turbosquid.com/3d-models/table-long-3d-1676699>



Sofa: <https://www.turbosquid.com/3d-models/sofa-table-model-1665327>



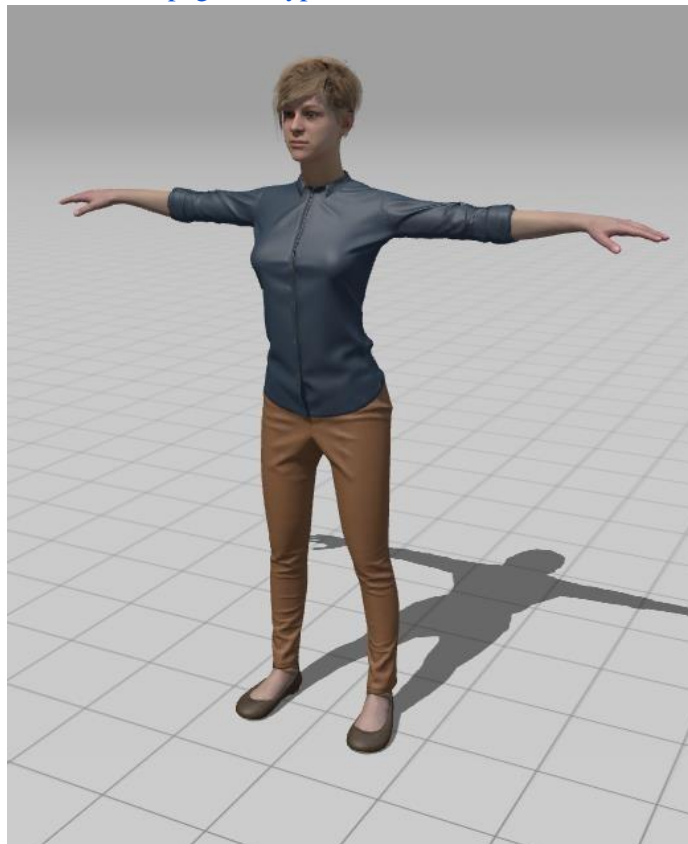
Tele: <https://www.turbosquid.com/3d-models/3d-living-room-1431240>



Ortiz: <https://www.mixamo.com/#/?page=3&type=Character>



Kate: <https://www.mixamo.com/#/?page=1&type=Character>





Megan: <https://www.mixamo.com/#/?page=1&type=Character>



## Conclusiones

Durante la realización de este proyecto se me presentaron varios inconvenientes y retos para poder lograr obtener resultados que cumplieran algunos de los objetivos planteados, gracias a los conocimientos adquiridos tanto en laboratorio como en el apartado teórico se logró comprender en cierta medida como resolver algunos de los problemas presentados durante la carga de modelos, el cargado de texturas y la modificación de los modelos utilizando el software 3ds Max

Durante la ubicación de los objetos fue un tanto tedioso el estar a prueba y error con los edificios y los vehículos del escenario, conforme se iba avanzando en el proyecto fuimos adquiriendo cierta noción de cuanto mover y en donde por lo que se obtuvo cierta experiencia sobre nuestro mundo.

Sobre la configuración que se realizó para las animaciones fue de gran ayuda la práctica de animación por teclas, además para lograr algunas de las animaciones tuvimos que reforzar nuestros conocimientos sobre la separación de polígonos sobre los modelos como por ejemplo la animación de los carros.

Finalizando puedo decir que personalmente este proyecto me ayudó a desarrollar mi creatividad y mi gusto por la edición de fotos, además que esto me dio ideas para futuros proyectos personales de mundos un poco más personalizados para así seguir aprendiendo sobre la generación de escenario y modelos 3D.