



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Computación gráfica e interacción  
humano computadora



---

## Proyecto Final

---

**PROFESOR:** Carlos Aldair Román Balbuena

**ALUMNO:** 319049792

**GRUPO:** 05

**SEMESTRE:** 2026-1

**ENTREGA:** 25 de noviembre de 2025

**MANUAL TECNICO**



---

# ÍNDICE

## Contenido

|  |    |
|--|----|
| 1.- Objetivos.....   | 3  |
| 2.- Estado del arte.....   | 3  |
| 3.- Diagrama de Gantt.....   | 4  |
| 4.- Diagrama de Flujo del software y pseudocódigo de animaciones ..... | 5  |
| 5.- Metodología de Software .....                                      | 9  |
| 6.- Alcance del proyecto.....  | 9  |
| 7.- Limitantes .....   | 12 |
| 8.- Elección de Software utilizado. ....                               | 14 |
| 9.- Evaluación del costo del proyecto.....                             | 16 |
| 10. – Documentación del Código .....                                   | 18 |
| 11. – Conclusiones .....   | 24 |
| 12. – Referencias.....   | 24 |

## 1.- Objetivos

- El alumno deberá aplicar y demostrar los conocimientos adquiridos durante el curso mediante la creación de una recreación 3D en OpenGL.
- Lograr recrear la fachada de la casa del protagonista del juego “Grand Theft Auto San Andreas”.
- Implementar 4 animaciones, Además de que cada animación debe estar en el contexto del ambiente.
- Realizar entradas de teclado, para que el usuario pueda interactuar con el ambiente 3D generado.
- Proporcionar el costo total del proyecto implementado.

## 2.- Estado del arte

Los recorridos virtuales son simulaciones digitales de ubicaciones reales o ficticias que permiten a los usuarios moverse e interactuar con un entorno sin la necesidad de estar presentes físicamente. Su aplicación actual se ha expandido más allá del mero entretenimiento, convirtiéndose en una herramienta esencial que transforma procesos de negocio y educativos al mejorar la accesibilidad, reducir costos operativos y aumentar la inmersión del usuario.

Las soluciones han evolucionado de simples imágenes 360° a entornos 3D complejos creados mediante fotogrametría o motores de juego (como Unreal Engine o Unity), ofreciendo una sensación de presencia total, especialmente a través de cascos de Realidad Virtual (VR).

Hoteles y destinos utilizan esta tecnología para crear muestrarios virtuales de sus instalaciones y servicios, aumentando la confianza del cliente antes de la reserva.

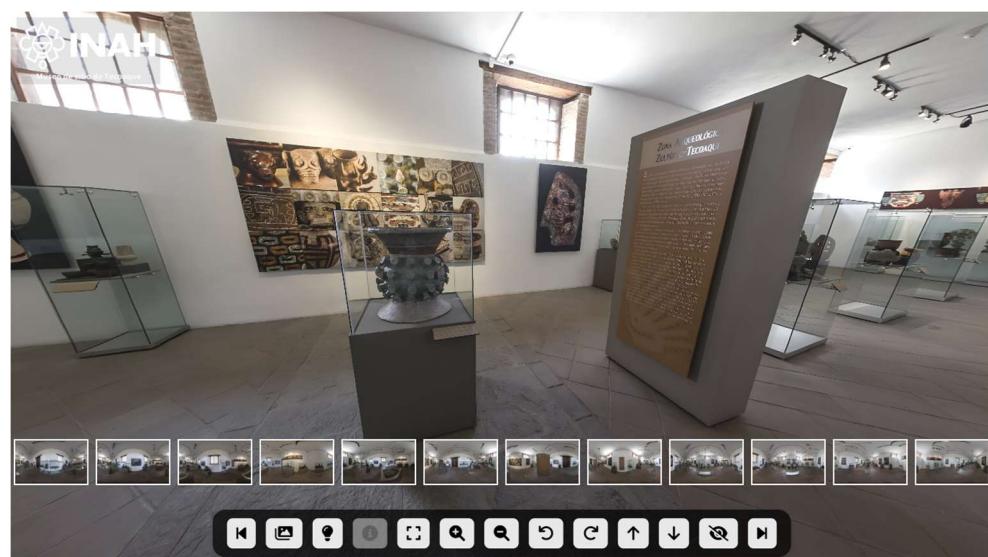
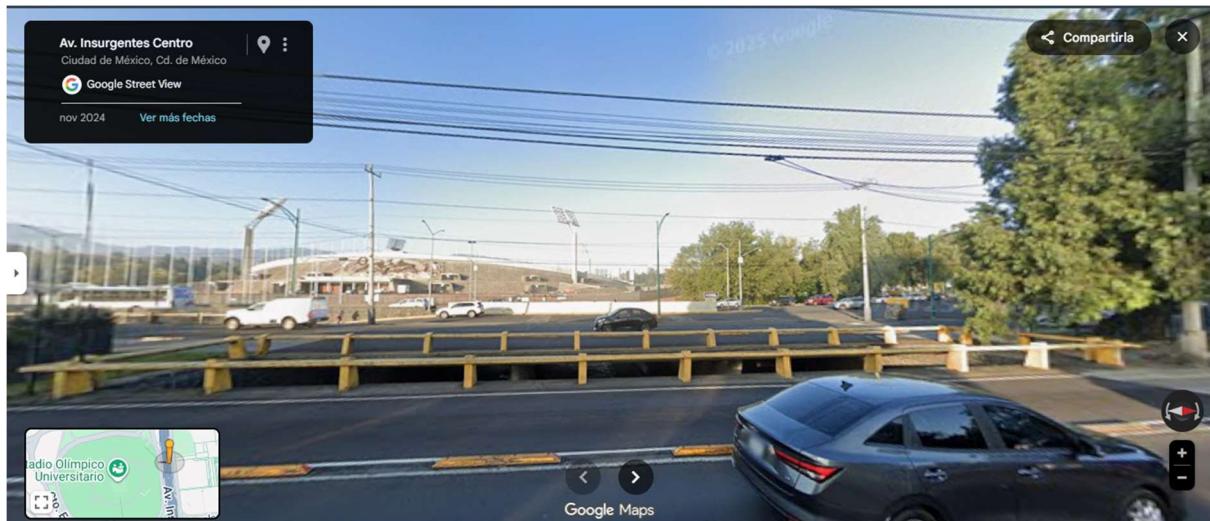
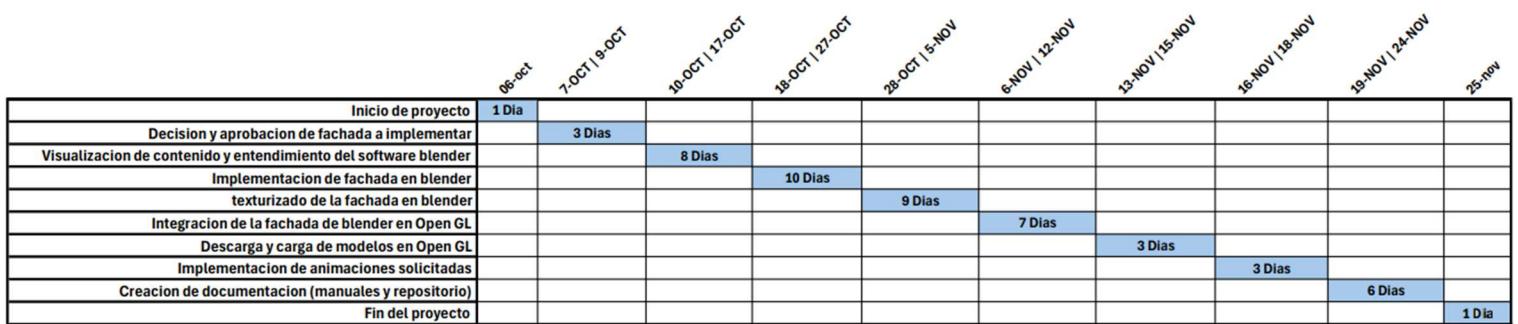


Imagen 1.1 – Recorrido virtual del museo de sitio de Tecoaque.



**Imagen 1.2 – Recorrido virtual de las calles en Google maps.**

### 3.- Diagrama de Gantt



**Imagen 1.3 - Diagrama de Gantt de las actividades realizadas durante el proyecto.**

## 4.- Diagrama de Flujo del software y pseudocódigo de animaciones

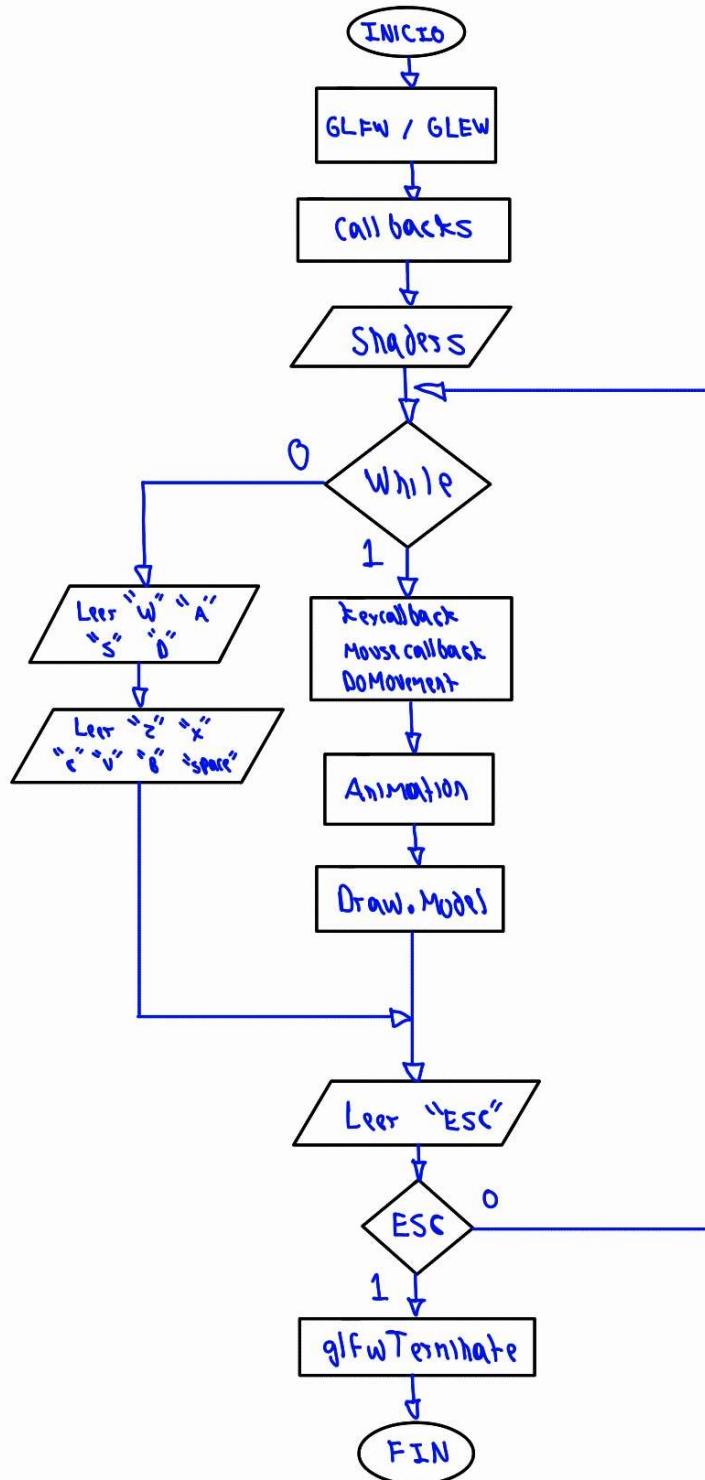


Imagen 1.4 – Diagrama de flujo de la ejecución del proyecto.

Si  $\text{rotacionPuerta} == 0^\circ$  entonces  
 $\text{rotacionPuerta} = 90^\circ$

De lo contrario

$\text{rotacionPuerta} = 0^\circ$

FIN SI

**Imagen 1.5** – pseudocódigo de la animación “Abrir o cerra puerta”.

Si  $\text{rotacionPuerta} \leq 0$  AND  $\text{puertaArriba} == \text{falso}$  entonces  
Disminuir  $\text{rotacionPuerta}$  0,5 unidades

Si  $\text{rotacionPuerta} < -20,5$  entonces  
 $\text{puertaArriba}$  es Verdadero

FIN SI

De lo contrario

Aumentar  $\text{rotacionPuerta}$  0,5 unidades

Si  $\text{rotacionPuerta} > -0,5$  entonces  
 $\text{puertaArriba}$  es Falso

FIN SI

FIN SI

**Imagen 1.6** – pseudocódigo de la animación “Abrir o cerrar puerta del garage”.

```
Si PosicionCortina == 0 AND CortinaAbierta == falso entonces
    Aumentar PosicionCortina 0.5 unidades
    Si PosicionCortina > 0.7 entonces
        CortinaAbierta es Verdadero
    Fin Si
De lo contrario
    0.5 unidades
    Si PosicionCortina < 0.01 entonces
        CortinaAbierta es Falso
    Fin Si
Fin Si
```

Imagen 1.7 – pseudocódigo de la animación “Abrir o cerrar cortina”.

```
Si Activo == Verdadero entonces
    Activar Luces Blancas
De lo contrario
    Desactivar Luces Blancas
Fin Si
```

Imagen 1.8 – pseudocódigo de la animación “Prender o apagar luces”.

Aumentar Rotacion de carro 0.1 unidades.

Aumentar timer "deltatime" veces.

Posicionllanta es 0.05 unidades.

**Si** timer >= Intervalo - cambio **entonces**

Posicionllanta es -0.05 unidades.

Decrementar timer "Intervalo - cambio" veces.

**Fin Si**

**Imagen 1.9 – pseudocódigo de la animación “Avanzar carro”.**

**Si** rotacion == 0 **entonces**

Mover personaje en el eje "z"

Rotar articulacion en sentido positivo

**De lo contrario**

Mover personaje en el eje "z"

Rotar articulacion en sentido negativo

**Fin Si**

**Si** Personaje.z == 0.5 **entonces**

Rotar personaje 90° en el eje "y"

**Fin Si**

**Si** rotacion.z == 90° **entonces**

Mover personaje en el eje "x"

Rotar articulacion en sentido positivo

**De lo contrario**

Mover personaje en el eje "x"

Rotar articulacion en sentido negativo

**Fin Si**

**Si** Personaje.x > 4.6 **entonces**

Detener Animacion

**Fin Si**

**Imagen 1.10 – pseudocódigo de la animación “movimiento personaje”.**



## 5.- Metodología de Software

La metodología que aplique en el proyecto fue Waterfall (cascada) es una metodología en la que las etapas se organizan de arriba a abajo. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden. Antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predisuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado.

Decidí implementar esta metodología, por que era la que mejor se adaptaba a mis necesidades. Al no tener un equipo de mas 2 integrantes, no debía tener reuniones de progreso sobre los avances del proyecto con otros compañeros, así como se hacen en otras metodologías agiles. El proyecto es individual y simplemente dependía de los avances que yo mismo realizaba.

Es por lo que en el diagrama de Gantt las actividades realizadas en el proyecto son secuenciales, por eso primero decidí que fachada implementar, luego aprender a modelar en blender, y proceder a modelar la casa, texturizarla y cargarla con las librerías de open gl. Para dejar para después el proceso de amueblar la casa con modelos descargados. Por último, en el Código, implementar las animaciones solicitadas, ya teniendo en cuenta los objetos con los que cuento y el espacio 3D donde se realizaran cada una.

## 6.- Alcance del proyecto

En este proyecto solo se queda hasta el diseño por fuera de la casa del protagonista del juego “Grand Theft Auto San Andreas”, en este caso me enfoque en desarrollar la vista frontal de la casa. Que sería la parte de la puerta blanca y el pequeño pasillo que se encuentra en el lado izquierdo. Además de la sección del garaje que pertenece a la casa del protagonista del juego.



**Imagen 2.1** – referencia de la fachada creada.



**Imagen 2.2** – Modelo de la fachada implementada en el proyecto.

Ya para el interior de la casa solo desarrolle 2 cuartos, la cocina y la sala. La cocina contiene los elementos físicos visuales más importantes, como los son la estufa, refrigerador y guarda platos que fueron hechos en Open GI. Los demás elementos fueron modelados desde Blender o descargados desde Sketchfab y cargados a Open GI.



**Imagen 2.3** – referencia de la cocina de la fachada.



**Imagen 2.4** – Modelo de la cocina realizada en el proyecto.

El último cuarto implementado fue la sala de la casa, este fue el cuarto más difícil de amueblar ya que tenía una mayor cantidad de elementos. y muchos de los elementos eran tardados de modelar, por lo que decidí solo crear los objetos con open gl, los más fáciles. En este caso yo modele con open gl la planta ubicada en la entrada de la cocina, una mesa, una silla y el cuadro fotográfico que contiene la foto de un paisaje del propio juego. Las demás fotografías y mesas para la planta las realice en blender, y los demás objetos son modelos descargados de Sketchfab.



**Imagen 2.5** – referencia de la sala de la fachada.



**Imagen 2.6** – Modelo de la sala realizada en el proyecto.

## 7.- Limitantes

Las limitaciones del proyecto son notorias ya que decidí omitir los cuartos de arriba de la casa, dentro de la casa solo tengo las escaleras de adorno, ya que no llevan a ningún lado y chocan con las paredes de la casa.

Para este proyecto decidí solo enfocarme en los 2 cuartos del primer piso ya que son los primeros que conocemos en el juego, son los más populares, que causan una sensación de recordarlos más fácil y además son los más reconocidos de la casa por las personas que jugaron el juego.



**Imagen 3.1** – escaleras de adorno implementadas.



**Imagen 3.2** – Escaleras del juego que llevan al piso de arriba, NO implementadas en este proyecto.



**Imagen 3.3** – Parte de arriba de la casa vacía, solo para adornar el exterior de la fachada.



### Imagen 3.4 – Cuartos del piso de arriba que NO fueron implementados en el proyecto.

Otra característica para tomar en cuenta fue la computadora en la que desarrolle el proyecto. Ya que los recursos con los que cuenta mi computadora son decentes. Ya que no cuento con una tarjeta gráfica en mi computadora, por lo que mi procesador y mi memoria RAM se destinan en el procesamiento de gráficos y vuelve más lento el proceso de compilación del proyecto.

Debido a esta limitante en los recursos de mi computadora, me retrase en el cargado de modelos en open gl, además de que a la hora de agregar los modelos descargados. La compilación del proyecto duraba hasta minuto y medio en ejecutarse, por lo que volvió lento el desarrollo del proyecto.

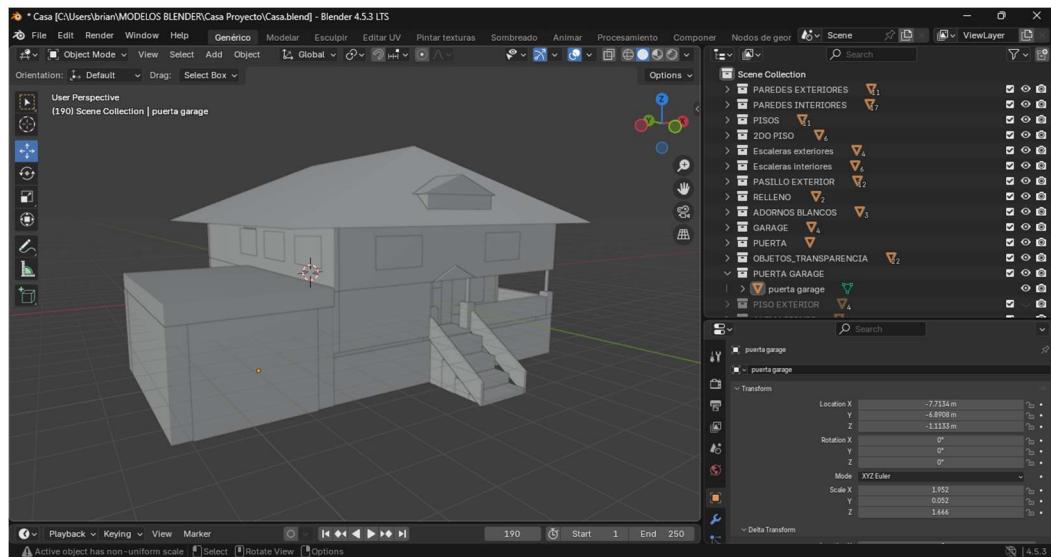
|   |   |   |  |
|---|---|---|--|
|  Almacenamiento<br><b>1.39 TB</b><br><br>347 GB de 1.39 TB usado |  Tarjeta gráfica<br><b>2 GB</b><br><br>AMD Radeon(TM) Graphics |  RAM instalada<br><b>16.0 GB</b><br><br>Velocidad: 2400 MT/s |  Procesador<br><b>AMD Ryzen 5 3450U with Radeon Vega Mobile Gfx</b><br><br>2.10 GHz |
|---|---|---|--|

Imagen 3.5 – recursos limitados de la computadora con la que realice el proyecto.

## 8.- Elección de Software utilizado.

### *Blender*

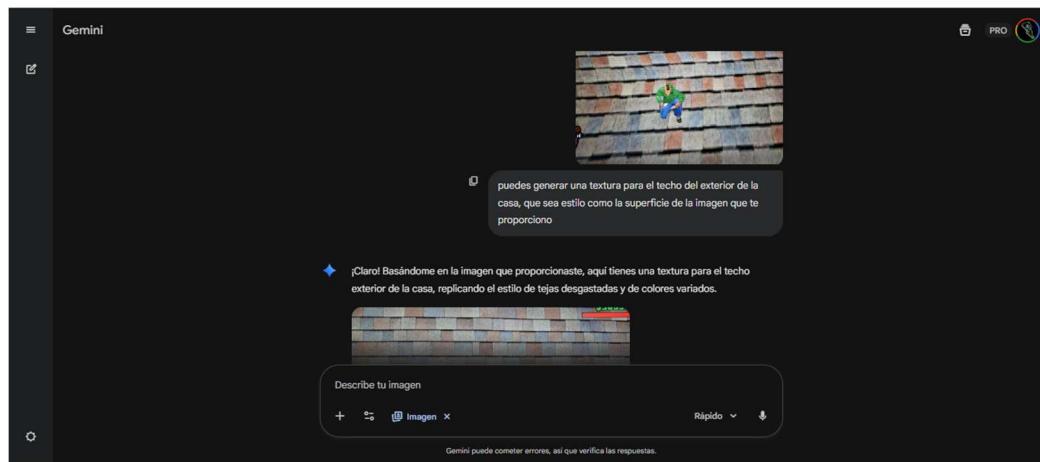
Para modelar la fachada de la casa utilice “Blender”. Yo decidí utilizarlo para modelar y texturizar toda la fachada que yo realicé. Aunque existen otros programas de modelado, yo preferí usar Blender debido a que es gratuito, fácil de instalar y no requiere crear una cuenta antes de usarlo, además existía mucho contenido de internet que te daban los fundamentos de como manejar el programa. Además, no solamente fueron esas las razones, si no que mis amigos que también llevan esta materia iban a utilizar Blender para modelar sus propias fachadas. Y podíamos tener apoyo entre nosotros mismos para cualquier problema que nos surgiera a cualquiera de nosotros.



**Imagen 3.6 – Modelado de la fachada en Blender.**

### **Generador de imágenes Gemini PRO**

Para poder texturizar la fachada o cualquier otro modelo se necesitan imágenes de cierto tamaño de pixeles, por lo que descargar imágenes de internet pude ser tardado y no siempre se encuentran las imágenes con las especificaciones que necesitamos. Por eso decidí utilizar el generador de imágenes de Gemini PRO, solo describía la imagen de la textura que deseaba obtener y además solía proporcionar una imagen del juego, para que tuviera un mejor contexto de lo que deseaba que me construyera. El único inconveniente que tuve fue cuando Gemini tardaba cada vez mas en generar las imágenes de las texturas que solicitaba, y en ocasiones la imagen que generaba era totalmente distinto a lo que le pedía o a lo que le proporcionaba como referencia.



**Imagen 3.7 – generación de las texturas con Gemini PRO.**

### Gimp

Muchas veces las imágenes que me genero Gemini tenían algunos detalles que deseaba eliminar o modificar, por lo que fue necesario editar las imágenes. Para eso utilice Gimp, el cual es un software de edición de imágenes. Utilice este software debido a que es gratuito, puedes descargarlo fácilmente y no te pide iniciar sesión con una cuenta para poder usarlo. Además, es un editor de imágenes que ya he utilizado en muchas ocasiones, por lo que no tuve que ver contenido o buscar referencias en internet. Es un software muy potente y me dejaba hacer todas las modificaciones que deseaba en las imágenes, como agregar el canal alfa a las imágenes que requieren transparencia, quitar marcas de agua de las imágenes de Gemini, cambiar el color de las texturas, cambiar el tamaño de pixeles de las imágenes o simplemente cambiar el formato de las imágenes.

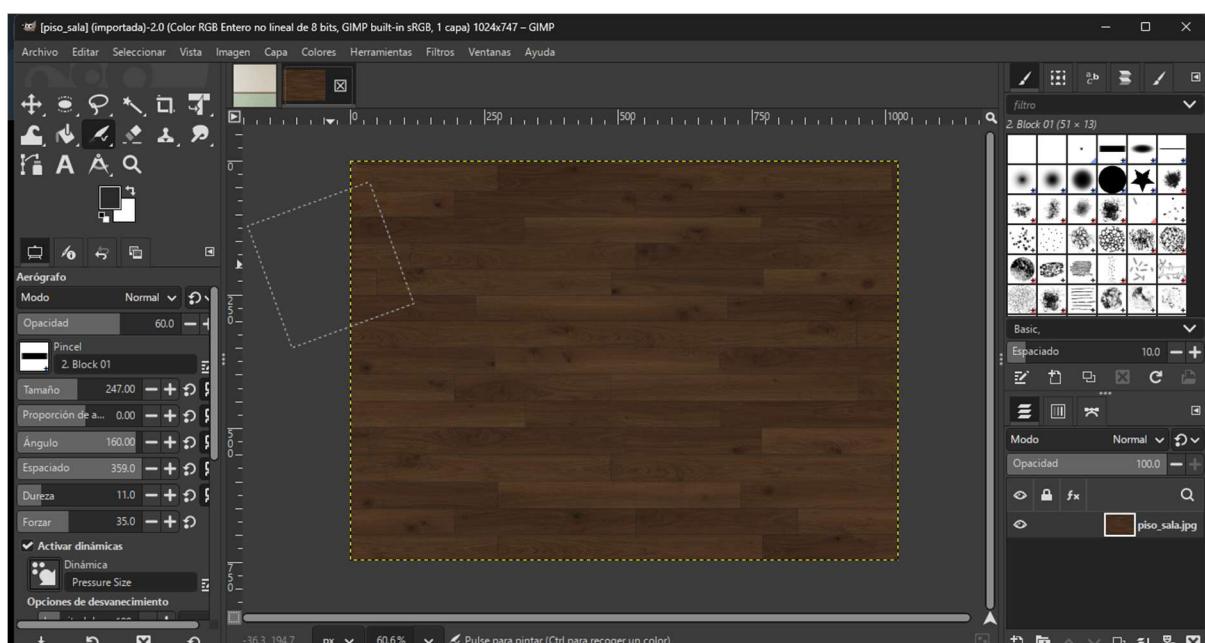


Imagen 3.8 – edición de las texturas con Gimp.

## 9.- Evaluación del costo del proyecto.

Para estimar el costo total del proyecto, se tomaron en cuenta los siguientes aspectos, el sueldo del programador, la computadora con la que se realizó el proyecto, el uso de licencias de pago, en este caso la única licencia de pago fue Gemini PRO para la generación de las imágenes de texturas, ya que los demás softwares, como Blender y Gimp son gratuitos y no requieren ningún pago de licencia.

También se consideró los pagos mensuales de planes para los servicios de internet y energía eléctrica. Ya por último se consideró el lugar físico donde se desarrollará el proyecto, en este caso una oficina ubicada en la Ciudad de México.



| Producto                  | Monto Mensual   | Monto Total del producto | Descripción  | Referencia  |
|---------------------------|---|--------------------------|--|---|
| Sueldo Programador        | \$25,000  | \$50,000                 | Es el sueldo del programador promedio en c++.          | <a href="https://mx.computrabajo.com/">https://mx.computrabajo.com/</a>   |
| Equipo (Computadora)      | <small>valor computadora<br/>5(depreciación)<br/>12 (meses del año)<br/>* 2(duración proyecto)</small><br>\$158.3 | \$316.6                  | Computadora con la que se realizó el proyecto.         | <a href="https://www.walmart.com.mx/ip/laptop-gamer-dell-inspiron-ryzen-5-16gb-1tb-256gb-ssd-vega-8-dell-3515-v3/00070452111687">https://www.walmart.com.mx/ip/laptop-gamer-dell-inspiron-ryzen-5-16gb-1tb-256gb-ssd-vega-8-dell-3515-v3/00070452111687</a> |
| Gemini Pro                | \$395   | \$790                    | Licencia de Gemini para la generación de las texturas. | <a href="https://gemini.google/subscriptions">https://gemini.google/subscriptions</a>   |
| Plan de Internet          | \$500   | \$1,000                  | Tarifa del plan de internet,                           | <a href="https://www.totalplay.com.mx/paquetes">https://www.totalplay.com.mx/paquetes</a>   |
| Plan de energía eléctrica | \$500   | \$1,000                  | Tarifa del plan de consumo de energía eléctrica.       | <a href="https://www.cfe.gob.mx/hogar/tarifas/">https://www.cfe.gob.mx/hogar/tarifas/</a>   |
| Oficina de trabajo        | \$3,500   | \$7,000                  | Oficina física donde se desarrolló el proyecto.        | <a href="https://www.reglas.com/">https://www.reglas.com/</a>   |

Para estimar el costo final, se realizó la suma de los productos del campo “Monto Total del Producto”, por lo que el costo total del proyecto es **\$60,106.6, sesenta mil ciento seis punto seis pesos mexicanos.**

## 10. – Documentación del Código

### Funciones

| Tipo de Función | Nombre de la Función | Parámetros que Recibe   | Descripción  |
|-----------------|----------------------|---|--|
| void            | KeyCallback          | GLFWwindow *window, int key, int scanCode, int action, int mode | Maneja los eventos de entrada del teclado (teclas presionadas/liberadas). Se usa para el control de la cámara, la luz, el inicio/grabación de <i>keyframes</i> , y las acciones de la máquina de estados.  |
| void            | MouseCallback        | GLFWwindow *window, double xPos, double yPos                    | Maneja los eventos de entrada del ratón (movimiento del cursor). Se usa para controlar la orientación (pitch y yaw) de la cámara sintética.  |
| void            | DoMovement           | Ninguno   | Procesa el estado actual del teclado (keys[]) para aplicar el movimiento de traslación de la cámara (W, A, S, D) de forma fluida utilizando deltaTime.   |
| void            | Animation            | Ninguno   | Contiene la lógica principal de la animación. Se encarga de actualizar las variables de posición y rotación (dogPosX, head, rotDog, etc.) en cada fotograma, especialmente durante la reproducción (play == true) y la interpolación de <i>keyframes</i> . |
| GLunit          | loadTexture          | Char: ruta del archivo de textura                               | Carga y configura texturas desde archivos de imagen.   |
| int             | main                 | Ninguno   | Función principal de la aplicación. Se encarga de la inicialización de GLFW/GLEW, la configuración de los <i>shaders</i> y <i>buffers</i> , la carga de modelos, y el bucle de renderizado (Game loop).  |

## Variabes

| Tipo de Variable | Nombre de la Variable       | Descripción   |
|------------------|-----------------------------|---|
| const GLuint     | WIDTH, HEIGHT               | Dimensiones iniciales de la ventana en píxeles.   |
| int              | SCREEN_WIDTH, SCREEN_HEIGHT | Dimensiones reales del framebuffer de la ventana obtenidas en tiempo de ejecución.                            |
| Camera           | camera                      | Instancia de la clase Camera que controla la matriz de vista y posición de la cámara en la escena 3D.         |
| GLfloat          | lastX, lastY                | Últimas coordenadas X e Y del cursor del mouse para calcular el desplazamiento.                               |
| bool[1024]       | keys                        | Array que almacena el estado (presionado/liberado) de todas las teclas del teclado.                           |
| bool             | firstMouse                  | Bandera que indica si es el primer movimiento del mouse para inicializar correctamente las coordenadas.       |
| glm::vec3[]      | pointLightPositions         | Array que almacena las posiciones en el espacio 3D de las luces puntuales de la escena.                       |
| glm::vec3        | Light1                      | Vector que define el color e intensidad de una luz variable en la escena (usada para efectos dinámicos).      |
| glm::vec3        | lightPos                    | Posición de una luz adicional en el espacio 3D (usualmente una luz puntual o foco, actualmente en el origen). |

|       |                                 |   |
|-------|---------------------------------|---|
| bool  | active                          | Controla si el sistema de iluminación (o alguna característica específica) está activo o desactivado globalmente. |
| float | rotCarro                        | Ángulo de rotación en grados del automóvil alrededor del eje Y.   |
| float | rotPuerta                       | Ángulo de rotación en grados de la puerta principal de la casa.   |
| float | rotPuertaGarage                 | Ángulo de rotación o posición vertical de la puerta del garage.   |
| float | posCortina                      | Desplazamiento horizontal de las cortinas para abrir/cerrar.  |
| float | tamllanta                       | Desplazamiento de las llantas del automóvil para efecto de rotación (simulación de avance).                       |
| float | timer                           | Acumulador de tiempo para controlar intervalos en animaciones discretas o cíclicas.                               |
| float | FLegs, RLegs                    | Ángulos de rotación para animar piernas y brazos del personaje (piernas delanteras y traseras).                   |
| float | personajePosX,<br>personajePosZ | Coordenadas X y Z de la posición del personaje en el mundo 3D.  |
| float | personajeRot                    | Ángulo de rotación en grados del personaje alrededor del eje Y.   |

|             |                        |  |
|-------------|------------------------|--|
| float       | posPuertaX, posPuertaZ | Desplazamientos de posición adicionales para la puerta principal.  |
| const float | INTERVALO_CAMBIO       | Intervalo de tiempo constante (0.1 segundos) para controlar el ritmo en animaciones por pasos.                                     |
| bool        | AnimCarro              | Bandera que activa/desactiva la animación del automóvil.   |
| bool        | AnimPuerta             | Bandera que activa/desactiva la animación de la puerta principal.  |
| bool        | AnimGarage             | Bandera que activa/desactiva la animación de la puerta del garage.   |
| bool        | AnimCortina            | Bandera que activa/desactiva la animación de las cortinas.   |
| bool        | AnimPersonaje          | Bandera que activa/desactiva la animación del personaje.   |
| bool        | puertaArriba           | Estado que indica si la puerta del garage está arriba (abierta).   |
| bool        | cortinaAbierta         | Estado que indica si las cortinas están abiertas.  |
| bool        | step                   | Bandera auxiliar para controlar el paso alternado en la animación del personaje (ej., para alternar el movimiento de las piernas). |

|          |                |   |
|----------|----------------|---|
| GLfloat  | deltaTime      | Tiempo transcurrido entre el frame actual y el anterior en segundos (para movimiento independiente de FPS). |
| GLfloat  | lastFrame      | Tiempo en segundos del último frame renderizado (usado para calcular deltaTime).                            |
| float[]  | vertices       | Array con datos de vértices para cubos o planos genéricos (posición, normal, coordenadas UV).               |
| GLuint[] | indices        | Array con índices para renderizado eficiente de triángulos mediante EBO.                                    |
| GLuint   | VBO            | Vertex Buffer Object - Almacena datos de vértices en la memoria de la GPU.                                  |
| GLuint   | VAO            | Vertex Array Object - Almacena la configuración de los atributos de vértices (punteros, stride, offset).    |
| GLuint   | EBO            | Element Buffer Object - Almacena los índices para el renderizado por elementos.                             |
| Shader   | lightingShader | Programa de shader para objetos con iluminación de Phong (el shader complejo).                              |

|           |   |   |
|-----------|---|---|
| Shader    | lampShader                              | Programa de shader simple para representar visualmente las fuentes de luz.                        |
| glm::mat4 | projection                              | Matriz de proyección perspectiva para transformar coordenadas 3D a 2D (Clip Space).               |
| glm::mat4 | view                                    | Matriz de vista que define la posición y orientación de la cámara.                                |
| glm::mat4 | model                                   | Matriz de modelo para transformaciones de objetos individuales antes de ser dibujados.            |
| glm::mat4 | modelTemp,<br>modelTemp1,<br>modelTemp2 | Matrices temporales utilizadas para la composición de transformaciones en el modelado jerárquico. |
| float     | velocidad                               | Velocidad base para movimientos y animaciones (usada para modular deltaTime).                     |
| Model     | Piso_Exterior, Skybox, Casa, etc.       | Instancias de modelos 3D cargados desde archivos OBJ para renderizar.                             |
| GLuint    | madera, marco, hoja, etc.               | Identificadores de texturas OpenGL cargadas desde archivos de imagen.                             |



## 11. – Conclusiones

En este proyecto final logré integrar cada uno de los conceptos que vimos en cada en cada una de las clases de teoría y laboratorio, para lograr integrar cada una de las funcionalidades, comencé a partir del Código de máquina de estados que me proporcionaron en el laboratorio de la materia, ya que lo considere el más completo para utilizarlo y agregar o eliminar funcionalidades. En la realización del proyecto no solamente utilice open gl, tuve que recurrir a otras herramientas de modelado 3D como “blender”, y también de software de manipulación de imágenes como “gimp”, para la creación de las imágenes que utilice como texturas tuve que utilizar la inteligencia artificial, en específico el generador de imágenes de “Gemini”, yo lo proporcione imágenes tomadas por mi celular del juego y le pedía a Gemini que genere las texturas.

Por último, para no tener que modelar yo todo el ambiente, fui a descargar modelos de la página “Skechtfab”, los cuales muchos no estaban bien texturizados y yo tuve que manipular sus coordenadas UV para que se pudieran visualizar correctamente en Open GI. Por último, agradezco que los profesores de teoría y de laboratorio dejaron la implementación libre del proyecto, ya que este proyecto no solamente implica cuestiones técnicas de las herramientas utilizadas, sino que también depende de la creatividad que el mismo alumno coloca en su proyecto.

## 12. – Referencias

Blender Flow [@BlenderFlow]. (s/f). *Como aplicar texturas fácil !! | | Blender 3.1 [[Object Object]]*. Youtube. Recuperado el 15 de noviembre de 2025, de <https://www.youtube.com/watch?v=ik9Ldysa6zE>

Casa de los Johnson. (s/f). Grand Theft Encyclopedia; Fandom, Inc. Recuperado el 15 de noviembre de 2025, de [https://gta.fandom.com/es/wiki/Casa\\_de\\_los\\_Johnson](https://gta.fandom.com/es/wiki/Casa_de_los_Johnson)

DansterDev [@DansterDev]. (s/f). *¡Modelar edificios con interior fácil y sencillo en blender! Explicado paso a paso! [[Object Object]]*. Youtube. Recuperado el 15 de noviembre de 2025, de <https://www.youtube.com/watch?v=Q1lillIriAs>

de Costas, A. [@AlejandrodeCostas]. (s/f-a). *Cómo exportar modelo a OBJ con texturas en Blender - Tutorial guardado en formato Wavefront OBJ 3D [[Object Object]]*. Youtube. Recuperado el 15 de noviembre de 2025, de <https://www.youtube.com/watch?v=Oes-ecXv8nc>



de Costas, A. [@AlejandrodeCostas]. (s/f-b). *TUTORIALES BLENDER - Cómo hacer un mapeado de texturas - Mapeo UV Unwrapping en español Parte 12 [[Object Object]]*. Youtube. Recuperado el 15 de noviembre de 2025, de <https://www.youtube.com/watch?v=Ek-Y62D9W4k>

Interactivas y Computación Gráfica, T. [@ArturoVMS]. (s/f). *Integración de GitHub Desktop con el Proyecto de Computación Gráfica [[Object Object]]*. Youtube. Recuperado el 15 de noviembre de 2025, de <https://www.youtube.com/watch?v=XZYgHmnB1vU>

*Metodologías de desarrollo de software: ¿qué son?* (s/f). Santander Open Academy. Recuperado el 15 de noviembre de 2025, de <https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>

Nisakai [@nisakai\_420]. (s/f). *GUÍA RÁPIDA: ¿Cómo modelar en Blender 3D? 🤓 | Tutorial de blender en español 🙌 [[Object Object]]*. Youtube. Recuperado el 15 de noviembre de 2025, de [https://www.youtube.com/watch?v=f\\_6QN\\_G3bk](https://www.youtube.com/watch?v=f_6QN_G3bk)

## Modelos Descargados

*Grand Theft Auto San Andreas - Balla - Download Free 3D model by Vesauq (@Trevoncuz2.0)*. (2025, junio 15). : <https://sketchfab.com/3d-models/grand-theft-auto-san-andreas-balla-cd393ab3131c4160a31be25a5c771ae8>

*Books and magazines - Download Free 3D model by Naira (@naira001)*. (2023, diciembre 27).: <https://sketchfab.com/3d-models/books-and-magazines-d0b76eada5bd495abcdfb2b20e6f7ee6>

Ddiaz Design (Director). (2025, enero 4). *1988 Lamborghini Countach 5000 QV - Download Free 3D model by Ddiaz Design (@ddiaz-design)*. : <https://sketchfab.com/3d-models/1988-lamborghini-countach-5000-qv-884741220ecc44a886f9232f7888a7f1>

Deharo, Y. (Director). (2018, julio 26). *20 Liquor bottles - Download Free 3D model by Yannick Deharo (@YannickDeharo)*. : <https://sketchfab.com/3d-models/20-liquor-bottles-4729b1cfa5b74db68924190242f8ac76>



---

*Dish Rack with Dishes - Download Free 3D model by HippoStance.* (2020, mayo 6). : <https://sketchfab.com/3d-models/dish-rack-with-dishes-ce69014964624fbb808a028e5c55d88b>

Elena, F. F. (Director). (2020, septiembre 3). *Saintpaulia flowers low-poly - Download Free 3D model by Elena FF (@elenafefor).* : <https://sketchfab.com/3d-models/saintpaulia-flowers-low-poly-6f78eed71b74755b3fc14f1ac5924ca>

*Mesa minimalista para TV - Download Free 3D model by Draxter\_pendragon.* (2025, julio 16). : <https://sketchfab.com/3d-models/mesa-minimalista-para-tv-badfe2c2c5a7494cac5bd052145f8f94>

*Old Television from 90's - Download Free 3D model by Zgon (@Z-gon).* (2022, febrero 22). : <https://sketchfab.com/3d-models/old-television-from-90s-5642074408e94782a7f2ee545e846b45>

*Plate and cup - Download Free 3D model by morrrtu1o.* (2023, noviembre 19).  
Serrano, V. H. F. (Director). (2020, diciembre 17). : <https://sketchfab.com/3d-models/plate-and-cup-7c2c9df4bcfb4dfbaa98836af4333f9e>

*Latas\_Coursera - Download Free 3D model by Victor Hugo Franco Serrano (@VictorHugoFrancoSerrano).* : <https://sketchfab.com/3d-models/latas-coursera-3768161dbb7b4f91907934c7fccaa7961>

Shekh, A. (Director). (2020, octubre 17). *Pine tree - Download Free 3D model by Andriy Shekh (@sheh5262).* : <https://sketchfab.com/3d-models/pine-tree-e52769d653cd4e52a4acf3041961e65>

*Sillas - Download Free 3D model by Elbolillo (@Elbolilloduro).* (2021, abril 9). : <https://sketchfab.com/3d-models/sillas-c9e85c037309413299b30dd666ab27c7>

*Sillón Le Mans - Download Free 3D model by Pablo.Portela.* (2020, agosto 13). : <https://sketchfab.com/3d-models/sillon-le-mans-79e6c35685a24da6899d884fa470df54>

*Three\_Seater\_Sofa - Download Free 3D model by Nelesh\_surve.* (2025, abril 12). : <https://sketchfab.com/3d-models/three-seater-sofa-bedd882faf354f87a28abcf93c718a3d>