

# Manual de Usuario

COMPUTACIÓN GRÁFICA E INTERACCIÓN HUMANO-COMPUTADORA

30 DE JULIO DEL 2021

SEMESTRE 2021-2

ARJONA MÉNDEZ ALBARRÁN SEBASTIÁN

RAMOS VILLASEÑOR CÉSAR MAURICIO

SANDOVAL MIRAMONTES JOAQUÍN

# ÍNDICE

---

Instalación y Ejecución.....	3
Modelos.....	7
Interacción con el entorno.....	9
Animaciones .....	9
Forma de trabajo .....	13
Diagrama de Gantt.....	15
Estimación de costos del proyecto .....	18
Evidencias de ejecución .....	18

# Instalación y Ejecución

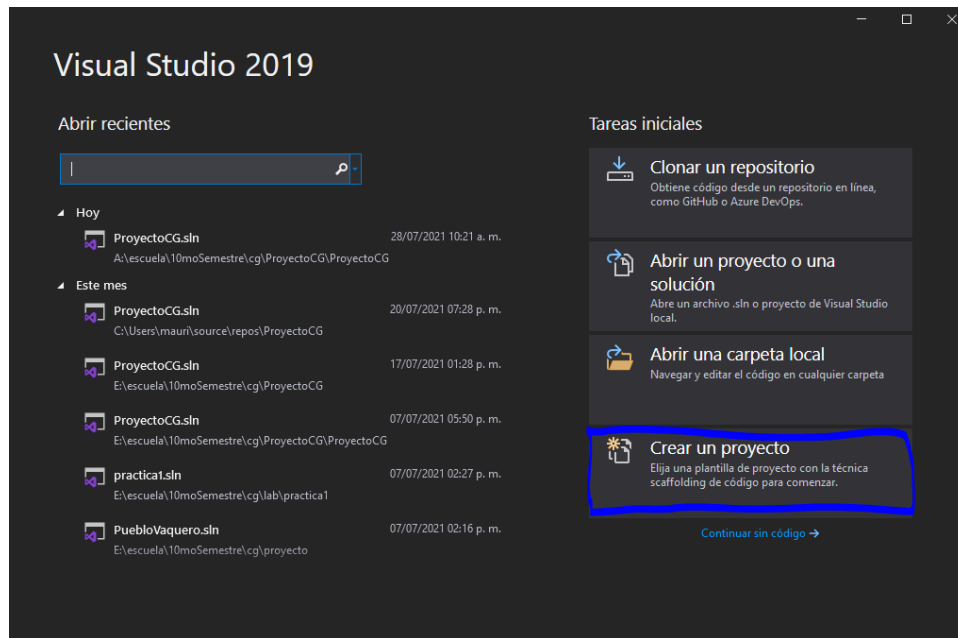
---

1. Es necesario clonar (descargar) el repositorio del proyecto en cualquier carpeta. El enlace en dónde se encuentra es el siguiente:
  - Vía HTTP: <https://github.com/Arjona99/ProyectoCG.git>
  - Vía SSH: <git@github.com:Arjona99/ProyectoCG.git>
  - Vía GitHub CLI: `gh repo clone Arjona99/ProyectoCG`
2. Dentro de la carpeta del repositorio (extraída si es el caso o por la generada por Git) se encuentra el archivo **Final.cpp**, este fichero contiene todo nuestro código fuente.
3. Para la ejecución vía **.exe** (por medio de fichero ejecutable), es necesario acceder a la carpeta del repositorio, ahí se encontrará el archivo **ProyectoCG.exe**, al hacer doble click en dicho fichero, correrá el proyecto.

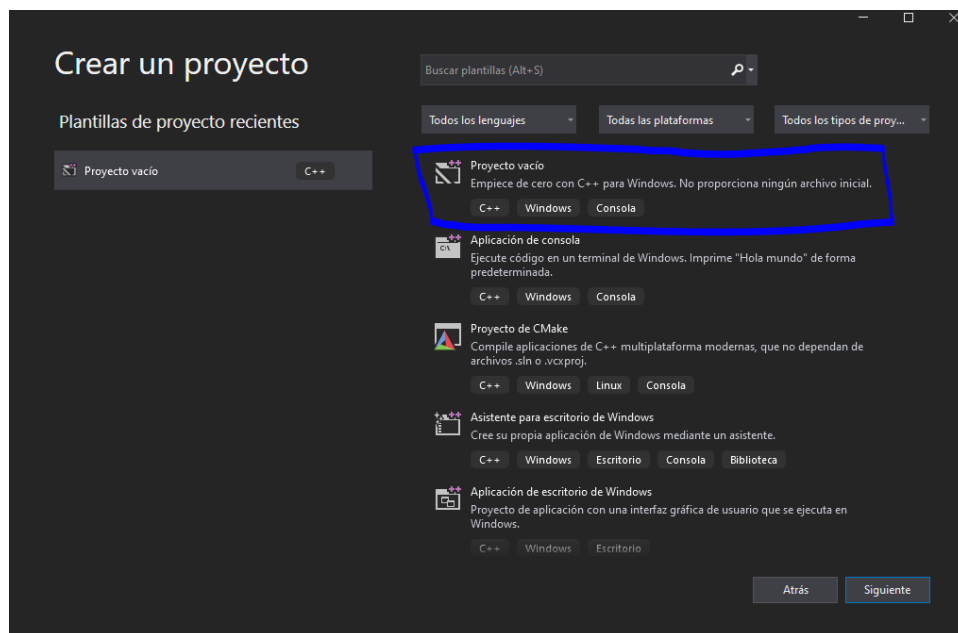
**NOTA:** Es importante no mover ningún otro archivo o cambiarlos de ruta, la estructura de archivos está definida de dicha manera para que tanto el archivo .exe como el código fuente puedan encontrar las dependencias necesarias para su ejecución.

Para la ejecución del proyecto vía Visual Studio 2019 realice lo siguiente:

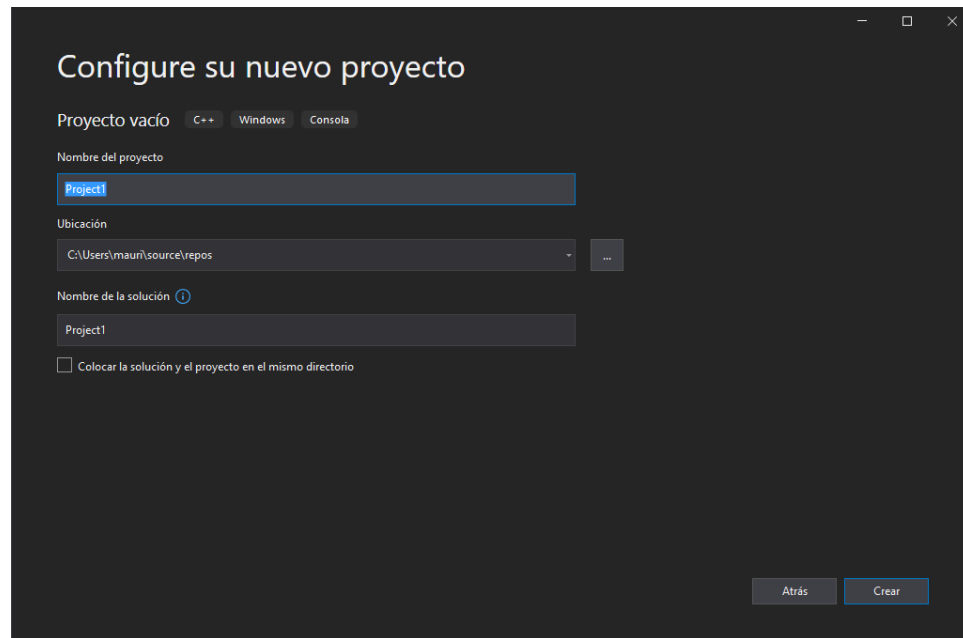
1. Cree un nuevo proyecto desde Visual Studio 2019, abra el programa y seleccione *Crear un proyecto*



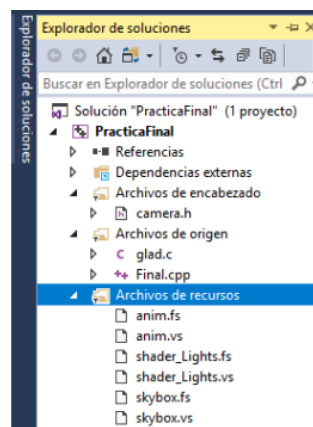
2. Escoja la opción *Proyecto Vacío* de C++, es necesario que previamente haya instalado dicha paquetería, esto se hace al ejecutar Visual Studio 2019 por primera vez



3. Decida el nombre del proyecto y la ubicación de este, asegurar que la casilla *Colocar la solución y el proyecto en el mismo directorio* se encuentre desmarcada.

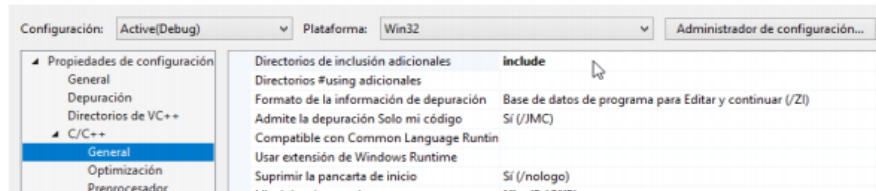


4. Copie todos los elementos de la carpeta del repositorio del proyecto dentro de la carpeta que recién se creó por Visual Studio 2019, esta carpeta se llamará justo como el nombre que se le asignó al crear el proyecto en el paso anterior y su ubicación será la ubicación dada también en el paso anterior.
5. Una vez con todos los elementos dentro de la carpeta creada por Visual Studio, será necesario ir a este programa recién abierto y dejar el Explorador de Soluciones como se indica la imagen (los *shader* no importan tanto y el archivo *camera.h* se encuentra dentro de la carpeta *include*). Además, configurar el proyecto como indican las imágenes.

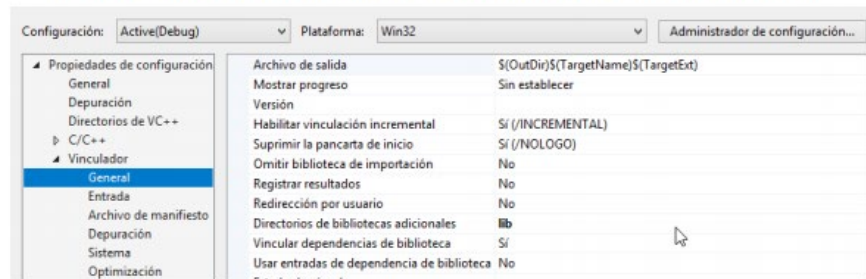


Se debe ir a las propiedades del proyecto, donde:

1. En C/C++ > General, en Directorios de inclusión adicionales se debe dejar como:

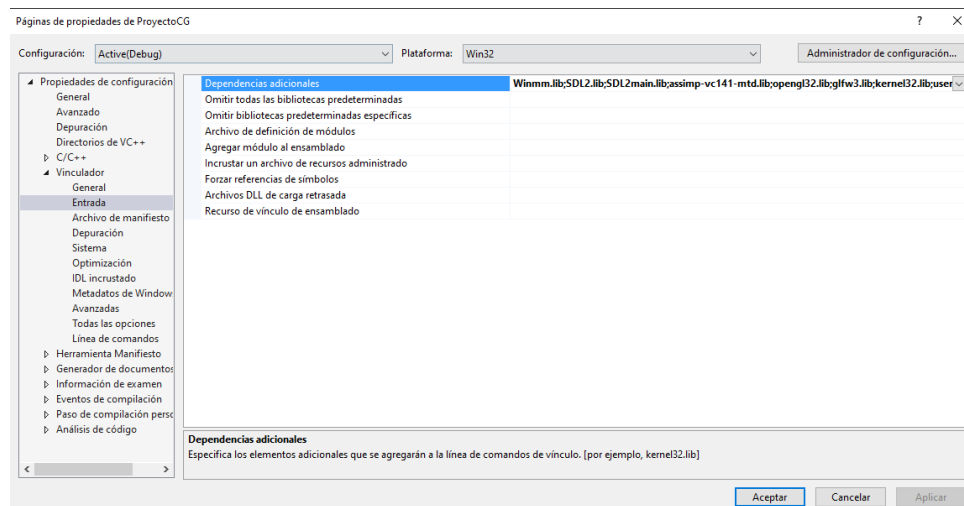


2. En Vinculador > General, Directorios de bibliotecas adicionales se debe dejar como:



6. En Vinculador > Entrada, Dependencias adicionales se debe dejar lo siguiente:

***Winmm.lib;SDL2.lib;SDL2main.lib;assimp-vc141-mtd.lib;opengl32.lib;glfw3.lib;kernel32.lib;user32.lib;gdi32.lib;winspool.lib;comdlg32.lib;advapi32.lib;shell32.lib;ole32.lib;oleaut32.lib;uuid.lib;odbc32.lib;odbccp32.lib;%AdditionalDependencies)***



7. Finalmente, ejecutar el código con el botón de la flecha verde en el panel superior, aun lado se indica con la leyenda *Depurador local de Windows*.

8. Podrían generarse errores por la versión de Visual Studio que se está utilizando (2017 en vez de 2019) con la biblioteca de audio al compilar. Si existen errores de compilación referidas a las líneas de la función **PlaySound** únicamente será necesario quitar el carácter L antes del *string* que indica el nombre del audio.

## Modelos

---

Los modelos utilizados en el este proyecto fueron obtenidos de las siguientes páginas que permiten la descarga de algunos modelos de manera gratuita e incluyen los archivos necesarios para poder importarlos a algún software de modelado (3DS Max) para su edición y adaptación según se necesite. Desarrollamos en 3Ds Max a partir de primitivas sobre un cono el diseño de los tipis indios.

- [CGTrader](#)
- [TurboSquid](#)
  - Vaca: <https://www.turbosquid.com/3d-models/cow-3d-model/1126261>
  - Toro: <https://www.turbosquid.com/3d-models/bull-3d-model-1393066>
  - Valla: <https://www.turbosquid.com/3d-models/medieval-wooden-fence-3ds-free/993091>
  - Árboles: <https://www.turbosquid.com/3d-models/3d-xfrogplants-honey-locust-gledista-triacanthos-1734005>
  - Rocas: <https://www.turbosquid.com/3d-models/3d-rock-model-1577462>
  - Molino: <https://www.turbosquid.com/3d-models/free-obj-model-windmill/674933>
  - Barril: <https://www.turbosquid.com/3d-models/free-max-model-barrel/903544>
- [Sketchfab](#)
  - Locomotora y vías: Willy Decarpentrie – Loco. <https://sketchfab.com/3d-models/loco-65b96c939dad45fd8b01798284ae670b>

- Vagón del Tren: Kefla – Train Wagon. <https://sketchfab.com/3d-models/train-wagon-cfa7a3d7f4f543ab92efb9510b8fab89>
- Bar: Yevheniia – Wild West Saloon. <https://sketchfab.com/3d-models/wild-west-saloon-7e24d622bfda479986ec6327568ddf65>
- Estación de Tren: Kefla - <https://sketchfab.com/3d-models/train-station-4d4921faa2904bd0b5b5fb6eb26e8154>
- Túnel: Merrittfx - <https://sketchfab.com/3d-models/tunnel-9e19628687a44de18c8c096e7f51d89d>
- Montañas: Rodrigo Gelmi - <https://sketchfab.com/3d-models/mountains-ae7c7d4938784b4d9f095101e90833e1>
- Minecraft Villager: Vincent Yanez - <https://sketchfab.com/3d-models/minecraft-villager-905c2547478f4bc9b4a8a1521966ab81>
- Tumbleweed: <https://sketchfab.com/3d-models/tumbleweed01-f9f2c474b4d44b868a45bf4bd00bd846>
- Granero: <https://sketchfab.com/3d-models/granja-western-texturizada-03f1cbb2229844c7a3266b5646b892cd>
- Torre de agua: <https://sketchfab.com/3d-models/free-water-tower-56fbfe448a2a41879c382444b9d1c8e1>
- Buitre: Devcells – Vulture: <https://sketchfab.com/3d-models/vulture-22996e35d9654d889726f5e7a656e876>
- [Free3D](#)

Una gran cantidad de modelos de casas y edificios fueron extraídos de este [enlace](#), ya que esta trata de un pueblo vaquero, pero muchos de los edificios no contenían texturas, así que estas fueron agregadas por nosotros.

Todos los modelos utilizados fueron gratitos y con licencia libre, así mismo para algunas texturas nos auxiliamos del sitio web [Textures](#), fueron editadas con la herramienta *Gimp* o *Adobe Photoshop*.



## Interacción con el entorno

---

Para interactuar con el ambiente virtual, se necesitará de un teclado y un mouse para utilizar las teclas y movimientos pertinentes.

- **Tecla W**      Movimiento hacia adelante
- **Tecla A**      Movimiento hacia la izquierda
- **Tecla S**      Movimiento hacia atrás
- **Tecla D**      Movimiento hacia la derecha
- **Tecla C**      Comenzar la animación por KeyFrames (vaca)
- **Tecla T**      Activa/Desactiva el movimiento del tren
- **Tecla X**      Activa la animación de la planta rodadora
- **Tecla M**      Inicia la reproducción de la música ambiente.

Para el movimiento de la cámara se acudirá al movimiento del mouse para realizar cambios de orientación, y el *scroll* del mouse para alejamiento y acercamiento de la misma.

## Animaciones

---

### Animación 1

La primera animación consta del movimiento de unos buitres en el cielo del ambiente virtual. El vuelo de los buitres consta por el movimiento de sus alas verticalmente y un desplazamiento del modelo con ayuda de traslaciones. Esta animación fue desarrollada por estados. Está activa por defecto y no es posible pararla.



Ilustración 1. Animación en ejecución.

## Animación 2

Esta animación realiza la caminata de una vaca, la cual está compuesta por distintos modelos (cuerpo y patas por separado). De esta manera, es posible manipular las patas independientemente para lograr la animación de caminata y desplazarla de un punto a otro, implementando traslaciones y rotaciones, tratando de asemejarse en lo más posible al desplazamiento de una vaca real, también se implementó sonido. Se activa con la tecla **C**, fue realizada por medio de *KeyFrames*, los cuales fueron precargados en el código, un total de 24. **NOTA:** Es necesario dejar que la animación termine antes de volver a inicializarla. Si la música está activa será necesario volver a activarla presionando la tecla **M**



Ilustración 2. Animación en ejecución.

## Animación 3

Esta animación es automática. En la animación, un aldeano caminando de un punto a otro de la aldea. Comienza desde la estación del sheriff, camina hacia el bar, gira a la izquierda y

camina hasta el fondo; después gira a la derecha y llega al final de la calle. Por último, realiza el mismo recorrido de regreso. Este recorrido se repite constantemente durante la ejecución. La animación fue hecha usando estados.



Ilustración 3. Animación en ejecución.

## Animación 4

Esta animación realiza el movimiento del tren a lo largo de las vías, esta goza de gran detalle ya que se puede ver como se mueven las llantas del tren y los ejes para simular el movimiento del tren lo más real posible. En un inicio el tren estará oculto y al presionar la tecla **T**, el tren saldrá por el túnel y se detendrá en la estación del tren. Y si se presiona nuevamente la tecla **T**, el tren continúa su recorrido hacia el túnel del lado opuesto de la superficie.



Ilustración 4. Animación en ejecución.

## Animación 5

Podemos observar en esta animación como una planta rodante con sombrero y pistola sale de su escondite para aproximarse a un enfrentamiento en la comisaría. Realiza un recorrido a lo largo de la calle frente al parque, compuesto por rotaciones y traslaciones, finalmente simula disparar hacia la comisaría. Esta animación se realizó por medio de cinco estados más un estado inicial (0). Implementa dos sonidos, una melodía clásica en un silbido para ambientar el movimiento de la planta, y sonidos de detonación de una pistola. Se inicializa con la tecla **X**. **NOTA:** Si la música está activa será necesario volver a activarla presionando la tecla **M**.



Ilustración 5. Animación en ejecución.



Ilustración 6. Animación en ejecución.

## Forma de trabajo



Para trabajar de manera eficiente y síncrona, se decidió utilizar Git como sistema de control de versiones, y GitHub como almacenamiento remoto del repositorio. En un principio se creó la rama **trunk**, en la cual se incluyó el archivo **README.md** y la configuración del archivo **.gitignore** para evitar hacer *commit* de archivos que no necesitan estar en el repositorio, entre ellos los generados por *VSCode*. Esto facilitó bastante el trabajo entre *commits*, pero dificultó solamente un poco la instalación.

Posteriormente, para lograr el trabajo síncrono, se realizó la repartición del trabajo: cada integrante se encargó de una rama propia para añadir los modelos que se necesitaban de su zona (se dividió el escenario en tres zonas, una para cada integrante).

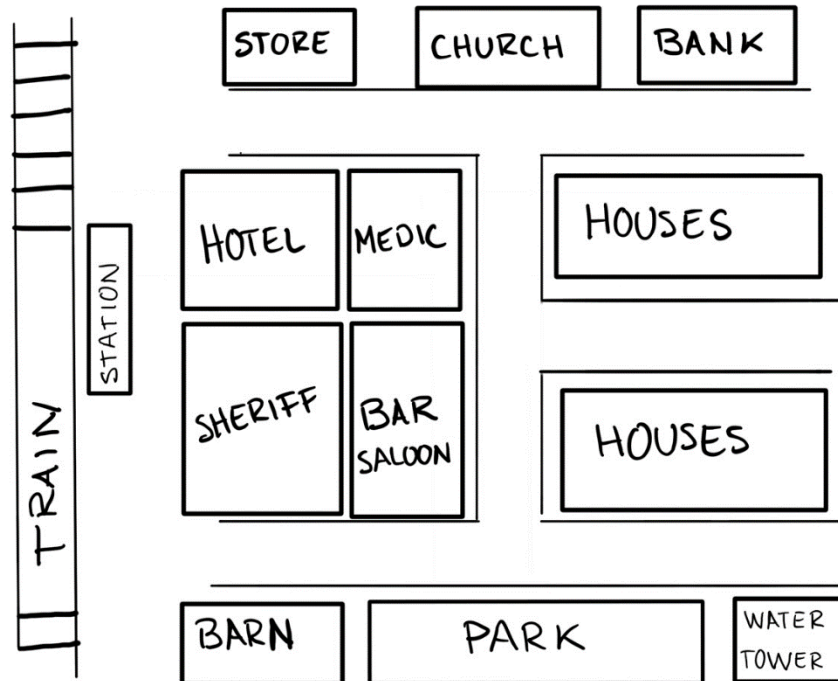


Ilustración 7. Boceto de nuestro escenario.

Una vez que los modelos estuvieran listos, se creaba un *Pull Request* de la rama para que los demás integrantes pudieran revisar el trabajo del responsable de la rama y aprobar los cambios, finalmente era posible hacer un *merge* a la rama **trunk** sin problema alguno.

```
mauri@DESKTOP-5K9LJ5J MINGW64 /a/escuela/10moSemestre/cg/ProyectoCG/ProyectoCG/ProyectoCG (ball-animation)
$ git log --graph --oneline --decorate
* fb4ef91 (HEAD -> ball-animation, origin/ball-animation) Animación de tumbleweed disparando
* c305bc3 (origin/trunk, origin/HEAD, trunk) Merge pull request #8 from Arjona99/cowboy-cow-animations
|
| * ab99a50 (cowboy-cow-animations) Animación de la vaca e implementación de audio
| * aa09939 Merge branch 'trunk' of https://github.com/Arjona99/ProyectoCG into trunk
|
| * a6824f1 Merge branch 'trunk' of https://github.com/Arjona99/ProyectoCG into trunk
|
| * 80472ad Reestauracion de casas y gallow
| * ed2b074 Ajuste de posición y texturas de Montañas
|
| * 5770c21 Merge pull request #7 from Arjona99/park
|
| * f9b91a6 (origin/park, park) Merge branch 'trunk' into park
|
| * d5193fe Merge pull request #5 from Arjona99/downtown
|
| * c02b110 Sincronización
|
| * 1001dff Tunel y montañas
| * afef49a Se agregó la enfermería.
| * 836a8ad Ajustar modelos y agregar Estación de tren y vías.
|
| * 4f7a5ac Added church, bank, store. Fix floor
| * 17c2ae3 Se agregaron 3 edificios nuevos
|
| * dcd6844 Merge de trunk
```

Ilustración 8. Evidencia de trabajo en Git.

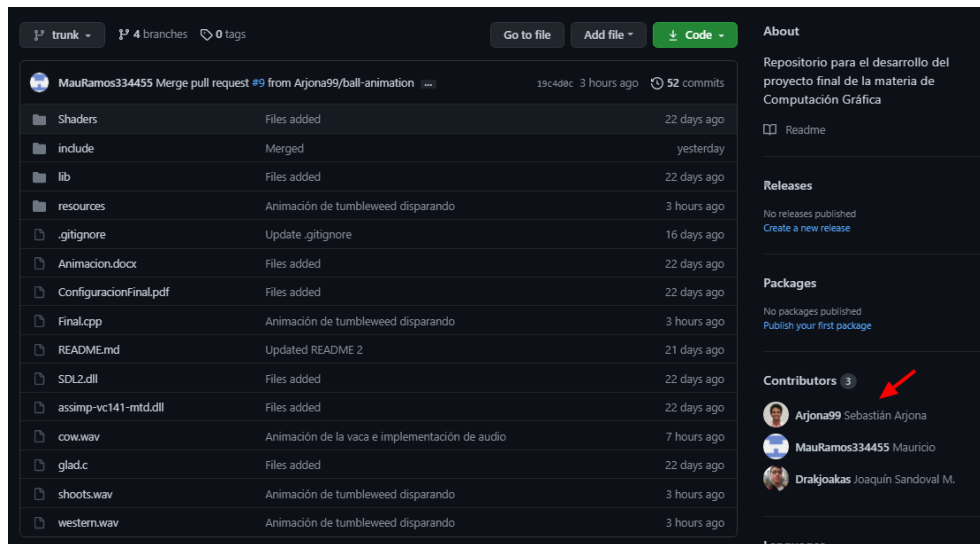


Ilustración 9. Evidencia de trabajo en GitHub.

Enlace del repositorio en GitHub: [Pueblo Vaquero](#)

## Diagrama de Gantt

Previo al comienzo del proyecto se realizó un diagrama de Gantt para hacer una estimación del tiempo que tomaría cada actividad. Como era de esperarse, los tiempos establecidos en un inicio no se cumplieron, así que también se mostrará un diagrama de Gantt con los tiempos reales.

# Diagrama de Gantt

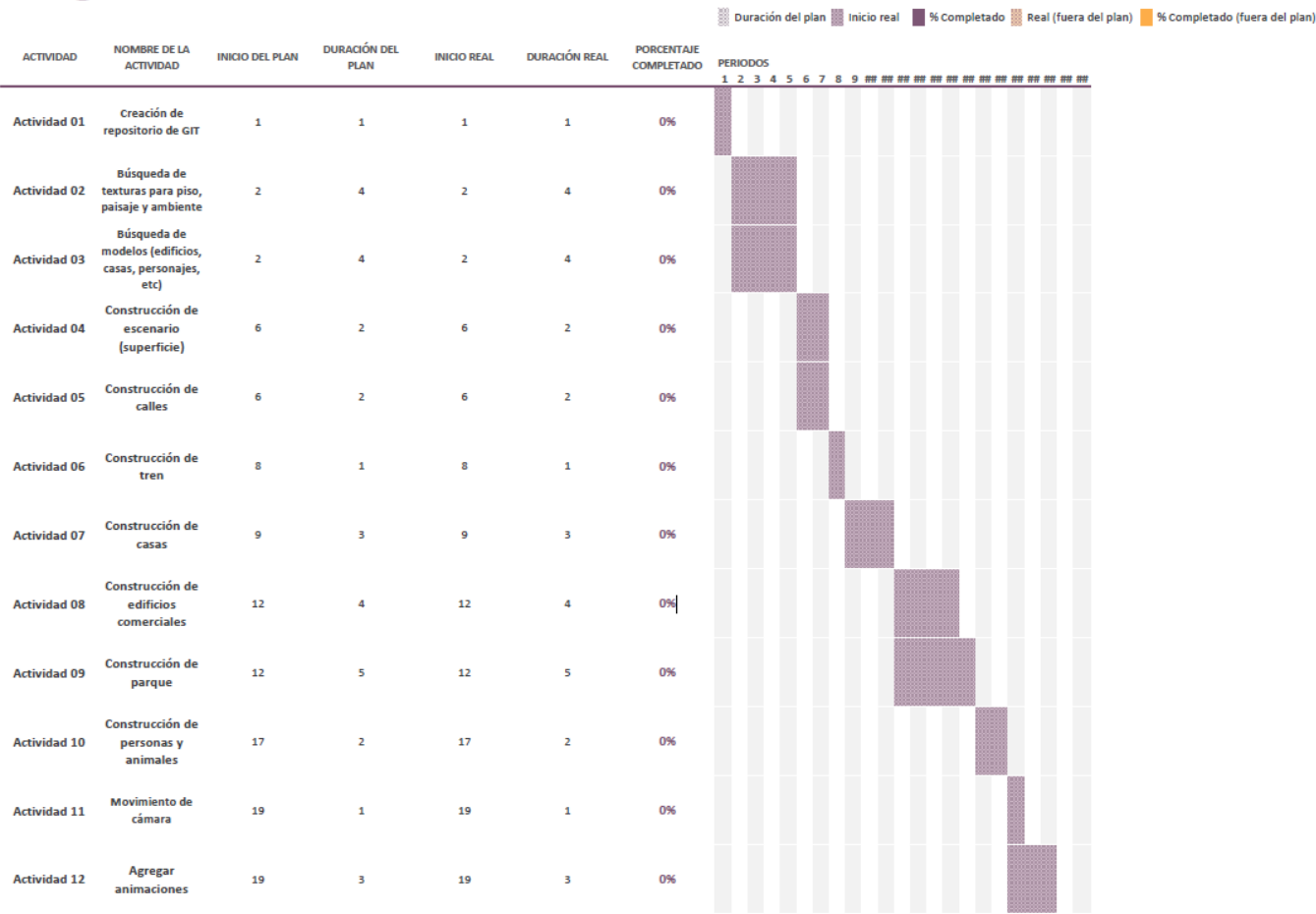


Ilustración 10. Estimación inicial



# Diagrama de Gantt

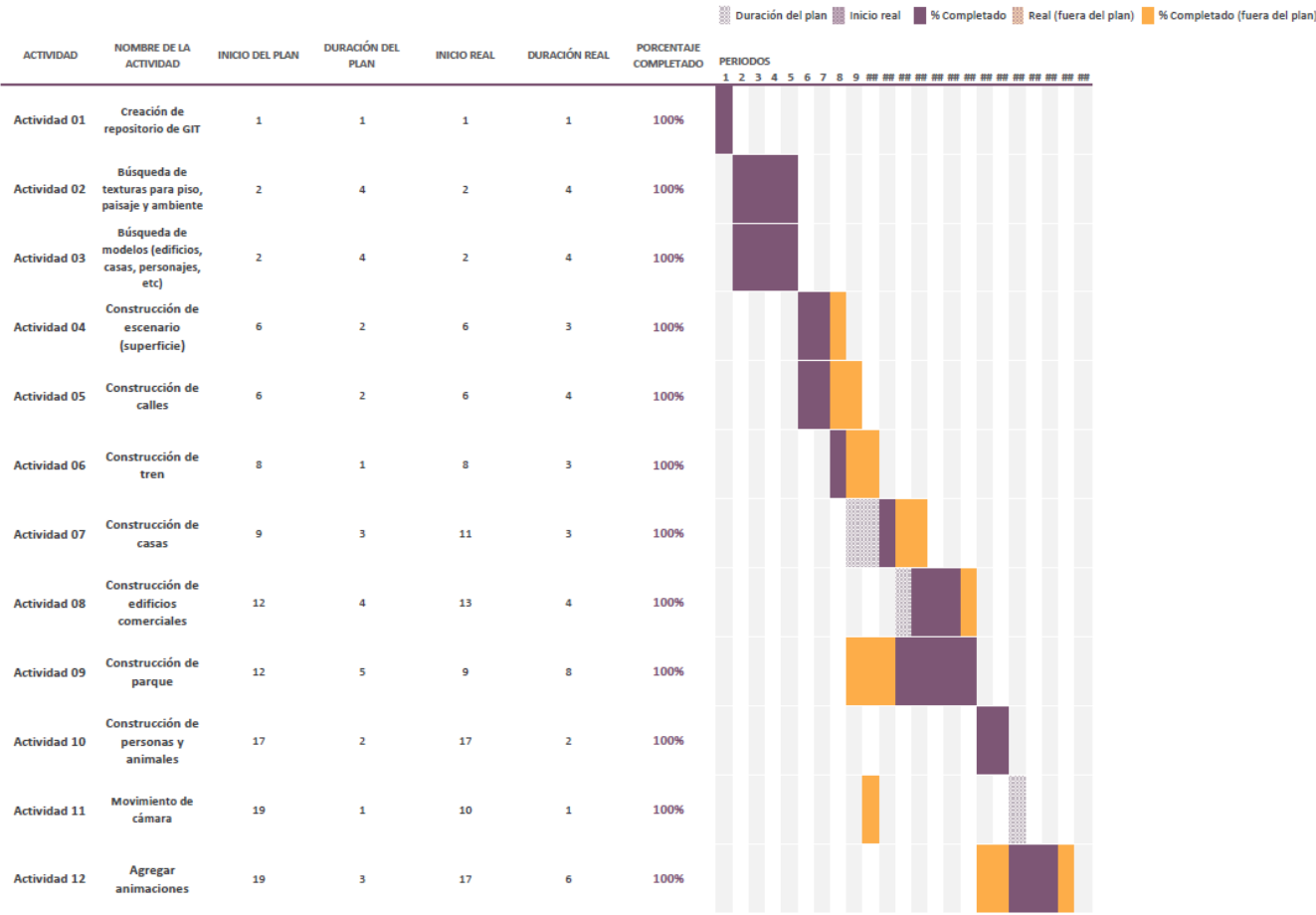


Ilustración 11. Avance real

## Estimación de costos del proyecto

---

CONCEPTO	COSTO
Programa 3DS MAX	\$2,813 mensual
Photoshop	\$449 mensual
Internet	\$500 mensual
Luz eléctrica	\$240 mensual
Honorarios	\$42,000 mensual (3 programadores)
<b>TOTAL</b>	<b>\$46,002.00</b>

## Evidencias de ejecución

---

Enlace al video demo del pueblo vaquero [aquí](#).