  


**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**Asignatura:** Computación Gráfica e Interacción Humano Computadora

**Grupo:** 5

**Semestre:** 2022-2

**Github como herramienta de trabajo colaborativo**

**Fecha Límite de Entrega:** 26/05/2022

**Profesor: José Roque Román Guadarrama**

**Alumnos:**

* **Colin Santos Luis Froylan**
* **Najera Noyola Karla Andrea**

# ¿Qué es el control de versiones?

El control de versiones es un sistema que permite rastrear y gestionar los cambios realizados en un archivo o conjunto de archivos. Se utiliza principalmente por programadores con el fin de tener un seguimiento detallado de las modificaciones realizadas en un código fuente, incluso permitiéndole analizar cambios y revertirlos en caso de ser necesario.

Hoy en día existen muchas herramientas para realizar el control de versiones de un proyecto; no obstante, el que tiene una mayor popularidad entre las comunidades de programación es GitHub.

# ¿Qué es GitHub?

Github es una herramienta de servicios basados en la nube que permiten almacenar un sistema de control de versiones (VCS) el cual recibe el nombre de Git, lo que permite a los desarrolladores trabajar en conjunto incluso a la distancia para mejorar el flujo de trabajo y la colaboración. Este servicio permite a los desarrolladores participar y realizar cambios en proyectos compartidos, a la par de tener un seguimiento completo de los avances de este.

Actualmente, GitHub aloja más de 100 millones de repositorios, principalmente de código abierto y pese a que su uso más popular es a través de la consola de comandos, existe una interfaz gráfica en la que usuarios inexpertos pueden usar la herramienta con facilidad.

# ¿Por qué decidimos usar GitHub?

Principalmente, en este proyecto se utilizó GitHub como sistema de almacenamiento y repositorio en línea debido a que nos permitió tener un trabajo en equipo bastante ágil a través de los comandos git commit, git push y git pull, que permitieron un gran nivel de sincronización en nuestro proyecto; cuando un compañero terminaba su parte, podía subirlo de manera inmediata al repositorio con los comandos git add, git commit y git push y el otro compañero lo podía descargar con el comando git pull. Esto nos ahorró mucho tiempo de transferencia de archivos, y también fue de mucha ayuda que se puede colocar un mensaje en GitHub cada que se sube un cambio.

**Captura de pantalla de un celular

Descripción generada automáticamente**

Prueba de uso de GitHub, con mensajes para los cambios y con muchos commits. También indica el momento en que fue subido/modificado cada archivo o carpeta.

Así mismo, si algo fallaba dentro de nuestro programa, o no queríamos incorporarlo, podíamos restablecerlo a la última versión subida en GitHub.

# Uso desde la interfaz gráfica de Visual Studio Code

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamenteDesde que comenzamos con la realización de este proyecto descubrimos que Visual Studio Code nos permitía usar Git de manera gráfica para controlar el repositorio de trabajo. Por ello, optamos por usarlo la mayor parte del tiempo desde la IDE, con el fin de facilitar cada uno de los commit.

En primer lugar, uno de los integrantes del equipo creo el proyecto siguiendo cada uno de los pasos indicados por el manual de configuración y al otro le basto con clonar el repositorio a partir de Visual Studio Code, lo que copiaba archivos e incluso la configuración de bibliotecas y elementos del proyecto.

Opciones de Visual Studio Code, aunque las que usamos fueron “Crear un proyecto” y “Clonar un repositorio”

Cabe destacar que ambos integrantes contábamos con un perfil de GitHub previamente creado y configurado en nuestro equipo y que la persona que creo el proyecto subió los archivos a su repositorio de manera pública.

Captura de pantalla de un celular

Descripción generada automáticamente

Repositorio del proyecto en el perfil del usuario suadero99 (Luis Froylan Colin santos)

Asimismo, fue necesario que el usuario propietario del proyecto colocara a la otra persona como colaboradora, de manera que ambas personas pudieran participar colaborativamente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Colaboradores del repositorio de GitHub

Contando con la configuración de Git para ambos usuarios, Visual Studio Code nos desplegaba una serie de opciones con las cuales podíamos hacer pull, commit y/o hacer push sobre los cambios del proyecto, con el fin de estar al corriente con los cambios que hacíamos entre ambos a los elementos del proyecto.

Captura de pantalla de un celular

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Opciones que presenta Visual Studio Code para el trabajo colaborativo con Git. Las que más usamos fueron “Pull”, “Enviar cambios” y “Sincronizar ramas”

Cabe destacar que, si bien todos los cambios hechos fueron realizados usando GitHub, entre ambas partes del equipo existió una comunicación constante con el fin de garantizar el pleno funcionamiento del proyecto. Esta comunicación fue por distintos medios, ya que involucro en muchas ocasiones trabajar los 2 en el mismo lugar o hacer algunas llamadas de Meet e incluso mensajes por redes sociales.

# Evidencias de uso de GitHub

A continuación, se presentan algunas capturas de pantalla que sirven como prueba del uso de GitHub (no son todos los commits realizados):

Captura de pantalla de computadora

Descripción generada automáticamente Captura de pantalla de computadora

Descripción generada automáticamente Captura de pantalla de computadora

Descripción generada automáticamente

Pruebas de commits a lo largo del desarrollo del proyecto

# Inconvenientes que tuvimos al usar GitHub

Afortunadamente, el uso de esta herramienta es muy sencillo, y nos facilitó mucho el desarrollo del proyecto, aunque también hubo puntos malos con esta herramienta. El principal problema fue que, cuando se compartío el repositorio en GitHub se creo con este un archivo .gitignore (archivo con la información a ignorar para subir a un repositorio) que no subía archivos .obj y la carpeta de recursos en su totalidad al repositorio, lo que ocasionó un problema en un principio al momento de sincronizar nuestros proyectos, pues un compañero no podía ver ni cargar los modelos. Nos dimos cuenta de que posiblemente pudo ser el archivo .gitignore y así fue, por lo que solo hubo que borrar del .gitignore los archivos .obj y borrar también la carpeta resources del listado de carpetas a ignorar y el problema fue solucionado.

Asimismo, tuvimos inconvenientes con la entrega del proyecto al profesor de laboratorio debido a que en su computadora no se cargaba adecuadamente nuestra iluminación. Tras hacer varias pruebas con compañeros con distintas computadoras descubrimos que este problema afectaba solo a aquellas personas con procesador Intel y, dado que ambos del equipo tenemos Ryzen 7, jamás nos percatamos de este defecto. Por ello, pese a que todo nuestro trabajo está realizado en GitHub, decidimos usar Google Drive para la entrega final con el fin de resolver problemas de compatibilidad entre procesadores AMD Ryzen e Intel.

# Enlace al repositorio del proyecto

<https://github.com/suadero99/ProyectoCG.git>

# Referencias

B., G. (2021, 8 marzo). *¿Qué es GitHub y Cómo Usarlo?* Tutoriales Hostinger. https://www.hostinger.mx/tutoriales/que-es-github#%C2%BFQue\_es\_GitHub\_y\_por\_que\_es\_tan\_popular