



Universidad Nacional Autónoma de México.



Facultad de Ingeniería.

Ingeniería en Computación.

Laboratorio de Computación Gráfica e Interacción Humano Computadora.

Alumnos:

- Brito Segura Ángel.
- Hernández Torres Agustín de Jesús.
- Huarte Nolasco Mario.

Documento: Experiencia utilizando GitHub.

Grupo de Teoría: 03.

Fecha límite de entrega: 25 / julio / 2021.

Semestre: 2021-2.

Un sistema de control de versiones como Github nos ayudó a almacenar los cambios de nuestro proyecto, de los múltiples archivos que lo conforman, a lo largo del tiempo. Esto nos permitió tener un control sobre las etapas del desarrollo de software y como no dependemos de un servidor central cada uno de nosotros teníamos una copia del repositorio en cada una de nuestras computadoras.

El primer paso que tuvimos que hacer fue instalar GIT, que es nuestro sistema de control de versiones distribuido que utilizamos. Para descargarlo accedimos desde su página oficial:

<http://git-scm.com/download/win>

Se eligió la opción de **Git for Windows Setup** correspondiente a la máquina de cada uno de nosotros y seguimos los pasos de instalación. Finalmente para verificar que la instalación fue exitosa y qué versión de GIT está instalada escribimos el comando:

```
$ git --version
```

Se recomienda usar el acceso en el explorador de Windows: *Clic derecho sobre una carpeta >> Git Bash here*

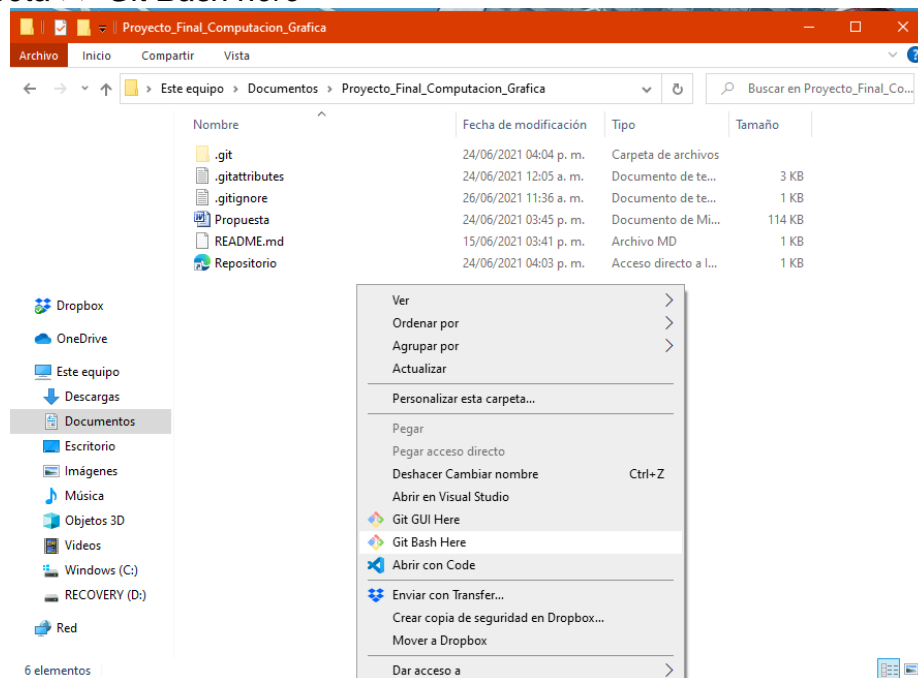


Figura 1: Explorador de archivos de Windows.

Lo primero que tuvimos que hacer después de la instalación fue configurar nuestro perfil de Git con nuestro nombre y correo asociado a una cuenta de Github (servidor en línea de repositorios remotos), es importante porque cada cambio que confirmamos estará asociado a esta información, ocupando los siguientes comandos:

```
$ git config --global user.name "nombre_usuario"
```

```
$ git config --global user.email correo@ejemplo.com
```

Una vez configurado, se clona el repositorio remoto en nuestras carpetas locales ocupando el siguiente comando:

**\$ git clone**

[https://github.com/angelbrittoFI/Proyecto\\_Final\\_Computacion\\_Grafica.git](https://github.com/angelbrittoFI/Proyecto_Final_Computacion_Grafica.git)

Nos solicitará el nombre de usuario y contraseña de Github para poder realizar esta acción. Esto creará un directorio del repositorio, donde hayamos ejecutado la orden anterior. Dentro de ese directorio se encontrará, además del directorio `.git` con toda la información del repositorio, una copia de trabajo de la última versión del proyecto que será nuestro repositorio local.

### **Subir archivos o modificaciones al repositorio**

Como buena práctica en Github, debido a que se trabaja de manera colaborativa, antes de realizar cualquier modificación en el repositorio local, se ocupa el siguiente comando para traer la última versión almacenada en la nube del proyecto:

**\$ git pull origin master**

Una vez que hemos hecho modificaciones en el repositorio local, procedemos a ejecutar el siguiente comando para conocer los comandos que se deben de realizar para que el repositorio remoto sea igual al local:

**\$ git status**

Para subir archivos al repositorio remoto, primero se debe de dar seguimiento a los archivos por lo que podemos hacerlo de dos maneras distintas:

**\$ git add nombre\_archivo.extension**

El comando anterior agrega un solo archivo dando su nombre y extensión.

**\$ git add .**

El comando anterior agrega todos los archivos modificados en la carpeta local que son diferentes a la última versión del proyecto.

Todos los cambios que se vayan realizando en nuestro proyecto deben ser identificados por un comentario. Añadimos el comentario mediante el siguiente comando:

**\$ git commit -m "Comentario del cambio"**

Finalmente, subimos los cambios al repositorio remoto tecleando en la terminal:

**\$ git push origin master**

Siempre que se quiera hacer algún cambio en el repositorio, se deben de seguir estos pasos.

Si por alguna razón los cambios realizados de manera local se quieren revertir a como estaban en el repositorio remoto, se ocupa el siguiente comando:

**\$ git restore .**

## Historial de cambios:

Para poder observar los cambios, a lo largo del tiempo, hechos en el repositorio remoto se puede acceder directamente al link del mismo:

[https://github.com/angelbritofI/Proyecto\\_Final\\_Computacion\\_Grafica](https://github.com/angelbritofI/Proyecto_Final_Computacion_Grafica)

Se da clic en la opción de *commits*:

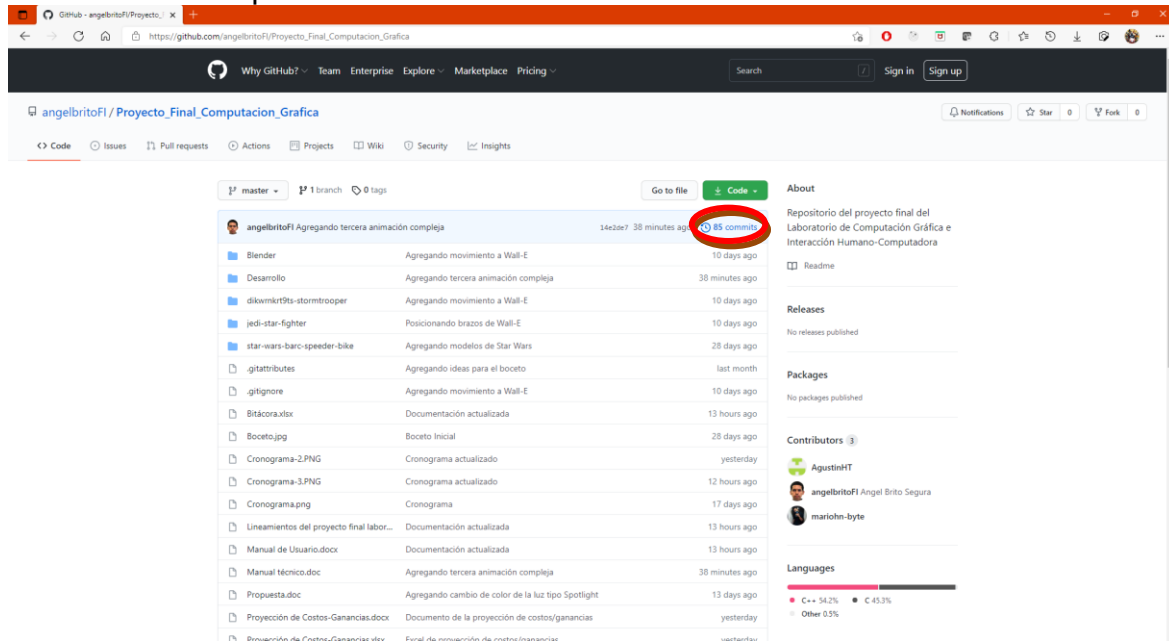


Figura 2: Página de Github.

Nos muestra el historial de todos los cambios hechos en el repositorio, con su fecha y desarrollador que realizó dicho cambio:

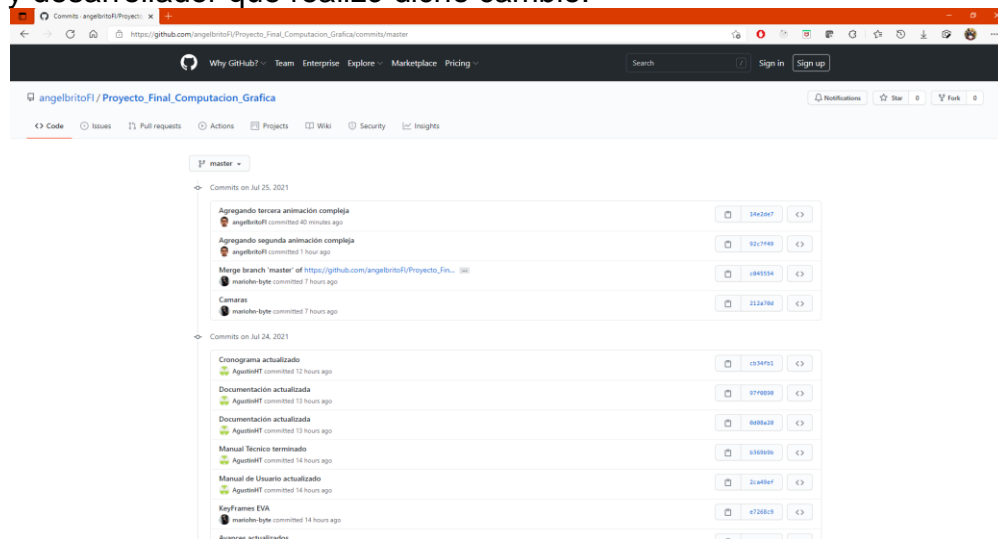


Figura 3: Historial de commits.

O bien, si se tiene clonado el repositorio, se ocupa el siguiente comando:

**\$ git log**

Este comando nos muestra el número de *commit*, el autor del cambio con su nombre de usuario y correo registrado en Github y fecha en que se realizó dicho cambio. Para mostrar más información se da **Enter** y si ya no se requiere más información se introduce la tecla **Q**.

### **Ignorar archivos:**

Para un mejor control del sistema de almacenamiento y evitar subir archivos innecesarios al repositorio remoto como archivos de configuración locales, se utilizó el archivo *.gitignore*. En este archivo se van poniendo la extensión o nombre del archivo a ignorar, incluso podemos ignorar carpetas completas para que en cada cambio del repositorio GIT no siga estos archivos plasmados en *.gitignore* para no ser controlados por el sistema.