





Herramienta de Desarrollo Colaborativa y Sistema de Almacenamiento: GitHub

Lázaro Martínez Annette Ariadna No. de cuenta: 316129189

Resumen

Este documento tiene la intención de comentar mi experiencia con el uso de GitHub como herramienta de desarrollo colaborativa y sistema de almacenamiento para el proyecto final de esta materia.

1. Sobre GitHub

GitHub es una herramienta basada en Git, el cual es un software de control de versiones diseñado por Linus Torvalds. GitHub es una compañía que ofrece un servicio de hosting de repositorios almacenados en la nube principalmente para desarrolladores de software. Cabe mencionar que GitHub tiene su versión de escritorio.

2. Experiencia Previa

Antes de utilizar GitHub para este proyecto, únicamente lo había utilizado para otro proyecto durante la carrera, el cual elaboramos en equipo para la materia de Estructura y Programación de Computadoras, por lo cual nos sirvió como herramienta de control de versiones y como herramienta colaborativa, sin embargo, la mayoría de los commits los realizamos desde la página web de GitHub.

De la misma forma, utilicé GitLab durante un proyecto de servidores en mi programa de becarios, cuando montamos un servidor DNS desde una máquina virtual en Amazon Web Services (AWS) que contaba con una distribución Linux como sistema operativo. En este caso, también se tomaba en cuenta utilizar la herramienta como herramienta colaborativa más que un control de versiones, pues únicamente lo utilizamos para la documentación, sin embargo, debido a la división de actividades, fui la principal escritora en el repositorio.

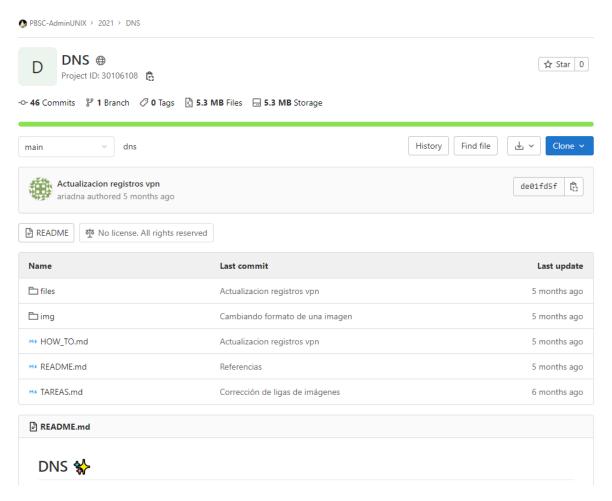


Figura 1: Proyecto en Gitlab

Puedo decir con seguridad que este repositorio en GitLab fue el que más me capacitó para herramientas de este tipo, principalmente debido a que todos los cambios se realizaron desde la máquina virtual, por lo que necesitábamos realizar todas las acciones (configuración, escrituras, modificaciones, etc.) desde la terminal y para esto fue necesario conocer los editores de texto de Linux, así como los comandos de git para la terminal.

3. Experiencia en este proyecto

Los primeros cambios del repositorio de este proyecto fueron principalmente para almacenar los modelos que elaboré con MagicaVoxel, por lo que los cambios los realicé en un principio desde la página web de GitHub, sin embargo, cuando empecé la creación del escenario fue necesario agregar varios archivos al repositorio, por lo que decidí empezar a utilizar la terminal de Windows para manejar los cambios.

En un principio creí que sería confuso porque solo había utilizado la herramienta desde Linux, sin embargo, los comandos son los mismos y el editor de texto de 'cmd' es muy similar al editor 'vi' de Linux, por lo que no tardé en comprender los aspectos de GitHub.

```
Edición README.md

Se elaboró una pequeña introducción del proyecto en el archivo README.md ¿debería ponerla en español e inglés?
# Please enter the commit message for your changes. Lines starting # with '#' will be ignored, and an empty message aborts the commit.
# On branch main
# Your branch is up to date with 'origin/main'.
# Changes to be committed:
# modified: README.md
# new file: img/portada.png
# Untracked files:
# .vs/

C:/Users/z3839/Documents/Gotham/.git/COMMIT_EDITMSG[+] [unix] (23:47 16/04/2022)
-- INSERT --
```

Figura 2: Editor en cmd

Además, esto me facilitó mucho los cambios, pues pude subir los archivos con mayor rapidez que en la versión en línea, además de que se facilitan los cambios y eliminaciones.

Finalmente, puedo mencionar que, debido a que el proyecto lo realicé sola, no fue necesario usar diferentes ramas del repositorio y la mayoría de los cambios los hice directo sobre la rama principal cuando me aseguraba de contar con una versión funcional.

Figura 3: Git push

Además, durante la realización del proyecto, adjunté por error un archivo con extensión .vs por actualizar los archivos por medio de expresiones regulares (una muy mala técnica para 'ahorrar tiempo'), el cual no solo no debía incluirse en el repositorio, sino que también era en exceso pesado, por lo que ocurrió un error al realizar el git push, pero ya había realizado el git commit, por lo que no podía eliminar el archivo del commit.

Para resolver este inconveniente, tuve que buscar en mis apuntes de git el comando para observar el código o identificador numérico de las versiones existentes en mi repositorio con el comando 'git log –oneline', desde el cual puede realizar un reset y obtener nuevamente la versión anterior del repositorio. Claramente, tuve que realizar nuevamente los cambios de la versión que no pude subir a GitHub, sin embargo, realicé un respaldo local de dichos archivos antes de aplicar el reset, por lo que no hubo mayor inconveniente y pude proseguir con el proyecto.

```
Símbolo del sistema - git log --oneline
                                                                                                                                                                                                                                   (HEAD -> main, origin/main, origin/HEAD) Animaciones iniciales
     1bc Musica
 441f76 Luces iniciales
 dafa47 Texturas con luz
 ea90ab Semaforos
 cadb6 Camaras
003a73 README.md modified
06ea54 Edición README.md
  a1759 Modificacion skybox
36a67 Optimizacion de modelos
    16a67 Optimizacion de modelos

P8cc2 Carga modelos pt3

17fff Carga de modelos pt 2.

1384 Carga de Modelos inicial

1266e Archivos adicionales p2

16c9 Archivos y librerías adicionales p1

16c13d Archivos iniciales de proyecto y avatar
     pef2 Texturas
       b3a Person models
     f095 Modificación de archivos vox
8342 Modificación de archivos obj
              Street elements
     149b Street elements
192f Adding transport models
192f Adding models
196a modificacion de carpetas
1987c Adition of models
     94e2 Add models
     e940 Modelos iniciales
13f4 Initial commit
```

Figura 4: Git log

4. Evidencias

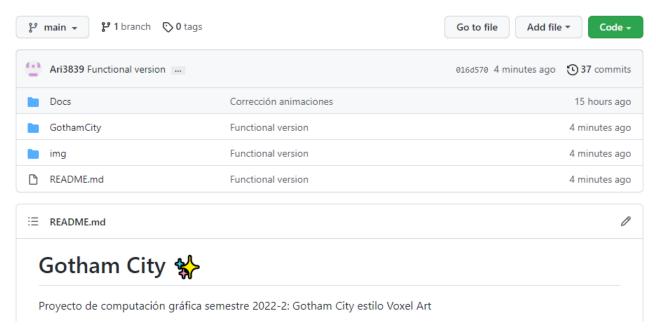


Figura 5: Repositorio

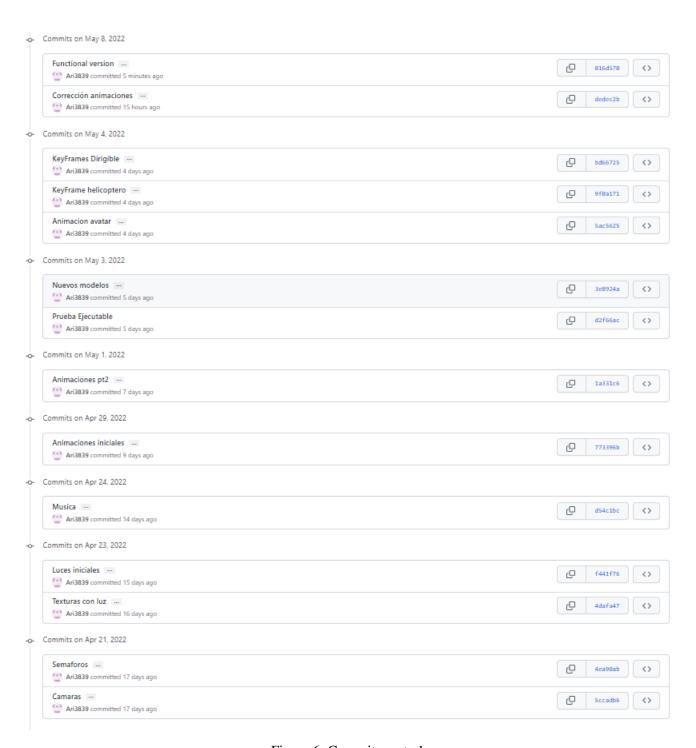


Figura 6: Commits parte 1

-o- Commits on Apr 16, 2022 README.md modified c003a73 $\langle \rangle$ Ari3839 committed 22 days ago Edición README.md ... g. <> Ari3839 committed 22 days ago Modificacion skybox ... Ari3839 committed 22 days ago Optimizacion de modelos ---3636a67 $\langle \cdot \rangle$ Ari3839 committed 22 days ago Commits on Apr 15, 2022 Carga modelos pt3 --f7b8cc2 (> Ari3839 committed 24 days ago -o- Commits on Apr 14, 2022 Carga de modelos pt 2. ... df717ff Ari3839 committed 24 days ago Carga de Modelos inicial 07b1384 Ari3839 committed 24 days ago -o- Commits on Apr 8, 2022 Archivos adicionales p2 475266e Ari3839 committed on 8 Apr Archivos y librerías adicionales p1 g. Ari3839 committed on 8 Apr Archivos iniciales de proyecto y avatar ---Verified g. 396e13d <> Texturas ... Verified Q 797bef2 $\langle \rangle$ Ari3839 committed on 8 Apr -o- Commits on Mar 31, 2022 Person models ---Verified 9 56b2b3a (> Ari3839 committed on 31 Mar -o- Commits on Mar 30, 2022 Modificación de archivos vox Q <> Verified Ari3839 committed on 30 Mar Modificación de archivos obj ... Verified g. d128342 <> Ari3839 committed on 30 Mar

Figura 7: Commits parte 2

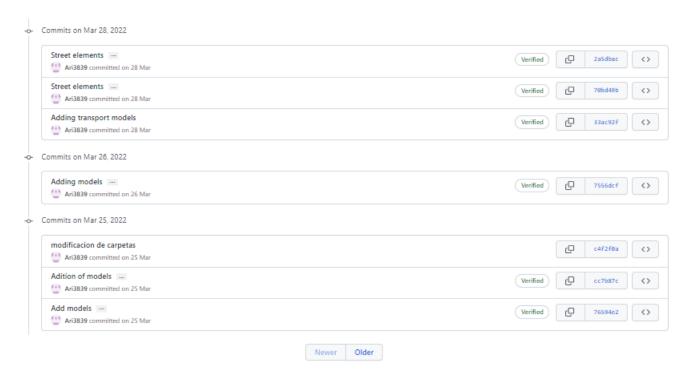


Figura 8: Commits parte 3

Referencias

- [1] colaboradores de Wikipedia. (2022a, marzo 15). Git. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Git#GitHub-Flow
- [2] colaboradores de Wikipedia. (2022b, marzo 17). GitHub. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/GitHub
- [3] Kinsta. (2020, 8 octubre). ¿Qué es GitHub? Una Guía para Principiantes sobre GitHub https://kinsta.com/es/base-de-conocimiento/que-es-github/
- [4] Hernández Bermúdez, A. L. (2021). Control de versiones con git AdminLinux PBSI UNAM-CERT. pbsc-adminunix. https://pbsc-adminunix.gitlab.io/git/