Universidad Nacional Autónoma de México
Professor: Ing. Jose Roque Roman Guadarrama
Computer Graphics and Human-Computer Interaction

[Engineering Faculty]
Deadline: 25/05/2022
Group 3. Semester 2022-2

Collaborative Development Tool and Storage System: GitHub

Lázaro Martínez Annette Ariadna
Student ID: 316129189

Abstract

This document has the intention of expressing my experience with the use of GitHub as a collaborative development tool and storage system during the final project of this subject.

1 About GitHub

GitHub is a tool based on Git, which is a version control software designed by Linus Torvalds. GitHub is a company that offers repository hosting services, primarily with cloud technology, for software developers. It is worth mentioning that GitHub has its own desktop version.

2 Previous Experience

Before using GitHub for this project, I had already used it for another project during the degree, but it was only the one we developed as a team for the Computer Structure and Programming subject. It served us as a version control tool and as a collaborative tool, however, most commits were made from the GitHub web page

Similarly, I used GitLab during a server project in my internship program, when we set up a DNS server from a virtual machine on Amazon Web Services (AWS) running a Linux distribution as operating system. In this case, we used the tool as a collaborative tool rather than a version control tool since we only used it for documentation, however, due to the division of activities, I was the main writer in the repository.

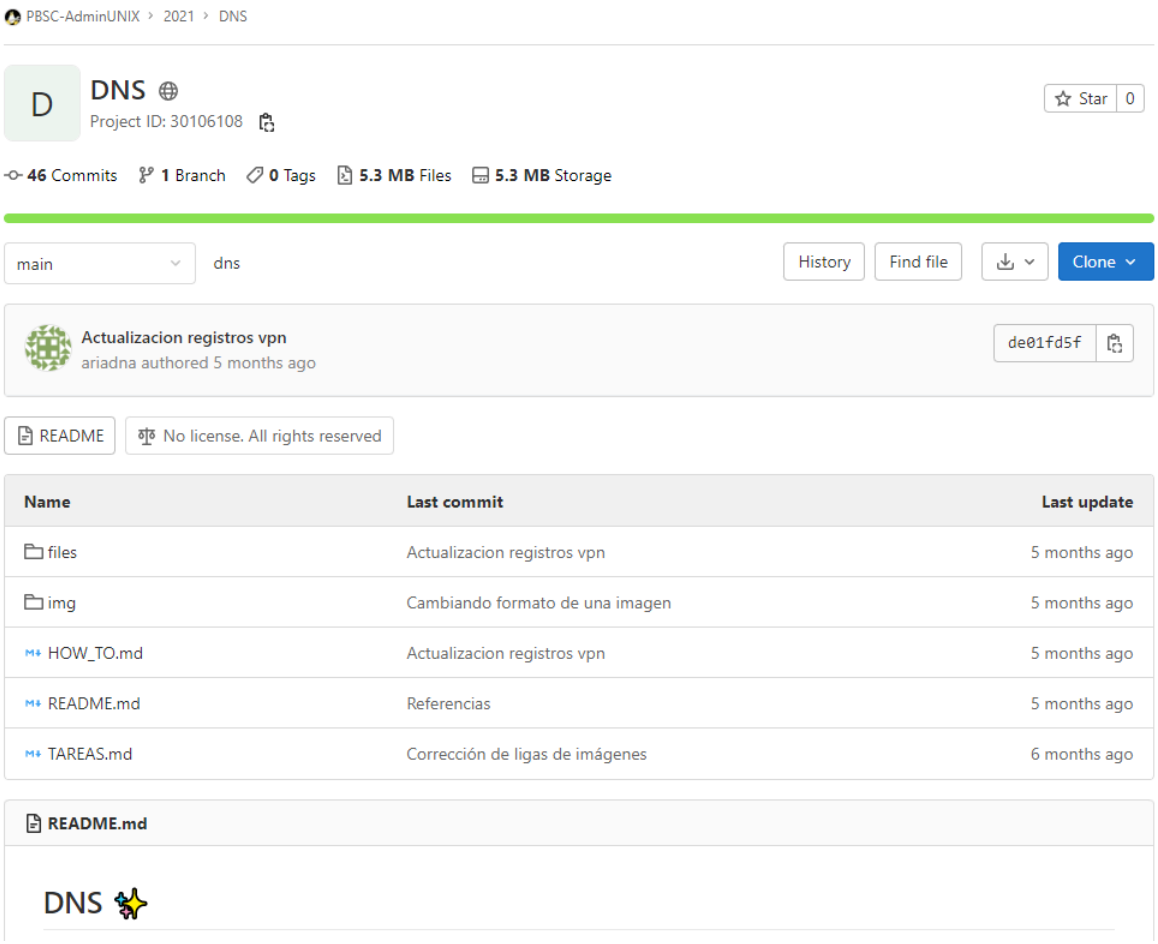

Figure 1: Project on Gitlab

I can safely say that this repository on GitLab was the one that trained me the most for tools of this type, mainly because all the changes were made from the virtual machine, so we needed to perform all the actions (configuration, writes, modifications, etc.) from the terminal and for this it was necessary to know the Linux text editors, as well as the git commands for the terminal.

3 My experience with this project

The first changes to the repository of this project were mainly to store the models I created with MagicaVoxel, therefore the changes were initially made from the GitHub website, however, when I started creating the scenario, it was necessary to add several files to the repository, so I decided to start using the Windows terminal to handle the changes.

At first I thought it would be confusing because I had only used the tool from Linux, however the commands are the same and the 'cmd' text editor is very similar to the Linux 'vi' editor, so it didn't take long for me to understand GitHub aspects.

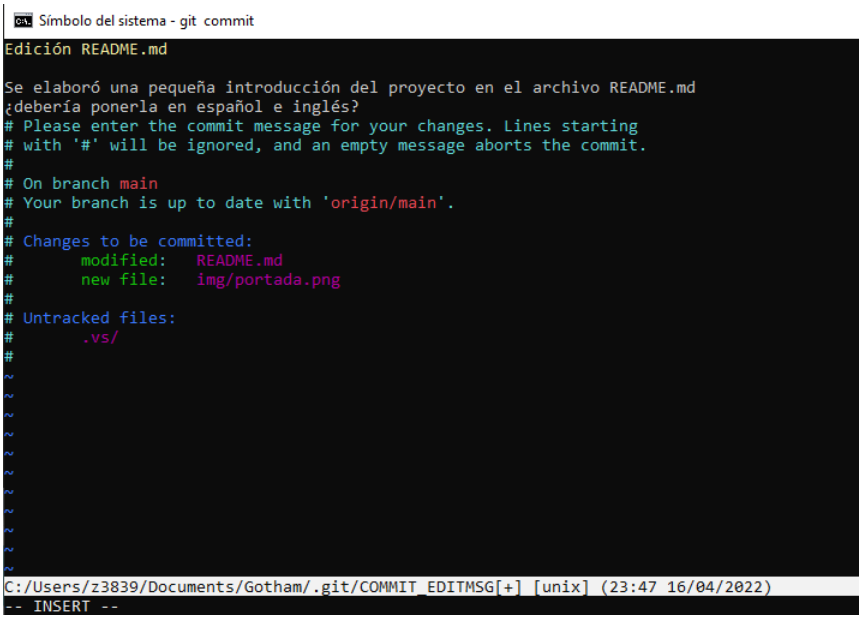

Figure 2: Text editor from cmd

In addition, this made it a lot easier for me to make the changes, as I was able to upload files faster than the online version, plus it makes changes and deletions much easier.

Finally, I can mention that, because the project was done by myself, it was not necessary to use different branches of the repository and most of the changes were made directly on the main branch when I was sure to have a working version.

```
C:\Users\z3839\Documents\Gotham>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 2.10 MiB | 3.70 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Ari3839/Gotham.git
   31a1759..d66ea54  main -> main

C:\Users\z3839\Documents\Gotham>
```

Figure 3: Git push

Also, during the realization of the project, I mistakenly attached a file with extension .vs by updating the files through regular expressions (a very bad technique to 'save time'), which not only should not be included in the repository, but was also overly heavy, thus it failed the git push, but I had already done the git commit, therefore I couldn't remove the file from the commit.

To solve this inconvenient, I had to look in my git notes for the command to observe the code or numeric identifier of the existing versions in my repository with the command 'git log –oneline', from which you can perform a reset and obtain a previous version of the repository. Clearly, I had to make the changes to the version that I couldn't upload to GitHub again, however, I made a local backup of those files before applying the reset, so there was no major inconvenience, and I was able to continue with the project.

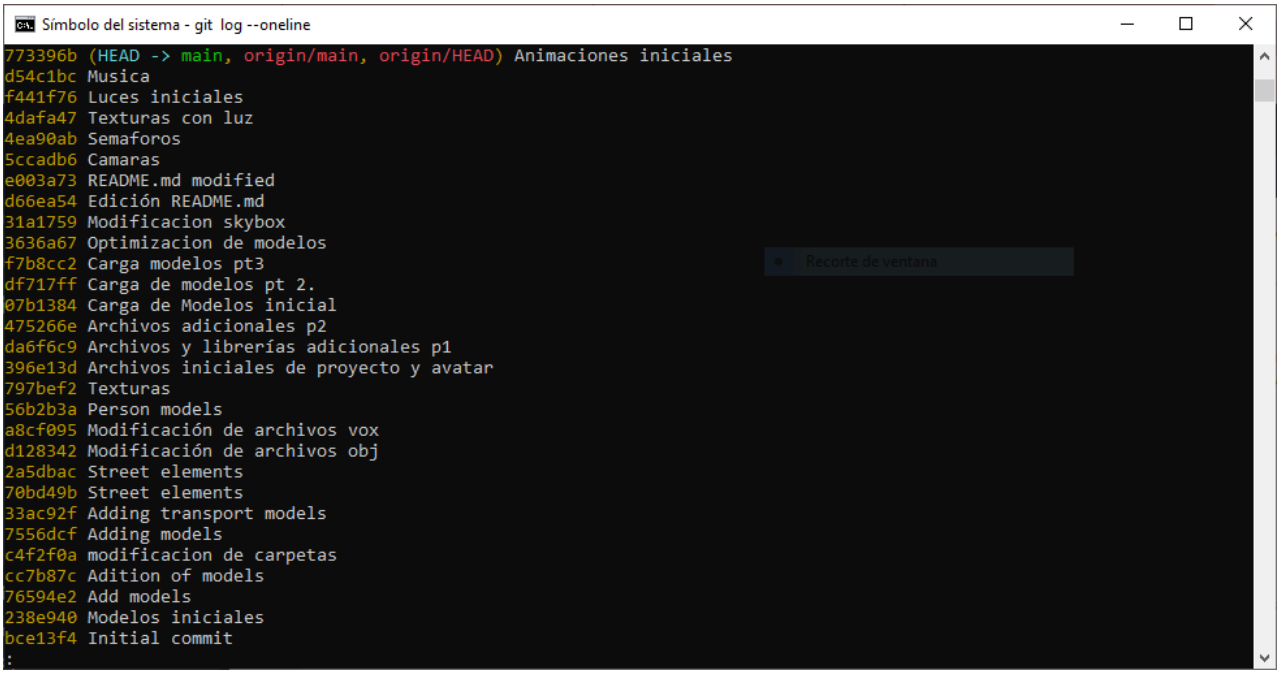

Figure 4: Git log

4 Evidencias

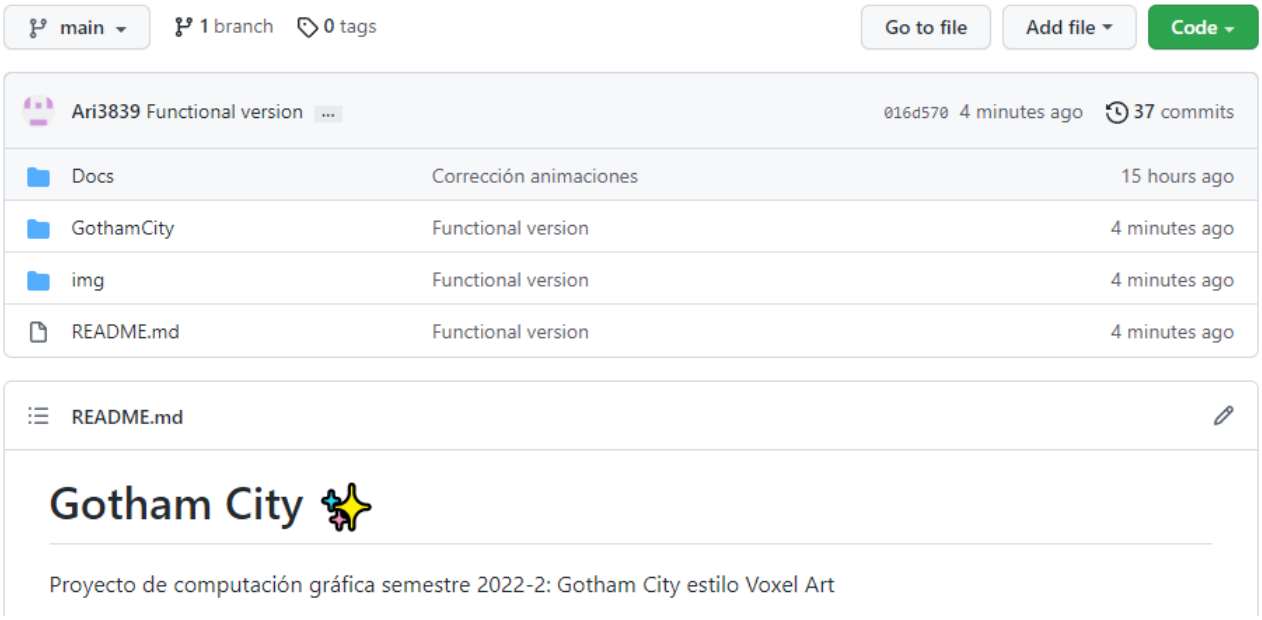

Figure 5: Repository

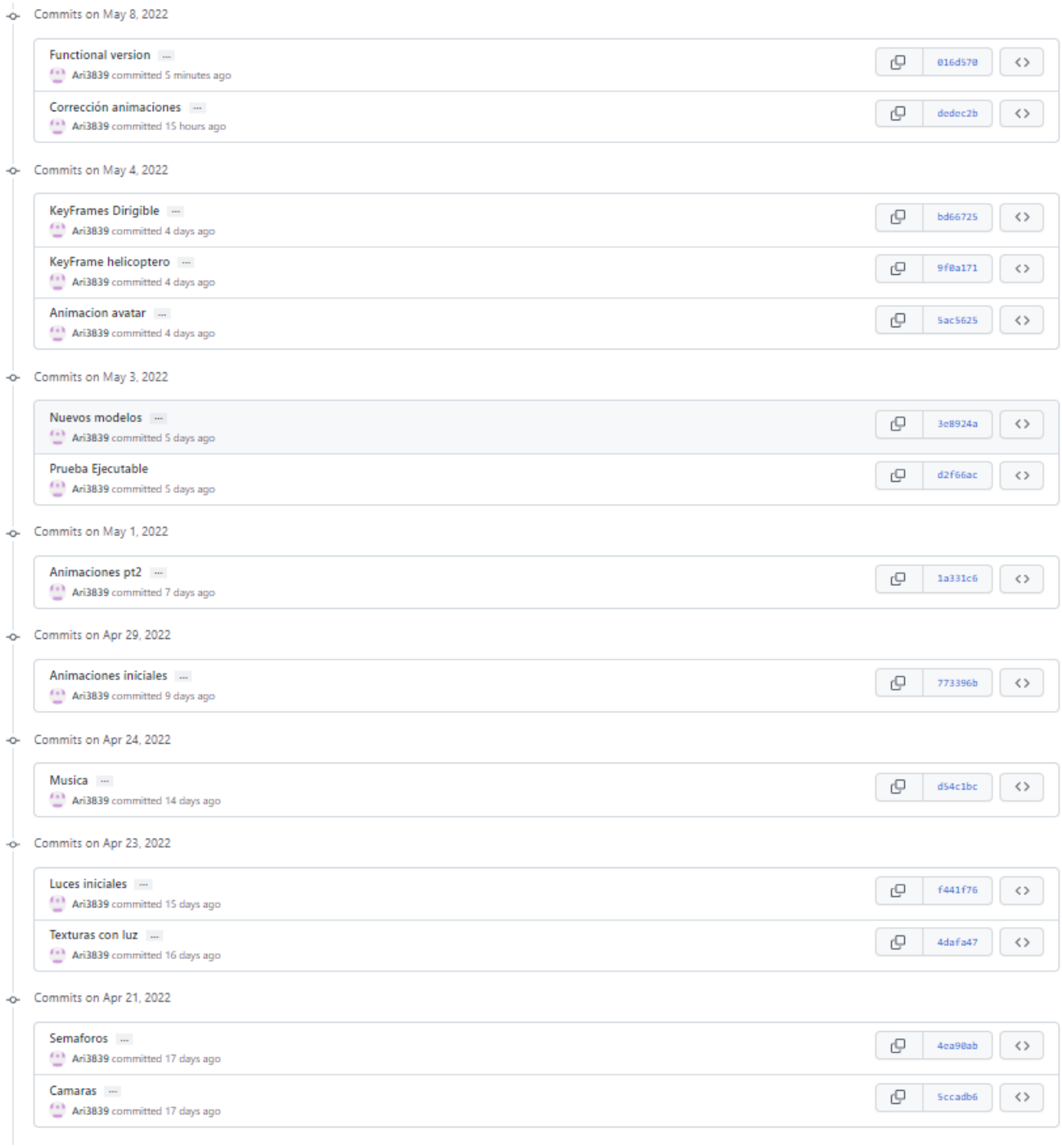

Figure 6: Commits part 1

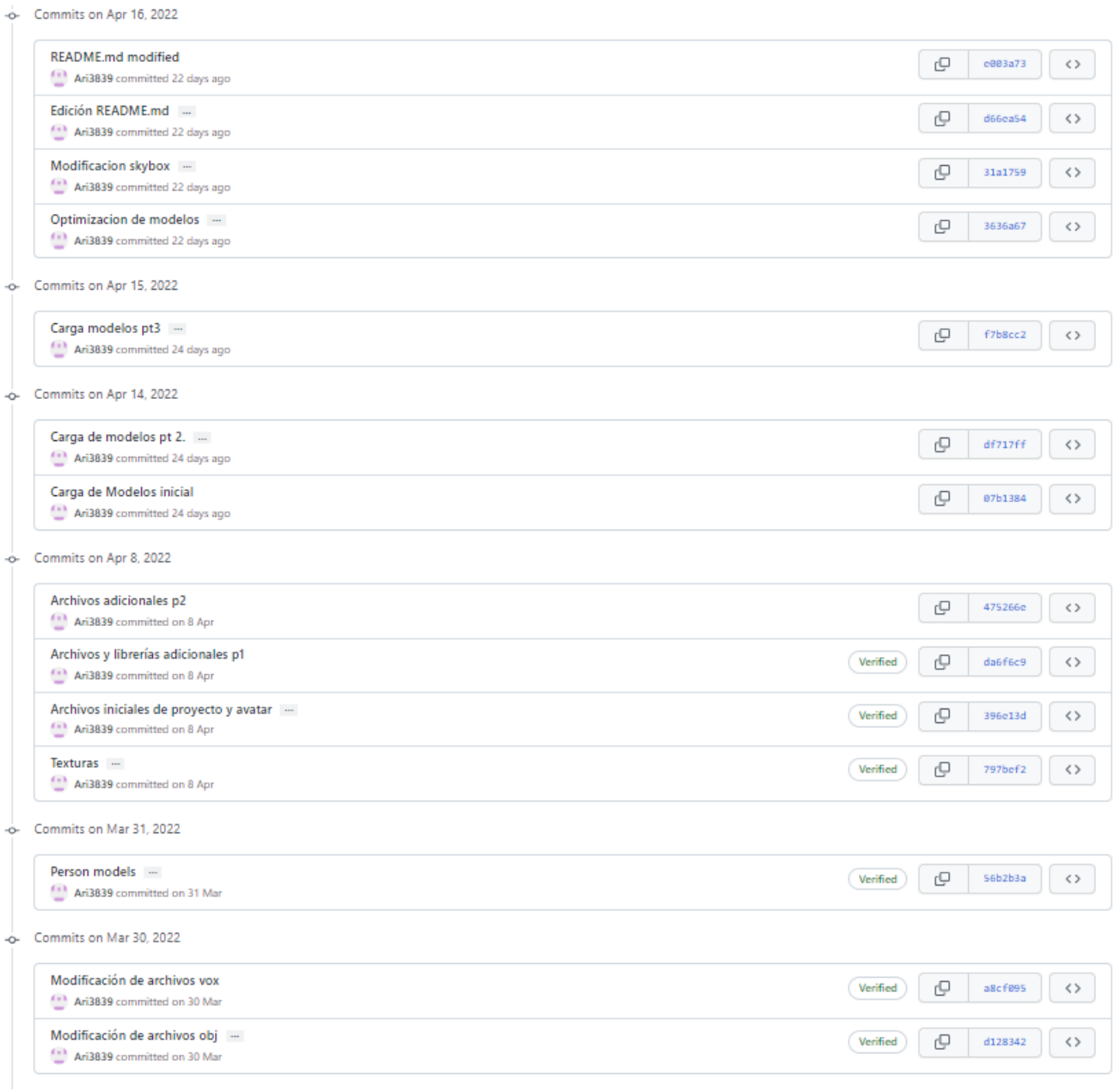

Figure 7: Commits part 2

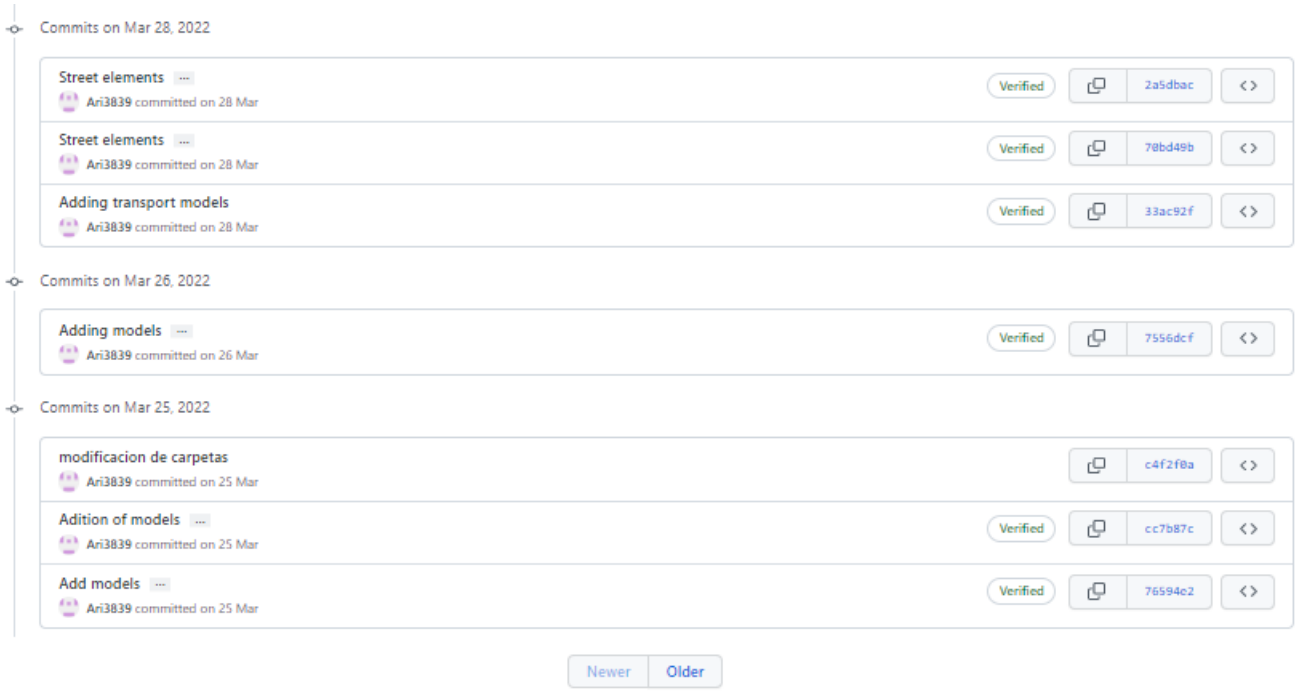

Figure 8: Commits part 3

References

[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/