

# Collaborative Learning Using Git with GitLab in Students of the Engineering Programming Course

Renzo Gustavo Bolivar Valdivia<sup>1</sup>[0000–0002–1984–5025]

Universidad Nacional de San Agustín de Arequipa, Perú  
rbolivarv@unsa.edu.pe  
<http://www.unsa.edu.pe>

**Abstract.** This research has used Git, software for Version Control, as well as the GitLab web platform for teaching programming collaboratively, which has strengthened collaborative learning in the classroom, a skill that for a long time was a very alternative important for team teaching of programming courses in engineering schools; The use of the Git tool helped the teacher to improve the teaching of programming and allowed through the GitLab platform for students to interact and develop the skills of critical thinking, teamwork, correcting and teaching to program low-performance students, the results of the capacities were visualized and quantified through the reports delivered by the GitLab web platform.

**Keywords:** Collaborative Learning · Programming · Version Control System · Git · GitLab.

## 1 Introduction

With the demands of technological advances in software, it is required that programming teams work in a coordinated and collaborative way, skills that every programmer must have today. In engineering careers, version control systems are used, a tool for learning software projects, in this research we present the results of using the GitLab web platform, which implemented collaborative learning in the programming course.

In **Using GitHub in the Classroom Predicts Student Learning Outcomes and Classroom Experiences: Findings from a Survey of Students and Teachers** [7] it was discovered that students who used GitHub (versus did not use) in the classroom felt that they learned more about teamwork and collaboration through their programming course. The data also supported that students who used GitHub in the classroom allows students to collaborate and contribute to the learning process of others.

In **GitHut como Herramienta Docente** [12] presents a teaching experience in the subject related to software engineering where GitHub is used as a

basic teaching tool for the development of the practical part of the course. This contribution focuses on motivating this experience, explaining its implementation, evaluating the benefits and risks, which concludes that Git when applied to teaching is a learning and teaching management tool; It also concludes that the emergence of platforms such as GitHub poses new challenges to teaching and should be integrated with existing educational platforms.

In **Git on the Cloud in the Classroom** [10] demonstrates the benefits of using Version Control in courses of any engineering career; they conclude that engineering students must be experts in the use of version control software as part of the presentation of their work, they also conclude that said software tool is simple to use to store versions of files in the cloud to coordinate and send the works.

In **Using GitHub in the Classroom – a Collaborative Learning Experience** [9] it presents the results of a collaborative learning experiment conducted through the use of GitHub in laboratory tasks, where they focus research on the direct interaction of students in the learning process of others. The students formed a more critical approach to the software developed, helped to improve not only their own software skills but also group learning.

In **Using GitHub Classroom To Teach Statistics** [3] through the teaching of Git and GitHub, they implemented GitHub Classroom in two educational environments: an introductory laboratory of computational statistics and a more advanced course of computational statistics. It was concluded that the Git workflow can be used in both introductory and advanced courses and by instructors without previous experience in GitHub, thus motivating the use of version control in students and having more confidence in their ability to collaborate using this tool.

In the present investigation, the web platform **GitLab** was used because it is open source and allows to create private projects unlike **GitHub**, a very important feature when deciding which web platform to use in the laboratory.

## **2 The Experience with Git and GitLab**

### **2.1 Space in which the Experience is Developed**

It has been developed and implemented with the students of programming of the first year of engineering in courses of **Computing I** and **Computation II**, in the professional career of Telecommunications Engineering of the National University of San Agustín of Arequipa, Peru.

### **2.2 Duration**

During the first and second semester of the 2017 academic year.

### 2.3 That motivated him to develop the experience

One of the pillars of the programming course is to be able to develop applications in groups, without a suitable tool it is not possible to develop this skill in class. During the development of my Master's thesis is that I know the **Git** and **GitLab** software tool very useful for editing my thesis in  $L^A T E X$ , programs and documents, after knowing that it can be used to teach collaborative learning in groups is that I decide to use the tool during my programming courses.

### 2.4 Software tool used

**Git** : Version control software developed by Mr. Linux Torvalds, designed to control versions of applications with a large number of source code files [5].

**GitLab** : Version control web service, allows you to store versions of files in the cloud with an open source license, sharing it privately or publicly [6].

### 2.5 How the experience developed

To obtain collaborative learning in software programming laboratories we have used the **GitLab** web platform.

First, several groups of 05 students were organized in the laboratory, initially they were asked to investigate the **Git** tool, so that they have prior contact with the tool; the entire course management was through the virtual Telecommunications Engineering classroom.

**Learning Git** Subsequently they were handed a digital manual with the basic **Git** commands [14].

The basic configuration commands of the **Git** environment:

```
git config --global user.name "__Your name__"
git config --global user.email "__Your mail__"
```

As an initial practice they entered a local folder on the computer, **Git** is initialized, its status is verified and if any changes were made to the files through the following commands:

```
git init
git status
git log
```

They were taught to modify their program within the initialized folder and can control the changes using the commands:

```
git add -A
git commit -m "___Detailed Comment of commit___"
```

Each student created their account on **GitLab**, the general working group was created **epit-programacion** grouping in number of 05 users. Students were trained to upload software projects from local folders to **GitLab**, using the commands:

```
git remote add origin https://renbol@gitlab.com/renbol/Proyect.git
git remote -v
git push origin master
git add -A
git commit -m "___Detailed Comment of the commit___"
git push origin master
```

Each laboratory group member must **clone** the **GitLab** repository on your local computer so you can contribute and send the changes to the repository **master** through the commands:

```
git clone https://renbol@gitlab.com/renbol/Proyect.git
git remote -v
git status
git add -A
git commit -m "___Detailed Comment of commit___"
git log
git fetch origin
git merge origin/master
git push origin master
```

## 2.6 Results of Collaborative Learning in GitLab

The collaborative procedure to learn to program in the engineering course was based on mastering Git and using GitLab to collaborate with the learning group. In GitLab through the menu of **Issues** (see Fig. 1), the students were assigned the programming tasks and could comment on their classmate's code, with feedback from all the members of the group, commented on and corrected their final code.

The result of all the comments of each member of the collaboration group as well as the corrections made to it can be observed and quantified in the menu **Comments** (see Fig. 1)

The teacher entered to review the code of each program and visualized the changes he has suffered since the code began to be written, until its completion even sequentially through the menu **Commits** (see Fig. 2), here the modifications to the program code are quantified.

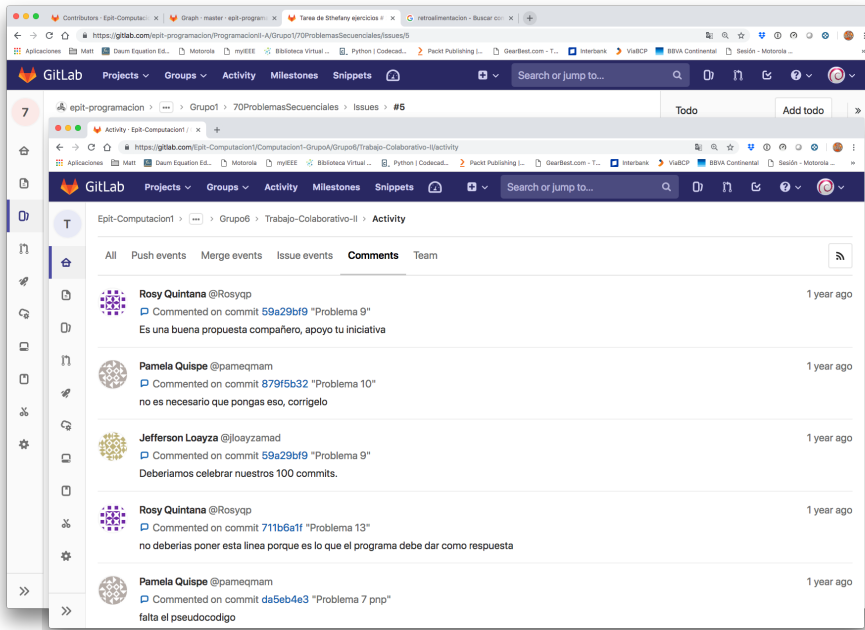


Fig. 1. GitLab Issues and Comments

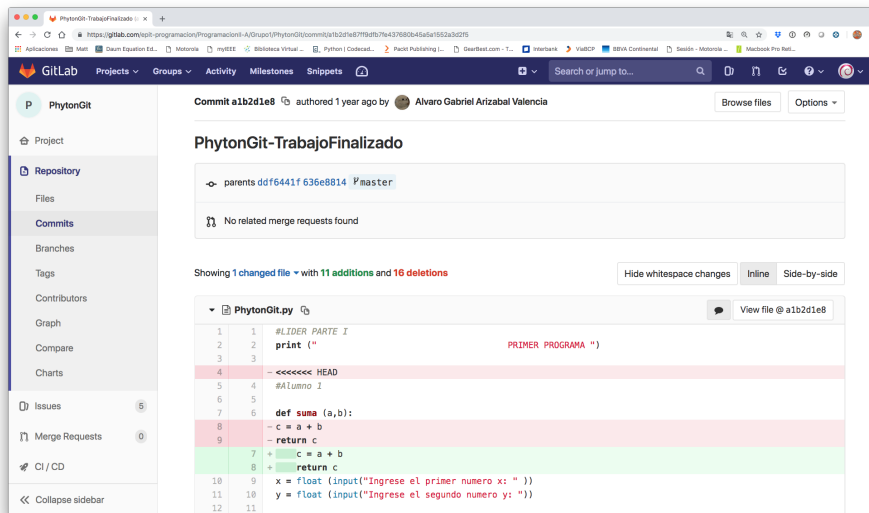
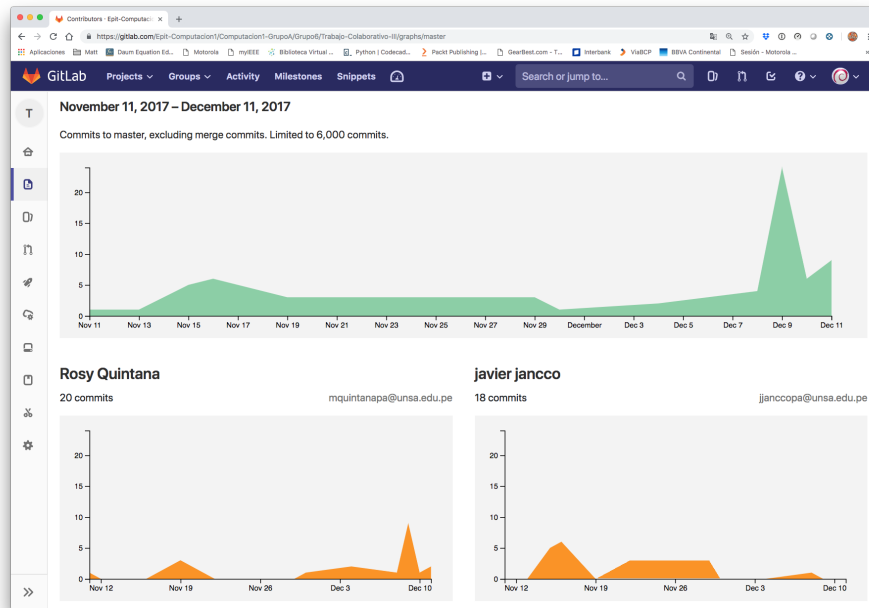


Fig. 2. GitLab Commits Code

The teacher also used the menu of **Contributors** (see Fig. 3) where you can observe in a graphic way how each group member has worked their code and a group graphics, where it is observed how students work at different times during the week.



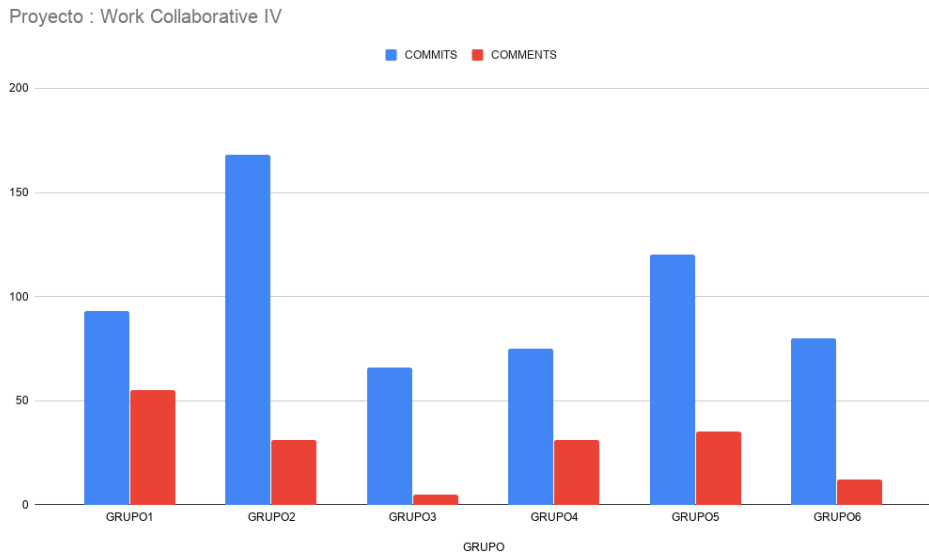
**Fig. 3.** GitLab Contributors

In the **Graph** menu the versions, contributions, modifications and results of the code are represented chronologically.

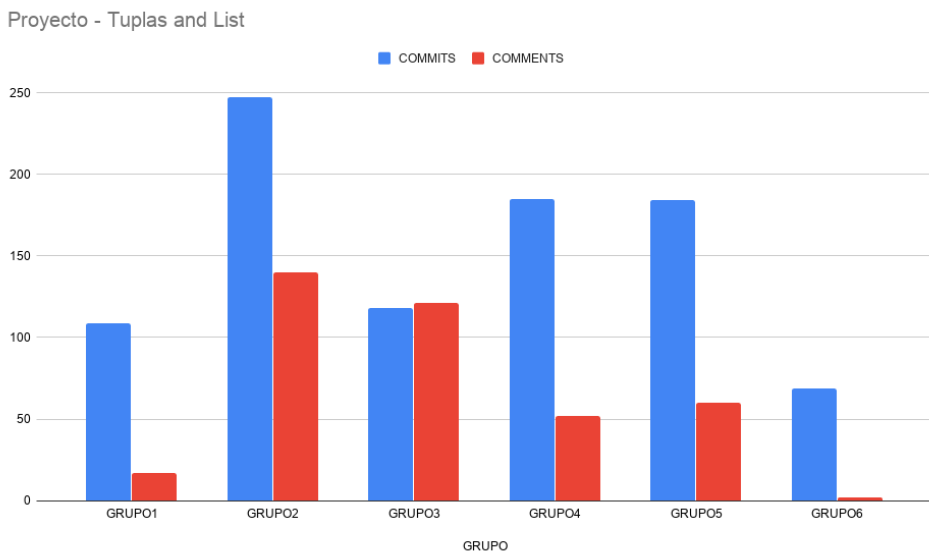
The quantifiable results are taken from the **Comments** menu and **Commits** menu that allowed us to have data of the collaborative work carried out by the group of students in each assigned software project.

We show results of the project called **Work Collaborative IV** in the course of Computing I (see Fig. 4) and of the project called **Tuples and List** in the course of Computing II (see Fig. 5).

In the graphs the collaborative learning is highlighted to the groups that if they worked collaborating with the learning of their peers, these groups have a red bar of up to 140 **Comments** the highest, the groups with red bar are also observed very small where the collaborative contribution is very low reaching the groups up to 2 **Comments** (see Fig. 5), in general we can indicate that the Computation II groups agreed to work collaboratively more than the Computation I groups.



**Fig. 4.** GitLab Graph Commits Comments Group Computation I



**Fig. 5.** GitLab Graph Commits Comments Group Computation II

### 3 Results and Discussion

#### 3.1 How has the impact on the student's learning processes been? How have you managed to show such an impact?

Collaborative learning of group software programming, through **GitLab** has been more organized, groups did not need to meet to distribute, sort and combine their contributions, there is feedback from the group to improve group learning, such as there are the files stored in the cloud, the work is easily accessible by the students and there is detailed control of each one's contribution to group learning.

Collaborative learning has been evidenced through the graphs and data of **GitLab** (see Fig. 4, 5, 2 and 3) it is also evidenced by the impact on the grades compared to 2016, in addition each student was able to quantify and rate the performance and contribution of their classmates using as a tool a **GitLab**, placing a grade that was averaged with that of the teacher, also based on the evidence of collaboration, the student had the option of requesting that one or more classmates leave the group due to poor performance.

#### 3.2 State the achievements and difficulties encountered in the experience

After working 02 semesters applying **Git** and **GitLab** in the computer classrooms of **Computation I** and **Computation II** it has been observed that:

1. Students in groups have learned more than working individually, the **GitLab** tool has allowed them to teach their low-performance peers to program.
2. The students did a more organized group learning, without having to meet physically, they also have detailed control of the changes made in their programs chronologically.
3. The teacher and student could rate the group and individual performance, thanks to the reports in **GitLab**.
4. Students of the **Computer I** course due to their educational habits, compared to the students of **Computer II**, it was difficult to change their learning habits, which caused their programming laboratory performance to be low.

#### 3.3 Describe the possibility that your experience has of being replicated in other courses, / activities / classroom / laboratory, etc.

The **Version Control Systems** with **Git** and/or **GitLab** should be used by engineering students both undergraduate and postgraduate in the preparation of any document individually and in groups, which will allow them to elaborate and



correct chronological of all his works in a very detailed way. **GitLab** should be a tool for engineering teachers in teaching their courses for collaborative learning, accompanied by ongoing training on its use and advantages.

## 4 Conclusions

In the present work it was demonstrated that the use of **Git** and **GitLab** as tools that help collaborative learning is very useful in the classroom and laboratory for the engineering teacher and student who study in the first year of studies, despite the fact that Engineering In Telecommunications it is not a career specialized in programming learning, very good results were obtained, which indicates that any professional career in engineering should use these tools. It was demonstrated how simple and practical it is for the student to use **Git**, in addition that the teacher can easily use the **GitLab** platform for collaborative learning in the classroom and laboratory. The experience was centered in **GitLab** for being an open source platform and where you can create private projects, 100% platform compatible with the **Git** tool, which give you the great advantage over other platforms. It would be a new challenge for engineering teaching to apply the **Git** and **GitLab** tools in all the courses that require writing reports or important documents such as thesis and research papers in a very simple and practical way.

## References

1. Social coding in GitHub : Transparency and Collaboration in an Open Software Repository. ACM Press (2012). <https://doi.org/10.1145/2145204.2145396>
2. Feliciano, J., Storey, M.A., Zagalsky, A.: Student experiences using GitHub in software engineering courses. In: Proceedings of the 38th International Conference on Software Engineering Companion - ICSE 16. ACM Press (2016). <https://doi.org/10.1145/2889160.2889195>
3. Fiksel, J., Jager, L.R., Hardin, J.S., Taub, M.A.: Using GitHub Classroom To Teach Statistics. *Journal of Statistics Education* **27**(2), 110–119 (may 2019). <https://doi.org/https://doi.org/10.1080/10691898.2019.1617089>
4. Francese, R., Gravino, C., Risi, M., Scanniello, G., Tortora, G.: On the Experience of Using Git-Hub in the Context of an Academic Course for the Development of Apps for Smart Devices. In: DMS. pp. 292–299 (2015)
5. Git, T.: Web Page Git (Nov 2019), <https://git-scm.com/>
6. GitLab, T.: Web Page GitLab (Nov 2019), <https://gitlab.com/>
7. Hsing, C., Gennarelli, V.: Using github in the classroom predicts student learning outcomes and classroom experiences: Findings from a survey of students and teachers. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education. pp. 672–678. ACM (2019)
8. Kalyan, A., Chiam, M., Sun, J., Manoharan, S.: A Collaborative Code Review Platform for GitHub. In: 2016 21st International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE (nov 2016). <https://doi.org/10.1109/ICECCS.2016.032>

9. Kertesz, C.Z.: Using GitHub in the classroom - a collaborative learning experience. In: 2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME). IEEE (oct 2015). <https://doi.org/10.1109/SIITME.2015.7342358>
10. Lawrance, J., Jung, S., Wiseman, C.: Git on the cloud in the classroom (2013). <https://doi.org/10.1145/2445196.2445386>
11. Marquardson, J., Schuetzler, R.: Teaching Tip: Learning by Teaching through Collaborative Tutorial Creation: Experience using Github and AsciiDoc. *Journal of Information Systems Education* **30**, 10–18 (01 2019)
12. Lopez Pellicer, F.J., Bejar, R., Latre, M.A., Nogueras Iso, J., Zarazaga Soria, F.J.: Github como herramienta docente. In: *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática*. pp. 66–73. Universitat Oberta La Salle (2015)
13. Ruiz, J.A.V., Palma, E.A.C., Intriago, J.M.L., González, A.H., Espinosa, M.M.: Configuración de Usuarios en Git Para Grupos de Desarrollo en Entornos Universitarios. *Revista San Gregorio* **1**(9), 84–89 (2015)
14. Scott Chacon, B.S.: *Pro Git*. APRESS L.P. (2014), <https://bit.ly/339ZogN>
15. Vasilescu, B., Filkov, V., Serebrenik, A.: Perceptions of diversity on GitHub: A user survey. In: *Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering*. pp. 50–56. IEEE Press (2015)
16. Zakiah, A., Fauzan, M.N.: Collaborative Learning Model of Software Engineering using Github for informatics student (apr 2016). <https://doi.org/10.1109/CITSM.2016.7577521>