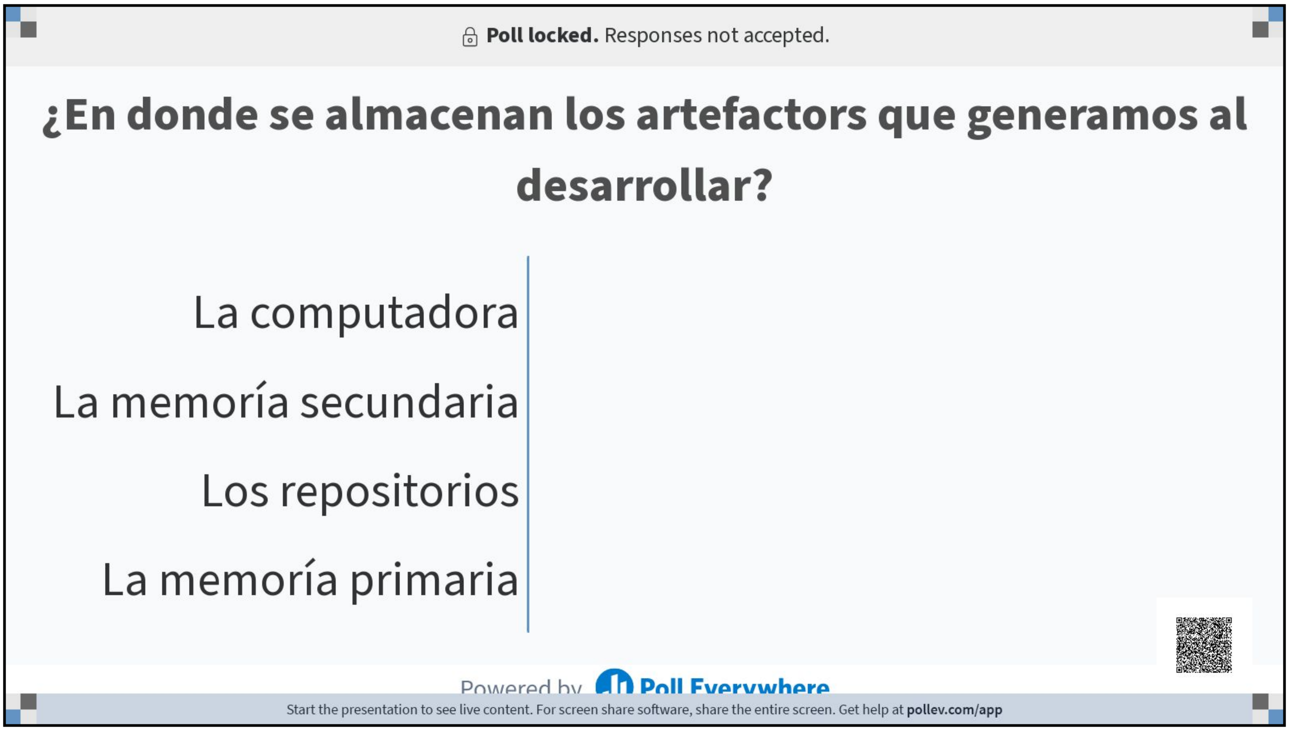
Repositorios de software y software abierto

Sivana Hamer - sivana.hamer@ucr.ac.cr
Escuela de Ciencias de la Computación

¿Qué artefactos generamos al desarrollar?

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**



Poll locked. Responses not accepted.

¿En donde se almacenan los artefactors que generamos al desarrollar?

La computadora

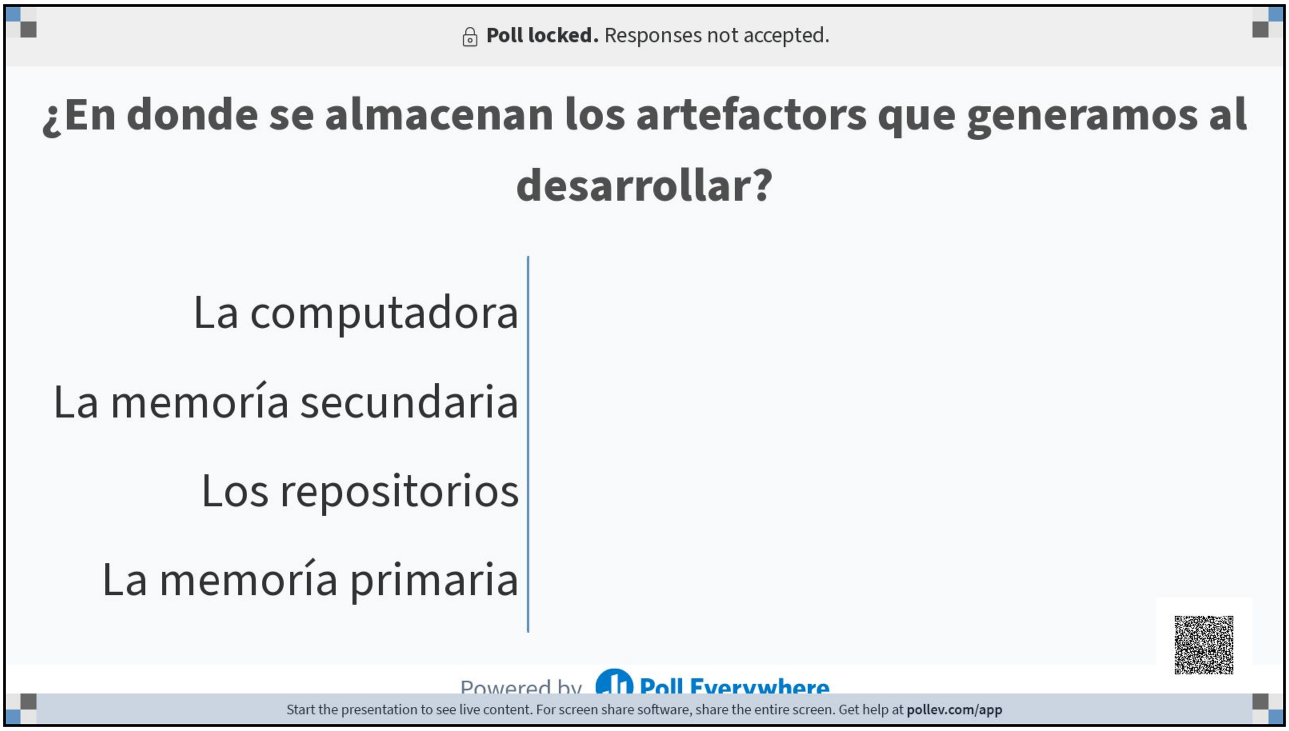
La memoría secundaria

Los repositorios

La memoría primaria

Powered by Poll Everywhere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app



Poll locked. Responses not accepted.

¿En donde se almacenan los artefactors que generamos al desarrollar?

La computadora

La memoría secundaria

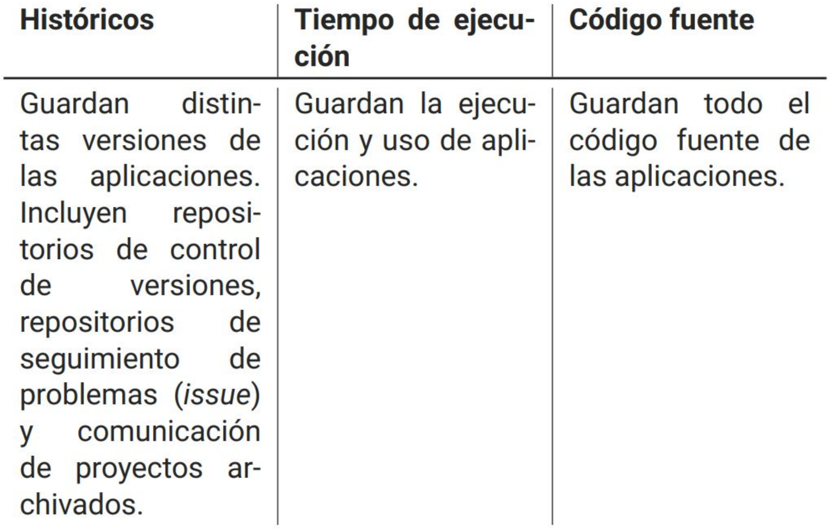
Los repositorios

La memoría primaria

Powered by Poll Everywhere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

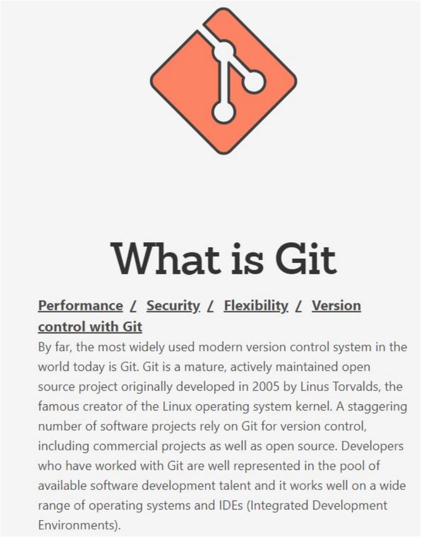
Existen distintos tipos de repositorios de software



Históricos	**Tiempo de ejecución**	**Código fuente**
Guardan distintas versiones de las aplicaciones. Incluyen repositorios de control de versiones, repositorios de seguimiento de problemas (*issue*) y comunicación de proyectos archivados.	Guardan la ejecución y uso de aplicaciones.	Guardan todo el código fuente de las aplicaciones.

GitHub es uno de los repositorios de código más usados…

GitHub *hostea* por debajo Git



What is Git

Performance / Security / Flexibility / Version control with Git

By far, the most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments).

When poll is active, respond at **pollev.com/sivanahamer104**

¿Alguien ha usado Git antes?

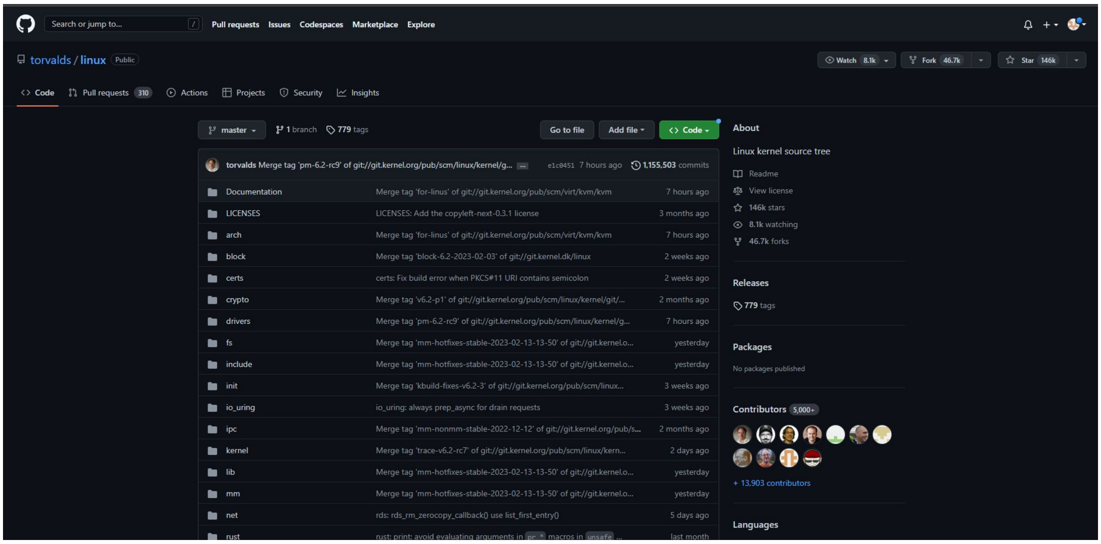
Si

No

Powered by Poll Everywhere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

En open source software, el código se encuentra abierto



torvalds Merge tag 'pm-6.2-rc9' of git://git.kernel.org/pub/scm/linux/kernel/g... e1c0451 7 hours ago 1,155,503 commits

Documentation	Merge tag 'for-linus' of git://git.kernel.org/pub/scm/virt/kvm/kvm	7 hours ago
LICENSES	LICENSES: Add the copyleft-next-0.3.1 license	3 months ago
arch	Merge tag 'for-linus' of git://git.kernel.org/pub/scm/virt/kvm/kvm	7 hours ago
block	Merge tag 'block-6.2-2023-02-03' of git://git.kernel.dk/linux	2 weeks ago
certs	certs: Fix build error when PKCS#11 URI contains semicolon	2 weeks ago
crypto	Merge tag 'v6.2-p1' of git://git.kernel.org/pub/scm/linux/kernel/git/...	2 months ago
drivers	Merge tag 'pm-6.2-rc9' of git://git.kernel.org/pub/scm/linux/kernel/g...	7 hours ago
fs	Merge tag 'mm-hotfixes-stable-2023-02-13-13-50' of git://git.kernel.o...	yesterday
include	Merge tag 'mm-hotfixes-stable-2023-02-13-13-50' of git://git.kernel.o...	yesterday
init	Merge tag 'kbuild-fixes-v6.2-3' of git://git.kernel.org/pub/scm/linux...	3 weeks ago
io_uring	io_uring: always prep_async for drain requests	3 weeks ago
ipc	Merge tag 'mm-nonmm-stable-2022-12-12' of git://git.kernel.org/pub/s...	2 months ago
kernel	Merge tag 'trace-v6.2-rc7' of git://git.kernel.org/pub/scm/linux/kern...	2 days ago
lib	Merge tag 'mm-hotfixes-stable-2023-02-13-13-50' of git://git.kernel.o...	yesterday
mm	Merge tag 'mm-hotfixes-stable-2023-02-13-13-50' of git://git.kernel.o...	yesterday
net	rds: rds_rm_zerocopy_callback() use list_first_entry()	5 days ago
rust	rust: print: avoid evaluating arguments in pr_* macros in unsafe ...	last month

About

Linux kernel source tree

Readme

View license

146k stars

8.1k watching

46.7k forks

Releases

779 tags

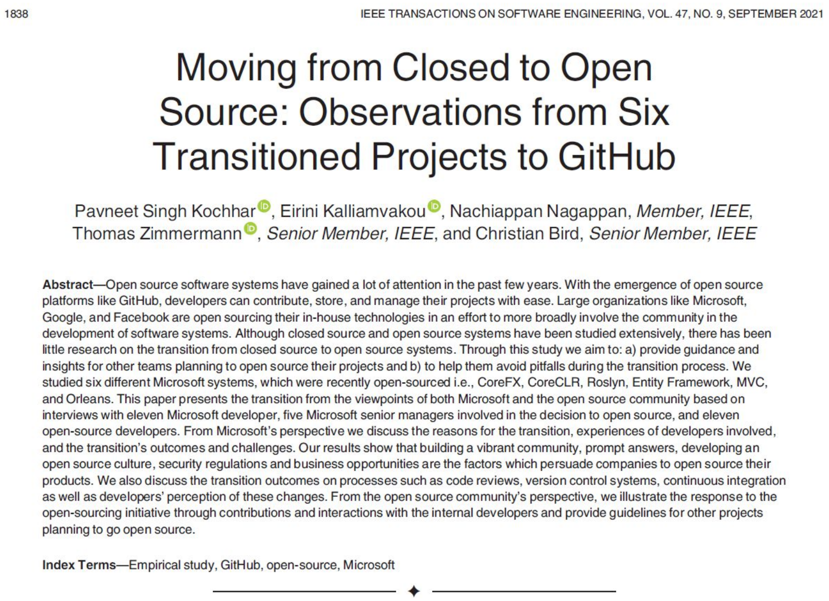
Packages

No packages published

Contributors 5,000+

+ 13,903 contributors

Languages



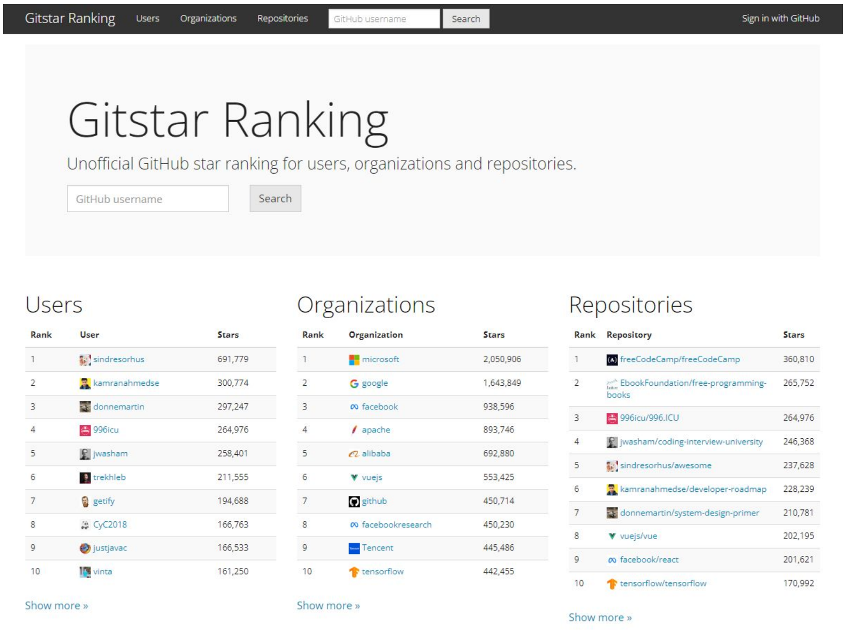
1838 IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 47, NO. 9, SEPTEMBER 2021

Moving from Closed to Open Source: Observations from Six Transitioned Projects to GitHub

Pavneet Singh Kochhar, Eirini Kalliamvakou, Nachiappan Nagappan, *Member, IEEE*, Thomas Zimmermann, *Senior Member, IEEE*, and Christian Bird, *Senior Member, IEEE*

Abstract—Open source software systems have gained a lot of attention in the past few years. With the emergence of open source platforms like GitHub, developers can contribute, store, and manage their projects with ease. Large organizations like Microsoft, Google, and Facebook are open sourcing their in-house technologies in an effort to more broadly involve the community in the development of software systems. Although closed source and open source systems have been studied extensively, there has been little research on the transition from closed source to open source systems. Through this study we aim to: a) provide guidance and insights for other teams planning to open source their projects and b) to help them avoid pitfalls during the transition process. We studied six different Microsoft systems, which were recently open-sourced i.e., CoreFX, CoreCLR, Roslyn, Entity Framework, MVC, and Orleans. This paper presents the transition from the viewpoints of both Microsoft and the open source community based on interviews with eleven Microsoft developer, five Microsoft senior managers involved in the decision to open source, and eleven open-source developers. From Microsoft's perspective we discuss the reasons for the transition, experiences of developers involved, and the transition's outcomes and challenges. Our results show that building a vibrant community, prompt answers, developing an open source culture, security regulations and business opportunities are the factors which persuade companies to open source their products. We also discuss the transition outcomes on processes such as code reviews, version control systems, continuous integration as well as developers' perception of these changes. From the open source community's perspective, we illustrate the response to the open-sourcing initiative through contributions and interactions with the internal developers and provide guidelines for other projects planning to go open source.

Index Terms—Empirical study, GitHub, open-source, Microsoft



Gitstar Ranking

Unofficial GitHub star ranking for users, organizations and repositories.

Users

Rank	User	Stars
1	sindresorhus	691,779
2	kamranahmedse	300,774
3	donnemartin	297,247
4	996icu	264,976
5	jwasham	258,401
6	trekhleb	211,555
7	getify	194,688
8	CyC2018	166,763
9	justjavac	166,533
10	vinta	161,250

Show more »

Organizations

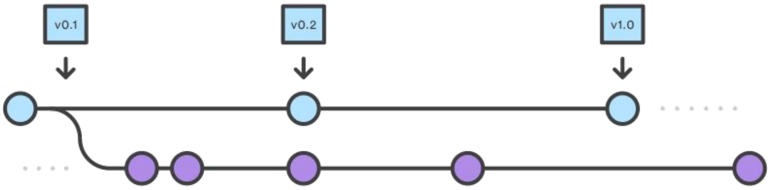
Rank	Organization	Stars
1	microsoft	2,050,906
2	google	1,643,849
3	facebook	938,596
4	apache	893,746
5	alibaba	692,880
6	vuejs	553,425
7	github	450,714
8	facebookresearch	450,230
9	Tencent	445,486
10	tensorflow	442,455

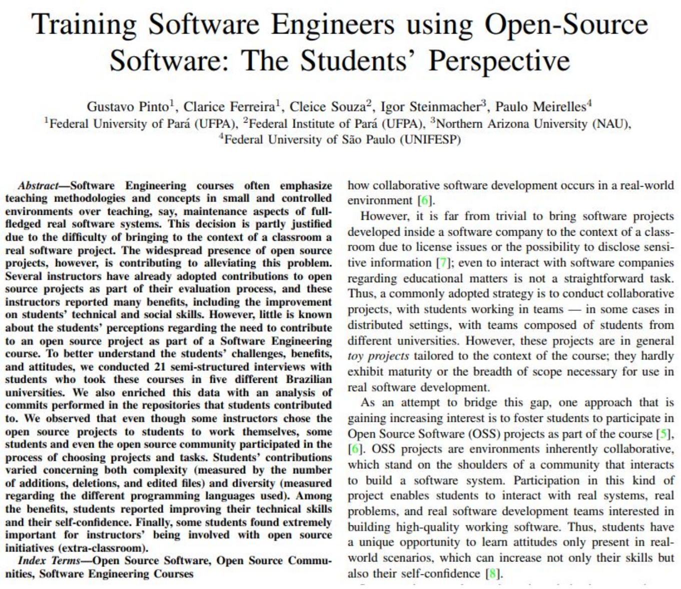
Show more »

Repositories

Rank	Repository	Stars
1	freeCodeCamp/freeCodeCamp	360,810
2	EbookFoundation/free-programming-books	265,752
3	996icu/996.ICU	264,976
4	jwasham/coding-interview-university	246,368
5	sindresorhus/awesome	237,628
6	kamranahmedse/developer-roadmap	228,239
7	donnemartin/system-design-primer	210,781
8	vuejs/vue	202,195
9	facebook/react	201,621
10	tensorflow/tensorflow	170,992

Show more »





Training Software Engineers using Open-Source Software: The Students' Perspective

Gustavo Pinto[1], Clarice Ferreira[1], Cleice Souza[2], Igor Steinmacher[3], Paulo Meirelles[4]
[1]Federal University of Pará (UFPA), [2]Federal Institute of Pará (UFPA), [3]Northern Arizona University (NAU),
[4]Federal University of São Paulo (UNIFESP)

***Abstract*—Software Engineering courses often emphasize teaching methodologies and concepts in small and controlled environments over teaching, say, maintenance aspects of full-fledged real software systems. This decision is partly justified due to the difficulty of bringing to the context of a classroom a real software project. The widespread presence of open source projects, however, is contributing to alleviating this problem. Several instructors have already adopted contributions to open source projects as part of their evaluation process, and these instructors reported many benefits, including the improvement on students' technical and social skills. However, little is known about the students' perceptions regarding the need to contribute to an open source project as part of a Software Engineering course. To better understand the students' challenges, benefits, and attitudes, we conducted 21 semi-structured interviews with students who took these courses in five different Brazilian universities. We also enriched this data with an analysis of commits performed in the repositories that students contributed to. We observed that even though some instructors chose the open source projects to students to work themselves, some students and even the open source community participated in the process of choosing projects and tasks. Students' contributions varied concerning both complexity (measured by the number of additions, deletions, and edited files) and diversity (measured regarding the different programming languages used). Among the benefits, students reported improving their technical skills and their self-confidence. Finally, some students found extremely important for instructors' being involved with open source initiatives (extra-classroom).**

***Index Terms*—Open Source Software, Open Source Communities, Software Engineering Courses**

how collaborative software development occurs in a real-world environment [6].

However, it is far from trivial to bring software projects developed inside a software company to the context of a classroom due to license issues or the possibility to disclose sensitive information [7]; even to interact with software companies regarding educational matters is not a straightforward task. Thus, a commonly adopted strategy is to conduct collaborative projects, with students working in teams — in some cases in distributed settings, with teams composed of students from different universities. However, these projects are in general *toy projects* tailored to the context of the course; they hardly exhibit maturity or the breadth of scope necessary for use in real software development.

As an attempt to bridge this gap, one approach that is gaining increasing interest is to foster students to participate in Open Source Software (OSS) projects as part of the course [5], [6]. OSS projects are environments inherently collaborative, which stand on the shoulders of a community that interacts to build a software system. Participation in this kind of project enables students to interact with real systems, real problems, and real software development teams interested in building high-quality working software. Thus, students have a unique opportunity to learn attitudes only present in real-world scenarios, which can increase not only their skills but also their self-confidence [8].

Hablemos del proyecto :)

Referencias

- Kagdi, H., Collard, M. L., & Maletic, J. I. (2007). A survey and taxonomy of approaches for mining software repositories in the context of software evolution. Journal of software maintenance and evolution: Research and practice, 19(2), 77-131.
- Hassan, A. E. (2008). The road ahead for mining software repositories. In 2008 frontiers of software maintenance (pp. 48-57). IEEE.
- Atlassian (2023). Getting git right. From: https://www.atlassian.com/git.
- Kochhar, P. S., Kalliamvakou, E., Nagappan, N., Zimmermann, T., & Bird, C. (2019). Moving from closed to open source: Observations from six transitioned projects to GitHub. IEEE Transactions on Software Engineering, 47(9), 1838-1856.

Referencias

- Lydia Hallie. (2020) 🌳🚀 CS Visualized: Useful Git Commands. Recuperado de: https://dev.to/lydiahallie/cs-visualized-useful-git-commands-37p1#pull