Innovating Computer Science Education with GitHub Integration

David Cantú Delgado

Tecnológico de Monterrey, México

Abstract

Developed skills in version control and collaborative development are essential in fields like Computer Science, Data Analytics, and Robotics. However, many students lack practical experience with tools such as GitHub, creating a disconnect between academic preparation and industry expectations. This study presents an educational intervention that integrates GitHub into an undergraduate Data Structures course to develop students' technical and collaborative competencies. Visual Studio Code was used as the development environment, and all course assignments were submitted through GitHub repositories to reinforce regular version control practices, facilitate information sharing among peers and instructors, and align with common industry workflows for collaborative project management. Students completed individual and team-based projects, engaging with essential workflows such as branching, conflict resolution, and pull request reviews. All students successfully completed the GitHub-based activities, demonstrating proficiency in version control and collaborative coding. Survey data indicated that 96% of students recognized the relevance of these skills for future academic or professional projects, and 95% expressed a high likelihood of continued GitHub use beyond the course. Qualitative feedback reflected increased confidence in software documentation and teamwork. Integrating professional tools into early coursework bridges critical gaps in technology education, reinforcing course content while aligning student learning with real-world development practices. These findings support the pedagogical value of embedding industry-standard tools like GitHub into the curriculum to foster practical, future-ready skills.

Keywords: *educational innovation, higher education, GitHub, computer science education, software engineering education.*

1. Introduction

In the software development industry, version control platforms like GitHub are essential for managing code, collaborating across teams, and maintaining project integrity. Despite its relevance, many undergraduate students in Computer Science and Information Technologies complete early-stage courses without meaningful exposure to these tools. This gap reduces their preparedness for real-world environments, where collaborative software development and effective documentation are standard expectations.

This educational innovation, "My First Commit: Using GitHub in Data Structures for Collaborative Projects and Software Development Innovation," addresses this challenge by integrating GitHub into a core Data Structures course. The initiative aims to develop students' technical competencies in version control, software documentation, and collaborative coding, ensuring that their academic experience reflects modern industry practices.

Through the consistent use of GitHub in individual and team-based assignments, students engage with key workflows such as branching, pull requests, and conflict resolution. Assignments are submitted through GitHub repositories, reinforcing continuous interaction with the platform and its core features. This structure allows students to become familiar with collaborative coding environments while improving their ability to manage and contribute to software projects.

The specific objectives of this intervention are: (1) to introduce practical GitHub skills and ensure that at least 80% of students can apply concepts such as version control, branching, and conflict resolution; (2) to have 90% of students integrate GitHub into their project workflow by the end of the course; and (3) to evaluate the impact of the innovation on project quality, tool adoption, and students' self-perceived technical readiness.

2. Context of the Innovation

The integration of technological tools in higher education is essential to prepare students for the demands of today's job market. In the field of software development, GitHub has become a key platform for version control and collaborative work [1]. However, many students in Computer Science and Information Technologies programs lack practical experience with GitHub, limiting their professional competence as they transition into the workforce [1].
This educational intervention is based on the hypothesis that incorporating GitHub into a Data Structures course would strengthen students' technical and collaborative skills, aligning their academic training with the expectations of the software industry. This assumption is supported by prior studies emphasizing the importance of integrating industry tools into academic curricula to foster relevant and applicable competencies. From a pedagogical perspective, personalized learning and consideration of individual learning styles—such as those proposed by Felder and Silverman are critical to ensuring that students reach their full potential, particularly when introduced to new technologies.[2]
Research has shown that the integration of digital tools into academic curricula enhances both teaching and learning by making instruction more relevant and engaging. Methodologies that support personalized learning, such as those described by von Feigenblatt et al. [3], enable the adaptation of practical activities to match the pace and needs of each student, fostering meaningful and autonomous learning. Additionally, the use of version control systems supports team collaboration and structured project organization—both of which are essential in modern software development.

To address these challenges, the intervention included:
Initial Training. Introductory sessions on GitHub and its integration with Visual Studio Code were delivered at the start of the course, providing students with a strong foundation. These sessions were designed with diverse learning styles in mind, combining visual explanations, hands-on exercises, and collaborative activities.
Repository Creation. Individual and group repositories were established for course activities, allowing students to gain experience managing code and collaborating within a structured environment. This approach promoted autonomy and self-efficacy, key elements of personalized learning [3].
Practical Assignments. A total of 15 individual activities and 4 major deliverables (2 individual, 2 team-based) required the use of GitHub for development and submission. These tasks were adapted to challenge both advanced learners and those needing more support, in line with active learning principles.
Assessment and Feedback. Continuous monitoring and formative assessment strategies were implemented to track progress and provide timely feedback. Personalized comments supported diverse learning preferences and ensured effective guidance.

3. Methodology

This educational intervention followed a structured and progressive methodology to ensure the effective integration of GitHub into the Data Structures course, using industry-standard tools. The main activities carried out during the implementation are described below:

3.1 Introduction to GitHub and Visual Studio Code

The first session introduced students to the fundamentals of version control and the role of GitHub in modern software development. It also included the installation and setup of Visual Studio Code (VS Code) as the integrated development environment (IDE) and the configuration of essential extensions such as the GitHub extension to streamline repository management.

3.2 Creation of Individual Repositories.

Each student was assigned a personal GitHub repository with individual access permissions. These repositories were synchronized with VS Code, enabling students to work locally on their machines and push changes to the remote repository on GitHub.

3.3 Hands-On Training Session.

Students participated in a practical training session where they learned to operate GitHub directly from within VS Code. In this session, they cloned repositories from GitHub into their local development environment, made and saved changes to files, followed by commits through the editor, pushed changes to the remote repository, and created and managed branches using VS Code's Git interface.

3.4 Cloning Repositories from GitHub into their Local Development Environment

Making and saving changes to files, followed by commits through the editor.

3.5 Pushing Changes to the Remote Repository

Creating and managing branches using VS Code's Git interface.

3.6 Shared Repository for In-Class Code

A central GitHub repository was created to host the code developed during lectures. Students cloned this repository in VS Code and used it as a base for practical exercises.

3.7 Training in Collaborative Workflow

Students developed collaboration skills by creating and switching between branches in VS Code, resolving merge conflicts using the built-in tools in the Git interface, and creating and reviewing pull requests in GitHub, reinforcing team-based coding practices.

3.8 Submission of Individual Assignments

Students completed 15 individual assignments based on topics from the Data Structures curriculum. In each activity, they modified files locally using Visual Studio Code, performed commits with clear, descriptive messages, and pushed their changes to their personal GitHub repositories.

3.9 Team-Based Evidence Projects

Four learning assessments were implemented to reinforce students' practical skills. Two of these were individual assessments focused on applying version control practices, while the other two were collaborative assignments in which students worked in teams using pull requests and resolved merge conflicts in shared branches, all managed through VS Code.

3.10 Ongoing Monitoring and Feedback

Throughout the course, students' repositories were monitored to assess their progress in managing projects using GitHub and VS Code, organizing and documenting commits, and participating in pull requests and collaborative workflows.

3.11 Final Evaluation and Course Closure

At the end of the course, students' repositories were evaluated based on their effective use of GitHub and VS Code, timely completion and delivery of assignments and evidence projects, quality of commit history and documentation, and teamwork and conflict resolution skills.
This methodology positioned GitHub and Visual Studio Code as core tools throughout the course, ensuring that students acquired practical competencies from the early stages of their academic journey and preparing them to meet the demands of the software development industry. The innovation specifically addresses the lack of hands-on experience with GitHub among students in Computer Science and Information Technologies during their exploration phase, which encompasses the first three semesters of the program. During this phase, students take foundational courses in basic sciences and core computing concepts, laying the groundwork for subsequent stages of their education. The absence of version control skills during this period limits their readiness to collaborate on software projects and misaligns their preparation with industry standards.
By integrating GitHub into the Data Structures course, a set of practical and collaborative activities was implemented to build competencies in version control, teamwork, and repository management.

The methodology included initial training, the use of Visual Studio Code as a support tool, and the submission of assignments through both individual and group repositories, fostering an applied learning experience.
The impact of the intervention was evident in the improvement of practical skills: 100% of students successfully completed activities using GitHub, demonstrating proficiency in commits, pull requests, and collaborative workflows. This experience prepares them for the focus stage of the program, where they develop specialized competencies, and for the specialization stage, which may include professional internships, research stays, or complementary concentrations.
Key outcomes of the intervention include the effective adoption of GitHub as a standard tool, enhanced peer collaboration, and stronger alignment with professional development workflows. Students perceived this innovation as a bridge to industry, reflected in their active participation and positive feedback. This educational proposal addressed a critical gap in the exploration phase and transformed the learning experience by equipping students with essential tools for academic and professional success in the later stages of their education.

4. Survey Results on the Use of GitHub in the Data Structures Course

4.1 Purpose of the Survey

The survey was designed to evaluate the impact of integrating GitHub into the Data Structures course as part of an educational innovation. This initiative aimed to develop practical competencies in the use of industry-standard tools, such as GitHub, to enhance students' technical and collaborative preparedness during the exploration stage of their academic journey. In addition, the survey sought to understand students' perceptions of their learning process, their comfort using the tool, and the future usefulness of the skills acquired.
Two-thirds of the students in the course responded to the survey. All participants were enrolled in Computer Science and Information Technologies programs—specifically, the B.S. in Computer Science and Technology, B.S. in Digital Transformation, and B.S. in Robotics and Digital Systems. As such, the responses reflect the direct experiences of students who actively participated in the implementation of the innovation.

4.2 Survey Structure

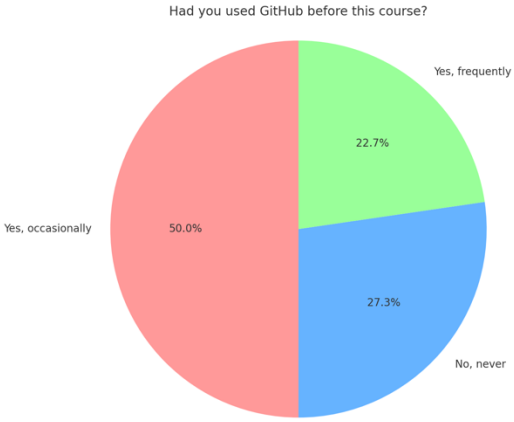
The survey included both closed and open-ended questions to assess the following areas:
Prior Experience. Level of familiarity with GitHub before the course.
Competency Acquisition. Understanding of basic concepts such as version control, commits, and pull requests.
Team Collaboration. Perceived efficiency and effectiveness of collaborative work using GitHub.
Future Utility. Anticipated application of acquired skills in academic and professional projects.
Overall Satisfaction. General opinion of GitHub integration in the course and suggestions for improvement.

5. Results

5.1 Had you used GitHub before this course?

According to the survey, 50% of students had used GitHub occasionally, 22.7% reported frequent use, and 27.3% had never used it before. These results indicate that while a portion of students entered the course with prior exposure to GitHub, the majority required instruction to build foundational knowledge of the tool.

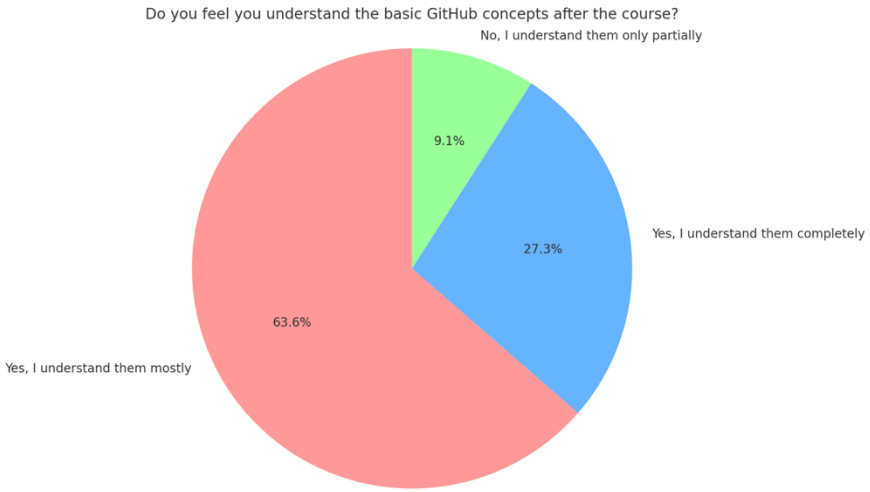
Figure 1. Students' Prior Experience Using GitHub Before the Course



5.2 After this course, do you feel you understand the basic concepts of GitHub (version control, repositories, commits, pulls, etc.)?

According to the survey, 63.6% of students indicated they understood the concepts “mostly,” while 27.3% stated they understood them “completely.” Only 9.1% reported that they understood the concepts “only partially.” These results demonstrate that the course had a strong positive impact on students’ comprehension of GitHub fundamentals, though a small group may still benefit from additional support or practice.

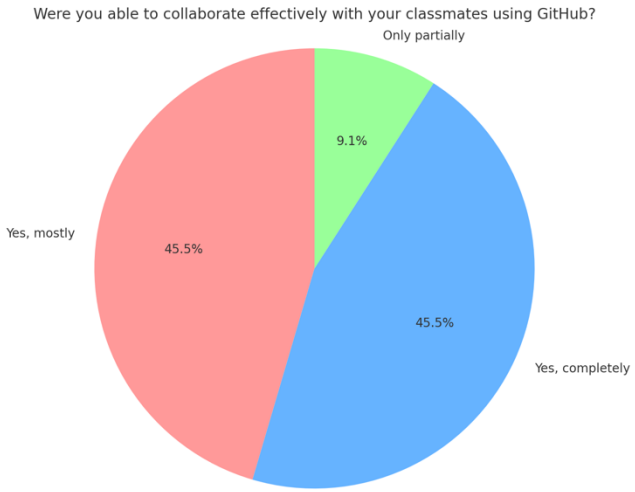
Figure 2. Students’ Understanding of GitHub Concepts After Completing the Course



5.3 During team activities, were you able to collaborate effectively with your classmates using GitHub?

90% of students reported that they were able to collaborate “mostly” or “completely” during team activities. This result indicates that GitHub contributed positively to improving the dynamics of collaborative work.

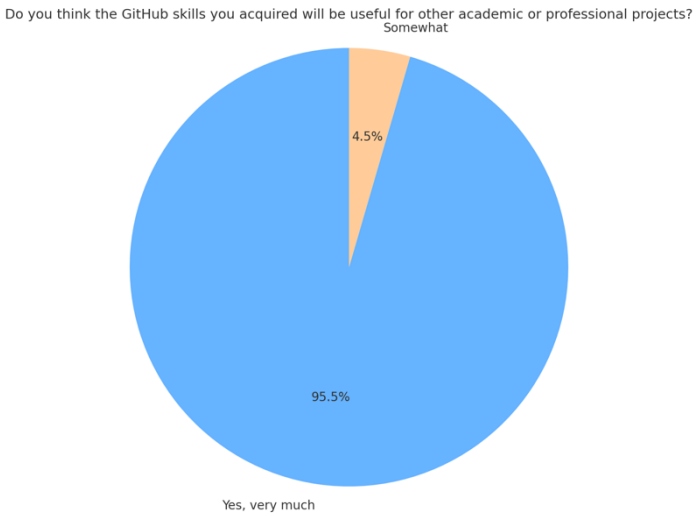
Figure 3. Students’ Perceptions of Collaboration Effectiveness Using GitHub



5.4 Do you think the skills you acquired in GitHub will be useful for other academic or professional projects?

96% of students responded that the skills they learned would be useful for future projects, highlighting the importance of these tools for their academic and professional development.

Figure 5. Students' Perceptions of the Usefulness of GitHub Skills for Future Projects



5.5 What was the most useful thing you learned about GitHub during this course?

Students' responses highlighted a variety of key takeaways from their learning experience with GitHub. The most frequently mentioned topics included:
Version Control. Many students found that understanding how version control works was the most valuable part of the course. They learned how to perform well-organized and documented commits and recognized the importance of maintaining a clear history of project changes.
Collaborative Work. The use of GitHub for team collaboration was cited as one of the most significant learning experiences. Students specifically mentioned skills such as creating and reviewing pull requests and resolving conflicts in shared branches.
Professional Workflow. Some students emphasized that learning to manage repositories and branches gave them exposure to industry-standard practices. This included using GitHub integrated with Visual Studio Code, which facilitated their project management.
Essential Commands. Several responses underscored the importance of mastering basic commands such as pull, push, and merge. Understanding how these actions translate to effective repository management was a key takeaway.

Organization and Documentation. Students learned to document changes using clear and descriptive commit messages. They appreciated the importance of clarity and structure in both individual and group projects.
New Perspectives. A few students expressed that the course changed their perception of GitHub, recognizing it as an indispensable tool in software development. As one student noted, “It’s essential for collaborative work in the industry” and “It simplifies academic and professional projects.”
Conclusion. Most students valued the hands-on, applied approach to learning GitHub, noting its impact on their understanding of collaborative work and project organization. This experience better prepared them to handle real-world workflows in both academic and professional contexts.

5.6 What aspects do you think could be improved when teaching GitHub in this course?

Student feedback identified several opportunities to improve GitHub instruction. The most common suggestions included:
More Depth in Advanced Concepts. Students suggested dedicating more time to advanced topics such as branch management and conflict resolution. Some proposed adding practical examples on the use of issues and best practices for managing collaborative projects.
Dedicated Introductory Sessions. Many students requested a fully dedicated introductory class, especially for those with no prior experience.
They recommended covering GitHub basics and initial configurations, including integration with Visual Studio Code.
More Realistic Exercises and Examples. Students recommended adding more practice-based activities inspired by real-world scenarios or larger projects to apply full GitHub workflows.
A common class project was also suggested to reinforce collaborative concepts in a realistic context.
Supplementary Resources. Some students asked for more support materials such as video tutorials and step-by-step guides to review outside of class.
Others recommended a centralized repository with detailed examples of each concept covered.
Adjusting Course Pacing. Some feedback indicated that key topics were covered too quickly, which hindered full comprehension.
Students suggested adjusting the pace and including dedicated Q&A sessions to clarify complex concepts.
Encouraging Autonomous Exploration. Several students proposed offering optional advanced tasks for those interested in exploring topics like GitHub Actions or advanced repository configurations.
Conclusion. While students appreciated learning GitHub in the course, they identified areas for improvement including deeper exploration of advanced concepts, more applied practice, and adjustments in teaching pace to support learners with varying levels of experience. These suggestions provide valuable insights for enhancing future course iterations.

5.7 Would you recommend integrating GitHub into other courses in the program? Why?

Most students responded positively regarding the integration of GitHub into other courses within the academic program. Their responses emphasized the following points:
An Essential Tool for Collaboration. Many students recommended incorporating GitHub into more courses due to its usefulness in team collaboration and project organization.
They highlighted how GitHub supports task distribution, progress tracking, and integration of contributions, significantly enhancing group dynamics.
Preparation for the Job Market. Students stressed that GitHub is an industry-standard tool, and that learning to use it across multiple courses would better prepare them for future professional challenges.
They argued that gaining experience with GitHub from various academic perspectives would increase their competitiveness in the job market.
Cross-Disciplinary Applications. Several responses noted that GitHub can be useful not only in programming courses, but also in interdisciplinary projects requiring version control and documentation.
They suggested integrating GitHub into subjects such as web development, databases, and software design to enhance practical learning.
Fostering Professional Habits. Students emphasized that using GitHub across courses would help develop professional habits such as documentation, branching strategies, and conflict resolution.
Continuous use would reinforce learning and make these practices second nature.

Implementation Recommendations. Some students recommended a gradual integration, starting with basic concepts in early courses and advancing to more complex functions in later ones. Others proposed collaborative projects spanning multiple courses to give GitHub use greater relevance and context.
Conclusion. Most students would recommend integrating GitHub into more courses across the curriculum, highlighting its academic and professional value. They believe a cross-disciplinary implementation would not only strengthen technical skills but also better prepare them for collaborative environments in the software industry. This feedback underscores the importance of GitHub as a fundamental tool in computing and information technologies education.

6. Conclusion

The innovation successfully achieved its stated objectives. Students acquired practical skills in GitHub and experienced its application in collaborative project settings. Survey responses demonstrated significant progress in understanding key concepts such as version control, branching, and pull requests, while also improving their ability to work effectively in teams.
Furthermore, the results suggest that GitHub is perceived as a vital tool for preparing students to meet the demands of the software industry. A substantial majority of students reported increased confidence in using GitHub and expressed a strong likelihood of applying these skills in future academic and professional projects. Their reflections confirmed that GitHub not only enhanced technical competencies but also facilitated the development of essential habits such as documentation, organization, and code management.
Student feedback also provided valuable insights for improving future iterations of the course. Suggestions included dedicating more time to advanced Git workflows, offering realistic project-based activities, adjusting the pace for better comprehension, and integrating GitHub into other relevant courses. These recommendations highlight the importance of aligning instruction with diverse learning needs and evolving industry practices.
Overall, the implementation of GitHub as a core learning tool transformed the student experience in the Data Structures course. It bridged the gap between theoretical knowledge and real-world software development practices, laying a strong foundation for students to succeed in later academic stages and in their professional careers. This experience reinforces the importance of incorporating industry-relevant tools early in the curriculum and serves as a replicable model for computing programs seeking to modernize and enhance their teaching practices.

7. Acknowledgements

The authors would like to acknowledge the financial and the technical support of Writing Lab, Institute for the Future of Education, Tecnologico de Monterrey, Mexico, in the production of this work.

REFERENCES

[1] Wolf, G. Use of the Git version control system as a replacement for a learning management system. IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, 2024, 19, 24–32.
[2] Escobar Jurado, S. I., Egea Arciniegas, T. G., Leal Bolaño, N. L., & Aguilera Cuenca, H. Estilos de aprendizaje: Tendencias y ruta para cualificar la práctica pedagógica. Sello Editorial UNAD. 2021
[3] Von Feigenblatt, O. F., Peña-Acuña, B., & Cardoso-Pulido, M. J. Aprendizaje personalizado y education maker: Nuevos paradigmas didácticos y otras aproximaciones. Ediciones Octaedro. 2022.