

# A cloud-accessible virtual lab architecture for teaching network automation: Integrating Cisco packet tracer REST API with docker, Cloudflare tunnel, and Google Colab

 Rodrigo Fernando Morocho Román<sup>1\*</sup>

<sup>1</sup>Technical University of Machala, Machala, Ecuador; rmorocho@utmachala.edu.ec (R.F.M.R.).

**Abstract:** This study designed and evaluated a reproducible, cloud-accessible virtual laboratory architecture for teaching network automation in higher education. The goal was to address the lack of cost-effective and scalable infrastructures that enable students to develop automation skills aligned with current industry demands. The proposed design integrated Cisco Packet Tracer's REST API, Docker containerization, Cloudflare Tunnel, and Google Colab to facilitate real-time programmatic interaction with simulated networks through Python. A design-based research (DBR) methodology was adopted, and a pilot was conducted with 35 undergraduate students enrolled in the "Network Management" course at the Technical University of Machala (Ecuador). The intervention consisted of three structured labs, a pre-test/post-test assessment, and a perception survey. Findings revealed statistically significant learning gains, with post-test scores showing marked improvement over pre-test results. Students also reported high satisfaction and engagement, highlighting the accessibility and novelty of combining cloud platforms with network simulators. The study concludes that this architecture provides a feasible and pedagogically sound model for integrating network automation into curricula. Practical implications include lowering barriers to access, fostering active learning, and offering institutions a scalable, low-cost alternative to physical labs.

**Keywords:** *Cisco packet tracer, Cloud-based education, Constructivist learning, Docker, Google Colab, network automation, REST APIs, Virtual laboratories.*

## 1. Introduction

The increasing complexity of modern networks and the widespread adoption of programmable infrastructures have created a growing demand for professionals skilled in network automation. As organizations transition toward intent-based networking and DevOps practices, educational institutions face the challenge of updating curricula to include hands-on training in automation, scripting, and software-defined infrastructures [1, 2]. However, deploying physical network laboratories with automation capabilities remains cost-prohibitive and logistically challenging, particularly for institutions with limited resources [3].

Virtual laboratories have emerged as a scalable and accessible alternative to physical labs. Tools such as Cisco Packet Tracer allow students to simulate network devices and protocols in an interactive environment without requiring hardware [4]. Nonetheless, while Packet Tracer is widely adopted, it has traditionally lacked built-in support for automation workflows. The introduction of a REST API in recent versions opens new opportunities for integrating automation into network simulations [5].

To execute automation scripts in an accessible environment, Google Colab offers a cloud-based platform where learners can write and run Python code directly from a browser. With built-in support for Python libraries such as *requests*, it is possible to consume APIs, process JSON data, and visualize

results without installing additional software [6, 7]. This makes Colab particularly attractive for educational contexts seeking to lower entry barriers for students.

A major limitation in connecting local simulators like Packet Tracer with cloud platforms is the absence of public IP addresses or open ports in typical classroom or home environments. To address this, technologies such as Cloudflare Tunnel can securely expose local services over the internet. When containerized with Docker, this setup becomes portable, reproducible, and easy to deploy, providing a complete cloud-accessible automation lab [8, 9].

Despite the potential of each individual component, limited research has explored the pedagogical impact of integrating Packet Tracer, Docker, Cloudflare Tunnel, and Google Colab into a unified and reproducible virtual lab architecture. Furthermore, most implementations emphasize technical feasibility rather than educational effectiveness or alignment with learning theories.

This study addresses that gap by designing a constructivist-aligned, cloud-accessible laboratory that enables students to interact programmatically with virtual networks using Python and REST APIs. A pilot was conducted with 35 undergraduate students from the Technical University of Machala, incorporating three structured lab sessions, a pre-test/post-test evaluation, and a perception survey to assess learning outcomes and student engagement.

The main contributions of this work are:

1. The design and deployment of an integrated laboratory using widely available and cost-free technologies.
2. Quantitative evidence of improved learning outcomes after structured automation labs.
3. Qualitative analysis of student perceptions regarding usability, relevance, and satisfaction.

Ultimately, the proposed approach democratizes access to modern networking education and equips students with practical skills aligned with current industry trends.

## 2. Literature Review

### 2.1. Network Virtualization and Automation in Education

The shift from hardware-centric to software-defined networking has reshaped how networks are designed, deployed, and managed. As a result, educational institutions are expected to adapt their curricula to prepare students for programmable networks and automation-driven environments. Network virtualization provides a foundation for scalable, reusable, and cost-effective laboratory environments where students can simulate real-world scenarios without the constraints of physical devices [10].

Tools such as Cisco Packet Tracer, GNS3, and EVE-NG are widely used in networking education. These platforms enable students to build, configure, and test network topologies in a controlled and safe environment [11]. While these simulators are essential, they often require complementary approaches to foster deeper engagement with automation and programming concepts [12].

Recent meta-analytic evidence underscores the substantial pedagogical benefits of virtual laboratory interventions. Li and Liang [13] reported a significant effect size (Hedges'  $g \approx 0.686$ ), with particularly strong impacts on student motivation ( $g \approx 3.571$ ) and engagement ( $g \approx 2.888$ ). These findings highlight the role of virtual labs not only as supplementary tools but also as demonstrably effective drivers of learning outcomes.

In this context, network automation is emerging as a critical competency. The adoption of DevOps practices in network operations—such as infrastructure as code (IaC), automated provisioning, and continuous monitoring—requires educational approaches that move beyond configuration tasks and embrace programmable interfaces and scripting [14]. Integrating such practices into teaching fosters both technical proficiency and an automation-oriented mindset, which are essential in NetDevOps roles [1].

## 2.2. REST APIs and JSON in Network Learning

REST (Representational State Transfer) APIs are the de facto standard for interacting with network devices and services. Unlike traditional CLI-based configuration, REST APIs provide programmatic access to configuration, monitoring, and automation functions. This promotes reproducibility and scalability [15]. In educational settings, introducing REST APIs enables students to interact with network elements in a structured and abstracted manner, aligning with modern software practices.

JSON (JavaScript Object Notation) is the preferred data interchange format in REST interactions. Its simplicity, readability, and compatibility with programming languages such as Python make it ideal for teaching structured data representation and processing. When students learn to parse and manipulate JSON using Python libraries (e.g., *json*, *requests*), they acquire essential skills for working in programmable environments and bridging the gap between network configuration and software development [16, 17].

From a pedagogical perspective, REST interaction encourages a constructivist approach to learning, where students actively explore, build, and troubleshoot through real API calls, thereby reinforcing theoretical concepts through experimentation [18].

## 2.3. Cloud-Based Learning Platforms and Reproducibility

Cloud-based platforms are increasingly relevant in education, as they provide on-demand access to computing resources without requiring local infrastructure. Google Colab, in particular, offers a browser-based environment where students can run Python code, access preinstalled libraries, and collaborate in real time [6]. This reduces technical barriers for learners and provides a consistent environment across different devices [7].

Containerization technologies such as Docker further enhance reproducibility by packaging applications together with their dependencies. This ensures that experiments and learning activities can be consistently replicated across operating systems and institutions [19]. In networking education, containerized environments allow educators to distribute preconfigured lab setups that students can launch quickly and uniformly [9].

Recent studies emphasize that cloud-based deployment of networking labs significantly improves scalability and accessibility. Al-Zubaidie, et al. [20] demonstrated how frameworks using Netkit with AWS EC2 enhance portability, reproducibility, and on-demand access for students, while also reducing hardware barriers and improving institutional efficiency.

Combining Google Colab with tunneling tools like Cloudflare Tunnel and simulators like Packet Tracer creates a novel pedagogical model that bridges local and remote resources. This integration enables real-time interaction with simulations running on local machines while maintaining the usability and shareability of cloud platforms. Few studies have systematically explored this combination in educational contexts, and even fewer have evaluated its impact on student learning outcomes and engagement.

## 2.4. Pedagogical Foundations and Virtual Labs

The design of technology-enhanced learning environments should be grounded in educational theory. Constructivist frameworks, such as those proposed by Piaget and Vygotsky, emphasize active learning, discovery, and knowledge construction through interaction and feedback [18]. Virtual labs align with these principles by offering exploratory environments where students experiment, reflect, and iterate.

Furthermore, interactive notebooks like those used in Colab support guided inquiry, a key component of constructivist pedagogy [21]. Through immediate feedback, visualization, and step-by-step experimentation, students build conceptual understanding and computational thinking simultaneously.

More recent contributions extend this paradigm by integrating advanced technologies into virtual laboratories. Terkaj, et al. [22] proposed a comprehensive framework that incorporates Digital Twin architectures and XR technologies to create immersive, safe, and cost-effective environments in engineering education. Similarly, Sellberg, et al. [23] conducted a scoping review of synthetic virtual laboratories across STEM disciplines, concluding that these models promote inclusivity and active learning while reshaping traditional pedagogical practices.

In summary, the convergence of simulation tools, cloud platforms, and containerized environments provides an unprecedented opportunity to democratize access to modern network education. However, there remains a research gap in evaluating integrated lab models that combine all these components in a cohesive, pedagogically grounded strategy.

### 3. Methodology

This study adopted a design-based research (DBR) approach to iteratively develop, implement, and evaluate a virtual laboratory environment aimed at teaching network automation and virtualization. DBR is well-suited for educational technology research as it allows the exploration of innovative instructional strategies in real-world learning settings [24, 25]. The goal was to assess both the pedagogical effectiveness and technical feasibility of a cloud-accessible, reproducible lab infrastructure using Cisco Packet Tracer, Docker, Cloudflare Tunnel, and Google Colab.

#### 3.1. Context and Participants

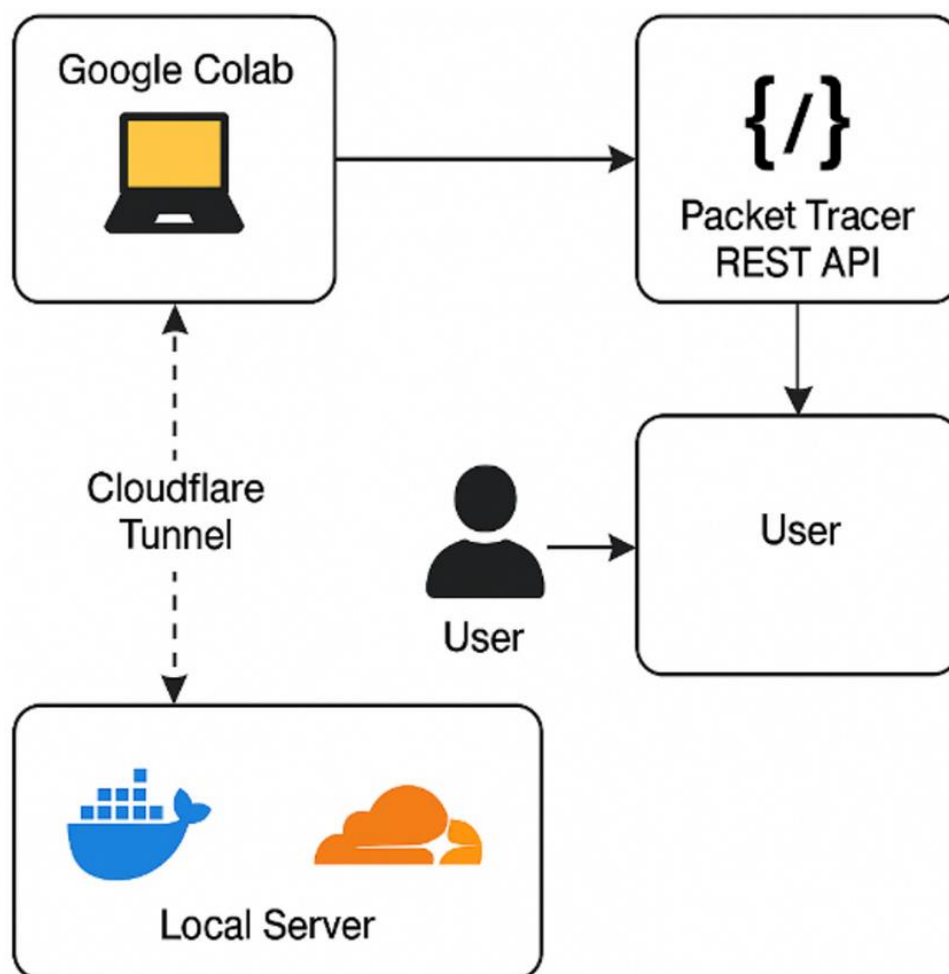
The pilot was conducted with 35 undergraduate students enrolled in the "Network Management" course of the Information Technology program at Universidad Técnica de Machala, Ecuador. The students had completed foundational coursework in networking and were familiar with tools such as Packet Tracer. This intervention was part of a scheduled module on "Virtualization and Automation of Networks" and constituted a core component of the course's practical activities.

#### 3.2. Architecture of the Virtual Lab

The laboratory architecture integrated four technologies to allow remote interaction with a simulated network environment:

- Cisco Packet Tracer 8.2.2, running locally, exposed its REST API to allow automation via HTTP requests [4].
- Docker containerized a lightweight Cloudflare Tunnel client to securely expose the local Packet Tracer API to the internet without requiring port forwarding or static IPs.
- Cloudflare Tunnel, a secure tunneling service, bridged Google Colab and the local simulation by creating a public endpoint that redirected traffic to the internal API [8].
- Google Colab was used as the cloud interface where students executed Python code to send and receive requests from the simulation.

This setup was reproducible, platform-independent, and required no local installation beyond Packet Tracer and Docker, making it highly suitable for scalable deployment. The architecture is illustrated in Figure 1.



**Figure 1.**  
Architecture of the Virtual Lab for Network Automation.

**Note:** The figure illustrates the integration of Cisco Packet Tracer REST API with Docker, Cloudflare Tunnel, and Google Colab to enable cloud-accessible and reproducible network automation labs.

The local execution of Cisco Packet Tracer and the Docker-based Cloudflare Tunnel was performed on a machine running Windows 11 Pro. The hardware configuration included a 12th-generation Intel Core i7 processor, 16 GB of RAM, and a 512 GB SSD. Although the laboratory was tested in a single operating system environment (Windows), the components involved—Docker, Cloudflare Tunnel, and the Packet Tracer REST API—are compatible with other platforms such as Linux and macOS. No performance issues were observed during the pilot with 35 undergraduate students, suggesting that the proposed architecture is scalable and stable for medium-scale educational deployments. Nonetheless, instructors intending to reproduce this setup should ensure that their local machines meet at least similar specifications for optimal performance.

### 3.3. Instructional Design and Activities

The intervention was structured into three progressive labs, each implemented as a Google Colab notebook:

1. Device Discovery: Students used HTTP GET methods to query connected network devices and retrieve interface data in JSON format.

2. Data Parsing and Filtering: Students learned to extract and process specific fields from API responses using Python's json module, reinforcing data handling and control flow.
3. Automation and Error Handling: Participants scripted automated configurations (e.g., assigning IP addresses) and added error handling mechanisms based on API status codes and messages.

Each notebook included narrative explanations, code cells, example outputs, and challenges that aligned with the learning outcomes of the course. The pedagogical strategy emphasized active learning and problem-solving, consistent with constructivist principles [18].

### 3.4. Evaluation Design

A mixed-methods evaluation was employed to assess the impact of the intervention from both quantitative and qualitative perspectives [26]:

- Pre-test and post-test assessments were applied to measure learning gains. Both consisted of 10 multiple-choice questions focused on key concepts such as virtualization, REST APIs, JSON, Docker, and automation.
- A perception survey using a 5-point Likert scale captured students' experiences regarding usability, engagement, and pedagogical value. The survey included five closed questions and two open-ended items for qualitative feedback.

The questionnaire items were validated through expert review and aligned with the intended learning outcomes.

### 3.5. Data Collection and Analysis

- Quantitative data from the tests were analyzed using descriptive statistics and a paired-sample **t-test** to determine whether the differences in pre- and post-test scores were statistically significant [27].
- Survey responses were analyzed to identify trends in student perception. Frequencies, means, and standard deviations were computed.
- Open-ended responses were coded and categorized using thematic analysis to extract student insights and recurring opinions [28].

The combination of objective scores and subjective perception allowed for a triangulated evaluation of the virtual lab's pedagogical impact.

## 4. Results and Discussion

### 4.1. Overview of the Pilot Study

The proposed virtual laboratory architecture was deployed and tested during a four-week pilot intervention with 35 undergraduate students. The instructional design included three practical labs and was complemented by a pre-test, post-test, and a perception survey. This structure enabled a holistic evaluation of the students' cognitive gains, engagement, and usability perception of the environment.

### 4.2. Quantitative Evaluation of Learning Gains

#### 4.2.1. Pre-test and Post-test Analysis

The pre- and post-tests included identical items designed to assess knowledge of virtualization concepts (e.g., containers, cloud tunnels), network automation (e.g., REST APIs, JSON), and the use of Google Colab for scripting tasks.

- Mean Pre-test Score: 3.11 out of 10 (SD = 1.23)
- Mean Post-test Score: 7.34 out of 10 (SD = 1.12)

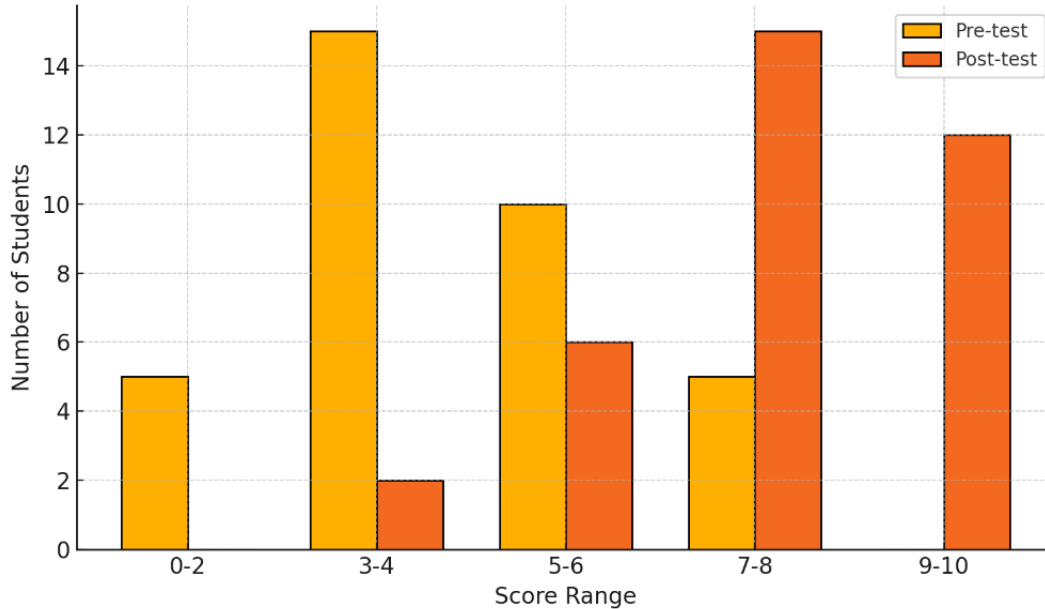
The learning gain was statistically significant, as confirmed by a paired-sample t-test:

$$t(34) = 15.21, p < 0.001$$

This result suggests that students significantly improved their conceptual understanding of network automation and virtualization. The high effect size indicates a strong instructional impact [29].

#### 4.2.2. Score Distribution and Performance Trends

Figure 2 shows the distribution of individual learning gains. Over 85% of students improved their scores by three or more points, with more than half scoring 8 or above on the post-test. This shift reflects deepened understanding and effective knowledge transfer, especially considering the students' initial unfamiliarity with Docker, JSON, and REST APIs.



**Figure 2.**

Distribution of Pre-test and Post-test Scores among Students.

**Note:** Data represent results from 35 undergraduate students. Scores improved significantly from the pre-test ( $M = 3.11$ ,  $SD = 1.23$ ) to the post-test ( $M = 7.34$ ,  $SD = 1.12$ ).

These findings align with studies indicating that active, notebook-based experimentation improves learning in computing disciplines [21, 30].

#### 4.3. Analysis of Student Perceptions

The post-lab survey, composed of five Likert-scale questions and two open-ended items, revealed strong agreement with key aspects of the learning experience (scale: 1 = Strongly Disagree, 5 = Strongly Agree):

**Table 1.**

Student Perceptions of the Virtual Lab (Likert-scale Responses).

Statement	Mean	SD
The labs helped me understand network automation concepts.	4.68	0.52
The integration of Google Colab and Packet Tracer was engaging.	4.77	0.43
Docker and Cloudflare Tunnel usage was well explained.	4.60	0.56
I would recommend this approach for future courses.	4.82	0.39
The labs motivated me to explore DevOps and automation.	4.74	0.48

**Note:** The table summarizes students' ratings ( $N = 35$ ) on a 5-point Likert scale regarding their learning experience with the virtual lab. Higher values indicate stronger agreement.

The highest-rated item was the recommendation of the method (Q4), indicating that students found the experience highly valuable and replicable. Open-ended responses emphasized **ease of access**, hands-on exploration, and the novelty of integrating cloud-based tools with network simulation.



*“It was the first time I worked with APIs and Python in this way, and I really liked seeing the responses live on Colab.”*

*“Now I understand what REST is, and I can even explain it to others.”*

These insights support the notion that combining simulation with live coding environments fosters engagement and ownership of learning [18].

#### 4.4. Pedagogical Implications

The success of this intervention suggests that interactive virtual labs can address key challenges in teaching network automation:

- **Bridging theory and practice:** Students applied abstract concepts in real-time through scripting and parsing API responses.
- **Reducing barriers to access:** The cloud-based design eliminated the need for specialized local installations.
- **Fostering critical skills:** Learners developed not only technical knowledge but also troubleshooting, data interpretation, and API documentation reading abilities.

These outcomes align with constructivist theories that advocate active, situated, and exploratory learning environments [18, 31]. Furthermore, this model supports the call for integrating DevOps principles into educational curricula to better reflect industry expectations [32].

#### 4.5. Limitations and Considerations for Future Design

Despite positive results, several limitations must be noted:

- **Limited API features:** Cisco Packet Tracer’s REST API offers basic functionality and does not support advanced features like routing protocol configuration or VLAN segmentation.
- **Dependency on internet connectivity:** In areas with poor bandwidth or network filtering, students may face tunnel instability or latency issues.
- **Scalability constraints:** Although no scalability issues were observed during the pilot with 35 students, Docker containers and Cloudflare tunnels require local resource allocation, which could potentially challenge instructors managing larger-scale or concurrent sessions in different environments.

Addressing these issues could involve enhancing the API, offloading processing to cloud VMs, or adopting more advanced simulation platforms (e.g., GNS3 or EVE-NG) with broader automation capabilities (Sharma et al., 2019).

Another limitation of this study is the absence of a control group. Although the design-based research (DBR) approach focuses on iterative refinement and real-world implementation, the lack of a control group may limit the ability to attribute learning gains solely to the intervention. Future studies could incorporate quasi-experimental designs to strengthen causal inferences.

## 5. Conclusions and Future Work

This study presented the design, implementation, and evaluation of a virtual, reproducible, and cloud-accessible laboratory environment to teach network automation and virtualization. The architecture integrated widely available tools—Cisco Packet Tracer, Google Colab, Docker, and Cloudflare Tunnel—to enable students to interact programmatically with simulated network devices through REST APIs and JSON data structures.

The pilot deployment with 35 undergraduate students demonstrated statistically significant learning gains, with post-test scores markedly higher than pre-test scores. The perception survey revealed high levels of satisfaction, particularly in aspects related to accessibility, engagement, and innovation. Students expressed enthusiasm toward the hands-on format and confidence in their ability to apply automation concepts in real-world scenarios.



From a pedagogical standpoint, this approach aligns with constructivist and experiential learning theories, emphasizing active experimentation, feedback, and reflection [18, 33]. The use of interactive Python notebooks in Colab encouraged discovery learning, enabling students to engage deeply with network configuration tasks, data parsing, and API-based automation. Moreover, the reproducibility of the environment through Docker containers and public repositories supports the growing emphasis on open, scalable, and cost-effective educational practices [19].

Nonetheless, the study also revealed practical constraints that must be considered in future deployments:

- The limited scope of the Packet Tracer REST API, which restricts complex automation scenarios.
- Tunnel reliability issues, particularly in constrained or filtered network environments.
- Reliance on internet connectivity, which can be a barrier in rural or under-resourced regions.

These limitations do not detract from the value of the intervention but rather inform the next iteration of this educational model.

### 5.1. Future Work

Building on the success of this pilot, several directions are proposed for future research and development:

1. Extension of Laboratory Content: Future labs will include advanced topics such as dynamic routing protocols, VLAN configuration, and automated troubleshooting logic using conditional API flows.
2. Integration with Advanced Simulation Platforms: Tools like GNS3 or EVE-NG, which support multi-vendor simulation and richer API interfaces, will be evaluated to complement or replace Packet Tracer where appropriate [10].
3. Scalability and Performance Testing: Larger-scale pilots will be conducted with expanded cohorts and multiple instructors to assess load balancing, tunnel stability, and orchestration at scale.
4. Cross-Institutional Collaborations: Partnerships with universities across Latin America will allow for comparative studies and validation of generalizability, helping to promote inclusive and regional network automation training.
5. Longitudinal Studies on Learning Outcomes: Follow-up studies will track the same cohort's performance in subsequent DevOps or cybersecurity courses to assess retention, skill transfer, and career impact.
6. Alignment with Pedagogical Frameworks: The integration of theoretical models such as the TPACK framework Mishra and Koehler [34] and SAMR model [35] will help articulate the instructional design principles that underpin the success of this model and guide its refinement.

### 5.2. Final Remarks

In an era where automation, virtualization, and cloud computing are redefining the landscape of IT and network engineering, education must evolve accordingly. The proposed virtual lab architecture offers a compelling blueprint for how modern network education can be delivered in a reproducible, cost-effective, and engaging way—regardless of institutional infrastructure or geographic constraints. This work contributes to a growing body of research advocating for democratized access to quality technical education through open-source tools and cloud technologies.

By continuing to refine the technical platform and align it more closely with robust instructional design theory, future iterations of this model hold the potential to shape a new standard in hands-on, scalable network education.

### Funding:

This research was conducted as part of a research project at the Technical University of Machala. The study did not receive external funding.

### Institutional Review Board Statement:

The study was conducted in accordance with the ethical standards of the Technical University of Machala. Formal ethical approval was not required for this type of educational research, as it involved standard classroom activities without sensitive personal data. All participants were informed about the purpose of the study and voluntarily agreed to participate.

### Transparency:

The author confirms that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

### Copyright:

© 2025 by the author. This open-access article is distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

### References

- [1] R. Yalamanchili and G. Kiran, "An educational framework for network automation using DevNet and cloud services," in *Proceedings of the 2022 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 2022.
- [2] B. Halabi and J. Williams, *Network automation and programmable networks: A practical guide to modern network infrastructure*. Indianapolis, IN: Cisco Press, 2021.
- [3] C. Cicconetti, L. Fanucci, P. Pagano, and T. Pecorella, "On the use of virtual labs for teaching computer networks," in *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON)*, 2020, doi: <https://doi.org/10.1109/EDUCON45650.2020.9125282>.
- [4] Cisco, "Cisco packet tracer REST API overview," Cisco Networking Academy, 2022. <https://www.netacad.com/>
- [5] R. Saini and S. Patel, "Automation of network simulation using REST API in cisco packet tracer," *International Journal of Computer Applications*, vol. 184, no. 25, pp. 1–6, 2022.
- [6] E. Bisong, *Google colab*. Berkeley, CA: Springer, 2019, pp. 59–64. [https://doi.org/10.1007/978-1-4842-4470-8\\_7](https://doi.org/10.1007/978-1-4842-4470-8_7)
- [7] A. Nelson and C. Hoover, "Building python notebooks with google colab for teaching data analysis," *Journal of Statistics Education* vol. 28, no. 2, pp. 151–158, 2020.
- [8] Cloudflare, "Expose your locally running web server to the internet with Cloudflare Tunnel," 2023. <https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/>
- [9] G. Kurt and Y. Yildirim, "Virtualization in network education: Using Docker containers for reproducible labs," *Education and Information Technologies*, vol. 27, pp. 825–841, 2022.
- [10] L. Zhang, J. Wang, and Y. Liu, "A hybrid simulation environment for teaching computer networks," *IEEE Transactions on Education*, vol. 63, no. 1, pp. 32–38, 2020.
- [11] A. Sharma, P. Gupta, and R. Rajput, "Comparative analysis of GNS3, Cisco packet tracer and EVE-NG for computer networks simulation," *International Journal of Computer Applications*, vol. 178, no. 7, pp. 32–36, 2019. <https://doi.org/10.5120/ijca2019918803>
- [12] S. P. Pereira, R. Queirós, and J. P. Leal, "An approach for teaching network automation with real-time simulation," in *Proceedings of the 13th International Conference on Computer Supported Education (CSEDU) (Vol. 2, pp. 190–197)*. SCITEPRESS, 2021.
- [13] J. Li and W. Liang, "Effectiveness of virtual laboratory in engineering education: A meta-analysis," *PloS one*, vol. 19, no. 12, p. e0316269, 2024. <https://doi.org/10.1371/journal.pone.0316269>
- [14] W. Chen, Y. Song, and D. Wang, "A practical teaching model of network automation based on DevOps," *Proceedings of the 2021 IEEE Frontiers in Education Conference (FIE)*, 2021, doi: <https://doi.org/10.1109/FIE49875.2021.9637242>.
- [15] F. Khan, H. Khan, and A. Baig, "Impact of REST APIs in software-defined networking," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 543–549, 2022. <https://doi.org/10.14569/IJACSA.2022.0130668>
- [16] T. Bray, "The javascript object notation (json) data interchange format," Internet Engineering Task Force, 2070–1721, 2014. <https://doi.org/10.17487/RFC8259>
- [17] R. Singh and V. Nath, "Learning API data handling with JSON using python," presented at the Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020.
- [18] D. Laurillard, *Teaching as a design science: Building pedagogical patterns for learning and technology*. New York: Routledge, 2012.

- [19] K. Grodecka and M. Jakubowski, "Reproducibility in teaching programming and data science: The role of containerization," *Education Sciences*, vol. 11, no. 12, p. 767, 2021. <https://doi.org/10.3390/educsci11120767>
- [20] M. Al-Zubaidie, R. A. Muhajjar, and L. A. Shihabe, "Computer networking and cloud-based learning/teaching environment using virtual labs tools: A review and future aspirations," *Mesopotamian Journal of Computer Science*, vol. 2024, pp. 176-203, 2024.
- [21] A. Rule, A. Tabard, and J. D. Hollan, "Exploration and explanation in computational notebooks," *ACM Computing Surveys*, vol. 53, no. 1, pp. 1-36, 2020.
- [22] W. Terkaj, E. Pessot, V. Kuts, Y. Bondarenko, S. L. Pizzagalli, and K. Kleine, "A framework for the design and use of virtual labs in digital engineering education," in *AIP Conference Proceedings*, 2024, vol. 2989, no. 1, p. 030003, doi: [https://doi.org/10.1007/978-3-031-40865-8\\_21](https://doi.org/10.1007/978-3-031-40865-8_21).
- [23] C. Sellberg, Z. Nazari, and M. Solberg, "Virtual laboratories in STEM higher education: A scoping review," *Nordic Journal of STEM Education*, vol. 8, no. 1, pp. 34-52, 2024.
- [24] T. Anderson and J. Shattuck, "Design-based research: A decade of progress in education research?," *Educational Researcher*, vol. 41, no. 1, pp. 16-25, 2012. <https://doi.org/10.3102/0013189X11428813>
- [25] F. Wang and M. J. Hannafin, "Design-based research and technology-enhanced learning environments," *Educational Technology Research and Development*, vol. 53, no. 4, pp. 5-23, 2005. <https://doi.org/10.1007/BF02504682>
- [26] J. W. Creswell and V. L. Plano Clark, *Designing and conducting mixed methods research*. Thousand Oaks, CA: SAGE Publications, 2018.
- [27] A. Field, *Discovering statistics using IBM SPSS statistics*. Thousand Oaks, CA: SAGE Publications, 2018.
- [28] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77-101, 2006. <https://doi.org/10.1191/1478088706qp0630a>
- [29] J. Cohen, *Statistical power analysis for the behavioral sciences*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.
- [30] J. M. Perkel, "Why jupyter is data scientists' computational notebook of choice," *Nature*, vol. 563, no. 7732, pp. 145-147, 2018.
- [31] L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press, 1978.
- [32] K. Mohan, A. Jain, and S. Raju, "Teaching DevOps and agile principles to university students: A pilot experience," *ACM Transactions on Computing Education*, vol. 20, no. 4, p. 21, 2020.
- [33] D. A. Kolb, *Experiential learning: Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall, 1984.
- [34] P. Mishra and M. J. Koehler, "Technological pedagogical content knowledge: A framework for teacher knowledge," *Teachers College Record*, vol. 108, no. 6, pp. 1017-1054, 2006. <https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- [35] R. R. Puentedura, "SAMR: A contextualized introduction," SAMR: A Contextualized Introduction, 2014.