



Mininet-GUI: Uma Abordagem Visual e Interativa para Experimentação em Redes SDN

Lucas M. Schneider¹, Emídio P. Neto¹, Felipe S. Dantas Silva¹

¹LaTARC Research Lab

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN)
Natal, RN – Brasil.

lucasschneider.dev@gmail.com, zshemidio@gmail.com, felipe.dantas@ifrn.edu.br

Abstract. *Mininet is a widely adopted emulator for prototyping and experimenting with Software-Defined Networking (SDN). Its main graphical interface, MiniEdit, however, offers limited functionality and low interactivity, which hampers adoption and constrains advanced experimentation. Although several alternative tools have been proposed, none has become a definitive replacement, largely due to the absence of essential integrated features. We present Mininet-GUI, a web-based tool that enables real-time creation, editing, and execution of Mininet topologies. Mininet-GUI allows dynamic manipulation of hosts, switches, controllers, and links, as well as direct command execution through the interface. Unlike existing approaches, it provides interactive topology editing, simplified terminal access via WebShell, and support for exporting and importing topologies in JSON and Python formats. Mininet-GUI is designed to make SDN experimentation more accessible and efficient for both beginners and advanced researchers. This paper describes its architecture, capabilities, and applicability through common use cases.*

1. Introdução

No âmbito da prototipação e experimentação de soluções para ecossistemas de redes de próxima geração, a exemplo do 5G/6G, o Mininet [Lantz et al. 2010] consolidou-se como ferramenta de referência para emulação de redes programáveis. O Mininet é amplamente reconhecido pela comunidade acadêmica e pela indústria por ser um habilitador para a criação ágil de ambientes de alta fidelidade, viabilizando a rápida experimentação de novos conceitos, protocolos e arquiteturas de Redes Definidas por Software (SDN) [Haleplidis et al. 2015].

A adoção prática de SDN pode ser maximizada por meio de interfaces intuitivas que reduzam as barreiras de entrada para o seu aprendizado. Entretanto, esse processo geralmente requer o uso combinado de ferramentas distintas e configurações manuais [Cosgrove 2016, Bouras et al. 2017, Salib and Lester 2018]. O próprio Mininet é distribuído com o MiniEdit, uma interface gráfica integrada que possibilita a criação e configuração de cenários de redes SDN. No entanto, o MiniEdit limita-se somente a funcionalidades básicas, como a criação de topologias e a configuração dos dispositivos, sem suporte a linha de comando ou scripts do Mininet, essenciais para experimentações avançadas. Essas limitações impulsionaram o desenvolvimento de soluções alternativas, como o Mininet Editor [Vyčítal 2019] e o Integrated [Barlowe et al. 2023], que introduzem novas perspectivas de interação do usuário através de uma interface web.

Um levantamento recente do estado da arte identificou diversas outras soluções voltadas à interação com o ambiente de experimentação do Mininet, propostas para mitigar as restrições das abordagens anteriores. Tais abordagens, no entanto, ainda possuem deficiências críticas, como a falta de integração da execução da emulação diretamente na interface, comprometendo a experiência de um ambiente verdadeiramente integrado para experimentação. Outras limitações incluem a ausência de uma interface web intuitiva, de um acesso ao console para execução de comandos e a geração automatizada de topologias. Essas lacunas são evidenciadas pelo fato de abordagens recentes continuarem recorrendo ao MiniEdit [Bhaskaran and M. 2024, Romanov et al. 2022].

Diante desse cenário, caracterizado pelo crescimento da demanda por soluções SDN e pelas limitações das ferramentas existentes, propomos o Mininet-GUI, uma ferramenta integrada para a experimentação operacionalizada através de uma interface gráfica que permite a criação, edição e execução de topologias em tempo real no Mininet. O Mininet-GUI não somente simplifica o processo de experimentação em SDN, mas também fornece perspectivas para fomentar a adoção de redes programáveis por uma nova geração de pesquisadores e profissionais ao maximizar as possibilidades de ensino desse paradigma por meio de uma abordagem interativa e simplificada. Ao integrar todas as funcionalidades essenciais em uma única plataforma, o Mininet-GUI reduz a necessidade de manipulação via CLI diretamente no host hospedeiro e a dependência de múltiplas ferramentas para a realização de experimentos completos.

As principais contribuições do Mininet-GUI incluem:

1. Interface web dinâmica para manipulação intuitiva da topologia de rede.
2. Emulação integrada, eliminando a necessidade de configurações externas.
3. Suporte a modificações na topologia em tempo real.
4. Geração automatizada de topologias, incluindo diversos modelos amplamente utilizados.
5. Terminal integrado dos nós via WebShell.
6. Visualização e criação das regras de fluxo.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 discute os principais trabalhos relacionados, justificando o Mininet-GUI. A Seção 3 detalha a arquitetura e implementação do Mininet-GUI, detalhando suas principais funcionalidades e fluxos operacionais. A Seção 4 descreve os cenários experimentais que ilustram a aplicabilidade da solução. Por fim, a Seção 5 apresenta as considerações finais e direções futuras para o aprimoramento da ferramenta.

2. Trabalhos Relacionados

O levantamento do estado da arte sobre as soluções de interação entre o usuário e o ambiente de experimentação com Mininet revelou que as soluções existentes geralmente foram implementadas para atender a demandas específicas, sem que nenhuma delas ofereça um conjunto abrangente de funcionalidades. Isso significa não haver uma solução unificada que integre todas as capacidades essenciais para um ambiente completo e interativo para a experimentação em SDN. A seguir, são descritas as funcionalidades identificadas nas soluções analisadas, organizadas nos itens F1 ao F10:

F1. Emulação integrada: executa e controla redes virtualizadas diretamente na ferramenta.

- F2. Suporte a modificações em tempo real:** permite a criação e remoção de dispositivos e links na topologia sem interromper a execução.
- F3. Interface web:** possibilita a interação via navegador, proporcionando uma experiência intuitiva e dinâmica.
- F4. Webshell:** fornece acesso à CLI dos dispositivos emulados para a execução de comandos através da interface web.
- F5. Exportação de script Python para o Mininet:** gera arquivos compatíveis com a API do Mininet em Python.
- F6. Suporte a múltiplos controladores customizáveis:** possibilita o uso de diversos controladores distintos com configurações personalizadas.
- F7. Manipulação gráfica:** permite a criação e posicionamento dos dispositivos através de operações com o mouse.
- F8. Persistência em banco de dados:** armazena configurações e topologias em uma base de dados.
- F9. Suporte a switches P4:** compatibilidade com switches programáveis em P4.
- F10. Geração automatizada de topologias:** possibilita a criação de topologias complexas, de forma automatizada, reduzindo a intervenção manual.

Embora o Mininet possua um mecanismo interno para a geração automática de topologias, disponível somente pela CLI do sistema operacional hospedeiro, essa capacidade não está presente nas ferramentas existentes. A Seção 3 detalha como o Mininet-GUI aprimora cada uma das funcionalidades apresentadas, garantindo uma abordagem modular e integrada na interface web.

A Tabela 1 compara o Mininet-GUI e as demais ferramentas existentes em relação às funcionalidades listadas.

Tabela 1. Comparação de funcionalidades (codificadas) e anos de publicação das principais ferramentas analisadas.

Ferramenta	Ano	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
MiniEdit	2010	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
VND	2013	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗
OSHI	2014	✓	✗	✓	✓	✗	✓	✓	✗	✗	✗
NetIDE	2016	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗
Vycital	2019	✗	✗	✓	✗	✓	✓	✓	✗	✗	✗
MiTE	2021	✗	✗	✗	✗	✓	✓	✓	✗	✓	✗
Integrated	2023	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗
Mininet-GUI	2025	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓

As subseções seguintes discutem os trabalhos relacionados em duas categorias: (1) Ferramentas sem integração à emulação e (2) Ferramentas com integração à emulação.

2.1. Ferramentas sem integração à emulação

As ferramentas identificadas nessa categoria não fornecem mecanismos integrados para iniciar e gerenciar a emulação diretamente na interface. A ausência desse recurso exige que o usuário configure e execute a emulação manualmente.

VND [Fontes et al. 2014] viabiliza a criação de cenários OpenFlow por meio de bibliotecas customizáveis em uma interface web. No entanto, sua implementação baseada em Adobe Flash tornou-se obsoleta, após a descontinuação da tecnologia em 2020.

Mininet Editor [Vyčítal 2019] oferece uma interface web e permite a definição explícita das configurações de conexões entre as portas dos dispositivos, porém sem integração com a execução da emulação.

MiTE [Sidiq et al. 2021] fornece compatibilidade com switches programáveis em P4. Contudo, assim como as outras ferramentas listadas, não integra a execução da emulação diretamente na ferramenta, obrigando o usuário a iniciá-la manualmente via terminal.

2.2. Ferramentas com integração à emulação

As demais ferramentas, embora ofereçam suporte à execução e controle da emulação na interface, estão restritas a funcionalidades específicas.

MiniEdit [Lantz et al. 2010], distribuído com o Mininet, possui uma interface gráfica simples. Os recursos oferecidos não fornecem acesso a funcionalidades avançadas e à criação de topologias complexas.

NetIDE [Aranda Gutiérrez et al. 2016] permite a integração de múltiplos controladores, possibilitando que a mesma aplicação opere independentemente da tecnologia subjacente, minimizando o *vendor lock-in*.

OSHI [Salsano et al. 2014] possibilita a criação de redes híbridas, combinando roteamento IP tradicional e comutação SDN, facilitando a reprodução de cenários realistas em larga escala.

[Barlowe et al. 2023] adota uma abordagem unificada para construção, exploração e armazenamento de múltiplos modelos SDN, utilizando uma base de dados orientada a grafos que permite a comparação de diversos parâmetros.

A análise das ferramentas existentes concluiu que nenhuma delas fornece suporte a modificações dinâmicas ou reconfigurações em tempo de execução, exigindo que o usuário realize diversas operações manuais, muitas vezes em distintas interfaces e aplicativos, para a condução da experimentação. A próxima seção apresenta o Mininet-GUI, uma ferramenta que preenche essa lacuna ao proporcionar um ambiente de experimentação integrado e interativo, maximizando as perspectivas de experimentação em SDN.

3. Mininet-GUI

A arquitetura do Mininet-GUI, apresentada na Figura 1, segue o modelo cliente-servidor, composto por um *backend*, responsável por instanciar e gerenciar a execução do *Mininet*, e um *frontend*, uma aplicação *Single Page Application* (SPA) que oferece uma interface interativa para a manipulação visual da topologia de rede em um grafo dinâmico. A comunicação entre esses componentes é intermediada pela API Broker, que integra uma API HTTP baseada no padrão OpenAPI e um gerenciador de conexões *WebSocket*, garantindo maior flexibilidade e modularidade à plataforma.

O *backend* do Mininet-GUI foi desenvolvido na linguagem Python, devido à API nativa do Mininet também ter sido implementada nessa linguagem. Para a construção da

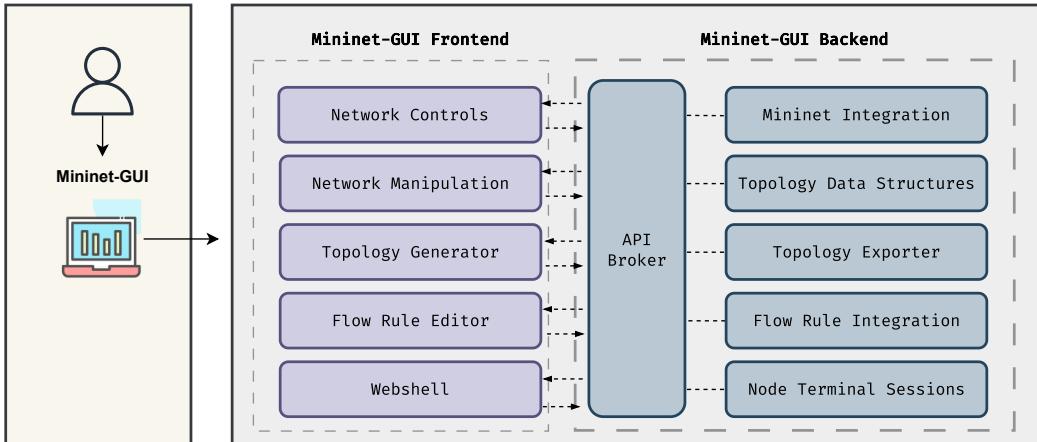


Figura 1. Arquitetura do Mininet-GUI.

API, foi utilizado o *FastAPI*¹, um *framework* moderno e eficiente para desenvolvimento de APIs em Python. O *backend* é composto por quatro principais componentes, cada um com funções específicas:

- **Mininet Integration:** gerencia a interação com as bibliotecas do Mininet, permitindo a criação e manipulação dinâmica dos elementos da rede, como *hosts*, *switches*, *controladores* e *links*. Além disso, é responsável pela inicialização e parada da rede, coleta de estatísticas, execução de comandos remotos nos nós e associação dinâmica de switches a controladores.
- **Topology Data Structures:** estruturas de dados que armazenam as informações da topologia, como os *hosts*, *switches* e controladores, em classes, listas, tuplas e dicionários do Python.
- **Topology Exporter:** facilita a importação e exportação de topologias nos formatos *JavaScript Object Notation* (JSON) e Python.
- **Flow Rule Integration:** oferece funcionalidades para edição e visualização das tabelas de fluxo dos dispositivos de rede.
- **Node Terminal Sessions:** implementa conexões *WebSocket* para gerenciar e exibir os terminais dos nós no *frontend* para sua utilização no Webshell.

O projeto de desenvolvimento do *frontend* do Mininet-GUI tem como premissa fornecer uma interface capaz de responder rapidamente às interações do usuário, sem a necessidade de recarregamentos completos da página, assim, proporcionando maior qualidade de experiência. Dessa forma, o *frontend* consiste em uma implementação baseada em *Vue.js*², um *framework* JavaScript amplamente utilizado para a construção de interfaces dinâmicas e reativas. Para a exibição da topologia da rede, o sistema emprega a biblioteca *vis-network*³, que permite representar os elementos da rede como um grafo interativo.

O *frontend* é composto pelos seguintes módulos:

¹<https://github.com/fastapi/fastapi>

²<https://vuejs.org>

³<https://github.com/visjs/vis-network>

- **Network Controls**: permite adicionar, remover e editar *hosts*, *switches* e *controladores*, além de reiniciar a topologia e executar testes como *pingall*.
- **Network Manipulation**: responsável pela visualização e manipulação interativa da topologia de rede.
- **Topology Generator**: oferece um mecanismo para geração automatizada de topologias de rede. Suporta os modelos *single*, *linear* e *tree*, com a quantidade de switches definida pelo usuário.
- **Flow Rule Editor**: consiste em uma ferramenta para visualização, criação e remoção de regras de fluxo OpenFlow nos switches.
- **Webshell**: permite a interação direta do usuário com os dispositivos da rede através da interface gráfica, eliminando a necessidade de acessar terminais externos.

O sistema também fornece suporte à exportação e importação de topologias no formato JSON e Python, facilitando o compartilhamento e a replicação de experimentos.

O principal desafio na implementação do Mininet-GUI foi permitir que os nós e links sejam criados e removidos em tempo de execução, sem que a emulação da rede precisasse ser interrompida ou reiniciada. Essa dificuldade foi superada com o uso dos métodos *start* e *stop* dos switches, que configuraram o dispositivo mesmo quando a rede já está em execução. A associação dos switches aos controladores também acontece dessa maneira, o que possibilita um ambiente dinâmico e interativo no frontend para a manipulação dos dispositivos da rede emulada.

Outro grande desafio foi a integração com o terminal dos nós do Mininet, através do componente de Webshell. No desenvolvimento do Webshell, foi necessário adicionar uma camada de compatibilidade com os sistemas Windows e Linux, pois os caracteres de controle (como quebra de linha e backspace, por exemplo) possuem diferenças entre os sistemas operacionais. Além disso, o Webshell necessitou da comunicação com o backend através de Websockets ao invés da API HTTP, pois o fluxo dos dados trocados entre o frontend e o backend ocorre em tempo real e bidirecionalmente.

4. Demonstração

O código-fonte do Mininet-GUI, juntamente com um vídeo da ferramenta em funcionamento, está disponível no repositório⁴ oficial, além da documentação completa do código-fonte. A ferramenta pode ser utilizada de duas formas diferentes: com a instalação a partir do código-fonte; e também com uma máquina virtual pré-configurada.

Esta demonstração tem como objetivo apresentar, na prática, os conceitos e funcionalidades descritos nas Seções 2 e 3, ilustrando o uso do Mininet-GUI em diferentes cenários de experimentação. Para isso, a demonstração será desenvolvida no âmbito dos seguintes casos de uso:

1. **Modelagem rápida**: criação ágil de topologias customizadas e execução de testes de conectividade e *troubleshooting* de redes.
2. **Avaliação de desempenho**: utilização de ferramentas externas, como o Iperf⁵, para análise de tráfego entre dispositivos na rede por meio do WebShell.

⁴<https://github.com/latarc/mininet-gui>

⁵<https://iperf.fr/>

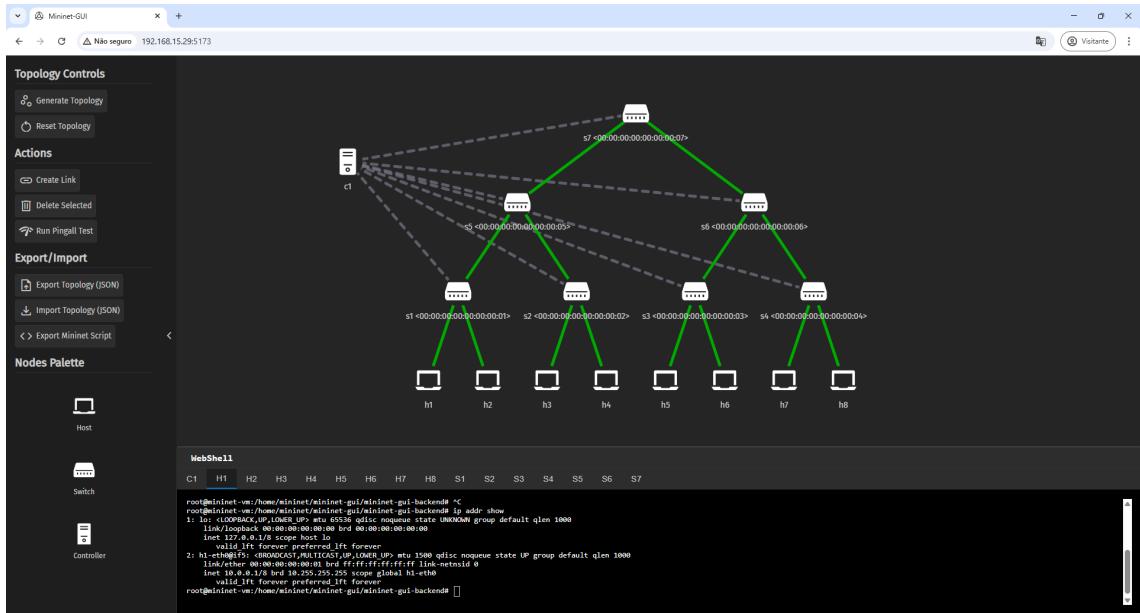


Figura 2. Interface web do Mininet-GUI.

3. Topologia Multi-Controlador: experimentações em infraestruturas de rede orquestradas por múltiplos controladores SDN.

A Figura 2 apresenta a interface do Mininet-GUI. A interface é composta por três áreas principais:

- **Barra lateral:** contém os controles interativos da rede, permitindo a geração e modificação de topologias, reinicialização da emulação, criação e remoção de *links*, *hosts*, *switches* e controladores, além da exportação e importação de topologias nos formatos JSON e script Python do Mininet.
- **Painel inferior:** apresenta o componente Webshell, fornecendo acesso direto aos terminais dos nós da rede, organizados em abas.
- **Área principal:** exibe a visualização gráfica da topologia, onde os ícones brancos representam os dispositivos e os links verdes indicam as conexões ativas. A associação entre switches e controladores SDN é representada pelas linhas pontilhadas na cor cinza.

5. Conclusão

O Mininet-GUI oferece uma interface gráfica moderna para criação, edição e execução de redes SDN no Mininet, tornando a experimentação mais acessível. Com uma arquitetura modular baseada no modelo cliente-servidor, permite a manipulação visual da topologia, simulações em tempo real, edição de regras de fluxo e acesso remoto via Webshell. Sua API facilita a geração, exportação e importação de topologias, fornecendo versatilidade tanto a iniciantes quanto a usuários avançados.

Em trabalhos futuros, pretende-se estender o Mininet-GUI para integrá-lo ao Mininet-WiFi e ao Containernet, ampliando sua aplicabilidade para experimentos com redes sem fio e virtualização baseada em *containers*. Além disso, serão integradas funcionalidades para visualização do tráfego de pacotes em tempo real e o suporte ao P4.

As novas contribuições potencializarão o Mininet-GUI como uma alternativa eficaz para experimentação com SDN.

Referências

- Aranda Gutiérrez, P., Rojas, E., Schwabe, A., Stritzke, C., Doriguzzi-Corin, R., Leckey, A., Petralia, G., Marsico, A., Phemius, K., and Tamurejo, S. (2016). Netide: All-in-one framework for next generation, composed sdn applications. In *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pages 355–356.
- Barlowe, S., Stanley, M., Lowry, N., Tipton, C., and Cruz, G. (2023). Towards an integrated framework for managing software-defined networking models. In *SoutheastCon 2023*, pages 23–30.
- Bhaskaran, S. and M., S. (2024). Study on networking models of sdn using mininet and miniedit. In *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pages 1159–1164.
- Bouras, C., Kollia, A., and Papazois, A. (2017). Teaching 5g networks using the onos sdn controller. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 312–317.
- Cosgrove, S. (2016). Teaching software defined networking: It's not just coding. In *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 139–144.
- Fontes, R. R., Oliveira, A. L. C., Sampaio, P. N. M., Pinheiro, T. R., and Figueira, R. A. R. B. (2014). Authoring of openflow networks with visual network description (sdn version) (wip). In *Proceedings of the 2014 Summer Simulation Multiconference*, SummerSim '14, San Diego, CA, USA. Society for Computer Simulation International.
- Haleplidis, E., Salim, J. H., Meyer, D., Hagsand, O., and Pentikousis, D. (2015). Software-defined networking (sdn): Layers and architecture terminology. *RFC 7426*.
- Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, New York, NY, USA. Association for Computing Machinery.
- Romanov, O., Nesterenko, M., Marinov, A., Skolets, S., and Burlaka, H. (2022). Sdn network modeling using the gui miniedit. In *2022 IEEE 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pages 01–06.
- Salib, E. H. and Lester, J. D. (2018). Hands-on labs and tools for teaching software defined network (sdn) to undergraduates. In *2018 ASEE Annual Conference & Exposition*.
- Salsano, S., Ventre, P. L., Prete, L., Siracusano, G., Gerola, M., and Salvadori, E. (2014). Oshi - open source hybrid ip/sdn networking (and its emulation on mininet and on distributed sdn testbeds). In *2014 Third European Workshop on Software Defined Networks*, pages 13–18.
- Sidiq, M. F., Pranata, M., and Basuki, A. I. (2021). Visual editor for streamlining p4-based programmable parser development. In *2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, pages 225–230.

Vyčítal, T. (2019). Gui editor pro mininet [online]. Diplomová práce, Univerzita Pardubice, Fakulta elektrotechniky a informatiky Pardubice. SUPERVISOR: Ing. Miroslav Dvořák, Dipl.tech.