

# ENTORNOS DE APRENDIZAJE A TRAVÉS DE LA EMULACIÓN PARA LA ENSEÑANZA DE REDES PROGRAMABLES

## LEARNING ENVIRONMENTS THROUGH EMULATION FOR TEACHING PROGRAMMABLE NETWORKS

D. I. Gallardo Álvarez<sup>1</sup>

J. P. Razón González<sup>2</sup>

E. G. Vargas Espinoza<sup>3</sup>

### RESUMEN

Para todo docente es trascendental mantenerse actualizado acerca de los métodos de enseñanza más efectivos, de manera que pueda favorecer el aprendizaje significativo de los alumnos. Indiscutiblemente, los tiempos y las formas en que las personas aprenden han cambiado, los métodos en los cuales los estudiantes eran sujetos pasivos han quedado atrás y ahora se busca involucrarlos activamente en su propia formación. Estas transformaciones imponen el reto, la necesidad y sobre todo la posibilidad de renovar las técnicas de enseñanza y el tipo de material educativo que se pone a disposición de los estudiantes, por ello, el objetivo principal del presente trabajo es mostrar el uso del emulador Mininet para crear entornos de aprendizaje que ayuden a la enseñanza de un concepto emergente que está revolucionando el campo de las comunicaciones denominado Redes Definidas por Software (SDN), sin la necesidad del equipo especializado requerido para su estudio; estas redes que se consideran programables permiten que los dispositivos sean orquestados de manera centralizada con la ayuda de algún protocolo como OpenFlow. Se presenta el diseño e implementación de dos escenarios de red desarrollados con Mininet, utilizando la metodología PDIOO propuesta por Cisco Systems, y se logra demostrar la funcionalidad y ventajas que tiene este emulador como herramienta de apoyo a la enseñanza de este tópico.

### ABSTRACT

For every teacher it is transcendental to keep up to date on the most effective teaching methods, so that it can favor the meaningful learning of the students. Unquestionably, times and the ways in which people learn have changed, the methods in which students were passive subjects have been left behind and now they are actively involved in their own training. These transformations impose the challenge, the need and above all the possibility of renewing the teaching techniques and the type of educational material that is made available to the students, therefore, the main objective of this work is to show the use of the Mininet emulator. to create learning environments that help teach an emerging concept that is revolutionizing the field of communications called Software Defined Networks (SDN), without the need for the specialized equipment required for its study; These networks that are considered programmable allow devices to be centrally orchestrated with the help of some protocol like OpenFlow. The design and implementation of two network scenarios developed with Mininet is presented, using the PDIOO methodology proposed by Cisco Systems, and it is possible to demonstrate the functionality and advantages of this emulator as a support tool for teaching this topic.

### ANTECEDENTES

Los avances en la tecnología y en específico las telecomunicaciones han orientado a la automatización de las redes de datos. Se dice que las Redes Definidas por Software o SDN por sus siglas en inglés (Software Defined Networking) son un nuevo modelo de las redes de

<sup>1</sup> Profesor de tiempo completo. Tecnológico Nacional de México/Instituto Tecnológico Superior de Irapuato. dennise.ga@irapuato.tecnm.mx

<sup>2</sup> Profesor de tiempo completo. Tecnológico Nacional de México/Instituto Tecnológico Superior de Irapuato. juan.rg@irapuato.tecnm.mx

<sup>3</sup> Profesor de tiempo completo. Tecnológico Nacional de México/Instituto Tecnológico Superior de Irapuato. elizabeth.ve@irapuato.tecnm.mx

datos, que produce cambios significativos en su construcción y operación; las SDN permiten el control de la red desde un único punto (controlador), gracias a la separación del plano de control y el plano de datos.

La arquitectura de las SDN consta de tres capas: capa de infraestructura, capa lógica y capa de aplicación. La capa de infraestructura está conformada por los dispositivos de red (switches y routers), los cuales pueden ser configurados por los controladores a través de la implementación de reglas de supervisión del tráfico para el cambio o políticas de seguridad. La capa media forma el controlador, el cual es el encargado de tomar las decisiones y programar las tablas de flujo de los elementos de la capa inferior para controlar el flujo de paquetes. Y la capa superior que es la capa de aplicación, la componen las aplicaciones creadas por los usuarios que son las que permiten personalizar las redes (Open Networking Foundation, 2023).

Esta idea de programar redes no es nueva, pero en la actualidad, la inteligencia de la red se centra lógicamente en controladores basados en software y los dispositivos de red se convierten solamente en dispositivos de reenvío de paquetes que pueden ser programados a través de una interfaz abierta como lo es OpenFlow.

OpenFlow es una interfaz abierta para el control remoto de las tablas de reenvío de los conmutadores de red, routers y puntos de acceso. El protocolo Openflow constituye la base de las SDN, este fue propuesto por diferentes miembros de las universidades de Stanford y California en Berkeley. En la actualidad, la Open Networking Foundation (ONF) se encarga de la definición del estándar (Figuerola, 2013).

OpenFlow Controller es el dispositivo principal en la arquitectura SDN. Es él quien toma las decisiones, implementa las reglas de la red, ejecuta las instrucciones que le proporcionan las diferentes aplicaciones y las distribuye a los diferentes dispositivos de la capa física de la red. Es quien determina cómo manejar los paquetes que no encajan en ninguna de las entradas de las tablas de flujo y quien gestiona dichas entradas, añadiendo o eliminando a través del canal seguro a los dispositivos OpenFlow (García et al., 2014).

De acuerdo con resultados del estudio State of the Network | 2019, hecho por Viavi Solutions, que consulta a los profesionales de TI sobre tecnologías emergentes o nuevas y sobre los despliegues dentro de sus organizaciones empresariales, se anuncia que las empresas están acelerando el despliegue de la tecnología SDN. Los hallazgos indican que aproximadamente 3 de cada 4 empresas, la implementaron para 2021 (Viavi Solutions Inc., 2023).

Hoy en día, en la mayoría de las organizaciones, las redes de telecomunicaciones se han convertido en parte esencial de su infraestructura tecnológica, por ello, la demanda de personal capaz de administrarlas está en crecimiento.

Por otro lado, para las instituciones educativas de nivel superior es difícil hablar de egresados altamente potenciales, pues son muchos los aspectos a medir y que intervienen en la competencia del universitario. Cada organización cuenta con sus propias necesidades a satisfacer y buscan al profesionista ideal para integrarlo y darle solución a sus problemas. Resulta difícil saber dónde el egresado llevará acabo sus actividades profesionales por lo que

se hace urgente la preparación del estudiante desde una perspectiva más amplia, que le permita ser competitivo en todos los campos.

En específico, el Tecnológico Nacional de México campus Irapuato siendo una institución de nivel superior, busca en los alumnos una formación integral a través de un proceso educativo de alta calidad con un enfoque cuyo propósito es adquirir conocimientos técnicos, científicos y humanísticos, lo cual permita al egresado convertirse en promotor y agente de cambio (Instituto Tecnológico Superior de Irapuato [ITESI], 2020).

En la carrera de Ingeniería en Sistemas Computacionales de esta institución, se cuenta con la especialidad “Redes de Computadoras”, en la cual se estudia el tópico SDN dentro de la materia Redes Convergentes y Cómputo en la nube, sin embargo, desafortunadamente en el laboratorio de prácticas existente no se cuenta con el equipo especializado para la enseñanza práctica de esta tecnología, por lo que, el tema únicamente se enseña de manera teórica. Dicha situación ha dado pie a que los maestros que imparten esta materia busquen alternativas que permitan a los estudiantes aprender mediante la experiencia al practicar con estas redes y así lograr un aprendizaje significativo.

Ante este panorama, se aprovecha esta área de oportunidad para investigar sobre el emulador “Mininet” y ver el impacto que tiene para la enseñanza de esta tecnología que ayude al fortalecimiento de conocimientos en los estudiantes, pero ¿De qué manera un emulador se considera efectivo en la enseñanza al no contar con equipo especializado?

La emulación es cuando una computadora se comporta como otro dispositivo. Mininet es un emulador de redes enfocado a las Redes Definidas por Software y el protocolo OpenFlow. Permite crear las SDN emulando redes mediante un escenario virtual, a través del lenguaje de programación Python. Dicha herramienta virtualiza hosts, switches y routers, lo que permite el desarrollo de una gran variedad de topologías con las que los estudiantes pueden practicar que, al no tener el equipo especializado, de otro modo no se podrían desarrollar.

Mininet usa una virtualización ligera para crear una red completa y cada host que se utiliza funciona como una máquina real, es decir, los dispositivos virtuales de Mininet se crean utilizando software en lugar de hardware. La eficiencia de Mininet proporciona una manera fácil de observar el comportamiento de la red además de experimentar con topologías SDN de manera virtual. Es fácil interactuar con la herramienta Mininet mediante el API que ofrece, debido a que su uso está destinado para el desarrollo, la enseñanza y la investigación (Lantz, et al., 2021).

## **METODOLOGÍA**

Este proyecto es dividido en dos etapas, la primera enfocada al desarrollo y estudio de los escenarios de red SDN y la segunda orientada al análisis de Mininet como herramienta de apoyo a la enseñanza de las redes definidas por software, considerando una muestra de estudiantes:

### Desarrollo de escenarios de red

La metodología elegida para el desarrollo de los escenarios de red es PDIOO, esta es propuesta por Cisco Systems y permite formalizar el diseño de una red en cinco fases: Planeación, Diseño, Implementación, Operación y Optimización (Teare, 2008).

#### Planeación

Dado el objetivo del proyecto, se establece que los escenarios de red se generen en un entorno virtual, y se diseñen y configuren considerando dos niveles en sus topologías para lograr en los alumnos un aprendizaje significativo y de manera incremental:

- Topologías de nivel básico
- Topologías de nivel avanzado

Por tanto, para la creación de estos escenarios, así como, las pruebas necesarias para la implementación de las SDN se utilizan las siguientes herramientas:

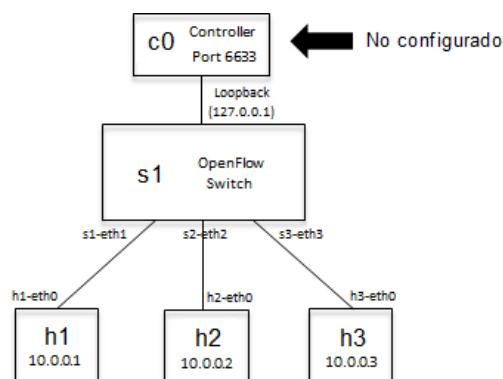
- Software de virtualización
- Sistema operativo Linux
- Sistema de ventanas
- Cliente de acceso remoto
- Analizador de protocolos

#### Diseño

Las topologías planteadas para el presente proyecto son diseñadas de acuerdo con el grado de dificultad mencionado anteriormente:

- a) Nivel básico: Una red se considera de este nivel porque maneja comandos básicos que permiten familiarizarse con la interfaz del Mininet y las herramientas utilizadas, además, la red que se crea dispone de un controlador, pero no configurado, es decir, los flujos de la red se configuran de manera manual. En la Figura 1 se muestra la topología propuesta para este nivel que incluye un switch y tres hosts.

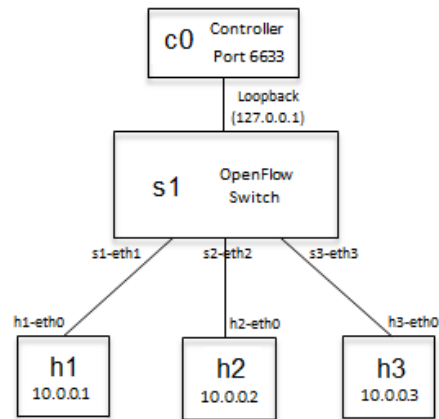
**Figura 1.** Topología de red – nivel básico



- b) Nivel Avanzado: Una red de nivel avanzado implica hacer uso de un controlador configurado, dicha configuración es realizada por medio de código en Python. En la

Figura 2 se muestra la topología propuesta para este nivel que incluye un switch, tres hosts y un controlador configurado, realizando la función del switch.

Figura 2. Topología de red – nivel avanzado



Implementación

Para la implementación de las redes se utilizan herramientas de hardware y software las cuales se describen en la Tabla 1:

Tabla 1. Hardware y software utilizado en la creación de las SDN

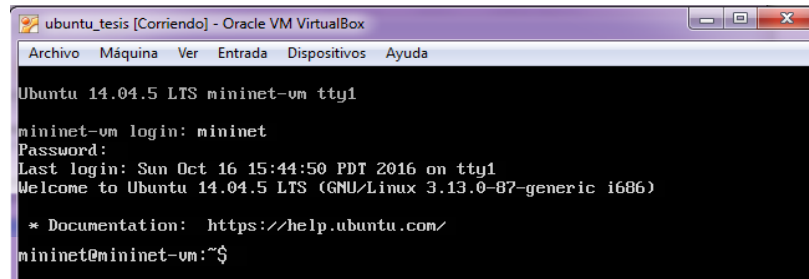
HARDWARE			SOFTWARE
Computadora			• VirtualBox (Libre, GPL)
	Mínimo	Utilizado	• Sistema Operativo Virtualizado Ubuntu 14.04 de 32 bits.
Procesador:	x86	AMD A6-4400M APU con Radeon(tm) HD Graphics 2.70 GHz	• Mininet 2.2.1.
Memoria RAM:	2 GB	6 GB	• Wireshark
Disco Duro:	250 GB	500 GB	• Putty Release 0.67
Sistema Operativo:	Windows XP o superior/ Linux	Windows 7 Ultimate	• Xming 6.9.0.31

En primer lugar, se instaló el software de virtualización “VirtualBox” para el despliegue de la máquina virtual “Mininet/Ubuntu VM”, la cual se encuentra a disposición de todos los usuarios en la página oficial de Mininet; dicha máquina virtual tiene instalado Mininet sobre el sistema operativo Ubuntu y su interfaz es línea de comandos. Una vez instalada, el acceso es a través de un usuario y contraseña como se muestra en la Figura 3. En este momento se pueden crear y configurar redes SDN en conjunto con una sesión SSH.

Cabe mencionar que, el uso de una interfaz de sistema de ventanas (Xming) en Windows es muy útil para configurar de forma independiente cada host virtualizado en Mininet, además

permite el acceso al programa Wireshark para realizar capturas de paquetes del flujo de datos con el protocolo OpenFlow de las redes creadas.

**Figura 3.** *Interfaz de Mininet*



### Operación

La funcionalidad de las topologías desarrolladas y la diferencia entre ellas es revisada con el comando ping y la herramienta Wireshark.

Para la topología uno, en la Figura 4 se observa el correcto funcionamiento mediante un ping entre el host 1 y el host 2.

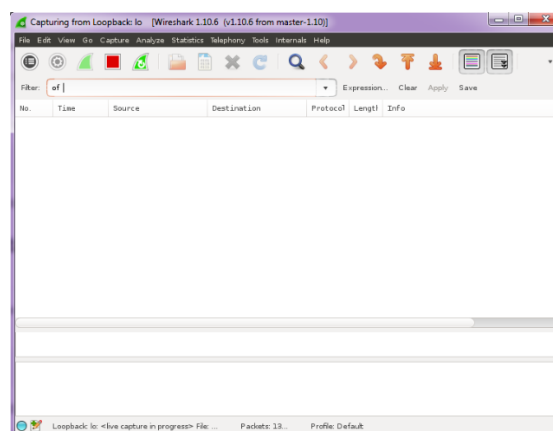
**Figura 4.** *Ping entre host 1 y host 2 – Topología uno*

```
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.944 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.049 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.049/0.348/0.944/0.421 ms
mininet>
```

En la Figura 5 se observa la ejecución de Wireshark, sin embargo, no muestra paquetes del protocolo OpenFlow (of), debido a que, en esta topología no existe un controlador programado.

**Figura 5.** *Flujo de paquetes con Wireshark – Topología uno*



Para la topología dos, en la Figura 6 se observa el correcto funcionamiento a través de un ping entre el host 1 y la dirección IP del host 2.

**Figura 6.** Ping entre host 1 e IP del host 2 – Topología dos

```

root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=77.2 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 77.266/77.266/77.266/0.000 ms
root@mininet-vm:~#

```

Es importante mostrar que debido a que es el primer paquete que se envía entre los hosts, este se envía como broadcast, pero el paquete llega con éxito solo al host de destino (ver Figura 7).

**Figura 7.** Flujo de tráfico tipo broadcast – Topología dos

```

root@mininet-vm:~# tcpdump -XX -n -i h2-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h2-eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:11:46.655153 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 7936, seq 1, length 64
0x0000: 0000 0000 0002 0000 0000 0001 0800 4500 .....E.
0x0010: 0054 28ae 0000 4001 5672 0a00 0002 0a00 .....T.
0x0020: 0002 0800 c739 1f00 0001 7534 6c57 3c36 ....S...uIW6
0x0030: 0900 0809 0a0b 0c0d 0a0f 1011 1213 1415 .....
0x0040: 1517 1819 1a1b 1c1d 1e1f 2021 2223 2425 .....*
0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 ...../012345
0x0060: 3637 .....67
12:11:49.655175 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 7936, seq 1, length 64
0x0000: 0000 0000 0001 0000 0000 0002 0800 4500 .....E.
0x0010: 0054 28ae 0000 4001 5672 0a00 0002 0a00 .....T.
0x0020: 0001 0000 c739 1f00 0001 7534 6c57 3c36 ....S...uIW6
0x0030: 0900 0809 0a0b 0c0d 0a0f 1011 1213 1415 .....
0x0040: 1517 1819 1a1b 1c1d 1e1f 2021 2223 2425 .....*
0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 ...../012345
0x0060: 3637 .....67

root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=53.5 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 53.585/53.585/53.585/0.000 ms
root@mininet-vm:~#

```

Finalmente, en la Figura 8, se presenta la ejecución de Wireshark que muestra el tráfico de red de la segunda topología. En el inciso a) se observa el primer envío de paquete (tipo broadcast) y en el inciso b) se observa el envío de paquetes subsiguientes (tipo unicast), función que realiza un switch.

**Figura 8.** Flujo de paquetes con Wireshark – Topología dos

**a) Envío de paquete broadcast**

No.	Time	Source	Destination	Protocol	Length	Info
9463	191.18918900	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
9465	191.19372100	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply
9573	195.09983300	00:00:00:00:00:01	127.0.0.1	OF 1.0	126	126 of_packet_in
9580	196.10229900	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
9582	196.11167600	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
9584	196.11214900	00:00:00:00:00:02	00:00:00:00:00:01	OF 1.0	126	126 of_packet_in
9585	196.11233000	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply
9587	196.15010600	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
9589	196.15033200	10.0.0.1	10.0.0.2	OF 1.0	182	182 of_packet_in
9591	196.18208200	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
9592	196.18254000	10.0.0.2	10.0.0.1	OF 1.0	182	182 of_packet_in
9593	196.18380200	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
9749	201.10295300	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
9750	201.10298400	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply
9752	201.18844200	00:00:00:00:00:02	00:00:00:00:00:01	OF 1.0	126	126 of_packet_in
9763	201.22710400	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
9763	201.22710400	00:00:00:00:00:01	00:00:00:00:00:02	OF 1.0	126	126 of_packet_in
9764	201.22919400	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
10063	206.10324000	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request

**b) Envío de paquete unicast**

No.	Time	Source	Destination	Protocol	Length	Info
86173	2245.7097230	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
86174	2245.7104980	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply
86200	2246.4667450	10.0.0.1	10.0.0.2	OF 1.0	182	182 of_packet_in
86201	2246.4829810	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
86204	2246.4880270	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
86462	2250.7109480	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
86463	2250.7130660	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply
86521	2251.4780670	00:00:00:00:00:01	00:00:00:00:00:02	OF 1.0	126	126 of_packet_in
86522	2251.4850980	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
86524	2251.4855510	00:00:00:00:00:02	00:00:00:00:00:01	OF 1.0	126	126 of_packet_in
86525	2251.4855210	127.0.0.1	127.0.0.1	OF 1.0	90	90 of_packet_out
87465	2255.7104940	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
87475	2255.7554380	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply
88677	2260.7111320	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
88679	2260.7286690	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply
89024	2265.7118120	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_request
89026	2265.7555680	127.0.0.1	127.0.0.1	OF 1.0	74	74 of_echo_reply



### Optimización

Debido al alcance del proyecto, esta fase no se incluye ya que las simulaciones no se implementan en un entorno real, sin embargo, los escenarios propuestos pueden ajustarse sin ningún inconveniente.

### Análisis de Mininet

Para este análisis es realizada una prueba piloto con 15 estudiantes de la carrera Ingeniería en Sistemas Computacionales del Tecnológico Nacional de México campus Irapuato, mismos que fueron elegidos considerando tres características:

- Estar cursando el noveno semestre de la carrera.
- Estar cursando la especialidad de Redes.
- Estar cursando la materia “Redes convergentes y Cómputo en la nube”.

En esta prueba, los estudiantes tuvieron dos horas por topología para realizarlas y el material de apoyo entregado fue un manual de la práctica y la máquina virtual con las características necesarias para su correcto funcionamiento.

Una vez que los estudiantes interactuaron con Mininet se hace necesario aplicar una encuesta con el objetivo de verificar el impacto del uso de la herramienta. Esta encuesta consta de cuatro preguntas cerradas, en la Tabla 2 se muestran los reactivos realizados.

**Tabla 2.** Cuestionario para estudiantes

No.	Pregunta	Respuesta			
1.	¿En qué porcentaje considera haber comprendido el concepto de redes definidas por software (SDN)?	0 - 25%	25 - 50%	50 - 75%	75 - 100%
2.	¿Cuánto tiempo le tomó realizar cada práctica?	Menos de 30 minutos	De 30 a 60 minutos	De 60 a 90 minutos	Más de 90 minutos
3.	¿Había realizado alguna práctica de este tópico?	SI		NO	
4.	¿Considera oportuno incluir prácticas para comprender mejor el tema?	SI		NO	

Como se puede observar, las primeras dos preguntas son de escala de valoración y se orientan a la revisión de la experiencia en cuanto a comprensión del tema y tiempo al realizar las topologías. Las otras dos son orientadas a la necesidad de realizar prácticas sobre el tópico Redes Definidas por Software para un mejor aprendizaje.

### RESULTADOS

Los resultados que se han tenido con el desarrollo de este proyecto incitan a seguir en la búsqueda de herramientas tecnológicas que ayuden a mantener los métodos de enseñanza atractivos para los estudiantes.

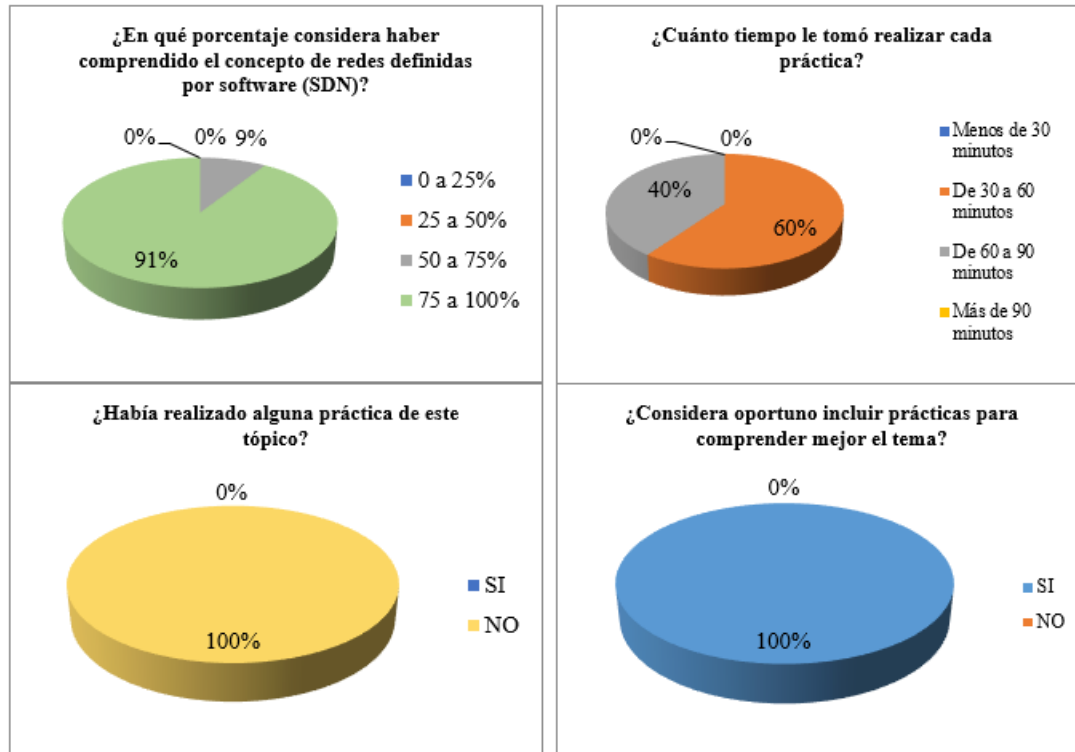
En cuestión tecnológica, se logró el despliegue de un entorno para el estudio de las SDN, mismo que actualmente se utiliza para la configuración de nuevos escenarios, con la intención de conformar un repositorio de prácticas que esté al alcance de los alumnos. Cabe



resaltar que, este entorno fue desplegado sobre una computadora portátil con características óptimas para examinar el funcionamiento de las SDN en un ambiente de laboratorio, por lo que, para un segundo proyecto se observa la necesidad de la implementación en un servidor dedicado para realizar un análisis profundo.

Con respecto a los resultados de las encuestas realizadas a los estudiantes, en la Figura 9 se observan las estadísticas de cada reactivo.

**Figura 9. Resultados – Cuestionario a alumnos**



Como se puede observar, la totalidad de alumnos encuestados respondieron haber comprendido el tema Redes Definidas por Software entre el 50 y el 100%, además de haber realizado cada escenario en un tiempo de entre 30 y 90 minutos (tiempo adecuado para su despliegue y análisis), lo cual indica que Mininet puede considerarse una excelente herramienta para la enseñanza de este tópico.

Por otro lado, los resultados de las otras dos preguntas arrojan que los estudiantes consideran oportuno incluir prácticas para un mejor aprendizaje del tema, ya que la mayoría ha escuchado hablar de las SDN, pero no ha trabajado con ellas.

## CONCLUSIONES

Dadas las transformaciones tecnológicas, las Instituciones de Educación Superior tienen ante sí la responsabilidad de buscar herramientas que fortalezcan la formación de los estudiantes.

Con este proyecto se ha ampliado el panorama sobre el material que se puede aprovechar para mejorar la formación académica de los estudiantes. Se ha cumplido el resultado esperado en un 100%, se logró profundizar en las Redes Definidas por Software, en su funcionamiento, arquitectura y el potencial que tiene este tipo de infraestructura.

Con el uso de la herramienta Mininet se adquieren conocimientos y habilidades técnicas sobre las SDN, permitiendo a los alumnos aprender la manera en la que se programan estas redes desde un entorno virtual, lo que ayuda a que se disminuya la brecha entre la teoría y la práctica.

Finalmente, la ventaja de trabajar con el emulador radica en la flexibilidad de uso que este brinda, ya que permite desarrollar diferentes tipos de redes utilizando dispositivos virtuales que soportan tecnologías de gran magnitud. Así mismo, los beneficios derivados que se destacan son la reducción de gastos en equipamiento informático, mayor grado de experimentación, además de ofrecer la oportunidad de instalar, configurar y administrar la propia instalación ampliando por tanto la formación en un tema que, sin duda, es una tendencia con gran impacto en el futuro.

## BIBLIOGRAFÍA

- Figuerola, N. (2013). *SDN-Redes definidas por software*.  
<https://articulosit.files.wordpress.com/2013/10/sdn.pdf>
- García, A., Rodríguez, C., Anaías, C. y Casmartíño, F. (2014). Controladores SDN, elementos para su selección y evaluación. *Telemática*, vol. 13(3).  
<https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/164>
- Instituto Tecnológico Superior de Irapuato (2020). *Filosofía Institucional*.  
[www.itesi.edu.mx/Acerca\\_de\\_ITESI/filosofia\\_institucional.html](http://www.itesi.edu.mx/Acerca_de_ITESI/filosofia_institucional.html)
- Lantz, B., Handigol, N., Heller, B., & Jeyakumar, V. (March 6, 2021). Introduction to Mininet. *GitHub*. <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
- Open Networking Foundation (2023). *Software-Defined Networking: The New Norm for Networks*.  
<https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>
- Teare, D. (2008). *Authorized Self-Study Guide: Designing for Cisco Internetwork Solutions (DESGN)* (2<sup>nd</sup> Ed.). Cisco Press.  
[http://pws.npru.ac.th/sarthong/data/files/Designing\\_for\\_Cisco\\_Internetwork\\_Solutions\\_DESGN\\_Authorized\\_CCDA.pdf](http://pws.npru.ac.th/sarthong/data/files/Designing_for_Cisco_Internetwork_Solutions_DESGN_Authorized_CCDA.pdf)
- Viavi Solutions Inc. (2023). *2019 State of the Network Study*.  
<https://www.stateofthenetwork.com/studies/2019.php>