



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

## Maestría en Gerencia de Sistemas

# Análisis técnico y económico de la implementación de una red SDN en el backbone de la Sede Matriz de la Universidad de las Fuerzas Armadas ESPE

## Diego Paúl Rojas Rivera

Maestrante

Trabajo de titulación, previo a la obtención del título de Magíster en Gerencia de Sistemas

UFA-ESPE

XVIII



# CONTENIDO



1.- Introducción



2.- Marco Teórico  
referencial



3.- Simulación red  
SDN en Mininet



4.- Análisis Técnico  
y económico



5.- Conclusiones y  
Recomendaciones



# Introducción SDN

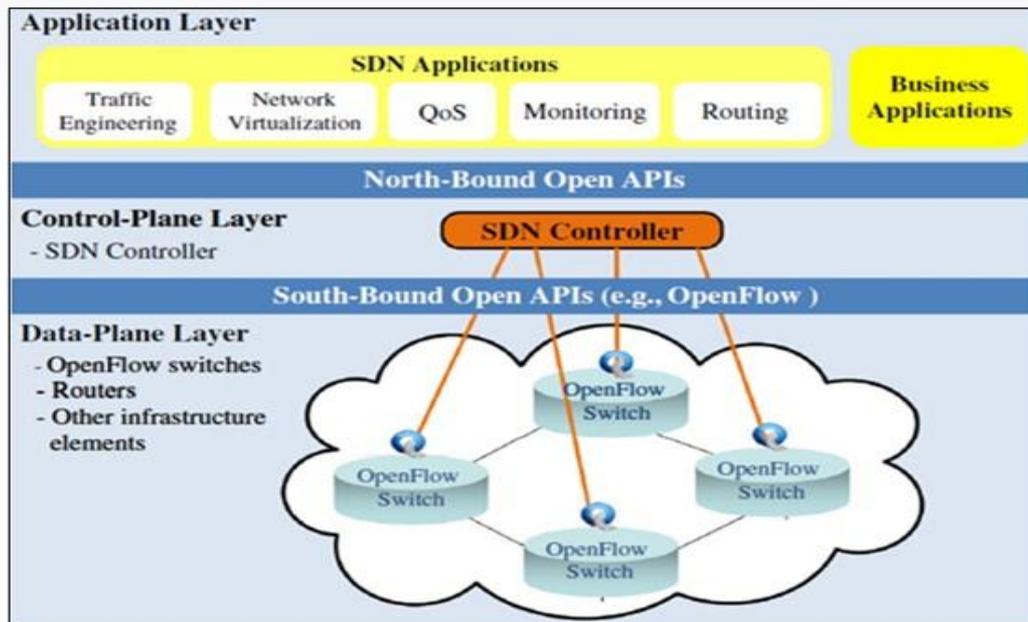
- ▶ En la actualidad tener una red segura, centralizada y escalable es un tema muy importante a considerar en la red de datos de una entidad, esto debido a que se tiene el temor que la misma puede ser vulnerable a ataques informáticos, problemas con la conmutación de datos ante una falla y la complejidad de implementar configuraciones y equipos en redes que no son escalables. Las entidades educativas deberían tomar en cuenta todos aspectos al momento implementar redes para la comunicación de los datos.

# Planteamiento del problema



- ▶ En la actualidad la mayoría de organizaciones cuentan con redes tradicionales como redes distribuidas dentro de su infraestructura; sin embargo, dichas redes no poseen seguridad centralizada en la transmisión de datos ni tampoco la escalabilidad y el modo de obtener el control de su red mediante un solo dispositivo, esto debido al desconocimiento de nuevas tecnologías que permiten realizar un control centralizado y optimizar de esta manera los recursos técnicos y económicos dentro de la organización.

# Redes definidas por software (SDN)

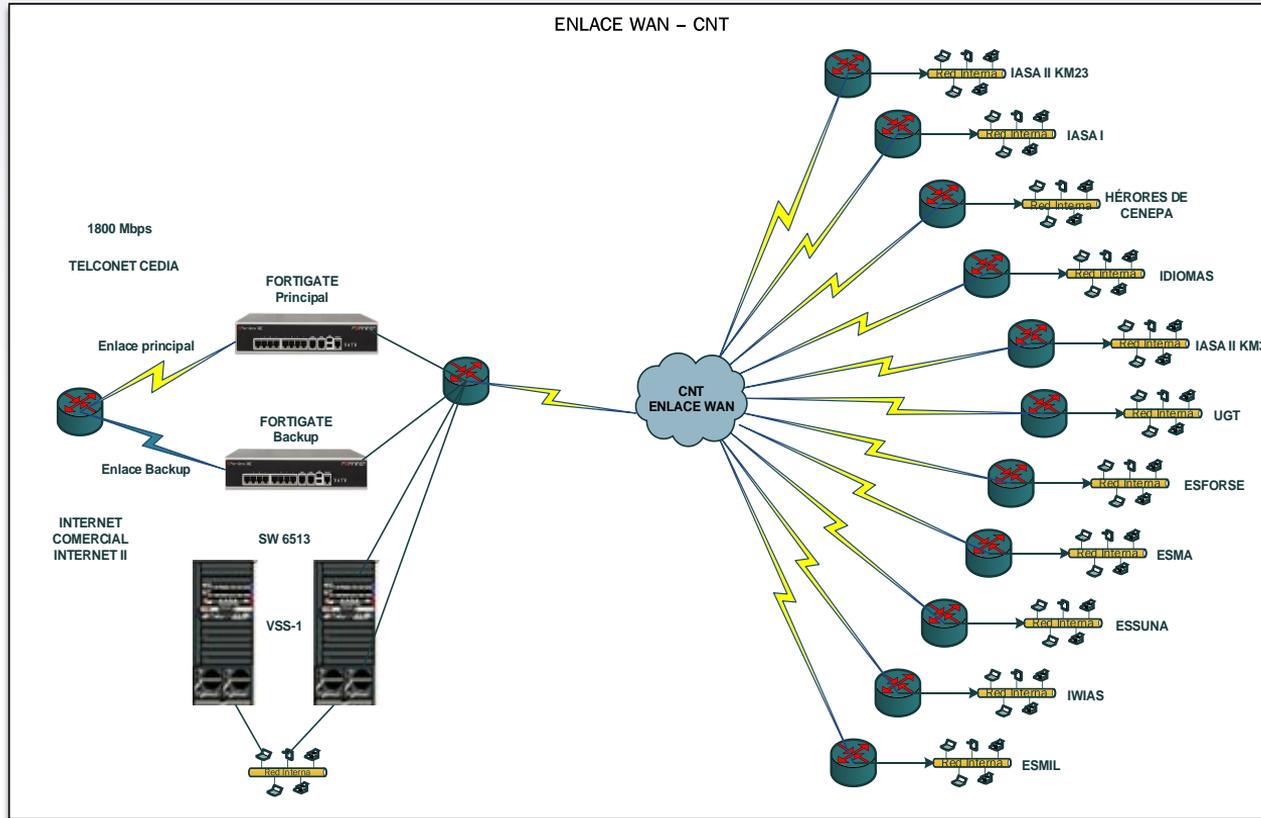


# OPENFLOW

OpenFlow es un protocolo que permite aplicar SDN a nivel de software y de hardware, y el cual constituye la primera interfaz completa para SDN (Pereira & Gamess, 2017). Esto se realiza a partir de la estandarización de los mensajes que se intercambian entre controladores y conmutadores. En términos generales, los mensajes dan las instrucciones o parámetros sobre cómo debe funcionar la conmutación y definir estadísticas de los flujos que se dan

Ingress Port	Src MAC	Dst MAC	Ether Type	VLAN ID	VLAN Priority	Src IPv4	Dst IPv4	IP Protocol	IP TOS	TCP/UDP Src	TCP/UDP Dst	Action	Priority	Counter
*	3c:07:54:*	*	*	Switching	*	*	*	*	*	*	*	Fwd Port 10	100	
*	*	*	Routing	*	*	*	192.168.1.*	*	*	*	*	Fwd Port 12	100	
Port 1	*	*	Replication/SPAN	*	*	*	*	*	*	*	*	Fwd Port 14..24	100	
*	*	*	Firewall/Security	*	*	*	*	*	*	*	23	Drop	100	
*	*	*	Inspection	*	*	*	*	0x06	*	*	*	Controller	100	
*	00:01:E7:*	*	*	VLAN10	*	*	*	*	*	*	80	Fwd Port 8	200	
*	*	*	Multi-action; NAT	*	*	*	192.168.1.*	*	*	*	80	Rewrite 10.1.2.3; Fwd Port 9	200	
			Local handling	*	*	*	10.*	*	*	*	*	Local	200	

# TOPOLOGIA WAN ACTUAL ESPE



# Simulación

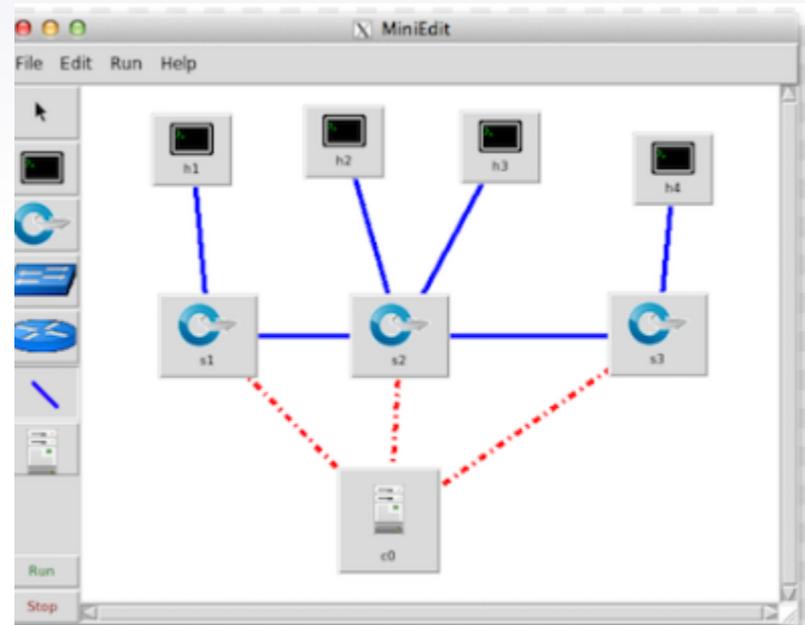
- ▶ Para la simulación, se determinó como método el emulador Mininet basado en Linux y se establecieron los siguientes componentes para la simulación:
- ▶ **Herramientas de desarrollo:**
- ▶ Máquina física para simulación 8GB RAM, procesador de 2 GHz.
- ▶ Máquina virtual Linux (Ubuntu, Red Hat u otra distribución)
- ▶ Controlador SDN seleccionado.
- ▶ Emulador Mininet (simulación mediante Miniedit / Phyton / línea de comandos).

Comparativa controladores SDN de código abierto

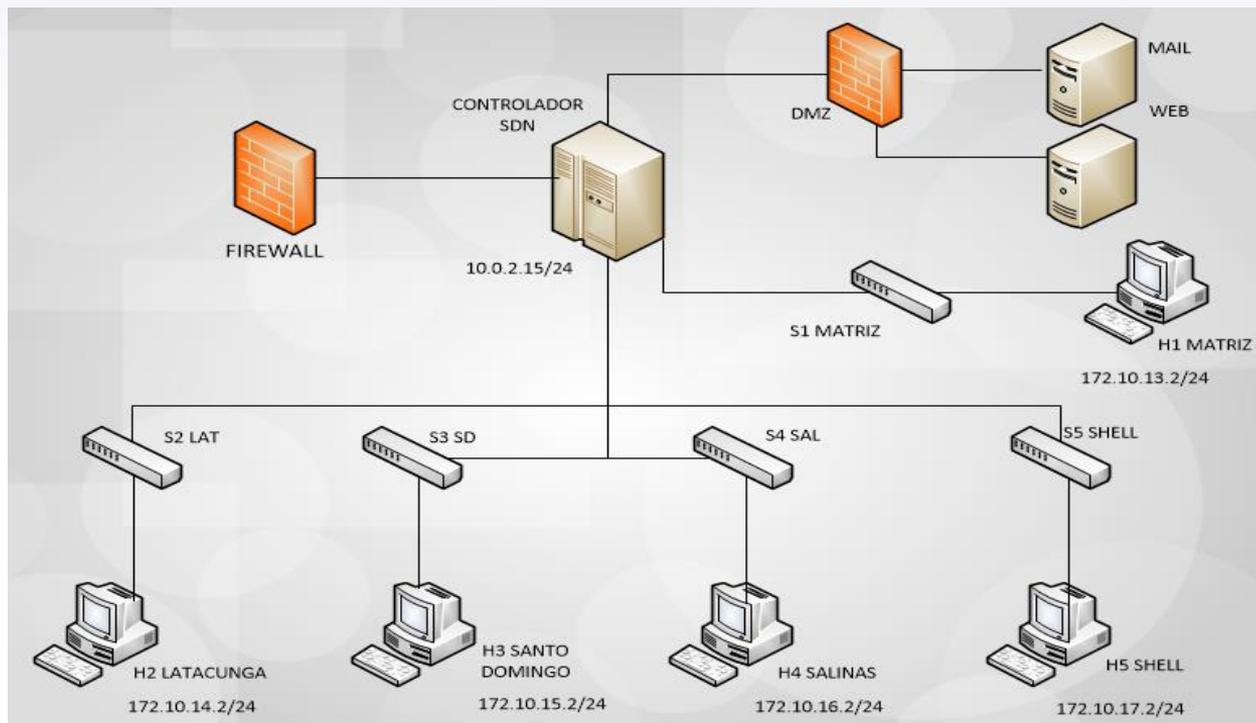
Característica	OpenDaylight <sup>a</sup>	Floodlight <sup>a</sup>	ONOS <sup>a</sup>	Ryu <sup>b</sup>
Lenguaje	Java	Java	Java	Phyton
Plataforma	Linux	Linux	Linux	Linux
	Windows	Windows	Windows	
	Mac	Mac	Mac	
Tipo de arquitectura	Distribuida	Centralizada	Distribuida	Arquitectura completa de framework
Protocolos southbound / northbound	OpenFlow	OpenFlow	OpenFlow	OpenFlow,
	REST	REST	Neutron	NETCONF
	NETCONF	JavaRPC		OF-config
Licencia	ELP	Apache	Apache	Apache

Tomado de: (Lasso & Puchaicela, 2021); (Pachés, 2020)

# SIMULACION



# TOPOLOGIA SDN A IMPLEMENTAR



## Topología de la red. Nomenclatura



Nomenclatura	Descripción
s1Matriz	Switch correspondiente a la Matriz
s2LT	Switch correspondiente a la sede Latacunga
s3SD	Switch correspondiente a la sede Santo Domingo
s4Sal	Switch correspondiente a la sede Salinas
s5Shell	Switch correspondiente a la sede Shell
Controlador SDN	Corresponde al controlador ubicado en la Matriz
h1, h2, h3, h4 y h5	hosts
-----	Northbound APIs
-----	Southbound APIs



# Comandos usados en mininet

```
floodlight@floodlight: ~/mininet/examples
File Edit View Search Terminal Help
mininet> help

Documented commands (type help <topic>):
=====
EOF      gterm  iperfudp  nodes      pingpair   py         switch
dpctl    help   link      noecho     pingpairfull  quit      time
dump     intfs  links     pingall    ports      sh         x
exit     iperf  net       pingallfull  px         source    xterm

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2
```

# Inicialización de la simulación y construcción de la red

```
floodlight@floodlight: ~/floodlight/example
File Edit View Search Terminal Help
floodlight@floodlight:~$ cd floodlight/
floodlight@floodlight:~/floodlight$ cd example/
floodlight@floodlight:~/floodlight/example$ ls
README graphDeps.py packetStreamerClientExample.py prueba2.py~
cli.py graphTopo.py prueba2.py
floodlight@floodlight:~/floodlight/example$ sudo python prueba2.py
[sudo] password for floodlight:
*** Adding controller
Unable to contact the remote controller at 10.0.2.15:6653
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h4 h1 h3 h2 h5
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> pingall
```

# Conexión al controlador vía web

Floodlight - Mozilla Firefox

Floodlight x Floodlight x +

localhost:8080 Search ☆ 📁 ⬇️ 🏠 🗨️ ☰

**Floodlight** [Dashboard](#) [Topology](#) [Switches](#) [Hosts](#)  Live updates

---

## Controller Status

**Hostname:** localhost:6633

**Healthy:** true

**Uptime:** 8304 s

**JVM memory bloat:** 84801016 free out of 204034048

**Modules loaded:** n.f.debugcounter.DebugCounterServiceImpl, n.f.accesscontrollist.ACL, n.f.testmodule.TestModule, n.f.ui.web.StaticWebRoutable, n.f.virtualnetwork.VirtualNetworkFilter, n.f.devicemanager.internal.DeviceManagerImpl, n.f.core.internal.OFSwitchManager, n.f.linkdiscovery.internal.LinkDiscoveryManager, n.f.loadbalancer.LoadBalancer, n.f.topology.TopologyManager, n.f.dhcpserver.DHCPserver, n.f.forwarding.Forwarding, n.f.flowcache.FlowReconcileManager, n.f.devicemanager.internal.DefaultEntityClassifier, n.f.storage.memory.MemoryStorageSource, n.f.jython.JythonDebugInterface, n.f.restserver.RestApiServer, org.sdnplatform.sync.internal.SyncManager, n.f.learningswitch.LearningSwitch, n.f.hub.Hub, n.f.firewall.Firewall, n.f.perfmon.PktInProcessingTime, n.f.core.internal.ShutdownServiceImpl, org.sdnplatform.sync.internal.SyncTorture, n.f.staticflowentry.StaticFlowEntryPusher, n.f.threadpool.ThreadPool, n.f.core.internal.FloodlightProvider, n.f.debugevent.DebugEventService,

---

## Switches (5)

DPID	IP Address	Vendor	Packets	Bytes	Flows	Connected Since
------	------------	--------	---------	-------	-------	-----------------

# Topología de la red vía web

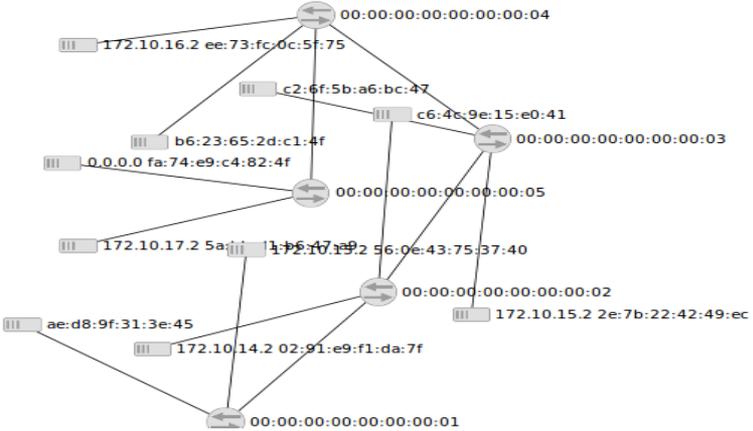
Floodlight  Dashboard Topology Switches Hosts  Live updates

### Switches (5)

DPID	IP Address	Vendor	Packets	Bytes	Flows	Connected Since
00:00:00:00:00:00:04	/10.0.2.15:37412	Nicira, Inc.	2553	411221	5	5/10/2022, 11:11:12 PM
00:00:00:00:00:00:05	/10.0.2.15:37411	Nicira, Inc.	1875	297470	7	5/10/2022, 11:11:12 PM
00:00:00:00:00:00:01	/10.0.2.15:37414	Nicira, Inc.	2482	411666	5	5/10/2022, 11:11:12 PM
00:00:00:00:00:00:03	/10.0.2.15:37413	Nicira, Inc.	2559	412064	5	5/10/2022, 11:11:12 PM
00:00:00:00:00:00:02	/10.0.2.15:37415	Nicira, Inc.	2576	417250	5	5/10/2022, 11:11:13 PM

Floodlight © Big Switch Networks, IBM, et. al. Powered by Backbone.js, Bootstrap, jQuery, D3.js, etc.

 Dashboard Topology Switches Hosts  Live updates



```
graph TD; S1((00:00:00:00:00:00:04)) --- S2((00:00:00:00:00:00:05)); S1 --- S3((00:00:00:00:00:00:01)); S1 --- S4((00:00:00:00:00:00:03)); S1 --- S5((00:00:00:00:00:00:02)); S2 --- S3; S2 --- S4; S2 --- S5; S3 --- S4; S3 --- S5; S4 --- S5;
```

# RESULTADOS PRUEBAS TECNICAS

```
mininet> pingall
*** Ping: testing ping reachability
h4 -> h1 h3 h2 h5
h1 -> h4 h3 h2 h5
h3 -> h4 h1 h2 h5
h2 -> h4 h1 h3 h5
h5 -> h4 h1 h3 h2
*** Results: 0% dropped (20/20 received)
```

```
*** Starting CLI:
mininet> h1 ping h2
PING 172.10.14.2 (172.10.14.2) 56(84) bytes of data.
64 bytes from 172.10.14.2: icmp_seq=1 ttl=64 time=15.5 ms
64 bytes from 172.10.14.2: icmp_seq=2 ttl=64 time=0.312 ms
64 bytes from 172.10.14.2: icmp_seq=3 ttl=64 time=0.052 ms
64 bytes from 172.10.14.2: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 172.10.14.2: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 172.10.14.2: icmp_seq=6 ttl=64 time=0.050 ms
^C
--- 172.10.14.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5001ms
rtt min/avg/max/mdev = 0.048/2.682/15.579/5.768 ms
```

# CAPTURA DE TRAFICO WIRESHARK

The screenshot shows the Wireshark interface running in a virtual machine named 'SDN\_G0 [Corriendo] - Oracle VM VirtualBox'. The main window title is 'Capturing from Loopback: lo [Wireshark 1.12.3 (Git Rev Unknown from unknown)]'. The filter bar is set to 'openflow\_v1'. The packet list pane displays a series of OpenFlow packets, alternating between 'Type: OFPT\_PACKET\_OUT' and 'Type: OFPT\_PACKET\_IN'. The status bar at the bottom indicates 'Loopback: lo: -live capture in pr... Packets: 66 - Displayed: 44 (66.7%)'.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::30ad:e5ff:fe93:bff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
2	0.000557000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
4	0.000785000	fe80::30ad:e5ff:fe93:bff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
5	0.001214000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
7	0.001508000	fe80::30ad:e5ff:fe93:bff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
8	0.001928000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
10	0.002079000	fe80::30ad:e5ff:fe93:bff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
11	0.002574000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
13	0.070236000	fe80::c23:24ff:fe54:6aff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
14	0.070773000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
16	0.071058000	fe80::c23:24ff:fe54:6aff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
17	0.071468000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
19	0.071623000	fe80::c23:24ff:fe54:6aff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
20	0.072037000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
22	0.196920000	fe80::1cd1:3bff:feb6:aaff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
23	0.197437000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
25	0.197780000	fe80::1cd1:3bff:feb6:aaff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
26	0.198177000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
28	0.496893000	fe80::c09:29ff:feb7:3eff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
29	0.497483000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT
31	0.497802000	fe80::c09:29ff:feb7:3eff02::fb		OpenFlow	191	Type: OFPT_PACKET_IN
32	0.498178000	127.0.0.1	127.0.0.1	OpenFlow	90	Type: OFPT_PACKET_OUT

# RESULTADOS PRUEBAS TECNICAS

*Medición de latencia con el comando ping*

<b>Nodos</b>	<b>Mínimo</b>	<b>promedio</b>	<b>Máximo</b>	<b>Desviación estándar</b>
<b>Matriz - Latacunga</b>	0.048	2.682	15.579	5.768
<b>Matriz - Santo Domingo</b>	0.038	7.114	42.092	15.643
<b>Matriz - Salinas</b>	0.0695	6.211	36.546	13.566
<b>Matriz - Shell</b>	0.054	6.420	37.722	13.999

# *Ventajas de la implementación*

- ▶ El sistema hace posible el control y la administración dinámica de la red.
- ▶ Es posible programar mediante códigos la infraestructura de la red.
- ▶ Habilitación de la programación de la red bajo demanda.
- ▶ Alta velocidad en provisión de clientes y servicios nuevos.
- ▶ Baja inversión inicial.
- ▶ Bajos costos de operación.
- ▶ Despliegue y control de la red simplificados.
- ▶ Mayor seguridad y menores vulnerabilidades en la red.
- ▶ Alta usabilidad.



Respecto del análisis económico realizado, se determinó que la red SDN simulada para el backbone de la Universidad de las Fuerzas Armadas es factible económicamente con una inversión de \$19.357,81, presentando los indicadores TIR de 29,09%, VAN \$17.713,14 y ROI de 6,95 (primer año). Esto indica que la implementación de la red es factible económicamente y que representa una optimización respecto de la actual gestión de datos.

Se ha determinado que el análisis de costo/beneficio realizado indica que es favorable para la institución actualizar e implementar un sistema de red SDN, ya que se constató que este es de USD 1,07. Esto, además de ser económicamente factible, permitirá manejar con mayor seguridad la red, controlar y administrar de manera eficiente mediante la seguridad centralizada y obtener un sistema escalable y flexible para integrar nuevas tecnologías a medida que surjan.



Respecto a la factibilidad técnica ofrecida por la red SDN, se conoció que presentó un buen desempeño en las pruebas realizadas. Respecto de la conectividad, esta se comprobó entre las cuatro sedes y la casa matriz, presentándose adecuada en todos los casos. El desempeño, por otra parte, se mostró adecuado; la latencia obtuvo valores promedio entre 2.682 y 6.420, con desviaciones estándar de entre 5.768 y 15.643. La pérdida de paquetes presentó en todos los casos analizados un 0%, mientras que la tasa de transferencia de datos fue igualmente adecuada. Por lo tanto, también es factible a nivel técnico.



Se recomienda implementar la red SDN simulada en la Universidad de las Fuerzas Armadas debido a los beneficios que esta representa y debido a su factibilidad económica, con lo que la gestión de la información puede ser más eficiente y segura. De igual manera, se recomienda implementar un sistema de gestión de información, que comprenda un seguimiento y evaluación continuos del funcionamiento. Las redes SDN son una tecnología relativamente nueva que está en constante avance y redescubrimiento. Por tan razón, es posible que el funcionamiento de esta presente brechas respecto de algunos rendimientos y/o costos de implementación debido a la necesidad de adaptar el hardware. Para esto, es importante que los administradores de la red estén constantemente evaluando su desempeño y verificando la interoperabilidad entre la red y otros servicios y tecnologías.



Se recomienda a la Empresa Pública de Movilidad del Norte que **ejecute y acoja las recomendaciones** realizadas en el informe de Evaluación Técnica Informática, a fin de que pueda **obtener la mejora** a corto plazo con procesos actuales y con los próximos que se deseen implementar.

# Preguntas?

Contactos:

- ▶ [drojas@espe.edu.ec](mailto:drojas@espe.edu.ec)

