



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA  
DE SISTEMAS**

**ARQUITECTURAS DE REDES DE  
COMPUTADORAS DEFINIDAS POR SOFTWARE:  
REVISIÓN BIBLIOGRÁFICA.**

**INFORME DE TRABAJO DE INVESTIGACIÓN:  
PARA OPTAR EL GRADO ACADEMICO DE BACHILLER  
EN INGENIERÍA DE SISTEMAS**

**Autor:**

**Santisteban Ynga, Bicry Raúl**

**Asesor:**

**Mg. Ing. Mejía Cabrera, Ivan**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Pimentel – Perú**

**2020**

## DEDICATORIA

Para las personas más importantes y especiales en mi vida: mi adorada madre, amadísimas esposa e hija, quienes, a diario con sus muestras de cariño, apoyo incondicional, paciencia y comprensión, se han convertido en la fortaleza de mi vida, la motivación mas hermosa que puede tener un hombre para salir adelante.

*¡Miren! Los hijos son una herencia de parte de  
Jehová; el fruto del vientre es un galardón”.*

Salmos 127:3

## AGRADECIMIENTO

*“Confía en Jehová con todo tu corazón  
y no te apoyes en tu propio entendimiento.  
6Tómalo en cuenta en todos tus caminos,  
y él hará rectas tus sendas.”.*  
*Proverbios 3:5-6*

Agradezco en primer lugar a mi Padre Celestial, Jehova Dios, pues sino fuera por su voluntad, nada de esto se habría logrado.

Tambien agradezco de manera muy especial a mi compañera de vida, mi esposa; su constante apoyo, animo y paciencia, es fundamental para lograr todo lo que nos proponamos.

A mi asesor, por su respaldo en el proyecto, su conocimiento, experiencia, tiempo y dedicación garantizaron el poder cumplir con los objetivos trazados en mi carrera profesional.

Espero sinceramente ser reciproco con cada uno de las personas que han sumado para llegar a la meta.

## RESUMEN

Nos encontramos en el umbral de la quinta revolución industrial (5RI), donde emerge una tecnología dinámica, adaptable, escalable y rentable que desvincula las funciones de control y datos en la infraestructura de la red.

Las Redes Definidas por Software (SDN), por sus siglas en inglés) adoptan una arquitectura que proporciona la programabilidad y centralización de la red. Esta investigación se centrará en el núcleo o cerebro de las SDN, conocida como controladores, son aquellos que poseen la inteligencia basada en software y está compuesta por tres capas: Infraestructura, Control y Aplicaciones.

Según estudios previos y las recomendaciones de la Open Networking Foundation (ONF), se detalla los componentes, funcionamiento y elementos para su selección. El despliegue en un entorno simulado basado en los indicadores de rendimiento, presentó resultados que permitirán a los administradores de red seleccionar el óptimo controlador según sus requerimientos.

Los resultados más notables, indican que la arquitectura SDN soportado con el controlador ONOS presenta mejor rendimiento.

**Palabras Clave** – Redes definida por software, SDN, controlador, capa de control, capa de datos.

## **ABSTRACT**

We are at the threshold of the fifth industrial revolution (5RI), where a dynamic, adaptable, scalable and cost-effective technology emerges that separates control and data functions in the network infrastructure.

Software Defined Networks (SDN) adopt an architecture that provides programmability and centralization of the network. This research will focus on the core or brain of the SDNs, known as controlled, are those that have software-based intelligence and is composed of three layers: Infrastructure, Control and Applications.

According to previous studies and the recommendations of the Open Networking Foundation (ONF), the components, operation and elements for their selection are detailed. The deployment in a simulated environment based on performance indicators, presented results that will allow network administrators to select the optimal controller according to their requirements.

The most notable results indicate that the SDN architecture supported with the ONOS controller has better performance.

**Keywords** - Software-defined networks, SDN, controller, control layer, data layer.

## CONTENIDO

DEDICATORIA.....	2
AGRADECIMIENTO .....	3
RESUMEN .....	4
ABSTRACT .....	5
ÍNDICE DE FIGURAS .....	8
ÍNDICE DE TABLAS .....	10
I. INTRODUCCIÓN .....	11
1.1. Antecedentes de estudio.....	13
1.2. Planteamiento del problema de Investigación. ....	14
1.3. Objetivos. ....	14
1.1.1. Objetivo general. ....	14
1.1.2. Objetivos específicos.....	15
1.4. Marco teórico conceptual.....	15
1.5. Nuevas tendencias en la Red.....	15
1.1.3. Redes basadas en intención .....	17
1.1.4. Computación en la Nube .....	18
1.1.5. BIG DATA .....	19
1.1.6. 5G .....	19
1.1.7. IoT .....	20
1.1.8. BYOD.....	20
1.6. Limitaciones de la Arquitectura tradicional.....	20
1.7. Redes Definidas por Software.....	22
1.1.9. Arquitectura.....	22
1.1.10. Interfaces .....	26
1.1.11. Modos de implementación.....	26
1.1.12. OpenFlow .....	29
1.1.13. Mininet.....	30
1.1.14. Cbench.....	30
1.1.15. Controlador SDN.....	31
1.1.16. Beneficios de las SDN.....	38
II. MATERIAL Y MÉTODOS.....	38
2.1. Método de Investigación.....	38
2.2. Plan de Investigación .....	39
2.2.1. Interrogantes de la Investigación.....	39
2.2.2. Protocolo de revisión.....	39

2.2.3.	Validación del Protocolo de revisión.....	40
2.3.	Procedimiento de la Investigación .....	40
2.2.4.	Identificar las investigaciones relevantes .....	40
2.2.5.	Seleccionar estudios primarios .....	42
2.2.6.	Evaluar la calidad de los estudios.....	47
2.2.7.	Extraer datos requeridos .....	49
2.2.8.	Sintetizar datos .....	50
2.4.	Documentos de la Investigación .....	62
2.2.9.	Validar el informe.....	62
III.	RESULTADOS.....	62
3.1.	Análisis .....	63
IV.	CONCLUSIONES .....	84
V.	RECOMENDACIONES .....	85
VI.	REFERENCIAS.....	103

## ÍNDICE DE FIGURAS

Figura 1 Crecimiento global de dispositivos móviles y conexiones .....	11
Figura 2 Demanda significativa de ancho de banda y video en el hogar conectado del futuro .....	12
Figura 3 Datos del consumo del Internet en Perú.....	15
Figura 4 Crecimiento del tráfico IP global. ....	16
Figura 5 Tecnologías que impulsan las nuevas demandas de la red .....	16
Figura 6 Red basada en intención.....	17
Figura 7 Modelo de servicios en la nube.....	18
Figura 8 Modelo de red en la nube según la ITU-T .....	19
Figura 9 Preocupaciones más importantes del estado actual de las redes. ....	21
Figura 10 Topolog´ia en Paradigma Convencional.....	22
Figura 11 Arquitectura SDN en capas y planos .....	23
<i>Figura 12</i> Arquitectura en capas de SDN RFC 7426 (2015) .....	24
Figura 13 Arquitectura SDN UIT-T Y.3300 .....	25
Figura 14 Arquitectura SDN según ONF .....	25
Figura 15 SDN Open Source via OpenFlow .....	27
Figura 16 SDN Open Source via Overlay .....	27
<i>Figura 17</i> SDN Open Source via APPs.....	28
Figura 18 Comparación de las alternativas de implementación .....	28
Figura 19 Componentes de SDN/OpenFlow .....	30
Figura 20 Arquitectura habitual de un controlador SDN. ....	32
Figura 21 Arquitectura del controlador Opendaylight .....	34
Figura 22 Arquitectura del controlador ONOS. ....	37
Figura 23 Método de revisión bibliográfica .....	38
Figura 24 Búsqueda inicial en la base de datos IEE.....	40
Figura 25 Resultado crudo de la búsqueda IEEE .....	41
Figura 26 Resultado crudo de la búsqueda en JabRef.....	41
<i>Figura 27</i> Búsqueda de cadena en el año 2015 al 2020 .....	42
Figura 28 Resultados filtrados en los últimos 5 años .....	46
Figura 29 Valoración de los artículos según el número de citas y la cantidad de vistas completas.....	62

Figura 30 Cantidad de artículos por año aplicando el filtro crudo tal cual arroja la regla de búsqueda.....	63
Figura 31 Resultados de búsqueda con filtro.....	67

## ÍNDICE DE TABLAS

Tabla 1 Principales controladores SDN de código abierto. ....	32
Tabla 2 Resultados más significativos sobre el rendimiento de controladores SDN. ....	43
Tabla 3 Resumen de impacto de los artículos (2015-2020). ....	47
Tabla 4 Valoración De Las Palabras Clave De Acuerdo A Inspec .....	51
Tabla 5 <i>Resultados de búsqueda de artículos</i> .....	64
Tabla 6 Autores inmersos en la investigación SDN .....	68
Tabla 7 <i>Métricas y elementos para evaluación de controladores</i> .....	71
Tabla 8 <i>Hallazgos importantes – Despliegue de las redes SDN</i> .....	75
Tabla 9 <i>Comparación de características</i> .....	84
Tabla 10 Matriz De Resumen De La Revisión Bibliografica De Los Articulos De Interes .....	86

## I. INTRODUCCIÓN

El acelerado avance en la tecnología, la proliferación de nuevas tendencias y la necesidad de la convergencia en los servicios (por ejemplo, en la nube) ha elevado la complejidad en la implementación y administración de la infraestructura de la red. La Open Networking Foundation (ONF,2012) identificó que la red tradicional es una red estática y compleja, contiene políticas inconsistentes, presenta falta de escalabilidad y dependencia de los proveedores de dispositivos. Asimismo, según “Cisco Annual Internet Report, 2018-2023”; en sus proyecciones cuantitativas sobre el crecimiento de los usuarios, dispositivos, conexiones de Internet y análisis global de banda ancha fija, WiFi y redes móviles 3,4 y 5G, expone lo siguiente para el 2023 en América Latina (LATAM) tendrá:

470 millones de usuarios de Internet (70 por ciento de la población regional).

520 millones de usuarios móviles (78 por ciento de la población regional).

2,1 mil millones de dispositivos / conexiones en red.

49 por ciento de todos los dispositivos en red conectados a dispositivos móviles y el

51 por ciento estará conectado o conectado a través de Wi-Fi

La velocidad promedio de banda ancha fija de LATAM alcanzará los 59.3 Mbps

La velocidad promedio de conexión móvil de LATAM alcanzará los 35.0 Mbps.

La velocidad promedio de Wi-Fi de LATAM desde dispositivos móviles alcanzará los 28.8 Mbps.

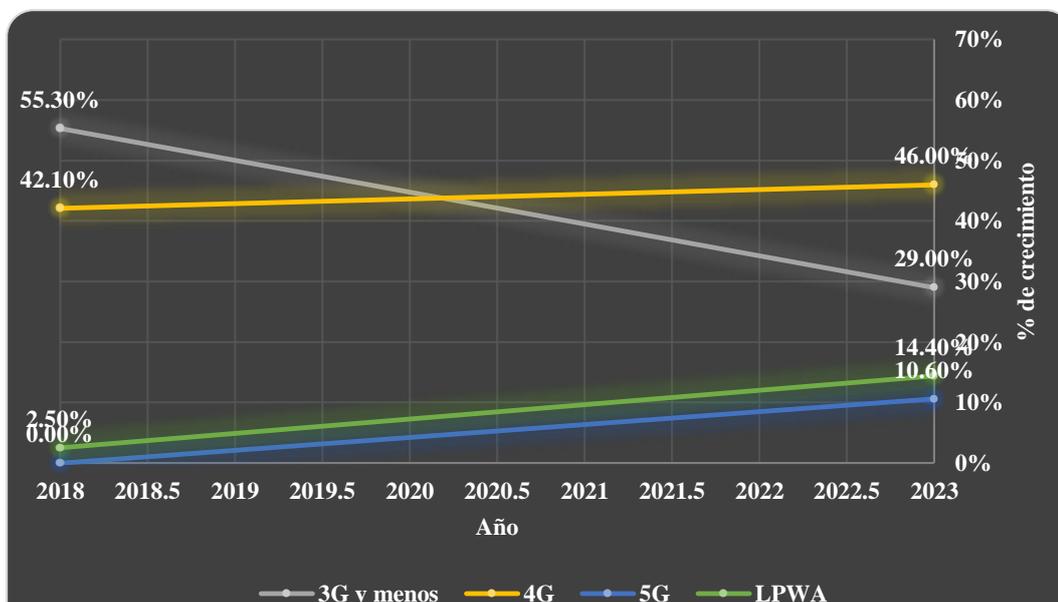


Figura 1 Crecimiento global de dispositivos móviles y conexiones

Fuente: Cisco Annual Internet Report, 2018-2023

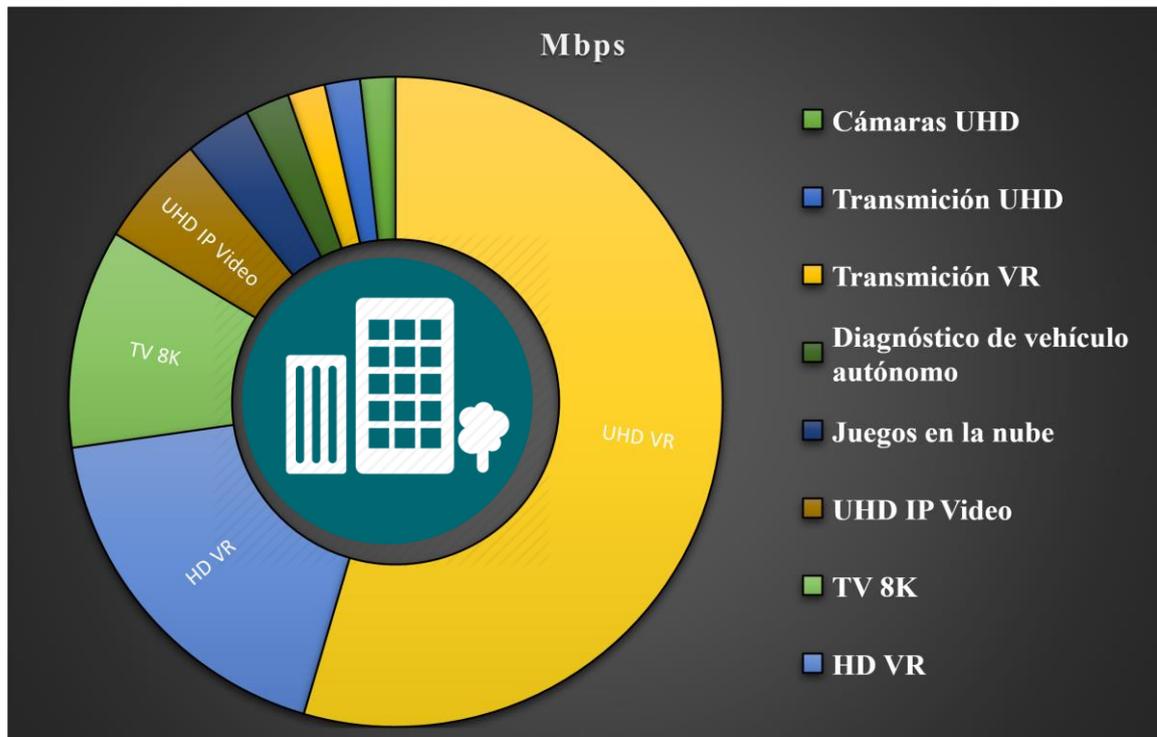


Figura 2 Demanda significativa de ancho de banda y video en el hogar conectado del futuro  
Fuente: Cisco Annual Internet Report, 2018–2023

En los  ltimos a os, los administradores de red se encuentran en la disyuntiva de migrar hacia una infraestructura de red centralizada y programable. Las Redes Definidas por Software; es una arquitectura emergente din mica, administrable, rentable, y adaptable, esta arquitectura separa las funciones de control y de env o de paquetes de la red, permitiendo que el control de la red sea directamente programable y que la infraestructura subyacente pueda ser abstra da para las aplicaciones y los servicios de red. En este trabajo se realizar  una revisi n bibliogr fica para comprender su arquitectura, componentes, despliegue y beneficios, sustentados en la revisi n de trabajos de investigaci n que hayan analizado o comparado el rendimiento del llamado controlador SDN, el encargado de gestionar los dispositivos por software. Se busca generar conocimiento a fin de proporcionar una gu a clave para la toma de decisiones y la elecci n m s  ptima para un escenario espec fico.

### **1.1. Antecedentes de estudio.**

Hasta la actualidad, se han realizado y presentado trabajos de evaluación comparativa de los controladores SDN.

(A. Tootoonchian, 2012) es uno de los pioneros investigadores que realizo una evaluación del rendimiento de controladores SDN NOT-MT, Maestro y Beacon. Se entiende que para la actualidad están desfasados, por ello no son utilizados para implementar la arquitectura.

(R. Khondoker, 2014) realizo una interesante comparación de los controladores Ryu, Pox, Trema, Floodlight y OpenDayLight; capturo las propiedades de interfaces hacia el sur, soporte de virtualización, contar con una GUI amigable, soporte en API REST, la velocidad en su codificación, el lenguaje de programación que fue desarrollado, capacidad de modularidad, el sistema operativo, TLS, compatibilidad con el protocolo OpenFlow y OpenStack. Su metodología se baso en la toma de decisiones de criterios multiples (MCD) y el proceso analítico de jerarquía (AHP).

(S. A. Shah, 2013) considera la arquitectura de lotes estáticos y el procesamiento por lotes adaptativos. como patrones para mejorar las implementaciones de los controladores, incluso afecta si deseamos insertar un nuevo diseño. Los investigadores se valieron de los controladores Beacon, NOX, Maestro y Floodlight, y los analizaron bajo las métricas de rendimiento latencia y escalabilidad, generando un banco de pruebas, con expectantes resultados.

(M. P. Fernandez, 2013) plantea la importancia del modo de operación, es decir las actuaciones en modo reactivo y proactivo de los controladores NOX, POX, Trema y Floodlight, el estudio recomienda adoptar el modo proactivo que permite instalar las reglas de reenvio en el mismo interruptor, a diferencia del modo reactivo, donde las reglas se instalan en el momento de recibir las solicitudes llamadas PACKET\_IN.

(S. Rowshanrad, 2016) realiza una evaluación y comparación de los controladores Floodlight y OpenDayLight en base a los parámetros de calidad de servicio QoS, que

incluye retraso y pérdida de paquetes en las diferentes topologías de red, y bajo los diferentes niveles de carga de tráfico.

(A. Shalimov, 2013) evalúa el desempeño de los controladores NOX, POX, Beacon, Floodlight, MuL, Maestro, Ryu en función de su latencia, rendimiento, escalabilidad, fiabilidad y seguridad

(D. Erickson, 2013) sostiene que el lenguaje de programación utilizado por un controlador SDN tiene un impacto significativo en su desempeño. Establece que Java al soportar plataforma multihilo supera a Python, C#, C, C++, quienes presentan deficiencias como demora en los tiempos de ejecución, mala gestión de memoria y no son compatibles con multi plataformas. El estudio evalúa los controladores Trema / C, POX / Python, Ryu / Python, Maestro / Java y Floodlight / Java y concluye que Beacon, presenta el mejor desempeño, y lo hace por estar soportado en Java.

(Salman, 2016) se encargó de evaluar los controladores MUL, Beacon, Maestro, ONOS, Ryu, OpenDayLight, Floodlight, NOX, IRIS, basado en Libfluid, y POX, usando las métricas de latencia y rendimiento, bajo un entorno de variación de interruptores y número de hilos a la instancia del controlador.

## **1.2. Planteamiento del problema de Investigación.**

¿Cuál es el estado del conocimiento concerniente a las Redes Definidas por Software, su arquitectura y rendimiento?

## **1.3. Objetivos.**

### **1.1.1. Objetivo general.**

Realizar una revisión del material bibliográfico sobre las Redes Definidas por Software, su arquitectura y elementos para su evaluación.

### 1.1.2. Objetivos específicos.

Elaborar el plan de investigación

Desarrollar el procedimiento de investigación

Generar la documentación de la investigación

### 1.4. Marco teórico conceptual.

Los fundamentos teóricos que soportan el proyecto son necesarios para identificar los componentes y beneficios que ostenta las Redes Definidas por Software, en especial, los controladores que son los encargados de la inteligencia y gestión centralizada de la red.

### 1.5. Nuevas tendencias en la Red

Las nuevas tecnologías en la red continúan requiriendo recursos de red y de administración más escalables debido a la necesidad de mayor velocidad de transmisión. Según el reporte producido por Data Reportal (Simon Kemp, equipo de Kepios, Hootsuite y We Are Social) especifica en Perú lo siguiente:



Figura 3 Datos del consumo del Internet en Perú

Según (Cisco, 2019) el tráfico de internet está en aumento y se estima que en el año 2021 el tráfico de internet llegara a 278 exabytes por mes, estos incrementos del tráfico de internet hacen reevaluar los enfoques de la arquitectura de red tradicional.

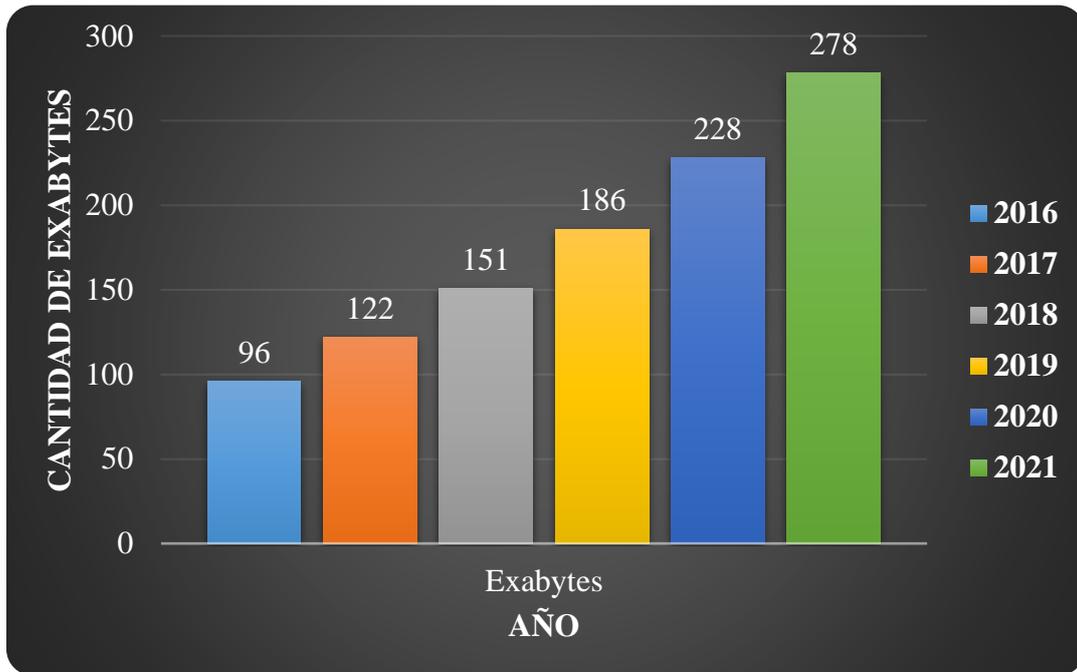


Figura 4 Crecimiento del tráfico IP global.

Según (Cisco, 2019), para el año 2022, los videos de Internet representarán el 82% de todo el tráfico del Internet empresarial, el tráfico de VR/AR se multiplicará por doce y el tráfico de videovigilancia de Internet se multiplicará por siete

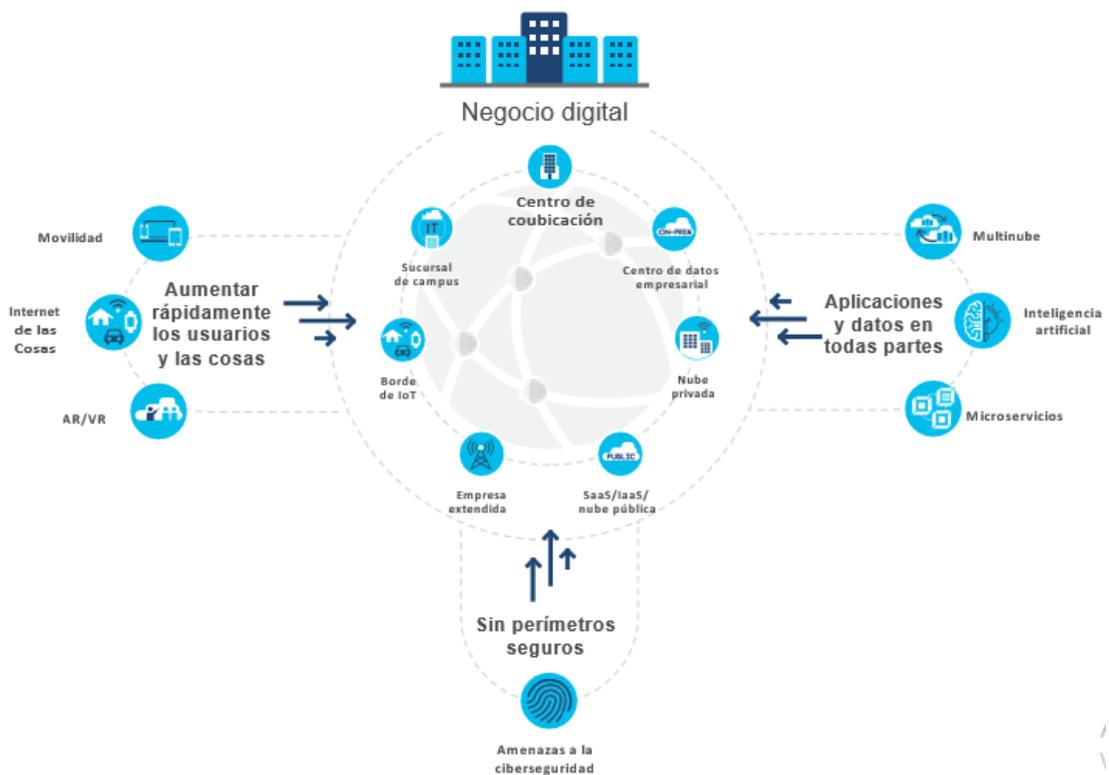


Figura 5 Tecnologías que impulsan las nuevas demandas de la red

### 1.1.3. Redes basadas en intención

Para el 2025, se proyecta que una empresa podría sustentar su requerimiento de red, soportado en el lenguaje natural, que se encargará de traducir de forma automática el requerimiento en acciones y políticas automatizadas, sin afectar otro servicio en ejecución. Esta propuesta se le conoce como la red basada en intención. (Cisco, 2019)

Ejemplo: Una organización requiere sensores ópticos de IoT inalámbricos para admitir una innovación empresarial que debe entregarse a través de una aplicación soportada en la tecnología de realidad aumentada. Así es como la necesidad y la intención del negocio se traducirían en una acción de red automatizada.

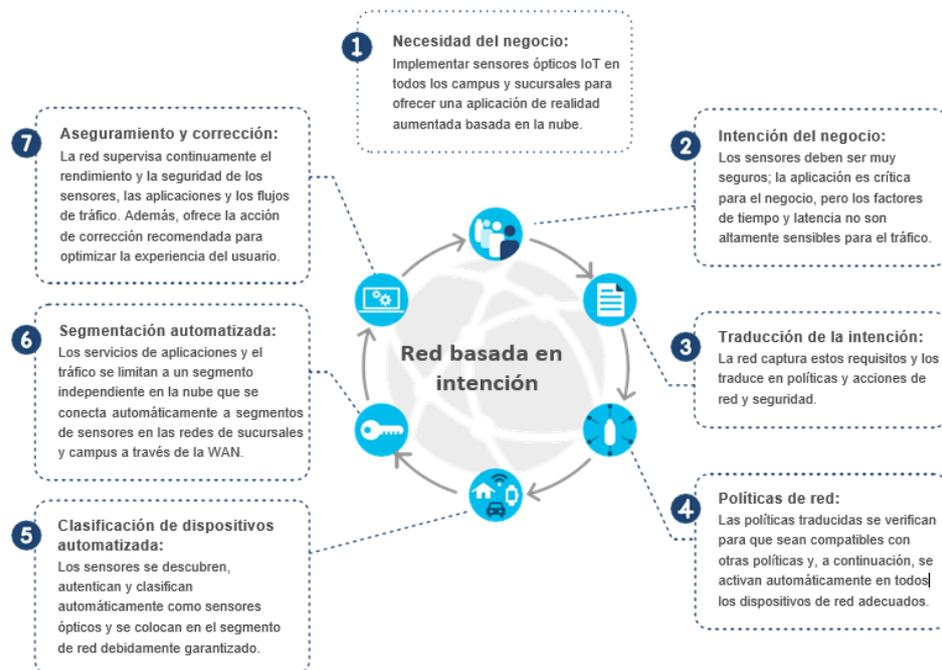


Figura 6 Red basada en intención

Según Gartner: “Aproximadamente el 70% de las labores de un administrador de red las realiza de forma mecánica y manual, lo que repercute en pérdida de tiempo, costo y la probabilidad de errores y reduce la flexibilidad”

### 1.1.4. Computación en la Nube

“La nube es un sistema computacional inteligente, complejo y poderoso en el cielo, al cual la gente simplemente se conecta” (Cloud Computing: Eyes on the Skies, Steve Hamm, Businessweek, abril 24 de 2008.)

Los servicios en la nube se presentan en tres formas: Software como servicio (SAAS), Plataforma como servicio (PAAS) e Infraestructura como servicio (IAAS). Podemos mencionar varios beneficios tales como: Reducción de costos, Adaptable y variable dependiendo del crecimiento de la empresa.

<i>On premises</i>	<i>Infraestructura como Servicio (IAAS)</i>	<i>Plataforma como Servicio (PAAS)</i>	<i>Software como Servicio (SAAS)</i>
Aplicación	Aplicación	Aplicación	Aplicación
Datos	Datos	Datos	Datos
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
Sistema Operativo	Sistema Operativo	Sistema Operativo	Sistema Operativo
Virtualización	Virtualización	Virtualización	Virtualización
Servidores	Servidores	Servidores	Servidores
Almacenamiento	Almacenamiento	Almacenamiento	Almacenamiento
Redes	Redes	Redes	Redes

Figura 7 Modelo de servicios en la nube

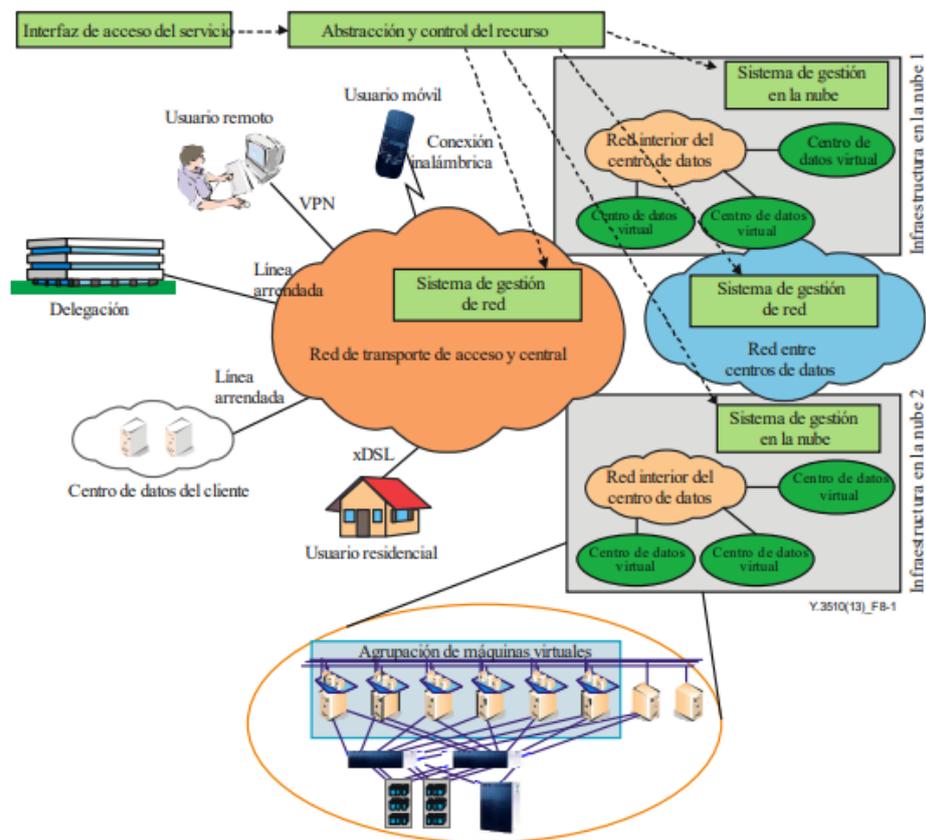


Figura 8 Modelo de red en la nube según la ITU-T

### 1.1.5. BIG DATA

En la actualidad, las empresas se presentan ante un gran desafío, el crecimiento de los datos en volumen, velocidad y variedad, es decir, han superado su capacidad de almacenamiento debido en gran parte al avance de la tecnología y la variedad de fuentes de datos. Se enfoca en el tratamiento y análisis de repositorios de datos de gran magnitud, por ejemplo, las bases de datos de un banco, etc.

### 1.1.6. 5G

5G es la quinta generación de redes de comunicaciones móviles basada y cimentada en la red 4GLTE. Cuenta con una velocidad de transferencia de 5000 megabits por segundo aproximadamente, mucho mayor que la 4GLTE. Aparece como respaldo a la tecnología 4G debido que se encuentra saturada y en su total capacidad.

### **1.1.7. IoT**

En un momento no muy lejano, los objetos que nos rodean serán inteligentes, es decir, podrán estar conectados a Internet y por ende poder administrarlos desde cualquier ubicación. Todo estará conectado y esta tecnología sigue evolucionando. Utiliza sensores interconectados entre los objetos, y aunque en la actualidad generan tráfico pesado, por ejemplo, las cámaras de seguridad, se está trabajando en optimizar los protocolos y sensores.

### **1.1.8. BYOD**

Se ha convertido en una tendencia, porque permite a los usuarios utilizar su dispositivo móvil para atender actividades del trabajo, por ejemplo, el correo electrónico, asegura que el usuario este siempre disponible y conectado, pero a su vez representa una brecha para la seguridad de la información, porque el usuario se encuentra más propenso a un ataque cibernético.

## **1.6. Limitaciones de la Arquitectura tradicional.**

Las arquitecturas tradicionales de redes no son óptimas para satisfacer todos los requerimientos de las empresas y los usuarios finales. En la actualidad, la preocupación sobre el estado de las redes tradicionales ha escalado a todas las empresas, según Gigaom Resarch en orden ascendente son:

Lento aprovisionamiento de nuevos clientes o servicios.

Complejidad de la red.

Retraso en el despliegue de nuevo servicios.

Adopción a tiempo de nuevas tecnologías.

Conectividad de la red.

Costo de la operación de la red.

Costo del equipamiento de la red.

Niveles de servicio de la red.

Seguridad y vulnerabilidades de la red.



Figura 9 Preocupaciones más importantes del estado actual de las redes.

La Open Networking Foundation (Onf,2012) nos indica de cuatro limitaciones generales de la arquitectura de red tradicional: Arquitectura estática y compleja: Al carecer de la capacidad para administrar los diferentes niveles de calidad de servicio, resulta en inconvenientes para atender los requerimientos de los usuarios, también cuando un dispositivo de red se agrega a la red, el administrador tiene que configurar manualmente en varios o todos los dispositivos de la infraestructura. Lo que conlleva a realizar las actualizaciones en la lista de control de acceso (ACL), configuración de VLAN, configuración de QoS y otros ajustes relacionados con el protocolo que sean necesarios.

Asimismo, según la Open Data Center Alliance (Odca,2014), las principales limitaciones de las redes tradicionales se relacionan con: Políticas inconsistentes, escasez de escalabilidad, dependencia de los proveedores, por ejemplo, si contamos con una red con dispositivos Cisco, actualmente es tedioso converger con otros dispositivos de otras marcas, lo que obliga a los usuarios a adquirir dispositivos del mismo proveedor.

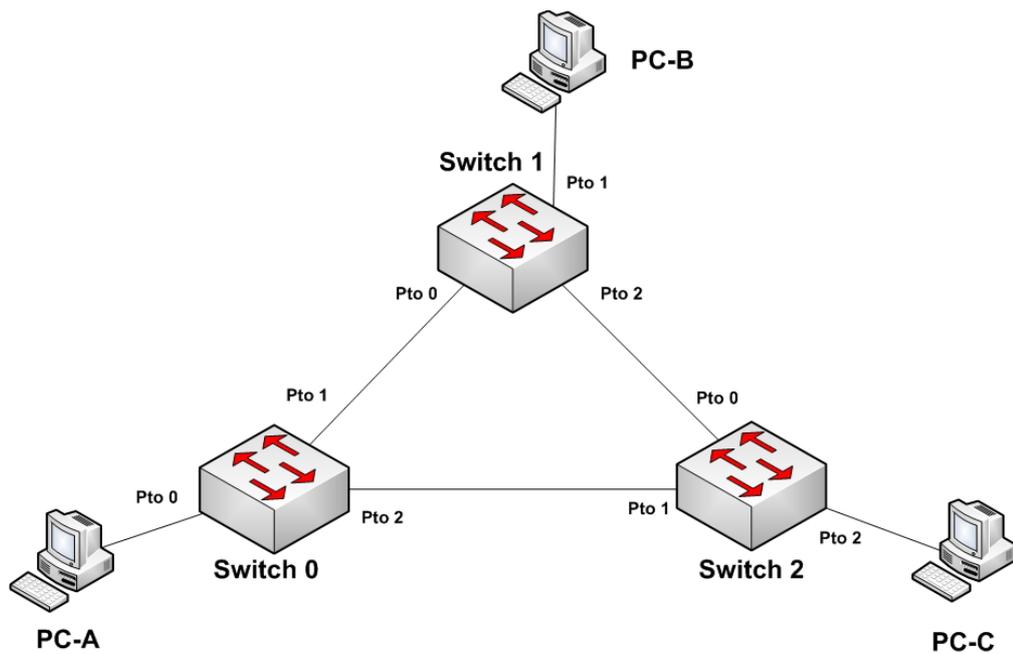


Figura 10 Topología en Paradigma Convencional

## 1.7. Redes Definidas por Software

Con el transcurso de los años, las redes SDN se han convertido en una de las tecnologías en redes de datos y comunicación, orientadas a la automatización y sistematización más sólida en el mercado. El modelo SDN, separa los planos de control y datos, abstrae la inteligencia de los dispositivos y la centraliza en su componente llamado controlador, quien toma las funciones de gestor y centralizador lógico, sumándole eficiencia, flexibilidad y programabilidad a la arquitectura.

### 1.1.9. Arquitectura

La emergente arquitectura de las Redes Definidas por Software, desempotra la función de reenvío (forwarding) de la función de control, lo hace con el objetivo de abstraer la inteligencia de los dispositivos de red (router, switch, entre otros), migrarla a su capa centralizadora y desde ahí programar las políticas e instrucciones, como el cálculo de las rutas, filtrado de paquetes, enrutamiento y demás indicaciones del flujo a los dispositivos subyacentes, en sus funciones de reparto y distribución. (All, 2014)

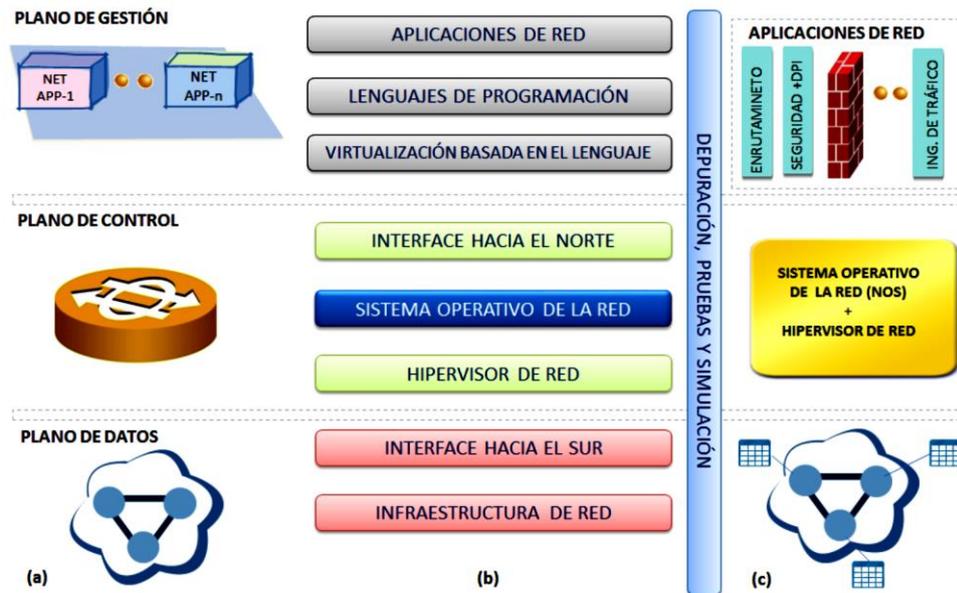


Figura 11 Arquitectura SDN en capas y planos

Según la IETF - RFC 7426. la arquitectura SDN se compone de lo siguiente:

**Application Plane:** Se conforma por las APPs y servicios que contienen las instrucciones del comportamiento de la red.

**Control Plane:** Es el ente principal que asume las decisiones y define la metodología de reenvío de paquetes y selección de dispositivos.

**User/Data Plane:** Es el encargado de ejecutar las acciones de reenviar, descartar o modificar los paquetes, tomando en cuenta las indicaciones determinadas por la capa superior.

**Management Plane:** Es el administrador encargado del monitoreo, configuración y mantenimiento de las operaciones.

**Operation Plane:** Especifica el estado de los dispositivos de red y sus componentes.

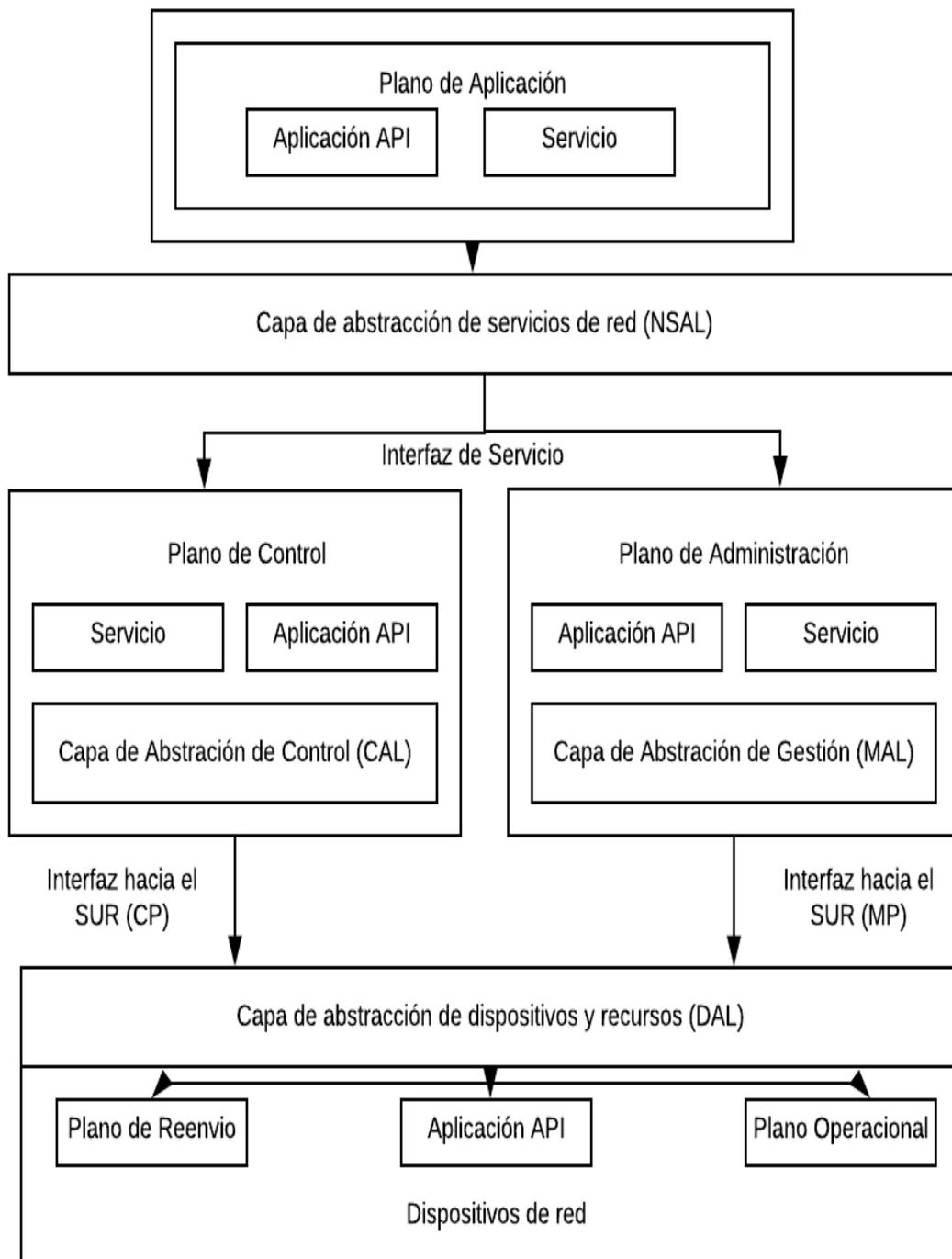


Figura 12 Arquitectura en capas de SDN RFC 7426 (2015)

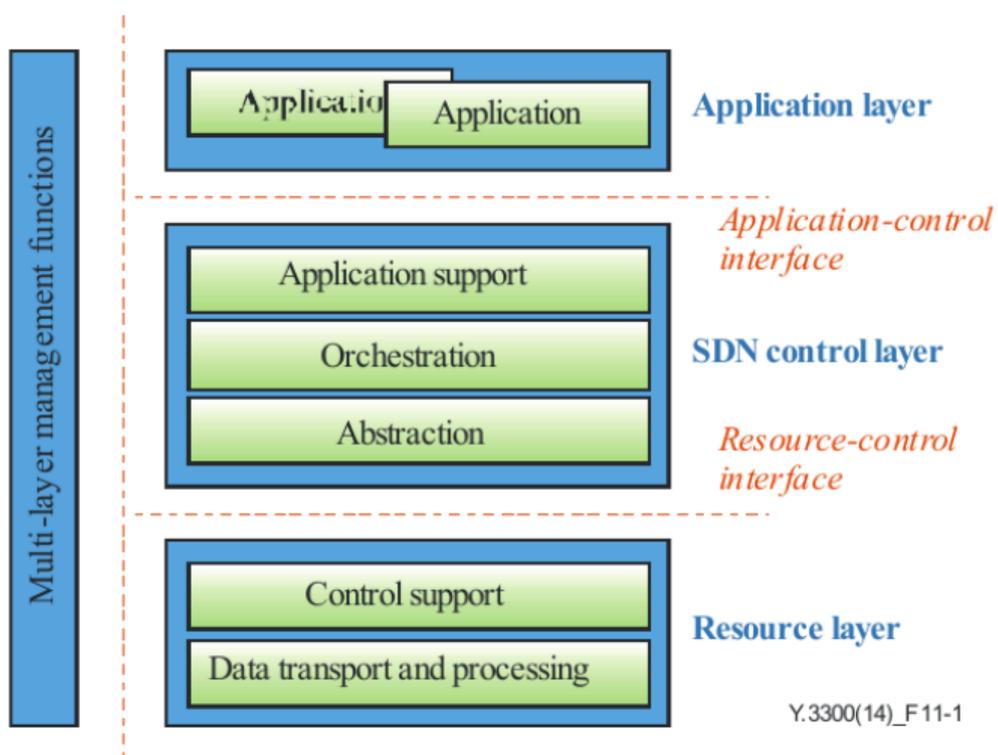


Figura 13 Arquitectura SDN UIT-T Y.3300

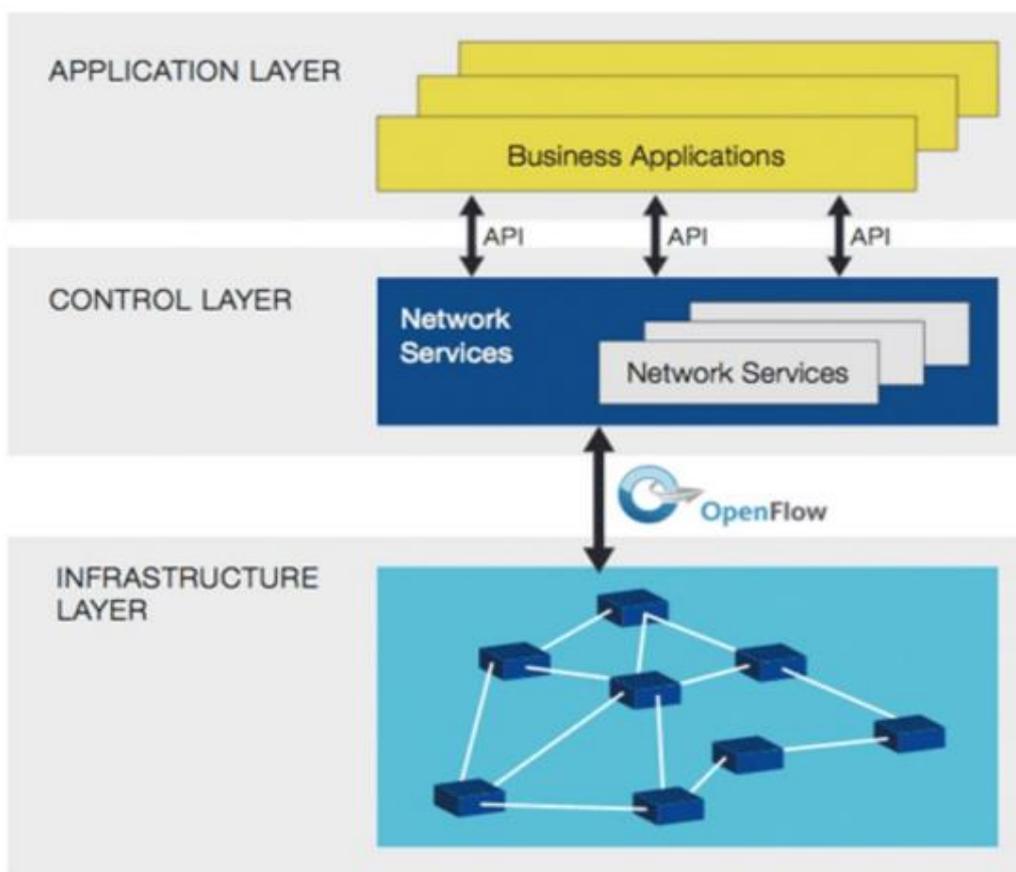


Figura 14 Arquitectura SDN según ONF

La arquitectura según la ONF se divide en tres capas: Infraestructura, Control y Aplicaciones. La interfaz de control del plano de datos funciona por medio del protocolo OpenFlow, que es el medio de comunicación entre el controlador SDN y los dispositivos de red, entre ellos router y switches, que son gestionados por la capa principal de la arquitectura, conocida como capa de control, equivale al sistema operativo de la red. Dentro de sus funciones principales esta la traducción de los requerimientos de la capa de Aplicación hacia los elementos de red, adicional a ello proporciona información como estadísticas y eventos. La capa de Aplicaciones se comunica hacia el controlador mediante las APIs hacia el norte, puede encargarse de los servicios de red, como calidad de servicio QoS o aplicaciones de negocio.

#### **1.1.10. Interfaces**

##### 1.1.1.1. Southbound Interface

Su importancia radica en el protocolo que utiliza para la comunicación entre los dispositivos de Plano de Datos y los elementos del Plano de Control, es donde se programan las instrucciones del reenvío en los dispositivos. Ejemplo: protocolo OpenFlow.

##### 1.1.1.2. Northbound Interface

Abstrae la complejidad de los dispositivos de red. A diferencia de la interfaz Southbound, no existe estandarización de esta interfaz, por lo que la portabilidad de Aplicaciones SDN sobre distintos controladores no está garantizada.

#### **1.1.11. Modos de implementación**

Como parte de la evolución tecnológica en SDN, debemos precisar las modalidades de implementación y su comparación, para posteriores pruebas y análisis de resultados.

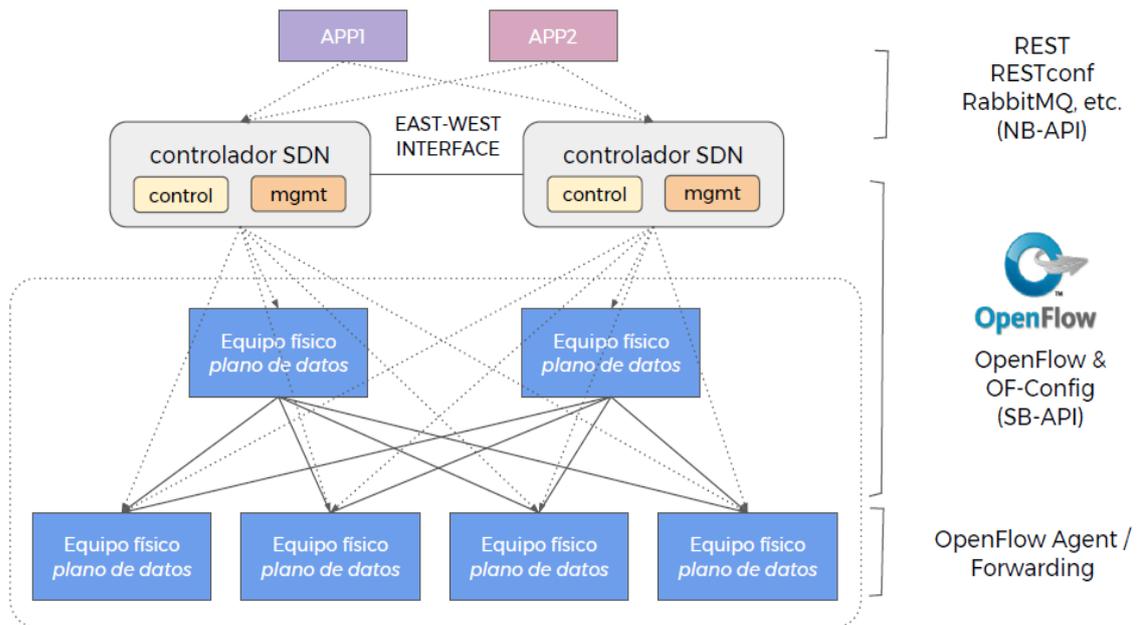


Figura 15 SDN Open Source via OpenFlow

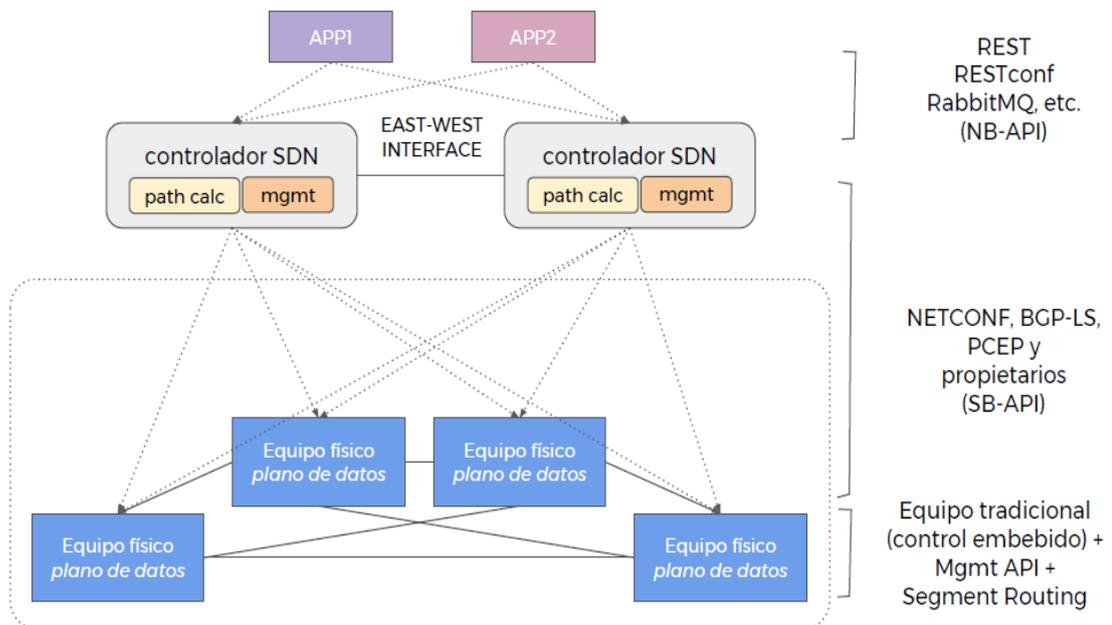


Figura 16 SDN Open Source via Overlay

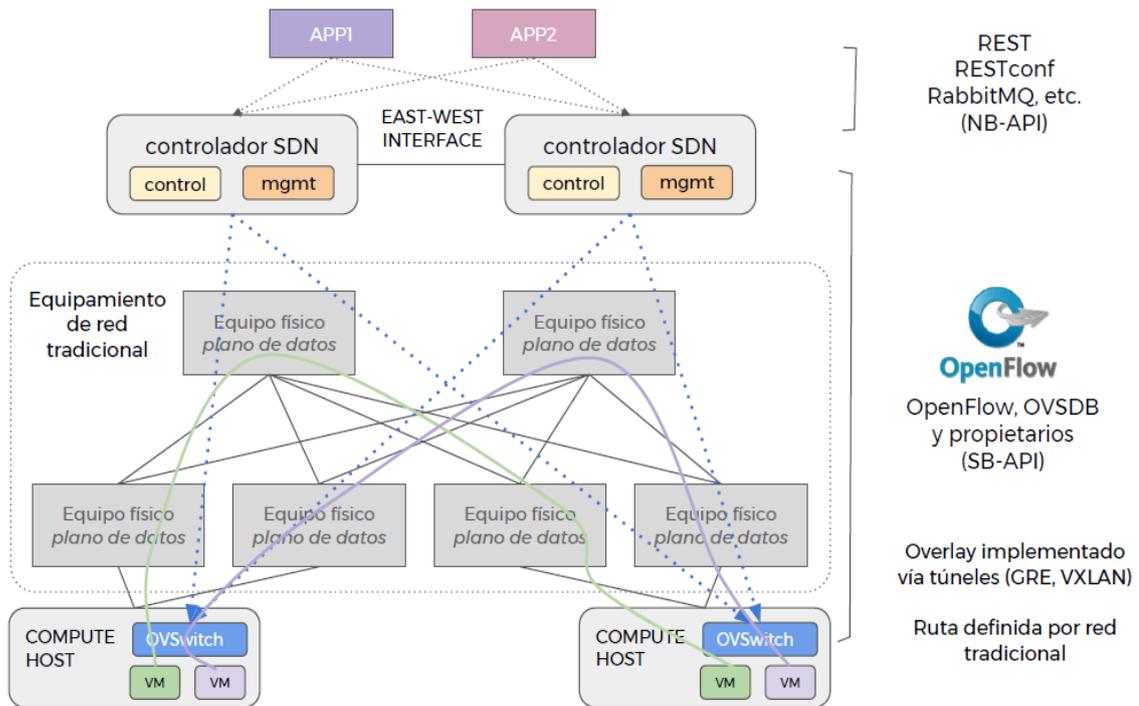


Figura 17 SDN Open Source via APPs

Modelo SDN / Característica	SDN Open Source (OpenFlow)	SDN via Overlay	SDN via APPIs
Caso de Uso principal	Varios	Centro de datos	Core/WAN
Hardware	Switches OpenFlow	Soporte VXLAN VTEP en servidores o switches	Routers tradicionales
Protocolo principal	OpenFlow	VXLAN	Segment Routing
Adopción en la industria	Baja	Media	Baja
Siguiente paso habitual en redes tradicionales	Implementar caso de uso específico	Implementar OpenStack, VMWare, NSX, Nuage VSP, Juniper Contral, entre otros.	Conseguir soporte de Segment Routing, PCC y BGP-LS en la red

Figura 18 Comparación de las alternativas de implementación

### 1.1.12. OpenFlow

Openflow es un protocolo de comunicación perteneciente a la interfaz sur - SouthBound API, se compone de una o más tablas de flujo y una grupal, cuya función es ubicar los paquetes y reenviarlas a un controlador externo. Hace uso de las tablas de flujo, o memorias TCAM y RAM, que la mayoría de fabricantes insertan en sus dispositivos de red. (Foundation, 2014)

El dispositivo OpenFlow se compone de tres partes:

Tabla de flujos: Indican como debe ser procesado un flujo determinado de paquetes.

Canal seguro: Se refiere al canal de comunicación entre el switch y el controlador que utiliza OpenFlow, debe garantizar la seguridad debido a que gestiona toda la red de forma centralizada.

Protocolo OpenFlow: Provee la opción de comunicación entre el conmutador y el controlador, soporta las instrucciones de insertar, eliminar, modificar y buscar en las entradas de la tabla de flujos a través del canal seguro.

#### 1.1.1.3. Arquitectura de SDN con OpenFlow

Los componentes de una arquitectura SDN/OpenFlow son:

El Switch OpenFlow

El controlador

El protocolo OpenFlow

El canal switch-controlador.

La ONF es la entidad que emite las especificaciones OpenFlow, las cuales se centran en el Switch OpenFlow, definiendo sus características, así como el protocolo de comunicación con el Controlador.

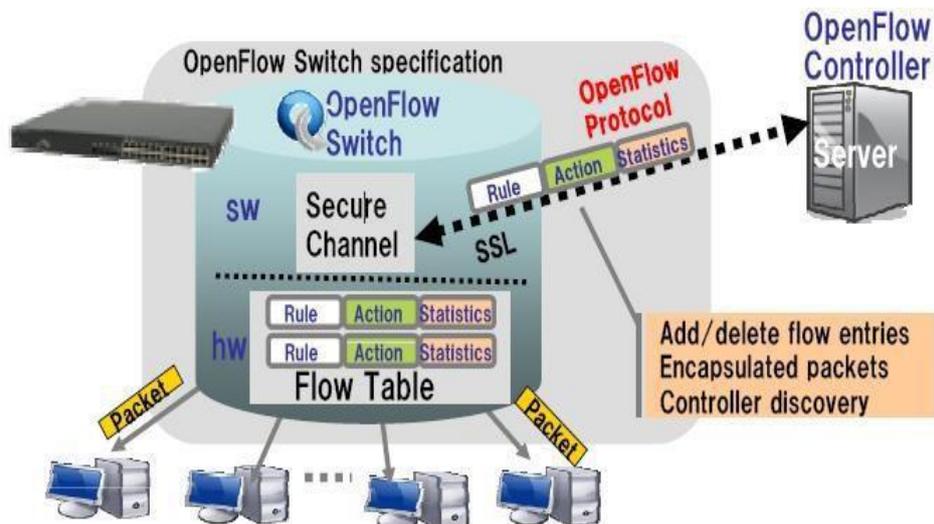


Figura 19 Componentes de SDN/OpenFlow

En noviembre de 2007 el OpenFlow Consortium, un grupo colaborativo de investigadores universitarios y administradores de red lanzó la primera versión 0.1.0. Se continuó con versiones aún experimentales: 0.2.0, 0.8.0, 0.8.1, 0.8.2, 0.8.9, publicadas en 2008 y 0.9 a mediados de 2009.

### 1.1.13. Mininet

Es un emulador de red que trae consigo dispositivos, que se configuran como hosts finales, conmutadores de red, enlaces y controladores, usado para despliegues de redes SDN y el protocolo OpenFlow, de fácil interacción y verificación de la red mediante línea de comandos (CLI), se soporta bajo licencia BSD de código abierto. Su sencillez lo demuestra por ejemplo al facilitar que con un solo comando *sudo mn* (utilizado por ejemplo en Ubuntu) se puede obtener información del estado de la red y crea una topología básica. También permite crear cualquier topología de red personalizada. Básicamente realiza la emulación de envío de paquetes, captura datos, tiempo de respuesta y retardo, ping o tcpdump para comprobar si hay conexión satisfactoria entre los hosts.

### 1.1.14. Cbench

Cbench, es una de las herramientas de benchmarker utilizada para probar controladores OpenFlow generando eventos de paquetes para nuevos flujos. Cbench realiza la emulación de un grupo de interruptores que se conectan a un

controlador, envían mensajes de paquetes y observan cómo se modifican los modos de flujo. Si cree que un cambio que está haciendo puede tener un impacto en el rendimiento, puede ser útil usar cbench para medirlo.

#### **1.1.15. Controlador SDN**

En la arquitectura de redes SDN, los controladores adoptan la inteligencia mediante software, para obtener la administración global de la red, la componen tres capas: Infraestructura, Control y Aplicaciones. El enfoque consta en otorgar a los administradores de red la optimización en el control sobre los dispositivos de red, separando el plano de datos, del plano de control que establece los parámetros de prioridad, direccionamiento, calidad de servicio. En síntesis, centralizar los componentes de red mediante el controlador (Alejandro García Centeno, 2014)

El controlador SDN se comunica con los switches Openflow para informar por dónde deben enrutar el tráfico. Lo logra mediante la denominada “Southbound Interface”, que es la capa que gestiona la comunicación entre los switches OpenFlow y el controlador. Por otro lado, la denominada Northbound Interface es la interfaz responsable de comunicar el controlador con aplicaciones externas, mediante la API. Estas ofrecen la capacidad de programar el controlador exponiendo modelos de datos de red y otras funcionalidades para ser utilizados por las aplicaciones externas.

En el intermedio de la Northbound Interface y la Southbound Interface se encuentra la lógica que permite la interacción entre las órdenes abstractas recibidas por la Northbound desde aplicaciones externas y las órdenes concretas que debe darle el controlador a los switches por medio de la Southbound. A esta capa se le suele denominar capa de abstracción de servicio (del inglés, service abstraction layer, SAL).

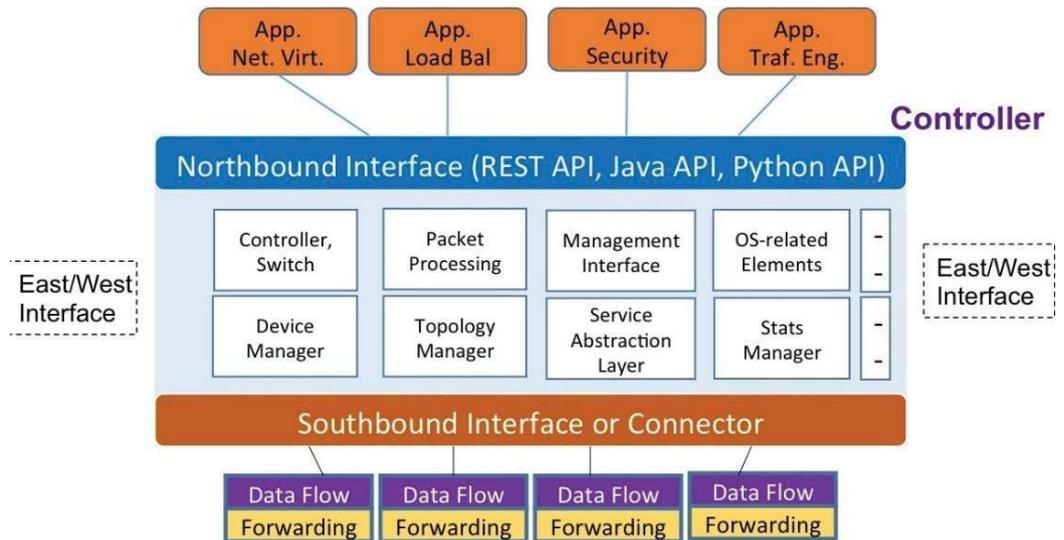


Figura 20 Arquitectura habitual de un controlador SDN.

En la actualidad para muchos administradores de red, las bondades de la arquitectura resultan inverosímil, debido a que obtener una red gestionada de forma centralizada, que simplifique las funciones y servicios, sin que afecte la comunicación, significaría romper la actual forma de ver la administración de las redes. Existe una gran variedad de software de controladores tanto de código abierto como de propietario que difieren de acuerdo a las estrategias de diseño que cada una utiliza, entre los que podemos mencionar los siguientes:

OpenDaylight, ONOS, Project Calico, The Fast Data Project, Project Floodlight, Beacon, NOX/POX, Open vSwitch vneio/sdnc, Ryu Controller, Cherry, Faucet, OpenContrail, Nuage Virtualized Services Controller (VSC) de Nokia, VortiQa Open Network Director de Freescale Semiconductor

Tabla 1 Principales controladores SDN de código abierto.

Controlador	Fuente	Licencia	Lenguaje
<b>OpenDayLight</b>	OpenDayLight	EPL	<b>Java</b>
<b>ONOS</b>	ON.Lab	Apache 2.0	<b>Java</b>
<b>NOX</b>	ICSI	GPL	<b>C++</b>
<b>POX</b>	ICSI	GPL	<b>Python</b>
<b>Beacon</b>	Stanford University	GPLv2, Stanford FOSS	<b>Java</b>
<b>Floodlight</b>	Big Switch Networks	Apache 2.0	<b>Java</b>

<b>Controlador</b>	<b>Fuente</b>	<b>Licencia</b>	<b>Lenguaje</b>
<b>Ryu</b>	NTT Communications	Apache 2.0	<b>Python</b>
<b>FLOWER</b>	Traveling GmbH	MIT License	<b>Erlang</b>
<b>Jaxon</b>	University of Tsukuba	GPLv3	<b>Java</b>
<b>Mul SDN</b>	Kulcloud	GPLv2	<b>C</b>
<b>NodeFlow</b>	Gary Berger	-	<b>Javascript</b>
<b>Trema</b>	NEC	<b>GPLv2</b>	<b>Ruby.C</b>

#### 1.1.1.4. OpenDayLight

OpenDaylight (ODL) es una plataforma de código abierto modular que permite personalizar y automatizar redes.

Desde el año 2014 cuenta con 10 lanzamientos, más de 1000 autores remitentes, más de 100K confirmaciones y potencia las redes de suscriptores globales 1B +. Su código OpenDaylight se ha integrado o integrado en más de 35 soluciones y aplicaciones de proveedores. También soporta ONAP, OpenStack y OPNFV. Utiliza las siguientes herramientas para su funcionamiento.

Maven: Usa pom.xml (Modelo de objetos de proyecto) para crear un script de las dependencias entre paquetes y describir qué paquetes cargar y comenzar.

OSGi: Es el back-end de OpenDaylight, ya que permite cargar dinámicamente paquetes y paquetes de archivos JAR, y enlazar paquetes para intercambiar información.

Interfaces JAVA: Las interfaces Java son utilizadas para escuchar eventos, especificaciones y patrones de formación.

API REST: Es el administrador de topología, el rastreador de host, el programador de flujo, el enrutamiento estático, etc.

### 1.1.1.4.1. Arquitectura

La arquitectura de OpenDaylight SDN Controller tiene varias capas. La capa superior la conforman las aplicaciones de gestión y lógicas de la red. La capa intermedia se refiere a la infraestructura y la capa inferior consiste en dispositivos físicos y virtuales (switch).

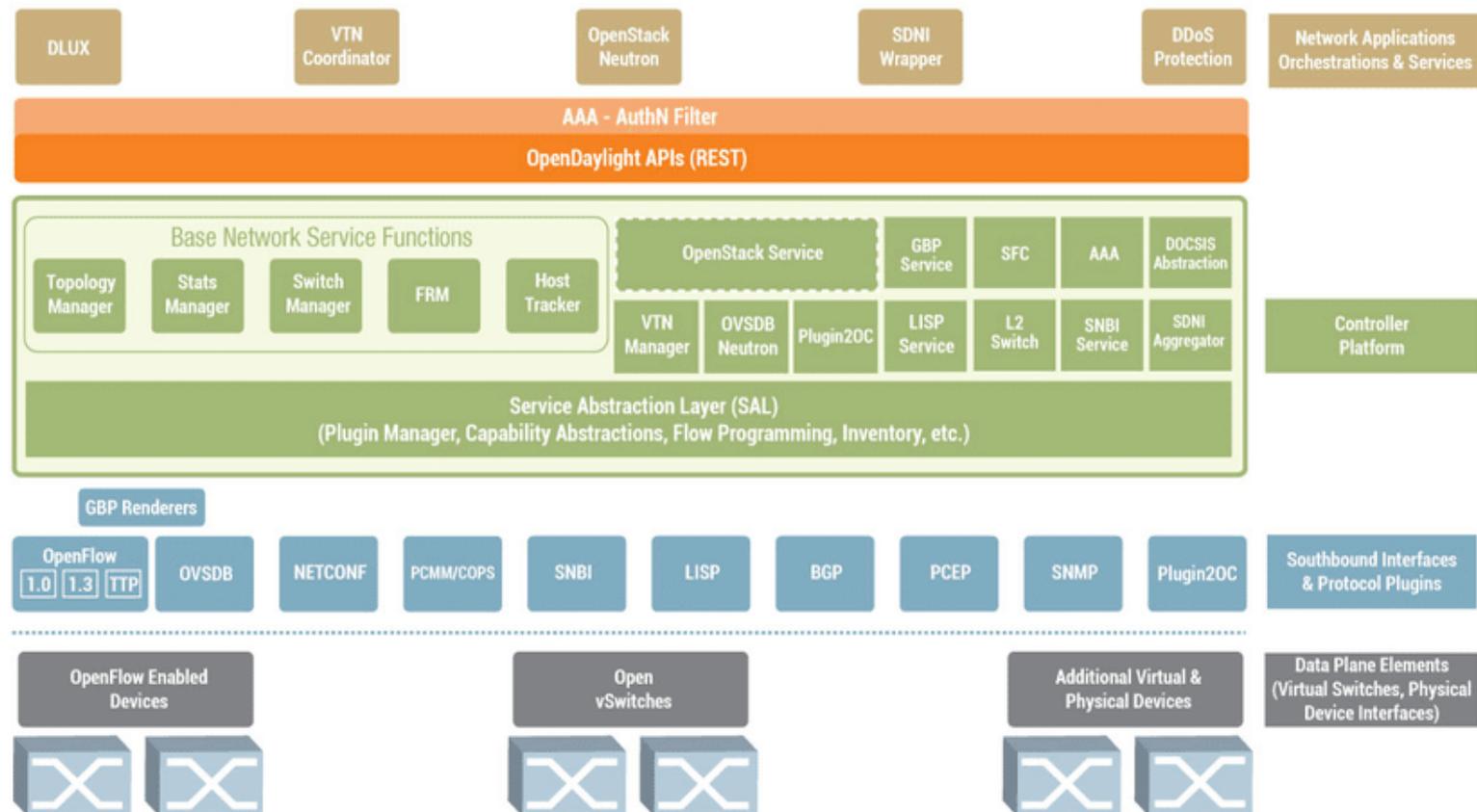


Figura 21 Arquitectura del controlador OpenDaylight (C.Fancy, 2018)

Según Opendaylight (2017), la capa intermedia aloja las north-bound y south-bound API. Opendaylight admite el marco REST bidireccional para la north-bound API. Las aplicaciones que se encuentran por encima de la capa intermedia, utilizan el controlador para recopilar la información de la red, ejecutar algoritmos para realizar análisis y usar esta información para realizar cambios en la red a través del controlador. La interfaz south-bound API puede soportar múltiples protocolos como Openflow y BGP-LS.

**Controlador de Servicios y Aplicaciones:** Esta capa se encuentra dividida en dos módulos, una plataforma de servicios y los servicios y aplicaciones de red. El módulo plataforma de servicios proporciona servicios específicos de autenticación, autorización y contabilización que permite al usuario autenticarse por medio de credenciales.

**Capa de Abstracción de Servicios:** El núcleo del controlador OpenDaylight es la capa de abstracción de servicios dirigida por modelos, conocida como MD-SAL por sus siglas en inglés (Model Driven -Service Abstraction Layer). Es quien permite a la interfaz sur soportar múltiples protocolos y donde interactúan los objetos y modelos que representan a los elementos de red y aplicaciones. La capa de abstracción de servicios oculta la complejidad de las API utilizando modelos YANG y permite interacciones por roles productor/consumidor; los productores son quienes implementan una API o los datos de una API y los consumidores son quienes utilizan esos datos.

**Interfaz Norte:** Expone protocolos como NETCONF, REST y RESTCONF.

**Interfaz Sur:** Se comunica con los elementos de red y abarca múltiples protocolos y plugin, como lo son OpenFlow, NETCONF, BGP, SNMP, LISP, etc.

#### 1.1.1.4.2. Módulos en OpenDayLight

Odl-mdsal-clustering: Este módulo es el encargado de permitir el funcionamiento del clúster de controladores.

Odl-l2switch-switch-ui: Este es el módulo que permite la conectividad en la red, permite aprender direcciones MAC e IP, también elimina bucles por lo que no es necesario instalar módulos adicionales de STP y permite la instalación de flujos en cada conmutador basados en parámetros de tráfico en la red.

Odl-dlux (core, apps-nodes, apps-topology): Estos módulos permiten utilizar la interfaz gráfica de ODL. Con las aplicaciones de nodos “odl-dluxapps-nodes” se podrán observar un listado de los dispositivos (conmutadores) detectados, así como sus flujos y conexiones. En la parte de topología se mostrará una representación gráfica de la red.

Odl-jolokia: Este módulo permite el acceso tipo REST a JMX con JSON a través de HTTP. Este módulo nos permite realizar consultas para verificar si el clúster se ha formado correctamente y que rol tienen los controladores.

#### 1.1.1.5. ONOS

Open Network Operating System (ONOS ®) es el controlador SDN de código abierto líder para crear soluciones SDN / NFV de próxima generación., admite la configuración y el control en tiempo real de la red, eliminando la necesidad de ejecutar protocolos de enrutamiento y control de conmutación dentro de la estructura de la red. permitiendo a los usuarios finales crear fácilmente nuevas aplicaciones de red sin la necesidad de alterar los sistemas de plano de datos. (Mamadou T.BAH, 2019) El controlador SDN de ONOS, se basa en una plataforma central con un conjunto de aplicaciones que interactúan de forma distribuida, modular y extensible. Se menciona que ofrece una gestión simplificada, para la configuración e implementación de nuevo software, hardware y

servicios. Ofrece una arquitectura con escalabilidad, resistencia y necesarias para cumplir con los rigores de los servicios de red.

#### 1.1.1.5.1. Arquitectura

La arista principal en la arquitectura ONOS son los servicios, que es formado por subsistemas que pertenecen a los niveles: Aplicación (App), Gestor (Manager) o Proveedor (Provider).

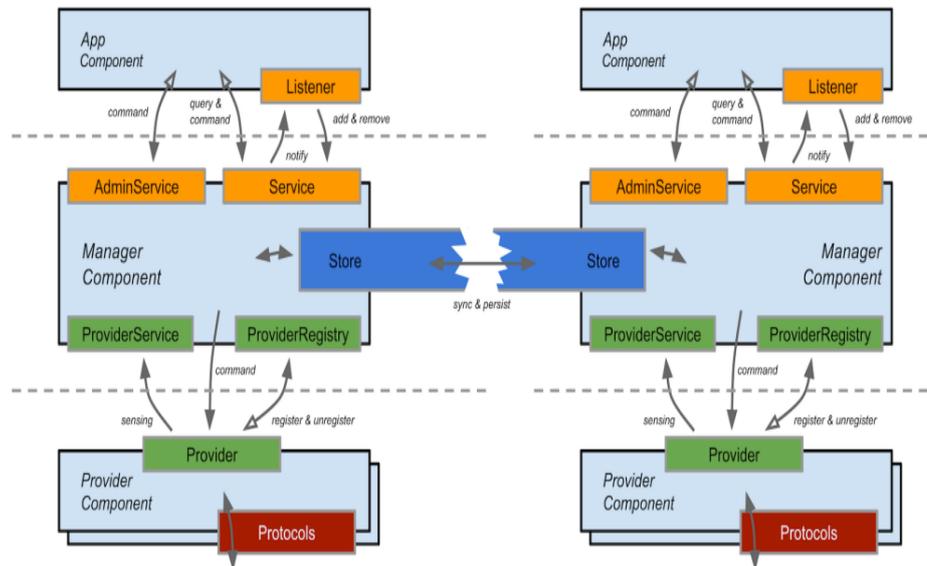


Figura 22 Arquitectura del controlador ONOS.

#### 1.1.1.5.2. Módulos en ONOS

*Org.onosproject.mlb:* Esta aplicación permite que los controladores que forman un clúster realicen automáticamente un balanceo de la cantidad de conmutadores conectados.

*Org.onosproject.fwd:* Esta aplicación permite el reenvío de paquetes tanto en capa 2 como en capa 3, también maneja direcciones IPv6. No es necesario instalar ningún módulo STP para eliminar bucles en la red.

*Org.onosproject.openflow:* Permite la comunicación con dispositivos que manejan el protocolo openflow.

### 1.1.16. Beneficios de las SDN

Al desvincular el plano de control del plano de datos para obtener un sistema centralizado las redes definidas por software SDN, proporcionan múltiples beneficios: Reduce costos en hardware y operativamente, ofrece un marco entendible y flexible, para reducir las brechas de inestabilidad que pueden ocasionar los errores operacionales, permite orquestar, gestionar la implementación, configuración o actualización de dispositivos, por ende, genera confiabilidad, seguridad, escalabilidad y alta disponibilidad.

## II. MATERIAL Y MÉTODOS

### 2.1. Método de Investigación

El procedimiento de la revisión bibliográfica en el proyecto se enfoca en proporcionar un marco de trabajo íntegro y reproducible para los administradores de red, que facilite la toma de decisiones en la gestión de una red de computadoras, basado en la arquitectura de las Redes Definidas por Software. Sin embargo, para aportar una base sólida y confiable se requiere investigar un rango de 5 años, entre el 2015 al 2020. A continuación, se realiza la descripción del método mencionado:

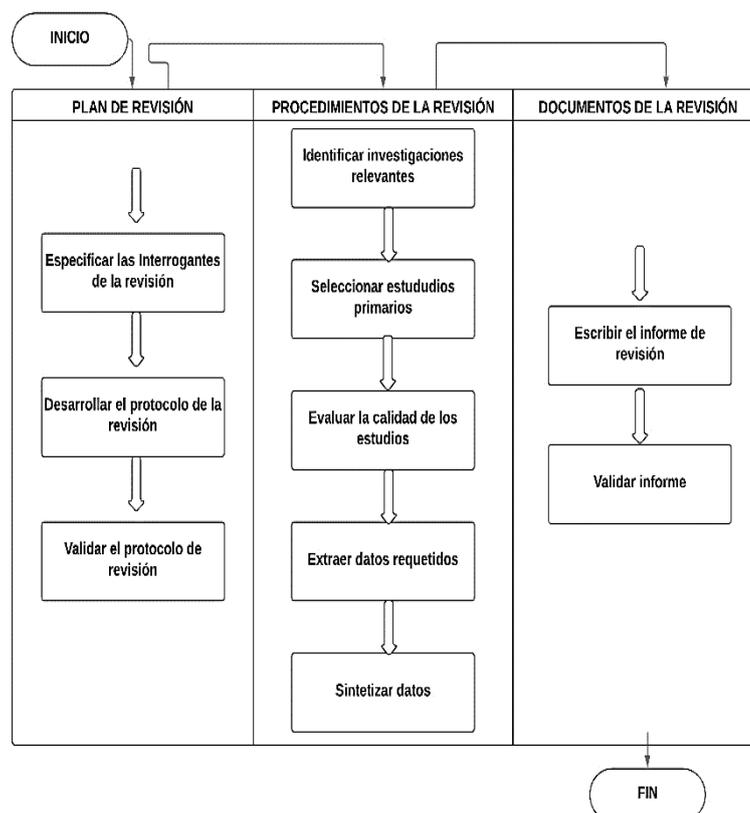


Figura 23 Método de revisión bibliográfica

## **2.2. Plan de Investigación**

Especifica los pasos a seguir en la revisión bibliográfica, para garantizar la integridad de la información, entre ellos:

Interrogantes de investigación, que el análisis del proyecto atenderá.

El procedimiento para la búsqueda de la información.

Elementos para la selección de artículos de investigación que se incluyen en la búsqueda.

### **2.2.1. Interrogantes de la Investigación**

Las incógnitas planteadas que el estudio responderá son:

PREG 1. ¿Cuántos estudios bibliográficos concernientes al paradigma de Redes Definidas por Software está disponible durante los últimos cinco años?

PREG 2. ¿Qué autores y metodología se han publicado durante los últimos cinco años?

PREG 3. ¿Qué métricas y características para la evaluación de los controladores, destacan en los estudios de Redes Definidas por Software?

PREG 4. ¿Qué problemas enfrentaron los estudios, como lo resolvieron y qué conclusiones nos guían en las siguientes investigaciones?

PREG 5. ¿Cuáles son los beneficios de adoptar las Redes Definidas por Software?

### **2.2.2. Protocolo de revisión**

El protocolo define el criterio para la búsqueda y posterior clasificación de los artículos científicos relacionados al tema, para ello nos sustentamos en la cantidad de citas de los artículos.

### 2.2.3. Validación del Protocolo de revisión

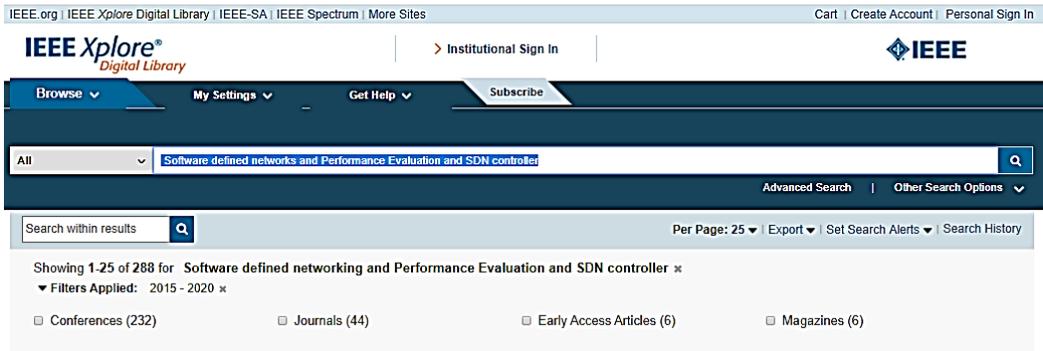
Se ha validado el protocolo de revisión. Las búsquedas se enfocan en revisiones sobre el protocolo y dar respuestas a las preguntas de investigación.

## 2.3. Procedimiento de la Investigación

### 2.2.4. Identificar las investigaciones relevantes

En este punto, se realiza la minuciosa búsqueda de investigaciones, con el fin de obtener información relevante y precisa de acuerdo a las preguntas planteadas. Por otro lado, la propuesta generó una cadena de búsqueda utilizando el conector booleano “and”; “*Software defined networks and Performance Evaluation and SDN controller*” en el rango del año 2015 al 2020.

Por ejemplo, se ha revisado la base de datos IEEE Xplore para ubicar fuentes de información, dando como resultado:



The screenshot shows the IEEE Xplore Digital Library search interface. At the top, there are navigation links for IEEE.org, IEEE Xplore Digital Library, IEEE-SA, IEEE Spectrum, and More Sites. The main header includes the IEEE Xplore logo and an Institutional Sign In button. Below the header, there are navigation tabs: Browse, My Settings, Get Help, and Subscribe. The search bar contains the query "Software defined networks and Performance Evaluation and SDN controller" and a search icon. Below the search bar, there are options for "Advanced Search" and "Other Search Options". The results section shows "Showing 1-25 of 288 for Software defined networking and Performance Evaluation and SDN controller". Filters Applied: 2015 - 2020. The results are categorized as: Conferences (232), Journals (44), Early Access Articles (6), and Magazines (6).

Figura 24 Búsqueda inicial en la base de datos IEE



### 2.2.5. Seleccionar estudios primarios

La selección de las investigaciones expresadas en el proyecto, se lleva a cabo gracias a los distintos indicadores, tales como, palabras clave, revistas, resumen, cantidad de referencias, basados en la regla de búsqueda planteada. A continuación, se describe las tendencias según el año de publicación y número de citas.

The image shows a vertical stack of four search result panels. Each panel has a search bar at the top with the text 'Search within results' and a magnifying glass icon. To the right of the search bar are links for 'Per Page: 25', 'Export', 'Set Search Alerts', and 'Search History'. Below the search bar, the text reads 'Showing 1-25 of [total] for (((All Metadata: Software defined networks) AND All Metadata: Performance Evaluation) AND All Metadata: SDN controller)'. Below this is a filter section 'Filters Applied: [year]'. The panels represent the years 2015, 2020, 2018, and 2020. The 2018 panel includes a filter section with checkboxes for 'Conferences (61)', 'Journals (14)', 'Magazines (3)', and 'Early Access Articles (1)'. The 2020 panel includes a filter section with checkboxes for 'Journals (4)', 'Early Access Articles (2)', and 'Conferences (1)'.

Figura 27 Búsqueda de cadena en el año 2015 al 2020

La base de datos ofreció 288 resultados con la coincidencia. No obstante, la abundante y mezclada información sobre Redes Definidas por Software, nos obliga a ser más específicos, escoger un tema y seleccionar los posibles artículos que respaldaran nuestro trabajo y objetivos. Nos interesa conocer cuantos trabajos de investigación existen, la problemática, la solución, los resultados, conclusiones y aporte brindado sobre el tema de investigación.

A continuación, se describe las tendencias de los estudios filtrados haciendo énfasis en el año de la publicación, descripción del artículo, el resumen que incluye, la cantidad de visitas y descargas de material completo.

Tabla 2 Resultados más significativos sobre el rendimiento de controladores SDN.

N°	Año	Artículo	Resumen	# de vistas
1	2019	Evaluación del tráfico TCP y UDP sobre redes definidas por software	En esta investigación se analizó el flujo de tráfico SDN, en los protocolos de red, TCP y UDP. Primero se emuló una infraestructura de red con la ayuda de Mininet, luego con la herramienta de medición iPerf se generó tráfico de prueba, que fue analizado en materia de paquetes por WireShark, en el modo de filtrado TCP y UDP.	41
2	2019	Una evaluación experimental de la latencia de configuración de flujo en redes distribuidas definidas por software	El estudio sostiene el riesgo de la implementación SDN en el modo único controlador, debido a que puede convertirse en un cuello de botella, que afecte el rendimiento de respuesta de paquetes y latencia en el flujo. A su vez, se presenta incidencias de escalabilidad.	1
3	2019	Hacia un control de flujo adaptativo y escalable OpenFlow-SDN: una encuesta	En esta encuesta, hemos presentado cuatro desafíos principales en OpenFlow-SDN que pueden considerarse como fuentes primarias de dicha sobrecarga de tráfico. Primero debemos captar conocimiento del funcionamiento del control de flujo de OpenFlow-SDN y los puntos críticos que afectan el rendimiento y la escalabilidad de la red, como la visibilidad centralizada lógicamente, el descubrimiento del estado del enlace, las reglas de flujo, problema de colocación y balanceo de carga de controladores.	725
4	2019	Criterios y metodologías de validación genérica para aplicaciones SDN	Los controladores de red ofrecen la configuración centralizada de la red, a su vez presentan el riesgo de tener errores de software y errores en las aplicaciones SDN, que amenazen las operaciones normales de las redes SDN. Para ello se debe validar y crear un banco de pruebas, con posterior análisis de los resultados automático.	53
5	2019	Una solución SDN para mejorar el rendimiento en redes dedicadas de área amplia	Con el fin de proporcionar flexibilidad y control en caso se presenten fallas en el funcionamiento de la red, se propone adoptar una superposición de red SDN distribuida y dispositivos de borde SDN.	144

N°	Año	Artículo	Resumen	# de vistas
6	2019	Una encuesta sobre flexibilidad del plano de datos y programabilidad en redes definidas por software	Como entendemos, una de las funciones del conmutador es el reenvío de paquetes obedeciendo las instrucciones indicadas previamente por el controlador SDN. Este documento ofrece una descripción general de la encuesta del plano de datos con un énfasis particular en el problema de la programabilidad y la flexibilidad.	780
7	2019	Evaluación de rendimiento de descubrimiento de topología de controladores OpenDaylight y ONOS	Cada año se desarrolla una variedad de controladores SDN de código abierto y comercial, que ofrecen a los operadores de red funciones avanzadas para lograr altos rendimientos y disponibilidad de la red. En este estudio se evalúa de forma experimental el rendimiento de los controladores ONOS y OpenDaylight en términos de descubrimiento de topología y actualización de topología	83
8	2019	Comparación de rendimiento de simuladores de red definidos por software para redes tácticas: Mininet vs. OPNET	Para toda nueva y emergente arquitectura de red, se debe generar banco de pruebas, análisis y comparación de resultados. Este estudio detecta que ninguna investigación muestra el uso de OPNET para SDN, por eso realiza una rigurosa comparación en base a criterios de demora, fluctuación de fase y RTT de extremo a extremo entre controladores e interruptores.	228
9	2018	Evaluación comparativa del controlador ONOS con OFCProbe	Para cuantificar varios indicadores de rendimiento del controlador SDN, se debe iniciar por evaluar su viabilidad en el contexto de diferentes características de la red, entre ellas: la topología y el tamaño de la red, el número de nodos y enlaces, se realiza usando la herramienta OFCProbe.	57
10	2018	Evaluación empírica de controladores SDN usando Mininet / Wireshark y comparación con Cbench	En este documento, se busca generar conocimiento sobre el rendimiento de los controladores SDN: ONOS, OpenMUL y POX, implementados en diferentes lenguajes de programación Java, C y Python, respectivamente, utilizando análisis de paquetes Mininet y Wireshark.	417
11	2018	Una evaluación comparativa del rendimiento de los controladores SDN populares	La rápida introducción de nuevos controladores en el mercado hace que sea difícil elegir un controlador adecuado. Este estudio investiga y evalúa el rendimiento de los controladores ONOS, Ryu, Floodlight y OpenDayLight en términos de latencia y rendimiento utilizando la herramienta Cbench	732

N°	Año	Artículo	Resumen	# de vistas
12	2018	Evaluación del rendimiento de los controladores SDN POX y floodlight en un entorno de emulación de mininet	El objetivo de este documento es explorar las características de los controladores POX y Floodlight y comparar su rendimiento	385
13	2018	Evaluación de los parámetros de QoS en diferentes arquitecturas SDN usando Omnet 4.6 ++	Este trabajo presenta la evaluación del rendimiento de los parámetros de QoS, MOS bajo Omnet 4.6 ++ para todos los tipos de topologías de redes definidas por software (SDN) .	171
14	2018	Análisis de rendimiento de la arquitectura del controlador de red definida por software: una encuesta basada en simulación	La plataforma SDN presenta nuevos desafíos para la escalabilidad y el rendimiento, el fortalecimiento de la seguridad, los protocolos de comunicación entre capas, etc. Este documento informa sobre una serie de tecnologías, modelos y herramientas para evaluar las métricas de rendimiento de los controladores SDN junto con los resultados de la simulación.	463
15	2017	Evaluación experimental de dos controladores OpenFlow.	Este trabajo realiza una evaluación experimental de dos controladores OpenFlow distribuidos de código abierto, a saber, ONOS y OpenDaylight. Específicamente, construye un banco de pruebas con Cbench para evaluar su rendimiento.	346
16	2017	Una evaluación cualitativa y cuantitativa de los controladores SDN	El crecimiento de la integración SDN en nuevos mecanismos y arquitecturas de red condujo al desarrollo de diferentes soluciones de controlador, con una amplia variedad de características. Ante aquello se proporciona un estudio de los controladores OpenDaylight (ODL), Open Network Operative System (ONOS), Ryu y POX), evaluando no solo su rendimiento, sino también sus características en un forma cualitativa	260

N°	Año	Artículo	Resumen	# de vistas
17	2017	Análisis comparativo de controladores de redes definidas por software (SDN): en términos de capacidades de manejo de tráfico	Este documento presenta un análisis de las capacidades de dos controladores OpenFlow SDN basados en Python de código abierto (POX y RYU). Los índices analizados incluyen el rendimiento de ambos controladores cuando están sujetos a la conmutación de capa 1 y capa 2.	548
18	2016	Evaluación del rendimiento y escalabilidad de Mininet en escenarios con grandes centros de datos.	Mininet presenta limitaciones debido a las restricciones de la plataforma de hardware / software donde se ejecuta, lo que le impide escalar a redes grandes. Presentamos una caracterización del rendimiento y la escalabilidad de Mininet a través de un análisis experimental para un centro de datos.	686
19	2015	Sobre el rendimiento de los controladores SDN: una comprobación de la realidad	Este documento tiene como objetivo hacer una verificación de la realidad del rendimiento actual de los controladores. Las mediciones se llevan a cabo en un entorno controlado, donde cada controlador se prueba con su propia configuración optimizada, lo que permite medir el rendimiento máximo, y obtener resultados intrínsecos.	1248

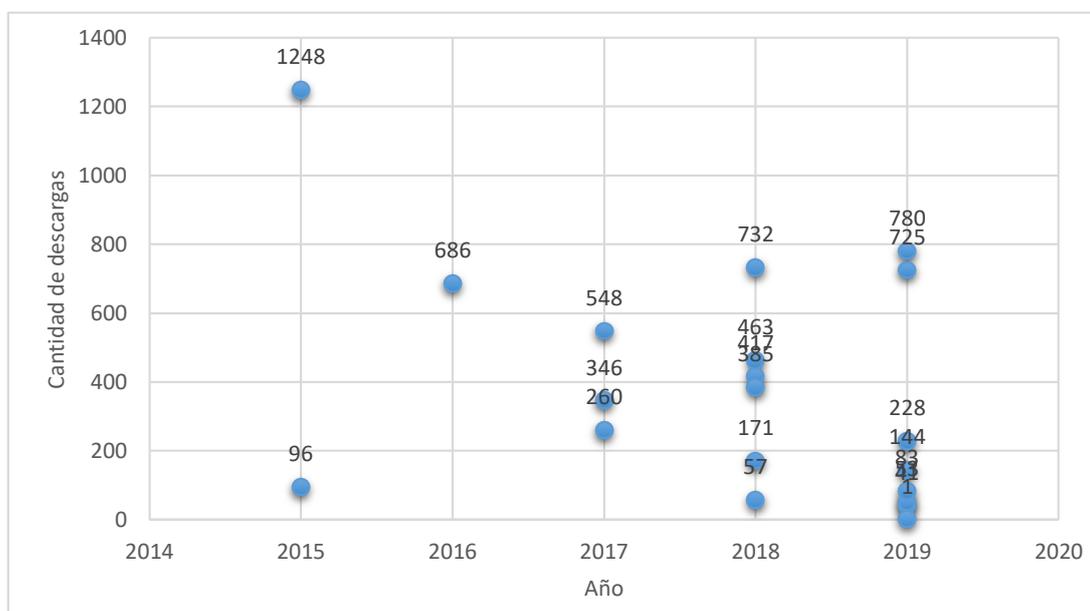


Figura 28 Resultados filtrados en los últimos 5 años

### 2.2.6. Evaluar la calidad de los estudios

Se suma al filtrado según la cadena de búsqueda y verificación individual, el número de citas que han referenciado, esta cuantificación nos indica el impacto y soporte que brindan a la comunidad de investigación. A continuación, se detalla lo sustentado:

Tabla 3 *Resumen de impacto de los artículos (2015-2020).*

N°	Año	Título	Origen (País)	Publicado en:	# de citas	# de vista
1	2019	Evaluación del rendimiento de descubrimiento de topología de los controladores OpenDaylight y ONOS	Paris, France	2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)	0	83
2	2019	Estudo Comparativo de Controladores Software-Defined Networking (SDN) - Controladores de Tráfico	Braga, Portugal	2019 14th Iberian Conference on Information Systems and Technologies (CISTI)	0	39
3	2018	Análisis de rendimiento de la arquitectura de controlador de red definida por software una encuesta basada en simulación	Bhubaneswar, India	2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)	1	463
4	2018	Comparación de tolerancia a fallos de los controladores SDN de ONOS y OpenDaylight	Ch <sup>^</sup> atillon, France	2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)	0	214
5	2018	Evaluación de los parámetros de QoS en diferentes arquitecturas SDN usando Omnet 4.6	Monastir, Tunisia Marruecos	2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)	0	171

<b>N°</b>	<b>Año</b>	<b>Título</b>	<b>Origen (País)</b>	<b>Publicado en:</b>	<b># de citas</b>	<b># de vista</b>
6	2018	Hacia redes SDN habilitadas para QoS	Marrakech, Morocco	2018 International Conference on Advanced Communication Technologies and Networking (CommNet)	1	162
7	2018	Una evaluación comparativa del rendimiento de los controladores SDN populares	Dubai, Emiratos Árabes Unidos	2018 Wireless Days (WD)	6	732
8	2018	Análisis De Rendimiento De Controladores De Flujo Abierto Basados En Python	Tadepalligudem, India	3rd International Conference on Electrical, Electronics, Engineering Trends, Communication, Optimization and Sciences (EEECOS 2016)	1	158
9	2017	Evaluación experimental de dos controladores OpenFlow	Toronto, ON, Canadá	2017 IEEE 25th International Conference on Network Protocols (ICNP)	3	346
10	2017	Implicación del rendimiento y análisis del protocolo OpenFlow SDN	Regina, Canada	2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)	5	472
11	2017	Redes definidas por software análisis comparativo de topologías con ONOS	Chennai, India	2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)	0	280
12	2017	Una evaluación cualitativa y cuantitativa de los controladores SDN	Almada, Portugal	2017 International Young Engineers Forum (YEF-ECE)	3	260
13	2016	Controladores SDN: Un estudio comparativo	Lemesos, Cyprus	2016 18th Mediterranean Electrotechnical Conference (MELECON)	23	1966

N°	Año	Título	Origen (País)	Publicado en:	# de citas	# de vista
14	2016	Investigación y análisis comparativo de la prueba de rendimiento en el controlador SDN	Wuhan, China	2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)	7	619
15	2016	Una medida de los tiempos de respuesta de varios controladores OpenFlow SDN con CBench	Marrakech, Marruecos	2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)	4	411
16	2015	Evaluación del rendimiento del controlador SDN de OpenDaylight	Hsinchu, Taiwan Pakistan	2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)	29	1858
17	2015	Sobre redes definidas por software y el diseño de Controladores SDN	Sydney, Australia	2015 6th International Conference on the Network of the Future (NOF)	7	692

### 2.2.7. Extraer datos requeridos

Para responder a las preguntas de investigación, se extrajeron los siguientes datos de cada estudio:

1. *Las palabras clave* que se utilizarán en un análisis de contenido para determinar el alcance de la investigación.
2. *Tipo de datos recogidos*: cualitativos, cuantitativo, encuesta, revisión de la literatura
3. *Información sintetizada* para determinar las problemáticas que enfrentaron y métodos para la solución.

### **2.2.8. Sintetizar datos**

Los resultados se sintetizan para poder determinar los temas principales de la investigación, clasificar cada artículo. Se realizó un análisis del contenido de las palabras clave de cada artículo haciendo énfasis en el significado de cada palabra o concepto el cual va más allá de simplemente contar palabras para examinarlas.

Se debe aclarar que para realizar este análisis se basa en la valoración de los términos controlados por INSPEC la cual es una importante base de datos de indexación de literatura científica y técnica, publicada por el Instituto de Ingeniería y Tecnología. Como resultado de la evaluación se expone el título de la publicación, autor, año de la publicación, palabras clave del autor, términos controlados por IEEE, términos controlados y no controlados por INSPEC número de citas y referencias de cada publicación. Además, se muestra una comparación entre el número de citas y la cantidad de veces que ha sido referenciado cada artículo revisad

Tabla 4 Valoración De Las Palabras Clave De Acuerdo A Inspec

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
1	2019	Evaluación del rendimiento de descubrimiento de topología de los controladores OpenDaylight y ONOS	Mamadou T.BAH ; A. Azzouni ; M.T. Nguyen ; Guy Pujolle	-	-	gestión de redes informáticas, evaluación del rendimiento de redes informáticas, redes definidas por software, topología de redes de telecomunicaciones	tiempo de descubrimiento de topología, tiempo de inactividad de la red, evaluación de rendimiento, paradigma de red, eficiencia de red, control de red, Controladores SDN de código abierto,, alto rendimiento de la red, gestión eficiente de la red, topología de red, controlador ONOS, controlador OpenDaylight	0	83

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
2	2019	Estudo Comparativo de Controladores Software-Defined Networking (SDN) - Controladores de Tráfico	Gonçalo Pereira ; José Silva ; Pedro Sousa	Redes de computadoras, SDN, OpenFlow	Linux, Java, Python, conmutadores, listas de materiales, Internet	interfaces de programa de aplicación, Internet, radio móvil, redes definidas por software, control de telecomunicaciones	microproyectos ilustrativos, aplicaciones de Internet, servicios distribuidos, redes definidas por software, controladores de tráfico, controlador SDN, infraestructuras de red	0	39
3	2018	Análisis de rendimiento de la arquitectura de controlador de red definida por software una encuesta basada en simulación	Madhukrishna Priyadarsini ; Padmalochan Bera ; Rohan Bampal	SDN, Rendimiento, Controlador, Parámetros, iPerf, Cbench, NOX, POX, Beacon	Interruptores, arquitectura de computadora, factores de tiempo, software, herramientas, rendimiento	seguridad de la red informática, protocolos, software de dominio público, arquitectura de software, redes definidas por software, métricas de software, evaluación del rendimiento del software	controladores SDN, análisis de rendimiento, redes definidas por software, control de redes, escalabilidad, entornos de red en vivo, métricas de rendimiento.	1	463

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
4	2018	Comparación de tolerancia a fallos de los controladores SDN de ONOS y OpenDaylight	José Manuel Sanchez Vilchez ; David Espinel Sarmiento	-	Sistemas de control, topología de red, resiliencia, topología, tolerancia a fallos, sistemas tolerantes a fallos, mecanizado virtual	tolerancia a fallas, redes definidas por software, confiabilidad de la red de telecomunicaciones, topología de la red de telecomunicaciones	plano de datos, escenarios de fallas, topologías de red, control de configuración, controladores OpenDaylight SDN, redes definidas por software, Mininet, ONOS	0	214
5	2018	Evaluación de los parámetros de QoS en diferentes arquitecturas SDN usando Omnet 4.6	Fatima Laassiri ; Mohamed Moughit ; Noureddine Idboufker	SDN, QoS, Controlador, Switch OpenFlow, ProtocolOpenFlow, Load Balancing, OmNet4.6	Switches, Protocolos, Calidad de servicio, , Topología, Redes definidas por software	calidad de servicio, protocolos de enrutamiento, redes definidas por software, tráfico de telecomunicaciones	Parámetros de QoS, características de tráfico, demora de extremo a extremo, redes definidas por software	0	171

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
6	2018	Hacia redes SDN habilitadas para QoS	Farah Chahlaoui ; Hamza Dahmouni	SDN, enrutamiento de tráfico, e2e Delay, Jitter, QoS	Calidad de servicio, retrasos, fluctuaciones, ancho de banda, sistemas de control	jitter, calidad de servicio, redes definidas por software	QoS, redes SDN, retraso de extremo a extremo, fluctuación de fase, costo de red autodefinido	1	162
7	2018	Una evaluación comparativa del rendimiento de los controladores SDN populares	Lusani Mamushiane ; Albert Lysko ; Sabelo Dlamini	Redes definidas por software (SDN), OpenFlow, controlador, Ryu, Floodlight, OpenDayLight, ONOS	Rendimiento, sistemas de control, medición, pruebas de referencia, comunicación inalámbrica, redes definidas por software, topología	evaluación del rendimiento de la red informática, toma de decisiones, redes definidas por software	evaluación comparativa, redes definidas por software, controladores de código abierto, evaluación de rendimiento, ONOS, Ryu, Floodlight, OpenDayLight, OpenFlow, Cbench.	6	732

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
8	2018	Análisis De Rendimiento De Controladores De Flujo Abierto Basados En Python	Karamjeet Kaur ; Sukhveer Kaur ; Vipin Gupta	Software Defined Networking, OpenFlow, Ryu, POX, Pyretic	-	computer centers, computer networks, Java, protocols, software defined networking, telecommunication network topology, telecommunication switching	OpenFlow version, controllers, java based controllers, performance analysis, python based openflow controllers, current traffic, workload, decentralized management, vendor specific devices, forwarding plane, control plane, plans, forwarding device, low cost merchant silicon physical device, virtual switch, switches, security devices	1	158

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
9	2017	Evaluación experimental de dos controladores OpenFlow	Mohamad Darianian ; Carey Williamson ; Israat Haque	-	Rendimiento, conjuntos de instrucciones, zócalos, servidores, sistemas de mensajes, sistemas de control, estudios de mercado	protocolos, software de dominio público, redes definidas por software, gestión de redes de telecomunicaciones, tráfico de telecomunicaciones	Redes definidas por software, SDN, funcionalidad de gestión de tráfico, controladores OpenFlow, ONOS, OpenDaylight, OpenStack análisis de rendimiento, protocolo OpenFlow SDN, red centralizada, plano de control, plano de datos, redes definidas por software, retraso de ruta de control, optimización de reenvío de flujo, controladores Openflow, controlador.	3	346
10	2017	Implicación del rendimiento y análisis del protocolo OpenFlow SDN	Aliyu Lawal Aliyu ; Peter Bull ; Ali Abdallah	SDN, OpenFlow, Retardo, Rendimiento y Escalabilidad	Retrasos, sistemas de control, software, hardware, control de procesos, protocolos, escalabilidad	evaluación del rendimiento de la red informática, protocolos, redes definidas por software, tráfico de telecomunicaciones	OpenFlow SDN, red centralizada, plano de control, plano de datos, redes definidas por software, retraso de ruta de control, optimización de reenvío de flujo, controladores Openflow, controlador.	5	472

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
11	2017	Redes definidas por software análisis comparativo de topologías con ONOS	Ashwin Rajaratnam ; Raturaj Kadikar ; Shanthi Prince ; M. Valarmathi	Redes definidas por software, ONOS, RTT, Mininet, controladores SDN, plano de control, plano de datos	Topología de red, topología, software, interfaces gráficas de usuario, protocolos, hardware, sistemas en tiempo real	centros informáticos, sistemas operativos de red, topología de red, calidad de servicio, redes definidas por software, enrutamiento de redes de telecomunicaciones, topología de redes de telecomunicaciones, tráfico de telecomunicaciones	RTT, tiempo de ida y vuelta, QoS, calidad de servicio, flujos de tráfico, controlador de sistema operativo de red abierta, topología de malla, topología de Torus, topología de red, redes definidas por software, redes de alta seguridad, aplicaciones innovadoras, centros de datos, enrutadores, conmutadores , redes de centros de datos, ONOS, topologías, análisis comparativo	0	280

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
12	2017	Una evaluación cualitativa y cuantitativa de los controladores SDN	Pedro Bispo; Daniel Corujo; Rui L. Aguiar	Redes definidas por software, OpenFlow, controlador SDN	Protocolos, sistemas de control, rendimiento, niobio, calidad de servicio, herramientas, escalabilidad	Comunicación móvil 5G, gestión de redes informáticas, gestión de movilidad (radio móvil), redes definidas por software, control de telecomunicaciones	evaluación cualitativa, evaluación cuantitativa, Controlador SDN, Redes definidas por software ,Red 5G, regla basada en el flujo, interfaz hacia el sur, Interfaz SB, función de red de alto nivel, interfaz hacia el norte, Interfaz NB, controlador de código abierto, OpenDaylight, ODL, sistema operativo de red abierta, ONOS, Herramienta Cbench.	3	260

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
13	2016	Controladores SDN: Un estudio comparativo	Ola Salman ; Imad H. Elhajj ; Ayman Kayssi ; Ali Chehab	SDN, controlador, Cbench, rendimiento	Java, topología, protocolos, interfaces gráficas de usuario, gestión de memoria, sistemas de control	-	-	23	1966
14	2016	Investigación y análisis comparativo de la prueba de rendimiento en el controlador SDN	Fan Yamei ; Liao Qing ; He Qi	SDN, Prueba, OpenDaylight, ONOS	Interruptores, topología, topología de red, rendimiento, protocolos, factores de tiempo	protocolos, redes definidas por software, gestión de redes de telecomunicaciones, topología de redes de telecomunicaciones	red definida por software, SDN, gestión de red, protocolo OpenFlow, controladores de código abierto, OpenDaylight, ONOS, instrumentos de prueba IXIA, Cbench, Mininet	7	619

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
15	2016	Una medida de los tiempos de respuesta de varios controladores OpenFlow SDN con CBench	C. Laissaoui; N. Idboufker; R. Ellassali; K. El Baamrani	-	Rendimiento, pruebas de referencia, software, protocolos, sistemas de control, arquitectura de red, seguridad	Redes definidas por software	tiempos de respuesta, CBench, controladores SDN, controladores OpenFlow, red definida por software, arquitectura de red, plano de control, plano de datos, solicitudes de ruta de datos, Floodlight, Beacon, Pox, Ryu	4	411

N°	Año	Título	Autores	Palabras clave del Autor	Términos de la IEEE	Términos controlados por INSPEC	Términos no controlados por INSPEC	# de citas en papers	# de vista de texto completo
16	2015	Evaluación del rendimiento del controlador SDN de OpenDaylight	Zuhran Khan Khattak, Muhammad Awais and Adnan Iqbal	Redes definidas por software, Benchmarking, Cbench, OpenDaylight, OpenFlow	Switches, pruebas de referencia, rendimiento, software, protocolos, Java	centros informáticos, evaluación del rendimiento de la red informática, redes definidas por software, control de telecomunicaciones, tráfico de telecomunicaciones, máquinas virtuales	evaluación de rendimiento, controlador OpenDaylight SDN, controlador de red definido por software, análisis de rendimiento, controlador de Floodlight, Cbench.	29	1858
17	2015	Sobre redes definidas por software y el diseño de Controladores SDN	Doan B. Hoang ; Minh Pham	Redes definidas por software, controlador SDN, interfaz norte, interfaz sur, interfaz este / oeste	Switches, Java, Protocolos, Topología de red, Topología, Arquitectura de computadora	toma de decisiones, interfaces de red, redes definidas por software	redes definidas por software, diseño de controlador SDN, plano de reenvío de datos, programabilidad, Beacon, OpenDaylight.	7	692

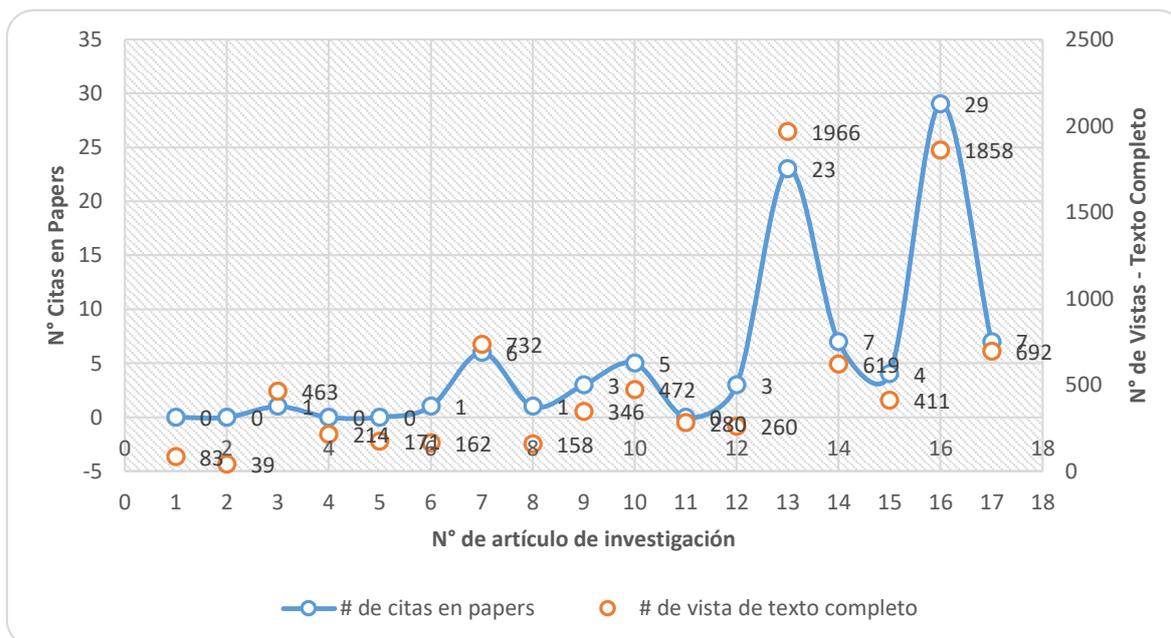


Figura 29 Valoración de los artículos según el número de citas y la cantidad de vistas completas.

## 2.4. Documentos de la Investigación

El objetivo de esta fase es documentar las respuestas a las preguntas de investigación presentadas en el paso 1, así como validar el informe.

### 2.2.9. Validar el informe

Los datos y el análisis de esta revisión integradora fueron validados en todo momento por los autores de este documento como parte del proceso de revisión.

## III. RESULTADOS

La interpretación de los resultados obtenidos se basa en la revisión bibliográfica de los artículos relacionados a las Redes Definidas por Software SDN, y en concreto el análisis comparativo del rendimiento del controlador. A continuación, se muestra en la Figura 35, la cantidad de estudios relacionados al tema antes mencionado ordenados de forma ascendente por año de publicación utilizando la regla de búsqueda con el filtro concatenado. Las evidencias nos indican que desde el año 2015 la tendencia al crecimiento en las investigaciones logró llegar a la cúspide el año 2018, luego presenta un declive hasta la actualidad, se espera que en materia de investigación en los próximos años crezcan los estudios comparativos, que serán el soporte para los administradores en red en lo que concierne a la toma de decisiones.

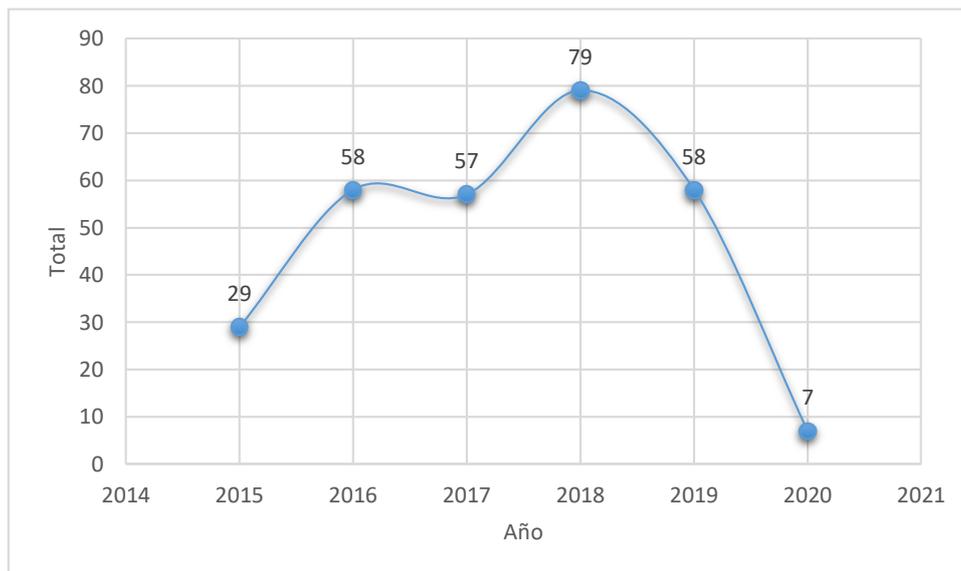


Figura 30 Cantidad de artículos por año aplicando el filtro crudo tal cual arroja la regla de búsqueda

Los 17 documentos sobre análisis de rendimiento de los controladores de Redes Definidas por Software seleccionados se muestran en la Tabla 4.

### 3.1. Análisis

En esta sección se presentan los resultados del análisis respondiendo a las preguntas de la investigación planteada anteriormente [2.2.1] desarrollado como parte de esta literatura de revisión.

#### **PREG 1. ¿Cuántos estudios bibliográficos concernientes al paradigma de Redes Definidas por Software está disponible durante los últimos cinco años?**

La cantidad de material bibliográfico concerniente a las Redes Definidas por Software, es muy amplia. Por ello es importante ser específicos con los criterios de búsqueda en las bases de datos. Debido a la importancia del controlador dentro de la arquitectura SDN, y la diversidad de arquitecturas, dispositivos y propuestas de implementación que surgen en el mercado; es menester comprender y analizar su rendimiento, limitaciones y beneficios.

Por lo tanto, desglosamos el amplio tema, y nos centramos en: Redes Definidas por Software, controlador SDN y los estudios de rendimiento de los mismos, a continuación, se muestra el resultado.

Tabla 5 Resultados de búsqueda de artículos

N°	Año	Título Original	Origen (País)	Autores	Publicado en:	# de citas	# de vistas
1	2019	Topology Discovery Performance Evaluation of OpenDaylight and ONOS Controllers	Paris, France	Mamadou T.BAH ; A. Azzouni ; M.T. Nguyen ; Guy Pujolle	2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)	0	83
2	2019	Comparative Study of Software-Defined Networking (SDN) Traffic Controllers	Braga, Portugal	Gonçalo Pereira ; José Silva ; Pedro Sousa	2019 14th Iberian Conference on Information Systems and Technologies (CISTI)	0	39
3	2018	Performance analysis of software defined network controller architecture—A simulation based survey	Bhubaneswar, India	Madhukrishna Priyadarsini ; Padmalochan Bera ; Rohan Bampal	2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)	1	463
4	2018	Fault Tolerance Comparison of ONOS and OpenDaylight SDN Controllers	Châtillon, France	José Manuel Sanchez Vilchez ; David Espinel Sarmiento	2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)	0	214
5	2018	Evaluation of the QoS parameters in different SDN architecture using Omnet 4.6++	Monastir, Tunisia Marruecos	Fatima Laassiri ; Mohamed Moughit ; Nouredine Idboufker	2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)	0	171

N°	Año	Título Original	Origen (País)	Autores	Publicado en:	# de citas	# de vistas
6	2018	Towards QoS-enabled SDN networks	Marrakech, Morocco	Farah Chahlaoui ; Hamza Dahmouni	2018 International Conference on Advanced Communication Technologies and Networking (CommNet)	1	162
7	2018	A comparative evaluation of the performance of popular SDN controllers	Dubai, Emiratos Árabes Unidos	Lusani Mamushiane ; Albert Lysko ; Sabelo Dlamini	2018 Wireless Days (WD)	6	732
8	2018	Performance analysis of python based openflow controllers	Tadepalligudem, India	Karamjeet Kaur ; Sukhveer Kaur ; Vipin Gupta	3rd International Conference on Electrical, Electronics, Engineering Trends, Communication, Optimization and Sciences (EEECOS 2016)	1	158
9	2017	Experimental evaluation of two OpenFlow controllers	Toronto, ON, Canadá	Mohamad Darianian ; Carey Williamson ; Israat Haque	2017 IEEE 25th International Conference on Network Protocols (ICNP)	3	346
10	2017	Performance Implication and Analysis of the OpenFlow SDN Protocol	Regina, Canada	Aliyu Lawal Aliyu ; Peter Bull ; Ali Abdallah	2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)	5	472

N°	Año	Título Original	Origen (País)	Autores	Publicado en:	# de citas	# de vistas
11	2017	Software defined networks: Comparative analysis of topologies with ONOS	Chennai, India	Ashwin Rajaratnam ; Raturaj Kadikar ; Shanthi Prince ; M. Valarmathi	2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)	0	280
12	2017	A qualitative and quantitative assessment of SDN controllers	Almada, Portugal	Pedro Bispo; Daniel Corujo; Rui L. Aguiar Ola	2017 International Young Engineers Forum (YEF-ECE)	3	260
13	2016	SDN controllers: A comparative study	Lemesos, Cyprus	Salman ; Imad H. Elhajj ; Ayman Kayssi ; Ali Chehab	2016 18th Mediterranean Electrotechnical Conference (MELECON)	23	1966
14	2016	Research and comparative analysis of performance test on SDN controller	Wuhan, China	Fan Yamei ; Liao Qing ; He Qi	2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)	7	619
15	2016	A measurement of the response times of various OpenFlow/SDN controllers with Cbench	Marrakech, Marruecos	C. Laissaoui; N. Idboufker; R. Ellassali; K. El Baamrani	2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)	4	411
16	2015	Performance evaluation of OpenDaylight SDN controller	Hsinchu, Taiwan Pakistan	Zuhran Khan Khattak, Muhammad Awais and Adnan Iqbal	2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)	29	1858

N°	Año	Título Original	Origen (País)	Autores	Publicado en:	# de citas	# de vistas
17	2015	On software-defined networking and the design of SDN controllers	Sydney, Australia	Doan B. Hoang ; Minh Pham	2015 6th International Conference on (NOF)	7	692

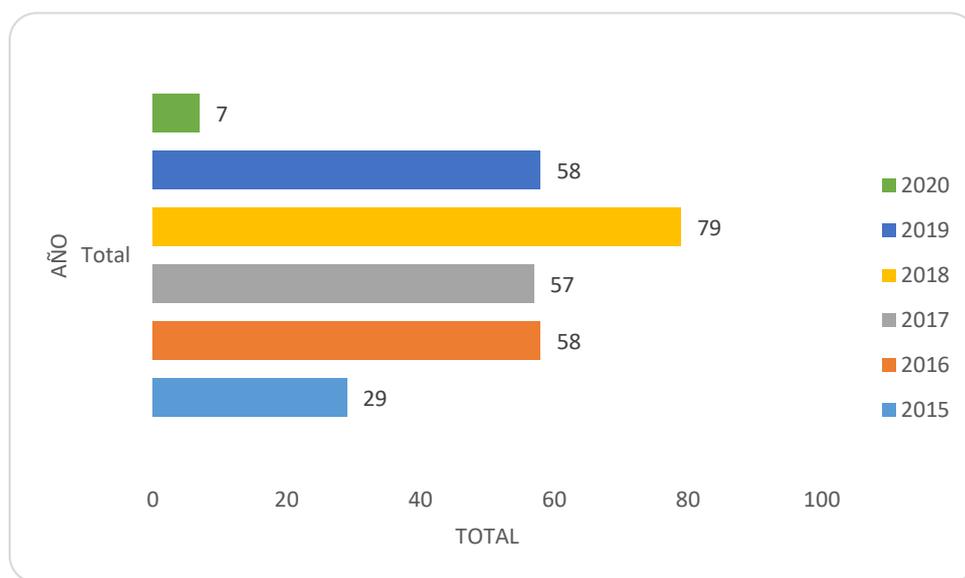


Figura 31 Resultados de búsqueda con filtro

**PREG 2. ¿Qué autores y metodología se han publicado durante los últimos cinco años?**

En este trabajo debemos reconocer el interés y trabajo realizado por los pioneros autores, entre ellos:

Tootoonchian quien fue uno de los primeros en llevar a cabo una evaluación del rendimiento de los controladores SDN (NOT-MT, Beacon y Maestro).

Khondoke quien presenta una comparación basada en características (interfaces hacia el sur, virtualización, GUI, API REST soporte, productividad en términos de velocidades de codificación, programación lenguaje, modularidad, sistema operativo, soporte TLS, madurez, compatible con la versión OpenFlow y OpenStack Soporte de neutrones) de los controladores Ryu, Pox, Trema, Floodlight y OpenDayLight.

Shah quien presenta dos arquitecturas: lotes estáticos (utilizados por Floodlight, Beacon y NOX) y procesamiento por lotes adaptativo (utilizado por Maestro) para mejorar las

implementaciones del controlador bajo varias métricas de rendimiento (latencia, escalabilidad del conmutador e hilo y escalabilidad) que resulto en un banco de pruebas personalizado.

Asímismo, Fernández, comparó las actuaciones en modo reactivo y proactivo de los controladores SDN. NOX, POX, Trema y Floodlight Rowshanrad evaluo y comparo Floodlight y OpenDayLight bajo los indicadores de calidad de servicio QoS que incluyen retraso y pérdida de paquetes, se realizo en diferentes topologías y niveles de cargas de tráfico. Shalimov evalúa los controladores NOX, POX, Beacon, Floodlight, MuL, Maestro, Ryu en función de la latencia, rendimiento, escalabilidad, fiabilidad y seguridad.

Por otro lado, Erickson, argumenta que el lenguaje de programación influye en el rendimiento de un controlador. Salman lleva a cabo una evaluación cualitativa de controladores SDN: MUL, Beacon, Maestro, ONOS, Ryu, OpenDayLight, Floodlight, NOX, IRIS, basado en Libfluid, y POX. Las métricas evaluadas son latencia y rendimiento. Actualmente, seguimos fortaleciendo y generando conocimiento, vital para la sostenibilidad de la tecnología SDN. A continuación, se lista autores que en los últimos cinco años continúan el legado de los pioneros.

Tabla 6 *Autores inmersos en la investigación SDN*

N°	Año	Artículo	Autores	Metología del Experimento
1	2019	Comparative Study of Software-Defined Networking (SDN) Traffic Controllers	Gonçalo Pereira, José Silva Pedro Sousa	microprojetos ilustrativos, o que permitiu efetuar um estudo comparativo no que se refere à curva de aprendizagem e algumas das principais APIs suportadas pelos controladores.
2	2019	Topology Discovery Performance Evaluation of OpenDaylight and ONOS Controllers	Mamadou T.BAH, A.Azzouni, M.T. Nguyen, Guy Pujolle	experimentally evaluate Cbench

N°	Año	Artículo	Autores	Metología del Experimento
3	2018	A Comparative Evaluation of the Performance of Popular SDN Controllers	Lusani Mamushiane, Albert Lysko, Sabelo Dlamini	Cbench ejecutado esencialmente en dos modos: rendimiento y modos de latencia.
4	2018	Evaluation of the QoS parameters in different SDN architecture using Omnet 4.6++	Fatima LAASSIRI Mohamed MOUGHIT Noureddine IDBOUFKER	Omnet 4.6++
5	2018	Ad. Software Defined Networking - SDN, architecture, components and functioning	Miller Ramírez Giraldo Ana María Lopez Echeverry	se propone usar una topología tipo de una red convencional describiendo su comportamiento, para después sobre la misma topología explicar lo que sucedería bajo el nuevo paradigma
6	2018	Performance analysis of python based openflow controllers	Karamjeet Kaur ; Sukhveer Kaur ; Vipin Gupta	Mininet Emulator
7	2018	Performance Analysis of Software Defined Network Controller Architecture—A Simulation Based Survey	Madhukrishna Priyadarsini, Padmalochan Bera, Rohan Bhampal	TCP and UDP mode using iPerf. cbench
8	2018	Towards QoS-enabled SDN networks	Farah CHAHLAOUI Hamza DAHMOUNI	evaluate the performance of real-time We present a simulation based study on the behavior of delay and jitter under increasing load and the impact of SDN network elements on the end-to-end (e2e) QoS of real time services
9	2017	Performance Implication and Analysis of the OpenFlow SDN Protocol	Aliyu Lawal Aliyu, Peter Bull & Ali Abdallah	discusses the performance implications of software switches in comparison with hardware switches used in OpenFlow networks.
10	2017	Software Defined Networks: Comparative Analysis of Topologies with ONOS	Ashwin Rajaratnam, Ruturaj Kadikar, Shanthi Prince M. Valarmathi	Mininet Emulator

N°	Año	Artículo	Autores	Metología del Experimento
11	2017	Experimental Evaluation of Two OpenFlow Controllers	Mohamad Darianian Carey Williamson Israat Haque	We conducted our experiments on stand-alone ONOS and OpenDaylight in three phases, namely physical server (baremetal) without Hyper-Threading (HT), physical server with HT, and virtual host.
12	2017	A Qualitative and Quantitative assessment of SDN Controllers	Pedro Bispo, Daniel Corujo, Rui L. Aguiar	benchmarking tool Cbench OpenStack
13	2016	SDN Controllers: A Comparative Study	Ola Salman Imad H. Elhaggi Ayman Kayssi Ali Chehab	Cbench ejecutado esencialmente en dos modos: rendimiento y modos de latencia.
14	2016	Research and Comparative Analysis of Performance Test on SDN Controller	Fan Yamei, Liao Qing, He Qi	IXIA Test Instrument(XG12) Cbench Mininet
15	2016	A measurement of the response times of various OpenFlow/SDN controllers with Cbench	C. Laissaoui, N. Idboufker, R. Ellassali and K. El Baamrani	Cbench ejecutado esencialmente en dos modos: rendimiento y modos de latencia.
16	2015	Performance Evaluation of OpenDaylight SDN Controller	Zuhran Khan Khattak, Muhammad Awaiss and Adnan Iqbal	Cbench ejecutado esencialmente en dos modos: rendimiento y modos de latencia.
17	2015	On Software-defined networking and the design of SDN Controllers	Doan B. Hoang Minh Pham	theoretical analysis
18	2015	Ad Controladores SDN, elementos para su selección y evaluación.	Alejandro García Centeno1, Carlos Manuel Rodríguez Vergel2, Caridad Anías Calderón3, Frank Camilo Casmartíño Bondarenko4	Cbench y Hcprobe.

**PREG 3. ¿Qué métricas y características para la evaluación de los controladores, destacan en los estudios de Redes Definidas por Software?**

La alta importancia del controlador SDN, radica en que dentro de sus funciones se encuentra la gestión global de los dispositivos subadyacentes, y complementarios. Por tal motivo, evaluarlo, generar banco de pruebas y analizar los resultados, serán el respaldo para elegir el más óptimo, según las necesidades o requerimientos de la red. A continuación, se detalla los resultados de los artículos en cuanto a que controladores fueron seleccionados para el estudio, bajo que características y las métricas a evaluar.

Tabla 7 Métricas y elementos para evaluación de controladores.

N°	Año	Artículo	Métricas de Comparación	Controladores Analizados	Características o Elementos de Evaluación
1	2019	Comparative Study of Software-Defined Networking (SDN) Traffic Controllers	Curva de Aprendizagem (Rápida, Moderada, Lenta) Exemplos de APIs Suportadas	Floodlight POX ONOS OpenDayLight	Ano Linguagem de Programação Plataforma Interface Documentação Versão OF Exemplos de Áreas de Trabalhos Desenvolvidos
2	2019	Topology Discovery Performance Evaluation of OpenDaylight and ONOS Controllers	topology discovery topology update throughput	ONOS OpenDaylight	Topology instances of the cluster
3	2018	A Comparative Evaluation of the Performance of Popular SDN Controllers	throughput and latency	Ryu Floodlight ONOS Opendaylight	REST API, Graphical user interface (GUI), Modularity, Orchestrator support, operating system (OS) supported, Partnership, Documentation, Programming language, Multi-threading support, TLS support, Virtualization, Application domain and architecture.

N°	Año	Artículo	Métricas de Comparación	Controladores Analizados	Características o Elementos de Evaluación
4	2018	Evaluation of the QoS parameters in different SDN architecture using Omnet 4.6++	evaluation of QoS parameters such as gigue, latency, lost packets, end-to-end delay,	Unspecified	Topology-
5	2018	Ad. Software Defined Networking - SDN, architecture, components and functioning	Arquitectura y componentes Herramientas disponibles para la implementacion Fabricantes con desarrollos	Unspecified	Architecture
6	2018	Performance analysis of python based openflow controllers	Programming Language	POX, RYU Pyretic.	REST API Support Open Source License Platform Support
7	2018	Performance Analysis of Software Defined Network Controller Architecture—A Simulation Based Survey	Latency Jitter Bandwidth Workload.	NOX, NOX-MT, POX, Rosemary, FloodLight, Beacon, Maestro, OpenDayLight.	Protocol of communication between controllers: Synchronization cost:
8	2018	Towards QoS-enabled SDN networks	delay jitter	Unspecified	OpenFlow 1.3
9	2017	Performance Implication and Analysis of the OpenFlow SDN Protocol	The delay accumulated by flows (datapath requests) as they traverse the control path. How many flows a controller can process per second	Floodlight Pox	Open Flow Programmin Language
10	2017	Software Defined Networks: Comparative Analysis of Topologies with ONOS	RTT Topology	Onos	North Bound APIs (NBIs) South Bound APIs(SBIs=
11	2017	Experimental Evaluation of Two OpenFlow Controllers	the throughput, latency, thread scalability	ONOS OpenDaylight.	Memory Usage Results CPU Utilization Thread Placement

N°	Año	Artículo	Métricas de Comparación	Controladores Analizados	Características o Elementos de Evaluación
12	2017	A Qualitative and Quantitative assessment of SDN Controllers	Latency Tests Throughput Tests	OpenDaylight (ODL), Open Network Operative System (ONOS), Ryu POX	NB interface SB interface Gul REST Apl Programmin Language Openflow Support Modularity Multi threading support Documentation
13	2016	SDN Controllers: A Comparative Study	performance	Onos Pox Nox Mul Beacon Floodlight Maestro Iris Ryu Libfluid_msg Libfluid_raw Opendaylight	Programming Language GUI Documentation Modularity Distributed Centralized Platform Support Southbound APIs Northbound APIs Partner Multithreading Support OpenStack Support Application Domain
14	2016	Research and Comparative Analysis of Performance Test on SDN Controller A measurement of the response times of various	time of Openflow channel, response time of topology discovery	OpenDaylight ONOS	System Structure Model Framework
15	2016	OpenFlow/SDN controllers with Cbench	throughput and latency	FloodLight, Beacon, Pox Ryu	Open Flow
16	2015	Performance Evaluation of OpenDaylight SDN Controller	throughput and latency	Floodlight OpenDaylight	East-West traffic Nort-South traffic
17	2015	On Software-defined networking and the design of SDN Controllers	structure core components	Beacon, OpenDaylight, Open Networking Operation System	Objective Architecture Core Network Services Southbound API/protocols Northbound API East/Westbound API Service Abstraction Layer / Plug-ins Management Interfaces Programming Language

N°	Año	Artículo	Métricas de Comparación	Controladores Analizados	Características o Elementos de Evaluación
18	2015	Ad Controladores SDN, elementos para su selección y evaluación.	soporte OpenFlow, virtualización de red, funcionalidad de la red, escalabilidad, rendimiento, programación de red, confiabilidad, seguridad de la red, monitorización centralizada y visualización, fabricantes de controladores SDN, soporte de plataformas y procesamiento.	Beacon Floodlight NOX POX Ryu Trema OpenDayLight (ODL).	Soporte OpenFlow: Virtualización de red: Funcionalidad de la red: Escalabilidad: Rendimiento:

**PREG 4. ¿Qué problemas enfrentaron los estudios, como lo resolvieron y qué conclusiones nos guían en las siguientes investigaciones?**

Como tecnología emergente, las SDN, se presentan ante dificultades, que se convierten en retos para los investigadores, a continuación, se presenta los incidentes previos, durante o post despliegue de los controladores. Asimismo, rescatar la forma que lo resolvieron y captar las conclusiones y recomendaciones, nos ayudara a entender holísticamente la implementación.

Tabla 8 *Hallazgos importantes – Despliegue de las redes SDN*

N <sup>o</sup>	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
1	2019	Evaluación del rendimiento de descubrimiento de topología de los controladores OpenDaylight y ONOS	El universo de investigaciones sobre el controlador SDN, se han centrado en que tan velozes en el procesamiento.	Se evaluó experimentalmente el desempeño de ONOS y ODL en términos de tiempo de descubrimiento de topología y tiempo de reacción en caso de cambios en la topología.	ONOS ofrece bajo su orquestación un máximo de velocidad de datos de usuario de hasta 30 Gbps, mientras que ODL no puede exceder 512 Mbps. Los resultados indicaron que ONOS puede gestionar una mayor cantidad de flujos, contribuyendo al óptimo rendimiento de los datos del usuario en la red	ONOS reacciona mejor para vincular la falla en una red, con un tiempo de reacción de aproximadamente 100 ms. Este controlador es adecuado para redes dinámicas en las que la topología puede cambiar con frecuencia
2	2019	Estudio Comparativo de Controladores Software-Defined Networking (SDN) - Controladores de Tráfico	La diversidad de controladores SDN así como sus características heterogéneas y potencialidades, dificultan su elección.	Un estudio comparativo entre algunos de los controladores discutidos anteriormente.	De esta experiencia más práctica, se obtuvo información muy relevante sobre las curvas de aprendizaje y algunas características compatibles con las API proporcionadas.	En un entorno OpenFlow, cualquier equipo que desea comunicarse con el controlador necesita apoyar este protocolo. Una vez verificado, es Secure realiza una conexión entre el controlador y el conmutador / enrutador

N.º	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
3	2018	Análisis de rendimiento de la arquitectura de controlador de red definida por software una encuesta basada en simulación	SDN, presenta nuevos desafíos hacia la escalabilidad y el rendimiento, el endurecimiento de la seguridad, protocolos de comunicación cruzados, etc.	Presentamos el estudio de la cantidad de parámetros y sus variaciones pueden afectar el rendimiento del controlador SDN.	De todas las simulaciones observamos que la POX es la versión avanzada de NOX proporciona mejor rendimiento. NOX y POX demostró funcionar eficientemente en tráfico homogéneo mientras que para heterogéneo el tráfico Beacon es el mejor.	Durante el tráfico intenso la carga es más pesada y por ende el rendimiento es menor en comparación con los que se asigno menos carga. Se sugiere que el controlador debería ser diseñado de tal manera que en alta carga mantenga su desempeño.
4	2018	Comparación de tolerancia a fallos de los controladores SDN de ONOS y OpenDaylight	¿Porqué el controlador SDN es el único responsable de proporcionar las decisiones correctas de reenvío al plano de datos para reenviar paquetes?	Implementar topologías de red personalizadas donde podemos inyectar varios tipos de fallas y monitorear el tráfico.	El controlador ONOS supera a ODL en términos de cambio a otros caminos adicionales para garantizar la continuidad del servicio.	El aspecto más importante es la capacidad de las aplicaciones SDN que controlan los controladores para detectar fallas en el plano de datos y eludirlas mediante el uso disponible recursos para garantizar la continuidad del servicio.

N <sup>o</sup>	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
5	2018	Evaluación de los parámetros de QoS en diferentes arquitecturas SDN usando Omnet 4.6	Los investigador es continúan experimentando para mejorar el rendimiento y sus capacidades para gestionar un gran número de equipos de red. Se basan en dos, la latencia y el número de peticiones procesadas por segundo	Se realizo una simulación de calidad de servicio en diversas topologías bajo OMNeT 4.6 ++.	La topología jerárquica es la más baja (200 ms), la fluctuación de fase mostrado por la topología de anillo es la más básica (6 ms) que indica el retardo más fiable, la latencia de o retardo dedicado por el tiempo en la topología de árbol 65ms es la más básica El número de paquetes perdidos que se ofrecen por la topología jerárquica es la más baja de 1% que determina la arquitectura más eficiente.	Los resultados indican que la topología jerárquica cuenta con el valor más alto de transmisión y por ende se convierte en la más fiable en comparación con estrella, anillo y árbol.
6	2018	Hacia redes SDN habilitadas para QoS	Se programa las tareas de gestión de tráfico a nivel de términos de retardo y jitter para aplicaciones en tiempo real.	Los autores construyeron un API de QoS que permite la provisión de QoS en las redes OpenFlow basados en controladores SDN.	Los resultados muestran que el número de paquetes generados disminuye la calidad de servicio.	Para garantizar la calidad propone el desarrollo de un modelo de optimización para los controladores, que en base a la naturaleza de tráfico (jitter o retrasar sensibles).

N <sup>o</sup>	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
7	2018	Una evaluación comparativa del rendimiento de los controladores SDN populares	<p>Dos de las preguntas más importantes frecuentemente formuladas son (a) ¿qué tan rápido puede responder un controlador a PACKET_IN mensajes?; y (b) cuántos mensajes PACKET_IN puede tener un controlador manejar por segundo?</p>	<p>Evaluamos a Ryu, Floodlight, ONOS y OpenDayLight. En términos de rendimiento y la latencia, con la herramienta Cbench.</p>	<p>ONOS tiene en general el mejor rendimiento a medida que aumenta el tamaño del plano de datos. Bajo diferentes cargas de trabajo (MAC), ONOS aún muestra una escalabilidad sobresaliente.</p>	<p>OpenDayLight presenta mejor compatibilidad con los proveedores de interfaces. ONOS tiene la capacidad de responder a solicitudes más rápido bajo varias cargas de tráfico, Ryu sobresale en las pruebas de latencia. Resumen la elección del controlador depende completamente de los requisitos del usuario.</p>
8	2018	Análisis De Rendimiento De Controladores De Flujo Abierto Basados En Python	<p>Las redes tradicionales presentan inflexibilidad, carecen de una administración descentralizada y los dispositivos no son programables..</p>	<p>Se comparo controladores basados en Python:POX, RYU y Pyretic.</p>	<p>RTT fue el menos en el caso de Ryu, mientras que fue máximo en el caso de Pyretic</p>	<p>El rendimiento de la aplicación creada en Ryu es muy rápido en comparación con POX y Pyretic. Ryu presenta una ventaja al admitir OpenFlow 1.4.</p>

N.º	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
9	2017	Evaluación experimental de dos controladores OpenFlow	Hasta la fecha, no se encuentra disponible un estudio que evalúe el rendimiento de ONOS y OpenDaylight.	Evaluar las últimas versiones de los controladores.	Observamos que el rendimiento de ONOS disminuyó ligeramente después de 12 hilos (entre 12-15 hilos). Esta tendencia refleja el retraso de comunicación entre subprocesos en CPU separada enchufes	En resumen, ONOS y OpenDaylight muestran un mayor rendimiento y menor latencia cuando Hyper-threading está habilitado. Nuestro estudio mostró que el rendimiento de ONOS y OpenDaylight difiere en el host virtual.
10	2017	Implicación del rendimiento y análisis del protocolo OpenFlow SDN	La arquitectura de red informática tradicional es compleja, inflexible y estática	Analizar controladores en términos de velocidad de bits, retraso promedio, tasa de paquetes y jitter promedio.	RYU es mejor en el manejo de enrutamiento de nuevo paquetes a su destino.	El estudio se puede utilizar como plantilla para desarrolladores SDN o investigadores para facilitar la elección del controlador SDN.

N°	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
1	2011	Redes definidas por software análisis comparativo de topologías con ONOS	Los diferentes servicios como Cloud requieren converger a un sistema más eficiente de red.	El controlador ONOS en diversas topologías, se analizan las configuraciones RTT para mensajes en 16 interruptores y 32 interruptores	La eficiencia de la red tiende a disminuir con un aumento en el número de interruptores. Por lo tanto, la topología de torus muestra ser la mejor para redes de alta velocidad y latencia extremadamente bajas en comparación a otras topologías.	La topología de torus es la más adecuada para alta velocidad y Redes de baja latencia. El RTT puede reducirse aún más seleccionando las rutas de baja latencia o ancho de banda alto a través de Controlador ONOS.
1	2012	Una evaluación cualitativa y cuantitativa de los controladores SDN	Los estudios se centran solo en el rendimiento y no están a la altura fecha, ya que las nuevas versiones de los controladores más populares son constantemente lanzadas.	Se realizaron varias pruebas de estrés usando Cbench, en modo latencia y rendimiento,	Ryu y OD-L se destacó, seguido de cerca por ONOS. ODL logra los mejores resultados cuando emulando cuatro o más interruptores mientras que los otros controladores tener un rendimiento similar, a pesar de la alta tasa de flujos por segundo por Ryu para un menor número de interruptores.	La comparaciónb cualitativa mostró que tanto ONOS como ODL son los más destacados

N <sup>o</sup>	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
13	2016	Controlador es SDN: Un estudio comparativo	Debido a la importancia del controlador hay una necesidad de evaluar y comparar su desempeño.	La comparación considera medidas cuantitativas (rendimiento y latencia), y criterios cualitativos. Con el fin de facilitar la elección del controlador apropiado para una aplicación específica dominio..	Los resultados muestran que los controladores y codificados por el lenguaje C dieron el más alto rendimiento, seguidos por los codificados de Java.	Se descubrió que OpenDaylight es una buena opción como controlador con todas las funciones. Su integración de los datos de IoT intermediario y nuevas interfaces hacia el sur específicas de IoT en su último lanzamiento lo convierten en el primer competidor en la elección del Controlador de "Internet del futuro".
14	2016	Investigación y análisis comparativo de la prueba de rendimiento en el controlador SDN	En la arquitectura SDN, el controlador puede cambiar la construcción y el servicio de la red. El controlador SDN es crucial para toda la red.	OpenDaylight y ONOS tienen una amplia aplicación y una arquitectura más razonable, por lo que son seleccionados para su evaluación.	Los resultados muestran que ONOS tiene un buen rendimiento en GUI, clústeres, conexión, conmutación y rendimiento. OpenDaylight tiene un mejor rendimiento en términos de descubrimiento de topología y estabilidad.	OpenDaylight tiene algunos problemas, como pérdidas de memoria.

N <sup>o</sup>	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
15	2016	Una medida de los tiempos de respuesta de varios controladores es OpenFlow SDN con CBench	La mayoría de las preocupaciones sobre esta nueva tendencia de red es: ¿qué tan rápido puede el controlador responder a solicitudes de ruta de datos y cuántas solicitudes de ruta de datos puede un controlador manejar por segundo.	Se realizó un estudio y evaluación de Floodlight, Beacon, Pox y Ryu en términos de latencia y rendimiento.	En general, Floodlight es el mejor controlador para usar al elegir entre él y Beacon, Ryu o Pox en una red que usa OpenFlow 1.0.	Las diferencias en los resultados del rendimiento podría ser el lenguaje de programación, en el aspecto de cómo ocupa los espacios de memoria. También podría ser la forma en que cada controlador está programado, la cantidad de procesos que desencadena para cada uno de estos eventos, el tamaño de los datos intercambiados durante cada comunicación de solicitud / respuesta y también los tamaños de memoria asignado para los almacenamientos intermedios.

N <sup>o</sup>	Año	Titulo	Problema que enfrentó	¿Que se hizo?	¿Qué resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes
16	2015	Evaluación del rendimiento del controlador SDN de OpenDaylight	Aunque existen varios estudios, OpenDaylight no ha sido considerado en ningún estudio de evaluación comparativa todavía.	Los autores presentan el esquema conocido como ECHO que contiene modelos de cadena de Markov distribución y se pueden utilizar para generar modelos de tráfico que pueden capturar las características en función del espacio y el tiempo.	Hemos encontrado que la evaluación comparativa de la EAD con Cbench no es muy exitosa. Parece que ODL tiene varios problemas tales como fugas de memoria.	Debido al gran número de pruebas insatisfactorias tenemos la intención de llevar a cabo código de profiling de la EAD. También hemos presentado una mejorada Cbench y su implementación prototipo.
17	2015	Sobre redes definidas por software y el diseño de Controladores SDN	Internet ha llegado a un punto en que es muy difícil de explorar nuevas arquitecturas que son más adecuados para aplicaciones emergentes	En particular, la revisión de esta investigación se basa en la comparación del diseño de los tres controladores: Beacon, OpenDaylight y Open Networking Operation System.	La interfaz hacia el sur es extendido para admitir otras API, protocolos de complementos y Servicios distintos de OpenFlow, como el servicio (DNS), Protocolo de configuración dinámica de host (DHCP), OVSDB,NETC ONF	A medida que SDN gana más y más adopción, los requisitos para los controladores SDN son: alta escalabilidad, disponibilidad y seguridad

### **PREG 5. ¿Cuáles son los beneficios de adoptar las Redes Definidas por Software?**

Las Redes Definidas por Software - SDN ostenta grandes beneficios, tales como; es directamente programable, al estar en desacople con las funciones de reenvío, es ágil, al permitir configurar ajustes en el flujo de tráfico en toda la red, mantiene una visión global de la infraestructura al centralizar la inteligencia, permite ser programado para optimizar los recursos a través de programas automatizados y dinámicos, y por ultimo al estar basado en normas de estándares abiertos simplifica el diseño y la operación de la red.

Tabla 9 *Comparación de características*

	<b>SDN</b>	<b>Redes convencionales</b>
<b>Característica</b>	Desacoplado del control de datos y control, ofrece programabilidad	Complejo control de la red. Diversidad de protocolos.
<b>Configuración</b>	Automatizado mediante la configuración con estructura de centralizado.	Configuración manual, expuesta a errores.
<b>Rendimiento</b>	Dinamicamente controlado, globalmente disponible con Capa cruzada de información	Configuración estática con limitada información.
<b>Innovación</b>	Fácil actualización de software y hardware, prueba suficiente ambiente, rápido despliegue	Difícil implementación de hardware y software, ofrece limitado entornos de prueba

## **IV. CONCLUSIONES**

La revisión bibliográfica exhaustiva permite adquirir conocimiento útil para posteriores estudios, y siendo recíprocos con ello, este trabajo presenta una revisión actualizada, compacta y experimental, del análisis de rendimiento de controladores SDN, se asumió el reto y los resultados obtenidos son muchos, entre ellos:

Comprender el funcionamiento de la arquitectura SDN, sus componentes, subcomponentes, funciones, características y criterios de evaluación. Actualmente la ONF, ha facilitado las métricas para proyectos de migración de redes tradicionales a redes

SDN. Como primera conclusión, contamos con numerosos estudios realizados sobre Redes Definidas por Software, pero se ha notado que no están clasificados, de la forma que sirva para la implementación sencilla del controlador, y esto se debe a que cada año se lanzan versiones con sus propias opciones de mejora, que cambian y no permiten alcanzar la estabilidad de un controlador. En este trabajo del universo de 288 coincidencias con el tema, solo 17 fueron seleccionadas para un análisis profundo y de base para este estudio.

Desde los autores pioneros hasta la actualidad, se han publicado diversas metodologías buscando generar conocimiento para implementaciones reales, comprendemos que el uso de las herramientas de evaluación de rendimiento, como, por ejemplo, CBench, han sido de gran utilidad para generar los bancos de pruebas, por tal motivo en este trabajo se tomara en cuenta utilizar la misma herramienta, verificando primero la compatibilidad con el controlador seleccionado. Se busca generar resultados en cuanto a latencia, perdida de paquetes como criterio de evaluación.

Según la literatura analizada, los controladores con características mas estables y que en su mayoría han sido utilizados, son ONOS y OpenDayLight, por su versatilidad y compatibilidad con Java, el lenguaje de programación que permite el uso de multiplataforma. La carencia de información loo requisitos previos para un correcto despliegue, el avance tecnológico y propuestas de los proveedores, de servicios y dispositivos de red, son algunos de los problemas que deben ser resueltos con trabajos de investigación, para disipar dudas y otorgar una visión global de la arquitectura y su funcionamiento. Las empresas cada dia se interesan mas en adoptar, migrar completamente o progresivamente a las redes SDN, debido a las bondades que solucionan las deficiencias de las redes tradicionales, y toman en cuenta los beneficios que las empresas que ya han adoptado esta tecnología sustentan.

## **V. RECOMENDACIONES**

Debido a que las redes SDN, son por el momento la mejor opción y se presentan como el futuro de las redes, debemos estar informados sobre las formas y resultados de cada implementación, que sirva de guía para posteriores análisis o despliegues.

Tabla 10 Matriz De Resumen De La Revisión Bibliografica De Los Articulos De Interes

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
1	2019	Evaluación del rendimiento de descubrimiento de topología de los controladores OpenDay light y ONOS	Mamadu T. BAH ; A. Azzouni ; M.T. Nguyen ; Guy Pujolle	gestión de redes informáticas, evaluación del rendimiento de redes informáticas, redes definidas por software, topología de redes de telecomunicaciones	Los controladores SDN necesitan estar actualizados o información sobre la topología de la red,	El universo de investigaciones sobre el controlador SDN, se han centrado en que tan veloz es en el procesamiento.	Se evaluó experimentalmente el desempeño de ONOS y ODL en términos de tiempo de descubrimiento de topología y tiempo de reacción en caso de cambios en la topología.	Se usa Cbench para evaluar las actuaciones de controladores en términos de rendimiento y latencia de solicitud	Los resultados indicaron que ONOS puede gestionar una mayor cantidad de flujos, contribuyendo al óptimo rendimiento de los datos del usuario en la red	ONOS reacciona mejor para vincular la falla en una red, Este controlador es adecuado para redes dinámicas en las que la topología puede cambiar con frecuencia	0	83

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
2	2019	Estudio Comparativo de Controladores Software-Defined Networking (SDN) - Controladores de Tráfico	Gonçalo Pereira ; José Silva ; Pedro Sousa	interfaces de programación de aplicación, Internet, radio móvil, redes definidas por software, control de telecomunicaciones	Las redes SDN continúan teniendo un gran impacto no solo operación de la red en sí, pero también en todo el desarrollo realizado a su alrededor	La diversidad de controladores SDN existentes, así como sus características heterogéneas y potencialidades, son factores que dificultan la elección de plataformas.	Un estudio comparativo entre algunos de los controladores discutidos anteriormente.	Se plantearon unos microproyectos que se desarrollaron utilizando una selección más restringido de cuatro controladores SDN (Floodlight, POX, ONOS y ODL).	De esta experiencia más práctica, se obtuvo información muy relevante sobre las curvas de aprendizaje controlador y algunas características compatibles con las APIs proporcionadas.	En un entorno OpenFlow, cualquier equipo que desea comunicar con el controlador necesita apoyar este protocolo.	0	39

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
3	2018	Análisis de rendimiento de la arquitectura de controlador de red definida por software una encuesta basada en simulación	Madhukrishna Priyadarsini ; Padmalochan Bera ; Rohan Bampal	seguridad de la red informática, protocolos, software de dominio público, arquitectura de software, redes definidas por software, métricas de software, evaluación del rendimiento del software		SDN, presenta nuevos desafíos hacia la escalabilidad y el rendimiento, el endurecimiento de la seguridad, protocolos de comunicación cruzados, etc.	Presentamos el estudio de la cantidad de parámetros y como sus variaciones pueden afectar el rendimiento del controlador SDN.	Varias herramientas se utilizan para calcular el rendimiento, latencia, tiempo de respuesta, jitter en SDN. Algunas herramientas se mencionan con su implementación prospectiva.	De todas las simulaciones observamos que POX es la versión avanzada de NOX que proporciona un mejor rendimiento. NOX y POX demostró funcionar eficientemente en tráfico homogéneo y heterogéneo Beacon es el mejor.	Se sugiere que el controlador debería ser diseñado de tal manera que en alta carga mantenga su desempeño.	1	463

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
4	2018	Comparación de tolerancia a fallos de los controladores SDN de ONOS y OpenDay light	José Manuel Sanchez Vilchez ; David Espinel Sarmiento	tolerancia a fallas, redes definidas por software, confiabilidad de la red de telecomunicaciones, topología de la red de telecomunicaciones	La centralización e inteligencia de red en el controlador SDN plantea muchas preguntas sobre las propiedades de tolerancia a fallas.	¿Porqué el controlador SDN es el único responsable de proporcionar las decisiones correctas de reenvío al plano de datos para reenviar paquetes?	Implementar topologías de red personalizadas donde podemos inyectar varios tipos de fallas y monitorear el tráfico.	Se revisa el estado del arte en el código abierto iniciativas para simular una infraestructura SDN, que incluyen iniciativas para simular el controlador SDN y las iniciativas para simular elementos del plano de datos. Se genera diferentes escenarios de fallas en una infraestructura SDN totalmente personalizable.	El controlador ONOS supera a ODL en términos de cambio a otros caminos adicionales para garantizar la continuidad del servicio.	El aspecto más importante es la capacidad de las aplicaciones SDN para detectar fallas en el plano de datos y eludirlas mediante el uso disponible recursos para garantizar la continuidad del servicio.	0	214

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
5	2018	Evaluación de los parámetros de QoS en diferentes arquitecturas SDN usando Omnet 4.6	Fatima Laassiri ; Mohamed Moughit ; Nouredine Idboufkeur	calidad de servicio, protocolos de enrutamiento, redes definidas por software, tráfico de telecomunicaciones	La evaluación del rendimiento de los parámetros de QoS, bajo OMNeT 4,6 ++ para todo tipo de topologías y simulación del protocolo OpenFlow	Los investigadores continúan experimentando. Se basan la latencia y el número de peticiones procesadas por segundo	Se realizo una simulación de calidad de servicio en diversas topologías bajo OMNeT 4,6 ++.	Se instala una o más entradas en el interruptor. A menudo, el controlador se instala la ruta completa para el paquete en la red mediante la modificación de las tablas de flujo de todos los switchs que componen la ruta	La topología jerárquica es la más baja. El número de paquetes perdidos que se ofrecen por la topología jerárquica es la más baja de 1% que determina la arquitectura más eficiente.	Los resultados indican que la topología jerárquica cuenta con el valor más alto de transmisión y por ende se convierte en la más fiable en comparación con estrella, anillo y árbol.	0	171

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
6	2018	Hacia redes SDN habilitadas para QoS	Farah Chahloui ; Hamza Dahmouni	jitter, calidad de servicio, redes definidas por software	Se centran en OpenFlow 1.3 para investigar el comportamiento de los indicadores de calidad de servicio en la SDN	Es necesario programar las tareas de gestión de tráfico a nivel de tráfico para satisfacer las necesidades en términos de retardo y jitter para aplicaciones en tiempo real y la pérdida de tiempo.	Los autores construyeron un API de QoS que permite la provisión de QoS en las redes OpenFlow basados en controladores SDN.	Usamos netperf para generar la UDP tráfico entre la fuente y su destinations. Comenzamos con la simulación de una sola de flujo para evaluar su retardo y jitter, a la que añadiremos un segundo tráfico y aumentamos la carga de fondo de tráfico progresivamente	Los resultados muestran que aumentar el número de paquetes generados disminuye la calidad de servicio.	Para garantizar la calidad de la experiencia de los usuarios, se propone el desarrollo de un modelo de optimización para los controladores.	1	162

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
7	2018	Una evaluación comparativa del rendimiento de los controladores SDN populares	Lusani Mamushi ; Albert Lysko ; Sabelo Dlamini	evaluación del rendimiento de la red informática, toma de decisiones, redes definidas por software	comparativa del rendimiento de controladores ONOS, Ryu, FloodLight y OpenDayLight SDN	¿qué tan rápido puede responder un controlador a PACKET_IN mensajes?; y (b) cuántos mensajes PACKET_IN puede tener un controlador manejar por segundo?	Evaluamos a Ryu, Floodlight, ONOS y OpenDayLight. En términos de rendimiento y la latencia, con la herramienta Cbench.	La evaluación se lleva a cabo usando Cbench en modo de latencia y modo de rendimiento.	ONOS tiene en general el mejor rendimiento a medida que aumenta el tamaño del plano de datos. Bajo diferentes cargas de trabajo (MAC), ONOS aún muestra una escalabilidad sobresaliente.	OpenDayLight presenta mejor compatibilidad con los proveedores de interfaces. ONOS tiene la capacidad de responder a solicitudes más rápido bajo varias cargas de tráfico.	6	732

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
8	2018	Análisis De Rendimiento De Controladores De Flujo Abierto Basados En Python	Karamjeet Kaur ; Sukhveer Kaur ; Vipin Gupta	computer centers, computer networks, Java, protocols, software defined networking, telecommunication network topology, telecommunication switching	Análisis de rendimiento de controladores de flujo abierto basados en python	Las redes tradicionales presentan inflexibilidad, carecen de una administración descentralizada y los dispositivos no son programables..	Se comparo controladores basados en Python:POX, RYU y Pyretic.	El experimento se realizó utilizando VMPlayer e importando la máquina Mininet en VMPlayer. Mininet emulador se utilizó para crear topología de red. Mininet se puede utilizar para crear redes de cualquier tamaño que contengan hosts, conmutadores y controladores	RTT fue el menos en el caso de Ryu, mientras que fue máximo en el caso de Pyretic	El rendimiento de la aplicación creada en Ryu es muy rápido en comparación con POX y Pyretic. Ryu presenta una ventaja al admitir OpenFlow 1.4.	1	158

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
9	2017	Evaluación experimental de los controladores OpenFlow	Mohamad Darianian ; Carey Williams ; Israat Haque	protocolos, software de dominio público, redes definidas por software, gestión de redes de telecomunicaciones, tráfico de telecomunicaciones	Evaluación experimental de dos controladores OpenFlow	Hasta la fecha, no se encuentra disponible un estudio que evalúe el rendimiento de ONOS y OpenDaylight.	Evaluar las últimas versiones de los controladores.	Ejecutamos controladores y Cbench en máquinas virtuales separadas por separado computar los nodos para la coherencia con la configuración en el físico anfitrión.	Observamos que el rendimiento de ONOS disminuyó ligeramente después de 12 hilos (entre 12-15 hilos). Esta tendencia refleja el retraso de comunicación entre subprocesos en CPU separada enchufes	En resumen, ONOS y OpenDaylight muestran un mayor rendimiento y menor latencia cuando Hyper-threading está habilitado	3	346

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
10	2017	Implicación del rendimiento y análisis del protocolo OpenFlow SDN	Aliyu Lawal Aliyu ; Peter Bull ; Ali Abdallah	evaluación del rendimiento de la red informática, protocolos, redes definidas por software, tráfico de telecomunicaciones	El controlador es el lugar donde toda la inteligencia de SDN reside; Por lo tanto, es importante probar las capacidades de controlador antes de implementarlo.	La arquitectura de red informática tradicional es compleja, inflexible y estática	Analizar controladores en términos de velocidad de bits, retraso promedio, tasa de paquetes y jitter promedio.	Como solución propuesta, MININET (emulador) se utiliza para emular las topologías para monitorear el tráfico y el rendimiento. El generador de tráfico de Internet (D-ITG) se utiliza para generar TCP tráfico a través de esas topologías personalizadas.	RYU es mejor en el manejo de enrutamiento de nuevo paquetes a su destino.	El estudio se puede utilizar como plantilla para desarrolladores SDN o investigadores para facilitar la elección del controlador SDN	5	472

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
1	2017	Redes definidas por software análisis comparativo de topologías con ONOS	Ashwin Rajaratnam ; Ruturaj Kadikar ; Shanthi Prince ; M. Valarmathi	centros informáticos, sistemas operativos de red, topología de red, calidad de servicio, redes definidas por software, enrutamiento de redes de telecomunicaciones, topología de redes de telecomunicaciones, tráfico de telecomunicaciones	Lo complejo el software de control se encuentra en un controlador centralizado que tiene una vista completa de la topología de la red y toma decisiones de enrutamiento.	Los diferentes servicios como Cloud requieren converger a un sistema más eficiente de red.	El controlador ONOS en diversas topologías, se analizan las configuraciones RTT para mensajes en 16 interruptores y 32 interruptores	La idea básica detrás de esta experimentación es proporcionar el importancia de ONOS como controlador y tipo de topología mejor adecuado para redes existentes y futuras.	La eficiencia de la red tiende a disminuir con un aumento en el número de interruptores. Por lo tanto, la topología de torus muestra ser la mejor para redes de alta velocidad y latencia extremadamente bajas en comparación a otras topologías.	La topología de torus es la más adecuada para alta velocidad y baja latencia.	0	280

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallázgos concretos)	Conclusiones relevantes	# de citas	# de vistas
1 2	20 17	Una evaluación cualitativa y cuantitativa de los controladores SDN	Pedro Bispo; Daniel Corujo; Rui L. Aguiar	Comunicación móvil 5G, gestión de redes informáticas, gestión de movilidad (radio móvil), redes definidas por software, control de telecomunicaciones	Una evaluación cualitativa y cuantitativa de los controladores SDN	Los estudios se centran solo en el rendimiento y no están a la altura fecha, ya que las nuevas versiones de los controladores más populares son constantemente lanzadas.	Se realizaron varias pruebas de estrés usando Cbench, en modo latencia y rendimiento ,	Por otro lado, para verificar el impacto del número de hosts emulados por conmutador, también ejecutamos Cbench en rendimiento modo, pero teniendo un número fijo de interruptores (16) y variando la cantidad de hosts por switch de 1000 a 10 millón.	Ryu y OD-L se destacó, seguido de cerca por ONOS. ODL logra los mejores resultados cuando emulando cuatro o más interruptores mientras que los otros controladores tienen un rendimiento similar.	La comparación cualitativa mostró que tanto ONOS como ODL son los más destacados	3	260

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallázgos concretos)	Conclusiones relevantes	# de citas	# de vistas
13	2016	Controladores SDN: Un estudio comparativo	Ola Salman ; Imad H. Elhajj ; Ayman Kayssi ; Ali Chehab	-	Controladores SDN: un estudio comparativo	Debido a la importancia del controlador hay una necesidad de evaluar y comparar su desempeño.	La comparación considera medidas cuantitativas (rendimiento y latencia), y criterios cualitativos. Con el fin de facilitar la elección del controlador apropiado para una aplicación específica dominio..	Cbench en el modo de rendimiento, envía tantos paquetes como sea posible para calcular el número máximo de paquetes manejados por el controlador. En el modo de latencia, envía un paquete y espera la respuesta para calcular el tiempo tomado para procesar un solo paquete por el controlador.	Los resultados muestran que los controladores codificados por el lenguaje C dieron el más alto rendimiento, seguidos por los codificados de Java.	OpenDaylight es una buena opción como controlador con todas las funciones. Su integración de los datos de IoT intermediario y nuevas interfaces específicas de IoT.	23	1966

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
14	2016	Investigación y análisis comparativo de la prueba de rendimiento en el controlador SDN	Fan Yamei ; Liao Qing ; He Qi	protocolos, redes definidas por software, gestión de redes de telecomunicaciones, topología de redes de telecomunicaciones	Investigación y análisis comparativo de la prueba de rendimiento en el controlador SDN	En la arquitectura SDN, el controlador puede cambiar la construcción y el servicio de la red. El controlador SDN es crucial para toda la red.	OpenDaylight y ONOS tienen una amplia aplicación y una arquitectura más razonable, por lo que son seleccionados para su evaluación.	La máquina de prueba que seleccionamos es de cinco PC. Cada PC es un modelo de doble núcleo y CPU Inter_CDC_1017U_1.6G, sincronizado a 1.6GHz, 2G de memoria. Los sistemas operativos se utilizan Ubuntu	Los resultados muestran que ONOS tiene un buen rendimiento en GUI, clústeres, conexión, conmutación y rendimiento. OpenDaylight tiene un mejor rendimiento en términos de descubrimiento de topología y estabilidad.	7	619	

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallázgos concretos)	Conclusiones relevantes	# de citas	# de vistas
15	2016	Una medida de los tiempos de respuesta de varios controladores OpenFlow SDN con CBench	C. Laissaoui; N. Idboufker; R. Ellassali; K. El Baamrani	Redes definidas por software	Una Medida De Los Tiempos De Respuesta De Varios Controladores Openflow / Sdn Con Cbench	La mayoría de las preocupaciones sobre esta nueva tendencia de redes: ¿qué tan rápido puede el controlador responder a solicitudes de ruta de datos y cuántas solicitudes de ruta de datos puede un controlador manejar por segundo.	Se realizo un estudio y evaluación de Floodlight, Beacon, Pox y Ryu en términos de latencia y rendimiento.	Para la herramienta Cbench elegimos ejecutarlo encima de Ubuntu 12.04 de 32 bits en una máquina virtual con 512 MB de RAM. Las pruebas se realizaron en los controladores en latencia y modos de rendimiento de Cbench con diferentes números de conmutadores.	En general, Floodlight es el mejor controlador para usar al elegir entre él y Beacon, Ryu o Pox en una red que usa OpenFlow 1.0.	Las diferencias en los resultados del rendimiento o podría ser el lenguaje de programación, los intercambiados durante cada comunicación de solicitud / respuesta y también los tamaños de memoria asignado para los almacenamientos.	4	411

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
16	2015	Evaluación del rendimiento del controlador SDN de OpenDaylight	Zuhran Khan Khattak, Muhammad Awais and Adnan Iqbal	centros informáticos, evaluación del rendimiento de la red informática, redes definidas por software, control de telecomunicaciones, tráfico de telecomunicaciones, máquinas virtuales	Evaluación del rendimiento del controlador SDN de OpenDaylight	Aunque existen varios estudios, OpenDaylight no ha sido considerado en ningún estudio de evaluación comparativa todavía.	Los autores presentan el esquema conocido como ECHO que contiene modelos de cadena de Markov distribución y se pueden utilizar para generar modelos de tráfico que pueden capturar las características en función del espacio y el tiempo.	Con el fin de probar los controladores SDN, se utilizó un total de 5 PC (HP Compaq 6300 PC de sobremesa) todo procesador Intel Core i5 tener. Todos los PCs estaban conectados entre sí a través de un interruptor. Uno de estos ordenadores se dedicó a ejecutar el controlador y en todos los demás, nos encontramos Cbench.	Hemos encontrado que la evaluación comparativa de la EAD con Cbench no es muy exitosa. Parece que ODL tiene varios problemas tales como fugas de memoria.	Debido al gran número de pruebas insatisfactorias tenemos la intención de llevar a cabo código de profiling de la EAD. También hemos presentado una mejorada Cbench y su implementación prototipo.	29	1858

N <sup>o</sup>	Año	Título	Autores	Términos controlados por INSPEC	Tema que aborda	Problema que enfrentó	¿Que se hizo?	¿Como se hizo? (El método que uso y las técnicas que uso)	¿Que resultados obtuvo? (Hallazgos concretos)	Conclusiones relevantes	# de citas	# de vistas
17	2015	Sobre redes definidas por software y el diseño de Controladores SDN	Doan B. Hoang ; Minh Pham	toma de decisiones, interfaces de red, redes definidas por software	Software Defined Networking-(SDN) se ha convertido en un paradigma de redes que puede eliminar las limitaciones de las infraestructuras de red actuales.	Internet ha llegado a un punto en que es muy difícil de explorar nuevas arquitecturas que son más adecuados para aplicaciones emergentes	En particular, la revisión de esta investigación se basa en la comparación del diseño de los tres controladores: Beacon, OpenDaylight y Open Networking Operation System.	En este artículo, la arquitectura SDN genérica es descrito, arquitectura del controlador SDN, y su estructura y componentes centrales se examinan..	La interfaz hacia el sur es extendido para admitir otras API, protocolos de complementos y Servicios distintos de OpenFlow para la gestión completa de dispositivos de red, como el servicio de nombres de dominio (DNS), (DHCP), OVSDB,NETCONF	A medida que SDN gana más y más adopción, los requisitos para los controladores SDN son: alta escalabilidad, disponibilidad y seguridad	7	692

## VI. REFERENCIAS

- SDN controllers: A comparative study. In Electrotechnical Conference (MELECON), 2016 18th Mediterranean (pp. 1-6). IEEE.
- A. Shalimov, D. Z. (2013). "Advanced study of SDN/OpenFlow controllers," . *IEEE Xplore*.
- A. Tootoonchian, S. G. (2012). On Controller Performance in Software-Defined Networks. *IEEE Xplore*.
- Alejandro García Centeno, C. M. (2014). Controladores SDN, elementos para su selección y evaluación. *Revista Telemática. Vol. 13. No. 3., 11.*
- Aliyu Lawal Aliyu, P. B. (2017). Performance Implication and Analysis of the OpenFlow SDN Protocol. *IEEE Xplore*, 6.
- All, O. N. (1 de Junio de 2014). SDN architecture. Palo Alto,, Bayshore Road.
- Anh Nguyen-Ngoc, S. R.-G. (2018). Benchmarking the ONOS Controller with OFCProbe. *IEEE Xplore*, 6.
- Ashwin Rajaratnam, R. K. (2017). Software Defined Networks: Comparative Analysis of Topologies with ONOS. *IEEE Xplore*, 5.
- C. Laissaoui, N. I. (2016). A measurement of the response times of various OpenFlow/SDN controllers with CBench. *IEEE Xplore*, 2.
- C.Fancy, M. (2018). Performance Evaluation of SDN controllers POX and Floodlight in Mininet Emulation Environment. *IEEE Xplore*, 5.
- CHAHLAOUI, F., & DAHMOUNI, H. (2018). Towards QoS-enabled SDN networks. *IEEE Xplore*, 7.
- Chehab, H, O. S., & Ali, E. A. (2016). SDN Controllers: A Comparative Study. *IEEE Xplore*, 6.
- Cisco. (2019). *Reporte sobre tendencias globales en redes 2020*.
- D. Erickson. (2013). "The beacon openflow controller," . *ACM SIGCOMM* .
- Darianian, M., Williamson, C., & Haque, I. (2017). Experimental Evaluation of Two OpenFlow Controllers. *IEEE Xplore*, 6.
- Echeverry, M. R. (2018). Redes de datos definidas por software - SDN, arquitectura, componentes y funcionamiento. *Journal de Ciencia e Ingenier'ia, vol. 10, no. 1, 7.*
- ENIO KALJIC, A. M. (2019). A Survey on Data Plane Flexibility and Programmability in Software-Defined Networking. *IEEE Xplore*, 37.
- Foundation, O. N. (27 de Marzo de 2014). OpenFlow Switch Specification.
- Gonçalo Pereira, J. S., & Sousa, P. (2019). Estudio Comparativo de Controladores Software-Defined Networking (SDN). *IEEE Xplore*, 6.

- Hoang, B., D., & Minh, P. (2015). On Software-defined networking and the design of SDN controllers. *IEEE Xplore*, 3.
- Javier Ortiz, J. L. (2016). Evaluation of performance and scalability of Mininet in scenarios with large data centers. *IEEE Xplore*, 6.
- José Manuel Sanchez Vilchez, D. E. (2018). Fault Tolerance Comparison of ONOS and OpenDaylight SDN Controllers. *IEEE Xplore*, 6.
- Kaur, K., Kaur, S., & Gupta, V. (2018). Performance Analysis of Python based OpenFlow controllers. *IEEE Xplore*, 4.
- Khattak, Z. K., & Iqbal, M. A. (2015). Performance Evaluation of OpenDaylight SDN Controller. *IEEE Xplore*, 6.
- LAASSIRI, F., MOUGHIT, M., & IDBOUFKER, N. (2018). Evaluation of the QoS parameters in different SDN architecture using Omnet 4.6++. *IEEE Xplore*, 6.
- Lusani Mamushiane, A. L. (2018). A Comparative Evaluation of the Performance of Popular SDN Controllers. *IEEE Xplore*, 6.
- M. P. Fernandez. (2013). "Comparing openflow controller paradigms scalability: Reactive and proactive," . *AINA*.
- Madhukrishna Priyadarsini, P. B. (2018). Performance Analysis of Software Defined Network Controller Architecture A Simulation Based Survey. *IEEE Xplore*, 7.
- Mamadou T.BAH, A. M. (2019). Topology Discovery Performance Evaluation of OpenDaylight and ONOS Controllers. *IEEE Xplore*, 7.
- MOHAMMED ALSAEEDI, M. M.-R. (2019). Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey. *IEEE Xplore*, 34.
- Pedro Bispo, D. C. (2017). A Qualitative and Quantitative assessment of SDN. *IEEE Xplore*, 6.
- R. Khondoker, A. Z. (2014). Feature-based comparison and selection of Software Defined Networking (SDN) controllers. *IEEE Xplore*.
- Ragaharini Jawaharan, P. M. (2018). Empirical Evaluation of SDN Controllers using Mininet/Wireshark and Comparison with Cbench. *IEEE Xplore*, 2.
- Ramón Parra Loera, V. M. (2015). Redes Definidas por Software: beneficios y riesgos de su implementación en Universidades. *CONAIC*, 7.
- Rastogi, A., & Bais, A. (2017). Comparative Analysis of Software Defined Networking (SDN) Controllers – In Terms of Traffic Handling Capabilities . *IEEE Xplore*, 6.
- S. A. Shah, J. F. (2013). "An architectural evaluation of SDN controllers," . *IEEE Xplore*.
- S. Rowshanrad, V. A. (2016). "Performance evaluation of SDN controllers: Floodlight and OpenDayLight," . *IIUM Engineering Journal*.
- Salman, O. E. (2016). SDN. *IEEE Xplore*.

Seungwoon Lee, J. A.-h. (2019). Performance Comparison of Software Defined Networking Simulators for Tactical Network : Mininet vs. OPNET. *IEEE Xplore*, 6.

Shozi, T., Dlamini, S., & O., P. M. (2019). An SDN Solution for Performance Improvement in Dedicated Wide-Area Networks. *IEEE Xplore*, 6.

Yamei1, F., Qing1, L., & Qi, H. (2016). Research and Comparative Analysis of Performance Test on SDN Controller. *IEEE Xplore*, 4.

Yimeng Zhao, L. I. (2015). On the Performance of SDN Controllers:A Reality Check. *IEEE Xplore*, 7.

Zhengxin Guo, Y. H. (2015). AN IMPLEMENTATION OFMULTI-DOMAIN SOFTWARE DEFINED NETWORKING. *IEEE Xplore*, 5.

Open Networking Foundation. Disponible en: <https://www.opennetworking.org>

Open Network Foundation. “SDN Architecture Overview”. Diciembre, 2013.

Disponible en: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf>

Open Network Foundation. “OpenFlow Switch Specification”. Marzo, 2015.

Disponible en: <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>

Open Network Foundation. “OpenFlow Switch Specification versión 1.0”. Diciembre, 2011.

Disponible en: <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.2.pdf> 74

List of SDN controller software. Disponible en: [https://en.wikipedia.org/wiki/List\\_of\\_SDN\\_controller\\_software](https://en.wikipedia.org/wiki/List_of_SDN_controller_software)

“What are SDN Northbound APIs,” Sdxcentral, [Online]

“Open Network Operating System (ONOS),” [Online].

Disponible en: <https://wiki.onosproject.org/display/ONOS/Wiki+Home>.

“Opendaylight,” [Online]. Disponible en: <https://www.opendaylight.org/>.

Wireshark. “Tshark”.

Disponible en: [https://www.wireshark.org/docs/wsug\\_html\\_chunked/AppToolstshark.html](https://www.wireshark.org/docs/wsug_html_chunked/AppToolstshark.html)