

PROPUESTA METODOLÓGICA PARA LA SELECCIÓN DE CONTROLADORES
DE REDES SDN A NIVEL EMPRESARIAL

ING. CARLOS JULIO QUIMBAYO RODRÍGUEZ

Trabajo de grado

TUTOR:

Ing. GUSTAVO ALONSO CHICA PEDRAZA, MSc.

UNIVERSIDAD SANTO TOMAS

FACULTAD DE INGENIERÍA DE TELECOMUNICACIONES

PROGRAMA DE MAESTRÍA EN TELECOMUNICACIONES Y REGULACIÓN TIC

BOGOTÁ D.C.

2020

NOTA DE ACEPTACIÓN

DIRECTOR

JURADO

JURADO

Bogotá 2020

AGRADECIMIENTOS

Agradezco a las personas que me apoyaron en el proceso de realizar este proyecto, entre los cuales se encuentran mis profesores de la Maestría en Ingeniería en Telecomunicaciones, sin los cuales no habría conseguido la información adecuada para lograr este suceso, infinitas gracias por su esfuerzo y dedicación para lograr una mejor Colombia a través de la educación.

También agradezco a mis compañeros de estudio, los cuales fueron siempre un apoyo y una voz de aliento en los momentos de incertidumbre, de los cuales conservo los mejores recuerdos y la amistad sincera de muchos.

Igualmente agradezco a mi familia, la cual he sacrificado en tiempo y en dedicación, sin embargo, siempre han sido pacientes y me han apoyado en los proyectos en los que me embarco; en especial a mi madre, la cual siempre ha sido mi fuente de inspiración y a mi hija, que la siempre ha sido la fuerza de voluntad que me lleva a lograr mis objetivos.

Por último, pero no menos importante, agradezco a mi tutor, Ingeniero Gustavo Alonso Chica Pedraza, infinitas gracias por su dedicación y esfuerzo para guiarme en el desarrollo del proyecto.

CONTENIDO

ACRONIMOS	VII
LISTA DE FIGURAS	X
LISTA DE TABLAS	0
RESUMEN	1
INTRODUCCIÓN	2
1. MARCO GENERAL DEL PROYECTO	4
1.1 PROBLEMÁTICA:	4
1.2 OBJETIVOS.....	7
1.2.1 <i>Objetivo General</i>	7
1.2.2 <i>Objetivos Específicos:</i>	8
1.3 ALCANCE.....	8
1.4 METODOLOGÍA.....	9
2. ESTADO ACTUAL DE LAS REDES DEFINIDAS POR SOFTWARE	13
2.1 LAS REDES DEFINIDAS POR SOFTWARE	13
2.1.1 <i>Empresas líderes en el desarrollo de las redes definidas por software</i>	18
2.1.2 <i>Actualidad de las redes definidas por software</i>	21
2.1.3 <i>Caracterización de las redes SDN a nivel empresarial</i>	22
2.2 CAPAS DE LA ARQUITECTURA DE REDES DEFINIDAS POR SOFTWARE.	26
2.2.1 <i>Plano de datos o de infraestructura</i>	27
2.2.2 <i>API's hacia el sur:</i>	28
2.2.3 <i>Plano de control</i>	29
2.2.4 <i>API's hacia el norte</i>	29
2.2.5 <i>Plano de aplicaciones:</i>	30
2.3 LOS CONTROLADORES SDN	31
2.3.1 <i>Controladores sin página Web</i>	36
2.3.2 <i>Controladores inactivos</i>	39
2.3.3 <i>Controladores sin actualización desde 2017</i>	43
2.3.4 <i>Controladores con menor información sobre sus características</i>	46
2.3.5 <i>Controladores con mayor información</i>	49

3.	DETERMINAR CUÁL SERIA EL MEJOR CONTROLADOR PARA REDES SDN A NIVEL EMPRESARIAL	56
3.1	LA TOMA DE DECISIONES.....	56
3.2	LA TOMA DE DECISIONES MULTICRITERIO.....	58
3.2.1	<i>Procesos Analíticos Jerárquicos (PAJ)</i>	58
3.3	DETERMINAR CUÁL ES EL CONTROLADOR PARA REDES SDN QUE MEJOR SE ADAPTE A LAS NECESIDADES DE UNA RED EMPRESARIAL	66
3.3.1	<i>Determinación del objeto de estudio</i>	66
3.3.2	<i>Elección de expertos</i>	67
3.3.3	<i>Elección de las alternativas</i>	68
3.3.4	<i>Elección de los criterios de decisión</i>	71
3.3.5	<i>Análisis de dominancia</i>	77
3.3.6	<i>Ponderación de los criterios</i>	80
3.3.7	<i>Cálculo de la prioridad global en el conjunto de alternativas</i>	84
3.3.8	<i>Análisis de resultados e informe final</i>	87
4.	ANÁLISIS DE SENSIBILIDAD A LA SELECCIÓN REALIZADA BAJO EL SOFTWARE EXPERT CHOICE.....	90
4.1	PROCESO DE ASIGNACIÓN DE PRIORIDADES GLOBALES MEDIANTE EL SOFTWARE EXPERT CHOICE.....	90
4.2	SENSIBILIDAD A LA VARIACIÓN MENOR AL 10%.....	94
4.3	SENSIBILIDAD A LA VARIACIÓN MAYOR AL 10%.....	97
4.4	COMPARACIÓN DIRECTA ENTRE CONTROLADORES.....	99
5.	PROPUESTA METODOLÓGICA	104
5.1	PASOS DE LA METODOLOGÍA.....	104
5.1.1	<i>Elección de expertos</i>	105
5.1.2	<i>Ponderación de los criterios</i>	106
5.1.3	<i>Cálculo de la prioridad global en el conjunto de alternativas</i>	107
5.1.4	<i>Análisis de resultados</i>	107
6.	CONCLUSIONES Y RECOMENDACIONES.....	108
6.1	CONCLUSIONES	108
6.2	RECOMENDACIONES	110
	BIBLIOGRAFÍA	112
7.	ANEXOS	122
	ANEXO A	122

7.1	<i>Manual de Uso Archivo de Propuesta Metodológica</i>	122
7.1.1	<i>Conocimientos necesarios</i>	122
7.1.2	<i>Pasos a seguir</i>	123
7.1.3	<i>Análisis de resultados</i>	126

ACRONIMOS

ACL: Access Control List

AHP: Analytic Hierarchy Process

ANP: Analytic Network Process

API: Application Programming Interface

ARCCN: Applied Research Center for Computer Networks

ARP: Address Resolution Protocol

BGP: Border Gateway Protocol

BSD: Berkeley Software Distribution

CAPWAP: Control and Provisioning for Wireless Access Point

Cisco ONE: Open Network Environment

Cisco XNC: Extensible Network Controller

CLI: Command Line Interface

COPS: Common Open Policy Service

CPD: Centros de Procesos de Datos

DHCP: Dynamic Host Configuration Protocol

DOVE: Distributed Overlay Virtual Ethernet

EPL: Eclipse Public License

ETRI: Electronics and Telecommunications Research Institute

FML: Fuzzy Markup Language

ForCES: Forwarding and Control Element Separation

GAN: Global Area Network

GPL: General Public License

GUI: Graphical User Interface

HP SDK: HP SDN Developer Kit

HP VAN SC: HP Virtual Application Networks SDN Controller

HP: Hewlett Packard

IBM PNC: IBM Programmable Network Controller

IEEE: Institute of Electrical and Electronics Engineers

IoT: Internet of Things

IS-IS: Intermediate System to intermediate System

LAN: Local Area Network

LGPL: Lesser General Public License

LISP: Locator ID Separation Protocol

LLDP: Link Layer Discovery Protocol

MAN: Metropolitan Area Network

MAUT: Multi Attribute Utility Theory

MCDM: Multiple Criteria Decision Making

MPLS: Multiprotocol Label Switching

NAT: Network Address Translation

NFV: Network Function Virtualization

NIB: Network Information Base

NV: Network Virtualization

ON Lab: Open Networking Laboratory

ONE PK: ONE Platform KIT

ONF: Open Networking Foundation

OSGI: Open Service Gateway Initiative

OSPF: Open Shortest Path First

OVN: Open Virtual Network)

OVS: Open Virtual switch

OVSDB: Open Virtual Switch Database Management Protocol

PAJ: Proceso Analítico Jerárquico

PAN: Personal Area Network

PCE: Path Computation Element

QoS: Quality of Service

REST: Representational State Transfer

RFC: Request for Comments

RSVP: Resource Reservation Protocol

SDN: Software Defined Networking

SNMP: Simple Network Management Protocol

SPF: Sender Policy Framework

STP: Spanning Tree Protocol

SXP: Scalable Group Tag eXchange Protocol

TCP: Transmission Control Protocol

TLS: Transport Layer Security

TTL: Time to Life

UDP: User Datagram Protocol

VPN: Virtual Private Network

WAN: Wide Area Network

XMPP: Extensible Messaging and Presence Protocol

LISTA DE FIGURAS

Figura 1 Comparación red tradicional vs red SDN	6
Figura 2 Diagrama de Gantt cronograma de trabajo	12
Figura 3 SDN primera versión.....	16
Figura 4 Arquitectura SDN.....	26
Figura 5 Arquitectura SDN completa.....	31
Figura 6 Jerarquía modelo PAJ	60
Figura 7 Respuesta sobre categorías de criterios de selección	71
Figura 8 Consideraciones de los expertos sobre aspectos de caracterización.....	72
Figura 9 Características relevantes para seleccionar un controlador SDN.....	73
Figura 10 Criterios para seleccionar el controlador SDN.....	76
Figura 11 Modelo jerárquico de Expert Choice	91
Figura 12 Vectores de prioridad Expert Choice.....	92
Figura 13 Calculo de la prioridad global por Expert Choice (método distribuido)	93
Figura 14 Calculo de la prioridad global por Expert Choice (método ideal)	93
Figura 15 Análisis de rendimiento (modo distribuido).....	94
Figura 16 Análisis de sensibilidad dinámico del resultado (modo distribuido)	95
Figura 17 Análisis de sensibilidad dinámico de la Categoría Técnico (+/- 10%).....	95
Figura 18 Análisis de sensibilidad dinámico de la Categoría Económico (+/- 10%).....	96
Figura 19 Análisis de sensibilidad dinámico de la Categoría Infraestructura (+/- 10%)	96
Figura 20 Análisis de sensibilidad dinámico de la Categoría Técnico (+/- 50%).....	97
Figura 21 Análisis de sensibilidad dinámico de la Categoría Económico (+/- 50%).....	98
Figura 22 Análisis de sensibilidad dinámico de la Categoría Infraestructura (+/- 50%)	98
Figura 23 Comparación directa controlador OpenDaylight y Floodlight.....	100
Figura 24 Comparación directa controlador OpenDaylight y ONOS.....	101
Figura 25 Comparación directa controlador Floodlight y ONOS.....	102
Figura 26 Comparación directa ONOS, Big Network Controller y Huawei SDN Agile Controller	102
Figura 27 Comparación directa Huawei SDN Agile Controller, Cisco Open SDN Controller y HP SDN VAN Controller	103
Figura 28 Diagrama de flujo de la metodología.....	105
Figura 29 Formulario de valoración de prioridad de criterios de decisión.....	106

Figura 30 Hoja Excel con valor de prioridad global de las alternativas.....	107
Figura 31 Archivo metodología SDN – hoja de cálculo Datos	124
Figura 32 Botones Experto 1,2 y 3.....	124
Figura 33 Formulario Experto 1	125
Figura 34 Error al ingresar datos	126
Figura 35 Resultados propuesta metodológica	126
Figura 36 Gráfico de torta – resultados.....	127

LISTA DE TABLAS

Tabla 1 Cronograma de trabajo	12
Tabla 2 Caracterización de las redes SDN empresariales	26
Tabla 3 Características de los controladores de Redes Definidas por Software.....	54
Tabla 4 Escala de valoración de Saaty	61
Tabla 5 Matriz de Comparación de Criterios	62
Tabla 6 Matrices de Comparación de Criterios en las Alternativas.....	63
Tabla 7 Índice de consistencia aleatorio	64
Tabla 8 Tabla de información criterios vs alternativas	65
Tabla 9 Caracterización de la red empresarial usada como ejemplo	67
Tabla 10 Alternativas de Controladores SDN.....	69
Tabla 11 Criterios técnicos.....	74
Tabla 12 Criterios económicos.....	75
Tabla 13 Criterios de infraestructura	75
Tabla 14 Valoración de las Características de las Alternativas	78
Tabla 15 Ponderación Criterios Padres (nivel 2)	81
Tabla 16 Ponderación Criterios Económicos (nivel 2.1)	81
Tabla 17 Ponderación Criterios de Infraestructura (nivel 2.1).....	81
Tabla 18 Ponderación Criterios Técnicos (Nivel 2.1)	82
Tabla 19 Ponderación Servicios controlador (nivel 2.2)	82
Tabla 20 Ponderación Conexiones controlador (nivel 2.2)	83
Tabla 21 Ponderación Vigencia controlador (nivel 2.2)	83
Tabla 22 Ponderación Características técnicas controlador (nivel 2.2)	83
Tabla 23 Ponderación Desarrollo del controlador (nivel 2.2)	84
Tabla 24 Ratio de Inconsistencia de las matrices evaluadas	84
Tabla 25 Calculo prioridad global (método distributivo).....	85
Tabla 26 Calculo prioridad global (método ideal)	86
Tabla 27 Resultados prioridad global (método distribuido).....	88
Tabla 28 Resultados prioridad global (método ideal)	88
Tabla 29 Consolidado de análisis de sensibilidad (Modo distribuido 10%).....	97
Tabla 30 Consolidado de análisis de sensibilidad (Modo distribuido >10%).....	99
Tabla 31 Escala de valoración de Saaty	123

RESUMEN

El presente trabajo de investigación tiene como propósito realizar un análisis, que permita determinar cuál es el controlador de Redes Definidas por Software que mejor se adapte a las necesidades específicas de una red empresarial, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura. Usando para ello, herramientas del área de toma de decisiones multicriterio, específicamente las desarrolladas en la aplicación del Proceso de Analítico Jerárquico (PAJ), el cual contempla el uso de valoraciones tanto cuantitativas como cualitativas, permitiendo el uso de la experiencia de expertos en el tema, de manera adicional, permite realizar un análisis de sensibilidad mediante el software especializado Expert Choice para confirmar la robustez de la elección realizada.

El desarrollo de este documento aborda de manera general el estado del arte de las redes definidas por software (SDN), las partes de su arquitectura y las empresas líderes en el desarrollo de la misma; enfocándose principalmente en la capa de control, donde se realiza especial énfasis en la búsqueda de información sobre los controladores desarrollados para redes SDN, sus características técnicas, fabricantes, páginas WEB, entre otras propiedades. Encontrando 35 controladores desarrollados por fabricantes, universidades y organizaciones no gubernamentales, algunos de los cuales son prototipos y otros fueron desarrollados para satisfacer necesidades específicas de esta nueva arquitectura, caracterizando cada uno de ellos a través de 23 variables. Destacándose por sus características los controladores Floodlight, HP SDN VAN Controller, ONOS, OpenDaylight, Big Network Controller, Cisco Open SDN Controller y Huawei SDN Agile Controller.

Luego, aborda el proceso de elección del controlador de redes SDN que mejor se adapte a las necesidades específicas de una red empresarial ejemplo, desarrollando el método PAJ sobre las 7 alternativas de controladores SDN que cuentan con las mejores características, para luego realizar un análisis de sensibilidad de esta elección usando el software Expert Choice y al final del documento, entrega una propuesta metodológica para facilitar la realización de esta elección.

Palabra claves

Multicriterio, Proceso Analítico Jerárquico, Redes definidas por Software, Controladores SDN.

INTRODUCCIÓN

Internet se ha convertido con el pasar de los años en el camino natural de toda la información que fluye en el mundo, creando la necesidad de transportar cada vez mayor cantidad de información con una calidad de servicio aceptable, tanto para satisfacer los servicios actuales como los planteados a futuro con tecnologías como internet de las cosas. Para lograr esto, se requiere de una gran cantidad de equipos de interconexión en la infraestructura de los operadores de servicio al igual que en otras grandes empresas; estos equipos de interconexión requieren ser configurados, administrados y trabajar coordinadamente entre sí, para lograr un buen funcionamiento y transportar paquetes de información de una fuente a un destino en el menor tiempo posible. Pero lograr esto para los administradores de redes es cada vez más complicado por la cantidad creciente de equipos a gestionar, además cada equipo de transmisión es en sí una caja negra, donde el fabricante mediante un software propietario realiza las funciones de enrutar, reenviar o descartar paquetes de información, y en donde realizar un cambio o necesitar encontrar la fuente de un error en la red, se vuelve una tarea dispendiosa, demorada y costosa entre otros inconvenientes; teniendo en cuenta esta problemática, se han presentado nuevos desarrollos para ayudar a los administradores de las redes en la gestión de los equipos de interconexión. [1]

Desde los años 90's, se han venido desarrollando diversas soluciones, entre las cuales se destacan las Redes Definidas por Software o redes SDN (Software Defined Networks), que plantean una nueva arquitectura de red, pero al ser una tecnología relativamente nueva, que aún se encuentra en desarrollo y que promueve un cambio radical en las redes, se requiere de un esfuerzo para encontrar información de su funcionamiento y aun mayor para entenderlo; lo anterior explica la complejidad al desarrollar aplicaciones y servicios para esta nueva arquitectura de red que permitan visualizar sus ventajas, por lo cual, muchos de los actuales servicios desarrollados para la misma, son adaptaciones de servicios y protocolos actuales, que no explotan todas las capacidades de esta nueva tecnología. Como caso excepcional esta Google, que desarrollo sus propios algoritmos y aplicaciones para esta arquitectura, con base en sus necesidades específicas y que le permitieron un gran avance en el enrutamiento de su red interna de transporte de información [2].

Las redes SDN se han desarrollado de forma abierta, teniendo avances en todas sus capas a través de universidades, organizaciones no gubernamentales y proveedores de equipos.

De forma especial en la capa de control, se han desarrollado diversos proyectos de controladores con diferentes características, lenguajes de desarrollo, sistemas operativos de operación, entre otros; y ante ello, emerge la dificultad de decidir cuál es el controlador que mejor se adapta a las necesidades de una red específica.

Es por ello que este proyecto pretende desarrollar un método analítico, que entregue como solución, cual es el controlador que mejor se adapta a las necesidades específicas de una red empresarial; usando para ello, el estudio y la caracterización de la mayor cantidad de controladores para redes SDN, y para su selección se plantea el uso de la herramienta de toma de decisión PAJ (Proceso Analítico Jerárquico), de modo que se pueda realizar una selección objetiva, ordenada y sustentada con base en las necesidades de la red, las características de los controladores y la priorización que tengan los administradores de red.

Este documento se encuentra organizado de la siguiente manera; en el capítulo uno, se contextualiza el trabajo mediante el planteamiento del problema, la descripción del título, los objetivos y el alcance proyectado. El capítulo dos, muestra el estado actual de las redes definidas por software a nivel general, el desarrollo de esta arquitectura a través del tiempo y sus partes, además de la caracterización de las redes SDN empresariales y un estudio comparativo entre los diversos controladores de redes SDN encontrados en la literatura sobre esta arquitectura. En el capítulo tres, se desarrolla la metodología de toma de decisiones PAJ, para seleccionar el controlador que mejor se adapte a las necesidades específicas de una red ejemplo. En el capítulo cuatro, se realiza un análisis de sensibilidad mediante el software Expert Choice, a la selección del controlador realizada en el capítulo tres. En el capítulo cinco, se realiza una propuesta metodológica que permita a una empresa seleccionar el controlador que mejor se ajuste a sus necesidades. Finalmente, en el capítulo seis, se describen las conclusiones del trabajo y algunas recomendaciones generales.

1. MARCO GENERAL DEL PROYECTO

En este capítulo se presentan los parámetros generales del proyecto desarrollado. En la sección 1.1, se presenta la problemática encontrada, la cual inspiró el desarrollo de este trabajo. En la sección 1.2, se presenta el objetivo general de este proyecto y los objetivos específicos del mismo. En la sección 1.3, se presenta el alcance del proyecto.

1.1 PROBLEMÁTICA:

En la actualidad, Internet se ha convertido en el camino natural de toda la información y comunicación que fluye a través del mundo, conectando cada vez a más personas a través de aplicaciones y/o servicios, como las redes sociales, aplicaciones de chat y/o videojuegos en línea, entre otros; de igual manera sirve como enlace para la comunicación entre personas y computadores, brindando acceso a la información alojada en ellos; a través de buscadores como Google o Yahoo! se puede acceder a bases de datos o a servicios como tiendas en línea, se puede usar servicios de geo-localización para encontrar la ruta más adecuada a través del tráfico matutino o simplemente para encontrar el video musical del momento.

Como se puede notar, nuestro diario vivir está rodeado de aplicaciones y/o servicios en red que nos brindan diversas comodidades, muchas de las cuales requieren el transporte de información de manera casi instantánea para poder cumplir su función, sin importar el lugar donde se encuentre el usuario, siempre y cuando tenga una conexión a internet habilitada. Pero lograr satisfacer la necesidad creciente de más y mejores conexiones de internet para los usuarios, requiere de una gran cantidad de equipos de interconexión en la infraestructura de los operadores de servicio, los cuales requieren ser configurados, administrados y trabajar coordinadamente entre sí, para lograr un buen funcionamiento de la red y que a su vez puedan cumplir su labor de forma eficiente; la cual es transportar la mayor cantidad de información entre un equipo fuente y un equipo destino, sin pérdidas de paquetes y en el menor tiempo posible [3].

Lograr lo anterior para los encargados de las redes, es cada vez más complicado por la cantidad creciente de equipos a gestionar, a ello se le debe sumar, problemas propios de las redes actuales, que en un principio fueron concebidas para un funcionamiento cliente -

servidor, donde a través de un computador, el usuario o cliente solicitaba información que se encontraba en un computador de grandes capacidades llamado servidor; estas primeras redes de interconexión son conocidas como redes estáticas, porque su arquitectura es muy rígida y permite pocos cambios en su configuración, ya que para poder realizar un cambio en las mismas, se debe realizar la reconfiguración de manera individual de todos los equipos involucrados. Otro inconveniente es que la actualización del Firmware de los equipos, está más ligado a la consideración de los fabricantes que a las necesidades del negocio y de los operadores de servicio, debido a que este software es de tipo propietario al igual que los programas de gestión; de manera adicional, las mejoras realizadas sobre las redes para poder ofrecer servicios y aplicaciones a los usuarios en la actualidad, ha requerido el desarrollo de múltiples protocolos, los cuales realizan funciones específicas y en muchos casos no pueden explotar todas las capacidades de los equipos, al generar incompatibilidades por no estar desarrolladas junto al Hardware y/o al firmware del fabricante del equipo. Por los problemas anteriormente mencionados entre otros, se han desarrollado nuevas tecnologías para ayudar a la gestión de administradores y operadores de redes, y a la red misma, a estar acorde a las necesidades de las aplicaciones y/o servicios actuales y en desarrollo [4].

Entre las nuevas tecnologías para mejorar la gestión y el funcionamiento de las redes de datos, se han planteado las Redes Definidas por Software, también llamadas redes SDN (Software Defined Networks), que más que una nueva tecnología es un nuevo tipo de arquitectura de red, que permite poder gestionar redes con gran cantidad de equipos y de distintos fabricantes de una manera centralizada, organizada y eficiente, dividiendo la red en tres capas o planos:

- Un plano de datos, que es en si el hardware del equipo (Switches, Routers, etc), al cual se le quitan funciones y se especializa su diseño en el envío o descarte de paquetes de información.
- Un plano de control centralizado para toda la red, que se encarga de la gestión de los equipos de interconexión y el enrutamiento de los paquetes de información, concentrando las directivas de funcionamiento de toda la red entregadas por el plano de aplicaciones, para luego enviarlas a el plano de datos de forma consistente; además almacena estadísticas de funcionamiento, comportamiento y fallos del

plano de datos, enviando esta información a la tercera capa o plano de aplicaciones, donde son analizadas y usadas como insumo para producir las directivas de la red.

- Un plano de aplicaciones, donde con base en la información y estadísticas recibidas desde el plano de control, se toman las decisiones por parte de las aplicaciones, las cuales entregan las directivas y servicios a la red; por ejemplo, control de acceso, enrutamiento eficiente, calidad de servicio, seguridad en la red, control de fallas, entre otros. Estas decisiones son enviadas a la capa de control, que centraliza esta información y la distribuye a la capa de datos.
- Para realizar la comunicación entre planos se utilizan programas de nombre API's (Interfaz de Programación de Aplicaciones) y protocolos abiertos, entre ellos el más conocido es Openflow.

En la Figura 1, se puede observar una comparación entre la arquitectura de la red tradicional y la arquitectura de la red SDN.

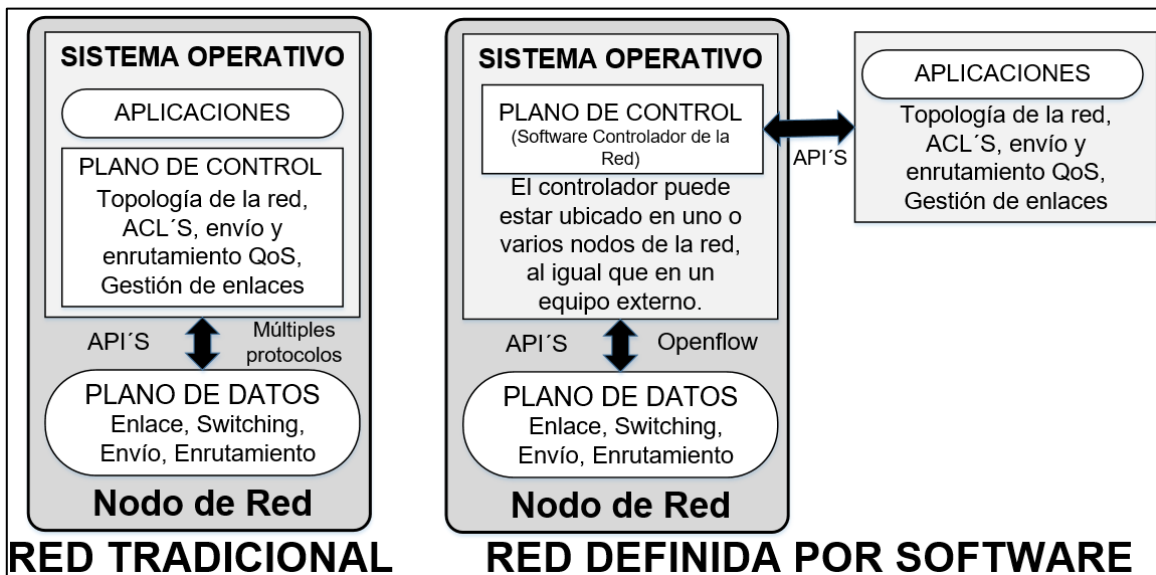


Figura 1 Comparación red tradicional vs red SDN [basado en [5]]

La parte primordial de esta nueva arquitectura de red es la capacidad de gestión y control centralizada, que permite de manera rápida y programada poder realizar cambios, como

configurar puertos y realizar el enrutamiento de paquetes mejorando el desempeño de la red, igualmente permite modificar servicios y aplicaciones, entre otras funciones, todo esto basado en las estadísticas de rendimiento y funcionamiento de la red misma. De manera adicional, el uso de protocolos de comunicación abiertos, permite el desarrollo de aplicaciones y/o mejoras por parte de los mismos proveedores de servicio y de los administradores de red, de tal manera que se pueden solucionar problemas específicos, sin necesidad de tener que esperar el lanzamiento de una mejora en el firmware por parte del fabricante de los equipos.

Al momento de realizar investigación, implementación y/o gestión de las redes SDN, se necesita conocer sus partes y cuáles son las variaciones de las mismas, de forma tal, que se use eficientemente cada herramienta disponible; en otras palabras, se debe tener conocimiento sobre los controladores de red SDN, los protocolos de comunicación usados y de las aplicaciones ya diseñadas, determinando sus capacidades y seleccionando la que mejor se adapte a la red en la que la queremos implementar [6].

Con base en lo anterior, este proyecto pretende enseñar el funcionamiento de esta nueva arquitectura de red, enfocándose en la capa de control, más específicamente, realizando un proceso de selección del controlador que mejor se adapte a las condiciones de una red en particular, evidenciando como con el uso de la teoría de toma de decisiones, se puede realizar una selección de manera ordenada, objetiva y sustentada, basada en las necesidades de la red, las características de los controladores y la priorización que tengan los administradores de red.

1.2 OBJETIVOS

1.2.1 Objetivo General

Plantear una propuesta metodológica para la selección de controladores SDN a nivel Empresarial, utilizando el proceso analítico jerárquico, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura.

1.2.2 Objetivos Específicos:

- 1** Identificar los criterios y variables que deben ser tenidos en cuenta para evaluar los Controladores de red utilizados en la arquitectura SDN, desde el punto de vista técnico, económico y de infraestructura a través de un estudio comparativo.
- 2** Seleccionar el mejor controlador para redes SDN a nivel empresarial usando como ejemplo una red real, teniendo en cuenta los criterios y/o variables identificados en el objetivo 1, mediante la aplicación del Proceso analítico Jerárquico.
- 3** Validar la selección del controlador a través de un análisis de sensibilidad utilizando el Software Expert Choice.
- 4** Plantear una propuesta metodológica para la selección de controladores SDN a nivel Empresarial utilizando el proceso analítico jerárquico, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura. basado en los objetivos 2 y 3.

1.3 ALCANCE

En este proyecto se pretende entregar un documento donde se explique el funcionamiento de las redes definidas por software, sus características y ventajas sobre las redes de transmisión usadas en la actualidad, esto a través el estado del arte realizado sobre esta nueva arquitectura de red.

Luego, el trabajo se enfoca en la capa de control de la arquitectura SDN, entregando información sobre las características de los controladores de red SDN encontrados en la literatura. Después, usando la herramienta de toma de decisiones PAJ, realizar una selección del controlador que mejor se adapte a las necesidades de una red empresarial tomada como ejemplo.

1.4 METODOLOGÍA

La metodología prevista para abordar este trabajo, que pretende plantear una propuesta metodológica para la selección de controladores SDN a nivel Empresarial utilizando el proceso analítico jerárquico, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura. Es el siguiente:

- **Etapa 1.** Identificación de la bibliografía para abordar el objeto de estudio: Se hará una revisión de fuentes bibliográficas en bases de datos con la finalidad de encontrar y seleccionar documentación a nivel académico, que estén relacionados con la Arquitectura de redes definidas por software. Para realizar esta etapa se requieren las siguientes actividades:
 - 1.1 Identificación de la documentación en literatura científica y académica, relacionada con las redes definidas por software.
 - 1.2 Realizar una priorización de lectura de la documentación, y realizar un resumen de las características de la arquitectura de redes definidas por software.
 - 1.3 Documentar la etapa 1.
- **Etapa 2.** Identificación de bibliográfica para identificar y caracterizar los controladores de las redes definidas por software: Se hará una revisión de fuentes bibliográficas en bases de datos con el fin de encontrar y seleccionar documentación a nivel académico y técnico, que estén relacionados con los controladores de las redes definidas por software. Para realizar esta etapa se requieren las siguientes actividades:
 - 2.1 Identificación de la documentación en literatura científica, académica y técnica relacionada con los diversos controladores de redes definidas por software.
 - 2.2 Identificar y caracterizar los controladores de las redes definidas por software.

2.3 Realizar una comparación entre las características de cada uno de los controladores encontrados.

2.4 Documentar la etapa 2.

- **Etapa 3.** Validar la importancia de cada una de las características de los controladores: Realizar una validación de la caracterización de los controladores y sus pesos respectivos encontrados en la etapa 2, mediante una consulta a expertos en el tema de las redes definidas por software.

3.1 Realizar una selección de expertos en redes definidas por software, basado en sus perfiles, conocimiento de las redes definidas por software e interés de colaborar en el ejercicio.

3.2 Realizar una selección de preguntas y realizar las encuestas a los expertos seleccionados.

3.3 Evaluación de los resultados.

3.4 Documentar la etapa 3.

- **Etapa 4.** Validar las respuestas de los expertos con el programa Expert Choice: Validar las características y pesos de los controladores obtenidos de los expertos en la etapa 3, mediante el uso del programa especializado Expert Choice.

4.1 Identificar los niveles jerárquicos de las características de los controladores de redes definidas por software.

4.2 Identificar los pesos de las características de los controladores de redes definidas por software.

4.3 Realizar los métodos de normalización necesarios.

4.4 Realizar la validación de la información con el programa Expert Choice.

4.5 Documentar la etapa 4.

- **Etapa 5.** Selección del controlador para la red empresarial ejemplo: Realizar la selección del controlador que mejor se adapte a la red empresarial ejemplo, basado en la información de las etapas 3 y 4.

5.1 Realizar una caracterización de las necesidades de la red empresarial ejemplo.

5.2 Realizar la selección del controlador de red SDN que mejor se ajuste a las necesidades de la red empresarial ejemplo.

5.3 Documentar la etapa 5.

- **Etapa 6.** Realizar la consolidación de la información: Realizar la consolidación de la información obtenida en todas las etapas de proceso, presentando el modelo de toma de decisiones analítico jerárquico como una solución, para seleccionar el controlador de redes definidas por software que mejor funcione en una red empresarial, dependiendo de sus necesidades.

Las etapas realizadas no implican necesariamente un desarrollo secuencial de las tareas estrictamente hablando, algunas de ellas se podrán realizar de forma paralela, lo que dependerá del estado de avance y la forma como se sobrelleven los posibles problemas que puedan aparecer en el desarrollo del proyecto.

En la Tabla 1, se observa el cronograma detallado de cada una de las actividades realizadas a lo largo del trabajo de grado, indicando el objetivo específico que satisfacen, el tiempo estimado de duración y las fechas en las que plantea realizarlas.

	etapa	Nombre de tarea	Duración	Comienzo	Fin
1. Identificar los criterios y variables que deben ser tenidos en cuenta para evaluar los Controladores de red utilizados en la arquitectura SDN, desde el punto de vista técnico, económico y de infraestructura a través de un estudio comparativo.	Etapa 1. Identificación de la bibliografía para abordar el objeto de estudio:	1.1 Identificación de la documentación sobre redes SDN.	262 días	01/07/17	01/07/18
		1.2 Priorización y lectura de la documentación.	197 días	01/10/17	01/07/18
		1.3 Documentación etapa 1.	174 días	01/11/17	02/07/18
	Etapa 2. Identificación de bibliografía para identificar y caracterizar los controladores de las redes definidas por software:	2.1 Identificación de la documentación sobre controladores SDN.	436 días	01/01/18	31/08/19
		2.2 Identificación y caracterización de los controladores SDN.	413 días	01/02/18	31/08/19
		2.3 Comparación de las características de los controladores SDN.	413 días	01/02/18	31/08/19
		2.4 Documentación etapa 2.	393 días	01/03/18	31/08/19
2. Seleccionar el mejor controlador para redes SDN a nivel empresarial usando como ejemplo una red real, teniendo en cuenta los criterios y/o variables identificados en el objetivo 1, mediante la aplicación del Proceso analítico Jerárquico.	Etapa 3. Validar la importancia de cada una de las características de los controladores	3.1 Realizar una selección de expertos en redes SDN.	43 días	01/08/19	30/09/19
		3.2 Seleccionar las preguntas y realizar las encuestas a los expertos en redes SDN.	45 días	01/09/19	31/10/19
		3.3 Evaluar y documentar los resultados de la etapa 3.	44 días	15/09/19	13/11/19

3. Validar la selección del controlador a través de un análisis de sensibilidad utilizando el Software Expert Choice.	Etapa 4. Validar las respuestas de los expertos con el programa Expert Choice	4.1 Identificar los niveles de las características de los controladores SDN.	42 días	06/10/19	30/11/19
		4.2 Identificar los pesos de las características de los controladores SDN.	51 días	06/10/19	13/12/19
		4.3 Normalizar	46 días	13/10/19	13/12/19
		4.4 Validar la información con el programa Expert Choice.	58 días	13/10/19	31/12/19
		4.5 Documentar la etapa 4.	69 días	13/10/19	15/01/20
4. Plantear una propuesta metodológica para la selección de controladores SDN a nivel Empresarial utilizando el proceso analítico jerárquico, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura. basado en los objetivos 2 y 3.	Etapa 5. Selección del controlador para la red empresarial ejemplo	5.1 Caracterizar las necesidades de la red ejemplo.	67 días	01/01/20	02/04/20
		5.2 realizar la selección del controlador para la red ejemplo.	65 días	01/02/20	30/04/20
		5.3 documentar la etapa 5	65 días	01/03/20	28/05/20
	Etapa 6. Realizar la consolidación de la información	6.1 consolidar la información	98 días	01/04/20	14/08/20

Tabla 1 Cronograma de trabajo [Elaboración propia]

En la Figura 2, se observa el diagrama de Gantt de las actividades explicadas en el cronograma de trabajo (Tabla 1), observando como el desarrollo de las mismas va en función del avance del trabajo de grado, de manera adicional se observa que algunas actividades fueron realizadas de manera paralela y otras de manera secuencial, ya que requerían un desarrollo previo.

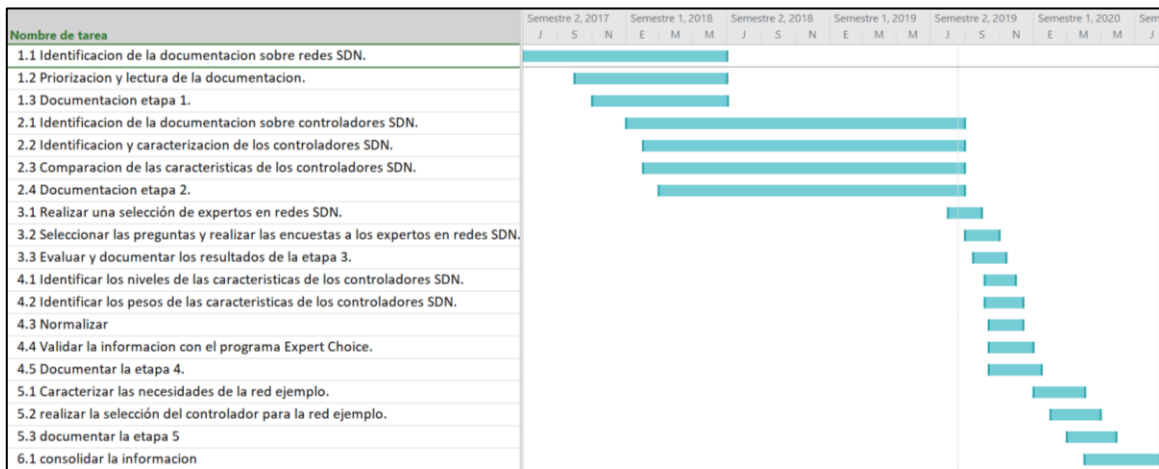


Figura 2 Diagrama de Gantt cronograma de trabajo [Elaboración propia]

2. ESTADO ACTUAL DE LAS REDES DEFINIDAS POR SOFTWARE

En este capítulo se presenta información sobre el estado actual de las redes definidas por software. En la sección 2.1, se presenta información sobre los inicios y desarrollo de esta arquitectura en el tiempo. En la sección 2.2, se presenta información sobre las partes en las que está dividida esta arquitectura de red. En la sección 2.3, se presenta información sobre los controladores de la red SDN y una tabla comparativa de sus características.

2.1 LAS REDES DEFINIDAS POR SOFTWARE

En un principio las redes de transmisión de paquetes de información, que terminarían convirtiéndose en la red de redes, más conocida como internet, se diseñaron para conexiones cliente-servidor, en otras palabras, la comunicación se realizaba entre un equipo de cómputo llamado cliente, que solicitaba información a un equipo de cómputo más potente llamado servidor en donde esta se encontraba alojada; este servidor respondía las solicitudes a medida que éstas iban llegando, enviándole la información solicitada a cada equipo cliente. Con este tipo de funcionamiento, las redes se enfocaban en satisfacer la necesidad de descarga de información por parte de los clientes, los cuales a su vez enviaban muy poca información hacia el servidor; este sistema funcionó adecuadamente hasta el desarrollo de los microprocesadores y la posterior masificación de los computadores personales en la década de los 80's, que llevó a un aumento considerable en la cantidad de conexiones entre los proveedores de servicio, las oficinas y/o los hogares de los usuarios. Con esta situación comenzaron a hacerse evidentes algunos inconvenientes en el funcionamiento de este tipo de redes, como su baja escalabilidad, el aumento de los tiempos de gestión y de configuración de manera directamente proporcional al aumento del tamaño de las mismas; así como inconvenientes adicionales dados por su arquitectura cerrada, como el tener actualizaciones de software y de funcionalidades dependientes de los fabricantes de los equipos, compleja innovación de servicios, entre otros. En respuesta a esto, se comenzó a pensar en el desarrollo de redes más fácilmente administrables, más fácilmente actualizables y cuyas funcionalidades no dependieran de la voluntad del fabricante de los equipos; dando como solución, el concepto de las redes definidas por software [7].

Los inicios de las redes definidas por software datan de los años 90's, con el proyecto de Redes Activas (Active Networking), cuyo concepto básicamente consistía en enviar dentro del tráfico de información, paquetes con micro-programas que debían ser extraídos y ejecutados por los switches, routers y demás equipos de interconexión en general, de esta manera se podía reconfigurar la red de una manera más rápida y eficiente, sin tener que gastar tiempo y otros recursos realizando las configuraciones equipo por equipo; pero esta idea fue perdiendo fuerza sobre todo por consideraciones de seguridad, ya que un atacante podría introducir programas maliciosos dentro de la red y hacerlos ejecutar afectando su funcionamiento.

El siguiente intento por hacer unas redes más fácilmente administrables, fueron las redes programables (Programmable Networks) en 1995, donde al igual que en las redes activas, se intentaba reconfigurar la red de forma más rápida mediante programas insertados, pero esta vez, los programas serían instalados directamente sobre cada nodo o equipo de interconexión de la red, y serían ejecutados dependiendo de los paquetes de información que estuvieran siendo procesados, de esta manera, se evitaban muchos inconvenientes de seguridad, pero era igualmente tedioso realizar las configuraciones previas y la inserción de los programas dentro de cada nodo [8].

El siguiente concepto y más cercano a las redes definidas por software, tiene su comienzo en un proyecto conjunto de las Universidades de Stanford y Berkeley, de nombre Ethane en el año 2006, donde los ingenieros Nick MacKeown y Scott Shenker, pretendían de manera general, facilitar la configuración de las redes de transmisión ethernet, en las cuales observaron, que para poder realizar cualquier cambio se debía realizar la configuración equipo por equipo, lo cual es una tarea muy dispendiosa y demorada, entonces para resolver este problema, conceptuaron que se podía utilizar un controlador central para toda la red y allí configurar todas las tareas necesarias de acceso, enrutamiento, seguridad, etc; este controlador se encargaría mediante un canal de comunicación seguro, de configurar todos los equipos de interconexión de la red, de esta manera toda la red seguiría las mismas políticas, minimizando la posibilidad de errores de configuración, de manera adicional, se podrían realizar cambios de forma más práctica, rápida e incluso pre-programados [4].

El problema principal de las redes de transmisión convencionales, que se pretendía solucionar con Ethane, es que los equipos utilizados en las redes de transmisión (Switches, Routers, Etc), son unas cajas negras para los administradores de las redes, donde los

fabricantes mediante un software propietario (Firmware), definen las políticas de procesamiento y envío de paquetes; la falta de conocimiento sobre esta programación del equipo, crea una arquitectura cerrada y poco receptiva al cambio, por lo cual, cualquier innovación depende de los mismos fabricantes de los equipos. En el mismo sentido, la transmisión de paquetes entre equipos se realiza mediante múltiples protocolos, cada uno diseñado para satisfacer una necesidad específica, junto con sus respectivas normalizaciones, existen aproximadamente 5500 RFC's creadas para controlar y ofrecer una funcionalidad tolerable en internet; a esto se le debe sumar, que los procesos realizados a cada paquete de información deben rehacerse en cada uno de los nodos de transmisión, lo que causa un retraso en la llegada de los paquetes a su destino y dificulta la posibilidad de ofrecer servicios como calidad de servicio de extremo a extremo [9].

De manera adicional, la programación cerrada de los equipos de transmisión, causa que los programas utilizados para realizar la gestión y administración de la red, solo puedan ser desarrollados por los mismos fabricantes de los equipos, y al ser los programas de gestión un factor diferenciador a la hora de elegir que marca de equipos se deben comprar, existe un programa casi por cada fabricante; razón por la cual, no existe la posibilidad de gestionar toda una red desde un solo software, si está compuesta por equipos de distintos fabricantes como sucede en la mayoría de los casos. Sumado a lo anterior, la mayoría de los programas de gestión son solo programas a través de los cuales se puede visualizar errores de la red, dando la posibilidad de realizar cambios equipo por equipo a través de canales de control dentro de banda, dado lo anterior, es imposible poder realizar la gestión de la red en tiempo real, lo cual vuelve la red poco dinámica y muy compleja al momento de necesitar realizar algún cambio; por ejemplo, al agregar equipos nuevos, configurar nuevos servicios o aplicaciones, gestionar caídas de enlaces o problemas de encolamiento, entre otros. De manera general, el desarrollo de mejoras sobre el software de gestión y el firmware de los equipos, está ligado más al deseo de los fabricantes que a la necesidad de los usuarios y del negocio [10].

En la Figura 3, se puede observar una versión inicial de la arquitectura SDN, donde se puede ver que los nodos de la red están conectados a un sistema operativo de red (controlador centralizado), el cual tiene una visión global del comportamiento de la red, permitiéndole tomar decisiones sobre la configuración de la misma, estas decisiones son informadas a los nodos a través de los canales seguros de comunicación.

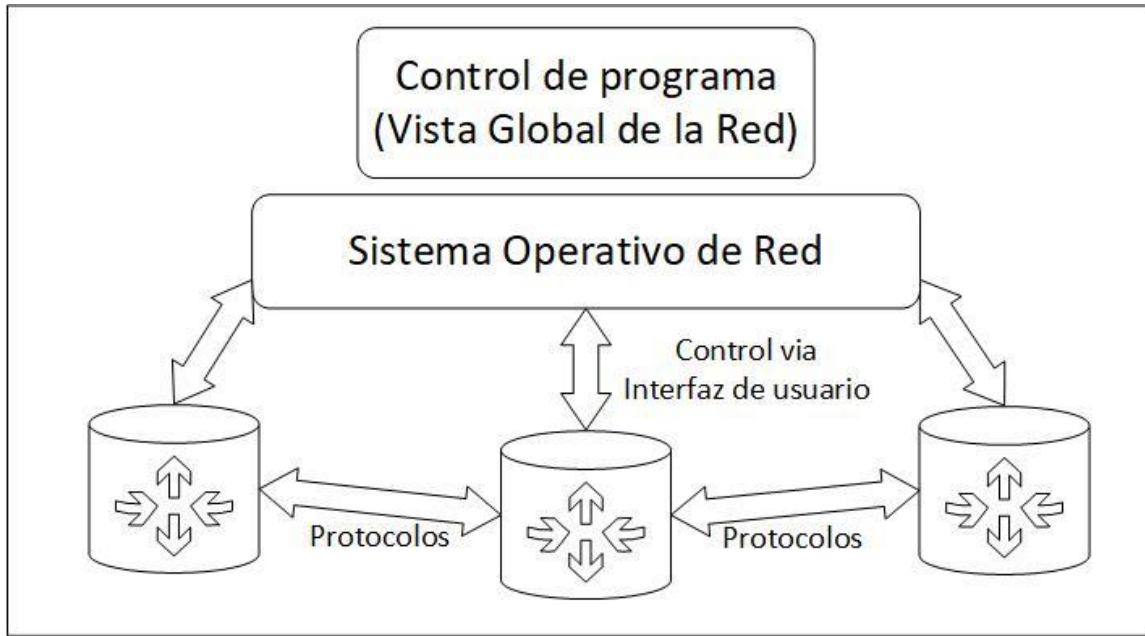


Figura 3 SDN primera versión [basado en [5]]

Continuando con el proceso de creación de la nueva arquitectura, en el año 2007, un grupo de trabajo de la universidad de Stanford, conformado por los ingenieros Nick MacKeown, Scott Shenker y el estudiante de doctorado Martin Casado, desarrollaron el protocolo Openflow y fundaron la empresa Nicira, dedicada a la virtualización de redes; este protocolo se convertiría en el eje fundamental de las redes definidas por software, tanto así, que algunas veces se confunde el protocolo, con el concepto de la nueva arquitectura de red, usándose como sinónimos.

La evolución del concepto de Ethane son las Redes definidas por software (SDN), en el año 2008 se presentó formalmente esta nueva arquitectura de red, que la divide en tres planos: un plano de datos, un plano de control y un plano de aplicaciones, comunicados entre sí por protocolos abiertos; esta arquitectura pretende facilitar la gestión de las redes de transmisión, además de acelerar su desarrollo y mejorar su funcionamiento [11].

Los principales fabricantes de equipos de transmisión como Hewlett Packard, Cisco, Huawei, entre otros; han visto las bondades y posibilidades de esta nueva arquitectura de red, de tal manera, que han desarrollado sus nuevos equipos con soporte para Openflow y en general para la arquitectura SDN. Algunos de estos equipos pueden trabajar de forma híbrida, es decir, pueden trabajar siguiendo los lineamientos de la arquitectura tradicional o bajo la

arquitectura SDN; de manera adicional, existen algunos equipos que permiten trabajar con las dos arquitecturas al mismo tiempo, seleccionando la arquitectura a usarse en cada puerto del equipo de transmisión. A continuación, algunos hitos en la evolución de esta arquitectura.

En el año 2011, se creó la Open Networking Foundation (ONF), por las empresas Deutsche Telekom, Facebook, Google, Microsoft, Verizon, and Yahoo; esta fundación tiene como objetivo principal, fortalecer de forma colaborativa con la estandarización, entendimiento y distribución de las Redes definidas por Software en el mundo, definiendo Openflow como el protocolo principal de la arquitectura. Actualmente la fundación cuenta con 113 empresas colaboradoras, entre fabricantes, proveedores de servicio y desarrolladoras de aplicaciones [12].

En el año 2012, se creó la fundación Floodlight, para el fortalecimiento y desarrollo del controlador para redes SDN de código abierto del mismo nombre; en el mismo año, la principal empresa dedicada a virtualización de servidores en el mundo (VMWare) compro la empresa Nicira [13]. En el año 2013, se creó la fundación OpenDaylight, para fomentar el desarrollo del controlador de para redes SDN del mismo nombre, contando con 25 grupos distribuidos alrededor del mundo y algo más de 600 desarrolladores trabajando en la evolución de este controlador de código abierto. Ambas organizaciones cuentan con documentación, blogs y soporte para quienes deseen trabajar con sus controladores [14].

En el año 2014, Hewlett Packard lanzo la primera tienda de aplicaciones para redes definidas por software, allí se pueden encontrar aplicaciones diseñadas por los propios desarrolladores de Hewlett Packard y también algunas desarrolladas por terceros; de esta tienda se pueden descargar versiones gratuitas, demos y versiones de pago de las distintas aplicaciones, entre las cuales se pueden encontrar aplicaciones para optimización, mejoras de seguridad, algunas de gestión y visualización de la red, entre otras [15].

Para poder realizar pruebas con la arquitectura SDN, actualmente existen herramientas de simulación como Mininet, NS-3 y Estinet; estos softwares trabajan sobre Linux, pero permiten su instalación en máquinas virtuales, dando así la posibilidad de trabajar sobre cualquier sistema operativo, sus características permiten el poder diseñar arquitecturas de redes a la medida y de esta manera poder evaluar su rendimiento. Además permiten gestionar la arquitectura diseñada por un controlador externo a elección del usuario o por

uno interno que viene pre-acondicionado con el software, entre otras muchas funciones; estos programas tienen la desventaja de depender de las capacidades de hardware del equipo de cómputo donde se encuentren instalados, dividiendo sus capacidades entre las necesidades del equipo en sí y las necesarias para realizar los cálculos de la máquina virtual [16].

2.1.1 Empresas líderes en el desarrollo de las redes definidas por software

Los líderes en la aplicación, estudio y desarrollo de esta nueva arquitectura de red, son principalmente empresas fabricantes de equipos de interconexión, dedicadas a soluciones de virtualización y/o al desarrollo de sistemas en la nube, entre otros; sus esfuerzos están encaminados a potencializar la arquitectura o a dirigir el avance de la misma, siendo vista esta nueva arquitectura de red, como una ventaja o amenaza para sus propios intereses. A continuación, veremos algunos de estos promotores de la arquitectura de Redes Definidas por Software.

La multinacional **Brocade**, desarrolladora de equipos de interconexión, es uno de los miembros de la Open Networking Foundation (ONF), y como tal, es uno de los pioneros en desarrollar equipos para la arquitectura de Redes Definidas por Software, al igual que en realizar pruebas de funcionamiento del protocolo Openflow, probando entre otras características el funcionamiento del modo híbrido a nivel de puertos físicos [17].

La multinacional **IBM**, como miembro fundador de la organización ONF y del Proyecto para desarrollar el controlador SDN OpenDaylight, también ha implementado en algunos de sus equipos soporte para el protocolo Openflow, al igual ha realizado pruebas de virtualización de redes, desacoplando la red virtual de la red física, además de haber desarrollado su propio controlador de nombre IBM PNC (Programmable Network Controller) [18].

La multinacional **Hewlett Packard**, al ser uno de los miembros de ONF, sigue una arquitectura de red semejante a la definida por esta organización, centrando el desarrollo de su concepto en los servicios y aplicaciones para las Redes Definidas por Software, para lo cual ha desarrollado su propio Controlador, de nombre HP VAN SC (HP Virtual Application Networks SDN Controller), el cual es compatible con Openflow y con algunas API's para realizar la comunicación entre las capas de la arquitectura. De manera adicional,

ha implementado una tienda de aplicaciones similar a Google App's [15], en la cual se pueden encontrar diversas soluciones para Redes Definidas por Software, la mayoría de las cuales han sido diseñadas por desarrolladores de la misma compañía, para lo cual cuentan con la herramienta de desarrollo de uso libre de nombre HP SDK (HP SDN Developer Kit), con la cual se pueden desarrollar aplicaciones para Redes Definidas por Software enfocándose especialmente en su compatibilidad con HP SDN VAN; esta herramienta cuenta con programas API's ya desarrollados, documentación de la herramienta, guías de programación, ejemplos de códigos y un entorno de simulación para realizar pruebas de funcionamiento.

La multinacional **Cisco**, líder en desarrollo e implementación de equipos de interconexión, es uno de los fabricantes más afectados con la implementación de esta nueva arquitectura de red más abierta, por lo tanto ha implementado su propia versión de la misma, ligándola a sus equipos para no perder fuerza en el mercado; esta visión la ha llamado Cisco ONE (Open Network Environment), en la cual ofrece mayor programabilidad de la red y brinda soluciones para Redes Definidas por Software puro al igual que híbrido. También cuenta con API's de comunicación compatibles con equipos de otros fabricantes dentro de la arquitectura SDN, brindando además virtualización de redes y soporte para el protocolo Openflow sobre sus equipos. De manera adicional, ofrece una herramienta gratuita para el desarrollo de aplicaciones para la arquitectura SDN versión Cisco, llamada ONE PK (ONE Platform KIT), igualmente ha desarrollado su propio controlador de red SDN, de nombre Cisco XNC (Extensible Network Controller). Cisco ONE además de soportar intercomunicación e interacción con soluciones de Redes Definidas por Software de otros fabricantes para funciones básicas, ofrece funcionalidades adicionales solo entre equipos de la marca Cisco [19].

La multinacional **VMWare**, es una filial de la empresa de virtualización de servidores EMC, que a su vez hace parte de DELL Incorporated; su campo de acción es la virtualización de servidores a través de máquinas virtuales, ofreciendo en un solo hardware la posibilidad de disponer de decenas de servidores virtuales separados entre sí. Este mismo concepto se puede aplicar a las redes de interconexión y es allí donde nace el concepto de la virtualización de redes (NFV), estableciendo cierta cantidad de redes virtuales separadas entre sí, sobre una capacidad de red física instalada, desligando el plano físico del digital; si a lo anterior, se le suma el concepto de SDN, dando la capacidad de gestión de estas redes virtuales con base en los requerimientos de las aplicaciones, se obtiene una red

virtualizada completamente inteligente. Lo anterior explica por qué esta compañía se encuentra interesada en el desarrollo de esta nueva arquitectura y por qué en el año 2012 compraron la compañía Nicira; empresa fundada por los creadores de Openflow y dedicada a la virtualización de redes. En el año 2013, presentaron su plataforma de virtualización de redes, con un controlador de red de nombre NSX, el cual mediante un switch virtual central, es capaz de interconectar las diferentes redes virtuales creadas, convirtiendo esta plataforma en una capa intermedia entre el Hardware y la capa de aplicaciones, que además soporta programas API's y diferentes protocolos entre ellos Openflow [20].

Como un caso especial de implementación de la arquitectura de Redes Definidas por Software en la industria, se encuentra el proceso de **Google**, al ser el mayor buscador de contenidos de Internet y tener que manejar en sus redes miles de solicitudes de búsqueda de videos, datos, etc. Lo primero que se debe saber, es que la red de Google está conformada por una red externa que maneja el acceso y solicitudes de búsqueda de los usuarios, la cual debe soportar muchos protocolos y peticiones al mismo tiempo; las peticiones son solucionadas a través de Centros de Procesos de Datos (CPD) diseminados alrededor del mundo, los cuales no cuentan con la misma información, es decir, están especializados por temáticas y los que tienen información duplicada deben mantenerla actualizada entre sí. Por esta razón, se necesita que estos CPD puedan comunicarse rápidamente tanto para realizar las búsquedas, como para sincronizar los cambios en la información, para lo cual cuentan con una red de transmisión interna propiedad de Google, por donde circulan las búsquedas, copias de seguridad, protocolos propios, entre otros; para tener un buen funcionamiento esta red interna debe contar con características especiales, como gran capacidad de escalabilidad, alta tolerancia a fallos, además de necesitar control de la red y aprovisionamiento de ancho de banda en tiempo real. Debido a estas consideraciones que las redes tradicionales no ofrecen, Google decidió implementar en su red interna los principios básicos de la arquitectura SDN, sobre protocolos diseñados a la medida por sus ingenieros y con el uso del protocolo Openflow; aumentando la tasa de uso de su infraestructura de entre el 30% y el 40% a un 70% de su capacidad [2].

2.1.2 Actualidad de las redes definidas por software

En el proceso de búsqueda de información sobre las Redes Definidas por Software, se encontró que en la academia, la mayoría de los proyectos desarrollados se enfocan en la adaptación de aplicaciones de la arquitectura convencional a la arquitectura de Redes Definidas por Software, con el fin de poder brindarle funciones similares, como en el caso de enrutamiento, seguridad, control de acceso, gestión y visualización de la red, entre otros; a continuación se nombran algunos de los trabajos encontrados.

En el año 2014, se desarrolló un trabajo de nivel de Maestría, de nombre “Diseño e implementación de una aplicación de red bajo la arquitectura SDN”, en la Pontificia Universidad Javeriana, donde se desarrolló una aplicación de gestión de paquetes UDP según las instrucciones del administrador de la red, sobre una arquitectura implementada con equipos de Hewlett Packard [21].

En el año 2015, se desarrolló un trabajo de grado, de nombre “Diseño e Implementación de una Aplicación de Balanceo de Carga para una Red Definida por Software (SDN)”, en la Escuela Politécnica Nacional de Ecuador, donde se desarrolló una aplicación de balanceo de carga, usando el lenguaje de desarrollo Python y el Controlador de redes RYU, de manera adicional se usó el programa de emulación Mininet para realizar pruebas [22].

En el año 2015, se desarrolló un trabajo de grado de nivel de Maestría, de nombre “Monitorización del plano de datos en redes definidas por software”, en la Universidad Complutense de Madrid, donde se desarrolló una aplicación para visualizar la cantidad de paquetes transmitidos y perdidos en una transmisión emulada sobre Mininet, usando el lenguaje de desarrollo Java y el controlador de red Floodlight [23].

En el año 2016, se desarrolló un trabajo de grado, de nombre “Desarrollo de una Aplicación para la Implementación de Calidad de servicio por Priorización de Trafico sobre una Red Definida por Software (SDN)”, en la Escuela Politécnica Nacional de Ecuador, donde se desarrolló una aplicación de priorización de tráfico basada en el puerto destino y origen del paquete, usando el lenguaje de desarrollo Java y el controlador de red Floodlight [24].

En el año 2016, se desarrolló un trabajo de grado, de nombre “Diseño e Implementación de una Aplicación Prototipo para Ofrecer el Servicio de DHCP Sobre una SDN”, en la Escuela Politécnica Nacional de Ecuador, donde se desarrolló una aplicación DHCP, usando el

lenguaje de desarrollo Python sobre el controlador RYU y el programa de emulación Mininet para realizar las pruebas [25].

Como caso especial, en Colombia, hay un proyecto avalado por Colciencias donde se utilizó la tecnología NetFPGA con el protocolo Openflow para experimentar el tratamiento de paquetes en un router remoto, a través de la Red Nacional Académica de Tecnología Avanzada (RENATA) [4].

Con respecto a trabajos desarrollados para la selección de controladores de redes SDN usando el método PAJ, se encontró en la literatura dos casos. En el año 2014, se desarrolló el trabajo de nombre “Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers”, donde se seleccionó el mejor controlador SDN de código abierto, evaluando los controladores POX, Trema, RYU, OpenDaylight y Floodlight; caracterizándolos a través de 14 variables, determinando que el mejor controlador SDN de código abierto era RYU [26].

En el año 2017, se desarrolló el trabajo de nombre “A Systematic and Generic Method for Choosing A SDN Controller”, donde se evaluaron los controladores de redes SDN ONOS, Opencontrail, OpenDaylight, Floodlight, RYU, OpenMul, POX, Trema, Beacon y NOX; caracterizados a través de 11 variables, determinando que los controladores RYU, OpenDaylight y ONOS contaban con las mejores características, para después realizarles una prueba de rendimiento a través del programa CBENCH, encontrando que el controlador OpenDaylight contaba con las mejores características y rendimiento [27].

2.1.3 Caracterización de las redes SDN a nivel empresarial

Las redes de transmisión de datos están divididas por su nivel de cobertura, el cual es directamente proporcional a la cantidad de equipos que pueden interconectar:

- Las redes PAN (Personal Area Network), se limitan normalmente a unos pocos metros y son usadas para interconectar terminales modernos como Teléfonos inteligentes, tabletas, computadores de escritorio y/o portátiles entre sí, de manera adicional, algunos sensores de internet de las cosas; esta comunicación se realiza mediante cable o por una variante inalámbrica WPAN (Wireless Personal Area

Network) usando tecnologías como Bluetooth, Wireless USB, Insteon, IrDA, ZigBee o Z-Wave.

- Las redes LAN (Local Area Network), están conformadas por más de un computador y su alcance se limita a un área bien definida y de pequeñas proporciones, como ejemplo, una casa, una oficina, un avión o un edificio. Necesitando otros componentes en la red como concentradores, puentes de red y conmutadores, es decir, hubs, bridges y switches; interconectados a través de medios cableados o por una variante inalámbrica WLAN (Wireless Local Area Network) definida por la familia de normas IEEE 802.11. Las redes LAN operan como una sola red de localización, aunque pueden servir a varios usuarios al mismo tiempo.
- Las redes MAN (Metropolitan Area Network), abarcan un área geográfica más extensa que una red LAN y pueden contener dentro de sí, varias redes LAN de una zona geográficamente cercana, comunicando varias sedes de una empresa o un campus universitario por medio de líneas arrendadas o propias; en caso de necesitar conectar sedes en varias ciudades o países, se debe usar la infraestructura de los operadores de servicio. También existe una variante inalámbrica WMAN (Wireless Metropolitan Access Network), que usa la tecnología WiMAX (Worldwide Interoperability for Microwave Access), definida por la norma IEEE 802.16, permitiendo la conexión en grandes áreas geográficas donde es complicado implementar otras infraestructuras.
- Las redes WAN (Wide Area Network) abarcan un área geográfica que se extiende a países o continentes, siendo toda esta infraestructura propiedad de una empresa u operador de servicio, que la gestiona y la alquila de manera privada.
- Las redes GAN (Global Area Network) abarcan un área geográfica global, uniendo varias redes WAN a través de enlaces satelitales o cables de fibra óptica submarinos, siendo su principal exponente internet; siendo toda esta infraestructura propiedad de una empresa u operador de servicio internacional, que la gestiona y la alquila de manera privada.

Dentro de los tipos de redes anteriormente vistas, la arquitectura de redes SDN puede alcanzar todo su potencial en las redes de mayor tamaño, siendo descartadas de inmediato las redes PAN y LAN pequeñas, como las de una casa, una oficina, un avión, o un edificio

pequeño; ya que este tipo de arquitectura no es aconsejable por su complejidad y por no ser eficiente tener un controlador para cada red pequeña, en función de la cantidad de tráfico de datos que se va a manejar [28].

Los tipos de redes donde se evidencia la implementación de la arquitectura de redes SDN, se pueden dividir en redes empresariales y redes de laboratorio, cada una con características específicas:

- Redes SDN de laboratorio: son redes de carácter demostrativo, las cuales pueden estar implementadas en físico o de manera digital desde cero, dependiendo generalmente del tamaño de la misma y la cantidad de recursos disponibles; por ejemplo, una red pequeña o con pocos equipos de transmisión puede ser implementada de manera física, pero una red extensa debe ser implementada de manera digital a través de un simulador, por el costo de implementación de la misma, tanto en tiempo como en recursos. Estas redes son diseñadas con características muy específicas dependiendo de la función o servicio que se desea demostrar o probar; por ejemplo, seguridad, resiliencia, enrutamiento, cuellos de botella, entre otros. Generalmente cuentan con pocos perfiles para su gestión y pocos usuarios, en caso de necesitar pruebas de rendimiento de la red o de un servicio en específico, se usan herramientas como el software Cbench, que permite enviar gran cantidad de tráfico para realizar pruebas de rendimiento, latencia, paquetes perdidos, etc [29].

Usualmente antes de realizar la implementación física de una red SDN empresarial, se realiza una implementación de laboratorio o demostrativa, para visualizar las posibles fallas y problemas que se pueden encontrar en la implementación de la red real.

- Redes SDN empresariales: son redes de producción, las cuales pueden ser cableadas y/o inalámbricas (Wireless SDN), cuya implementación tiene un fuerte componente físico, al tener gran cantidad de equipos distribuidos en diferentes nodos, sin dejar de lado que se pueden usar tecnologías digitales como Virtualización de Funciones de Red (NFV) y almacenamiento en la nube. Usualmente el proyecto de implementación de una red SDN empresarial se desarrolla en fases, donde la implementación de una red SDN de laboratorio de

manera digital o física, juega un papel crucial en momento de detectar fallas. Estas redes son diseñadas en función de satisfacer las necesidades actuales y a futuro de los usuarios y administradores de las redes, entregando la posibilidad de implementar nuevas funciones y servicios, de manera adicional, deben poder correlacionarse con las redes con tecnología anterior o legacy, manteniendo el servicio a sus múltiples usuarios; para lo cual, debe cumplir con múltiples requisitos como estabilidad, seguridad, resiliencia, escalabilidad, gran rendimiento y buena documentación entre otros [30].

Antes de colocar en funcionamiento estas redes, se deben realizar exhaustivas pruebas de funcionamiento; esto a través de equipos certificadores y generadores de tráfico, realizando pruebas de rendimiento, latencia, paquetes perdidos, enrutamiento, seguridad, entre otras, para de esta manera asegurar que los errores dentro de la misma se reducen a su mínima posibilidad.

Dentro de las redes SDN empresariales implementadas, se destaca el caso de la compañía Google, que durante tres años trabajo en la implementación de la arquitectura SDN en su red interna, la cual comunica sus 12 Centros de Datos distribuidos alrededor del mundo. Dentro de las necesidades específicas de esta red, se encuentra que transporta información multimedia y de protocolos propios de búsqueda, a través de múltiples enlaces y cientos de puertos por nodo, con una demanda elástica de ancho de banda en función de picos de trasmisión, transportando del orden de terabytes por segundo, teniendo una tasa de uso de la infraestructura entre el 30% y el 40%, debido a las provisiones por fallo en la red. Teniendo un gran problema en el tema de escalabilidad de la red, debido al incremento de tráfico en función de la cantidad de usuarios y servicios ofrecidos; el cual bajo después de la implementación de la arquitectura SDN, debido a que la tasa de uso de la red aumento en algunos enlaces a cerca del 100% y en general al 70%, pudiendo de esta manera transportar más información a través de menos enlaces y menos equipos [2].

En la Tabla 2, se observa un resumen de las características de las redes SDN de laboratorio y de las redes SDN empresariales.

Tabla 2 Caracterización de las redes SDN empresariales [Elaboración propia]

Características	Red SDN laboratorio	Red SDN empresarial
tamaño	Redes pequeñas y medianas.	Redes grandes
implementación	Simulada o física dependiendo de su tamaño	Física
Uso	Demostrativo o modelo	Producción
Usuarios	Pocos usuarios	Múltiples Usuarios
Requerimientos	Dependen de su uso y de la función o servicio que se desea probar o mostrar.	Estabilidad, escalabilidad, seguridad, buen rendimiento, resiliencia, Buena documentación, entre otros.

2.2 CAPAS DE LA ARQUITECTURA DE REDES DEFINIDAS POR SOFTWARE.

La arquitectura de las redes definidas por software (SDN), como se mencionó anteriormente, divide las funciones de la red de transmisión tradicional en tres capas o planos; una capa de datos o infraestructura, una capa de control o gestión, y una capa de aplicaciones. De manera adicional, para poder realizar la comunicación entre estas tres capas, se utilizan programas llamados API's (Interfaz de Programación de Aplicaciones) o se hace uso de protocolos abiertos como Openflow. La arquitectura SDN se puede ver gráficamente en la Figura 4.

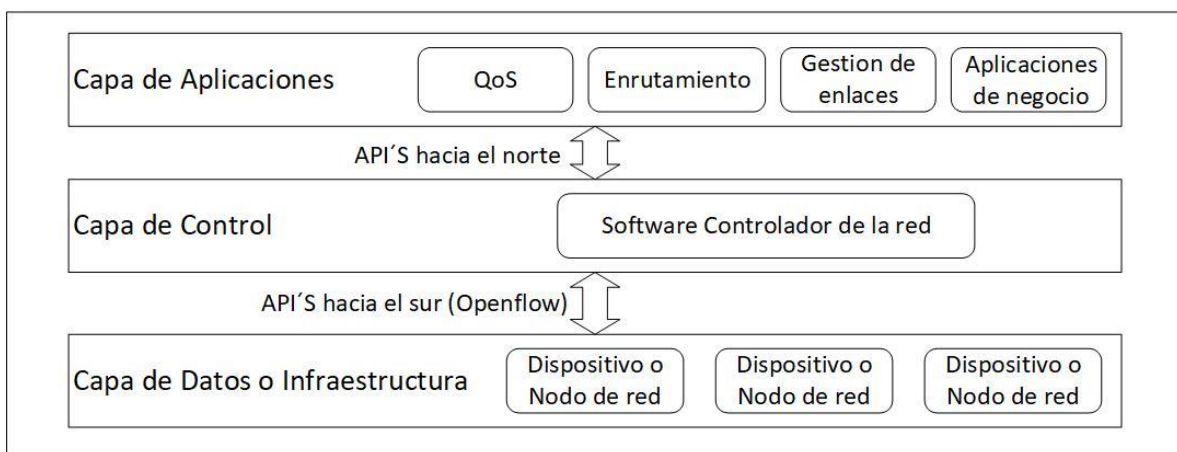


Figura 4 Arquitectura SDN [basado en [31]]

2.2.1 Plano de datos o de infraestructura

Es el hardware o los equipos de transmisión en sí, es decir, switches, routers, etc. A los cuales se les quitan las funciones que tenían en la arquitectura convencional, como el enrutamiento de paquetes, procesos de control de acceso y políticas de seguridad entre otros; enfocando el diseño y desarrollo de los mismos en función de la recepción y reenvío de paquetes a través de flujos, de la forma más eficiente posible. Las instrucciones sobre el enrutamiento de paquetes y las políticas de comportamiento de la red son dadas desde la capa de control a cada equipo de la misma, esto a través de protocolos abiertos, de los cuales el más conocido y distribuido es Openflow. Dejando en los equipos solo una pequeña capacidad de hardware para realizar procesamiento y almacenamiento de información, en este espacio se aloja un pequeño software API's (Interface de Programación de Aplicaciones), encargado de procesar las instrucciones de la capa de control; además de alojar las tablas de flujo, en donde se almacenan las instrucciones que indican que proceso seguir con cada paquete de información que llega al equipo, indicándole si un paquete de información con una característica específica; por ejemplo, su dirección IP destino, dirección IP fuente, si es TCP o UDP entre otras, debe ser rechazado o reenviado, o si debe ser tratado de forma especial por hacer parte de un servicio en tiempo real; como por ejemplo, una video llamada o una aplicación específica que necesita calidad de servicio, de esta manera, el equipo sabe por cual flujo de paquetes, cola y/o puerto específico debe salir cada paquete para llegar a su destino. Además de permitirle a la red realizar algunos otros procesos dependiendo de las aplicaciones instaladas; por ejemplo, la eliminación de paquetes que provienen de direcciones IP antes catalogadas como maliciosas, en caso que se tenga una aplicación de seguridad con esta función instalada en la capa de aplicaciones [32].

Algunos fabricantes de equipos han sostenido la capacidad de procesamiento y almacenamiento en algunos modelos, permitiendo de esta manera, que los equipos puedan funcionar bajo la arquitectura convencional y/o bajo la arquitectura de redes definidas por software, estos equipos son llamados equipos híbridos; esto con el fin de evitar que la interrupción en la comunicación entre el equipo de transmisión y el controlador de la red (cuando se encuentren trabajando bajo la arquitectura de redes definidas por software), pueda generar un error global en la red y desencadene en una excesiva pérdida de paquetes, pudiendo retroceder a la arquitectura convencional en caso de ser necesario; de manera adicional, algunos fabricantes como HP, Huawei, NEC, IBM, entre otros, han

incluido en algunos modelos la capacidad de elegir cuales puertos físicos en específico deben trabajar bajo una u otra arquitectura, funcionando las dos arquitecturas al mismo tiempo dentro de un mismo equipo.

2.2.2 API's hacia el sur:

“Consiste en un protocolo o interfaz de programación bien definido entre los elementos de la capa de datos y la capa de control” [16], siendo Openflow el protocolo más utilizado, aunque también existen otros como ForCES, Netconf, PCEP.

Openflow es un protocolo de código abierto, cuya versión más reciente es la 1.5.1 [33], la cual puede ser encontrada en la página WEB de la fundación ONF (Open Networking Foundation). Este protocolo permite la comunicación bidireccional entre el plano de datos y el plano de control, en sus primeros años se desarrolló como un proyecto conjunto entre la Universidad de Stanford y la Universidad de Berkeley, pero actualmente su desarrollo y estandarización está a cargo de la fundación ONF.

Para establecer y sostener la comunicación entre el controlador y el equipo de transmisión, Openflow utiliza protocolos de transporte estandarizados como TCP y UDP, entre otros; debido a que no tiene desarrollada esta función dentro de sí mismo. También usa el protocolo de encriptamiento TSL, para establecer la seguridad en los canales de comunicación, mediante estos canales seguros se llenan las tablas de flujos, definiendo que acciones tomar con los paquetes de información que ingresan al equipo por alguno de sus puertos; bien sea, reenviarlos en un flujo preestablecido a su destino, encapsularlo con destino al controlador para que defina qué acciones tomar con paquetes con sus mismas características o eliminarlo en caso de algún ataque a la red u otro acontecimiento. Igualmente, estos enlaces sirven como canales de transporte de estadísticas y fallas desde el equipo de transmisión hacia el controlador, permitiéndole conocer en tiempo real inconvenientes en la red; por ejemplo, la saturación de paquetes en los puertos, colas o flujos, al igual que desconexiones entre equipos o caída de puertos, etc. De esta forma las aplicaciones instaladas en el plano de aplicaciones, pueden conocer el funcionamiento de la red en tiempo real, tomando decisiones y realizando reconfiguraciones para que funcione más eficientemente [4].

2.2.3 Plano de control

Esta parte de la arquitectura es un software llamado controlador, el cual actúa como un sistema operativo centralizado para toda la red, concentrando las decisiones sobre el comportamiento, enrutamiento y los cambios que deben realizarse en los equipos y en los flujos de datos; estas decisiones son tomadas por los servicios instalados en la capa de aplicación, los cuales le brindan sus funcionalidades a la red. Algunos controladores ya tienen preinstaladas algunas aplicaciones básicas; por ejemplo, el protocolo de descubrimiento, enrutamiento, direccionamiento IP, entre otros [34].

Entre las ventajas de tener un plano de control centralizado, se encuentran el poder enviar instrucciones de forma automática y consistente a toda la red, basándose en su comportamiento en tiempo real, evitando inconvenientes por errores de digitación, instrucciones en conflicto y de manera adicional permite realizar cambios de forma más rápida; de manera adicional, abstrae la complejidad de la capa de datos permitiendo el desarrollo de aplicaciones con más facilidad, al no tener que preocuparse de las capas inferiores. El controlador también concentra la información que envían los equipos de la capa de datos, registrando estadísticas, eventos y problemas de rendimiento entre otras informaciones relacionadas con el comportamiento de la red; entregando la información pertinente a cada aplicación a través de API's, que sirven como canales de comunicación entre la capa de aplicaciones y la capa de control. Estos datos son necesarios para que las aplicaciones puedan tener una visión global de la red, tomar decisiones y solicitar cambios en tiempo real en la configuración de la red; por ejemplo, en caso que un equipo pierda la conexión con otro equipo o se encuentre sobrecargada su memoria de encolamiento, una aplicación de enrutamiento puede decidir cambiar la ruta de los flujos de datos que pasan por este equipo, para evitar la pérdida de paquetes [31]. Existen diversos desarrollos de controladores, entre ellos están HP VAN SDN, Beacon, NOX, OpenDaylight, Floodlight, NOX, POX, entre otros.

2.2.4 API's hacia el norte

Son un conjunto de programas API's, usualmente desarrollados junto con las aplicaciones, usando el lenguaje RESTful (Transferencia de Estado Representacional), aunque también existen otros lenguajes de programación como FML, Frenetic, Procera, entre otros; los

cuales son seleccionados dependiendo de la necesidad específica de cada aplicación. Estas API's sirven de canal de comunicación bidireccional entre el plano de aplicaciones y el plano de control, en un proceso donde, después que el controlador expone todos sus servicios a las aplicaciones, es decir, los datos almacenados de la red (estadísticas, eventos, rendimiento, etc), las aplicaciones solicitan las estadísticas y reportes específicos de eventos necesarios para su funcionamiento; una vez realizado el procesamiento de la información de la red, las aplicaciones entregan al controlador las políticas de funcionamiento para ser distribuidas entre la capa de datos, después de esto, continúan censando el comportamiento de la red, basado en la información que les llega a través de estos mismos canales [35].

2.2.5 Plano de aplicaciones:

En esta parte de la arquitectura de red, están alojados los servicios brindados mediante aplicaciones, las cuales son un conjunto de interfaces o programas, ya que la definición de aplicaciones en Redes Definidas por Software es muy amplia, cubriendo desde servicios de red, como QoS, control de movilidad, control de acceso, monitoreo de tráfico y seguridad, hasta aplicaciones propias del negocio, es decir, desarrolladas por los usuarios para satisfacer sus necesidades propias; como en el caso de la empresa Iskratel, que desarrollo una aplicación que ofrece servicios dinámicos por sesión, para el aprovisionamiento a abonados residenciales y comerciales dentro de su propia red SDN [36].

En la Figura 5, se puede observar el funcionamiento de la arquitectura de redes definidas por software en su conjunto, en la parte inferior se observa la capa de datos, donde se hace referencia a algunos equipos que la componen; esta capa se encuentra conectada a través de canales de comunicación con la capa de control, estos canales pueden ser enlaces a través del protocolo Openflow, un protocolo similar o alguna API desarrollada para tal fin. En la capa de control, se puede observar algunos ejemplos de programas que han sido diseñados para ser controladores de la red, los cuales se comunican con otros controladores de su mismo nivel jerárquico, haciendo referencia a la posibilidad de tener uno o varios controladores dirigiendo la misma red, teniendo la claridad que solo uno actúa como maestro y los demás como esclavos; de esta manera se puede asegurar el funcionamiento de la red a pesar del fallo de un controlador, de manera adicional, esta

comunicación entre controladores permite servicios como calidad de servicio de extremo a extremo en redes amplias. Entre la capa de control y la capa de aplicaciones, se pueden observar algunos de los lenguajes de programación que son usados para el desarrollo de programas API's, los cuales sirven como canales de comunicación entre el controlador y las aplicaciones. Por último, en la parte superior de la ilustración, se puede observar en la capa de aplicaciones algunos ejemplos de servicios que pueden ser desarrollados para esta arquitectura; por ejemplo, control de tráfico, control de acceso, monitoreo de la red, programas de enrutamiento, programas que aseguren calidad de servicio o programas específicos para las necesidades del negocio o de los usuarios.

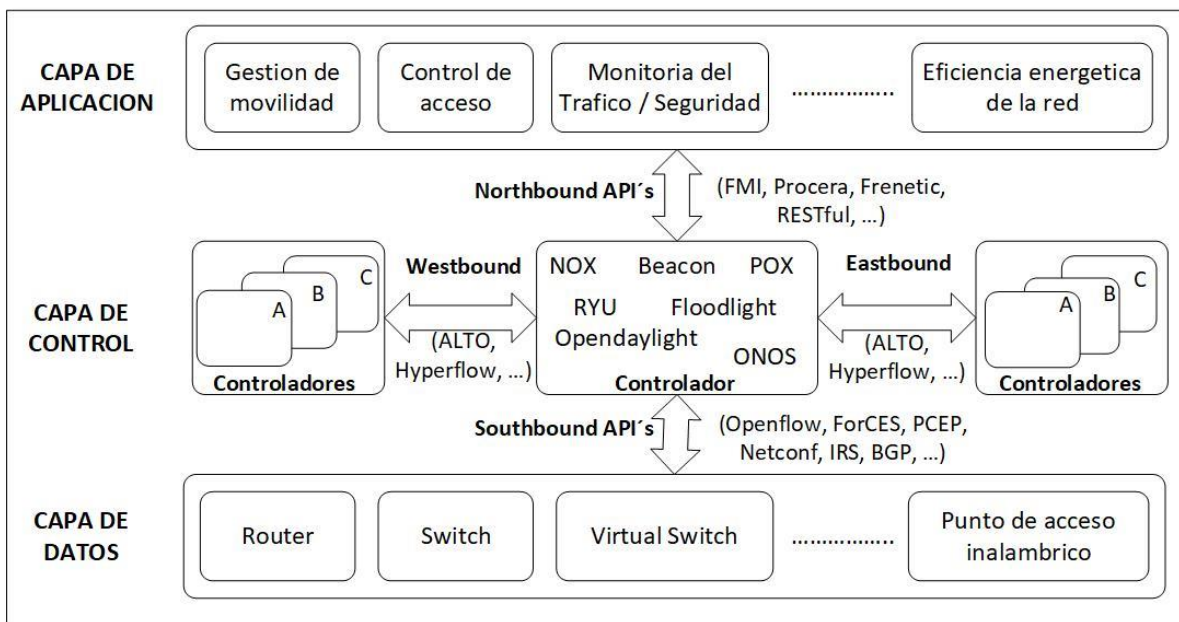


Figura 5 Arquitectura SDN completa [basado en [16]]

2.3 LOS CONTROLADORES SDN

Existen múltiples controladores de redes definidas por software desarrollados a lo largo de la evolución de la arquitectura, en la literatura analizada se encontró que de manera general los mismos son caracterizados por las siguientes variables:

- **Versión del protocolo Openflow soportado:** indica las versiones del controlador Openflow que soporta el controlador, de ello dependen las funciones y servicios que

nos entrega el mismo; por ejemplo, soporte para IPv4 y/o IPv6, soporte para enlaces ópticos, soporte de tunelamiento, etc. A la fecha de realización de este trabajo se han entregado las versiones del 1.0 al 1.5 del protocolo Openflow.

Aunque a primera vista resulte conveniente contar con soporte para la última versión del mismo, debido a los cambios y mejoras realizadas en el código, se pueden presentar errores de reconocimiento y configuración; por lo cual es indispensable revisar si las aplicaciones requeridas y los equipos de transmisión soportan la misma versión del protocolo Openflow que el controlador seleccionado para la red, y de esta manera poder evitar inconvenientes.

- **API hacia el sur:** indica que protocolos soporta en controlador para realizar la comunicación hacia la capa de datos (Hardware de la red), en la mayoría de los casos los controladores traen soporte para algunas de las versiones del protocolo Openflow; de manera adicional, también soportan algunos protocolos de la red ethernet como ARP, DHCP, BGP, IS-IS, etc. De esta manera dan la posibilidad de habilitar conexiones con equipos de transmisión que funcionen bajo configuración híbrida, es decir redes SDN y redes ethernet; además de posibilitar la conexión con redes virtualizadas y con funcionalidades en la nube.
- **API hacia el norte:** indica que protocolos, API's o lenguajes de programación soporta el controlador para poder comunicarse y ofrecer los servicios del controlador a la capa de aplicaciones.
- **Tipo de Interfaz:** indica el tipo de interfaz que usa el controlador para comunicarse con el usuario, la cual puede ser básica a través de una Interfaz de Línea de Comandos (CLI), o de una forma más amigable a través de una Interfaz Gráfica de Usuario (GUI), la cual se puede acceder a través de un navegador WEB (WEB GUI) o a través de una aplicación diseñada para tal fin.
- **Aplicaciones de Enrutamiento:** indica que aplicaciones han sido diseñadas junto con el controlador o por terceros, para ofrecer servicios de enrutamiento en la red; por ejemplo, STP, DHCP, ARP, NAT, balanceo de carga, etc.
- **Aplicaciones de Medición y Monitoreo:** indica que aplicaciones han sido diseñadas junto con el controlador o por terceros, para ofrecer servicios de medición

y seguimiento de la red; por ejemplo, monitoreo de la red, revisión de la topología, gestión de estadísticas, etc.

- **Aplicaciones de Seguridad y Confianza:** indica que aplicaciones han sido diseñadas junto con el controlador o por terceros, para ofrecer servicios de seguridad y resiliencia en la red; por ejemplo, corrección de fallos, control de acceso, seguimiento del comportamiento de aplicaciones y usuarios, etc.
- **Integración con la Nube y Virtualización:** indica si el controlador soporta aplicaciones de integración con servicios en la nube, de servicios de virtualización de redes y virtualización de funciones (NFV).
- **Máximo Número de Flujos Soportados:** indica el número máximo de flujos (conexiones), que puede soportar el controlador y que se han medido al ser sometido a pruebas de rendimiento. Se debe tener en cuenta que este valor varía en función de las capacidades del equipo de hardware con el que se desarrolló la prueba, debido a que por ser tan costoso en recursos el realizar una prueba en equipos reales, en su mayoría estas pruebas han sido realizadas mediante programas de simulación de redes, como Mininet.
- **Código abierto o propietario:** indica si el controlador ha sido desarrollado y distribuido bajo una licencia que permite a los usuarios tener acceso al código fuente, de esta manera puede ser estudiado y modificado sin mayores restricciones para su uso; en algunos casos con la posibilidad de ser redistribuirlo, bajo los términos de la licencia con la cual fue adquirido originalmente.
- **Sistemas Operativos:** indica sobre cuales sistemas operativos funciona el controlador, esto con el objetivo de saber en cuales debe ser instalado.
- **Soporta Multihilos:** Indica si el controlador realiza procesamiento de la información de manera lineal o si funciona bajo una arquitectura multihilos, la cual permite realizar varios procesos de manera simultánea, de esta manera se reduce el tiempo de procesamiento de las tareas.
- **Consistencia de la Información:** indica si el diseño del controlador tiene entre sus funciones garantizar la coherencia y la estabilidad de la información de la red,

además de asegurar que la misma sea desplegada y ejecutada en todos los nodos de la red de forma simultánea; evitando de esta manera errores de configuración, en caso que se retome el control de la red después de una pérdida de comunicación entre los equipos de transmisión y el controlador, o en caso que sea necesario realizar el proceso de cambio de rol de un controlador esclavo a maestro.

- **Ambientes de Uso:** indica los tipos de redes para los que fue diseñado el controlador y para los cuales están desarrolladas sus características; por ejemplo, redes pequeñas si cuenta con un bajo número de flujos de conexión, o si cuenta con una aplicación que le permita poder interactuar con las funcionalidades de la nube, entre otras características.
- **Sistema Distribuido o Centralizado:** indica si el diseño del controlador es de tipo centralizado, permitiendo una alta consistencia de la información de la red, pero a la vez, dejando que exista un punto único de falla, creando así una alta vulnerabilidad en la red. Si el diseño es de tipo distribuido, permite mayor resiliencia ante algún evento de falla en el controlador, pero se debe tener especial énfasis en la manera de mantener la coherencia en la información de la red que posee cada una de las instancias del controlador; permitiendo también dos tipos de arquitectura entre las instancias del controlador, una en la que todos los equipos se comunican dentro de un mismo nivel de jerarquía, siendo esta una distribución plana; y otra en la que existen unas instancias del controlador que concentran la información de otras instancias, que se encuentran debajo de su nivel de jerarquía, siendo esta una distribución de tipo jerárquica.
- **Tolerancia a Fallos:** indica si el diseño del controlador tiene entre sus funciones el soportar el fallo y la posterior recuperación de algunos de sus componentes, manteniendo el servicio con la menor interrupción posible, o si dentro de su arquitectura tiene redundancia de algunos elementos para evitar puntos únicos de falla.
- **Fabricante:** indica cual es la empresa desarrolladora del controlador.
- **Documentación:** indica que tan extensa es la documentación encontrada sobre el controlador, pudiendo catalogarse en Pobre cuando no se tiene una página WEB del controlador y se encuentran solo uno o dos artículos sobre mismo; Buena

cuando el controlador tiene su propia página WEB pero cuenta solo con información básica sobre su configuración, de manera adicional se encuentran entre uno y tres artículos sobre el mismo; Alta cuando el controlador cuenta con su propia página Web donde se encuentra información detallada sobre su configuración (manuales), además se encuentran más de tres artículos sobre el mismo.

- **Tipo de Licencia de Uso:** Indica el tipo de licencia o permisos que le otorga el desarrollador del controlador al usuario del mismo, de esta manera tenemos la licencia GPL (GNU General Public License) que nos permite el uso libre del software, estudiar su código fuente, realizarle modificaciones y distribuirlos a cualquier costo, pero siempre bajo la misma licencia de uso GPL y exigiendo la publicación del código fuente modificado. La licencia Apache y la licencia BSD (Berkeley Software Distribution) permiten los mismos usos de la licencia GPL, pero no obligan a que la distribución de software modificado sea bajo la misma licencia del software origen y tampoco que su código fuente sea de carácter abierto. La licencia EPL (Eclipse Public License) y la licencia LGPL (GNU Lesser General Public License) permiten la combinación de software libre con software propietario, exigiendo la publicación del código fuente solo cuando el software es considerado un trabajo derivado del original. Las licencias propietarias o privativas, generalmente no permiten estudiar su código fuente, además prohíben la modificación y distribución del software, permitiendo su uso solo bajo las condiciones expresas que otorgue el desarrollador.
- **Lenguaje de Programación:** indica el tipo de lenguaje usado para desarrollar el controlador, el cual le brinda características como velocidad de procesamiento, modularidad, capacidad de integración con otros controladores o aplicaciones, etc.
- **Versión del Programa:** Indica la versión del controlador, de esta manera se puede hacer un seguimiento a la evolución del mismo.
- **Página del Controlador:** Indica la dirección WEB de la página del controlador, donde se puede encontrar información básica de su fabricante, configuración del controlador, información de contacto con los desarrolladores, etc.

- **Fecha de Actualización de la Página:** indica la última fecha de actualización de la página WEB, dándonos una idea del tiempo de seguimiento del proyecto por parte de sus desarrolladores.

En la investigación sobre los controladores para redes definidas por software, se encontraron 35 controladores documentados con al menos un artículo; con base en la información encontrada, cada controlador fue caracterizado en las 23 variables anteriormente señaladas. Para facilitar el proceso de revisión de los controladores, estos fueron divididos en 5 grupos, los cuales se encuentran a continuación.

2.3.1 Controladores sin página Web

En este grupo se reúnen los controladores que no cuentan con una página Web propia, dificultando la obtención de información sobre los mismos; de manera adicional, se debe tener en cuenta que no se puede corroborar si aún se encuentran activos estos proyectos. En este grupo se encuentran los controladores Meridiam, Rosemary, ONIX, Hiperflow, Smart-Light, Fleet y Elasticon; a continuación, se encuentra en breve resumen de sus características principales.

- **Meridiam:** Fue desarrollado por el instituto de investigación Thomas J. Watson de IBM, su inicio tomo como base el controlador Floodlight, por lo tanto, fue desarrollado en lenguaje JAVA; su objetivo principal era implementar las ventajas de la arquitectura SDN en el ambiente de la nube, enfocando el desarrollo de sus características para ofrecer Infraestructura como Servicio y Plataforma como Servicio. Su concepto divide la red completa en 5 partes: la primera son los terminales, que son los mismos elementos de red (routers, switches, etc), los cuales son agrupados según las necesidades y sus características; la segunda son los canales de comunicación desde los grupos de terminales hacia los servicios de red que requieran; la tercera capa de la red agrupa todos los servicios; la cuarta capa de la red es el control de los enlaces que interconectan los servidores y las aplicaciones, agrupando también las capacidades asignadas y los enlaces entre grupos de terminales donde se tiene el almacenamiento de la red virtual; por último se tiene el canal de comunicación entre la capa de control y las aplicaciones [37].

- **Rosemary:** Fue desarrollado por la empresa SRI Internacional usando lenguaje C, tiene una arquitectura centralizada especialmente creada para cubrir las fallas de otros controladores en robustez y resiliencia, tratando de afectar lo menos posible el rendimiento del controlador. Su concepto se divide en cuatro bloques funcionales: el primero es una Capa de Abstracción de Datos (DAL), que concentra la información de los equipos de la capa de Hardware; el segundo es un Kernel que provee los servicios básicos necesarios a las aplicaciones de red; el tercero es un sistema de control de uso de librerías y por ultimo un monitor de recursos para las aplicaciones, creando instancias distintas para la ejecución de las aplicaciones y las funciones del controlador. De manera adicional, Rosemary tiene funcionalidades como control de acceso, autenticación y arranque en modo seguro, donde inicializa solo con las funciones básicas que el administrador seleccione [38].
- **ONIX:** Fue desarrollado por las empresas Nicira Networks, NTT OSRG y Google en el año 2010, usando lenguaje C++; su objetivo inicial era permitir el rápido desarrollo de aplicaciones de control para las redes definidas por software, por lo cual soporta aplicaciones desarrolladas en lenguajes C++, Python y Java. Además de asegurar un buen rendimiento en el procesamiento de paquetes, al dividir el control de la red en distintas instancias del controlador, cada una de las cuales toman el control de una parte de la red. También cuenta con características de resiliencia y alta disponibilidad debido a su arquitectura distribuida, asegurando la consistencia de la información en todas las instancias del controlador, usando a una API propia que permite la comunicación con un servidor NIB (Network Information Base), donde se almacena entre otros, la topología y los eventos de la red. Su licencia es de tipo propietario [39].
- **Hiperflow:** Fue desarrollado por la Universidad de Toronto en el año 2010, tomando como base el controlador NOX y usando lenguaje C++; siendo el primer controlador con una arquitectura de control distribuida plana, en otras palabras, la red es dividida en subconjuntos de Switches, los cuales son controlados desde una de las diversas instancias del controlador, pero a su vez estas instancias comparten la información de eventos específicos entre sí, para mantener una mirada global de la red. Esto permite reducir la cantidad de procesamiento que debe soportar cada una de las instancias del controlador y a la vez permite mayor disponibilidad, previendo el caso

de que se pierda la comunicación de una instancia del controlador con su grupo de Switches, los cuales serían tomados por otra instancia del controlador [37].

- **Smart-Light:** Es un controlador pensado para redes pequeñas o medianas, desarrollado por la universidad de Lisboa; su diseño fue basado en el controlador Floodlight y escrito en lenguaje Java, cuenta con una arquitectura de tipo distribuida y soporta Openflow en su versión 1.3. El objetivo principal de este controlador era satisfacer los problemas de resiliencia de la arquitectura SDN, pero manteniendo la consistencia de la información dentro de un rendimiento aceptable, para que en caso de pérdida o falla de comunicación entre el controlador primario (maestro) y la capa de datos, uno de los otros controladores es seleccionado como el nuevo controlador maestro; y para asegurar la continuidad en el sistema, toda la información de la capa de datos es continuamente actualizada y centralizada, en un almacenamiento externo compartido bajo la forma NIB (Network Information Base). De manera adicional, cada controlador cuenta con una pequeña base de datos cache para mejorar el rendimiento de la red; además de asegurar la conexión entre controladores y el almacenamiento de la información de la red, mediante una API llamada Zookeeper [40].
- **Fleet:** Desarrollado por la Universidad de Carnegie Mellon, Fleet es el primer controlador que se desarrolló con el objetivo de evitar problemas de red, por actividades maliciosas o por malas configuraciones de los administradores de la red; usando herramientas como limitar el número de administradores hasta máximo 10, sistemas de autenticación de controladores en los equipos de la capa de datos, realizando protocolos de monitoreo del comportamiento de los controladores, comparando sus políticas y configuraciones de red para determinar cuál puede ser malicioso. Además cuenta con protocolos para actuar en caso de sobrecarga de paquetes y recuperación de la red; de manera adicional, para asegurar la resiliencia ante fallos, su arquitectura es físicamente distribuida pero lógicamente centralizada, evitando un único punto de falla [41].
- **Elasticon:** Es un controlador de tipo distribuido, basado en el controlador Floodlight y desarrollado en lenguaje JAVA, cuyo objetivo principal es poder asegurar que siempre exista mínimo un controlador para gestionar la red, asegurando que cada elemento de la capa de datos tenga comunicación con varias instancias del

controlador, siendo una de estas instancias configurada como maestra y las demás como esclavas. De manera adicional, realiza continuamente un seguimiento a los límites predefinidos de comportamiento de la red, tanto para cambiar el controlador maestro como para realizar el balanceo de carga entre las instancias del controlador, manteniendo la red siempre dentro de su mejor rendimiento; asegurando de esta manera la entrega de los paquetes de datos a pesar de fallas en la red y que estos puedan ser procesados en el orden correcto, además de utilizar una base de datos independiente de la red, para mantener la consistencia de la información [42].

2.3.2 Controladores inactivos

En este segundo grupo, se encuentran los controladores cuyos proyectos no han sido modificados después del año 2015 y/o se encuentran oficialmente inactivos, esto basado en la información encontrada en su página Web o en los artículos sobre los mismos. En este grupo se encuentran los controladores NOX (-MT), POX, Maestro, Pane, Open MUL, Open IRIS, al igual que el controlador Beacon, el cual, a pesar de haber tenido su última actualización en el año 2017, ya fue declarado oficialmente inactivo; el mismo caso sucede con el controlador Active Fabric Controller, que fue declarado inactivo y reemplazado en los equipos de transmisión de su fabricante por el controlador OpenDaylight.

- **NOX:** Fue el primer controlador diseñado para la arquitectura SDN, creado por la Universidad de Stanford y la empresa Nicira Networks en el año 2008, usando lenguaje C++ y entregado bajo la licencia GNU GPL (GNU General Public License); su código fue donado para la investigación y el desarrollo de proyectos relacionado con las Redes Definidas por Software, siendo el controlador más usado en los primeros años del desarrollo de esta nueva arquitectura de red, empleando para desarrollar sus tareas un hilo de trabajo cooperativo, el cual resultaba ser un cuello de botella al ir aumentando la cantidad de paquetes entrantes y salientes. En el año 2011, para mejorar este aspecto se entregó al público una mejora del controlador desarrollado por la misma compañía de nombre NOX-MT, que realiza el desarrollo de tareas mediante múltiples hilos de trabajo, aumentando su capacidad de procesamiento de paquetes de algunos miles de paquetes por segundo a un poco

más de un millón de paquetes por segundo, esto dependiendo de las características técnicas del equipo donde es instalado. Actualmente su versión más reciente es la 1.0, la cual soporta la versión 1.0 del protocolo Openflow, contando además con diversas aplicaciones para facilitar su funcionamiento, entre las que se encuentran interfaz gráfica, descubrimiento de topología, monitoreo y visualización de la red, además de una aplicación de enrutamiento y autenticación de nodos de red; de manera adicional, cuenta con algunas aplicaciones desarrolladas por terceros [43]. Más información puede ser encontrada en la herramienta de desarrollo colaborativo GitHub, donde tiene su propia página [44].

- **POX:** Basado en el controlador NOX, fue desarrollado por la empresa Nicira Networks en el año 2011, usando el lenguaje de desarrollo Python, su código fuente fue entregado al público bajo la licencia GNU GPL (GNU General Public License). Siendo el lenguaje Python más sencillo y práctico para desarrollar aplicaciones, la apuesta era una evolución más rápida de funcionalidades para este controlador, pero sacrificando un poco el rendimiento. POX en su versión entregada en el año 2014, de nombre Dart, soporta el protocolo Openflow en sus versiones 1.0, 1.1 y 1.2, contando con diversas funcionalidades y/o aplicaciones como: una interfaz gráfica que permite ver la topología de la red y monitorear los cambios en la misma, aplicaciones de enrutamiento como Spanning Tree, DHCP, NAT y ARP; Además de aplicaciones de balanceo de carga entre otras. Los controladores POX y NOX son usados generalmente para la investigación y no para uso comercial, debido a su poca capacidad de gestión en redes extensas [45]. Más información puede ser encontrada en la herramienta de desarrollo colaborativo Git Hub, donde tiene su propia página [46].
- **Maestro:** Es un controlador creado por la Universidad de Rice (Houston, EE. UU) en el año 2011, desarrollado en lenguaje JAVA y liberado bajo la licencia de software libre GNU GPL V2.1 (GNU General Public License), soportando el protocolo Openflow en su versión 1.0. Su objetivo principal es ayudar en el desarrollo de aplicaciones de control de red, por lo cual cuenta entre sus características generales con procesamiento de tareas a través de múltiples hilos, además de aprovechar el paralelismo y la optimización de funciones para mejorar su rendimiento, realizando trabajos complejos con pocos recursos, siendo estas características ideales para ser usado en centros de datos. También cuenta con aplicaciones como monitoreo y

descubrimiento de la topología de red, enrutamiento y autenticación de nodos; además de soportar conexiones a hipervisores, con el fin de contrarrestar el punto único de falla que amenaza a los controladores de tipo centralizado y mejorar la gestión en redes extensas [47]. Más información puede ser encontrada en la herramienta de desarrollo colaborativo GitHub, donde tiene su propia página [48].

- **Pane:** Es un controlador desarrollado por la universidad de Brown en el año 2013, escrito en lenguaje Haskell, tiene una arquitectura de tipo distribuida y fue liberado bajo la licencia BSD 3.0. Este controlador es un prototipo con características de seguridad, rendimiento y con un comportamiento estable, al mismo tiempo que entrega permisos de configuración a los administradores de red, usuarios y aplicaciones. Entre sus características generales soporta Openflow en su versión 1.0 y también el protocolo MPLS, además soporta el lenguaje 4D en sus conexiones hacia los equipos de la capa de datos; de manera adicional, cuenta con una API propia para su conexión con la capa de aplicaciones y su configuración se puede realizar mediante línea de comandos [49]. Más información puede ser encontrada en la página web del proyecto [50].
- **Open MUL SDN:** Es un controlador desarrollado por la fundación OpenMUL en lenguaje C; entre sus características principales se encuentra que tiene procesamiento multihilos, soporta Openflow en sus versiones 1.0, 1.3 y 1.4. Este controlador fue elaborado con el propósito de ser confiable ante fallos y de manera adicional que mantuviera un buen rendimiento, para ello cuenta con una arquitectura de tipo distribuida y con una implementación altamente flexible, siendo un controlador elaborado de forma modular y además con la capacidad de soportar múltiples protocolos de conexión, tanto hacia la capa de aplicación como hacia la capa de datos. También cuenta con una API Northbound multinivel que permite la comunicación y el almacenamiento de múltiples aplicaciones, entre ellas, funciones que vienen por defecto con el controlador, como el enrutamiento de paquetes, visualización de topología, gestión mediante una interfaz gráfica y también conexión con la nube a través de OpenStack [51]. Más información puede ser encontrada en la página web del proyecto [52].
- **Open IRIS:** Es un Controlador desarrollado por el instituto coreano de investigación de nombre ETRI (Electronics and Telecommunications Research Institute);

basándose en el controlador Floodlight e implementado en lenguaje Java. Teniendo como objetivo principal satisfacer las necesidades de disponibilidad y escalabilidad en los sistemas de control de las redes SDN, usando entre otras herramientas, dos conexiones de comunicación desde el controlador hacia el Switch para asegurar disponibilidad, actuando una como comunicación maestra y la otra como esclava; a través de estos enlaces de comunicación supervisa constantemente el estado de los Switches, enviándoles mensajes y en caso de que alguno no responda, este equipo es sacado de la lista de Switches activos y se inicia un proceso para detectar si tiene alguna falla. Además, utiliza una base de datos externa para asegurar que la información de configuración de la red siempre este accesible, igualmente en caso de detectar la conexión de un Switch nuevo o la desconexión de alguno, realiza automáticamente el balanceo de los flujos de transporte entre los Switches activos. Dentro de sus características básicas soporta el protocolo Openflow en sus versiones 1.0 a 1.3, además de funcionar con otros protocolos como OVSDB, SNMP, BGP, PCEP y LISP entre otros; también cuenta con una interfaz gráfica para realizar la gestión de distintas aplicaciones como monitoreo de red, topología, gestión de fallos, entre otras [53]. Más información puede ser encontrada en la página web del Instituto [54].

- **Beacon:** Es un controlador desarrollado en lenguaje JAVA por la Universidad de Stanford, con el objetivo de tener un controlador con un alto nivel de rendimiento y productividad; fue liberado en el año 2010 bajo la licencia GNU GPL (GNU General Public License). Dentro de sus características generales soporta el procesamiento de tareas multihilos, también soporta la versión 1.0 del protocolo Openflow mediante la API OpenFlowj; de manera adicional, cuenta con funcionalidades o aplicaciones como: interfaz gráfica que le permite ver la topología de la red y realizar el monitoreo de la misma, también cuenta con aplicaciones de enrutamiento, aislamiento de redes, firewall y balanceo de carga, entre otras. Además de contar con aplicaciones desarrolladas por terceros como: OpenVAS, que permite realizar escaneo de vulnerabilidades en la red, igualmente cuenta con aplicaciones que permiten su interconexión con tecnologías de arquitectura en la nube como OpenStack y Open Nébula [55]. Este controlador al igual que NOX y POX es muy usado en el desarrollo de investigaciones sobre las redes definidas por software, pero no comercialmente [56]. Más información puede ser encontrada en la página web del proyecto [57].

- **Active Fabric Controller:** Es un controlador desarrollado por la compañía DELL, cuyo objetivo principal es poder trabajar en conjunto con los instrumentos y funcionalidades que entregan las plataformas en la nube, esto a través de OpenStack y su interfaz Neutrón; interconectando estos dos mundos, la nube y el hardware, de esta manera poder garantizar una alta disponibilidad y resiliencia en la red, además de poder asegurar calidad de servicio en la transmisión de paquetes a través de políticas de enrutamiento de extremo a extremo. Entre sus características principales se encuentra que fue diseñado bajo el sistema operativo Linux y esto le permite mayor rapidez de procesamiento, también cuenta con conexión a una base de datos externa, permitiendo la recuperación de fallos y asegurando la consistencia de la información; además de soportar Openflow en su versión 1.3 para la comunicación con la capa de datos y usar REST API en la comunicación con la capa de aplicaciones [58]. Este controlador fue descartado por DELL para ser usado en sus equipos y fue reemplazado por OpenDaylight. Más información puede ser encontrada en la página web del fabricante [59].

2.3.3 Controladores sin actualización desde 2017

En este tercer grupo se encuentran los controladores cuyos proyectos se encuentran Inactivos desde el año 2017, es decir, desde ese año no se ha realizado alguna modificación en su página WEB, indicando que no se ha realizado un seguimiento adecuado a su evolución o que desde esa fecha no se realiza ningún avance en el proyecto. En este grupo se encuentran los controladores Trema, Programmable Flow, Disco, Yanc, Beehive, RUNOS, Ryu NOS y LOOM; los cuales, en su mayoría, no han tenido actualización de su página WEB desde el año 2016, teniendo solamente el controlador Trema, Ryu, NOS y Disco actualizaciones en el año 2017.

- **Trema:** Es un controlador marco creado por la compañía NEC, con el fin de facilitar el desarrollo de otros controladores de redes definidas por software, fue desarrollado en lenguaje C y liberado en el año 2009 bajo la licencia de software libre GNU GPL (GNU General Public License); soporta el protocolo Openflow en sus versiones 1.0 y 1.3, también permite el desarrollo y prueba de nuevas aplicaciones en lenguaje Ruby, contando para esto con aplicaciones base, con servicios como revisión de

topología de la red, enrutamiento, aislamiento de redes en capa 2. Además, permite el uso de la herramienta de pruebas de funcionamiento Cbench. De manera adicional, cuenta con una aplicación de emulación de red, que permite la implementación de redes sencillas para realizar pruebas en los controladores y aplicaciones diseñadas [51]. Más información puede ser encontrada en la herramienta de desarrollo colaborativo GitHub, donde tiene su propia página [60].

- **Programmable Flow:** Es un controlador basado en Trema, fue desarrollado por la compañía NEC en lenguaje C y liberado en el año 2010 bajo la licencia de software libre GNU GPL (GNU General Public License); se encuentra en su versión 5.0 y soporta el protocolo Openflow en sus versiones 1.0 y 1.3, además su gestión se realiza a través de una interfaz gráfica. Este controlador permite el desarrollo y prueba de nuevas aplicaciones creadas en lenguaje de programación Ruby, dando la posibilidad de tomar como base las actualmente desarrolladas, por ejemplo, monitoreo y topología de la red, enrutamiento, calidad de servicio, integración con arquitecturas en la nube como OpenStack, entre otras; esto con el fin de desarrollar aplicaciones más complejas [61]. Más información puede ser encontrada en la página web del proyecto [62].
- **Disco:** Es un controlador de tipo distribuido plano, desarrollado en lenguaje JAVA por Thales Community and Security en 2014, tomando como base el controlador Floodlight. Su funcionamiento divide el controlador en varias instancias y cada instancia mantiene la gestión y el control de una parte de la red; pero manteniendo la gestión de extremo a extremo de la red, mediante un canal de control liviano pero altamente manejable, que realiza la conexión entre las instancias del controlador, teniendo de esta manera una visión global de la red, lo cual le permite un control de la red interdominio e intradominio a pesar que su información se encuentre separada [63]. Más información puede ser encontrada en la página web del proyecto [64].
- **Yanc:** Es un controlador desarrollado por la universidad de Colorado en el año 2013, usando lenguaje C y C++; está basado en el sistema operativo Linux y sigue las políticas de la filosofía UNIX, cuyo objetivo era desarrollar un controlador que pudiera interactuar con otros controladores y aprovechar las aplicaciones diseñadas para los mismos, sin importar el lenguaje específico en el que hubieran sido escritos, para lo cual entrega y recibe todos los datos de la red en archivos estándar tipo I/O.

Entre sus características generales soporta Openflow en sus versiones 1.0 a 1.3, tiene una arquitectura distribuida plana y su gestión se realiza mediante línea de comandos [65]. Más información puede ser encontrada en la herramienta de desarrollo colaborativo Git Hub, donde tiene su propia página [66].

- **Beehive – Kandoo:** Es un controlador de tipo distribuido, desarrollado por la Universidad de Toronto en el año 2012 bajo el nombre de Kandoo, aunque después cambiaría el nombre del proyecto a Beehive en el año 2013; este controlador tiene como objetivo principal asegurar un buen rendimiento de la red incluso cuando se está recuperando de una falla, contando entre sus características el no necesitar toda la información de la red para poder funcionar correctamente, es decir, solo con parte o incluso con solo la información de un equipo de transmisión (Switch, Router, etc), puede gestionar, reaprender, optimizar y distribuir la información en las demás instancias del controlador continuando con el funcionamiento de red. Dentro de sus características generales soporta Openflow en sus versiones 1.0, 1.1 y 1.2, además de contar con aplicaciones básicas de enrutamiento, monitoreo y de gestión de fallas; de manera adicional, cuenta con un almacenamiento interno de la información de la red de tipo liviano, permitiéndole gran capacidad de reacción [67]. Más información puede ser encontrada en la herramienta de desarrollo colaborativo Git Hub, donde tiene su propia página [68].
- **Runos:** Es un prototipo de controlador SDN desarrollado en el año 2014, por el grupo ARCCN (Applied Research Center for Computer Networks) de la Universidad Estatal de Moscú (Rusia); fue escrito en lenguaje C++ y liberado bajo licencia Apache 2.0. Siendo su objetivo principal intentar solucionar algunos problemas de los controladores de redes SDN al ejecutar varias aplicaciones al mismo tiempo, momentos donde se pueden generar problemas de direccionamiento de flujos al igual que el solapamiento de políticas de red, también intenta facilitar el proceso de desarrollo de aplicaciones, haciendo que las tareas del controlador y de los equipos de transmisión sean más accesibles al programador, aprovechando la abstracción de las capas inferiores y su arquitectura de tipo modular. Algunas de sus características principales son el soporte del protocolo Openflow en sus versiones 1.0 y 1.3, además cuenta con una interfaz gráfica para la gestión de aplicaciones, entre las cuales se encuentran el monitoreo, enrutamiento y topología de la red, autenticación

y control de usuarios, aislamiento de tráfico entre otras. Más información puede ser encontrada en la página web del proyecto [69]

- **Loom:** Es un controlador prototipo desarrollado por Universidad de Ciencia y Tecnología (AGH), de Cracovia (Polonia), con una arquitectura distribuida, cuyo objetivo principal es obtener un controlador con alta escalabilidad y robusto, que soporte 10.000 equipos de enrutamiento y 100.000 equipos finales; entre sus características principales se encuentra que soporta el protocolo Openflow en sus versiones 1.3 y 1.4, además de mantener una única base de datos para la información de la red. Más información puede ser encontrada en la página web del proyecto [70].
- **Ryu NOS:** Es un controlador desarrollado por la empresa japonesa NTT OSRG, usando el lenguaje de programación Python y liberado en el año 2013 bajo la licencia gratuita apache 2.0, fue diseñado para permitir el rápido desarrollo de nuevas aplicaciones de control y administración de recursos, soportando el protocolo Openflow en sus versiones 1.0 a 1.5; es un controlador multiplataforma pero está limitado por su entorno de ejecución a través de máquinas virtuales, además de no contar con una alta modularidad. Tiene entre sus aplicaciones básicas una interfaz gráfica limitada, revisión topología y monitoreo de la red, Spanning Tree, Firewall, recuperación de fallos, además de soportar la interconexión con arquitecturas en la nube a través de OpenStack [45]. Más información puede ser encontrada en la página web del proyecto [71].

2.3.4 Controladores con menor información sobre sus características

En este grupo se encuentran los controladores de cuyos proyectos se tiene la menor información, realizando una comparación entre los 13 controladores restantes y teniendo en cuenta la información sobre las 23 variables de caracterización antes mencionadas. En este grupo se encuentran los controladores IBM SDN VE, Faucet, Open Contrail Controller, North Star, Blue Planet SDN Controller; A saber, del controlador Blue Planet SDN Controller solo contamos con información en 15 de las 23 variables, del controlador North Star contamos con información en 17 de las 23 variables, de los controladores IBM SDN VE y

Faucet contamos con información sobre 20 de las 23 variables, y del controlador Open Contrail Controller contamos con información en 21 de las 23 variables.

- **IBM SDN For Virtual Enviroment:** Es un controlador basado en OpenDaylight lanzado en el año 2013, que muestra la visión de IBM sobre la evolución de las tecnologías de la información, no concentrándose únicamente en las Redes Definidas por Software sino proyectando todo un ambiente virtualizado, que comprende además la virtualización de funciones de red (NFV), la virtualización de redes (NV) y el almacenamiento de información en la nube, todo esto soportado en OpenStack a través de la interfaz Neutrón, entregando altas capacidades de resiliencia mediante servidores virtuales, actuando uno como activo o maestro y los demás en espera hasta un máximo de cuatro; de manera adicional, entrega una interfaz de nombre DOVE para realizar la interconexión con los switches virtuales. Entre sus características principales, se encuentra soporte para el protocolo Openflow en sus versiones 1.0 y 1.3, además de otros protocolos como BGP, LISP, PCEP, entre otros; cuenta también con una interfaz gráfica que permite la gestión de aplicaciones propias y de terceros como topología y monitoreo de la red, balanceo de carga, aislamiento de clientes, autenticación y control de usuarios entre otras [72]. Más información puede ser encontrada en la página web del fabricante [73].
- **Faucet:** Es un controlador de código abierto desarrollado por la universidad de Waikato (Nueva Zelanda) en el año 2015, está basado en el controlador Ryu NOS y fue escrito en lenguaje Python con menos de 1000 líneas de código, lo cual hace que sea muy rápida su respuesta; entre otras características funciona con una arquitectura centralizada, soportando el protocolo Openflow en su versión 1.3, de manera adicional, a través de su interfaz gráfica permite la gestión del enrutamiento de paquetes y el control de usuarios de la red, además de ser compatible con servicios virtualizados (NFV). Más información puede ser encontrada en la página web del proyecto [74]
- **Open Contrail Controller:** Es un controlador de código abierto desarrollado en lenguaje C++ en el año 2012 por la empresa Juniper Networks, siendo liberado para uso libre bajo la licencia Apache 2.0. Este controlador permite la fácil integración de las redes físicas con las redes virtuales a través de los Vrouter contrail, siendo estos

routers virtuales junto a APIS como Neutrón los que permiten el acceso a servicios en la nube, los cuales son prestados mediante plataformas como OpenStack, Amazon Web Services, entre otros; Siendo en sí, más que un controlador, una plataforma integral de virtualización. De forma más global, Open Contrail Controller solo es una parte de la red completa de enrutamiento, virtualización y servicios en la nube creada por Linux Fundación de nombre Tungsten [75]. Entre sus características generales, permite el desarrollo de aplicaciones en diversos lenguajes, además de contar con aplicaciones como aislamiento de clientes, Ethernet VPN, firewall, gestión y análisis de estadísticas, gestión de fallos y de enrutamiento de flujos de paquetes, entre otras. Más información puede ser encontrada en la página web del proyecto [76].

- **North Star:** Es un controlador desarrollado por Juniper Networks, que permite el control del flujo de paquetes a través de varios protocolos usados en la red actual como MPLS, BGP, IS-IS, entre otros; de manera adicional, apoyándose en la fuerza del área de desarrollo de la compañía, ha logrado implementar numerosas aplicaciones para su controlador, como una interfaz gráfica, monitorización de la topología, enrutamiento, balanceo de carga, recuperación de fallos entre otras. También permite la conexión con servicios en la nube a través de una interfaz con OpenStack, además de estar trabajando en una interfaz que le permita la conexión con redes de transporte óptico; la documentación sobre su implementación es bastante detallada, aunque información adicional sobre su funcionamiento y arquitectura es difícil de encontrar por tener una licencia de uso propietaria [76]. Más información puede ser encontrada en la página web del proyecto [77].
- **Blue Planet SDN Controller:** Es un controlador desarrollado por la empresa Ciena Incorporated, cuyo objetivo más que el desarrollo de un controlador, es la construcción de una red adaptativa que automáticamente controle sus recursos, tomando decisiones basadas en la información del análisis del comportamiento de la misma red; para ello cuenta con varias API's de código abierto que funcionan como interfaces para la comunicación entre el controlador y los Switches, al igual que para la comunicación entre el controlador y las aplicaciones, esto con el fin de poder engranar eficientemente con Switches y aplicaciones de cualquier desarrollador y/o fabricante de forma rápida y sencilla, también cuenta con la herramienta Blue Planet Toolkit, que permite el desarrollo de aplicaciones y/o

soluciones de forma conjunta entre desarrolladores y clientes. Este controlador cuenta de manera adicional con simuladores de red en la nube para realizar pruebas, teniendo un fuerte desarrollo en interfaces para virtualización de funciones de red (NFV) y para comunicación con redes ópticas. Más información puede ser encontrada en la página web del proyecto [78].

2.3.5 Controladores con mayor información

En este grupo de controladores se encuentran los que se distinguen por tener la información más completa sobre los mismos en la literatura encontrada, contando con información en la mayoría de las 23 variables de caracterización antes mencionadas; cada uno cuenta con su propia página WEB, además de estar vigentes y haber sido actualizada su información como mínimo en el año 2018, de estos controladores cuatro son software propietario y tres son software libre. Estos controladores son Floodlight, HP SDN VAN Controller, ONOS, OpenDaylight, Big Network Controller, Cisco Open SDN Controller y Huawei SDN Agile Controller. A saber, se cuenta con información en todas las 23 variables de caracterización de cada uno de ellos, excepto en los controladores Cisco Open SDN Controller y Huawei SDN Agile Controller de los cuales se cuenta con información en 22 de las 23 variables de caracterización.

- **FloodLight:** Es un controlador creado por la empresa de desarrollo de equipos de transmisión Big Switch Networks, siendo una evolución del controlador Beacon; fue desarrollado bajo el lenguaje JAVA y liberado en el año 2012 bajo la licencia de software libre Apache. Entre sus características generales se encuentra que soporta el protocolo Openflow en sus versiones 1.0 a 1.5, cuenta además para su evolución y desarrollo de aplicaciones, con la ayuda de los programadores de la organización no gubernamental de su mismo nombre (Floodlight), al igual que con el apoyo de los desarrolladores de la empresa Big Switch Networks; su desarrollo está dirigido por la Open Network Foundation (ONF), fundación que fue creada para apoyar el desarrollo y la implementación de la arquitectura SDN. Este controlador es usado en numerosos proyectos de investigación por su buena documentación y fácil configuración, además, soporta redes híbridas y puede operar sobre Switch físicos y virtuales. Su más reciente versión es la 1.2, liberada en marzo de 2016, entre sus

aplicaciones cuenta con una interfaz gráfica para revisar su topología y realizar el monitoreo de la misma, además de servicios de enrutamiento, balanceo de carga, aislamiento de clientes, recuperación rápida de fallos, calidad de servicio, firewall, control de acceso entre otras [79]. Más información puede ser encontrada en la página web del proyecto [13].

- **HP SDN VAN Controller:** Fue creado por la empresa fabricante de equipos de transmisión Hewlett Packard bajo lenguaje JAVA, en un principio fue basado en el controlador Beacon y después en su evolución OpenDaylight. Para su desarrollo, mejoras y soporte de fallos cuenta con el apoyo de los desarrolladores de HP, teniendo una evolución y versiones mejoradas de forma rápida; su versión más reciente es la 2.8.8 liberada en el año 2018, la cual cuenta con soporte para Openflow en sus versiones 1.0 y 1.3 [37]. De manera adicional, cuenta con su propia tienda de aplicaciones para redes SDN, que funciona de forma similar a la de aplicaciones de sistemas operativos como Android; también tiene una suite de desarrollo de aplicaciones, que entrega las herramientas necesarias para facilitar la creación de aplicaciones que funcionen sobre el mismo controlador [15]. Más información puede ser encontrada en la página web del proyecto [80].
- **ONOS:** Sistema Operativo de Red Abierta (ONOS), Es un controlador de código abierto escrito en lenguaje JAVA, desarrollado por la fundación Open Networking Laboratory (ON Lab)¹ en el año 2014, tomando como base el controlador Floodlight y utilizando programadores de ambas organizaciones para su desarrollo. Su arquitectura es de tipo distribuida, siendo una implementación orientada a la administración, configuración y despliegue de nuevos servicios, su diseño sigue la arquitectura OSGI (Open Service Gateway initiative); que incluye abstracciones que hacen fácil el desarrollo de nuevos servicios y funciones, debido a que divide la red en 7 subsistemas: Subsistema de Dispositivos, de Enlace, de Host, de Topología, de Ruta, para Reglas de Flujo y de paquetes. Apoyado en el soporte de desarrollo de la fundación ONOSProject.org, realiza la entrega de actualizaciones de versión aproximadamente cada tres meses, su más reciente entrega es Quail en enero de

¹ ON Lab está conformada por AT&T, Ciena, Cisco, Ericsson, Fujitsu, Huawei, Intel y NTT Communication.

2019. De manera adicional se puede encontrar en su página web [81], mucha documentación para implementarlo, realizar investigación y crear desarrollos para el controlador; su interfaz gráfica es muy intuitiva y permite aplicar a las topologías varios filtros para mejorar su presentación [45]. Más información puede ser encontrada en la página web del proyecto [64].

- **OpenDaylight:** Es un controlador de código abierto escrito en lenguaje JAVA, usando como base el controlador Beacon; su primera versión fue liberada en el año 2013 bajo la licencia EPL V1.0, siendo desarrollado por Linux Foundation y por la organización del mismo nombre (OpenDaylight), contando con el apoyo de una red de más de mil desarrolladores distribuidos en sus más de 50 organizaciones miembro ha logrado entregar 12 versiones, siendo la más reciente Magnesium en marzo de 2020. Dentro de sus características principales se puede decir que es un controlador con arquitectura de tipo distribuido plana, permitiendo una alta disponibilidad, además, permite gran capacidad de desarrollo de funciones y aplicaciones debido a que soporta múltiples protocolos de comunicación, entre ellos Openflow en sus versiones 1.0, 1.3 y 1.4; de manera adicional, permite satisfacer necesidades de incorporación de tecnologías nuevas como IoT, NFV y Clusterización. Aunque su interfaz gráfica es poco intuitiva, permite la gestión de numerosas aplicaciones [82]. Más información puede ser encontrada en la página web del proyecto [83].
- **Big Network Controller:** Fue desarrollado por Big Switch Networks y presentado en el año 2012, está basado en el controlador Floodlight, siendo una versión más robusta de este y que cuenta siempre con las más recientes mejoras del mismo. Fue diseñado para facilitar la integración de las redes físicas con las funcionalidades de la nube, contando con una interfaz a este mundo a través de OpenStack y también permite la virtualización de funciones de red (NFV), a través de las funcionalidades de VMWare; de manera adicional, soporta las mismas aplicaciones diseñadas para Floodlight, como el monitoreo y visualización de la red a través de una interfaz gráfica, enrutamiento de paquetes, entre otras; además de tener aplicaciones propias para análisis de datos [84]. Más información puede ser encontrada en la página web del proyecto [85].

- **Cisco Open SDN Controller:** Este controlador fue desarrollado por la empresa fabricante de equipos de transmisión Cisco, siendo una versión mejorada del controlador OpenDaylight, con características adicionales que funcionan solo entre equipos Cisco, de esta manera se trata de amarrar el concepto de SDN a una marca específica, obligando a comprar solo hardware de la misma compañía si se quiere tener acceso a estas funcionalidades adicionales. Para lograr la evolución continua de este controlador, Cisco se apoya en los desarrolladores de la empresa y de manera adicional usa DEVNET [86] para el desarrollo de aplicaciones, la cual es una comunidad en línea de desarrolladores ofrecida por Cisco para fomentar el desarrollo de aplicaciones que funcionen bajo su plataforma, entregando acceso a foros, herramientas de desarrollo y de prueba, además de soporte. Esto le permite entregar un controlador robusto y bien documentado, el cual cuenta entre sus características principales con soporte para el protocolo Openflow en sus versiones 1.0 y 1.3, soportando también redes IP/MPLS a través de su propio protocolo Cisco Opflex; además de entregar aplicaciones propias de enrutamiento, gestión de estadísticas, balanceo de carga, autenticación de usuario entre otras, las cuales pueden ser fácilmente gestionadas a través de su interfaz gráfica. Todo esto a pesar de contar con soporte solo hasta el mayo 31 de 2020, fecha a partir de la cual el proyecto queda en el limbo [19]. Más información puede ser encontrada en la página web del fabricante [87].
- **Huawei SDN AGILE Controller:** Es un controlador diseñado por la multinacional china Huawei, tomando como base el controlador ONOS y de manera adicional cuenta con compatibilidad con el controlador OpenDaylight a través del lenguaje REST API; dentro de sus principales características de diseño se encuentra que permite la interconexión con la nube a través de una interfaz con OpenStack, permitiendo la interconexión de redes físicas y funcionalidades en la nube, además de facilitar el uso de tecnologías de Virtualización de Funciones de Red (NFV) e Internet de las Cosas (IoT). También permite a través de su interface gráfica el monitoreo y gestión tanto de la topología de red física como de la red virtual, permitiendo la implementación rápida de cambios y aplicaciones en las redes creadas. Cuenta con soporte para los protocolos Openflow, Netconf, PCEP, BGP-LS, SNMP, entre otros; para la comunicación entre el controlador y la capa de aplicaciones cuenta con soporte para el lenguaje REST API, además cuenta con

aplicaciones de monitoreo, enrutamiento, recuperación de fallos de red, balanceo de carga, autenticación y control de usuarios, calidad de servicio entre otras [88]. Más información puede ser encontrada en la página web del fabricante [89].

En la Tabla 3, “Características de los controladores de Redes Definidas por Software” se puede observar la información de las 23 variables que caracterizan los controladores de redes SDN anteriormente mencionados.

Tabla 3 Características de los controladores de Redes Definidas por Software [Elaboración Propia]

Características	Meridian	Rosemary	ONIX	HiperFlow	SMart-Light	Fleet	ElastiCon	NOX (-MT)	POX	Maestro	Pane	Open MUL	Open IRIS	Beacon	Active Fabric Controller	Trema	Programable flow	Disco	Yanc	Beehive	RUNOS	LOOM	IBM SDN VE	Faucet	Open Contrail Controller	North Star	Blue Planet SDN Controller	Ryu NOS	Floodlight	HP SDN VAN Controller	ONOS	OpenDaylight	Big Network Controller	Cisco Open SDN Controller (APIC)	Huawei SDN Agile Controller
V. Openflow	V 1.0, 1.3	V 1.0, 1.3	V 1.0	V 1.0	v 1.3			V 1.0	V 1.0, 1.1, 1.2	V 1.0	V 1.0	V 1.0, 1.3, 1.4	V 1.0, 1.1, 1.2, 1.3	V 1.0	v 1.3	V 1.0, 1.3	V 1.0, 1.3	V 1.0	V 1.0, 1.1, 1.2, 1.3	V 1.0, 1.1, 1.2	V 1.0, 1.3	V 1.3, 1.4	V 1.0, 1.3	V 1.3							V 1.0, 1.2, 1.3, 1.4, 1.5	V 1.0, 1.1, 1.2, 1.3, 1.4, 1.5	V 1.0, 1.3	V 1.0	
API (Sur)	Openflow	Openflow, XMPP	Openflow, OVSDb, IS-IS, OSPF	Openflow	Openflow		TCP	Openflow, TCP, SNMP	Openflow, TCP, OVSDb	Openflow, TCP	Openflow, MPLS, 4D	Openflow, OVSDb, OF-Config, Netconf	Openflow, OVSDb, SNMP, BGP, PCEP, LISP	OpenflowJ API (Openflow), TCP, Beacon Datapath API	Openflow, TCP	Openflow, TCP	Openflow, TCP	Openflow, OSPF, BGP, RSVP	Openflow, LLDP	Openflow	Openflow, BGP	Openflow	Openflow, PCEP, BGP, LISP, DOVE (Distributed Overlay Virtual Ethernet)	Openflow, BGP, LACP, ARP	BGP, Netconf, SNMP, XMPP (Canal de Control de Vrouter),	BGP, ISIS, OSPF, PCEP, Netconf	Openflow, MPLS, Netconf, SNMP, BGP	Openflow, Netconf, Vrrp, Netflow, OF-Config, OVSDb, Sflow	Openflow, L3 Agent, L2 Agent	Openflow, Netconf, PCEP, OVSDb, BGP, P4, TL1, SNMP	Openflow, BGP, OSPF	Openflow, Cisco Opflex, Netconf, PCEP, BGP-LS, SNMP, IGP, OVSDb	Openflow, Netconf, PCEP, BGP-LS, SNMP, IGP, OVSDb		
API (Norte)	REST API	REST API	NVP, NBAPI, ONIX, NB API		REST API	REST API	REST API	Pythonic (REST API)	REST API	REST API	REST API	REST API	REST API	REST API, REST WS	REST API	REST API	REST API	REST API	REST API, File System	REST API, JAVA RCP	REST API	JSON	SDN-VE North API (REST API)	REST API	REST API	REST API	REST API	REST API	REST API	REST API	REST API	REST API, Java API (OSGI), Restconf, XMPP, Maven	REST API, Java	REST API	REST API
Interfaz	WEB GUI	CLI	GUI		CLI			CLI WEB, GUI (Python, QT4)	WEB GUI (Python, QT4)	CLI, WEB GUI	CLI	CLI, WEB GUI	CLI, WEB GUI	CLI, WEB GUI	WEB GUI	CLI	CLI, GUI	GUI	CLI	CLI	CLI, WEB GUI	CLI	CLI, WEB GUI	CLI, WEB GUI	CLI, WEB GUI	WEB GUI	CLI	WEB CLI, GUI básica	CLI, WEB GUI	GUI (Shell)	CLI, WEB GUI	CLI, WEB GUI	WEB GUI	CLI, WEB GUI	
Aplicación Enrutamiento			Enrutamiento					Enrutamiento, STP básico, OpenRoads (integración con redes fijas y móviles)	Enrutamiento, Balanceo de carga, STP, DHCP, NAT, ARP	Enrutamiento, STP		PRISM APP (enrutamiento), NAT	Enrutamiento, Balanceo de carga	IRouting Engine interface (Enrutamiento), Balanceo de carga	Enrutamiento, Balanceo de carga, DHCP, QoS, Optimización de tráfico	Enrutamiento	Enrutamiento, Calidad de Servicio				Enrutamiento	Enrutamiento, NAT, DHCP, QoS		Enrutamiento, Balanceo de carga, STP	Enrutamiento	Enrutamiento, Balanceo de carga	Enrutamiento, Balanceo de carga	Enrutamiento	Enrutamiento, Calidad de servicio, STP	Enrutamiento (SPF), Balanceo de carga	Enrutamiento (SPF)	Enrutamiento, Virtual Switch, Calidad de servicio, Balanceo de carga, VLAN, NAT	Enrutamiento, Balanceo de carga	Enrutamiento, NAT, Calidad de servicio, Balanceo de carga	
Aplicación Medición	Monitoreo, Topología	Gestión de recursos	Monitoreo, Topología		Topología			Monitoreo, Topología	Monitoreo, Topología	Monitoreo, Topología		Topología, Gestión de estadísticas	Monitoreo, Topología	Monitoreo, ITopology interface (Topología)	Monitoreo, Topología	Monitoreo, Topología	Monitoreo, Topología	Monitoreo, Topología	Monitoreo, Topología	Monitoreo	Monitoreo, Topología, Gestión de estadísticas		Monitoreo, Topología, Gestión de estadísticas, eventos y rendimiento		Monitoreo, Gestión de estadísticas	Monitoreo, Topología, Gestión de estadísticas	Gestión de estadísticas, Automatización de uso recursos	Monitoreo, Topología	Monitoreo, Topología	Monitoreo, Topología	Monitoreo, Topología	Monitoreo, Topología, Gestión de estadísticas y equipos	Monitoreo, Topología, Gestión de estadísticas	Monitoreo, Topología, Gestión de ancho de banda y de tráfico	
Aplicación Seguridad		Aislamiento y Monitoreo de aplicaciones, Control de acceso y autenticación, Firewall	Gestión de fallos, VPN, VLAN, Ethane application (Seguridad en redes empresariales)	WheelFS (Gestión de fallos)	Zookeeper (Consistencia de la información de la red)	Autenticación de controlador, Gestión de fallos, Seguimiento del comportamiento de los controladores		Autenticación de usuario, FortNix (resolución de conflicto de reglas)	Firewall	Autenticación de equipos, Hipervisor		Firewall, Gestión de fallos	Firewall, VM Security, Open VPN, Recuperación de redes, Sistemas de protección, VLAN OpenVAS (escáner de vulnerabilidades)	VLAN, Autenticación de Usuario, Gestión de fallos, Recuperación de redes	VLAN	VLAN, VPN		Gestión de fallos	Gestión de fallos	SSHFS (Seguridad acceso remoto); WheelFS (Consistencia en la distribución de reglas)	Gestión de fallos	Autenticación y control de usuarios, VPN, VLAN, Gestión de fallos		Autenticación y control de usuarios	VLAN, Firewall, Eth VPN, Gestión de fallos y re-enrutamiento de paquetes	Gestión de fallos, Autenticación de Usuario, VPN	Seguimiento de aplicaciones	Firewall, Gestión de fallos, VLAN, Snort (detección de intrusos)	ACL, Firewall, Autenticación, autorización y seguimiento de aplicaciones, Detección de anomalías, VLAN, Gestión de fallos	Autenticación de usuario y de aplicaciones, Sistemas de protección (2n+1), ACL	Gestión de fallos, Autenticación de usuario, Seguimiento de aplicaciones	Autenticación y seguimiento del comportamiento de usuario	IP sec VPN, Autenticación y control de usuarios; Gestión de fallos		
Integración (Nube y virtualización)	OpenStack, IBM Smart Cloud Provisioning		DVS (Virtualización de la red)					OVN (framework de virtualización)	OVS (Open Virtual Switch)			OpenStack	OpenStack, Open Nebula	OpenStack, Open NFV	OpenStack	OpenStack	OpenStack, Center Virtual Machine Manager (Microsoft)				OpenStack	OpenStack	OpenStack	NFV	OpenStack, CloudStack, Amazon Web Services, OpenNFV	OpenStack	Cloud, NFV	OpenStack	OpenStack	OpenStack, OPNFV, ONAP	OpenStack, Nutanix	OpenStack, OPNFV, ONAP	OpenStack, NFV		
No. Flujos Soportados		10 M/s	2.2 M/s	30 m/s	367 m/s			5.3 M/s (NOX-MT), 30 m/s (NOX Classic)	35 m/s	4.8 M/s		2.7 M/s	5 M/s	13.8 M/s	160 m/s	200 m/s			200 m/s	8 M/s							20 mil/s	2.5 M/s	2.3 M/s	1 M/s	106 m/s	2.5 M/s		300 m/s	
Código Abierto	Si	No	No	No	Si			Si	Si	Si	Si	Si	Si	Si	No	Si	Si	No	No	Si	Si	Si	No	Si	Si	No	No	Si	Si	No	Si	Si	No	No	No
Sistemas Operativos	Basados en Cloud	Linux		Linux	Linux			Linux (Ubuntu, Debian, Redhat, Gentoo)	Linux, MAC OS, Windows	Linux, MAC OS, Windows	Linux, MAC OS	Linux	Linux, MAC OS, Windows	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux	Linux, Windows, Android, MAC OS	Linux, MAC OS, Windows	Linux	Linux, MAC OS, Windows	Linux, MAC OS, Windows	Linux	Linux	
Multihilos	Si	Si	No	Si	Si			Si (NOX-MT)	No	Si		Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si	Si
Consistencia	No	Alta	Alta	Débil	Alta	No	Débil	No	No	No	Media	No	Alta	No	Media	No	Alta	Débil	Débil	No	Alta				No		No	NO	Débil	Alta	Débil	Media	Débil		
Ambientes de Uso	Redes pequeñas, Infraestructura en la nube,	Redes empresariales, Data center	Redes empresariales, Data center, Infraestructura en la nube	Redes pequeñas,	Redes pequeñas	Redes pequeñas	Redes pequeñas	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center	Redes pequeñas	Redes pequeñas	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la Nube	Redes empresariales	Redes empresariales	Redes empresariales	Redes empresariales	Redes pequeñas	Redes pequeñas	Redes empresariales, Data center, Infraestructura en la Nube	Redes empresariales, Infraestructura en la nube	Redes empresariales, Infraestructura en la nube	Redes empresariales, Infraestructura en la nube	Redes empresariales, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Data center, Infraestructura en la nube	Redes empresariales, Infraestructura en la nube	
Distribuido o Centralizado	Centralizado	Centralizado	Distribuido plano	Distribuido plano	Distribuido Plano	Distribuido jerárquico	Distribuido	Centralizado	Centralizado	Centralizado	Distribuido plano	Distribuido	Distribuido jerárquico	Centralizado	Distribuido jerárquico	Centralizado	Centralizado	Distribuido plano	Distribuido plano	Distribuido jerárquico	Distribuido plano	Distribuido plano	Distribuido plano	Centralizado	Distribuido	Distribuido		Centralizado	Centralizado	Distribuido	Distribuido Plano	Distribuido plano	Distribuido plano	Distribuido plano	
T. a Fallos	No	No	Si	Si	Si	No	Si	No	No	No	Si	No	Si	No	Si	No	No	Si	Si	Si	Si	Si	Si	No	Si	Si	Si	No	No	Si	Si	Si	Si	Si	Si
Fabricante	IBM T. J. Watson Research Center	SRI International	Nicra Networks, NTT OSRG y Google	Universidad de Toronto	Universidad de Lisboa	Carnegie Mellon University		Nicra Networks, Universidad de Stanford	Nicra Networks	Universidad de Rice	Universidad de Brown	Open Mul Foundation	ETRI	Universidad de Stanford	Dell	NEC Labs	NEC Labs	Thales Community and Security	Universidad de Colorado	Universidad de Toronto	ARCCN, Universidad estatal de Moscú	AGH Universidad de Ciencia y Tecnología, Polonia	IBM	Universidad de Wkatato, Nueva Zelanda	Juniper Networks	Juniper Networks	Ciena Inc	NTT OSRG, VA Linux	Big Switch Networks (Proyecto Floodlight)	Hewlett Packard	ON Lab	Fundación OpenDaylight (Múltiples fabricantes)	Big Switch Networks	Cisco Systems	Huawei Technologies

Documentación	Pobre	Pobre	Pobre	Pobre	Pobre	Pobre	Pobre	Buena	Buena	Buena	Pobre	Pobre	Buena	Buena	Buena	Buena	Buena	Pobre	Pobre	Pobre	Buena	Pobre	Buena	Buena	Pobre	Buena	Buena	Alta	Alta	Alta	Alta	Alta	Alta	Alta					
Licencia		Propietario	Propietario	Propietario	GPLv2L			GPLv3	GPLv3	LGPLv2.1	BSD 3.0	GPLv2	Apache 2.0	GPLv2	Propietaria	GPLv2	GPLv2	Propietaria	Propietaria	Apache 2.0	Apache 2.0	apache 2.0	Propietaria	Apache 2.0	Apache 2.0	Propietaria	Propietaria	Apache 2.0	Apache 2.0	Propietaria	Apache 2.0	EPL V1.0	Propietaria	Propietaria	Propietaria				
Lenguaje	Java	C	Python, C++, Java	C++	Java		Java	C++	Python	Java	Haskell	C	Java	Java	Linux	C, Ruby	C, Ruby	Java	C, C++	C++	Erlang	Java	Python	C++, Java, Python			Python	Java	Java	Java	Java	Java	Java	Java	Java				
Versión	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0		1.0	2.08	1.0		1.0	5.0	1.1									1.2	1.0				1.3	1.2	2.7.16	1.10.0	1.3	4.7	1.2	3.0
Página	No tiene	No tiene	No tiene	No tiene	No tiene	No tiene	No tiene	https://github.com/noxrepo/nox-classic/wiki	https://noxrepo.github.io/pox-dochtml	http://zhengcai.github.io/mastro-platform	http://pane.cs.brown.edu	http://www.openmum.org	https://openiris.etri.re.kr	https://openflow.stanford.edu/display/Beacon/Home.html	https://www.dell.com/learn/cotes/cobsd1/networking?&bsd&-ck=mn	https://tremagithub.io/trema	https://www.necam.com/PFlow	http://anr-discourse.lyon.fr	https://github.com/ngn-colorado/yanc	https://github.com/kandoo/bee-hive-netctrl	https://arcon.github.io/runos	https://github.com/FlowFoward/FlowFoward	https://www.ibm.com/services/network/software-defined	https://faucet.readthedocs.io/en/latest/intro.html ; https://github.com/faucetsdn/faucet	http://www.opencontrail.org	https://www.juniper.net/documentation/en_US/northstar3.0.0/information-products/pathway-pages/3.0/northstar-user-guide.html	https://www.blueplanet.com	http://osrg.github.io/ryu	http://www.projectfloodlight.org	http://h17007.www1.hp.com/ww1/hpe.com/us/en/networking/library/index.aspx?ca=sdn#XlsDWSHG2w	https://onosp.roject.org	https://www.opendaylight.org	https://www.bigswitch.com/tags/big-network-controller	https://www.cisco.com/c/en/us/products/cloud-systems-management/open-sdn-controller/index.html	https://www.huawei.com/en/products/enterprise-networking/sdn-controller				
Actualización Página	N/A	N/A	N/A	N/A	N/A	N/A	N/A	2012	2015	2015	2013	2015	2015	2017	2019	2017	2016	2017	2016	2016	2016	2016	2018	2019	2019	2019	2019	2019	2019	2019	2019	2019	2019	2019	2019	2019	2019		

3. DETERMINAR CUÁL SERIA EL MEJOR CONTROLADOR PARA REDES SDN A NIVEL EMPRESARIAL

En este capítulo se presenta información sobre la toma de decisiones y el proceso para determinar cuál es el controlador para redes SDN que mejor se adapte a las necesidades de una red empresarial usando en método PAJ (Procesos Analíticos Jerárquicos). En la sección 3.1, se presenta información general sobre la toma de decisiones. En la sección 3.2, se presenta información sobre la toma de decisiones multicriterio y el método PAJ. En la sección 3.3, se presenta el procedimiento para determinar cuál es el controlador para redes SDN que mejor se adapte a las necesidades de una red empresarial usando en método PAJ.

3.1 LA TOMA DE DECISIONES.

En la vida diaria todas las personas deben tomar decisiones, algunas de las cuales son tomadas sin mucho razonamiento del porqué se selecciona una u otra opción, por ejemplo, al elegir que ropa se debe usar para salir a la calle o por cual ruta se prefiere llegar a un destino, usualmente no se detienen mucho tiempo en analizar todas las posibilidades y sus consecuencias, debido a que estas decisiones no representan una gran relevancia en su diario vivir o en su futuro, por lo tanto no gastan mucho tiempo en analizar si han tomado efectivamente la decisión correcta. Pero en algunas ocasiones, deben tomar decisiones que pueden repercutir en sus vidas o en la vida de terceros de forma importante, por ejemplo, en el caso de seleccionar cuál carrera profesional estudiar, que tipo de carro comprar, en que negocio deben invertir, etc. Estas decisiones son tomadas frecuentemente después de realizar un análisis de las opciones y sus consecuencias a largo plazo, para luego seleccionar la que se considera mejor opción dentro del grupo de posibilidades; aunque se debe tener en cuenta que en algunas ocasiones la opción seleccionada termina no siendo la mejor, debido a que las personas suelen tomar decisiones sesgadas, bien sea debido a que sus creencias terminan favoreciendo alguna de las opciones, por falta de información sobre el tema o por no tener claro el objetivo de la decisión. Para evitar estos inconvenientes, la teoría de toma de decisiones se encarga de entregar un método lógico para seleccionar la mejor opción, a través de métodos matemáticos que posibilitan la toma de decisiones en ambientes impredecibles o con poca información relevante, entregando la

certeza de haber seleccionado la mejor opción, con base en el procedimiento realizado con la información con la que se cuenta en el momento [90].

Cuando se deben tomar decisiones complejas, como en el caso de algún proyecto de ingeniería, el decisor debe evaluar múltiples variables, conceptos y criterios, algunos de los cuales pueden estar en conflicto, como lo indico el economista Herbert Alexander Simon en 1955, “en las complejas organizaciones actuales, éstas no actúan intentando maximizar una determina función de utilidad, sino que se plantean distintos objetivos a la vez, la mayoría de los cuales son incompatibles entre sí, por lo que finalmente lo que se pretende es conseguir un determinado nivel de satisfacción para cada uno de ellos” [91], siguiendo este concepto nace la Toma de Decisiones Multicriterio (MCDM), que pretende abordar la toma de decisiones en un contexto de distintos objetivos y en un entorno incierto, contando con un procedimiento lógico que permite llegar a una determinación y poder sustentar que la decisión tomada es la más acorde a las características del problema, siendo esto necesario debido a:

- La complejidad del proyecto hace que sea imposible para el decisor tomar una decisión argumentada, debido a limitantes como la capacidad de memoria y atención, tomando en cuenta el hecho que, el ser humano solo puede realizar comparaciones de características entre máximo tres variables, realizando de forma natural y habitual comparaciones entre solo dos variables; con base en lo anterior, evaluar un proyecto con múltiples variables y alternativas excede el alcance del decisor.
- Ante la magnitud de un proyecto existe la tendencia a variar las metas, además de ser selectivo en la adquisición y procesamiento de la información pertinente, debido al sesgo natural que puede tener el decisor.
- La necesidad de tomar las mejores decisiones en el entorno empresarial, debido a la alta competitividad del sector, sumado a la legislación que exige cada vez mayor seguridad, calidad y rigor en los productos, documentos y determinaciones.

3.2 LA TOMA DE DECISIONES MULTICRITERIO

Es un área de investigación que viene desarrollándose desde mediados del siglo XX, la cual mediante el análisis matemático entrega herramientas como métodos, modelos y técnicas, a los tomadores de decisión para que puedan seleccionar la mejor opción evaluando múltiples variables, criterios de decisión, puntos de vista, entre otras informaciones; mejorando la efectividad, eficacia y eficiencia de las decisiones tomadas, su documentación y sustento. De esta investigación y análisis se desprenden métodos como La teoría de Utilidad Multiatributo (MAUT), el Proceso Analítico Jerárquico (PAJ), el Proceso Analítico de Red (ANP), el método Scoring, entre otros.

Entre los métodos de toma de decisión multicriterio más usados en diferentes áreas del conocimiento como gobierno, industria, negocios, salud, educación, entre otros; se encuentra el método de Procesos Analíticos Jerárquicos (PAJ), debido a su baja complejidad, su buena estructuración y a la posibilidad de considerar tanto aspectos cuantitativos como cualitativos al momento de realizar la selección, permitiendo de esta manera contar con elementos subjetivos de selección como la experiencia y el criterio de expertos en el tema [92].

3.2.1 Procesos Analíticos Jerárquicos (PAJ)

El Proceso de Análisis Jerárquico o AHP (Analytic Hierarchy Process), es un método de toma de decisiones introducido en el año 1977 por el matemático Iraquí Thomas L. Saaty, el cual está diseñado para resolver problemas de criterios múltiples de una manera eficiente, organizando la información del problema, descomponiéndolo y analizándolo por partes. De manera adicional, este proceso toma en cuenta datos cuantitativos y también cualitativos relativos a las alternativas de decisión, es decir, toma en cuenta la experiencia y las opiniones del tomador de decisión, siendo estos difíciles de incorporar en otros métodos de toma de decisión, debido a la complejidad de su cuantización; este proceso se realiza al solicitarle al tomador de la decisión o a un experto en el tema, que entregue evaluaciones subjetivas respecto a la importancia relativa de cada uno de los criterios de decisión y su nivel de preferencia sobre cada una de las alternativas, las cuales son procesadas matemáticamente, entregando como resultado, una jerarquización con prioridades que muestra las preferencias globales para cada una de las alternativas, en

caso de usar la ayuda de más de un experto, realiza una combinación y consenso de manera matemática de las opiniones, entregando un promedio geométrico de sus respuestas [93].

El proceso desarrollado por este método está dividido en 5 partes principales:

- **Modelado del problema:** el modelo PAJ para mejorar el entendimiento del problema, lo descompone en tres niveles o jerarquías:
 - **Nivel 1:** En este nivel superior, se ubica el objetivo general o meta del desarrollo del proceso.
 - **Nivel 2:** En este nivel intermedio, se ubican los criterios de decisión, los cuales a su vez pueden ser divididos en subcriterios para facilitar la comprensión del problema.
 - **Nivel 3:** En este nivel inferior, se ubican las alternativas u opciones de solución que presenta el problema.

Para explicar de forma más clara el proceso, usaremos un ejemplo del documento “Decisión con incertidumbre, decisión multicriterio y teoría de juegos” [94], el problema nos indica que se debe decidir sobre tres alternativas de rutas para una carretera, evaluándolas por su costo, impacto medioambiental y tiempo de ejecución; una vez se tiene la información del problema se procede a asignar los datos a cada nivel:

- **Nivel 1** (El propósito del problema): Trazado de una carretera.
- **Nivel 2** (Los criterios de decisión): Costo, Impacto Medioambiental, Tiempo de ejecución.
- **Nivel 3** (Las alternativas): Rutas A, B, C.

El modelado del problema puede ser graficado para mejorar el entendimiento y la estructuración del mismo, como se puede observar en la Figura 6, donde se muestra de manera adicional la ubicación de los subcriterios, en caso de que sean necesarios para la comprensión problema.

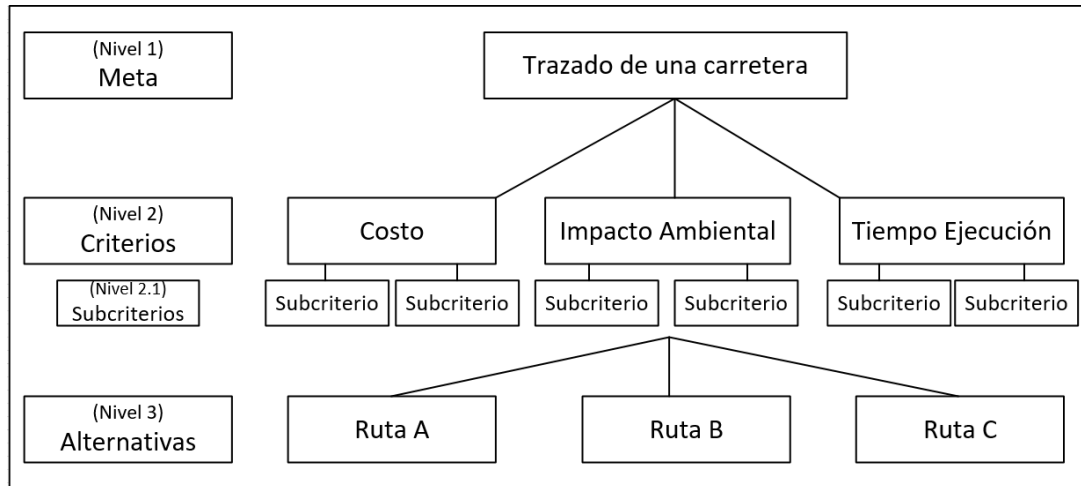


Figura 6 Jerarquía modelo PAJ [Elaboración propia]

- Sintetizar los juicios:** Paso seguido a la jerarquización del problema, se deben realizar comparaciones pareadas en cada nivel de jerarquía, tanto para los criterios de decisión entre sí, como para las alternativas de solución del problema, con el fin de escalar las preferencias relativas del decisor o experto entre dos elementos, uno de los cuales es evaluado y el otro es usado como elemento de comparación, buscando determinar la importancia o contribución de cada uno de los elementos al elemento de nivel superior al que se encuentran ligados. Esta valoración se realiza con base a datos concretos de los elementos o realizando juicios sobre la importancia o preferencia del elemento evaluado con relación al de comparación, para esto se construye una matriz de comparación, la cual se llena con base a una escala ya preestablecida, donde se usan valores enteros de 1 a 9 para determinar el nivel de importancia o preferencia del elemento evaluado contra el elemento de comparación, usando la mayoría de las veces solo los números impares, los números pares son usados para lograr mayor precisión, igualmente se pueden asignar valores inversos como $1/3$, $1/5$, $1/7$ y $1/9$, si el elemento evaluado tiene menor importancia con relación al elemento con el que es comparado. En la Tabla 4, se puede ver la escala numérica para valoración de Thomas L. Saaty.

Tabla 4 Escala de valoración de Saaty [Basado en [95]]

Escala Numérica	Escala Verbal	Explicación
1	Igual importancia	Los dos elementos contribuyen igualmente a la propiedad o criterio
3	Moderadamente más importante	La experiencia previa favorece al elemento evaluado sobre el de comparación
5	Frecuentemente más importante	La experiencia previa favorece fuertemente al elemento evaluado sobre el de comparación
7	Mucho más importante	El elemento evaluado domina el de comparación.
9	Importancia extrema	El elemento evaluado domina fuertemente el de comparación.

De forma matemática, la matriz de comparación es explicada de la siguiente manera, “Sea A una matriz $n \times n$, donde a_{ij} sea el elemento (i,j) de A, para $i = 1, 2, \dots, n$ y $j = 1, 2, \dots, n$. Decimos que A es la matriz de comparaciones pareadas de n alternativas, si a_{ij} es la medida de la preferencia del renglón i cuando se le compara con la alternativa de la columna j, cuando $i = j$, el valor de a_{ij} será igual a 1, pues se está comparando la alternativa consigo misma.” [96]

Con relación al ejercicio de ejemplo, se puede suponer que el decisor ya ha emitido sus juicios, entregando la siguiente matriz de comparaciones, Tabla 5, indicando en la primera fila que el criterio de decisión “Costo” tiene una valoración de 2 cuando es comparado con el criterio de decisión “Impacto Medioambiental” y tiene un valor de 5 cuando es comparado con el criterio de decisión “Tiempo de Ejecución”; igualmente en la segunda fila se observa que el criterio de decisión “Impacto Medioambiental” tiene una valoración de 3 cuando es comparado con el criterio de decisión “Tiempo de Ejecución”, los valores de las otras tres comparaciones pareadas son el inverso de las valoraciones anteriores.

Tabla 5 Matriz de Comparación de Criterios [tomado de [94]]

	Costo	Impacto medioambiental	Tiempo de ejecución
Costo	1	2	5
Impacto medioambiental	1/2	1	3
Tiempo de ejecución	1/5	1/3	1

- Establecer prioridades:** Como resultado del procesamiento matemático de las matrices de comparación realizadas en el proceso de sintetización, se pueden establecer los vectores de prioridades o pesos relativos de cada uno de los elementos, con base a la importancia relativa o contribución de cada uno de los elementos al elemento de nivel superior al que se encuentran ligados, teniendo en cuenta que la suma de las ponderaciones de cada nivel debe ser igual a 1.

Estos valores se encuentran con la solución del sistema $\frac{W_j}{W_i} = a_{ij}$, o lo que es lo mismo $W_i - (a_{ij} * W_j) = 0$, se debe tener en cuenta que estos sistemas no suelen tener una solución precisa dadas las inconsistencias del decisor, por lo tanto se usan las que más se aproximan; continuando con el ejercicio de ejemplo, el mismo se plantea matemáticamente con las siguientes ecuaciones, donde se tienen en cuenta las variables de desviación.

$$\min \sum_{i=1}^3 (n_i + P_i) \quad \text{Ecuación 1.}$$

$$w_1 - 2w_2 + n_1 - P_1 = 0 \quad \text{Ecuación 2.}$$

$$w_1 - 5w_3 + n_2 - P_2 = 0 \quad \text{Ecuación 3.}$$

$$w_2 - 3w_3 + n_3 - P_3 = 0 \quad \text{Ecuación 4.}$$

$$w_1 + w_2 + w_3 = 1 \quad \text{Ecuación 5.}$$

Solucionando las ecuaciones se obtienen los valores de los pesos para cada uno de los criterios de decisión, siendo el del criterio “costo” igual a $w_1 = 0.588$; el del criterio “impacto medioambiental” igual a $w_2 = 0.294$; y el del criterio “tiempo de ejecución” igual a $w_3 = 0.118$; teniendo variables de desviación $n_1 = n_2 = p_1 = p_2 = p_3 = 0$; $n_3 = 0.06$.

Luego de obtener los valores de los pesos de los criterios de decisión, debemos conocer la valoración de cada uno de los criterios de decisión en relación a cada una de las alternativas; en caso de no contar con datos específicos, se pueden recurrir nuevamente a las comparaciones pareadas, solicitándole al tomador de decisión o al experto, que nos entregue una valoración en una matriz de comparaciones por cada criterio de decisión, comparando las distintas alternativas de solución con respecto a ese criterio específico, estas matrices de comparación deben ser procesadas de forma similar como se realizó con la Matriz de Comparación de Criterios, Tabla 5. Continuando con el ejercicio de ejemplo, podemos suponer que el decisor ha emitido sus juicios entregando las siguientes matrices de comparaciones, Tabla 6, indicando en la tabla de comparación “Costo”, en la fila 1 que la alternativa A tiene una valoración de 6 en comparación con la alternativa B y una valoración de 3 en comparación con la alternativa C, es decir, la alternativa A es menos costosa que la alternativa B y la C, de manera similar se pueden revisar los valores de las otras dos matrices del ejercicio de ejemplo.

Tabla 6 Matrices de Comparación de Criterios en las Alternativas [tomado de [94]]

Costo	A	B	C
A	1	6	3
B	1/6	1	1/2
C	1/3	2	1

Impacto Medioamb.	A	B	C
A	1	1/9	1/5
B	9	1	2
C	5	1/2	1

Tiempo Ejecución	A	B	C
A	1	1/2	1/4
B	2	1	1/2
C	4	2	1

- Revisar la consistencia de los juicios:** Para validar las respuestas dadas en las matrices de comparación por los tomadores de decisión y/o los expertos, se debe determinar qué tan consistentes son las valoraciones dadas por ellos, tratando de evitar contradicciones, de tal forma que, si dentro de sus valoraciones $C1=2*C2$ y $C2=2*C3$, debe darse que $C1=4*C3$, pero si este valor es diferente, se dice que la matriz es inconsistente; normalmente el grado de consistencia de una matriz no es perfecto, debido tanto a la valoración del decisor como a la escala de valoración usada, por lo tanto, se establece un valor máximo de inconsistencia para la matriz comparación, el cual se encuentra mediante el siguiente proceso. Primero se debe conocer el índice de consistencia aleatorio (RI), el cual varía en función de la

dimensión de la matriz (n); este valor se obtuvo mediante la simulación de 100.000 matrices recíprocas generadas aleatoriamente, utilizando la escala de valoración de Saaty, entregando una escala que se puede ver en la Tabla 7.

Tabla 7 Índice de consistencia aleatorio [tomado de [95]]

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RI	0	0	0,525	0,882	1,115	1,252	1,341	1,404	1,452	1,484	1,513	1,535	1,555	1,57	1,583	1,595

También debemos conocer el índice de consistencia de la matriz, el cual se encuentra mediante la siguiente ecuación.

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad \text{Ecuación 6.}$$

Siendo λ_{\max} el resultado de la suma ponderada de cada celda por el vector prioridad de cada criterio evaluado, esta suma es dividida por el vector prioridad, sus resultados son sumados y a este valor se le saca la media aritmética. Teniendo los valores de RI y de CI, se puede encontrar el valor de la ratio de consistencia (CR), el cual se calcula con la siguiente ecuación.

$$CR = \frac{CI}{RI} \quad \text{Ecuación 7.}$$

Si este cociente es igual a 0,10 o menor, es decir, menor del 10%, entonces se acepta la consistencia de la matriz y por lo tanto los valores entregados por los expertos son validados; si el valor de este cociente es mayor al 10%, se deben revisar los juicios emitidos por los expertos y de ser necesario, se debe volver a realizar el proceso.

- **Decisión final:** Continuando con el proceso, una vez se tengan los pesos de cada uno de los criterios de decisión, al igual que las valoraciones de cada uno de los criterios de decisión con respecto a las alternativas, se procede a realizar una tabla con esta información; continuando con el ejercicio ejemplo, esta información se puede observar en la Tabla 8.

Tabla 8 Tabla de información criterios vs alternativas [tomado de [94]]

Criterios	Costo	Impacto medioambiental	Tiempo ejecución
Pesos	0.588	0.294	0.118
A	0.667	0.069	0.143
B	0.111	0.621	0.286
C	0.222	0.31	0.571

Por último, para determinar cuál de las alternativas es la mejor opción, se debe sumar los productos de los pesos de cada criterio de decisión por los valores de esos criterios en relación a cada alternativa; continuando con el ejercicio ejemplo, la suma de estos productos en la alternativa A es igual a:

$$(0.667 * 0.588) + (0.069 * 0.294) + (0.143 * 0.118) = 0.429 \quad \text{Ecuación 8.}$$

la suma de estos productos en la alternativa B es igual a:

$$(0.111 * 0.588) + (0.621 * 0.294) + (0.286 * 0.118) = 0.294 \quad \text{Ecuación 9.}$$

la suma de estos productos en la alternativa C es igual a:

$$(0.222 * 0.588) + (0.310 * 0.294) + (0.571 * 0.118) = 0.289 \quad \text{Ecuación 7.}$$

De esta manera se obtienen los valores globales de cada una de las alternativas, los cuales son comparados entre sí, siendo la alternativa más adecuada la que presente el valor más alto, con una diferencia de por lo menos el 10% sobre la anterior, de lo contrario sería considerado un empate estadístico; siguiendo con el ejercicio ejemplo, los valores globales para cada una de las alternativas son $A = 0.429$; $B = 0.294$ y $C = 0.289$; se logra determinar que la mejor opción es la alternativa A, seguida de la alternativa B y la alternativa C, las dos últimas se encuentran en empate estadístico en segundo lugar, por contar con una diferencia menor al 10% de la una con respecto a la otra.

El modelo PAJ permite dar un valor numérico a elementos diversos o difíciles de medir como la experiencia o el criterio de expertos en el tema, permitiéndole ser usado en procesos de selección, clasificación, priorización de alternativas, comparación de procesos

y gestión de calidad entre otros. El proceso matemático del modelo PAJ puede ser desarrollado mediante fórmulas en una tabla de Excel, con el fin de facilitar el proceso cuando se enfrenta un problema con gran cantidad de criterios de decisión a evaluar y múltiples alternativas de solución, siguiendo cada uno de los procesos básicos del mismo realizados en el ejercicio ejemplo [91].

3.3 DETERMINAR CUÁL ES EL CONTROLADOR PARA REDES SDN QUE MEJOR SE ADAPTE A LAS NECESIDADES DE UNA RED EMPRESARIAL

Para realizar este trabajo se decidió usar el método PAJ, debido a su baja complejidad, la posibilidad de trabajar con gran cantidad de información, además de poder usar la experiencia y juicio de expertos en el tema, de tal manera que se desarrolle el proceso de selección de la manera más fielmente posible a la usada en la realidad; de manera adicional, el proceso permite realizar una comprobación de la selección realizada, mediante el análisis de sensibilidad en el software Expert Choice.

Siguiendo las tendencias en el uso del método PAJ y para entregar un proceso más organizado, se ha decidido dividir el proceso en 7 pasos o niveles, que nos irán llevando a la solución del problema propuesto [95].

3.3.1 Determinación del objeto de estudio

El objeto de este estudio es seleccionar el controlador para redes SDN que mejor se adapte a las necesidades de una red empresarial, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura. Para ello, se usa de ejemplo una red empresarial con las siguientes características: es una red de gran tamaño, de tipo campus universitario, con más de 40 edificios, restaurantes, cafeterías, entre otros; la cual cuenta con nueve nodos distribuidos, con múltiples enlaces y cientos de puertos por nodo, para entregar acceso a servicios de internet y a servicios de la red interna, manejando tráfico del orden de 400 Gigabytes por segundo. Brindando servicio 24 horas al día, 365 días al año a estudiantes, administrativos, profesores e invitados, los cuales pueden sumar más de 10 mil personas por día usando su infraestructura [97].

En la Tabla 9, se puede observar la comparación entre las características de una red SDN empresarial y la red empresarial usada como ejemplo, encontrando que cumple con las características para poder implementar la infraestructura de redes SDN.

Tabla 9 Caracterización de la red empresarial usada como ejemplo [Elaboración Propia]

Características	Red SDN empresarial	Red SDN empresarial Ejemplo
tamaño	Redes grandes	Red de un campus universitario, con infraestructura propia y arrendada para conectar sus más de 40 edificios. (Red MAN)
implementación	Física	Física (alámbrica e inalámbrica)
Uso	Producción	Brindar acceso a internet y a otros servicios internos a administrativos, estudiantes, profesores e invitados.
Usuarios	Múltiples Usuarios	Más de 10 mil personas por día.
Requerimientos	Estabilidad, escalabilidad, seguridad, buen rendimiento, resiliencia, Buena documentación, entre otros.	Estabilidad, escalabilidad, seguridad, buen rendimiento, resiliencia, Buena documentación, manejar tráfico de más de 400 Gb por segundo, entre otros.

3.3.2 Elección de expertos

Siguiendo el proceso selección del controlador, se solicitó ayuda a 6 expertos en Redes Definidas por Software, teniendo en cuenta su generosidad para colaborar en el desarrollo del proceso y el conocimiento sobre el tema, el cual fue evaluado con base a la documentación encontrada en el proceso de búsqueda de información. Estando tres de los expertos vinculados a la empresa usada como ejemplo, los cuales serían consultados en caso de desarrollar un proyecto que involucre las Redes Definidas por Software u otro proyecto de telecomunicaciones, los otros tres expertos no están vinculados en la actualidad con la empresa usada como ejemplo, pero dos de ellos estuvieron vinculados anteriormente a la empresa en cuestión, a continuación, sus perfiles:

- Experto 1: Ingeniero Electrónico de la Universidad Nacional de Colombia, MSc en Ingeniería Electrónica con énfasis en Telecomunicaciones de la Pontificia Universidad Javeriana. Con experiencia en Administración de redes LAN en sistemas Alcatel-Lucent y HP, administración de servicios DHCP y DNS en AIX y

Windows Server 2019, administración de herramientas de gestión y monitoreo como Alcatel Omnivista, HP IMC, Observer y Zabbix. Gestión de DNS en plataforma F5 Big-IP, monitoreo de telefonía IP Nortel y Aruba Networks. Participación en proyectos de telecomunicaciones y gestión de servicios de usuario.

- Experto 2: Ingeniero Electrónico y MSc en Ingeniería Electrónica de la Pontificia Universidad Javeriana. Con experiencia en Redes de Datos, Telecomunicaciones, Sistemas Embebidos, Redes Ópticas de Transporte y Procesamiento A/D del audio.
- Experto 3: Ingeniero en Electrónica y Telecomunicaciones, y MSc en Telemática de la Universidad del Cauca. Con experiencia en Redes de Datos y Telecomunicaciones, Enrutamiento y Conmutación, Internet de las Cosas (IoT) y Gestión de Proyectos.
- Experto 4: Ingeniero Electrónico de la Pontificia Universidad Javeriana, MSc en Telecomunicaciones Ópticas y Tecnologías Fotónicas del Politécnico di Torino. Con experiencia en Venta de Soluciones, Infraestructura de Centros de Datos, Cableado Estructurado, Redes LAN, Sistemas de Voz y Seguridad, Liderazgo de Equipos y en Gestión de Proyectos.
- Experto 5: Ingeniero Electrónico y de Sistemas de la Pontificia Universidad Javeriana, MSc en Ciencias de la Computación de la Universidad de los Andes. Con experiencia en Redes de Datos, Sistemas Digitales y Desarrollo de Software.
- Experto 6: Ingeniera de Telecomunicaciones de la Universidad Santo Tomás, Especialista en Gerencia de Proyectos de Telecomunicaciones de la Universidad del Rosario, MSc en seguridad informática de la Universidad Abierta de Cataluña, España. Con experiencia en Telecomunicaciones, Formulación y Ejecución de Proyectos de Soluciones de Conectividad, Comunicaciones Unificadas y Centros de Datos.

3.3.3 Elección de las alternativas

Realizar la selección del controlador que mejor se adapte a las necesidades de una red empresarial, evaluando un número tan grande de alternativas, resulta inviable en términos

de recursos como tiempo, capacidad de cómputo, entre otros; sumado a lo anterior, la versión del software Expert Choice que se va a usar para realizar la validación de la selección del controlador, cuenta solo con la licencia de demostración debido al costo de su licenciamiento, por lo tanto viene con algunas restricciones. Debido a lo anterior, se debe realizar un proceso de selección de variables, de manera que se pueda realizar el estudio con un grupo más pequeño de controladores, los cuales deben estar vigentes y tener las mejores características, es por esto que se decidió realizar el proceso con los controladores Floodlight, HP SDN VAN Controller, ONOS, OpenDaylight, Big Network Controller, Cisco Open SDN Controller y Huawei SDN Agile Controller, ya que son controladores vigentes y cuentan con buena documentación, teniendo información en la mayoría de las variables que caracterizan los controladores SDN, como se puede observar en la Tabla 10.

Tabla 10 Alternativas de Controladores SDN [Elaboración Propia]

Características	Floodlight	HP SDN VAN Controller	ONOS	OpenDaylight	Big Network Controller	Cisco Open SDN Controller	Huawei SDN Agile Controller
V. Openflow	v 1.0; 1.1; 1.2; 1.3; 1.4; 1.5;	v 1.0; 1.3;	v 1.0; 1.3;	v 1.0; 1.3; 1.4;	v 1.0; 1.1; 1.2; 1.3; 1.4; 1.5;	V 1.0; 1.3;	V 1.0;
API (Sur)	Openflow; Indigo Agent and Oxygent (agente con librerías y lenguajes para acceder a múltiples switches).	Openflow; L3 Agent; L2 Agent	Openflow; Netconf; PCEP; OVSDB; BGP; P4; TL1; SNMP;	Openflow; Netconf; PCEP; OVSDB; BGP; P4; SNMP; YANG; IISP; COAT; LACP; Cisco Opflex; CAPWAP; OF-Config; SXP; USC; PCMM; COPS;	Openflow; BGP; OSPF;	Openflow; Netconf; PCEP; OVSDB; BGP; Cisco Opflex;	Openflow; Netconf; PCEP; OVSDB; BGP; SNMP; IGP;
API (Norte)	REST API; JAVA; RPC; Quantum;	REST API; JAVA API (OSGI);	REST API; OSGI framework;	REST API; JAVA API (OSGI); Restconf; XMPP; Maven;	REST API; JAVA	REST API	REST API
Interfaz	WEB GUI; CLI;	GUI (Shell)	WEB GUI; CLI;	WEB GUI; CLI;	WEB GUI; CLI;	WEB GUI;	WEB GUI; CLI;
Aplicación Enrutamiento	Enrutamiento; Virtual Switch; Balanceo de carga; Calidad de servicio; NAT; Balanceador de costos; DHCP; HAND (controlador de red basado en métricas); Proxy ARP; QoS; SE-Floodlight;	Enrutamiento (shortest path forwarding);	Enrutamiento;	Enrutamiento (shortest path forwarding); Balanceo de carga;	Enrutamiento; Virtual Switch; Balanceo de carga; Calidad de servicio; VLAN; NAT;	Enrutamiento; Balanceo de carga;	Enrutamiento; Balanceo de carga; Calidad de servicio; NAT;

Aplicación Medición	Monitoreo; Topología;	Monitoreo; Topología;	Monitoreo; Topología;	Monitoreo; Gestión de estadísticas; Gestión de equipos;	Monitoreo; Topología; Gestión de estadísticas;	Monitoreo; Topología; Gestión de estadísticas;	Monitoreo; Topología; Gestión de ancho de banda y de tráfico;
Aplicación Seguridad	Lista de Control de Acceso (ACL); Firewall; Autenticación, Autorización y seguimiento de aplicaciones; Detección de anomalías; ; Aislamiento de redes (VLAN); Recuperación de redes;	Lista de Control de Acceso (ACL); Autenticación de usuario y de aplicaciones; Sistemas de protección 2n+1;	Recuperación de fallos;	Autenticación, autorización y limitación de Administradores de red;	Autenticación de usuario; Recuperación de fallos; seguimiento de aplicaciones;	Autenticación de usuario; Seguimiento de comportamiento de usuario;	Autenticación y control de usuarios; Recuperación de fallos; IPsec VPN;
Integración (Nube y Virtualización)	OpenStack;	OpenStack;	OpenStack (Neutrón); OPNFV;	OpenStack (Neutrón); OPNFV; ONAP;	OpenStack; Nutanix;	OpenStack (Neutrón); OPNFV; ONAP;	OpenStack (Neutrón); NFV;
No. Flujos Soportados	2,5 Mill/s	2,3 Mill/s	1 Mill/s	106 mil/s	2,5 Mill/s		300 mil/s
C. F. Abierto	Si	No	Si	Si	No	No	No
Sistemas Operativos	Linux; MAC OS; Windows;	Linux;	Linux; MAC OS; Windows;	Linux; MAC OS; Windows;	Linux;	Linux;	Linux;
Multihilos	Si	Si	Si	Si	Si	Si	Si
Consistencia	No	Débil	Alta	Débil	Media	Débil	
Ambientes de Uso	Redes Empresariales; Data Center; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;	Redes Empresariales; Data Center; Infraestructura en la Nube;	Redes Empresariales; Data Center; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;
Distribuido o Centralizado	Centralizado	Distribuido	Distribuido Plano	Distribuido Plano	Distribuido	Distribuido Plano	Distribuido
T. a Fallos	No	Si	Si	Si	Si	Si	Si
Fabricante	Big Switch Networks (Project Floodlight);	HP	ON Lab	OpenDaylight Foundation (Apoyo Múltiples fabricantes)	Big Switch Networks	Cisco Systems	Huawei Technologies
Documentación	Buena	Alta	Alta	Alta	Alta	Alta	Alta
Licencia	Apache 2.0	Propietaria	Apache 2.0	EPL V1.0	Propietaria	Propietaria	Propietaria
Lenguaje	JAVA	JAVA	JAVA	JAVA	JAVA	JAVA	JAVA
Versión	1.2	2.7.16	1.10.0	1.3	4,7	1,2	3.0
Página	http://www.projectfloodlight.org/	http://h17007.www1.hp.com/usa/en/networking/library/index.aspx?cat=sdn#.XIsDWShKg2w	https://onosproject.org/	https://www.opendaylight.org/	https://www.bigswitch.com/tags/big-network-controller	https://www.cisco.com/c/en/us/products/cloud-systems-management/open-sdn-controller/index.html	https://e.huawei.com/es/products/enterprise-networking/sdn-controller

Actualización Pagina	2018	2019	2019	2019	2019	2019	2019
-------------------------	------	------	------	------	------	------	------

3.3.4 Elección de los criterios de decisión

Continuando con el proceso de selección del controlador que mejor se adapte a las necesidades propias de una red empresarial, se deben seleccionar los criterios de decisión relevantes al momento de seleccionar un controlador, para esto, se usó como base las características encontradas en la literatura que nos definen cada uno de los controladores. En un primer acercamiento se determinó que los criterios de decisión estarían dentro de las categorías de infraestructura, técnica y económica, sin embargo, para asegurar que estas tres categorías de criterios de decisión eran suficientes para caracterizar los controladores de redes SDN, se realizó una primera encuesta a los expertos, donde se les pregunto, si estaban de acuerdo con que los controladores SDN se pudieran definir bajo estas tres categorías o si se requería alguna adicional, dando como resultado, que dos de los seis expertos consideraron suficientes estas tres categorías, los otros cuatro indicaron que faltaban elementos adicionales, esto se puede ver en la Figura 7.

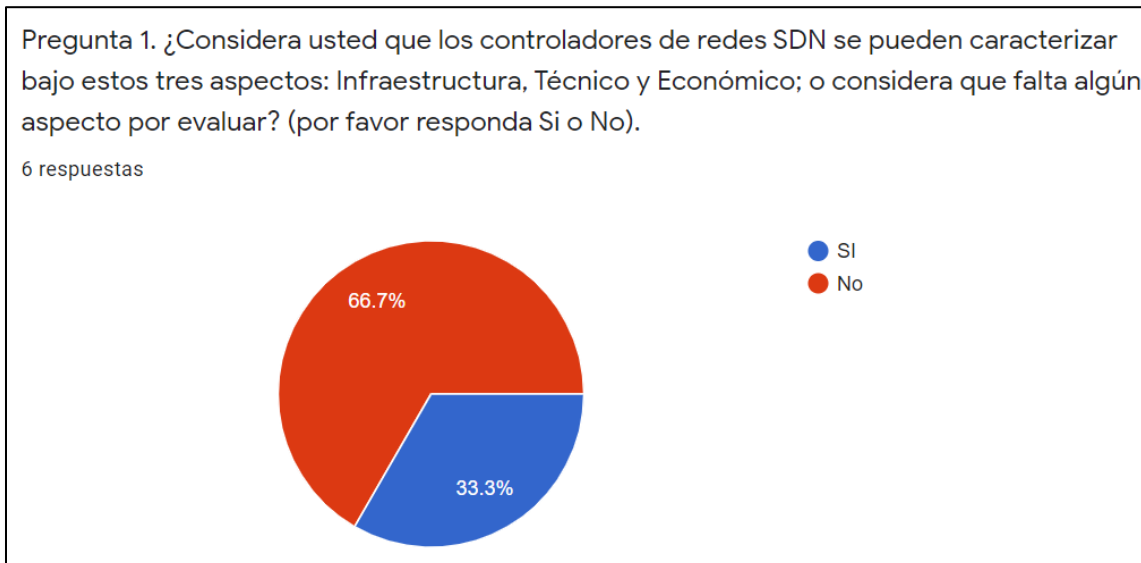


Figura 7 Respuesta sobre categorías de criterios de selección [Fuente google docs]

Dentro de las respuestas de los 4 expertos que consideraron que faltaban elementos para poder caracterizar los controladores de redes SDN, se encontró que estos elementos faltantes estaban considerados dentro de las características encontradas en la literatura sobre el tema, por lo cual se determinó que estas tres categorías de criterios son suficientes para poder caracterizar los controladores de redes SDN, estas consideraciones pueden verse en la Figura 8.

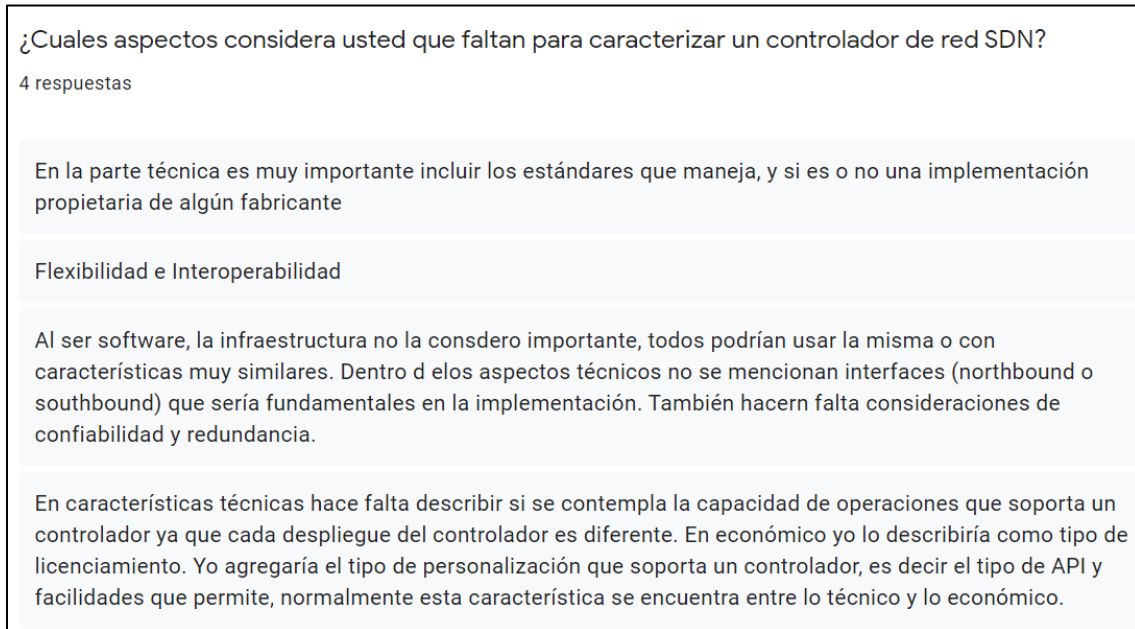


Figura 8 Consideraciones de los expertos sobre aspectos de caracterización [Fuente google docs]

Para determinar la relevancia de las características encontradas en la literatura al momento de caracterizar los controladores SDN, en la misma encuesta, en una segunda pregunta, se les pidió a los expertos indicar si consideraban estas características relevantes al momento de seleccionar un controlador de redes SDN, entregando como resultado, que en su mayoría las características eran relevantes en el concepto de los expertos, solo una de ellas fue descartada ya que cinco de los seis expertos encontraron que esta característica era irrelevante a la hora de seleccionar un controlador de redes SDN, esta característica es “versión del controlador”, esto se puede observar en la Figura 9, donde la opinión de los expertos se refleja en el tamaño de las barras de colores, cuando la característica es considerada relevante suma a la barra azul y si la característica es considerada irrelevante suma a la barra roja barra roja.

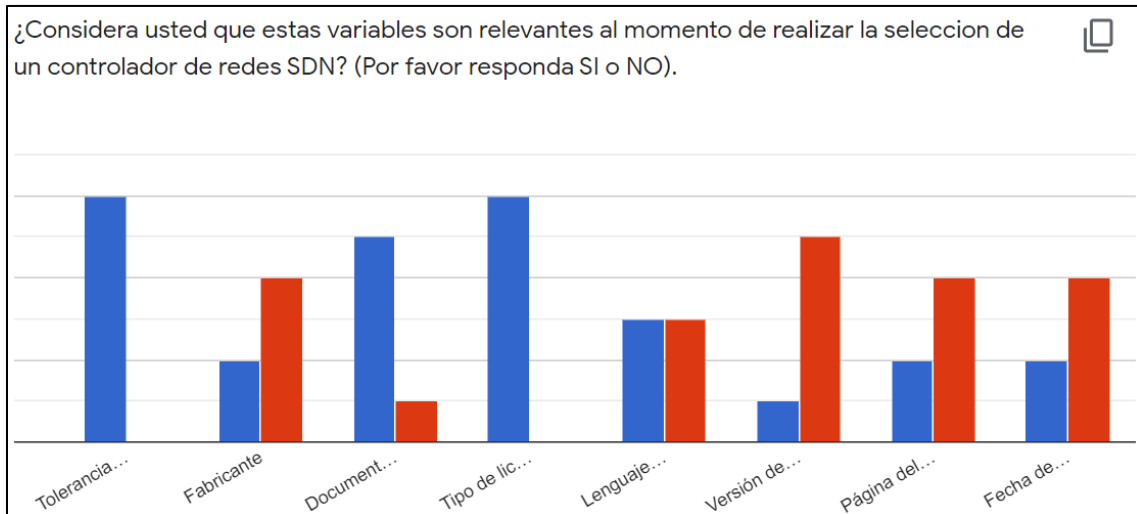


Figura 9 Características relevantes para seleccionar un controlador SDN [Fuente google docs]

Con base en lo anterior, se determinó que las otras 22 características sobre los controladores de redes SDN eran relevantes al momento de realizar una selección de los mismos; con el fin de hacer el proceso de selección más fácil, estas características o criterios de decisión se dividieron en las tres categorías de criterios, siendo estos:

- **Criterios técnicos:** Son los criterios de decisión de carácter técnico que tienen los controladores, informando sus capacidades, lenguaje de desarrollo, vigencia, etc.
- **Criterios económicos:** Son los criterios de decisión que indican si el controlador o su implementación trae consigo un gasto económico.
- **Criterios de infraestructura:** Son los criterios de decisión que indican cual es la infraestructura necesaria para implementar el controlador.

A su vez la categoría de criterios técnicos, fue subdividida en cinco en subcategorías, debido a que tiene muchos criterios de decisión asociados, estas son:

- **Servicios del controlador:** Son los criterios de decisión que indican que servicios ofrece el controlador.
- **Conexiones del controlador:** Son los criterios de decisión que indican los protocolos que soporta el controlador en sus comunicaciones hacia la capa de datos y la capa de aplicaciones.

- **Vigencia del controlador:** Son los criterios de decisión que nos indican si el proyecto del controlador aún se encuentra activo y cuál fue la fecha de su última actualización.
- **Características técnicas del controlador:** Son los criterios de decisión que indican cuales son las características técnicas específicas del controlador.
- **Desarrollo del controlador:** Son los criterios de decisión que indican aspectos específicos del desarrollo del controlador.

En la Tabla 11, se encuentran las 5 subcategorías en las que se dividieron los Criterios Técnicos de Decisión, y los Criterios Técnicos asociados a cada una de estas subcategorías.

Tabla 11 Criterios técnicos [Elaboración Propia]

Servicios del controlador	Conexiones del controlador	Vigencia del controlador	Características técnicas del controlador	Desarrollo del controlador
N° de Aplicaciones de enrutamiento	API hacia el sur	Página del Controlador	Consistencia de la Información	Fabricante
N° de Aplicaciones de Medición y Monitoreo	API hacia el norte	Fecha de Actualización de la Página	Versión del protocolo Openflow soportado	Lenguaje de Programación
N° de Aplicaciones de Seguridad y confianza			Tolerancia a Fallos	Documentación
Tipo de interfaz de usuario			Soporta Multihilos	
			Máximo Número de Flujos Soportados	

En la Tabla 12, se encuentran los criterios de decisión asociados a la categoría Criterios Económicos de Decisión.

Tabla 12 Criterios económicos [Elaboración Propia]

Criterios económicos del controlador		
código abierto o propietario	Sistemas Operativos	Tipo de licencia de uso (permisos de uso)

En la tabla 13, se encuentran los criterios de decisión asociados a la categoría Criterios de Infraestructura de Decisión.

Tabla 13 Criterios de infraestructura [Elaboración Propia]

Criterios de infraestructura		
Ambientes de uso	Integración con la Nube y Virtualización	Sistema Distribuido o Centralizado

En la Figura 10, se encuentra la organización jerárquica de los criterios de decisión, en la parte superior se observa la meta del ejercicio, que es encontrar el controlador SDN que mejor se adapte a las características de una red empresarial, esta parte de la jerarquía es también llamada nivel 1.

Un poco más abajo, se observan las tres categorías en las que se han dividido los criterios de decisión que caracterizan los controladores SDN, siendo estas categorías los Criterios Económicos, de Infraestructura y Técnicos, esta parte de la jerarquía es llamada Nivel 2 o Criterios Padres. Debajo de las categorías de los Criterios Económicos y de Infraestructura se pueden observar los criterios de decisión propios de cada categoría, debajo de la categoría Criterios Técnicos se tienen las 5 subcategorías: Servicios del controlador, Conexiones del controlador, Vigencia del Controlador, Características técnicas del controlador y Desarrollo del controlador, esta parte de la jerarquía es llamada Nivel 2.1 o Criterios Hijos. Debajo de estas subcategorías se pueden observar los criterios de decisión técnicos propios de cada subcategoría, esta parte de la jerarquía es llamada Nivel 2.2 o Criterios Nietos.

En la base de la jerarquía se pueden observar las 7 alternativas de controladores a evaluar, siendo esta parte el nivel 3.

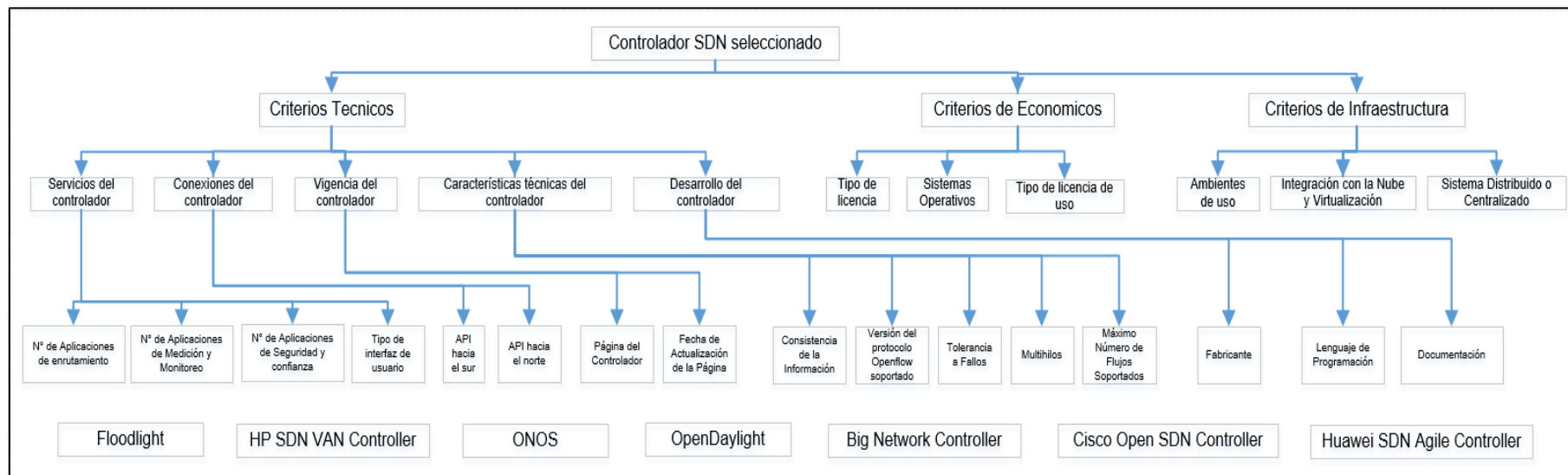


Figura 10 Criterios para seleccionar el controlador SDN [Elaboración Propia]

3.3.5 Análisis de dominancia

Continuando con el proceso de selección del controlador SDN, se debe realizar el análisis de dominancia, donde se evalúa cada una de las alternativas criterio a criterio, con el fin de identificar las alternativas que son dominadas al igual que las alternativas que no son dominadas y son candidatas a ser seleccionadas, es decir, mediante la cuantificación de los criterios de decisión de cada una de las alternativas y su análisis, se pretende encontrar una alternativa que sea superior en todos los criterios de decisión a las otras, de esta manera se puede determinar el mejor controlador SDN en esta fase del proceso. Este valor podría ser asignado por los expertos sobre el tema, realizando un proceso de matriz de decisión, pero para ese caso, cada uno de los expertos debería contar con información profunda sobre cada uno de los controladores seleccionados como alternativa, con el fin de realizar una valoración de cada uno de sus criterios con respecto a los demás controladores; de manera adicional, su conocimiento sobre un controlador en particular podría causar un sesgo en la valoración, por lo tanto, se determinó que la opción más viable era realizar una asignación de valores a los criterios de decisión, teniendo en cuenta el nivel de desarrollo de este criterio de decisión dentro del global de las alternativas, esto basado en la información obtenida a través de la investigación sobre los controladores SDN, la cual permitió realizar la Elección de las Alternativas en la sección 3.3.3.

En este proceso se realizó una valoración de cada criterio de decisión de forma individual, donde la alternativa que contaba con mayores componentes dentro de cada criterio de decisión obtenía la valoración más alta, y la alternativa que contaba con menores componentes dentro de cada criterio de decisión obtenía la valoración más baja; en los criterios de decisión donde todas las alternativas contaban con igual número de componentes se les dio un valor de "1" a todas y en los criterios de decisión donde la alternativas no contaba con componentes se les dio un valor de "0", el resultado de este proceso de valoración se puede observar en la tabla 14.

Tabla 14 Valoración de las Características de las Alternativas [Elaboración Propia]

Características	Floodlight	HP SDN VAN Controller	ONOS	OpenDaylight	Big Network Controller	Cisco Open SDN Controller	Huawei SDN Agile Controller
V. Openflow	v 1.0; 1.1; 1.2; 1.3; 1.4; 1.5;	v 1.0; 1.3;	v 1.0; 1.3;	v 1.0; 1.3 ; 1.4;	v 1.0; 1.1; 1.2; 1.3; 1.4; 1.5;	V 1.0; 1.3;	V 1.0;
Valoración	5	2	2	3	5	2	1
API (Sur)	Openflow; Indigo Agent and Oxygent (agente con librerías y lenguajes para acceder a múltiples switches).	Openflow; L3 Agent; L2 Agent	Openflow; Netconf; PCEP; OVSDB; BGP; P4; TL1; SNMP;	Openflow; Netconf; PCEP; OVSDB; BGP; P4; SNMP; YANG; IISP; COAT; LACP; Cisco Opflex; CAPWAP; OF-Config; SXP; USC; PCMM; COPS;	Openflow; BGP; OSPF;	Openflow; Netconf; PCEP; OVSDB; BGP; Cisco Opflex;	Openflow; Netconf; PCEP; OVSDB; BGP; SNMP; IGP;
Valoración	1	1	3	5	1	2	3
API (Norte)	REST API; JAVA; RPC; Quantum;	REST API; JAVA API (OSGI);	REST API; OSGI framework;	REST API; JAVA API (OSGI); Restconf; XMPP; Maven;	REST API; JAVA	REST API	REST API
Valoración	4	2	2	5	2	1	1
Interfaz	WEB GUI; CLI;	GUI (Shell)	WEB GUI; CLI;	WEB GUI; CLI;	WEB GUI; CLI;	WEB GUI;	WEB GUI; CLI;
Valoración	2	1	2	2	2	1	2
Aplicación Enrutamiento	Enrutamiento; Virtual Switch; Balanceo de carga; Calidad de servicio; NAT; Balanceador de costos; DHCP; HAND (controlador de red basado en métricas); Proxy ARP; QoS; SE-Floodlight;	Enrutamiento (shortest path forwarding);	Enrutamiento;	Enrutamiento (shortest path forwarding); Balanceo de carga;	Enrutamiento; Virtual Switch; Balanceo de carga; Calidad de servicio; VLAN; NAT;	Enrutamiento; Balanceo de carga;	Enrutamiento; Balanceo de carga; Calidad de servicio; NAT;
Valoración	5	1	1	2	4	2	3
Aplicación Medición	Monitoreo; Topología;	Monitoreo; Topología;	Monitoreo; Topología;	Monitoreo; Gestión de estadísticas; Gestión de equipos;	Monitoreo; Topología; Gestión de estadísticas;	Monitoreo; Topología; Gestión de estadísticas;	Monitoreo; Topología; Gestión de ancho de banda y de tráfico;
Valoración	2	2	2	3	3	3	3

Aplicación Seguridad	Lista de Control de Acceso (ACL); Firewall; Autenticación, Autorización y seguimiento de aplicaciones; Detección de anomalías; ; Aislamiento de redes (VLAN); Recuperación de redes;	Lista de Control de Acceso (ACL); Autenticación de usuario y de aplicaciones; Sistemas de protección 2n+1;	Recuperación de fallos;	Autenticación, autorización y limitación de Administradores de red;	Autenticación de usuario; Recuperación de fallos; seguimiento de aplicaciones;	Autenticación de usuario; Seguimiento de comportamiento de usuario;	Autenticación y control de usuarios; Recuperación de fallas; IPsec VPN;
Valoración	5	3	1	2	3	2	3
Integración (Nube y Virtualización)	OpenStack;	OpenStack;	OpenStack (Neutrón); OPNFV;	OpenStack (Neutrón); OPNFV; ONAP;	OpenStack; Nutanix;	OpenStack (Neutrón); OPNFV; ONAP;	OpenStack (Neutrón); NFV;
Valoración	1	1	2	3	2	3	3
No. Flujos Soportados	2,5 Mill/s	2,3 Mill/s	1 Mill/s	106 mil/s	2,5 Mill/s	N/A	300 mil/s
Valoración	5	5	3	1	5	0	1
C. F. Abierto	Si	No	Si	Si	No	No	No
Valoración	1	0	1	1	0	0	0
Sistemas Operativos	Linux; MAC OS; Windows;	Linux;	Linux; MAC OS; Windows;	Linux; MAC OS; Windows;	Linux;	Linux;	Linux;
Valoración	3	1	3	3	1	1	1
Multihilos	Si	Si	Si	Si	Si	Si	Si
Valoración	1	1	1	1	1	1	1
Consistencia	No	Débil	Alta	Débil	Media	Débil	N/A
Valoración	0	1	5	1	3	1	0
Ambientes de Uso	Redes Empresariales; Data Center; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;	Redes Empresariales; Data Center, Infraestructura en la Nube;	Redes Empresariales; Data Center; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;	Redes Empresariales; Infraestructura en la Nube;
Valoración	3	2	3	3	2	2	2
Distribuido o Centralizado	Centralizado	Distribuido	Distribuido Plano	Distribuido Plano	Distribuido	Distribuido Plano	Distribuido
Valoración	1	1	1	1	1	1	1
T. a Fallos	No	Si	Si	Si	Si	Si	Si
Valoración	0	1	1	1	1	1	1
Fabricante	Big Switch Networks (Project Floodlight);	HP	ON Lab	OpenDaylight Fundation (Apoyo Multiples fabricantes)	Big Switch Networks	Cisco Systems	Huawei Technologies
Valoración	1	1	1	1	1	1	1
Documentación	Buena	Alta	Alta	Alta	Alta	Alta	Alta
Valoración	4	5	5	5	5	5	5
Licencia	Apache 2.0	Propietaria	Apache 2.0	EPL V1.0	Propietaria	Propietaria	Propietaria
Valoración	1	1	1	1	1	1	1
Lenguaje	JAVA	JAVA	JAVA	JAVA	JAVA	JAVA	JAVA
Valoración	1	1	1	1	1	1	1
Pagina	http://www.projectfloodlight.org/	http://h17007.www1.hp.com/usa/en/networking/library/index.aspx?cat=sdn#.XlsDWShKg2w	https://onosproject.org/	https://www.opendaylight.org/	https://www.bigswitch.com/tags/big-network-controller	https://www.cisco.com/c/en/us/products/cloud-systems-management/open-sdn-controller/index.html	https://e.huawei.com/es/products/enterprise-networking/sdn-controller

Valoración	1	1	1	1	1	1	1
Actualización Pagina	2018	2019	2019	2019	2019	2019	2019
Valoración	1	2	2	2	2	2	2
Valoraciones Máximas	8	4	8	11	8	5	6
Porcentaje V. Máximas	16,0%	8,0%	16,0%	22,0%	16,0%	10,0%	12,0%
Puntaje Global	48	36	44	48	47	34	37
Porcentaje P. Global	16,3%	12,2%	15,0%	16,3%	16,0%	11,6%	16,3%

Con base en este proceso se determinó que, ninguna de las alternativas predomina con la valoración más alta en todos los criterios de decisión, algunas alternativas comparten en algunos criterios la valoración más alta y todas tienen a lo menos 4 criterios donde poseen las valoraciones más altas (en este conteo no se tuvieron en cuenta los criterios donde todas las alternativas tienen igual número de elementos y cuentan con la misma valoración); por lo tanto, no se puede determinar cuál es el mejor controlador en esta fase del proceso, pero si se puede visualizar que existen alternativas que tienen mejor puntuación y pueden ser opcionadas a ser elegidas, como en el caso de los controladores Opendaylight, Floodlight, Big Network Controller, ONOS y Huawei SDN Agile Controller.

3.3.6 Ponderación de los criterios

La ponderación de los criterios se realizó aplicando la media geométrica a las respuestas entregadas en las comparaciones pareadas por los expertos, este método es usado debido a que la media geométrica permite entregar una valoración bastante objetiva en términos de discrepancias, ya que minimiza el impacto de las valoraciones sesgadas o muy distantes, teniendo en cuenta la posibilidad que algún experto valore un criterio de decisión que no sea de su completo dominio; los resultados de estos procesos se pueden observar en las tablas siguientes. El proceso desarrollado para obtener estos resultados se puede observar más detenidamente en [91].

En la tabla 15, se observa la media geométrica de las valoraciones entregadas por los expertos en las comparaciones pareadas, de manera adicional se puede observar el vector prioridad para cada una de las categorías de criterios de decisión, siendo para la categoría

de criterios técnicos de 0,71, para la categoría de criterios económicos de 0,154 y para la categoría de criterios de infraestructura de 0,136, es decir, una consideración de importancia de 71%, 15,4% y 13,6% respectivamente, la suma de estos ponderados cubre el 100% de la asignación posible. Esta misma información sobre los otros niveles de criterios de decisión se pueden encontrar en las tablas 16, 17, 18, 19, 20, 21, 22 y 23.

Tabla 15 Ponderación Criterios Padres (nivel 2) [Elaboración propia]

	Infraestructura	Técnica	Económica	Vector prioridad
Infraestructura	1,000	0,220	0,755	0,136
Técnica	4,545	1,000	5,433	0,710
Económica	1,325	0,184	1,000	0,154
Suma Columnas				1,000

Tabla 16 Ponderación Criterios Económicos (nivel 2.1) [Elaboración propia]

	Código abierto o propietario	Sistemas operativos	Tipo de licencia de uso	Vector prioridad
Código abierto o propietario	1,000	2,466	2,498	0,539
Sistemas operativos	0,405	1,000	2,265	0,291
Tipo de licencia de uso	0,400	0,442	1,000	0,169
Suma Columnas				1,000

Tabla 17 Ponderación Criterios de Infraestructura (nivel 2.1) [Elaboración propia]

	Ambientes de uso	Integración con la nube	Sistema distribuido o centralizado	Vector prioridad
Ambientes de uso	1,000	3,672	3,512	0,622
Integración con la nube	0,272	1,000	2,542	0,244
Sistema distribuido o centralizado	0,285	0,393	1,000	0,134
Suma Columnas				1,000

Tabla 18 Ponderación Criterios Técnicos (Nivel 2.1) [Elaboración propia]

	Servicios controlador	Conexiones controlador	Vigencia controlador	Características técnicas controlador	Desarrollo del controlador	Vector prioridad
Servicios controlador	1,000	4,561	4,223	3,260	3,225	0,456
Conexiones controlador	0,219	1,000	2,466	1,817	2,513	0,191
Vigencia controlador	0,237	0,405	1,000	1,442	1,698	0,120
Características técnicas controlador	0,307	0,550	0,693	1,000	4,539	0,157
Desarrollo del controlador	0,310	0,398	0,589	0,220	1,000	0,076
Suma Columnas						1,000

Tabla 19 Ponderación Servicios controlador (nivel 2.2) [Elaboración propia]

	Aplicaciones de enrutamiento	Aplicaciones de medición y monitoreo	Aplicaciones de seguridad y confianza	Tipo de interfaz de usuario	Vector prioridad
Aplicaciones de enrutamiento	1,000	3,595	2,904	2,542	0,481
Aplicaciones de medición y monitoreo	0,278	1,000	1,587	2,667	0,223
Aplicaciones de seguridad y confianza	0,344	0,630	1,000	2,513	0,185
Tipo de interfaz de usuario	0,393	0,375	0,398	1,000	0,111
Suma Columnas					1,000

Tabla 20 Ponderación Conexiones controlador (nivel 2.2) [Elaboración propia]

	API hacia el sur	API hacia el norte	Vector prioridad
API hacia el sur	1,000	3,460	0,776
API hacia el norte	0,289	1,000	0,224
Suma Columnas			1,000

Tabla 21 Ponderación Vigencia controlador (nivel 2.2) [Elaboración propia]

	Tiene página el controlador	Fecha de actualización de la página	Vector prioridad
Tiene página el controlador	1,000	4,460	0,817
Fecha de actualización de la página	0,224	1,000	0,183
Suma Columnas			1,000

Tabla 22 Ponderación Características técnicas controlador (nivel 2.2) [Elaboración propia]

	Consistencia de la información	Versión del protocolo Openflow soportado	Tolerancia a fallos	Soporta multihilos	Máximo número de flujos soportados	Vector prioridad
Consistencia de la información	1,000	2,542	3,267	2,884	2,140	0,374
Versión del protocolo Openflow soportado	0,393	1,000	2,542	2,335	2,570	0,237
Tolerancia a fallos	0,306	0,393	1,000	2,402	3,026	0,173
Soporta multihilos	0,347	0,428	0,416	1,000	2,942	0,129
Máximo número de flujos soportados	0,467	0,389	0,330	0,340	1,000	0,087
Suma Columnas						1,000

Tabla 23 Ponderación Desarrollo del controlador (nivel 2.2) [Elaboración propia]

	Fabricante	Lenguaje de programación	Documentación	Vector prioridad
Fabricante	1,000	2,080	3,267	0,536
Lenguaje de programación	0,481	1,000	3,267	0,333
Documentación	0,306	0,306	1,000	0,131
suma				1,000

En el desarrollo del proceso, se revisó la inconsistencia de cada una de las matrices de decisión de las comparaciones pareadas entregadas por los expertos, del mismo modo, se revisó la inconsistencia de las matrices de la media geométrica de sus respuestas, verificando que en ningún caso superara el 10% de inconsistencia. En la tabla 24, se puede observar un ejemplo de este proceso.

Tabla 24 Ratio de Inconsistencia de las matrices evaluadas [Elaboración propia]

λ_{max}	3,024
CI	0,012
RI	0,525
CR	2%

3.3.7 Cálculo de la prioridad global en el conjunto de alternativas

Teniendo en cuenta que todos los criterios y subcriterios estuvieron valorados a maximizar, no se requiere realizar ningún proceso de convergencia de los mismos, caso contrario si alguno de ellos hubiera sido valorado a minimizar. El siguiente paso a seguir, es hacer el proceso de normalización, el cual se ha realizado mediante dos métodos que conservan la proporcionalidad, el método distribuido y el método ideal.

El método distribuido, divide la valoración de cada uno de los criterios de cada una de las alternativas en la suma de las valoraciones de todas las alternativas de ese mismo criterio, luego ese valor es multiplicado por los vectores prioridad de cada uno de los criterios y subcriterios relacionados; como ejemplo, el valor de 0,043 dado a las aplicaciones de

enrutamiento de la alternativa Floodlight, que se encuentra con asterisco (*) en la Tabla 24, es el resultado de la valoración individual de ese criterio en esa alternativa (5), dividido por la suma de los valores de ese criterio para todas la alternativas (18), este resultado fue multiplicado por el vector prioridad de las aplicaciones de enrutamiento (0,481), de los servicios del controlador (0,456) y de los criterios técnicos (0,71). Los resultados de los criterios de cada alternativa son sumados, entregando un valor de prioridad global para cada una de las alternativas, los cuales se pueden observar en la parte inferior de la Tabla 25.

Tabla 25 Calculo prioridad global (método distributivo) [Elaboración propia]

Nivel 1	Nivel 2	Nivel 3	Floodlight	HP SDN VAN Controller	ONOS	Opendaylight	Big Network Controller	Huawei SDN Agile Controller	Cisco Open SDN Controller
Criterios de infraestructura	Ambientes de uso		0,015	0,010	0,015	0,015	0,010	0,010	0,010
	Integración con la nube y virtualización		0,002	0,002	0,004	0,007	0,004	0,007	0,007
	Sistema distribuido o centralizado		0,003	0,003	0,003	0,003	0,003	0,003	0,003
Criterios económicos	Código abierto o propietario		0,028	0,000	0,028	0,028	0,000	0,000	0,000
	Sistemas operativos		0,010	0,003	0,010	0,010	0,003	0,003	0,003
	Tipo de licencia de uso		0,004	0,004	0,004	0,004	0,004	0,004	0,004
Criterios técnicos	servicios del controlador	Aplicaciones de enrutamiento	0,043*	0,009	0,009	0,017	0,035	0,026	0,017
		Aplicaciones de medición y monitoreo	0,008	0,008	0,008	0,012	0,012	0,012	0,012
		Aplicaciones de seguridad y confianza	0,016	0,009	0,003	0,006	0,009	0,009	0,006
		Tipo de interfaz de usuario	0,006	0,003	0,006	0,006	0,006	0,006	0,003
	Conexiones del controlador	API hacia el sur	0,007	0,007	0,020	0,033	0,007	0,020	0,013
		API hacia el norte	0,007	0,004	0,004	0,009	0,004	0,002	0,002
	Vigencia del controlador	Página del controlador	0,010	0,010	0,010	0,010	0,010	0,010	0,010
		Fecha de actualización de la página	0,001	0,002	0,002	0,002	0,002	0,002	0,002
	Capacidades técnicas del controlador	Consistencia de la información	0,000	0,004	0,019	0,004	0,011	0,000	0,004
		Versión del protocolo Openflow Soportado	0,007	0,003	0,003	0,004	0,007	0,001	0,003
		Tolerancia a fallos	0,000	0,003	0,003	0,003	0,003	0,003	0,003
		Soporte Multihilos	0,002	0,002	0,002	0,002	0,002	0,002	0,002

		Máximo número de flujos soportados	0,002	0,002	0,001	0,000	0,002	0,000	0,000
	Desarrollo del controlador	Fabricante	0,004	0,004	0,004	0,004	0,004	0,004	0,004
		Lenguaje de programación	0,003	0,003	0,003	0,003	0,003	0,003	0,003
		Documentación	0,001	0,001	0,001	0,001	0,001	0,001	0,001
Resultado Global			0,178	0,095	0,161	0,183	0,142	0,128	0,112

El método ideal, divide la valoración de cada uno de los criterios de cada una de las alternativas en la valoración más alta de todas las alternativas para ese mismo criterio, luego ese valor es multiplicado por los vectores prioridad de cada uno de los criterios y subcriterios; como ejemplo, el valor de 0,156 dado a las aplicaciones de enrutamiento de la alternativa Floodlight, que se encuentra con asterisco (*) en la Tabla 26, es el resultado de la valoración individual de ese criterio en esa alternativa (5), dividido por la valoración más alta de ese criterio para todas la alternativas (5), este resultado fue multiplicado por el vector prioridad de las aplicaciones de enrutamiento (0,481), de los servicios del controlador (0,456) y de los criterios técnicos (0,71). Los resultados de los criterios de cada alternativa son sumados, y luego son normalizados, al ser divididos los resultados de cada una de las alternativas en la suma de los resultados de todas las alternativas, este proceso entrega un valor de prioridad global para cada una de las alternativas, los cuales se pueden observar en la parte inferior de la Tabla 26.

Tabla 26 Calculo prioridad global (método ideal) [Elaboración propia]

Nivel 1	Nivel 2	Nivel 3	Floodlight	HP SDN VAN Controller	ONOS	Opendaylight	Big Network Controller	Huawei SDN Agile Controller	Cisco Open SDN Controller
Criterios de infraestructura	Ambientes de uso		0,084	0,056	0,084	0,084	0,056	0,056	0,056
	Integración con la nube y virtualización		0,011	0,011	0,022	0,033	0,022	0,033	0,033
	Sistema distribuido o centralizado		0,018	0,018	0,018	0,018	0,018	0,018	0,018
Criterios económicos	Código abierto o propietario		0,083	0,000	0,083	0,083	0,000	0,000	0,000
	Sistemas operativos		0,045	0,015	0,045	0,045	0,015	0,015	0,015
	Tipo de licencia de uso		0,026	0,026	0,026	0,026	0,026	0,026	0,026
Criterios técnicos	servicios del controlador	Aplicaciones de enrutamiento	0,156*	0,031	0,031	0,062	0,125	0,093	0,062
		Aplicaciones de medición y monitoreo	0,048	0,048	0,048	0,072	0,072	0,072	0,072

		Aplicaciones de seguridad y confianza	0,060	0,036	0,012	0,024	0,036	0,036	0,024
		Tipo de interfaz de usuario	0,036	0,018	0,036	0,036	0,036	0,036	0,018
	Conexiones del controlador	API hacia el sur	0,021	0,021	0,063	0,105	0,021	0,063	0,042
		API hacia el norte	0,024	0,012	0,012	0,030	0,012	0,006	0,006
	Vigencia del controlador	Página del controlador	0,070	0,070	0,070	0,070	0,070	0,070	0,070
		Fecha de actualización de la página	0,008	0,016	0,016	0,016	0,016	0,016	0,016
	Capacidades técnicas del controlador	Consistencia de la información	0,000	0,008	0,042	0,008	0,025	0,000	0,008
		Versión del protocolo Openflow Soportado	0,026	0,011	0,011	0,016	0,026	0,005	0,011
		Tolerancia a fallos	0,000	0,019	0,019	0,019	0,019	0,019	0,019
		Soporte Multihilos	0,014	0,014	0,014	0,014	0,014	0,014	0,014
		Máximo número de flujos soportados	0,010	0,010	0,006	0,002	0,010	0,002	0,000
	Desarrollo del controlador	Fabricante	0,029	0,029	0,029	0,029	0,029	0,029	0,029
		Lenguaje de programación	0,018	0,018	0,018	0,018	0,018	0,018	0,018
		Documentación	0,006	0,007	0,007	0,007	0,007	0,007	0,007
	Suma Columnas			0,794	0,494	0,712	0,819	0,674	0,636
Resultado Global			0,169	0,105	0,152	0,174	0,144	0,135	0,120

3.3.8 Análisis de resultados e informe final

Con Base en los resultados del proceso de cálculo de la prioridad global en el conjunto de alternativas, el cual tiene en cuenta las valoraciones de los criterios de cada una de las alternativas, al igual que, los vectores de prioridad entregados por la media geométrica de las respuestas de los expertos, se puede indicar que la alternativa que obtiene la mejor puntuación siguiendo el método distribuido de normalización, es Opendaylight con un 0,183, es decir, un 18,3% del 100% de la asignación posible, seguido por Floodlight con un 17,8% y ONOS con un 16,1%, en otras palabras, el controlador para redes SDN que mejor se adapta a las necesidades de la red empresarial ejemplo, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura es el controlador Opendaylight. Pero se debe tener en cuenta que, para tener un resultado definitivo, las alternativas deben estar separadas al menos un 10% entre sí, de lo contrario se considera un empate estadístico. En la Tabla 27, se observa que las puntuaciones del controlador Opendaylight y del controlador Floodlight tienen solo un 2,73% de diferencia, por lo cual, se encuentran en un empate estadístico en el primer lugar, técnicamente el mismo caso sucede en el segundo

lugar entre el controlador Floodlight y el controlador ONOS, al igual que en el cuarto lugar entre el controlador Big Switch Networks y el controlador Huawei SDN Agile Controller, pero en estos casos la diferencia está más cercana al valor límite, por lo cual se puede desestimar y dejar sus posiciones según su puntuación.

Tabla 27 Resultados prioridad global (método distribuido) [Elaboración propia]

	Alternativa	prioridad	Menos 10%	% diferencia
1	Opendaylight	0,183	0,165	2,73%
2	Floodlight	0,178	0,160	9,55%
3	ONOS	0,161	0,145	11,8%
4	Big Network Controller	0,142	0,128	9,86%
5	Huawei SDN Agile Controller	0,128	0,115	12,5%
6	Cisco Open SDN Controller	0,112	0,101	15,17%
7	HP SDN VAN Controller	0,095	N/A	N/A

Siguiendo el método ideal de normalización, Tabla 28, se observa que la alternativa que obtiene la mejor puntuación es Opendaylight con un 0,174, es decir, un 17,4% del 100% de la asignación posible, seguida por Floodlight con un 16,9% y ONOS con un 15,2%; en otras palabras, el controlador para redes SDN que mejor se adapta a las necesidades de la red empresarial ejemplo, teniendo en cuenta condicionantes técnicos, económicos y de infraestructura es el controlador Opendaylight. Pero igual que en caso anterior, la diferencia en la puntuación entre el controlador Opendaylight y el controlador Floodlight es de solo 2,87%, por lo tanto, se considera que se encuentran en un empate estadístico en el primer lugar, igual sucede en tercer lugar entre el controlador ONOS y el controlador Big Switch Networks, y en el cuarto lugar entre el controlador Big Switch Networks y el controlador Huawei SDN Agile Controller.

Tabla 28 Resultados prioridad global (método ideal) [Elaboración propia]

	Alternativa	prioridad	Menos 10%	% diferencia
1	Opendaylight	0,174	0,157	2,87%
2	Floodlight	0,169	0,152	10%
3	ONOS	0,152	0,137	5,26%
4	Big Network Controller	0,144	0,13	6,25%
5	Huawei SDN Agile Controller	0,135	0,121	11,1%
6	Cisco Open SDN Controller	0,120	0,108	12,5%
7	HP SDN VAN Controller	0,105	N/A	N/A

Los resultados de los procesos de normalización por los métodos distribuido e ideal son similares, entregando el mismo orden de puntuación, igualmente coinciden en que los controladores Opendaylight y Floodlight se encuentran en un empate estadístico en el primer lugar. Como último paso del método PAJ, se hace necesario realizar un análisis de sensibilidad, con el fin de establecer la robustez de los resultados presentados y resaltar los puntos fuertes de los controladores seleccionados en primer y segundo puesto de puntuación.

4. ANÁLISIS DE SENSIBILIDAD A LA SELECCIÓN REALIZADA BAJO EL SOFTWARE EXPERT CHOICE

En este capítulo se presenta información sobre el proceso de análisis de sensibilidad de la selección realizada en el capítulo anterior, usando para ello el software Expert Choice. En la sección 4.1, se presenta el proceso de ingreso de datos al programa Expert Choice. En la sección 4.2, se presenta el proceso de análisis de sensibilidad con variaciones menores al 10%. En la sección 4.3, se presenta el proceso de análisis de sensibilidad con variaciones mayores al 10%. En la sección 4.4, se presenta un análisis de los puntos fuertes de cada alternativa en una comparación entre pares.

El proceso de análisis de sensibilidad tiene como objeto confirmar que los resultados del estudio son fruto de la robustez del proceso y no del azar, además de entregar una solución concluyente al proceso matemático visto en el capítulo anterior, para ello se usa el software especializado para toma de decisiones de nombre Expert Choice; siendo este un paquete informático que permite implementar la metodología PAJ, después de ser alimentado con la información entregada en las matrices de decisión por los expertos y la valoración de los criterios de cada alternativa, entregando como resultado una priorización de las alternativas, usando tanto el método distribuido como el método ideal.

Luego este escenario es sometido al análisis de sensibilidad, en el cual se realiza la variación en un valor menor al 10% de uno de los vectores de prioridad, se analiza como este cambio afecta los otros pesos de los criterios de decisión y la priorización de las alternativas, entregando la información de este nuevo escenario tanto de manera numérica como gráfica, representándola en una gráfica de sensibilidad de valoración, de sensibilidad dinámica, de sensibilidad de gradiente, de sensibilidad de diferencias ponderadas y un gráfico en dos dimensiones. Además, entrega la posibilidad de realizar un análisis de cada una de las alternativas en comparación directa con otra.

4.1 PROCESO DE ASIGNACIÓN DE PRIORIDADES GLOBALES MEDIANTE EL SOFTWARE EXPERT CHOICE

El programa Expert Choice como primer paso, entrega el modelo jerárquico del problema de decisión, dando la posibilidad de observar errores en su configuración, encontrando de

esta manera en la parte superior de la jerarquía, el objeto de estudio del problema, del cual se desprenden los nodos jerárquicos o criterios de decisión, de estos nodos a su vez se desprenden subnodos o subcriterios (en caso de contar con ellos), en la parte inferior o base de la jerarquía, se pueden encontrar las alternativas evaluadas. En la Figura 11, se observa el modelo jerárquico de este problema en particular, encontrando en la parte superior el objeto de estudio del problema, nivel 1, de él se desprenden las tres categorías de criterios de decisión, dando lugar al nivel 2, de cada una de estas categorías de criterios de decisión se desprenden subcriterios que dan lugar a los niveles 2.1 y 2.2, finalmente en la base de la jerarquía se encuentran las 7 alternativas evaluadas en este proceso, nivel 3.

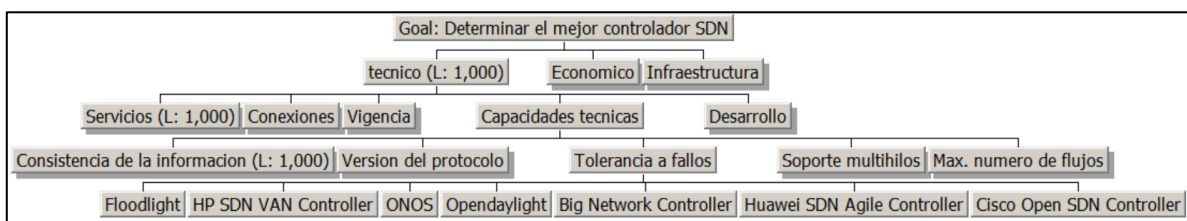


Figura 11 Modelo jerárquico de Expert Choice [Fuente simulador]

Continuando con el proceso, se deben ingresar al programa los valores de las matrices de decision entregadas por los expertos y la informacion de la valoracion de los criterios para cada una de las alternativas; estos valores son procesados y como resultado son entregados los pesos relativos de cada uno de los criterios, corroborando en este punto, que los calculos realizados en el capitulo anterior estan correctos, teniendo una variacion minima, del orden de decimales.

En la Figura 12, se observa los pesos relativos de cada uno de los criterios de decisión, teniendo la informacion del valor del peso relativo de cada criterio con respecto a los otros criterios de su mismo subnivel (L: local) y con respecto a todos los otros criterios (G:Global), cuyo valor resulta de multiplicar el valor de su peso local por el valor del peso global del criterio superior, es decir, del que se desprende.

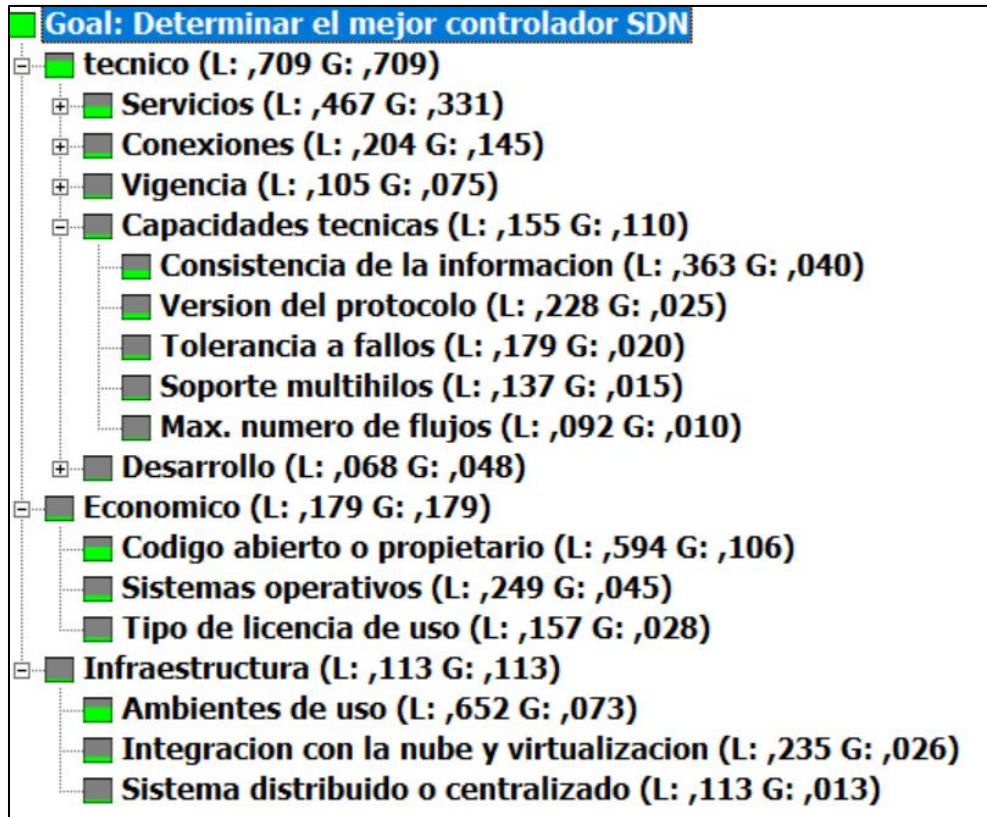


Figura 12 Vectores de prioridad Expert Choice [Fuente simulador]

El siguiente paso, es el calculo y la normalizacion de las prioridades globales de las alternativas, el cual se puede realizar mediante el metodo distribuido o el metodo ideal; en la Figura 13, se ve el resultado del proceso realizado mediante el metodo distribuido, entregando un calculo de prioridades similar al realizado en el capitulo anterior a traves de este mismo metodo, variando sus valores solo en decimales, ademas nos indica un valor de inconsistencia total del 9%.

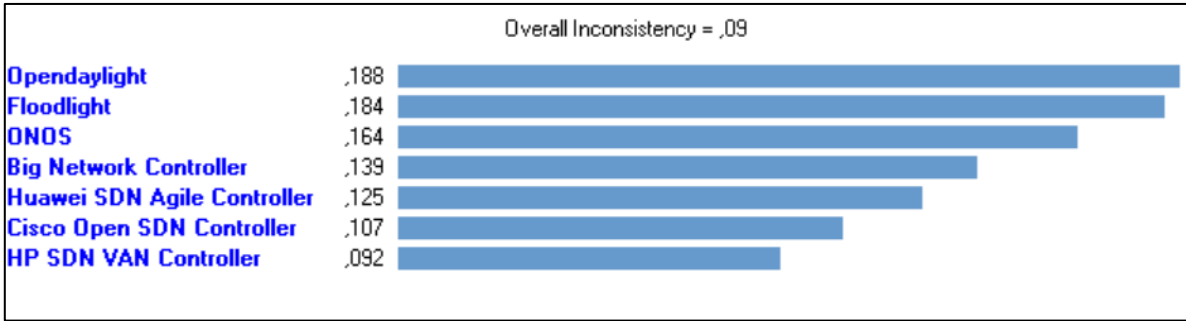


Figura 13 Calculo de la prioridad global por Expert Choice (método distribuido) [Fuente simulador]

En la Figura 14, se ve el resultado del proceso realizado mediante el método ideal, entregando igualmente un calculo de prioridades globales de las alternativas similar al realizado en el capítulo anterior mediante este mismo metodo, variando sus valores solo en decimales, tambien nos indica un valor de inconsistencia total de 9%.

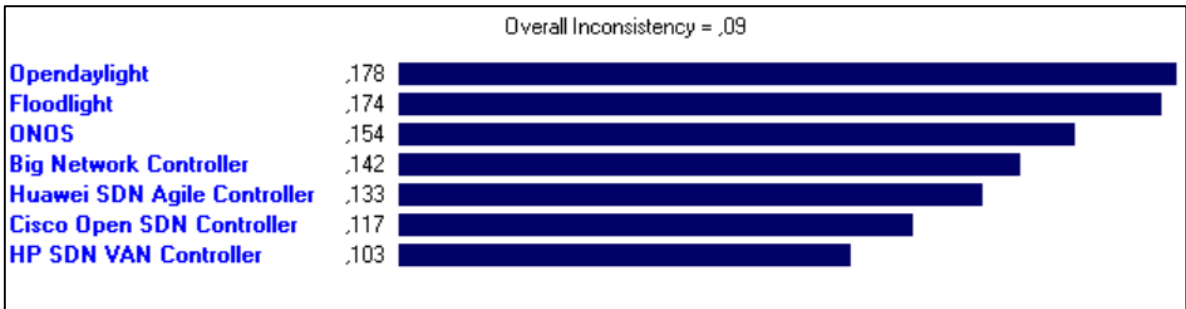


Figura 14 Calculo de la prioridad global por Expert Choice (método ideal) [Fuente simulador]

Ambos procesos indican que el controlador seleccionado debe ser el controlador Opendaylight, seguido respectivamente por los controladores Floodlight, ONOS, Big Network Controller, Huawei SDN Agile Controller, Cisco Open SDN Controller y HP SDN VAN Controller; también se puede ver que la diferencia entre el primer y segundo lugar es menor al 10%, por lo cual están empatados estadísticamente. Estos resultados son similares a los entregados por el proceso matemático realizado en el capítulo anterior, entregando el mismo orden de prioridades globales a las alternativas, dando mayor confianza a los resultados obtenidos.

Igualmente, el software Expert Choice permite ver gráficamente el desempeño de cada una de las alternativas con respecto a cada una de las categorías de criterios de decisión; en la

Figura 15, se observa que los controladores Opendaylight, Floodlight y ONOS, se destacan como primeros en las tres categorías de criterios de decisión evaluadas, teniendo un mejor consolidado el controlador Opendaylight, seguido muy de cerca por el controlador Floodlight, que tiene una menor puntuación en el criterio de infraestructura, en el tercer lugar el controlador ONOS, tiene una mejor puntuación que el controlador Floodlight en el criterio de infraestructura pero tiene una puntuación más baja en el criterio técnico, el cual tiene un alto grado de preferencia para los expertos consultados.

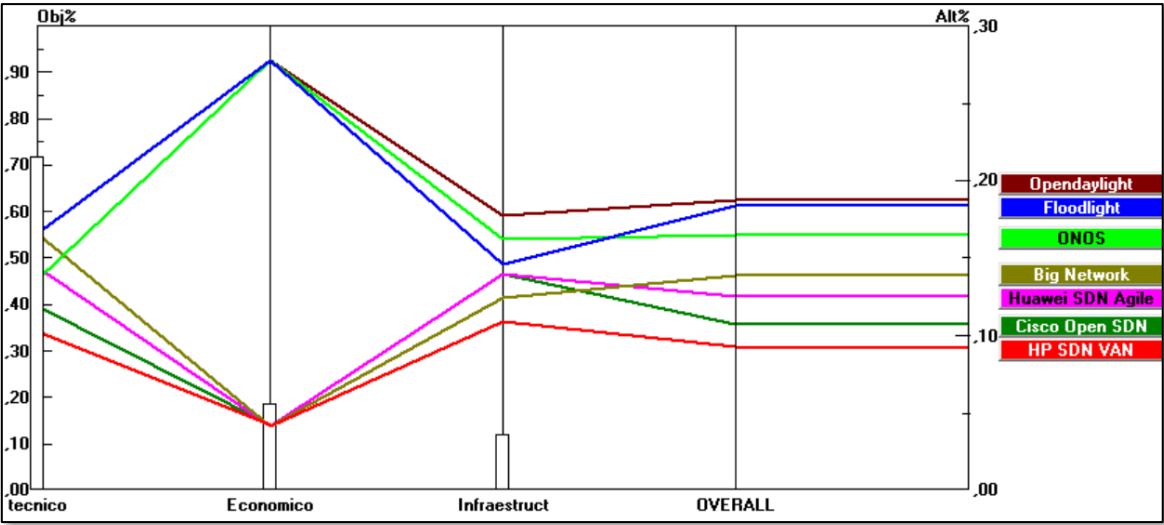


Figura 15 Análisis de rendimiento (modo distribuido) [Fuente simulador]

4.2 SENSIBILIDAD A LA VARIACIÓN MENOR AL 10%

Usando el software Expert Choice, mediante la función grafica de sensibilidad dinámica, se realizó una variación de +/- 10% sobre los valores de las categorías de los criterios de decisión, con el fin de revisar si estos cambios generaban una reorganización de las alternativas. En la Figura 16, se observa el análisis de sensibilidad dinámico del resultado sin variaciones, donde se puede observar el aporte de cada una las categorías de criterios de decisión a la valoración de cada alternativa, encontrando que las tres alternativas mejor valoradas cuentan con un sobresaliente aporte de la categoría económica con respecto a las demás.

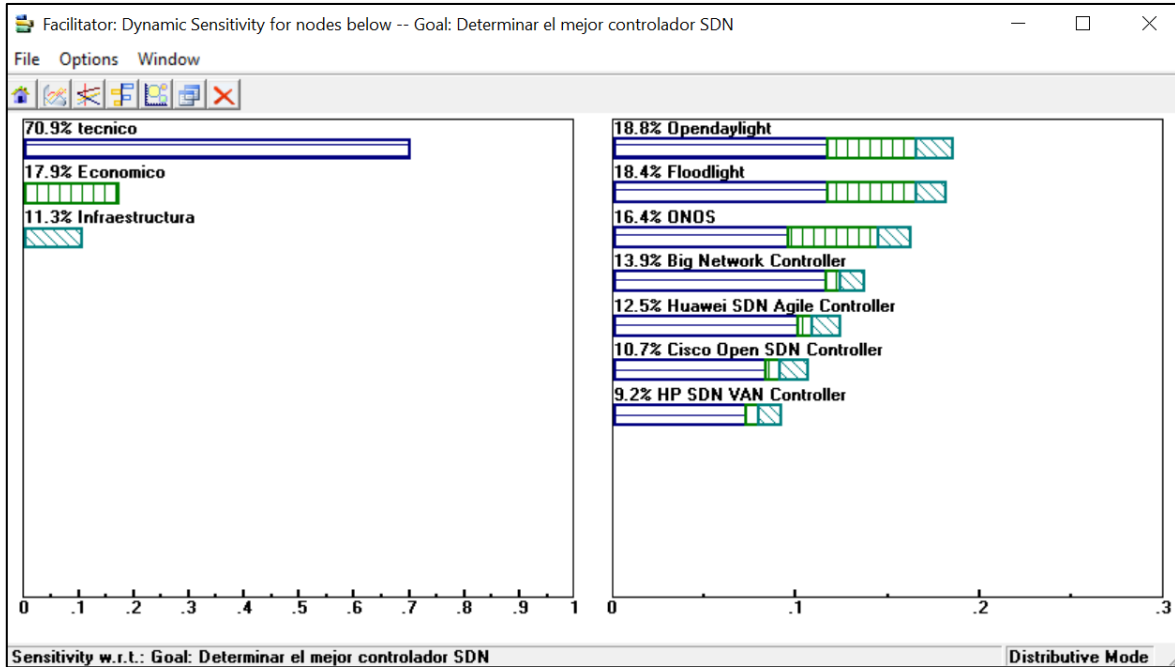


Figura 16 Análisis de sensibilidad dinámico del resultado (modo distribuido) [Fuente simulador]

En la Figura 17, se observa el analisis de sensibilidad dinamico realizado sobre la Categoría Tecnico. En la Figura 18, se observa el analisis de sensibilidad dinamico realizado sobre la Categoría Economico. En la Figura 19, se puede observar el analisis de sensibilidad dinamico realizado sobre la Categoría Infraestructura.

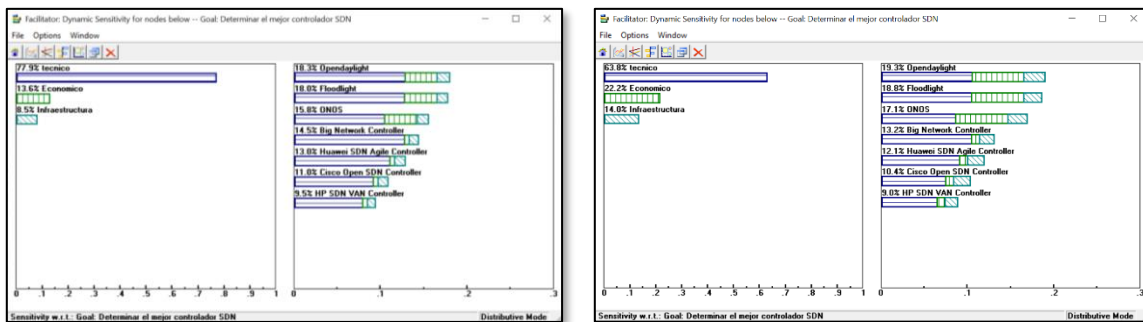


Figura 17 Análisis de sensibilidad dinámico de la Categoría Técnico (+/- 10%) [Fuente simulador]

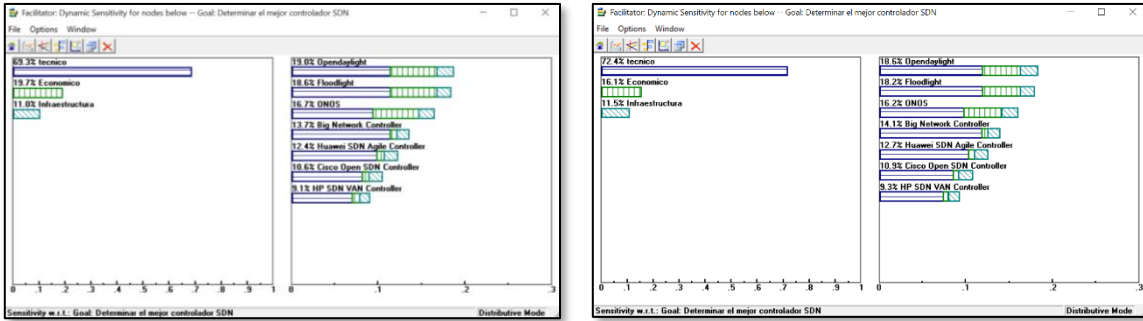


Figura 18 Análisis de sensibilidad dinámico de la Categoría Económico (+/- 10%) [Fuente simulador]

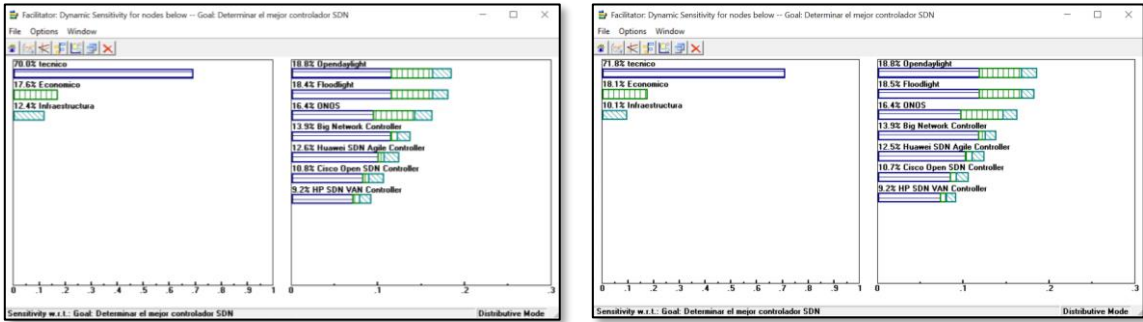


Figura 19 Análisis de sensibilidad dinámico de la Categoría Infraestructura (+/- 10%) [Fuente simulador]

En la Tabla 29, se observa el consolidado del análisis de sensibilidad dinámico, encontrando que a pesar de existir variaciones en las valoraciones de las alternativas, continúan conservando el orden, siendo el controlador OpenDaylight el mejor valorado, seguido de cerca por el controlador Floodlight, con una diferencia menor al 10%; estos resultados certifican la robustez del proceso realizado al igual que de la solución encontrada.

Tabla 29 Consolidado de análisis de sensibilidad (Modo distribuido 10%) [Elaboración propia]

Criterios				Alternativas						
Variación	Técnico	económico	infraestructura	OpenDaylight	Floodlight	ONOS	Big Network Controller	Huawei SDN Agile Controller	Cisco Open SDN Controller	HP SDN VAN Controller
0%	70.9%	17.9%	11.3%	18.8%	18.4%	16.4%	13.9%	12.5%	10.7%	9.2%
+10%	77.9%	13.6	8.6%	18.3%	18.0%	15.8%	14.5%	13.0%	11.0%	9.5%
-10%	63.8%	22.2%	14.0%	19.3%	18.8%	17.1%	13.2%	12.1%	10.4%	9.0%
+10%	69.3%	19.7%	11.0%	19.0%	18.6%	16.7%	13.7%	12.3%	10.6%	9.1%
-10%	72.4%	16.1%	11.5%	18.6%	18.2%	16.2%	14.1%	12.7%	10.9%	9.3%
+10%	70.0%	17.6%	12.4%	18.8%	18.4%	16.4%	13.9%	12.6%	10.8%	9.2%
-10%	71.8%	18.1%	10.1%	18.8%	18.5%	16.4%	13.9%	12.5%	10.7%	9.2%

4.3 SENSIBILIDAD A LA VARIACIÓN MAYOR AL 10%

Continuando con el proceso, se realizaron variaciones mayores al 10% con el fin de sacar conclusiones mucho más robustas, modificando los valores de las categorías de los criterios de decisión hasta un +/- 50%. En la Figura 20, se observa el análisis de sensibilidad dinámico realizado sobre la Categoría Técnico. En la Figura 21, se observa el análisis de sensibilidad dinámico realizado sobre la Categoría Económico. En la Figura 22, se observa el análisis de sensibilidad dinámico realizado sobre la Categoría Infraestructura.

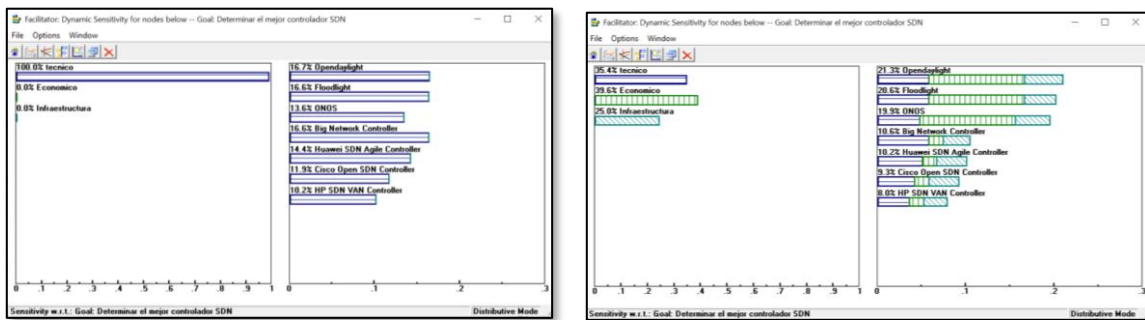


Figura 20 Análisis de sensibilidad dinámico de la Categoría Técnico (+/- 50%) [Fuente simulador]

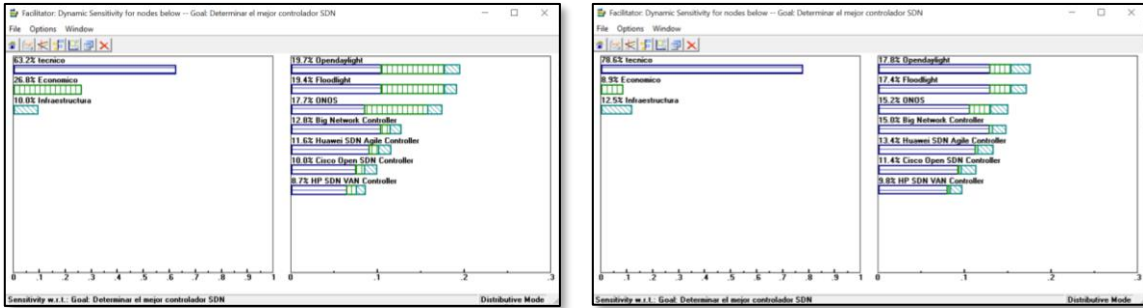


Figura 21 Análisis de sensibilidad dinámico de la Categoría Económico (+/- 50%) [Fuente simulador]

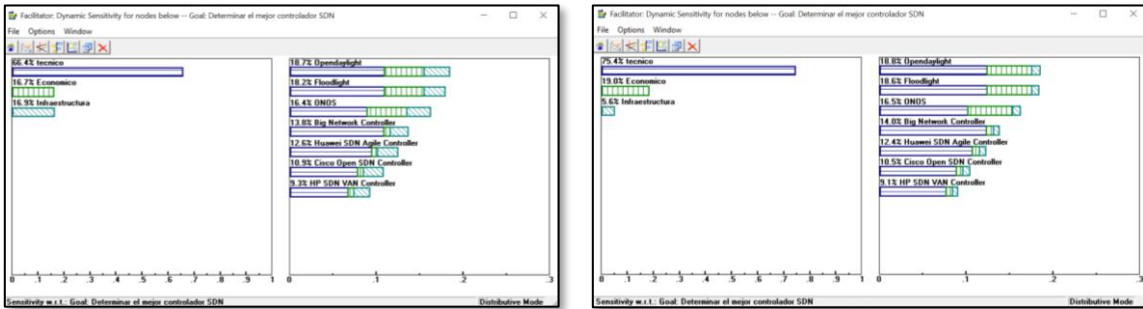


Figura 22 Análisis de sensibilidad dinámico de la Categoría Infraestructura (+/- 50%) [Fuente simulador]

En la Tabla 30, se observa el consolidado del análisis de sensibilidad dinámico, con una modificación de hasta el 50% en las valoraciones de las categorías de criterios de decisión, donde se encuentra que en todos los escenarios los controladores Opendaylight y Floodlight mantienen el primer y segundo lugar respectivamente. Solo en el escenario donde se modifica el valor de la categoría del criterio técnico hasta en un 50% adicional, se obtiene una variación en el orden, donde el Controlador ONOS a partir de una modificación de +11,5%, pierde su posición de tercer puesto con el controlador Big Network Controller; luego a partir de una modificación de +22.5%, el controlador ONOS pierde nuevamente su posición con el controlador Huawei SDN Agile Controller, indicando que estos controladores dos controladores tienen mejores características técnicas. Llegando al final del escenario, el controlador Big Network Controller iguala la valoración del controlador Floodlight. De esta manera se corrobora la robustez del proceso realizado al igual que de la solución encontrada.

Tabla 30 Consolidado de análisis de sensibilidad (Modo distribuido >10%) [Elaboración propia]

Criterios				Alternativas						
Variación	Técnico	económico	infraestructura	OpenDaylight	Floodlight	ONOS	Big Network Controller	Huawei SDN Agile Controller	Cisco Open SDN Controller	HP SDN VAN Controller
0%	70.9%	17.9%	11.3%	18.8%	18.4%	16.4%	13.9%	12.5%	10.7%	9.2%
+50%	100%	0%	0%	16.7%	16.6%	13.6%	16.6%	14.4%	11.9%	10.2%
-50%	35.4%	39.6%	25.0%	21.3%	20.6%	19.9%	10.6%	10.2%	9.3%	8.0%
+50%	63.2%	26.8%	10.0%	19.7%	19.4%	17.7%	12.8%	11.6%	10.0	8.7%
-50%	78.6%	8.9%	12.5%	17.8%	17.4%	15.2%	15.0%	13.4%	11.4%	9.8%
+50%	66.4%	16.7%	16.9%	18.7%	18.2%	16.4%	13.8%	12.6%	10.9%	9.3%
-50%	75.4	19.0	5.6%	18.8%	18.6%	16.5%	14.0%	12.4%	10.5%	9.1%

4.4 COMPARACIÓN DIRECTA ENTRE CONTROLADORES

Entre las características del software Expert Choice, entrega la posibilidad de realizar una comparación directa entre alternativas, permitiendo de esta manera evaluar los puntos fuertes de cada una. En la Figura 23, se observa un comparativo entre el controlador OpenDaylight y el controlador Floodlight, los cuales quedaron en primer y segundo lugar respectivamente en la selección realizada, con valoraciones tan similares que técnicamente se encuentran en un empate estadístico, en la figura se observa que en las categorías de criterios técnicos y económicos no tienen diferencia, solo existe una pequeña diferencia en la categoría de criterios de infraestructura que termina favoreciendo al controlador OpenDaylight.

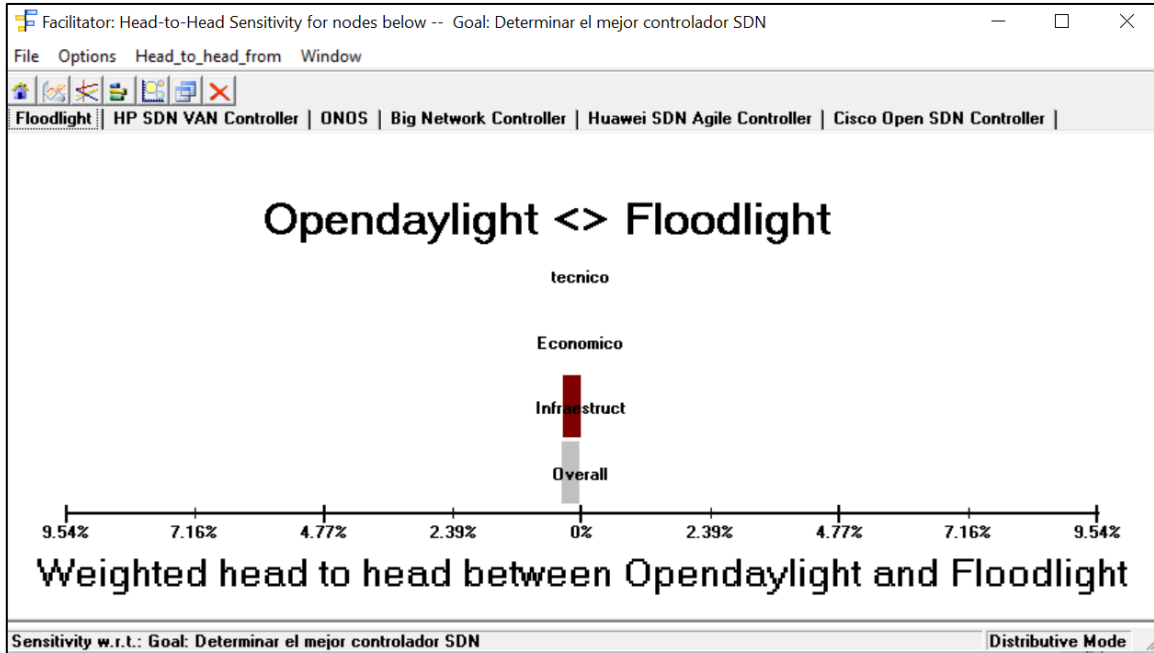


Figura 23 Comparación directa controlador OpenDaylight y Floodlight [Fuente simulador]

En la Figura 24, se observa una comparación directa entre el controlador OpenDaylight y el controlador ONOS, los cuales quedaron el primer y tercer lugar respectivamente en la selección realizada; en la figura se observa que el controlador Opendaylight supera al controlador ONOS en las categorías de criterios técnicos ampliamente y de infraestructura en solo un poco, además de tener un empate en la categoría de criterios económicos, de esta manera se explica la separación en las valoraciones entre ambos controladores, que termina favoreciendo al controlador OpenDaylight.

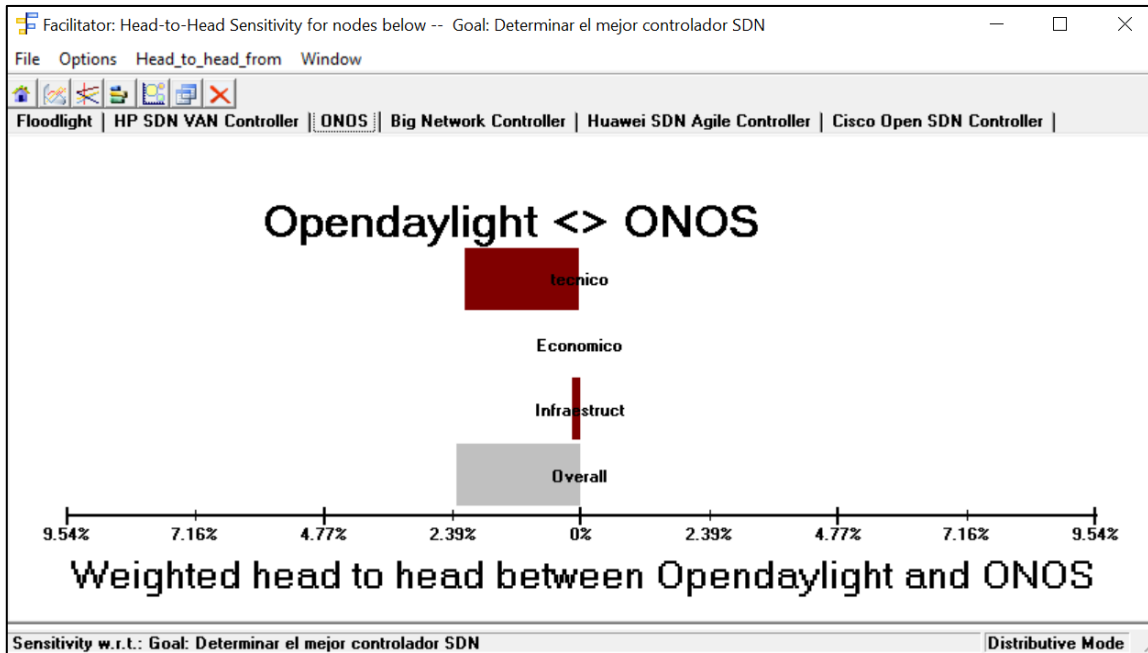


Figura 24 Comparación directa controlador OpenDaylight y ONOS [Fuente simulador]

En la Figura 25, se observa una comparación directa entre el controlador Floodlight y el controlador ONOS, los cuales quedaron el segundo y tercer lugar respectivamente en la selección realizada; en la figura se observa que el controlador Floodlight supera al controlador ONOS en la categoría de criterios técnicos ampliamente, pero en la categoría de criterios de infraestructura el controlador ONOS tiene un poco de ventaja, además tienen un empate en la categoría de criterios económicos, al realizar la valoración global se explica la separación en las valoraciones entre ambos controladores, que termina favoreciendo al controlador Floodlight.

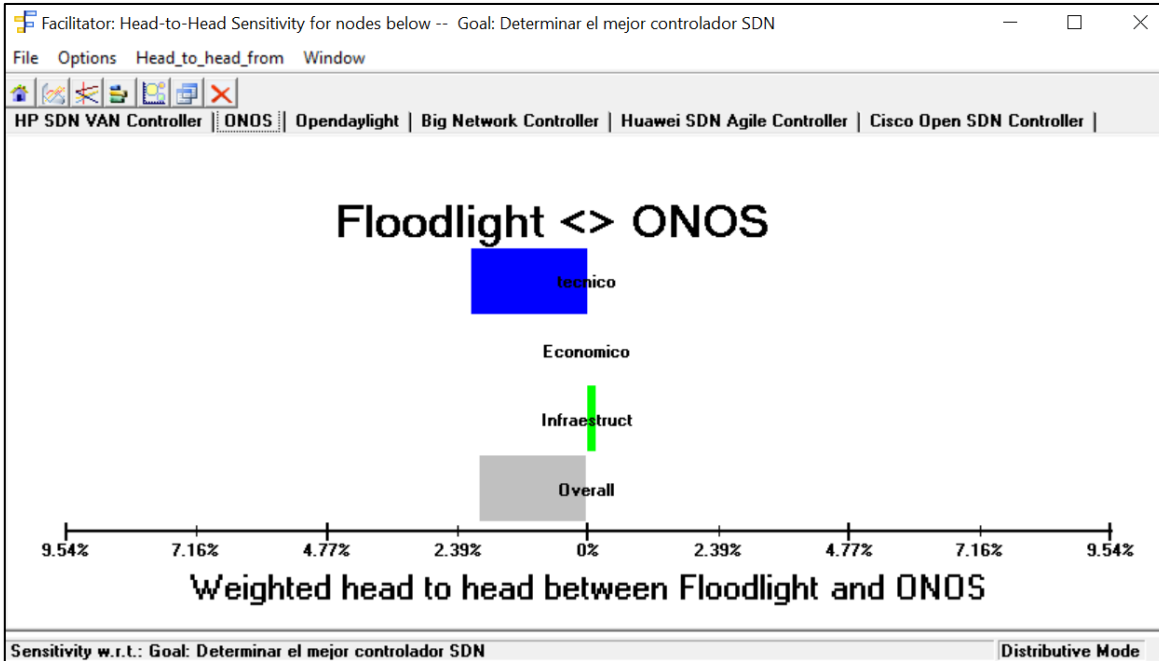


Figura 25 Comparación directa controlador Floodlight y ONOS [Fuente simulador]

En las Figuras 26 y 27, se observa una comparación entre los controladores que en la selección quedaron en los puestos tercero, cuarto, quinto y sexto con los controladores que quedaron en la posición directamente inferior, es decir, el controlador que en la selección ocupó el tercer puesto es comparado con el término en la cuarta posición y el que terminó en el sexto lugar es comparado con el que terminó en el séptimo puesto. De esta manera se puede observar las fortalezas y debilidades de cada uno de los controladores, las cuales determinaron el posicionamiento de cada uno en la selección realizada.

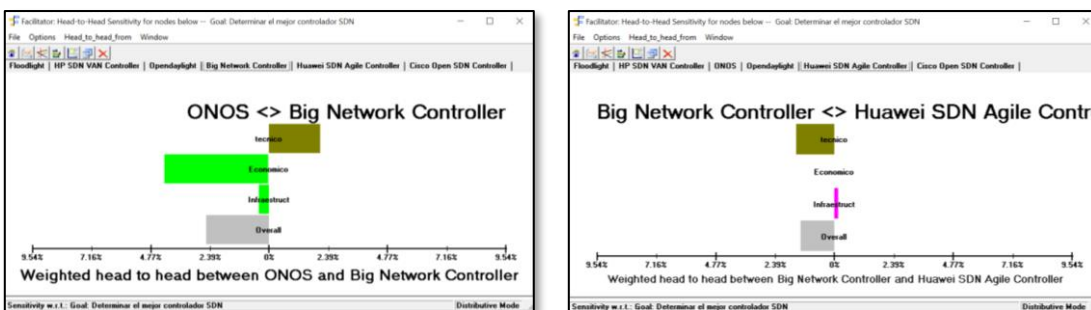


Figura 26 Comparación directa ONOS, Big Network Controller y Huawei SDN Agile Controller [Fuente simulador]

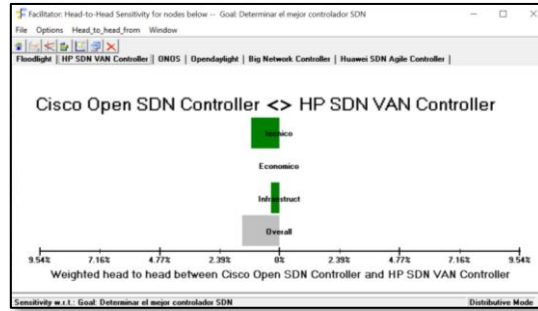
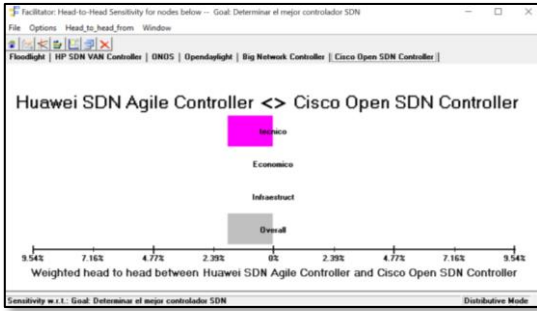


Figura 27 Comparación directa Huawei SDN Agile Controller, Cisco Open SDN Controller y HP SDN VAN Controller [Fuente simulador]

5. PROPUESTA METODOLÓGICA

La propuesta metodológica desarrollada en este capítulo, se enfoca en realizar la selección del controlador de redes SDN que mejor se adapte a las necesidades de una red específica bajo criterios económicos, de infraestructura y técnicos, usando para ello el método PAJ, siguiendo el proceso realizado anteriormente en el documento. El desarrollo de la propuesta metodológica incluye la utilización de un archivo Excel, el cual facilita la selección del controlador. El modo de uso de este archivo se explica en el anexo A de este documento.

La selección propuesta es realizada entre 7 alternativas de controladores SDN, siendo estos los controladores OpenDaylight, Floodlight, ONOS, Big Network Controller, Huawei SDN Agile Controller, Cisco Open SDN Controller y HP SDN VAN Controller. Estos controladores fueron seleccionados en el desarrollo del proceso para determinar cuál es el controlador SDN que mejor se adapta a las necesidades de una red empresarial (Sección 3.3.3), debido a que están vigentes y tienen las mejores características entre los 35 controladores encontrados en la literatura revisada sobre el tema. De manera adicional, la selección y valoración de cada uno de los criterios de decisión, se realizó mediante un proceso en el que participaron expertos en el tema (Sección 3.3.4) y se analizó la información encontrada sobre la caracterización de los controladores SDN (Sección 3.3.5). Entregando una propuesta metodológica donde mediante la valoración pareada de criterios de decisión desarrollada por de tres expertos, se obtiene el cálculo de prioridad global de cada una de las alternativas y se puede tomar una decisión sobre cuál es el controlador SDN que mejor se adapta a las necesidades de la red evaluada.

5.1 PASOS DE LA METODOLOGÍA

A continuación, se presentan los pasos que se deben seguir para llevar a feliz término la metodología entregada. En la Figura 28, se observa el diagrama de flujo de la metodología, indicando los pasos del proceso, los cuales deben ser desarrollados secuencialmente; excepto en el paso de la ponderación de los criterios realizada por los expertos, la cual se puede realizar paralelamente por cada uno de ellos, pero deben estar realizadas las tres ponderaciones para continuar en el proceso.

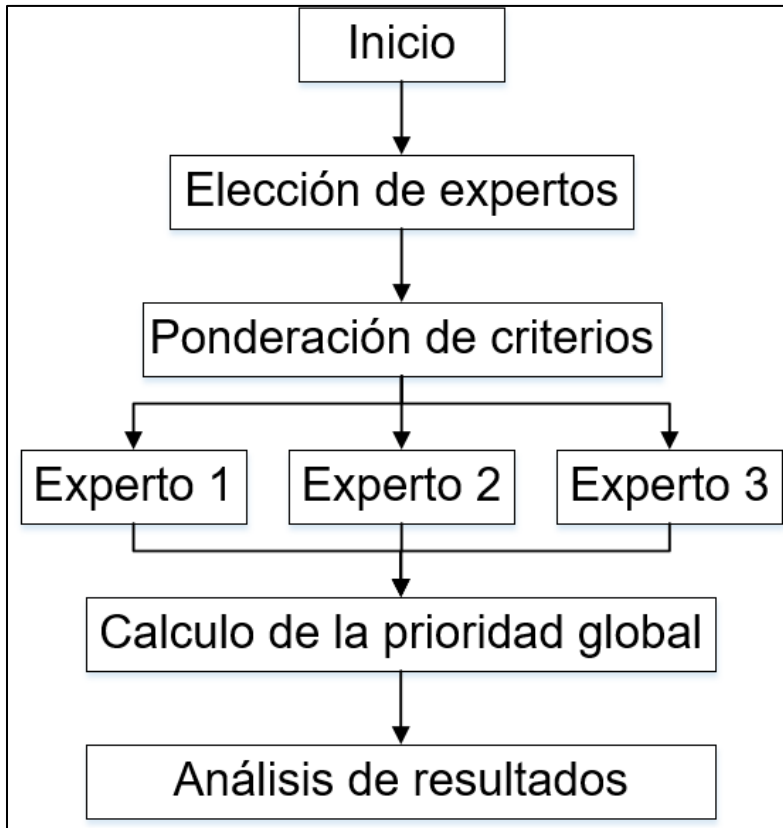


Figura 28 Diagrama de flujo de la metodología [Elaboración propia]

5.1.1 Elección de expertos

En el proceso se debe realizar la selección de 3 expertos en temas de redes y que se encuentren familiarizados con las redes SDN, preferiblemente estos expertos deben ser parte de la empresa o deben estar familiarizados con los objetivos de la empresa en cuestión, de esta manera se asegura que la valoración de los criterios de decisión realizada por ellos, está alineada con los objetivos de la empresa al realizar la implementación de esta nueva arquitectura. En caso que los expertos sean parte de la empresa, de ser posible, deben ser de áreas distintas de la empresa, de tal modo que se garantice la pluralidad en las opiniones, teniendo información de la parte técnica, administrativa y económica.

Una vez elegidos los expertos, se procederá a identificarlos como expertos 1, 2 y 3, cuyo nombramiento no tiene mayor importancia más que la de identificarlos en el desarrollo del proceso.

5.1.2 Ponderación de los criterios

Los expertos deben entregar las valoraciones de prioridad de los criterios de decisión, según su experiencia y alineados con los objetivos de la empresa, para esta valoración se usa la tabla de valoración de Saaty (Tabla 4); estas valoraciones son recopiladas mediante un formulario desarrollado para cada uno de ellos, donde se realizan 40 comparaciones pareadas divididas en 9 matrices de decisión, como se puede observar en la Figura 29. El procesamiento de la información proporcionada por los tres expertos, entrega el vector de prioridad relativo de cada uno de los criterios de decisión, la información sobre el Ratio de inconsistencia (RC) de las matrices evaluadas y el valor de prioridad global de las alternativas para cada uno de los expertos (Método distribuido); teniendo solo visible las dos últimas informaciones en la hoja "Datos" del archivo de Excel desarrollado, con el fin de evaluar si el valor del Ratio de Inconsistencia (RC) supera el 10% en alguna de las matrices, ante lo cual se debe proceder a repetir la valoración de los criterios de decisión el formulario por parte del experto.

The screenshot shows a web-based form titled "Español" for evaluating decision criteria. It is organized into several sections:

- CATEGORIAS DE CRITERIOS DE DECISION:** Includes dropdowns for "Tecnica vs Infraestructura", "Tecnica vs Economica", "Economica vs Infraestructura", and a numeric input field with values 1, 3, 5, 7, 9.
- CRITERIOS ECONOMICOS:** Includes dropdowns for "Codigo Abierto o propietario vs Sistemas operativos", "Codigo Abierto o propietario vs Tipo de licencia de uso", and "Sistemas operativos vs Tipo de licencia de uso".
- CRITERIOS DE INFRAESTRUCTURA:** Includes dropdowns for "Ambientes de uso vs Integración con la nube", "Ambientes de uso vs Sistema distribuido o centralizado", and "Integración con la nube vs Sistema distribuido o centralizado".
- CRITERIOS TECNICOS:** A grid of dropdowns comparing various technical aspects like "Servicios del controlador vs Conexiones del controlador", "Vigencia del controlador", "Características técnicas del controlador", and "Desarrollo del controlador".
- SERVICIOS DEL CONTROLADOR:** Includes dropdowns for "Aplicaciones de enrutamiento vs Aplicaciones de medición y monitoreo" and "Aplicaciones de seguridad y confianza vs Tipo de interfaz de usuario".
- CONEXIONES DEL CONTROLADOR:** Includes "API hacia el sur vs API hacia el norte" and "VIGENCIA DEL CONTROLADOR: Tiene pagina el controlador vs Fecha actualización de la pagina".
- CARACTERISTICAS TECNICAS DEL CONTROLADOR:** A grid of dropdowns comparing "Consistencia de la información vs Versión protocolo Openflow", "Consistencia de la información vs Tolerancia a fallos", "Consistencia de la información vs Soporte multihilos", "Consistencia de la información vs N° de flujos soportados", "Versión protocolo Openflow vs Soporte multihilos", "Versión protocolo Openflow vs N° de flujos soportados", "Tolerancia a fallos vs Soporte multihilos", "Tolerancia a fallos vs N° de flujos soportados", and "Soporte multihilos vs N° de flujos soportados".
- DESARROLLO DEL CONTROLADOR:** Includes dropdowns for "Fabricante vs Lenguaje de programación", "Fabricante vs Documentación", and "Lenguaje de programación vs Documentación".

At the bottom, there are two buttons: "GUARDAR DATOS" and "SALIR".

Figura 29 Formulario de valoración de prioridad de criterios de decisión [Elaboración propia]

5.1.3 Cálculo de la prioridad global en el conjunto de alternativas

Después de ser entregadas las valoraciones de los criterios de decisión por parte de los tres expertos mediante los formularios y ser validados los Ratios de Inconsistencia de cada una de las 9 matrices de decisión, las cuales en ningún caso deben superar el 10%; se debe proceder a revisar los valores de prioridad global (Método distribuido) de cada una de las 7 alternativas, los cuales se pueden observar en la parte inferior de la hoja “Datos” del archivo de Excel desarrollado, como se muestra en la Figura 30.

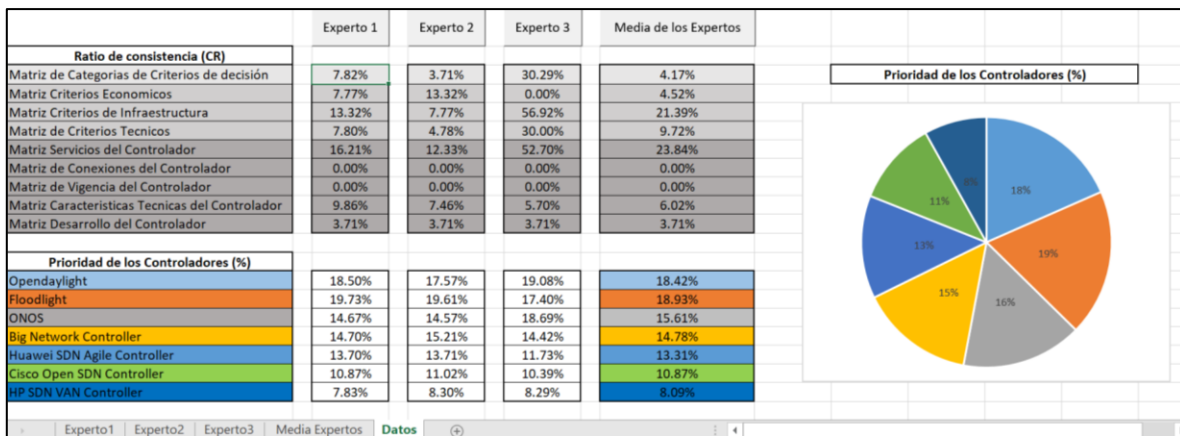


Figura 30 Hoja Excel con valor de prioridad global de las alternativas [Elaboración propia]

5.1.4 Análisis de resultados

Con base en los resultados del proceso de cálculo de la prioridad global en el conjunto de alternativas (Método distribuido), encontrado en la parte inferior de la hoja “Datos” del archivo de Excel desarrollado, se puede determinar cuál de las 7 alternativas es el controlador que mejor se adapta a las necesidades de la red evaluada, bajo criterios económicos, de infraestructura y técnicos; siendo este el controlador que tenga el valor de prioridad global más alto entre el grupo de alternativas. Se debe tener en cuenta que, para tener una solución definitiva, la diferencia del valor de prioridad global entre alternativas debe ser superior al 10%, de lo contrario, estas alternativas se encuentran en un empate estadístico. El modo de uso del archivo Excel que permite llegar a esta solución, se explica en el anexo A de este documento.

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Los métodos de toma de decisión multicriterio son poderosas herramientas de apoyo para los encargados de elegir la mejor opción entre un grupo de posibilidades, ya que orientan el proceso, permiten encontrar una solución para problemas complejos y con gran cantidad de variables a tener en cuenta, de manera adicional, eliminan la posibilidad de tener sesgos involuntarios por alguna de las alternativas y entregan la posibilidad de realizar una selección debidamente sustentada con base en la información con la que se cuenta en el momento.
- El método de toma de decisiones multicriterio PAJ, es un método sencillo pero muy poderoso y robusto, que permite encontrar la solución a problemas complejos y con gran cantidad de variables a tener en cuenta, usando información tanto cuantitativa como cualitativa, permitiendo el uso de opiniones y la experiencia de expertos en el tema, a la vez que minimiza la posibilidad de tener un sesgo que favorezca a alguna de las alternativas por parte del tomador de decisión.
- Las Redes Definidas por Software (SDN), más que una nueva tecnología es una nueva arquitectura de red, que permite la evolución de las redes de comunicación a entornos completamente digitales, mediante su unión con tecnologías como la Virtualización de Funciones de Red (NFV), el ambiente en la nube y la Virtualización de Redes (NV). Permitiendo el desarrollo de nuevos servicios y funcionalidades, como los Sistemas de Internet de las Cosas (IoT) y la calidad de servicio de extremo a extremo.
- Los principales líderes en el desarrollo de las redes SDN son las mismas empresas fabricantes de equipos de transmisión, previendo las ventajas que el desarrollo de esta arquitectura les puede generar en el futuro; realizando este apoyo de manera directa a través de desarrollos propios o mediante la financiación de otras organizaciones como la Open Network Foundation (ONF).
- En el proceso de búsqueda de información relevante sobre los controladores de redes SDN, se identificaron 23 características que definen los controladores en la mayoría de los casos; estas mismas fueron sometidas a evaluación por parte de los

expertos sobre su pertinencia al momento realizar la selección de un controlador de redes SDN, los cuales consideraron que 22 de estas características eran relevantes. Estas 22 características seleccionadas fueron usadas en el método PAJ como criterios de decisión.

- En el proceso de elección de los criterios de decisión relevantes al momento de realizar la selección de un controlador de redes SDN, a los expertos se les consulto si los aspectos técnicos, de infraestructura y económicos eran suficientes al momento de caracterizar un controlador de redes SDN, como se planteó en el objetivo específico 1 de este trabajo; analizando sus respuestas, se determinó que estos tres aspectos son suficientes al momento de caracterizar un controlador de redes SDN. Estos tres aspectos fueron usados como categorías, en las cuales fueron divididos los criterios de decisión encontrados.
- La principal característica de las redes SDN es su control centralizado, que permite poder gestionar toda la red como un único equipo, independientemente de su tamaño, número de dispositivos que la conformen y fabricantes. Actualmente, se han desarrollado gran cantidad de proyectos de controladores de redes SDN, intentando satisfacer distintas necesidades como seguridad, confiabilidad, resiliencia, escalabilidad, entre otras; encontrando en el desarrollo de este documento 35 alternativas de controladores SDN, de los cuales después de realizar una selección por fases, se destacaron 7, por encontrarse vigentes, tener las mejores características y documentación, siendo estos los controladores OpenDaylight, Floodlight, ONOS, Big Network Controller, Huawei SDN Agile Controller, Cisco Open SDN Controller y HP SDN VAN Controller.
- Mediante el desarrollo del método PAJ para seleccionar el controlador que mejor se adapte a las necesidades específicas de una red empresarial ejemplo, se identificó que la mejor opción de controlador para este caso en particular es OpenDaylight, seguido del controlador Floodlight; se debe tener en cuenta que la diferencia en los valores de prioridad global entre ambos controladores fue inferior a un 3%, por lo cual se encuentran en un empate estadístico. A través de una comparación directa entre estos dos controladores, se determinó que existe una pequeña diferencia en la categoría de criterios de infraestructura a favor del controlador OpenDaylight.

- Mediante el proceso de análisis de sensibilidad a través del software Expert Choice a la selección realizada, se logró evidenciar la robustez del método PAJ, al no tener cambios en el orden de los controladores a pesar de realizar variaciones en la valoración de las categorías de criterios de decisión en un +/- 10%; igualmente, al realizar variaciones en la valoración de las categorías de criterios de decisión en un +/- 50%, se evidenciaron cambios en el orden de los controladores solo en el escenario donde se modifica el valor de la categoría del criterio técnico hasta en un 50% adicional, donde varían los controladores ubicados en el tercer puesto, y solo al final del escenario el controlador Big Network Controller logra igualar la valoración del controlador Floodlight, ubicado en segundo lugar. Evidenciando de esta manera la certeza de la solución encontrada y la fortaleza de todo el proceso realizado.
- La relevancia de haber considerado aspectos de infraestructura y económicos, además de aspectos técnicos en el desarrollo de este documento, se evidencia al momento de realizar el análisis de sensibilidad a través del software Expert Choice, específicamente en el escenario donde se modifica el valor de la categoría del criterio técnico hasta en un 50% adicional, allí se observa efectivamente que los controladores Big Network Controller y Huawei SDN Agile Controller, superan en las valoraciones de criterios técnicos al controlador ONOS, pero al analizar en conjunto todas sus características, este último controlador es más robusto y termina ocupando el tercer lugar por encima de estos dos.
- Debido al tiempo de recolección de la información necesaria y a la complejidad del método PAJ, para realizar la selección del controlador de redes SDN de mejores características técnicas, económicas y de infraestructura, siguiendo las especificaciones del tomador de decisión y/o los expertos de redes convocados, se hace necesario el desarrollo de una metodología que permita poder realizar esta selección de una manera más rápida, práctica y que aproveche todo el proceso desarrollado en este trabajo.

6.2 RECOMENDACIONES

- Se recomienda al momento de realizar un análisis para tomar una decisión en cualquier aspecto, usar los métodos de toma de decisión, ya que estos permiten

tener un proceso organizado, que evita la desviación del objetivo principal y la posibilidad de un sesgo involuntario por algunas de las alternativas de solución; además de entregar un proceso bien documentado y sustentado del cómo se llegó a la solución, con base en la información del momento.

- Se recomienda al momento de realizar un método de toma de decisión, tomar el tiempo suficiente para buscar la documentación pertinente, debido a que esto facilita el proceso, le da más robustez a la decisión tomada y más certeza sobre la validez de la solución final.
- Se recomienda al momento de realizar la selección del controlador que mejor se adapte a las necesidades de una red específica, usar la propuesta metodológica entregada en este documento; en caso de necesitar mayor certeza de la decisión, se puede aprovechar la tabla 7 como base para desarrollar el proceso, validando la información sobre las características de los controladores SDN a partir del año 2019.
- Se propone como continuación de este trabajo, realizar un estudio de rendimiento sobre los controladores elegidos como alternativas en el desarrollo de este documento, lo que permitirá robustecer los resultados encontrados y ampliar el conocimiento sobre los mismos.

BIBLIOGRAFÍA

- [1] ONF (Open Networking Fundation), Software Defined Networking: The New Norm for Networks, 2012.
- [2] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski y Arjun Singh, B4: Experience with a Globally-Deployed, Google Inc, 2013.
- [3] R. M. Park Campos y E. E. Baack Valle, Despliegue y evaluación de desempeño de una red, Instituto Tecnológico Autónomo de México ; Universidad de Quintana Roo, 2012.
- [4] D. F. Blandón Gómez, «Openflow: El Protocolo Del Futuro,» Revista UCP (Universidad Catolica de Pereira), Manizalez, 2013.
- [5] J. Machado, A. f. Ramos y J. C. Cuellar, «Implementacion de Openflow sobre NetFPGA,» Revista Ingenieria y Region, Universidad ICESI, Cali, 2014.
- [6] W. Braun y M. Menth, Software Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices, German Federal Ministry of Education and Research (BMBF), 2014.
- [7] A. F. Ruiz, Estado del arte redes definidas por software (SDN), Pereira: Universidad Catolica de Pereira, 2014.
- [8] D. Kreutz, F. M. V. Ramos, F. Paulo Verissimo, C. E. Rothenberg, S. Azodolmolky y S. Uhlig, «Software Defined Networking: A Comprehensive Survey,» IEEE, 2014.
- [9] Ó. Roncero Hervás, Software Defined Networking, Univesidad politecnica de Cataluña, 2014.
- [10] O. Oladunjoye, Software Defined Networking The Emerging Paradigm To Computer Networking, Rourkela, India: National Institute of Technology, 2017.

- [11 C. J. Jacob H., J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley y H. L. Owen,
] Advancing Software-Defined Networks: A Survey, Atlanta: Georgia Institute of
Technology, 2017.
- [12 ONF, ONF What Why, 2016.
]
- [13 Project Floodlight, «projectfloodlight.org,» 27 10 2019. [En línea]. Available:
] <http://www.projectfloodlight.org/floodlight/>.
- [14 OpenDaylight Project, «opendaylight.org,» 27 10 2019. [En línea]. Available:
] <https://www.opendaylight.org/>.
- [15 Hewlett Packard Enterprise, «SDN Apps,» 27 10 2019. [En línea]. Available:
] <https://community.arubanetworks.com/t5/SDN-Apps/ct-p/SDN-Apps>.
- [16 B. Valencia, S. Santacruz, L. Becerra y J. Padilla, «Mininet: una herramienta versátil
] para emulación y prototipado de Redes Definidas por Software,» Universidad Católica
de Pereira, Pereira, 2015.
- [17 lumina networks, «lumina networks,» 31 10 2019. [En línea]. Available:
] <https://www.luminanetworks.com/>.
- [18 IBM, «Servicios de IBM para redes definidas por software.,» 31 10 2019. [En línea].
] Available: <https://www.ibm.com/co-es/services/network/software-defined>.
- [19 Cisco, «Cisco XNC SDN Controller,» 31 10 2019. [En línea]. Available:
] <https://www.cisco.com/c/en/us/support/cloud-systems-management/extensible-network-controller-xnc-version-1-5/model.html>.
- [20 VMware, «VMware NSX,» 31 10 2019. [En línea]. Available:
] <https://www.vmware.com/co/products/nsx.html>.
- [21 D. A. Maldonado Hidalgo, «Diseño e implementación de una aplicación de red bajo la
] arquitectura SDN,» Pontificia Univerisidad Javeriana, Bogotá, 2014.

- [22 M. E. Olaya Yandun, Diseño e Implementacion de una Aplicacion de Balanceo de Carga para una Red Definida por Software (SDN), Ecuador: Escuela Politecnica Nacional, 2015.
- [23 I. Rojas García, Monitorizacion del Plano de Datos en Redes Definidas por Software, Universidad Complutense de Madrid, 2015.
- [24 E. M. Moscoso Clerque, «Desarrollo de una Aplicacion para la Implementacion de Calidad de servicio por Priorizacion de Trafico sobre una Red Definida por Software (SDN),» Escuela politecnica Nacional, Quito, 2016.
- [25 M. A. Velazco Berrezueta, Diseño e Implementacion de una Aplicacion Prototipo para Ofrecer el Servicio de DHCP Sobre una SDN, Ecuador: Escuela Politecnica Nacional, 2016.
- [26 R. Khondoker, A. Zaalouk, R. Marx y K. Bayarou, Feature-based Comparison and Selection of Software Defined Networking (SDN) Controllers, Fraunhofer Institute for Secure Information Technology, 2014.
- [27 O. Belkadi y Y. Laaziz, A Systematic and Generic Method for Choosing A SDN Controller, Morocco: Department of Communication systems and Computer science, 2017.
- [28 D. Sakellaropoulou, A Qualitative Study of SDN Controllers, Athens University of economics and business, 2017.
- [29 sourceforge, «Cbench,» 27 agosto 2013. [En línea]. Available: <https://sourceforge.net/projects/cbench/>. [Último acceso: Julio 11 2020].
- [30 H. Dominik, W. Lukasz y J. Jürgen, Analysis of Realizing a Future Industrial Network by Means of Software-Defined Networking (SDN), IEEE, 2016.
- [31 A. García Centeno, C. M. Rodríguez Vergel y C. A. ., F. C. C. B. Calderón, Controladores SDN, elementos para su selección y evaluación..

- [32 K. S. Sahoo, S. K. Mishra, S. Sahoo y B. Sahoo, Software Defined Network The Next Generation Internet Technology, Rourkela, India: National Institute of Technology, 2017.
- [33 Open Networking Foundation (ONF), Openflow Switch Specification v1.5.1, 2015.
- [34 A. Imran, SDN Controllers Security Issues, Finlandia: University of Jyväskylä, 2017.
- [35 H. I. Kobo, A. M. A. Mahfouz y G. P. Hancke, A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements, University of Pretoria, South Africa ; City University of Hong Kong, Hong Kong, 2017.
- [36 iskratel, «iskratel.com,» 31 10 2019. [En línea]. Available: <https://www.iskratel.com/en/company/press-centre/y2013/m10/98-Iskratel-showcases-one-of-industry-s-first-SDN-applications>.
- [37 M. Paliwal, D. Shrimankar y O. Tembhurne, Controllers in SDN: A Review Report, Nagpur, India: Visvesvaraya National Institute of Technology, 2018.
- [38 S. Shin, Y. Songy, T. Leey, S. Leey, J. Chungy, P. Porrasy, V. Yegneswarany, J. Noh y B. B. Kang, Rosemary A Robust, Secure, and High-Performance Network Operating System, KAIST; Atto Research Korea; SRI International, 2014.
- [39 T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama y S. Shenker, Onix: A Distributed Control Platform for Large-scale Production Networks, Nicira, Google, NEC, University of Berkeley, 2010.
- [40 F. Botelho, A. Bessani, F. M. y V. R. P. Ferreira, SmartLight A Practical Fault-Tolerant SDN Controller, University of Lisbon, Portugal, 2014.
- [41 S. Matsumoto, S. Hitz y A. Perrig, Fleet Defending SDNs from Malicious Administrators, Escuela politecnica federal de Zurich (ETH) Suiza; Universidad Carnegie Mellon (CMU) EE UU, 2014.

- [42 A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman y R. R. Kompella, *ElastiCon: An Elastic Distributed SDN Controller*, West Lafayette IN, USA: Purdue University, 2014.
- [43 N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown y S. Shenker, *NOX: Towards an Operating System for Networks*, Nicira Networks; stanford University; University of California, 2008.
- [44 MurphyMc, «GitHub noxrepo/nox-classic,» 19 10 2019. [En línea]. Available:
] <https://github.com/noxrepo/nox-classic/wiki>.
- [45 F. Bannour, S. Souihi y A. Mellouk, *Distributed SDN Control Survey, Taxonomy, and Challenges*, University of Paris-Est Créteil, 2018.
- [46 McCauley, «Github noxrepo.github.io,» 19 10 2019. [En línea]. Available:
] <https://noxrepo.github.io/pox-doc/html/>.
- [47 Z. Cai, A. L. Cox y T. S. E. Ng, *Maestro: A System for Scalable OpenFlow Control*, Rice University, 2010.
- [48 zhengcai, «GitHub maestro platform,» 19 10 2019. [En línea]. Available:
] <http://zhengcai.github.io/maestro-platform/>.
- [49 A. D. Ferguson, A. Guha, C. Liang, R. Fonseca y S. Krishnamurthi, *Participatory Networking: An API for Application Control of SDNs*, Universidad de Brown, 2013.
- [50 Brown University, «PANE Controller,» 19 10 2019. [En línea]. Available:
] <http://pane.cs.brown.edu>.
- [51 V. R. S. Raju, *SDN Controllers Comparison*, Bangalore: SRIT, 2018.
]
- [52 Openmul.org, «Open MUL Controller,» 19 10 2019. [En línea]. Available:
] <http://www.openmul.org>.

- [53 B. Lee, S. H. Park, J. Shin y S. Yang, IRIS: The Openflow-based Recursive SDN
] Controller, Electronics and Telecommunications Research Institute (ETRI), Daejeon,
Republic of Korea, 2014.
- [54 B. Lee, «GitHub Open Iris Controller,» 19 10 2019. [En línea]. Available:
] <https://openiris.etri.re.kr>.
- [55 D. Erickson, The Beacon OpenFlow Controller, Stanford University, 2013.
]
- [56 S. A. Shah, J. Faiz, M. Farooq, A. Shafi y S. A. Mehdi, An Architectural Evaluation of
] SDN Controllers, Pakistan: School of EECS, National University of Sciences and
Technology (NUST), 2017.
- [57 D. Erickson, «Openflow Beacon,» 19 10 2019. [En línea]. Available:
] <https://openflow.stanford.edu/display/Beacon/Home.html>.
- [58 DELL Inc., Active Fabric Controller (AFC) User Guide, 2014.
]
- [59 DELL, «Active Fabric Controller,» 19 10 2019. [En línea]. Available:
] <https://www.dell.com/learn/co/es/cobsdt1/networking?s=bsd&~ck=mn>.
- [60 N. Karanatsios, «Github Trema,» 19 10 2019. [En línea]. Available:
] <https://trema.github.io/trema/>.
- [61 NEC, NEC ProgrammableFlow: An Open and Programmable Network Fabric for
] Datacenters and the Cloud, 2012.
- [62 NEC, «ProgrammableFlow Controller,» 19 10 2019. [En línea]. Available:
] <https://www.necam.com/PFlow/>.
- [63 K. Phemius, M. Bouet y J. Leguay, DISCO: Distributed Multi-domain SDN Controllers,
] Francia: Thales Communications & Security, 2013.

- [64] agence nationale de la recherche, «Disco SDN Controller,» 19 10 2019. [En línea].
] Available: <http://anr-disco.ens-lyon.fr/>.
- [65] M. Monaco, O. Michel y E. Keller, Applying Operating System Principles to SDN
] Controller Design, University of Colorado, 2013.
- [66] sohilladhani, «GitHub Yanc Controller,» 19 10 2019. [En línea]. Available:
] <https://github.com/ngn-colorado/yanc>.
- [67] S. H. Yeganeh y Y. Ganjali, Kandoo: A Framework for Efficient and Scalable Offloading
] of Control Applications, University of Toronto, 2012.
- [68] soheilhy, «Beehive Controller,» 19 10 2019. [En línea]. Available:
] <https://github.com/kandoo/beehive-netctrl>.
- [69] Applied Research Center for Computer Networks, «GitHub Runos Controller,» 19 10
] 2019. [En línea]. Available: <https://arccn.github.io/runos/>.
- [70] kozorezal, «GitHub Loom Controller,» 19 10 2019. [En línea]. Available:
] <https://github.com/FlowForwarding/loom>.
- [71] Ryu SDN Framework Community, «Ryu SDN Framework,» 19 10 2019. [En línea].
] Available: <http://osrg.github.io/ryu/>.
- [72] C. Dixon, D. Olshefski, V. Jain, C. DeCusatis, W. Felter, J. Carter, M. Banikazemi, V.
] Mann, J. M. Tracey y R. Recio, Software defined networking to support the software
defined environment, IBM, 2014.
- [73] IBM, «IBM SDN VE,» 19 10 2019. [En línea]. Available:
] <https://www.ibm.com/services/network/software-defined>.
- [74] Faucet Developers, «GitHub Faucet Controller,» 19 10 2019. [En línea]. Available:
] <https://github.com/faucetsdn/faucet>.

[75 Tungsten Fabric, «Open Contrail Controller,» 19 10 2019. [En línea]. Available:
] <http://www.opencontrail.org/>.

[76 Juniper Networks, Contrail Networking, 2018.

]

[77 Juniper Networks, «NorthStar Controller,» 19 10 2019. [En línea]. Available:
] https://www.juniper.net/documentation/en_US/northstar3.0.0/information-products/pathway-pages/3.0/northstar-user-guide.html.

[78 Ciena Corporation, «Blue Planet,» 11 11 2019. [En línea]. Available:
] <https://www.blueplanet.com>.

[79 S. Asadollahi, B. Goswami, A. S. Raoufy y H. G. J. Domingos, Scalability of software
] defined network on floodlight controller using OFNet, India: Saurashtra University
Rajkot; Christ University Bangalore, 2017.

[80 HP Enterprise, «HP SDN VAN Controller,» 19 10 2019. [En línea]. Available:
] <http://h17007.www1.hpe.com/us/en/networking/library/index.aspx?cat=sdn#.XIsDWS hKg2w>.

[81 OpenNetworking.org, «ONOS Controller,» 19 10 2019. [En línea]. Available:
] <https://onosproject.org>.

[82 J. Medved, R. Vargas, A. Tkacik y K. Gray, OpenDaylight: Towards a Model-Driven
] SDN Controller Architecture, Cisco Systems, 2014.

[83 OpenDaylight Project, «OpenDaylight Controller,» 19 10 2019. [En línea]. Available:
] <https://www.opendaylight.org>.

[84 Big Switch Networks, Big Cloud Fabric™—Enterprise Cloud, 2019.

]

[85 Big Switch Networks, «Big Network Controller,» 19 10 2019. [En línea]. Available:
] <https://www.bigswitch.com/tags/big-network-controller>.

- [86 Cisco, «DEVNET,» 12 11 2019. [En línea]. Available: <https://developer.cisco.com/>.
]
- [87 Cisco, «Cisco Open SDN Controller,» 19 10 2019. [En línea]. Available:
] <https://www.cisco.com/c/en/us/products/cloud-systems-management/open-sdn-controller/index.html>.
- [88 Huawei Technologies, Huawei Agile Controller-DCN 3.0 Datasheet, 2017.
]
- [89 Huawei, «Huawei SDN Agile Controller,» 19 10 2019. [En línea]. Available:
] <https://e.huawei.com/es/products/enterprise-networking/sdn-controller>.
- [90 F. Aguilar, Teoría de la decisión e incertidumbre: modelos normativos y descriptivos,
] España: Instituto de Estudios Sociales Avanzados, 2004.
- [91 J. Aznar B. y F. Guijarro M., Nuevos metodos de valoracion Modelos multicriterio,
] Valencia - España: Universidad Politecnica de Valencia, 2012.
- [92 J. V. Pilar, Herramientas para la Gestión y la Toma de Decisiones, Salta - Argentina:
] Hanne, 2012.
- [93 N. Bulling, A survey of multi-agent decision making, Berlin - Alemania: Clausthal
] University of Technology, 2014.
- [94 B. Vitoriano, Teoria de la Decision: Decisión con Incertidumbre, Decisión Multicriterio
] y Teoría de Juegos, Universidad Complutense de Madrid, 2007.
- [95 G. A. Chica Pedraza, Estudio y Análisis de la viabilidad de la implementación de la
] tecnología PLT (Power Line Telecommunications) en Colombia en el ámbito de la
transmisión de datos sobre redes de baja tensión, Bogotá: Universidad Nacional de
Colombia, 2012.

[96 T. Hurtado y G. Bruno, El Proceso de Analisis Jerarquico (AHP) como herramienta] para la toma de decisiones en la seleccion de proveedores, Lima - Peru: Universidad Nacional Mayor de San Marcos, 2005.

[97 R. Q. J. Sebastian, Caracterizacion y analisis del desempeño de la red de datos de la] Pontificia Universidad Javeriana., Bogotá: Pontificia Universidad Javeriana, 2017.

7. ANEXOS

ANEXO A

7.1 Manual de Uso Archivo de Propuesta Metodológica

En esta sección se encuentra la información necesaria para hacer uso del archivo de Excel, que permite realizar la elección del controlador de redes SDN que mejor se adapte a las necesidades específicas de una red.

Este archivo fue desarrollado en el software de hojas de cálculo Excel, debido a su baja complejidad y a la alta difusión del modo de uso de este programa, aprovechando sus funcionalidades de programación usando Visual Basic y el uso de macros, las cuales permiten ejecutar tareas de manera automática. El archivo comprende 5 hojas de cálculo y un formulario desarrollado para cada uno de los 3 expertos, donde se realizan 40 comparaciones pareadas divididas en 9 matrices de decisión, usando la escala de Saaty; permitiendo de esta manera disminuir la probabilidad de error, al realizar el ingreso de los valores en las matrices desarrolladas en las hojas de cálculo de forma automática.

Entregando como resultado del procesamiento de los datos, el Ratio de inconsistencia de cada una de las matrices por experto y el calculado a través de la media geométrica de las tres valoraciones; el cálculo de la prioridad global del conjunto de alternativas por experto y el calculado a través de la media geométrica de las tres valoraciones. Además de un gráfico de torta, donde por colores está representado el valor calculado a través de la media geométrica de la prioridad global del conjunto de alternativas, todos estos valores calculados siguiendo el método distribuido.

7.1.1 Conocimientos necesarios

Los conocimientos necesarios para hacer uso de este archivo son solo conocimientos básicos de Excel y la escala de valoración de Saaty, que se puede observar en la tabla 31; si se requiere manipular el archivo se debe sumar conocimientos sobre fórmulas y macros en Excel, el proceso del método PAJ y conocimientos básicos de Visual Basic.

Tabla 31 Escala de valoración de Saaty [Elaboración propia]

Escala Numérica	Escala Verbal	Explicación
1	Igual importancia	Los dos elementos contribuyen igualmente a la propiedad o criterio
3	Moderadamente más importante	La experiencia previa favorece al elemento evaluado sobre el de comparación
5	Frecuentemente más importante	La experiencia previa favorece fuertemente al elemento evaluado sobre el de comparación
7	Mucho más importante	La elemento evaluado domina sobre el de comparación.
9	Importancia extrema	La elemento evaluado domina fuertemente sobre el de comparación.

7.1.2 Pasos a seguir

El usuario debe ejecutar el archivo de Excel de nombre “Metodología SDN” dando doble clic sobre el mismo, el cual abrirá una ventana y donde se encuentran 5 hojas de cálculo de nombres “expertos 1, 2 y 3, Media Expertos y Datos”; paso seguido se debe dar clic en la hoja de cálculo de nombre “Datos”, donde se encuentra una interfaz similar a la figura 31; en la parte superior-izquierda están los nombres de las 9 matrices de decisión en las que están divididos los criterios de decisión, a su derecha el porcentaje de Ratio de Inconsistencia de cada una de las matrices, discriminada por cada uno de los expertos y el cálculo de la media geométrica de las tres valoraciones.

En la parte inferior-izquierda se encuentra los nombres de las 7 alternativas de controladores de redes SDN, a su derecha el porcentaje de prioridad global de cada una de las alternativas, discriminado por cada uno de los expertos y el cálculo de la media geométrica de las tres valoraciones. De manera adicional, en la parte derecha se encuentra un gráfico de torta, donde están representados los porcentajes de prioridad global de cada una de las alternativas, calculado a través de la media geométrica de las tres valoraciones.

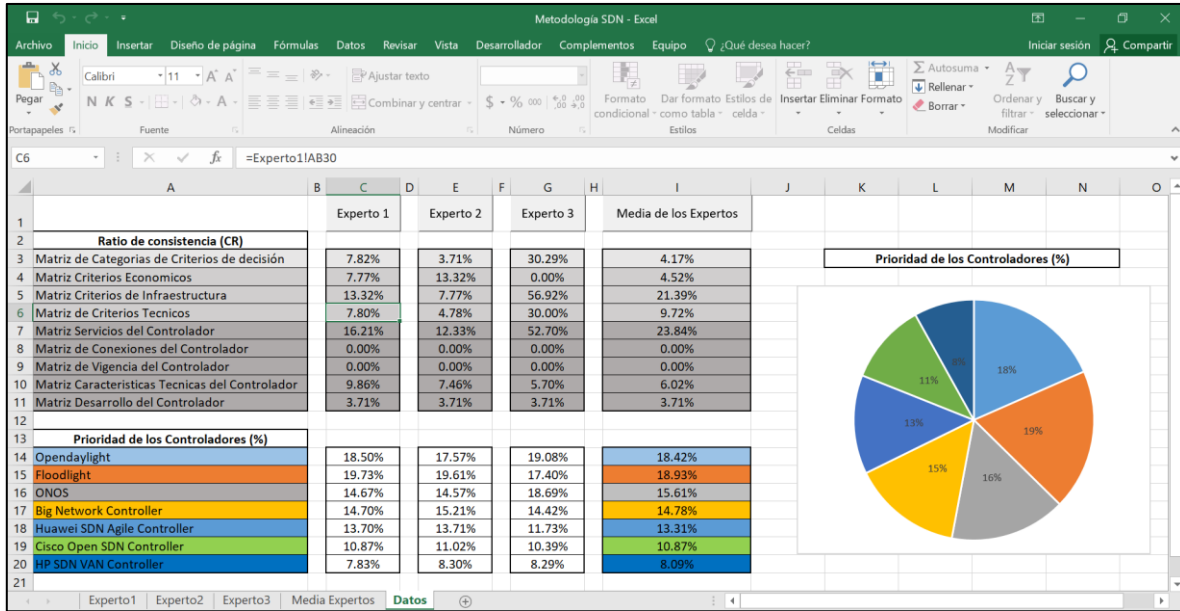


Figura 31 Archivo metodología SDN – hoja de cálculo Datos [Elaboración propia]

Paso seguido se deben rellenar los formularios de cada uno de los expertos, para esto se debe dar clic en los botones de nombre “Experto 1, 2 y 3” de manera individual, estos están ubicados en la parte superior.

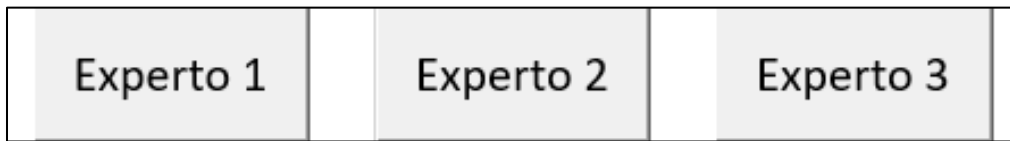


Figura 32 Botones Experto 1,2 y 3 [Elaboración propia]

Después de dar clic en cualquiera de estos botones, por ejemplo, en “Experto 1”, se despliega una ventana con el formulario para este experto, como se observa en la Figura 33, el cual debe ser diligenciado completamente, eligiendo valores entre 1 y 9 de la lista desplegable, siguiendo la escala de valoración de Saaty anteriormente vista (Tabla 31); una vez completado el formulario se debe dar clic en el botón “Guardar Datos” de la parte inferior del formulario y seguidamente en el botón “Salir”; de esta manera la valoración entregada será ingresada en las matrices de la hoja de datos “Experto1” de manera automática. Este proceso debe ser repetido para ingresar los valores de los expertos 2 y 3, dando clic en los botones correspondientes para cada uno.

Experto1

CATEGORIAS DE CRITERIOS DE DECISION
 Tecnica vs Infraestructura Tecnica vs Economica Economica vs Infraestructura

CRITERIOS ECONOMICOS
 Codigo Abierto o propietario vs Sistemas operativos Codigo Abierto o propietario vs Tipo de licencia de uso Sistemas operativos vs Tipo de licencia de uso

CRITERIOS DE INFRAESTRUCTURA
 Ambientes de uso vs Integración con la nube Ambientes de uso vs Sistema distribuido o centralizado Integración con la nube vs Sistema distribuido o centralizado

CRITERIOS TECNICOS
 Servicios del controlador vs Conexiones del controlador Servicios del controlador vs Vigencia del controlador Servicios del controlador vs Características técnicas del controlador Servicios del controlador vs Desarrollo del controlador Conexiones del controlador vs Vigencia del controlador Vigencia del controlador vs Características técnicas del controlador Vigencia del controlador vs Desarrollo del controlador Características técnicas del controlador vs Desarrollo del controlador

SERVICIOS DEL CONTROLADOR
 Aplicaciones de enrutamiento vs Aplicaciones de medición y monitoreo Aplicaciones de enrutamiento vs Aplicaciones de seguridad y confianza Aplicaciones de enrutamiento vs Tipo de interfaz de usuario Aplicaciones de medición y monitoreo vs Aplicaciones de seguridad y confianza Aplicaciones de medición y monitoreo vs Tipo de interfaz de usuario

Aplicaciones de seguridad y confianza vs Tipo de interfaz de usuario

CONEXIONES DEL CONTROLADOR-----**VIGENCIA DEL CONTROLADOR**
 API hacia el sur vs API hacia el norte Tiene pagina el controlador vs Fecha actualización de la pagina

CARACTERISTICAS TECNICAS DEL CONTROLADOR
 Consistencia de la información vs Version protocolo Openflow Consistencia de la información vs Tolerancia a fallos Consistencia de la información vs Soporte multihilos Consistencia de la información vs N° de flujos soportados Version protocolo Openflow vs Tolerancia a fallos

Version protocolo Openflow vs Soporte multihilos Version protocolo Openflow vs N° de flujos soportados Tolerancia a fallos vs Soporte multihilos Tolerancia a fallos vs N° de flujos soportados Soporte multihilos vs N° de flujos soportados

DESARROLLO DEL CONTROLADOR
 Fabricante vs Lenguaje de programación Fabricante vs Documentación Lenguaje de programación vs Documentación

GUARDAR DATOS SALIR

Figura 33 Formulario Experto 1 [Elaboración propia]

En caso de olvidar ingresar un valor en alguna de las comparaciones pareadas del formulario, al regresar a la hoja de Cálculo “Datos”, encontrará en la celda correspondiente al Ratio de Inconsistencia de esa matriz de decisión la palabra “Error”, como se puede observar en la Figura 34, igualmente si no se ingresa la valoración de alguno de los tres expertos. De manera adicional, se debe verificar que el valor de Ratio de Inconsistencia de todas las matrices debe ser igual o inferior a 10.00%, de lo contrario, se debe volver a diligenciar el formulario de este experto.

	Experto 1	Experto 2	Experto 3	Media de los Expertos
Ratio de consistencia (CR)				
Matriz de Categorías de Criterios de decisión	Error	3.71%	3.71%	Error
Matriz Criterios Economicos	Error	3.71%	3.71%	Error
Matriz Criterios de Infraestructura	Error	3.71%	3.71%	Error
Matriz de Criterios Tecnicos	Error	17.61%	7.45%	Error
Matriz Servicios del Controlador	Error	17.74%	12.33%	Error
Matriz de Conexiones del Controlador	0.00%	0.00%	0.00%	0.00%
Matriz de Vigencia del Controlador	0.00%	0.00%	0.00%	0.00%
Matriz Características Técnicas del Controlador	Error	11.87%	21.72%	Error
Matriz Desarrollo del Controlador	Error	3.71%	3.71%	Error

Figura 34 Error al ingresar datos [Elaboración propia]

7.1.3 Análisis de resultados

Los valores de las prioridades globales de las alternativas son entregados en porcentaje, se debe tener en cuenta que la diferencia entre valores de las alternativas debe ser mayor a un 10%, de lo contrario estas alternativas se encuentran en un empate estadístico. En la Figura 35, se observa los valores de la prioridad global de cada una de las alternativas, discriminada por expertos y la calculada a través de la media geométrica de las tres valoraciones (en colores). En la Figura 36, se observa el gráfico de torta, donde están representados los porcentajes de prioridad global de cada una de las alternativas calculado a través de la media geométrica de las tres valoraciones.

13	Prioridad de los Controladores (%)				
14	Openaylight	18.50%	17.57%	19.08%	18.42%
15	Floodlight	19.73%	19.61%	17.40%	18.93%
16	ONOS	14.67%	14.57%	18.69%	15.61%
17	Big Network Controller	14.70%	15.21%	14.42%	14.78%
18	Huawei SDN Agile Controller	13.70%	13.71%	11.73%	13.31%
19	Cisco Open SDN Controller	10.87%	11.02%	10.39%	10.87%
20	HP SDN VAN Controller	7.83%	8.30%	8.29%	8.09%

Figura 35 Resultados propuesta metodológica [Elaboración propia]

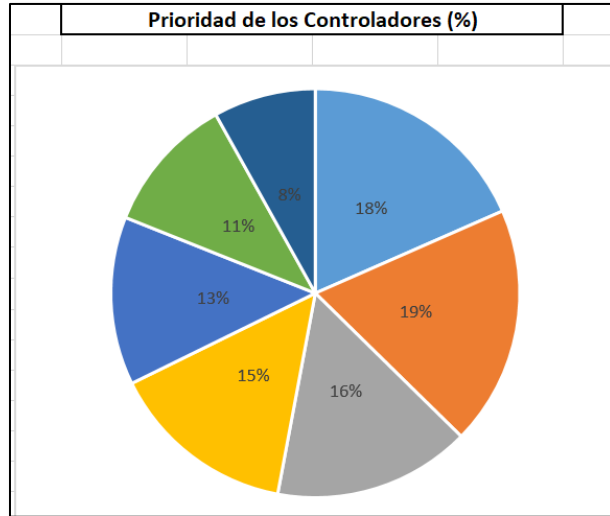


Figura 36 Gráfico de torta – resultados [Fuente Excel]