

Universidad de los Andes
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
Consejo de Estudios de Postgrado
Postgrado en Telecomunicaciones



UNIVERSIDAD
DE LOS ANDES

ANÁLISIS COMPARATIVO DE LA CALIDAD EN APLICACIONES DE EMULACIÓN DE REDES INALÁMBRICAS DEFINIDAS POR SOFTWARE, UTILIZANDO EL MODELO DE ESTIMACIÓN DE PROYECTOS EN ENTORNOS DE INCERTIDUMBRE

TRABAJO DE GRADO PRESENTADO ANTE LA ILUSTRE UNIVERSIDAD DE LOS ANDES PARA
OPTAR AL GRADO DE MAGÍSTER SCIENTIAE EN TELECOMUNICACIONES.

Elaborado por: Gerlyn Eduardo Duarte
Tutor: Ph.D. Francisco Valdés Souto
Ph.D. Jesús Calderón Vielma

Junio 2016
Mérida-Venezuela



Este obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional

Una demora en tus metas jamás la veas como una derrota, percíbela como una oportunidad para crecer, aprender y mejorar cada día, esto te dará las herramientas necesarias para demostrar en algún momento tu valía, en especial a todos aquellos que no creyeron en tí...

Gerlyn Eduardo Duarte



UNIVERSIDAD DE LOS ANDES


Facultad de Ingeniería
Escuela de Eléctrica
Consejo de Estudios de Postgrado
Postgrado en Telecomunicaciones

VEREDICTO


Quienes suscribimos, integrantes del jurado designado por el Consejo de Estudios de Postgrado de la Universidad de Los Andes para conocer el veredicto sobre el Trabajo de Grado presentado por el(la) Ing(a). **GERLYN EDUARDO DUARTE PRIETO**, cédula de identidad No. 6.348.304, para optar al grado de **Magíster Scientiae en Telecomunicaciones**, el cual se intitula: **"ANÁLISIS COMPARATIVO DE LA CALIDAD EN APLICACIONES DE EMULACIÓN DE REDES INALÁMBRICAS DEFINIDAS POR SOFTWARE, UTILIZANDO EL MODELO DE ESTIMACIÓN DE PROYECTOS EN ENTORNOS DE INCERTIDUMBRE"**.

Hacemos constar lo siguiente:

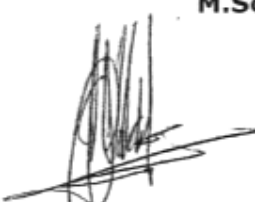
PRIMERO: Que hoy jueves 30-06-2016, a las 03:00 p.m., nos constituimos como Jurado en el Salón de Reuniones de la Dirección de Relaciones Interinstitucionales, Universidad de Los Andes, siendo Presidente del mismo el **Dr. Jesús Calderón Vielma (Universidad de Los Andes)** y estando bajo la modalidad de videoconferencia el tutor y al mismo tiempo Jurado Externo al Postgrado en Telecomunicaciones, Dr. Francisco Valdes Souto (Universidad Nacional Autónoma de México, UNAM). **SEGUNDO:** A continuación discutimos si se procedía a su defensa pública. Luego de considerar las observaciones y críticas de cada miembro del Jurado, acordamos por unanimidad autorizar a su presentación. **TERCERO:** A las 03:15 p.m., de este mismo día, se procedió al acto público de sustentación del Trabajo de Grado presentado a requerimiento del Jurado. **CUARTO:** Una vez concluida la sustentación correspondiente, el Jurado interrogó al aspirante sobre diversos aspectos a que el trabajo se refiere. **QUINTO:** Seguidamente, el presidente del Jurado invitó al público asistente a formular preguntas y observaciones sobre el trabajo presentado. **SEXTO:** Una vez concluido el acto de presentación, el Jurado procedió a la deliberación final y concluyó que: **SE APRUEBA EL TRABAJO DE GRADO PRESENTADO A NUESTRA CONSIDERACIÓN, CON MENCIÓN PUBLICACIÓN.**



Dr. Jesús Calderón Vielma
Tutor



M.Sc. José Rafael Uzcátegui M.
Jurado



Dr. Francisco Valdes Souto
Tutor - Jurado Externo al Postgrado

Coordinación de Postgrado en Telecomunicaciones
Escuela de Ingeniería Eléctrica
Universidad de Los Andes
Mérida, Venezuela
Email: post-telecom@ula.ve
Web: <http://gitel.ing.ula.ve/mtl.html>



Agradecimientos

A DIOS PADRE TODOPODEROSO, por cada día que me ha dado de vida, por haber hecho posible todos los logros que he alcanzado hasta el momento, por llenarme de tantas bendiciones, y en especial por ayudarme a levantar en cada tropiezo que se me ha presentado, por demostrarme que no importa lo empinado y dolorosa que puede ser la vida siempre estará junto a mí, para apoyarme y guiarme para salir adelante.

A MI FAMILIA, la cual tuvo la paciencia y fe suficiente de que lo lograría, a todos ustedes que día y noche le pedían a Dios para que pudiese alcanzar este logro, gracias por estar allí siempre y creer en mí a pesar de todo.

A MI TUTOR, Francisco Valdés Souto Profesor de la Universidad Nacional Autónoma de México (UNAM), no existen palabras con las cuales le pueda expresar mi gratitud por la oportunidad brindada, así como la paciencia y comprensión tenida. De igual forma el dedicar su tiempo y transmitirme sus conocimientos para la realización de este trabajo, a pesar de la distancia. Le doy gracias a Dios por haberme permitido encontrarlo y contar con usted, estaré toda mi vida agradecido, un millón de gracias.

Al Profesor Ramón Fontes de la Universidade Estadual de Campinas (Brasil), por su apoyo desinteresado, incondicional y transmisión de conocimientos a lo largo de esta travesía. Por permitirme formar parte del equipo de pruebas y ser uno de los privilegiados en trabajar con tus herramientas. De igual forma siempre te estaré agradecido por todo.

A MI TUTOR Jesús Calderón Profesor de la Universidad de Los Andes (ULA), gracias por haberme apoyado y brindado la oportunidad de trabajar de forma desinteresada con usted, y motivarme efusivamente a terminar este trabajo.

Al Profesor Nelson Pérez Coordinador de la Maestría en Telecomunicaciones, por su sincera amistad y apoyo incondicional en todo lo que he necesitado a lo largo de esta travesía, por su paciencia durante todo este tiempo, y por todos los consejos que me brindaste en esta etapa de estudio, un millón de gracias, hermano.

A TODOS MIS AMIGOS que siempre han confiado en mí, apoyándome desinteresadamente, nombrarlos a cada uno podría ocurrir que cometa la equivocación de dejar a alguno por fuera, no quiero cometer esa injusticia, esta demás ya que cada uno de ustedes saben quiénes son. GRACIAS HERMANOS.

A TODAS AQUELLAS PERSONAS que algún día pasaron por mi vida y dejaron huella en mi corazón, algunas de ustedes creyeron en mí y otras no, a todos ustedes también les agradezco ya que también fueron fuente de motivación de una forma u otra para luchar y no claudicar sin importar las condiciones adversas en las que me encontrara.

A ESTA CIUDAD Y A ESTA UNIVERSIDAD las cuales tanto me han dado, pero tanto me han quitado...

A TODOS GRACIAS DE TODO CORAZÓN, CUENTEN CONMIGO SIEMPRE,
DIOS LOS BENDIGA A CADA UNO DE USTEDES...

Dedicatoria

A DIOS PADRE TODOPODEROSO, por todo el amor, gracia, vida y luz que me ha brindado, por ser mi guía y fortaleza en los momentos difíciles, sin ti no lo hubiese logrado Señor, mil gracias.

A LOS AMORES DE MI VIDA, MIS HIJOS SEBASTIÁN EDUARDO y SOFÍA DEL CARMEN, ustedes son la bendición y el tesoro más grande que Dios me pudo haber dado, no me cansaré de dar las gracias a mi señor por ello. Muchas veces pensé rendirme, pero luego recordé que ustedes seguirán mis pasos y por eso debía seguir luchando. Este logro es mi pequeño aporte para que al crecer vean que con sacrificio y la ayuda de Dios se pueden lograr grandes cosas, espero que comprendan que los esfuerzos son parte del triunfo. Jamás se den por vencidos, siempre podrán lograr todo lo que se propongan, confío en ambos. Cuando ya no esté físicamente acompañándolos, cuenten con Dios que siempre estará a su lado protegiéndolos y apoyándolos, crean y confíen en el, ÁMENLO sobre todas las cosas. Cúdense, protéjense, apóyense, respétense y ÁMENSE entre ustedes dos toda la vida. Son y serán siempre mi razón de vida, de lucha y de seguir adelante. LOS AMO Y AMARE POR SIEMPRE MIS TESOROS.

A MI MADRE CARMEN SOFÍA quien de una manera desinteresada me brindaste amor, confianza y dedicación el tiempo que estuviste a mi lado, aunque en estos momentos no estás junto a nosotros, se que este triunfo te llenaría de gozo también, y le doy gracias a Dios por cada día que estuviste junto a nosotros, porque la vida y el amor son los dones mas preciosos que el nos da, solo nos queda saber disfrutar cada momento, y yo los disfrute junto a ti, TE AMARE POR SIEMPRE MI AMORCITO.

A MI ESPOSA FRANCIS, a pesar de todo soy feliz a tu lado, gracias por haber llegado a mi vida y haberme dado el tesoro más grande que poseo. Este logro te lo dedico también de corazón, por las distintas formas de apoyo que me brindaste. TE AMO MUCHO GORDA.

AL RESTO DE MI FAMILIA, por creer en mí siempre a pesar de todo, y por el deseo que siempre tuvieron de que este momento llegara, este triunfo es de todos ustedes también, LOS AMO MUCHO.

A los que nunca creyeron en mí y a todos los que de una u otra forma me han estimulado y brindado su amistad y hermandad.

Gerlyn Eduardo Duarte

Índice general

Agradecimientos	IV
Dedicatoria	VI
Resumen	XVI
Abstract	XVII
1. Introducción	1
1.1. Introducción	2
1.2. Planteamiento del problema	7
1.3. Justificación	8
1.4. Delimitación de la Investigación	9
1.5. Objetivos	9
1.5.1. Objetivo general	9
1.5.2. Objetivos específicos	9
2. Marco Teórico	11
2.1. Antecedentes	12
2.1.1. Simuladores de SDWN	12
2.1.2. Modelo EPCU	13
2.2. Definición de términos básicos SDN.	14
2.2.1. OpenFlow	14
2.2.2. Redes definidas por Software	15
2.2.2.1. Controlador	16

2.2.2.2. Switch OpenFlow	17
2.2.3. Mininet	17
2.2.4. OpenNet	18
2.2.5. Mininet-WiFi	19
2.2.6. Visual Network Description (VND)	19
2.3. Calidad de Software	19
2.3.1. Norma ISO/IEC 25000	19
2.3.2. Calidad del producto software	20
2.3.2.1. Características y subcaracterísticas de calidad interna- externa	22
2.3.2.2. Características y subcaracterísticas de la calidad de uso	23
2.4. Modelo de estimación de proyectos en entornos de incertidumbre (EPCU)	23
2.5. Experiencia de Usuario (UX)	25
2.5.1. Escala para la Usabilidad de los Sistemas (SUS).	27
2.6. Distribución Normal	29
2.7. Distribución Normal Estandarizada	31
2.8. Puntaje Z	31
2.9. Coeficiente Alfa de Cronbach (α)	32
3. Marco Metodológico	34
3.1. Etapas de la Investigación	35
3.1.1. Revisión Bibliográfica	35
3.1.2. Diagnóstico de la situación	35
3.1.3. Instalación y Configuración de los Servidores	36
3.1.4. Realizar pruebas pertinentes	36
3.1.5. Diseño e implementación de instrumentos de recolección de datos	36
3.1.6. Medición y Evaluación de la Calidad del Producto Software y de la Experiencia de Usuario	36
3.1.7. Elaboración del Informe Final	37
3.2. Población y Muestra	37
3.3. Técnicas e Instrumentos de Recolección de Datos	38

3.4. Técnicas de Análisis de Datos	40
3.5. Análisis Cuantitativo de la calidad del producto software	40
4. Presentación y Análisis de los Resultados	42
4.1. Diagnóstico de la situación	43
4.2. Instalación y Configuración de los servidores	44
4.3. Pruebas de rendimiento y uso del software	45
4.3.1. Pruebas con OpenNet	45
4.3.2. Pruebas con Mininet-WiFi	52
4.4. Medición y Evaluación de la Calidad del Producto Software y de la Experiencia de Usuario	58
4.4.1. Medición de la calidad del producto software de OpenNet	59
4.4.1.1. Calidad Interna/Externa de OpenNet	60
4.4.1.2. Calidad de Uso de OpenNet	62
4.4.2. Medición de la calidad del producto software de Mininet-WiFi	63
4.4.2.1. Calidad Interna/Externa de Mininet-WiFi	64
4.4.2.2. Calidad de Uso de Mininet-WiFi	65
4.4.2.3. Comparación de la Calidad interna/externa y de uso entre OpenNet y Mininet-WiFi	67
4.4.3. Medición de la Experiencia de Usuario	68
4.4.4. Interrelación de la Usabilidad con la UX en OpenNet	71
4.4.5. Interrelación de la Usabilidad con la UX en Mininet-WiFi	79
4.4.6. Índices de calidad del producto Software empleando el modelo EPCU	82
4.5. Discusión de Resultados	83
5. Conclusiones y Recomendaciones	85
Conclusiones	85
Recomendaciones	87
Bibliografía	89

A. Glosario de Términos	93
B. Instalación y Configuración de OpenNet y Mininet-WiFi	96
B.1. Instalación y configuración de OpenNet.	97
B.2. Instalación y configuración de Mininet-WiFi.	101
C. Instrumento de Validación de la Calidad del Producto Software de las aplicaciones SDWN, basado en la ISO/IEC 25000	104
D. Instrumento de evaluación de la escala para la usabilidad de los sistemas (SUS)	121
E. Análisis y Resultados Estadísticos	125
F. Scripts usados con OpenNet	178
G. Scripts usados con Mininet-WiFi	183

Índice de figuras

2.1. Comparación de trabajos relacionados sobre redes inalámbricas bajo diferentes plataformas experimentales.	13
2.2. Arquitectura lógica SDN	15
2.3. Modelo generalizado de un Switch OpenFlow.	17
2.4. La familia de normas ISO/IEC 25000	20
2.5. Modelo de referencia de Calidad de Software	21
2.6. modelo de calidad del producto definido por la ISO/IEC 25010	21
2.7. Pasos del Modelo EPCU	25
2.8. Contexto global de la Experiencia de Usuario	26
2.9. Escala de porcentaje de usabilidad de SUS	28
2.10. Rankings de Percentiles de SUS	28
2.11. Número de Usuarios requeridos para realizar un test	29
2.12. Distribución Normal Tipificada	30
4.1. Ejecución del Controlador POX	46
4.2. Consola de OpenNet al arrancar	46
4.3. Ayuda de OpenNet.	47
4.4. Información de los dispositivos existentes en la topología	47
4.5. Probando conectividad mediante paquetes ICMP	48
4.6. Terminal Xterm del host 1 (h1)	48
4.7. Parámetros de la NIC en una de las estaciones emuladas	48
4.8. Iperf cliente-servidor en consolas de Xterm en los host h1-h2	49
4.9. Gráfica del Throughput resultado de la prueba de rendimiento	50

4.10. NetAnim ejecutando la topología seleccionada	50
4.11. NetAnim realizando la simulación	51
4.12. Visualización de algunos parámetros en NetAnim	51
4.13. Escenario virtual sencillo de una SDWN usando VND	52
4.14. Red SDWN con VND	53
4.15. Consola principal de Mininet-WiFi	53
4.16. Opciones de Ayuda de Mininet-WiFi	54
4.17. Parámetros inalámbricos de la NIC en una de las estaciones emuladas	54
4.18. Parámetros de estaciones emuladas en Mininet-WiFi	55
4.19. Pruebas de conectividad con paquetes ICMP desde la consola xterm de uno de los nodos	55
4.20. Pruebas de rendimiento de nodos inalámbricos con Iperf	56
4.21. Gráfica del Jitter resultado de la prueba de rendimiento	56
4.22. Gráfica de Movilidad de nodos en Mininet-WiFi	57
4.23. Gráfica de Movilidad de nodos en Mininet-WiFi	57
4.24. Características de la Calidad Interna/Externa en OpenNet	60
4.25. Características de la Calidad de Uso para OpenNet	60
4.26. Evaluación de la calidad interna/externa en OpenNet	61
4.27. Índice de calidad interna/externa de OpenNet con mecanismo EPCU (con autorización de www.mepe.com.mx)	61
4.28. Evaluación de la calidad de Uso en OpenNet	62
4.29. Índice de calidad de Uso de OpenNet con mecanismo EPCU (con autoriza- ción de www.mepe.com.mx)	62
4.30. Características de la Calidad Interna/Externa en Mininet-WiFi	64
4.31. Características de la Calidad de Uso para Mininet-WiFi	64
4.32. Evaluación de la calidad interna/externa en Mininet-WiFi	65
4.33. Índice de calidad interna/externa de Mininet-WiFi con mecanismo EPCU (con autorización de www.mepe.com.mx)	65
4.34. Evaluación de la calidad de uso en Mininet-WiFi	66
4.35. Índice de calidad de Uso de Mininet-WiFi con mecanismo EPCU (con autorización de www.mepe.com.mx)	66

4.36. Comparación gráfica de la calidad interna/externa	67
4.37. Comparación gráfica de la calidad de uso	68
4.38. Valor Global SUS de OpenNet y Mininet-WiFi	70
4.39. Relación de los software SDWN	71
4.40. Puntuaciones SUS de OpenNet aproximado a la curva Normal	72
4.41. Valoraciones normalizadas para OpenNet	73
4.42. Curva Normal de las puntuaciones SUS de OpenNet a $N(0, 1)$	73
4.43. Curva Normal de puntuaciones SUS de OpenNet a $N(\mu, \sigma)$	74
4.44. Linealización de la curva Normal de los valores SUS de OpenNet	75
4.45. Curva Normal de las valoraciones de los expertos en OpenNet a $N(0, 1)$	76
4.46. Curva Normal de las valoraciones de los Expertos en OpenNet a $N(\mu, \sigma)$	76
4.47. Determinación del índice de la calidad de uso de OpenNet a partir de la media corregida (con autorización de www.mepe.com.mx).	78
4.48. Comparación de las características de la calidad de uso en OpenNet	79
4.49. Puntuaciones SUS de Mininet-WiFi aproximado a la curva Normal	79
4.50. Valoraciones normalizadas para Mininet-WiFi	80
4.51. Curva Normal de las valoraciones de los Expertos en Mininet-WiFi a $N(0, 1)$	81
4.52. Determinación del índice de la calidad de uso de Mininet-WiFi a partir de la media corregida (con autorización de www.mepe.com.mx).	82
4.53. Comparación de las características de la calidad de uso en Mininet-WiFi	82
B.1. Línea de tiempo de los cambios hechos en OpenNet	97
B.2. Comandos para instalar OpenNet	98
B.3. Ejecución del Controlador POX	98
B.4. Comandos para ejecutar OpenNet	98
B.5. Ejecución de OpenNet	99
B.6. Comandos para ejecutar NetAnim	99
B.7. Ejecución de NetAnim	99
B.8. Script de la tarea que se debe modificar en Ansible	100
B.9. Pasos para instalar Mininet-WiFi	101
B.10. Ejecución de Mininet-WiFi	102

B.11. Contribuciones de los desarrolladores a nivel de código al proyecto Mininet-
WiFi 103

Índice de Tablas

2.1. Comparación entre los Simuladores existentes y OpenNet	12
2.2. Características de algunos controladores SDN de código abierto.	16
2.3. Rango de Alfa de Cronbach	33
4.1. Características técnicas de los servidores	44
4.2. Estadística de Fiabilidad	58
4.3. Resultados del análisis estadístico descriptivo del cuestionario de calidad para OpenNet	59
4.4. Resultados del análisis estadístico descriptivo del cuestionario de calidad para Mininet-WiFi	63
4.5. Valores de evaluación de la Escala de usabilidad del sistema (SUS)	69
4.6. Resultados del análisis estadístico de la Escala de usabilidad del sistema (SUS)	69
4.7. Resultados Normalizados de la Usabilidad y UX de OpenNet	72
4.8. Resultados Normalizados de la Usabilidad y UX de Mininet-WiFi	80
4.9. Índice de la Calidad del producto Software basado en la norma ISO/IEC 25000	82

Análisis Comparativo de la Calidad en Aplicaciones de Emulación de Redes Inalámbricas Definidas por Software, utilizando el Modelo de Estimación de Proyectos en Entornos de Incertidumbre.

Gerlyn Eduardo Duarte¹, Francisco Valdés Souto², Jesús Calderón Vielma¹

1: Universidad de los Andes, Venezuela.

2: Universidad Nacional Autónoma de México.

Resumen

Las redes definidas por software (*SDN=Software Defined Network*) son una arquitectura emergente de redes de telecomunicaciones, destacándose como una tecnología capaz de ofrecer un control más flexible y dinámico a la hora de gestionar el diseño y funciones de la red, donde se separan los planos de control y enrutamiento. Esto permite la centralización de los controles de la red basada en software. Se puede decir que es posible tener flexibilidad y rapidez al momento de configurar, implementar y administrar diferentes protocolos personalizados para los requerimientos de distintas redes. Actualmente, en una red convencional, cada conmutador y/o enrutador tienen un software propietario que le dice qué actividad ejecutar. Generalmente los análisis y puesta en práctica de redes SDN se centra principalmente en la parte cableada de la red de datos. Sin embargo, todas las ventajas de las SDN sólo se obtienen si se logra abarcar la totalidad de la infraestructura, desde el centro de datos al extremo de la red, permitiendo extender todos los beneficios de SDN hasta la red inalámbrica, para conseguir a su vez una red inalámbrica de área local definida por software (*SDWN=Software Define Wireless Networks*). En este trabajo se evaluó la calidad del producto software de forma cuantitativa de dos programas desarrollados bajo fuentes abiertas denominados *OpenNet* y *Mininet-WiFi*, los cuales permiten emular y/o simular una SDWN. La evaluación de la calidad interna/externa y de uso del producto software de estos programas se apoyó en el conjunto de normas *ISO/IEC 25000*. Estas normas definen las categorías y subcategorías de calidad de forma cualitativa, pero no especifican un método estándar específico para realizar la evaluación. Para resolver esto se utilizó un modelo basado en lógica difusa denominado Modelo de Estimación de Proyectos en Entornos de Incertidumbre (*EPCU=Estimation of Projects in Contexts of Uncertainty*), lo que permitió determinar de manera formal y bajo criterios consistentes y repetibles, cual software es el que tiene mayor índice de calidad de producto. Adicionalmente se determinó la estimación de la experiencia de usuario (*UX=User eXperience*) con la intención de interrelacionarla con la usabilidad del producto software, utilizándola como un nuevo valor de entrada en la determinación de la calidad de uso del modelo EPCU. Por otra parte se efectuaron emulaciones de redes inalámbricas definidas por software de diferentes topologías permitiendo analizar la movilidad de los dispositivos emulados. De igual forma se ejecutaron mediciones del ancho de banda de las SDWN emuladas, así como la interconectividad entre ellas. Se evidenció la facilidad y el rendimiento de trabajar con las herramientas de emulación de SDWN estudiadas. Se comprobó que los emuladores, permiten interactuar y ejecutar cualquier proceso como si se ejecutara directamente en un equipo con GNU/Linux de forma eficaz.

Palabras Clave: SDN, Mininet, SDWN, EPCU, Calidad del producto software.

Comparative Analysis of Quality in Software-Defined Wireless Network-Emulating Applications by Using the Estimation of Projects in Contexts of Uncertainty Model.

Gerlyn Eduardo Duarte¹, Francisco Váldez Souto², Jesús Calderón Vielma¹

1: University of the Andes.

2: National Autonomous University of Mexico.

Abstract

Software-defined networks (SDN) are an emerging architecture of telecommunications network, standing as a technology capable of providing a more flexible and dynamic control in managing the design and functions of the network, where the control plane and routing are separated. This allows the centralization of control network based on software. It is said that it is possible to have flexibility and speed when configuring, deploying and managing various custom protocols for different network requirements. Currently, in a conventional network, each switch or router has its own software that tells it which activity must run. Generally, the analysis and implementation of SDN networks are mainly focused on the wired side of the data network. However, the full benefits of SDN are obtained only if it covers the entire infrastructure, from the data center to the network end, allowing the extension of the full benefits of SDN to the wireless network, and also allowing a local area wireless network to be defined by software, SDWN (Software Define Wireless Networks). In this work, the quality of the software product was evaluated quantitatively, for two software developed by open sources such as OpenNet and Mininet-WiFi, both software allow to simulate and emulate one SDWN. The evaluation of the internal/external quality and the software product usage quality of these software were supported on the set of ISO/IEC 25000 rules. These rules define the categories and subcategories of quality in a qualitative form, but it does not specify an specific standard method for doing the evaluation. To resolve this, it was used a model based on fuzzy logic techniques, using the Estimation of Projects in Contexts of Uncertainty (EPCU), which made possible to determine in a formal way and following a consistent and repeatable criteria, which software was the one with the highest quality, and the one that fill the users expectations as well. User experience estimation was determined equally (UX=User experience). On the other hand, wireless network emulations were defined by software of different topologies allowing analyze the movement of emulated devices. Similarly, the bandwidth measurements of the emulated SDWN and interconnectivity between them were also executed. Performance and ease of working with the SDWN-studied emulation tools were evidenced. Finally, it was found that the emulators allow you to interact and run any process as though if it were running directly on a computer with GNU/Linux in an effective way.

Keywords: Data Communications Network, SDN, Mininet, SDWN, EPCU, Software Quality product.

Capítulo 1

Introducción

1.1. Introducción

La actualización de la infraestructura de red hacia nuevas tecnologías y protocolos es una tarea engorrosa y costosa. Actualmente, si se desean realizar cambios significativos se requiere reemplazar los dispositivos de red por completo. En ese sentido, se puede decir que estas redes tal como están configuradas, siguen teniendo el impedimento de realizar cambios de modo dinámico, siendo esta su mayor dificultad desde el punto de vista de gestión y administración, es por ello que podemos afirmar que las mismas tienen poca escalabilidad y se han quedado obsoletas.

El diseño de las redes de datos actuales fue pensado para que el control de flujo y el enrutamiento se ejercieran directamente desde los dispositivos, formando parte de una estructura de gestión estática, jerárquica y completamente dependientes de la infraestructura de red. En un escenario real donde las redes actuales tienen la tendencia a la saturación: es difícil procesar toda la cantidad de información que se está generando desde los diferentes servicios que utilizan tecnología IP tales como: Voz sobre IP (VoIP), Transmisiones Streaming, BigData, Cloud Computing, Virtualización de servidores, Internet de las cosas (IoT), entre otros.

Otro punto a resaltar es la red de datos inalámbrica, ya que la constante instalación sea planificada o no, de dispositivos para el acceso inalámbrico, originan una interferencia recíproca y por lo tanto, la disminución del rendimiento de la red, convirtiéndose a la larga en un entorno inalámbrico inestable.

Esta problemática sería mejorada con la implementación de una red definida por software (*SDN=Software Defined Network*), en la red de área local inalámbrica (*WLAN=Wireless Local Area Network*), la cual brinda la posibilidad de tener una red inalámbrica centralizada y flexible mediante la aplicación de distintas reglas a diferentes flujos de datos, permitiendo así gestionar numerosos puntos de acceso en una WLAN tradicional, siendo esta una solución que depende menos de la intervención del hombre y sobre todo de un hardware más especializado en el procesamiento de datos; es allí donde entra en juego el protocolo OpenFlow y las redes SDN (Duarte and Lobo, 2015).

Una red inalámbrica definida por software (*SDWN=Software Defined Wireless Networks*) no sólo permite crear reglas en el flujo de datos, sino también controlar el comportamiento de los parámetros de los dispositivos inalámbricos (canales, control de acceso, potencia, etc.) (Frost and Sullivan, 2014).

A lo largo de este documento, los términos *SDN* y *SDWN* harán referencia respectivamente a redes definidas por software, y a redes inalámbricas definidas por software.

Los fundamentos iniciales de las SDN fueron creadas en el año 2007 en la Universidad de Standford por un grupo de trabajo conformado por Martín Casado, Nick McKeown, Scott

Shenker entre otros (Casado et al., 2007). Posteriormente McKeown et al. (2008) desarrolló el protocolo OpenFlow, en este protocolo es donde están basadas las SDN. Aunque fue en 2011 que Scott Shenker y Nick McKeown en conjunto con seis compañías del área (Deutsche Telekom, Facebook, Google, Microsoft, Verizon, y Yahoo) fundaron la Open Networking Foundation (ONF) (Erickson, 2011). Esta fundación buscaba fomentar el uso de OpenFlow, la creación de estándares y la implantación de SDN más allá de las universidades.

Hoy en día las SDN se están diseñando y estructurando en diferentes entornos de redes tanto cableadas como inalámbricas, entre los que más resaltan están los centros de datos de prestadores de servicio. Grandes compañías y consorcios están invirtiendo en ellas, tanto en la parte de investigación como en la producción y generación de dispositivos que soporten el protocolo OpenFlow. Es por ello que las SDN se presentan como la solución que ayudará a mejorar todos los problemas que hasta ahora han tenido los administradores de red a consecuencia de la complejidad e inflexibilidad de las redes actuales.

El objetivo principal de estas redes consiste en separar el plano de control del plano de datos en los dispositivos de una red.

Un controlador SDWN puede obtener información de los dispositivos alrededor de cada punto de acceso que contengan en su firmware el protocolo OpenFlow, permitiendo conocer toda la topología de la red. Con esa información, es posible reducir la interferencia de los puntos de acceso adyacentes mediante el control de la potencia. Por otra parte una SDWN puede hacer que múltiples puntos de acceso actúen como un solo gran punto de acceso inalámbrico virtual, para evitar desconexiones de enlaces innecesarios (Suresh et al., 2012).

De la misma manera Tseng et al. (2005) demostró que existe una mejora significativa en el rendimiento de una red inalámbrica al utilizar un controlador centralizado.

En un estudio realizado por Chan et al. (2014) se comparó a Mininet y ns-3, los cuales son los dos simuladores/emuladores más utilizados para las SDN. De igual forma en ese estudio se contrastó a EstiNet con los anteriores simuladores/emuladores; con la intención de ver el alcance para la simulación y/o emulación de las SDWN, ratificando que Mininet no soporta el modelado del canal inalámbrico ni la movilidad de los dispositivos. Por su parte, ns-3 cuenta con un apoyo limitado para los controladores definidos por software. EstiNet es un simulador comercial y por lo tanto no facilita la flexibilidad de modificación a nivel de su código fuente.

El resultado de esta investigación llegó a la conclusión de que los emuladores y/o simuladores existentes no son adecuados para SDWN. De igual forma, los autores exponen todas las limitaciones de esos programas para implementar SDWN en términos de apoyo, funcionalidad inalámbrica, compatibilidad de controladores y extensibilidad.

Es por ello que Chan et al. (2014) crearon un nuevo simulador/emulador de SDWN, el cual llamaron *OpenNet*, el mismo se construyó en base a Mininet y a ns-3. Estos dos

programas tienen una gran flexibilidad para desarrollar e implementar nuevos protocolos experimentales. Además, OpenNet hereda la compatibilidad con versiones anteriores de ns-3 y el controlador de compatibilidad de Mininet. OpenNet aprovecha las ventajas como software de emulación de Mininet y los módulos para canales inalámbricos y móviles de Ns-3, incluyendo la simulación de los puntos de acceso con soporte para OpenFlow. OpenNet está basado en código abierto (Chan, 2014).

Cabe destacar que Mininet por sí solo, no permite implementar ningún modelo ni topología inalámbrica, ya que no posee ningún modelo para canales inalámbricos ni movilidad.

Un trabajo reciente realizado por Fontes (2015), muestra un nuevo programa creado para la simulación de SDWN. El mismo es un Fork de Mininet, esta aplicación hace posible emular diferentes topologías de SDWN, la misma ha sido llamada *Mininet-WiFi* por su autor.

Mininet-WiFi se complementa con Visual Network Description (VND) la cual permite crear topologías de redes virtuales SDN desde una interfaz web-gráfica (Fontes et al., 2014). A partir de ahora al hacer referencia a Visual Network Description se simplificará con el acrónimo VND.

En los actuales momentos existe un gran interés por la calidad de los productos y servicios, no escapando el producto software de este interés.

Medir la calidad del producto software permitiría obtener valores cuantificables que ayudan a realizar algunas comparaciones entre dos o más programas. En otras palabras si no se mide, no hay una forma real de determinar cuál es el mejor o más óptimo.

El software hoy en día es cada vez más complejo y sofisticado, es por esto que está en aumento la necesidad de que los productos tengan buena calidad, la realización de la medición de la calidad del producto software de manera formal es complicada, pudiendo esta ser determinante a la hora de seleccionar el más adecuado, permitiendo ahorrar a las empresas recursos y tiempo valioso, entre otros.

Se debe resaltar que si bien la calidad es un concepto subjetivo, para el caso del software está definida en la familia de normas ISO/IEC 25000, la cual establece un marco de trabajo común para evaluar la calidad del producto software, proporcionando una guía sobre el uso de la nueva serie de estándares internacionales relacionada con la calidad del producto, guía que se conoce como Requisitos y Evaluación de Calidad de Productos de Software (*SQuaRE=System and Software Quality Requirements and Evaluation*), en adelante (SQuaRE).

Esta familia de normas especifican las categorías y subcategorías a evaluar de la calidad interna/externa y de uso del producto software, pero no establece de manera estandarizada cómo se deben de medirse estas.

Por lo anteriormente mencionado, con la finalidad de verificar cuantitativamente la calidad de los productos estudiados se emplea el modelo de estimación de proyectos en entornos de incertidumbre (*EPCU=Estimation of Projects in Contexts of Uncertainty*) diseñado por Valdés and Abran (2007), siendo aplicado por Valdés (2013) en la evaluación de las categorías de calidad interna/externa y de uso del producto software, este modelo se fundamenta en técnicas de lógica difusa aprovechando que este marco matemático permite manejar incertidumbre asociada a las variables subjetivas, permitiendo así usar mecanismos de evaluación cuantitativos de la calidad del producto software. Para referirnos a este modelo en este trabajo, en adelante se utilizarán las siglas (EPCU). Por otra parte actualmente no existen umbrales de métricas de calidad recomendados y aceptados universalmente.

El presente trabajo, se enfoca en la evaluación de la calidad del producto software que emule y/o simule SDWN en este caso OpenNet como en Mininet-WiFi utilizando para ello el modelo EPCU, centrándose en la creación de escenarios virtuales similares a los desarrollados en Mininet, posibilitando la experimentación con el protocolo OpenFlow.

Adicionalmente, se efectúan mediciones del ancho de banda de las SDWN emuladas con ambos software, así como la evaluación de la interconectividad entre los dispositivos móviles conectados para ver su eficiencia, lo mismo que analizar la capacidad de emular la movilidad de dispositivos inalámbricos, esto se realizó con la intención de comprobar el alcance de cada software para emular y/o simular las SDWN.

Otro factor importante a determinar hoy en día en el desarrollo de software es la experiencia de usuario (*UX=Users eXperience*). Pudiéndose caracterizar la experiencia del usuario a través de la calidad en uso del producto software González-Sánchez et al. (2012). Para efectos de este trabajo en adelante al referirse a la experiencia de usuario se simplificara como UX.

En este estudio se estima la UX para OpenNet y Mininet-WiFi como complemento a la evaluación de la calidad de uso la cual se determina por el modelo EPCU, para ello se utiliza un estándar diseñado para tal fin como es la Escala para la Usabilidad de los Sistemas (*SUS=System Usability Scale*). Los valores obtenidos en la escala SUS se tipificaran utilizando la normalización z-score para obtener un resultado compatible que servirá de entrada de la calidad de uso en el modelo EPCU, esto se plantea como un factor de corrección de la usabilidad en la medición de la calidad externa. Estandarizar o normalizar los valores permite comparar puntuaciones de dos distribuciones diferentes.

En la actualidad, se desarrolla un creciente interés por evaluar la calidad de uso y la UX de los productos software. Peñalva (2014) sugiere, que un modelo de medición de la calidad

de uso del producto software debe identificar las causas del interés y apego del usuario en una experiencia de interacción, el proceso debe ser dinámico y realizado en un lapso de tiempo que garantice que las ideas de los usuarios estén consolidadas ya que cambian según pasa el tiempo.

De igual forma González-Sánchez et al. (2012) indicó que en la UX toman importancia los aspectos subjetivos de la calidad de uso debido a la propia evolución del concepto de satisfacción. En los enfoques tradicionales la percepción de “satisfacción” era analizada según la eficacia y eficiencia, de modo que si los usuarios perciben como eficaz y eficiente el uso del producto, se da por hecho que están satisfechos. Pero hay aspectos de la UX, como placer, confort, o utilidad, que contribuyen de manera significativa a la satisfacción del usuario con el sistema. Factores que han llegado a formar parte del estándar de la calidad en uso ISO/IEC 25010.

Igualmente indica González-Sánchez et al. (2012), que debido a la gran cantidad de opciones a escoger que tiene el usuario, el concepto de usabilidad como principal referencia de la calidad, queda solapado por otro aspecto mucho más amplio como es la experiencia de usuario, la cual intenta profundizar dentro de la satisfacción del usuario para que las sensaciones del proceso de uso sean las mejores.

Por todo lo anteriormente citado, el proceso de medición y evaluación de la UX en esta investigación tiene su eje principal en determinar la calidad de uso, utilizando para ello la usabilidad determinada a partir del la UX, la cual servirá como valor de entrada para obtener un nuevo índice de la calidad de uso en el modelo EPCU, esto permite tener una perspectiva más amplia con un grupo de usuarios sin conocimientos de uso sobre los software empleados.

1.2. Planteamiento del problema

Las redes de telecomunicaciones actuales han sido concebidas tradicionalmente con una estructura compleja y con escasa flexibilidad, es allí donde se destaca la importancia de implementar una SDN.

A nivel de simulación y/o emulación de SDN se han creado varios programas que facilitan el diseño y pruebas de las mismas, siendo Mininet la herramienta más utilizada por la comunidad académica para desarrollar proyectos de investigación de emulación de SDN según resaltan Barona (2013) y Fontes (2015).

El principal problema al emular y/o simular se ha presentado en la parte de las redes inalámbricas, donde existen pocos software que permiten la simulación/emulación de estas redes, y en especial que sean de fuentes abiertas, ya que esta condición permite la adaptación de los mismos a cualquier SDWN a través de su código fuente. De igual forma se han creado unas pocas soluciones propietarias de código cerrado la cuales tienen un alto costo.

La emulación de redes ha sido ampliamente utilizada en la evaluación del desempeño, en los protocolos de pruebas de las mismas, así como en diversas cuestiones relacionadas con la investigación en redes de computadoras.

Es importante incorporar las redes inalámbricas a la infraestructura de una SDN, por lo tanto contar con un simulador y/o emulador que permita realizar pruebas en una SDWN es muy importante ya que permitirá a largo plazo crear SDWN bien diseñadas, redundantes y resilientes, y debiese ser tomado en cuenta por los administradores de red ya que esto les simplificará sus funciones.

La simulación/emulación de redes inalámbricas no es una tarea fácil, en la misma hay que controlar distintos factores, tales como la modulación, la interferencia electromagnética, la movilidad, la potencia, el canal, entre otros.

Otro factor que se debe destacar, es que mayoritariamente los estudios científicos sobre SDN, solo abarcan la parte cableada de las redes de telecomunicaciones.

Es por ello que son pocos los emuladores y/o simuladores de redes inalámbricas definidas por software existentes. En los artículos propuestos por Chan et al. (2014) y por Fontes et al. (2015) han creado dos simuladores y/o emuladores de SDWN desarrollados bajo código abierto.

De todo lo anteriormente señalado surgen las siguientes interrogantes:

¿De los programas existentes para la simulación y/o emulación de SDWN se puede determinar cuál es el que posee mejor calidad del producto software?

¿Las comparaciones actuales sobre software para SDWN han sido objetivas?

¿Se puede medir la calidad del producto software de las herramientas estudiadas de una forma objetiva y cuantitativa?

¿Se puede hacer una evaluación objetiva de la experiencia de usuario?

La norma SQuaRE permite que se realice la evaluación y comparación de la calidad interna/externa y de uso del producto software en los programas OpenNet y Mininet-WiFi, dejando claro que estas normas no establecen de manera estandarizada cómo deben ser medidas la calidad del producto software, por ello se apoyara en el modelo EPCU esta investigación, ya que permite realizar una evaluación de la calidad considerando los aspectos subjetivos de la calidad de uso como lo establece González-Sánchez et al. (2012), por tanto se considera que se ajusta a lo que se quiere alcanzar.

El modelo EPCU se basa en el criterio del experto en el proceso para determinar cuantitativamente la calidad del producto software.

Del mismo modo se destaca que las aplicaciones analizadas para emular SDWN carecen de suficiente información sobre su uso y estructura, la información encontrada es la brindada por los autores en sus artículos científicos y la misma es solo referencial.

Otra dificultad que se presenta es la de encontrar usuarios calificados que hayan utilizado estos programas que puedan llenar el instrumento de validación de la calidad interna/externa y de uso del producto software.

El estudio de la UX de los software de emulación/simulación de SDWN, cobra relevancia debido a su utilidad para mejorar la usabilidad de los sistemas. Por tal motivo es fundamental determinar la UX para conocer la apreciación del usuario final sobre la calidad del producto software. Actualmente la importancia de la UX se hace presente en el desarrollo y estudio de nuevos productos software. Una buena experiencia, satisface a los usuarios y la curva de adopción del producto se acelera. La calidad del producto software, junto con la calidad del proceso, se puede considerar como los aspectos más importantes actualmente en el desarrollo de Software. La UX se está convirtiendo en otro factor esencial a tomar en cuenta en la creación del producto software.

1.3. Justificación

Las SDWN se fundamentan al igual que las SDN en el protocolo OpenFlow, lo que les permite tener un control centralizado de los puntos de acceso inalámbrico. Este control centralizado posibilita la modificación de los diferentes parámetros inalámbricos, lo que facilitará la administración de las mismas.

Este estudio aporta una nueva visión de lo ya expuesto por Chan et al. (2014) y Fontes et al. (2015). El principal objetivo es evaluar la calidad del producto software en OpenNet y en Mininet-WiFi.

Se decidió integrar Mininet-WiFi con Visual Network Description (VND) otra herramienta hecha por el mismo desarrollador que creo Mininet-WiFi, las cuales se auto complementan.

Un adecuado aprovechamiento en el uso de estos programas en la etapa de diseño de las SDWN posibilita analizar los distintos entornos y parámetros modificables en las redes inalámbricas, permitiendo así poder emular diferentes SDWN.

Por tal motivo es de vital importancia esta investigación ya que evaluará y comparará de manera cuantitativa cuál de los dos programas analizados tendrán mejor índice de calidad del producto software según la familia de normas ISO/IEC 25000.

En principio la determinación de la calidad interna/externa y de uso del producto software se realiza con la ayuda del modelo EPCU, el mismo permite contrastar con argumentos válidos los estudios realizados por Chan et al. (2014) y Fontes et al. (2015), ya que los mismos no hacen referencia a los criterio tomados para la comparación realizada con los otros software, y por ende a los resultados obtenidos sobre el software más óptimo para simular y/o emular SDWN.

1.4. Delimitación de la Investigación

La prioridad en este estudio es evaluar y comparar la calidad del producto software de los programas OpenNet y de Mininet-WiFi con apoyo del modelo EPCU, estos programas permiten realizar emulación/simulación de SDWN. El estudio del modelo EPCU recae fuera de los propósitos de esta investigación, puesto que fue discutido en detalle por Valdés and Abran (2007), Valdés and Abran (2010), Valdés (2011), al igual que por Valdés (2012).

1.5. Objetivos

1.5.1. Objetivo general

Evaluar cuantitativamente la calidad interna/externa y de uso del producto software en las aplicaciones de emulación y/o simulación de redes inalámbricas definidas por software OpenNet y Mininet-WiFi, fundamentado en las especificaciones de la norma ISO/IEC 25000 y apoyado en el modelo EPCU.

1.5.2. Objetivos específicos

- Comparar el funcionamiento, desempeño y alcance de cada software para emular y/o simular las SDWN, realizando pruebas de rendimiento de diferentes escenarios en ambos software.

- Desarrollar los instrumentos de medición adecuados para determinar la calidad del producto software a medir.
- Evaluar la calidad del producto software de OpenNet y Mininet-WiFi, utilizando el modelo EPCU.
- Determinar el método de valoración para medir la Experiencia de Usuario (UX) de OpenNet y Mininet-WiFi, con la finalidad de integrarlo como una entrada de la variable usabilidad del modelo EPCU.

Capítulo 2

Marco Teórico

2.1. Antecedentes

La mayoría de los estudios en el campo de las SDN sólo se han centrado en las redes cableadas, son pocos los casos de estudio que se han enfocado en el análisis de redes inalámbricas definidas por software, más escasos son los estudios sobre emuladores y/o simuladores de estas redes.

En esta parte se tratan los antecedentes de la investigación que reflejan los avances y el estado actual de la simulación y/o emulación de las SDWN, los cuales sirven de modelo o ejemplo para futuras investigaciones. De la misma forma esta investigación se ha centrado en aquellos antecedentes que tuvieron relación directa o indirecta con el modelo EPCU, tópico muy importante del proyecto. A continuación se citan estos casos de estudio y se da una breve descripción de los resultados obtenidos en las mismas.

2.1.1. Simuladores de SDWN

Ya referido anteriormente el trabajo de Chan et al. (2014), el cual llevó a cabo un estudio que describe un análisis de las herramientas más utilizadas para simular/emular SDWN. La tabla 2.1 muestra la comparación de OpenNet con los otros emuladores realizada por los autores de la investigación.

Tabla 2.1: Comparación entre los Simuladores existentes y OpenNet

Simulador	Versión de OpenFlow	Simulación/Emulación	Funcionalidad Inalámbrica	Compatibilidad con Controladores	Extensibilidad
Mininet	1.3.1	E	No	Si	Si
NS-3	0.8.9	S/E	No Explora	No	Si
EstiNet	1.3.1	S/E	Si	Si	No
OpenNet	1.3.1	S/E	Si	Si	Si

Adaptado de: Chan et al. (2014)

Sun et al. (2015) analizó las propiedades de las SDN y sus ventajas de ser adoptada esta tecnología en entornos inalámbricos, dejando claro que la explotación de SDN en las redes inalámbricas se encuentra todavía en etapas iniciales.

Igualmente ya se hizo referencia del trabajo de Fontes et al. (2015), el cual desarrolló un software que permite emular SDWN, siendo el mismo una mejora de Mininet que soporta emulaciones inalámbricas. También realizó una comparación con los software que permiten la emulación y simulación de redes inalámbricas. La figura 2.1 muestra los resultados de dicha comparación.

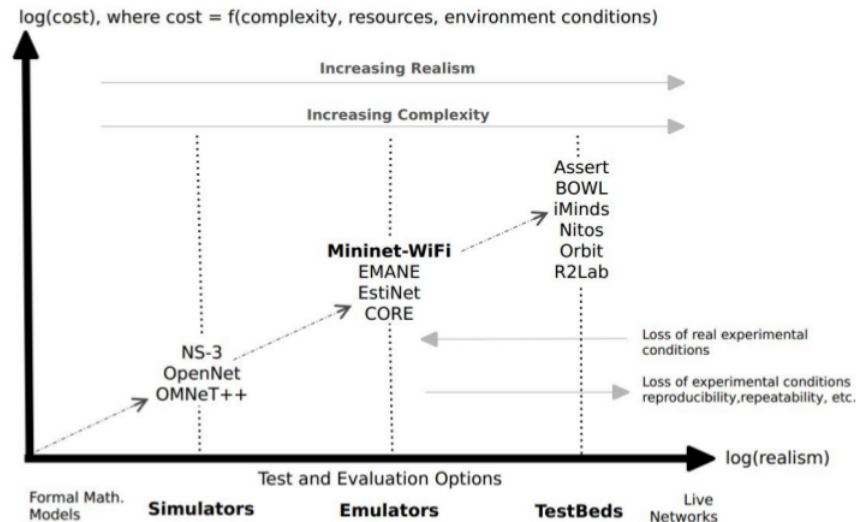


Figura 2.1: Comparación de trabajos relacionados sobre redes inalámbricas bajo diferentes plataformas experimentales.

Fuente: Fontes et al. (2015)

Hay que resaltar que en estas investigaciones ninguna especifica cuál es el mecanismo o norma que soporta los resultados de las comparaciones entre software realizadas.

2.1.2. Modelo EPCU

Valdés and Abran (2007) propusieron el modelo EPCU, este modelo está fundamentado en lógica difusa, y trata de resolver el problema de estimación de proyectos en su inicio, en ese momento es donde se encuentran una gran cantidad de incertidumbre y la información del proyecto es vaga (es decir, generalmente descrito por variables lingüísticas). “*El modelo es empleado generalmente en entornos de incertidumbre, es decir, cuando las decisiones se basan principalmente en variables cualitativas y en el juicio del experto, como lo es el caso de estimaciones o evaluación de algunos aspectos de los proyectos de software*”.

En las fases tempranas de un proyecto de desarrollo de software, la mayoría de las variables son lingüísticas, o cualitativas, y generalmente las estimaciones se desarrollan en un ambiente de incertidumbre. El modelo EPCU es útil en estas fases.

“*Este modelo considera los factores siguientes: Las variables lingüísticas usadas por los expertos para describir las variables de entrada del proceso de estimación en base a la experiencia (cuando estas entradas se basan en la información disponible de carácter impreciso o ambiguo para estimar un proyecto). La forma en que los expertos combinan los valores lingüísticos para la estimación del proyecto*”. Valdés (2011) formalizó y perfeccionó el modelo mediante un estudio de doctorado, fijando las bases conceptuales del mismo. “*Este modelo permite evaluar a partir del juicio de experto, mediante mecanismos formales de manejo de información imperfecta y con base en el estándar de calidad ISO/IEC 25000,*

2.2. DEFINICIÓN DE TÉRMINOS BÁSICOS SDN.

la calidad de los productos de software tanto desde el punto de vista de su construcción (calidad interna/externa) como desde el punto de vista del usuario (calidad de uso)”

Valdés (2013) realizó un estudio donde evaluó 44 proyectos de software para estimar la calidad del producto software de cada uno basándose en la norma ISO/IEC 25000 SQuaRE. En cada proyecto el juicio del experto fue el que determinó la asignación de valores a las variables de entrada. Se procedió a calcular el índice de calidad por proyecto utilizando el contexto EPCU definido para calidad interna/externa y de uso.

El modelo EPCU permite convertir la experiencia del experto en valores cuantitativos, dando convicción al usuario, de la evaluación de la calidad del producto de software.

En ese estudio Valdés (2013), se pudo observar que es posible obtener un valor cuantitativo formal, calculado de una misma forma para los distintos proyectos analizados basándose en las categorías de calidad definidas por SQuaRE, manejándolas como variables de entrada y en un contexto EPCU definido. El tener valores cuantitativos obtenidos formalmente permite realizar comparaciones de la calidad en cada software.

En la literatura no se encontraron estudios que analicen la calidad del producto software y la UX en el contexto de programas emuladores de SDN y SDWN.

2.2. Definición de términos básicos SDN.

2.2.1. OpenFlow

OpenFlow es una de las bases fundamentales de las redes SDN. Se trata de un protocolo de red Open Source utilizado para que un controlador gire instrucciones a los dispositivos de red en relación a que destinos debe enviar los paquetes. Para ello, las decisiones de envío de los paquetes se centralizan para que la red se pueda programar independientemente de los dispositivos existentes. OpenFlow es un protocolo que permite ejecutar procedimientos y especificaciones experimentales en las redes de datos de un campus. De esa manera, se puede implementar a un switch Ethernet, router y a puntos de acceso inalámbrico, solo cambiando el firmware (software alojado en el hardware) de dichos dispositivos, siempre y cuando este nuevo firmware tenga el soporte al protocolo, permitiendo esto definir los flujos de datos mediante software. Al instalar un firmware que contenga OpenFlow se puede tener acceso a las tablas de flujo, es decir, a las reglas que le indican a los routers y switches cómo direccionar el tráfico de la red. (McKeown et al., 2008), (HP-Networking, 2012).

OpenFlow facilita el acceso a los dispositivos de red para la programación simplificada mediante una interfaz estándar. La facilidad a la hora de programar permite configurar una capa de control solvente para poder centralizar la inteligencia de la red y brindar esa capacidad de programación tan pregonada por la tecnología SDN. Actualmente Google y

2.2. DEFINICIÓN DE TÉRMINOS BÁSICOS SDN.

Amazon han implementado SDN en toda su red, OpenFlow ha sido implementado en los nuevos dispositivos de red de los más importantes fabricantes.

Este protocolo ha sido diseñado para que se apoye en tres componentes: Las tablas de flujos instaladas en cada uno de los switches que indicarán a cada dispositivo que hacer con el tráfico. Estas tablas indican a los switches como deben proceder los paquetes a través de la red. Un canal seguro que es el que permite conectar a los switches con el controlador, es a través de este canal que ambos dispositivos “conversan” OpenFlow. El controlador, que será el componente principal y el que dialoga con todos los dispositivos y se encarga de centralizar y transmitir la información a la red, manteniendo un paso distribuido a través de los switches y routers.

2.2.2. Redes definidas por Software

Las redes definidas por software (SDN=Software Defined Network) son una arquitectura de red emergente que se destaca como una tecnología capaz de ofrecer un control más flexible y dinámico a la hora de gestionar el diseño y funciones de la red, donde se separan los planos de control y enrutamiento. Esto permite la centralización de los controles de la red basada en software. Hoy en día se han desarrollado muchos controladores SDN, tanto de código abierto como comerciales, se puede decir que es uno de los principales aspectos a tener en cuenta en el entorno actual al diseñar una SDN (HP-Networking, 2012).

La Figura 2.2 muestra a detalle la estructura de la arquitectura de una red definida por software. La cual está conformada por capas, la más baja que es la de infraestructura, que es donde se encuentran los dispositivos de red (switches OpenFlow), la segunda capa es la de Control y por último la de aplicaciones.

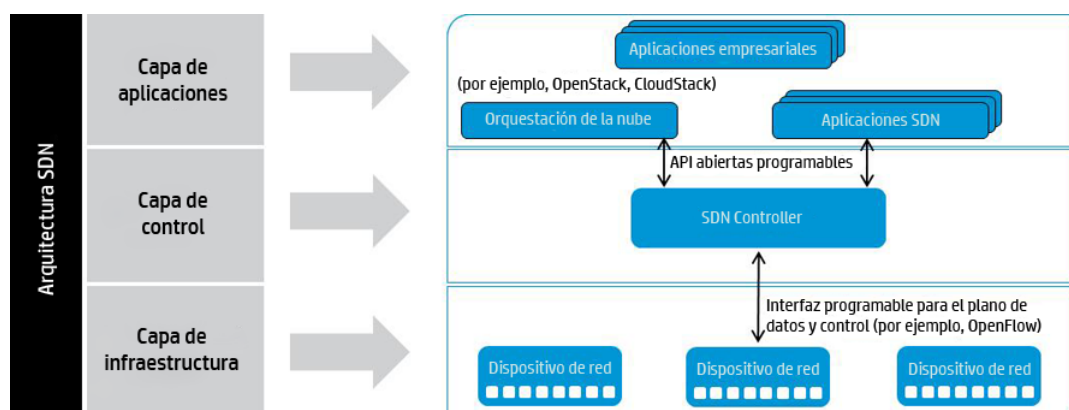


Figura 2.2: Arquitectura lógica SDN
Fuente: HP-Networking (2012)

En otras palabras las SDN son redes gestionadas remotamente de forma centralizada por un programa informático. Un software que genera un mapa esquemático de la red y sobre

2.2. DEFINICIÓN DE TÉRMINOS BÁSICOS SDN.

el que puede controlarse el flujo de datos. En las redes actuales, si por ejemplo queremos cambiar la manera como un dispositivo (un router, o un switch) gestiona cierto tráfico, se debe intervenir sobre él de forma manual. Con las SDN, no es necesario y pueden cambiarse todas las configuraciones de los dispositivos de red desde un control centralizado, lo que agiliza y optimiza el tráfico (Uceda, 2012).

2.2.2.1. Controlador

Un controlador es el componente principal de una SDN, el mismo es el encargado de gestionar los flujos de datos y de controlar las comunicaciones entre los dispositivos y las aplicaciones, basándose para esto del protocolo OpenFlow. El controlador es un dispositivo físico que hace las funciones de servidor. El controlador es el que definirá los parámetros de cada flujo de datos y cómo deben ser procesados los paquetes hacia el switch OpenFlow que corresponde.

En la tabla 2.2 podemos ver alguna de las características presentes en los principales controladores bajo código abierto existentes.

Tabla 2.2: Características de algunos controladores SDN de código abierto.

	Beacon	Floodlight	NOX	POX	Trema	Ryu	ODL
Soporte OpenFlow	OF v1.0	OF v1.0	OF v1.0	OF v1.0	OF v1.3	OF v1.0, v1.2, v1.3 y extensiones Nicira	OF v1.0
Virtualización	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Mininet y Open vSwitch	Construcción de una herramienta virtual de simulación	Mininet y Open vSwitch	Mininet y Open vSwitch
Lenguaje de desarrollo	Java	Java	C++	Python	Rudy/C	Python	Java
Provee REST API	No	Si	No	No	Si (Básica)	Si (Básica)	Si
Interfaz Gráfica	Web	Web	Python+, QT4	Python+, QT4, Web	No	Web	Web
Soporte de plataformas	Linux, Mac OS, Windows y Android para móviles	Linux, Mac OS, Windows	Linux	Linux, Mac OS, Windows	Linux	Linux	Linux, Mac OS, Windows
Soporte de OpenStack	No	Si	No	No	Si	Si	Si
Multiprocesos	Si	Si	Si	No	Si	No	Si
Código Abierto	Si	Si	Si	Si	Si	Si	Si
Tiempo en el mercado	4 años	2 años	6 años	1 años	2 años	1 años	5 meses
Documentación	Buena	Buena	Media	Pobre	Media	Media	Media

Fuente: Centeno et al. (2014)

2.2.2.2. Switch OpenFlow

El switch OpenFlow es un dispositivo de conmutación que tiene un firmware que soporta el protocolo OpenFlow, diferenciándose principalmente en la forma cómo recibe las políticas de control del flujo y cómo hace la actualización de las tablas de flujo; este conmutador usa un control externo por medio del cual agrega, modifica o elimina reglas en las tablas de flujo ya existentes en cada uno de los switches y para agregar o eliminar nuevas tablas de flujo con sus reglas (Park and Baack, 2012).

El switch OpenFlow está compuesto por las tablas de flujos instaladas en cada uno de ellos, además ellas indicarán a cada dispositivo como deben proceder los paquetes a través de la red, se puede decir que las mismas son el hardware del switch. Otro componente del switch OpenFlow es el canal seguro que es el que permite conectar a los switches con el controlador, es a través de este canal que ambos dispositivos intercambian se comunican mediante el protocolo OpenFlow. El canal seguro es el componente software de un switch OpenFlow.

La Figura 2.3 muestra los componentes principales de un switch OpenFlow, pudiendo visualizar cada uno de los componentes mencionados anteriormente.

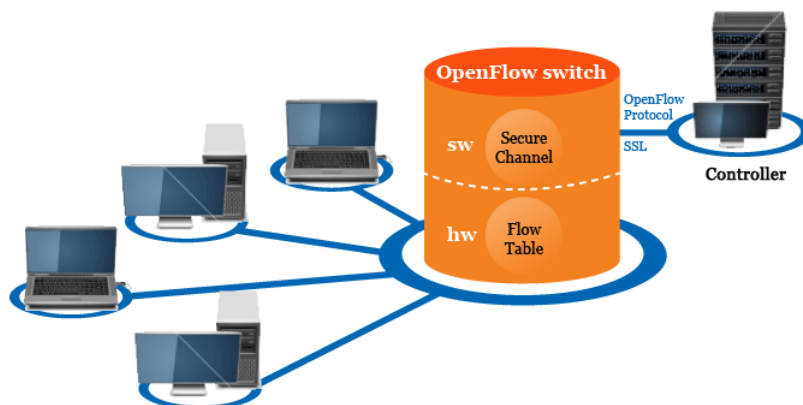


Figura 2.3: Modelo generalizado de un Switch OpenFlow.
Fuente: Xinguard (2014)

2.2.3. Mininet

Es un emulador de red que permite la creación de escenarios de redes virtuales. Mininet es un proyecto Open Source con una licencia BSD. Está basado en las funcionalidades que presta la virtualización de un sistema operativo GNU/Linux, permitiendo la creación de cientos de nodos, ya sean estos switches, routers o controladores en un solo ordenador. La virtualización con GNU/Linux permite el funcionamiento de procesos individuales con interfaces, tablas ARP y de enrutamiento diferentes. Fue elaborado con el objeto de que

2.2. DEFINICIÓN DE TÉRMINOS BÁSICOS SDN.

cualquier persona, lo pueda descargar, correr, evaluar, explorar, realizar cambios sobre entornos previamente creados ó mejor aún diseñar sus propios escenarios de prueba.

La mayor parte del código de Mininet es desarrollada con Python y una pequeña parte con lenguaje C. Adicionalmente Mininet usa una interfaz de línea de comandos (CLI) para crear, gestionar y compartir los escenarios creados.

Una de las principales características de Mininet es que sus switches soportan el protocolo OpenFlow y por tanto permiten la experimentación de SDN.

Los programas que se ejecutan en Mininet pueden enviar paquetes a través de lo que parece ser una interfaz de Ethernet real, con una velocidad de enlace y con retardo. Los paquetes se procesan por lo que parecen verdaderos dispositivos de red, con una determinada cantidad de colas.

Mininet ayuda a emular una red definida por software y experimentar con OpenFlow. Mininet brinda una gran cantidad de ventajas al emular una red. Es rápido y fácil de ejecutar, la puesta en marcha de una red emulada tarda sólo unos segundos. Mininet permite crear topologías personalizadas de un solo switch, grandes topologías como Internet, la red troncal de un campus, un centro de datos, o cualquier otra topología de red (Barona, 2013; Lantz et al., 2010); (Duarte and Lobo, 2015).

Todos los programas que se ejecuten en GNU/Linux está disponible para que funcionen en Mininet, desde los servidores de Internet hasta las herramientas de TCP para el monitoreo de paquetes.

Mininet se puede ejecutar en cualquier equipo de computación desde un portátil, en un servidor, en una máquina virtual, en una máquina con GNU/Linux nativo (Mininet se incluye con Ubuntu 12.10 en adelante), o en la nube (por ejemplo, Amazon EC2) Se puede crear y ejecutar experimentos en Mininet escribiendo simples (o complejos de ser necesario) scripts de Python (Barona, 2013; Lantz et al., 2010).

2.2.4. OpenNet

OpenNet es una modificación de Mininet combinado con ns-3. Esta aplicación fue creada por Chan et al. (2014). La principal razón de esta combinación es la posibilidad que brinda ns-3 para la conexión inalámbrica y la movilidad, Mininet sólo permite realizar simulaciones de redes cableadas. A través de esta unión, SDN se puede aplicar en redes de tipo inalámbricas, además de las redes cableadas.

2.2.5. Mininet-WiFi

Fontes et al. (2015) define que Mininet-WiFi es una modificación de Mininet, la cual permite la emulación de redes inalámbricas definidas por software. Mininet-WiFi añade estaciones virtualizadas WiFi (STA) y puntos de acceso inalámbricos (APs), basado en los controladores de los dispositivos móviles más comunes en GNU/Linux. Soporta la mayoría de los controladores inalámbricos del sistema operativo GNU/Linux. Las características proporcionadas por los controladores inalámbricos permiten que MiniNet-WiFi tenga un control preciso sobre los paquetes de las redes inalámbricas emuladas. MiniNet-WiFi está siendo desarrollada como una extensión limpia de alta fidelidad del emulador Mininet, añadiendo nuevas abstracciones y clases de apoyo a los controladores inalámbricos de GNU/Linux.

2.2.6. Visual Network Description (VND)

Es una plataforma web la cual tiene una interfaz gráfica que permite la creación de topologías de SDN para realizar la simulación y el análisis de las mismas. VND permite realizar experimentos de SDN, generando scripts para usar en Mininet. Estos scripts están relacionados con el escenario de la red, las tablas de flujo, y las configuraciones de QoS creadas en una interfaz gráfica sencilla y amigable. VND también se puede instalar y ejecutar localmente configurando un servidor web.

VND tiene como objetivo facilitar la creación de escenarios de red generalmente (experimentos) para su posterior simulación bajo diferentes herramientas OpenFlow, como Mininet. Para ello, VND exporta scripts de Python con la descripción de las normas y tablas de flujo con el fin de ser ejecutado en controladores OpenFlow. Además permite la interoperabilidad ya que proporciona también la generación automática de archivos NSDL, los cuales pueden ser importados por cualquier herramienta de simulación y análisis NSDL. También genera archivos compatibles con ns-3 (Fontes, 2013; Fontes et al., 2014).

2.3. Calidad de Software

2.3.1. Norma ISO/IEC 25000

La norma ISO/IEC 25000 define un modelo detallado para el uso de las nuevas series y estándares internacionales, el mismo ha sido llamado Requisitos y Evaluación de Calidad de Productos de Software - SQuaRE. Su principal objetivo es guiar la evaluación de calidad de productos software estableciendo criterios para la especificación de requisitos de calidad de software, sus métricas y su evaluación.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones. Cuyo objetivo principal es guiar el desarrollo de los productos de software mediante la especificación de requisitos y evaluación de características de calidad (ISO25000.com, 2014). La figura 2.4 muestra como está estructurada la familia de normas ISO/IEC 25000.



Figura 2.4: La familia de normas ISO/IEC 25000
Fuente: ISO25000.com (2014)

“ISO/IEC 25010 El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado” (ISO25000.com, 2014).

2.3.2. Calidad del producto software

Se define como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.

Según Piedrahita (2007) el modelo de calidad propuesto por SQuaRE se divide en dos partes: calidad interna/externa, y calidad en uso. La calidad interna se relaciona con las propiedades estáticas del software; la externa hace alusión a las características asociadas con la ejecución del software en el hardware y el sistema operativo. En términos generales, la calidad interna/externa puede ser definida como una sola, debiendo ser vista esencialmente como la calidad relativa a la construcción del software. La calidad en uso considera la calidad del sistema en su ambiente operacional para los usuarios, para las tareas específicas que

realizan, en otras palabras está relacionada al uso del software. El modelo SQuaRE supone que la calidad de los procesos presenta dependencia tanto de la calidad interna/externa, como de la calidad en el uso; y estas últimas, a su vez, se ven influenciadas por la calidad en el desarrollo de software, tal como se observa en la Figura 2.5

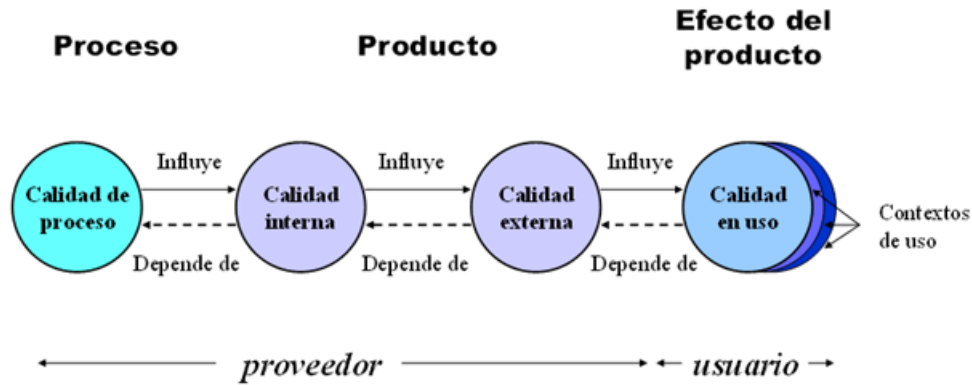


Figura 2.5: Modelo de referencia de Calidad de Software
Fuente: ISO/IEC-25010 (2007)

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad como se puede verificar en la figura 2.6:



Figura 2.6: modelo de calidad del producto definido por la ISO/IEC 25010
Fuente: ISO25000.com (2014)

De acuerdo a la norma ISO/IEC 25000 como referencia Valdés (2013): “*este modelo evalúa la calidad desde dos perspectivas: la calidad del producto como tal (calidad interna/externa), y la calidad de uso del software (externa). Cada perspectiva considera diversas características, y a su vez cada característica puede tener una o más subcaracterísticas*”.

Valdés (2013) utilizó las categorías definidas por SQuaRe para especificar las características y subcaracterísticas de la calidad interna/externa del producto software de la siguiente forma:

2.3.2.1. Características y subcaracterísticas de calidad interna-externa

1. *Adecuación funcional: Conjunto de atributos que atañen a la existencia de un conjunto de funciones y sus propiedades específicas. Estas funciones satisfacen las necesidades implícitas y establecidas. Consideran las subcaracterísticas de Adecuación, Precisión y Adherencia a Normas.*
2. *Fiabilidad: Conjunto de atributos que atañen a la capacidad del software para mantener su nivel de prestación bajo condiciones establecidas durante un tiempo establecido. Considera las subcaracterísticas de: Disponibilidad, Tolerancia a fallos, Recuperación y Adherencia a Normas.*
3. *Eficiencia de Rendimiento: Capacidad del producto de software para proporcionar un rendimiento apropiado relacionado con el total de recursos utilizados bajo condiciones establecidas. Considera las subcaracterísticas de: Tiempo de respuesta, Utilización de recursos, Adherencia a Normas.*
4. *Operabilidad: Capacidad del producto software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo ciertas condiciones específicas. Considera las subcaracterísticas de: Reconocimiento de adecuación, Capacidad de aprendizaje, Facilidad de uso, Útil, Atractivo, Accesible técnicamente, Adherencia a Normas.*
5. *Seguridad: Capacidad del producto software para alcanzar niveles aceptables de riesgo hacia la gente, negocio, software, propiedad o medio ambiente, en un contexto específico de uso. Considera las subcaracterísticas de: Confidencialidad, Integridad, No rechazo, Responsabilidad, Autenticidad, Adherencia a Normas.*
6. *Compatibilidad: Capacidad del producto software de coexistir con otros programas independientes en un entorno común y compartiendo recursos también comunes. Considera las subcaracterísticas de: Capacidad de replazo, capacidad de coexistencia, Interoperabilidad, Adherencia a Normas.*
7. *Mantenibilidad: Capacidad del producto software para ser modificado. Considera las subcaracterísticas de: Modularidad, Reusabilidad, Capacidad de análisis, Capacidad de cambios, Estable modificación, Capacidad de Testing, Adherencia a Normas.*
8. *Portabilidad: Capacidad del producto software para ser transferido de un entorno a otro. El entorno se interpreta tanto a nivel software y hardware, como aquel entorno relacionado con la organización. Considera las subcaracterísticas de: Portabilidad, Adaptabilidad, Capacidad de instalación, Adherencia a Normas.*

2.4. MODELO DE ESTIMACIÓN DE PROYECTOS EN ENTORNOS DE INCERTIDUMBRE (EPCU)

Dentro de la norma ISO 25000 (2005) también se encuentra la evaluación de la calidad de uso, la cual podemos definir como “la capacidad del producto software para permitir a determinados usuarios alcanzar objetivos especificados con efectividad, productividad, seguridad y satisfacción en contextos de uso especificados”. Como se puede apreciar en esta parte de la norma, la efectividad es una característica que se evalúa implícitamente a través de una serie de características como la eficiencia y la funcionalidad (ISO/IEC-25010, 2007; ISO25000.com, 2014).

Basándose en las normas SQuaRE Valdés (2013) detalla las características y subcaracterísticas de la calidad de uso de la siguiente forma:

2.3.2.2. Características y subcaracterísticas de la calidad de uso

1. *Usabilidad: Capacidad del producto para permitir a los usuarios alcanzar objetivos específicos con precisión en un contexto específico de uso. Considera las subcaracterísticas de Efectividad de uso, Eficiencia de uso, Satisfacción de uso, Adherencia a Normas.*
2. *Flexibilidad de Uso: Conformidad de contexto de uso, extensión de contexto de uso y accesibilidad de uso. Considera las subcaracterísticas de: Conformidad de contexto de uso, Extensión de contexto de uso, Accesibilidad de uso, Adherencia a Normas.*
3. *Seguridad de Uso: Capacidad del producto para alcanzar niveles aceptables de riesgo y proporcionar seguridad en el entorno. Considera las subcaracterísticas de: Seguridad y salud del operador, Seguridad y salud pública, Daño al entorno de uso, Daños comerciales de uso, Adherencia a Normas.*

Valdés (2013) destaca que: “Aunque algunas de estas características se pueden medir de manera objetiva, la mayor parte requieren una evaluación subjetiva. Esto hace que la manera más utilizada, más rápida, menos costosa y quizá la que más refleja la realidad, sea la utilización de la experiencia de los empleados de una organización. Sin embargo, esto no permite realizar comparaciones objetivas y presenta algunos otros problemas como que le pertenece al experto y no a la organización, no se puede replicar sistemáticamente y no contribuye a la madurez de la ingeniería de software.”

2.4. Modelo de estimación de proyectos en entornos de incertidumbre (EPCU)

El modelo EPCU fue propuesto por Valdés and Abran (2007), fundamentado en lógica difusa, la cual es un marco matemático que permite trabajar con incertidumbre. La lógica difusa está probada en gran medida en ámbitos industriales de forma eficiente, lo que permite que el modelo pueda ser empleado en entornos borrosos o de incertidumbre en distintas áreas.

2.4. MODELO DE ESTIMACIÓN DE PROYECTOS EN ENTORNOS DE INCERTIDUMBRE (EPCU)

Este modelo permite solucionar el problema que se genera al estimar proyectos en las primeras etapas del ciclo de vida del mismo, siendo en estas etapas donde se encuentra una gran cantidad de incertidumbre, donde la información del proyecto generalmente es cualitativa. Para la evaluación y determinación de variables cualitativas el modelo EPCU puede ser utilizado.

El modelo parte de las opiniones o juicio de expertos en el proceso a estudiar, ya que el mismo permite obtener de una serie de variables subjetivas generadas en un entorno de información imperfecta (información vaga o ambigua), o para ser más específicos aquella información cuyos datos se describen generalmente utilizando variables lingüísticas, pudiendo las mismas convertirse en entornos o en variables cuantificables (Valdés, 2013).

Valdés (2012) señala que el modelo EPCU permite determinar un valor cuantitativo suficientemente razonable, generado por la evaluación de reglas de inferencia definidas por los expertos. De manera descriptiva el modelo realiza una serie de pasos dónde se identifican las variables de entrada, en esta investigación las mismas son las categorías de la calidad interna/externa y de uso del producto software.

El modelo EPCU se compone de seis pasos como se muestra en la figura 2.7, se debe resaltar que los tres (3) primeros pasos están relacionados al proceso de configuración del contexto del modelo, estos pasos son realizados por los expertos y se refieren a la configuración del modelo o generación de un contexto a utilizar denominado “Contexto EPCU”. Valdés (2012) define como contexto EPCU al conjunto de variables (de entrada y salida) y las relaciones entre estas que puedan afectar a un proyecto o conjunto de proyectos similares, de manera simplificada aquí es donde se definen las variables de entrada y salida, así como sus relaciones entre sí de manera lingüística.

Una vez que se ha establecido el contexto EPCU, el mismo puede ser utilizado para realizar la estimación/evaluación con independencia de los expertos, esto hace que las empresas tengan un activo basado en la experiencia de los expertos sin necesidad de contar con ellos, ya que el modelo EPCU habilita la replicación sistemática de conocimiento (Valdés, 2012).

Los últimos tres (3) pasos que se muestran en la figura 2.7 es donde se utiliza el modelo EPCU.

Resumiendo se puede indicar que el modelo EPCU se desarrolla en los siguientes pasos:

- Especificación de las variables de entrada.
- Especificación de la variable de salida.
- Generación de las reglas de inferencia.
- Fusificación.

- Evaluación de las reglas de inferencia
- Defusificación

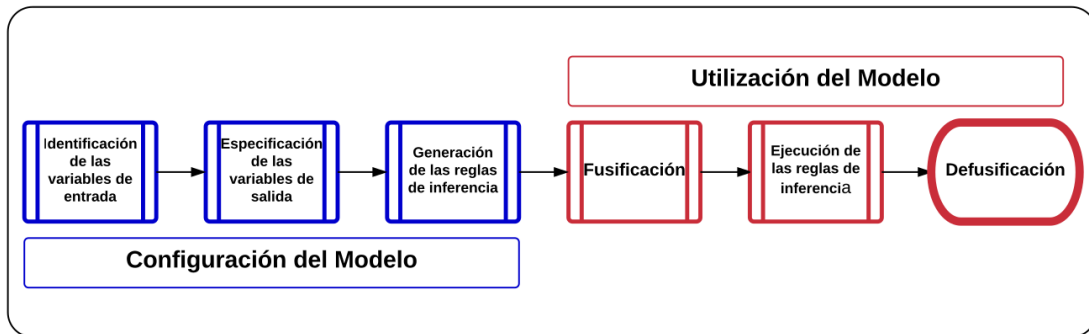


Figura 2.7: Pasos del Modelo EPCU
Adaptado de: Valdés (2011)

2.5. Experiencia de Usuario (UX)

La Experiencia de Usuario (User eXperience=UX) se refiere a lo que experimenta el usuario antes, durante y después de interactuar con el producto. Si no se incorpora al usuario, no se puede hacer UX. El diseño de la experiencia de usuario resulta fundamental hoy en día, porque comprender a los usuarios y sus verdaderas motivaciones y necesidades, permite considerar desde ese lugar qué interfaz, qué contenidos y qué interacciones lograrán el resultado buscado, y finalmente, validar con los usuarios los resultados que produce la interfaz propuesta. La UX es por tanto la percepción dejada en la mente de alguien después de una interacción con el producto según Kambrica (2014) y Zapata (2011).

La validación de UX se puede hacer de forma directa a través de pruebas con usuarios, entrevistas cualitativas, relevamiento de modelos mentales, o de forma indirecta realizando heat maps, A/B testing, click maps. La diferencia entre validación directa e indirecta, no es solamente metodológica. La validación directa suele ser cualitativa, es decir, permite saber por qué los usuarios hacen lo que hacen, mientras que la validación indirecta muestra que es lo que hacen, pero no dice por qué. Por eso, es bueno validar en primer lugar la experiencia de forma cualitativa como “ajuste grueso”, y luego en forma cuantitativa para buscar el “ajuste fino” (Kambrica, 2014).

Aunque la interacción humano computador (HIC) al igual que la UX son temas que fueron creados en los años 90, a nivel de Ingeniería de software comienza ser relevante su uso. Mayoritariamente la UX ha sido tomada en cuenta por el marketing y diseño de productos distintos al software. A partir de la última década se ha vuelto algo común en los equipos de desarrollo y diseño de software de todo tipo. Esto último se debe al auge que existe hoy en día por la gran variedad y cantidad de dispositivos táctiles como tabletas y

Smartphone lo que ha creado un gran interés en los desarrolladores que están creando sus aplicaciones centradas en la UX. Se puede afirmar que hoy en día la experiencia que tienen las personas al utilizar los productos y servicios interactivos que ofrece una organización es una de las nuevas áreas en las que la UX permite incrementar de modo notable la calidad de uso. La UX ayuda a que sean altamente satisfactorios el uso de los servicios tecnológicos que ofrecen las empresas hoy en día, incrementando la rentabilidad de las mismas.

En la figura 2.8 podemos ver la importancia y el contexto que abarca hoy en día la UX en el diseño y creación de software.



Figura 2.8: Contexto global de la Experiencia de Usuario
Fuente: Romero (2011)

Estudios realizados en UX como los de Hassan Montero and Martín Fernández (2005); Ortega Santamaría (2011) han ratificado lo propuesto por D'Hertefelt (2000) sobre que la UX representa un cambio emergente del propio concepto de usabilidad, donde el objetivo no se limita a mejorar el rendimiento del usuario en la interacción, eficacia, eficiencia y la facilidad de aprendizaje, sino que se intenta resolver el problema estratégico de la utilidad del producto y el problema psicológico del placer y diversión de su uso.

La UX y la usabilidad son claves en el ciclo de vida iterativo del diseño y desarrollo de cualquier producto interactivo o interfaz. Pero ¿cómo saber si en un sistema se están teniendo en cuenta y en qué grado se está haciendo? ¿En qué basarse para afirmar que un software tiene una mejor experiencia de usuario que otro? ¿Cómo detectar si se están cometiendo errores y cómo priorizarlos?.

Existen técnicas asociadas a la experiencia de usuario (UX) tales como encuestas de Uso, Test de Tareas y Test de Tarjetas de Uso, mediante el análisis de éstos llevando a cabo determinadas tareas en el sistema que se pretende evaluar o bien mediante focus groups que permitan a los usuarios opinar sobre determinados aspectos del sistema. También se puede realizar una inspección o evaluación sin usuarios llevadas a cabo por expertos o profesionales de la usabilidad (Zapata, 2011).

La UX es hoy en día un tema fundamental a tener en cuenta en el desarrollo de cualquier producto y el desarrollo de software no escapa de este contexto. Muchos de los principios básicos permanecen desde su creación, pero el desarrollo de nuevas tecnologías como: apps, dispositivos táctiles, wearables, interfaces basadas en gestos, o en emociones, hace que las herramientas de UX estén evolucionado con estas nuevas posibilidades. Es por ello que la evaluación de la UX está tomando relativa importancia, ya no es solo vista como herramienta de marketing, la misma va interrelacionada con la usabilidad. Hoy en día los usuarios demandan que todos los productos tengan una gratificante UX, es allí donde radica la importancia de la UX.

Así mismo González-Sánchez et al. (2012) señala que se deben considerar otros aspectos como: *que la usabilidad es 'absorbida' dentro de la UX; en segundo lugar, UX es una forma elaborada de satisfacción (una de las tres propiedades básicas de la usabilidad). La usabilidad es utilizada para medir y evaluar los objetivos pragmáticos (orientados a cómo el usuario realiza la tarea), mientras la UX (como forma elaborada de la satisfacción) se centra en la evaluación de los objetivos hedónicos (cómo se relaciona, siente y aprecia el usuario la realización de una tarea determinada). En ambos casos estas vertientes se intentan cubrir dentro del nuevo estándar de la calidad en uso ISO/IEC 25010.*

2.5.1. Escala para la Usabilidad de los Sistemas (SUS).

La Escala para la Usabilidad de los Sistemas (SUS=System Usability Scale) fue desarrollada por Brooke (1996) motivado por la necesidad de medir las opiniones subjetivas de los usuarios sobre el uso de los software, realizando una validación directa de UX.

La escala SUS se aplica después que el usuario tuvo la oportunidad de utilizar el software, la misma se evalúa inmediatamente, antes que el usuario tuviese la oportunidad de pensar largamente o discutir sobre el uso del software. La escala SUS fue desarrollado en 1986 como parte de la introducción de la ingeniería de usabilidad a los sistemas de oficina de Digital Equipment Co. Ltd., la misma se desarrolla en una escala de Likert de 1 a 5 y está compuesta por diez (10) preguntas cerradas. SUS es aplicable a software, páginas web, dispositivos electrónicos Smart entre otros.

La figura 2.9 representa una escala de porcentaje de usabilidad hecha por Bangor et al. (2009), esta escala esta derivada tras valorar más de 3500 test SUS, la misma hoy en día sirve de referencia para verificar en que porcentaje de usabilidad se encuentra su evaluación. Se debe aclarar que la escala no son porcentajes matemáticos si no lo que llamaron porcentaje de usabilidad basado en el valor obtenido al realizar el test.

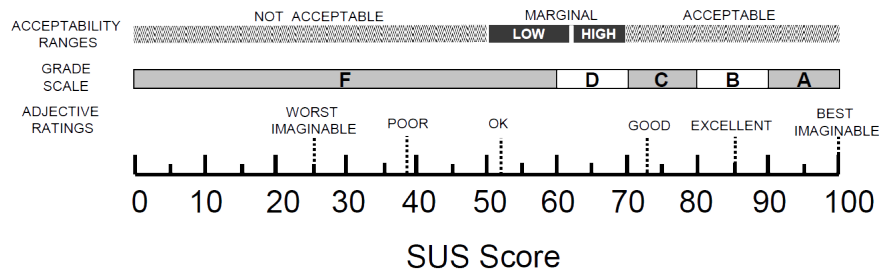


Figura 2.9: Escala de porcentaje de usabilidad de SUS
Fuente: Bangor et al. (2009)

Posteriormente Sauro (2011), crea una distribución aproximada de las puntuaciones SUS sobre la base de más de 5.000 observaciones de evaluaciones en términos de ranking de percentiles, la cual se muestra en la figura 2.10, siendo creada a partir de la escala de calificaciones de Bangor et al. (2009). Esta distribución fue creada normalizando las puntuaciones SUS.

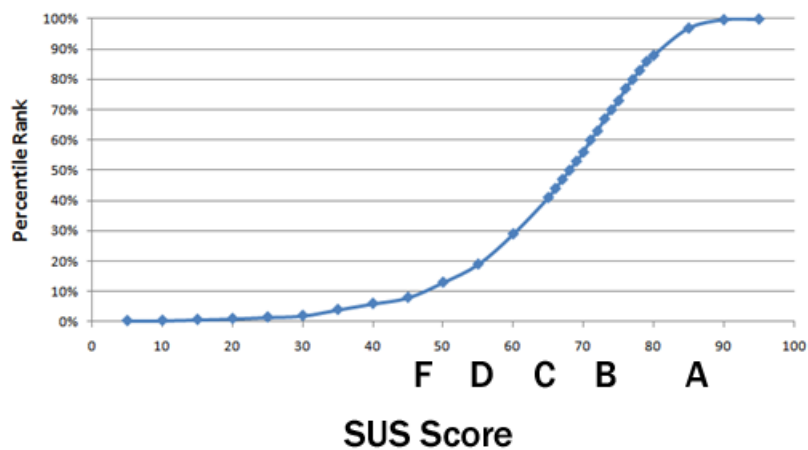


Figura 2.10: Rankings de Percentiles de SUS
Fuente: Sauro (2011)

Nielsen (2000) demostró en una investigación realizada, que a partir de cinco (5) entrevistas a usuarios ya es un número suficiente para hacer un estudio cualitativo sobre la usabilidad que tenga validez.

Se puede visualizar en la figura 2.11 la gráfica creada por Nielsen (2000) donde se muestra que al realizar pruebas de usabilidad con por lo menos cinco (5) usuarios, se obtendrá suficiente información para mejorar la solución de diseño, aproximadamente en 80% de los problemas de usabilidad presentes en el producto.

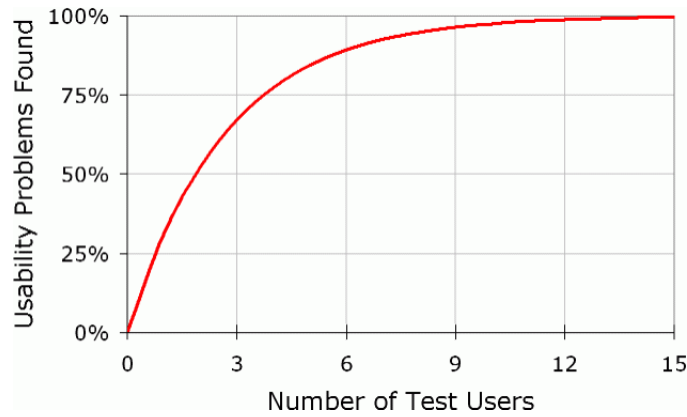


Figura 2.11: Número de Usuarios requeridos para realizar un test
Fuente: Nielsen (2000)

2.6. Distribución Normal

En estadística y probabilidad se llama distribución normal, distribución de Gauss o distribución gaussiana, a una de las distribuciones de probabilidad de variable continua que con más frecuencia aparece aproximada en fenómenos reales. Esta distribución permite modelar una gran cantidad de fenómenos naturales, sociales, demográficos y psicológicos entre otros.

La distribución Normal representa el modelo continuo más importante en Estadística, tanto por su aplicación directa, veremos que muchas variables de interés general pueden describirse por dicho modelo, como por sus propiedades, que han permitido el desarrollo de numerosas técnicas de inferencia estadística. En realidad, el nombre de Normal proviene del hecho de que durante un tiempo se creyó, por parte de médicos y biólogos, que todas las variables naturales de interés seguían este modelo.

Expresa Tauber (2001) que una variable aleatoria normal es aquella cuya función de densidad que viene dada por una fórmula, la cual representa una familia de funciones caracterizada por unos parámetros dados (media y desviación típica), explicando intuitivamente el significado de estos parámetros.

En la ecuación 2.1 expresa la curva de la función Normal.

$$f(x) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}; \quad N(\mu, \sigma) \quad (2.1)$$

Señala Tauber (2001) que prácticamente todos los casos de la distribución normal se encuentran entre tres desviaciones típicas por debajo de la media y tres desviaciones típicas por encima de la media (un total de seis desviaciones típicas).

La figura 2.12 muestra la curva normal, en la cual el 68,26% de las observaciones caen en el rango comprendido entre -1σ y $+1\sigma$, el 95,44% de las observaciones caen entre -2σ y $+2\sigma$ y el 99,74% de las observaciones caen entre -3σ y $+3\sigma$.

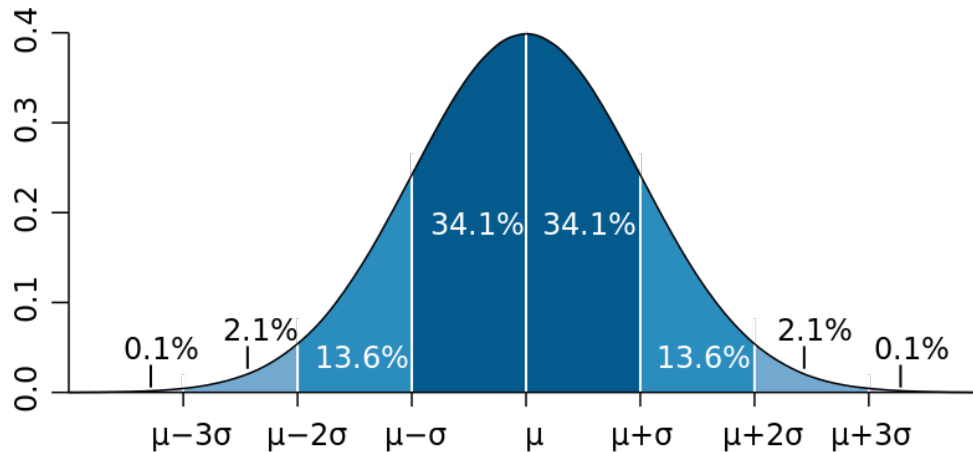


Figura 2.12: Distribución Normal Tipificada
Fuente: <https://commons.wikimedia.org/wiki/>

Las principales características de la curva Normal son las siguientes:

- El campo de existencia es cualquier valor real, es decir, $(-\infty, +\infty)$.
- Es simétrica respecto a la media μ .
- Tiene un máximo en la media μ .
- Crece hasta la media μ y decrece a partir de ella.
- La media, la mediana y la moda coinciden en el mismo punto (el centro)
- En los puntos $\mu - \sigma$ y $\mu + \sigma$ presenta puntos de inflexión.
- El eje de abscisas es una asíntota de la curva.
- El área del recinto determinado por la función y el eje de abscisas es igual a la unidad (1).
- Es simétrica respecto al eje que pasa por $x = \mu$, deja un área igual a 0,5 a la izquierda y otra igual a 0,5 a la derecha.
- Es unimodal.

2.7. Distribución Normal Estandarizada

La Distribución Normal Estándar o tipificada, es una técnica muy útil a la hora de comparar datos de diferentes distribuciones que estén en unidades de medida o escalas distintas, la misma se basa en un caso particular de la distribución Normal.

Mediante la tipificación, las medidas obtenidas por cualquier sujeto en variables o escalas diferentes, siempre tendrán una media o promedio aritmético de cero (0), una varianza y una desviación estándar de uno (1), se denota como $X \sim N(0,1)$ (Marín, 2005; Castro, 2012; Hernández Sampieri et al., 2014).

Al igual que toda distribución Normal, en la distribución Normal tipificada la Media, Moda y Mediana presentan el mismo valor, en este caso será uno (1).

La ecuación 2.2 muestra la función de densidad de una curva Normal tipificada:

$$f(x) = \frac{1}{\sqrt{2\Pi}} e^{-\frac{x^2}{2}}; \quad N(0,1) \quad (2.2)$$

2.8. Puntaje Z

Una variable aleatoria normal está estandarizada al expresar su valor como el número de desviaciones estándar (σ) que se encuentran a la izquierda o derecha de su media (μ). Éste es realmente sólo un cambio en las unidades de medida que se usan. La variable aleatoria normal estandarizada, mide la distancia entre una observación y la media, medidas en unidades de desviación estándar, es una medida de posición relativa.

Z mide el verdadero nivel dentro de cada conjunto de datos de las diferentes variables a analizar, al haber eliminado los parámetros de centro y dispersión. Es como si dos conjuntos los redujéramos a la misma escala para poder compararlos Marín (2005); Mendenhall et al. (2010); Castro (2012); Hernández Sampieri et al. (2014).

El puntaje Z se determina a partir de la siguiente ecuación:

- Para datos de una muestra,

$$Z = \frac{x_i - \bar{X}}{S} \sim N(0,1) \quad (2.3)$$

- Para datos poblaciones,

$$Z = \frac{x_i - \mu}{\sigma} \sim N(0,1) \quad (2.4)$$

en donde:

2.9. COEFICIENTE ALFA DE CRONBACH (α)

Z = dato estandarizado o normalizado

x_i = valor nominal del dato a estandarizar

\bar{X} = media aritmética del conjunto de datos o muestra

μ = media aritmética de la población

S = desviación estándar del conjunto de datos o muestra

σ = desviación estándar de la población

Según define Marín (2005), la tipificación de variables resulta muy útil para eliminar su dependencia respecto a las unidades de medidas empleadas. En realidad, una tipificación equivale a una transformación lineal.

Siendo:

$$Z = \frac{x_i - \mu}{\sigma} = \frac{1}{\sigma}x_i - \frac{\mu}{\sigma}$$

$$Z = ax_i + b \quad (2.5)$$

Donde:

$$a = \frac{1}{\sigma}; \quad b = -\frac{\mu}{\sigma}$$

Como ya se explicó anteriormente, Marín (2005) indica que “la variable tipificada expresa el número de desviaciones típicas que dista de la media cada observación. Por ello, se puede comparar la posición relativa de los datos de diferentes distribuciones. Otra situación habitual se presenta cuando se hace un cambio de unidades de medida”.

2.9. Coeficiente Alfa de Cronbach (α)

El Alfa de Cronbach (α) es un coeficiente que sirve para medir la fiabilidad de una escala de medida, para escalas politómicas (Cronbach, 1951).

El alfa de Cronbach permite estimar la fiabilidad de un instrumento de medida a través del conjunto de ítems, que se espera que midan el mismo constructo o dimensión teórica. La fiabilidad evalúa el grado de consistencia en que un instrumento mide lo que debe medir.

$$\alpha = \left(\frac{k}{k-1} \right) \left(1 - \frac{\sum S_i^2}{S_{total}^2} \right) \quad (2.6)$$

Donde k es el número de ítems de la prueba, S_i^2 es la varianza de los ítems (desde 1...i) y S_{total}^2 es la varianza de la prueba total.

2.9. COEFICIENTE ALFA DE CRONBACH (α)

Ledesma et al. (2002) indican que el método de consistencia interna es el camino más habitual para estimar la fiabilidad de pruebas, escalas o test, cuando se utilizan conjuntos de ítems o reactivos que se espera midan el mismo atributo o campo de contenido. Los principales coeficientes de estimación son sencillos de computar y están disponibles como opción de análisis en los programas estadísticos más conocidos. El coeficiente Alfa de Cronbach es, sin duda, el más ampliamente utilizado por los investigadores.

Los valores del alfa de Cronbach varían entre 0 y 1. Gliem and Gliem (2003) sugieren las valoraciones que se visualizan en la Tabla 2.3 al evaluar los coeficientes del alfa de Cronbach.

Tabla 2.3: Rango de Alfa de Cronbach

Alfa de Cronbach	>0,9	[0,8-0,9)	[0,7-0,8)	[0,6-0,7)	[0,5-0,6)	<0,5
Valoración	Excelente	Bueno	Aceptable	Cuestionable	Pobre	Inaceptable

Adaptado de: Gliem and Gliem (2003)

Capítulo 3

Marco Metodológico

En el presente capítulo se describe el marco metodológico donde se definió el tipo y diseño de la investigación, así como la población y el tipo de instrumento de recolección de datos para recabar información sobre las variables de estudio. El desarrollo de esta investigación se realiza de forma secuencial, siguiendo una serie de pasos aplicados de manera metodológica.

El enfoque que se adoptó en esta investigación fue la de una combinación de investigación documental y de campo.

Arias (2012) define la investigación o el diseño documental como un proceso que se basa en la búsqueda, recuperación, análisis, crítica e interpretación de datos obtenidos y registrados por otros investigadores en fuentes impresas, audiovisuales o electrónicas. Como en toda investigación, el propósito de este diseño es el aporte de nuevos conocimientos. De la misma manera Arias (2012) define la investigación de campo como aquella que se basa en una recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes. De allí su carácter de investigación no experimental. Del mismo modo el proyecto se puede enfocar como un proyecto de desarrollo tecnológico.

3.1. Etapas de la Investigación

La investigación se realizó en las siguientes etapas o fases:

3.1.1. Revisión Bibliográfica

En esta etapa se realizó la búsqueda y recopilación de la información documental relacionada con el área de estudio: Protocolo OpenFlow, SDN, SDWN, calidad del producto software, la norma ISO/IEC 25000 y del modelo EPCU entre otros. Con el propósito de recaudar la mayor información posible sobre la temática mencionada, esto permitió la fácil y clara comprensión del problema y por consiguiente el logro de una óptima solución.

3.1.2. Diagnóstico de la situación

En esta fase se procedió a determinar las distintas herramientas y requisitos necesarios para la instalación y configuración de OpenNet y Mininet-WiFi. Simultáneamente se determinó cuáles serán las aplicaciones que serán utilizadas para las pruebas de rendimiento.

3.1.3. Instalación y Configuración de los Servidores

Se procedió a instalar y configurar dos servidores bajo una plataforma GNU/Linux, esto con la finalidad de alojar las aplicaciones SDWN estudiadas, al igual que otros servicios o aplicaciones necesarias para la realización de pruebas de rendimiento y de usabilidad de OpenNet y Mininet-WiFi.

3.1.4. Realizar pruebas pertinentes

Esta etapa consistió en trabajar con algunas de las topologías inalámbricas ya recreadas en cada programa para realizar pruebas de rendimiento y movilidad. Se realizaron ciertas pruebas técnicas que proporcionaron información objetiva e independiente sobre cómo se comportan OpenNet y Mininet-WiFi en diferentes escenarios.

3.1.5. Diseño e implementación de instrumentos de recolección de datos

En esta etapa se planeó y programó la evaluación de calidad del producto software de OpenNet y de Mininet-WiFi. Se determinó y diseño un instrumento de medición a través de un cuestionario cerrado de selección simple (ver Apéndice C), para determinar la calidad del producto software empleando el modelo EPCU, la misma se le aplico a usuarios avanzados en el uso de OpenNet y Mininet-WiFi. Una vez definidos y relacionados los criterios para medir cada característica de la calidad interna/externa y de uso del producto software de las aplicaciones estudiadas, se ejecutó el proceso de recolección de datos, computó de las métricas e indicadores elementales, y por último, se procedió a documentar los resultados encontrados.

De igual forma en esta etapa se determinó el instrumento de validación de experiencia de usuario de los programas analizados, para ello se decantan por la escala para la usabilidad de los sistemas (SUS) (ver Apéndice D). La escala para la usabilidad de los sistemas, es un cuestionario de diez ítems que se aplica a los usuarios con la finalidad de medir la facilidad percibida en el uso de software, hardware, juegos y sitios web al interactuar con los mismos.

3.1.6. Medición y Evaluación de la Calidad del Producto Software y de la Experiencia de Usuario

Se seleccionó el modelo de estimación de proyectos en entornos de incertidumbre (EPCU) como el proceso de evaluación para determinar la calidad del producto software basado en la norma ISO/IEC 25000, en virtud de que ya se contaba con un estudio (referencia) que se utilizó para realizar la evaluación de calidad de un software.

Se utilizaron los mismos contextos definidos en el estudio, tanto para calidad interna/externa y de uso, los contextos definidos por Valdés (2013) fueron utilizados a través del sitio www.mepe.com.mx.

El modelo EPCU permitió obtener criterios de evaluación para cada característica y subcaracterística de forma cuantificable. Una vez definidos y consensuados los criterios, se llevó a cabo el proceso de evaluación de la calidad del producto software con el fin de analizar, evaluar, comparar la calidad interna/externa y de uso en OpenNet y en Mininet-WiFi.

Se realizaron actividades de análisis y comparación de los indicadores de la calidad del producto software y la experiencia de usuario de OpenNet y Mininet-WiFi, y la justificación de los resultados.

Se realizó la interpretación de los resultados obtenidos en las actividades experimentales. Se utilizaron software de código abierto y mecanismos de documentación para facilitar la interpretación de los datos y su seguimiento, tal es el caso de GNU PSPP, GeoGebra, Epidat, RLPlot y la hoja de cálculo Calc integrada en suite ofimática LibreOffice, esto permitió dar soporte al marco de medición y evaluación.

3.1.7. Elaboración del Informe Final

Esta etapa comprende la última fase de la investigación, donde se redacta y elabora el informe final estableciendo las conclusiones y recomendaciones necesarias una vez desarrollados los análisis cuantitativos y cualitativos que describen las características de la calidad del producto software de las aplicaciones analizadas.

3.2. Población y Muestra

Según Arias (2012) la población, es un conjunto finito o infinito de elementos con características comunes de interés para el investigador. Para esta investigación se estableció como criterio de selección de la muestra el nivel avanzado de los usuarios en el uso de OpenNet y/o Mininet-WiFi. Por lo anteriormente mencionado se realizó un muestreo intencional basado en el dominio avanzado por parte de los usuarios de los emuladores SDWN analizados, este muestreo lo define Arias (2012) como aquel donde los elementos son escogidos con base a criterios o juicios preestablecidos por el investigador.

En el caso de Mininet-WiFi el desarrollador del programa creó un grupo de discusión cerrado el cual permitió tener una interacción con diferentes profesionales expertos en el área que están trabajando con este software, ese grupo está compuesto por Profesores universitarios con un nivel de estudio Ph.D. o candidatos a algún doctorado del área, así como

3.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

estudiantes de algún Máster universitario del área, de los cuales solo diez (10) respondieron el instrumento de evaluación, el mismo fue creado para recabar información sobre la calidad del producto software fundamentado en la ISO/IEC 25000. En el caso de OpenNet existe un grupo de discusión compuesto por los desarrolladores donde el nivel de estudio es semejante a los del grupo mencionado anteriormente, respondiendo el instrumento de evaluación cinco (5) usuarios avanzados en el manejo del software. Esto permitió obtener información de personas calificadas en el uso del software, y así se pudo establecer acciones específicas, que promovieron un aporte relevante al análisis de la calidad del producto software de las aplicaciones estudiadas.

En el caso de la evaluación de la experiencia de usuario se decidió tomar una muestra de siete (7) usuarios compuesto por personal que labora en el área de las tecnologías de la información del rectorado de la Universidad de Los Andes en Mérida-Venezuela, recopilando información necesaria que sirvió de inicio para el análisis de la experiencia de usuario del software analizado. Este personal tuvo contacto por primera vez con el software al momento de las pruebas. Este número de usuarios que evaluó la UX de Mininet-WiFi y de OpenNet cumplió con los requisitos mínimos para esta evaluación según detalla Nielsen (2000), y se muestra con mejor detalle en la figura 2.11.

3.3. Técnicas e Instrumentos de Recolección de Datos

Para la recopilación de los datos se elaboró un instrumento de evaluación de la calidad interna/externa y de uso de OpenNet y Mininet-WiFi. Se utilizó un cuestionario en línea como instrumento que permitió recabar información por parte de los expertos en el uso del software analizado, respondiendo el experto de acuerdo a su criterio y experiencia.

El cuestionario se diseñó con la intención de obtener las entradas requeridas por el modelo EPCU para la determinación cuantitativa de la calidad del producto software, el mismo se presentó con una serie de preguntas cerradas. Se consideraron valores cualitativos para analizar las variables, el mismo brindó tres opciones cualitativas (Baja, Promedio, Alta), las cuales fueron estimadas numéricamente mediante valores comprendidos de 0 a 5, pudiendo los valores contener un decimal.

De acuerdo con su forma, el instrumento estaba compuesto de 82 ítems o preguntas, dividido en 10 secciones, las preguntas se basaron en los atributos, características y subcaracterísticas de la calidad interna/externa y de uso del producto software definidas en la norma ISO/IEC 25000. Este instrumento se puede analizar en el Apéndice C.

La confiabilidad o fiabilidad del instrumento de evaluación se determinó por medio del método de consistencia interna basado en el Alfa de Cronbach, el cual midió el grado de correlación interna entre los ítems.

3.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Hernández Sampieri et al. (2014) señala que la confiabilidad de un instrumento de medición se refiere al grado en que su aplicación repetida al mismo individuo u objeto produce resultados iguales.

La validez del contenido del instrumento utilizado igualmente se evaluó a través del juicio de expertos conocedores del área a investigar. El juicio de expertos lo define Escobar-Pérez and Cuervo-Martínez (2008) *”como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones”*.

El instrumento empleado en esta investigación para evaluar la calidad del producto software de las aplicaciones analizadas, fue revisado por un Profesor investigador de la Universidad Nacional Autónoma de México Ph.D., creador del modelo EPCU, especialista en dimensionamiento y estimación de proyectos de software, así como evaluación en factibilidad y desempeño de proyectos; y por un Profesor investigador de la Universidad de Los Andes Ph.D., especialista en comunicaciones industriales y Coordinador General del Grupo de Investigación en Ingeniería Empresarial y Gestión Automatizada. La validación de este instrumento el cual fue construido para una población de estudio específica, fue adaptado mediante un procedimiento de traducción (equivalencia semántica), siendo el mismo validado en el idioma a presentar a la población de estudio, en este caso el Inglés.

Para determinar la experiencia de usuario se utilizó un cuestionario estándar el cual se encuentra validado científicamente y ha sido diseñado para tal fin conocido como: Escala para la Usabilidad de los Sistemas (SUS).

Esta escala que fue desarrollada por Brooke (1996) y se implementó en este estudio debido a la necesidad de medir las opiniones subjetivas de los usuarios sobre el uso de los software SDWN para realizar una validación directa de UX, esto permitirá obtener un valor de usabilidad distinto para utilizarlo con el modelo EPCU.

Se debe dejar claro que el software analizado para simulación y/o emulación de SDWN es para usuarios específicos y/o con un nivel de conocimientos técnicos de medio a avanzado, en este caso el instrumento para determinar la UX de los software fue aplicado a un grupo distinto de profesionales que laboran en el área de las Tecnologías de la Información en la Universidad de Los Andes, cuyo nivel en el área es variado (bajo a avanzado), destacando que ninguno tenían experiencia ni conocimiento alguno sobre el uso y existencia de los dos software estudiados.

La escala SUS se aplicó después que el usuario tuvo la oportunidad de utilizar el software, la misma se evaluó inmediatamente antes que el usuario tuviese la oportunidad de pensar largamente o discutir sobre el uso del software. Se tomó como referencia el cuestionario efectuado por Cortés (2000). Este instrumento de evaluación de la experiencia de usuario para el software estudiado fue aplicado en línea.

3.4. Técnicas de Análisis de Datos

Luego de aplicar los instrumentos de validación sobre calidad del producto software al grupo de expertos, Estos datos fueron procesados con los software de fuentes abiertas GNU PSPP, GeoGebra, Epidat, RPlot y la hoja de cálculo Calc de la suite ofimática LibreOffice, para obtener los estadísticos descriptivos de la información recolectada, esto permitió obtener las entradas requeridas por el modelo EPCU para determinar la calidad interna/externa y de uso de Mininet-WiFi y de OpenNet. Al tratarse de estadísticos descriptivos, los datos que se muestran son medidas de tendencia central, de dispersión y medidas de distribución.

De igual manera se procedió a calcular los estadísticos descriptivos de la experiencia de usuario determinada. Posteriormente se normalizaron o tipificaron los valores generados en la puntuación SUS para obtener valores corregidos que permiten determinar una nueva usabilidad del producto.

Una vez obtenidos los valores corregidos y la desviación estándar del conjunto de datos SUS, se procede a calcular el puntaje Z que le correspondería en una curva normal a esa proporción de área de la curva original. De este modo, para un valor corregido en particular de la usabilidad, su puntaje Z asociado nos indica a cuántas unidades de desviación estándar del promedio se encuentra dicho valor. En otras palabras, en la curva normal contamos en cantidad de desviaciones estándar.

3.5. Análisis Cuantitativo de la calidad del producto software

Con la información recopilada y analizada, mediante el uso de estadísticos descriptivos se determinó la media aritmética de los datos la cual sirvió como valor de entrada de los mecanismos del modelo EPCU para determinar la calidad interna/externa y de uso de OpenNet y Mininet-WiFi.

Se procedió a emplear el mecanismo EVA20131206SQRINT desarrollado por Valdés (2014). Este mecanismo permite evaluar cuantitativamente y de forma sencilla la calidad Interna/Externa del producto software, el mismo se basa en las categorías definidas por el estándar ISO/IEC 25000 y está fundamentado en el modelo EPCU.

Así mismo se utilizó el mecanismo EVA20131206SQREXT el cual está igualmente fundamentado en el modelo EPCU, este mecanismo también fue creado por Valdés (2014), la finalidad del mismo es ofrecer un mecanismo que permita evaluar cuantitativamente y de forma sencilla la calidad de uso del producto software, basado en las categorías definidas por el estándar ISO/IEC 25000.

3.5. ANÁLISIS CUANTITATIVO DE LA CALIDAD DEL PRODUCTO SOFTWARE

Posteriormente se determinó el porcentaje de usabilidad de SUS del software analizado, como paso siguiente se procedió a normalizar los valores obtenidos de ese porcentaje para poder interrelacionarlos con la usabilidad obtenida del instrumento de evaluación.

Con el puntaje Z generado de la normalización y la desviación estándar de los datos obtenidos del análisis estadístico de los instrumentos de evaluaciones de los software OpenNet y Mininet-WiFi, al igual que la normalización de los valores obtenidos del porcentaje de usabilidad de SUS se determinó una media aritmética corregida utilizando la ecuación 2,3. Se empleó esta ecuación ya que los datos provienen de una muestra determinada.

Siendo esta media aritmética corregida un nuevo valor que se utilizó como entrada de la usabilidad para la determinación de un nuevo índice de calidad de uso en el modelo EPCU. Finalmente se analizó la diferencia de ambos valores de la calidad de uso a partir del modelo EPCU.

Capítulo 4

Presentación y Análisis de los Resultados

En el capítulo 3 se presentó la metodología empleada en este estudio. En este capítulo se exponen los resultados de la ejecución de la investigación y también el análisis de los resultados.

4.1. Diagnóstico de la situación

En esta fase se procedió a determinar las distintas herramientas y requisitos necesarios para la instalación y configuración de OpenNet y Mininet-WiFi. Del mismo modo se determinó cuáles serán las aplicaciones que serán utilizadas para las pruebas de rendimiento.

Acá se observó que ambos software tienen paquetes comunes pero de distintas versiones, esto pudiese generar conflictos al momento de la instalación de hacerlos en un solo equipo, por esto se decidió instalar cada software en equipos distintos.

Un detalle a tomar en consideración es el hecho que OpenNet y Mininet-WiFi solo están diseñados para funcionar bajo el Sistema Operativo GNU/Linux, específicamente para distribuciones Ubuntu en la versión 13.04 o posterior para Mininet-WiFi, y OpenNet se puede instalar en equipos que posean una distribución Fedora 20 en adelante y versiones de Ubuntu a partir de la versión 12.04.

Para la realización de las pruebas de rendimiento se decidió utilizar Iperf, la cual es una herramienta cliente-servidor de código abierto que sirve para medir el ancho de banda y la calidad de un enlace de red. El funcionamiento habitual de Iperf para medir el rendimiento de la red consiste en crear flujos de datos TCP para medir el rendimiento de la carga útil, así como UDP para permitir al usuario especificar el tamaño de los datagramas y proporcionar resultados del rendimiento y de los paquetes perdidos. Iperf permite ajustar varios parámetros que pueden ser usados al hacer las pruebas en una red, así como para optimizar y ajustar la red. Iperf puede ser asociado con Jperf para tener una interfaz gráfica manejable.

Para diagnosticar la conectividad, el funcionamiento y la disponibilidad de las redes emuladas, se determinó utilizar la herramienta PING (Packet Internet Groper). PING es una utilidad que sirve para diagnosticar en redes de computadoras el estado de la comunicación del host local con uno o varios equipos remotos de una red IP por medio del envío de paquetes ICMP de solicitud (ICMP Echo Request) y de respuesta (ICMP Echo Reply).

El Protocolo de Mensajes de Control de Internet o ICMP (por sus siglas en inglés de Internet Control Message Protocol) es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP), se puede afirmar que es la herramienta de administración de red más conocida. Es una herramienta simple que envía paquetes de un host destino a un host remoto para verificar si está respondiendo y, por ende, si es accesible a través de la red.

4.2. Instalación y Configuración de los servidores

Al hablar o determinar la calidad del producto software, hoy en día una de las características que cobra valor es la portabilidad, en especial la capacidad del producto software para ser instalado en un entorno específico. La mayoría de las características pueden que varíen entre diferentes productos software ya que los mismos son distintos, pero la instalación por ser el primer contacto que tiene el usuario con el producto software debe ser agradable.

Actualmente existen una gran cantidad de opciones que permiten o facilitan la instalación y configuración de un software. Al existir esa gran variedad de opciones el usuario no se complica a la hora de elegir un software donde su instalación sea complicada. Es por ello que se hace determinante para los desarrolladores el crear un mecanismo de instalación accesible al usuario final.

Esta fase se ocupó de realizar la instalación y configuración del sistema operativo GNU/Linux en los equipos portátiles que cumplieron la función de servidores. Se decidió la instalación y configuración de dos equipos distintos para que no se crearan conflictos con diferentes versiones de paquetes que utilizan ambos programas, como es el caso de Open vSwitch (OVS) uno de los componentes esenciales para la simulación, emulación y virtualización de dispositivos que utilizan el protocolo OpenFlow. Se debe destacar que se pudo haber usado un solo equipo con la intención de optimizar los recursos e instalar uno de los programas en el servidor físico, y virtualizar un servidor para instalar el segundo de los programas analizados, o en su defecto virtualizar ambos servidores en un solo computador, pero al querer medir el rendimiento de ambos programas y al ser esta investigación un análisis comparativo se decidió hacer la instalación física y no virtualizar, para que las comparaciones no tuviesen esa condición inicial, aprovechando la disponibilidad de estos computadores.

En el Apéndice B se muestra un compendio que se elaboró sobre cómo se realizó la instalación y configuración de OpenNet y Mininet-WiFi, así como los problemas que se presentaron en este proceso y sus soluciones.

La Tabla 4.1 muestra las características técnicas de los dos equipos portátiles que sirvieron como servidor.

Tabla 4.1: Características técnicas de los servidores

Servidor	Procesador	Frecuencia	Núcleos	RAM	Sistema Operativo
1	Core 2 Duo T6500; 32 bits	2.10 GHz	2	2 GB	Ubuntu 14.04.3 LTS
2	Core i5 i5-2410M; 64 bits	2.30 GHz	8	8 GB	Xubuntu 15.10

4.3. Pruebas de rendimiento y uso del software

Esta etapa consistió en la ejecución de pruebas de rendimiento y de uso de ambos programas, en los dos servidores, arrojando resultados de rendimiento semejantes en ambos casos. Así mismo se recrearon algunas topologías SDWN en cada programa, con la intención de conocer mejor el mismo y para realizar pruebas de rendimiento y movilidad, las pruebas respectivas posibilitaron verificar el funcionamiento de los software y que los mismos si estén diseñados para emular y/o simular SDWN. La realización de estas pruebas de rendimiento permitió medir el ancho de banda y la tasa de transferencia, en cada enlace y el punto de conexión a la red del usuario, las pruebas a realizar serán las siguientes:

Iperf la cual es una herramienta que se utiliza para evaluar el desempeño en las comunicaciones de redes informáticas, sirvió para medir el ancho de banda efectivo de una red de datos simulada, además permitió la evaluación de las comunicaciones entre dos dispositivos de la red usando los protocolos TCP o UDP.

Otra prueba realizada para medir la conectividad entre dispositivos fue usar el comando PING (Packet Internet Groper), el mismo es una utilidad para el diagnóstico de la conexión entre dos dispositivos en una red TCP/IP mediante el envío de paquetes ICMP (Internet Control Message Protocol) de solicitud y respuesta. También especifica la pérdida de paquetes. La ejecución del comando PING mostró la siguiente información:

- El nombre del “host” destino.
- El número de paquetes transmitidos.
- El número de paquetes recibidos.
- La pérdida porcentual de paquetes.
- El promedio, el valor máximo y el valor mínimo del RTT.

Las pruebas de uso en esta etapa consistieron en el conocimiento de los programas analizados en lo concerniente al manejo y su configuración.

4.3.1. Pruebas con OpenNet

Se procedió a realizar los siguientes pasos para ejecutar OpenNet con la finalidad de conocer su uso y funcionamiento:

- Antes de comenzar a utilizar OpenNet, fue necesario cargar un controlador POX para que la red se pueda interconectar y así funcione el reenvío de paquetes, ya que OpenNet no posee un controlador precargado, se decidió usar el controlador POX, ejecutando el

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

comando que se muestra en Figura 4.1. POX es un controlador SDN, desarrollado en Python que permite programar y controlar switches OpenFlow de forma sencilla. POX es un controlador muy atrayente a nivel académico y de simulación por la facilidad de modificar su código.

```
gerlyn@gerlyn-mininet:~/OpenNet-master/pox$ sudo python pox.py forwarding.l2_learning
sudo: imposible resolver el anfitrión gerlyn-mininet
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
INFO:core:POX 0.2.0 (carp) is up.
INFO:openflow.of_01:[00-00-00-00-00-07 1] connected
INFO:openflow.of_01:[00-00-00-00-00-06 2] connected
INFO:openflow.of_01:[00-00-00-00-00-01 4] connected
INFO:openflow.of_01:[00-00-00-00-00-05 6] connected
INFO:openflow.of_01:[00-00-00-00-00-03 5] connected
INFO:openflow.of_01:[00-00-00-00-00-04 3] connected
INFO:openflow.of_01:[00-00-00-00-00-02 7] connected
█
```

Figura 4.1: Ejecución del Controlador POX

- Se ejecutan los pasos ya mostrados en la figura B.4 del Apéndice B, permitiendo que se cargue OpenNet. La figura 4.2 muestra la consola de OpenNet luego del proceso de inicio del mismo al ejecutar el script wifiroaming.py, se puede ver el arranque de la red además de la configuración de los host, el arranque del controlador y de los 7 switches de la topología seleccionada. El script wifiroaming.py para esta topología se puede ver en el Apéndice D.

```
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
Wifi pcap path: /tmp/pcap/wifi
CSMA pcap path: /tmp/pcap/csma
Netanim path: /tmp/xml/wifi-wired-bridged4.xml
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
Testing network connectivity
*** h1 : ('ping 10.10.10.2 -c 3',)
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
From 10.10.10.1 icmp_seq=1 Destination Host Unreachable
From 10.10.10.1 icmp_seq=2 Destination Host Unreachable
From 10.10.10.1 icmp_seq=3 Destination Host Unreachable

--- 10.10.10.2 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2017ms
pipe 3
*** Starting CLI:
opennet> █
```

Figura 4.2: Consola de OpenNet al arrancar

Ya cargado OpenNet se comienzan a conocer algunos de los comandos que este programa contiene, en la figura 4.3 se ve un conjunto de opciones que se pueden ejecutar desde la consola de OpenNet, en este caso se ejecutó el comando help.

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

```
opennet> help

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

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

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

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

opennet> █
```

Figura 4.3: Ayuda de OpenNet.

La figura 4.4 detalla por medio de los comandos nodes y dump los dispositivos existentes y otros parámetros importantes como la dirección IP, el proceso que está ejecutándose en la red emulada mostrando la información de cada conexión.

```
opennet> nodes
available nodes are:
c0 h1 h2 s1 s2 s3 s4 s5 s6 s7
opennet> dump
<Host h1: h1-eth0:10.10.10.1 pid=28571>
<Host h2: h2-eth0:10.10.10.2 pid=28574>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None,
s1-eth5:None,s1-eth6:None,s1-eth7:None pid=28579>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None pid=28582>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=28585>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=28588>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None pid=28591>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None pid=28594>
<OVSSwitch s7: lo:127.0.0.1,s7-eth1:None,s7-eth2:None pid=28597>
<RemoteController c0: 127.0.0.1:6633 pid=28565>
opennet> █
```

Figura 4.4: Información de los dispositivos existentes en la topología

Dentro de las pruebas de rendimiento a realizar se evaluó la interconexión entre host utilizando el comando ping para ello, obteniendo una respuesta positiva al envío y recepción de paquetes ICMP como se muestra en la figura 4.5, en la misma se aprecian los paquetes enviados y los recibidos al ejecutar dicha acción, al igual que el tiempo que tarda cada paquete en llegar al host destino, y otros parámetros o estadísticas de interconexión como el ttl y el rtt.

```

opennet> h1 ping h2 -c 7
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=56.1 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=10.6 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=7.82 ms
64 bytes from 10.10.10.2: icmp_seq=4 ttl=64 time=5.95 ms
64 bytes from 10.10.10.2: icmp_seq=5 ttl=64 time=5.60 ms
64 bytes from 10.10.10.2: icmp_seq=6 ttl=64 time=6.20 ms
64 bytes from 10.10.10.2: icmp_seq=7 ttl=64 time=10.3 ms

--- 10.10.10.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6007ms
rtt min/avg/max/mdev = 5.600/14.668/56.137/17.036 ms
opennet> █

```

Figura 4.5: Probando conectividad mediante paquetes ICMP

Luego se procedió abrir desde OpenNet un emulador de terminal Xterm para uno de los host simulados, en la figura 4.6 se puede apreciar el terminal del host 1 (h1). En este terminal Xterm o consola del equipo h1, en el mismo se puede ejecutar cualquier comando u operación de las existentes en sistemas GNU/Linux.



Figura 4.6: Terminal Xterm del host 1 (h1)

En la figura 4.7 se aprecia la consola principal de OpenNet y el terminal del host 1 emulado, donde se ejecutaron comandos para ver los diferentes parámetros de configuración de la tarjeta de red (NIC) del equipo h1. Se puede verificar en la misma figura que los parámetros de configuración de la NIC pertenecen a una red cableada, por lo que el comportamiento de OpenNet en esta parte de una SDWN es de simulación la misma, mas no de emulación.

```

opennet> h1 ifconfig
h1-eth0 Link encap:Ethernet direcciónHW 5a:01:f8:47:77:8e
Direc. inet:10.10.10.1 Difus.:10.255.255.255 Másc:255.0.0.0
Dirección inet6: fe80::5801:f8ff:fe47:778e/64 Alcance:Enlace
ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
Paquetes RX:189 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:14 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colatX:500
Bytes RX:29944 (29.9 KB) TX bytes:1116 (1.1 KB)

lo Link encap:Bucle local
Direc. inet:127.0.0.1 Másc:255.0.0.0
Dirección inet6: ::1/128 Alcance:Anfitrión
ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colatX:0
Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)

opennet> █

```

```

root@gerlyn-mininet: /home/gerlyn/OpenNet-master/mininet/examples/
root@gerlyn-mininet:/home/gerlyn/OpenNet-master/mininet/examples/opennet# ifconfig
h1-eth0 Link encap:Ethernet direcciónHW 5a:01:f8:47:77:8e
Direc. inet:10.10.10.1 Difus.:10.255.255.255 Másc:255.0.0.0
Dirección inet6: fe80::5801:f8ff:fe47:778e/64 Alcance:Enlace
ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
Paquetes RX:189 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:14 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colatX:500
Bytes RX:29944 (29.9 KB) TX bytes:1116 (1.1 KB)

lo Link encap:Bucle local
Direc. inet:127.0.0.1 Másc:255.0.0.0
Dirección inet6: ::1/128 Alcance:Anfitrión
ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colatX:0
Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)

root@gerlyn-mininet:/home/gerlyn/OpenNet-master/mininet/examples/opennet# ifconfig
lo no wireless extensions.
h1-eth0 no wireless extensions.

root@gerlyn-mininet:/home/gerlyn/OpenNet-master/mininet/examples/opennet#

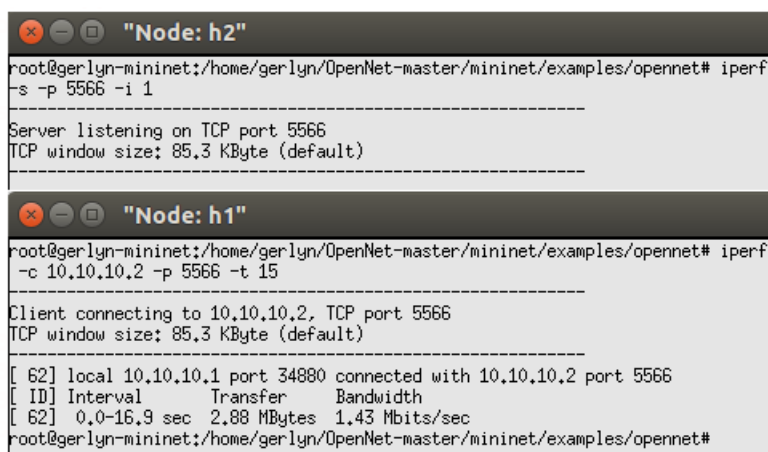
```

Figura 4.7: Parámetros de la NIC en una de las estaciones emuladas

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

Luego de realizarse las pruebas de conectividad se procedió a ejecutar la herramienta Iperf para ver el desempeño de las conexiones de la red emulada. Como se ilustra en la figura 4.8 donde se pueden apreciar los dos terminales xterm de dos host a los que se le aplicó la herramienta Iperf. Esta herramienta tiene muchas opciones a ejecutar.

Las pruebas con Iperf consisten en enviar desde el servidor, un flujo de datos a una tasa de transmisión constante por el puerto indicado en la regla evaluada hasta un host remoto, el cliente captura los datagramas en ventanas de 1 segundo, y a medida de que recibe el flujo de datos publica los parámetros de rendimiento. Los parámetros que se muestran a continuación son los más elementales, del lado del cliente en Node: *h1* la sentencia para ejecutarlo es la siguiente: *iperf -c 10.10.10.2 -p 5566 -t 15*, donde la opción *-c* significa cliente, luego *-p* se refiere a el número de puerto TCP, el parámetro *-t* indica el tiempo que duró la transmisión, en este caso se le coloco 15 segundos. En el lado del servidor en el host (*h2*), los parámetros son los mismos agregando la *-s* que significa servidor y la opción *-i* que especifica el intervalo medido en segundos en el cual se volverá a realizar la medición.



```
root@gerlyn-mininet:~/home/gerlyn/OpenNet-master/mininet/examples/opennet# iperf
-s -p 5566 -i 1
-----
Server listening on TCP port 5566
TCP window size: 85.3 KByte (default)
-----

root@gerlyn-mininet:~/home/gerlyn/OpenNet-master/mininet/examples/opennet# iperf
-c 10.10.10.2 -p 5566 -t 15
-----
Client connecting to 10.10.10.2, TCP port 5566
TCP window size: 85.3 KByte (default)
-----
[ 62] local 10.10.10.1 port 34880 connected with 10.10.10.2 port 5566
[ ID] Interval      Transfer    Bandwidth
[ 62] 0.0-16.9 sec  2.88 MBytes  1.43 Mbits/sec
root@gerlyn-mininet:~/home/gerlyn/OpenNet-master/mininet/examples/opennet#
```

Figura 4.8: Iperf cliente-servidor en consolas de Xterm en los host h1-h2

Luego de haber ejecutado la prueba de rendimiento con iperf entre los nodos emulados *h1* y *h2*, se procedió a graficar con Gnuplot el Throughput del flujo TCP en un tiempo dado.

Gnuplot es un programa con licencia bajo software libre, que sirve para la visualización gráfica de datos científicos. Permite realizar gráficos 2D y 3D de curvas, líneas de nivel y superficies, tanto a partir de funciones como de datos discretos.

Para la construcción de la gráfica Gnuplot fue ejecutado directamente desde un terminal Xterm del nodo *h2*, esto demuestra que desde un host emulado se puede ejecutar cualquier proceso existente en GNU/Linux. En la figura 4.9 se puede apreciar la gráfica, los valores de la misma se muestran en la tabla incrustada a la derecha del mismo gráfico.

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

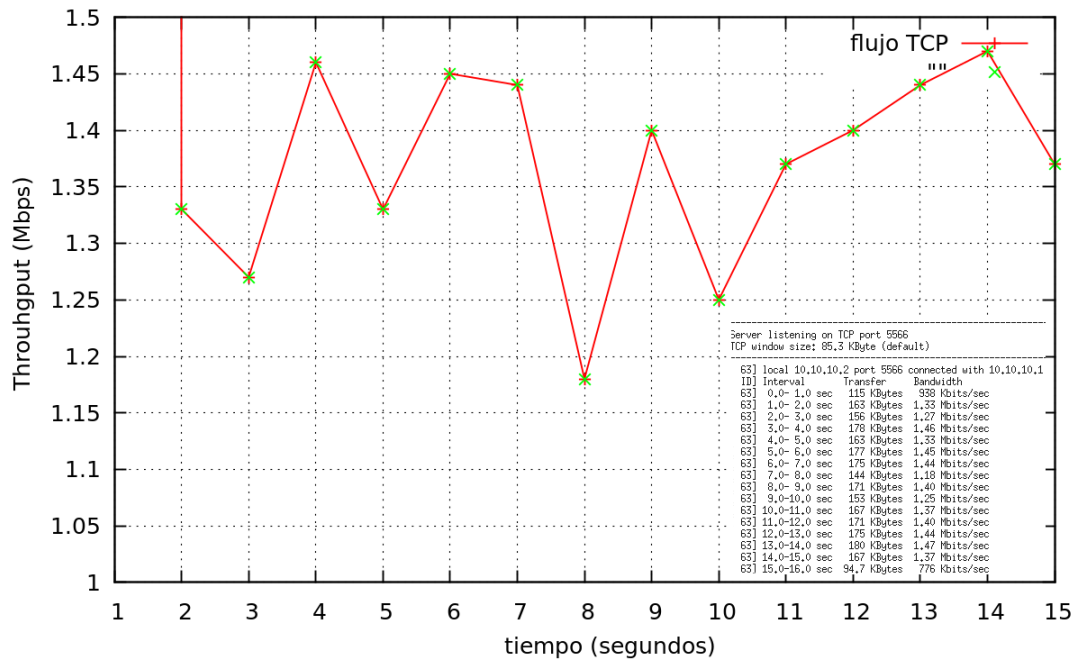


Figura 4.9: Gráfica del Throughput resultado de la prueba de rendimiento

Después de haber realizado las pruebas de rendimiento se procedió a ejecutar Net/Anim, de la misma forma como se muestra en la figura B.7 del Apéndice B. Luego de haber cargado Net/Anim y de haber ejecutado OpenNet se procedió cargar el archivo XML para realizar la simulación inalámbrica de la topología escogida. A continuación, se volvió a ejecutar el script wifroaming.py y NetAnim se visualizó como se ve en la figura 4.10.

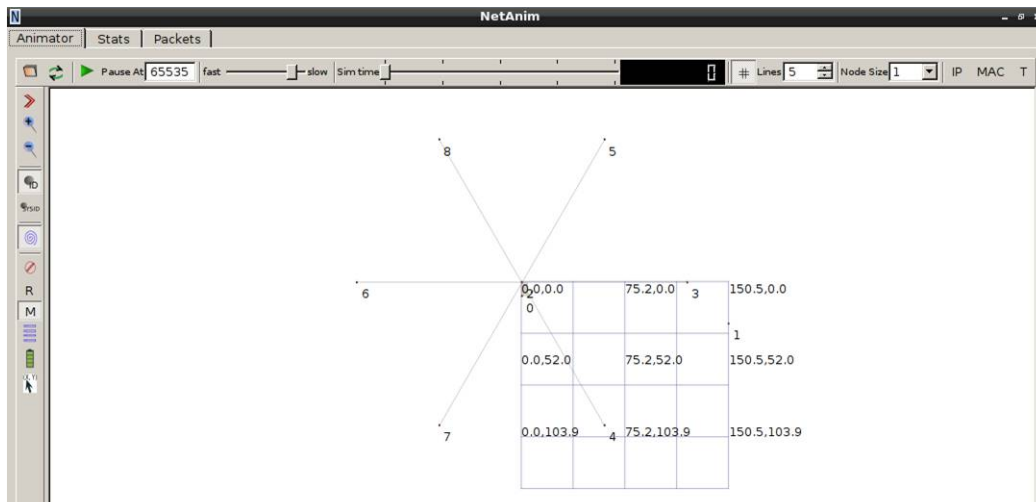


Figura 4.10: NetAnim ejecutando la topología seleccionada

Seguidamente se comenzó a ejecutar la simulación con NetAnim, pudiendo evaluar en este paso la movilidad de los host, en un rango o cuadrícula configurada por el usuario, como se muestra en la figura 4.11.

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

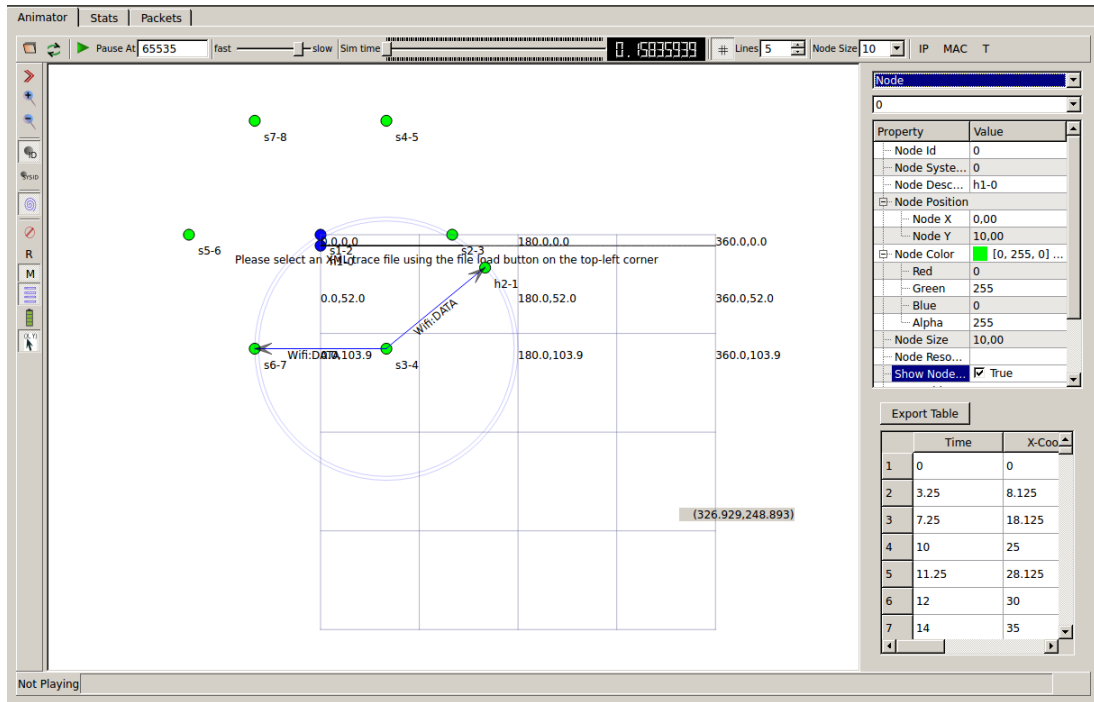


Figura 4.11: NetAnim realizando la simulación

En la figura 4.12 se puede apreciar como se ve NetAnim luego de modificar algunas variables de los dispositivos de la red simulada, como la posición y el tamaño de los nodos. NetAnim permite visualizar varios parámetros como los acuses de recibos (ACK) de la recepción de los mensajes, posición y trayectoria de los nodos, paquetes ICMP, tiempos de transmisión entre otros.

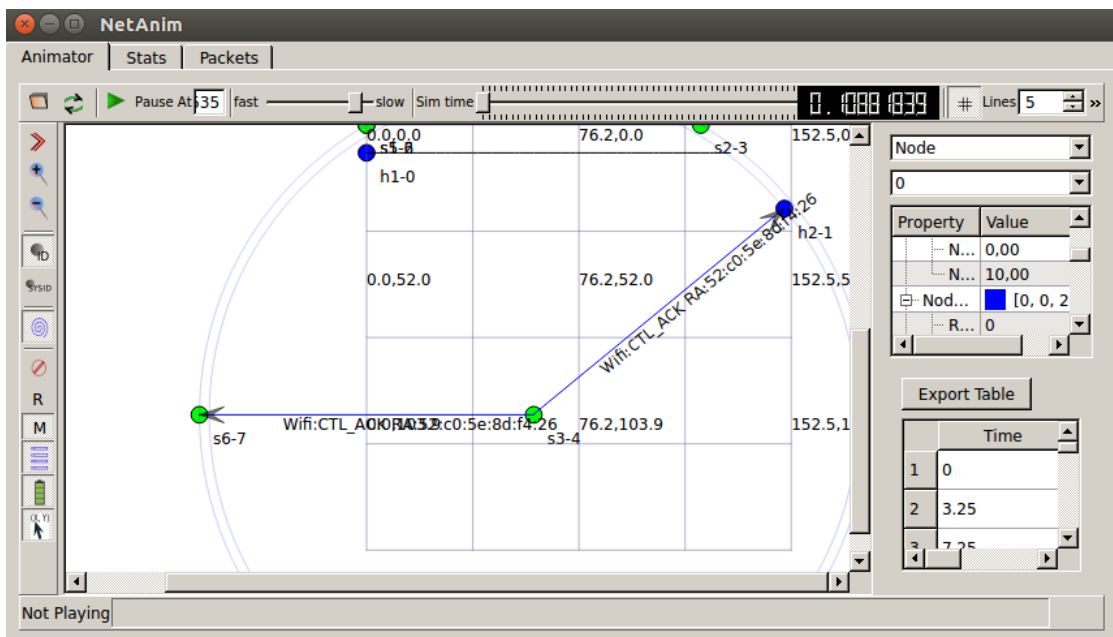


Figura 4.12: Visualización de algunos parámetros en NetAnim

4.3.2. Pruebas con Mininet-WiFi

Se recrearon algunas topologías SDWN usando VND e integrándolo con Mininet-WiFi. Al igual que con el software anterior se realizaron pruebas de rendimiento las cuales permitieron medir el ancho de banda y la tasa de transferencia, así como pruebas de conectividad utilizando paquetes ICMP, movilidad y de uso.

VND permitió crear algunos escenarios virtuales de diferentes topologías, los cuales se emularon como una SDWN agregándole los dispositivos necesarios para que cumpla con ello, luego se pudo exportar esta topología en códigos de Python a Mininet-WiFi, donde se realizaron diferentes pruebas de conectividad entre los dispositivos, así mismo se procedió a medir el ancho de banda y el rendimiento de los enlaces. VND permitió la creación de redes virtuales simulando estaciones y puntos de acceso inalámbricos (APs que soportan OpenFlow), al igual que enlaces, switches y controladores. Se utilizó un controlador POX de forma centralizada.

La Figura 4.13 muestra un escenario virtual sencillo creado en VND, donde se pueden observar los distintos dispositivos SDWN interconectados, como el punto de acceso inalámbrico (con soporte OpenFlow), el controlador OpenFlow y dos (2) computadores portátiles, los cuales están conectados inalámbricamente.

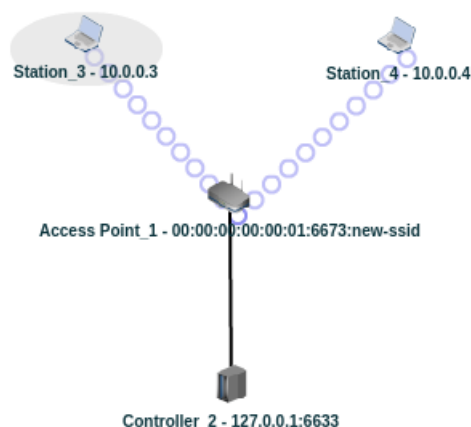


Figura 4.13: Escenario virtual sencillo de una SDWN usando VND

Los Algoritmos generados por VND para las topologías trabajadas con Mininet-WiFi se pueden ver en el Apéndice G.

Con VND se crearon topologías de forma gráfica y sencilla a nivel de usuario, sin requerir que la configuración sea por comandos. En la figura 4.14 se observa otro escenario virtual creado en VND, en el cual se le han agregado otros dispositivos a la SDWN.

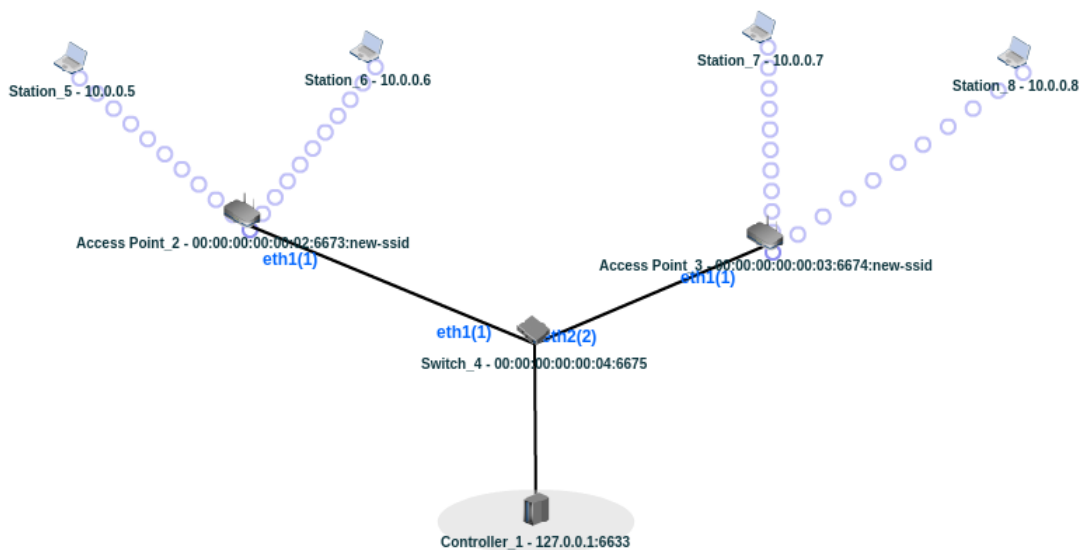


Figura 4.14: Red SDWN con VND

La emulación permitirá analizar la capacidad de Mininet-WiFi en la creación de redes SDWN. Se realizó la emulación de varios escenarios de redes de datos.

Mininet-WiFi permitió la creación de redes virtuales emulando hosts, enlaces, switches y controladores. Los códigos de python para ejecutarlos fueron exportados de las topologías recreadas en VND. La emulación permite usar los prototipos y códigos usados en redes reales, además de extrapolar los resultados logrados; lo que es una ventaja frente a una red simulada. Con Mininet-WiFi procederemos a realizar pruebas sencillas para determinar la conectividad de los enlaces así como su ancho de banda. La figura 4.15 muestra la consola principal de Mininet-WiFi al momento de iniciar su ejecución e interconectarse con los dispositivos de la topología.

```
gerlyn @ gerlyn-bears ~/mininet-wifi/mininet/examples
└─$ sudo python wifiMobilityModel.py
*** Creating nodes
*** Associating and Creating links
Associating stal to apl
Associating sta2 to apl
*** Starting network
Mobility started at 0 second(s)
*** Running CLI
*** Starting CLI:
mininet-wifi> █
```

Figura 4.15: Consola principal de Mininet-WiFi

Los documentación y los comandos de acceso a la ayuda, es un factor muy importante para la ejecución de un programa, en el caso de Mininet-WiFi existen tres opciones más que no tiene OpenNet, en la figura 4.16 se puede ver el listado de los mismos.

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

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

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

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

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

mininet-wifi> █
```

Figura 4.16: Opciones de Ayuda de Mininet-WiFi

En la figura 4.17 se observan los parámetros Wireless de una de las estaciones emuladas en Mininet-WiFi, acá se pudo apreciar que Mininet-WiFi si permite emular una SDWN comportándose cada nodo como un dispositivo inalámbrico de una SDWN física. Se pueden ver los valores de conexión y detalles del punto de acceso inalámbrico.

```
mininet-wifi> stal iwconfig
stal-wlan0 IEEE 802.11abgn ESSID:"new-ssid"
          Mode:Managed Frequency:2.412 GHz Access Point:
02:00:00:00:02:00
          Bit Rate:1 Mb/s Tx-Power=20 dBm
          Retry short limit:7 RTS thr:off Fragment thr
:off
          Encryption key:off
          Power Management:off
          Link Quality=70/70 Signal level=-30 dBm
          Rx invalid nwid:0 Rx invalid crypt:0 Rx invali
d frag:0
          Tx excessive retries:0 Invalid misc:0 Missed
beacon:0

lo      no wireless extensions.

mininet-wifi> stal iwconfig
stal-wlan0 IEEE 802.11abgn ESSID:"new-ssid"
          Mode:Managed Frequency:2.412 GHz Access Point: 02:00:00:00:02:00
          Bit Rate:1 Mb/s Tx-Power=20 dBm
          Retry short limit:7 RTS thr:off Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=70/70 Signal level=-30 dBm
          Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
          Tx excessive retries:0 Invalid misc:0 Missed beacon:0

lo      no wireless extensions.
```

Figura 4.17: Parámetros inalámbricos de la NIC en una de las estaciones emuladas

La figura 4.18 muestra los parámetros de las interfaces Wireless de uno de los nodos de la SDWN de la figura 4.14. Se visualiza a la izquierda los parámetros mencionados desde la consola propia de Mininet-WiFi y a la derecha directamente desde la consola Xterm de la estación inalámbrica 1 (stal), mostrando que los datos son los mismos en ambos casos.

```

mininet-wifi> stal ifconfig
lo          Link encap:Bucle local
           Direc. inet:127.0.0.1 Másc:255.0.0.0
           Dirección inet6: ::1/128 Alcance:Anfitrión
           ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
           Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
           Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0
           colisiones:0 long.colaTX:0
           Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)

stal-wlan0 Link encap:Ethernet direcciónHW 02:00:00:00:00:00
           Direc. inet:10.0.0.2 Difus.:10.255.255.255 Másc:255.0.0.0
           Dirección inet6: fe80::ff:fe00:0/64 Alcance:Enlace
           ACTIVO DIFUSIÓN MULTICAST MTU:1500 Métrica:1
           Paquetes RX:17 errores:0 perdidos:0 overruns:0 frame:0
           Paquetes TX:6 errores:0 perdidos:0 overruns:0 carrier:0
           colisiones:0 long.colaTX:1000
           Bytes RX:3390 (3.3 KB) TX bytes:628 (628.0 B)

root @ gerlyn-bears ~/mininet-wifi/examples
└─ # ifconfig -a
lo          Link encap:Bucle local
           Direc. inet:127.0.0.1 Másc:255.0.0.0
           Dirección inet6: ::1/128 Alcance:Anfitrión
           ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
           Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
           Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0
           colisiones:0 long.colaTX:0
           Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)

stal-wlan0 Link encap:Ethernet direcciónHW 02:00:00:00:00:00
           Direc. inet:10.0.0.2 Difus.:10.255.255.255 Másc:255.0.0.0
           Dirección inet6: fe80::ff:fe00:0/64 Alcance:Enlace
           ACTIVO DIFUSIÓN MULTICAST MTU:1500 Métrica:1
           Paquetes RX:17 errores:0 perdidos:0 overruns:0 frame:0
           Paquetes TX:6 errores:0 perdidos:0 overruns:0 carrier:0
           colisiones:0 long.colaTX:1000
           Bytes RX:3390 (3.3 KB) TX bytes:628 (628.0 B)

root @ gerlyn-bears ~/mininet-wifi/examples
└─ #

```

Figura 4.18: Parámetros de estaciones emuladas en Mininet-WiFi

Luego de conocer cómo funciona y como está estructurado Mininet-WiFi se procedió a realizar las pruebas donde se evaluó la interconexión entre dos estaciones inalámbricas, utilizando para ello él envió de paquetes ICMP entre nodos, como se muestra en la figura 4.19. En dicha figura se observan dos (2) pruebas realizadas entre dos estaciones inalámbricas, en este caso desde un terminal Xterm de una de las estaciones emuladas.

```

root @ gerlyn-bears ~/Documentos/Material Tesis 131215/Corridas Artículo/Nuevascorridas/
Iperf - Firewall.cx/jperf-2.0.2 - Windows GUI
└─ # ► ping 10.0.0.8 -c 5
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=2.65 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=2.70 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=2.70 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=2.68 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=2.66 ms

--- 10.0.0.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.653/2.680/2.709/0.069 ms
root @ gerlyn-bears ~/Documentos/Material Tesis 131215/Corridas Artículo/Nuevascorridas/
Iperf - Firewall.cx/jperf-2.0.2 - Windows GUI
└─ # ► ping 10.0.0.7 -c 5
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=1 ttl=64 time=2.66 ms
64 bytes from 10.0.0.7: icmp_seq=2 ttl=64 time=2.70 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=64 time=2.70 ms
64 bytes from 10.0.0.7: icmp_seq=4 ttl=64 time=2.71 ms
64 bytes from 10.0.0.7: icmp_seq=5 ttl=64 time=2.67 ms

--- 10.0.0.7 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.662/2.691/2.711/0.050 ms

```

Figura 4.19: Pruebas de conectividad con paquetes ICMP desde la consola xterm de uno de los nodos

Ya ejecutadas las pruebas de conectividad se procedió a medir el desempeño de las conexiones de una de las topologías emuladas utilizando para ello la herramienta Iperf. Como se ilustra en la figura 4.20 donde se puede apreciar los terminales xterm de las dos estaciones inalámbricas a los que se le aplicó la herramienta Iperf, donde una se desempeñó como servidor y la otra como cliente. Esta herramienta tiene muchas opciones a ejecutar. Para esta prueba se decidió medir el Jitter en uno de los enlaces, es por ello que en este caso se requiere medir el flujo UDP. Por defecto Iperf se usa para medir el flujo TCP, al utilizarlo para análisis de flujo UDP debe utilizarse tanto en el servidor como en el cliente el argumento `-u`. El resultado obtenido contiene una métrica extra para el packet loss. Esta debe ser lo más pequeña posible como es natural. Los parámetros que se colocaron en el lado

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

del cliente en la estación 8 (*sta8*) son los siguientes: `iperf -c 10.10.10.2 -p 5566 -t 15` la opción `-c` significa cliente, luego el número de puerto TCP el parámetro `-t` indica el tiempo que duró la transmisión, en este caso se le coloco 15 segundos. En el lado del servidor en la estación 7 (*sta7*) los parámetros son los mismos agregando la `-s` que significa servidor y la opción `-i` que especifica el intervalo medido en segundos en el cual se volverá a realizar la medición.

```
"Node: sta7"
root @ gerlyn-bears ~/Documentos/Tesis Gerlyn/Tesis GEDP/Scripts
└─ # ▶ iperf -c 10.0.0.8 -u -b 10M -t 15 -p 5566
-----
Client connecting to 10.0.0.8, UDP port 5566
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 21] local 10.0.0.7 port 47903 connected with 10.0.0.8 port 5566

"Node: sta8"
root @ gerlyn-bears ~/Documentos/Tesis Gerlyn/Tesis GEDP/Scripts
└─ # ▶ iperf -s -u -p 5566 -i 1
-----
Server listening on UDP port 5566
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 21] local 10.0.0.8 port 5566 connected with 10.0.0.7 port 47903
[ ID] Interval      Transfer    Bandwidth    Jitter    Lost/Total Datagrams
ms
[ 21] 0.0- 1.0 sec  1.19 MBytes 10.0 Mbits/sec  0.010 ms    0/ 851 (0%)
[ 21] 1.0- 2.0 sec  1.19 MBytes 10.0 Mbits/sec  0.024 ms    0/ 850 (0%)
[ 21] 2.0- 3.0 sec  1.19 MBytes 10.0 Mbits/sec  0.018 ms    0/ 850 (0%)
[ 21] 3.0- 4.0 sec  1.19 MBytes 10.0 Mbits/sec  0.025 ms    0/ 851 (0%)
[ 21] 4.0- 5.0 sec  1.19 MBytes 10.0 Mbits/sec  0.025 ms    0/ 850 (0%)
[ 21] 5.0- 6.0 sec  1.19 MBytes 10.0 Mbits/sec  0.014 ms    0/ 850 (0%)
[ 21] 6.0- 7.0 sec  1.19 MBytes 10.0 Mbits/sec  0.024 ms    0/ 851 (0%)
[ 21] 7.0- 8.0 sec  1.19 MBytes 10.0 Mbits/sec  0.014 ms    0/ 850 (0%)
[ 21] 8.0- 9.0 sec  1.19 MBytes 10.0 Mbits/sec  0.006 ms    0/ 850 (0%)
[ 21] 9.0-10.0 sec  1.19 MBytes 10.0 Mbits/sec  0.020 ms    0/ 851 (0%)
[ 21] 10.0-11.0 sec 1.19 MBytes 10.0 Mbits/sec  0.022 ms    0/ 850 (0%)
```

Figura 4.20: Pruebas de rendimiento de nodos inalámbricos con Iperf

Se procedió a graficar con Gnuplot el Jitter del flujo UDP en un tiempo dado, con los datos obtenidos en la prueba de rendimiento realizada con Iperf, cuyos resultados se muestran en la figura anterior. Para la construcción de la gráfica Gnuplot fue ejecutado directamente desde un terminal del nodo *sta8*. En la figura 4.21 se puede apreciar la gráfica del flujo UDP.

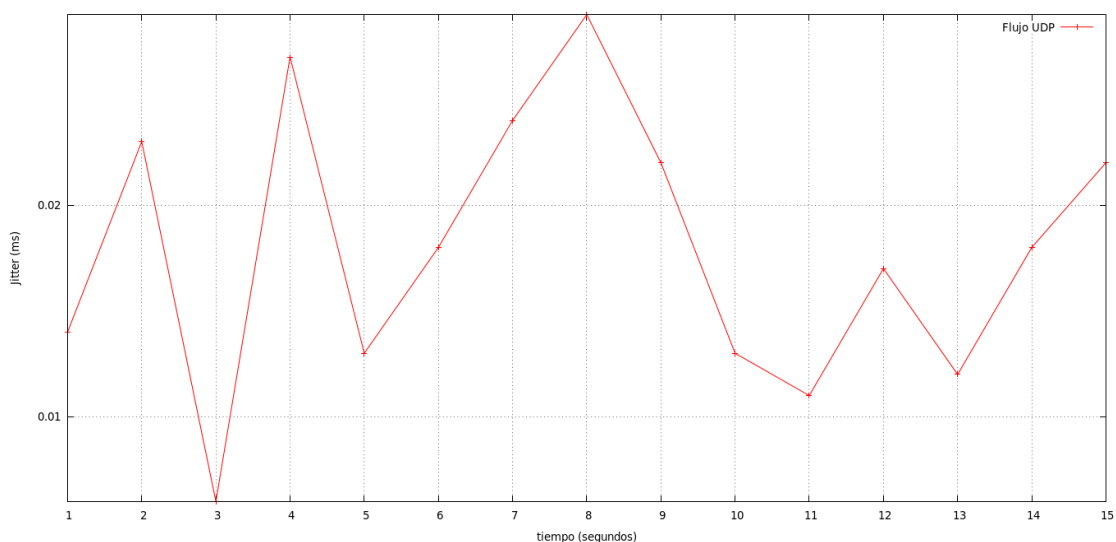


Figura 4.21: Gráfica del Jitter resultado de la prueba de rendimiento

4.3. PRUEBAS DE RENDIMIENTO Y USO DEL SOFTWARE

Mininet-WiFi permite evaluar la movilidad de los nodos de una red SDWN emulada. Como se destaca en la gráfica 4.22 emular la movilidad de los dispositivos portátiles es posible. La parte izquierda de la figura muestra el resultado de una prueba de rendimiento enviando paquetes ICMP, los primeros paquetes son recibidos sin ningún problema, luego existe una espera hasta el paquete 34 que es donde confirma que comienzan a llegar nuevos paquetes, esto se debe a que la estación 2 (*sta2*) que es la que recibe los paquetes, está emulando la movilidad y en ese lapso de tiempo la misma salió de la zona de cobertura inalámbrica del punto de acceso (*ap1*) por lo cual pierde la conexión con la estación 1 (*sta1*), esto se puede apreciar en la gráfica derecha de la figura 4.23 generada por Mininet-WiFi, al ejecutar el script.

La figura 4.23 permite visualizar como es el recorrido por la zona de cobertura de estos dispositivos portátiles y la conexión con el *ap1*.

Mininet-WiFi para realizar la emulación de movilidad inalámbrica y propagación por ahora cuenta con los siguientes modelos: RandomWalk, Truncated Levy Walk, Random Direction, Random WayPoint, Gauss-Markov.

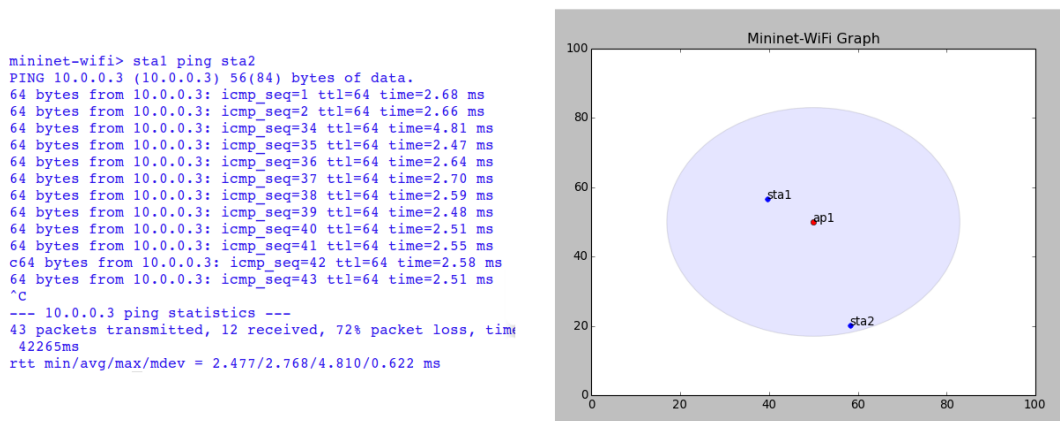


Figura 4.22: Gráfica de Movilidad de nodos en Mininet-WiFi

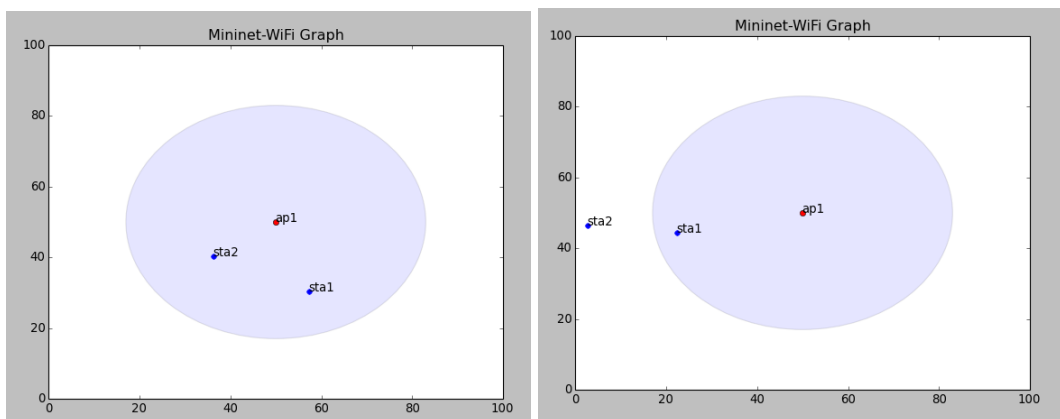


Figura 4.23: Gráfica de Movilidad de nodos en Mininet-WiFi

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

4.4. Medición y Evaluación de la Calidad del Producto Software y de la Experiencia de Usuario

Se realizó la evaluación de la calidad del producto software de las aplicaciones de simulación y/o emulación de SDWN, se seleccionó para ello el modelo de estimación de proyectos en entornos de incertidumbre (EPCU) como el proceso de evaluación para determinar la calidad del producto software basado en la norma ISO/IEC 25000.

El modelo realizó una serie de pasos que permitieron identificar las variables de entrada, siendo estas variables las características y subcaracterísticas de la calidad Externa/Interna de la norma antes mencionada, al igual que determinó una variable de salida, que en esta ocasión fue un índice de calidad y las reglas de inferencia que relacionan las etiquetas lingüísticas (poco, promedio, mucho) de las variables de entrada, con el grado de pertenencia (muy baja, baja, promedio, alta, muy alta) de la variable de salida.

El tener un valor que sea medible de la calidad del producto software, sin duda, dará certeza al usuario, sobre todo si el también participa en la evaluación de la calidad externa del producto de software.

En esta etapa se realizaron actividades de análisis descriptivo y comparación de los indicadores de la calidad del producto software y la experiencia de usuario relacionada con el uso de OpenNet y Mininet-WiFi. Para el análisis de los resultados obtenidos en las actividades experimentales se usaron para facilitar la interpretación de los datos y su seguimiento los software GNU PSPP, GeoGebra, Epidat, RLPLot y la hoja de cálculo Calc integrada en suite ofimática LibreOffice, esto permitió dar soporte al marco de medición y evaluación.

Con el promedio aritmético de los datos obtenidos del cuestionario sobre la calidad del producto software, se calculó el índice de calidad externa/interna y de uso de cada software utilizando para ello los mecanismos basados en el modelo EPCU, de tal manera que se obtuvo como valor de salida un índice de calidad promedio que representa la calidad del producto software.

La fiabilidad del instrumento de medición se determinó a través la obtención del alfa de Cronbach, la misma evaluó el grado de consistencia del cuestionario, en la tabla 4.2 se muestra que el valor obtenido fue de $\alpha = 0,89$, fundamentado en la tabla 2.3 se aprecia la valoración del mismo es bueno, con un valor cercano a uno (1), esto indica que el instrumento de evaluación de la calidad del producto software presenta una alta fiabilidad interna.

Tabla 4.2: Estadística de Fiabilidad

Alfa de Cronbach	Nro de elementos
0,89	82

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

4.4.1. Medición de la calidad del producto software de OpenNet

Se ejecutó el análisis descriptivo para procesar estadísticamente los valores generados en los cuestionarios sobre calidad de producto software para OpenNet, arrojando los resultados que se sintetizan en la tabla 4.3. El cuestionario para determinar la calidad del producto software para OpenNet se aplicó de forma on-line, siendo respondido por siete (7) usuarios con un nivel avanzado en el área los cuales tienen estudios en Ph.D. o están en proceso de obtener el mismo, estos usuarios son procedentes de la República China-Taiwan.

De los datos mostrados en la tabla 4.3 se trabajó para ser más detallados solo con la media aritmética de cada característica de la norma ISO/IEC 25000, dicho promedio se utilizó como valor de entrada para evaluar y determinar la calidad del producto software empleando los mecanismos basados en el modelo EPCU. El promedio se obtuvo considerando las opiniones de todos los participantes que respondieron el instrumento de evaluación.

Tabla 4.3: Resultados del análisis estadístico descriptivo del cuestionario de calidad para OpenNet

Variable	N	Media	Moda	Desv Std	Varianza	Mínimo	Q1	Mediana	Q3	Máximo	Curtosis	Asimetría
FUNCIONABILIDAD	3	3,8	4	0,35	0,12	3,4	4	4	4	4	-1,5	-1,73
CONFIABILIDAD	12	2,7	2,4	1,02	1,04	0	3,4	2	4	4	2,012	-1,55
MANTENIBILIDAD	15	3,3	3,4	1,02	1,05	1	4	3	5	5	0,128	-0,77
EFICIENCIA	6	4,1	4	0,24	0,006	4	4	4	4,6	4,6	1,2	2,45
PORTABILIDAD	14	3,5	5	1,83	3,35	0	5	4	5	5	-0,4	-1,21
COMPATIBILIDAD	5	3,1	.	1,85	3,41	0	4,5	3	4,6	4,6	-0,359	-1,55
OPERABILIDAD	7	1,3	2	1,11	1,24	0	3	1	3	3	-1,917	0,25
SEGURIDAD	8	1,2	0	1,82	3,32	0	2,9	0	4	4	-1,074	1,06
USABILIDAD	11	3,6	3,4	0,87	0,76	2	4,4	3	5	5	-0,605	-0,24
FLEXIBILIDAD	3	4	3,8	0,35	0,12	3,8	4,4	3,8	4,4	4,4	-1,5	1,73
SEGURIDAD DE USO	5	4,8	5	0,21	0,04	4,5	5	5	5	5	-1,241	-0,52

Todas las variables son compuestas (integradas de varios ítems). La columna N representa el número de casos válidos para cada variable, donde N es el número de preguntas que tiene el instrumento de evaluación de la calidad del producto software. El N total de la muestra es de 82, como se aprecia en la tabla 4.3, el número de casos es distinto en las diferentes variables. La variable con mayor promedio fue la Seguridad de Uso y la más baja la Seguridad. Como ya se ha dicho anteriormente por ser un análisis estadístico descriptivo estos datos corresponden a medidas de tendencia central, medidas de distribución y dispersión de las variables, en el Apéndice E se pueden ver más detallados estos resultados.

La figura 4.24 muestra los diagramas de caja de la distribución de datos mostrados en la tabla 4.3, los cuales fueron el resultado de la valoración de los expertos, a cada característica de la calidad interna/externa de la norma ISO 25000 sobre el software OpenNet.

De esta figura se desprende que la valoración dada por los expertos para cada característica, oscila dentro de un rango entre los valores 2 y 4 de la escala preestablecida, exceptuando la Portabilidad y la Seguridad los cuales la valoración dada por los expertos fue más variada en estas características.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

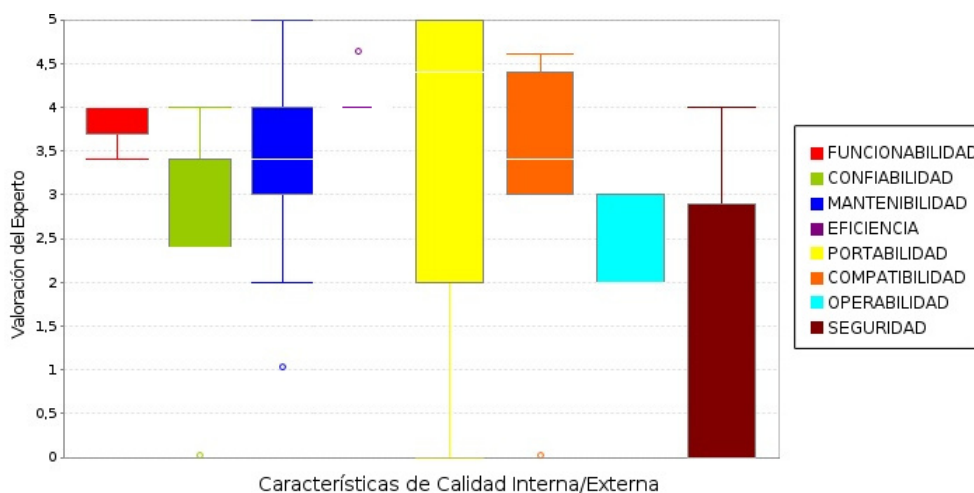


Figura 4.24: Características de la Calidad Interna/Externa en OpenNet

La figura 4.25 presenta un resumen de la distribución de los valores de las características de la calidad de uso para el software OpenNet, donde se puede apreciar que la Seguridad de Uso fue la característica que tuvo mayor valoración, la Flexibilidad también tuvo una valoración homogénea entre todos los expertos semejante a la de la Seguridad de Uso, en contra parte en la Usabilidad su distribución de datos fue más heterogénea, y la valoración de la misma se aprecia que es más baja que las otras dos características.

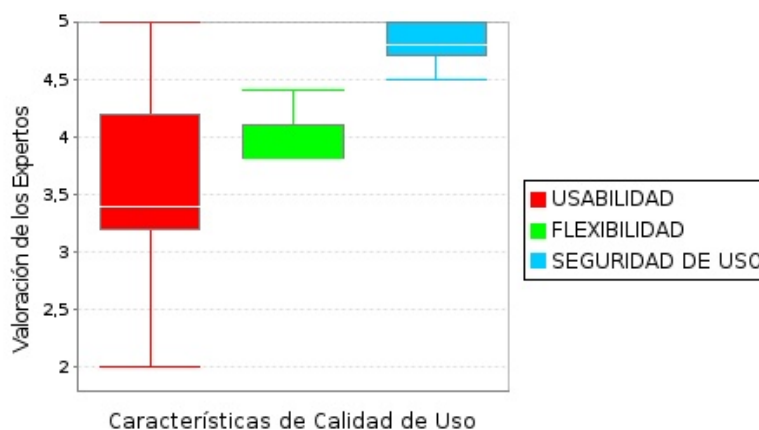


Figura 4.25: Características de la Calidad de Uso para OpenNet

4.4.1.1. Calidad Interna/Externa de OpenNet

Se puede observar en la figura 4.26 la gráfica que se generó de relacionar la media aritmética de las preguntas o ítems del cuestionario con las características de la calidad interna/externa para OpenNet mostradas en la tabla 4.3.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

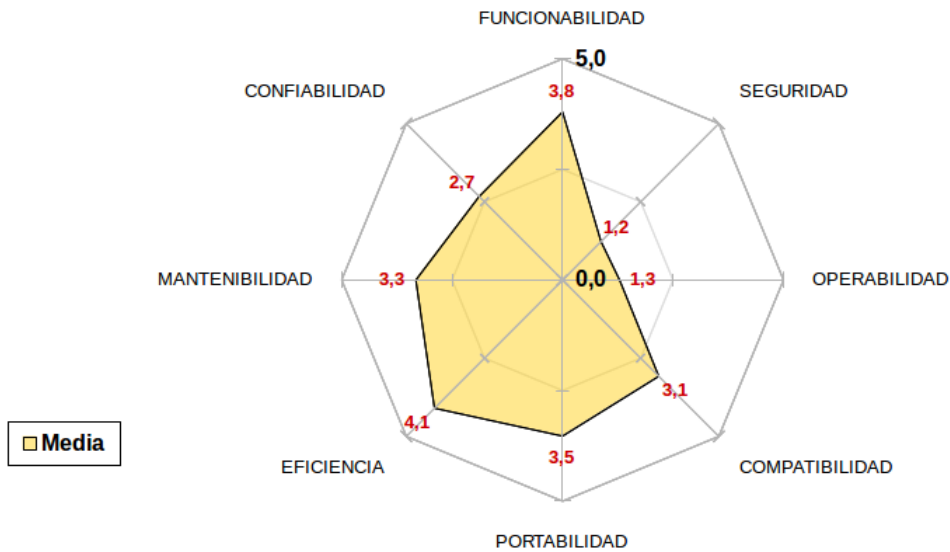


Figura 4.26: Evaluación de la calidad interna/externa en OpenNet

La media aritmética de cada característica de calidad del estándar ISO/IEC 25000 representa la evaluación de los expertos a las variables de entrada, las cuáles se determinaron en base al instrumento de evaluación de la calidad del producto software.

El funcionamiento y uso del mecanismo EVA20131206SQRINT que está fundamentado en el modelo EPCU se puede observar en la figura 4.27.

El valor de salida del mecanismo o índice de calidad para OpenNet fue de 57,34%, el mismo se obtuvo considerando la evaluación de las variables de entrada en dicho mecanismo.

El índice de calidad que se obtiene de la aplicación del modelo EPCU, el cual se relaciona a través de reglas de inferencia, puede ser alcanzado vinculando distintos valores de las entradas de cada categoría, es decir, se puede obtener un mismo índice de calidad con distintos valores en estas categorías.

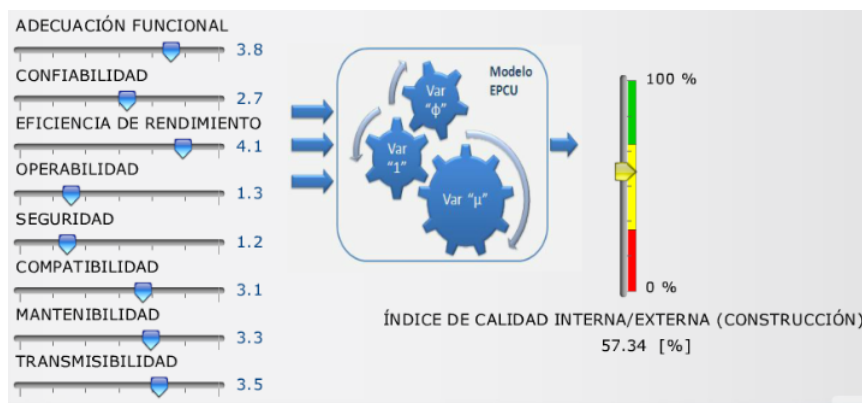


Figura 4.27: Índice de calidad interna/externa de OpenNet con mecanismo EPCU (con autorización de www.mepe.com.mx)

Fuente: <http://www.mepe.com.mx/mecanismos-por-categoria/>

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

4.4.1.2. Calidad de Uso de OpenNet

Como se aprecia en la figura 4.28, se procedió a graficar la media aritmética obtenida del análisis estadístico del instrumento de evaluación, de las características de calidad de uso de la norma ISO/IEC 25000.

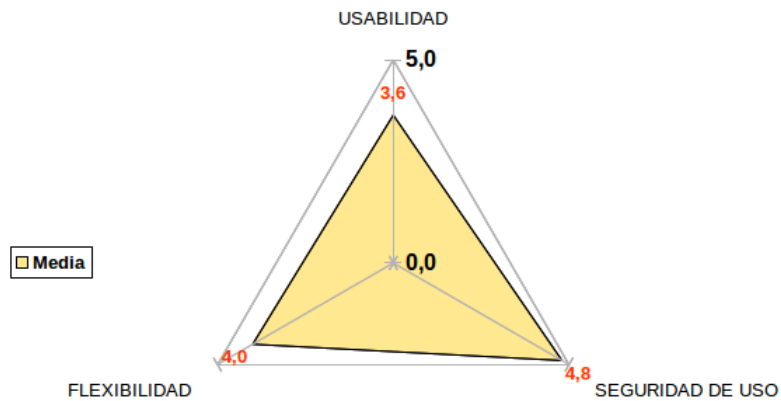


Figura 4.28: Evaluación de la calidad de Uso en OpenNet

De igual forma se realizó la evaluación de la calidad de uso. Para este cálculo se utilizó el mecanismo EVA20131206SQREXT que está idénticamente fundamentado en el modelo EPCU, la ejecución del mismo se puede observar en la figura 4.29. Para este caso de obtención de la calidad de uso OpenNet obtuvo un valor de 96,83 %.

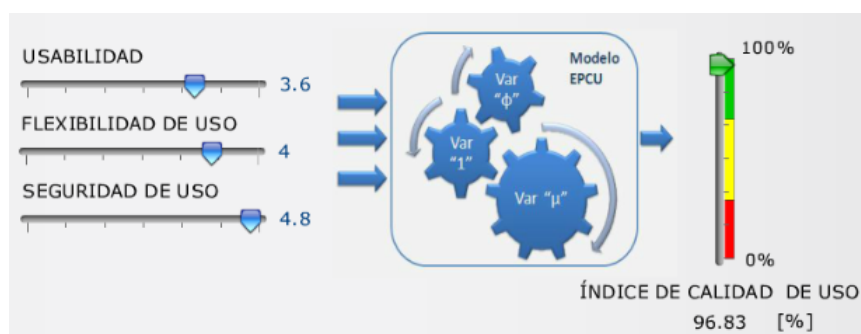


Figura 4.29: Índice de calidad de Uso de OpenNet con mecanismo EPCU (con autorización de www.mepe.com.mx)

Fuente: <http://www.mepe.com.mx/mecanismos-por-categoria/>

Los valores obtenidos con los mecanismos EPCU para OpenNet presenta una calidad interna baja 57,34 % en relación al nivel obtenido de la calidad que percibe el usuario (calidad externa) que fue de 96,83 %.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

4.4.2. Medición de la calidad del producto software de Mininet-WiFi

La tabla 4.4 es el resultado del análisis descriptivo de los valores generados en los cuestionarios sobre calidad de producto software para Mininet-WiFi. De la población estudiada, diez (10) sujetos respondieron el cuestionario, estando compuesta la población y la muestra determinada por expertos en el manejo de este software. Estos usuarios de igual forma que los del software anterior poseen un nivel avanzado en el área, teniendo estudios a nivel de Ph.D. o están en proceso de obtener el mismo, la procedencia de estos es de la República del Brasil y de China-Taiwán en su mayoría. Al igual que en el caso analizado anteriormente se trabajó con la media aritmética de cada característica de la norma ISO/IEC 25000, dicho promedio permitió evaluar los mecanismos para determinar la calidad del producto software empleando el modelo EPCU.

Tabla 4.4: Resultados del análisis estadístico descriptivo del cuestionario de calidad para Mininet-WiFi

Variable	N	Media	Moda	Desv. Std.	Varianza	Mínimo	Q1	Mediana	Q3	Máximo	Curtosis	Asimetría
FUNCIONABILIDAD	3	4,2	.	0,21	0,04	4	4	4	4,4	4,4	-1,5	1,29
CONFIABILIDAD	12	4,4	4	0,53	0,28	3,5	4	4	5	5	-1,342	0,03
MANTENIBILIDAD	15	4,2	4 & 5	1,05	1,1	1	4	5	5	5	3,69	-2,18
EFICIENCIA	6	4,4	5	0,74	0,54	3,5	3,5	5	5	5	-1,561	-0,71
PORTABILIDAD	14	4	5	1,77	3,12	0	4	5	5	5	1,439	-1,96
COMPATIBILIDAD	5	4,1	5	2,01	4,05	0,5	2,5	5	5	5	0,25	-2,24
OPERABILIDAD	7	4,6	5	0,53	0,29	4	4	5	5	5	-1,917	-0,37
SEGURIDAD	8	1,8	0	1,75	3,07	0	0	2	3	4,5	-1,345	0,37
USABILIDAD	11	4,6	5	0,49	0,24	4	4	5	5	5	-1,757	-0,43
FLEXIBILIDAD	3	4,7	5	0,58	0,33	4	4,5	5	5	5	-1,5	-1,73
SEGURIDAD DE USO	5	4,8	4,7	0,16	0,03	4,6	4,65	5	4,95	5	-1,422	0,52

Se aprecia en la tabla 4.4 los promedios de las variables que sirvieron de entradas de los mecanismos de evaluación del modelo EPCU, en la determinación de los índices de calidad interna/externa y de uso, las cuales se determinaron a partir del instrumento de evaluación son compuestas (integradas de varios ítems). Al igual que en el caso de OpenNet, la columna N representa el número de casos válidos para cada variable, donde N representa el número de preguntas en el instrumento de evaluación de la calidad del producto software. El N total de la muestra es de 82, el número de casos es distinto en las diferentes variables. La variable con mayor promedio fue la Seguridad de Uso y la más baja la Seguridad.

Los datos de la tabla 4.4 se sintetizan en la figura 4.30, donde se pueden observar los diagramas de caja de los valores obtenidos de las características de la calidad interna/externa, generada del instrumento de evaluación sobre el software Mininet-WiFi. A partir de esta figura se observa que la valoración dada por los expertos para cada característica, las mismas se encuentran dentro de un rango de datos entre 3,5 y 5 en la mayoría de las características, exceptuando la Compatibilidad y la Seguridad los cuales la valoración dada por los expertos si fue más variada, siendo esta ultima la que presentó menor valoración.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

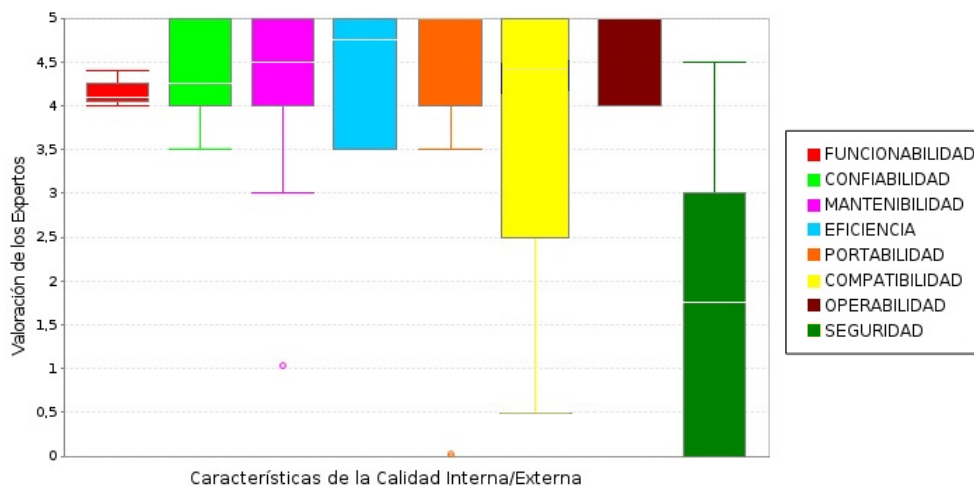


Figura 4.30: Características de la Calidad Interna/Externa en Mininet-WiFi

La figura 4.31 muestra la distribución de los valores de las características de la calidad de uso para el software Mininet-WiFi, donde se puede apreciar que la Seguridad de Uso fue la característica que tuvo mayor valoración en la opinión de todos los expertos que respondieron el instrumento de evaluación. La Flexibilidad posee una valoración homogénea y alta, al igual que la Seguridad de Uso, a diferencia de la Usabilidad en la cual su distribución de datos fue más heterogénea y de menor valoración, en comparación con las otras dos características. También se debe destacar que el rango de datos de las tres características evaluadas está entre 4 y 5 de la escala definida.

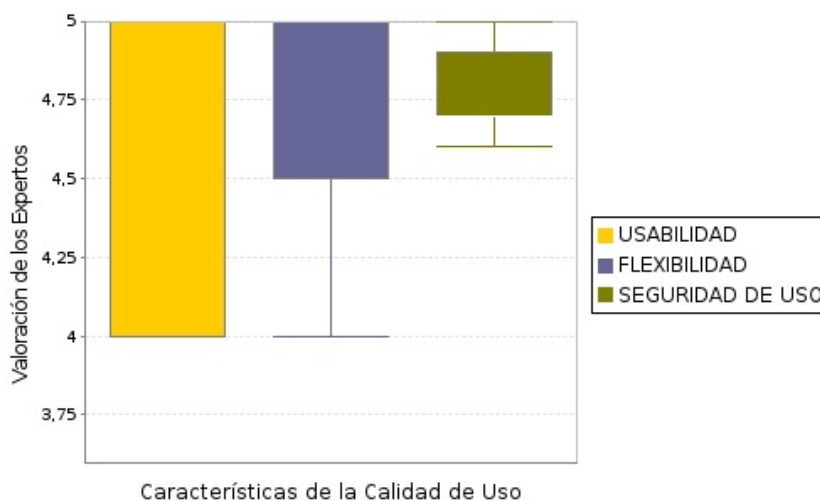


Figura 4.31: Características de la Calidad de Uso para Mininet-WiFi

4.4.2.1. Calidad Interna/Externa de Mininet-WiFi

Se procedió a graficar la media aritmética de las características pertenecientes a la calidad de producto interna/externa en este caso para Mininet-WiFi. Una forma alternativa de presentar el promedio de estas características se puede observar en la figura 4.32.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

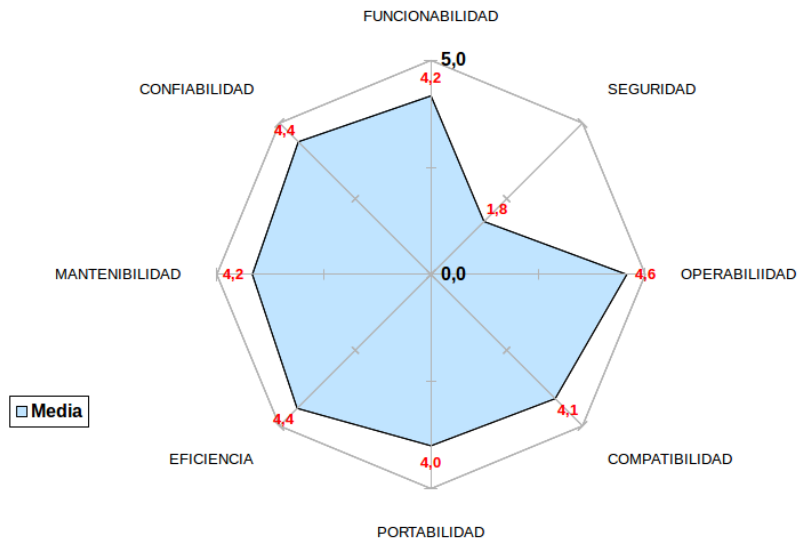


Figura 4.32: Evaluación de la calidad interna/externa en Mininet-WiFi

Como ya se mencionó anteriormente, las variables de entrada para el cálculo del índice de la calidad interna/externa del producto software se representan con la media aritmética, para este caso es el resultado de la ponderación de las respuestas respondidas por los expertos. El funcionamiento y uso del mecanismo EVA20131206SQRINT se puede observar en la figura 4.33. De este mecanismo se puede apreciar que Mininet-WiFi obtuvo un valor o índice de calidad de 84,38 %, lo que representa que Mininet-WiFi posee una alta calidad interna/externa de software.

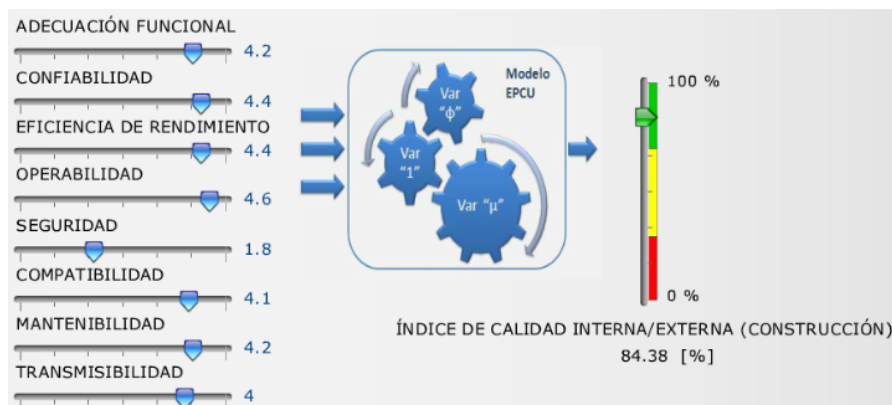


Figura 4.33: Índice de calidad interna/externa de Mininet-WiFi con mecanismo EPCU (con autorización de www.mepe.com.mx)

Fuente: <http://www.mepe.com.mx/mecanismos-por-categoria/>

4.4.2.2. Calidad de Uso de Mininet-WiFi

La gráfica de la media aritmética de las características de uso de la norma ISO/IEC 25000 se puede apreciar en la figura 4.34. La misma visualiza una calidad de uso casi perfecta en el rango de valores que se fijó con anterioridad.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

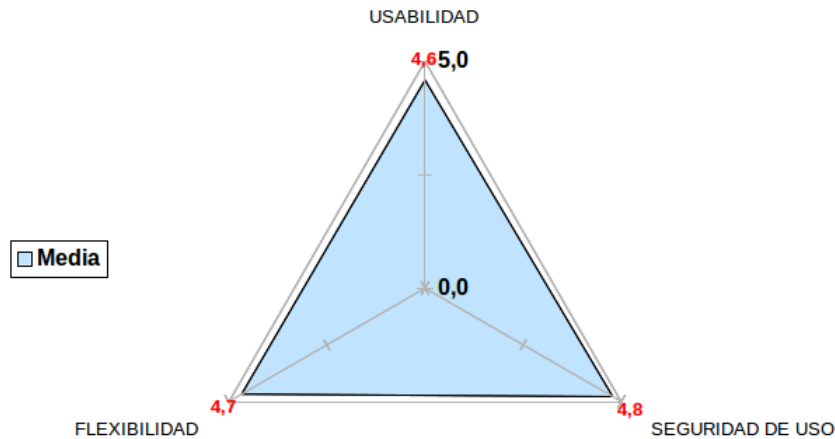


Figura 4.34: Evaluación de la calidad de uso en Mininet-WiFi

Por medio de la evaluación de la calidad de uso utilizando los mecanismos fundamentados en el modelo EPCU, se pudo comprobar que lo visualizado en la figura 4.34 en relación a la alta calidad de uso de Mininet-WiFi es valedero. En la figura 4.35 se muestra el mecanismo EVA20131206SQREXT. Para este caso la calidad de uso de Mininet-WiFi obtuvo un valor de 100 % en la medición del índice de calidad de uso.

Hay que recordar que para la valoración de los índices de calidad realizadas por estos mecanismos, estos se encuentran fundamentados en operaciones bajo lógica difusa y no en operaciones aritméticas.

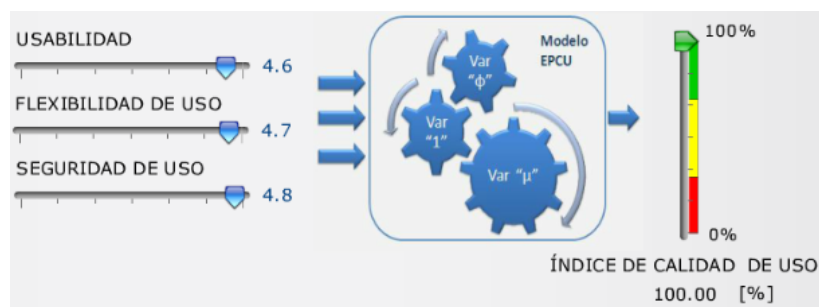


Figura 4.35: Índice de calidad de Uso de Mininet-WiFi con mecanismo EPCU (con autorización de www.mepe.com.mx)

Fuente: <http://www.mepe.com.mx/mecanismos-por-categoria/>

En el caso de Mininet-WiFi al emplear los mecanismos del modelo EPCU, se obtuvo índices de calidad altos, en el caso de la calidad interna/externa fue de 84,38 %, y en el de calidad de uso de 100 %.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

4.4.2.3. Comparación de la Calidad interna/externa y de uso entre OpenNet y Mininet-WiFi

Valdés (2013) señala que la calidad de un software, refiriéndose únicamente al índice de calidad, al valor cuantitativo obtenido de la aplicación de los mecanismos basados en el modelo EPCU al relacionar las distintas categorías a través de reglas de inferencia, puede ser alcanzada con distintos valores de las categorías, es decir, se podría obtener un mismo valor de 89% de calidad con distintos valores de las categorías.

Las diferencias entre OpenNet y Mininet-WiFi se reflejan gráficamente en la relación de la media aritmética con las características de la calidad del producto software en la figura 4.36, estos valores repercuten directamente en el índice de calidad, en el caso de la calidad interna y externa, y en la figura 4.37 para la calidad de uso.

En el caso de la calidad de construcción (interna/externa) se puede ver la gran diferencia en cuanto a calidad de software entre ambos programas, mas no en el caso de la calidad de uso donde sus valores son cercanos. OpenNet presenta una buena calidad de uso de (96.83%) sin embargo a nivel de la calidad interna/externa muy baja (57,34%).

Mininet-WiFi presentó un valor de índice de la calidad interna/externa de 84,38% y de calidad de uso de 100%. A nivel de índice de calidad de uso no existen diferencias estadísticamente significativa, no así en el caso del índice de calidad interna/externa.

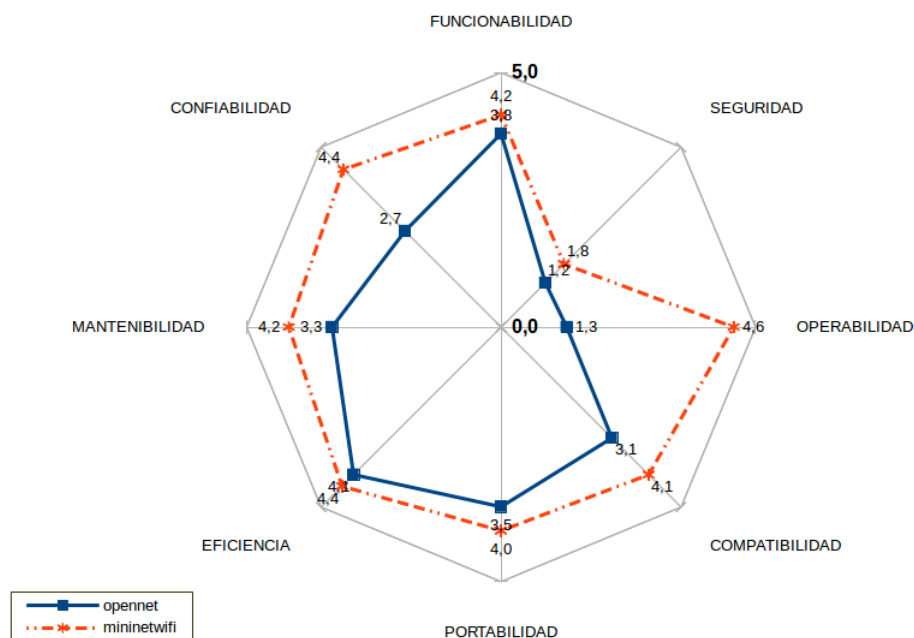


Figura 4.36: Comparación gráfica de la calidad interna/externa

Por ello se debe realizar la importancia de medir la experiencia de usuario para profundizar la visión de los usuarios en relación al uso de OpenNet.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

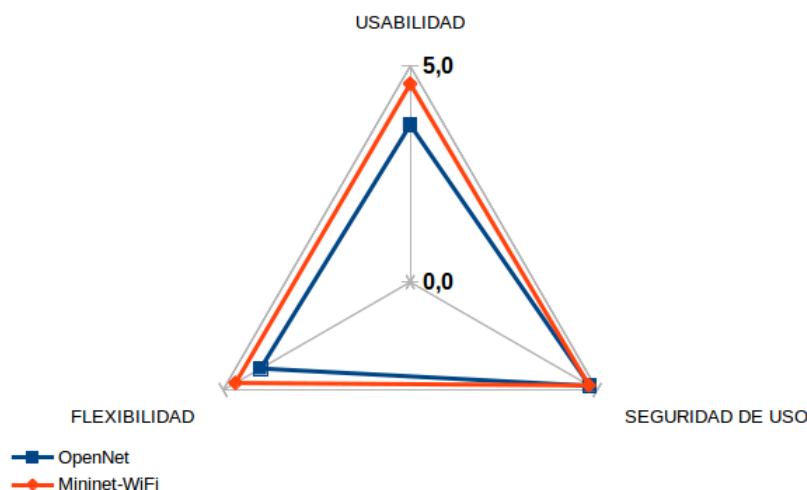


Figura 4.37: Comparación gráfica de la calidad de uso

4.4.3. Medición de la Experiencia de Usuario

Como ya se mencionó anteriormente la UX puede caracterizarse a través de la calidad en uso del producto software, ya que la misma representa un cambio emergente del propio concepto de usabilidad.

El grupo de usuarios que evaluará la UX, es capacitado por el equipo de expertos y reciben el material de evaluación. Dicho grupo trabajó de forma independiente durante aproximadamente 40 minutos en cada software. El material consiste en un formulario con una lista de tareas que se deben responder vía on-line. Además reciben un checklist para comentar sus impresiones o dificultades y evaluar su experiencia. Posteriormente, el equipo se encargó de analizar los resultados e impresiones individuales de los evaluados, siguiendo los estándares de la metodología de la Revisión de Usabilidad (Usability Review). La muestra de esta etapa consistió en siete (7) usuarios de diferentes perfiles (novatos, intermedios y expertos) que laboran en el sector de las tecnologías de la información y comunicación en el Rectorado de la Universidad de Los Andes en Mérida-Venezuela. Hay que destacar que estos usuarios no poseen conocimiento alguno en el área de las SDN, SDWN y calidad de software, ni tampoco conocían la existencia y uso de las aplicaciones estudiadas.

Al ser estos usuarios un grupo distinto y los cuales se centraron solo en conocer y usar ambos software por un tiempo determinado, para luego responder la escala SUS, esto permitió tener un valor menos sesgado en referencia a la usabilidad de OpenNet y Mininet-WiFi, que la brindada por los grupos de expertos en OpenNet y en Mininet-WiFi.

Se utilizó para esta valoración la Escala de usabilidad del sistema (SUS), La tabla 4.5 muestra la valoración de esta escala.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

Tabla 4.5: Valores de evaluación de la Escala de usabilidad del sistema (SUS)

1	2	3	4	5
Muy en desacuerdo	En desacuerdo	Ni de acuerdo, ni en desacuerdo	De acuerdo	Muy de acuerdo

Para la evaluación de SUS la misma se fundamenta en una regla sobre la valoración de las preguntas, de la forma siguiente:

- Las preguntas con número impar impactan de manera positiva la escala. El procedimiento para calcular el aporte de cada una se hace de la siguiente manera:

$$\text{Valor reflejado por el encuestado} - 1 \text{ (menos uno)} \quad (4.1)$$

- Las afirmaciones con número par impactan de manera negativa. El procedimiento para calcular el aporte de cada una se hace de la siguiente manera:

$$5 - \text{Calificación obtenida} \quad (4.2)$$

Estas operaciones aritméticas permiten obtener una calificación parcial de cada afirmación, las cuales se suman y el total se multiplica por 2,5.

$$\text{Valor global SUS} = \left[\sum (\text{preguntas}_{\text{impares}}) + \sum (\text{preguntas}_{\text{pares}}) \right] \times 2,5 \quad (4.3)$$

Esto permitirá obtener un valor entre 0 y 100 que será el valor global de SUS, resaltando que este valor no es un porcentaje.

En la tabla 4.6 podemos observar los parámetros estadísticos que se generaron del cuestionario SUS aplicado a los encuestados (ver Apéndice E).

Tomando como referencia la media aritmética calculada del valor global SUS, que se generó de las respuestas dadas por los usuarios entrevistados, OpenNet presenta un factor muy bajo de 13,4. Si este valor lo comparamos con la escala de la figura 2.9, se obtendrá un valor de porcentaje de usabilidad demasiado bajo. En caso de Mininet-WiFi su resultado en la misma escala de la figura 2.9 representa un porcentaje de usabilidad bueno del orden de los 77,6.

Tabla 4.6: Resultados del análisis estadístico de la Escala de usabilidad del sistema (SUS)

Software	Media	Varianza	Desv. Estándar	Mínimo	1 ^{er} Cuartil	Mediana	3 ^{er} cuartil	Máximo
OpenNet	13,4	3,1	3,3	10	11	13	15	20
Mininet-WiFi	77,6	4,2	4,6	70	75	78	80	85

La figura 4.38 compara los resultados obtenidos en el cálculo del valor global SUS por cada uno de los usuarios entrevistados sobre la UX en los software estudiados. En la parte

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

superior se refleja la curva de la puntuación recibida por Mininet-WiFi, estando todos sus valores por encima o muy cerca de la media del valor global SUS para este programa. En la parte inferior se visualiza la gráfica del factor SUS por usuario y su relación con su media en OpenNet, estando la mayoría de los datos por debajo de la misma. Este gráfico es bastante revelador en cuanto a la diferencia existente entre los dos software en relación a la puntuación obtenida con el valor global SUS, para determinar la UX.

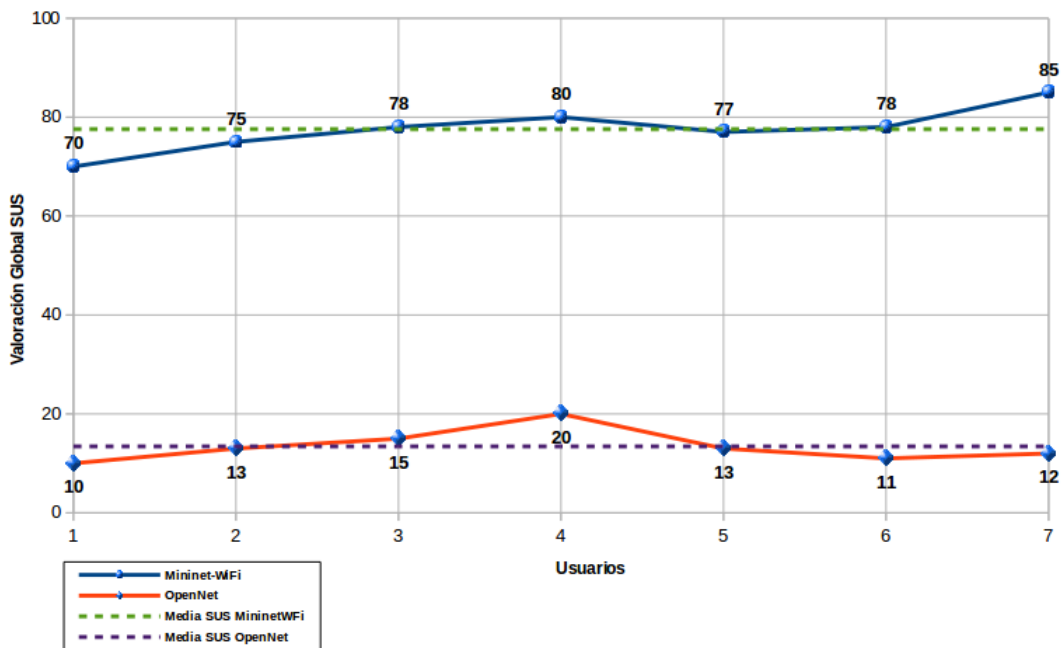


Figura 4.38: Valor Global SUS de OpenNet y Mininet-WiFi

Se puede apreciar en la figura 4.39 el diagrama de caja obtenido de los datos de la tabla 4.6. En dicha figura se puede ver la diferencia en la valoración SUS de cada software. Los datos en cada caja se distribuyen de forma homogénea, esta distribución es debido a la cercanía de las valoraciones SUS de cada usuario.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

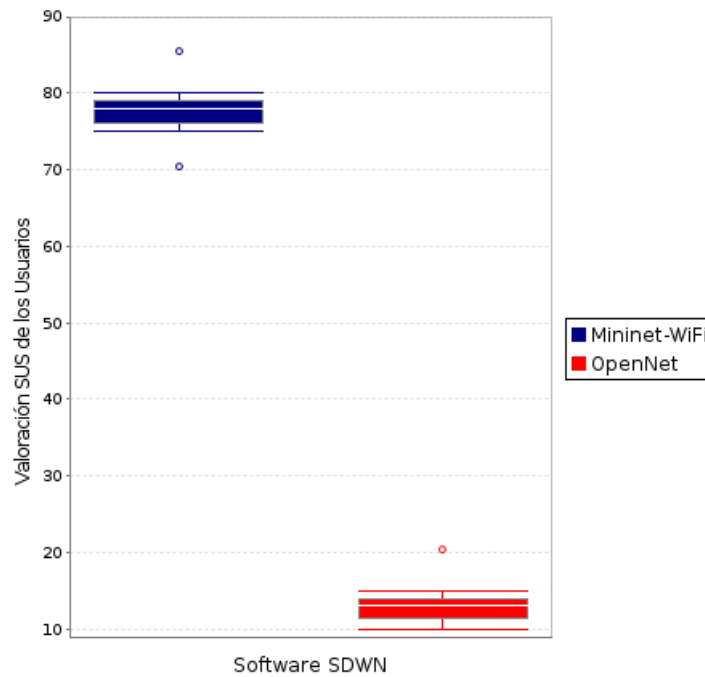


Figura 4.39: Relación de los software SDWN

4.4.4. Interrelación de la Usabilidad con la UX en OpenNet

No se pueden comparar directamente la UX con la Usabilidad, por las diferentes escalas en las que ambas se encuentran.

Para contrastar ambos resultados se toma en consideración el hecho que el conjunto de datos de las valoraciones obtenidas tanto para la Usabilidad y las puntuaciones SUS, se pueden aproximar sus histogramas a una curva Normal con los parámetros μ (*Media aritmética*) y σ (*Desviación Estándar*). Como ambas muestras se comportan de modo Normal, entonces se tipifican y observan las puntuaciones sobre una distribución de referencia $X \sim N(0, 1)$.

La figura 4.40 muestra el histograma de las Puntuaciones SUS de OpenNet, aproximándose los mismos a una curva Normal.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

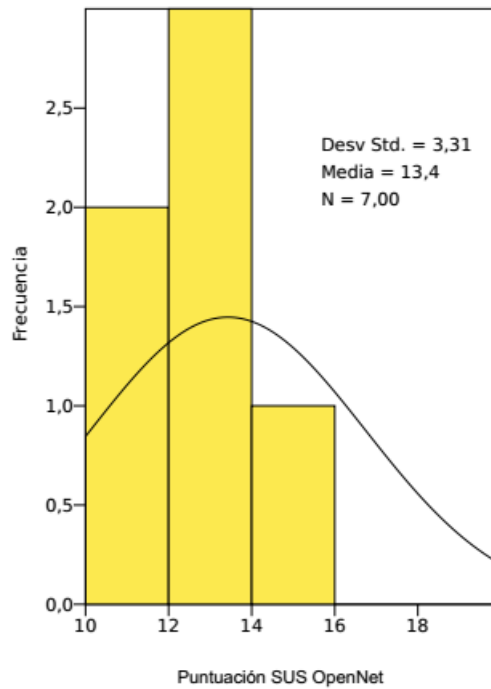


Figura 4.40: Puntuaciones SUS de OpenNet aproximado a la curva Normal

En la tabla 4.7 se encuentran los resultados normalizados de la Usabilidad en la Calidad de Uso de OpenNet, dichos resultados fueron asignados en el instrumento de evaluación que permitió cuantificar las características de la norma ISO/IEC 25000 por el grupo de expertos en OpenNet ($Z_{EXP\ OpenNet}$). De igual forma se visualiza la puntuación SUS al determinar la UX por el grupo de usuarios seleccionados ($Z_{SUS\ OpenNet}$), ambas mediciones fueron tipificadas utilizando la ecuación 2.3.

Tabla 4.7: Resultados Normalizados de la Usabilidad y UX de OpenNet

Software	N	media	σ	S	Min	Q1	Mediana	Q3	Max
ZSUS Opennet	7	0	0,9258	1	-1,036	-0,7338	-0,1295	0,4748	1,9857
ZEXP Opennet	5	0	0,9535	1	-1,8788	-0,7307	-0,2714	0,8768	1,5657

En la figura 4.41 se aprecian los diagramas de caja de los valores normalizados de OpenNet, los mismos se construyeron en base a los valores de la tabla 4.7.

Se puede apreciar en esta figura que entre ambos gráficos existe un intervalo en común o la intersección de ambos diagramas de caja, comprendido el intervalo entre $[-0,7307; 0,4748]$.

Por otra parte se debe considerar que ambas puntuaciones difieren del número de usuarios, es por ello que para la determinación de la media corregida utilizada como nueva entrada del modelo EPCU, se tomaron valores que están dentro de ese intervalo de datos.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

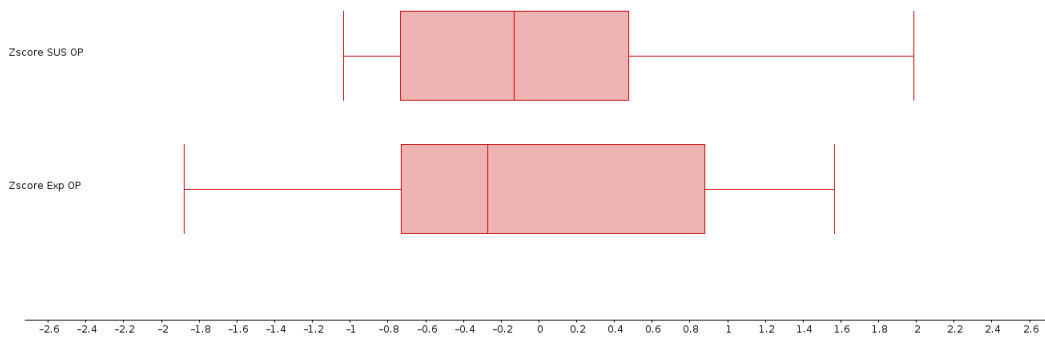


Figura 4.41: Valoraciones normalizadas para OpenNet

La figura 4.42 muestra la distribución Normal del rango de datos tipificados $N(0, 1)$, la zona sombreada de la misma representa la intersección entre los diagramas de caja de la figura 4.41 de los valores tipificados de $Z_{SUS} OpenNet$ y de $Z_{EXP} OpenNet$. Esta zona representa el área bajo la curva o el intervalo donde se determina la probabilidad de que cualquier valor estará dentro de ese rango.

Asimismo se aprecia en la figura los valores mínimos y máximos tipificados z_{min} y z_{max} , los cuales son los valores mínimo y máximo del rango en común de la gráfica de la figura 4.43.

En este caso la probabilidad a determinar del área bajo la curva sera la siguiente:

$$P(-0,7338 \leq z \leq 0,4748)$$

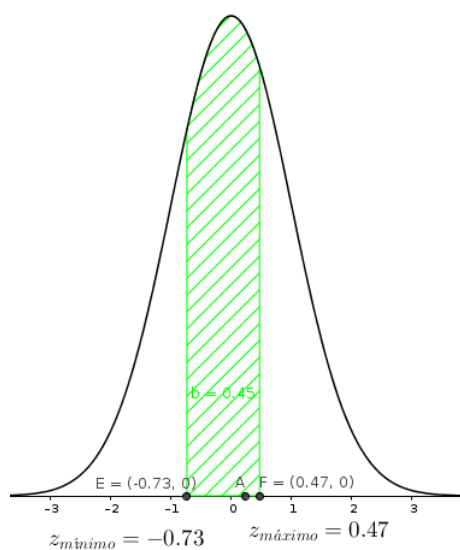


Figura 4.42: Curva Normal de las puntuaciones SUS de OpenNet a $N(0, 1)$

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

En la figura 4.43 se visualizan los datos de la distribución normal sin tipificar de las puntuaciones SUS de OpenNet, esta gráfica nos permitirá obtener un valor x_1 el cual sirve de dato para la determinación de la media corregida, así como los valores mínimos y máximos del rango sombreado, la probabilidad de la zona sombreada será la siguiente, para esta escala:

$$P(10,98 \leq x \leq 14,95)$$

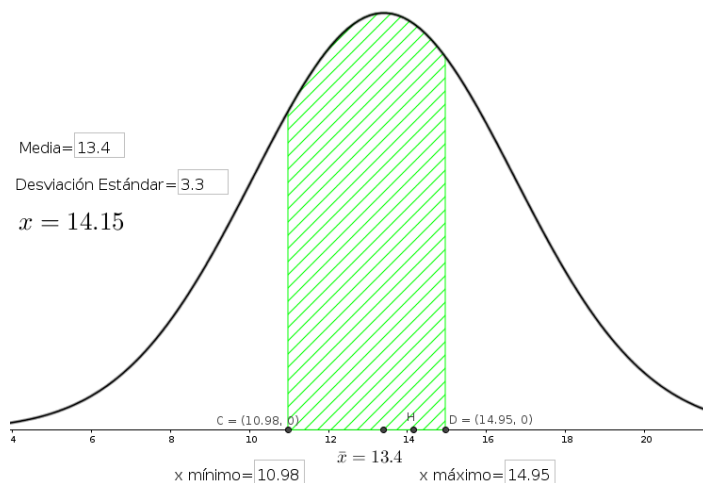


Figura 4.43: Curva Normal de puntuaciones SUS de OpenNet a $N(\mu, \sigma)$

Como ya se referencio anteriormente, la normalización o tipificación de datos se puede tomar como una función lineal, como se aprecia en la figura 4.44.

La gráfica de la izquierda de la figura 4.44, muestra los valores de la puntuación SUS de OpenNet sin tipificar aproximados a una curva normal $N(\mu, \sigma)$, la recta de la derecha es la trasformación lineal de la curva normal tipificada, esta recta fue construida con los datos de la puntuación SUS de OpenNet a partir de la ecuación 2.5.

La tipificación de un valor de la curva se determina de igual forma con la ecuación 2.3, a continuación se presenta un ejemplo:

$$z_{sus} = g(x) = \frac{x_i - \bar{X}}{S} = \frac{14,15 - 13,4}{3,3} = 0,23$$

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

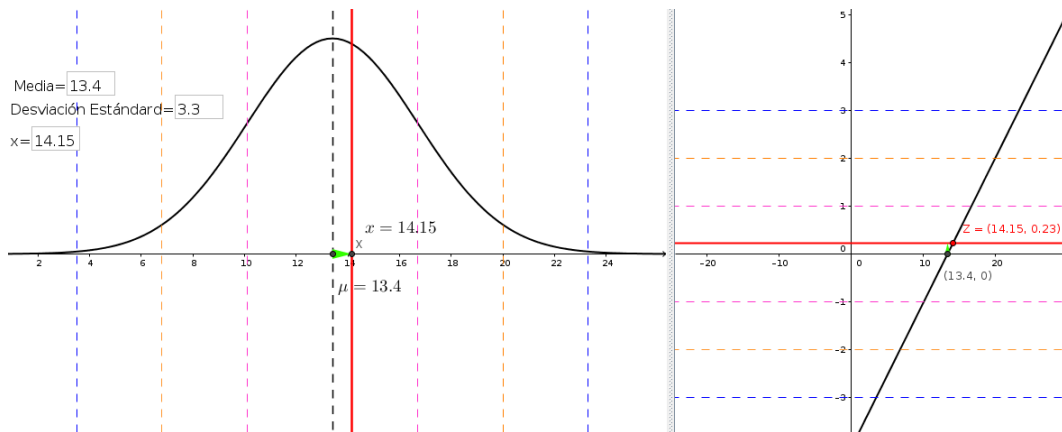


Figura 4.44: Linealización de la curva Normal de los valores SUS de OpenNet

La determinación de la recta de la figura derecha se determina a partir de la siguiente ecuación:

$$x_i = Sz + \bar{X}$$

$$x = 3,3 \times 0,23 + 13,4 = 14,15 \quad (4.4)$$

Si se toma un valor $x_i = 14,15$ de la figura 4.43 o 4.44, al tipificar se obtiene el siguiente resultado:

$$z_{sus} = \frac{x_i - \bar{X}}{S} = \frac{14,15 - 13,4}{3,3} = 0,23 \quad (4.5)$$

La probabilidad de las figuras 4.42 y 4.41 aproximados a la curva normal y la de los valores tipificados son las mismas aunque estén en escalas diferentes, podemos resumir que la probabilidad de ambas es la siguiente:

$$P(10,98 \leq x \leq 14,95) = P(-0,7338 \leq z \leq 0,4748) = 0,45$$

Se puede apreciar el comportamiento de las valoraciones de los expertos a una curva Normal tipificada $N(0,1)$ en la figura 4.45, de igual forma la zona sombreada representa la intersección de los diagramas de caja de la figura 4.41, cuyo rango de z esta en el siguiente intervalo:

$$z_{EXP} = [-0,73 ; 0,47]$$

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

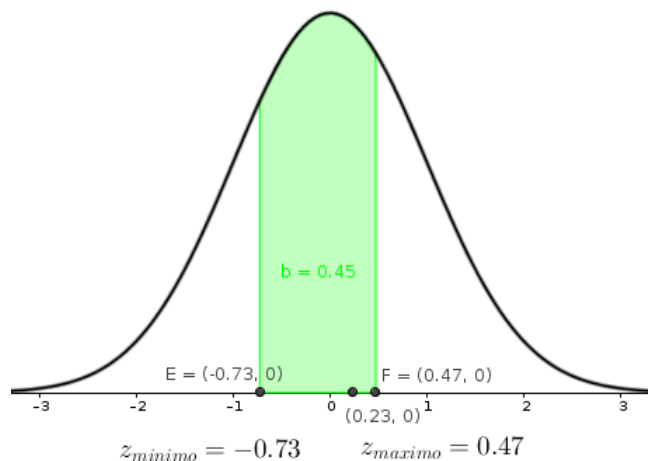


Figura 4.45: Curva Normal de las valoraciones de los expertos en OpenNet a $N(0, 1)$

La figura 4.46, muestra la aproximación a una curva Normal de las valoraciones realizadas por los expertos sobre OpenNet.

Según los valores de medida central para este conjunto de datos, la probabilidad de estas curvas será la siguiente:

$$P(2,97 \leq x \leq 4,01) = P(-0,7338 \leq z \leq 0,4748) = 0,45$$

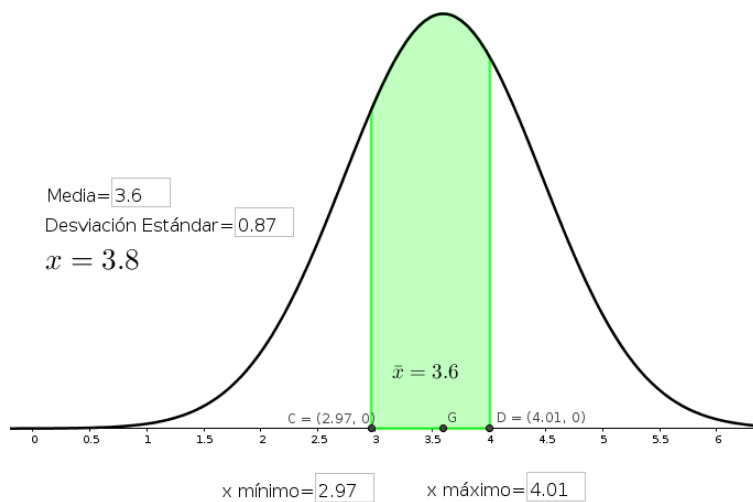


Figura 4.46: Curva Normal de las valoraciones de los Expertos en OpenNet a $N(\mu, \sigma)$

Para estimar la media corregida ($\mu_{\text{corregida}}$) que se utilizó como valor de entrada de la Usabilidad en la determinación del índice de calidad de uso en el modelo EPCU, se realizó de la siguiente forma:

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

Se procedió a calcular dos valores de z_i tomando los datos ya conocidos de \bar{x}_i , S_x , variando el x_i de las figuras 4.43 y la 4.46, que permita satisfacer esta igualdad:

$$z_1 = z_2$$

- De la figura 4.43 se aprecia que los valores de x_i deben estar dentro de este rango:

$$x_{SUS} = [10,98 - 14,95]$$

El valor de la media será constante para este caso y es de $\bar{x}_{SUS} = 3,6$, al igual que la desviación estándar $S_{SUS} = 0,87$.

Por ejemplo al tomar un valor de $x_1 = 14,15$, se obtiene el siguiente valor de z_1 :

$$z_1 = \frac{14,15 - 13,4}{3,3} = 0,23 \quad (4.6)$$

- Como segundo paso se calculó a partir de la gráfica de la figura 4.46 otro valor de z_i , se tomó un dato de x_i que permita que:

$$z_1 = z_2$$

Este valor asumido de x_i debe estar dentro del siguiente rango de datos:

$$x_{EXP} = [2,97 - 4,01]$$

La media será $\bar{x}_{EXP} = 3,6$, al igual que la desviación estándar de los datos es $S_{EXP} = 0,87$.

Se tomó un valor de x_i que este dentro del rango, siendo el mismo de $x_2 = 3,8$, obteniéndose el siguiente valor de z_2 :

$$z_2 = \frac{3,8 - 3,6}{0,87} = 0,23 \quad (4.7)$$

Por lo tanto tenemos de las ecuaciones 4.6 y 4.7 que:

$$z_1 = z_2$$

Se igualaron las ecuaciones,

$$\frac{14,15 - 13,4}{3,3} = \frac{3,8 - 3,6}{0,87} = 0,23 \quad (4.8)$$

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

Se asumió que $\mu_{corregida} \neq \bar{x}_{EXP}$, para luego proceder a sustituir de la ecuación 4.8.

$$\frac{14,15 - 13,4}{3,3} = \frac{3,8 - \mu_{corregida}}{0,87} = 0,23 \quad (4.9)$$

quedando que:

$$\mu_{corregida} = (x_i - \bar{x}_{SUS}) \times \frac{S_{EXP}}{S_{SUS}} \quad (4.10)$$

Se tomaron los diferentes valores de medidas de tendencia central de la tabla 4.7, realizando con estos datos iteraciones utilizando la ecuación 4.10, para luego determinar el promedio de los resultados arrojados en dichas iteraciones, permitiendo esto obtener un valor estimado de la $\mu_{corregida}$, la cual sirvió para determinar el índice de calidad de uso corregido empleando el modelo EPCU. El resultado que se obtuvo fue el siguiente:

$$\mu_{corregida} = 3$$

Se aprecia que este valor, el cual será la nueva entrada de la Usabilidad en el mecanismo EPCU, es menor al que se determinó por el instrumento de evaluación a los expertos.

En la figura 4.47 se pueden apreciar las variables de entrada y el nuevo valor del índice de calidad de uso generado de la interrelación de la Usabilidad y la UX, utilizando el mecanismo fundamentado en el modelo EPCU.

Este nuevo índice de calidad de uso de OpenNet obtuvo un valor de 85,98%. Se puede apreciar que es significativamente menor en relación al índice donde no se interrelaciona la UX.

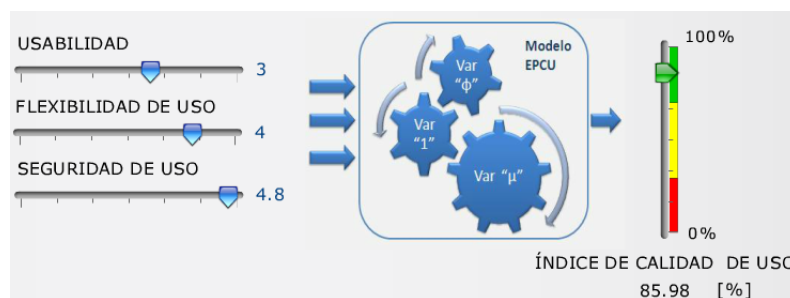


Figura 4.47: Determinación del índice de la calidad de uso de OpenNet a partir de la media corregida (con autorización de www.mepe.com.mx).

Fuente: <http://www.mepe.com.mx/mecanismos-por-categoria/>

La siguiente figura muestra la comparación de las entradas de las dos evaluaciones de la calidad de uso de OpenNet.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

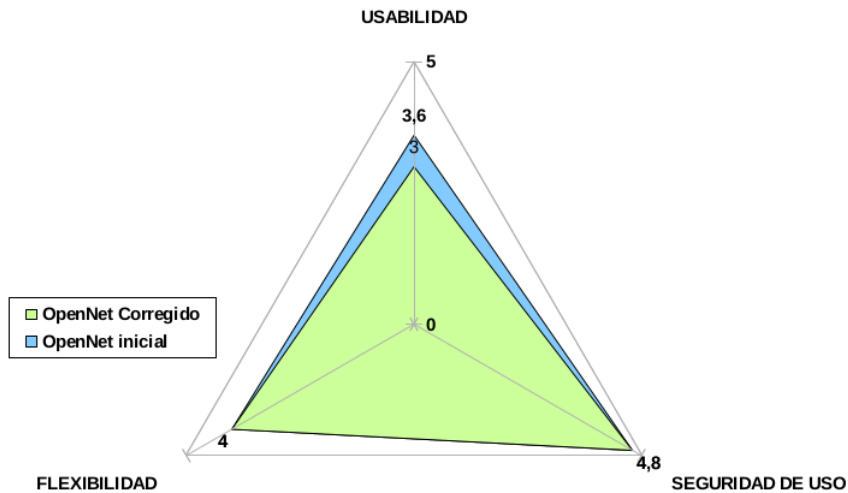


Figura 4.48: Comparación de las características de la calidad de uso en OpenNet

4.4.5. Interrelación de la Usabilidad con la UX en Mininet-WiFi

La interrelación entre la Usabilidad como característica de la calidad de uso y la UX en Mininet-WiFi, se ha estimado de igual forma que se hizo con OpenNet.

La figura 4.49 muestra una aproximación a una distribución Normal a partir del histograma construido de las Puntuaciones SUS de Mininet-WiFi.

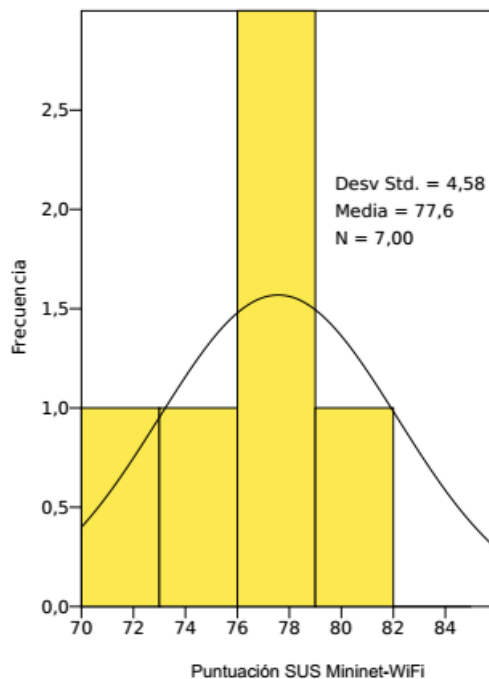


Figura 4.49: Puntuaciones SUS de Mininet-WiFi aproximado a la curva Normal

La siguiente tabla muestra los valores normalizados de la Usabilidad en la Calidad de Uso de Mininet-WiFi, dichos valores son el resultado de las evaluaciones hechas por los expertos

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

en este software ($Z_{EXP} Mininet - WiFi$). Así mismo se visualiza la puntuación SUS obtenida al determinar la UX por el grupo de usuarios seleccionados ($Z_{SUS} Mininet - WiFi$), ambas mediciones fueron tipificadas utilizando la ecuación 2.3.

Tabla 4.8: Resultados Normalizados de la Usabilidad y UX de Mininet-WiFi

Software	N	media	σ	S	Min	Q1	Mediana	Q3	Max
ZSUS Mininet-WiFi	7	0	0,9258	1	-1,6541	-0,5618	0,0936	0,5306	1,6229
ZEXP Mininet-WiFi	10	0	0,9535	1	-1,2039	-1,2039	0,8335	0,8335	0,8335

En la figura 4.50 se aprecian los diagramas de caja de los valores tipificados de Mininet-WiFi, a partir de los datos proporcionados en la tabla 4.8.

Se puede observar que entre ambos gráficos existe un rango donde se interceptan ambos diagramas de caja, el cual está comprendido en el siguiente intervalo $[-0,5618; 0,5306]$.

De igual forma que con OpenNet ambas puntuaciones difieren del número de usuarios, es por ello que para la determinación de la media corregida que se utilizó como nueva entrada del modelo EPCU, se tomaron valores que están dentro de ese intervalo de datos.

También se aprecia en la figura 4.50 que la distribución de los datos de $Z_{EXP} Mininet - WiFi$ es más homogénea que la de $Z_{SUS} Mininet - WiFi$, las puntuaciones SUS hechas por los usuarios son mas dispersas que las evaluaciones aportadas por los expertos.

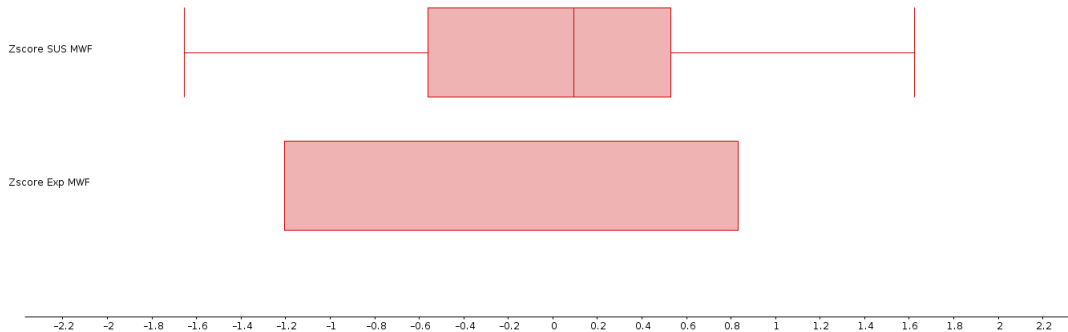


Figura 4.50: Valoraciones normalizadas para Mininet-WiFi

La figura 4.51, muestra una distribución aproximada a una curva Normal del intervalo de intersección de la figura 4.50, con los valores máximo y mínimo del intervalo.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

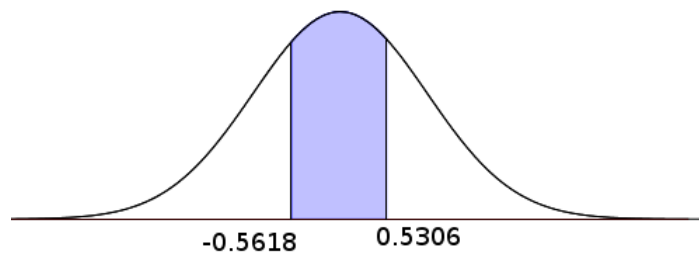


Figura 4.51: Curva Normal de las valoraciones de los Expertos en Mininet-WiFi a $N(0, 1)$

Según los valores de medida central para el conjunto de datos de la intersección en Mininet-WiFi, la probabilidad de estas curvas será la siguiente:

$$P(4,22 \leq x \leq 4,76) = P(-0,5618 \leq z \leq 0,5306) = 0,42$$

Esta probabilidad indica que existe un 42% de que cualquier valor escogido para esa distribución, este dentro de ese intervalo.

Se determinó la $\mu_{\text{corregida}}$ de la misma manera como se hizo con OpenNet, iterando los valores de tendencia central de la tabla 4.8 utilizando la ecuación 4.10.

Luego se procedió a determinar el promedio de los resultados de las iteraciones, permitiendo con esto obtener un valor estimado de la $\mu_{\text{corregida}}$, la cual sirvió como entrada de la característica Usabilidad que determinó el índice de calidad de uso corregido empleando el modelo EPCU, el resultado obtenido fue el siguiente:

$$\mu_{\text{corregida}} = 4,5$$

Se puede observar en la figura 4.47, las variables de entrada y el nuevo valor del índice de calidad de uso generado de la interrelación de la Usabilidad y la UX para Mininet-WiFi, utilizando el mecanismo EPCU.

En el caso de Mininet-WiFi el valor cuantitativo obtenido de la ejecución del modelo EPCU que determina el índice de calidad de uso, obtuvo un valor de 100%.

Se puede apreciar que a pesar que la Usabilidad tuvo un valor un poco menor con este método, al relacionar las distintas categorías a través de las reglas de inferencia el índice fue igual que el que no se interrelaciona con la UX, pudiéndose obtener el mismo valor de índice de calidad de uso con distintos valores en una de las categorías.

4.4. MEDICIÓN Y EVALUACIÓN DE LA CALIDAD DEL PRODUCTO SOFTWARE Y DE LA EXPERIENCIA DE USUARIO

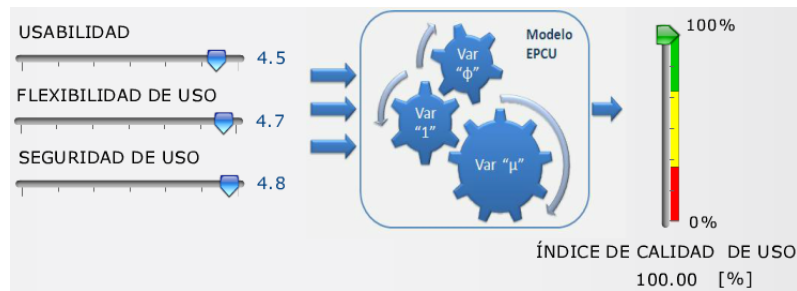


Figura 4.52: Determinación del índice de la calidad de uso de Mininet-WiFi a partir de la media corregida (con autorización de www.mepe.com.mx).

Fuente: <http://www.mepe.com.mx/mecanismos-por-categoria/>

La figura 4.53 muestra la comparación de las entradas de las dos evaluaciones de la calidad de uso de Mininet-WiFi.

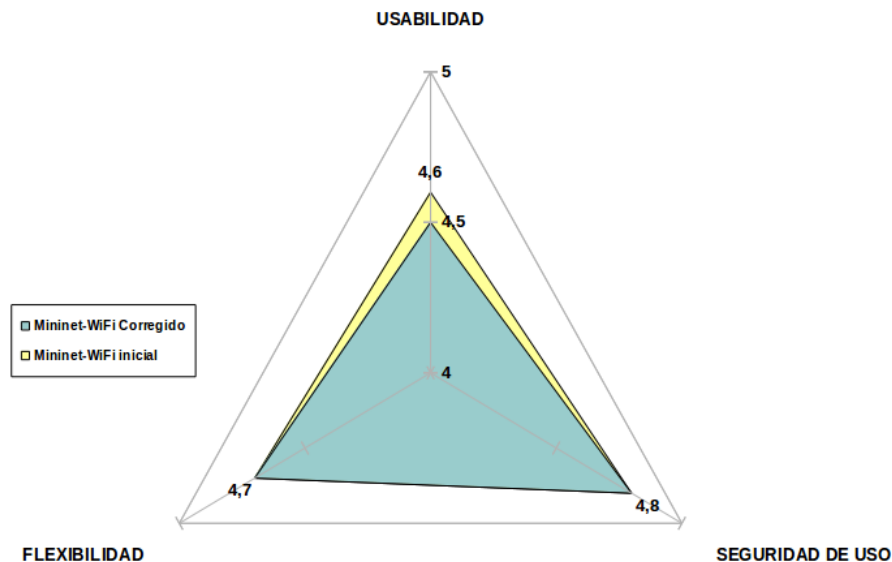


Figura 4.53: Comparación de las características de la calidad de uso en Mininet-WiFi

4.4.6. Índices de calidad del producto Software empleando el modelo EPCU

En la tabla 4.9 se muestran los resultados obtenidos del análisis comparativo de la calidad del producto software en las aplicaciones de emulación y/o simulación de redes inalámbricas definidas por software (SDWN), utilizando el modelo de estimación de proyectos en entornos de incertidumbre (EPCU), el cual se fundamenta en la norma ISO/IEC 25000. Se puede apreciar en dicha tabla que Mininet-WiFi obtuvo mejores índices de calidad que OpenNet.

Tabla 4.9: Índice de la Calidad del producto Software basado en la norma ISO/IEC 25000

Índice de Calidad	OpenNet	Mininet-WiFi
Calidad Interna/Externa (Construcción)	57,34%	84,38%
Calidad de Uso (Experto)	96,83%	100,00%
Calidad de Uso (UX-SUS)	85,98%	100,00%

4.5. Discusión de Resultados

En el estudio actual, la comparación entre OpenNet y Mininet-WiFi se muestra de forma cuantificable, permitiendo esto aclarar los resultados que Chan et al. (2014) y Fontes et al. (2015) asumen sobre el software de mejores prestaciones para la simulación/emulación de SDWN, en el cual quedaban una serie de interrogantes que hasta ahora la literatura científica referida a estos software no había sido capaz de aclarar, para ser más específico lo referente a la forma de determinar de forma confiable cual era el software de mejor calidad para la emulación de las redes inalámbricas definidas por software.

Este trabajo permitió verificar de forma medible, cual es el mejor software de los analizados en términos de calidad, esto se determinó apoyándonos en el modelo EPCU el cual está fundamentado en la norma ISO/IEC 25000 referida a la calidad del producto software.

A continuación citaremos algunos aspectos que se generaron en esta investigación:

- Una de las características que la norma define es la portabilidad, donde la facilidad de instalar el producto es muy importante. En este caso se pudo verificar que aunque OpenNet ha mejorado considerablemente este aspecto al automatizar su proceso de instalación por medio de Ansible aún les queda mucho por hacer. Mininet-WiFi permite una variedad opciones para el uso del programa, y en el caso de la instalación localmente la misma es más sencilla que la de OpenNet.
- Al realizar pruebas de rendimiento y uso se comprobó que Mininet-WiFi si emula los procesos inalámbricos, en contra parte OpenNet se basa en la simulación de los mismos.
- Al proceder a realizar la medición y evaluación de la calidad interna y externa de ambos software empleando para ello el mecanismo EVA20131206SQRINT el cual se basa en el modelo EPCU, Mininet-WiFi arrojó resultados altos en lo relacionado a los valores del índice de calidad interna/externa con un 84,38 %. En contraste OpenNet su índice de calidad fue de 57,34 %.
- En lo concerniente a la evaluación de la calidad de uso se empleó el mecanismo EVA20131206SQREXT, el índice de calidad de uso de OpenNet acá fue de 96,83 %, y el de Mininet-WiFi de 100 %, ambos software poseen un alto índice de calidad de uso.

La calidad de la interacción influye de manera decisiva en que los usuarios se decanten por un software u otro.

- Supeditados a los resultados anteriores referentes a los índices de calidad de uso, se procedió a medir la UX en ambos software con la finalidad de interrelacionarlo con

la usabilidad y generar una nueva entrada de la misma para emplearlo en el modelo EPCU.

- Los resultados de este estudio indican que con la interrelación entre la Usabilidad y la UX, se pueden llegar a resultados un poco mas aceptables, esto debido a que en el análisis inicial la valoración de uso la dan personas las cuales conocen perfectamente el software, son usuarios expertos que están acostumbrados al uso del mismo. Con la UX el usuario evalúa en función a la facilidad y agrado hacia el software entre otros parámetros. El resultado más sorprendente que se observo en este estudio es que el puntaje SUS de UX para OpenNet fue extremadamente bajo, esto origino un menor índice de calidad de uso en la interrelación de la UX y la Usabilidad, donde su valor fue de 85,98% en contraste con la primera evaluación que fue de 96,83%. Mininet-WiFi obtuvo un resultado catalogado como bueno y cuantitativamente obtuvo el mismo valor del 100% en las dos evaluaciones del índice de calidad de uso empleando el mecanismo EPCU.
- Se comprueba que el modelo EPCU permite determinar la calidad del producto software de forma cuantitativa a partir de datos cualitativos, de forma confiable y accesible a cualquier usuario.

Capítulo 5

Conclusiones y Recomendaciones

Conclusiones

En esta tesis se evaluó cuantitativamente la calidad interna/externa y de uso del producto software en las aplicaciones de emulación y/o simulación de redes inalámbricas definidas por software OpenNet y Mininet-WiFi, fundamentado en las especificaciones de la norma ISO/IEC 25000 y apoyado en el modelo EPCU.

La ISO/IEC 25020 que será la encargada de definir las métricas de calidad del producto software y que todavía está en pendiente de publicación, por lo que no existe un acuerdo respecto a los indicadores y umbrales que se deben considerar para poder determinar la calidad de un producto software de manera estandarizada. A través del modelo EPCU fue posible determinar un valor cuantitativo consistente, de manera formal permitiendo hacer comparaciones acerca de la calidad del producto de software de una manera práctica, tomando como referencia la valoración de los expertos.

Se comparó el funcionamiento y desempeño de los emuladores de redes inalámbricas definidas por software OpenNet y Mininet-WiFi. Esto permitió verificar el comportamiento y recursos necesarios para poder ejecutarlos. De igual manera se efectuaron pruebas de rendimiento de diferentes topologías en ambos software, esto permitió visualizar el comportamiento a hora de emular o simular los procesos.

Se desarrolló e implementó un instrumento de medición adecuado, el mismo permitió conocer la valoración de expertos en referencia al uso, configuración y alcance de cada software, fundamentado o soportado en la familia de normas ISO/IEC 25000 el cual sirvió para determinar la calidad del producto software a medir. Vale resaltar que estos expertos han producido o están en el proceso de crear información de carácter científica sobre ambos software, lo que indica que son usuarios avanzados en el manejo de estos programas.

El modelo EPCU permitió evaluar a partir del juicio de expertos la calidad del producto software de OpenNet y Mininet-WiFi, mediante mecanismos formales de manejo de información imperfecta y con base en el estándar de calidad ISO/IEC 25000. Siendo la valoración del experto un dato cualitativo, donde el modelo EPCU basado en lógica difusa facilitó evaluar la calidad interna/externa y de uso de OpenNet y Mininet-WiFi de forma cuantitativa.

Se determinó la escala para la usabilidad de los sistemas como el método de valoración para medir la experiencia de usuario (UX) de OpenNet y Mininet-WiFi, esto con la finalidad de interrelacionar la Usabilidad de la calidad de uso con la UX, para luego emplear ese resultado como una entrada de datos en el modelo EPCU. La relevancia de la UX está claramente apoyada por los resultados obtenidos. La usabilidad es entendida actualmente como facilidad de uso de un producto, siendo muy limitante esta visión, por ello se emplea la caracterización de la experiencia del usuario a través de la calidad de uso, lo que origina un concepto más amplio al integrar la calidad de uso con la UX, incluyendo no sólo aspectos de operatividad sino aquellos que están relacionados al soporte, entre otros, estando la usabilidad realmente centrada en el usuario.

Finalmente, los resultados de esta investigación demuestran en lo que respecta a calidad del producto software, que Mininet-WiFi es el software que presento mejor índice de calidad en el uso de los mecanismos del modelo EPCU, del mismo modo la valoración de la UX tuvo valores catalogados en el porcentaje de usabilidad como buenos, así como en la interrelación de la UX y la usabilidad se obtuvieron resultados semejantes con el modelo EPCU. En contraste los resultados de OpenNet no estuvieron al mismo nivel que los de Mininet-WiFi, siendo más bajos en todas las evaluaciones realizadas.

Se evidencia la facilidad y el rendimiento de trabajar con las herramientas de emulación de SDWN empleadas, permitiendo de igual forma la estructuración de distintas topologías de SDWN.

Se comprueba que los programas estudiados, permiten interactuar y ejecutar cualquier proceso como si se ejecutara directamente en un equipo con GNU/Linux de forma eficaz.

Se analiza el comportamiento de OpenNet y de Mininet-WiFi y se comprueba si es posible diseñar e implementar distintos escenarios virtuales, recreando algunas topologías inalámbricas.

Recomendaciones

Es posible fijar estrategias relevantes y factibles que sirvan para incrementar el índice de calidad de un software, realizando acciones que permitan mejorar las categorías de calidad que brinda la norma ISO/IEC 25000. Esto se debe a que el índice de calidad es consecuencia de los distintos valores de las categorías definidas por ese estándar.

Sería importante reexaminar y/o mejorar los siguientes aspectos de esta investigación:

- El cuestionario de evaluación de la valoración de los expertos, en especial los ítems relacionados con la flexibilidad y seguridad de uso que sirven para determinar la calidad de uso.
- Aumentar la población y la muestra de usuarios expertos en el uso de los programas, para obtener el juicio de mas expertos, podrá dar mejores resultados.
- Adiestrar durante más tiempo a los usuarios encargados de evaluar la experiencia de usuario puede equilibrar la diferencia que existe entre los dos software.

Sería recomendable que futuras investigaciones abordasen los siguientes temas:

- La estimación del tamaño funcional de OpenNet y de Mininet-WiFi.
- Para la valoración de la UX emplear otras técnicas o test existentes para determinarla como son las mediciones de componentes emocionales, determinación de zonas de calor, entre otros. De igual forma se puede realizar una inspección o evaluación sin usuarios llevadas a cabo por expertos o profesionales de la usabilidad.

Bibliografía

- Arias, F. G. (2012). *El Proyecto de Investigación. Introducción a la metodología científica*. 5ta. Fidas G. Arias Odón.
- Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123.
- Barona, L. (2013). Propuesta de escenarios virtuales con la herramienta vnx para pruebas del protocolo openflow. Tesis de maestría, ingeniería de redes y servicios telemáticos, Universidad Politécnica de Madrid, España.
- Brooke, J. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., and Shenker, S. (2007). Ethane: taking control of the enterprise. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 1–12. ACM.
- Castro, L. (2012). Introducción a la probabilidad y la estadística. Recuperado de: <http://www.fic.umich.mx/lcastro/5normal.pdf>.
- Centeno, A. G., Vergel, C. M. R., Calderón, C. A., and Bondarenko, F. C. C. (2014). Controladores sdn, elementos para su selección y evaluación. *Revista Telem@tica*, 13(3):10–20.
- Chan, M. (2014). Introduction to opennet. Recuperado de: <http://www.slideshare.net/rascov/20140824-open-net>.
- Chan, M.-C., Chen, C., Huang, J.-X., Kuo, T., Yen, L.-H., and Tseng, C.-C. (2014). Opennet: A simulator for software-defined wireless local area network. In *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pages 3332–3336. IEEE.
- Cortés, A. F. (2000). Algunos cuestionarios conocidos de usabilidad. Recuperado de: <http://www.sidar.org/recur/desdi/traduc/es/visitable/nuevos/CuestCon.htm>.
- Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334.

- D'Hertefelt, S. (2000). Emerging and future usability challenges: designing user experiences and user communities. Recuperado de: <http://users.skynet.be/fa250900/future/vision20000202shd.htm>.
- Duarte, G. and Lobo, R. (2015). Emulación de escenarios virtuales, en una sdwlan (software defined wireless local area network), de un campus universitario. *Ingeniería al Día*, 1(2).
- Erickson, D. (2011). Open networking foundation formed to speed network innovation. Recuperado de: <http://archive.openflow.org/wp/2011/03/open-networking-foundation-formed-to-speed-network-innovation/>.
- Escobar-Pérez, J. and Cuervo-Martínez, A. (2008). Validez de contenido y juicio de expertos: una aproximación a su utilización. *Avances en medición*, 6:27–36.
- Fontes, R. (2013). Visual network description (vnd). Recuperado de: <http://www.ramonfontes.com/visual-network-description/>.
- Fontes, R. (2015). Mininet-wifi - emulator for software-defined wireless networks. Recuperado de: <https://github.com/intrig-unicamp/mininet-wifi>.
- Fontes, R., Afzal, S., Brito, S., Santos, M., and Esteve Rothenberg, C. (2015). Mininet-WiFi: emulating Software-Defined wireless networks. In *2nd International Workshop on Management of SDN and NFV Systems / DMTF Datacenter and Cloud Management Workshop, ManSDN/NFV-DCM 2015*, Barcelona, Spain.
- Fontes, R., Oliveira, A., Pinheiro, T., Sampaio, P., and Figueira, R. (2014). Authoring of openflow networks with visual network description (sdn version). In *46th Summer Computer Simulation Conference*.
- Frost and Sullivan (2014). Software defined networking: en busca de la automatización de la red. White paper, Frost and Sullivan-Cisco.
- Gliem, R. R. and Gliem, J. A. (2003). Calculating, interpreting, and reporting cronbach's alpha reliability coefficient for likert-type scales. Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education.
- González-Sánchez, J. L., Montero-Simarro, F., and Gutiérrez-Vela, F.-L. (2012). Evolución del concepto de usabilidad como indicador de calidad del software. *El Profesional de la información*, 21(5):529–536.
- Hassan Montero, Y. and Martín Fernández, F. J. (2005). La experiencia del usuario. *No Solo Usabilidad*, (4).
- Hernández Sampieri, R., Fernández Collado, C., and Baptista Lucio, P. (2014). *Metodología de la investigación*. 6ta edition.

- HP-Networking (2012). Openflow: habilitación de la tecnología para redes definidas por software. Recuperado de: <http://h17007.www1.hp.com/ve/es/solutions/technology/openflow/index.aspx>.
- ISO25000.com (2014). Portal iso 25000. Recuperado de: <http://iso25000.com/index.php/normas-iso-25000>.
- ISO/IEC-25010 (2007). Iso/iec 25010:2007. software engineering. software product quality requirements and evaluation (square).
- Kambrica (2014). Ui, ux, ixd: ¿cuál es la diferencia? Recuperado de: <http://www.kambrica.com/blog/ui-ux-ixd-cual-es-la-diferencia/>.
- Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, page 19. ACM.
- Ledesma, R., Molina, G., and Valero, P. (2002). Análisis de consistencia interna mediante alfa de cronbach: un programa basado en gráficos dinámicos. *Psico-USF*, 7(2):143–152.
- Marín, J. (2005). Estadística descriptiva y análisis de datos. Recuperado de: <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/EDescrip/EstDescript.html>.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.
- Mendenhall, W., Beaver, R., and Beaver, B. (2010). *Introducción a la probabilidad y estadística*. 13 edition.
- Nielsen, J. (2000). Why you only need to test with 5 users. Recuperado de: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- Ortega Santamaría, S. (2011). *Introducción a la usabilidad y su evaluación*, chapter 1. FUOC. Fundació per a la Universitat Oberta de Catalunya, 1 edition.
- Park, R. and Baack, E. (2012). Despliegue y evaluación de desempeño de una red openflow. *Reporte Técnico LRAV 11112*.
- Peñalva, M. d. C. (2014). Un modelo de evaluación de la calidad de aplicaciones web en e-government. Master's thesis, Facultad de Informática.
- Piedrahita, S. (2007). Construcción de una herramienta para evaluar la calidad de un producto software.

- Romero, J. (2011). Maximizando la experiencia de usuario. Recuperado de: <http://www.influyeerp.com/maximizando-la-experiencia-de-usuario>.
- Sauro, J. (2011). *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC.
- Sun, W., Li, H., and Wu, J. (2015). Analysis on applying sdn in wireless networks from operation view.
- Suresh, L., Schulz-Zander, J., Merz, R., Feldmann, A., and Vazao, T. (2012). Towards programmable enterprise wlans with odin. In *Proceedings of the first workshop on Hot topics in software defined networks*, pages 115–120. ACM.
- Tauber, L. M. (2001). *La construcción del significado de la distribución normal a partir de actividades de análisis de datos*. PhD thesis, Universidad de Sevilla.
- Tseng, C.-C., Yen, L.-H., Chang, H.-H., and Hsu, K.-C. (2005). Topology-aided cross-layer fast handoff designs for ieee 802.11/mobile ip environments. *Communications Magazine, IEEE*, 43(12):156–163.
- Uceda, S. (2012). Un nuevo código de circulación para internet. Recuperado de: <http://cnmcblog.es/2012/06/22/un-nuevo-codigo-de-circulacion-para-internet>.
- Valdés, F. (2013). Midiendo la calidad del software. Recuperado de: <http://sg.com.mx/revista/40/midiendo-la-calidad-del-software>.
- Valdés, F. and Abran, A. (2007). Industry case studies of estimation models using fuzzy sets. *Proceedings of the IWSM-Mensura*, page 87.
- Valdés, F. and Abran, A. (2010). Comparing the estimation performance of the epcu model with the expert judgment estimation approach using data from industry. In *Software Engineering Research, Management and Applications 2010*, pages 227–240. Springer.
- Valdés, F. (2011). *Design of a Fuzzy Logic Software Estimation Process*. PhD thesis, École de Technologie Supérieure, Université du Québec, Montreal.
- Valdés, F. (2012). *Design Of A Fuzzy Logic Estimation Process For Software Projects: Estimation of Projects in a Context of Uncertainty EPCU Model*. LAP Lambert Academic Publishing.
- Valdés, F. (2014). Mecanismos de estimación de proyectos y evaluación. Recuperado de: <http://www.mepe.com.mx/>.
- Xinguard (2014). Openflow principle. Recuperado de: <http://www.xinguard.com/en/content.aspx?id=70>.
- Zapata, M. (2011). Métodos de evaluación sin usuarios. *Universitat Oberta de Catalunya*.

Apéndice A

Glosario de Términos

Nomenclatura

Contenedor	Un contenedor es simplemente un proceso para el sistema operativo que, internamente, contiene la aplicación que queremos ejecutar y todas sus dependencias. La aplicación contenida solamente tiene visibilidad sobre el sistema de ficheros virtual del contenedor y utiliza indirectamente el kernel del sistema operativo principal para ejecutarse.
Emulación	Emular significa imitar las acciones de otro, procurando igualarlas e incluso excederlas (Diccionario de la Real Academia Española, 23a edición). Esto significa que un emulador reproduce todos los procesos internos que realizaría el sistema nativo, y busca cubrir todos los comportamientos respectivos implementando los mismos mecanismos.
Fork	Una bifurcación (fork en inglés), en el ámbito del desarrollo de software, es la creación de un proyecto en una dirección distinta de la principal u oficial tomando el código fuente del proyecto ya existente. Comúnmente se utiliza el término inglés. Como resultado de la bifurcación se pueden llegar a generar proyectos diferentes que cubren necesidades distintas aunque similares.
GitHub	GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.
Jitter	Se define técnicamente como la variación en el tiempo en la llegada de los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino. Es un efecto de las redes de datos no orientadas a conexión y basadas en conmutación de paquetes. Como la información se discretiza en paquetes cada uno de los paquetes puede seguir una ruta distinta para llegar al destino.

Máquina Virtual	En informática una máquina virtual es un software que simula a una computadora y puede ejecutar programas como si fuese una computadora real. Este software en un principio fue definido como un duplicado eficiente y aislado de una máquina física.
Norma	Norma es un término que proviene del latín y significa “escuadra”. Una norma es una regla que debe ser respetada y que permite ajustar ciertas conductas o actividades. En el ámbito del derecho, una norma es un precepto jurídico.
Simulación	Se puede definirse como la experimentación con un modelo que imita ciertos aspectos de la realidad. Esto permite trabajar en condiciones similares a las reales, pero con variables controladas y en un entorno que se asemeja al real pero que está creado o acondicionado artificialmente. La idea es que la simulación permita comprobar el comportamiento de una persona, objeto o de un sistema en ciertos contextos que, si bien no son idénticos a los reales, ofrecen el mayor parecido posible. Así, es posible corregir fallos antes de que la experiencia, efectivamente, se concrete en el plano de lo real.
SSH	Secure SHell, es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente.
Throuhgput	Caudal o tasa neta de información de usuario transmitida por una red de telecomunicación digital.
WLAN	Una red de área local inalámbrica (WLAN) es una red que cubre un área equivalente a la red local de una empresa, con un alcance aproximado de cien metros. Permite que las terminales que se encuentran dentro del área de cobertura puedan conectarse entre sí.

Apéndice B

Instalación y Configuración de OpenNet y Mininet-WiFi

B.1. Instalación y configuración de OpenNet.

Se instaló y configuró OpenNet siguiendo los pasos que señala la página del proyecto alojada en la plataforma de desarrollo colaborativo de códigos GitHub:

<https://github.com/dlinknctu/OpenNet>.

Tener alojado el proyecto en GitHub presenta una serie de ventajas incluyendo el control de versiones y la posibilidad de que otros desarrolladores participen en el mismo, ofreciendo herramientas muy útiles para el trabajo en equipo, dentro de un proyecto, por solo citar algunas.

Se debe resaltar que desde la última versión publicada por los desarrolladores la instalación y configuración de OpenNet se facilita bastante, en versiones anteriores era un proceso tedioso y engorroso. En fecha reciente se le realizó una actualización importante ya que el proceso de instalación y configuración se encuentra automatizado por medio de Ansible.

Se puede decir que Ansible es una herramienta Open Source escrita en Python que permite gestionar configuraciones de equipos por medio de tareas, realizando un despliegue automático de las aplicaciones a instalar, así como muchas otras tareas de forma limpia y transparente al usuario final. Hay que destacar que Ansible no necesita instalar un agente en cada una de las máquinas que queremos configurar, al contrario de otras herramientas de automatización de procesos como Chef y Puppet. Ansible para la ejecución de tareas controla los equipos a través de SSH.

En la figura B.1 se aprecia la línea de tiempo del proyecto OpenNet en GitHub, donde se detalla la fecha donde se introdujo Ansible para el proceso de automatización de la instalación.

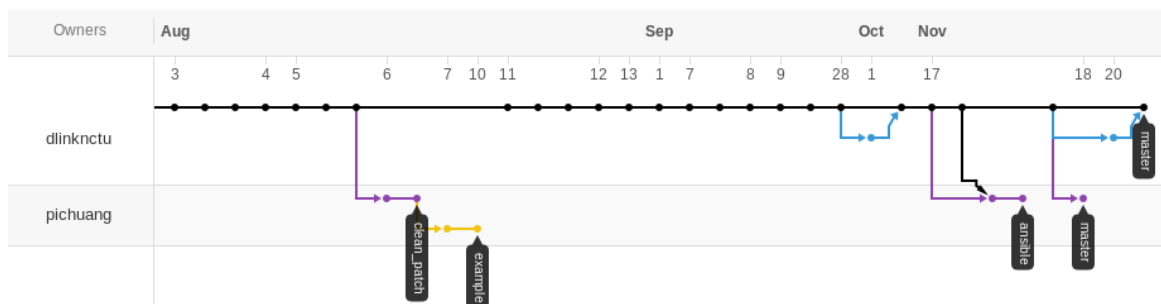


Figura B.1: Línea de tiempo de los cambios hechos en OpenNet

Fuente: *<https://github.com/dlinknctu/OpenNet>.*

Para comenzar a instalar OpenNet se ejecutó en un terminal del servidor donde fue instalado, una serie de comandos los cuales se muestran en la figura B.2.

B.1. INSTALACIÓN Y CONFIGURACIÓN DE OPENNET.

```
$ sudo su -
# apt-get install git ssh
# git clone https://github.com/dlinknctu/OpenNet.git
# cd OpenNet
# ./configure.sh
# ./install.sh master
```

Figura B.2: Comandos para instalar OpenNet
Fuente: <https://github.com/dlinknctu/OpenNet>

Al terminar de ejecutarse la instalación, antes de comenzar a utilizar OpenNet, fue necesario arrancar un controlador ya que OpenNet no posee un controlador precargado, se decidió usar el controlador POX, ejecutando el comando que se muestra en Figura B.3. POX es un controlador SDN, desarrollado en Python que permite programar y controlar switches OpenFlow de forma sencilla. POX es un controlador muy atrayente a nivel académico y de simulación por la facilidad de modificar su código.

```
gerlyn@gerlyn-mininet:~/OpenNet-master/pox$ sudo python pox.py forwarding.l2_learning
sudo: imposible resolver el anfitrión gerlyn-mininet
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
INFO:core:POX 0.2.0 (carp) is up.
INFO:openflow.of_01:[00-00-00-00-00-07 1] connected
INFO:openflow.of_01:[00-00-00-00-00-06 2] connected
INFO:openflow.of_01:[00-00-00-00-00-01 4] connected
INFO:openflow.of_01:[00-00-00-00-00-05 6] connected
INFO:openflow.of_01:[00-00-00-00-00-03 5] connected
INFO:openflow.of_01:[00-00-00-00-00-04 3] connected
INFO:openflow.of_01:[00-00-00-00-00-02 7] connected
█
```

Figura B.3: Ejecución del Controlador POX

La figura B.4 muestra la serie de pasos finales para ejecutar OpenNet.

```
$ sudo su -
# cd OpenNet
# ./waf_shell.sh
# cd ../../mininet/examples/opennet
# python wifiroaming.py
```

Figura B.4: Comandos para ejecutar OpenNet
Fuente: <https://github.com/dlinknctu/OpenNet>

La figura B.5 muestra el arranque de OpenNet y la interconexión de los dispositivos de la topología, en este caso la configuración de la topología se realiza en el archivo wifiroaming.py.

B.1. INSTALACIÓN Y CONFIGURACIÓN DE OPENNET.

```
root@gerlyn-mininet:/home/gerlyn/OpenNet-master/mininet/examples/opennet# python wifiroaming.py
*** Adding controller
Wifi pcap path: /tmp/pcap/wifi
CSMA pcap path:/tmp/pcap/csma
Netanim path: /tmp/xml/wifi-wired-bridged4.xml
*** Starting network
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
Testing network connectivity
*** h1 : ('ping 10.10.10.2 -c 3',)
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=64 time=299 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=64 time=19.8 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=64 time=10.9 ms

--- 10.10.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 10.912/110.243/299.959/134.199 ms
*** Starting CLI:
opennet> █
```

Figura B.5: Ejecución de OpenNet

Antes de ejecutar OpenNet se requirió la ejecución de NetAnim, lo cual se realizó ejecutando la serie de comandos en la consola, como se muestra en la figura B.6.

```
$ sudo su -
# cd OpenNet/ns-allinone-$NS3_VERSION/netanim-$NETANIM_VERSION
# ./NetAnim
```

Figura B.6: Comandos para ejecutar NetAnim
Fuente: <https://github.com/dlinknctu/OpenNet>

Con esa serie de pasos quedo instalado OpenNet, al igual que NetAnim el cual es el visualizador gráfico del simulador ns-3, la figura B.7 muestra la pantalla inicial de NetAnim sin cargar la simulación hecha con OpenNet.

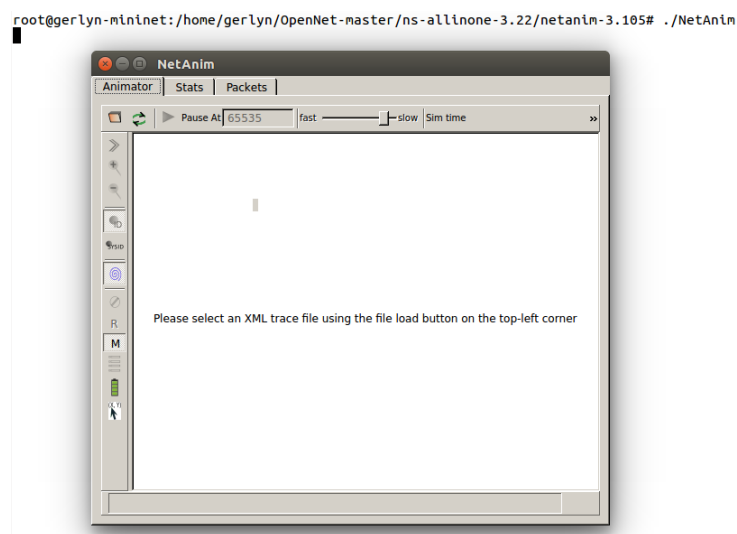


Figura B.7: Ejecución de NetAnim

B.1. INSTALACIÓN Y CONFIGURACIÓN DE OPENNET.

Un inconveniente que se presentó al instalar OpenNet en el equipo de 32 bits, fue el hecho que la tarea creada en Ansible para automatizar la instalación de los paquetes necesarios no permitía instalar OVS. Esto era consecuencia de que dicha tarea estaba especificada en el archivo de configuración para que solo se realicen instalaciones en equipos de 64 bits, generando un error que no permitía la instalación, por lo cual se editó dicha tarea encontrándose la misma en la siguiente ruta: **ansible/roles/openvswitch/tasks/main.yml** para poder instalar OVS. En esta tarea se sustituyó en todas las líneas del script donde dice **amd64.deb** por **i386.deb**, esto permitió descargar e instalar el OVS para 32 bits. Se debe resaltar que en ninguna parte los desarrolladores referencian ese detalle, ni en foros ni listas de discusión, por lo cual la instalación actual solo permite hacerse en equipos de 64 bits. Se debe descargar el archivo Open-Master.zip del repositorio en GitHub y hacer la ejecución del archivo install.sh localmente. La figura B.8 muestra la parte del script donde se debe hacer la modificación.

```
25   with_items:
26     - openvswitch-common_{{ OVS_VERSION }}-1_amd64.deb
27     - openvswitch-switch_{{ OVS_VERSION }}-1_amd64.deb
28     - openvswitch-pki_{{ OVS_VERSION }}-1_all.deb
29     - python-openvswitch_{{ OVS_VERSION }}-1_all.deb
30   args:
31     chdir: "{{ temp_location }}"
32     notify: Restart OpenvSwitch daemon
33     tags: openvswitch
34   - name: Start OpenvSwitch Daemon
35     service: name=openvswitch-switch state=started enabled=yes
36     tags: openvswitch
37   - name: Check OpenvSwitch Database Version
38     command: "ovs-vsctl --version"
39     register: ovsdb_version
40     tags: openvswitch
41   - debug: msg="OpenvSwitch Database Version {{ ovsdb_version.stdout }}"
42     tags: openvswitch
43   - name: Remove OpenvSwitch source code
44     file:
45       path: "{{ temp_location }}/{{ item }}"
46       state: absent
47   with_items:
48     - openvswitch-{{ OVS_VERSION }}
49     - openvswitch-{{ OVS_VERSION }}.tar.gz
50     - openvswitch-switch_{{ OVS_VERSION }}-1_amd64.deb
51     - openvswitch-common_{{ OVS_VERSION }}-1_amd64.deb
52     - openvswitch-datapath-dkms_{{ OVS_VERSION }}-1_all.deb
53     - openvswitch-datapath-source_{{ OVS_VERSION }}-1_all.deb
54     - openvswitch-dbg_{{ OVS_VERSION }}-1_amd64.deb
55     - openvswitch-ipsec_{{ OVS_VERSION }}-1_amd64.deb
56     - openvswitch-pki_{{ OVS_VERSION }}-1_all.deb
57     - openvswitch-test_{{ OVS_VERSION }}-1_all.deb
58     - openvswitch-testcontroller_{{ OVS_VERSION }}-1_amd64.deb
59     - openvswitch-vtep_{{ OVS_VERSION }}-1_amd64.deb
60     - python-openvswitch_{{ OVS_VERSION }}-1_all.deb
61   tags: openvswitch
```




Figura B.8: Script de la tarea que se debe modificar en Ansible

B.2. Instalación y configuración de Mininet-WiFi.

Al igual que OpenNet, el proyecto Mininet-WiFi se encuentra alojado en GitHub en el siguiente enlace:

<https://github.com/intrig-unicamp/mininet-wifi>

A diferencia del software anterior, Mininet-WiFi facilita otras fuentes alternativas para su instalación y uso, como son: la descarga de una imagen de una máquina virtual (VM=Virtual Machine) con Ubuntu 14.03.4 Trusty, con todos los paquetes requeridos para ejecutar Mininet-WiFi ya instalados y configurados, o la descarga de una imagen de un Docker Linux Containers.

Docker es un proyecto de código abierto con el que fácilmente se pueden crear contenedores personalizados, ligeros y portables que sirven de base para nuevas imágenes de sistemas operativos, aplicaciones o para poner directamente en marcha servicios. Estos contenedores de Docker se pueden definir como máquinas virtuales ligeras, menos exigentes con los procesadores y memorias de los equipos donde se ejecutarán. Es una virtualización a nivel del sistema operativo. En Docker se puede virtualizar un sistema operativo GNU/Linux con todas las aplicaciones que necesitemos. Los Docker Linux Container permiten ayudar a los desarrolladores y administradores de sistemas a presentar sus aplicaciones con todas sus dependencias conjuntamente, y conseguir que se ejecute a través de otros sistemas y máquinas, sin ningún problema.

Se probaron las tres opciones de instalación y uso de Mininet-WiFi, arrojando todos resultados similares en cuanto a rendimiento y usabilidad. Para efectos de esta investigación se decidió trabajar con la opción existente en GitHub para que el análisis comparativo estuviese en igualdad de condiciones, no por ello, los otros dos métodos que permiten usar Mininet-WiFi no dejan de ser una ventaja al compararlo con otros programas semejantes por su portabilidad.

Se procedió a realizar la instalación de Mininet-WiFi ejecutando la serie de comandos que se muestra en la figura B.9.

```
step 1: $ sudo apt-get install git
step 2: $ git clone https://github.com/intrig-unicamp/mininet-wifi
step 3: $ cd mininet-wifi
step 4: $ sudo util/install.sh -Wnfv
```

Figura B.9: Pasos para instalar Mininet-WiFi

Fuente: *<https://github.com/intrig-unicamp/mininet-wifi>*

Mininet-WiFi cuenta para su instalación con un script escrito en bash el `install.sh`, el cual generó la automatización de la instalación y configuración del mismo. Este script es una modificación del archivo de configuración de Mininet.

B.2. INSTALACIÓN Y CONFIGURACIÓN DE MININET-WIFI.

Luego para ejecutar Mininet-WiFi y correr un ejemplo de una topología inalámbrica simple compuesta por 2 estaciones y 1 access point y el controlador, se ejecutó los comandos de la figura B.10.

```
gerlyn @ gerlyn-bears ~
└─$ sudo mn --wifi --ssid=new_ssid
[sudo] password for gerlyn:
*** Creating network
*** Adding controller
*** Adding hosts and stations:
sta1 sta2
*** Adding switches and access point(s):
ap1
*** Adding links and associating station(s):
(sta1, ap1) (sta2, ap1)
*** Starting controller(s)
c0
*** Starting switches and access points
ap1 ...
*** Starting CLI:
mininet-wifi> █
```

Figura B.10: Ejecución de Mininet-WiFi

De la plataforma GitHub donde está alojado el proyecto Mininet-WiFi, se pudo obtener evidencia exacta que el mismo si es una modificación del proyecto Mininet, estando este último alojado de la misma forma en esta plataforma. Esto se puede deducir de la figura B.11 la cual muestra la aportación de códigos por parte de los desarrolladores al proyecto Mininet-WiFi. De esta figura se desprenden datos interesantes, la imagen muestra una gráfica superior de color verde que representa los aportes totales de código al proyecto Mininet-WiFi, así mismo las gráficas inferiores de color naranjas indican como ha sido la contribución individual de los desarrolladores, en este caso se puede visualizar el aporte de Bob Lanz el cual es uno de los creadores de Mininet. La fecha inicial que allí se visualiza es la del proyecto Mininet. Es bueno acotar que de forma alterna esta plataforma indica la fecha de inicio del proyecto Mininet-WiFi el cual esta referenciado a mayo de este año. Al unir las gráficas que se generan por las aportaciones individuales a nivel de código hechas al proyecto, en una especie de línea de tiempo se comprueba que el mismo es un valor exacto de la gráfica superior que representa al proyecto Mininet-WiFi.

B.2. INSTALACIÓN Y CONFIGURACIÓN DE MININET-WIFI.

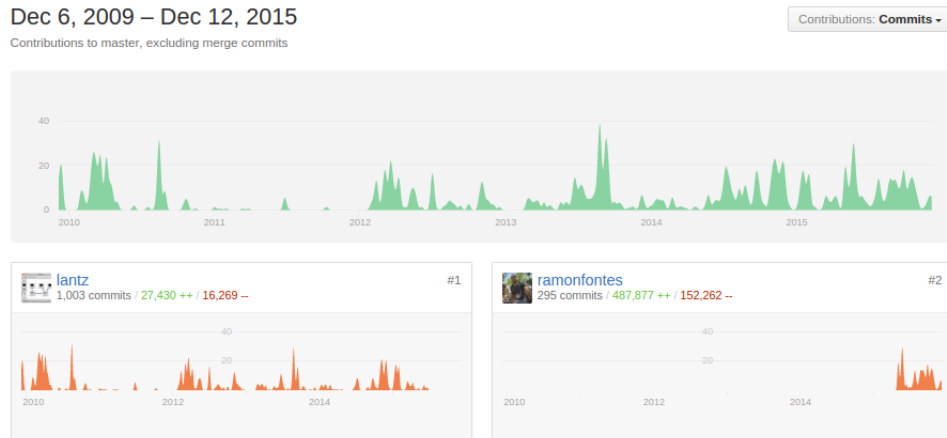


Figura B.11: Contribuciones de los desarrolladores a nivel de código al proyecto Mininet-WiFi

Fuente: <https://github.com/intrig-unicamp/mininet-wifi>

Apéndice C

Instrumento de Validación de la Calidad del Producto Software de las aplicaciones SDWN, basado en la ISO/IEC 25000

25. **Is it necessary a level of knowledge to execute the updates? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

26. **Does the software able an easy delete of an update? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

27. **Are there information included in the components documentation about what does each component, what parameters are needed and what data returns? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

28. **Are all the software components documented? (Functions, methods and/or classes) ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

29. **Does it possess proper update availability? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

30. **Does this software allow modifications that can include software corrections, improves or adaptation to changes in the environment and specifications of functionality requirements? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

EFFICIENCY (PERFORMANCE)

Please complete the following questionnaire indicating a value between 0-5.

59. **The non-refusal required events are processed using digital sign? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

60. **Does the system authenticate the identity of a user or a resource? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

61. **Does the software has mechanisms to allow the prevention of non-authorized access, accidental or deliberated to the products functionality? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

62. **Is it correct the implementation of data encryption according to the requirement specification? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

63. **Is the audit track complete in relation to user's system access and data? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

USABILITY

Please complete the following questionnaire indicating a value between 0-5.

The qualitative variable values are considered for three (Low, Average, High) corresponding to a range of 0-5, where the highest value is 5 and the lowest 0.

70. **Is the software delay on completing a task less than expected? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

71. **Could it be considered that this software allows a quick learning about its functionality to the inexperienced users? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

72. **Do you consider that this software adheres to the standards, conventions, style guides, regulations related to the usability? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

73. **Does this software allow the user to learn about its functionality? A really important consideration here is software's documentation. ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

74. **Does this software possesses the capability of being understood, learned, used and also is attractive to the user when used under specific conditions? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

FLEXIBILITY OF USE

Please complete the following questionnaire indicating a value between 0-5.

The qualitative variable values are considered for three (Low, Average, High) corresponding to a range of 0-5, where the highest value is 5 and the lowest 0.

75. **Can a physically handicap user completely access to the software functions? ***

Marca solo un óvalo.

0 1 2 3 4 5

Low High

76. **Is easy to adapt the software to the user's needs? ***

Marca solo un óvalo.

0 1 2 3 4 5

Low High

77. **Is it high the percentage of context in which the software could be used? ***

Marca solo un óvalo.

0 1 2 3 4 5

Low High

OPERATORS SECURITY AND HEALTH

Please complete the following questionnaire indicating a value between 0-5.

The qualitative variable values are considered for three (Low, Average, High) corresponding to a range of 0-5, where the highest value is 5 and the lowest 0.

78. **Is the environmental impact by the systems elaboration and use minimum and/or disposable? ***

Marca solo un óvalo.

0 1 2 3 4 5

Low High

79. **Is it low and/or disposable the incidence of risk for third parts (people) affected by the systems use? ***

Marca solo un óvalo.

0 1 2 3 4 5

Low High

80. **Is the impact on health and security on product users by its use minimum and/or disposable? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

81. **Does the environmental impact on the development and use of the system is minimal and/or disposable? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

82. **Is the frequency of health problems between the product users minimum and/or disposable? ***

Marca solo un óvalo.

	0	1	2	3	4	5	
Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	High

Apéndice D

Instrumento de evaluación de la escala para la usabilidad de los sistemas (SUS)

Evaluación de la Experiencia de Usuario sobre programas de Redes Inalámbricas definidas por Software

A continuación les presento el siguiente cuestionario, el cual les agradezco de antemano su colaboración al responder.

El mismo se basa en determinar la Experiencia de Usuario del Software para redes inalámbricas definidas por software.

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta.

Atentamente.

Gerlyn Eduardo Duarte

***Obligatorio**

Cuestionario basado en la Escala de Usabilidad del Sistema (System Usability Scale, SUS)

Este cuestionario se basa en la escala de usabilidad del sistema (SUS por sus siglas en inglés), el mismo fue desarrollada por John Brooke en 1996 debido a una fuerte necesidad en la comunidad de usabilidad para obtener una herramienta que pueda recolectar rápida y fácilmente la calificación subjetiva de un usuario respecto con la facilidad de uso de un producto.

1. **Creo que me gustaría utilizar el sistema con frecuencia ***

Siendo 1 la nota más baja y 5 la más alta:

Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

2. **He encontrado el sistema innecesariamente complejo ***

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:

Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

3. Pensé que el sistema era fácil de usar *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

4. Creo que necesito el apoyo de una persona para poder utilizar el sistema *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

5. Me pareció que las diversas funciones en este sistema fueron bien integradas *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

6. Pensé que era demasiado inconsistente este sistema *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

7. Imagino que la mayoría de la gente aprende a utilizar este sistema muy rápidamente *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

8. Me pareció que el sistema es muy engorroso de usar *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

9. Me sentí muy seguro utilizando el sistema *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

10. Necesitaba aprender un montón de cosas antes de que pudiera usar este sistema *

Teniendo en cuenta que acaba de hacer uso de un software, responda por favor la siguiente encuesta; siendo 1 la nota más baja y 5 la más alta:
Marca solo un óvalo.

	1	2	3	4	5	
Bajo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alto

Apéndice E

Análisis y Resultados Estadísticos

Parámetros Estadísticos de OpenNet

GET DATA

/TYPE=TXT

/FILE="/home/gerlyn/Documentos/Tesis Gerlyn/Análisis

EstadísticoOpennet.csv"

/IMPORTCASES=ALL

/ARRANGEMENT=DELIMITED

/DELCASE=LINE

/FIRSTCASE=2

/DELIMITERS=","

/QUALIFIER=""

/VARIABLES=

FUNCIONABILIDAD F3.0

CONFIABILIDAD F3.0

MANTENIBILIDAD F3.0

EFICIENCIA F3.0

PORTABILIDAD F3.0

COMPATIBILIDAD F3.0

OPERATIVIDAD F1.0

SEGURIDAD F3.0

USABILIDAD F3.0

FLEXIBILIDAD F3.1

SEGURIDAD_DE_USO F3.0.

FRECUENCIES

FRECUENCIES

/VARIABLES= FUNCIONABILIDAD CONFIABILIDAD

MANTENIBILIDAD EFICIENCIA PORTABILIDAD COMPATIBILIDAD

OPERATIVIDAD SEGURIDAD USABILIDAD FLEXIBILIDAD

SEGURIDAD_DE_USO

/FORMAT=AVALUE TABLE

/STATISTICS=DEFAULT VARIANCE SKEWNESS MODE MEDIAN

/MISSING=INCLUDE

/HISTOGRAM=NORMAL.

FUNCIONABILIDAD

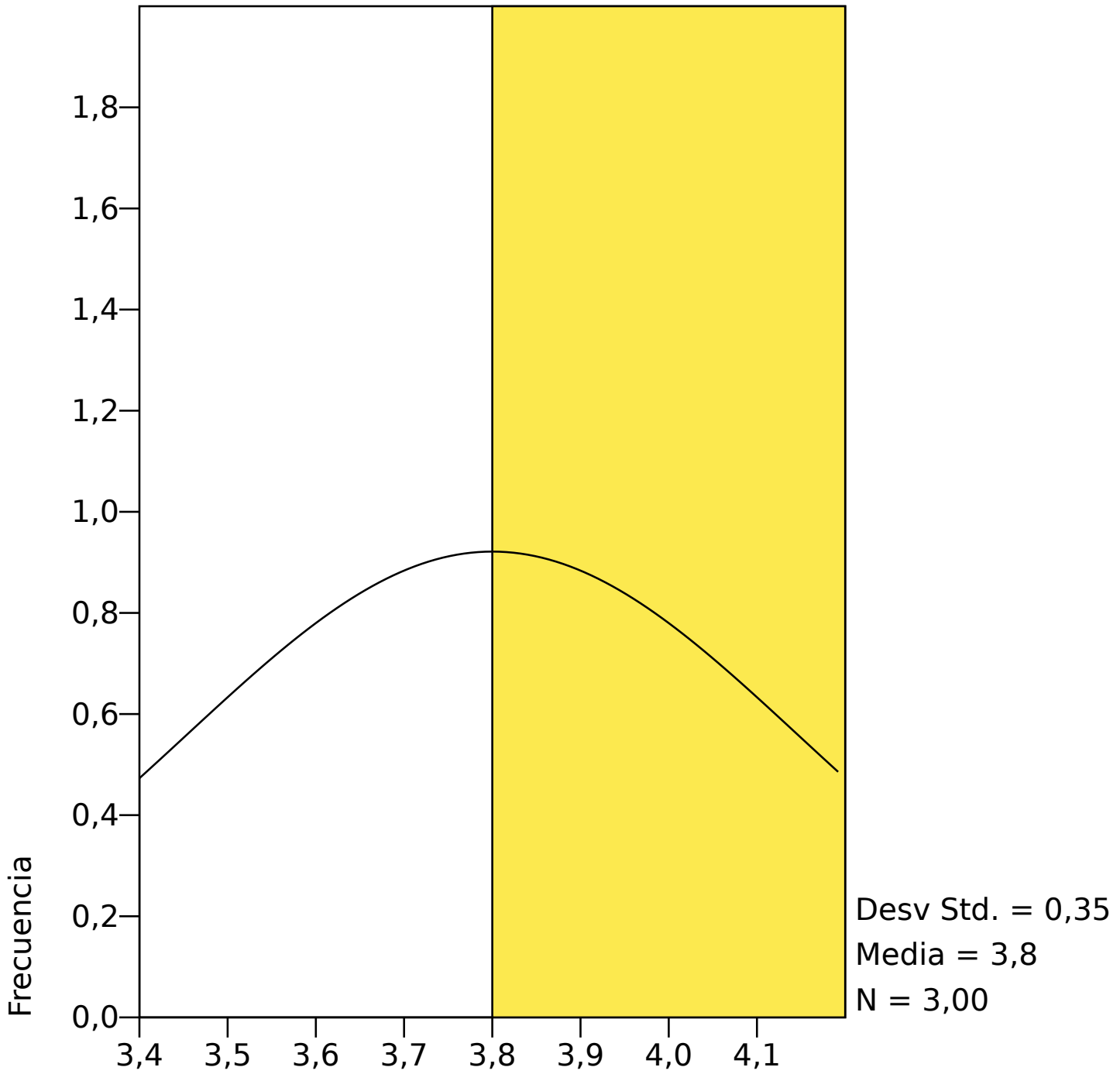
<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	3	1	6,67	33,33	33,33
	4	2	13,33	66,67	100,00
	.	12	80,00	Perdidos	

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	<i>Total</i>	15	100,0	100,0	

FUNCIONABILIDAD

<i>N</i>	<i>Válido</i>	3
	<i>Perdidos</i>	12
<i>Media</i>		3,80
<i>Modo</i>		4,00
<i>Desv Std</i>		,35
<i>Asimetría</i>		-1,73
<i>Mínimo</i>		3,40
<i>Máximo</i>		4,00
	50 (Mediana)	4

HISTOGRAM



FUNCIONABILIDAD

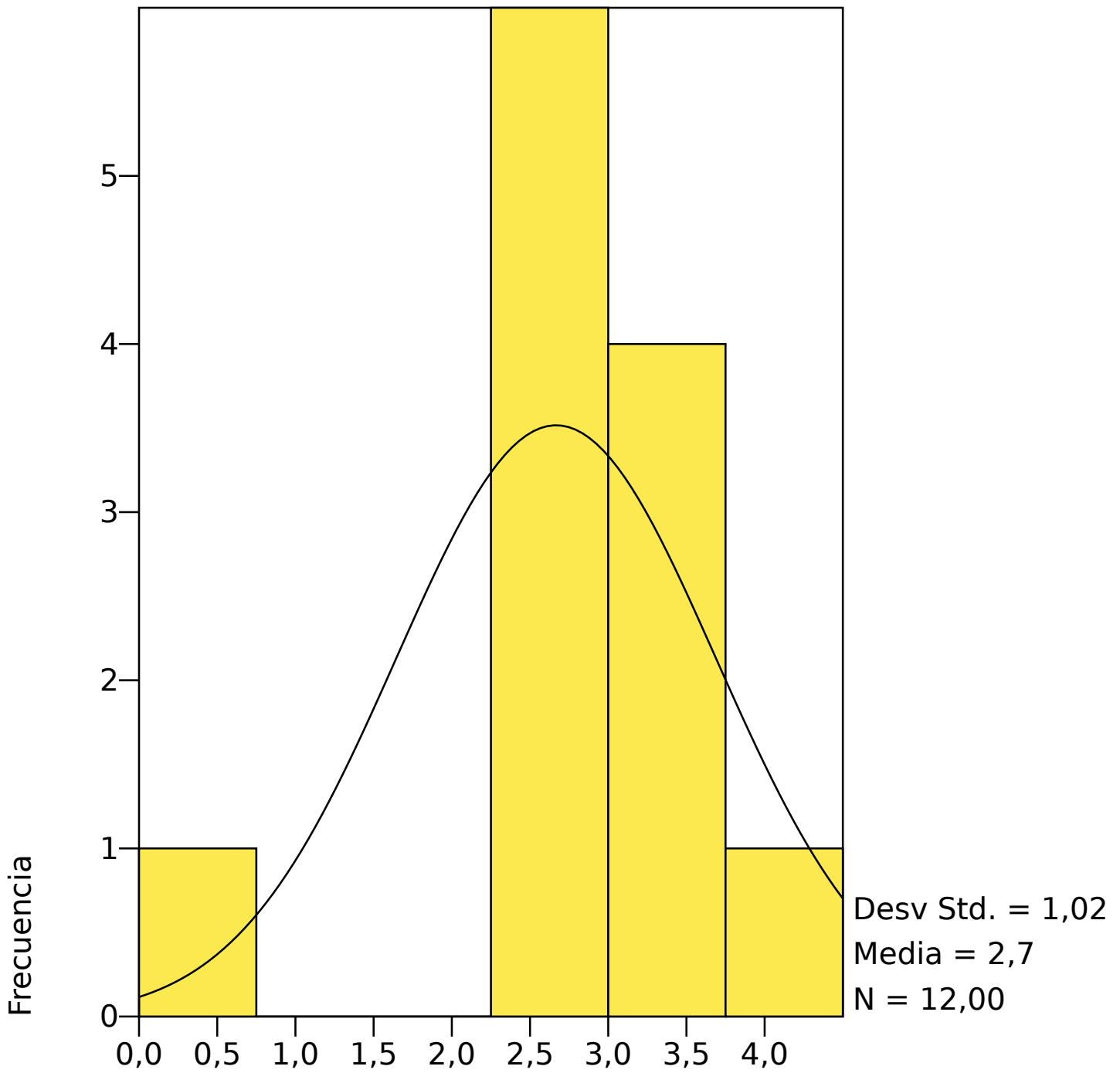
CONFIABILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	0	1	6,67	8,33	8,33
	2	6	40,00	50,00	58,33
	3	4	26,67	33,33	91,67
	4	1	6,67	8,33	100,00
	.	3	20,00	Perdidos	
<i>Total</i>		15	100,0	100,0	

CONFIABILIDAD

<i>N</i>	<i>Válido</i>	12
	<i>Perdidos</i>	3
<i>Media</i>		2,67
<i>Modo</i>		2,40
<i>Desv Std</i>		1,02
<i>Asimetría</i>		-1,55
<i>Mínimo</i>		,00
<i>Máximo</i>		4,00
	50 (Mediana)	2

HISTOGRAM



CONFIAIBILIDAD

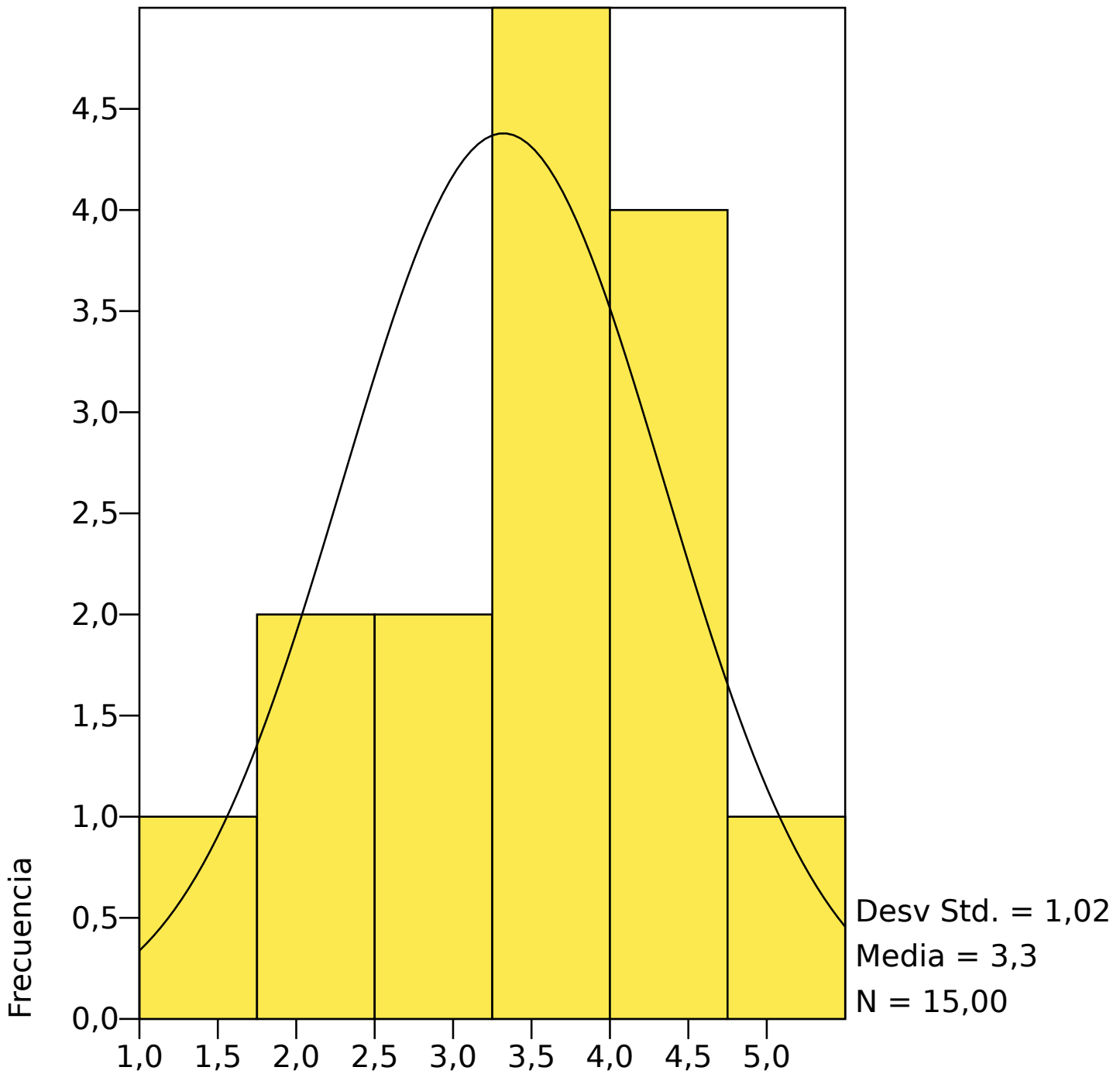
MANTENIBILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	1	1	6,67	6,67	6,67
	2	2	13,33	13,33	20,00
	3	2	13,33	13,33	33,33
	3	4	26,67	26,67	60,00
	4	1	6,67	6,67	66,67
	4	3	20,00	20,00	86,67
	4	1	6,67	6,67	93,33
	5	1	6,67	6,67	100,00
<i>Total</i>		15	100,0	100,0	

MANTENIBILIDAD

<i>N</i>	<i>Válido</i>	15
	<i>Perdidos</i>	0
<i>Media</i>		3,32
<i>Modo</i>		3,40
<i>Desv Std</i>		1,02
<i>Asimetría</i>		-,77
<i>Mínimo</i>		1,00
<i>Máximo</i>		5,00
	50 (Mediana)	3

HISTOGRAM



MANTENIBILIDAD

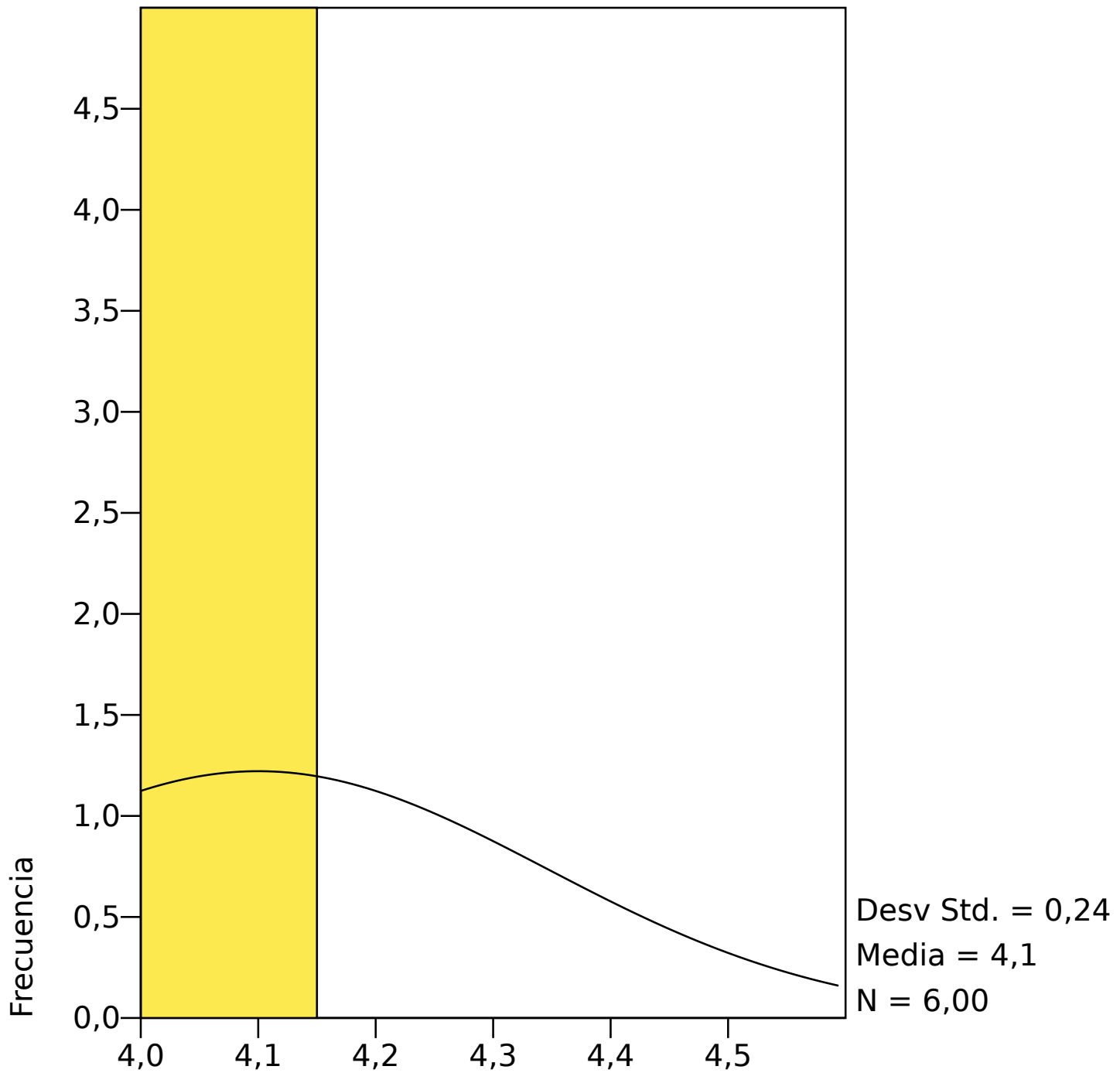
EFICIENCIA

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	4	5	33,33	83,33	83,33
	5	1	6,67	16,67	100,00
	.	9	60,00	Perdidos	
<i>Total</i>		15	100,0	100,0	

EFICIENCIA

<i>N</i>	<i>Válido</i>	6
	<i>Perdidos</i>	9
<i>Media</i>		4,10
<i>Modo</i>		4,00
<i>Desv Std</i>		,24
<i>Asimetría</i>		2,45
<i>Mínimo</i>		4,00
<i>Máximo</i>		4,60
	50 (Mediana)	4

HISTOGRAM



EFICIENCIA

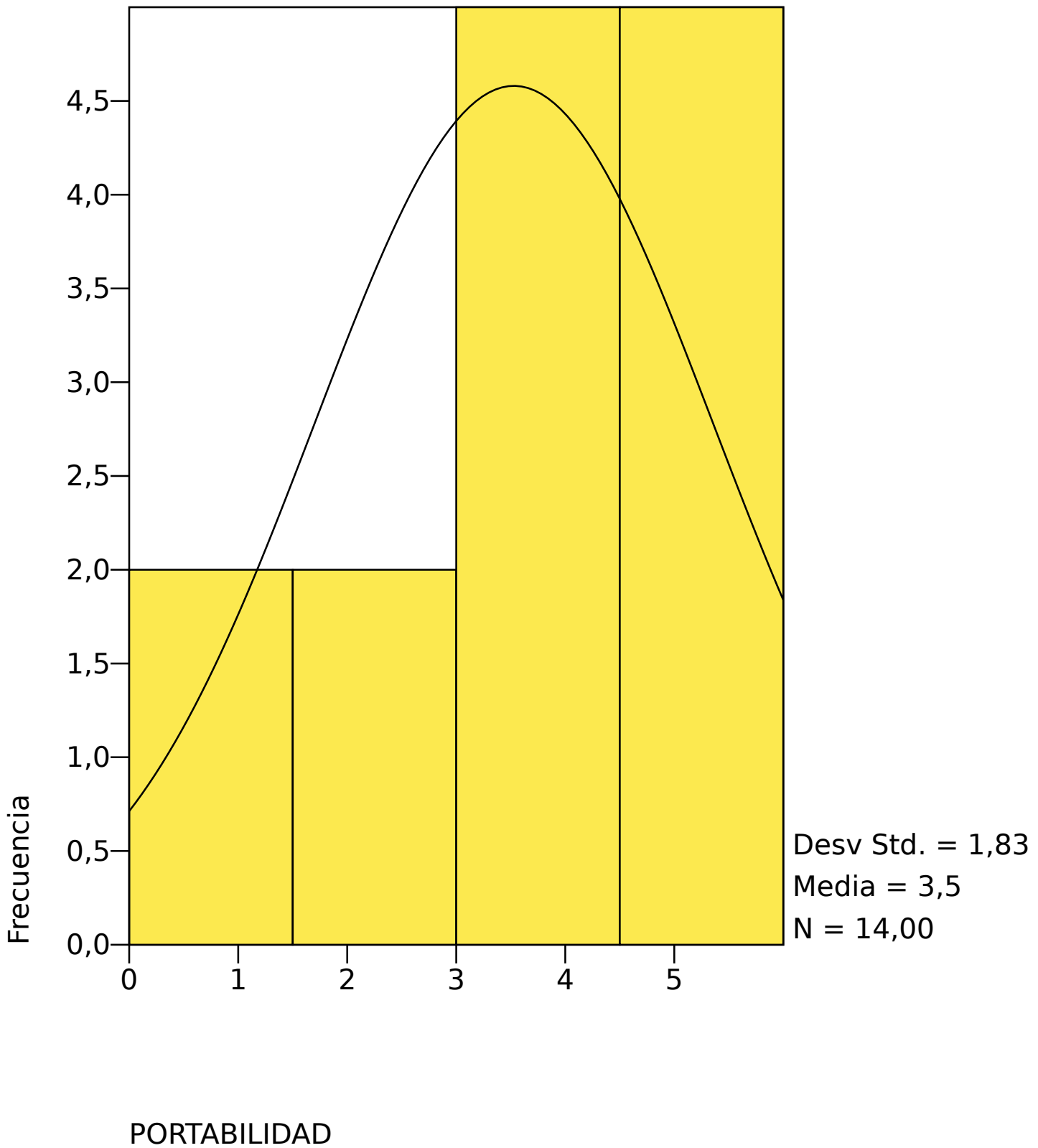
PORTABILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	0	2	13,33	14,29	14,29
	2	1	6,67	7,14	21,43
	2	1	6,67	7,14	28,57
	4	2	13,33	14,29	42,86
	4	3	20,00	21,43	64,29
	5	1	6,67	7,14	71,43
	5	4	26,67	28,57	100,00
	.	1	6,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

PORTABILIDAD

<i>N</i>	<i>Válido</i>	14
	<i>Perdidos</i>	1
<i>Media</i>		3,53
<i>Modo</i>		5,00
<i>Desv Std</i>		1,83
<i>Asimetría</i>		-1,21
<i>Mínimo</i>		,00
<i>Máximo</i>		5,00
	50 (Mediana)	4

HISTOGRAM



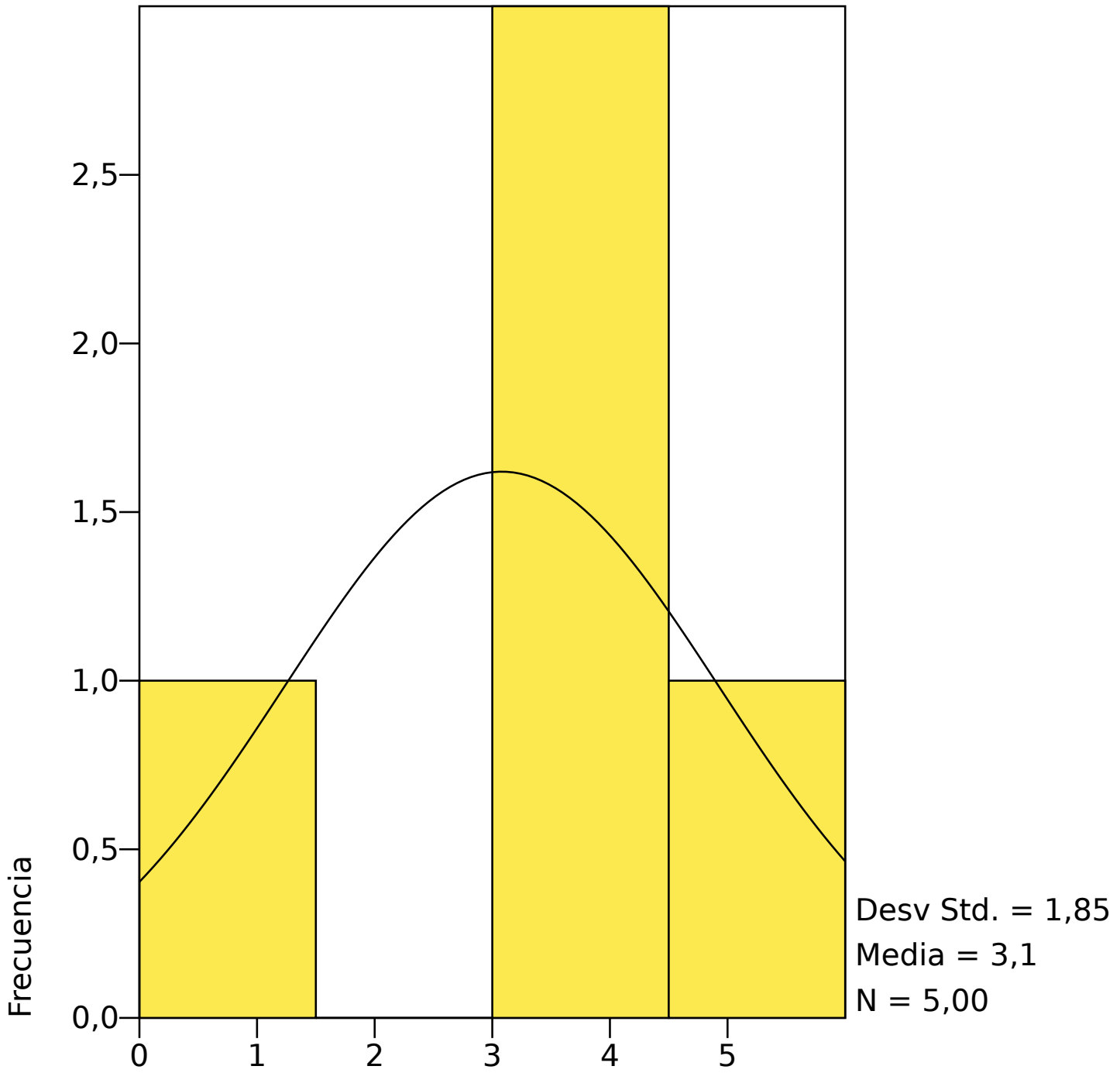
COMPATIBILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	0	1	6,67	20,00	20,00
	3	1	6,67	20,00	40,00
	3	1	6,67	20,00	60,00
	4	1	6,67	20,00	80,00
	5	1	6,67	20,00	100,00
	.	10	66,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

COMPATIBILIDAD

<i>N</i>	<i>Válido</i>	5
	<i>Perdidos</i>	10
<i>Media</i>		3,08
<i>Modo</i>		.
<i>Desv Std</i>		1,85
<i>Asimetría</i>		-1,55
<i>Mínimo</i>		,00
<i>Máximo</i>		4,60
	50 (Mediana)	3

HISTOGRAM



COMPATIBILIDAD

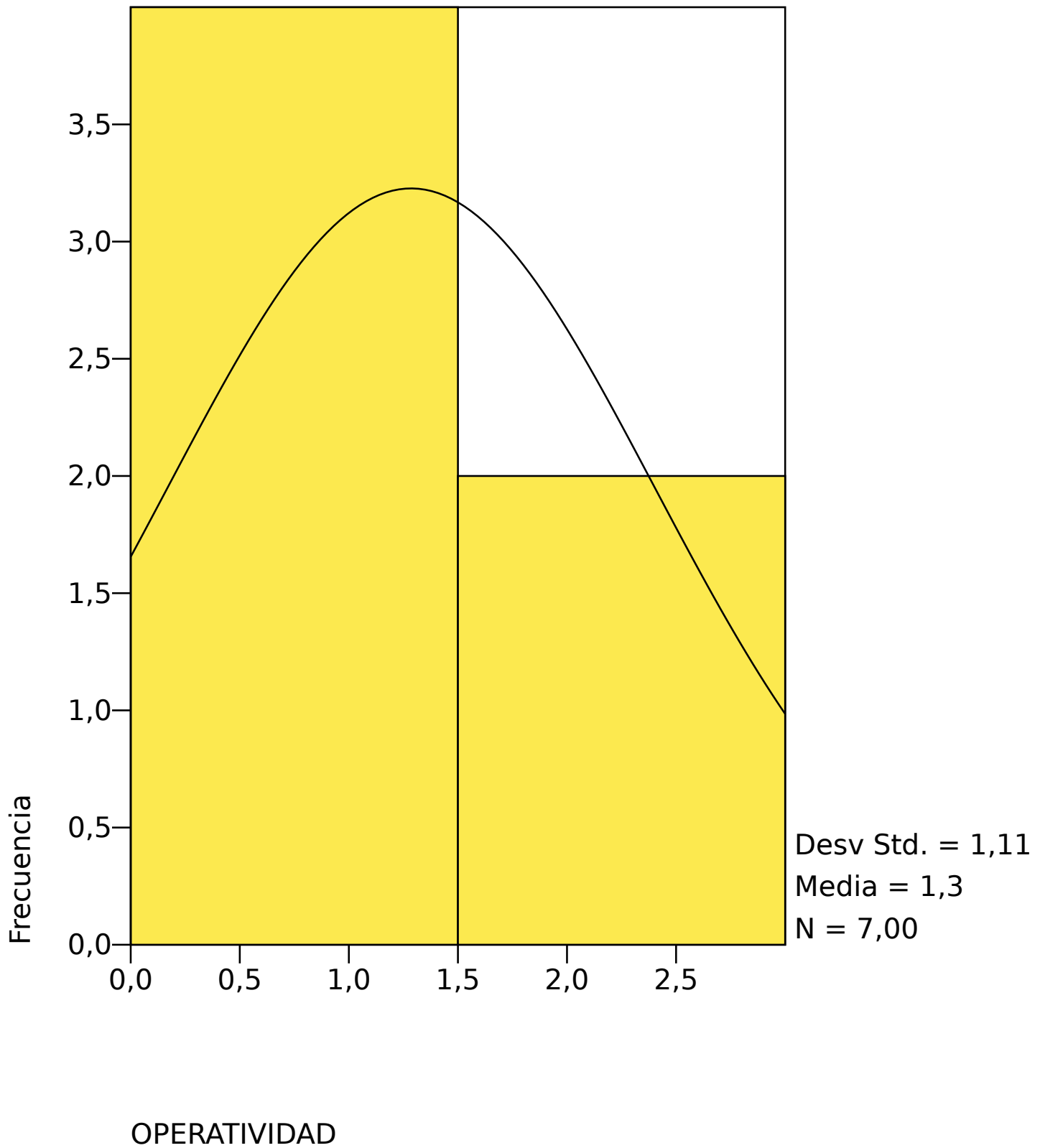
OPERATIVIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	0	2	13,33	28,57	28,57
	1	2	13,33	28,57	57,14
	2	2	13,33	28,57	85,71
	3	1	6,67	14,29	100,00
	.	8	53,33	Perdidos	
<i>Total</i>		15	100,0	100,0	

OPERATIVIDAD

<i>N</i>	<i>Válido</i>	7
	<i>Perdidos</i>	8
<i>Media</i>		1,29
<i>Modo</i>		.
<i>Desv Std</i>		1,11
<i>Asimetría</i>		,25
<i>Mínimo</i>		,00
<i>Máximo</i>		3,00
	50 (Mediana)	1

HISTOGRAM



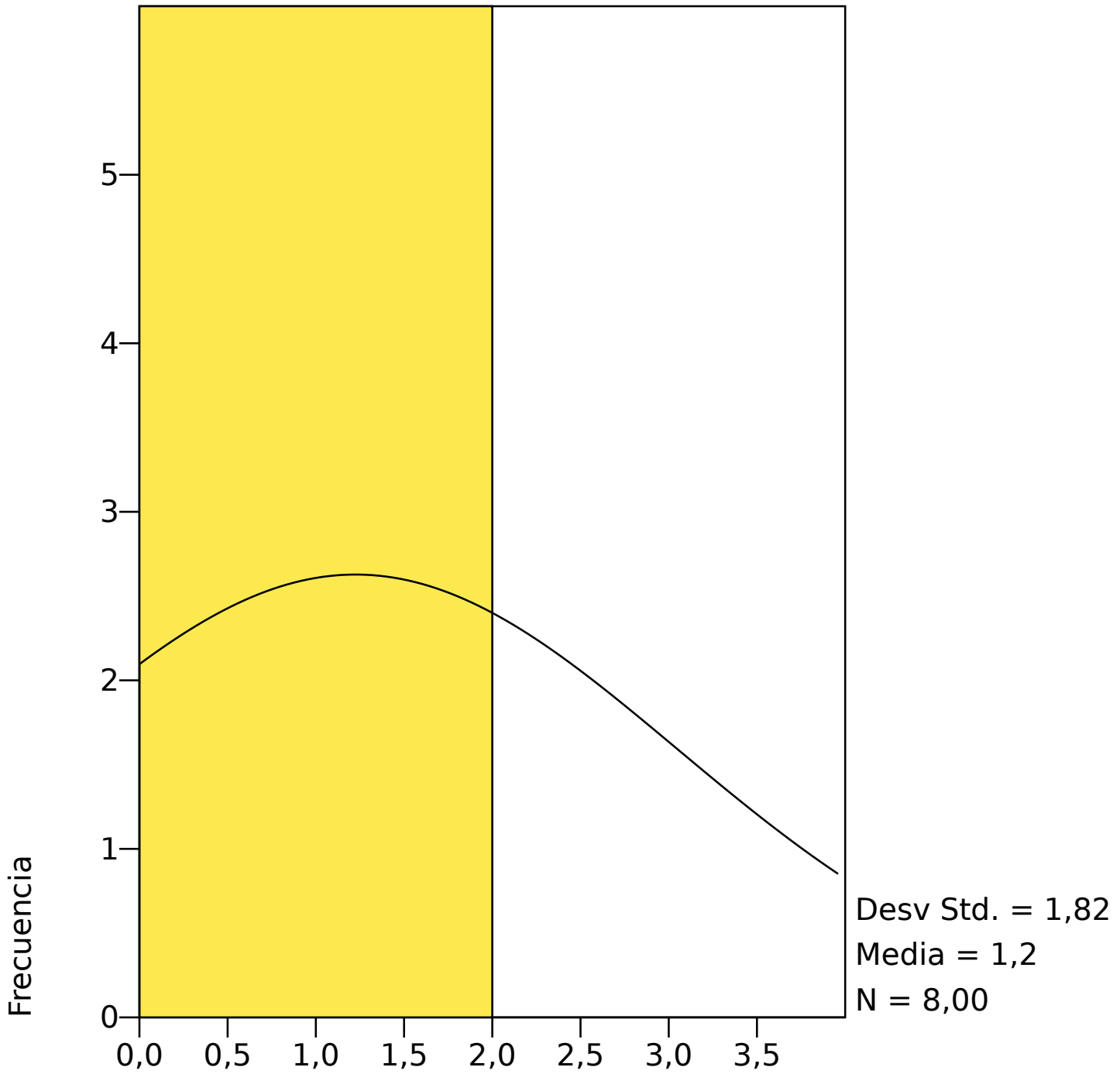
SEGURIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	0	5	33,33	62,50	62,50
	2	1	6,67	12,50	75,00
	4	2	13,33	25,00	100,00
	.	7	46,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

SEGURIDAD

<i>N</i>	<i>Válido</i>	8
	<i>Perdidos</i>	7
<i>Media</i>		1,23
<i>Modo</i>		,00
<i>Desv Std</i>		1,82
<i>Asimetría</i>		1,06
<i>Mínimo</i>		,00
<i>Máximo</i>		4,00
	50 (Mediana)	0

HISTOGRAM



SEGURIDAD

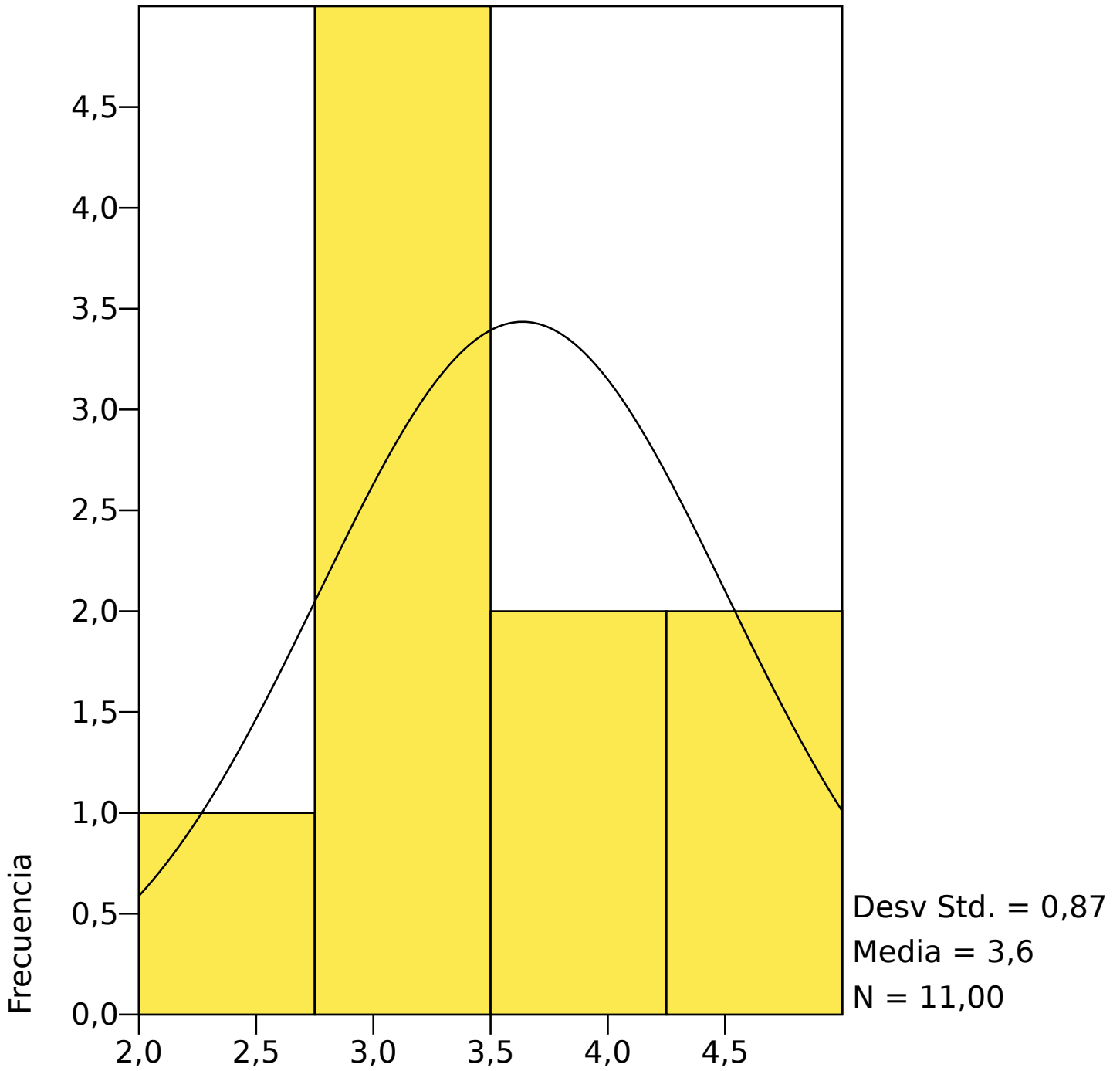
USABILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	2	1	6,67	9,09	9,09
	3	1	6,67	9,09	18,18
	3	1	6,67	9,09	27,27
	3	3	20,00	27,27	54,55
	4	2	13,33	18,18	72,73
	4	1	6,67	9,09	81,82
	5	1	6,67	9,09	90,91
	5	1	6,67	9,09	100,00
	.	4	26,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

USABILIDAD

<i>N</i>	<i>Válido</i>	11
	<i>Perdidos</i>	4
<i>Media</i>		3,64
<i>Modo</i>		3,40
<i>Desv Std</i>		,87
<i>Asimetría</i>		-,24
<i>Mínimo</i>		2,00
<i>Máximo</i>		5,00
	50 (Mediana)	3

HISTOGRAM



USABILIDAD

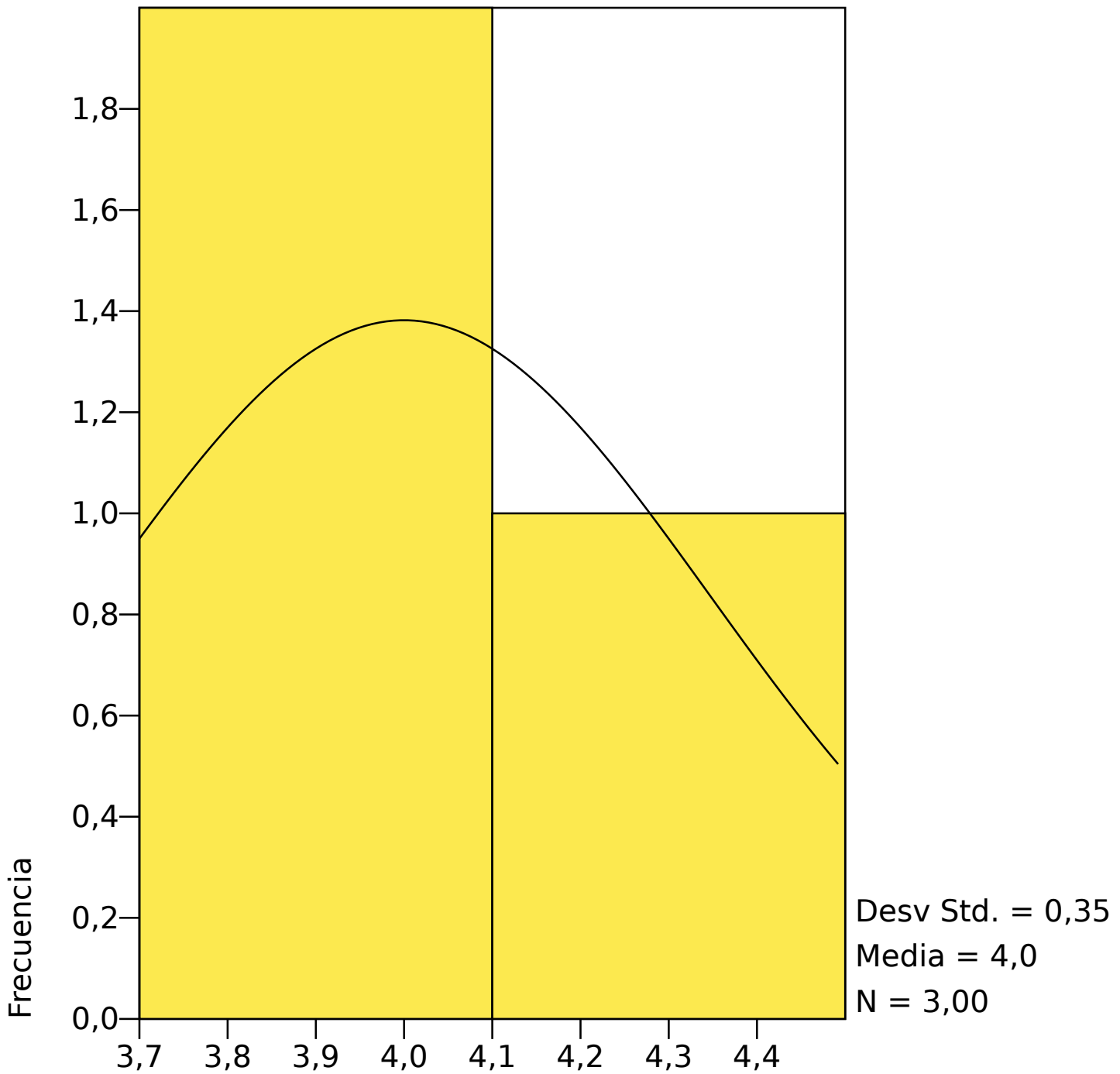
FLEXIBILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	3,8	2	13,33	66,67	66,67
	4,4	1	6,67	33,33	100,00
	.	12	80,00	Perdidos	
<i>Total</i>		15	100,0	100,0	

FLEXIBILIDAD

<i>N</i>	<i>Válido</i>	3
	<i>Perdidos</i>	12
<i>Media</i>		4,00
<i>Modo</i>		3,80
<i>Desv Std</i>		,35
<i>Asimetría</i>		1,73
<i>Mínimo</i>		3,80
<i>Máximo</i>		4,40
	50 (Mediana)	3,8

HISTOGRAM



FLEXIBILIDAD

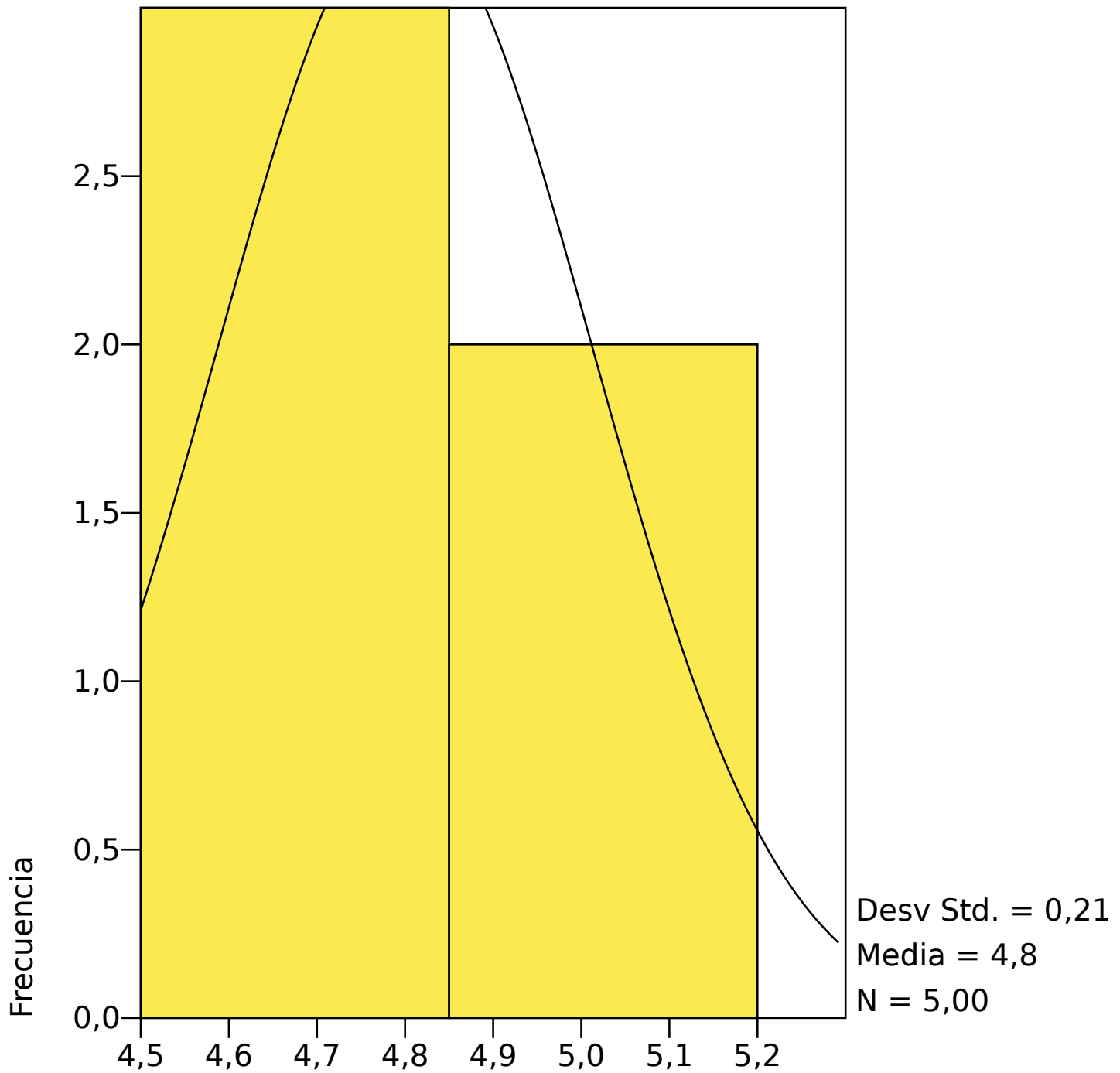
SEGURIDAD DE USO

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	5	1	6,67	20,00	20,00
	5	1	6,67	20,00	40,00
	5	1	6,67	20,00	60,00
	5	2	13,33	40,00	100,00
	.	10	66,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

SEGURIDAD DE USO

<i>N</i>	<i>Válido</i>	5
	<i>Perdidos</i>	10
<i>Media</i>		4,80
<i>Modo</i>		5,00
<i>Desv Std</i>		,21
<i>Asimetría</i>		-,52
<i>Mínimo</i>		4,50
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



SEGURIDAD_DE_USO

DESCRIPTIVES

DESCRIPTIVES

/VARIABLES= FUNCIONABILIDAD CONFIABILIDAD MANTENIBILIDAD
EFICIENCIA PORTABILIDAD COMPATIBILIDAD OPERATIVIDAD
SEGURIDAD USABILIDAD FLEXIBILIDAD SEGURIDAD_DE_USO
/MISSING=INCLUDE
/STATISTICS=VARIANCE.

Casos válidos = 15; casos con valor(es) perdido(s) = 12.

<i>Variable</i>	<i>N</i>	<i>Varianza</i>
FUNCIONABILIDAD	3	,12
CONFIABILIDAD	12	1,04
MANTENIBILIDAD	15	1,05
EFICIENCIA	6	,06
PORTABILIDAD	14	3,35
COMPATIBILIDAD	5	3,41
OPERATIVIDAD	7	1,24
SEGURIDAD	8	3,32
USABILIDAD	11	,76
FLEXIBILIDAD	3	,12
SEGURIDAD_DE_USO	5	,04

Parámetros Estadísticos de Mininet-WiFi

GET DATA

```
/TYPE=TXT  
/FILE="/home/gerlyn/Documentos/Tesis Gerlyn/Promedios Preguntas  
MininetWiFi NEW.csv"  
/IMPORTCASES=ALL  
/ARRANGEMENT=DELIMITED  
/DELCASE=LINE  
/FIRSTCASE=2  
/DELIMITERS=","  
/QUALIFIER=""  
/VARIABLES=  
  FUNCIONABILIDAD F3.0  
  CONFIABILIDAD F3.0  
  MANTENIBILIDAD F3.0  
  EFICIENCIA F3.0  
  PORTABILIDAD F3.0  
  COMPATIBILIDAD F3.0  
  SEGURIDAD F3.0  
  USABILIDAD F3.0  
  FLEXIBILIDAD F1.0  
  SEGURIDAD_DE_USO F3.0.
```

FREQUENCIES

FREQUENCIES

```
/VARIABLES= FUNCIONABILIDAD CONFIABILIDAD  
MANTENIBILIDAD EFICIENCIA PORTABILIDAD COMPATIBILIDAD  
SEGURIDAD USABILIDAD FLEXIBILIDAD SEGURIDAD_DE_USO  
/FORMAT=AVALUE TABLE  
/STATISTICS=DEFAULT SEMEAN VARIANCE SKEWNESS RANGE  
MODE MEDIAN  
/MISSING=INCLUDE  
/HISTOGRAM=NORMAL.
```

FUNCIONABILIDAD

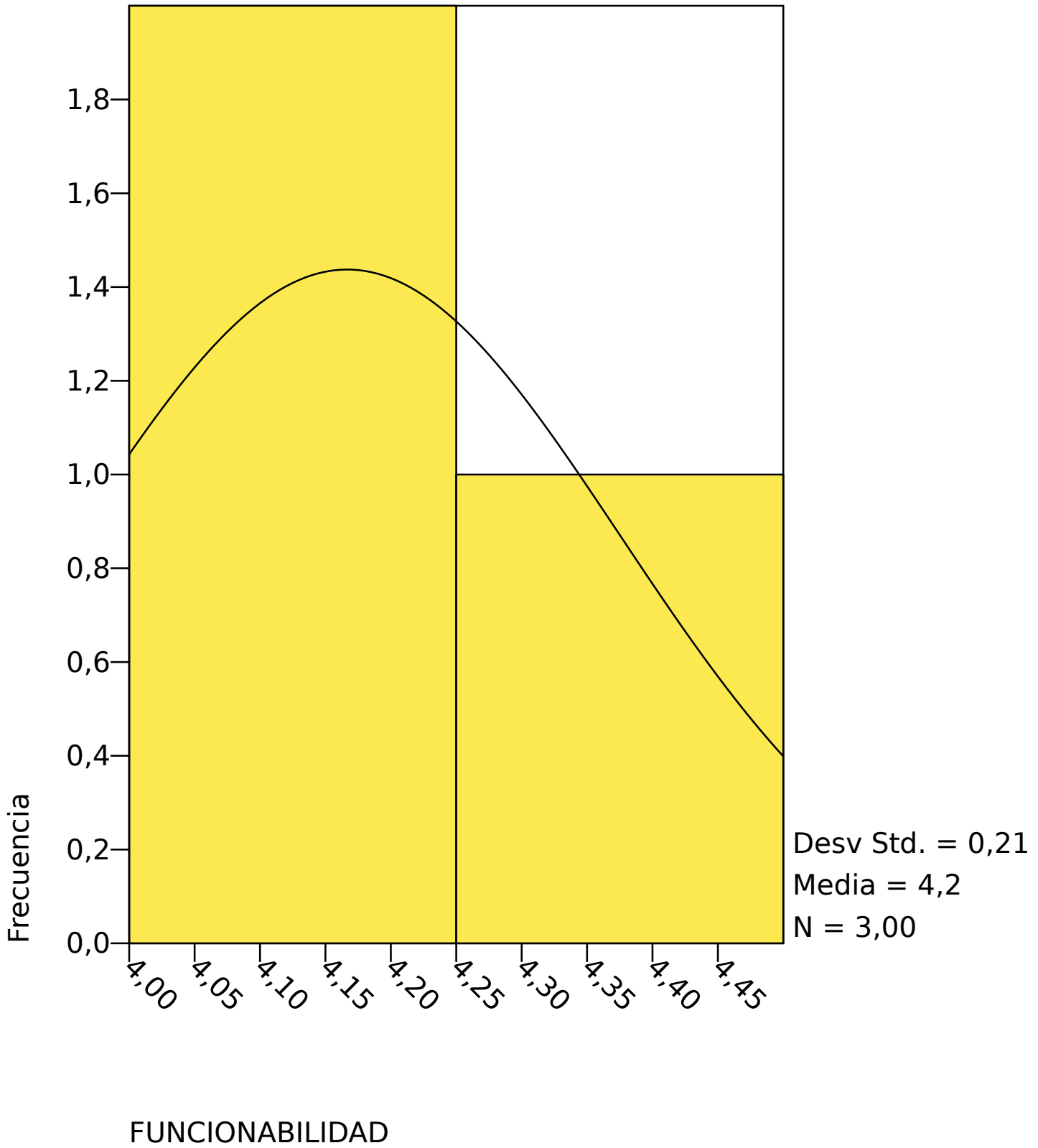
<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	4	1	6,67	33,33	33,33
	4	1	6,67	33,33	66,67
	4	1	6,67	33,33	100,00
	.	12	80,00	Perdidos	

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	<i>Total</i>	15	100,0	100,0	

FUNCIONABILIDAD

<i>N</i>	<i>Válido</i>	3
	<i>Perdidos</i>	12
<i>Media</i>		4,17
<i>Err.Est.Media</i>		,12
<i>Modo</i>		.
<i>Desv Std</i>		,21
<i>Asimetría</i>		1,29
<i>Intervalo</i>		,40
<i>Mínimo</i>		4,00
<i>Máximo</i>		4,40
	50 (Mediana)	4

HISTOGRAM



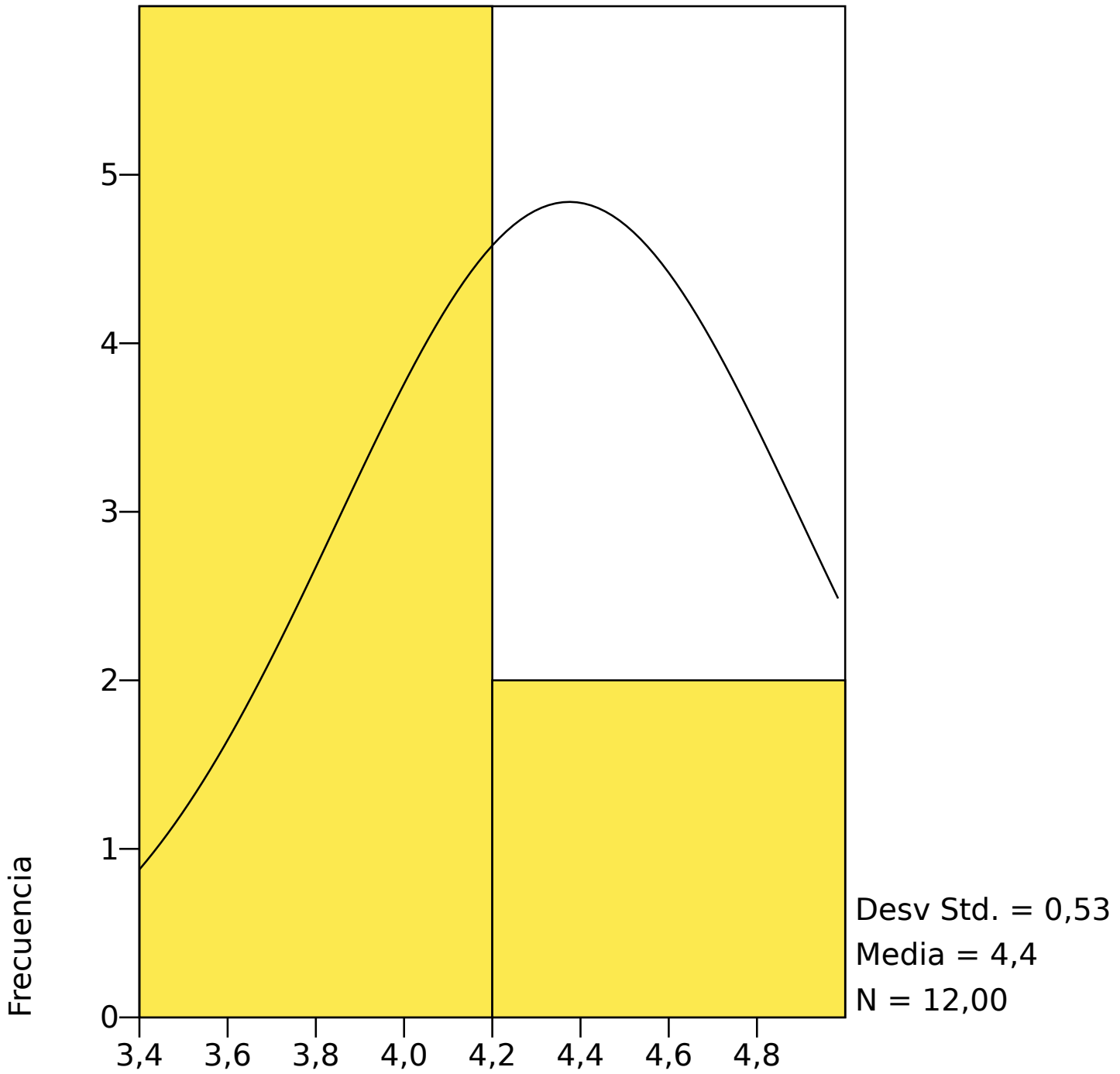
CONFIABILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	4	1	6,67	8,33	8,33
	4	5	33,33	41,67	50,00
	5	2	13,33	16,67	66,67
	5	4	26,67	33,33	100,00
	.	3	20,00	Perdidos	
<i>Total</i>		15	100,0	100,0	

CONFIABILIDAD

<i>N</i>	<i>Válido</i>	12
	<i>Perdidos</i>	3
<i>Media</i>		4,38
<i>Err.Est.Media</i>		,15
<i>Modo</i>		4,00
<i>Desv Std</i>		,53
<i>Asimetría</i>		,03
<i>Intervalo</i>		1,50
<i>Mínimo</i>		3,50
<i>Máximo</i>		5,00
	50 (Mediana)	4

HISTOGRAM



CONFIAIBILIDAD

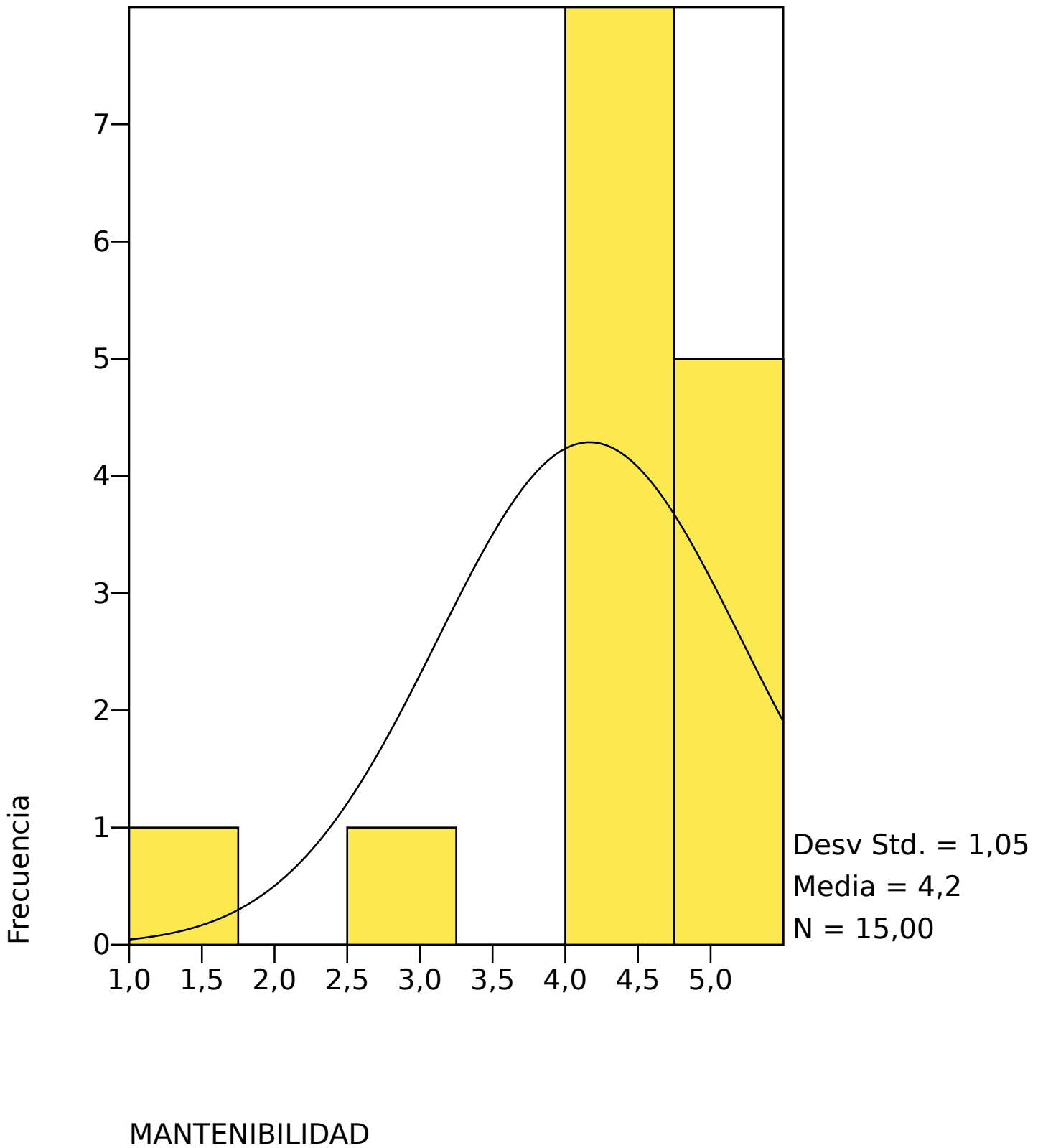
MANTENIBILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	1	1	6,67	6,67	6,67
	3	1	6,67	6,67	13,33
	4	5	33,33	33,33	46,67
	5	3	20,00	20,00	66,67
	5	5	33,33	33,33	100,00
<i>Total</i>		15	100,0	100,0	

MANTENIBILIDAD

<i>N</i>	<i>Válido</i>	15
	<i>Perdidos</i>	0
<i>Media</i>		4,17
<i>Err.Est.Media</i>		,27
<i>Modo</i>		.
<i>Desv Std</i>		1,05
<i>Asimetría</i>		-2,18
<i>Intervalo</i>		4,00
<i>Mínimo</i>		1,00
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



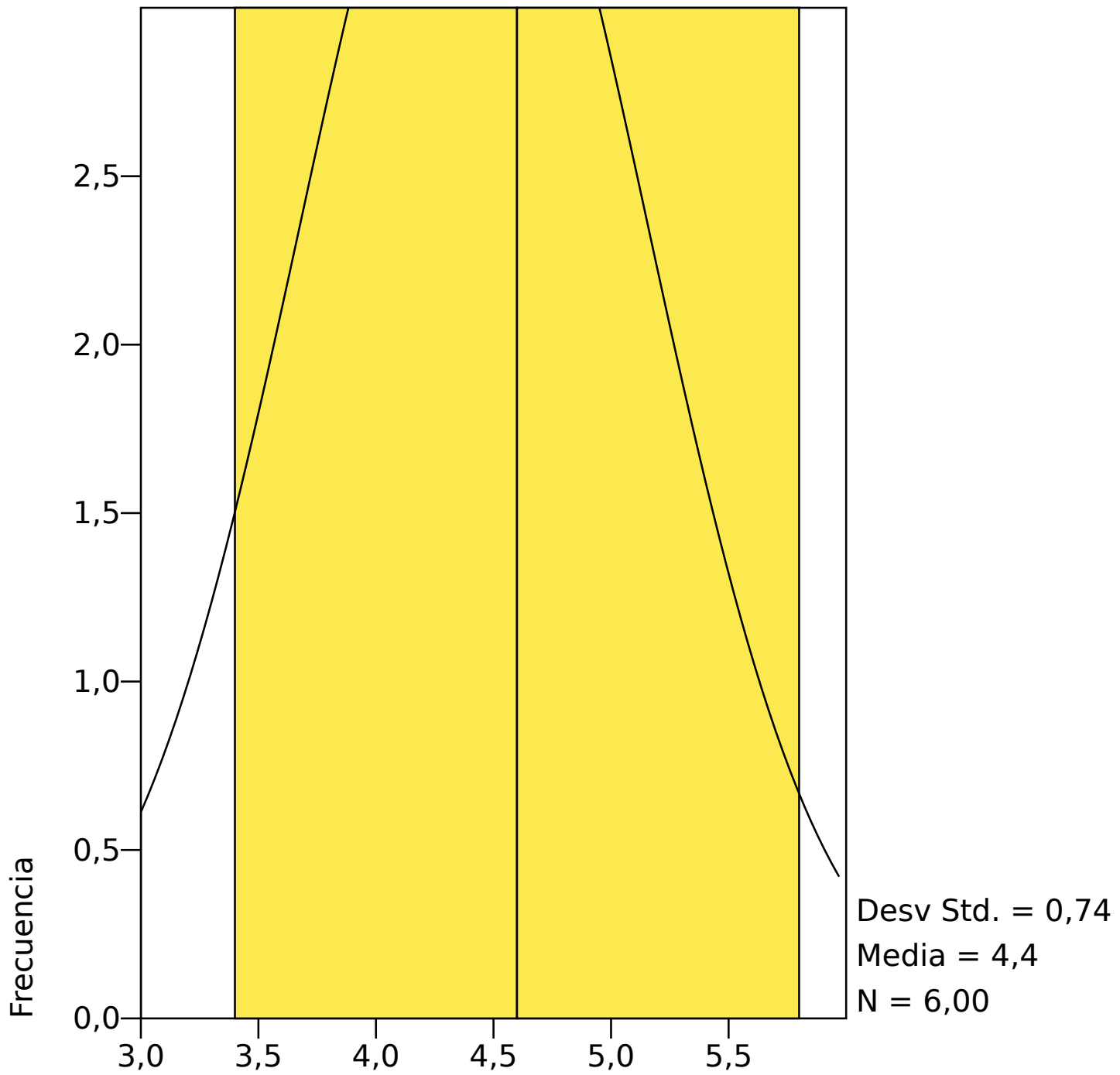
EFICIENCIA

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	4	2	13,33	33,33	33,33
	5	1	6,67	16,67	50,00
	5	3	20,00	50,00	100,00
	.	9	60,00	Perdidos	
<i>Total</i>		15	100,0	100,0	

EFICIENCIA

<i>N</i>	<i>Válido</i>	6
	<i>Perdidos</i>	9
<i>Media</i>		4,42
<i>Err.Est.Media</i>		,30
<i>Modo</i>		5,00
<i>Desv Std</i>		,74
<i>Asimetría</i>		-,71
<i>Intervalo</i>		1,50
<i>Mínimo</i>		3,50
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



EFICIENCIA

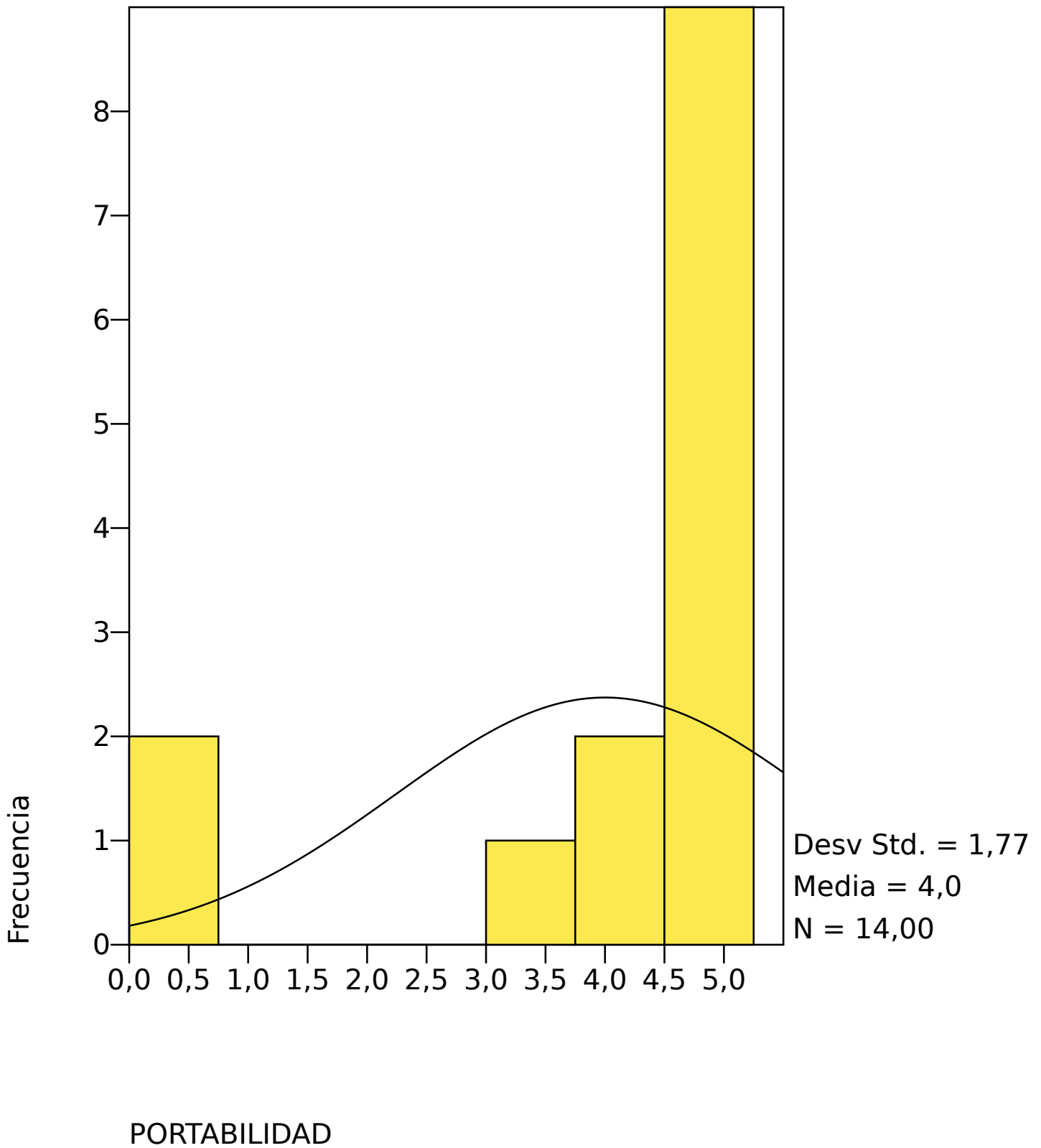
PORTABILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	0	2	13,33	14,29	14,29
	4	1	6,67	7,14	21,43
	4	2	13,33	14,29	35,71
	5	1	6,67	7,14	42,86
	5	8	53,33	57,14	100,00
	.	1	6,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

PORTABILIDAD

<i>N</i>	<i>Válido</i>	14
	<i>Perdidos</i>	1
<i>Media</i>		4,00
<i>Err.Est.Media</i>		,47
<i>Modo</i>		5,00
<i>Desv Std</i>		1,77
<i>Asimetría</i>		-1,96
<i>Intervalo</i>		5,00
<i>Mínimo</i>		,00
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



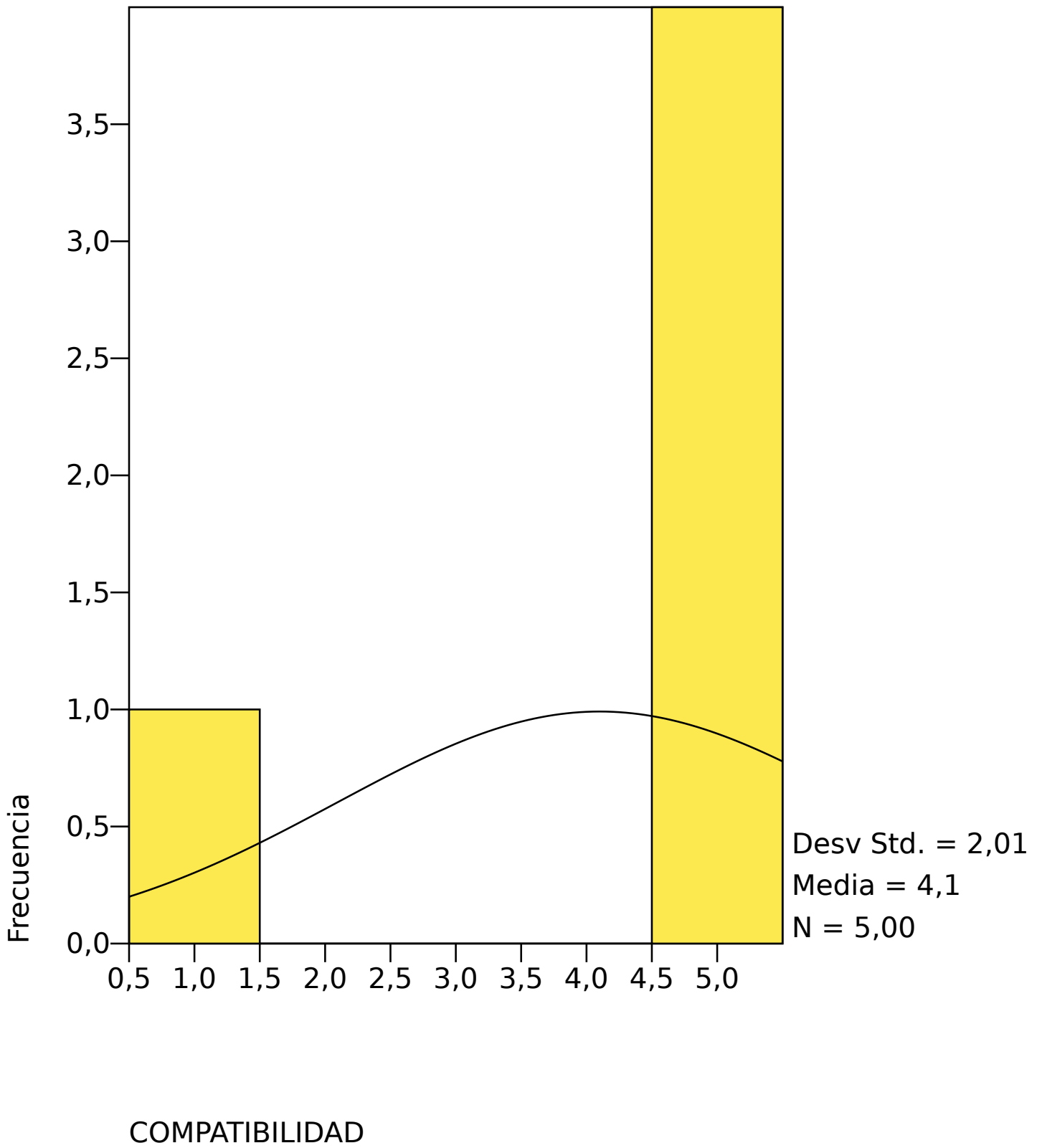
COMPATIBILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	1	1	6,67	20,00	20,00
	5	4	26,67	80,00	100,00
	.	10	66,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

COMPATIBILIDAD

<i>N</i>	<i>Válido</i>	5
	<i>Perdidos</i>	10
<i>Media</i>		4,10
<i>Err.Est.Media</i>		,90
<i>Modo</i>		5,00
<i>Desv Std</i>		2,01
<i>Asimetría</i>		-2,24
<i>Intervalo</i>		4,50
<i>Mínimo</i>		,50
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



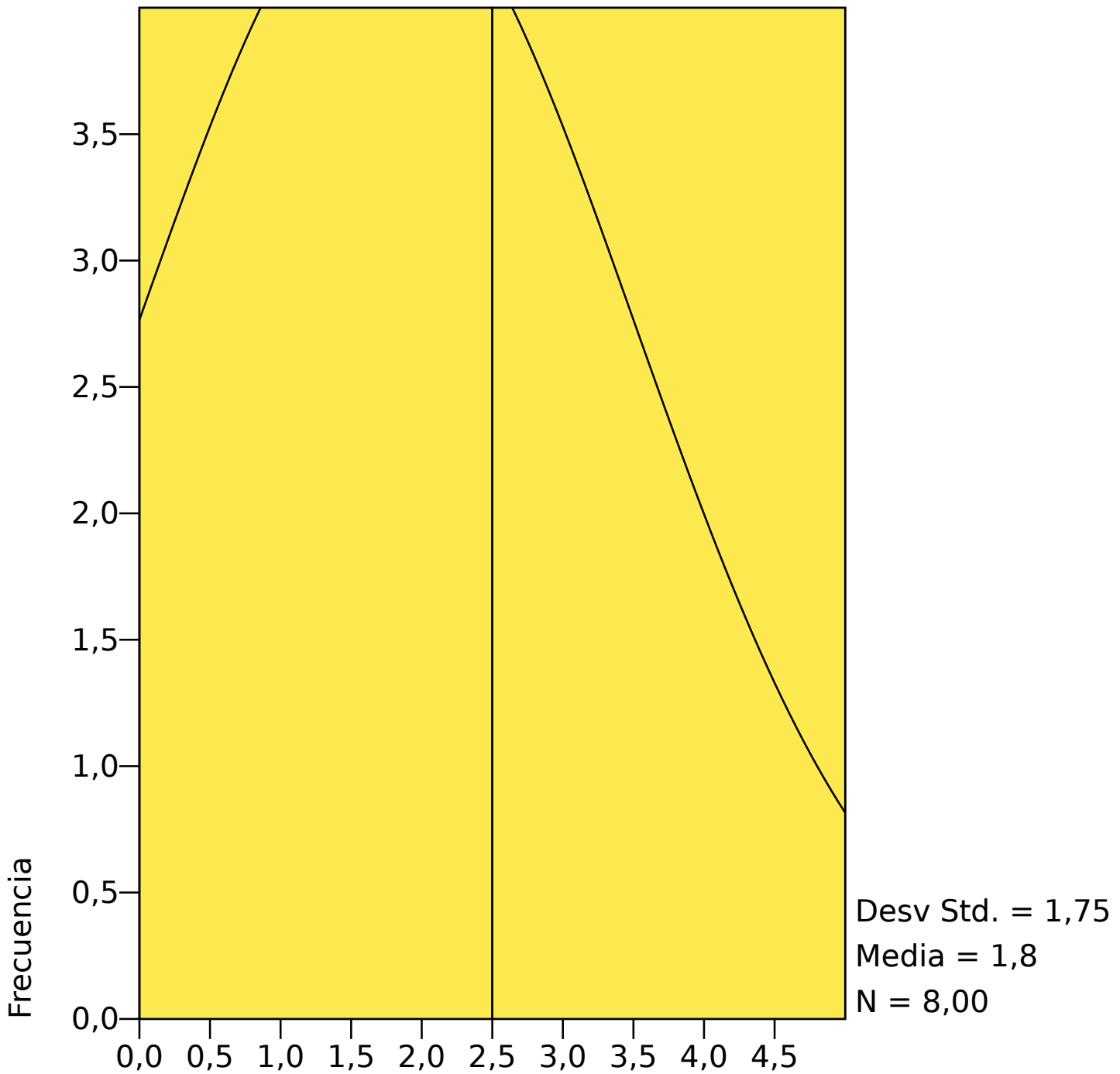
SEGURIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	0	3	20,00	37,50	37,50
	1	1	6,67	12,50	50,00
	3	2	13,33	25,00	75,00
	4	1	6,67	12,50	87,50
	5	1	6,67	12,50	100,00
	.	7	46,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

SEGURIDAD

<i>N</i>	<i>Válido</i>	8
	<i>Perdidos</i>	7
<i>Media</i>		1,75
<i>Err.Est.Media</i>		,62
<i>Modo</i>		,00
<i>Desv Std</i>		1,75
<i>Asimetría</i>		,37
<i>Intervalo</i>		4,50
<i>Mínimo</i>		,00
<i>Máximo</i>		4,50
	50 (Mediana)	2

HISTOGRAM



SEGURIDAD

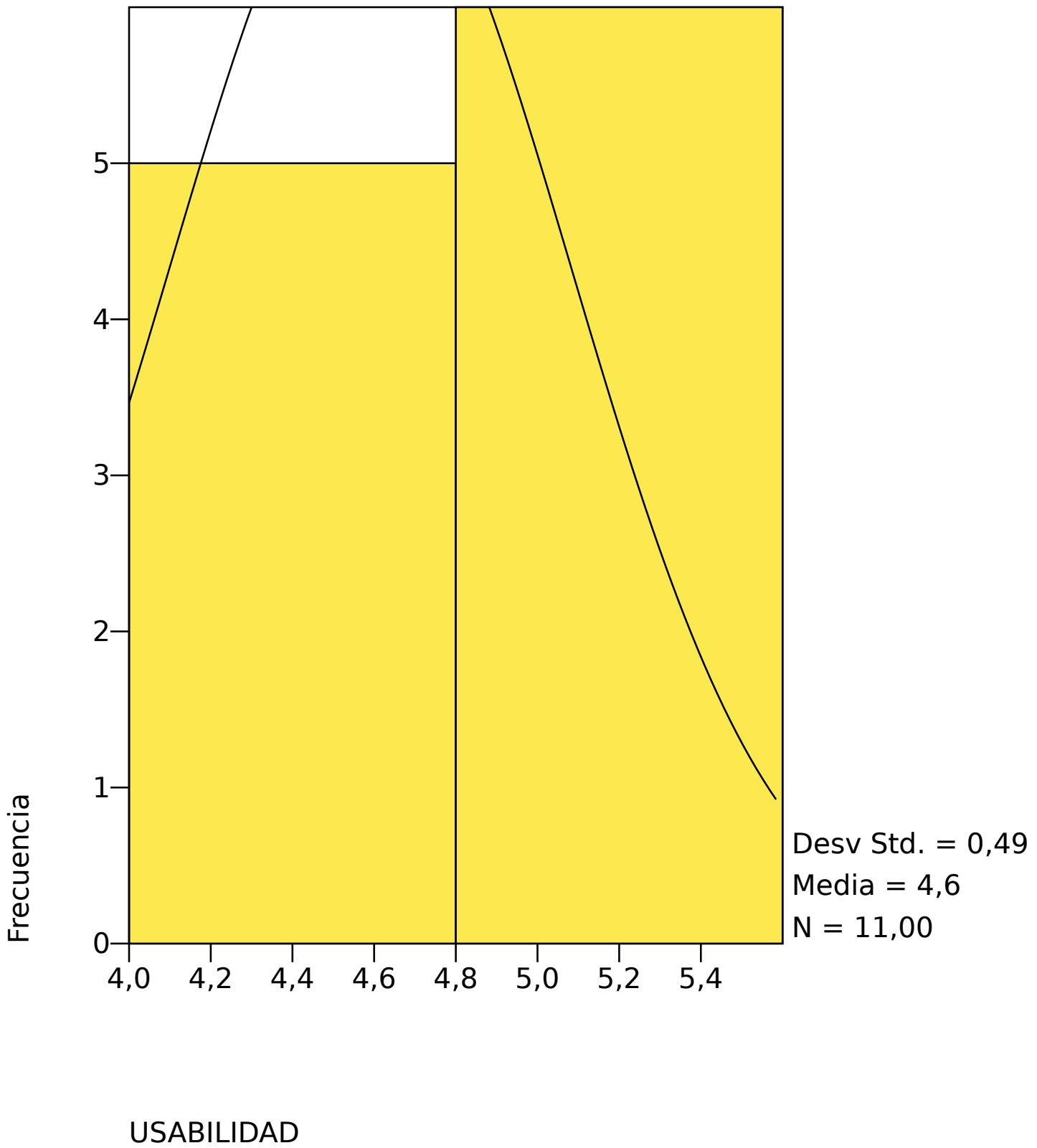
USABILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	4	4	26,67	36,36	36,36
	5	1	6,67	9,09	45,45
	5	6	40,00	54,55	100,00
	.	4	26,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

USABILIDAD

<i>N</i>	<i>Válido</i>	11
	<i>Perdidos</i>	4
<i>Media</i>		4,59
<i>Err.Est.Media</i>		,15
<i>Modo</i>		5,00
<i>Desv Std</i>		,49
<i>Asimetría</i>		-,43
<i>Intervalo</i>		1,00
<i>Mínimo</i>		4,00
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



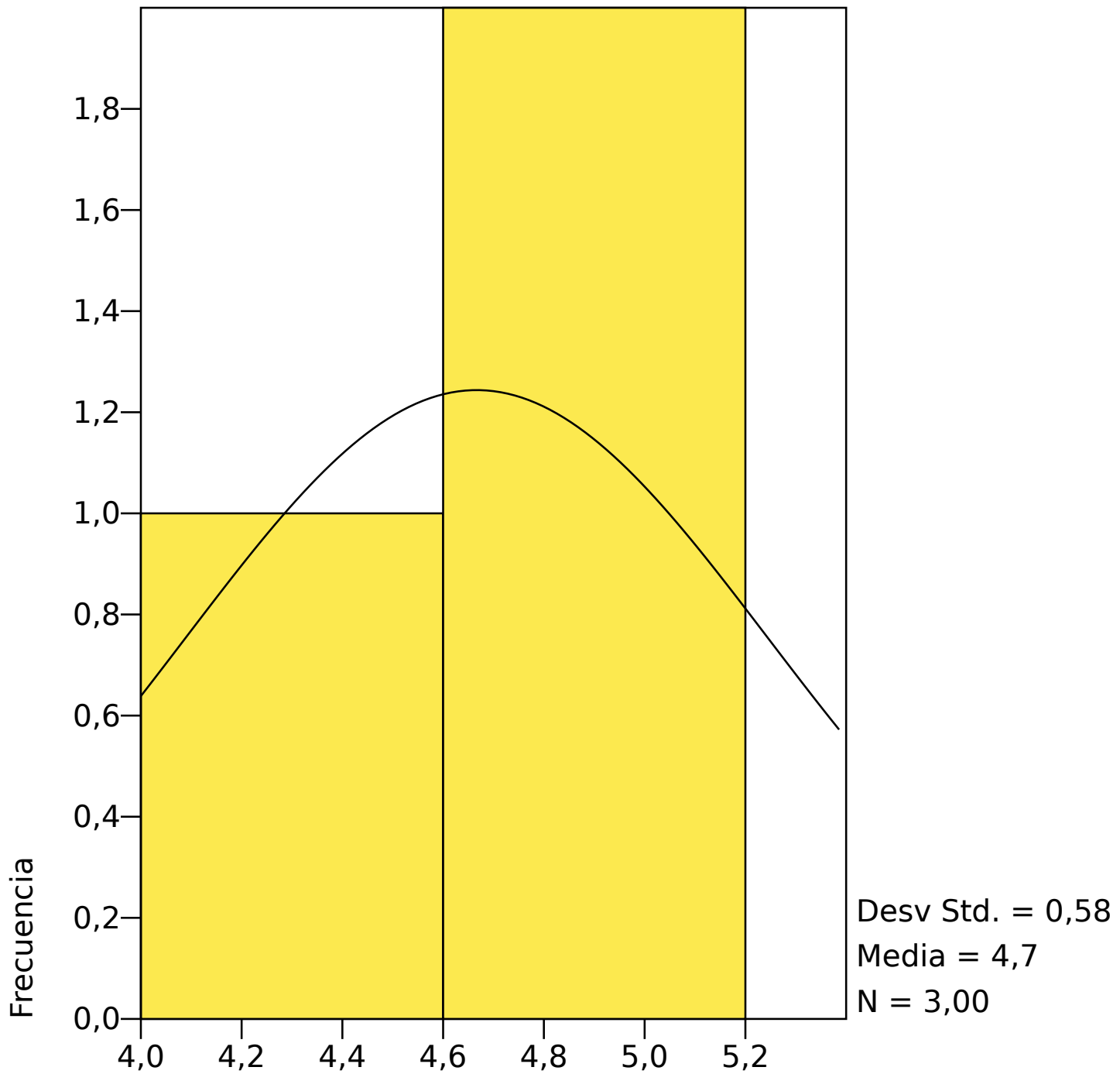
FLEXIBILIDAD

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	4	1	6,67	33,33	33,33
	5	2	13,33	66,67	100,00
	.	12	80,00	Perdidos	
<i>Total</i>		15	100,0	100,0	

FLEXIBILIDAD

<i>N</i>	<i>Válido</i>	3
	<i>Perdidos</i>	12
<i>Media</i>		4,67
<i>Err.Est.Media</i>		,33
<i>Modo</i>		5,00
<i>Desv Std</i>		,58
<i>Asimetría</i>		-1,73
<i>Intervalo</i>		1,00
<i>Mínimo</i>		4,00
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



FLEXIBILIDAD

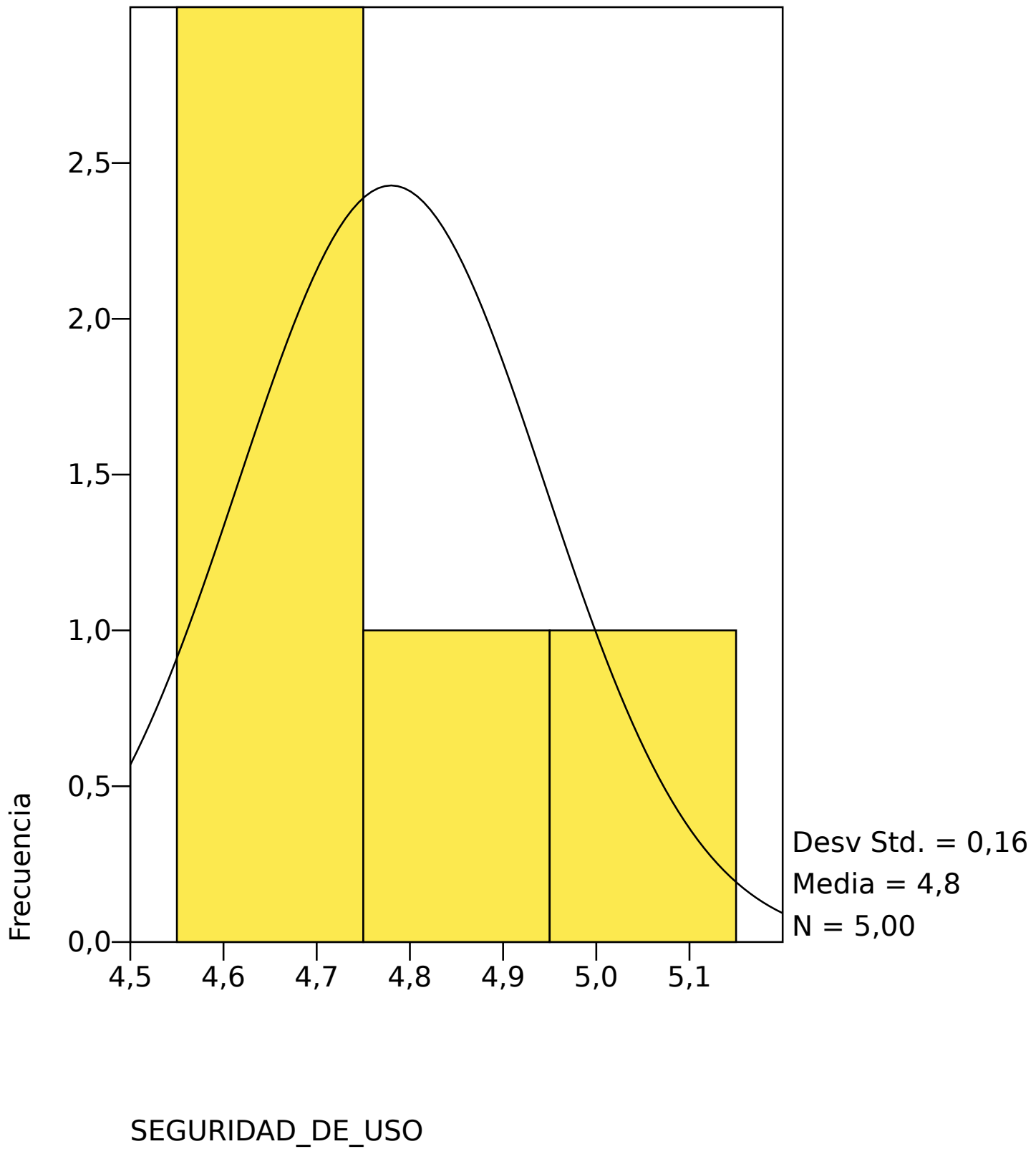
SEGURIDAD_DE_USO

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	5	1	6,67	20,00	20,00
	5	2	13,33	40,00	60,00
	5	1	6,67	20,00	80,00
	5	1	6,67	20,00	100,00
	.	10	66,67	Perdidos	
<i>Total</i>		15	100,0	100,0	

SEGURIDAD_DE_USO

<i>N</i>	<i>Válido</i>	5
	<i>Perdidos</i>	10
<i>Media</i>		4,78
<i>Err.Est.Media</i>		,07
<i>Modo</i>		4,70
<i>Desv Std</i>		,16
<i>Asimetría</i>		,52
<i>Intervalo</i>		,40
<i>Mínimo</i>		4,60
<i>Máximo</i>		5,00
	50 (Mediana)	5

HISTOGRAM



DESCRIPTIVES

DESCRIPTIVES

/VARIABLES= FUNCIONABILIDAD CONFIABILIDAD MANTENIBILIDAD
EFICIENCIA PORTABILIDAD COMPATIBILIDAD SEGURIDAD USABILIDAD
FLEXIBILIDAD SEGURIDAD_DE_USO

/MISSING=INCLUDE

/STATISTICS=VARIANCE.

Casos válidos = 15; casos con valor(es) perdido(s) = 12.

<i>Variable</i>	<i>N</i>	<i>Varianza</i>
FUNCIONABILIDAD	3	,04
CONFIABILIDAD	12	,28
MANTENIBILIDAD	15	1,10
EFICIENCIA	6	,54
PORTABILIDAD	14	3,12
COMPATIBILIDAD	5	4,05
SEGURIDAD	8	3,07
USABILIDAD	11	,24
FLEXIBILIDAD	3	,33
SEGURIDAD_DE_USO	5	,03

Parámetros Estadísticos de la UX

GET FILE="/home/gerlyn/Documentos/Tesis Gerlyn/Análisis UX SUS.sav".

FREQUENCIES

FREQUENCIES

/VARIABLES= Mininet_WiFi OpenNet
 /FORMAT=AVALUE TABLE
 /STATISTICS=DEFAULT SEMEAN VARIANCE MODE MEDIAN
 /MISSING=INCLUDE
 /HISTOGRAM=NORMAL.

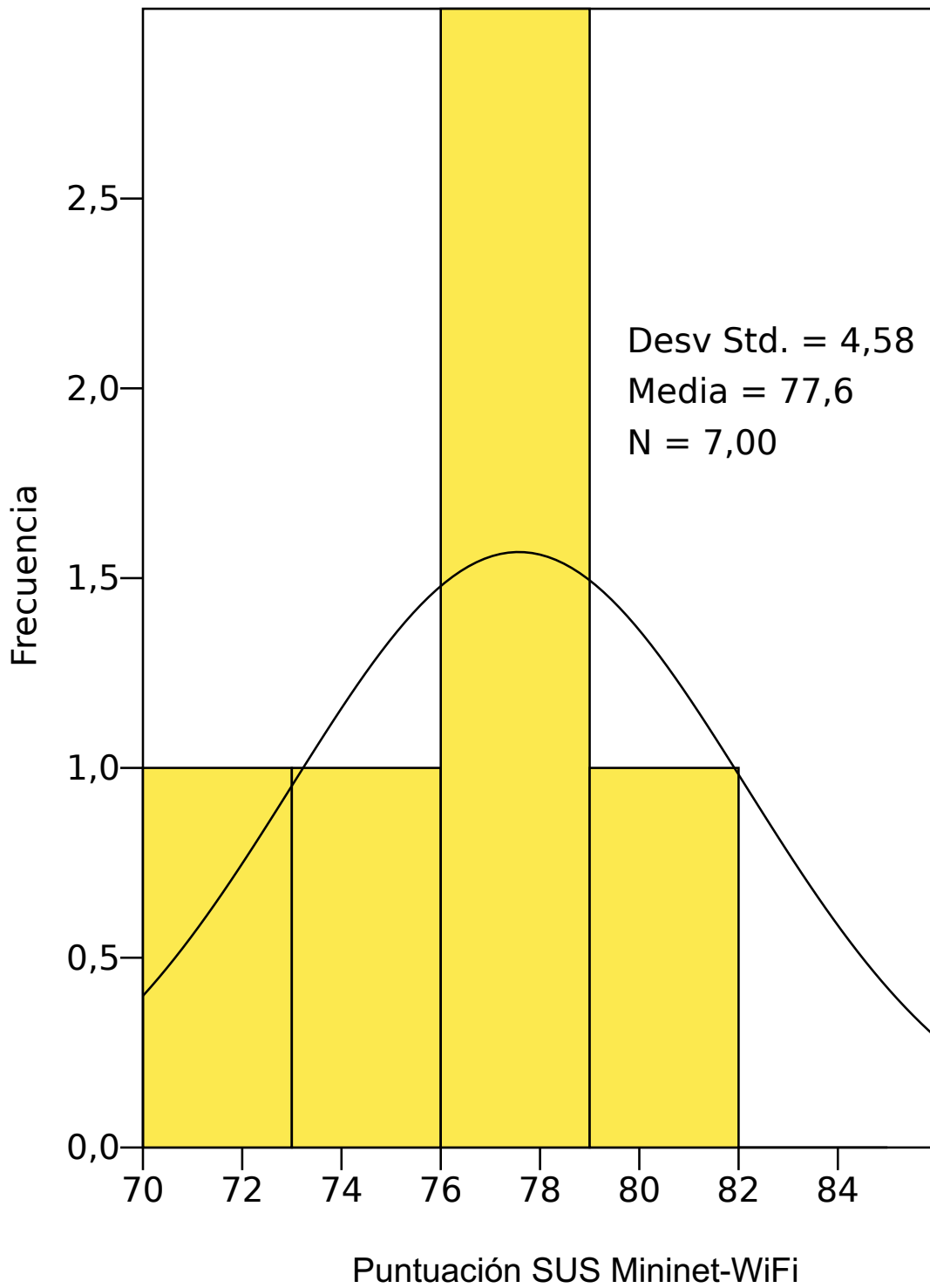
Mininet_WiFi

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	70	1	14,29	14,29	14,29
	75	1	14,29	14,29	28,57
	77	1	14,29	14,29	42,86
	78	2	28,57	28,57	71,43
	80	1	14,29	14,29	85,71
	85	1	14,29	14,29	100,00
<i>Total</i>		7	100,0	100,0	

Mininet_WiFi

<i>N</i>	<i>Válido</i>	7
	<i>Perdidos</i>	0
<i>Media</i>		77,57
<i>Err.Est.Media</i>		1,73
<i>Modo</i>		78,00
<i>Desv Std</i>		4,58
<i>Mínimo</i>		70,00
<i>Máximo</i>		85,00
	50 (Mediana)	78

HISTOGRAM



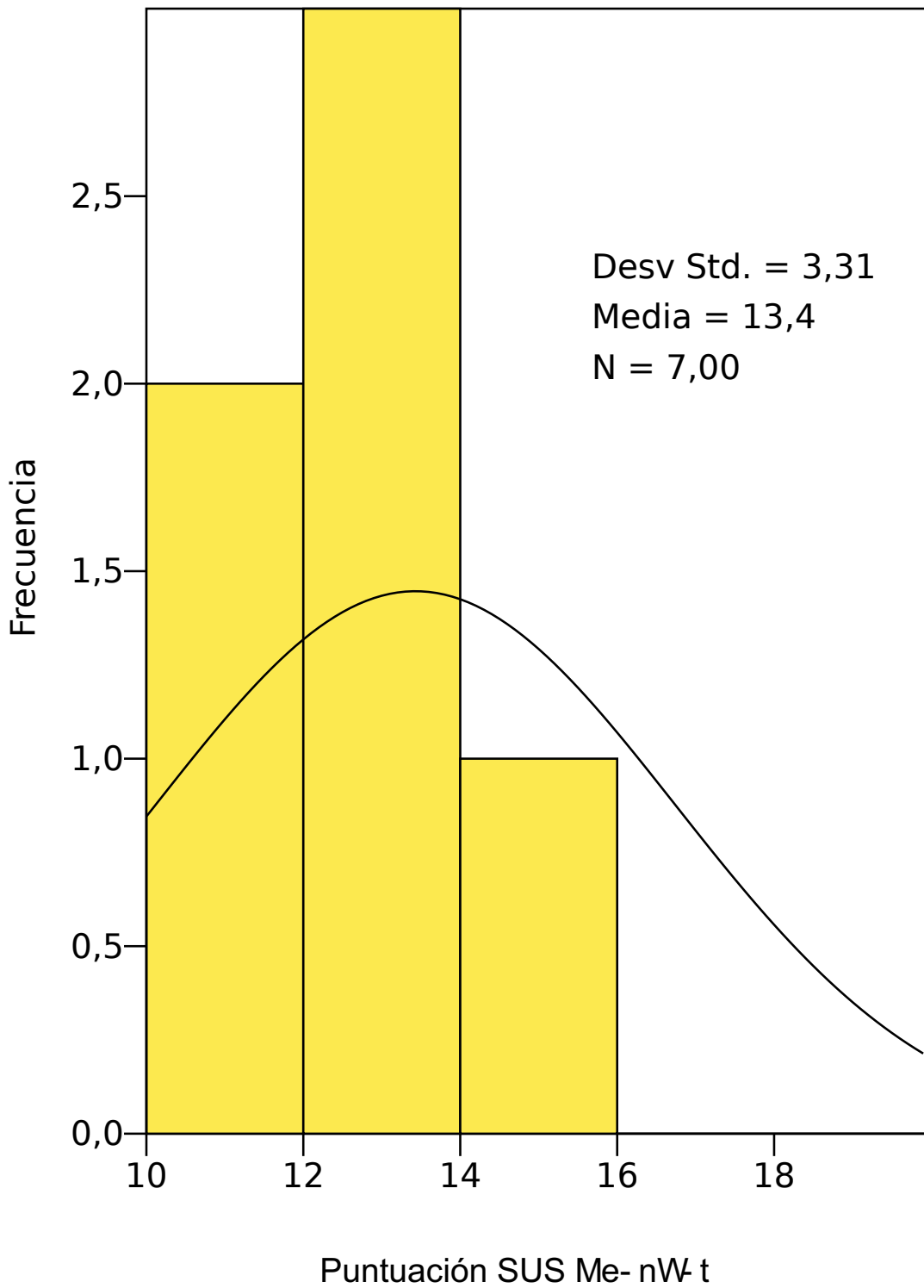
OpenNet

<i>Etiqueta de Valor</i>	<i>Valor</i>	<i>Frecuencia</i>	<i>Porcentaje</i>	<i>Porcentaje Válido</i>	<i>Porcentaje Acumulado</i>
	10	1	14,29	14,29	14,29
	11	1	14,29	14,29	28,57
	12	1	14,29	14,29	42,86
	13	2	28,57	28,57	71,43
	15	1	14,29	14,29	85,71
	20	1	14,29	14,29	100,00
<i>Total</i>		7	100,0	100,0	

OpenNet

<i>N</i>	<i>Válido</i>	7
	<i>Perdidos</i>	0
<i>Media</i>		13,43
<i>Err.Est.Media</i>		1,25
<i>Modo</i>		13,00
<i>Desv Std</i>		3,31
<i>Mínimo</i>		10,00
<i>Máximo</i>		20,00
	50 (Mediana)	13

HISTOGRAM



DESCRIPTIVES

DESCRIPTIVES

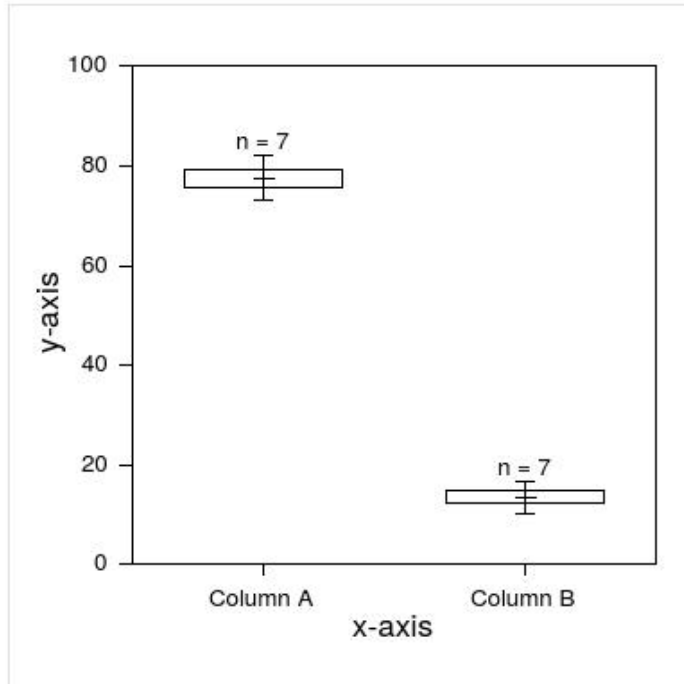
```
/VARIABLES= Mininet_WiFi OpenNet  
/MISSING=INCLUDE  
/STATISTICS=VARIANCE.
```

Casos válidos = 7; casos con valor(es) perdido(s) = 0.

<i>Variable</i>	<i>N</i>	<i>Varianza</i>
Mininet_WiFi	7	20,95
OpenNet	7	10,95

Estadísticos Descriptivos SUS Score

Compare Means of Two Groups



Column A:

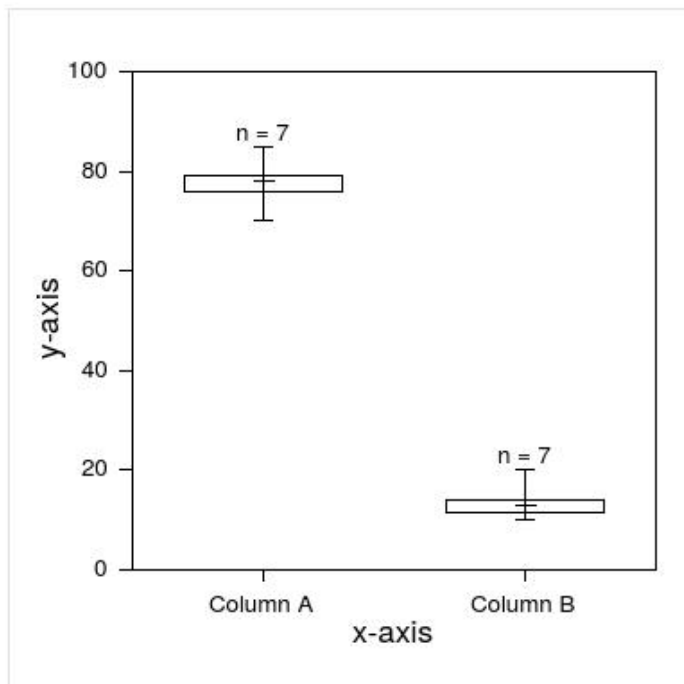
Mean = 77,5714
Std.Dev. = 4,57738
N = 7
Std.Err. = 1,73009
95% C.I. = 4,23337
Kurtosis = 1,4142
Skewness = -0,0652418

Column B:

Mean = 13,4286
Std.Dev. = 3,30944
N = 7
Std.Err. = 1,25085
95% C.I. = 3,06072
Kurtosis = 2,56795
Skewness = 1,48035

t-Test:

Student's t = 30,0448
P < 0.0001
P(corr.) < 0.0001



Column A:

Median = 78
25% = 76
75% = 79
N = 7
Min. = 70
Max. = 85

Column B:

Median = 13
25% = 11,5
75% = 14
N = 7
Min. = 10
Max. = 20

u-Test:

Mann-Whitney U = 49
z = 3,1305
P = 0,00174512
z(corr.) = 3,1374
P(corr.) = 0,00170454

Apéndice F

Scripts usados con OpenNet

Script wifiroaming.py OpenNet

```
#!/usr/bin/python
```

```
""" This example shows how to create an empty Mininet object (without a topology object) and add nodes to it manually. """ import sys import os
```

```
import mininet.net import mininet.node import mininet.cli import mininet.log import mininet.ns3
```

```
from mininet.net import Mininet, MininetWithControlNet from mininet.node import Controller, RemoteController from mininet.cli import CLI from mininet.log import setLogLevel, info from mininet.ns3 import *
```

```
import ns.core import ns.network import ns.wifi import ns.csma import ns.wimax import ns.uan import ns.netanim
```

```
from mininet.opennet import *
```

```
""" nodes is a list of node descriptions, each description contains several node attributes.
```

```
name: The node identifier. type: The character of this node, can be host or switch.
```

```
position: The position of this node. velocity: The velocity of this node, which will be used in host mobility model. The default mobility model is ns3::ConstantVelocityMobilityModel.
```

```
mobility: The mobility model of this node. setListPoritionAllocate() will take two parameters, MobilityHelper and ListPositionAllocate. After install the PositionAllocate in the MobilityHelper, setListoritionAllocate() will return the MobilityHelper. """
```

```
nodes = [ { 'name': 'h1', 'type': 'host', 'ip': '10.10.10.1', 'position': (0.0, 10.0, 0.0), 'velocity': (2.5, 0, 0) }, { 'name': 'h2', 'type': 'host', 'ip': '10.10.10.2', 'mobility': setListPositionAllocate( createMobilityHelper("ns3::RandomWalk2dMobilityModel",n0="Bounds", v0=ns.mobility.RectangleValue(ns.mobility.Rectangle(50,50))), createListPositionAllocate(x1=150,y1=30,z1=0)) }, { 'name': 's1', 'type': 'switch', 'position': (0.0, 0.0, 0.0) }, { 'name': 's2', 'type': 'switch', 'position': (120.0, 0.0, 0.0) }, { 'name': 's3', 'type': 'switch', 'position': (60.0, 60.0*(3**0.5), 0.0) }, { 'name': 's4', 'type': 'switch', 'position': (60.0, -60.0*(3**0.5), 0.0) }, { 'name': 's5', 'type': 'switch', 'position': (-120.0, 0.0, 0.0) }, { 'name': 's6', 'type': 'switch', 'position': (-60.0, 60.0*(3**0.5), 0.0) }, { 'name': 's7', 'type': 'switch', 'position': (-60.0, -60.0*(3**0.5), 0.0) }, ]
```

```
""" wifintfs is a list of wifi interface descriptions, each description contains some wifi attributes. """
```

```
wifintfs = [ {'nodename': 'h1', 'type': 'sta', 'channel': 1, 'ssid': 'ssid'}, {'nodename': 'h2', 'type': 'sta', 'channel': 11, 'ssid': 'ssid'}, {'nodename': 's1', 'type':
```

```

'ap', 'channel': 1, 'ssid': 'ssid'}, {'nodename': 's2', 'type': 'ap', 'channel': 6, 'ssid':
'ssid'}, {'nodename': 's3', 'type': 'ap', 'channel': 11, 'ssid': 'ssid'}, {'nodename':
's4', 'type': 'ap', 'channel': 11, 'ssid': 'ssid'}, {'nodename': 's5', 'type': 'ap',
'channel': 6, 'ssid': 'ssid'}, {'nodename': 's6', 'type': 'ap', 'channel': 11, 'ssid':
'ssid'}, {'nodename': 's7', 'type': 'ap', 'channel': 11, 'ssid': 'ssid'}, ]

```

""" links is a list of Ethernet links. """

```

links = [ {'nodename1': 's1', 'nodename2': 's2'}, {'nodename1': 's1', 'node-
name2': 's3'}, {'nodename1': 's1', 'nodename2': 's4'}, {'nodename1': 's1',
'nodename2': 's5'}, {'nodename1': 's1', 'nodename2': 's6'}, {'nodename1': 's1',
'nodename2': 's7'}, ]

```

```

def WifiNet():

```

""" Create an Wifi network and add nodes to it. """

```

net = Mininet()

```

```

info( '*** Adding controller\n' )

```

```

net.addController( 'c0', controller=RemoteController, ip='127.0.0.1', port=6633 )

```

""" Initialize the WifiSegment, please refer ns3.py """

```

wifi = WifiSegment(standard = ns.wifi.WIFI_PHY_STANDARD_80211g)

```

```

wifinodes = []

```

""" Initialize nodes """

```

for n in nodes:

```

""" Get attributes """

```

nodename = n.get('name', None)
nodetype = n.get('type', None)
nodemob = n.get('mobility', None)
nodepos = n.get('position', None)
nodevel = n.get('velocity', None)
nodeip = n.get('ip', None)

```

""" Assign the addfunc, please refer Mininet for more details about addHost and addSwitch """

```

if nodetype is 'host':

```

```

    addfunc = net.addHost

```

```

elif nodetype is 'switch':

```

```

    addfunc = net.addSwitch

```

```

else:

```

```

    addfunc = None

```

```

if nodename is None or addfunc is None:

```

```

    continue

```

```

""" Add the node into Mininet """
node = addfunc (nodename, ip=nodeip)

""" Set the mobility model """
mininet.ns3.setMobilityModel (node, nodemob)
if nodepos is not None:
    mininet.ns3.setPosition (node, nodepos[0], nodepos[1], nodepos[2])
if nodevel is not None:
    mininet.ns3.setVelocity (node, nodevel[0], nodevel[1], nodevel[2])

""" Append the node into wifinodes """
wifinodes.append (node)

""" Initialize Wifi Interfaces """
for wi in wifiintfs:

    """ Get attributes """
    winodename = wi.get('nodename', None)
    witype = wi.get('type', None)
    wichannel = wi.get('channel', None)
    wissid = wi.get('ssid', None)

    """ Assign the addfunc, please refer the WifiSegment in ns3.py """
    if witype is 'sta':
        addfunc = wifi.addSta
    elif witype is 'ap':
        addfunc = wifi.addAp
    else:
        addfunc = None
    if winodename is None or addfunc is None or wichannel is None:
        continue

    """ Get wifi node and add it to the TapBridge """
    node = getWifiNode (wifinodes, winodename)
    addfunc (node, wichannel, wissid)

""" Initialize Ehternet links between switches """
for cl in links:
    clnodename1 = cl.get('nodename1', None)
    clnodename2 = cl.get('nodename2', None)
    if clnodename1 is None or clnodename2 is None:
        continue
    clnode1 = getWifiNode (wifinodes, clnodename1)
    clnode2 = getWifiNode (wifinodes, clnodename2)
    if clnode1 is None or clnode2 is None:
        continue

```

```

net.addLink( clnode1, clnode2 )

""" Enable Pcap output"""
pcap = Pcap()
pcap.enable()
print pcap

""" Enable netanim output"""
anim = Netanim("/tmp/xml/wifi-wired-bridged4.xml", nodes)
print anim

""" Update node descriptions in the netanim """
for node in wifinodes:
    anim.UpdateNodeDescription (node.nsNode, str(node) + '-' + str(node.nsNode.GetId()))
    if isinstance(node, mininet.node.OVSSwitch):
        color = (0, 255, 0)
    elif isinstance(node, mininet.node.Host):
        color = (0, 0, 255)
    anim.UpdateNodeColor (node.nsNode, color[0], color[1], color[2])

""" Start the simulation """
info( '*** Starting network\n' )
net.start()
mininet.ns3.start()

info( 'Testing network connectivity\n' )
wifinodes[0].cmdPrint( 'ping 10.10.10.2 -c 3' )

CLI( net )

info( '*** Stopping network\n' )
mininet.ns3.stop()
info( '*** mininet.ns3.stop()\n' )
mininet.ns3.clear()
info( '*** mininet.ns3.clear()\n' )
net.stop()
info( '*** net.stop()\n' )

if name == 'main': setLogLevel( 'info' ) WifiNet()

```

Apéndice G

Scripts usados con Mininet-WiFi

Script Dos Host Mininet-WiFi

#!/usr/bin/python

```
""" Script created by VND - Visual Network Description (SDN version) """ from
mininet.net import Mininet from mininet.node import Controller, RemoteCon-
troller, OVSKernelSwitch, IVSSwitch, UserSwitch from mininet.link import Link,
TCLink from mininet.cli import CLI from mininet.log import setLogLevel

def topology():

    "Create a network."
    net = Mininet( controller=Controller, link=TCLink, switch=OVSKernelSwitch )

    print "*** Creating nodes"
    ap1 = net.addBaseStation( 'ap1', ssid= 'new-ssid', mode= 'g', channel= '1' )
    c2 = net.addController( 'c2' )
    sta3 = net.addStation( 'sta3', wlans=1, mac='00:02:00:00:00:03', ip='10.0.0.3/8' )
    sta4 = net.addStation( 'sta4', wlans=1, mac='00:02:00:00:00:04', ip='10.0.0.4/8' )

    print "*** Creating links"
    net.addLink(ap1, sta4)
    net.addLink(ap1, sta3)

    print "*** Starting network"
    net.build()
    c2.start()
    ap1.start( [c2] )

    print "*** Running CLI"
    CLI( net )

    print "*** Stopping network"
    net.stop()

if name == 'main': setLogLevel( 'info' ) topology()
```

Script 2 apsw Mininet-WiFi

#!/usr/bin/python

""" Script created by VND - Visual Network Description (SDN version) """ from mininet.net import Mininet from mininet.node import Controller, RemoteController, OVSKernelSwitch, IVSSwitch, UserSwitch from mininet.link import Link, TCLink from mininet.cli import CLI from mininet.log import setLogLevel

def topology():

 "Create a network."

 net = Mininet(controller=Controller, link=TCLink, switch=OVSKernelSwitch)

 print "*** Creating nodes"

 c1 = net.addController('c1')

 ap2 = net.addBaseStation('ap2', ssid= 'new-ssid', mode= 'g', channel= '1')

 ap3 = net.addBaseStation('ap3', ssid= 'new-ssid', mode= 'g', channel= '1')

 s4 = net.addSwitch('s4', listenPort=6675, mac='00:00:00:00:00:04')

 sta5 = net.addStation('sta5', wlans=1, mac='00:02:00:00:00:05', ip='10.0.0.5/8')

 sta6 = net.addStation('sta6', wlans=1, mac='00:02:00:00:00:06', ip='10.0.0.6/8')

 sta7 = net.addStation('sta7', wlans=1, mac='00:02:00:00:00:07', ip='10.0.0.7/8')

 sta8 = net.addStation('sta8', wlans=1, mac='00:02:00:00:00:08', ip='10.0.0.8/8')

 print "*** Creating links"

 net.addLink(sta8, ap3)

 net.addLink(sta7, ap3)

 net.addLink(sta6, ap2)

 net.addLink(sta5, ap2)

 net.addLink(ap3, s4)

 net.addLink(ap2, s4)

 print "*** Starting network"

 net.build()

 c1.start()

 s4.start([c1])

 print "*** Running CLI"

 CLI(net)

 print "*** Stopping network"

 net.stop()

if name == 'main': setLogLevel('info') topology()