UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR

INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO DE INVESTIGACIÓN

EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional

Autor:

D. Rafael Cano Parra

Tutores:

Dr. D. Miguel Luis Bote Lorenzo

Dr. D. Eduardo Gómez Sánchez

Valladolid, 11 de diciembre de 2012

TÍTULO:	**Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional**
AUTOR:	**D. Rafael Cano Parra**
TUTORES:	**Dr. D. Miguel Luis Bote Lorenzo** **Dr. D. Eduardo Gómez Sánchez**
DEPARTAMENTO:	**Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

Tribunal

PRESIDENTE:	**Dr. D. Ioannis Dimitriadis Damoulis**
VOCAL:	**Dr. D. Juan Ignacio Asensio Pérez**
SECRETARIO:	**Dr. D. Jesús María Vegas Hernández**
FECHA:	**11 de diciembre de 2012**
CALIFICACIÓN:	

Resumen del TFM

La simulación es una técnica muy utilizada e importante en ingeniería, y en especial en telemática, porque permite modelar, observar y analizar sistemas de redes de forma barata sin necesidad de montar un sistema real. La simulación de barrido de parámetros de redes telemáticas puede ayudar a conocer el funcionamiento de protocolos en distintas condiciones de la red y en distintos tamaños de redes, por lo que supone un avance significativo como utilidad en educación. Sin embargo, las simulaciones de barrido de parámetros son costosas computacionalmente, y para conseguir un tiempo de respuesta bajo adecuado a un contexto educativo, hace falta paralelizar las simulaciones en distintos recursos. Los simuladores actuales no proporcionan esta característica, pero existe una propuesta que lo resuelve, el Entorno de Simulación de Red Distribuido (DNSE), utilizando las tecnologías del *grid* computacional y de los servicios WSRF, que presentan limitaciones. La principal es que no permite el escalado automático en función de la demanda, sino que requiere la intervención manual de un administrador. Además, las evaluaciones del prototipo indicaron limitaciones en la usabilidad, fundamentalmente debidas a la interfaz gráfica de usuario. Estas limitaciones se pueden superar con la computación en nube, los servicios REST y el uso de tecnologías Web para la interfaz, por lo que se propone el DNSE3 como una nueva versión que implementa estas tecnologías. El DNSE3 consta de ocho servicios y permite escalar horizontalmente los servicios de simulación, que usan el simulador de red ns-3, y analizar gráficamente los resultados en una interfaz Web. Finalmente, se propone una evaluación basada en el análisis de características siguiendo las aproximaciones

de modo de detección y de caso de estudio.

Palabras clave

Apoyo al aprendizaje, computación en nube, DNSE, ns-3, servicios REST, simulador de red.

Abstract

Simulation is an important technique widely used in engineering, especially in telematics, because it allows modeling, observing and analyzing network systems cheaply without the need to mount a real system. The parameter sweep simulation of computer networks may help to understand the functioning of protocols in different network conditions and different sizes of networks, which represents a significant step as an utility in education. However, the parameter sweep simulations are computationally expensive, and to get a proper response time under an educational context, we need to parallelize the simulations in different resources. The current simulators do not provide this feature, but there is a proposal that solves it, the Distributed Network Simulator Environment (DNSE), using the technologies of grid computing and WSRF services, which have limitations. The main issue is that does not allow automatic scaling based on demand, but requires manual intervention from an administrator. Furthermore, the prototype evaluations indicated limitations in usability, due basically to the graphical user interface. These limitations can be overcome with cloud computing, REST services and the use of Web technologies to the interface; then we propose DNSE3 as a new version that implementing these technologies. DNSE3 is composed by eight services and allows to scale horizontally the simulation services, which use the ns-3 network simulator, and analyze the results graphically in a Web interface. Finally, we propose an evaluation based on the featured analysis following the detection mode and the case study approaches.

Keywords

Cloud computing, DNSE, e-learning, network simulator, ns-3, REST services.

Agradecimientos

En primer lugar, quisiera agradecer la inmensa paciencia que han tenido conmigo mis dos tutores, Miguel y Edu, durante toda la realización de este Trabajo Fin de Máster. Muchos han sido los motivos que les he dado para "sacarles de sus casillas", pero aún así nunca han dejado de confiar en mí y sólo he recibido grandes consejos, buenas palabras y muchos ánimos. ¡Gracias!

También quisiera agradecer de forma especial todo el apoyo recibido por Yannis y Asen, que han prestado mucha parte de su limitado tiempo para solucionar mis problemas (¡y muy grandes!) y aportarme consejos e ideas muy valiosas. Espero seguir recibiéndolas en el futuro ;-)

Luego extender mi agradecimiento a todas esas personas que forman el grupo de investigación GSIC/EMIC[1]. Me recibieron con los brazos abiertos para que administrase sus redes y ordenadores allá en el año 2010 y han hecho realidad uno de mis sueños: formar parte de esa gran familia de investigadores de calidad y excelencia competitiva. Es un honor seguir trabajando con todos vosotros.

También quiero agradecer el apoyo que me han prestado todos mis amigos: para los que han estado a mi lado durante el día a día aguantándome en riguroso directo, para los que han estado en la lejanía leyendo mis momentos de desahogo, y para los nuevos amigos que ha traído consigo el MUI-TIC. A todos ellos (perdonadme que no os nombre, pero sois taaaantos...), muchísimas gracias.

Y para finalizar, quizás uno de los agradecimientos más grandes y de todo corazón: a mi FAMILIA. Son muchos los que *forman* mi familia, pero no tantos los que *son* mi familia. A todos esos, que bien saben quienes son, un besazo muy grande, os lo merecéis.

Y en especial a mi mamá Eva, a mi papá Evelio, a mi hermana Yolanda, a mis tías Marivi y Trini, a mi prima Diana, a mis tíos Antonio y Gerardo, a mi *cuñado* Alberto... y, aunque ya no estés conmigo, a mi abuela Avelina. Yo no sé cómo lo hacéis, pero a vuestro lado me siento tan arropado que todos mis problemas, dudas... mágicamente desaparecen. Sois el pilar básico de mi existencia, ¡qué haría sin vosotros! Os quiero y os querré muchísimo.

[1]Todos estos y más: https://www.gsic.uva.es/miembros.php

Índice general

1. Introducción **1**
1.1. Introducción y motivación 1
1.2. Objetivos del trabajo 3
1.3. Metodología de investigación 6
1.4. Organización del documento 7

2. Estudio del Estado del Arte **9**
2.1. Simulación de redes 9
2.1.1. Definición 9
2.1.2. Ejemplos de simuladores de redes 10
2.1.3. Paralelización y distribución 13
2.1.4. Aplicación en la educación 13
2.2. Computación en nube 14
2.2.1. Definición 14
2.2.2. Paradigmas de computación distribuida 15
2.2.3. Clasificación 16
2.2.4. Modelos de despliegue 18
2.2.5. Ventajas e inconvenientes 19
2.2.6. Aplicación en la educación 21
2.2.7. Aplicación en la simulación en educación 23

3. Propuesta y Diseño de Aplicación **25**
3.1. Introducción 25
3.2. Requisitos de la aplicación 27
3.3. Arquitectura del sistema 29
3.4. Servicios del sistema 34
3.4.1. Servicio de Almacenamiento 34
3.4.2. Servicio de Colas 37
3.4.3. Servicio de Escalado 39
3.4.4. Servicio de Estadística 41
3.4.5. Servicio de Informe 43
3.4.6. Servicio de Monitorización 46
3.4.7. Servicio de Orquestación 49
3.4.8. Servicio de Simulación 58
3.5. Selección de tecnologías 60
3.5.1. REST y RESTful 61

3.5.2. Infraestructura de nube computacional 62
3.5.3. Simulador de red 64
3.6. Plan de implementación 65

4. Discusión y Propuesta de Evaluación 69
4.1. Discusión 69
4.2. Propuesta de evaluación 74
4.2.1. Modo de detección 75
4.2.2. Caso de estudio 76

5. Conclusiones y Trabajos Futuros 81
5.1. Conclusiones 81
5.2. Trabajos futuros 84

6. Referencias 87

A. Glosario de Términos 91

Índice de figuras

1.1. Esquema general del trabajo de investigación en el que se recoge el contexto, los objetivos global y parciales, la contribución y la propuesta de evaluación. 5

3.1. Diagrama de casos de uso de la aplicación DNSE3. 31
3.2. Diagrama de arquitectura de servicios de la aplicación DNSE3. 33
3.3. Diagrama de casos de uso de los escenarios de uso del Servicio de Almacenamiento. 35
3.4. Diagrama de modelo de dominio del Servicio de Almacenamiento. 36
3.5. Diagrama de recursos del Servicio de Almacenamiento. 36
3.6. Diagrama de casos de uso de los escenarios de uso del Servicio de Colas. 38
3.7. Diagrama de modelo de dominio del Servicio de Colas. 38
3.8. Diagrama de recursos del Servicio de Colas. 38
3.9. Diagrama de casos de uso de los escenarios de uso del Servicio de Escalado. 40
3.10. Diagrama de modelo de dominio del Servicio de Escalado. 40
3.11. Diagrama de recursos del Servicio de Escalado. 41
3.12. Diagrama de casos de uso de los escenarios de uso del Servicio de Estadística. 42
3.13. Diagrama de modelo de dominio del Servicio de Estadística. 42
3.14. Diagrama de recursos del Servicio de Estadística. 43
3.15. Diagrama de casos de uso de los escenarios de uso del Servicio de Informe. 44
3.16. Diagrama de modelo de dominio del Servicio de Informe. 45
3.17. Diagrama de recursos del Servicio de Informe. 45
3.18. Diagrama de casos de uso de los escenarios de uso del Servicio de Monitorización. 47
3.19. Diagrama de modelo de dominio del Servicio de Monitorización. 47
3.20. Diagrama de recursos del Servicio de Monitorización. 48
3.21. Diagrama de casos de uso de los escenarios de uso de la gestión de usuarios del Servicio de Orquestación. 51
3.22. Diagrama de casos de uso de los escenarios de uso de la gestión de sesiones del Servicio de Orquestación. 51
3.23. Diagrama de casos de uso de los escenarios de uso de la gestión de paquetes de simulación del Servicio de Orquestación. 52
3.24. Diagrama de casos de uso de los escenarios de uso de la gestión de simulaciones del Servicio de Orquestación. 53

3.25. Diagrama de casos de uso de los escenarios de uso de la gestión de visualizaciones del Servicio de Orquestación. 54
3.26. Diagrama de casos de uso de los escenarios de uso de la gestión de informes del Servicio de Orquestación. 55
3.27. Diagrama de casos de uso de los escenarios de uso de la gestión de configuración del Servicio de Orquestación. 55
3.28. Diagrama de modelo de dominio del Servicio de Orquestación. 56
3.29. Diagrama de recursos del Servicio de Orquestación. 57
3.30. Diagrama de casos de uso de los escenarios de uso del Servicio de Simulación. 59
3.31. Diagrama de modelo de dominio del Servicio de Simulación. 59
3.32. Diagrama de recursos del Servicio de Simulación. 60
3.33. Diagrama de componentes para un ejemplo de despliegue de la aplicación DNSE3. 68

Índice de tablas

2.1. Comparativa de las características más relevantes de los simuladores de red. 13
2.2. Clasificación de las oportunidades y limitaciones de la nube computacional. 21

3.1. Estimación temporal de la implementación de la aplicación DNSE3. . . . 67

Capítulo 1

Introducción

1.1. Introducción y motivación

En muchas ocasiones resulta inviable poder estudiar el comportamiento de sistemas, aplicaciones y plataformas en entornos reales, por ejemplo, por el elevado gasto económico que conlleva o por las dificultades técnicas debidas a sus grandes dimensiones. Para estos casos se utiliza la simulación, una de las técnicas más ampliamente utilizadas en la actualidad en multitud de sectores, como por ejemplo las finanzas, la industria, el militar o la educación. La simulación permite el modelado de objetos reales o hipotéticos en un ordenador para su posterior estudio [1, 2].

En el sector de la educación, la simulación ha permitido que los estudiantes puedan aplicar los conceptos teóricos adquiridos en un escenario práctico, apoyando el aprendizaje. Por ejemplo, los estudiantes de ingeniería de industriales pueden modelar una pieza de un sistema y estudiar el comportamiento de su maleabilidad cuando se le aplican diferentes fuerzas. Las simulaciones están presentes en multitud de áreas dentro del sector de la educación, entre ellas las ingenierías relacionadas con las Tecnologías de la Información y las Comunicaciones (TIC). Y dentro de las TIC, una de las simulaciones que más relevancia tiene en un contexto educativo es la simulación de redes telemáticas para el análisis del rendimiento de los sistemas de comunicación [3, 4].

La simulación de redes telemáticas es la metodología que implementa una red en un ordenador para su posterior evaluación. Permite probar escenarios que son difíciles o costosos de simular en entornos reales [1]. También es usada para la evaluación de nuevas arquitecturas de comunicaciones y protocolos de red en un entorno controlado y reproducible. Por ello, permite modelar una red de ordenadores arbitraria en la que se especifican tanto el comportamiento de los nodos como los canales de comunicación que utilizan. Además, se permiten variar las condiciones de la topología de la red a través de la configuración de un conjunto de parámetros de simulación [3]. Por ejemplo, se puede estudiar el comportamiento de un protocolo de enrutamiento de red realizando su simulación sobre varias topologías en las que se van modificando su conjunto de parámetros para hallar aquel con el que la red tiene un menor tiempo de transmisión.

En un contexto educativo puede tener gran interés la realización de muchas simulaciones de redes telemáticas realizando un barrido de parámetros. El barrido de parámetros aplica los valores de un rango determinado a los parámetros de la red, generando un con-

junto de simulaciones individuales. Esto permite que los estudiantes puedan comprobar a través de la comparación qué conjunto de parámetros es el más adecuado para lograr un objetivo señalado por el profesor [5, 6, 7].

Una técnica para la ejecución de una simulación de barrido de parámetros es la división de la misma en tantas simulaciones individuales como opciones de conjuntos de parámetros posibles, y la ejecución de cada una de las simulaciones individuales por separado [5, 8]. Esta técnica puede requerir mucho tiempo para la ejecución de una simulación de barrido de parámetros, además de consumir muchos recursos computacionales. El uso de computación distribuida ofrece a este método una ejecución concurrente de las simulaciones individuales, pudiendo mejorar la escalabilidad tanto en términos de tamaño de red como rapidez de ejecución [8].

En la literatura están presentes una multitud de simuladores de redes telemáticas de software libre que son usados en entornos académicos e investigadores [3, 1]. Los más relevantes son el ns-2[1] y su predecesor ns-3[2], OMNeT++[3], J-Sim[4], SimPy[5] y JiST/SWANS[6]. Sin embargo, aunque algunos ofrecen la capacidad de ejecutar simulaciones en paralelo de forma interna para modelos de redes que permiten la división de la red a nivel de paquete, no son capaces de ejecutar la división resultante de una simulación de barrido de parámetros en un entorno de computación distribuida [8].

El Entorno de Simulación de Red Distribuido (*Distributed Network Simulator Environment*, DNSE) es una propuesta de Bote et al. [5] que resuelve este problema. Es un entorno compuesto de varios servicios cuya función principal es la ejecución de simulaciones individuales y de barrido de parámetros usando el simulador de red ns-2. Utiliza la tecnología del *grid* computacional para distribuir las simulaciones en varias instancias del servicio de simulación entre las diferentes máquinas que lo forman. También usa servicios WSRF (*Web Service Resource Framework*) para la comunicación e interacción interna entre los servicios que lo componen. El DNSE consigue que la ejecución de una simulación de barrido de parámetros se adecúe al tiempo disponible para una sesión de laboratorio de redes telemáticas (normalmente de dos horas).

Sin embargo, el DNSE presenta algunas limitaciones debido al uso de estas tecnologías. La primera es que el *grid* computacional no permite realizar una escalabilidad dinámica y automática de sus recursos computacionales. Es necesaria la intervención de un administrador que aumente o disminuya el número de máquinas y servicios disponibles en el *grid* computacional según las necesidades de la aplicación [9]. La segunda es que los servicios WSRF, por su composición, no proporcionan un mecanismo sencillo para la monitorización del estado de los servicios, dificultando su sondeo para conocer las necesidades de escalado.

Las limitaciones anteriores pueden superarse con el uso de tecnologías de computación en la nube y servicios REST (*Representational State Transfer*). La nube computacional permite realizar una escalabilidad dinámica y automática de forma sencilla y flexible de sus recursos computacionales. De esta forma, habilita la capacidad de manejar

[1] `http://www.isi.edu/nsnam/ns/`
[2] `http://www.nsnam.org/`
[3] `http://www.omnetpp.org/`
[4] `https://sites.google.com/site/jsimofficial/`
[5] `http://simpy.sourceforge.net/`
[6] `http://jist.ece.cornell.edu/`

de forma automática por la aplicación el incremento y disminución de los recursos según la demanda, pudiendo realizar más trabajos de forma simultánea en paralelo [10]. Para las simulaciones de barrido de parámetros, la aplicación se puede autoescalar de forma horizontal aumentando el número de servicios de simulación de forma dinámica en un entorno de nube computacional, para que la ejecución de las simulaciones individuales que se generan se realicen en menos tiempo. Posteriormente, puede disminuir el número de servicios de simulación para liberar los recursos computacionales que se habían ocupado. Así, las tecnologías de computación en nube permiten una computación distribuida con mejor flexibilidad que la que puede ofrecer el *grid* computacional [2].

A su vez, el uso de servicios REST para la comunicación entre los diferentes servicios que componen la aplicación, permiten que cada uno de los servicios tenga una interfaz programática pública, invocable a través del estándar HTTP, el cual proporciona principalmente cuatro métodos de invocación (consulta, creación, modificación y eliminación) para la realización de tareas simples, pero potentes, sobre los recursos de cada servicio. En estas invocaciones se realiza un intercambio de representaciones que permiten la interoperabilidad entre los servicios, así como con otros agentes externos, lo cual facilita la reutilización de los servicios de la aplicación para otros entornos [11, 12].

En este trabajo de investigación se propone una nueva versión del DNSE usando la computación en nube y los servicios REST. Se la ha denominado Entorno de Simulación de Red Distribuido III (DNSE3) y permite, igualmente, la realización de simulaciones de redes telemáticas de barrido de parámetros, entre otras funciones, enfocada para el apoyo al aprendizaje en las sesiones de laboratorio.

Así, se aplicará el beneficio de escalabilidad dinámica y automática de la nube computacional para la realización de simulaciones en un entorno distribuido que sea capaz de adaptarse de forma autónoma a las necesidades del mismo, para conseguir un ahorro temporal y poder ajustar los tiempos de simulación (para simulaciones de barridos de parámetros) a una sesión de prácticas de laboratorio [8, 13]. La aplicación estará dividida en ocho servicios que realizarán ciertas tareas de forma autónoma, eficiente, efectiva y en paralelo [10, 14]. Los servicios estarán diseñados siguiendo el modelo arquitectural REST que facilitará la comunicación e invocación entre ellos [11].

Por otro lado, se mejorará la interfaz de usuario para que, además de ofrecer una visión adaptada a las necesidades pedagógicas, se aprovechen las ventajas de la Web tanto para la interacción con la aplicación de forma ubicua y multiplataforma [14, 15], como para la facilidad de comunicación entre los diferentes servicios de la aplicación DNSE3 a través de servicios Web [16].

Adicionalmente, debido a las restricciones temporales de este Trabajo Fin de Máster, sólo se ha realizado en análisis y diseño de la propuesta de aplicación DNSE3, quedando pendiente la implementación de la misma y la consiguiente evaluación a partir del prototipo. A pesar de ello, se propone tanto un plan de implementación como una propuesta de evaluación basada en el análisis de características.

1.2. Objetivos del trabajo

En la sección anterior se han identificado y descrito dos problemas relevantes que se presentan en los entornos de simulación de redes telemáticas cuando son utilizados como

apoyo al aprendizaje en un entorno universitario. De acuerdo con ello, en este trabajo se persigue el siguiente objetivo:

La propuesta de un entorno de simulación de redes telemáticas basado en tecnologías de computación en nube y servicios REST que permita la realización de simulaciones de barrido de parámetros en un contexto educativo.

Para lograr este objetivo, se proponen una serie de objetivos parciales que ayuden y guíen la consecución del objetivo principal, que se describen a continuación:

- *Estudiar las ventajas y los inconvenientes de la nube computacional en general y específicamente en entornos educativos.*
  Para conseguir este objetivo se realizará un estudio del estado del arte del paradigma de la computación en la nube, señalando las ventajas y los inconvenientes identificadas por diferentes autores en el uso en general, y las que se aplican y surgen en un entorno educativo.

- *Estudiar las ventajas y los inconvenientes de los simuladores de red en general y específicamente en entornos educativos.*
  Para conseguir este objetivo se realizará un estudio del estado del arte de los simuladores de redes telemáticas, señalando las ventajas y los inconvenientes identificados por diferentes autores en el uso en general, y las que se aplican y surgen en un entorno educativo.

- *Ilustrar la potencialidad de la nube computacional en un contexto educativo mediante la propuesta de una aplicación concreta e identificar los problemas no cerrados.*
  Partiendo de la propuesta realizada en este trabajo de investigación, ilustrar las oportunidades de la nube computacional que podrían aplicarse en otras aplicaciones en un contexto educativo, e identificar los problemas abiertos que continúan sin resolverse.

- *Proponer la evaluación del entorno de simulación de red distribuido aplicado en un contexto educativo para el apoyo al aprendizaje.*
  Se propondrá una evaluación basada en el análisis de las características, aplicando dos aproximaciones de evaluación: una evaluación intermedia aplicando el modo de detección en el que se analizan los requisitos software de la propuesta, y una evaluación final aplicando un caso de estudio real en el que está involucrada la propuesta del entorno de simulación de red distribuido en un contexto educativo real para el apoyo al aprendizaje.

El objetivo global y los objetivos parciales, junto con el contexto de investigación, la contribución y la propuesta de evaluación de este trabajo se resumen de manera esquemática en la Figura 1.1.

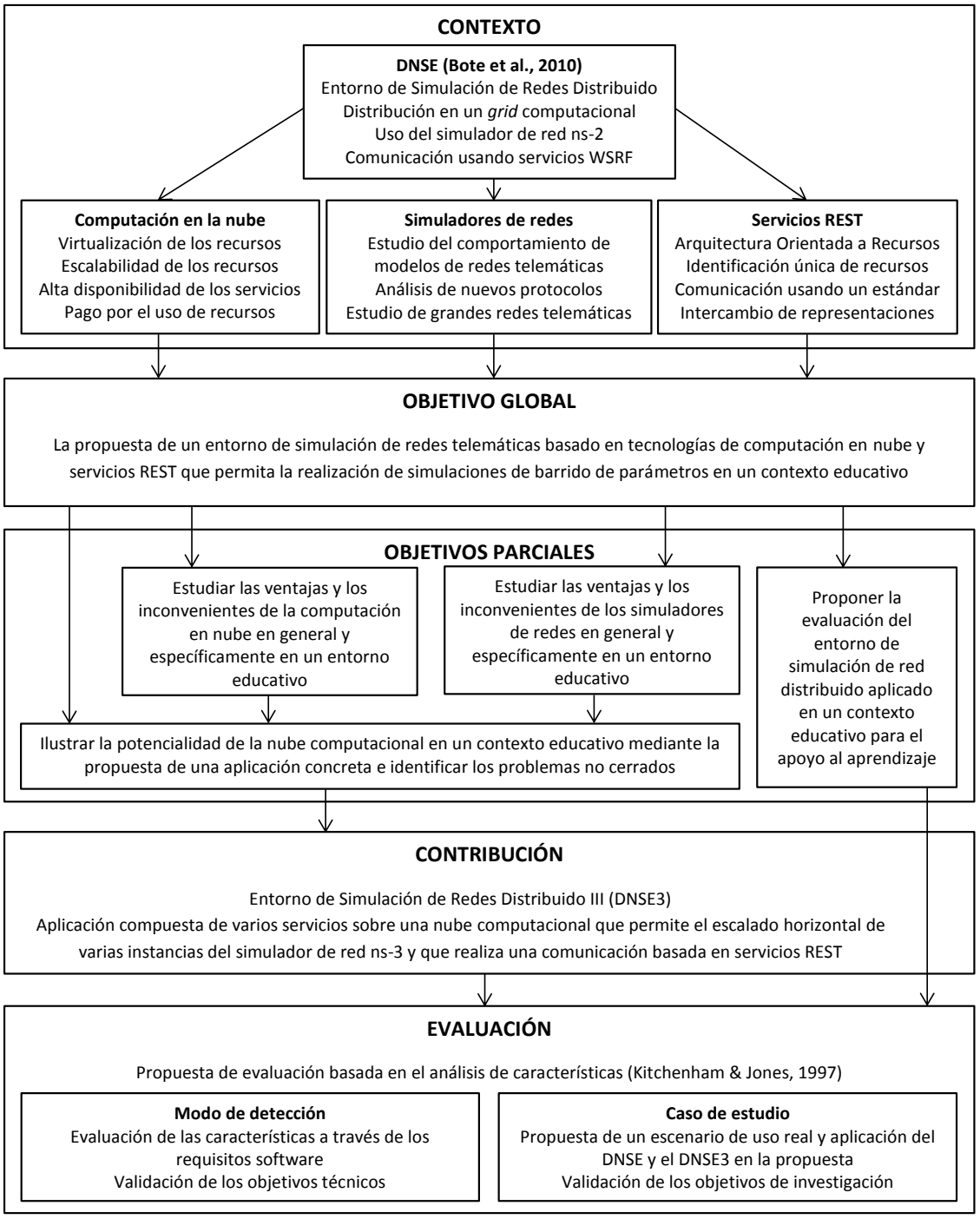

Figura 1.1: Esquema general del trabajo de investigación en el que se recoge el contexto, los objetivos global y parciales, la contribución y la propuesta de evaluación.

1.3. Metodología de investigación

La metodología de investigación aplicada para la consecución de los objetivos de este trabajo es el método de ingeniería propuesto por Adrion [17]. Este método está compuesto de cuatro etapas: observar las soluciones existentes, proponer una solución mejor, desarrollar la nueva solución y evaluar la nueva solución. Estas cuatro etapas se han ido aplicando de forma iterativa a lo largo del trabajo para poder refinar progresivamente la solución presentada.

El método de ingeniería puede ser el más adecuado para la realización de este trabajo de investigación debido a que el objetivo principal que se persigue es el desarrollo de una solución nueva para realizar la simulación de red en entornos de apoyo al aprendizaje que haga uso de la nube computacional. La solución propuesta debe ser novedosa, por lo que ha sido necesario realizar una observación de las soluciones existentes. El desarrollo de la nueva solución no ha llegado a producir una pieza de software que pueda evaluarse como tal, pero se propone una evaluación de dicha pieza de software para que, una vez desarrollada, pueda aplicarse.

Una descripción más detallada de cada una de las cuatro fases del método de ingeniería es la siguiente:

1. **Observar las soluciones existentes.** La primera etapa de este método tiene como objetivo detectar los problemas que se van a abordar en el trabajo de investigación a partir del estudio de las propuestas existentes. Por ello, se ha realizado una exploración de la literatura existente relacionada con la nube computacional y los simuladores de redes telemáticas aplicados en el apoyo al aprendizaje. Esta exploración ha permitido identificar los problemas que ya han sido descritos en la introducción: la distribución de las ejecuciones de las simulaciones no permite un aumento y disminución de los recursos disponibles de forma dinámica y no se tiene una interfaz gráfica de usuario integrada que permita el análisis e interpretación de la gran cantidad de resultados generados.

2. **Proponer una solución mejor.** En la segunda etapa del método de ingeniería se debe hacer una propuesta de solución que supere las limitaciones detectadas en las soluciones existentes. Para ello, antes es necesario detallar y estudiar las posibles aproximaciones que permitan resolver dichas limitaciones a través de una captura de requisitos. Dichos requisitos son heredados, en parte, de la aplicación previa propuesta por Bote et al. [5] y obtenidos directamente de los profesores y alumnos que hacen uso de este entorno de simulación de red distribuido. Una vez definidos, se propone una arquitectura software orientada a servicios de la aplicación, que se adapte a la nube computacional, y se realiza el estudio con el análisis y diseño de cada servicio, usando una metodología orientado a recursos.

3. **Desarrollar la nueva solución.** En esta etapa se debe implementar la solución propuesta. A partir del estudio de la etapa anterior se implementará la arquitectura del nuevo sistema que pueda superar las limitaciones de las soluciones existentes. Puesto que esta parte no estará dentro del tiempo de desarrollo de este trabajo de investigación, se realiza una propuesta del plan de implementación que podrá llevarse a cabo para desarrollar la nueva solución.

4. **Evaluar la nueva solución.** En la última etapa del método de ingeniería es necesario mostrar que la solución desarrollada supera los problemas que fueron detectados en la primera etapa de esta metodología. Puesto que el desarrollo de un prototipo del sistema no se ha podido llevar a cabo, se realizará una evaluación teórica del sistema. Para ello, inicialmente se realizará la evaluación basándose en las características de la aplicación, y posteriormente se propondrá un escenario de uso real en el que podrá utilizarse.

No obstante, se ha de señalar que esta metodología no sigue un proceso totalmente secuencial, puesto que algunas fases se pueden ir desarrollando de forma paralela, pero en cambio sí que sigue un proceso iterativo en el que tras finalizar la última etapa se vuelve a la primera para refinar los resultados obtenidos.

1.4. Organización del documento

En el presente Capítulo 1 de **Introducción** se presenta inicialmente el marco teórico y la motivación del trabajo de investigación, para luego describir el objetivo principal y los objetivos parciales, la metodología de investigación y la presente organización del documento.

En el Capítulo 2 de **Estudio del Estado del Arte** se realiza un estudio del estado del arte de la nube computacional describiendo sus características básicas, sus oportunidades y sus limitaciones. También se enumeran y describen las soluciones actuales de simuladores de redes telemáticas de software libre. Posteriormente, en cada uno de ellos, se relacionan con su aplicación en el apoyo al aprendizaje o *e-learning*.

En el Capítulo 3 de **Propuesta y Diseño de Aplicación** se describe inicialmente el contexto de la propuesta de un entorno de simulación de redes telemáticas distribuido para el apoyo al aprendizaje y basado en una nube computacional, que prentende solucionar las dos limitaciones identificadas: el DNSE3. Luego se enumeran los requisitos de la aplicación y se presenta la arquitectura software orientada a servicios definida, en el que se identifican ocho servicios. Estos servicios se describen por separado posteriormente, realizando una elicitación de requisitos software para cada uno de ellos, y aplicando la metodología de análisis y diseño orientado a recursos. Finalmente se realiza una selección de las tecnologías que va a utilizar el DNSE3 y se propone un plan de implementación.

En el Capítulo 4 de **Discusión y Propuesta de Evaluación** se reflexiona sobre la aplicación de las oportunidades de la nube computacional y la influencia de las limitaciones para la aplicación para el apoyo al aprendizaje propuesta de simulación de redes telemáticas DNSE3. Para la evaluación de la propuesta para comprobar que se han resuelto las limitaciones identificadas, y puesto que no ha sido desarrollada, se realiza una evaluación basada en el análisis de las características de la aplicación y se propone un escenario de uso real donde se utiliza la aplicación por los profesores y los alumnos en una clase de prácticas de laboratorio.

En el Capítulo 5 de **Conclusiones y Trabajos Futuros** se presentan las principales conclusiones extraídas del trabajo de investigación, haciendo un breve repaso de las aportaciones de la misma, junto con una discusión de las posibles líneas de trabajo futuro.

Finalmente, en el Capítulo 6 de **Referencias** se enumeran las referencias citadas durante todo el trabajo de investigación, por orden de aparición en el mismo.

Adicionalmente, se presenta un Apéndice A de **Glosario de Términos** en el que se describen las siglas, abreviaturas y conceptos más relevantes que puedan resultar de interés a la hora de comprender el trabajo de investigación por el lector.

Capítulo 2

Estudio del Estado del Arte

En este capítulo se presenta un estudio del estado del arte de los tres tópicos tratados en este trabajo de investigación: la nube computacional (*cloud computing*), las simulaciones de redes telemáticas (*network simulations*) y el apoyo al aprendizaje (*e-learning*).

En la primera sección se realiza un estudio del estado del arte sobre la simulación de redes telemáticas y su aplicación en la educación. En ella se define qué son y las características principales, los ejemplos más relevantes de simuladores de redes, así como las soluciones para la paralelización y distribución de las simulaciones. Posteriormente se estudian las ventajas e inconvenientes en un contexto educativo.

En la segunda sección se realiza un estudio del estado del arte sobre la computación en nube y su aplicación tanto para la educación como para la simulación. Así, se define qué es la computación en nube, sus orígenes, una clasificación por tipo de abstracción y modelo de despliegue, las ventajas e inconvenientes en general, y los mismos para su aplicación en contextos educativos y en simuladores.

2.1. Simulación de redes

2.1.1. Definición

Tal y como afirman Weingartner et al. [3], *la simulación de redes es sin duda la metodología de evaluación más predominante en el área de las redes telemáticas*. Una red telemática estará compuesta por la descripción del comportamiento de los nodos que la componen y de sus canales de comunicación, así como de un conjunto de parámetros que permitan configurar de forma dinámica el comportamiento de los nodos y los canales de comunicación [3].

La ejecución de simulaciones de redes telemáticas tiene dos objetivos principales [8]: el primero es poder observar el comportamiento de protocolos y otros mecanismos de comunicación aplicando diversas condiciones de red, y el segundo es entender el efecto real de un nuevo protocolo, mecanismo, servicio de red, ataque o aplicación previamente a ser aplicado en una red real, sobre todo cuando será desplegado en una gran red como Internet.

Adicionalmente, Wang et al. [18] identifican seis características que se presentan como ventajas en los simuladores de redes frente a la utilización de escenarios reales:

1. Para escenarios complejos, que en redes reales no podrían reproducirse, los resultados obtenidos tras su simulación son fiables.

2. La función de predicción de redes es mucho mejor que cualquier otro método utilizado en escenarios reales.

3. Tiene un amplio rango de uso para la optimización y la expansión de redes o diseños de nuevas redes.

4. El coste de uso inicial es bajo, permitiendo el uso a varias personas sin coste adicional, y pudiendo reutilizar los modelos ya creados para otras simulaciones.

5. Gran flexibilidad, pudiendo modificar los valores del modelos para observar el comportamiento de protocolos y entender la red, de forma que se puedan comprobar por comparación las ventajas e inconvenientes de varias estrategias.

6. Se puede realizar de forma sencilla la reproducción y el análisis de los resultados, pudiendo además reproducir circunstancias especiales que no son posibles en redes reales.

2.1.2. Ejemplos de simuladores de redes

Puesto que hay una gran diversidad de implementaciones de simuladores de redes, para este estudio se seleccionarán y describirán aquellos más representativos [1, 3]. De entre ellos, existen soluciones de software libre, como es el caso de ns-2, ns-3, OMNeT++, J-Sim, SimPy y JiST/SWANS, y otras soluciones de software propietario como son OPNET Modeler, NetSim y Qualnet.

El simulador de red ns-2[1] es una herramienta que data de 1995 y que se basa en el paradigma de la simulación basada en eventos discretos, en el que el tiempo de ejecución de una simulación es proporcional al número de eventos que se quieren procesar [8]. Está escrito en el lenguaje de programación C++, que también se utiliza para modelar el comportamiento de los nodos en la simulación, y además realiza la integración de *scripts* TCL (*Tool Command Language*), modificables de forma rápida e interactiva, para controlar el proceso de simulación y aspectos como la topología de la red [3]. Esto permite que se puedan realizar cambios de forma dinámica en la configuración de las simulaciones sin tener que volver a recompilar el modelado de la red. Permite la simulación del protocolo TCP, de enrutamiento y de *multicasting* sobre redes fijas e inalámbricas. Destacar que en entornos educativos su uso ha sido tan generalizado que se ha convertido en el estándar *de facto* de los simuladores de redes [3, 8]. Pero este simulador tiene dos limitaciones importantes: la cantidad elevada de memoria que precisa cuando el modelo de simulación es muy grande (con muchos nodos y canales), y la cantidad de tiempo de computación necesario para terminar la ejecución de una simulación es proporcional al tamaño del modelo de simulación utilizado. Aunque estas limitaciones pueden resolverse parcialmente utilizando la paralelización de las ejecuciones de las simulaciones en varias instancias del simulador de red, sólo es aplicable en aquellos modelos que permiten ser divididos en tareas más pequeñas y que pueden ser ejecutadas de manera individual [8].

[1] `http://www.isi.edu/nsnam/ns/`

La siguiente versión de este simulador es el ns-3[2], publicado en 2006, que basa muchas de sus características en su antecesor, como son la simulación basada en eventos discretos y el tipo de simulaciones que permite realizar. Está reescrito completamente en C++, inclusive la descripción de los modelos y los procesos de simulación que ya no precisan el uso de los *scripts* TCL (los entornos actuales ya no presentan la limitación temporal de compilación de los modelos en el lenguaje C++). Por tanto, su interfaz de la aplicación ya no es compatible con el ns-2, aunque permite la portabilidad de los escenarios a la nueva interfaz en C++ [3]. Los resultados de las simulaciones se generan en ficheros de trazas en paquetes *pcap*. Estos paquetes pueden ser utilizados por otras herramientas para su visualización gráfica y análisis, como por ejemplo Wireshark[3] [1]. El simulador de red ns-3 además resuelve las principales limitaciones de su antecesor sobre el uso elevado de memoria y tiempo de computación para grandes escenarios de simulación. Para ello hace uso de la paralelización, dividiendo internamente el escenario en pequeñas tareas individuales que ejecutará de manera paralela, pero no permite la distribución de las mismas en distintas máquinas [3].

OMNeT++[4] [19] es otro simulador de red multiprocesos escrito en C++ y basado en eventos discretos utilizado para modelar redes de comunicaciones, multiprocesadores y otros sistemas distribuidos o paralelos. Es una aplicación modular basada en componentes, de forma que los modelos de simulación individuales o atómicos pueden componerse en modelos más grandes. Así, permite la paralelización de las simulaciones ejecutando cada modelo atómico de forma paralela. Los modelos individuales se escriben en su propio lenguaje NED (*NEtwork Description*), que se traduce de forma transparente por la herramienta a C++ [1]. Este simulador de red no es uno como tal, ya que integra otras herramientas como son un analizador gráfico de los resultados que ayuda a reducir el tiempo de pruebas, o un entorno de desarrollo integrado para facilitar el desarrollo de modelos de simulación.

J-Sim[5] [20] o JavaSim es un simulador de red escrito en Java que también se basa en componentes como OMNeT++, de forma que los modelos de simulación se pueden crear de forma jerárquica, construidos a partir de componentes autocontenidos. Estos componentes son escritos en Java, y se utilizan *scripts* TCL para el ensamblado de estos componentes formando un modelo complejo. Contiene un editor gráfico para la construcción de estos *scripts* basado en XML, que los traduce posteriormente a TCL. También permite la paralelización de las simulaciones haciendo uso de la clase *Thread* de Java, que ejecuta varias instancias de su simulador [1].

SimPy[6] es un simulador de red de eventos discretos orientado a procesos que está implementado en Python y que proporciona una interfaz programática en el mismo lenguaje. Los modelos de simulaciones están compuestos por entidades básicas que se ejecutan como procesos, permitiendo la paralelización de las ejecuciones, aunque no su distribución entre diferentes máquinas. También permite la sincronización de los procesos de simulación y ofrece comandos para el control de los datos de simulación [3].

[2] `http://www.nsnam.org/`
[3] `http://www.wireshark.org/`
[4] `http://www.omnetpp.org/`
[5] `https://sites.google.com/site/jsimofficial/`
[6] `http://simpy.sourceforge.net/`

JiST/SWANS[7] es un simulador de red implementado y basado en Java, que está compuesto de dos partes: JiST o *Java in Simulation Time* y SWANS o *Scalable Wireless Ad hoc Network Simulator*, que lo complementa con simulaciones para redes móviles *ad hoc*. Este simulador de red está basado en entidades que representan elementos de la red (como son los nodos), y cuya orquestación se realiza a través de invocaciones a métodos entre entidades. Esto permite que se pueda realizar una ejecución paralela de las simulaciones a través de la definición de puntos de sincronización, con la que se consigue una mejora del rendimiento, pero no permite la distribución de las ejecuciones entre diferentes máquinas. El inconveniente principal es que el software ha dejado de estar mantenido por su autor [3].

En lo referente a software privativo, el más representativo es OPNET Modeler[8]. Es un software propiedad de OPNET Technologies Inc. escrito inicalmente en C y portado a C++, que contiene una larga selección de modelos de protocolos. Permite realizar modelos jerárquicos de gran profundidad, pero sólo permite topologías fijas sin opción a cambiar parámetros. También contiene un editor gráfico de modelos que se almacena en un fichero binario con formato propietario, lo que dificulta su interoperabilidad. A su vez, integra una interfaz gráfica de usuario para la interpretación de los resultados generados similar a la de OMNeT++. Tiene la capacidad de realizar la paralelización de la ejecución de los modelos de simulación como alternativas anteriores [1].

NetSim[9] es un simulador de eventos discretos, escrito en Java y desarrollado por Tetcos en 1997, que proporciona un entorno de modelado y simulación orientado a objetos para la simulación y análisis de escenarios de comunicación de datos y voz. Permite la creación de software independiente de la plataforma y de forma rápida, que puede ser integrado en un documento HTML a través de un *applet* de Java [1].

Qualnet[10] es un simulador de red comercial escrito en C++ y diseñado principalmente para redes inalámbricas, aunque también predice simulaciones de redes cableadas y mixtas. Utiliza el "lenguaje" de simulación paralelo Parsec[11] desarrollado por el Laboratorio de Computación Paralela de la UCLA (PCL) y el modelo de simulación GloMoSim (*Global Mobile system Simulation*). Parsec divide el modelo de simulación en entidades y provee una interfaz de simulación minimalista para ello. El núcleo de Parsec es usado para proveer planificaciones de eventos y servicios de simulación paralelos, y se caracteriza por proveer una muy eficiente infraestructura de simulación paralela para simulaciones de redes inalámbricas. También contiene una interfaz gráfica de usuario integrada para el análisis de los resultados de las simulaciones [1].

En la Tabla 2.1 se pueden encontrar, de forma resumida, algunas de las características más relevantes y comparables para los diferentes simuladores de red. Entre ellas está la licencia de uso que se le aplica, los lenguajes de programación que utilizan, la implementación de la paralelización de la ejecución de las simulaciones y la integración de una interfaz gráfica de usuario para el modelado y/o el análisis de las simulaciones.

[7] `http://jist.ece.cornell.edu/`
[8] `http://www.opnet.com/solutions/network_rd/modeler.html`
[9] `http://www.boson.com/netsim-cisco-network-simulator`
[10] `http://www.qualnet.com/`
[11] `http://pcl.cs.ucla.edu/projects/parsec/`

Simulador de red	Licencia	Lenguajes	Paralelización	Interfaz gráfica
ns-2	Gratuita	C++ y TCL	No	No
ns-3	Gratuita	C++	Sí	No
OMNeT++	Gratuita	C++	Sí	Sí
J-Sim	Gratuita	Java y TCL	Sí	No
SimPy	Gratuita	Python	Sí	No
JiST/SWANS	Gratuita	Java	Sí	No
OPNET Modeler	Comercial	C y C++	Sí	Sí
NetSim	Comercial	Java	Sí	No
Qualnet	Comercial	C++ y Parsec	Sí	No

Tabla 2.1: Comparativa de las características más relevantes de los simuladores de red.

2.1.3. Paralelización y distribución

Entre todos los ejemplos de simuladores de red vistos, una de las características más relevantes es la posibilidad de paralelización de las simulaciones, pero la imposibilidad de la distribución de las mismas entre diferentes máquinas. Las limitaciones presentes en la ejecución de simulaciones (consumo de mucha memoria y tiempo de procesamiento) se solucionaron ofreciendo la capacidad de desarrollar simulaciones detalladamente y a nivel de paquete para conseguir modelos de redes telemáticas a gran escala que pudieran dividirse en pequeñas tareas para su paralelización [8].

Fujimoto et al. [8] identifican dos formas para conseguir paralelizar las simulaciones de red. La primera consiste en crear una herramienta de simulación de red nueva que permita la paralelización, sin embargo este tipo va a tener una dependencia del entorno elevada para conseguir que la ejecución sea eficiente, así como requiere mucho tiempo y esfuerzo para conseguir un sistema usable. La segunda consiste en permitir la ejecución de simulaciones de forma paralela interconectando varias instancias de un mismo simuladores o de diferentes simuladores entre sí. Esta solución es muy utilizada en simulaciones militares. Así, se permite explotar conjuntamente modelos y software de diferentes simuladores en un único sistema.

Los autores siguieron la segunda opción de paralelización y crearon el entorno de simulación de red *Parallel / Distributed Network Simulator* (PDNS) para conseguir realizar simulaciones paralelas y/o distribuidas, utilizando internamente varias instancias del simulador de red ns-2 y la herramienta GTNetS[12] para permitir la una federación escalable de simuladores de red. Sin embargo, necesitaron realizar modificaciones en el simulador de red ns-2 para que pudiese soportar la ejecución distribuida, y también utilizar el enrutamiento NIx-Vector para evitar el consumo de mucha memoria de ns-2, consiguiendo simulaciones de redes más grandes con muchos más nodos [8].

2.1.4. Aplicación en la educación

La realización de simulaciones es una técnica muy utilizada en educación para el apoyo al aprendizaje. Resulta prácticamente imposible para una institución educativa poder

[12] `http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/`

disponer de material novedoso y en gran cantidad para que los alumnos puedan experimentar con él. Por ejemplo, en las prácticas de laboratorios de redes telemáticas se podría usar hardware para el estudio y análisis de protocolos, pero existen varias limitaciones que lo impiden, entre ellas: los cambios frecuentes de protocolo pueden requerir la inversión periódica en nuevo hardware en casos en los que no se pueda actualizar mediante software; el material debería adaptarse al número de alumnos y, en algunos casos, se necesitaría una gran inversión; y la imposibilidad de estudiar el comportamiento de grandes redes telemáticas o sobre escenarios especiales [2, 4, 18].

El uso de simuladores de redes en el área de las redes telemáticas en el campo de la educación supuso un avance significativo, resolviendo las limitaciones anteriores. La posibilidad de modelar grandes sistemas de redes, estudiar los nuevos protocolos y disponer de un simulador para cada alumno, permite que el aprendizaje de cada alumno mejore con respecto al uso de hardware [2, 18].

Uno de los simuladores de red más utilizado durante años en educación para el apoyo al aprendizaje ha sido el ns-2 [5]. Sin embargo, la falta de paralelización de las tareas de simulación no permite adaptar el tiempo necesario para ejecutar las simulaciones de grandes redes telemáticas a las horas de las clases prácticas. A su vez, el análisis de los resultados obtenidos resulta complicado, teniendo que utilizar otras herramientas como Nam[13] o Wireshark[14] [7, 18].

Para solucionar estas limitaciones presentes en el simulador de red ns-2, algunos autores ofrecieron nuevas herramientas como el PDNS de Fujimoto et al. [8] descrito anteriormente, o el *Distributed Network Simulator Environment* (DNSE) de Bote et al. [5], un entorno de simulación de redes distribuido que permite el modelado y ejecución de simulaciones individuales o de barrido de parámetros (esto es, asignar un rango de valores a los parámetros de la simulación) a través del simulador de red ns-2 y en un entorno de paralelización como es un grid computacional.

Aún así, hay más simuladores de red que permiten la paralelización, pero que no son utilizados en entornos educativos para el apoyo al aprendizaje por diversas causas, entre ellas la complicada curva de aprendizaje que necesitan para poder diseñar e implementar un modelo de simulación interesante, la dificultad presente ante el análisis de los resultados debido a la complejidad de dichos resultados o la falta de una interfaz gráfica de usuario que facilite esta operación, y la alta dedicación de tiempo para la ejecución de una simulación de grandes proporciones o con características peculiares.

2.2. Computación en nube

2.2.1. Definición

La nube computacional o la computación en nube (en inglés, *cloud computing*) es un término que hoy en día aún sigue generando mucha confusión en cuanto a qué se refiere exactamente. Los usuarios suelen utilizar el concepto de “en la nube” para todos aquellos servicios que son ofrecidos a través de Internet. Pero el término de “nube” no sólo se refiere a los servicios finales de usuario, sino que agrega a más niveles de servicio,

[13] `http://www.isi.edu/nsnam/nam/`
[14] `http://www.wireshark.org/`

tal y como afirma Armbrust et al. en [21]. En el artículo intentan aclarar todos los conceptos para los términos usados en la nube, y afirman que "nube" se refiere tanto a las aplicaciones de software ofrecidas a través de Internet, como al hardware y los sistemas software en los centros de datos que ofrecen estos servicios. Como se verá más adelante, los servicios ofrecidos se dividen entres niveles: la infraestructura para referirse al hardware, la plataforma para referirse a los sistemas software y el software para referirse a las aplicaciones.

Aún así, hay múltiples definiciones del término de nube computacional, tal y como indican Voorsluys et al. en [9]. El autor de este trabajo propone una definición agregada de todas ellas: un sistema de computación distribuido formado por varios computadores interconectados y vistos como una unidad, que ofrece generalmente una serie de recursos virtualizados de procesamiento, almacenamiento y red, que pueden ajustarse de forma dinámica a través del escalado, los cuales se pagan por su uso, y que se rigen por un acuerdo de licencia de software personalizado. Pero todas las definiciones coinciden en señalar cuatro puntos clave que describen la nube computacional, como son el pago por el uso de los recursos, la percepción "infinita" de recursos gracias a la escalabilidad, la gestión a través de una interfaz unificada, y la virtualización de sus recursos.

2.2.2. Paradigmas de computación distribuida

La nube computacional actualmente es un paradigma emergente en el campo de la computación distribuida, pero tiene sus orígenes en otros paradigmas presentes en este mismo campo [22]. Es por tanto que muchos de los conceptos en los que está basada la computación en la nube están presentes en el *cluster* computacional y en el *grid* computacional que se describen a continuación.

El *cluster* computacional surge debido a la limitación de las supercomputadoras a ciertos problemas que no podían ser tratados con ellas. Surge como una solución para dichos problemas, con el añadido de poder ofrecer muchos recursos computacionales homogéneos a un precio económico. Un *cluster* computacional se define como *una colección de computadoras distribuidas o paralelas que están interconectadas entre sí a través de redes de alta velocidad* [23]. Sus principales características son la alta disponibilidad, el balanceado de carga y la computación masiva. Al estar compuesto por una multitud de nodos computacionales, la cantidad de recursos computacionales aumenta según se añaden más nodos y la carga computacional puede repartirse entre todos ellos, y en caso de fallo en uno de los nodos, el resto no se verán afectados. Para el usuario, todos los nodos computacionales son vistos como si se tratase de una única máquina virtual.

En el caso del *grid* computacional, éste se compone de la agregación de varios recursos autónomos distribuidos en diferentes lugares para la consecución de un objetivo común, que pueden ser asignados de forma dinámica en la ejecución dependiendo de la disponibilidad, capacidad, rendimiento, coste y requisitos de calidad de servicio. Se puede acceder a ellos de forma transparente a través de un *middleware* que divide y distribuye las piezas de una tarea concreta en ellos [23]. El *grid* computacional estará formado por una federación de empresas e instituciones que comparten sus recursos para conseguir un beneficio mutuo. Uno de los aspectos clave del *grid* computacional fue el desarrollo de estándares basados en servicios Web (por ejemplo, SOAP, WSDL o WSRF) que permitiesen la comunicación con los recursos distribuidos de forma que pareciese un único sistema

virtual. La implementación *middleware* más relevante es Globus Toolkit, que implementa varios de los estándares anteriores para permitir el despliegue de servicios y aplicaciones orientadas a servicios. Esta solución tiene dos limitaciones importantes, como son la garantía de la calidad del servicio debido a la influencia en el rendimiento de la ejecución del resto de servicios, y la complicación de configurar los diferentes servicios en ejecución con requisitos diferentes (sistema operativo, biblioteca, etc.) [9].

A pesar de que parecería lógico señalar que la nube computacional se presenta como la evolución lógica del *grid* computacional, esto no es así, y de acuerdo con Buyya et al. [22] se identifica más acorde a una nueva generación de centros de datos compuestos de nodos virtualizados a través de tecnologías de hipervisores[15] que permiten la escalabilidad dinámica de recursos negociada con un acuerdo de nivel de servicio, y que es accesible a través de servicios Web. En cambio, hereda ciertas características de las dos aproximaciones, como por ejemplo el uso de servidores sencillos, un único propietario, una red dedicada y de alta velocidad para su interconexión o una gestión centralizada de sus recursos en el caso del *cluster* computacional, y una escalabilidad elevada o una negociación basada en acuerdos de nivel de servicio en el caso del *grid* computacional. Además, unifica algunas de las características propias de cada uno como son la gestión y asignación de recursos tanto de forma centralizada como distribuida.

Caben destacar otras dos diferencias relevantes entre los tres paradigmas. La primera se refiere a la titularidad de los recursos computacionales ofrecidos, donde en un *grid* está repartido entre varias instituciones y empresas (cada una ofrece sus recursos), mientras que en un *cluster* o una nube normalmente van a ser propiedad de un proveedor que lo ofrecerá como servicio a los clientes [22]. La otra diferencia es relativa a la homogeneidad de los recursos computacionales ofrecidos, siendo muy homogéneos en un *cluster* y muy heterogéneos en un *grid*, lo cual en este último dificulta el desarrollo de *middlewares* y el despliegue y gestión de servicios [5, 23]. Para la nube computacional los recursos pueden ser heterogéneos, pero gracias a la virtualización y el uso de hipervisores no es tan relevante para el desarrollo, gestión y despliegue de *middlewares* [2].

2.2.3. Clasificación

Una de las características que identifican a la nube computacional como una nueva generación de computación distribuida es la capacidad de ofrecer servicios no sólo en un nivel computacional, sino que la abstracción de estos recursos puede llevarse hasta dos niveles más generales. Por tanto, los tres niveles en los que principalmente se clasifica a una nube computacional son [4, 9, 10, 14, 16]:

- Infraestructura como Servicio (*Infrastructure as a Service* o IaaS) es el nivel más bajo de los sistemas de computación en el que se ofrecen recursos de computación, de almacenamiento y de comunicación virtualizados y bajo demanda, pudiendo comprar dichos recursos y facturarlos en base a la cantidad consumida (por ejemplo, número de instancias ejecutadas, duración de la ejecución, almacenamiento adicional consumido, etc.), gracias al escalado dinámico que permite iniciar y parar instancias en función de las que están en ejecución. Es la evolución de los servidores

[15] Un hipervisor es la plataforma software que permite el acceso a los recursos hardware (procesador, memoria, red, etc.) de un ordenador, presentados como una máquina virtual [9].

privados virtuales tradicionales. Permite la ejecución de servidores bajo demanda como máquinas virtuales, pudiendo elegir entre varias opciones de sistemas operativos. Dan la posibilidad de instalar nuevo software en los servidores, propio u ofrecido por defecto, que puede ser personalizado, gestionado y controlado por el usuario tal y como lo puede hacer en un servidor tradicional. Para la administración de este nivel se suele utilizar una interfaz Web, así como otras interfaces programáticas. El ejemplo más utilizado en este nivel es Amazon Elastic Compute Cloud (EC2)[16], aunque existen otros como ServePath GoGrid[17] o RackSpace Cloud[18].

Uno de los servicios que está incluido en este nivel es el de Almacenamiento como Servicio (*Data as a Service* o DaaS), que debido a las oportunidades y retos que presenta tiene el suficiente carácter como para definirse como un nivel más paralelo al IaaS [24]. El almacenamiento en la nube es utilizado en diversas situaciones, como por ejemplo, ampliar el almacenamiento de una instancia, utilizarlo como espacio para copias de seguridad, sincronizar contenido entre diferentes dispositivos, o compartir contenido. Para ello, hace uso de las funcionalidades que puede ofrecer la nube para gestionar un almacenamiento virtual. Existen muchas soluciones para el almacenamiento en la nube y para un amplio público, existiendo soluciones empresariales como Amazon Elastic Block Storage (ESB)[19] o RackSpace CloudFiles[20], o soluciones más personales como Amazon Simple Storage Service (S3)[21], Dropbox[22], Google Drive[23] o Microsoft SkyDrive[24]. También suele ser empleado como almacenamiento de las bases de datos, con soluciones específicas como Amazon SimpleDB[25].

- Plataforma como Servicio (*Platform as a Service* o PaaS), es un nivel de abstracción superior con respecto al IaaS que hace que la nube pueda ser programable, es decir, que permita crear y desplegar aplicaciones sin tener constancia del hardware utilizado (como el número de procesadores o la cantidad de memoria utilizada). Para ello se dispone de aplicaciones como herramientas, entornos o lenguajes de programación concretos para el desarrollo de aplicaciones Web por los usuarios, que permiten separar la programación y pruebas del despliegue final. Algunos de los ejemplos en este nivel son Google App Engine[26], que permite los lenguajes de programación Java y Python, o Windows Azure[27], que permite lenguajes de programación que usen las bibliotecas de .NET.

- Software como Servicio (*Software as a Service* o SaaS), es el nivel más alto de los sistemas de computación, como abstracción tanto del PaaS como del IaaS, en el que

[16] http://aws.amazon.com/ec2/
[17] http://www.gogrid.com/
[18] http://www.rackspace.com/cloud/
[19] http://aws.amazon.com/ebs/
[20] http://www.rackspace.com/cloud/public/files/
[21] http://aws.amazon.com/s3/
[22] http://www.dropbox.com/
[23] http://www.google.com/intl/es/drive/start/index.html
[24] http://skydrive.live.com/
[25] http://aws.amazon.com/simpledb/
[26] http://developers.google.com/appengine/
[27] http://www.windowsazure.com/

se ofrecen aplicaciones a los usuarios para usos muy diversos sin conocer detalles sobre la plataforma o la infraestructura que lo soporta. Uno de los primeros servicios en este nivel fue ofrecer el servicio de correo electrónico basado en la Web ejecutado sobre una plataforma e infraestructura de nube computacional. De hecho, prácticamente todas las soluciones existentes están basadas en la web con una distribución a través de Internet, lo cual permite que sean accesibles desde cualquier lugar y desde una multitud de dispositivos. Son cada vez más usuarios los que están empezando a migrar el uso de aplicaciones de escritorio a aplicaciones web en la nube que ofrecen la misma funcionalidad. Permite que los usuarios no tengan que preocuparse sobre el mantenimiento y actualización del software, y además simplifica a los desarrolladores su desarrollo y despliegue. Algunos de los ejemplos más relevantes en este nivel son Google Apps[28], que contiene una diversidad de aplicaciones para usuarios finales (editores de documentos, hojas de cálculo, presentaciones y dibujos), Salesforce.com[29] y Appian Cloud BPM[30], aplicaciones para la gestión de empresas con respecto a la cartera de clientes y el proceso de negocio, o SmugMug[31] y Picasa[32] para compartir fotografías y vídeos. Además, en este nivel se pueden incorporar algunas de las aplicaciones de DaaS, puesto que se ofrecen a su vez como infraestructura (con una interfaz programática) y como software (con una interfaz Web de usuario), tal y como es el caso de Dropbox[33] o Google Drive[34].

Una de las tendencias en la nube computacional es utilizar el término "como servicio" junto al nombre de cualquier tipo de servicio para referirse a la virtualización del mismo ofrecido a través de Internet [25]. A esta tendencia se la denomina de forma común "X" como Servicio o *"X" as a Service* (XaaS), donde la letra "X" puede ser, además de infraestructura, plataforma, software o almacenamiento, otros servicios como bases de datos, comunicación, seguridad, gestión de identidades o escritorio [26].

2.2.4. Modelos de despliegue

Además de una división por niveles de la nube computacional, se puede realizar una división por modelos de despliegue en función de la distribución física de sus servidores y de su contenido. Inicialmente se identificaron dos modelos de despliegue [21]: nube pública y nube privada. Pero el uso de estos dos modelos ha originado el surgimiento de otros dos [4, 9]: la nube híbrida y la nube comunitaria. Las características que identifican a cada uno de estos cuatro modelos de despliegue son las siguientes:

- Una nube pública es aquella que se ofrece al público en general y en la que el modelo de negocio es el pago por el uso de sus recursos computacionales, de almacenamiento o de comunicación. Por ejemplo, Amazon EC2 es una nube pública.

[28] `http://www.google.com/apps/intl/es/index.html`
[29] `http://www.salesforce.com/`
[30] `http://www.appian.com/bpm-software/cloudbpm.jsp`
[31] `http://www.smugmug.com/`
[32] `http://picasaweb.google.com/`
[33] `http://www.dropbox.com/`
[34] `http://www.google.com/intl/es/drive/start/index.html`

- Una nube privada se refiere a un centro de datos instalado en una empresa o una institución para su uso interno.

- Una nube híbrida es la combinación de una nube privada con una nube pública, de forma que se puedan ampliar los recursos limitados de una nube privada con los ofrecidos por una nube pública de forma temporal en los picos de carga, consiguiendo una sensación de recursos "ilimitados".

- Una nube comunitaria es aquella solución intermedia entre una nube pública y una nube privada, donde el acceso a la misma está restringido a un grupo de instituciones que forman una comunidad, con el fin de llevar a cabo un objetivo en común compartiendo los recursos de la nube computacional.

2.2.5. Ventajas e inconvenientes

Independientemente del modelo de despliegue que siga una nube computacional, ésta va a tener unos ventajas u oportunidades y unos incovenientes o limitaciones comunes. La clasificación de beneficios y limitaciones realizada por la comunidad científica es muy diversa [2, 9, 10, 15, 16, 21, 23], pero hay ciertos rasgos en los que coinciden muchos de los autores anteriores, y que serán los que se describan en este estudio.

El beneficio principal de la nube computacional, ya que es la base del paradigma, es la virtualización de todos sus recursos (computación, almacenamiento, servicios, etc.). La virtualización genera la falsa apariencia a los usuarios de recursos "infinitos" que pueden ser solicitados en cualquier momento para ser utilizados [2, 9]. Esto se realiza a través de un autoservicio de los recursos, pudiendo solicitar la reserva de los mismos en el momento que se necesiten y liberarlos cuando ya no sean necesarios, sin necesidad de tener un acuerdo de uso previo [9, 10]. También existe la posibilidad de automatizar dicha reserva de recursos por las aplicaciones sin la necesidad de intervención humana [9].

El uso de los recursos suele estar sujeto a un pago por su uso, con tarifas ajustadas a un espacio temporal corto (por ejemplo, una hora) que permite un ahorro en los costes al usuario. Esto es, cuando se necesitan los recursos por un breve periodo de tiempo debido a una carga puntual, se pueden reservar esa hora y liberarlos a su término si ya no son necesarios, pagando sólo por esa hora [10, 16]. La forma de medir el uso de los recursos se realiza a través de métricas que, además de contabilizar el tiempo que se han estado usando, puede contabilizar otros tamaños como la cantidad de ancho de banda utilizado o el número de procesamientos realizados [9, 16]. Otra parte del ahorro está presente también en los costes iniciales, ya que el pago por uso libera al usuario de un compromiso inicial por unos recursos que no va a necesitar, pero le permite que en el momento que su aplicación empiece a crecer, la ampliación de la capacidad de los recursos crecerá junto a ella [2, 10, 21].

La característica de aumentar y disminuir los recursos de forma dinámica según las necesidades de la aplicación se le denomina elasticidad o escalabilidad de la nube computacional, y es una de sus características más atractivas [2, 10, 15, 21]. Esta escalabilidad es hacia arriba cuando se reservan y amplian los recursos, y es hacia abajo cuando se liberan o disminuyen los mismos [9, 23]. Por otro lado, la escalabilidad también se puede clasificar según el tipo de recursos a escalar, siendo vertical cuando se escalan los recur-

sos hardware virtualizados, u horizontal cuando se escalan las instancias en ejecución. Además, puesto que cada cliente tiene unas necesidades diferentes y una infraestructura de nube computacional puede ser utilizada por muchos usuarios, se permite la personalización de los recursos computacionales (por ejemplo, número de procesadores, cantidad de memoria o capacidad de almacenamiento) a las necesidades de cada usuario [9, 10]. Gracias a la virtualización y la escalabilidad se consigue también un ahorro energético, primeramente debido a la compartición de los recursos de un grupo de máquinas entre varios usuarios, y luego debido a la posibilidad de desactivar o suspender aquellas máquinas físicas que no estén siendo utilizadas temporalmente [4, 9].

Otra de las ventajas de la nube computacional es la alta disponibilidad que es capaz de ofrecer. Puesto que está compuesta por múltiples nodos repartidos por varios lugares físicos, en caso de fallo en alguno de ellos, la carga puede ser distribuida por el resto de los nodos que están disponibles [2, 9, 10, 15]. Por ejemplo, si fallase una instancia en un grupo de servidores, automáticamente se podría iniciar una nueva instancia en otro grupo de servidores saludable [9]. A su vez, y para que no afecte al servicio temporalmente, se puede realizar una redundancia de los servicios, de forma que se pueden tener dos instancias del servicio en lugares diferentes para que, si falla uno de ellos, sea el otro el encargado de seguir ofreciendo el servicio sin interrupción [2]. Finalmente, la nube computacional suele llevar asociada la realización de copias de seguridad periódicas, protegiéndose así ante este tipo de fallos [9].

Pero como todo paradigma, la nube computacional tiene ciertas limitaciones o retos pendientes de superar. El que más repercusión tiene en la comunidad científica es el de la seguridad y privacidad de los datos [10, 16]. Armbrust et al. [21] aseguran que el carácter público de la nube computacional expone a los sistemas a ser más atacados. Esta limitación afecta a todos los niveles de la pila de la nube computacional [9]. Toda empresa e institución tiene ciertos datos de carácter confidencial o que requieren un tratamiento sujeto a leyes de privacidad locales, y necesitan la garantía de que sus datos van a estar seguros [16]. El primer sistema de seguridad es la virtualización, que permite aislar a los diferentes recursos virtualizados, pero no todo puede ser virtualizado y no están exentos de fallos, y debido a que se comparten recursos físicos, se podría acceder a datos no autorizados [21]. Actualmente ya se emplean tecnologías bien conocidas como la codificación de datos, el uso de redes locales virtuales (*Virtual Local Area Network* o VLAN) o cortafuegos que mejoran la seguridad y privacidad, pero no son válidas para todos los casos y se necesitan buscar mejores soluciones [9].

Otro reto está presente en la garantía de disponibilidad de los servicios de la nube computacional y su interoperabilidad. Cuando un usuario utiliza un servicio de la nube computacional, éste es ofrecido por un tercero de forma remota a través de Internet y no contiene la garantía suficiente de que, en caso de que el servicio deje de estar disponible, sus datos se queden bloqueados en ese servicio sin poder utilizarlos [4, 9, 10, 15, 21]. Por ello, debería garantizarse la interoperabilidad entre los servicios de la nube computacional, de forma que se puedan realizar migraciones entre los diferentes servicios para no tener un bloqueo de los datos [4, 9, 21, 23]. Esto se consigue con la estandarización y actualmente, a pesar de los esfuerzos puestos por diversas organizaciones, no todos los proveedores aplican los estándares o éstos no están completamente definidos [9, 10, 13, 16, 21, 23].

Relativo a los recursos físicos y el rendimiento, los usuarios quieren tener garantías de que los servicios que se le ofrecen y por los que pagan van a ofrecer un mínimo rendimiento [21]. Para ello es necesario definir buenas políticas de calidad de servicio y llegar a acuerdos de nivel de servicio para poder garantizar dicho rendimiento [16]. Aún así, existen problemas presentes como los cuellos de botella en la transferencia de datos a través de las redes de comunicaciones que, ante una elevada carga de trabajo, pueden reducir la calidad del servicio [10, 21]. También en el caso de realizar la escalabilidad de los recursos deberá garantizarse un tiempo de respuesta adecuado desde su petición hasta que están disponibles para garantizar la disponibilidad del servicio [10, 21, 23]. Por otra parte, el rendimiento garantizado no deberá perjudicar a la gestión eficiente de los recursos, ya que una mala gestión podría suponer un mayor coste energético [9, 23].

En la Tabla 2.2 se presenta un resumen con todas las oportunidades y limitaciones de la nube computacional descritas anteriormente.

La nube computacional
Oportunidades
Virtualización de los recursos [2, 9]
Apariencia de recursos infinitos [21]
Autoservicio de los recursos [9, 10]
Compromiso inicial innecesario [2, 10, 21]
Elasticidad o escalado dinámico [2, 9, 10, 15, 21, 23]
Personalización de los recursos [9, 10]
Pago por el uso de los recursos [2, 9, 16, 21]
Alta disponibilidad de los servicios [2, 9, 10, 15]
Fiabilidad y copias de seguridad de los datos [2, 9]
Menor coste computacional [2]
Ahorro energético [4, 9]
Limitaciones
Seguridad y privacidad de los datos [9, 10, 16, 21, 23]
Compartición de los recursos físicos [21]
Continuidad de los servicios [9, 10, 15, 21]
Bloqueo de los datos [10, 21]
Estandarización [9, 10, 13, 16, 23]
Rendimiento impredecible de los recursos [16, 21]
Cuellos de botella en la transferencia de datos [10, 21]
Rapidez en la escalabilidad [10, 21, 23]
Gestión eficiente de los recursos [9, 23]
Fallos en grandes sistemas distribuidos [9, 10, 21, 23]

Tabla 2.2: Clasificación de las oportunidades y limitaciones de la nube computacional.

2.2.6. Aplicación en la educación

Como ya se ha visto, la nube computacional contiene muchos beneficios que pueden ser aplicados a otros campos como oportunidades ante las limitaciones presentes en ellos.

Uno de estos campos es la educación, y más concretamente, en la utilización de software para el apoyo al aprendizaje [4, 14]. La primera utilización de la computación en la nube en el campo educativo fue en el nivel de SaaS con el uso del correo electrónico, en el que instituciones educativas migraron sus servicios de correo electrónico a este tipo de soluciones [4]. Proveedores como Google o Microsoft, que dominan el mercado del sector educativo, ofrecen servicios en la nube computacional de forma gratuita a este sector, tales como el correo electrónico (Gmail[35] y Outlook[36]) o suites de ofimática (Google Apps[37] y Office Web Apps[38]). Aunque las instituciones educativas también quieren migrar otros servicios como los sistemas de gestión del aprendizaje (*Learning Management Systems* o LMS) para disminuir los costes relativos a las licencias, el hardware y el mantenimiento [14]. A su vez, la nube computacional da pie a la creación de nuevo software de apoyo al aprendizaje aplicando sus ventajas [15].

Uno de los problemas que están presentes en los entornos educativos son los picos de carga puntuales, como por ejemplo en la realización de matrículas, los periodos de exámenes o la realización de prácticas de laboratorio [4, 14]. Para ello, tradicionalmente las instituciones educativas invierten en nuevo hardware para poder dar soporte a esos picos de carga, quedando posteriormente infrautilizado. En este aspecto, la nube computacional puede ayudar con la escalabilidad de sus recursos, ofreciendo una escalabilidad hacia arriba en esos momentos puntuales y pagando por el uso de los recursos en ese periodo, consiguiendo un ahorro en los costes del nuevo hardware a la institución [4].

Y es que el ahorro es uno de los puntos clave en las instituciones educativas, sobre todo en el marco actual económico en el que la financiación directa e indirecta ha sido bruscamente reducida [4]. Como ya se comentó, muchas empresas ofrecen parte de su software en la nube de forma gratuita a las instituciones educativas. Con ello, las empresas pretenden que los estudiantes se acostumbres a su uso y que posteriormente, en su futuro laboral, utilicen dichos servicios pagando por su uso. Por parte de las instituciones, el uso de servicios gratuitos o de pago por uso permite un ahorro sustancial en hardware, software, mantenimiento y soporte [4]. Además, les permite concentrar todos sus esfuerzos en el núcleo del negocio, como es la educación y la investigación, relegando la parte computacional a proveedores externos [14].

Por otro lado, la nube computacional puede ofrecer una disponibilidad de los servicios mayor a la que una institución educativa puede ofrecer con sus propios recursos. Por ejemplo, Google garantiza que sus sistemas tienen una disponiblidad del 99,9 %, un porcentaje que las instituciones educativas no pueden alcanzar sin una alta inversión [14]. Por tanto, gracias a la utilización de servicios en la nube y su alta disponibilidad, los usuarios tendrán una mayor satisfacción y las instituciones educativas no tendrán que preocuparse sobre caídas del servicio o pérdida de los datos [4, 14]. Además, también se garantiza que los servicios ofrecidos estarán disponibles desde cualquier lugar, gracias a la ubicuidad ofrecida, y accesibles desde cualquier dispositivo, incluidos los dispositivos móviles, gracias a la utilización de interfaces basadas en la Web [14, 15]. A su vez, permite que los estudiantes puedan utilizar el software de forma colaborativa, compartir documentación, una mejora en la comunicación entre sí y la posibilidad de realizar revisiones entre pares

[35] `http://mail.google.com/`
[36] `http://www.outlook.com/`
[37] `http://www.google.com/apps/intl/es/index.html`
[38] `http://office.microsoft.com/es-es/web-apps/`

[4, 15].

Otra de las ventajas que la nube computacional ofrece a las instituciones educativas es su aportación a la consecución de los objetivos de impacto medioambiental. Debido a la externalización de los servicios que ofrece, el consumo energético utilizado en el hardware de la institución se disminuye, y la huella medioambiental de los centros de datos que se aplica a estos estudios es más baja gracias a la optimización del uso de los recursos hardware que se emplea en ellos. Y es que el crecimiento digital al que están sometidas las instituciones educativas hoy en día para adaptarse a las nuevas tecnologías, implicaría un aumento del consumo energético que la nube computacional puede solventar [4].

Aún con todas las ventajas expuestas anteriormente, las instituciones educativas son reticentes a migrar sus servicios a la nube computacional debido a la falta de confianza originada por las limitaciones que están presentes en este paradigma [4]. La principal y más importante es la relativa a la seguridad de los datos, ya que aunque muchos de ellos no son relevantes (por ejemplo, material didáctico o generado por los alumnos en plataformas abiertas), los datos privados de los estudiantes y trabajadores sí lo son [4, 14]. Pasar estos datos a un servicio en la nube supone perder el control físico de los datos, por lo que es necesario firmar acuerdos de nivel de servicio personalizados que garanticen la seguridad y privacidad de este tipo de datos [4].

Otro de los riesgos que limita la implementación a instituciones educativas es el miedo al bloqueo de sus datos en los servicios de la nube computacional. Los esfuerzos realizados para la migración de sus datos o la implementación de un nuevo servicio en la nube computacional son elevados [4]. En el caso de que uno de los servicios en la nube computacional dejase de estar operativo o la institución no pudiese sufragar los gastos por el uso del servicio, las instituciones podrían correr el riesgo de perder sus datos debido a la imposibilidad de migrarlos a otro servicio [4]. Y esto es debido a la falta de estándares entre los servicios de la nube computacional que permitan una interoperabilidad entre sí, que debe solucionarse para que la confianza en ellos por estas instituciones permitan asegurarse una integridad de sus datos [14].

2.2.7. Aplicación en la simulación en educación

De entre las diversas áreas de las instituciones educativas, tanto McDonald et al. [4] como Rajaei et al. [2] coinciden en que uno de las principales áreas en la que la nube computacional puede solventar sus limitaciones es en las aplicaciones de simulación. Las simulaciones sirven para estudiar el comportamiento de un sistema, aplicación o plataforma. El coste de las herramientas de simulación puede ser elevado en situaciones en las que, por ejemplo, los modelos de simulación son complejos (como simulaciones de evolución climática o de mecánica de fluidos), o se manejan un alto número de elementos (como simulaciones de una red de datos con muchos nodos o de un sistema multicelular). En estos casos las simulaciones requieren mucho coste computacional, por lo que para alcanzar tiempos de respuesta razonables en su uso, puede requerir una gran inversión en infraestructura de hardware que no es viable para las instituciones educativas.

Existen algunos casos en los que los simuladores pueden requerir un uso de una infraestructura distribuida, dividiendo una simulación a gran escala en tareas individuales y repartiéndolas entre varias computadoras para su ejecución, y que no se pueden so-

lucionar con la paralelización de las simulaciones, por ejemplo porque el algoritmo no permite dicha paralelización o porque son paralelizables en un grano muy fino. Por ello, se ha estudiado la aplicación de los paradigmas del campo de la computación distribuida en los simuladores [2, 4]. Las supercomputadoras pueden ofrecer la capacidad suficiente computacional para su ejecución, pero para grandes simulaciones pueden no ser suficientes o rentables. Con el *cluster* computacional y el *grid* computacional se puede repartir la capacidad de procesamiento entre varias computadoras, pero la flexibilidad es limitada a la hora de ampliar los recursos computacionales bajo necesidades puntuales. Por ello, la nube computacional puede ofrecer más flexibilidad gracias a la percepción "infinita" de recursos computacionales y al escalado dinámico de sus recursos [2, 27].

Capítulo 3

Propuesta y Diseño de Aplicación

3.1. Introducción

Para este trabajo de investigación se propone un prototipo de aplicación educativa para el aprendizaje electrónico en el campo de simulación de redes TCP/IP, aplicado en un entorno universitario de las ramas de ingeniería de la telecomunicación e informática, que aproveche las ventajas tecnológicas que ofrece la computación en la nube para la distribución de los servicios de los que estará compuesta y la escalabilidad de recursos utilizados en la simulación. También se pretende mejorar la interpretación de los resultados obtenidos en las simulaciones a través de una interfaz gráfica de usuario que utilice las tecnologías actuales para la visualización de contenidos digitales. Esta aplicación se denominará **Entorno de Simulación de Red Distribuido III** o en inglés *Distributed Network Simulator Environment III* (DNSE3).

El origen de la aplicación DNSE3 proviene de una aplicación anterior propuesta por Bote et al. [5] en 2010, llamada Entorno de Simulación de Red Distribuido (en inglés, *Distributed Network Simluation Environment* o DNSE), que a su vez está basada en otra aplicación de los mismos autores llamada Entorno de Simulación de Red Colaborativo (en inglés, *Collaborative Network Simulation Environment* o CNSE) [28]. El DNSE utiliza el paradigma del *grid* computacional para la distribución de sus servicios entre varios servidores, los cuales pueden estar alojados en diferentes ubicaciones físicas, pero a través de la conexión a Internet que tiene cada uno de ellos, pueden comunicarse entre sí como si estuviesen en un mismo lugar. Para la ejecución de las simulaciones de redes TCP/IP hace uso del simulador de red *ns-2*. Se pueden realizar simulaciones individuales (esto es, con sólo un conjunto de valores para los parámetros definidos) o simulaciones de barrido de parámetros (esto es, asignar varios conjuntos de valores para los parámetros definidos, realizando varias simulaciones individuales para completar el barrido de parámetros). Puesto que el *ns-2* sólo puede ejecutar una simulación individual de forma simultánea, se aprovecha el *grid* computacional para ejecutar varias simulaciones individuales de las que se compone una simulación de barrido de parámetros, de forma simultánea y distribuida, consiguiendo de esta manera un ahorro en tiempo de ejecución. En cambio, al realizar varias simulaciones individuales, los resultados obtenidos se presentan más numerosos y, por tanto, más complicados de analizar por su gran volumen. Por ello, el DNSE presenta una visualización de los resultados a través de gráficas de diferentes parámetros, visua-

lizaciones animadas del comportamiento de la red u obteniendo una serie de estadísticas predefinidas del bruto de datos generado.

Por tanto, los objetivos que se alcanzan con el DNSE son el ahorro de tiempo a la hora de realizar simulaciones de barrido de parámetros, distribuyendo las diferentes tareas de simulación entre el conjunto de servidores del *grid* computacional, y la evaluación de los resultados obtenidos de forma simplificada, ofreciendo para ello una interfaz gráfica de usuario que facilita su interpretación. Puesto que esta aplicación está enfocada dentro del aprendizaje electrónico en un entorno universitario, será utilizada en las clases de laboratorio en las asignaturas relativas a las redes telemáticas. Normalmente, estas clases tienen una duración breve (aproximadamente de dos horas) y es recomendable poder realizar las tareas de simulación dentro del espacio asignado, por ello el DNSE consigue adecuar el tiempo empleado para la simulación a las de una clase de laboratorio a través de la distribución de las simulaciones en un *grid* computacional. A su vez, los alumnos necesitan adquirir algún valor pedagógico de las simulaciones, y gracias a la interfaz gráfica de usuario ofrecida por el DNSE son capaces de generar gráficas, resultados estadísticos y visualizaciones animadas que permiten comprender mejor dichos resultados.

Los objetivos conseguidos con el DNSE son los que se pretenden conseguir también con la aplicación DNSE3, pero además intentando mejorar los resultados obtenidos para cada uno dentro del contexto del aprendizaje electrónico para las clases de laboratorio de las asignaturas de redes telemáticas. Por tanto, para conseguir ahorrar en el tiempo empleado para ejecutar las simulaciones individuales de una simulación de barrido de parámetros, se aprovecharán las ventajas de la nube computacional relativas a la escalabilidad de los recursos computacionales, adaptando los mismos según la carga de trabajo estimada para ese momento, y consiguiendo adicionalmente un ahorro en cuanto a los recursos hardware utilizados y el consumo energético asociado al mismo para la institución universitaria. También se actualizará la versión del simulador de red, utilizando el *ns-3* que incorpora nuevas funcionalidades para la ejecución de diferentes métodos y protocolos en la simulación de redes TCP/IP, así como un formato de los resultados aplicando un formato común utilizado por múltiples aplicaciones del entorno de simulaciones de redes y que amplía las capacidades de análisis de los mismos, aunque lamentablemente se pierde la capacidad de visualizar de forma animada los resultados de una simulación. Con este nuevo formato común para los resultados se facilitará su interpretación, aunque será necesario adaptar la interfaz gráfica de usuario a este nuevo formato y, también, a los nuevos requisitos solicitados por los profesores y alumnos que utilizaron el DNSE. Para ello se utilizará una interfaz gráfica de usuario basada en la Web que permitirá, además de la gestión de la aplicación por parte del usuario, la generación visual y más cómoda de gráficas y estadísticas, la generación de informes de los paquetes de simulación y la posibilidad de anotaciones personalizadas en simulaciones y gráficas. También se obtienen ventajas adicionales como una ejecución de la interfaz gráfica de usuario independiente de la plataforma y una ubicuidad en la ejecución de la misma.

Relativo al diseño de la aplicación DNSE3, se utilizará una arquitectura orientada a servicios (*Service-Oriented Architecture* o SOA), que permitirá dividir las principales funcionalidades de la aplicación en diferentes servicios, intentando proporcionar la mayor independencia funcional para cada uno de los servicios y permitiendo así su posterior reutilización en otras aplicaciones para aquellos más generalistas. La división en servicios

se aprovechará, junto con la escalabilidad horizontal proporcionada por la nube computacional, para permitir la posiblidad de aumentar o disminuir el número de servicios según la demanda de recursos actual de la aplicación. Para ello, se deberá realizar una gestión de métricas que sean capaces de controlar diferentes factores de cada uno de los entornos de los servicios, así como una gestión de reglas que controlen dichas métricas y permitan llevar a cabo la escalabilidad horizontal del sistema.

Puesto que la aplicación DNSE3 va a estar dividida en diferentes servicios que se ejecutarán en diferentes espacios virtuales de una nube computacional, se necesitará un sistema de comunicación entre ellos para poder llevar a cabo las funcionalidades de la aplicación. El diseño arquitectural REST (*Representational State Transfer*), basado en una arquitectura orientada a recursos (*Resource-Oriented Architecture* o ROA), permite diseñar un elemento software para realizar una comunicación utilizando el protocolo HTTP, entre otras cosas, pudiendo aplicar una serie de acciones básicas (consultar, crear, modificar y eliminar, entre otras) sobre un conjunto de recursos de los que está compuesto. Por tanto, se diseñarán los servicios siguiendo el diseño arquitectural REST para que ofrezcan diversos recursos de la aplicación DNSE3 sobre los que realizar diferentes acciones. A su vez, se definirá para cada uno de ellos un formato común de interfaz programática de la aplicación (*Application Program Interface* o API) que permita conocer al resto de los servicios, e incluso a otros elementos software externos, cómo establecer la comunicación con dicho servicio.

En las siguientes secciones de este capítulo se define de forma más precisa la aplicación DNSE3 propuesta. Primero se realiza una elicitación de los requisitos software de la aplicación en general. Posteriormente, se describe la arquitectura del sistema en el que se identifican las funciones básicas y la descomposición de servicios. A continuación, se obtienen los requisitos específicos y se realiza un análisis y diseño orientado a recursos (*Resource-Oriented Analysis and Design* o ROA/D) [29] para cada uno de los servicios de la aplicación. Luego, se realiza un estudio sobre la selección de las tecnologías utilizadas en el DNSE3. Y finalmente, se presenta un plan de implementación para la construcción y despliegue de la aplicación DNSE3.

3.2. Requisitos de la aplicación

Los requisitos que han sido tomados para la aplicación DNSE3 han sido heredados, en su mayor parte, de la aplicación anterior DNSE, puesto que la funcionalidad básica que se desea aplicar es la misma. Esta aplicación está teniendo buenos resultados durante su puesta en práctica en las clases de laboratorio de las asignaturas de redes telemáticas, y así se puede ver en la evaluación realizada por Bote et al. [5], aunque también tiene ciertas limitaciones y mejoras que tanto alumnos como profesores desean aplicar. En la entrevista realizada a un profesor de la asignatura de Teletráfico impartida en el Grado de Ingeniería Telemática de la Universidad de Valladolid, expuso todos los comentarios de mejora recibidos por los otros profesores de la asignatura y por los alumnos que la han utilizado durante este curso y cursos anteriores. Todos los comentarios factibles se han tenido en cuenta para ser incorporados como nuevos requisitos en la aplicación DNSE3.

La aplicación DNSE3 va a permitir la simulación de diferentes parámetros sobre el diseño de una red TCP/IP. Se podrán realizar simulaciones individuales dando un solo

valor a los parámetros de la red, o simulaciones de barrido de parámetros dando un solo valor o un rango de valores a los parámetros de la red. Estas simulaciones estarán contenidas en un paquete de simulación que describirá, entre otras cosas, los parámetros a los que asignar los valores. Los paquetes de simulación estarán almacenados en un repositorio de ficheros remoto, y se podrán subir desde una ubicación local, trabajar con ellos en la aplicación y descargarlos posteriormente para tener una copia, incluyendo la opción de eliminar el paquete de simulación del repositorio de ficheros. La aplicación permitirá una gestión CRUD (*Create, Record, Update and Delete*) de usuarios a través de cuentas de usuarios, identificados por un nombre de usuario y una contraseña. El repositorio de ficheros estará dividido en cuentas de usuario y los paquetes de simulación estarán asignados a una cuenta de usuario, por lo que se podrán listar los paquetes de simulación de cada usuario.

En un paquete de simulación se podrán crear nuevas simulaciones individuales y de barrido de parámetros, así como modificar sus valores o eliminarlas. Las simulaciones se podrán ejecutar para obtener un conjunto de valores de salida con los resultados y se podrán tomar anotaciones. Los valores de salida podrán consultarse con sus datos en bruto, o se podrá realizar un tratamiento de los mismos, a través de las variables disponibles en cada uno, para generar resultados gráficos y estadísticos. Sobre las gráficas se podrán realizar anotaciones, y para los resultados estadísticos será necesario seleccionar un método estadístico implementado por la aplicación y asignarle los valores correspondientes a los parámetros solicitados para dicho método. En cualquier momento se podrá generar un informe del paquete de simulación con la información del mismo, esto es, sus datos descriptivos, sus simulaciones individuales y de barrido de parámetros, y los resultados generados de dichas simulaciones.

Por tanto, a través de la descripción anterior, se definen tres funciones principales en la aplicación DNSE3:

- La **gestión de los paquetes de simulación,** que incluirá la **gestión de las cuentas de usuario** (con la creación, consulta, modificación y eliminación de las cuentas de usuario) y la **gestión de la sesión de usuario** (con el inicio, consulta y cierre de la sesión), que permitirá la subida de paquetes de simulación a las cuentas de usuario, para posteriormente poder consultarlos, modificarlos, eliminarlos y descargarlos.

- La **gestión de las simulaciones,** que permitirá la creación, modificación, eliminación, ejecución y anotación de las simulaciones individuales y de barrido de parámetros.

- La **gestión de las visualizaciones,** que permitirá la visualización de los datos de salida generados para una simulación, y la generación, eliminación y anotación de las gráficas específicas para ciertas variables de salida, así como la generación y eliminación de estadísticas para ciertos valores de salida utilizando un método estadístico implementado por la aplicación. Sobre el paquete de simulación se podrá realizar la **gestión del informe**, pudiendo generar, descargar y eliminar dicho informe con todo su contenido.

Los usuarios de esta aplicación serán los profesores y alumnos de los laboratorios de las asignaturas de redes telemáticas a las que va dirigida, por lo que su perfil ya incluye el

suficiente nivel y conocimiento técnico como para entender qué realizar y cómo llevarlo a cabo.

La aplicación DNSE3 cumplirá con una serie de restricciones y dependencias que han sido obtenidas a partir del estudio del estado del arte del capítulo 2 para la investigación, siendo esta aplicación un medio tecnológico para llevar a cabo la evaluación de la investigación y obtener así las conclusiones oportunas sobre ella. Las restricciones y dependencias son las siguientes:

- La aplicación deberá ejecutarse en un entorno de **nube computacional** en la que se puedan distribuir las tareas de ejecución de las simulaciones, ya sea la propia de una simulación individual o las varias simulaciones individuales de una simulación de barrido de parámetros, entre varias instancias de la misma para conseguir un escalado horizontal. También dependerá de la interfaz programática definida por la nube computacional y sus instancias, tanto para la obtención de los valores propios de la ejecución de una instancia, como de las acciones necesarias para la escalabilidad horizontal hacia arriba o hacia abajo.

- Las simulaciones deberán ser ejecutadas por el **simulador de red *ns-3***, dependiendo por tanto del formato que define para la entrada de la ejecución de una simulación individual y del que define en los resultados obtenidos de dicha simulación.

- Los servicios en los que se descompondrá la aplicación DNSE3 deben seguir el **diseño arquitectural REST**, tanto para su definición interna y su interfaz programática, como para realizar la comunicación con los otros servicios y con la nube computacional.

- La interfaz gráfica de usuario de la aplicación DNSE3 debe ser una **interfaz Web**, de forma que pueda ser ejecutada desde cualquier navegador Web, consiguiendo una independencia de la plataforma y dando la posibilidad de ubicuidad de la aplicación.

Con respecto al rendimiento de la aplicación DNSE3, se han definido unos requisitos de rendimiento que garanticen que la solución diseñada pueda ser utilizada por varios alumnos en una clase de laboratorio. Por ello, deberá permitir ejecutar como mínimo 1.000 simulaciones individuales de forma simultánea. También deberá permitir conectarse como mínimo a 40 usuarios de forma simultánea y almacenar como mínimo 40 paquetes de simulación y 10 ficheros de salida por cada simulación.

Todo el contenido de esta sección está detallado y ampliado con información más precisa en el informe técnico de Especificación de Requisitos Software [30] que acompaña este Trabajo Fin de Máster.

3.3. Arquitectura del sistema

Para la arquitectura del sistema se ha utilizado una representación arquitectónica basada en el Modelo 4+1 [31, 32] que define cinco vistas diferentes para mostrar diferentes aspectos del sistema. Estas vistas son las siguientes: vista lógica, vista de implementación, vista de procesos, vista de despliegue y vista de casos de uso. Puesto que la aplicación

DNSE3 no está terminada en su totalidad para este Trabajo Fin de Máster (a falta de la implementación, despliegue y puesta en práctica), sólo es posible completar dos de las cinco vistas, como son la vista de casos de uso en la que se definen los actores y los escenarios de uso de la aplicación, y la vista lógica en la que se definen los diferentes paquetes o subsistemas de la aplicación, que en una arquitectura orientada a servicios (SOA) serán los diferentes servicios de los que estará compuesta la aplicación y que se identificarán de forma que tengan una atomicidad relevante para realizar una función específica y un bajo acoplamiento entre sí para ser lo más genéricos posibles.

En la **vista de casos de uso**, inicialmente se definen los actores que van a estar involucrados en el sistema, que en este caso serán tres: el *usuario* de la aplicación que realizará todos y cada uno de los casos de uso del sistema, la *nube computacional* que representa a la instancia en la que se ejecutarán los servicios de la aplicación, y el *simulador de red ns-3* como herramienta externa donde se ejecutarán cada una de las simulaciones individuales. Estos dos últimos actores, aunque no ejecutarán ninguna funcionalidad del sistema, tendrán una función significativa en la aplicación y por eso se representan. Estos actores van a tener, por tanto, una interacción activa o pasiva con los escenarios de uso definidos para la aplicación DNSE3, que se corresponderán con cada uno de los requisitos funcionales de la sección anterior.

Los escenarios de uso se han clasificado en cuatro categorías atendiendo a las principales funciones que realizan en el sistema, y son: *usuarios* para la gestión de cuentas de usuario y sus sesiones, *paquetes* para la gestión de los paquetes de simulación, *simulaciones* para la gestión de las simulaciones de un paquete de simulación, y *visualizaciones* para la gestión de los resultados generados de las simulaciones y el informe del paquete de simulación. En el diagrama de casos de uso de la Figura 3.1 están representadas todas las categorías anteriores como paquetes, conteniendo cada una los escenarios de uso o casos de uso de la aplicación DNSE3, y su relación con cada uno de los actores del sistema.

En la **vista lógica** se ha seguido una arquitectura orientada a servicios (SOA) que descompondrá la aplicación DNSE3 en diferentes servicios encargados de llevar a cabo la funcionalidad del sistema con la mayor autonomía e independencia posible. Aunque antes de describir los servicios se identificarán aquellos agentes externos con los que interactuarán. Estos agentes externos son los siguientes: el *usuario* encargado de iniciar cada una de las funcionalidades del sistema, y que las ejecutará a través de la *interfaz Web* de la aplicación, siendo la cara visible de la aplicación al exterior y que ofrecerá todas las funcionalidades del sistema; por otro lado la *nube computacional* será donde se estén ejecutando los servicios y donde se llevarán a cabo las tareas de recogida de métricas y gestión de la escalabilidad horizontal, y el *simulador de red ns-3* usado para llevar a cabo la ejecución de las simulaciones individuales y posterior recogida de sus resultados.

Relativo a los servicios que compondrán la aplicación DNSE3, se ha hecho una clasificación en tres tipos de servicios según el nivel de acoplamiento que tienen con respecto a a la infraestructura de la nube computacional y la aplicación DNSE3. De esta forma, los tres tipos de servicios son: los servicios propios de la infraestructura, los servicios genéricos de la aplicación y los servicios propios de la aplicación.

Los *servicios propios de la infraestructura* son aquellos que tienen muy bajo acoplamiento con la aplicación DNSE3, pero muy alto acoplamiento con la infraestructura de la nube computacional, en donde su funcionalidad se basa en la gestión de la misma. Los

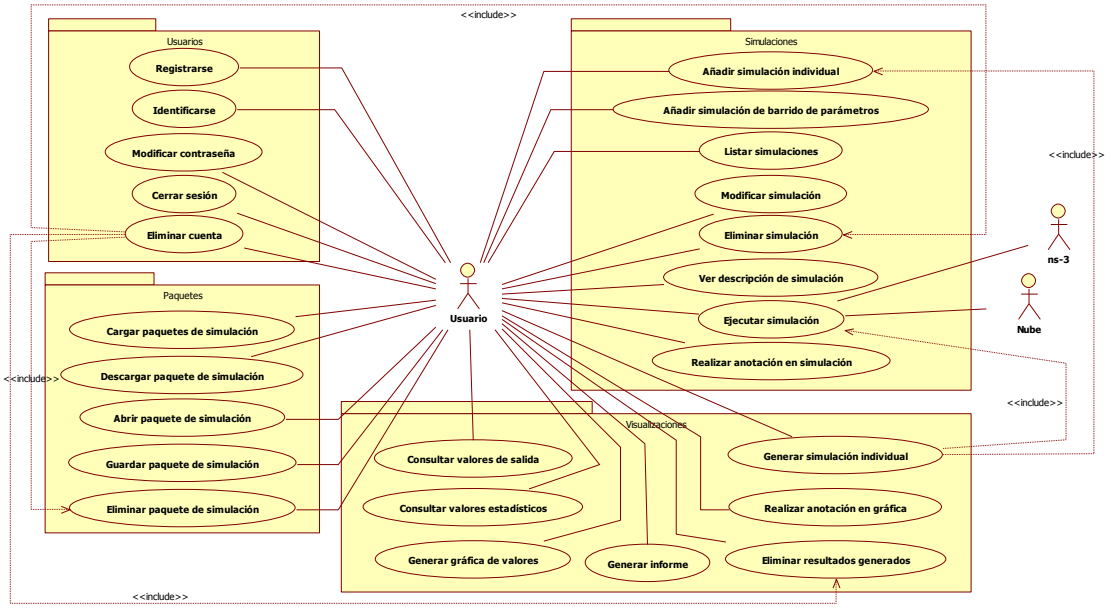

Figura 3.1: Diagrama de casos de uso de la aplicación DNSE3.

tres servicios identificados y clasificados en esta categoría son:

- **SPI-001. Servicio de Almacenamiento.** Este servicio contiene un repositorio de ficheros genérico y autoescalable con la capacidad de almacenar, recuperar, modificar y eliminar ficheros y directorios. Se usará en la aplicación DNSE3, principalmente, para el almacenamiento de los paquetes de simulación de cada usuario, y su contenido asociado.

- **SPI-002. Servicio de Monitorización.** Este servicio se encargará de almacenar las métricas de los otros servicios del sistema, de forma que los servicios puedan registrarse y cancelar su registro en él, y cuando estén registrados puedan publicar y eliminar tanto sus métricas definidas en el servicio, como los valores que van tomando sus métricas a lo largo de su ejecución. En la aplicación DNSE3 se registrarán y publicarán métricas los servicios de colas (SGA-001) y los servicios de simulación (SGA-002), y las consultarán los servicios de escalado (SPI-003).

- **SPI-003. Servicio de Escalado.** Este servicio se encargará de realizar la escalabilidad horizontal de los servicios de la nube computacional. Para ello, permite la definición, modificación y eliminación de una serie de reglas, cuya función es controlar los valores de las métricas de un tipo de servicios. Dependiendo del valor que tomen estas métricas, el servicio realizará una escalabilidad horizontal hacia arriba (ejecutando un nuevo servicio) o hacia abajo (teminando la ejecución de un servicio) según se haya definido. En la aplicación DNSE3 las reglas controlarán los valores de los servicios de colas (SGA-001) y los servicios de simulación (SGA-002) para realizar un escalado horizontal de los servicios de simulación (SGA-002).

Los *servicios generales de la aplicación* son aquellos que tienen un nivel de acoplamiento bajo con la aplicación DNSE3 (y, por tanto, podrán reutilizarse en otras aplicaciones futuras), pero no tienen relación con la infraestructura de la nube computacional. Los tres tipos de servicios que están identificados en esta categoría son:

- **SGA-001. Servicio de Colas.** Este servicio contiene una cola genérica para realizar el procesamiento de elementos de forma asíncrona. Los servicios podrán insertar elementos en la cola en los que adjuntarán un mensaje con el contenido a procesar por otro servicio. Los mensajes tendrán un estado de ejecución asociado, siendo "pendiente" cuando son insertados, "en ejecución" una vez el servicio lo ha recogido para procesar, y "finalizado" cuando el servicio ha terminado de procesarlo y así lo ha indicado. También contendrá un tiempo de vida de ejecución estimado que servirá para conocer si la ejecución se está llevando a cabo o, en caso contrario, ha fallado y no se le ha informado de ello; en este último caso el elemento pasará a estado "pendiente" nuevamente. Por otra parte, este servicio se registrará en un servicio de monitorización (SPI-002) para publicar sus métricas. En la aplicación DNSE3, será el servicio de orquestación (SPA-001) el que insertará los elementos de simulación, y serán los servicios de simulación (SGA-002) los encargados de recogerlos y procesarlos.

- **SGA-002. Servicio de Simulación.** Este servicio se encargará de ejecutar las simulaciones individuales utilizando el simulador de red *ns-3*. Recogerá las simulaciones pendientes de procesar de uno de los servicios de colas (SGA-001) registrados en el servicio de monitorización (SPI-002), las procesará y a su término indicará que han sido procesadas. Estos elementos tendrán un mensaje con una estructura del contenido conocida por el servicio. Si durante la ejecución el tiempo de vida estimado por un elemento de cola está próximo a alcanzar el tiempo dedicado a la simulación, el servicio lo actualizará para indicar que aún continúa en ejecución. Por otra parte, el servicio se registrará en el servicio de monitorización (SPI-002) y publicará sus métricas y, de forma periódica, los valores asociados a ellas. En la aplicación DNSE3, el mensaje de un elemento de la cola contendrá la referencia al directorio del servicio de almacenamiento (SPI-001) donde se encuentran los ficheros de simulación individual a ejecutar, y será en ese mismo destino donde almacene los resultados de la simulación. A su vez, en la aplicación éste será el servicio que se emplee para realizar la escalabilidad horizontal de la misma.

- **SGA-003. Servicio de Estadística.** Este servicio ofrecerá una serie de métodos estadísticos implementados en él que se usarán para calcular estadísticas sobre un conjunto de datos. Tendrá la capacidad de generar, calcular, obtener y eliminar una estadística, utilizando para ello un conjunto de valores de entrada y un conjunto de valores para los parámetros del método estadístico, sobre los que realizará el cálculo para obtener un conjunto de valores de salida. En la aplicación DNSE3, el servicio de orquestación (SPA-001) será el encargado de formatear el conjunto de valores de entrada adaptándolos al formato establecido en el método estadístico, así como proporcionar el conjunto de valores para los parámetros. Posteriormente, una vez se haya realizado el cálculo de estadística, recogerá el conjunto de valores de salida obtenidos para su almacenamiento.

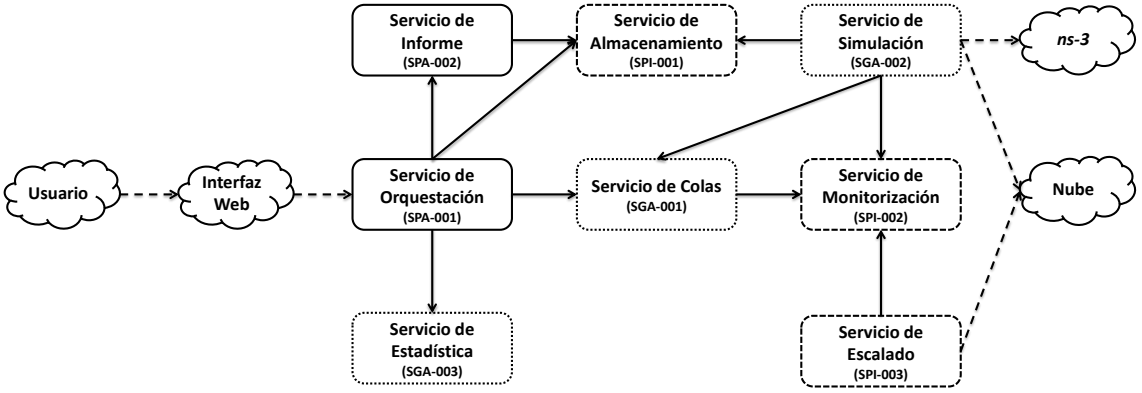

Figura 3.2: Diagrama de arquitectura de servicios de la aplicación DNSE3.

Finalmente, los *servicios propios de la aplicación* son aquellos que tienen un nivel de acoplamiento muy alto con respecto a la aplicación DNSE3, puesto que formarán parte de su núcleo y tendrán funcionalidades muy concretas, y no tienen apenas relación directa con la parte de infraestructura de la nube computacional. Para esta categoría se han definido dos servicios, que son:

- **SPA-001. Servicio de Orquestación.** Este servicio será el encargado de realizar toda la orquestación de la aplicación DNSE3, conociendo las acciones a llevar a cabo para cada una de sus funcionalidades y llevándolas a cabo de forma ordenada a través de la interacción con el resto de servicios que forman la aplicación. En especial, este servicio contendrá la funcionalidad de gestión de las cuentas de usuario y de las sesiones de las mismas, y llevará a cabo las tareas de división de una simulación de barrido de parámetros en tantas simulaciones individuales como sean necesarias, y del formateado y filtrado de todos los datos de los ficheros que forman una simulación para realizar otras acciones (por ejemplo, el cálculo de una estadística o la generación de una gráfica).

- **SPA-002. Servicio de Informe.** Este servicio será el encargado de generar un informe completo de un paquete de simulación de la aplicación DNSE3. Para ello, el servicio de orquestación (SPA-001) le indicará el directorio del servicio de almacenamiento (SPI-001) donde está almacenado el paquete de simulación y el servicio generará un informe con los valores descriptivos del mismo, de sus simulaciones y de los resultados generados para cada una de ellas. También permitirá la descarga del contenido del informe en un formato específico (como mínimo, en XML, PDF y HTML) y su eliminación.

Todos estos servicios anteriores, junto con los agentes externos, formarán todo el entorno de la aplicación DNSE3. La interacción entre todos ellos está definida en la Figura 3.2, donde un usuario interaccionará con la interfaz Web, y ésta ejecutará las acciones en el Servicio de Orquestación (SPA-001). Este servicio se encargará de solicitar el resto de acciones a los otros servicios: para la gestión de los datos de la aplicación el Servicio de Almacenamiento (SPI-001), para la gestión de las simulaciones el Servicio de Colas (SGA-001), para la gestión de las estadísticas de una simluación el Servicio de Estadística

(SGA-003) y para la gestión de los informes de un paquete de simulación el Servicio de Informe (SPA-002). El Servicio de Monitorización (SPI-002) tendrá registrados un Servicio de Colas (SGA-001) y varios Servicios de Simulación (SGA-002), y el Servicio de Escalado (SPI-003) consultará sus valores para llevar a cabo la escalabilidad horizontal con la nube computacional. El Servicio de Almacenamiento (SPI-001) será consultado por otros servicios durante su ejecución, como son el Servicio de Informe (SPA-002) o un Servicio de Simulación (SGA-002). Este último además recogerá las simulaciones individuales de un Servicio de Colas (SGA-001) e interactuará con el simulador de red *ns-3* para su ejecución.

Todo el contenido de esta sección está detallado y ampliado con información más precisa en el informe técnico de Arquitectura de Software [33] que acompaña este Trabajo Fin de Máster.

3.4. Servicios del sistema

Una vez que se han identificado todas las funcionalidades principales de la aplicación DNSE3 y se ha realizado la descomposición de la misma en varios servicios, falta analizar qué tipo de funcionalidades va a llevar a cabo cada uno de los servicios de todas las funcionalidades principales, así como la realización de la elicitación de requisitos software y el análisis y diseño orientado a recursos (ROA/D) para cada uno de ellos. De esta forma, los servicios quedan definidos de forma completa para poder llevar a cabo su implementación posterior. En las siguientes subsecciones se describirán los aspectos anteriores para cada uno de los ocho servicios de los que consta el DNSE3.

Un aspecto común que seguirán cada uno de ellos es su arquitectura lógica de software, basada en la propuesta por la metodología ROA/D. Esta arquitectura consta de cuatro capas: una capa superior de *interfaz de usuario*, no presente en cada servicio de forma específica, pero sí presente en la aplicación DNSE3 interactuando con el servicio de forma directa o a través del Servicio de Orquestación (SPA-001); una segunda capa de *recursos* donde se realizarán las transferencias de representaciones invocando los métodos HTTP sobre los recursos y que definirá la interfaz programática del servicio; una tercera capa de *objetos* que define el modelo de datos del servicio a través de los objetos y la lógica asociada a ellos, y la última capa de *datos* encargada de la persistencia de los objetos e, indirectamente, de los recursos.

3.4.1. Servicio de Almacenamiento

El Servicio de Almacenamiento (SPI-001) va a realizar las tareas de almacenamiento y gestión de ficheros en una estructura de directorios, permitiendo el listado, creación, modificación y eliminación de una estructura de directorios contenida en el repositorio de ficheros del servicio, así como la subida, descarga, actualización y eliminación de los ficheros en dichos directorios.

La aplicación DNSE3 tendrá una interacción directa y pasiva con otros servicios de la misma, como son:

- El *Servicio de Orquestación (SPA-001)* contendrá el esquema de la estructura de directorios que estará definida para a aplicación y será el que introduzca los ficheros

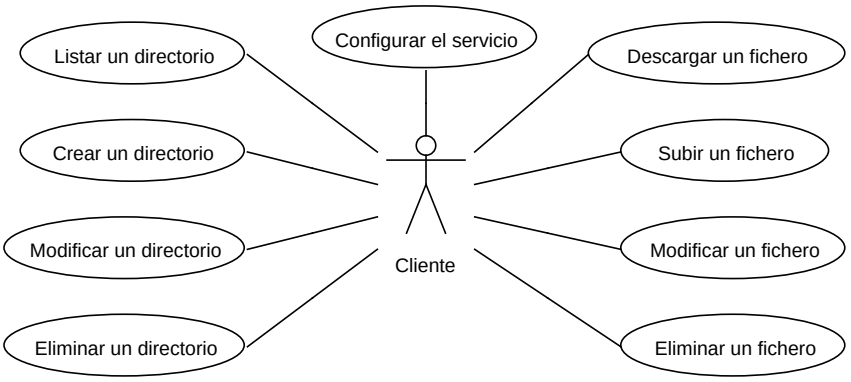

Figura 3.3: Diagrama de casos de uso de los escenarios de uso del Servicio de Almacenamiento.

y directorios por cada paquete de simulación de cada usuario, y también recogerá los ficheros de un paquete de simulación para poder ser descargados por un usuario.

- El *Servicio de Informe (SPA-002)* recogerá todos los datos relativos a un paquete de simulación del lugar que se le ha indicado en el repositorio de ficheros para generar su informe, y almacenará los resultados en el mismo lugar, así como los ficheros generados en distintos formatos.

- El *Servicio de Simulación (SGA-002)* recogerá todos los datos relativos a una simulación individual del lugar que se le ha indicado en el repositorio de ficheros para ejecutar la simulación, y almacenará los resultados obtenidos en el mismo lugar.

A estos tres servicios se les denominará *clientes*, y serán los elementos software del DNSE3 que interactúen con este servicio de almacenamiento. También este servicio interactuará de forma activa con otro elemento software como es el *repositorio de ficheros local*, referente al almacenamiento físico donde estarán alojados los directorios y ficheros del servicio. Este repositorio de ficheros local podrá estar localizado en la propia instancia del servicio o en un servicio externo de almacenamiento de ficheros en la nube. De este último se podrían aprovechar las ventajas de sensación de almacenamiento "infinito", garantía de la integridad del contenido al estar distribuido y replicado, o acceso ubicuo al mismo.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Crear, listar, modificar y eliminar directorios del repositorio, contenidos en el directorio raíz o en subdirectorios a partir de éste.

- Cargar, descargar, actualizar y eliminar ficheros de los directorios del repositorio, ya sea el directorio raíz o los subdirectorios a partir de éste.

Estas funciones están reflejadas en el diagrama de casos de uso de la Figura 3.3, en donde se relacionan con los elementos software de su entorno. Además, a través de la interfaz de comandos, se podrá configurar el servicio para definir su puerto de escucha.

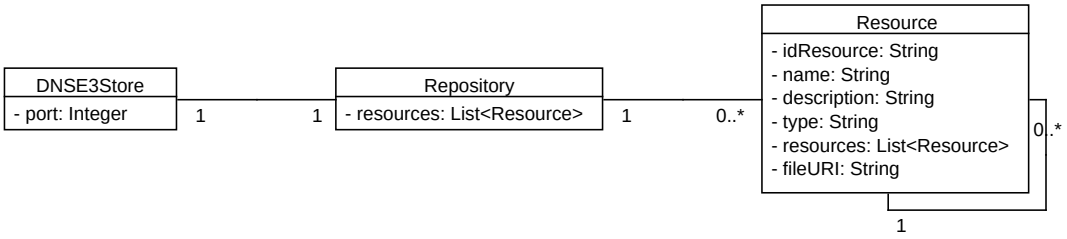

Figura 3.4: Diagrama de modelo de dominio del Servicio de Almacenamiento.

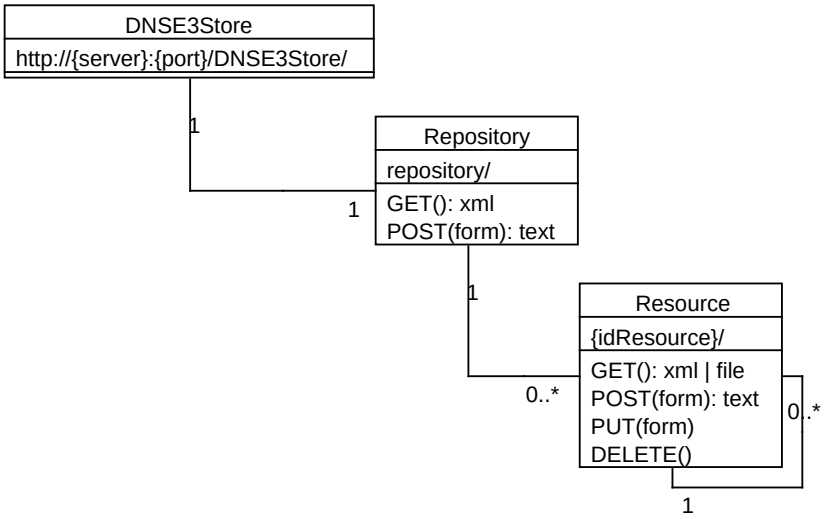

Figura 3.5: Diagrama de recursos del Servicio de Almacenamiento.

Internamente, el servicio va a tener un modelo de dominio compuesto por tres entidades: *DNSE3Store* como entidad principal del servicio, *Repository* como el repositorio de ficheros y que será el directorio raíz del servicio, y *Resource* que representa tanto a un directorio como a un fichero del repositorio. Las propiedades y relaciones de cada entidad pueden verse en la Figura 3.4.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de tres recursos con el mismo nombre que los objetos, siguiendo la jerarquía representada en la Figura 3.5, y en la que se representan también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar el informe técnico de Servicio de Almacenamiento [34] que acompaña este Trabajo Fin de Máster.

3.4.2. Servicio de Colas

El Servicio de Colas (SGA-001) va a realizar las tareas de gestión de los elementos de una cola para la comunicación asíncrona entre diferentes servicios de un mismo sistema basado en servicios. La libre disposición del contenido del mensaje de los elementos de la cola permite que el servicio sea generalista, es decir, que pueda reutilizarse en en otros sistemas. También realizará la publicación de métricas en un Servicio de Monitorización (SPI-002), relativas a la ocupación de la cola y el ratio de procesamiento de sus elementos por los servicios externos.

La aplicación DNSE3 tendrá una interacción directa, tanto activa como pasiva, con otros elementos software o servicios de la misma. A estos servicios se les denominará *clientes*, y son:

- El *Servicio de Orquestación (SPA-001)* incluirá en el mensaje del elemento de la cola, para cada simulación individual, el contenido con el acceso a los recursos de la simulación para la recogida de los datos y deposición de los resultados, y añadirá este elemento a la cola del servicio a la espera de ser procesado.

- El *Servicio de Simulación (SGA-002)* solicitará a la cola el siguiente elemento de tipo simulación que está pendiente de procesar, del que obtendrá el contenido del mensaje para procesarlo y dejar los resultados de la simulación, indicando en éste que está procesado. En el caso de que el tiempo de simulación sobrepase el tiempo de vida estimado, se actualizará para indicar al servicio que el elemento sigue en procesamiento.

- El *Servicio de Monitorización (SPI-002)* recibirá el registro de este servicio, la publicación de sus métricas y, de forma periódica, los valores que van tomando dichas métricas a lo largo del tiempo.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Insertar elementos en la cola para que se realice el procesamiento..

- Solicitar el elemento siguiente de la cola pendiente de procesar y modificar el tiempo de vida de un elemento de la cola que está siendo procesado.

- Consultar uno o todos los elementos de la cola y eliminar un elemento de la cola que ya ha sido procesado de forma satisfactoria.

Estas funciones están reflejadas en el diagrama de casos de uso de la Figura 3.6, en donde se relacionan con los elementos software de su entorno. Además, a través de la interfaz de comandos, se podrá configurar el servicio para definir su puerto de escucha y la URI del Servicio de Monitorización (SPI-002) en el que se registrará y publicará sus métricas y valores asociados.

Internamente, el servicio va a tener un modelo de dominio compuesto por tres entidades: *DNSE3Queue* como entidad principal del servicio, *Elements* como el conjunto de elementos que están en la cola, independientemente de su estado de procesamiento, y *Element* que representa a un elemento de la cola en particular. Las propiedades y relaciones de cada entidad pueden verse en la Figura 3.7.

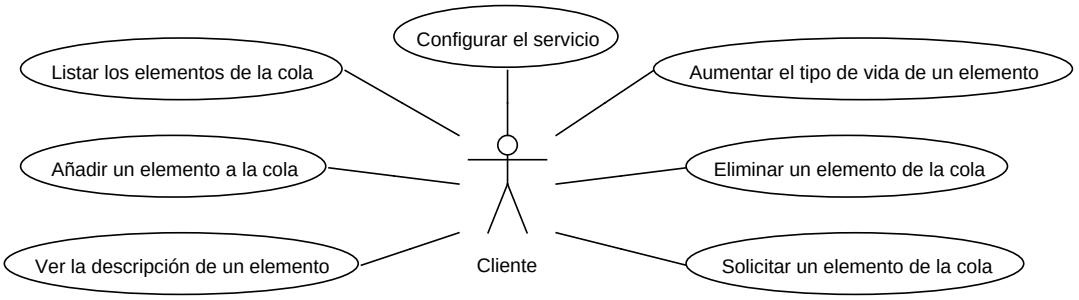

Figura 3.6: Diagrama de casos de uso de los escenarios de uso del Servicio de Colas.

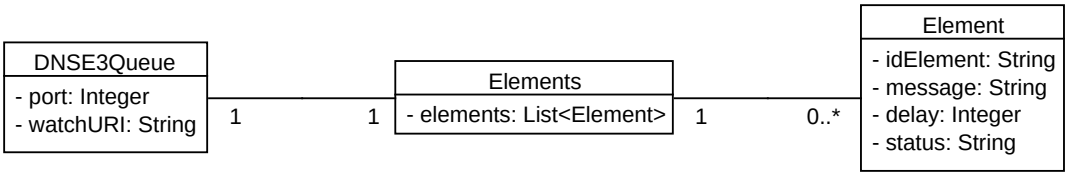

Figura 3.7: Diagrama de modelo de dominio del Servicio de Colas.

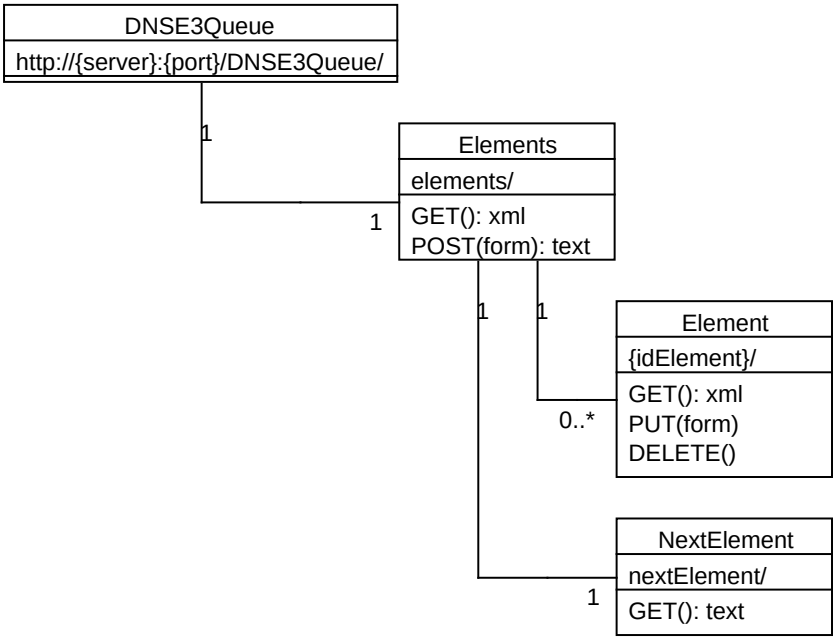

Figura 3.8: Diagrama de recursos del Servicio de Colas.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de tres recursos con el mismo nombre que los objetos, además de un recurso a mayores llamado *NextElement* que representa al siguiente elemento pendiente de ser procesado. Los recursos tendrán una jerarquía como la que está representada en la Figura 3.8, y en la que están representados también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar el informe técnico de Servicio de Colas [35] que acompaña este Trabajo Fin de Máster.

3.4.3. Servicio de Escalado

El Servicio de Escalado (SPI-003) va a realizar las tareas de gestión de los servicios que están ejecutándose en un sistema de nube computacional a través de una escalabilidad horizontal hacia arriba (instanciando nuevas imágenes de un servicio) o hacia abajo (terminando una instancia de un servicio en ejecución). Para la gestión de dichos servicios se definen una serie de reglas que, a través de un conjunto de valores de referencia, consultan las métricas correspondientes en un Servicio de Monitorización (SPI-002) determinado que son publicadas por otros servicios y que permiten conocer el estado actual del sistema en el entorno de nube computacional para adaptar los servicios en ejecución actuales a los requisitos reales de la carga del sistema.

La aplicación DNSE3 tendrá una interacción directa, tanto activa como pasiva, con otros elementos software de la misma, y son:

- Los *clientes* serán los encargados de la creación, consulta, modificación y eliminación de las reglas que llevan la gestión de las tareas de escalabilidad horizontal en el sistema.

- El *Servicio de Monitorización (SPI-002)* será consultado para recoger los valores que toman las métricas de un tipo de servicio en particular según las especificaciones definidas en la regla.

- La *nube computacional* sobre la que realizará las acciones correspondientes para llevar a cabo la escalabilidad horizontal, que en el caso de que sea hacia arriba lanzará una nueva instancia de un tipo de servicio determinado, y en el caso de que sea hacia abajo terminará una instancia de un tipo de servicio, siguiendo el criterio de menor ocupación actual de sus recursos.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Listar todas las reglas que están definidas en el servicio.

- Insertar una nueva regla en el servicio que compare el valor de una métrica definida en un servicio por el de un valor de referencia para que gestione el escalado de un servicio.

- Consultar, modificar y eliminar una regla existente en el servicio.

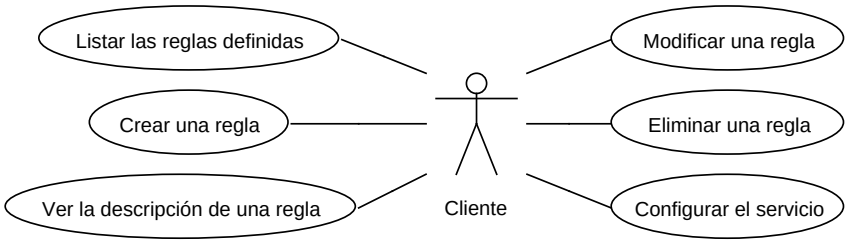

Figura 3.9: Diagrama de casos de uso de los escenarios de uso del Servicio de Escalado.

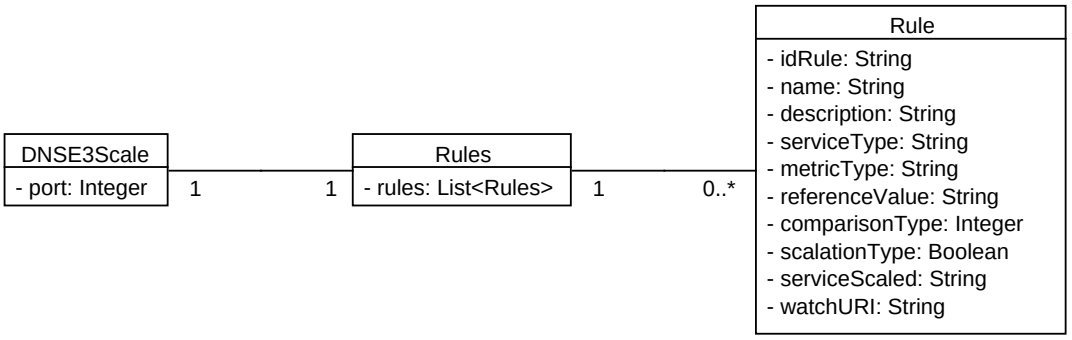

Figura 3.10: Diagrama de modelo de dominio del Servicio de Escalado.

Estas funciones están reflejadas en el diagrama de casos de uso de la Figura 3.9, en donde se relacionan con los elementos software de su entorno. Además, a través de la interfaz de comandos, se podrá configurar el servicio para definir su puerto de escucha.

Internamente, el servicio va a tener un modelo de dominio compuesto por tres entidades: *DNSE3Scale* como entidad principal del servicio, *Rules* como el conjunto de reglas del servicio, y *Rule* que representa a una regla en particular. Las propiedades y relaciones de cada entidad pueden verse en la Figura 3.10.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de tres recursos con el mismo nombre que los objetos, con una jerarquía como la que está representada en la Figura 3.11, y en la que están representados también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar en el informe técnico de Servicio de Escalado [36] que acompaña este Trabajo Fin de Máster.

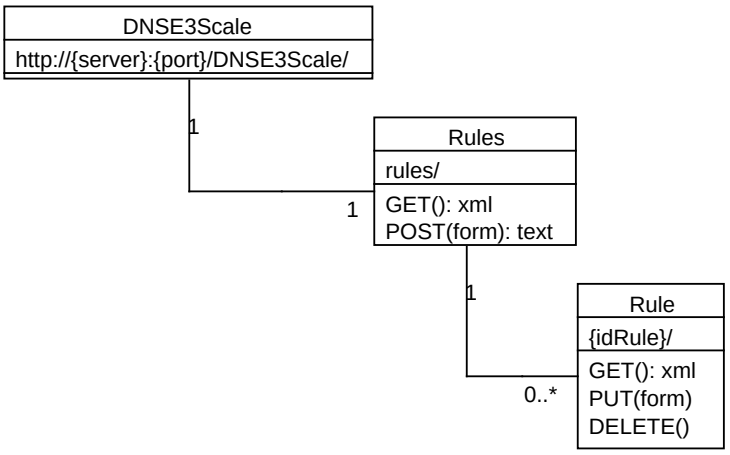

Figura 3.11: Diagrama de recursos del Servicio de Escalado.

3.4.4. Servicio de Estadística

El Servicio de Estadística (SGA-003) realizará la aplicación de un método estadístico predefinido en el servicio sobre un conjunto de datos de entrada que, junto con una configuración de parámetros personalizada, obtiene un conjunto de datos de salida con los resultados estadísticos de los datos. Los método estadísticos a utilizar estarán implementados en el propio servicio, y cada método estadístico va a poder contener una serie de conjuntos de datos de entrada, configuración de parámetros y conjunto de datos de salida a los que se les llamará *estadística*.

La aplicación DNSE3 tendrá una interacción directa de forma pasiva con un único elemento software de la misma, al que se le llamará en general *cliente*, y:

- El *Servicio de Orquestación (SPA-001)* será el encargado de la gestión de las estadísticas de los resultados de una simulación, aplicados para uno o varios de los métodos estadísticos, así como la obtención de la descripción de los métodos estadísticos que hay implementados en el servicio para ser mostrados al usuario final.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Listar todos los métodos estadísticos que están implementados en el servicio y obtener la descripción detallada de un método estadístico en concreto.

- Listar las estadísticas definidas para un método estadístico en particular, así como crear, obtener la descripción detallada y eliminar una estadística en concreto.

Estas funciones están reflejadas en el diagrama de casos de uso de la Figura 3.12, en donde se relacionan con los elementos software de su entorno. Además, a través de la interfaz de comandos, se podrá configurar el servicio para definir su puerto de escucha.

Internamente, el servicio va a tener un modelo de dominio compuesto por siete entidades: *DNSE3Stats* como entidad principal del servicio, *Methods* como el conjunto de

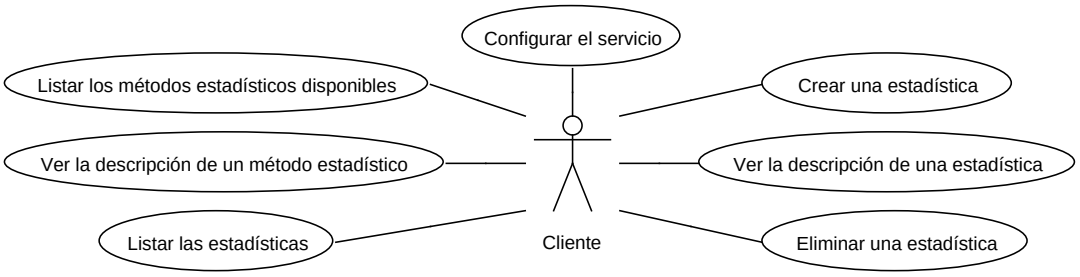

Figura 3.12: Diagrama de casos de uso de los escenarios de uso del Servicio de Estadística.

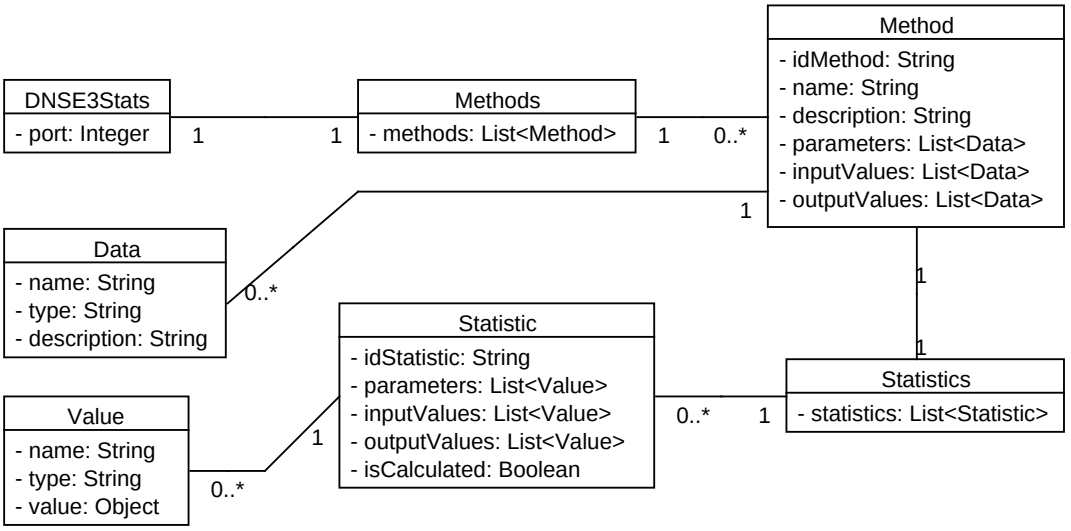

Figura 3.13: Diagrama de modelo de dominio del Servicio de Estadística.

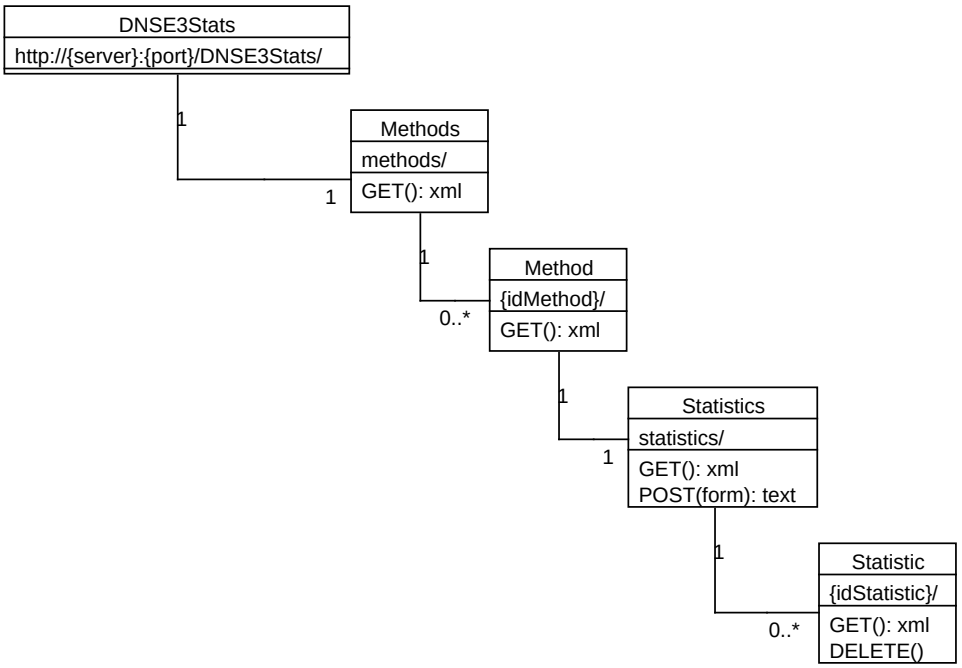

Figura 3.14: Diagrama de recursos del Servicio de Estadística.

métodos estadísticos implementados en servicio, *Method* que representa a un método estadístico en particular, *Statistics* como el conjunto de estadísticas de un método estadístico, *Statistic* como una estadística en particular, *Data* como la descripción de un dato, y *Value* como el valor de un dato. Las propiedades y relaciones de cada entidad pueden verse en la Figura 3.13.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de cinco recursos con el mismo nombre que los objetos anteriores, a excepción de *Data* y *Value* que formarán parte del recurso. El servicio tendrá una jerarquía como la que está representada en la Figura 3.14 y en la que están representados también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar en el informe técnico de Servicio de Estadística [37] que acompaña este Trabajo Fin de Máster.

3.4.5. Servicio de Informe

El Servicio de Informe (SPA-002) realizará la generación de un informe en un formato preestablecido (por ejemplo, XML, HTML o PDF) de un paquete de simulación de un usuario de la aplicación DNSE3, sobre el que se representará la información general de dicho paquete, así como la información de cada una de las simulaciones individuales y de barrido de parámetros,que contiene y sus resultados generados sobre datos en bruto, gráficas y estadísticas. También se añadirán aquellas anotaciones tomadas en las simulaciones

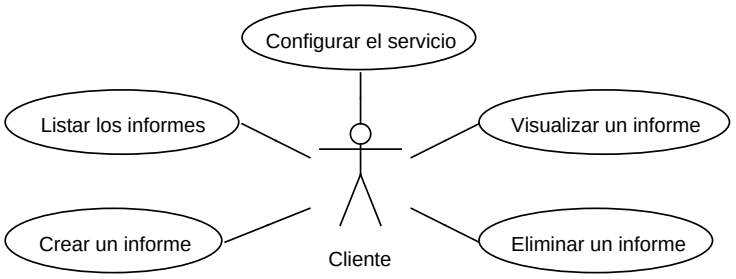

Figura 3.15: Diagrama de casos de uso de los escenarios de uso del Servicio de Informe.

y en las gráficas.

La aplicación DNSE3 tendrá una interacción directa, tanto de forma activa como pasiva, con dos elementos software de la misma, como son:

- El *Servicio de Orquestación (SPA-001)* o *cliente* que será el encargado de solicitar a este servicio la generación de un informe para un paquete de simulación, la descarga del mismo en un formato determinado o la eliminación de un informe que ya no es necesario.

- El *Servicio de Almacenamiento (SPI-001)* que será al que se le soliciten todos los datos de un paquete de simulación para la realización del informe, según la URI que se le indique al servicio, y donde se almacenarán los resultados del informe y los ficheros generados asociados al mismo.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Listar todos los informes que se han generado en el servicio.

- Visualizar, crear y eliminar un informe del servicio.

Estas funciones están reflejadas en el diagrama de casos de uso de la Figura 3.15, en donde se relacionan con los elementos software de su entorno. Además, a través de la interfaz de comandos, se podrá configurar el servicio para definir su puerto de escucha.

Internamente, el servicio va a tener un modelo de dominio compuesto por tres entidades: *DNSE3Report* como entidad principal del servicio, *Reports* como el conjunto de informes del servicio, y *Report* como un informe en particular. Las propiedades y relaciones de cada entidad pueden verse en la Figura 3.16.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de tres recursos con el mismo nombre que los objetos anteriores, formando una jerarquía como la que está representada en la Figura 3.17 y en la que están representados también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar en el informe técnico de Servicio de Informe [38] que acompaña este Trabajo Fin de Máster.

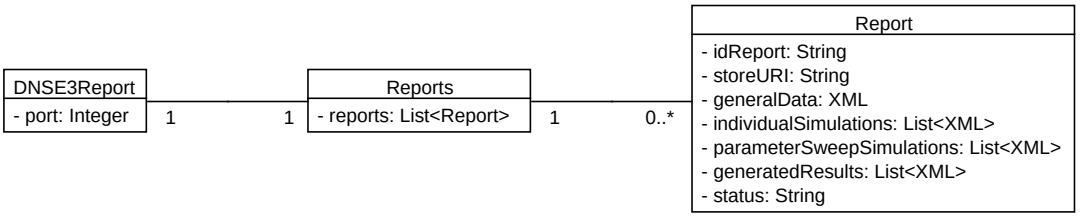

Figura 3.16: Diagrama de modelo de dominio del Servicio de Informe.

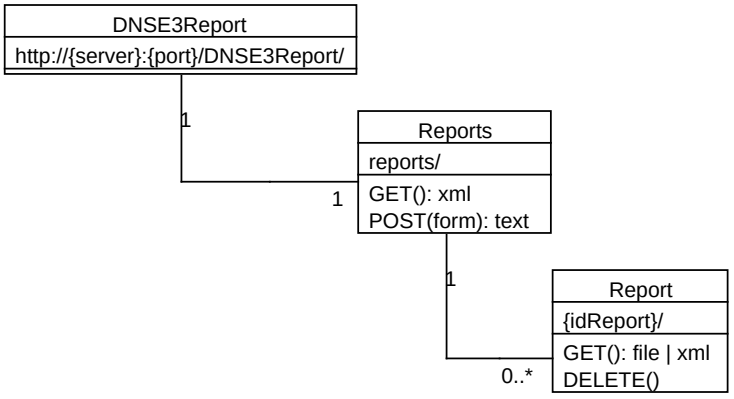

Figura 3.17: Diagrama de recursos del Servicio de Informe.

3.4.6. Servicio de Monitorización

El Servicio de Monitorización (SPI-002) realizará la monitorización de servicios a través de una serie de métricas definidos en ellos. Los servicios podrán registrarse en el servicio y realizar la publicación de diversas métricas y los valores correspondientes que van tomando a lo largo del tiempo de su ejecución. Estos valores pueden ser consultados posteriormente por otros servicios para poder gestionar los servicios registrados. También se podrán eliminar las métricas (junto con sus valores) y cancelar el registro de los servicios en cualquier momento.

La aplicación DNSE3 tendrá una interacción directa y de forma pasiva con tres elementos software de la misma, como son:

- El *Servicio de Colas (SGA-001)* y *Servicio de Simulación (SGA-002)*, como elementos que ofrecen información al servicio, registrarán sus servicios y publicarán sus métricas, e irán publicando de forma periódica los valores que van tomando las métricas en ese instante.

- El *Servicio de Escalado (SPI-003)*, como elemento que recoge información del servicio, consultará los servicios que están registrados tanto de tipo "colas" como de tipo "simulación", y recogerá los últimos valores almacenados para las métricas que tienen registradas. Estos valores servirán para analizar sus reglas y poder realizar el escalado horizontal de la aplicación DNSE3.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Consulta de listados y descripciones de los servicios (todos o de un tipo determinado) que están registrados, las métricas que están publicadas y los valores de las métricas almacenados (todos o el última publicado).

- Registro y cancelación del registro de un servicio.

- Publicación y eliminación de una métrica de un servicio.

- Publicación del valor de una métrica de un servicio.

Estas funciones están reflejadas en el diagrama de casos de uso de la Figura 3.18, en donde se relacionan con los elementos software de su entorno. Además, a través de la interfaz de comandos, se podrá configurar el servicio para definir su puerto de escucha.

Internamente, el servicio va a tener un modelo de dominio compuesto por siete entidades: *DNSE3Watch* como entidad principal del servicio, *Services* como el conjunto de servicios registrados, *Service* como un servicio registrado en particular, *Metrics* como el conjunto de métricas publicadas de un servicio, *Metric* como una métrica publicada en particular, *Values* como el conjunto de valores publicados para una métrica y *Value* como un valor publicado en particular. Las propiedades y relaciones de cada entidad pueden verse en la Figura 3.19.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de seis recursos con el mismo nombre que los objetos anteriores, a excepción de *Value* que estará integrado en

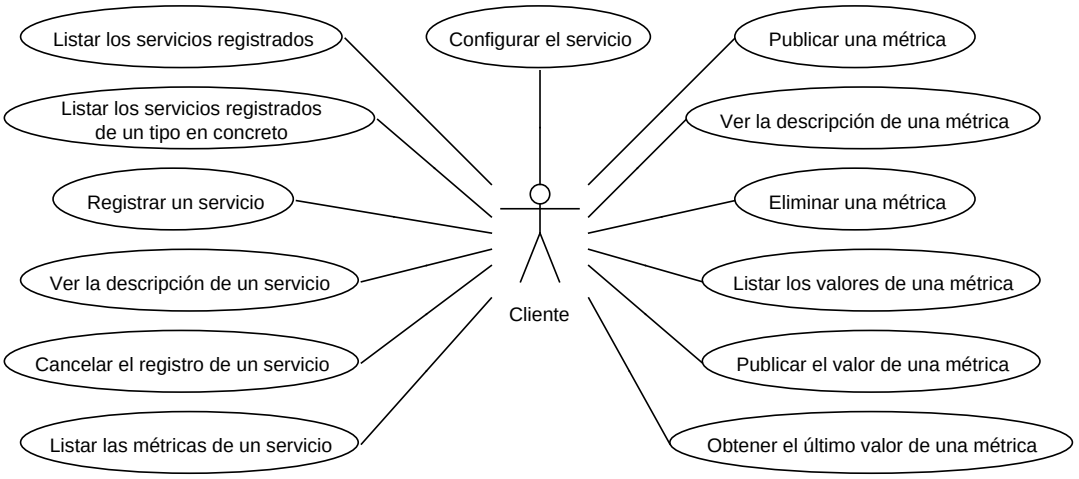

Figura 3.18: Diagrama de casos de uso de los escenarios de uso del Servicio de Monitorización.

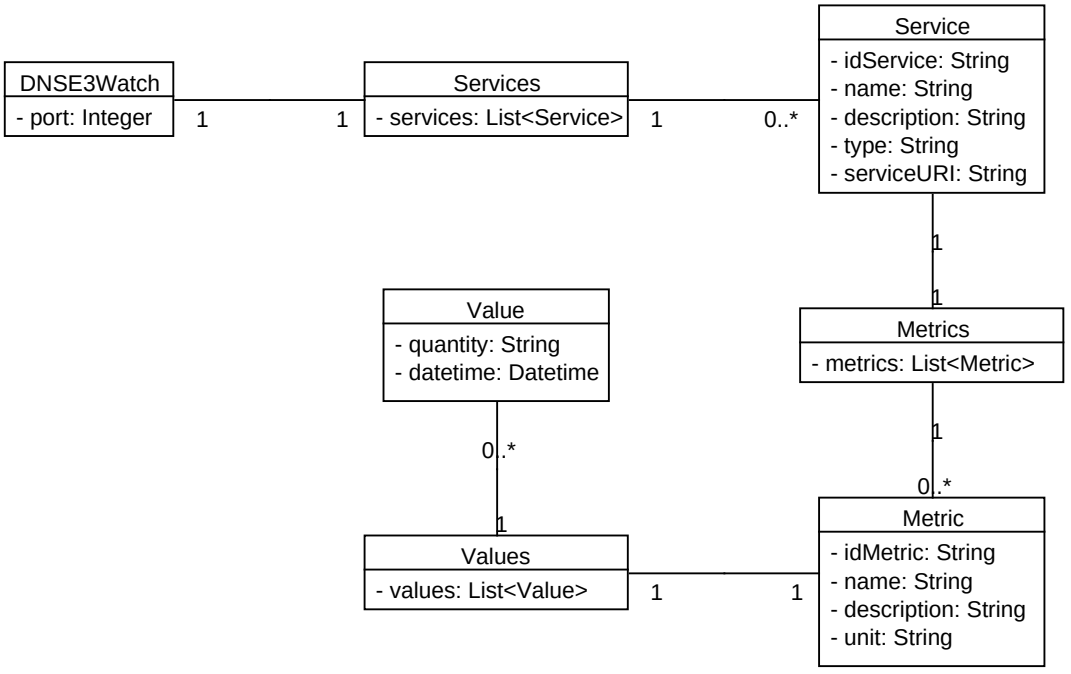

Figura 3.19: Diagrama de modelo de dominio del Servicio de Monitorización.

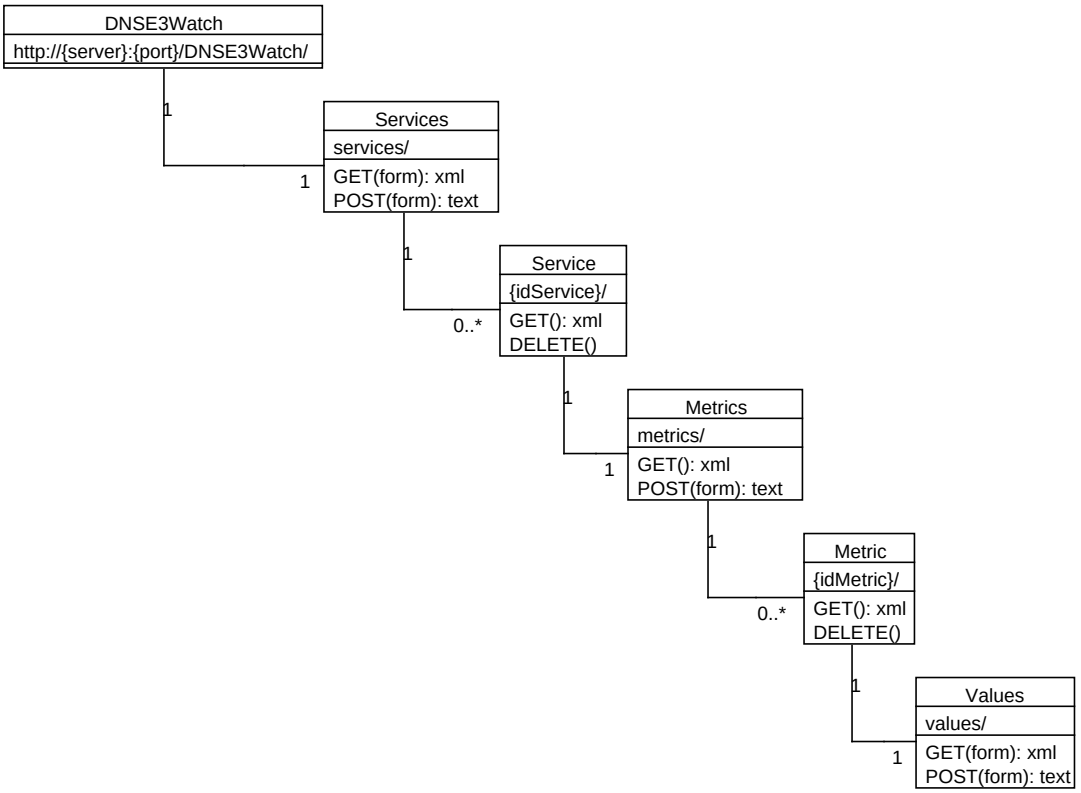

Figura 3.20: Diagrama de recursos del Servicio de Monitorización.

Values. Este servicio tendrá una jerarquía como la que está representada en la Figura 3.20 y en la que están representados también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar en el informe técnico de Servicio de Monitorización [39] que acompaña este Trabajo Fin de Máster.

3.4.7. Servicio de Orquestación

El Servicio de Orquestación (SPA-001) realizará la orquestación de toda la aplicación DNSE3, es decir, una vez recibida una acción por parte del usuario a través de la interfaz Web, conocerá los pasos que hay que realizar y los llevará a cabo de forma ordenada llamando a cada uno de los otros servicios que componen la aplicación. También contendrá la gestión de las cuentas de usuario y la gestión de las sesiones de dichas cuentas para la personalización del entorno de trabajo, compuesto de paquetes de simulación, para cada uno de los usuarios.

Cabe destacar que este servicio llevará a cabo dos funcionalidades relevantes y dignas de mención, como son la descomposición de una simulación de barrido de parámetros en tantas simulaciones individuales como sean necesarias par poder llevar a cabo la simulación, las cuales serán añadidas como elementos en la cola del Servicio de Colas (SGA-001) para su posterior procesado por un Servicio de Simulación (SGA-002), así como un filtrado de los datos recuperados desde los ficheros de resultados de las simulaciones generados por el Servicio de Simulación (SGA-002), que estarán contenidos en el espacio de trabajo del Servicio de Almacenamiento (SPI-001) correspondiente al paquete de simulación que contiene dichas simulaciones, para reformatearlos a la hora de ser pasados a otros componentes del sistema, como por ejemplo para el cálculo de estadísticas por el Servicio de Estadística (SGA-003) o la generación de gráficas por la interfaz Web.

Por tanto, este servicio tendrá una interacción directa con cinco elementos software de la aplicación DNSE3, uno de forma pasiva (la interfaz Web) y cuatro de forma activa (los servicios de la aplicación). Descrita de manera más específica, los elementos software con los que interactúa son:

- La *interfaz Web* del usuario de la que recibe la ejecución de cada una de las acciones que inician los usuarios a través de ella, y sobre la que devuelve los resultados generados de dichas acciones. Será la parte visible de la aplicación DNSE, correspondiente a la capa de *interfaz de usuario* de la metodología ROA/D, en la que este servicio actuará como un servicio más de la aplicación y como un intermediario de las comunicaciones entre la interfaz Web y el resto de servicios de la aplicación.

- El *Servicio de Almacenamiento (SPI-001)* que contendrá almacenada toda la información de la aplicación DNSE3 a través de la creación de las cuentas de usuario y los paquetes de simulación de cada usuario, según una estructura jerárquica de directorios y ficheros en el repositorio bien conocida por el servicio.

- El *Servicio de Colas (SGA-001)* en el que se añadirán las simulaciones individuales pendientes de ejecución, ya sean provenientes de una simulación individual como

tal o de la división de una simulación de barrido de parámetros, y sobre las que se consultará su estado de procesamiento durante la ejecución de la misma.

- El *Servicio de Estadística (SGA-003)* en el que se consultarán los métodos estadísticos disponibles para ser listados al usuario, y sobre el que se enviarán los datos en bruto concretos de una simulación para el cálculo de su estadística, aplicando un método determinado, y recogiendo los resultados de su cálculo.

- El *Servicio de Informe (SPA-002)* en el que se pedirá la generación de un informe a partir de toda la información contenida en un paquete de simulación de un usuario, y del que se solicitarán los ficheros con el contenido del informe en un formato determinado.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Listado, visualización, creación, modificación y eliminación de cuentas de usuario de la aplicación.

- Identificación, consulta y cierre de sesiones de las cuentas de usuario de la aplicación.

- Listado, visualización, carga, descarga, modificación y eliminación de paquetes de simulación de un usuario.

- Listado, visualización, creación, modificación eliminación y ejecución de simulaciones de un paquete de simulación.

- Listado y visualización de los datos de salida generados por la ejecución de una simulación.

- Listado, visualización, creación y eliminación de gráficas para un conjunto de valores de salida de la ejecución de una simulación.

- Listado descriptivo de los métodos estadísticos disponibles en la aplicación para el cálculo de estadísticas de una simulación.

- Listado, visualización, creación y eliminación de estadísticas para un conjunto de valores de salida de la ejecución de una simulación.

- Creación y visualización de anotaciones en simulaciones y gráficas de simulaciones.

- Generación, descarga y eliminación de informes de un paquete de simulación en distintos formatos.

Estas funciones están reflejadas en los diferentes diagramas de casos de uso en donde se relacionan con los elementos software de su entorno, y los cuales se han clasificado según la funcionalidad principal de gestión de las partes de la aplicación. Así, la gestión de usuarios está descrita en la Figura 3.21, la gestión de sesiones esta descrita en la Figura 3.22, la gestión de paquetes de simulación está descrita en la Figura 3.23, la gestión

de simulaciones está descrita en la Figura 3.24, la gestión de visualizaciones está descrita en la Figura 3.25, la gestión de informes está descrita en la Figura 3.26, y la gestión de configuración del servicio está descrita en la Figura 3.27.

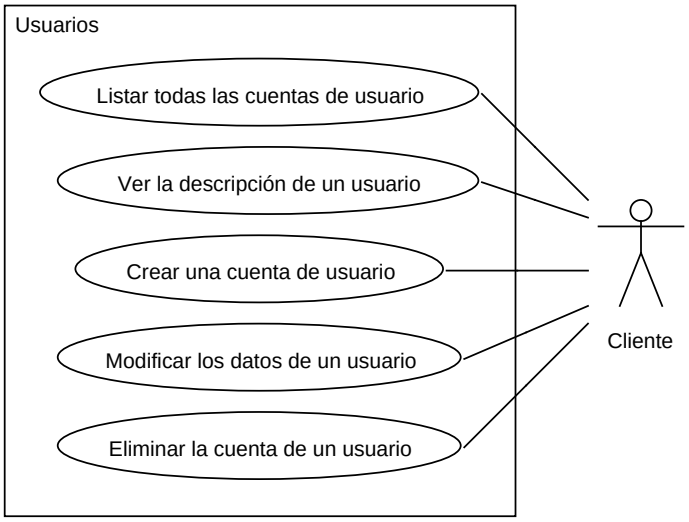

Figura 3.21: Diagrama de casos de uso de los escenarios de uso de la gestión de usuarios del Servicio de Orquestación.

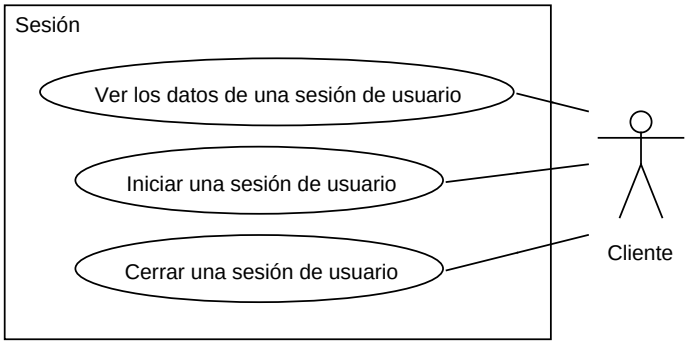

Figura 3.22: Diagrama de casos de uso de los escenarios de uso de la gestión de sesiones del Servicio de Orquestación.

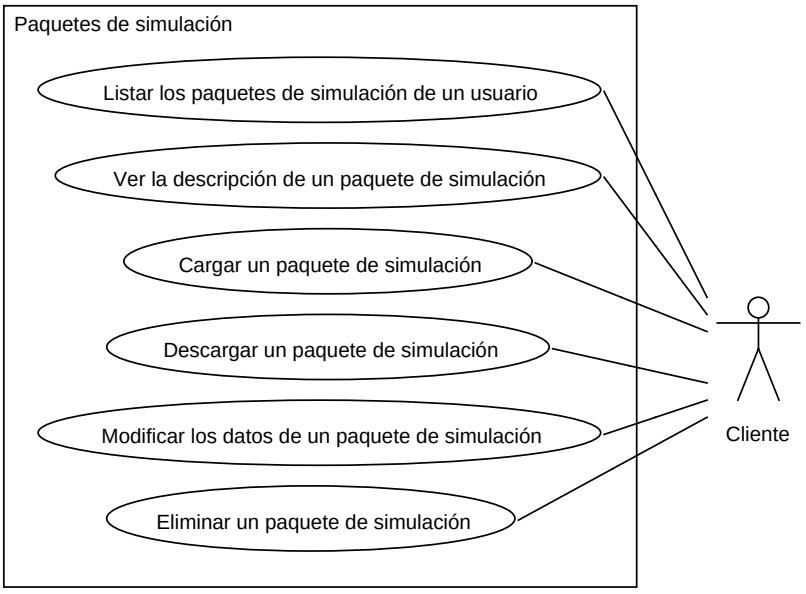

Figura 3.23: Diagrama de casos de uso de los escenarios de uso de la gestión de paquetes de simulación del Servicio de Orquestación.

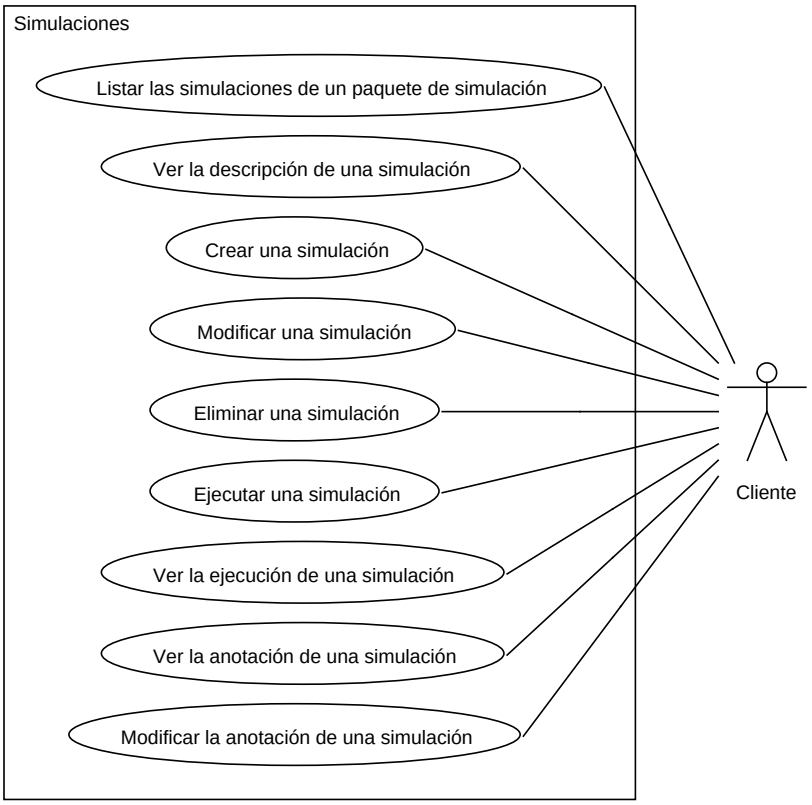

Figura 3.24: Diagrama de casos de uso de los escenarios de uso de la gestión de simulaciones del Servicio de Orquestación.

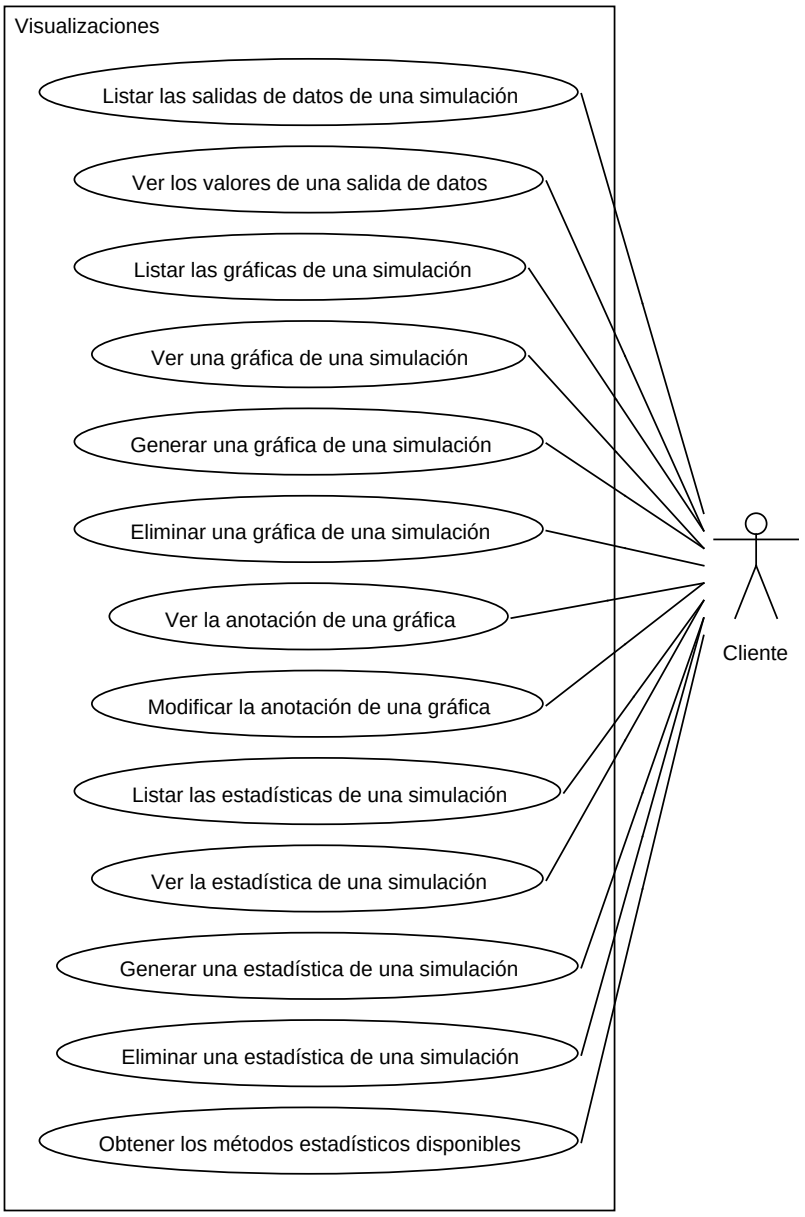

Figura 3.25: Diagrama de casos de uso de los escenarios de uso de la gestión de visualizaciones del Servicio de Orquestación.

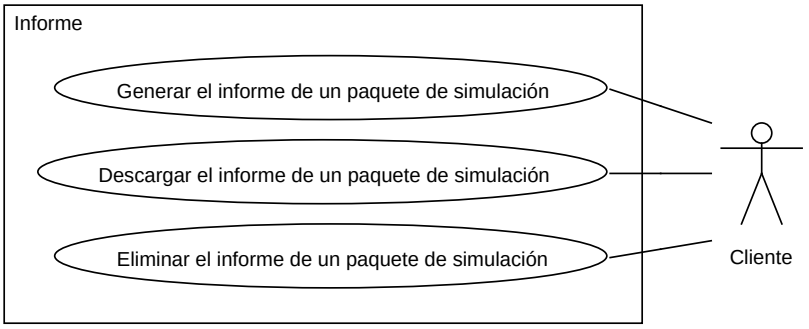

Figura 3.26: Diagrama de casos de uso de los escenarios de uso de la gestión de informes del Servicio de Orquestación.

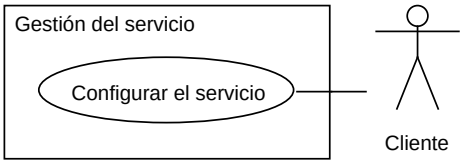

Figura 3.27: Diagrama de casos de uso de los escenarios de uso de la gestión de configuración del Servicio de Orquestación.

Internamente, el servicio va a tener un modelo de dominio compuesto por 19 entidades. La entidad principal del servicio será el *DNSE3Orch*, partiendo de él las entidades relacionadas con los usuarios y la sesión de las cuentas de usuario. Asociados a una cuenta de usuario están los paquetes de simulación, compuesto de simulaciones y un informe de resultados. Estas simulaciones contendrán entidades con respecto a su ejecución, anotación y visualizaciones, como son las estadísticas (y los métodos estadísticos disponibles) y las gráficas (y sus anotaciones). Para la gestión de los datos del conjunto de datos de entrada, de salida y parámetros, se tiene una estructura de datos en el que están representados el dato como tal, su tipo y el conjunto de valores que toma en un momento determinado. El conjunto de entidades anterior puede verse en la Figura 3.28, así como las propiedades de cada una de ellas y las relaciones que tienen entre sí.

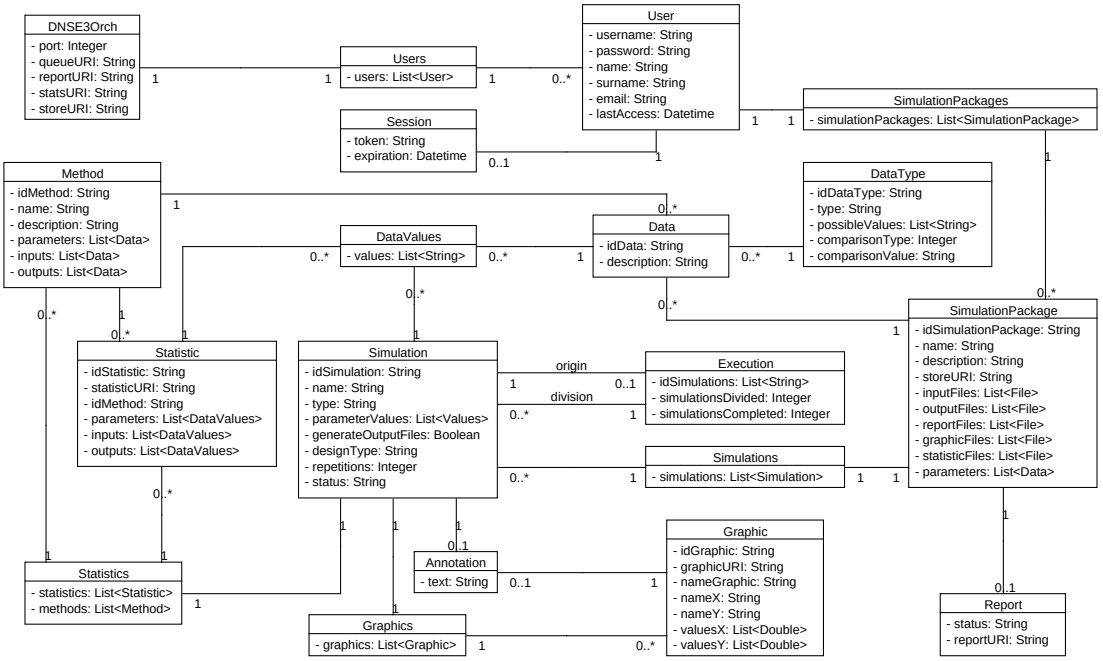

Figura 3.28: Diagrama de modelo de dominio del Servicio de Orquestación.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de veinte recursos, donde muchos de ellos heredan el nombre de las entidades en la capa de objetos, y aparecen otros nuevos como los referentes a las visualizaciones de datos en bruto o recursos virtuales como la descarga de los paquetes de simulación. En cambio, los que no aparecen representados son la estructura de los datos de los conjuntos de entrada, de salida y parámetros, que están integrados dentro de los recursos. Todos estos recursos están organizados siguiendo una jerarquía como la que está representada en la Figura 3.29 y en la que están también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar en el informe técnico de Servicio

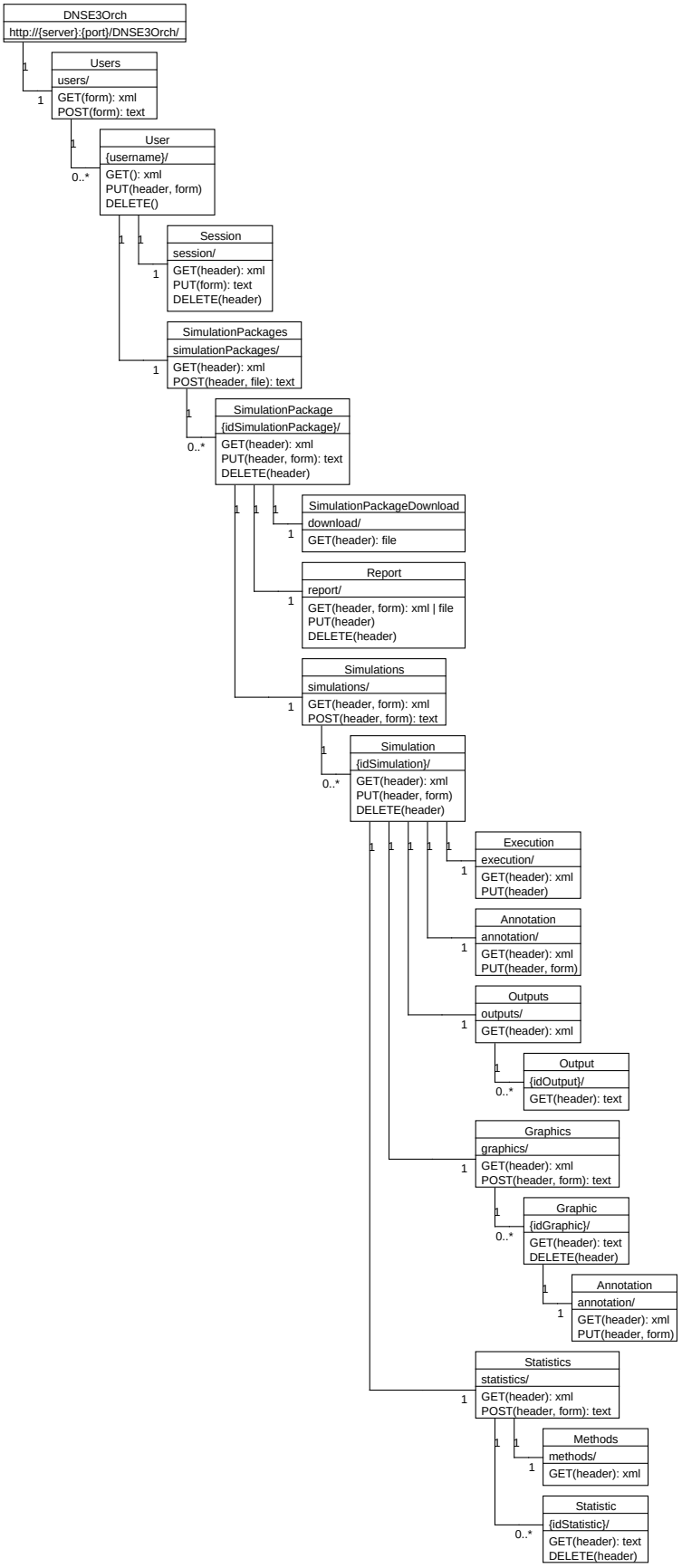

Figura 3.29: Diagrama de recursos del Servicio de Orquestación.

de Orquestación [40] que acompaña este Trabajo Fin de Máster.

3.4.8. Servicio de Simulación

El Servicio de Simulación (SGA-002) realizará la recogida automática de las simulaciones individuales que están pendientes de ejecución en un Servicio de Colas (SGA-001) de los que están registrados en un Servicio de Monitorización (SPI-002), para proceder a su ejecución utilizando el simulador de red *ns-3* y almacenar los resultados generados. Para ello, el servicio recoge los datos de la simulación del espacio asignado en el repositorio de ficheros del Servicio de Almacenamiento (SPI-001) y deposita los resultados de la simulación en el espacio asignado para ello en el mismo. A su vez, el servicio se registra en el Servicio de Monitorización (SPI-002), publica sus métricas definidas y, de forma puntual, va publicando los valores que van teniendo dichas métricas a lo largo de su ejecución.

Por tanto, este servicio tendrá una interacción directa y de forma activa con cinco elementos software de la aplicación DNSE3. Tres de ellos serán servicios de la aplicación (a los que también se les llamará *clientes*) y dos de ellos serán elementos externos al sistema, pero necesarios para su ejecución. Los cinco elementos son los siguientes:

- El *Servicio de Monitorización (SPI-002)* se utilizará para la recogida del listado de los Servicios de Colas (SGA-001) disponibles en la aplicación DNSE3, y para el registro del servicio en él, así como la publicación de sus métricas y sus valores correspondientes.

- El *Servicio de Colas (SGA-001)* se utilizará para la recogida de una simulación individual pendiente de procesar en la cola, y posteriormente, tras su ejecución, señalar en la misma la terminación de su procesado.

- El *Servicio de Almacenamiento (SPI-001)* se utilizará para recuperar la información de la simulación individual de su espacio de almacenamiento y, posteriormente tras su ejecución, depositar los resultados generados por la misma.

- El simulador de red *ns-3* se utilizará para llevar a cabo la ejecución de una simulación individual, proporcionándole los *scripts* necesarios, y para su posterior recogida de los resultados generados a través de la salida estándar.

- La *nube computacional*, y en concreto la instancia donde se está ejecutando, para obtener los valores actuales de las métricas que se han publicado en el Servicio de Monitorización (SPI-002) y que hacen referencia a la carga actual del servicio.

Las funciones principales que va a realizar este servicio y que se ofrecerán a través de su interfaz programática son las siguientes:

- Listar las simulaciones en ejecución y las métricas disponibles en el servicio.

- Consultar el estado de una simulación en ejecución y el valor actual de una métrica del servicio.

- Eliminar una simulación en ejecución.

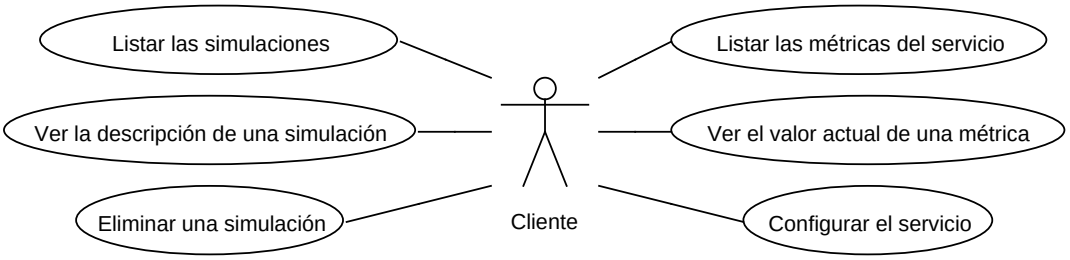

Figura 3.30: Diagrama de casos de uso de los escenarios de uso del Servicio de Simulación.

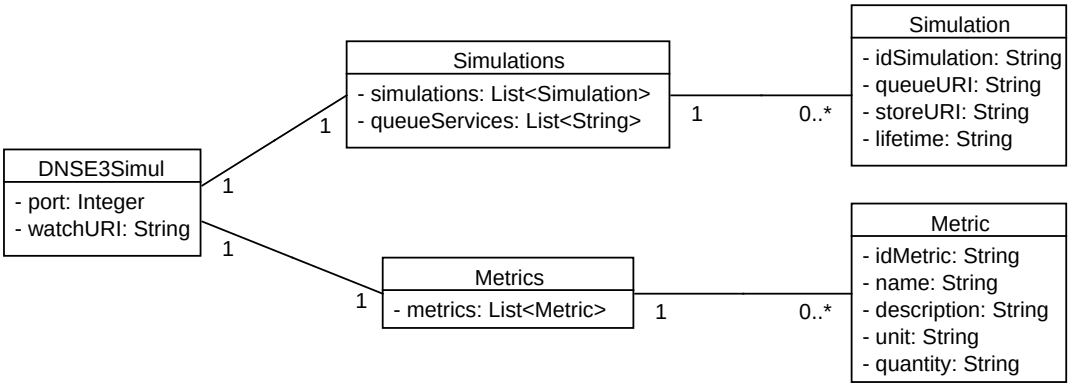

Figura 3.31: Diagrama de modelo de dominio del Servicio de Simulación.

Estas funciones están reflejadas en el diagrama de casos de uso de la Figura 3.30, en donde se relacionan con los elementos software de su entorno. Además, a través de la interfaz de comandos, se podrá configurar el servicio para definir su puerto de escucha y el Servicio de Monitorización (SPI-002) en el que se registrará y publicará sus métricas y los valores asociados a ellas.

Internamente, el servicio va a tener un modelo de dominio compuesto por cinco entidades: *DNSE3Simul* como entidad principal del servicio, *Simulations* como el conjunto de simulaciones en ejecución, *Simulation* como una simulación en ejecución en particular, *Metrics* como el conjunto de métricas publicadas del servicio y *Metric* como una métrica publicada y su valor actual en particular. Las propiedades y relaciones de cada entidad pueden verse en la Figura 3.31.

Las entidades anteriores, incluidas en la capa de objetos de la arquitectura lógica, serán la base para la capa de recursos. Por tanto, el servicio constará de cinco recursos con el mismo nombre que los objetos anteriores, que estarán organizados siguiendo una jerarquía como la que está representada en la Figura 3.32 y en la que están representados

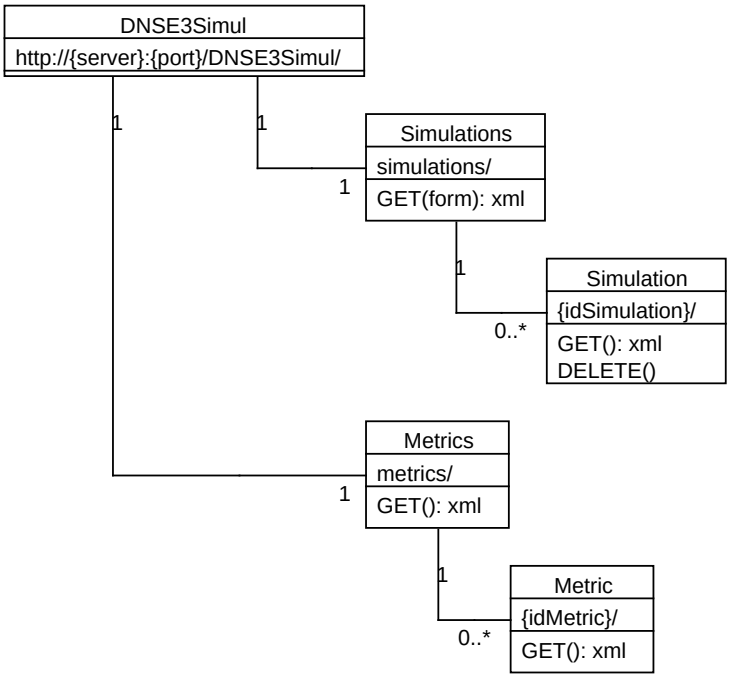

Figura 3.32: Diagrama de recursos del Servicio de Simulación.

también los métodos que tendrá cada uno de los recursos y el espacio de URIs del servicio.

Para obtener una descripción completa del servicio, junto a su elicitación de requisitos software, análisis y diseño detallados, se puede consultar en el informe técnico de Servicio de Simulación [41] que acompaña este Trabajo Fin de Máster.

3.5. Selección de tecnologías

La selección de las tecnologías a utilizar en un proyecto software es una parte importante, ya que la evolución de las siguientes fases de implementación y puesta en práctica dependen en su mayoría de una buena elección. Aunque en un proyecto de investigación esta selección no tiene tanta relevancia como en un proyecto industrial, también hay que realizar un estudio de las opciones disponibles para cada una de las tecnologías utilizadas en proyecto, analizando sus beneficios e inconvenientes, y justificando la elección más idónea.

En este proyecto de investigación hay tres tipos de tecnologías fundamentales: la implementación de los servicios seguirá el estilo arquitectural REST (y en particular su especificación para los servicios Web RESTful [11]), el despliegue de los servicios se hará en una infraestructura de nube computacional, y la simulación de las redes telemáticas se llevará a cabo a través de un simulador de red.

3.5.1. REST y RESTful

REST (*REpresentational State Transfer*) es un diseño arquitectural basado en recursos donde cada uno de ellos está referenciado con un identificador único. Estos recursos se comunican entre sí a través de un estándar, en el cual realizan el intercambio de representaciones con el contenido de la comunicación. RESTful [11] es una especificación de este diseño arquitectural que está basado en los estándares definidos para la Web. Por ello, para la identificación única de cada recurso utiliza una URI (*Unique Resource Identifier*) que representa a un único recurso, para la comunicación entre los recursos hace uso del protocolo HTTP (*HyperText Transfer Protocol*), y para el intercambio de representaciones utiliza estándares como el lenguaje XML (*eXtensible Markuk Languaje*), entre otros.

RESTful no es una implementación de REST, sino una especificación que define las pautas que deben seguirse para la implementación de un servicio basado en una arquitectura orientada a recursos (ROA). Actualmente en cualquier lenguaje de programación se permite la posibilidad de implementar la especificación RESTful desde cero, puesto que existen implementaciones para los estándares utilizados en la misma (esto es, URI, HTTP o XML). Por ejemplo, la plataforma Java de Oracle contiene las bibliotecas necesarias para la implementación de una aplicación que haga uso de estos estándares. Pero, debido a que esta implementación lleva un coste de tiempo adicional al del proyecto a desarrollar, así como la posibilidad de introducir vulnerabilidades o cometer fallos que desestabilicen la implementación final, se hace uso de *frameworks* ya desarrollados y probados. Estos *frameworks* proporcionan una API (*Application Program Interface*) que facilita la implementación, y aseguran una calidad lo suficientemente alta en aspectos como eficiencia y seguridad.

Por tanto, los *frameworks* más relevantes en el campo de la implementación de la especificación RESTful son los siguientes:

- **RESTlet Web API Framework for Java**[1]**.** Es un proyecto iniciado en diciembre de 2005 por Jérôme Louvel, y se presenta como una solución a la falta de implementación de la especificación RESTful en el lenguaje Java, solucionando las limitaciones ofrecidas por los *Servlets*. Utiliza Jetty[2] para separar la implementación del protocolo HTTP del soporte para *Servlets*. Proporciona la posibilidad de implementar tanto el lado del cliente como el lado del servidor, pudiendo incluir ambos en un mismo servicio. La API que ofrece consta de un núcleo con la funcionalidad básica y varias bibliotecas de apoyo que implementan el uso de los estándares utilizados (por ejemplo, conectores HTTP, formularios Web o documentos XML). Entre sus características está el soporte nativo de REST, soporte para varias ediciones de la plataforma Java (Java SE/EE, GWT, GAE, Android u OSGi), un completo servidor web (con posibilidad de usar estándares de seguridad), multitud de conectores de varios protocolos (HTTP, SMTP, POP, JDBC, FILE, etc.) y de representaciones (POJO, XML, XSLT, RSS, Atom, etc.), una configuración flexible, implementaciones de seguridad de otros servicios (como Google, Amazon u OAuth) y facilidad para la escalabilidad.

[1] `http://www.restlet.org/about/`
[2] `http://jetty.codehaus.org/jetty/`

- **ASP.NET Web API**[3]**.** Proyecto creado a partir del RESTful WCF API de la *Windows Community Foundation*, es una solución empresarial para dar soporte para REST a las tecnologías de programación de Microsoft, en este caso, para ASP.NET. El objetivo de este *framework* es construir y consumir servicios HTTP desde un amplio rango de dispositivos, pensada para construir plataformas siguiendo las especificaciones de RESTful. Las características que incorpora son las siguientes: un modelo de programación moderno para el protocolo HTTP, soporte completo para el encaminamiento en ASP.NET, negociación del contenido y formatos personalizables, modelo de unión y validación, filtros, creación de consultas (OData), facilidad para hacer pruebas unitarias, mejoras en *Inversion of Control* (IoC), configuración basada en el código, y anfitrión propio sin necesidad de usar IIS (el servidor Web de Microsoft).

- **JAX-RS - Java API for RESTful Web Services**[4]**.** Es una API para el lenguaje de programación Java que da soporte para la creación de servicios Web basados en el estilo arquitectural REST. Está incluido como una parte oficial de Java Enterprise Edition desde la versión 6. Las principales características que definen esta tecnología son: API del cliente a bajo nivel, filtros y manipuladores, soporte para validaciones, hipermedia, procesamiento de peticiones asíncronas en la parte del servidor, integración con el inyector de dependencias JSR 330, especificación del tipo del contenido de la respuesta en la parte del servidor, y API de configuración unificada.

Los tres *frameworks* dan soporte completo a las especificaciones básicas de RESTful, como son la creación de clientes y servidores que implementen el protocolo HTTP y sus métodos básicos de CRUD, el intercambio de representaciones entre los recursos en diferentes formatos estándares o la configuración de los servicios personalizada. Por otra parte, RESTlet y JAX-RS utilizan el lenguaje de programación Java, liberado bajo la licencia GNU GPL de software libre, mientras que ASP.NET Web API, como su nombre indica, utiliza el lenguaje ASP.NET que es propietario de Microsoft.

Por tanto, debido a la minimización de los costes en el desarrollo e implementación de esta propuesta, la mejor opción es el uso de un lenguaje de software libre, como es Java. De entre los dos *frameworks* relevantes en este lenguaje de programación, como ambos aportan las funcionalidades básicas de RESTful, se seleccionará aquel que integre funcionalidades adicionales que puedan servir tanto para ampliar el servicio en un futuro como para facilitar su implementación. Así, la tecnología seleccionada será **RESTlet Web API Framework for Java**, por el amplio abanico de funcionalidades adicionales, como por ejemplo el soporte ofrecido para varias plataformas (incluyendo plataformas móviles como Android) o las implementaciones de seguridad para la integración con otros servicios.

3.5.2. Infraestructura de nube computacional

La propuesta de aplicación, como ya se ha visto, está compuesta por ocho servicios. No todos los servicios requieren necesariamente su ejecución en una infraestructura de

[3] `http://www.asp.net/web-api`
[4] `http://java.net/projects/jax-rs-spec`

nube computacional, pero es recomendable que todos ellos se ejecuten en ella para aprovechar la ventaja de la escalabilidad de la aplicación ofrecida por la computación en nube, y que la misma pueda ser explotada junto con la facilidad que ofrece el diseño arquitectural de REST para realizar la escalabilidad de un servicio. En cambio, hay otros servicios que sí requieren el uso de una nube computacional, como son el Servicio de Simulación para la consulta de ciertos valores para conocer el estado actual de la instancia en la que se ejecuta, y el Servicio de Escalado para realizar la ejecución de instancias nuevas de un servicio o terminar alguna de las ya existentes. También el Servicio de Almacenamiento podrá utilizar la computación en nube para realizar un autoescalado del espacio de almacenamiento según sea necesario.

Entre las diferentes alternativas tecnológicas de infraestructura de computación en nube, sólo se analizarán aquellas más relevantes que estén bajo una licencia que permitan el libre uso sin la necesidad del pago por licencia o uso. Estas soluciones suelen estar diseñadas como nubes privadas, aunque suelen ser compatibles con las API desarrolladas por las nubes públicas (por ejemplo, Amazon EC2) que permiten convertirse en nubes híbridas sin un coste adicional en la infraestructura, más allá del pago por el uso de las nubes públicas. La justificación de buscar una solución de código abierto se debe a que las instituciones educativas, y sobre todo en proyectos de investigación, la estimación y justificación de este tipo de gastos puede resultar complicada, siendo más sencilla la compra de hardware para instalar una nube computacional privada con posibilidad de reutilización en otros proyectos. Además, las alternativas de código abierto suelen estar diseñadas para ejecutarse sobre sistemas operativos de la misma índole (por ejemplo, GNU/Linux Ubuntu Server o GNU/Linux Fedora), puesto que es necesaria una fuerte cohesión con los servicios de virtualización de computación, almacenamiento y red del sistema operativo.

En el estudio realizado por Laszewski et al. [42] identifican a cuatro infraestructuras de computación en nube de código abierto como las más relevantes: Nimbus[5], OpenNebula[6], Eucalyptus[7] y OpenStack[8]. Las cuatro propuestas contienen varios aspectos en común, como son la implementación de una interfaz REST, la compatibilidad con las API de Amazon EC2 y Amazon S3, el soporte para los dos *hypervisor* más frecuentes (KVM y XEN) para la virtualización de los recursos computacionales, varias opciones para la virtualización de la red, una solución completa e independiente para la virtualización del almacenamiento, y credenciales seguros de identificación usando el protocolo estándar X509.

Laszewiski et al. concluyen que ninguna de las soluciones de infraestructura de nube computacional de código abierto es "la mejor solución", sino que cada una tiene unas características que pueden adaptarse mejor o peor al uso al que van a estar destinadas. Además, sin la existencia de un *testbed* que permita comparar las características de las cuatro soluciones, no hay posibilidad de realizar una conclusión justa. Por otro lado, señalan que las dos soluciones más populares son Eucalyptus y OpenStack, y que las limitaciones desfavorables frente a Nimbus y OpenNebula (por ejemplo, la facilidad de su instalación) están próximas a resolverse en la siguiente versión, y el autor de este

[5] `http://www.nimbusproject.org/`
[6] `http://opennebula.org/`
[7] `http://www.eucalyptus.com/`
[8] `http://www.openstack.org/`

Trabajo Fin de Máster puede confirmar que están superadas con base en su experiencia personal con ambas infraestructuras. Adicionalmente, tras el estudio de las características básicas de las cuatro infraestructuras, Eucalyptus y OpenStack contienen más posibilidades implementadas frente a Nimbus y OpenNebula para la virtualización de los recursos computacionales (compatibilidad con más *hypervisor*) y de las redes (varios modos de uso), entre otras.

Por tanto, se ha decidido seleccionar dos de las tecnologías existentes de infraestructura de nube computacional de código abierto: **Eucalyptus y OpenStack**. Así, los ocho servicios de la aplicación DNSE3 deberán tener compatibilidad, al menos, con estas dos infraestructuras, pudiendo realizar la comunicación con ellas para la recuperación de valores para las métricas de un servicio o para realizar la escalabilidad horizontal hacia arriba o hacia abajo.

3.5.3. Simulador de red

La elección de la tecnología utilizada para la simulación de redes telemáticas está limitada, en parte, a la elección tomada por Bote et al. [5] para su propuesta de la aplicación DNSE, una propuesta sobre la que está basada la aplicación DNSE3 y de la que hereda la funcionalidad básica de ejecución de simulaciones individuales y de barrido de parámetros. El DNSE hace uso del simulador de red *ns-2*, una herramienta de simulación de código abierto utilizada para analizar un gran abanico de protocolos de red utilizados en redes cableadas e inalámbricas. El *ns-2* es la herramienta más utilizada en los laboratorios de las asignaturas de redes telemáticas por la amplia información que presentan los resultados de las simulaciones y que permiten a los estudiantes entender el comportamiento y las características de los protocolos de red.

En el estudio del estado del arte de los simuladores de redes telemáticas en educación realizado en en el Capítulo 2 están identificados varios simuladores de red alternativos al *ns-2*, entre los que se incluye la nueva versión de éste, el simulador de red *ns-3*, que resuelve las limitaciones que presentaba su antecesor. Además del *ns-2* y el *ns-3*, se describen otros simuladores de red de código abierto, como son: OMNeT++, JiST, J-Sim, Simpy y SSFNet. Todos ellos son totalmente válidos para ser usados como apoyo en la aplicación DNSE3 para la realización de las simulaciones de redes telemáticas, puesto que la principal característica que se necesita para la propuesta (la realización de simulaciones de red individuales que generen unos resultados de trazas que puedan ser estudiados posteriormente por los alumnos) está disponible en todos ellos.

Sin embargo, puesto que la propuesta del DNSE3 es una evolución del DNSE y esta aplicación ya está siendo utilizada en los laboratorios de las asignaturas de redes telemáticas, es conveniente seguir manteniendo el formato de definición de las redes telemáticas muy similar. De esta forma, la transición será menos costosa para los profesores que ya tienen definidas las propuestas de los paquetes de simulación. También, puesto que el DNSE3 mejora las limitaciones del DNSE referentes a los tiempos de ejecución y evaluación de resultados en un entorno pedagógico, es consistente mantener dicha evolución en la herramienta de simulación de red, de forma que el uso del *ns-3* frente al *ns-2* suponga también una solución a las limitaciones que presentaba dicha versión (un uso de memoria elevado ante simulaciones de redes de gran escala y la combinación de los lenguajes C++ y TCL para la descripción de la red telemática). A su vez, mantener la línea evolu-

tiva del simulador de red permitirá que en la evaluación del DNSE3 se puedan realizar comparaciones más objetivas con su antecesor DNSE.

Por tanto, el simulador de red que seleccionado para la aplicación DNSE será el ***ns-3***, frente a las otras alternativas de código libre ya estudiadas en el estado del arte.

3.6. Plan de implementación

Debido a la limitación de 375 horas dedicadas por el autor a este Trabajo Fin de Máster, la implementación de la propuesta de aplicación no ha podido llevarse a cabo. Con ello, tampoco podrá llevarse a cabo una evaluación real de la propuesta, como se verá más adelante, donde la aplicación DNSE3 sea probada por profesores y alumnos para recoger registros de su uso y opiniones de los usuarios. Sin embargo, en esta sección se propone un plan de implementación a llevar a cabo en un futuro, y de esa forma poder realizar la evaluación real de este trabajo de investigación.

La implementación de la aplicación DNSE3 es el siguiente paso a realizar tras el diseño en todo proyecto de ingeniería. Ya en el diseño, la aplicación está descrita con un nivel de detalle suficiente para que en la implementación sólo sea necesario afrontar los problemas técnicos con las tecnologías seleccionadas. Para el DNSE3 será necesario implementar cada uno de los ocho servicios de los que está compuesto. La tecnología seleccionada para implementar RESTful (RESTlet Web API Framework for Java) requiere la utilización del lenguaje de programación Java para su desarrollo. La dificultad no será por igual en cada uno de los servicios debido a los elementos software de terceros con los que interactúa en alguno de ellos. Por ello, se define una implementación iterativa, donde se irán completando y probando pequeñas partes de los servicios de forma individual y como aplicación.

Inicialmente, en la primera iteración, será necesario implementar el esqueleto básico de los servicios, de forma que se defina su jerarquía de recursos y su espacio de URIs. Se completará con la definición de los métodos para cada uno de los servicios, pero sin ningún tipo de funcionalidad más allá de la creación de los clientes en cada uno de ellos para las comunicación e interacción entre sí, y la definición del esquema de las representaciones aceptadas y generadas, más los códigos de repuesta y error. Así, hasta este punto se podrá probar que los servicios tienen sus interfaces programáticas bien definidas y que realizan la interacción entre sí correctamente. De momento no será necesario hacer uso la nube computacional para el despliegue de los servicios en las pruebas.

La siguiente iteración corresponderá a la implementación de la capa de objetos para que la anterior capa de recursos pueda ser implementada por completo. Por ello, se crearán las clases identificadas en el modelo de dominio, junto con sus propiedades y sus métodos, implementando también la funcionalidad básica (es decir, sin interactuar ni con la capa de datos ni con los elementos software de terceros) de cada uno de los métodos. Posteriormente se unirán ambas capas, de forma que se complete toda la funcionalidad de cada uno de los recursos, haciendo las invocaciones necesarias a los objetos de la capa inferior. Así, se podrá probar que cada uno de los servicios es capaz de gestionar la información de cada recurso y que las invocaciones de las acciones de cada uno de los recursos realizan las funciones que deben. En este iteración tampoco será necesario el despliegue de los servicios en una nube computacional para realizar las pruebas.

La tercera iteración se corresponderá a realizar la unión de los servicios con los elementos software de terceros. En este caso, los servicios implicados serán el Servicio de Simulación, el Servicio de Escalado y el Servicio de Almacenamiento. En el Servicio de Simulación se recogerán, a través de la interfaz programática de la instancia de la nube computacional y de llamadas por líneas de comandos, los valores reales y actuales para cada una de las métricas, ya sea si se le solicitan por la interfaz programática o si se están recogiendo para ser publicados en el Servicio de Monitorización. También se invocará en el simulador de red ns-3, por línea de comandos, la ejecución de una de las simulaciones individuales recogida del Servicio de Colas, esperando al término de la ejecución para recoger por la salida estándar las trazas con los resultados de la misma. El Servicio de Escalado utilizará la interfaz programática de la nube computacional para ejecutar nuevas instancias de un tipo determinado, o para terminar la ejecución de alguna de ellas, previa consulta de sus valores de las métricas para seleccionar aquella con más recursos libres. Finalmente, el Servicio de Almacenamiento, para esta propuesta, hará uso de un espacio físico de almacenamiento en la instancia de la nube computacional (como parte del volumen volátil de la instancia o como un volumen de almacenamiento asignado a la misma), por lo que la implementación se realizará utilizando las bibliotecas nativas de gestión de ficheros y directorios. Al finalizar esta iteración se probará que estos tres servicios son capaces de comunicarse con los elementos software de terceros y, en conjunto como aplicación, llevar a cabo todas las funcionalidades correctamente.

La aplicación DNSE3 ya será capaz, hasta esta iteración, de realizar todas sus funcionalidades, pero no tiene ningún tipo de persistencia de los datos. Por tanto, la cuarta iteración se utilizará para unir la capa de datos a la capa de objetos de la aplicación. Existirán dos tipos de persistencia de los datos: aquella llevada a cabo por el Servicio de Almacenamiento a través de los diretorios y ficheros que aloja, y la propia de cada servicio individual para gestionar sus objetos. En el Servicio de Almacenamiento se alojarán los datos relativos a los usuarios y todo su contenido en un espacio virtual que asignará, por ejemplo, creando un directorio raíz DNSE3 y añadiendo en él una estructura jerárquica con los identificadores de los usuarios, paquetes de simulación, informe, simulaciones y visualizaciones. Así, cuando se inicie el Servicio de Orquestación podrá recuperar todos los datos desde este servicio y continuar trabajando con ellos de forma persistente. En cambio, el resto de servicios (Colas, Escalado, Estadística, Informe, Monitorización y Simulación) tendrán una persistencia en sus propios servicios, almacenando el contenido de los objetos bajo una tecnología de persistencia para Java (por ejemplo, Hibernate). Las pruebas en esta iteración irán relacionadas a comprobar que la persistencia es estable, parando e iniciando los servicios sin que afecte al contenido de los recursos.

Finalmente, quedará por implementar la capa de interfaz Web, que se hará en la quinta iteración. Se desarrollará una interfaz Web de usuario que realice todas y cada una de las funciones de la aplicación DNSE3, y que se comunicará con la interfaz programática del Servicio de Orquestación para llevarlas a cabo. Las tecnologías utilizadas para la interfaz Web no son importantes para el desarrollo de la propuesta, pero se recomienda la utilización de los lenguajes de programación HTML y JavaScript para el lado del cliente, y el uso de PHP para el lado del servidor. Con esta iteración se probará que la interfaz Web de usuario realice todas las funciones de la aplicación DNSE3 correctamente, interactando con el Servicio de Orquestación para llevarlas a cabo de forma exitosa.

Por tanto, el plan de implementación constará de cinco iteraciones, cuya división se corresponde con cada una de las capas definidas por el modelo arquitectural REST (recursos, objetos, datos e interfaz Web), además de una iteración adicional para la unión con los elementos software de terceros. La estimación de trabajo para estas cinco iteraciones es de 10 semanas de duración, con una media de trabajo de 30 horas semanales. La división por iteraciones está representada en la Tabla 3.1.

Iteración	**Estimación temporal**
1. Capa de recursos	2 semanas
2. Capa de objetos	3 semanas
3. Elementos software de terceros	2 semanas
4. Capa de datos	1 semana
5. Capa de interfaz Web	2 semanas
TOTAL	10 semanas

Tabla 3.1: Estimación temporal de la implementación de la aplicación DNSE3.

Por otra parte, para las pruebas realizadas en la tercera, cuarta y quinta iteración, se propone un diagrama de componentes para el despliegue de los servicios y la interfaz Web de la aplicación DNSE3, junto a otros elementos software de terceros, que está representado en la Figura 3.33. Este despliegue está dimensionado para ser ejecutado en un escenario en el que sólo se da soporte a los laboratorios de una asignatura de una sola titulación (escenario habitual en una universidad). Constará inicialmente de cuatro instancias en la nube computacional que ejecutarán uno o varios de los componentes de la aplicación, y en el momento en el que la cola tenga pendiente alguna simulación individual, se ejecutará una quinta instancia (o más si son necesarias) para la ejecución de las simulaciones.

Así, las cinco posibles instancias estarán compuestas por los siguientes componentes:

- **Instancia de inicio.** Contendrá tanto la interfaz Web del usuario como el Servicio de Orquestación. La interfaz Web realizará llamadas REST al Servicio de Orquestación, y éste se comunicará con los servicios de las instancias de apoyo y de almacenamiento.

- **Instancia de almacenamiento.** Contendrá el Servicio de Almacenamiento y el sistema de ficheros físico donde estarán alojados los directorios y ficheros reales.

- **Instancia de apoyo.** Contendrá los servicios de apoyo a la aplicación DNSE3, como son el Servicio de Colas, el Servicio de Estadística y el Servicio de Informe. Estos servicios, a su vez, se comunicarán con la instancia de almacenamiento y con la instancia de gestión a través de REST. También tendrán la persistencia asociada a una base de datos propia en la instancia.

- **Instancia de gestión.** Contendrá los servicios de gestión de la nube computacional, como son el Servicio de Monitorización y el Servicio de Escalado, donde el segundo llamará a través de REST al primero y a la nube computacional. También tendrán la persistencia asociada a una base de datos propia en la instancia.

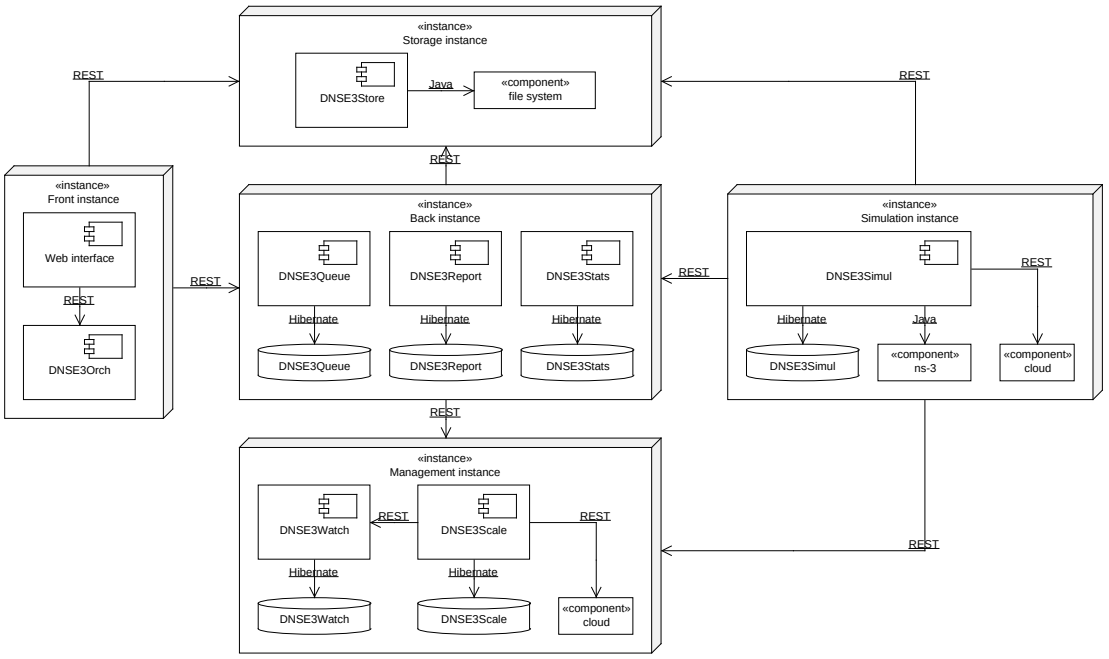

Figura 3.33: Diagrama de componentes para un ejemplo de despliegue de la aplicación DNSE3.

- **Instancia de simulación.** Contendrá el Servicio de Simulación y se comunicará a través de REST con las instancias de apoyo, de almacenamiento y de gestión. Internamente se comunicará por REST con la instancia de la nube computacional y por línea de comandos con el simulador de red *ns-3* que tendrá la instancia. También tendrá la persistencia asociada a una base de datos en la instancia.

Esta propuesta de despliegue de la aplicación DNSE3 es únicamente una propuesta, tal y como su nombre indica. Puede ser modificada o alterada en todos sus niveles tanto como sea necesario para adaptar la aplicación a las necesidades requeridas.

Capítulo 4

Discusión y Propuesta de Evaluación

4.1. Discusión

Los entornos de simulación presentan grandes ventajas frente a otras alternativas en entornos educativos, ya que proporcionan la capacidad de realizar pruebas prácticas como apoyo al aprendizaje teórico con un bajo coste y una alta capacidad de recursos de simulación. Sin embargo, tienen algunas limitaciones, ya que las soluciones existentes no están adaptadas para ser usadas por los alumnos, debido a la complejidad en el modelado de las simulaciones y a la dificultad en analizar los resultados generados.

En el caso de los simuladores de redes telemáticas, existen muchas alternativas que permiten la realización de simulaciones de diferentes topologías de redes y con una diversidad de protocolos. Sin embargo, estas herramientas presentan la complejidad de modelado y análisis de resultado de los simuladores anteriores, entre otras.

La solución presentada en este trabajo de investigación, el entorno de simulación de redes distribuido DNSE3, pretender resolver parte de las grandes limitaciones presentes en otras alternativas, aplicando las ventajas ofrecidas tanto por la nube computacional como por los servicios REST.

En el trabajo de Bote et al. [5], en el que proponen un entorno de simulación de redes distribuido (DNSE), ya se identificaron limitaciones presentes en soluciones anteriores, como por ejemplo el simulador de red ns-2, que resolvieron. La primera fue el interés que tiene la realización de simulaciones de barrido de parámetros para el apoyo al aprendizaje, puesto que permite la posibilidad de realizar un conjunto de simulaciones individuales señalando previamente el rango de valores que deben tomar los parámetros de la simulación. La segunda fue la posibilidad de realizar simulaciones de forma paralela y distribuida, empleando para ello el simulador de red ns-2 y el paradigma del grid computacional. Y la tercera fue proporcionar un entorno gráfico de usuario para el análisis de los resultados obtenidos tras la simulación, donde los alumnos de forma sencilla podían consultar los resultados, hallar datos estadísticos o visualizar tanto el comportamiento de la simulación como gráficas de diferentes parámetros.

Sin embargo, el DNSE presenta dos limitaciones relevantes que el DNSE3 pretende resolver. La primera es que, aunque el grid computacional permite la distribución de las simulaciones entre varias computadoras consiguiendo un ahorro temporal en su ejecución, no permite la escalabilidad de los recursos computacionales para adaptarse a la demanda

de simulaciones individuales presentes en cada momento. Por tanto, los recursos computacionales, aunque pueden aumentarse añadiendo más computadoras al grid computacional que ejecuten los servicios del DNSE, no permite la posibilidad de aumentar y reducir la capacidad computacional dinámicamente durante su ejecución.

La segunda limitación es la mejora de las características y funciones de la aplicación tras la experiencia de uso, aportadas por los profesores y alumnos de las asignaturas donde se ha utilizado el DNSE. Estos usuarios indicaron como mejorable las funciones presentadas por la interfaz gráfica de usuario, que aunque permitía realizar las tareas principales, se sugería que se pudiese tener una interfaz Web que permitiese la ubicuidad de la aplicación y una mejor experiencia de usuario, como por ejemplo, tomar anotaciones en las simulaciones o en las gráficas, o la realización de informes de un paquete de simulación.

Así, el DNSE3 pretende resolver ambas limitaciones con el uso de una nube computacional en sustitución del *grid* computacional, y esta elección ha apoyado también a la decisión del uso de servicios REST en sustitución de los servicios WSRF (*Web Services Resource Framework*). También se ha propuesto la implementación de una interfaz gráfica de usuario basada en la Web que incorpore las nuevas funcionalidades (ya vistas en la sección 3.2) y mejore la experiencia de usuario, proporcionando a su vez ubicuidad y ejecución multiplataforma.

Con respecto a otras alternativas de simuladores de redes ya vistas en la sección 2.1, de las seis soluciones presentes de software libre, sólo una de ellas incorpora una interfaz gráfica de usuario que permita el análisis de los resultados obtenidos de forma sencilla, como es OMNeT++. El resto de soluciones (ns-2, ns-3, J-Sim, SimPy y JiST/SWANS) no contienen esta característica, dificultando el análisis de los resultados y empobreciendo el apoyo al aprendizaje. Con respecto a la paralelización de las simulaciones (esto es, permitir la división de una simulación en varias partes y ejecutarlas de forma paralela para conseguir un mejor resultado temporal), sólo el simulador de red ns-2 no presenta esta característica, pero el resto de soluciones tampoco permiten realizar la escalabilidad de los recursos computacionales utilizados en dicha paralelización con el que permitir conseguir una mejora del rendimiento y una adaptación del entorno a las necesidades de la aplicación.

Las alternativas de simuladores de redes comerciales, como son OPNET Modeler, NetSim y Qualnet fueron descartadas desde el inicio por la implicación del pago de licencias, que en el caso de un entorno educativo supone un desembolso de una cantidad lo suficientemente grande, por la gran cantidad de alumnos que la van a utilizar, como para no ser viable para las instituciones educativas.

Con respecto al uso de la nube computacional en el DNSE3, la mayor ventaja ofrecida ha sido la posibilidad de ofrecer un escalado horizontal de los recursos disponibles para ejecutar las simulaciones. Esto permite que se puedan ejecutar varios Servicios de Simulación de la aplicación, donde cada uno de ellos está compuesto por una instancia que contiene el simulador de red ns-3, pudiendo ampliar o reducir el número de instancias según las simulaciones individuales pendientes de ejecución. Esta aproximación está basada en la segunda opción propuesta por Fujimoto et al. [8] para la paralelización de las simulaciones, pero adaptada a la escalabilidad.

Por tanto, se aprovechan tres tipos de ventajas, como son la paralelización ofrecida por el simulador de red ns-3, la distribución de las simulaciones entre varias instancias de

simuladores de red, y la escalabilidad horizontal de las instancias ofreciendo más o menos número de las mismas según la carga actual pendiente. Si la reducción de tiempo que se tuvo en la propuesta del DNSE de Bote et al. [5] fue considerable, en este caso se supone que la reducción de tiempo será mayor, ya que se ofrecen más recursos computacionales y más instancias del simulador de red para ejecutar todas las simulaciones.

La principal ventaja de utilizar una nube computacional con respecto a otras alternativas de computación distribuida como el *grid* o *cluster* computacional (descritos en la sección 2.2) es, precisamente, la escalabilidad dinámica de sus recursos computacionales y la visión "infinita" de recursos disponibles. Con esta aproximación, la reserva y liberación de los recursos computacionales se realiza automáticamente por la aplicación sin necesidad de intervención por parte del administrador o usuario, utilizando sólo aquellos que son necesarios, con el consiguiente ahorro económico y eficiencia energética que conlleva.

Relativo al uso de la nube computacional está el uso del modelo arquitectural REST, o comúnmente llamado como "servicios REST". En principio no sería necesario utilizar los servicios REST con el uso de la nube computacional, pero la ventaja obtenida por la conjugación de ambos es significativa. Una de las características de la nube computacional es que su interfaz programática está ofrecida, al menos, como servicios REST [22], por lo que si la aplicación está basada en servicios REST se supone una mayor interoperabilidad entre ésta y la nube computacional a la hora de realizar el escalado horizontal.

Una de las características que se suele aplicar con el uso del modelo arquitectural REST y que es muy recomendable a la hora de realizar el escalado horizontal, es la división del sistema en diferentes servicios, es decir, seguir una arquitectura orientada a servicios (SOA). De esta forma, cada parte del sistema (o servicio) cumplirá un conjunto de funciones comunes y concretas, que con el escalado se podrán replicar para conseguir realizar las tareas más rápidamente. Éste es el caso del Servicio de Simulación, en el que cuantas más instancias se ejecuten de él, más simulaciones individuales pendientes podrá ejecutar, y más rápido se podrán realizar las simulaciones de barrido de parámetros compuestas de un conjunto de simulaciones individuales.

La identificación de servicios según las funciones del DNSE3 ha generado ocho servicios, ya vistos en la sección 3.4. Además, se realizó una clasificación de los mismos según la dependencia con la infraestructura de nube computacional y con la aplicación. Es por ello que algunos de ellos como el Servicio de Almacenamiento, el Servicio de Monitorización y el Servicio de Escalado pueden ser reutilizados en otras aplicaciones, ya que no contienen ningún tipo de dependencia con la aplicación. Así, el almacenamiento contienen una interfaz programática genérica de gestión de directorios y ficheros, la monitorización es un caso similar, con la posibilidad de registrar cualquier tipo de servicio y publicar cualquier tipo de métrica, y el de escalado permite realizar un escalado horizontal sobre cualquier tipo de servicio, aunque en este último están presentes ciertas limitaciones con la gestión de métricas y escalabilidad que se tratarán más adelante.

Otros como el Servicio de Colas, el Servicio de Simulación o el Servicio de Estadística tienen una mayor dependencia con el tipo de aplicación, como es la simulación. Aunque la gestión de colas se ha definido lo más genérica posible, quizás que un elemento de la cola pueda estar en tres estados distintos (pendiente, en ejecución o finalizado) está ligado al tipo de aplicación, siendo los mismos estados en los que se encuentra una simulación. Este

servicio se podría reutilizar si el tipo de aplicación tiene estos tres estados o equivalentes. En el caso de las simulaciones de redes a través de la herramienta ns-3, el formato en que se recogen sus valores y depositan los resultados en el Servicio de Almacenamiento crea dependencia doble: en el formato de los valores recogidos y depositados, y la interfaz programática del Servicio de Almacenamiento. Y para el cálculo de estadísticas, este servicio define unos formatos sobre la entrada y salida de valores que deben cumplirse y que, en principio, están pensados para usarse con los resultados de las simulaciones del DNSE3. Podría reutilizarse para otras aplicaciones, pero los valores para el cálculo estadístico y recogida de resultados deberán ser equivalentes a los de una simulación.

El resto de servicios son totalmente dependientes de la aplicación que no podrán ser reutilizados en otras aplicaciones. Por ejemplo, el Servicio de Orquestación es muy dependiente del contexto, conteniendo todas las funcionalidades de la aplicación y requiriendo el uso del resto de servicios para poder llevarlas a cabo. Y el Servicio de Informe está muy acoplado con el formato de un paquete de simulación, de forma que sólo realizará informes de este tipo y siguiendo el formato ya determinado.

Finalmente, el uso de tecnologías Web para la interfaz gráfica de usuario ofrece varias ventajas, entre ellas la posibilidad de su ejecución como aplicación multiplataforma, ya que desde cualquier navegador Web se podrá acceder a la interfaz, o la ubicuidad del DNSE3 que, al poderse ofrecer a través de Internet desde la nube computacional, el acceso al mismo se podrá realizar desde cualquier punto del planeta con conexión a Internet.

Con respecto a la seguridad de la aplicación DNSE3, la nube computacional tiene ciertas limitaciones en la seguridad, sobre todo en las nubes públicas, por la compartición de los recursos hardware entre varias instancias y al estar expuestas desde Internet, como ya se ha visto en la sección 2.2. Pero también ofrece soluciones tradicionales como son las redes locales virtuales (VLAN) o la configuración de cortafuegos por cada instancia, con las que se puede restringir el acceso a las instancias. Así, en la aplicación se pueden ejecutar los servicios de tal forma que aquellos que no necesiten estar expuestos directamente a Internet (todos excepto la interfaz Web y el Servicio de Orquestación), estarán configurados tanto en una VLAN privada como un control de acceso a través del cortafuegos de tal forma que se protejan de un acceso no autorizado. Por otra parte, el DNSE3 contiene una seguridad interna a través de la interfaz programática de control de acceso de usuarios, de forma que un usuario tiene que estar identificado en la aplicación y haber obtenido el *token* de sesión para poder invocar a los métodos de cada recurso.

Aún así, los problemas presentes en la seguridad de la nube computacional siguen estando, ya que los recursos hardware utilizados por las instancias (núcleos de procesamiento, memoria de acceso aleatorio o infraestructura de red) son compartidos a través de la virtualización. Dicha virtualización es realizada a través de un hipervisor, y como cualquier tipo de elemento software de un sistema operativo, puede contener fallos de seguridad que permitan el acceso a otros recursos sobre los que un usuario no está autorizado.

Aunque la escalabilidad horizontal ofrecida por la nube computacional es una ventaja que favorece a uno de los objetivos del DNSE3 (en concreto, conseguir adaptar el tiempo de ejecución de las simulaciones al de una clase de prácticas de laboratorio), la inmediatez en el aumento de los recursos computacionales es un problema conocido [10, 21, 23]. Esta limitación se presenta debido al tiempo que transcurre entre que el sistema detecta que

necesita nuevos recursos computacionales y hasta que éstos están disponibles. En el caso del DNSE3, este tiempo es entre que las reglas de escalado detectan la falta de instancias de servicios de simulación y las ejecutan, hasta que estas instancias empiezan a consumir simulaciones pendientes desde el servicio de colas. Aunque ese periodo de tiempo suele ser pequeño (entre uno y dos minutos), la eficiencia conseguida con la escalabilidad se ve, en parte, limitada. Como solución se podría utilizar una heurística que pueda detectar la necesidad de escalar y anticiparse a ella, pero también se corre el riesgo de que no ocurra y se generen gastos adicionales que no eran necesarios.

Para realizar la escalabilidad, en el DNSE3 se utiliza un sistema basado en reglas y métricas, tal y como recomiendan autores como Voorsluys et al. [9] o Fujimoto et al. [8]. Este sistema permite que se recojan los valores de las métricas de diferentes servicios unificados en un Servicio de Monitorización y que, de forma regular, un Servicio de Escalado los consulte y compare con las reglas que tiene definidas, y en caso de ser válidas, ejecutar el criterio de escalabilidad asignado. Para el DNSE3 se ha descrito un formato simple de reglas: asignar un tipo comparativo que evalúe un valor de una métrica de los tipos de servicio registrados en un servicio de monitorización, con un valor fijo asignado, y en caso de cumplirse la comparación, realizar el escalado horizontal hacia arriba (ejecutando una nueva instancia) o hacia abajo (terminando la instancia con menos ocupación de sus recursos) de un tipo de servicio determinado. Puesto que este tipo de reglas contienen varias limitaciones, como por ejemplo comparar una única métrica por regla, realizar un tipo de comparación simple o escalar una única instancia de un tipo por cada regla que se cumpla, todas ellas son relativas al formato y lenguaje para su definición. Así, una limitación es la falta de un lenguaje y formato de definición de reglas para el escalado de un sistema en una nube computacional.

Asociado al lenguaje y formato de definición de reglas, está el problema presente a la hora de seleccionar el tipo de métricas más adecuado para realizar la escalabilidad de un sistema y así garantizar una calidad del servicio adecuada al contrato de nivel de servicio acordado. Existen multitud de métricas que pueden utilizarse, como por ejemplo el porcentaje de uso del procesador o la cantidad de memoria ocupada, pero según el conocimiento del autor, no existe un estudio que recomiende y discuta las ventajas e inconvenientes de utilizar una u otra en diferentes escenarios para garantizar la calidad y el contrato acordados.

Como punto final de la discusión en este trabajo de investigación, el autor quiere ofrecer unas pautas generales sobre las ventajas e inconvenientes de las *lecciones aprendidas* con respecto al desarrollo de aplicaciones para el apoyo al aprendizaje en la nube computacional. Estos aspectos son acordes con los expuestos por McDonald et al. [4] o Scatler [14] en sus informes técnicos sobre la aplicación de la nube computacional en la educación.

Cuando una institución educativa quiere dar el paso hacia el uso y desarrollo de aplicaciones en la nube computacional, el objetivo primordial es conseguir un ahorro económico con esa medida, siempre y cuando no se comprometan sus datos confidenciales. Para el caso de aplicaciones educativas de apoyo al aprendizaje, la seguridad en los datos no es relevante, por lo que hay una mayor predisposición al dar este salto para este tipo de aplicaciones.

Las ventajas de la escalabilidad de los recursos computacionales con el uso de nubes

públicas (menos utilizadas) o nubes híbridas (opción más común) permiten que el ahorro económico sea importante por el pago de sólo aquellos recursos que se utilizan, y por la externalización de parte del soporte y mantenimiento hardware y software. La posibilidad de que las aplicaciones de apoyo al aprendizaje tengan una capacidad "infinita" de ampliación a través del pago de pequeñas cantidades por su uso, permiten ofrecer un aprendizaje en el que las herramientas son capaces de adaptarse a las clases y su metodología, y no al contrario, como por ejemplo adaptar los grupos para una clase de 20 alumnos para el uso de un simulador de red del que se tiene cinco licencias y que están instalados en cinco ordenadores fijos en un aula.

Sin embargo, el riesgo latente del bloqueo de las aplicaciones y los datos bajo una plataforma de nube computacional es el freno de la migración a este paradigma. Una institución educativa seguramente no podrá permitirse el coste adicional que supone la adaptación de las aplicaciones a otra plataforma, que es necesario debido a la falta de estandarización entre ellas. La garantía de interoperabilidad entre las plataformas generaría mayor confianza, sin el miedo presente a una posible migración costosa.

4.2. Propuesta de evaluación

La propuesta de evaluación para este trabajo de investigación está basada en el análisis de características propuesto por Kitchenham et al. [43]. En él describen, entre otras cosas, cuatro aproximaciones para organizar una evaluación de análisis de características: modo de detección, caso de estudio, experimento formal, y entrevista. De las cuatro aproximaciones, se han seleccionado las dos primeras, ya que son las que mejor se ajustan a la propuesta realizada, pudiendo obtener resultados relevantes sobre las funcionalidades del DNSE3 y de su aplicación en un escenario real.

Por tanto, la evaluación se llevará a cabo en dos etapas: una inicial previa al desarrollo de la propuesta (el modo de detección) y otra tras desarrollar el DNSE3 y aplicarlo en un entorno real (el caso de estudio). Así, previamente al esfuerzo del desarrollo, se podrá evaluar por agentes externos y expertos que la aplicación está diseñada con la mayor aproximación para cumplir los objetivos propuestos, tanto técnicos como de investigación. Si la evaluación es negativa, se deberán revisar las funcionalidades de la propuesta para ser adaptadas a los objetivos. Una vez el diseño del DNSE3 pase la evluación de modo de detección, se pasará a su desarrollo y despliegue, siguiendo el escenario propuesto en el caso de estudio de la segunda parte de la evaluación. Así, se podrá comprobar definitivamente si los objetivos técnicos y de investigación se han cumplido.

El punto de inicio de la evaluación será el diseño de la aplicación DNSE3 para el modo de detección, y el despliegue de la implementación de la misma para el caso de estudio. Las funciones que soportará la aplicación están recogidas en la sección 3.2 de esta memoria, y con más detalle en el informe técnico de Especificación de Requisitos Software [30]. El DNSE3 consta de 25 requisitos funcionales que serán invocados por un único usuario. Este usuario será el alumno y/o profesor de una asignatura del área de las redes telemáticas, y conocerá las nociones previas necesarias sobre los conceptos teóricos de simulación de redes telemáticas. Los resultados de la evaluación serán útiles tanto para los investigadores como para los desarrolladores de software, para conocer si se han cumplido los objetivos de investigación y técnicos, respectivamente.

Para la evaluación, Kitchenham et al. definen cuatro roles y responsabilidades: el promotor de la investigación, en este caso el grupo GSIC/EMIC de la Universidad de Valladolid, y en particular los tutores de este trabajo de investigación como miembros de dicho grupo; el evaluador, que será el autor de este trabajo de investigación; los usuarios tecnológicos, que en la primera parte serán dos profesores de la asignatura de Teletráfico del Grado de Ingeniería Telemática de la Universidad de Valladolid, y en la segunda parte serán los alumnos de esa asignatura junto con los profesores; y los asesores de la herramienta, que serán los dos profesores anteriores.

4.2.1. Modo de detección

Esta aproximación de evaluación tiene como objetivo validar que los escenarios de uso de cada uno de los requisitos software de la aplicación DNSE3 cumplen, en mayor o menor medida, los objetivos técnicos y de investigación descritos en este trabajo. Esta evaluación se llevará a cabo una vez terminada la fase de diseño de la propuesta de aplicación. Por limitaciones de tiempo, a pesar de tener terminado el diseño, no se ha podido ejecutar la evaluación, pero sí la descripción de la propuesta.

La herramienta que se evaluará será el Entorno de Simulación de Red Distribuido III (DNSE3) que forma parte de la propuesta hecha por el autor del trabajo de investigación en el Capítulo 3. Esta propuesta contiene un listado de características que serán las que se evalúen, y que están descritas en la sección 3.2 del mismo capítulo y ampliadas en el informe técnico de Especificación de Requisitos Software [30]. Se evaluarán cada una de las 25 funcionalidades definidas para el DNSE3 en base a tres criterios: el uso potencial de esa funcionalidad, su aporte al apoyo al aprendizaje, y la dificultad de uso para los alumnos.

Por tanto, el documento de evaluación constará de tres partes, una para cada criterio, en el que se enumerarán cada una de las 25 funcionalidades. Para cada una de ellas se describirán los pasos que realiza, dando así al evaluador la suficiente información como para conocer qué se realiza en ellas. El nivel de detalle de la descripción serán los pasos más relevantes que ejecuta, incluyendo con qué servicios se comunica y los pasos realizados en los mismos. Para cuantificar el valor de cada criterio para cada una de las funcionalidades se utiliza una escala basada en seis opciones: muy en desacuerdo, en desacuerdo, poco de acuerdo, bastante de acuerdo, de acuerdo, y muy de acuerdo. Se toma una escala de seis valores (un número par) para que los resultados que se obtengan siempre tiendan hacia uno de los lados y se pueda conocer la tendencia (positiva o negativa) de las valoraciones.

Una vez que están los documentos de evaluación, se entregan a los participantes en el rol de usuarios tecnológicos. En este rol y para esta aproximación son los dos profesores que imparten o impartieron las prácticas de laboratorio de la asignatura de Teletráfico del Grado de Ingeniería Telemática de la Universidad de Valladolid. Estos dos profesores proporcionarán información suficientemente relevante como para conocer si la herramienta es útil en la programación de la parte práctica de su asignatura correspondiente al tema de simulación de redes telemáticas, sobre los siguientes aspectos:

1. La aplicación es capaz de proporcionar la suficiente funcionalidad como para poder cumplir los objetivos pedagógicos definidos para el temario de la parte práctica relativo a la simulación de redes telemáticas de la asignatura de Teletráfico. En con-

secuencia, también proporcionará la suficiente funcionalidad para otras asignaturas que contengan prácticas de simulaciones de redes telemáticas en su temario.

2. La aplicación es capaz de proporcionar un apoyo suficiente al aprendizaje a través de su funcionalidad, de forma que puede mejorar el aprendizaje en los alumnos de la asignatura, y en consecuencia, mejorar la adquisición de los objetivos pedagógicos definidos.

3. La aplicación es capaz de proporcionar una funcionalidad con un nivel conceptual suficiente como para ser entendido por los alumnos de la asignatura de Teletráfico en un entorno universitario, y en consecuencia, en otros entornos universitarios cuyo nivel conceptual se asemeje al de esta asignatura.

Esta evaluación no está destinada directamente a validar la hipótesis del objetivo de investigación de este trabajo de investigación, sino que sirve como evaluación intermedia que permita conocer si el trabajo que se está desarrollando sigue un buen camino. Así, si la herramienta propuesta es capaz de satisfacer las necesidades de los profesores de las asignaturas a las que va destinada, se podrá asegurar que el contexto en el que está enmarcada dicha propuesta es el indicado y que, una vez desarrollada, se podrán validar los objetivos que tiene que cumplir con respecto al rendimiento y la usabilidad. Y es que la funcionalidad que se evalúa en esta aproximación será la que posteriormente se traslade, en cierta medida, para conseguir el rendimiento apropiado y la usabilidad de la interfaz de usuario.

4.2.2. Caso de estudio

Esta aproximación de evaluación se llevará a cabo una vez se haya desarrollado la aplicación DNSE3 y se haya desplegado en las instancias de una nube computacional, por ejemplo, siguiendo el ejemplo de despliegue detallado en la sección 3.6. El objetivo de esta evaluación es poder validar los objetivos de este proyecto de investigación (definidos en la sección 1.2). Para ello, se propone un escenario de uso real en el que estará desplegada la aplicación para ejecutar sus funcionalidades con usuarios reales y en un entorno educativo real.

Se continuará con la evaluación de la herramienta DNSE3 y estarán involucrados en ella el evaluador (el autor de este trabajo), los usuarios tecnológicos (un profesor y 45 alumnos) y los asesores de la herramienta (el mismo profesor y los desarrolladores del software). El DNSE3 se empleará en las dos sesiones de prácticas, de dos horas de duración cada una, de la asignatura de Teletráfico del tercer curso del Grado de Ingeniería Telemática de la Universidad de Valladolid. Los participantes de esta asignatura son: un profesor, responsable de realizar el supuesto práctico y de diseñar la metodología de aprendizaje que se llevará a cabo, y 45 alumnos que deberán realizar el supuesto práctico empleando la herramienta DNSE3 propuesta una parte de ellos, y la herramienta DNSE de Bote et al. [5] otra parte de ellos. Estos 45 alumnos estarán repartidos por grupos, formando 15 grupos de tres alumnos cada uno, y asignándose 10 grupos al uso de la herramienta DNSE3 y 5 grupos al uso del DNSE. Los grupos de alumnos podrán ejecutar ambas herramientas tanto en sus ordenadores personales como en los proporcionados en el aula de prácticas.

El objetivo del supuesto práctico de simulación que propondrá el profesor será optimizar los parámetros de un diseño de una red telemática. Para ello, se requiere el uso de simulaciones de barrido de parámetros, ya que para buscar el valor óptimo de los parámetros, será necesario ejecutar varias simulaciones individuales con una configuración distinta de sus parámetros que permita comparar los resultados y obtener el conjunto más próximo al óptimo. Con este supuesto práctico se podrán evaluar los dos objetivos de este trabajo de investigación: la mejora de los tiempos de ejecución de las simulaciones de barrido de parámetros a través del uso de la escalabilidad dinámica ofrecido por la nube computacional y apoyado por los servicios REST, y la facilidad del análisis de los resultados a través de una interfaz gráfica de usuario basada en tecnologías Web.

El desarrollo del ejercicio práctico se desarrollará tal y como se explica a continuación:

1. El profesor repartirá a cada grupo de prácticas una copia del paquete de simulación con el que trabajarán. Este paquete de simulación contendrá la descripción del modelo de la red telemática a simular, pero no tendrá creada ninguna simulación y, por tanto, no habrá ninguna asignación de parámetros.

2. Cada uno de los grupos abrirá la aplicación correspondiente (DNSE3 o DNSE) y cargará el paquete de simulación en ella. A partir de ese momento, tendrán que ir creando simulaciones individuales y de barrido de parámetros, ir ejecutándolas y analizando los resultados obtenidos, para hallar los valores óptimos para los parámetros de esa topología de red.

3. Con respecto a la aplicación DNSE3, un grupo abrirá el navegador Web y solicitará la ruta donde esté alojada. Posteriormente se registrará en la aplicación y se identificará para entrar a su espacio de trabajo. En él subirá el paquete de simulación que le ha pasado el profesor y lo abrirá. A partir de este momento creará tantas simulaciones como crea necesarias para hallar los valores óptimos de los parámetros. Este ejemplo estará centrado en la creación de una simulación de barrido de parámetros. Por tanto, se creará asignándole un nombre y una descripción, y se fijará un conjunto de valores a cada uno de los datos específicos solicitados: el rango de valores de cada parámetro, el número de repeticiones y el tipo de diseño. Esta simulación se almacenará en el repositorio de ficheros, en el espacio asignado para este paquete de simulación.

4. Posteriormente, el grupo solicitará la ejecución de la simulación que ha creado. El DNSE3 recogerá los detalles de la simulación y realizará la división de la misma en tantas simulaciones individuales como sean necesarias. Por ejemplo, se puede suponer una división en 100 simulaciones individuales por cada simulación de barrido de parámetros. Por tanto, si los 10 grupos de prácticas que utilizan el DNSE3 realizan la misma simulación de barrido de parámetros, se tendrán como máximo 1.000 simulaciones individuales pendientes de ejecutar, que estarán en la cola de la aplicación.

5. Previamente a la sesión práctica, el Servicio de Escalado del DNSE3 contendrá ya definidas unas reglas de escalado. En ellas se ha fijado que cuando el Servicio de Colas tenga más de 100 simulaciones individuales pendientes de procesar, será necesario ampliar la capacidad de procesamiento ejecutando una nueva instancia del

Servicio de Simulación. La regla que define la escalabilidad horizontal hacia abajo versa que cuando un Servicio de Simulación tenga un porcentaje de ocupación de su procesador menor del 5 %, este servicio sea terminado. Por tanto, en el momento en que se pasa a tener 1.000 simulaciones pendientes, el Servicio de Escalado irá ejecutando una a una las 10 instancias necesarias del Servicio de Simulación. Estas instancias consultarán la cola e irán ejecutando las simulaciones pendientes, y según vayan quedando desocupadas las instancias, el Servicio de Escalado irá terminando su ejecución.

6. Una vez se han ejecutado todas las simulaciones, los grupos consultarán los resultados obtenidos. En las recomendaciones del profesor se les informa que primero analicen los resultados en bruto obtenidos, y que posteriormente obtengan los datos estadísticos de aquellas variables que ellos crean más relevantes y que generen las gráficas correspondientes. Durante la realización de estas tareas, el DNSE3 va realizando consultas a los servicios asociados para obtener las peticiones.

7. Por otra parte, algunos grupos comprueban que los resultados obtenidos para un conjunto de parámetros, en una de las gráficas, no es acorde a lo esperado, por lo que realizan la anotación en dicha gráfica y ejecutan la simulación individual para ese conjunto. El DNSE3 entonces introduce esa simulación a la cola, que es ejecutada en un breve periodo de tiempo. Al estar el sistema ejecutando simulaciones individuales constantemente, siempre hay disponible alguna instancia del Servicio de Simulación con una ocupación de su procesador superior a 5 % y, por tanto, el rendimiento es mayor que cuando se iniciaron las 1.000 simulaciones individuales de forma repentina, donde el sistema tardó en responder durante la ejecución de las instancias.

8. Una vez que los grupos han terminado de ejecutar todas las simulaciones y han analizado los resultados obtenidos, se les recomienda generar un informe del paquete de simulación en formato PDF que contenga todos los datos del mismo. Este informe deberán adjuntarlo posteriormente al informe principal del supuesto práctico, para ser revisado por el profesor. También se les recomienda descargar una copia del paquete de simulación, como medida preventiva.

9. Con respecto a los grupos que les fue asignado el uso del DNSE, las tareas que realizan son las mismas que los grupos que tienen asignado el DNSE3, y el proceso que se lleva a cabo es el que ya explicaron Bote et al. en [5].

10. Si alguno de los grupos no ha podido terminar el supuesto práctico en las sesiones prácticas, se les ofrece la posibilidad de terminarlo en las horas no presenciales. También se les indica que, si quieren, posteriormente se les realizará un cuestionario con preguntas sobre el uso de la aplicación fuera del aula de prácticas, con fines investigadores. Este cuestionario tendrá preguntas sobre la experiencia de uso y el rendimiento ofrecido fuera del aula en comparación con el que tuvieron en el aula durante las sesiones prácticas.

Durante el desarrollo de este supuesto práctico, tanto el evaluador como los desarrolladores del software estuvieron controlando y almacenando los registros de ejecución del

DNSE y del DNSE3. En ellos se almacenaban datos como el uso del procesador y de la memoria durante la ejecución de las simulaciones, o el número de instancias presentes en el sistema DNSE3 con respecto a la ocupación de la cola y el porcentaje del procesador. A su vez, se analizaba el comportamiento de los usuarios frente a la interfaz gráfica de usuario, recogiendo valores como la facilidad de uso (¿encontraban las opciones rápidamente? ¿necesitaban consultar al profesor el significado de alguna de las opciones?) o si el comportamiento de la interfaz era el esperado.

Con todos los valores recogidos durante el supuesto práctico, tanto el evaluador y los desarrolladores realizan una clasificación con respecto a cada una de las funcionalidades de las que constan dichas aplicaciones (realizando un mapeado con las del DNSE) y para cada uno de las dos herramientas, y les asigna una puntuación de seis valores: nada útil, poco útil, medianamente útil, bastante útil, útil y muy útil. Posteriormente se comparan ambas y se identifican si el DNSE3 mejora el aprendizaje con respecto al DNSE.

También se aprovechan los valores recogidos para realizar la comparación de la ejecución entre el DNSE3 y el DNSE, verificando si el primero mejora el tiempo de ejecución de las simulaciones de barrido de parámetros y si su interfaz gráfica de usuario ha ayudado a los alumnos a analizar mejor los resultados obtenidos. Por tanto, si los valores obtenidos son los esperados, se podrán validar los objetivos del trabajo de investigación, y la propuesta de aplicación desarrollada hace posible llevarlos a cabo.

Capítulo 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

El uso de simulaciones es una técnica importante en el sector educativo, permitiendo que los alumnos puedan conocer el comportamiento de un sistema sin necesitar los recursos físicos para realizar el estudio. Los simuladores se utilizan en muchos campos en educación, por ejemplo en la ingeniería industrial para las piezas de una máquina o en la ingeniería telemática para las redes de comunicaciones.

En el caso de la ingeniería telemática, el avance logrado por la utilización de redes telemáticas en el sector educativo es significativo. Permite el modelado grandes redes telemáticas y estudiar el comportamiento de los protocolos, nodos y otros elementos que la forman sin tener que disponer del hardware necesario, en muchas ocasiones inviable por su elevado coste económico. También permite estudiar el comportamiento de nuevos protocolos, mecanismos, aplicaciones o sistemas antes de ser aplicados en un contexto real.

La utilización de simuladores de redes telemáticas para el apoyo al aprendizaje permite que los alumnos puedan estudiar el comportamiento de una topología de red asignándole diferentes valores a sus parámetros de configuración, pudiendo comparar qué conjunto de valores es más adecuado para llevar la red al punto de funcionamiento deseado. Para ello, los alumnos necesitan ejecutar una simulación de barrido de parámetros en una topología de red, analizando posteriormente los resultados obtenidos para cada una de las opciones paramétricas posibles.

Sin embargo, los simuladores de red actuales, ya vistos en la sección 2.1, no ofrecen la opción de realizar simulaciones de barrido de parámetros, debiendo el alumno configurar y ejecutar cada simulación individual por separado y de forma secuencial. Una característica que sí ofrecen la mayoría de los simuladores de red estudiados es la paralelización de la ejecución de las simulaciones, pero esta técnica no es compatible con la posibilidad de tener un rango de valores para sus parámetros.

La solución para la ejecución de simulaciones de barrido de parámetros de forma concurrente es la utilización de la computación distribuida. De esta forma se pueden tener varias instancias de un simulador de red y ejecutar en cada una de ellas la correspondiente simulación individual. La eficiencia temporal conseguida con la distribución de las simulaciones es significativa.

Bote et al. [5] propusieron el DNSE, una aplicación que resolvía esta limitación usando un *grid* computacional para la distribución de las simulaciones en varias instancias del simulador de red ns-2. El tiempo de respuesta alcanzado permitía adaptar la ejecución de las simulaciones de barrido de parámetros dentro del tiempo disponible en una sesión de laboratorio. Sin embargo, esta solución no permite la escalabilidad dinámica y automática de los recursos computacionales para disponer de más instancias del simulador de red en caso de necesidad, sino que precisa de la intervención de un administrador que detecte dicha necesidad para añadir más nodos al *grid* computacional.

La solución propuesta y utilizada para resolver esta limitación es el uso de la computación en nube. Permite realizar una escalabilidad dinámica de sus recursos potencialmente "infinitos", ampliándola o disminuyéndola para adaptarse a las necesidades en cada momento, y automatizable por la propia aplicación ejecutada en la nube. Así, podrá permitir aumentar o disminuir el número de instancias del simulador de red en el entorno de simulación de red distribuido automáticamente a través de la definición de unas reglas de escalado que controlan una serie de métricas publicadas puntualmente por los servicios a escalar sobre su estado, sin necesidad de la intervención de un administrador. Por tanto, cuando se presente una carga muy alta de simulaciones individuales de varias simulaciones de barrido de parámetros, el sistema podrá autoescalar sus recursos para ejecutarlas en un tiempo razonable.

Por ello es importante que, tanto la infraestructura de nube computacional como los servicios que se ejecutan en ella, contengan una interfaz programática pública y compartida que permita la invocación de métodos de control, como por ejemplo para la ejecución de una nueva instancia o para la consulta del valor publicado para una métrica de un servicio.

Por otra parte, el uso de la nube computacional en un entorno educativo ofrece además otras ventajas a las instituciones educativas, como es un ahorro económico por la externalización de la gestión y mantenimiento del hardware, pagando sólo por el uso de los recursos consumidos, así como una alta disponibilidad de sus servicios. Sin embargo, la migración de los servicios en general de una institución educativa a la nube es lenta, debido a los problemas relativos a la seguridad de los datos confidenciales y el bloqueo de los datos en una infraestructura debido a la no interoperabilidad entre sí por la falta de estandarización.

La propuesta de este trabajo de investigación es el diseño de una nueva versión del DNSE que se ha llamado Entorno de Simulación de Redes Distribuido III (o sus siglas en inglés, DNSE3). El DNSE3 utiliza las ventajas de la nube computacional para ofrecer simulaciones de parámetros de forma distribuida y permitiendo además la escalabilidad horizontal de los recursos. Integra el simulador de red ns-3 (evolución del ns-2) para la ejecución de las simulaciones individuales. La aplicación está compuesta por ocho servicios con una elevada independencia funcional y una interfaz gráfica de usuario basada en tecnologías Web. La descomposición en servicios permite realizar la escalabilidad horizontal hacia arriba o hacia abajo por la ejecución de más o menos instancias de los mismos. La independencia funcional de los servicios también permite que puedan ser reutilizados en otras aplicaciones.

Cada uno de los servicios del DNSE3 está diseñado siguiendo el modelo arquitectural REST, basado en una arquitectura orientada a recursos. Así, cada servicio está compuesto

de varios recursos (usuarios, simulaciones, gráficas, etc.) que son identificados de forma unívoca. Estos servicios pueden comunicarse entre sí usando estándares e intercambiando representaciones.

El uso de REST, y en particular RESTful, permite que los servicios sean interoperables entre sí a través de la definición de una interfaz programática común conocida públicamente, que es invocable a través de los métodos sencillos pero pontentes definidos en el estándar HTTP (por ejemplo, POST permite la creación de nuevos recursos), y el intercambio de representaciones en formatos estándares como los formularios Web o documentos XML.

La interfaz gráfica de usuario proporciona la gestión de las funcionalidades de la aplicación y permite un análisis de los resultados de las simulaciones adaptada para el apoyo al aprendizaje y siguiendo un diseño visual con interés pedagógico. El uso de tecnologías Web permite que la aplicación esté disponible de forma ubicua y pueda ejecutarse en cualquier navegador Web. El resto de tecnologías seleccionadas para su uso en el DNSE3 serán Eucalyptus y OpenStack para la infraestructura de nube computacional, RESTlet para la implementación de los servicios REST, y ns-3 con respecto al simulador de red. En el caso de Eucalyptus y OpenStack, ambos ofrecen una interfaz programática de servicios REST que permiten la consulta del estado de la propia infraestructura y la gestión de sus recursos, por ejemplo, para ejecutar nuevas instancias de un servicio a través del servicio de escalado.

Debido a la falta de tiempo para la ejecución completa de este trabajo de investigación, la implementación de la aplicación propuesta y la evaluación de la misma no ha podido llevarse a cabo. Sin embargo, el autor ha propuesto un plan de implementación y una evaluación. En el plan de implementación del DNSE3 se explican los pasos a seguir a través de cinco iteraciones y se propone un modelo de despliegue sencillo.

A su vez, se propone una evaluación basada en el análisis de características para validar los objetivos tecnológicos y de investigación inherentes en el trabajo de investigación. Para la evaluación se siguen dos aproximaciones: el modo de detección y el caso de estudio. La aproximación de modo de detección realiza una evaluación de los requisitos software del DNSE3, puntuando cada uno de ellos, en una escala de seis puntos, según tres criterios: el uso potencial del requisito, su aporte en el apoyo al aprendizaje y la dificultad de uso para los alumnos. Con esta aproximación se pretende validar que los requisitos software que se han definido cumplen los objetivos técnicos marcados para el DNSE3 y, por tanto, ayudan a la consecución del objetivo principal de la investigación.

Con la segunda aproximación de caso de estudio se realizará una evaluación en la que se comparan las características del DNSE3 con las heredadas del DNSE, buscando la validación del objetivo global de esta investigación. Por ello se pretende validar si el DNSE3 permite la realización de simulaciones de barrido de parámetros en un contexto educativo, de forma que permita que la ejecución de estas simulaciones se adapte al tiempo de las clases de laboratorio y que el análisis de los resultados sea interpretable de forma sencilla por los alumnos de la asignatura para poder conseguir los objetivos pedagógicos.

El caso de estudio que se propone divide a una clase del área de redes telemáticas en 15 grupos, donde 10 de ellos usarán el DNSE3 y el resto el DNSE para realizar un mismo supuesto práctico. En este supuesto práctico tendrán que realizar, al menos, una simulación de barrido de parámetros y un análisis de los resultados obtenidos. Durante el

caso de estudio se recogerán datos relativos al funcionamiento de las aplicaciones, de la nube computacional y de la interacción de los usuarios con la interfaz Web. Estos datos serán clasificados según las características de la aplicación y puntuados en una escala de seis valores por los desarrolladores y los evaluadores. Los resultados arrojados por las puntuaciones se estudiarán y entonces se podrá validar la pregunta de investigación del trabajo.

5.2. Trabajos futuros

El primer trabajo futuro será completar las tareas que han quedado pendientes en este trabajo de investigación y que ya han sido comentadas en las conclusiones. La primera es la puesta en práctica de la aproximación de modo de detección propuesta para la evaluación de análisis de características. Se deberán componer las encuestas con los criterios propuestos, para ser rellenadas por los dos usuarios tecnológicos y así recoger y analizar los resultados por el evaluador. Los resultados obtenidos podrán validar los objetivos técnicos de la propuesta del DNSE3.

El siguiente trabajo futuro es la implementación del DNSE3 para poder llevar a cabo la aproximación de caso de estudio propuesta de la evaluación de análisis de características. Se desplegará el DNSE3 en el modelo propuesto en el plan de implementación y se ejecutará el supuesto práctico en las clases de laboratorio, recogiendo todos los datos necesarios para la evaluación. Posteriormente, entre los desarrolladores y los evaluadores realizarán la clasificación, puntuación y análisis de los resultados obtenidos. Con ellos se podrá validar la pregunta de investigación: ¿permite el DNSE3 la realización de simulaciones de barrido de parámetros en un contexto educativo?

Los siguientes trabajos futuros fueron detectados durante la discusión del capítulo 4. El primero surge por la limitación de la nube computacional en cuanto al retardo presente en la escalabilidad horizontal hacia arriba (es decir, la ejecución de nuevas instancias de un servicio). Aunque no se ha podido comprobar si este retardo se presenta en la realidad, se ha tenido en cuenta en el supuesto práctico de la segunda parte de la evaluación de la propuesta, ya que ha sido identificado por varios autores durante el estudio del estado del arte en el capítulo 2. Una propuesta para la solución de este problema es el uso de heurísticas fiables que detecten con anterioridad la necesidad de escalado hacia arriba y con una tasa de fallos muy baja. Otra posible solución podría ser la investigación en un tipo de tecnología que permita la ejecución de instancias en espacios de tiempo más breves, a través de replicación, uso de cachés o suspensión de instancias.

Otro trabajo futuro es la mejora en la definición e implementación del formato para la construcción de reglas de escalado. Uno de los pilares de la nube computacional es la posibilidad de realizar el escalado de sus recursos, y por eso necesita una definición de un formato de reglas que sea simple, completo y potente, que permita más libertad de acción y expresión. Por ejemplo, el formato debería permitir la definición de reglas que involucrasen varias métricas de varios tipos de servicios, que incluyesen acciones diferenciadas dependiendo del estado en el que se encuentren, o que permitiese una personalización de las tareas a realizar más allá de la ejecución o terminación de una nueva instancia.

Finalmente, se detecta como potencial trabajo futuro el estudio de métodos más avanzados en la infraestructura de nube computacional que faciliten su autoescalabilidad. Ac-

tualmente, el método más utilizado es la preconfiguración de imágenes de máquinas virtuales con los servicios a ofrecer, de forma que la escalabilidad horizontal hacia arriba consista en la ejecución de más instancias de este tipo de servicio. Para ello, otro servicio será el encargado de controlar los valores de las métricas publicadas por los servicios a través de software, y aplicar las reglas de escalado para detectar la necesidad de ampliar o disminuir sus recursos. Este método podría ser mejorado si la infraestructura de nube computacional integrase algún servicio avanzado de gestión de la escalabilidad de los servicios que puede ofrecer.

Capítulo 6

Referencias

[1] S. Siraj, A. K. Gupta, and R. Badgujar, "Network Simulation Tools Survey," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE 2012)*, vol. 1, pp. 201–210, junio 2012.

[2] H. Rajaei and J. Wappelhorst, "Clouds & grids: a network and simulation perspective," in *Proceedings of the 14th Communications and Networking Symposium (CNS'11)*, (San Diego, CA, USA), pp. 143–150, Society for Computer Simulation International, 2011.

[3] E. Weingartner, H. vom Lehn, and K. Wehrle, "A Performance Comparison of Recent Network Simulators," in *IEEE International Conference on Communications (ICC'09)*, pp. 1–5, junio 2009.

[4] D. McDonald, A. MacDonald, and C. Breslin, "Final report from the JISC Review of the Environmental and Organisational Implications of Cloud Computing in Higher and Further Education," tech. rep., University of Strathclyde, mayo 2010.

[5] M. L. Bote-Lorenzo, J. I. Asensio-Pérez, E. Gómez-Sánchez, G. Vega-Gorgojo, and C. Alario-Hoyos, "A Grid Service-based Distributed Network Simulation Environment for Computer Networks Education," *Computer Applications in Engineering Education*, in press.

[6] A. Juan, J. Faulin, J. Marqués, and P. Fonseca, "Castelldefels project: modeling and simulation of the computer system that gives support to the virtual campus of the Open University Of Catalonia," in *2007 Winter Simulation Conference*, pp. 2364–2364, IEEE, 2007.

[7] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, *et al.*, "Advances in network simulation," *Computer*, vol. 33, no. 5, pp. 59–67, 2000.

[8] R. M. Fujimoto, K. Perumalla, A. Park, H. Wu, M. H. Ammar, and G. F. Riley, "Large-Scale Network Simulation: How Big? How Fast?," in *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS 2003)*, pp. 116–123, IEEE, 2003.

[9] W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to cloud computing," *Cloud Computing*, pp. 1–41, 2011.

[10] N. Leavitt, "Is Cloud Computing Really Ready for Prime Time?," *IEEE Computer Society*, vol. 42, pp. 15–20, enero 2009.

[11] L. Richardson and S. Ruby, *RESTful web services*. Franham: O'Reilly, 2007.

[12] J. Louvel, T. Templier, and T. Boileau, *Restlet in Action*. Manning Publications, 13 ed., 2012.

[13] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Cáceres, *et al.*, "The RESERVOIR Model and Architecture for Open Federated Cloud Computing," *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 1–17, 2009.

[14] N. Sclater, "Cloud Computing in Education," tech. rep., UNESCO Institute for Information Technologies in Education, septiembre 2010.

[15] L. Johnson, A. Levine, and R. Smith, "The 2009 Horizon Report," tech. rep., 2009.

[16] C. N. Höfer and G. Karagiannis, "Cloud computing services: taxonomy and comparison," *Journal of Internet Services and Applications*, vol. 2, no. 2, pp. 81–94, 2011.

[17] W. R. Adrion, "Research Methodology in Software Engineering: Summary of the Dagstuhl Workshop on Future Directions in Software Engineering," *ACM SIGSOFT Software Engineering Notes*, vol. 18, no. 1, pp. 36–37, 1993.

[18] A. Wang and W. Jiang, "Research of Teaching on Network Course Based on NS-3," in *First International Workshop on Education Technology and Computer Science (ETCS'09)*, vol. 2, pp. 629–632, marzo 2009.

[19] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, p. 60, 2008.

[20] A. Sobeih, W. Chen, J. Hou, L. Kung, N. Li, H. Lim, H. Tyan, and H. Zhang, "J-sim: A simulation environment for wireless sensor networks," in *Proceedings of the 38th annual Symposium on Simulation*, pp. 175–187, IEEE Computer Society, 2005.

[21] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[22] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599–616, 2009.

[23] N. Sadashiv and S. M. D. Kumar, "Cluster, grid and cloud computing: A detailed comparison," in *Sixth International Conference on Computer Science and Education (ICCSE 2011)*, pp. 477–482, IEEE, 2011.

[24] T. Dillon, C. Wu, and E. Chang in *24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), title = Cloud Computing: Issues and Challenges, month = abril, year = 2010, pages = 27–33,*.

[25] B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," in *Fifth International Joint Conference on INC, IMS and IDC (NCM'09)*, pp. 44–51, agosto 2009.

[26] H. E. Schaffer, "X as a Service, Cloud Computing, and the Need for Good Judgment," *IT Professional*, vol. 11, pp. 4–5, septiembre-octubre 2009.

[27] E. Millman, D. Arora, and S. Neville, "STARS: A Framework for Statistically Rigorous Simulation-based Network Research," in *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA 2011)*, pp. 733–739, IEEE, 2011.

[28] M. L. Bote-Lorenzo, J. I. Asensio-Pérez, E. Gómez-Sánchez, G. Vega-Gorgojo, Y. A. Dimitriadis, and S. Guíñez-Molinos, "A Grid Service-based Collaborative Network Simulation Environment for Computer Networks Education," in *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, pp. 56–60, 2007.

[29] J. Louvel, T. Templier, and T. Boileau, *Restlet in Action*, ch. Designing a RESTful web API. Manning Publications, 13 ed., 2012.

[30] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Especificación de Requisitos Software," Tech. Rep. versión 2.3, Universidad de Valladolid, 2012.

[31] P. Kruchten, "The 4+1 View Model of Architecture," *IEEE Software*, vol. 12, pp. 42–50, november 1995.

[32] P. Kruchten, *The Rational Unified Process: An Introduction*. Addison-Wesley, 3 ed., 2004.

[33] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Arquitectura Software," Tech. Rep. versión 1.3, Universidad de Valladolid, 2012.

[34] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Almacenamiento," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[35] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Colas," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[36] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Escalado," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[37] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Estadística," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[38] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Informe," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[39] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Monitorización," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[40] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Orquestación," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[41] R. Cano-Parra, "Entorno de simulación de redes TCP/IP usando servicios REST basados en la nube computacional, Servicio de Simulación," Tech. Rep. versión 1.2, Universidad de Valladolid, 2012.

[42] G. von Laszewski, J. Diaz, F. Wang, and G. Fox, "Comparison of Multiple Cloud Frameworks," in *IEEE 5th International Conference on Cloud Computing (CLOUD 2012)*, pp. 734–741, junio 2012.

[43] B. A. Kitchenham and L. Jones, "Evaluating Software Engineering Methods and Tools, Part 7: Planning Feature Analysis Evaluation, journal = ACM SIGSOFT Software Engineering Notes, volume = 22, number = 4, pages = 21–24, year = 1997, publisher = ACM,"

Apéndice A

Glosario de Términos

API. *Application Program Interface*, en español Interfaz de Programación de Aplicaciones, es un conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado en otro software como una capa de abstracción.

Cola. En el contexto del Servicio de Colas, es el listado de elementos que contiene el servicio y que se implementará siguiendo una cola FIFO (*First Input, First Output*, es decir, el primero en entrar en la cola será el primero que salga de ella).

CRUD. *Create, Record, Update, Delete.* Son cuatro tareas comunes realizadas en los sistemas informáticos correspondientes a la vida de un objeto, como es la creación, guardado, actualización y eliminado de dicho objeto.

Distribución de recursos dinámica. Se denomina al reparto de un número finito de trabajos de simulaciones individuales entre un número de recursos computacionales disponibles que pueden aumentarse o disminuirse en un momento determinado según el número de trabajos que haya disponibles.

DNSE. *Distributed Network Simulation Environment*, en español Entorno de Simulación de Red Distribuido, es un entorno de simulación de redes TCP/IP distribuido en el que, a través de un paquete de simulaciones, se permiten realizar y obtener resultados de simulaciones individuales o de barrido de parámetros.

DNSE3. Es el nombre designado para la tercera versión del DNSE y que es la propuesta que se presenta en este trabajo de investigación.

Ejecución. Término utilizado para referirse a la acción de aplicar una serie de funciones implementadas por el simulador *ns-3* a un conjunto de parámetros definidos en una red TCP/IP concreta.

Elemento. En el contexto del Servicio de Colas, un elemento de la cola será aquel objeto que contiene la información necesaria para su procesamiento, el estado de procesamiento en el que se encuentra y el tiempo de vida máximo para su ejecución.

Escalabilidad horizontal. La escalabilidad es la característica que tiene un sistema de adaptarse a las condiciones actuales del entorno de ejecución, realizando de forma dinámica una adaptación de su entorno de forma que los recursos que están siendo utilizados sean los menores posibles. En una nube computacional, la escalabilidad horizontal se refiere a la instanciación de nuevas imágenes de un servicio en particular o a la terminación de las mismas según sea necesario.

Estadística. Se refiere a un conjunto de datos de entrada y una configuración de los parámetros de un método estadístico en concreto, sobre los que se aplica dicho método

para obtener un conjunto de datos de salida que serán los datos estadísticos resultantes.

HTML. *HyperText Markup Language* o Lenguaje de Marcas de Hipertexto, es un lenguaje de marcas predominante en la elaboración de páginas web que permite describir y traducir su estructura e información en forma de texto.

IaaS. *Infrastructure as a Service*, en español Infraestructura como Servicio, es el nivel más bajo de la nube computacional que se compone de los recursos hardware y los componentes físicos.

Informe. Es el documento, en un formato determinado, que contendrá la descripción de toda la información perteneciente a un paquete de simulación, tales como su configuración, sus simulaciones independientes y de barrido de parámetros, y de los resultados (datos en bruto, gráficas y estadísticas) y anotaciones (de simulación y de gráfica) realizadas sobre ellas.

Instanciación. La instanciación de una imagen de un servicio en la nube computacional se refiere al inicio de la ejecución de un nuevo tipo de servicio dentro del entorno de nube computacional del sistema a través de una imagen donde está instalado y definido dicho servicio.

Método estadístico. Es la descripción del contenido de un conjunto de datos de entrada, una configuración de parámetros y un conjunto de datos de salida sobre los que se aplicarán una serie de funciones matemáticas particulares para cada método, dando como resultado una estadística.

Métrica. Una métrica es la definición de una característica particular de un servicio del sistema que puede tomar uno o varios valores diferentes a lo largo de la ejecución, y que definen el estado actual o pasado en el que se ha encontrado el servicio.

Monitorización. La monitorización de servicios en una nube computacional se refiere a la gestión centralizada de los valores de las diferentes métricas ofrecidas por los servicios, de forma que cada servicio puede registrarse y publicar sus métricas y valores para que otros servicios puedan consultar el estado del mismo para otras gestiones particulares.

ns-3 El proyecto *ns-3* es una aplicación de software libre que implementa un simulador de red de eventos discretos para sistemas de Internet utilizado para fines educativos y de investigación.

Orquestación. En el contexto de este servicio, se refiere al control del flujo de las diferentes funciones a aplicar tras la solicitud de una acción, ya sea realizando ciertas funciones atómicas o redirigiendo dichas funciones a uno de los servicios que componen el sistema.

PaaS. *Platform as a Service*, en español Plataforma como Servicio, es el nivel intermedio de la nube computacional donde diferentes infraestructuras se organizan de diferentes formas para ofrecer una plataforma a los usuarios.

Paquete de simulación. Es un fichero comprimido que contiene tanto los metadatos que describen el diseño y los parámetros de la red TCP/IP a simular, como la descripción de las diferentes simulaciones definidas, los ficheros de los resultados generados, y las anotaciones realizadas en las simulaciones y las gráficas.

Parámetro. Es un tipo de dato especial configurado para un paquete de simulación o un método estadístico que, dependiendo del valor que se le asigne en una simulación o una estadística, dará un resultado distinto tras la ejecución de una simulación o la aplicación de un método estadístico sobre una estadística.

PDF. *Portable Document Format* o Formato de Documento Portátil, es un formato multiplataforma de almacenamiento de documentos digitales que permite la inclusión de texto, gráficos, enlaces y contenido multimedia.

Repositorio de ficheros. Lugar virtual en el que se almacenan los paquetes de simulación de cada usuario y localizado en el Servicio de Almacenamiento.

REST. *Representational State Transfer*, en español Transferencia de Estado Representacional, es un conjunto de criterios de diseño para la Arquitectura Orientada a Recursos (*Resource-Oriented Architecture*, ROA).

Servicio. Es un componente del sistema DNSE3 que tiene la capacidad de realizar una funcionalidad específica con cierta independencia dentro del sistema para el reparto de la carga del mismo.

SaaS. *Software as a Service*, en español Software como Servicio, es el nivel más alto de la nube computacional que representa al software que es ofrecido a los usuarios finales y que está ejecutándose en una de las plataformas disponibles.

Simulación de barrido de parámetros. Es un tipo de simulación en el que para cada parámetro definido en el paquete de simulación se proporcionan una serie de valores, de tal forma que una simulación de barrido de parámetros equivale a un conjunto de simulaciones individuales.

Simulación individual. Es un tipo de simulación en el que a cada parámetro definido en el paquete de simulación se le proporciona un único valor.

SOA. *Service-Oriented Architecture*, en español Arquitectura Orientada a Servicios, es una arquitectura de software que define la utilización de servicios para dar soporte a los requisitos funcionales del sistema.

Tiempo de vida. Se establece para un elemento de la cola del Servicio de Colas, y es el tiempo máximo que un elemento puede estar en estado de ejecución antes de que el servicio restablezca su estado a disponible nuevamente, suponiendo que no ha podido finalizarse su ejecución.

TCL. *Tool Command Language*, en español Lenguaje de Herramientas de Comando, es un lenguaje de script que ha sido concebido con una sintaxis sencilla para un aprendizaje fácil, sin detrimento en la funcionalidad y la expresividad, que es usado principalmente para el desarrollo rápido de prototipos.

TFM. Trabajo Fin de Máster, es la actividad en la que está englobada la realización de este documento.

URI. *Uniform Resource Identifier*, en español Identificador de Recursos Uniforme, es un identificador uniforme de recursos en Internet que consta de una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para identificar recursos, como documentos e imágenes en Internet, por su localización.

Visualización. Término utilizado para referirse a la acción de consultar los resultados que se han generado tras la ejecución de una simulación, y que puede realizarse de tres modos: consulta de los datos en bruto obtenidos para cada variable, la aplicación de un método estadístico a dichos datos, y la generación de gráficas a partir de dichos datos.

XML. *eXtensible Markup Language*, en español Lenguaje eXtensible de Marcado, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C) que no sólo ha nacido para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.