

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Análisis de enfoques de asignación de recursos en arquitectura Cloud-Fog
Computing

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Telemática

Presentado por:

Anel Ivette Martínez Chávez

Marlon Alexander Segarra Zambrano

GUAYAQUIL - ECUADOR

Año: 2020

DEDICATORIA

El presente trabajo es dedicado a las personas que más amo y representan un gran significado en mi vida, mis padres quienes me han apoyado en cada una de las etapas de mi vida. Especialmente a mi padre, que me demostró que casi nada es imposible, sino se lo intenta y se aprende, porque nunca se debe rendirse a un obstáculo sino se ha enfrentado a este sin miedo a las consecuencias que pueda suceder. A mi madre, que a lo largo de los años cultivo ese lado de amor y perdón que se debe brindar a los demás. A ellos dos, con mi inmenso amor dedico este trabajo y mi vida.

DEDICATORIA

El presente trabajo es dedicado a las personas que más amo y que constituyen un pilar fundamental en mi vida, a mi madre quien me enseñó que si algo se está tornando difícil hay que tomarse un descanso y retomarlo de nuevo con ideas frescas ya que algunas dificultades hay que tomárselas con calma además de darme ánimos para intentarlo de nuevo en caso de tener alguna falla, a mi padre quien me enseñó que uno debe informarse antes de dar ideas o establecer distintos puntos de vista. A mis hermanos quienes me enseñaron que a pesar de las dificultades se puede salir adelante y seguirse superando. A ellos dedico este trabajo, mi esfuerzo y dedicación.

AGRADECIMIENTO

Agradezco a mis padres por siempre estar presentes, brindándome su apoyo y su amor en cada una de las dificultades que se me han presentado. A mi pareja sentimental, que siempre me ha brindado su apoyo y ayuda incondicional en cada momento de alegría o tristeza. A mi sobrina, quien me ha sacado una risa en momentos de estrés completo. A mi hermano a quien le agradezco su apoyo y atención que nos brinda siempre. A todos ellos les agradezco con mucha gratitud y amor su presencia en mi vida.

AGRADECIMIENTOS

Mi más sincero agradecimiento a mis padres los cuales me apoyaron mucho a lo largo de toda mi carrera impulsándome a seguir adelante. A mi pareja sentimental por brindarme su ayuda y apoyo incondicional en todo momento ya sea de alegría, tristeza o de estrés total. Agradezco a mis hermanos por siempre tener confianza en mi y apoyarme con mis metas a futuro brindándome consejos que han servido a lo largo de mi vida. A ellos les agradezco siempre su confianza, amor, apoyo y su presencia en mi vida que es de vital importancia para mí.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Marlon Segarra y Anel Martínez* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Marlon Segarra

Autor 1

Anel Martínez

Autor 2

EVALUADORES

.....
Nombre del Profesor

PROFESOR DE LA MATERIA

.....
Nombre del Profesor

PROFESOR TUTOR

RESUMEN

Los recientes avances en tecnología han proporcionado múltiples entornos de desarrollo para las industrias de tecnología de la información en cuanto a su infraestructura de información, los cuales se adaptan de acuerdo con las funciones que desempeñan estas industrias. Sin embargo, no logran cubrir la creciente demanda de información que los usuarios están generando actualmente. Considerando el uso eficiente de los componentes de la infraestructura (almacenamiento, procesamiento y red), en este trabajo se evalúa una topología híbrida, lo que significa combinar procesamiento de nube y niebla. El objetivo principal es proporcionar un mecanismo que permita seleccionar el esquema más adecuado para una arquitectura dada después de comparar el desempeño de los diferentes esquemas de asignación de recursos. Para ello, desarrollamos e implementamos un simulador basado en una herramienta de código abierto (Ifogsim). Este código de fuente abierta está basado en Java. Nuestro simulador se implementó para permitir a las empresas realizar comparaciones rápidas de diferentes enfoques de asignación de recursos. Para tener una mejor ilustración de las métricas de rendimiento, el simulador incluye una función para elegir el tipo de gráfico (es decir, gráficos de barras o de líneas). Para fines de comparación, ejecutamos la simulación para diferentes arquitecturas variando el número de nodos, y obtuvimos varios gráficos comparativos con las principales métricas de rendimiento como el consumo de energía, la latencia promedio, el tiempo de ejecución del algoritmo y el uso de la red. A partir de los resultados numéricos, se puede observar que el enfoque híbrido mostró mejores resultados en términos de las métricas de desempeño evaluadas, lo que se puede atribuir debido a la mejor distribución de la carga en la red.

Palabras Clave: Ifogsim, nube, niebla, híbrido

ABSTRACT

Recent advances in technology have provided multiple development environments for the information technology industries regarding their information infrastructure, which are adapted according to the functions these industries perform. Nevertheless, they fail to cover the ever-growing demand of information that users are currently generating. Considering the efficient use of infrastructure components (storage, processing, and network), in this work, a hybrid topology is evaluated, which means combining cloud and fog processing. The main goal is to provide a mechanism such as the most suitable scheme for a given architecture can be selected after comparing the performance of the different resource allocation schemes. For this purpose, we developed and implemented a simulator based on an open-source tool (Ifogsim). This open-source code is based on Java. Our simulator was implemented to allow companies to perform rapid comparisons of different resource allocation approaches. To have a better illustration of the performance metrics, the simulator includes a feature to choose the graph type (i.e., bar or line graphs). For comparison purpose, we run the simulation for different architectures varying the number of nodes and obtained several comparative graphs with the main performance metrics such as energy consumption, average latency, algorithm execution time and the use of the network. From the numerical results, it can be noticed that the hybrid approach showed better results in terms of the evaluated performance metrics, which can be attributed due to the better load distribution over the network.

Keywords: *Ifogsim, cloud, fog, hybrid*

INDICE GENERAL

RESUMEN	I
ABSTRACT	II
ABREVIATURAS	V
SIMBOLOGÍA.....	VII
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1	1
1 INTRODUCCIÓN.....	1
1.1 Descripción del problema	2
1.2 Justificación del problema.....	3
1.3 OBJETIVOS	4
1.3.1 Objetivo General.....	4
1.3.2 Objetivos Específicos	4
1.4 Metodología.....	5
1.4.1 Análisis comparativo de la comunicación en la nube, la niebla o híbrida.....	5
1.4.2 Comparativa entre las herramientas de código abierto	6
1.5 Resultados esperados	7
CAPÍTULO 2	7
2 Marco teórico.....	7
2.1 Componentes de la arquitectura de computación en la nube y en la niebla	8
2.2 Tipos de Arquitecturas	9
2.3 Enfoques de asignación de recursos.	12
2.3.1 Centralizada.....	13

2.3.2	Distribuido	13
2.3.3	Hibrido.....	14
2.4	Programas de Simulacion de Codigo Abierto	16
2.5	Limitaciones.....	18
2.6	Estructura del Modelo de Simulacion a implementar.....	20
CAPITULO 3		23
3	ANALISIS DE RESULTADOS	23
3.1	Algoritmos de administración de recursos	23
3.1.1	Algoritmo Hibrido (Edge-Ward).....	23
3.1.2	Algoritmo Centralizado (Only-Cloud).....	23
3.1.3	Algoritmo Distribuido (Mapping).....	23
3.2	Simulador Cloud-Fog Simulator.....	24
3.2.1	Ventana Principal	24
3.2.2	Ventana de configuración de la nube y niebla	25
3.2.3	Ventana de resultados	27
3.3	Casos de estudio.....	29
3.3.1	Caso 1: Comparativa de dos dispositivos	29
3.3.2	Caso 2: Comparativa de 1 a 8 dispositivos	32
3.4	Análisis de Costos.....	37
CAPITULO 4		39
4	CONCLUSIONES Y RECOMENDACIONES	39
4.1	Conclusiones.....	40
4.2	Recomendaciones.....	41
BIBLIOGRAFIA		42

ANEXOS	44
ANEXO 1: MANUAL DE INSTALACION.....	44
Requerimientos de instalación.....	45
Pasos de instalación	46
ANEXO 2: MANUAL DE USUARIO.....	51
Ejecución del programa	51
Ventana principal del simulador	51
Ventana de configuración	52
Ventana de configuración de nodos en la nube	52
Ventana de configuración de nodos en la niebla	54
Ventana de resultados	55

ABREVIATURAS

MIPS Million Instructions per Second

BW	Bandwidth
TI	Tecnología de la información
IoT	Internet of Things
F2C	Fog to Cloud
QoS	Quality of Services
CSV	Comma separated values
JSON	JavaScript Object Notation
YAFS	Yet Another Fog Simulator
CPU	Central Processing Unit
RAM	Random Access Memory
OSM	Open Street Map
VM	Virtual Machine
XML	Extensible Markup Language
SDN	Software Defined Networking

SIMBOLOGÍA

MJ Mega Joules

KB Kilo Bytes

ÍNDICE DE FIGURAS

Figura 1.1 Curva de tendencias tecnológicas 2020 [1]	3
Figura 2.1 Topología general de recursos [8].....	10
Figura 2.2: Arquitectura Cloud-Fog-Edge [9].....	11
Figura 2.3: Arquitectura Cloud-Fog-Fog-Edge [8]	11
Figura 2.4 : Arquitectura Cloud-Fog-Fog-Fog-Edge [10].....	12
Figura 2.5:Enfoque centralizado	13
Figura 2.6: Enfoque distribuido	14
Figura 2.7: Enfoque hibrido	14
Figura 2.8: Escenario principal a resolver.....	21
Figura 2.9: Esquema de simulación.....	22
Figura 3.1: Ventana principal del simulador Cloud/Fog Computing.....	25
Figura 3.2: Configuración de nodos en la nube.....	26
Figura 3.3: Configuración de nodos en la niebla.	27
Figura 3.4: Ventana de resultados de la simulación.....	28
Figura 3.5: Topología usando 2 dispositivos	29
Figura 3.6: Energía consumida con respecto a la comparativa de dos dispositivos.....	30
Figura 3.7: Latencia promedio con 1 y 2 dispositivos.	31
Figura 3.8: Tiempo de ejecución del algoritmo con 1 y 2 dispositivos.....	31
Figura 3.9: Uso de la red con 1 y 2 dispositivos.	32
Figura 3.10: Topología usando 8 dispositivos	33
Figura 3.11: Energía consumida con varios dispositivos.	34
Figura 3.12: Latencia promedio con varios dispositivos.....	35
Figura 3.13: Tiempo de ejecución de varios dispositivos.....	36
Figura 3.14: Uso de la red de varios dispositivos.	37

ÍNDICE DE TABLAS

Tabla 3.1 : Descripción de parámetros de ventana principal del simulador.	24
Tabla 3.2: Descripción de parámetros de ventana de configuración en la nube y la niebla.	26
Tabla 3.3: Descripción de métricas de resultados.....	28
Tabla 3.4: Análisis de costos de personal.	38

CAPÍTULO 1

1 INTRODUCCIÓN

En los últimos años el campo tecnológico ha ganado cada vez más fuerza, principalmente para la administración eficiente de los recursos en los sistemas de información, que recopilan gran cantidad de datos en tiempo real para su procesamiento y análisis respectivo, en donde la tolerancia a fallos y la sincronización son indispensables para otorgar una respuesta inmediata a un servicio. Principalmente se ha priorizado, el uso de la computación de la nube (cloud computing) que maneja una infraestructura centralizada (centro de datos), como una herramienta fundamental para lograr la recopilación y procesamiento de datos de múltiples dispositivos, cuyos servicios proporcionan funciones de copias de seguridad, intercambio y respaldo de datos, brindando a las empresas el beneficio de reducción de costos en la infraestructura de hardware y la garantía de tener una alta disponibilidad y confiabilidad de los datos en la nube.

A pesar del buen funcionamiento que ha tenido este nuevo paradigma, existen algunas limitaciones vitales al momento de transmitir y procesar la información tales como la latencia, que es el tiempo de espera de una respuesta a una solicitud y el tiempo de respuesta que está conformada por el tiempo de procesamiento (tiempo que le toma al sistema procesar una solicitud) y el tiempo de transmisión (tiempo que le toma al mensaje ir del cliente al servidor o el tiempo de retorno al cliente).

Lo que genera una interrogante acerca de la necesidad de subir información a la nube o en su defecto hacer un procesamiento local lo más cerca posible de la fuente de información para obtener un mayor control en tiempo real, como el uso de la computación en la niebla que se encuentra cercana a la red de sensores y actuadores, con millones de nodos desplegados que operan con un mayor tiempo de respuesta, por lo cual lo más óptimo sería realizar un procesamiento local para ejecutar una acción en un tiempo adecuado, y en el caso en que la información suministrada no pueda ser procesada localmente por cuestiones

de memoria o poder de procesamiento se recurre a enviar la información a la nube para su respectivo almacenamiento.

Debido a los factores antes mencionados, en este trabajo se propone realizar el análisis de desempeño de esta arquitectura, mediante su simulación dentro de un ambiente virtualizado, que permita una oportuna gestión de la administración de los recursos (infraestructura y datos), disminución de la sobrecarga de control de procesamientos, con la finalidad de maximizar el uso de los recursos, el desempeño y el tiempo de respuesta de los servicios.

1.1 Descripción del problema

La computación en la nube sin lugar a duda ha facilitado la gestión de la infraestructura en las empresas , mejorando el manejo de la información en el sector TI, sin embargo, para demandas específicas que incluyen la actualización de información en tiempo real es intolerable que haya degradación en la latencia, como es el caso de las plantas de fabricación o los sistemas de control de tráfico, es por esto que la computación en la niebla aborda demandas tales como la latencia, el tiempo de descarga, la optimización de energía y la seguridad entre la comunicación de dispositivos, pero este procesamiento tiene también sus limitaciones en cuanto a la capacidad de procesamiento por lo que ambas tecnologías se complementan para generar un sistema integrado.

En la figura 1.1 se determina la curva Gartner para el año 2020 el auge en la tecnología involucra el Internet de las cosas y sus aplicaciones tales como ciudades inteligentes, monitoreo de salud, seguimiento logístico e industrias inteligentes, lo cual ha dificultado el uso del cloud computing debido al gran volumen de datos generado, lo que hace que muchas empresas dejen de lado su uso, sin embargo, la implementación de estas nuevas tecnologías involucra una velocidad de procesamiento mayor, así como optimizar el tiempo de respuesta y las operaciones que se deben efectuar para obtener un resultado, es por esto que surge el concepto de Hybrid cloud computing el cual involucra la integración del fog computing para poder complementarse y brindar una mejor arquitectura en cuanto a la comunicación de estos dispositivos, los cuales dependiendo de la cantidad de datos que manejen serán manejados por la nube o por la niebla.

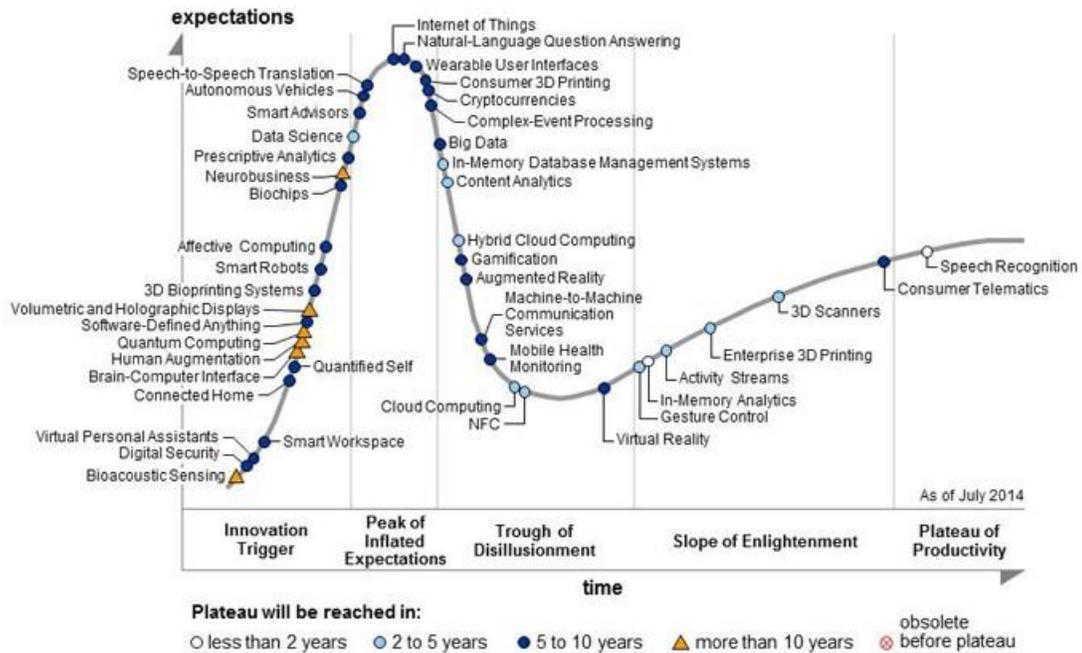


Figura 1.1 Curva de tendencias tecnológicas 2020 [1]

La mejor forma de visualizar el comportamiento de estos componentes es mediante un simulador en el cual se podrá apreciar la comparativa de los distintos enfoques (nube, niebla, e híbrido) para el manejo óptimo de los recursos en la nube y en la niebla a fin de obtener un mejor desempeño en el sistema.

1.2 Justificación del problema

El uso del cloud y fog computing en sistemas que dependan del tiempo de respuesta, la latencia y la administración de recursos puede producir un retraso importante en dispositivos tales como actuadores en donde su correcto funcionamiento depende del rápido procesamiento de la información. Por los factores desarrollados para el análisis de la

optimización de los recursos, es necesario la integración de la computación en la nube y la niebla, para brindar principalmente a las empresas una óptima identificación y resolución de problemáticas de gestión, con el objetivo de la reducción de los costos desarrollados para la implementación de una amplia infraestructura desplegada a nivel geográfico de los nodos de red que se encargan enviar la trama de comunicación.

Con el fin de reducir el ancho de banda, la latencia de la red, reducción de los cuellos de botellas generados, reducción de fallas de comunicación y procesamiento de la data, mediante un aumento de velocidad del análisis de los datos con un procesamiento local en caso de la que la información a procesar es necesaria de manejarla en determinados nodos por su nivel de prioridad, para la correcta accesibilidad desde diversos puntos remotos, dependiendo de la sensibilidad de los datos en el tiempo de transmisión y procesamiento de la localización de que la data sea almacenada y procesada en nodos de agregación cercanos a las fuentes de información o en la nube para realizar la respectiva solicitud de servicio. Obteniendo como resultado un mejor manejo del almacenamiento y procesamiento de la información, reduciendo los tiempos de transmisión y ejecución de tareas.

1.3 OBJETIVOS

1.3.1 Objetivo General

- Evaluar la topología híbrida mezclando procesamiento en la nube y en la niebla mediante la simulación de dicha arquitectura en una herramienta de código abierto con la finalidad de comparar el rendimiento respectivo de los diferentes esquemas de asignación de recursos.

1.3.2 Objetivos Específicos

- Evaluar las herramientas de código abierto disponibles para la simulación de las arquitecturas de computación el nube, niebla, redes de sensores y actuadores mediante la comparación de estas, para la apropiada selección de la herramienta más idónea.

- Determinar el tipo de enfoque a utilizar, por medio del uso de los ejemplos proporcionados por la herramienta de simulación a fin de determinar aquel que permite el manejo óptimo de los recursos en la industria TI.
- Evaluar los resultados obtenidos de las pruebas experimentales realizadas para el análisis de los parámetros de interés (latencia y tiempo de respuesta).

1.4 Metodología

Referente al estudio inicial realizado sobre los distintos tipos de computación en la nube y en la niebla y de acuerdo con los trabajos realizados se procede a describir los distintos factores a tomar en consideración para la correcta gestión de los recursos además de realizar la respectiva comparación entre las diferentes herramientas de código abierto en donde se tomará en cuenta los parámetros de gran importancia para una correcta comunicación con el cliente a fin de proporcionar una mejor calidad en cuanto a la optimización de recursos ofreciendo una simulación del procesamiento local (computación en la niebla) o remoto(computación en la nube) de la información suministrada por los dispositivos.

1.4.1 Análisis comparativo de la comunicación en la nube, la niebla o híbrida

Actualmente la industria de TI se enfoca en su mayoría al manejo de recursos en la nube [2] debido a su capacidad de procesamiento y almacenamiento dentro de grandes centros de datos remotos, sin embargo, se poseen ciertas dificultades en cuanto a acceder a la información sin conexión a internet o un potencial riesgo en la seguridad al tener información sensible almacenada.

El manejo de recursos en la niebla por otra parte desempeña un papel de intermediario o mediador entre los grandes centros de datos y los dispositivos finales, es por esto por lo que se ubican lo más cerca posible del cliente, por lo que los tiempos de respuesta son menores en este caso y al poseer una gran diversidad de protocolos de comunicación, facilita el

manejo de la seguridad en los dispositivos, sin embargo, su capacidad de almacenamiento es limitada.

Debido a estos factores antes mencionados se vuelve necesaria la adopción de una arquitectura híbrida en donde las deficiencias de la nube sea compensada por la niebla y de esta forma realizar un análisis comparativo de la comunicación en la arquitectura híbrida (nube y niebla) con respecto a una arquitectura centralizada de la nube o distribuida como la niebla, es imperativo el análisis de los beneficios y prejuicios que ofrecen ambas para poder solventar las deficiencias de cada uno y poder obtener una alta confiabilidad.

Entre los principales factores a considerar en un escenario combinado, son la alta escalabilidad, la baja latencia y flexibilidad en el mantenimiento y configuración de los dispositivos conectados, además del análisis son el manejo del procesamiento de la data de forma remota con respecto al procesamiento de los datos que se encuentran más cerca de la fuente de información, según el tiempo de respuesta a una petición solicitada por el cliente.

1.4.2 Comparativa entre las herramientas de código abierto

Con la finalidad de obtener los parámetros de rendimiento óptimos para la simulación de una arquitectura híbrida y obtener mejores resultados de análisis de gestión de recursos, se consideraron distintas herramientas de simulación de código abierto existentes.

Una de las herramientas de simulación analizadas es Ifogsim [3], la cual permite la simulación de entornos en la nube y en la niebla, a diferencia de otras herramientas de simulación como los mencionados en [4], donde se enfocan únicamente a la nube o a la niebla, además de que la arquitectura brindada es de un modelo de capas, la cual es responsable de una tarea en específico (actuadores, sensores) que siguen un modelamiento de flujo de datos distribuidos para las aplicaciones construidas en conjunto con los elementos de procesamiento que mantienen una comunicación entre sí.

Entre las características principales brindadas por esta herramienta, en cuanto al modelamiento de administración de recursos, se encuentra la evaluación de los métodos de gestión de recursos en la red, para asegurar el cumplimiento de la calidad de servicio mediante la elección del mejor candidato para la asignación de los recursos en un determinado modulo, en caso de congestión de red.

Otra de las características a destacar de esta herramienta de simulación es, el monitoreo del consumo de energía durante la simulación y disponibilidad de los recursos en la red es de vital importancia debido a que los dispositivos implementados en estos entornos poseen limitaciones de batería.

1.5 Resultados esperados

De acuerdo al análisis realizado en cuanto a las comparativas sobre una arquitectura de comunicación conjunta de la nube y la niebla, y las herramientas de simulación de código abierto existentes, se determina la implementación de un modelo de comunicación de la nube y la niebla, en donde se realice la administración de los recursos de la red, reflejando el manejo del tiempo de procesamiento y el tiempo de espera durante la ejecución de una petición establecida por parte cliente, dejando un modelo simulado y una arquitectura que conlleve una correcta administración de los recursos en la red.

CAPÍTULO 2

2 Marco teórico

En el presente marco teórico, se fundamentará los conceptos primordiales a evaluar durante el desarrollo de esta investigación. En primer lugar, se determinará los modelos que corresponden a las capas de comunicación que intervienen en el desarrollo del proyecto y

el medio de implementación de estos modelos. Además de la referencia de los campos aplicativos en los cuales se han desarrollado soluciones TI relacionados a la arquitectura propuesta.

2.1 Componentes de la arquitectura de computación en la nube y en la niebla

Una arquitectura es el modelo, estructura o diseño conceptual de un sistema, en donde se definen los requerimientos e implementaciones necesarias para las funciones operacionales con el conjunto de componentes que conforman una arquitectura. A continuación, se detallarán los componentes involucrados en una arquitectura en la nube y en la niebla.

Nube(Cloud Computing): modelo de conexión de múltiples servidores, computadores y distintos dispositivos conectados entre sí a través del internet, que permiten el acceso de la red por solicitud a un determinado grupo compartido de recursos informáticos (servicios, aplicaciones, almacenamiento y servidores), mediante el acceso de la interfaz de los dispositivos de los clientes (computadores, celulares) que se conectan a los sistemas de procesamiento y almacenamiento de los datos para la solicitud de recursos al proveedor de servicios [5].

Niebla (Fog Computing): escenario de arquitectura horizontal donde una gran cantidad de dispositivos descentralizados heterogéneos (inalámbricos o autónomos) conocidos como “fog nodes” los cuales se encuentran cerca de los dispositivos finales tales como sensores, actuadores y computadores, manteniendo una comunicación distribuida entre sí y con la red, para el cumplimiento de las funciones de computación, control, almacenamiento y procesamiento sin intervención de agentes terceros, otorgando altos niveles de velocidad de transmisión de data confiable y baja latencia para la recepción de respuestas de análisis de datos mayor [5].

Redes de sensores (Sensor Networks): constituida por una serie de nodos sensores que se encuentran ampliamente desplegados en distintos puntos geográficos que se consideran inaccesibles y que se comunican entre sí o directamente con una estación, base externa o

Gateway (dispositivo de enlace de equipos), que se encargan primordialmente del censado, computación y comunicación inalámbrica de la data recopilada al coordinador [6].

Dispositivo de Borde (Edge Device): componente de hardware encargado del control del flujo de datos y limitaciones entre dos redes. Estos dispositivos cumplen con funciones como enrutamiento, procesamiento, transmisión y monitoreo, además de que funcionan como puntos de entrada o salida según la tarea a la cual este siendo aplicado.

2.2 Tipos de Arquitecturas

Debido al surgimiento de nuevas tecnologías como el uso de sensores y actuadores para el procesamiento de varias tareas y gran volumen de datos con el fin de obtener una respuesta eficiente, ha surgido una evolución para la computación en la nube, tal como aplicar procesamiento en la niebla y en la nube para obtener los beneficios de ambas capas, como se observa en la figura 2.1 en donde se muestran los distintos tipos de procesamiento.

La computación en la nube surgió en la última década con el propósito de facilitar la segmentación de los recursos informáticos de los dispositivos de borde, pero este paradigma evidenció problemas en cuanto al procesamiento y transmisión en tiempo real, baja seguridad y alta carga de red, obstaculizando el desempeño de los nuevos servicios de IoT [7]. Debido a estas falencias surgió la computación en la niebla, que cuenta con baja latencia, eficiencia energética y recursos, sin embargo, el almacenamiento es un gran desafío debido a que para obtener una alta consistencia de los datos requiere más esfuerzo.

Pero en la actualidad se aplica una combinación de ambas capas, con el fin de obtener beneficios y solventar las falencias que se ejecutaban por separada en estas capas. Existe un novedoso paradigma de computación que ha sido propuesto recientemente con el objetivo de mejorar la integración de Fog Computing y Cloud Computing a través de la gestión coordinada de los recursos subyacentes, teniendo en cuenta las peculiaridades inherentes a cada modelo de computación y permitiendo la ejecución paralela y distribuida de servicios en distintos recursos de niebla / nube.

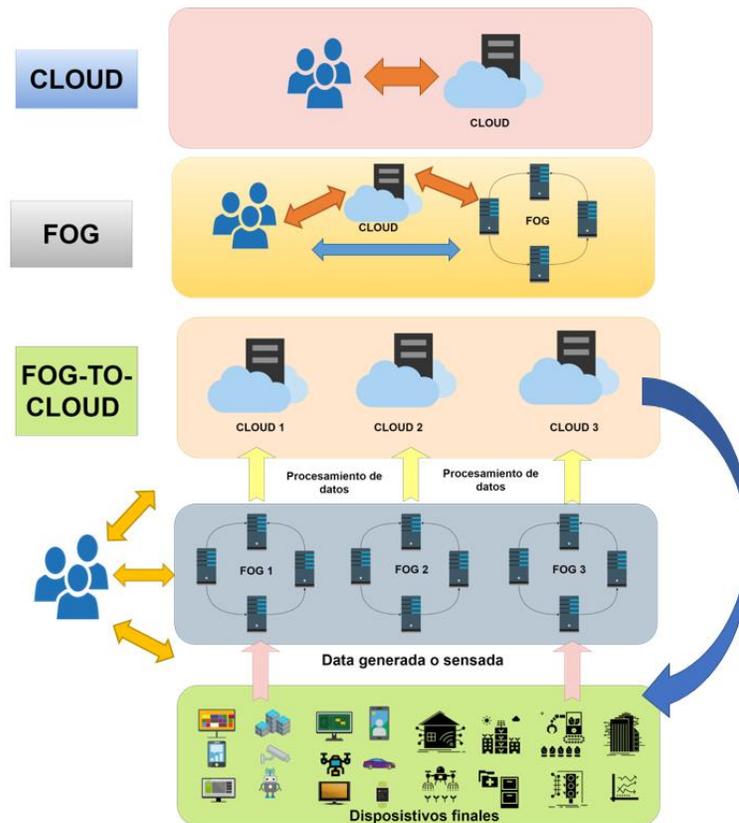


Figura 2.1 Topología general de recursos [8]

Este nuevo paradigma ofrece distintos tipos de arquitecturas, las cuales serán detalladas a continuación:

Cloud-Fog-Edge: en la arquitectura mostrada en la figura 2.2, implementan una nube central en la cual realiza el procesamiento y almacenamiento de datos masivos y la toma de decisiones, en la capa en la niebla usan una nube local en donde se encuentra conectados nodos adyacentes que se encargan del acceso a las comunicaciones entre los nodos, y finalmente la capa de borde en donde se encuentran un conjunto de dispositivos que comparten recursos computacionales y almacenamiento para generar una cooperación dinámica que cubra con la demanda de requisitos.

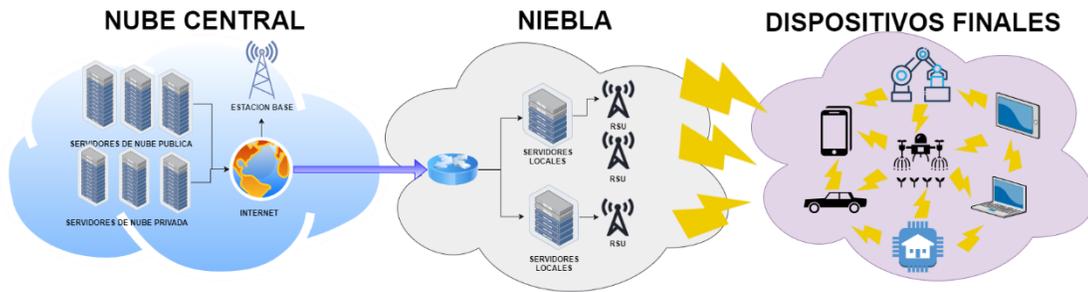


Figura 2.2: Arquitectura Cloud-Fog-Edge [9]

Cloud-Fog-Fog-Edge: en la arquitectura mostrada en la figura 2.3, se observa que la capa de la nube se encarga de realizar las tareas de cálculos computacionales, almacenamiento y conectividad. Existen dos dominios de fog, el primer dominio se encarga del procesamiento de las aplicaciones en tiempo real para el rendimiento del servicio, en el segundo maneja la paralelización de los servicios para su respectiva coordinación, y por último en el borde están los dispositivos finales conectados.

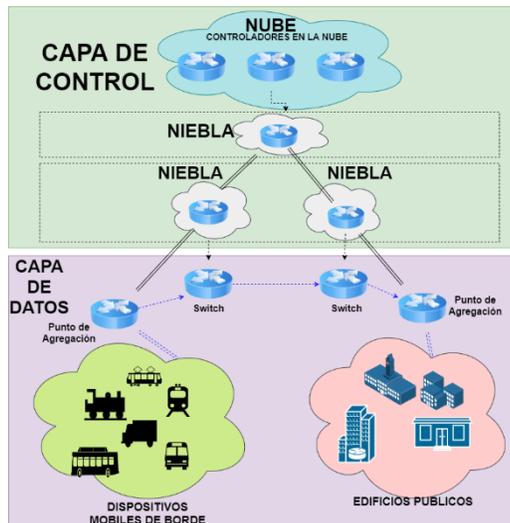


Figura 2.3: Arquitectura Cloud-Fog-Fog-Edge [8]

Cloud-Fog-Fog-Fog-Edge: en la arquitectura presentada en la figura 2.4, se usa la primera capa de la nube en donde se recopila los datos e información de cada nodo de computación de la capa 2 en donde se encuentran conectadas decenas de dispositivos intermedios que reciben el flujo de datos provenientes de la capa 3 en donde están localizados los nodos de borde, que identifican patrones de flujo de datos entrantes de los sensores para la

respectiva extracción de características computacionales provenientes de la capa 4 que es donde se encuentran los componentes directos de recepción de la data de los sensores, y la capa de borde, donde están localizados los sensores o dispositivos finales.

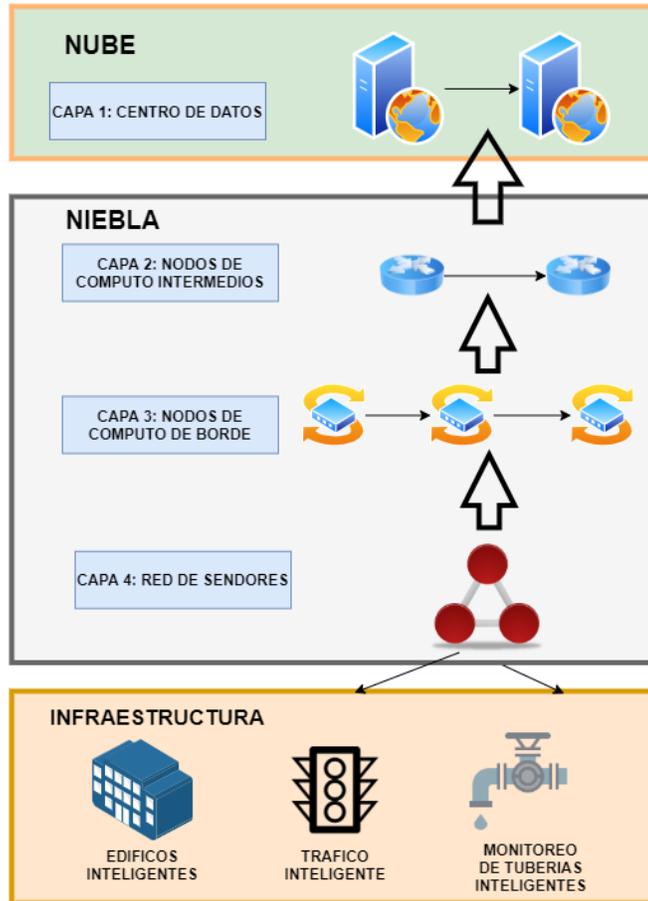


Figura 2.4 : Arquitectura Cloud-Fog-Fog-Fog-Edge [10]

2.3 Enfoques de asignación de recursos.

La asignación de recursos se define como el proceso de programación y distribución de los recursos disponibles en un determinado tiempo, con el objetivo de maximizar la eficiencia del uso de estos recursos. Es fundamental implementar la asignación de recursos, en las arquitecturas mencionadas debido a las falencias del manejo que se evidencian en la implementación de las capas individuales, como la deficiencia de la transmisión en tiempo real(nube) y la gestión de almacenamiento(niebla). Logrando con una combinación de

ambas arquitecturas, un modelo de reducción de recursos de procesamiento y comunicación.

2.3.1 Centralizada

En este tipo de asignación [11], como se observa en la figura 2.5 el control centralizado es una técnica la cual se basa en un controlador central el cual maneja toda la carga de trabajo de los dispositivos situados en el borde de la red, en donde el controlador que dirige toda la ejecución de las operaciones es la nube, es una de las arquitecturas más utilizadas, a pesar de que presenta dificultades manejando la escalabilidad.

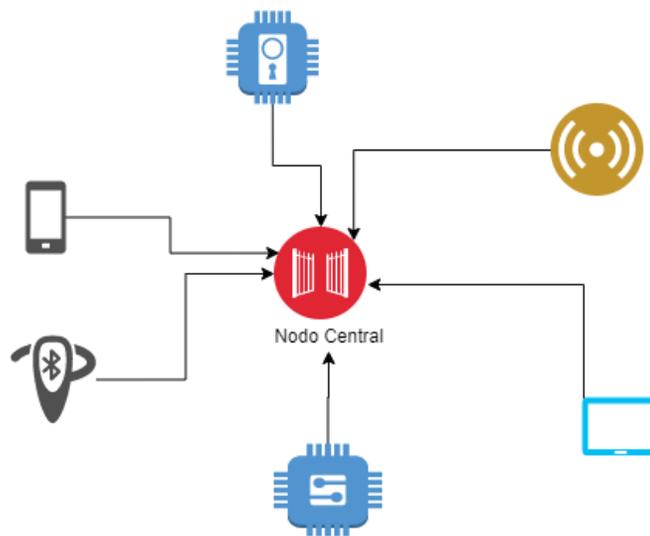


Figura 2.5:Enfoque centralizado

2.3.2 Distribuido

En cuanto al enfoque distribuido [11] también nos menciona que, a diferencia de la centralizada, esta nos permite distribuir la carga de trabajo a los dispositivos directamente conectados al dispositivo final como se muestra en la figura 2.6.

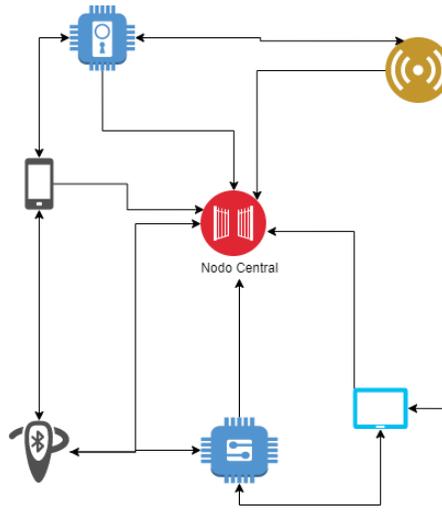


Figura 2.6: Enfoque distribuido

2.3.3 Híbrido

En la asignación híbrida se permite combinar lo mejor de los dos tipos de arquitectura como se muestra en la figura 2.7 brindando de esta forma confiabilidad, escalabilidad y fácil mantenimiento además de la capacidad de soportar una evolución en la tecnología [8].

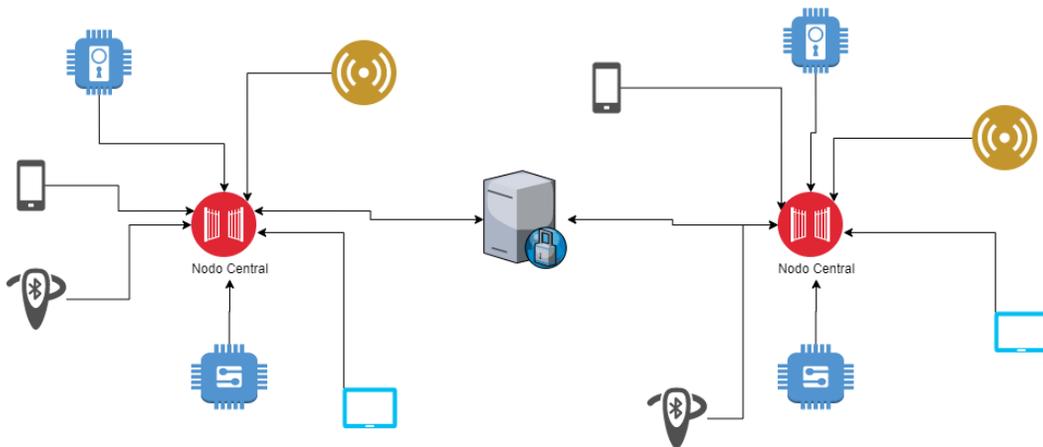


Figura 2.7: Enfoque híbrido

2.3.4 Propuestas de esquemas híbridos de asignación de recursos

En [10], se presenta el modelo de una arquitectura enfocada a la gestión de recursos, que mantiene como base un módulo central de un sistema F2C (fog-to-cloud) orientado a una ciudad inteligente. La función de esta arquitectura es la evaluación de los recursos y

servicios según las preferencias y políticas de uso compartido de los usuarios, mediante la selección de recursos de acuerdo con el modelo de costos, en donde se considera la dimensión de costos y el consumo energético.

Otro estudio aplicado en el campo de la medicina es presentado en [12], para el análisis de la coordinación entre capas (nube, niebla y borde) y la estructuración e integración de los servicios, dependiendo del grado de control y gestión de cada uno. Mostrando las mejoras que ofrece la aplicación del modelo F2C en cuanto a la optimización del tiempo de ejecución y aceleración para un determinado servicio médico comparándolo con un escenario propuesto en la nube.

En [13], se plantea un análisis de los servicios dinámicos de IoT según los paradigmas de la computación en la nube y la niebla, además de la combinación de ambas en un modelo F2C, realizando el respectivo análisis de un modelo de ejecución paralelo y distribuido, lo cual garantizará un mejor desarrollo de procesamiento de los servicios, resultados de análisis del tiempo de retraso y carga del núcleo de la red.

La tabla 2,1, detalla las fortalezas y debilidades de cada tipo de arquitectura tales como problemas en escalabilidad o la confiabilidad que posee cada enfoque de asignación de recursos. Se evidencia que el esquema centralizado posee un alto mantenimiento, niveles de ajustes y evolución al contrario del esquema distribuido que posee un bajo nivel en estos puntos. Para el esquema híbrido todas las características son moderadas, lo cual nos dice que presenta un mayor equilibrio.

Característica	Centralizada	Híbrida	Distribuida
Confiabilidad	Bajo	Moderado	Alto
Mantenimiento	Alto	Moderado	Bajo
Estabilidad	Bajo	Moderado	Alto
Escalabilidad	Bajo	Moderado	Alto
Ajustes	Alto	Moderado	Bajo
Evolución	Alto	Moderado	Bajo

Tabla 2.1: Comparación de arquitecturas de control [8]

2.4 Programas de Simulación de Código Abierto

Para la asignación de recursos en cloud-fog se analizaron diversas herramientas de código abierto en donde se compararon factores claves tales como: lenguaje de programación utilizado, topología, inyección de fallas, modelo de aplicación, movilidad, escalabilidad, entre otras, con la finalidad de identificar la herramienta idónea que permita un correcto control para la ejecución paralela y distribuida de servicios en distintos recursos de niebla / nube. A continuación, se presentarán las herramientas sujetas al análisis respectivo.

Cloudsim: es una herramienta de simulación la cual nos permite modelar y simular tanto el sistema como el comportamiento de los componentes de este en la nube, centros de datos, máquinas virtuales, y políticas para el aprovisionamiento de recursos las cuales se pueden ampliar con facilidad y con un esfuerzo limitado [14].

Ifogsim: Es una herramienta de simulación la cual permite analizar las políticas de manejo de recursos en la niebla tales como la latencia, consumo de energía, congestión de red y costos operativos. Entre sus principales simulaciones se encuentran dispositivos de borde, centros de datos en la nube, enlaces de red los cuales son usados para medir el rendimiento. Todas aquellas aplicaciones que se ejecutan en dispositivos de niebla llevan una comunicación Publicador/Suscriptor en donde los resultados de los procesos acaban en las acciones tomadas por los actuadores [15].

YAFS (Yet Another Fog Simulator): se encarga de analizar el desempeño y programación de estrategias de enrutamiento, que brinda al usuario métricas tales como tiempo de uso de la red, repuesta y espera, además del cálculo de métricas de QoS. Los datos generados son presentados en un archivo .CSV, así como su topología es definida mediante un archivo de formato JSON. Implementa el lenguaje de programación Python 2.7, y no soporta otras versiones de Python actualizadas [16].

EmuFog: entorno de emulación para la niebla desarrollado en Java, que permite al usuario generar aplicaciones basadas en Docker. Una de las funcionalidades que ofrece, es el seguimiento de eventos de cada nodo local, mediante un registro del consumo de CPU y memoria, además de permitir la ubicación de los nodos en la niebla, definiendo el nivel en la topología con distintas políticas de ubicación y carga de trabajo, pero no permite visualizar una interfaz gráfica para métricas globales como el tiempo de respuesta [17].

FogTorchπ: herramienta de simulación desarrollada en Java, que implementa el método de Monte Carlo para el desarrollo de variaciones en los enlaces de comunicación, y obtener las probabilidades de asignación de perfiles de QoS. Las métricas medidas son la garantía de QoS y el consumo de recursos de niebla, que representan en sus resultados el porcentaje de ejecuciones, el promedio agregado de RAM y del almacenamiento consumido [18].

EdgeCloudSim: simulador de modelamiento de redes desarrollado en Java, que permite realizar un modelamiento computacional mediante la ejecución de tareas y programación de VM, el modelado en la niebla se enfoca en la movilidad, orquestación y descarga. Implementa un entorno modular cuya configuración es administrada con 3 archivos, parámetros de simulación, un archivo XML (Lenguaje de marcado extensible) y la topología del servidor. Una de sus características primordiales, es que permite la definición de movilidad de enlaces de red para la medición de diferentes aspectos de entornos nube/niebla para la evaluación de servicios IoT [19].

Cupcarbon: simulador de redes de sensores inalámbricos para Smart City e Internet de las cosas (IoT), caracterizado por su implementación en dos entornos de simulación, en donde uno de sus entornos se emplea para la simulación de móviles (objetos voladores y vehículos), y su segundo entorno para la simulación de eventos discretos en redes de sensores inalámbricos empleado en base al primer entorno. Permite el diseño y prototipado directo de sensores en el mapa con OSM (OpenStreetMap), para la capa de aplicación de los nodos. Su programación y configuración de cada nodo sensor es desarrollado en un script llamado SenScript que son programados en Python [20].

En la tabla 2.2 se muestra las comparativas entre varias herramientas de simulación planteadas anteriormente, las cuales se definen como derivaciones de Cloudsim a excepción de EmuFog y YAFS.

Características	Simuladores					
	iFogSim	CloudSim SDN	YAFS	EmuFog	FogTorchπ	Edge CloudSim
Lenguaje	Java	Java	Python	Java	Java	Java
Topología	Arbol	Arbitrario	Arbitrario	Arbitrario	Arbitrario	Arbitrario
Inyección de fallas	No	No	Si	Si	No	No
Modelo de aplicación	Modular	Cadena de Servicios	Modular	Basado en Docker	Modular	Modular
Mobilidad	No	No	Customizada por el usuario.	No	No	Customizada por el usuario.
Modelo de costo y energía	Si	Si	Si	Parcial	Si	Parcial
Federación y escalabilidad	Si	Si	Si	Parcial	No	Parcial

Tabla 2.2: Comparación técnica de simuladores Cloud/Fog [21].

En el caso de Ifogsim, se la considera una de las herramientas de simulación nube/niebla más utilizadas debido a la amplia gama de librerías y funciones que ofrecen al usuario para que este desarrolle su propio simulador en base a esta herramienta, debido a que ofrece un mayor soporte para la simulación de algoritmos y mecanismos de tiempo de ejecución y consumo de recursos (CPU y memoria).

2.5 Limitaciones

En la bibliografía revisada se identificaron las siguientes limitaciones:

- No presentan un enfoque general de las arquitecturas, sino un enfoque de casos de estudio en donde predominan:
 1. Vista general de las consideraciones a tomar en la arquitectura para el manejo de recursos, todo orientado más a Smart City en donde nos muestra todas las formas de manejo de recursos posibles.
 2. Vista del modelo F2C utilizando servicios paralelos y distribuidos teniendo una vista del desempeño de cada uno comparándolos en términos de delay y procesamiento utilizando Tareador el cual es una herramienta usada para estrategias de paralelización.
- Únicamente implementan las funciones propias de las herramientas de simulación de código abierto para el desarrollo de los proyectos, pero no buscan generar una combinación de soluciones propias implementando un mayor nivel de programación.
- Usan programas externos para la visualización de las gráficas obtenidas a partir de los resultados generados por las herramientas de simulación.

Referente a las limitaciones encontradas para las herramientas de simulación de código abierto:

- La mayoría de las herramientas de simulación son variaciones de Cloudsim, es decir implementan sus funcionalidades y usan el mismo código, por lo que implementar nuevos comportamientos o modelos de simulación puede resultar complejo debido a la extensión del código a implementar junto con las funcionalidades de Cloudsim.
- EmuFog es una herramienta de emulación y no de simulación, por lo que no se puede aplicar en ambientes reales.

- YAFS es compatible solo con versión de Python 2.7, pero no con las versiones actuales, además de que no existe mucha documentación relevante sobre la aplicación de esta herramienta de simulación.

2.6 Estructura del Modelo de Simulación a implementar

La herramienta seleccionada para la simulación es Ifogsim, debido a que brinda la implementación de entornos en la nube y en la niebla, con un modelo de capas que se encarga de una tarea en específico. Las principales ventajas que nos otorga, es el modelamiento de la administración de recursos, evaluación de recursos de red, monitoreo del consumo energético y disponibilidad de los recursos.

El escenario planteado en la figura 2.8 se enfoca en la correcta toma de decisiones de una empresa para realizar la administración correcta de los recursos o cambios necesario para la optimización de los servicios dentro de su compañía. Se basa en un gerente de múltiples empresas las cuales se encargan de la manufactura de varios componentes los cuales utilizan sensores y actuadores, que se encargan de recopilar toda la información de las maquinarias industriales y posteriormente realizar el envío, almacenamiento, procesamiento e interacción con los dispositivos finales (actuadores). Sin embargo, este proceso involucra un tiempo de respuesta más extendido, por lo que el principal objetivo es buscar una manera de disminuir ese tiempo de respuesta implementando distintos modelos, ya sea usando componentes en la nube, en la niebla o implementando un modelo híbrido, para lo cual se implementaría un simulador para observar la asignación de recursos, permitiendo al usuario el análisis de métricas de rendimiento.

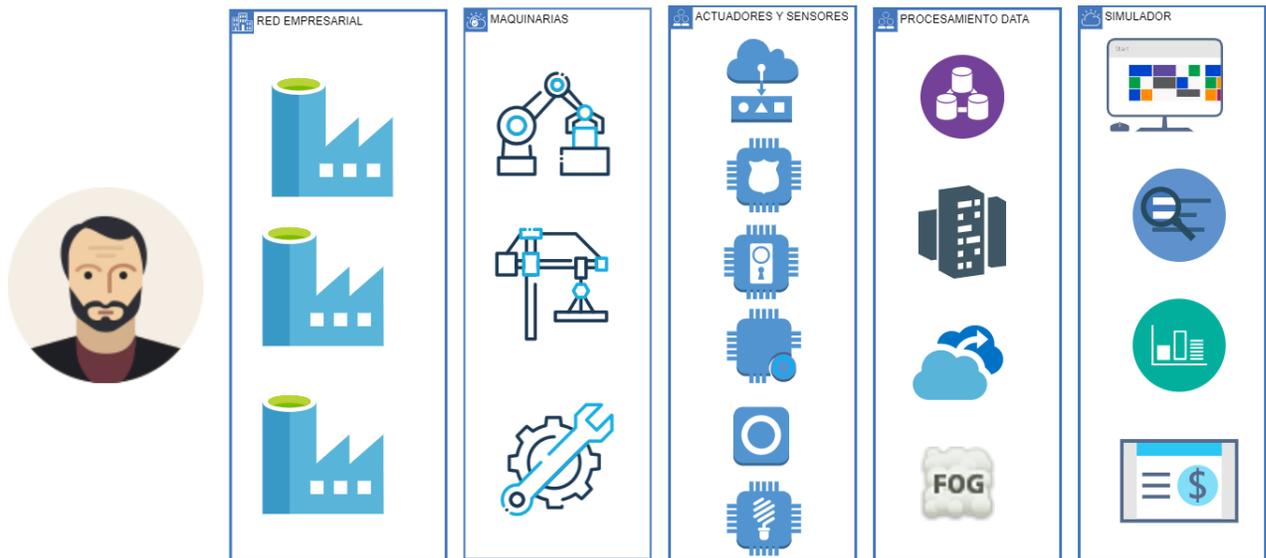


Figura 2.8: Escenario principal a resolver

El esquema de manejo de la herramienta para este escenario se basa en la implementación de Ifogsim, un simulador de código abierto basado en Java. Se desarrollará un ejecutable con una interfaz amigable para el usuario en donde este podrá ingresar los parámetros de simulación de la nube y la niebla, además de definir la cantidad de dispositivos a simular, y de esta forma poder generar modelos de prueba diversos para observar la gestión de recursos mediante resultados gráficos que representan la energía consumida, delay, uso del CPU y la red lo cual se encuentra detallado en la figura 2.9.

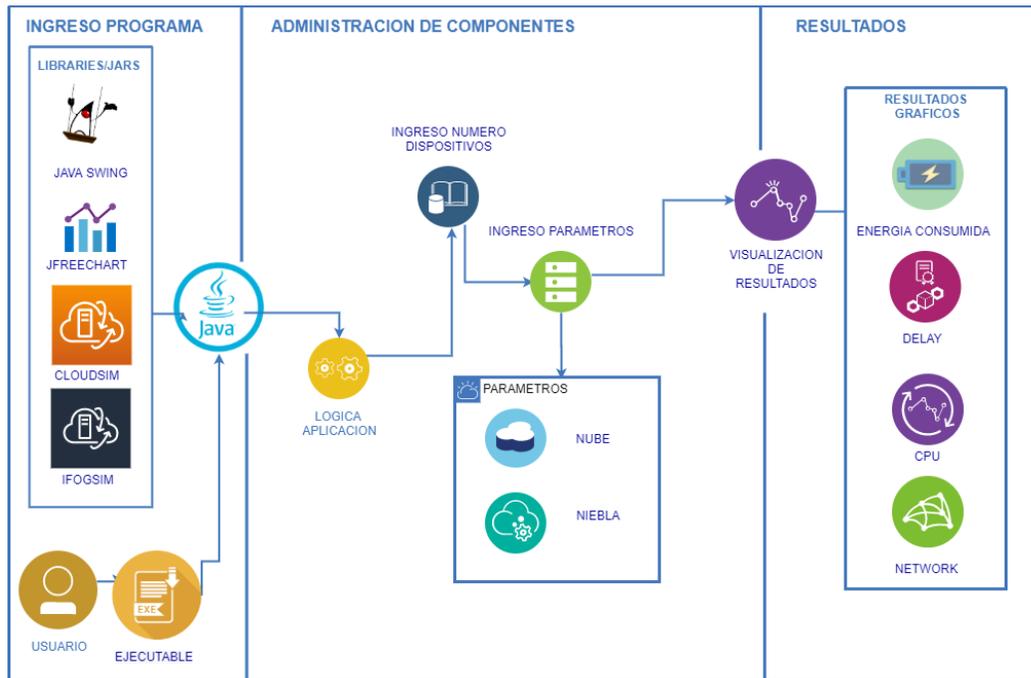


Figura 2.9: Esquema de simulación

CAPITULO 3

3 ANALISIS DE RESULTADOS

3.1 Algoritmos de administración de recursos

Para evaluar el comportamiento del desempeño de diferentes estrategias de asignación de recursos, este trabajo ha considerado tres algoritmos diferentes. Los algoritmos seleccionados son: Edge-Ward (Hibrido), Cloud-only (Centralizado) y Mapping (Distribuido). A continuación, haremos una breve explicación de cada uno de ellos.

3.1.1 Algoritmo Hibrido (Edge-Ward)

El algoritmo es implementado para una estrategia Edge, en caso de que un nodo de la niebla no logre cumplir con los requisitos de una aplicación (insuficiente capacidad computacional), el algoritmo se encarga de seleccionar dispositivos adicionales en la capa de la niebla dentro de la jerarquía de la topología de la red que posea los requisitos solicitados. Las solicitudes son atendidas en orden de llegada, dependiendo de la disponibilidad de recursos informáticos en cada nivel jerárquico, en caso de que ningún dispositivo en la niebla cumpla con los requisitos, este es programado en la nube. Este algoritmo permite un mejor rendimiento para la planificación de recursos de CPU de los dispositivos en la niebla.

3.1.2 Algoritmo Centralizado (Only-Cloud)

Utiliza la implementación tradicional en la nube en la cual la captura de datos realizada por los sensores se envía hacia los centros de datos, se procesa y finalmente se envía de regreso a los actuadores.

3.1.3 Algoritmo Distribuido (Mapping)

En este algoritmo las solicitudes se asignan a los dispositivos en la niebla, sin importar sus capacidades y requisitos, en caso de que esta sea insuficiente se forma una cola de procesamiento en el nodo para poder atender los requisitos de la aplicación.

3.2 Simulador Cloud-Fog Simulator

El simulador tiene una interfaz que permite al usuario realizar tres algoritmos diferentes antes mencionados para una arquitectura específica. Una vez que la simulación termina, muestra una ventana con los resultados de estos tres métodos en donde podremos ver las distintas métricas las cuales son: energía consumida, latencia promedio, tiempo de ejecución del algoritmo y uso de la red. A continuación, se procederá a realizar una descripción de las ventanas y funcionalidades principales del simulador.

3.2.1 Ventana Principal

En esta vista se podrá configurar el número de dispositivos en cada capa (nube y niebla), en la tabla 3.1 se describirá los parámetros de la ventana principal que se visualiza en la imagen 3.1.

Nombre de parámetro	Descripción
Dispositivos en la nube	Selecciona la cantidad de dispositivos en la nube a simular.
Rango de dispositivos en la niebla	Define el rango de dispositivos en la niebla a simular, con un rango desde 1 a 12.
Configuración general de dispositivos en la niebla	Define si la configuración se realizara de manera individual, o de forma generalizada en todos los dispositivos en la niebla.
Numero de dispositivos por equipo	Representa el número de dispositivos por cada nodo en la niebla.
Tipo de grafico	Selección de representación de resultados en formato de grafico de barras o grafico de líneas.

Tabla 3.1 : Descripción de parámetros de ventana principal del simulador.

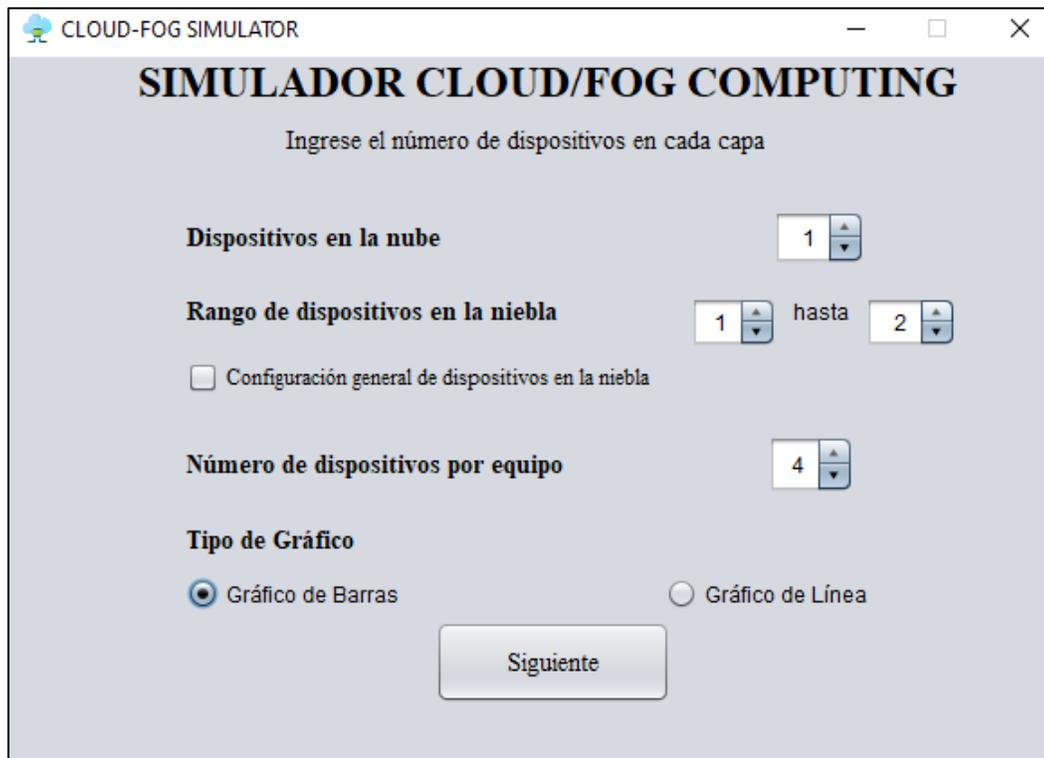


Figura 3.1: Ventana principal del simulador Cloud/Fog Computing

3.2.2 Ventana de configuración de la nube y niebla

En la imagen 3.2 y 3.3 se observa la ventana de configuración de la nube y la niebla, donde se definen los siguientes parámetros: nombre, ancho de banda de subida, ancho de banda de bajada, millón de Instrucciones por segundo (MIPS) y cantidad de memoria. Los cuáles serán descritos para conocimiento del lector en la tabla 3.2.

Nombre de parámetro	Descripción
Nombre de nodo	Se determina el nombre del nivel jerárquico de la capa, en este caso es la capa de la nube y en los casos de los nodos de la niebla se determina el nombre del nivel del dispositivo de niebla.
Ancho de banda de bajada (Downlink BW)	Se refiere a la velocidad de bajada de los datos.
Ancho de banda de bajada (Uplink BW)	Se refiere a la velocidad de subida de los datos.

MIPS	Sus siglas significan millones de instrucciones por segundo, se encargan de medir la potencia de procesamiento y el rendimiento del CPU.
Cantidad de memoria (RAM)	Indica la cantidad de memoria principal del dispositivo en donde se guardará los programas y datos.

Tabla 3.2: Descripción de parámetros de ventana de configuración en la nube y la niebla.

Cabe recalcar que los valores ingresados en este momento son de prueba, al momento de usar el simulador se deberá obtener estos datos del equipo técnico encargado de manejar la estructura en la nube o en la niebla según el caso debido a que estos parámetros dependen del tipo de procesador y la velocidad del reloj.

The screenshot shows a configuration window for a cloud node. The title bar reads 'CLOUD-FOG SIMULATOR'. The main title is 'CONFIGURACIÓN DE NODOS EN LA NUBE' and the subtitle is 'Nodo Cloud'. The configuration parameters are as follows:

Parámetro	Valor
Nombre	cloud
Downlink BW	10000
Uplink BW	100
MIPS	44800
RAM	40000

A 'Siguiete' button is located at the bottom center of the window.

Figura 3.2: Configuración de nodos en la nube

En la siguiente imagen se realiza las configuraciones de los dispositivos en la niebla, los cuales pueden ser configurados todos a la vez o individualmente dependiendo de las características de cada dispositivo, tomando en cuenta que el caso de la nube el equipo

técnico deberá suministrar la información requerida y para el caso de la niebla se cuenta con opciones de memoria RAM.

Nombre	<input type="text" value="fog"/>
Downlink BW	<input type="text" value="10000"/>
Uplink BW	<input type="text" value="10000"/>
MIPS	<input type="text" value="2800"/>
RAM	<input type="text" value="4000"/>

Figura 3.3: Configuración de nodos en la niebla.

3.2.3 Ventana de resultados

Una vez acaben las simulaciones se mostrará la ventana de resultados como se observa en la Figura 3.4 en donde podremos ver la comparativa de la cantidad de dispositivos y las distintas métricas las cuales son: energía consumida, latencia promedio, tiempo de ejecución del algoritmo y uso de la red. En la tabla 3.3 se realizará una descripción de las métricas reflejadas en las ventanas de resultados.

Métricas	Descripción
Energía Consumida	Representa la sumatoria del consumo energético de todos los dispositivos en la capa de la nube y la niebla que intervienen en la simulación, su unidad de medida es representada en mega joule.
Latencia Promedio	Tiempo promedio de ejecución de instrucciones desde su fuente hasta el actuador pasando por el procesamiento de los dispositivos en la nube y la niebla, su unidad de medida es mili segundos.
Tiempo de ejecución del algoritmo	Tiempo que se demora en realizar la simulación de cada enfoque, ya sea centralizado, distribuido o híbrido. Su unidad de medida es mili segundos.
Uso de la red	Representa la cantidad de uso de la red en la aplicación, su unidad de medida es kilo bytes.

Tabla 3.3: Descripción de métricas de resultados

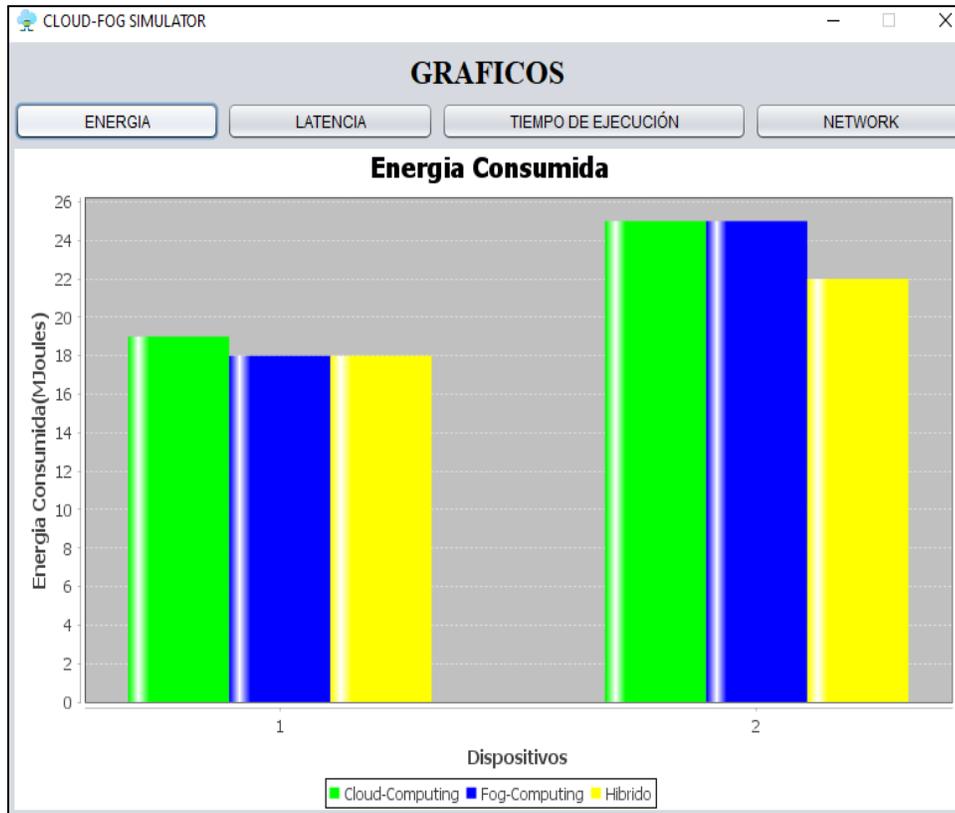


Figura 3.4: Ventana de resultados de la simulación.

3.3 Casos de estudio

El primer escenario de simulación es definiendo el rango de 1 a 2 dispositivos en la niebla y representándolos por gráficos de barra. Y el segundo escenario de simulación se representará con un rango de 1 a 8 dispositivos en la niebla y representándolos por grafico de líneas.

3.3.1 Caso 1: Comparativa de dos dispositivos

Se implementó el escenario de 1 a 2 dispositivos, debido a que permite el estudio de un ambiente de prueba para el desarrollo de un proyecto personal del usuario o el emprendimiento de un proyecto a desarrollar a futuro, en el cual el usuario puede obtener métricas de eficiencia de la topología y compararlos con los distintos enfoques que puede implementar. En la figura 3.5 se puede evidenciar la infraestructura implementada para el caso de 2 dispositivos, en donde se determina dos nodos en la niebla, y en cada uno de estos nodos se determina 2 dispositivos (sensor o actuador) que se encargan de simular el comportamiento en un ambiente real.

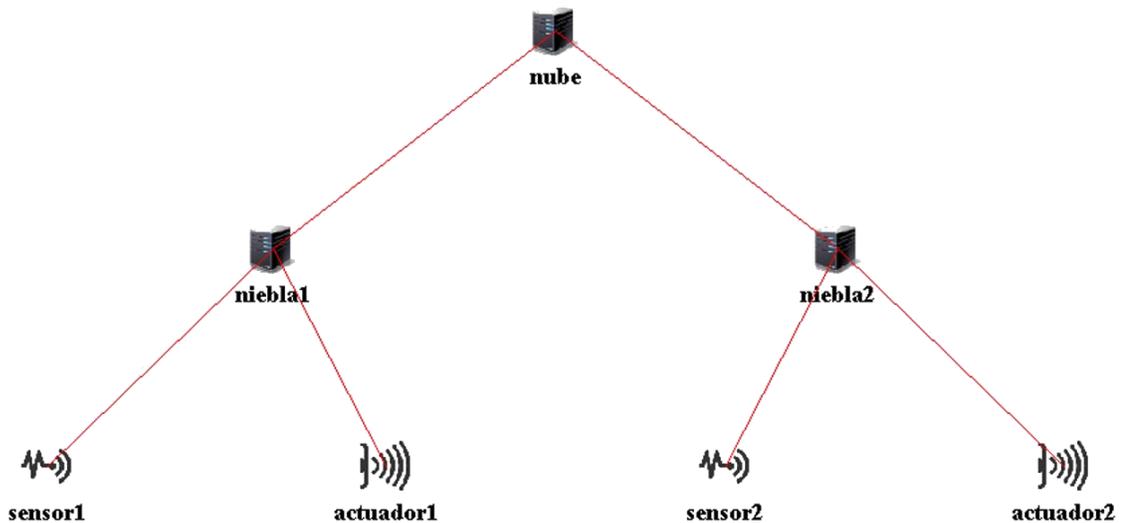


Figura 3.5: Topología usando 2 dispositivos

Una vez realizada la simulación se puede observar que cuando se aumenta la cantidad de dispositivos a utilizar el consumo de energía aumenta de igual forma, sin embargo, el

algoritmo que demuestra menos consumo es aquel que usa el enfoque híbrido como se observa en la Figura 3.6 esto es debido a que el procesamiento se realiza más cerca del borde de la red y de ser necesario se utilizan los otros nodos, por lo que estos pueden permanecer en un estado suspendido hasta que se requiera de su capacidad de procesamiento.

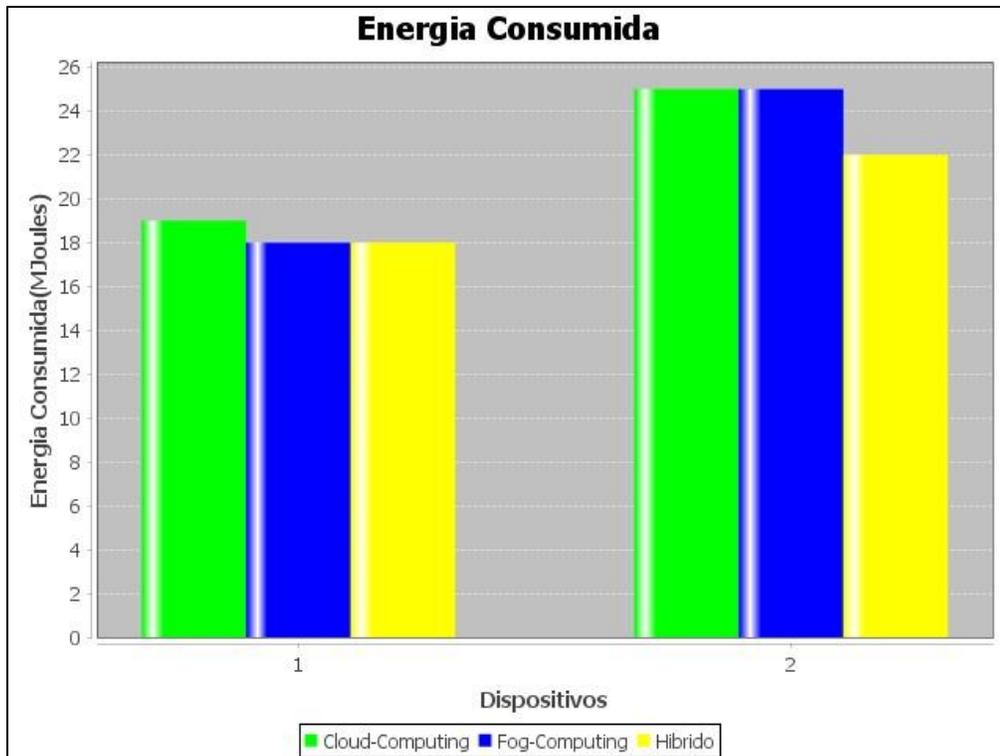


Figura 3.6: Energía consumida con respecto a la comparativa de dos dispositivos.

Con respecto a la latencia promedio podemos observar en la Figura 3.7 que el enfoque de la niebla es mayor al de la nube cuando solo hay un dispositivo, esto se debe a que este algoritmo en particular utiliza colas cuando el poder de procesamiento no es suficiente y para este caso se ejecutaron alrededor de 10140 instrucciones, además podemos ver un decrecimiento significativo cuando se trata del enfoque híbrido por su capacidad de utilizar tanto la nube como la niebla.

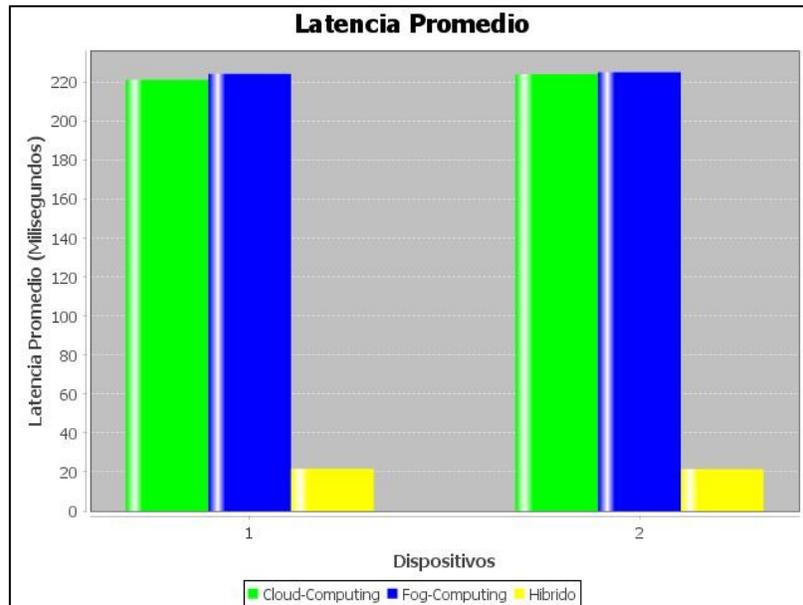


Figura 3.7: Latencia promedio con 1 y 2 dispositivos.

El tiempo de ejecución tiene una relación directa con el número de dispositivos a utilizar ya que mientras más dispositivos haya, el tiempo puede extenderse o reducirse dependiendo del algoritmo utilizado, como se observa en la figura 3.8 en el caso de cloud computing se evidencia un incremento en el tiempo de ejecución con un mayor número de dispositivos, a diferencia de fog computing e híbrido en donde se reduce el tiempo de ejecución, debido a la distribución de la capacidad computacional entre los nodos.

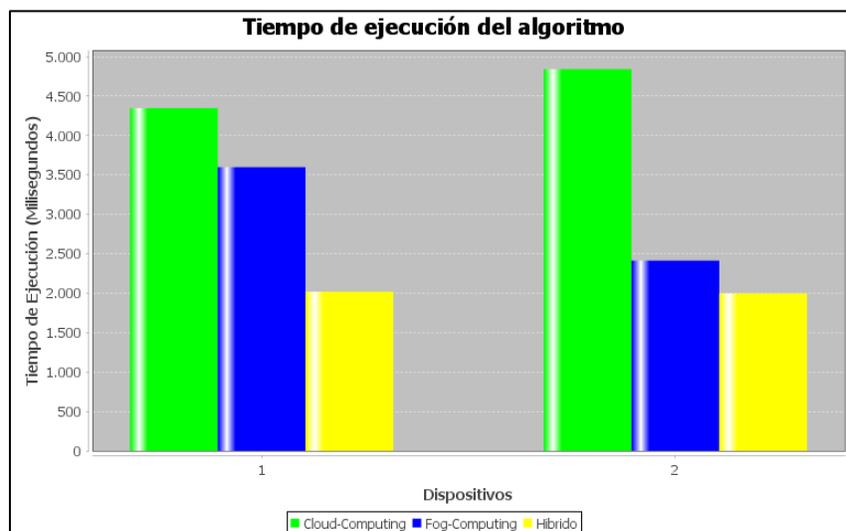


Figura 3.8: Tiempo de ejecución del algoritmo con 1 y 2 dispositivos.

En cuanto al uso de la red, se observa en la figura 3.9 se determina que el uso de la red es la latencia por capacidad de la red usada por la aplicación, en el caso de fog e hibrido es mayor el uso de la red, debido a que implementa un mayor uso de nodos en la niebla, a diferencia de cloud, que únicamente realiza el procesamiento en la nube.

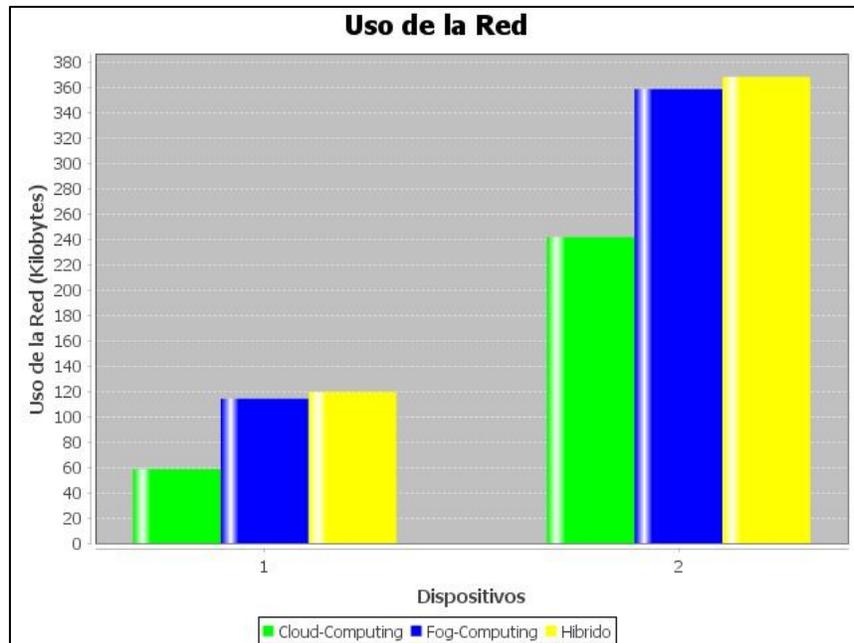


Figura 3.9: Uso de la red con 1 y 2 dispositivos.

3.3.2 Caso 2: Comparativa de 1 a 8 dispositivos

Se desarrolló una comparativa de 8 dispositivos, que es aplicable a empresas medianas o grandes, en donde sus estaciones de trabajo están localizadas en distintas áreas, que se encargan de la comunicación de varias redes con una más grande, logrando establecer el estudio eficiente de las métricas que se establecen para el desarrollo de inversión o cambio en la infraestructura.

Como se evidencia en la figura 3.10, se puede observar una topología de árbol, la cual implementa un modelo jerárquico en donde el nivel de la nube es el nodo padre, y los dispositivos de la niebla son los nodos hijos de la nube, que se encuentran conectados punto a punto, y de esta forma permitirá la comunicación con el nodo contiguo del nivel inferior.

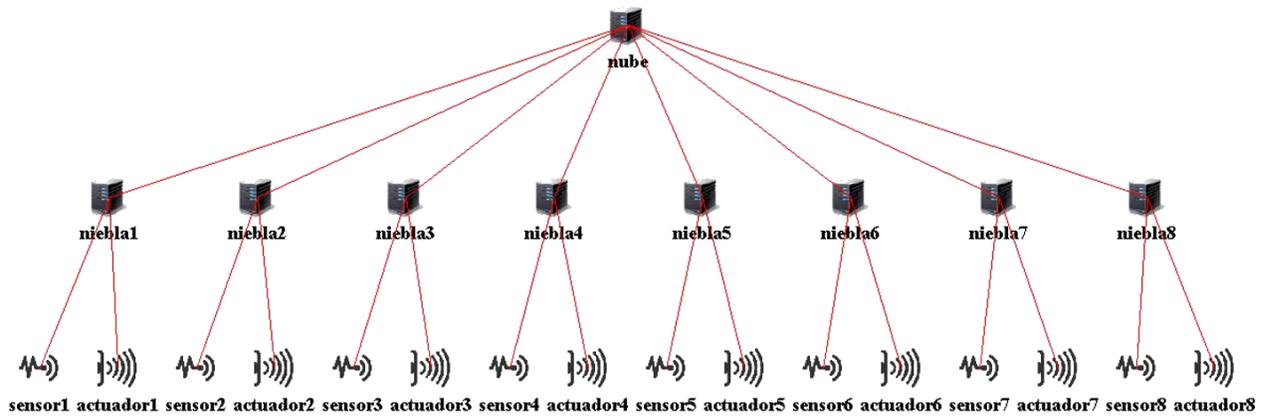


Figura 3.10: Topología usando 8 dispositivos

Este tipo de topología permite una amplia escalabilidad, debido a que se puede agregar nuevos nodos fácilmente implementando un concentrador de red, permitiendo un amplio acceso a los dispositivos dentro de la red facilitando la detección de errores para su inmediata corrección.

Se realizó la ejecución de la simulación con una configuración generalizada para un mayor número de dispositivos y seleccionando el gráfico de línea, luego procedimos a ingresar las respectivas configuraciones, noten que los parámetros de la niebla solo se los ingreso una vez, en este caso se esperó un poco más de tiempo en la simulación debido a que son más dispositivos que crear y por cada uno se crea una máquina virtual. Hay que tomar en cuenta que se puede realizar una simulación con un mayor número de nodos, pero el tiempo de espera para obtener los resultados sería mayor, además de que la simulación de un mayor número de nodos depende de la capacidad computacional del equipo en donde se esté ejecutando el simulador.

En la figura 3.11 se observó una diferencia marcada en los tres algoritmos utilizados siendo la nube el algoritmo only cloud, la niebla el algoritmo Mapping y el híbrido el algoritmo Edge Ward. Para la energía consumida se observa que para el caso de fog la energía consumida es mayor debido a la cola generada cuando la capacidad de procesamiento es insuficiente

y el estado del dispositivo se queda activo y en el caso de cloud es mayor también debido a que todo el procesamiento se está centralizando en la capa de la nube. Y para el híbrido es menor el consumo de energía debido a que se distribuye entre la niebla y la nube.

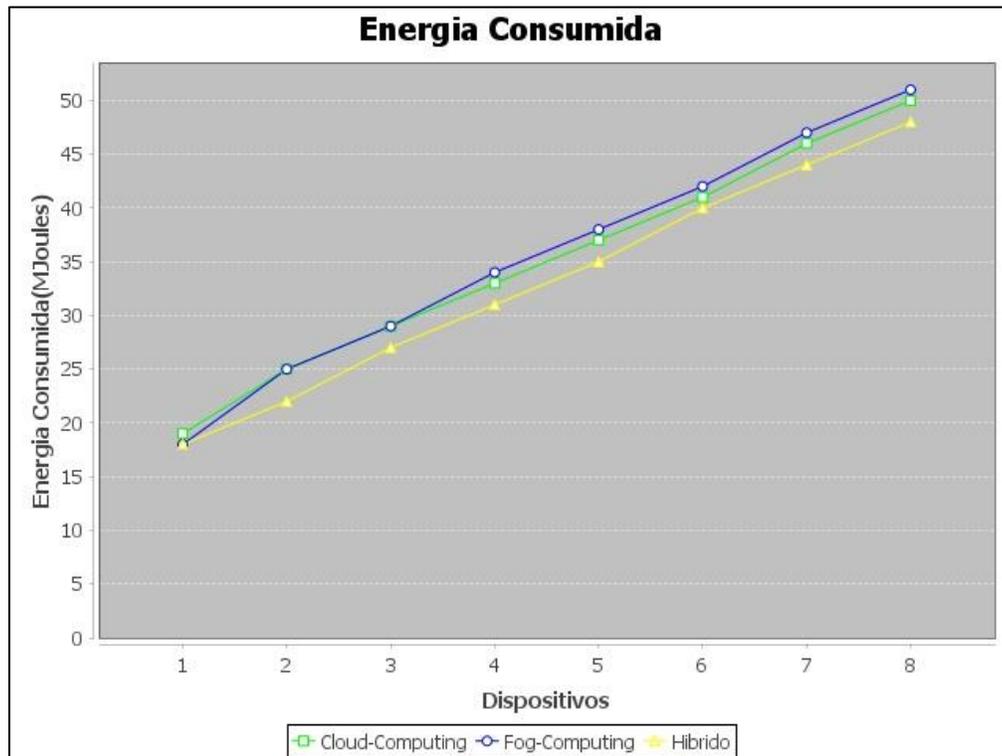


Figura 3.11: Energía consumida con varios dispositivos.

En la figura 3.12 se muestra la latencia promedio que son alrededor de unas 60000 instrucciones, la latencia promedio se mantiene en el algoritmo híbrido mientras que en los otros se dispara, en donde en fog computing tiene problemas de escalabilidad por lo que la latencia debería ser mayor en topologías grandes con un mayor número de transferencia de datos, mientras que en la nube todo el procesamiento se encuentra centralizado, y debe de recorrer una ruta mayor.

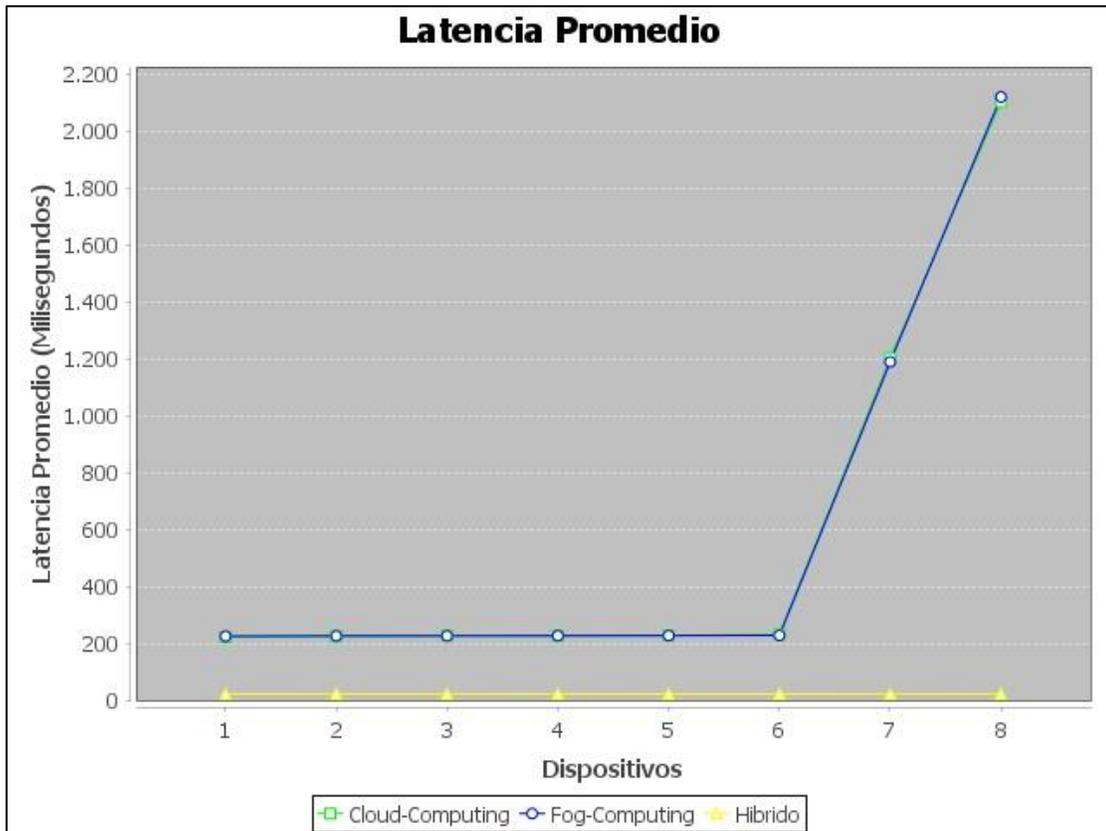


Figura 3.12: Latencia promedio con varios dispositivos.

En la figura 3.13 para el tiempo de ejecución del algoritmo, en el algoritmo Mapping el tiempo de ejecución es menor, esto se debe a que se utilizan colas si no se tiene la capacidad de memoria suficiente en el nodo por lo que este permanecerá más tiempo activo. En cloud es mayor al tiempo de ejecución debido a que se procesa todo e la nube generando una congestión, y en el caso de hibrido es menor el tiempo de ejecución porque se distribuye los procesamientos entre la niebla y la nube.

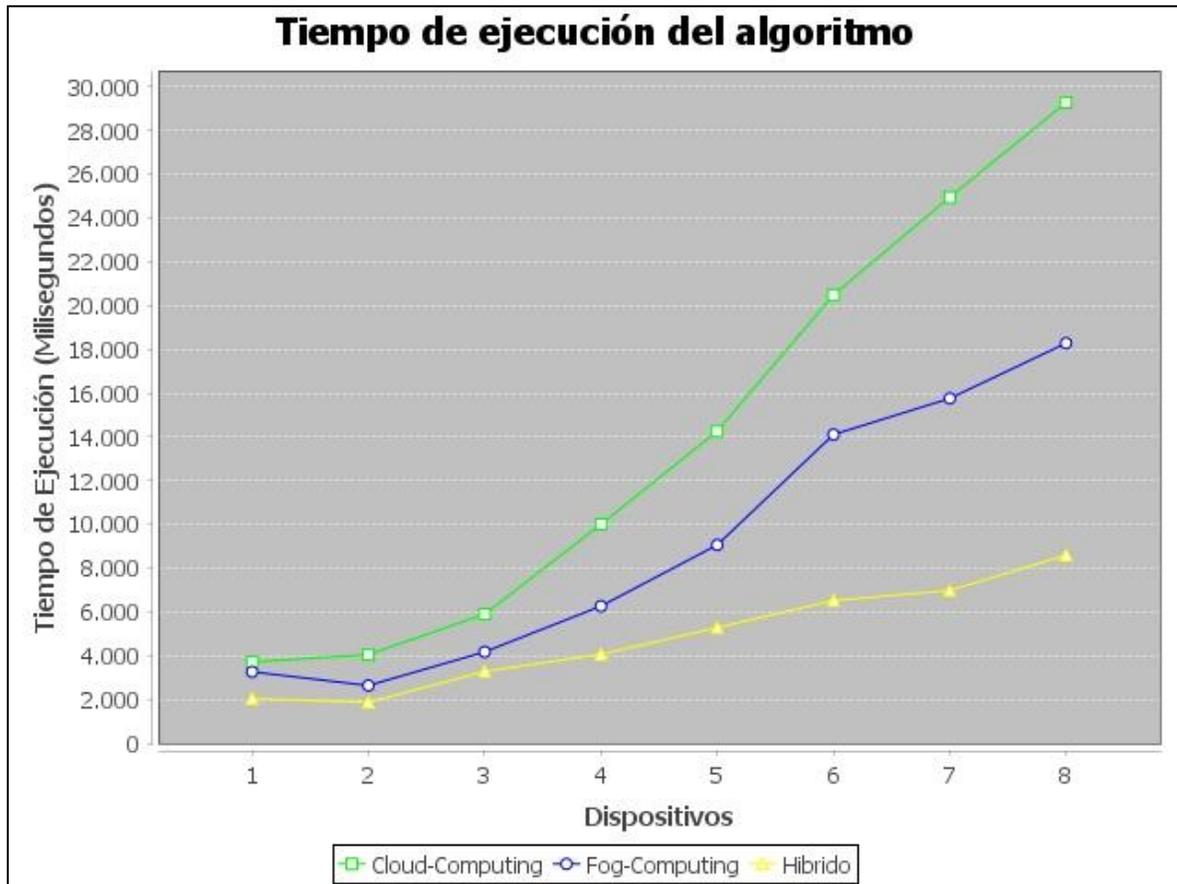


Figura 3.13: Tiempo de ejecución de varios dispositivos.

En la figura 3.14 se observó que a medida que los nodos en la niebla son considerados en el procesamiento, disminuye el uso de la red debido a que el crecimiento descontrolado de la red en modelos basados solo en la nube puede producir congestión de la misma y degradación de las aplicaciones.

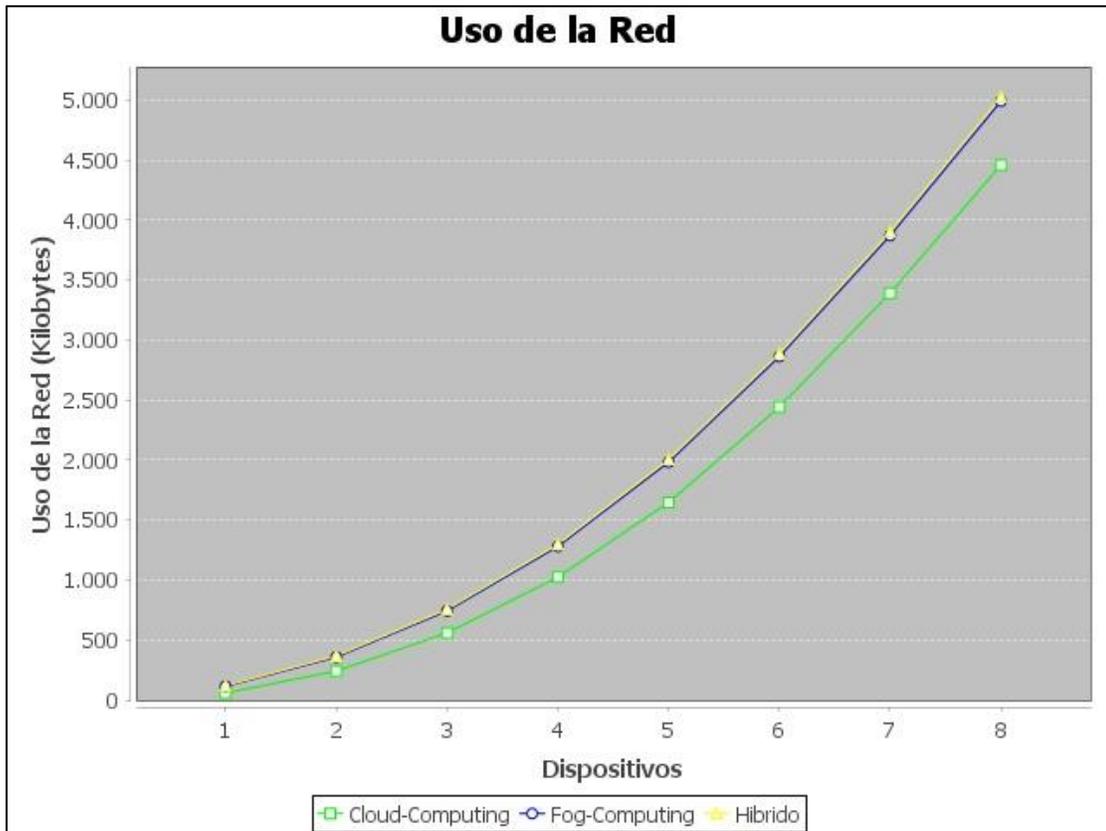


Figura 3.14: Uso de la red de varios dispositivos.

3.4 Análisis de Costos

El análisis de costos para el desarrollo de este trabajo únicamente está enfocado al personal encargado de la programación del simulador, debido a que el lenguaje de programación en el cual está desarrollado el simulador es código abierto y no tiene ningún tipo de restricciones de pago por su uso.

Además de que los modelos de los casos de estudio implementados para las comparativas están enfocados en simulaciones, en donde el usuario administra las características de los equipos a implementar que cumplan con los objetivos de su empresa y de esta forma lograra obtener resultados que sean aplicables para el desarrollo o mejora de una infraestructura dentro de una empresa. Brindando al usuario la libertad de modificar el tipo de modelo a implementar para el desarrollo de la infraestructura.

A continuación, se definirá en la tabla 3.4 los costos por un periodo de tiempo de 4 meses. En donde no se incluye el mantenimiento ni actualización del programa, en caso de que la empresa desee obtener nuevas funcionalidades o análisis de distintas métricas, deben de solicitar el servicio de modificación.

Rol	Número	Costo mensual	Numero de meses	Total
Programador Junior	2	\$1300	4	\$10400
TOTAL				\$10400

Tabla 3.4: Análisis de costos de personal.

CAPITULO 4

4 CONCLUSIONES Y RECOMENDACIONES

En el trabajo desarrollado se evidencio la importancia de una comparativa de distintos enfoques de asignación de recursos para la ayuda de toma de decisiones del tipo de topología y tecnología a implementar, sea la nube, niebla o hibrido.

Se puede obtener métricas tales como energía consumida, tiempo de ejecución de algoritmo, latencia y consumo de red, usando un software abierto como Ifogsim desarrollado en Java, en donde se pueden establecer distintos escenarios a simular. Además de que puedes brindar al usuario una configuración personalizada de los nodos en la nube o en la niebla, estableciendo sus parámetros de forma individual o generalizada según sea el caso por desarrollar.

Cuando se implementan un número mayor de 12 nodos para las simulaciones, no se obtiene una buena apreciación de los resultados gráficos debido a que se apilan las gráficas y no se observa correctamente los cambios en el desempeño de cada algoritmo, en este caso para la simulación de un número mayor de 12 nodos es preferible obtener los resultados únicamente en archivos .csv en donde se podría realizar el análisis posterior de estos resultados en una herramienta grafica la cual permita mostrar más opciones.

El rendimiento del computador en donde se esté realizando la simulación, debido a que, al realizar simulaciones de nodos de más de 20 nodos en computadoras sin muchos recursos computacionales, se empiezan a generar problemas de ejecución y la cantidad de tiempo que toma en mostrar los resultados es muy elevado, debido a que se generan por cada nodo una máquina virtual y esto consume demasiados recursos computacionales ocasionando una interrupción en la operatividad del equipo.

4.1 Conclusiones

1. Se verifico luego de las pruebas realizadas que el enfoque hibrido posee un mejor rendimiento de los resultados obtenidos en energía consumida, tiempo de ejecución del algoritmo, latencia y consumo de red debido a que este enfoque permite realizar una distribución de la carga en la red, en donde los nodos en la niebla se encargaban principalmente del procesamiento de los datos, y en caso de no tener recursos disponibles en este, se realiza un post procesamiento en la nube.
2. Se decidió usar la herramienta de simulación Ifogsim debido a que en las comparativas realizadas se determinó que esta herramienta posee un mayor campo de aplicación, debido a que permite la implementación de los tres algoritmos de ejecución de la nube, niebla e hibrido, a diferencia de otras herramientas de simulación que únicamente permitían realizar un análisis en la de máximo dos enfoques, pero no de los tres enfoques a la vez.
3. Se determinaron tres principales enfoques a utilizar en los cuales se evaluó la asignación de recursos en toda la topología, teniendo como resultados que el enfoque hibrido obtiene un mejor desempeño en comparación con los enfoques de la nube y la niebla, en donde se centraliza todo el procesamiento o se crean colas.
4. Se identificó que en la implementación de las comparativas realizadas los enfoques de la nube y la niebla poseían valores aproximados para los recursos de energía consumida, latencia, pero en cuanto al uso de la red en la nube es menor debido a que la información se procesa de forma centralizada y en cuanto al tiempo de ejecución, este es menor de las tres debido a que se debe realizar todo el proceso de transporte de la información desde y hacia la nube.

4.2 Recomendaciones

1. Implementar una herramienta grafica externa del programa de simulación, para una mejor visualización de los resultados gráficos en el caso de un mayor número de nodos a simular, usando los datos almacenados a partir de un archivo .csv en donde se encuentre almacenado los resultados primordiales de la ejecución de los resultados obtenidos.
2. Establecer una interfaz del desarrollo de la topología generada con la cantidad de nodos en la nube y en la niebla, para poder observar de forma gráfica el tipo de arquitectura que se está simulando.
3. Mantener un historial de las simulaciones realizadas para poder implementar posteriormente estos archivos para realizar un análisis del comportamiento de los recursos asignados según el número de nodos simulados en cada tipo de enfoque.
4. Considerar el uso de otra herramienta de código abierto en caso de ser necesario el uso de nodos con movilidad que permita simular el movimiento en el dispositivo.

BIBLIOGRAFIA

- [1] Grdar esweb., «Gartner: tendencias tecnológicas en auge al 2020,» Julio 2015. [En línea]. Available: <https://www.grdar.com/es/blog/tendencias-tecnologicas-en-auge-al-2020>.
- [2] R. F. El-Gazzar, « A Literature Review on Cloud Computing Adoption Issues in Enterprises,» *IFIP Advances in Information and Communication Technology*, vol. 429, pp. 214-242, 2014.
- [3] A. V. D. K. G. B. Harshit Gupta, «iFogSim: A toolkit for modeling and simulation of resource,» *Software: Practice and Experience*, vol. 47, nº 9, pp. 1275-1296, 2017.
- [4] S. Margariti, V. Dimakopoulos y G. Tsoumanis, «Modeling and Simulation Tools for Fog Computing—A Comprehensive Survey from a Cost Perspective,» *Future Internet*, vol. 12, nº 5, p. 89, 2020.
- [5] M. A. Y. y. D. Schaefer, «Fog Computing as a Complementary Approach to Cloud Computing,» de *2019 International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Arabia Saudita, 2019.
- [6] W. S. Y. S. y. E. C. I. F. Akyildiz, «A survey on sensor networks,» *IEEE Communications Magazine*, vol. 40, nº 8, pp. 102-114, Agosto 2002.
- [7] M. Avram, «Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective,» *Procedia Technology*, vol. 12, nº 7, pp. 529-534, 2014.
- [8] E. M. J. G. S. S. Xavi Masip, «Collaborative Mechanism for Hybrid Fog-Cloud Scenarios,» de *Fog and Fogonomics: Challenges and Practices of Fog Computing, Communication, Networking, Strategy, and Economics*, Hoboken, New Jersey, USA, Wiley, 2020, pp. 7-60.
- [9] Y. Z. S. G. W. X. a. K. Y. R. Yu, «Toward Cloud-Based Vehicular Networks with Efficient Resource Management,» *IEEE Network*, vol. 27, nº 5, pp. 48-55, Septiembre-Octubre 2013.

- [10] B. & C. Z. & H. G. & W. T. & H. H. & Y. Q. ang, «A hierarchical distributed fog computing architecture for big data analysis in smart cities.,» de *ASE International Conference on Big Data*, Kaohsiung, Taiwan, 2015.
- [11] B. V. Cheol-Ho Hong, «Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms,» *ACM Computing Survey*, vol. 52, nº 5, p. 97, 2019.
- [12] E. M.-T. G. T. A. J. y. G. R. X. Masip-Bruin, «Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems,» *IEEE Wireless Communications*, vol. 23, nº 5, pp. 120-128, Noviembre 2016.
- [13] X. M.-B. E. M.-T. S. S.-L. J. G. G. R. A. J. A. y. J. F. V.B. Souza, «Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures,» *Future Generation Computer Systems*, vol. 87, pp. 1-15, Octubre 2018.
- [14] R. R. R. B. A. D. R. C. y. B. R. Calheiros, «CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,» *Software: Practice and Experience*, vol. 41, nº 1, pp. 23-50, Agosto 2010.
- [15] A. V. D. S. K. G. y. R. B. Harshit Gupta, «iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge,» *Software: Practice and Experience*, vol. 47, nº 9, pp. 1275-1296, Junio 2017.
- [16] C. G. a. C. J. I. Lera, «YAFS: A Simulator for IoT Scenarios in Fog Computing,» *IEEE Access*, vol. 7, pp. 91745-91758, 2019.
- [17] L. G. H. G. E. S. a. U. R. R. Mayer, «EmuFog: Extensible and scalable emulation of large-scale fog computing infrastructures,» *2017 IEEE Fog World Congress (FWC)*, pp. 1-6, 2017.
- [18] S. F. A. I. A. Brogi, «Deploying Fog Applications: How Much Does It Cost, By the Way?,» *8th International Conference on Cloud Computing and Services Science*, 2018.
- [19] C. O. A. E. C. Sonmez, «EdgeCloudSim: An environment for performance evaluation of edge computing systems,» *Trans Emerging Tel Tech*, vol. 29, nº 11, pp. 2161-3915, 2018.

- [20] A. Bounceur, «CupCarbon: A New Platform for Designing and Simulating Smart-City and IoT Wireless Sensor Networks (SCI-WSN),» de *Association for Computing Machinery*, New York, USA, 2016.
- [21] K. V. M. C. E. M. David Perez Abreu, «A comparative analysis of simulators for the Cloud to Fog continuum,» *Simulation Modelling Practice and Theory*, vol. 101, 2020.

ANEXOS

ANEXO 1: MANUAL DE INSTALACION

Requerimientos de instalación

Para el proceso de instalación del programa se deben de tomar en cuenta las siguientes indicaciones establecidas para el componente de hardware y software que son primordiales para una correcta ejecución del simulador.

Requerimientos de hardware

En la tabla 1, se definen los componentes mínimos requeridos para la ejecución del simulador, en caso de desear ejecutar con mayor rapidez e implementar en las simulaciones un mayor número de nodos es recomendable usar computadores con una mayor capacidad computacional.

Hardware	
Número de núcleos	4
Procesador	Intel Core i5
Memoria RAM	8 Gb
Disco Duro	1 Tb

Tabla 5. Requerimientos mínimos de hardware

Requerimientos de software

En la tabla 2, se define los componentes de software que son necesarios para la instalación, en donde se deben de cumplir con las versiones establecidas para los programas determinados. En la figura 1, se puede observar en las propiedades del proyecto las distintas librerías que posee, en donde la mayoría viene por defecto en la instalación del programa a diferencia de las que son mencionadas en la tabla 2, las cuales se requiere su respectiva instalación.

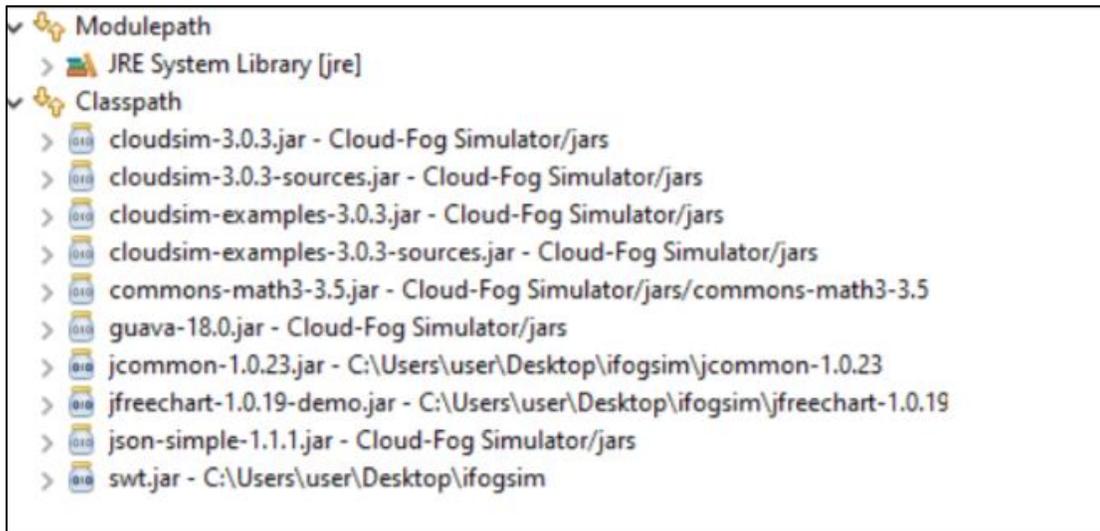


Figura 1. Librerías utilizadas en el programa

Software		
Programas	Nombre	Versión
	Eclipse	2020-09
	Java	8
Librerías	jfreechart	1.0.19
	jcommon	1.0.23
	swt	-
	commons-math	3.5

Tabla 6. Requerimientos de software

Pasos de instalación

1. Descargar el archivo comprimido con el programa y descomprimirlo en una ubicación a elección del usuario

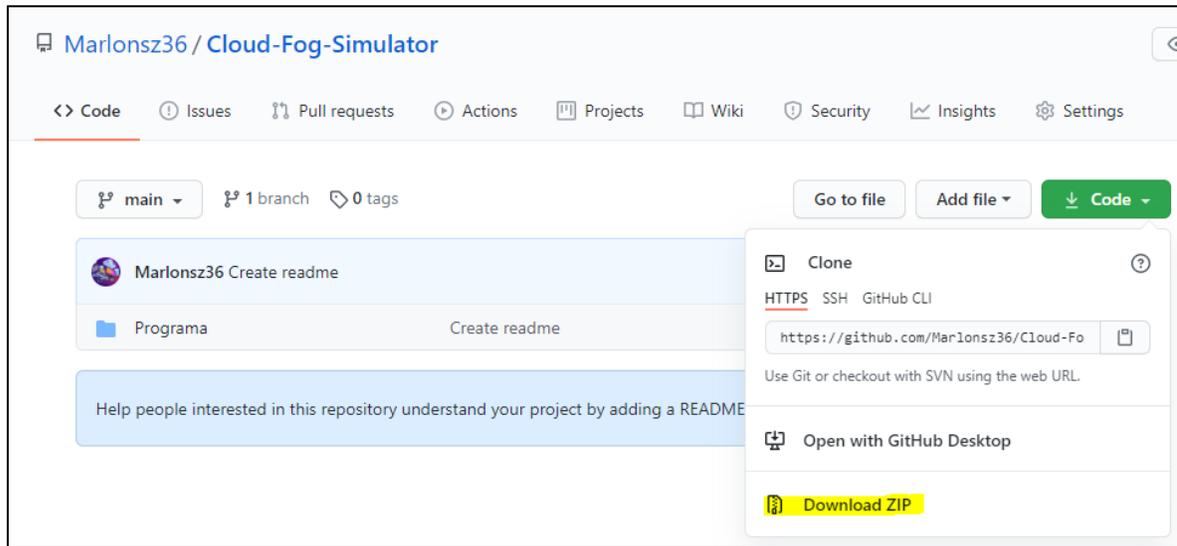


Figura 2. Descarga del simulador desde el repositorio

2. Abrir el programa Eclipse, puede escogerse el espacio de trabajo mostrado a continuación o escoger una a gusto del usuario

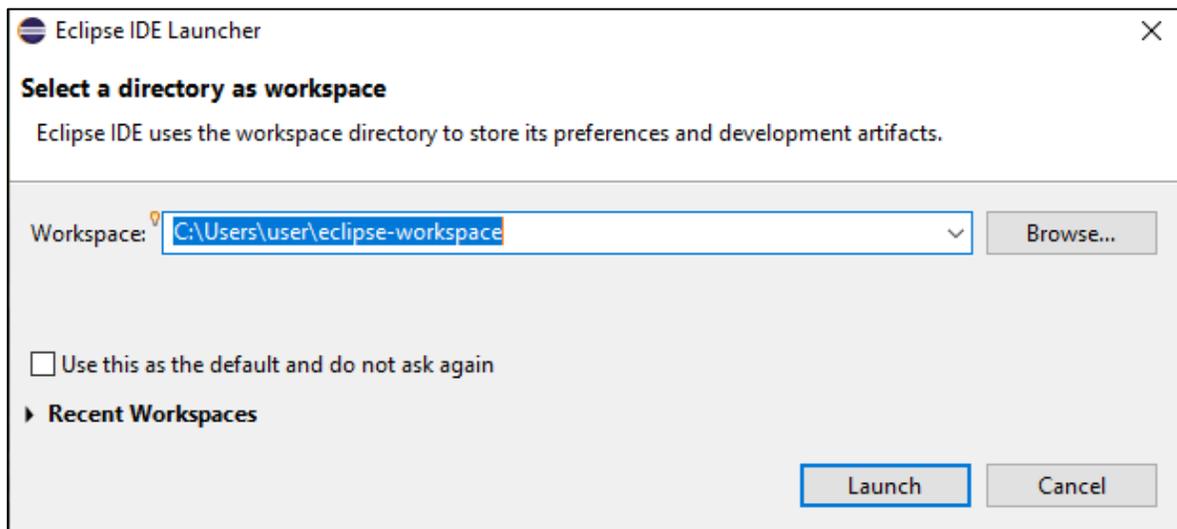


Figura 3. Ruta de almacenamiento del programa

3. Crear un nuevo proyecto en eclipse.

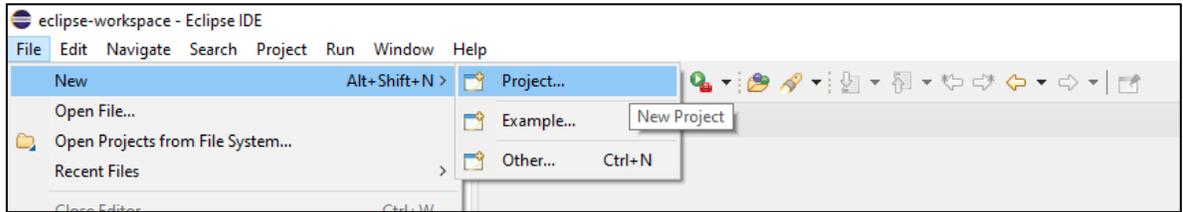


Figura 4. Creación de un nuevo proyecto

4. Asignar un nombre al proyecto, luego seleccionar la ubicación del proyecto descomprimido, y finalmente dar clic en finalizar.

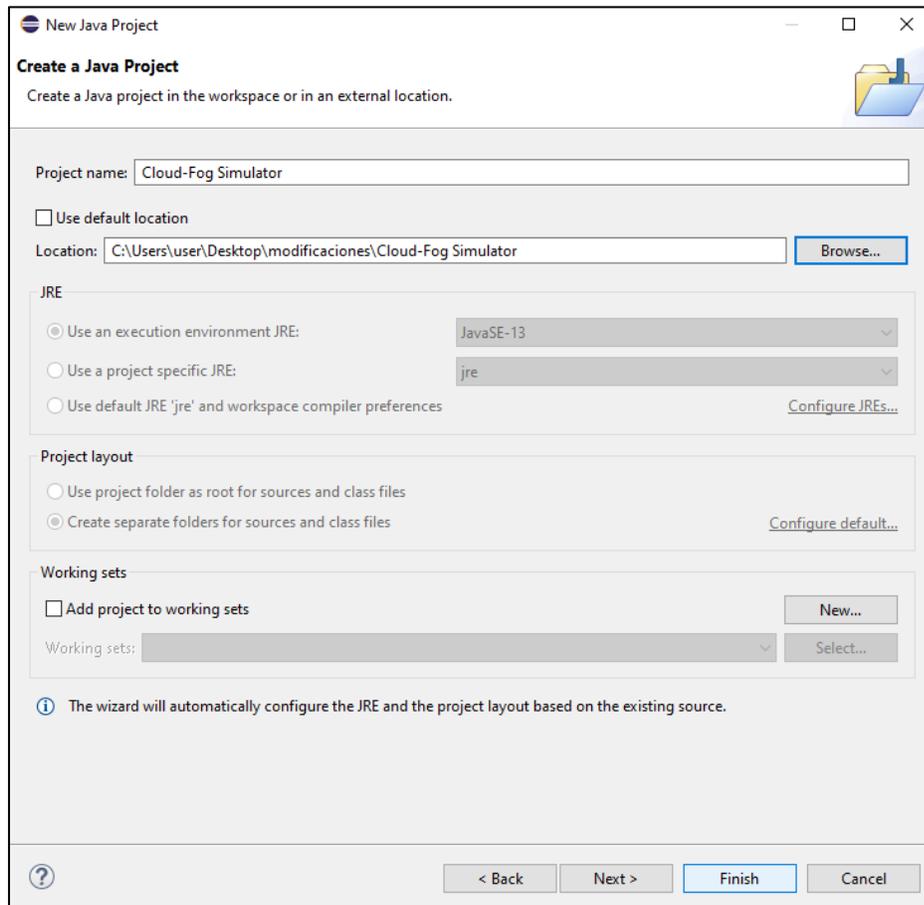


Figura 5. Asignación de nombre del proyecto

5. A continuación, tenemos los archivos de la carpeta listos para explorar y ejecutar el programa. Como se puede observar en la imagen, el proyecto está compuesto por varias carpetas las cuales se tiene la topología, resultados, las salidas del programa y los distintos paquetes que se implementaran.

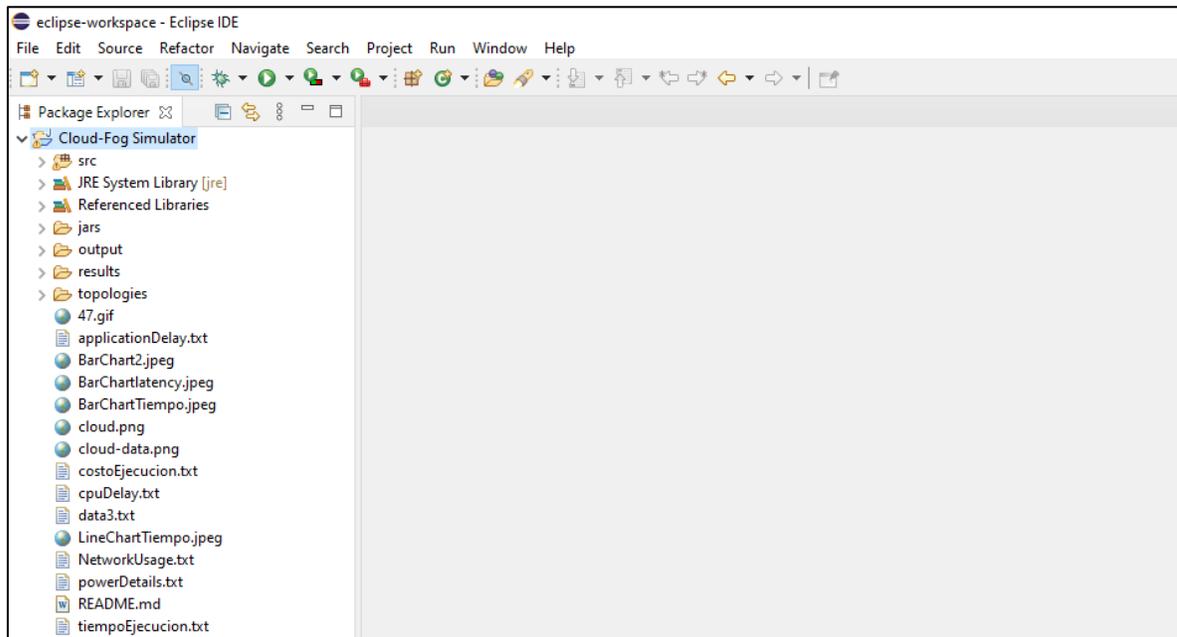


Figura 6. Estructura del proyecto

6. Para poder ejecutar el simulador se debe buscar la carpeta src y buscar el paquete “org.ifogsim.simuation” y ubicarse en la clase “ventana Principal”, damos clic derecho y seleccionamos la opción Run as, Java Application

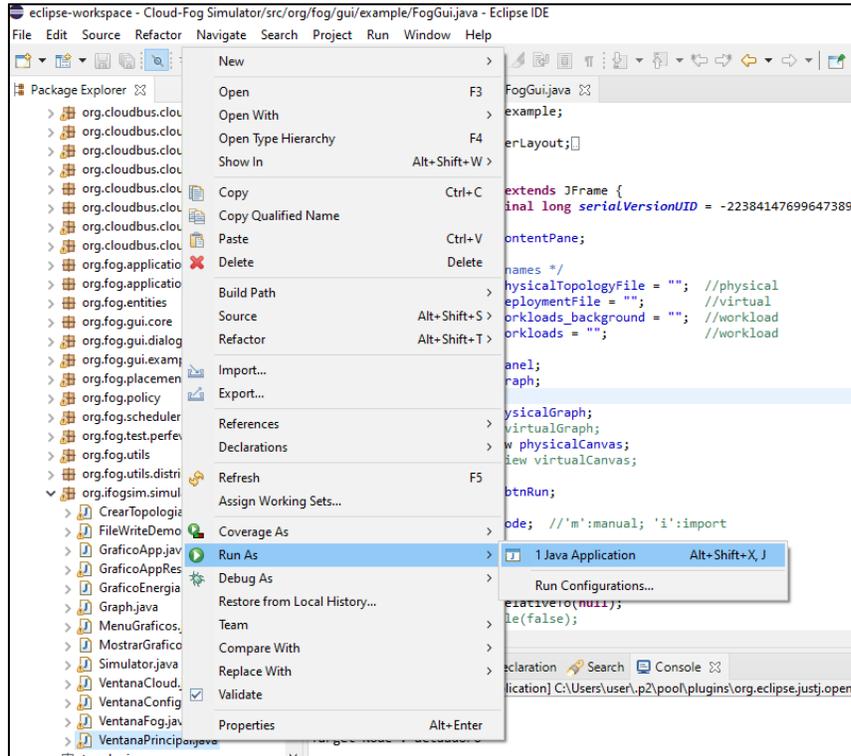


Figura 7. Ejecución del programa

- Finalmente se muestra la ventana principal del simulador en donde el usuario podrá modificar los parámetros necesarios para la simulación.

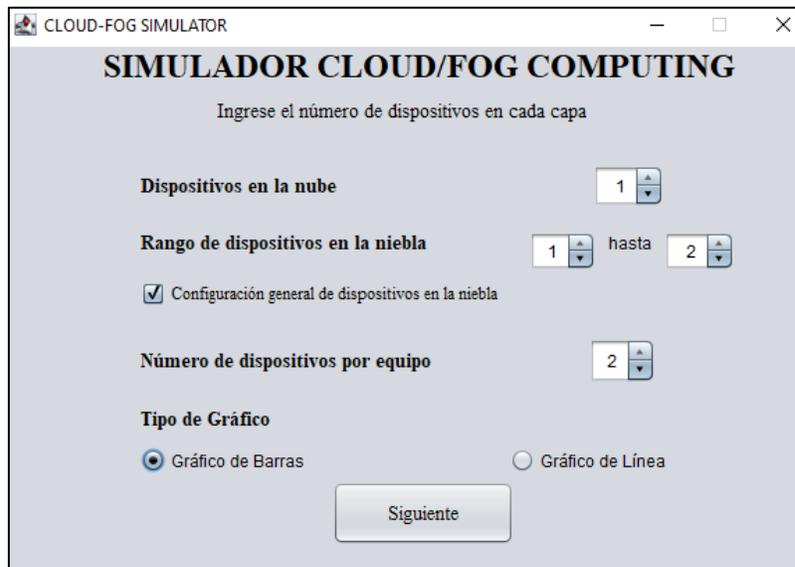


Figura 8. Ventana Principal del simulador

ANEXO 2: MANUAL DE USUARIO

Ejecución del programa

Dar clic en el ejecutable del programa, como se puede observar en la figura 1, el ejecutable del programa es un ejecutable de Java.

powerDetails	29/1/2021 21:54	Documento de te...	1 KB
README.md	12/5/2020 7:47	Archivo MD	2 KB
Simulator	9/2/2021 20:55	Executable Jar File	8 KB
tiempoEjecucion	29/1/2021 21:53	Documento de te...	1 KB

Figura 8. Ejecutable del programa

Ventana principal del simulador

Al iniciar el simulador, aparecerá la ventana principal del programa en donde se deberá detallar los parámetros necesarios para la simulación de las distintas arquitecturas que desea desarrollar el usuario.

CLOUD-FOG SIMULATOR

SIMULADOR CLOUD/FOG COMPUTING

Ingrese el número de dispositivos en cada capa

Dispositivos en la nube 1

Rango de dispositivos en la niebla 1 hasta 2

Configuración general de dispositivos en la niebla

Número de dispositivos por equipo 2

Tipo de Gráfico

Gráfico de Barras Gráfico de Línea

Siguiente

Figura 9. Ventana principal del simulador.

A continuación, en la tabla 1, se determinará una explicación de los parámetros principales de la ventana principal del simulador.

Dispositivos en la nube	Representa el número de dispositivos en la nube que el usuario desea simular.
Rango de dispositivos en la niebla	Representa el rango de dispositivos en la niebla que el usuario desea simular.
Configuración general de dispositivos en la niebla	Determina la opción de implementar una configuración general de todos los dispositivos en la niebla con las mismas características establecidas, en caso de no seleccionar esta opción el usuario determinara distintas características para cada nodo en la niebla.
Numero de dispositivos por equipo	Representa el número de equipos que posee cada nodo en la niebla.
Tipo de grafico	Brinda al usuario la opción de observar los resultados en gráficos de barra o de línea.

Tabla 7. Descripción de parámetros de simulación

Ventana de configuración

Ventana de configuración de nodos en la nube

A continuación, aparecerá la siguiente ventana en donde se deberá detallar los parámetros requeridos del nodo en la nube, se recomienda la ayuda de los responsables del área de TI que son los que proveen la información de las características de los componentes que posee la infraestructura. En la tabla 2, se determinará una explicación de los parámetros determinados en la configuración de los nodos en la nube y en la niebla.

The image shows a software window titled 'CLOUD-FOG SIMULATOR'. Inside, the main heading is 'CONFIGURACIÓN DE NODOS EN LA NUBE' and the sub-heading is 'Nodo Cloud'. There are five text input fields arranged vertically, labeled 'Nombre', 'Downlink BW', 'Uplink BW', 'MIPS', and 'RAM'. At the bottom center, there is a button labeled 'Siguiete'.

Figura 10. Ventana de configuración de nodos en la nube

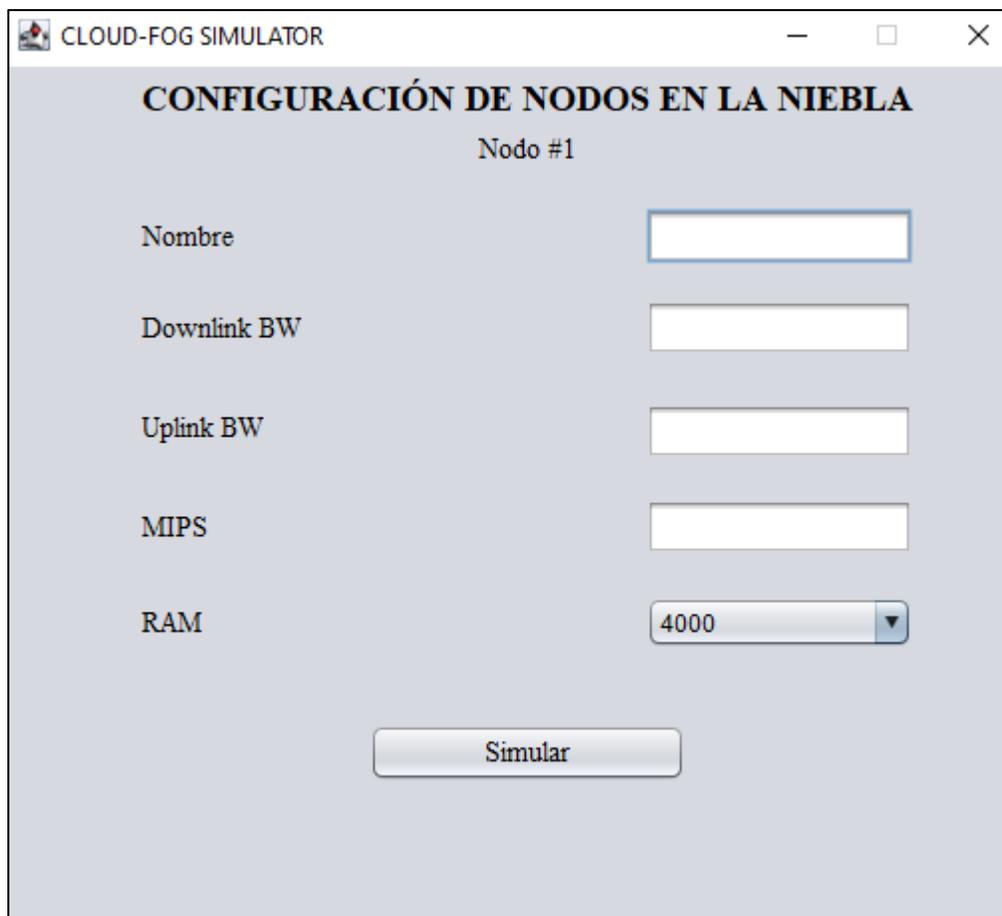
Parámetro	Descripción
Nombre	Representa el nombre del nodo (nube o niebla)
Downlink	Es la velocidad de bajada de la data.
Uplink	Es la velocidad de subida de la data.
MIPS	Representa la potencia y rendimiento del CPU.
RAM	Representa la memoria ram de los dispositivos.

Tabla 8. Descripción de parámetros de nodos en la nube/niebla

Ventana de configuración de nodos en la niebla

A continuación, se deberá introducir toda la información pertinente a cada uno de los nodos que se utilicen para la configuración de los nodos en la niebla, como se determinó en la tabla 2 poseen los mismos parámetros que los definidos en la nube.

Nota: En caso de escoger la opción generalizada solo deberá ingresar los parámetros una sola vez.



The image shows a software window titled "CLOUD-FOG SIMULATOR" with a subtitle "CONFIGURACIÓN DE NODOS EN LA NIEBLA" and "Nodo #1". The window contains five input fields for configuration: "Nombre" (text), "Downlink BW" (text), "Uplink BW" (text), "MIPS" (text), and "RAM" (a dropdown menu currently showing "4000"). A "Simular" button is located at the bottom center of the window.

Figura 11. Ventana de configuración de nodos en la niebla

Ventana de resultados

Una vez damos clic en simular deberemos esperar y posteriormente nos aparecerá la siguiente ventana en donde deberemos dar clic a la métrica de interés para visualizar la gráfica.

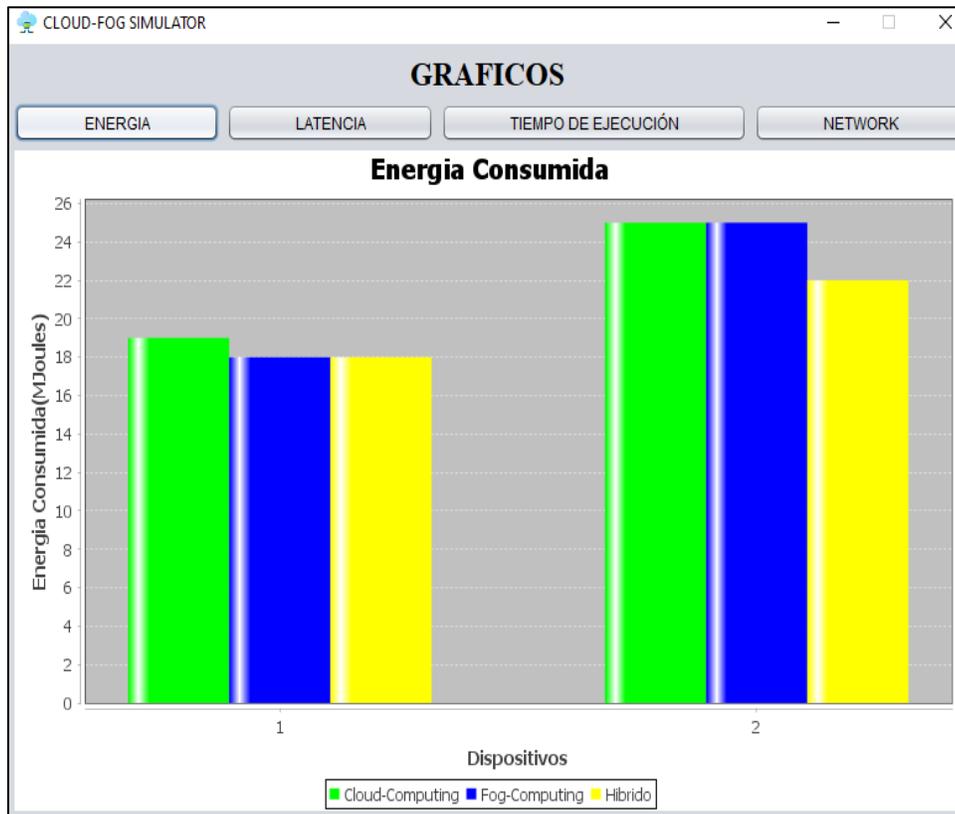


Figura 12. Ventana de resultados