

**Facultad de Ingeniería**

**Escuela de Ingeniería en Telemática**

**Curso**

Inteligencia en Redes de Datos.

**Tema**

CloudSim para un proveedor de infraestructura como servicio

**Estudiante**

Edward Alfaro Murillo

20230113205

**Profesor**

Carlos Andrés Méndez Rodríguez

**Periodo**

3Q, 2024

**Tabla de Contenido**

**Introducción**

* Contexto de la Computación en la Nube
* CloudSim como Herramienta de Simulación

**Justificación**

* Ventajas de la Computación en la Nube
* Importancia del Dimensionamiento en la Nube

**Objetivos**

* Objetivo General
* Objetivos Específicos

**Marco Teórico**

* Definición de Cloud Computing
* IaaS (Infraestructura como Servicio)
* CloudSim: Características y Funcionalidades

**Desarrollo de la Simulación**

* Configuración de CloudSim
* Ejemplos: Simulación de Recursos en un Entorno IaaS
* Configuración de Hosts y VMs
* Creación y Ejecución de Cloudlets

**Desarrollo y Análisis de Resultados**

* Comparación entre Ejemplo 1 y Ejemplo 2
* Análisis del Comportamiento de la Infraestructura IaaS

**Bibliografía**

**Introducción**

En la actualidad, la computación en la nube (Cloud Computing) ha revolucionado la forma en que las organizaciones gestionan sus recursos tecnológicos y sus infraestructuras informáticas. Esta tecnología permite el acceso a recursos compartidos como servidores, almacenamiento y aplicaciones a través de internet, eliminando la necesidad de inversiones significativas en hardware y ofreciendo flexibilidad y escalabilidad. Sin embargo, para evaluar y optimizar las infraestructuras basadas en la nube, es esencial contar con herramientas de simulación que permitan modelar diferentes escenarios sin la necesidad de desplegar físicamente los sistemas. En este contexto, CloudSim emerge como una solución poderosa y eficiente. CloudSim es una plataforma de simulación en Java diseñada para modelar y simular entornos de computación en la nube, permitiendo a los investigadores y desarrolladores probar y evaluar diversas arquitecturas de infraestructura, algoritmos de gestión de recursos y políticas de distribución de carga. A través de CloudSim, es posible estudiar el rendimiento de los recursos en la nube, la asignación de tareas y la gestión del consumo de energía, entre otros aspectos fundamentales para la toma de decisiones en la gestión de infraestructuras de nube. Esta herramienta es fundamental para avanzar en el diseño de soluciones más eficientes y sostenibles en el ámbito de la computación en la nube.

La computación en la nube (Cloud Computing) ha emergido como una de las tecnologías más transformadoras en la era digital, ofreciendo un modelo de entrega de servicios de TI que permite a las organizaciones y usuarios acceder a recursos informáticos a través de internet, sin la necesidad de poseer ni gestionar la infraestructura subyacente. Esta tecnología facilita una amplia gama de servicios, como almacenamiento, procesamiento de datos y aplicaciones, los cuales son accesibles bajo demanda y con un modelo de pago basado en el uso. Entre los principales beneficios de la computación en la nube se encuentran la escalabilidad, la flexibilidad, la reducción de costos operativos y la facilidad de acceso a servicios avanzados sin la necesidad de una inversión inicial significativa en hardware y software.

El diseño y la gestión de infraestructuras en la nube requieren un enfoque cuidadosamente planificado, ya que implica coordinar una serie de componentes complejos, como servidores, redes, almacenamiento y la asignación dinámica de recursos. En este sentido, los avances en el campo de la simulación han permitido a los investigadores y desarrolladores crear entornos virtuales que imitan el funcionamiento de las infraestructuras de nube, lo que facilita el análisis y la mejora de la eficiencia de estas infraestructuras sin la necesidad de realizar costosos despliegues físicos. En este contexto, CloudSim se ha posicionado como una herramienta clave para la simulación y evaluación de arquitecturas y servicios en la nube.

**Justificación**

La magnitud de los datos, sumados al incremento del uso de la tecnología, obliga a las empresas a disponer de un centro de datos principal y un centro de datos alterno que se encuentren conectados y sincronizados entre sí, para mantener un respaldo de la información e implementar metodologías como alta disponibilidad y redundancia, con la finalidad de reducir el tiempo de respuesta en sus sistemas.

Se determina que para disponer de un centro de datos propio, se debe realizar una inversión económica inicial bastante alta y continua durante toda la vida del data center, para disponer de un centro de datos alterno en donde se pueda respaldar la información, se debe realizar una inversión semejante a la realizada en la adquisición del primero; para evitar costear esos gastos, se recomienda que una alternativa es contratar los servicios de cloud, de este modo se puede reducir los costos de construcción, instalación, configuración y mantenimiento, reduciendo el tiempo de implementación de los sistemas base y sistemas propios para el modelo de negocio de la entidad contratante del cloud.

Al optar por la opción de cloud computing es muy importante realizar un correcto dimensionamiento que permita conocer la capacidad y los recursos que se deben contratar, este proceso genera un inconveniente al desconocer si los recursos a contratar son los necesarios o si están sobredimensionados; para simplificar el proceso de estudio del dimensionamiento se desarrolló CloudMeter, que es una herramienta desarrollada en Java EE y como dependencia principal utiliza la librería CloudSim, que permite realizar una simulación de recursos en el cloud y como resultado presenta un informe con la actividad de la simulación que permite conocer si los recursos seleccionados puestos en funcionamiento son los adecuados para satisfacer los requerimientos.

Sin embargo, en este proyecto no se incluirá la parte del CloudMeter, se busca hacer la investigación enfocada principalmente en el CloudSim, fue argumentado su importancia como herramienta del programa por lo útil que es para una simulación de recursos en la nube.

**Objetivos**

**Objetivo general**

* Investigar sobre cómo desarrollar un software mediante CloudSim, para simular un proveedor de Infraestructura como Servicio.

**Objetivos específicos**

* Investigar sobre el diseño de un software que tenga funcionalidades de proveedor de infraestructura como servicio.
* Investigar el código y librerías necesaria para la simulación.
* Proponer soluciones para la prueba de la simulación.
* Realizar pruebas del software de simulación.

**Definición de Cloud Computing**

Para empezar explicando este algoritmo, cloud computing se define como un tipo de servicios tecnológico e informático, donde un usuario accede a un conjunto de recursos compartidos y que son ofertados por un proveedor como un servicio a usuarios que posean una suscripción o paguen una membresía; los usuarios acceden a los recursos por medio de internet.

Los recursos que un proveedor comparte con sus clientes o usuarios del cloud computing son asignados con un proceso automatizado el cual requiere de una gestión mínima, de este modo ahorrando tiempo al usuario que quiere hacer uso de manera inmediata un servicio después de contratarlo.

El Cloud Computing se define como una metodología que se caracteriza por disponer y proveer un conjunto de recursos informáticos, mismos que al ser compartidos son de costo accesible, también se debe considerar que este modelo de computación es escalable y su uso es bajo demanda.

Los servicios que la computación en la nube ofrece, son cada día más utilizados de manera directa o indirecta, por toda persona que utilice una computadora conectada a internet; esto debido a que el cloud computing provee servicios de alojamiento y procesamiento que simplifican el proceso para disponer información o servicios publicados en internet.

En la actualidad cualquier persona puede hacer uso de los servicios que ofrece el cloud computing debido a que la mayoría de los proveedores de este servicio, ofertan una prueba gratuita de sus servicios en un rango limitado de tiempo, otros proveedores asignan créditos gratuitos para que el cliente pueda probar sus servicios, pasado el tiempo de prueba, se debe realizar un pago para una suscripción y el costo varía en función de los recursos que el usuario consuma.

**IaaS (Infraestructura como Servicio)**

La Infraestructura como Servicio (IaaS, Infrastructure as a Service) pertenece al modelo XaaS. Con el IaaS, el proveedor ofrece una infraestructura TIC en forma de abono. El concepto de infraestructura se compone de: servidores, hardware de almacenamiento, suministro de red y máquinas virtuales (VM, Virtual Machines). Por lo tanto, también comprende los servicios relacionados como: seguridad y copias de seguridad con regularidad, las cuales se realizan a menudo en el modelo IaaS. Con IaaS, el cliente usa la infraestructura del proveedor para desarrollar softwares propios o comprados. Las empresas escogen a menudo el modelo IaaS para reducir la (gran) inversión que supone un hardware o para posponerla un tiempo. Sin embargo, este modelo se suele basar en el desarrollo realizado sobre PaaS (plataforma como servicio) o SaaS (software como servicio).

**“Alquilando” la infraestructura**

La IaaS es utilizada normalmente por empresas que no desean trabajar mucho tiempo con los mismos paquetes o modelos de software. A menudo, son aquellas las organizaciones que usan paquetes temporales o que prueban o desarrollan su propio software las que se aprovechan de las diferentes posibilidades que ofrece el modelo IaaS. Por ejemplo, las diferentes versiones de prueba de un nuevo paquete pueden realizarse durante un tiempo en el servidor del proveedor. Durante esta fase de prueba, solo se paga un “alquiler” por utilizar la infraestructura. Por lo tanto, la empresa que está desarrollando no está obligada a comprar un servidor potente.

**Los modelos de costes de IaaS**

La Infraestructura como Servicio se ofrece en varias formas. La más común es la de un abono con un precio fijo por hora, día, mes o año (para las pruebas de software). A veces, la factura se crea en base al uso de la capacidad de almacenamiento o procesamiento de las VM.

**Referencia bibliográfica**

CloudSim es un marco de código abierto, que se utiliza para simular la infraestructura y los servicios de computación en la nube. Está desarrollado por la organización CLOUDS Lab y está escrito íntegramente en Java. Se utiliza para modelar y simular un entorno de computación en la nube como un medio para evaluar una hipótesis antes del desarrollo de software para reproducir pruebas y resultados.

Por ejemplo, si tuviera que implementar una aplicación o un sitio web en la nube y quisiera probar los servicios y la carga que su producto puede manejar y también ajustar su rendimiento para superar los cuellos de botella antes de arriesgarse a la implementación, entonces dichas evaluaciones podrían realizarse simplemente codificando una simulación de ese entorno con la ayuda de varias clases flexibles y escalables proporcionadas por el paquete CloudSim, sin costo alguno.

**Beneficios de la simulación sobre el despliegue real**:

No se requiere inversión de capital. Con una herramienta de simulación como CloudSim no hay coste de instalación ni mantenimiento.

Fácil de usar y escalable. Puede cambiar los requisitos, como agregar o eliminar recursos, cambiando solo unas pocas líneas de código.

Los riesgos pueden evaluarse en una etapa anterior. En Cloud Computing, la utilización de bancos de pruebas reales limita los experimentos a la escala del banco de pruebas y hace que la reproducción de los resultados sea una tarea extremadamente difícil. Con la simulación, puede probar su producto en casos de prueba y resolver problemas antes de la implementación real sin ninguna limitación.

No hay necesidad de enfoques de prueba y error. En lugar de confiar en evaluaciones teóricas e imprecisas que pueden conducir a un rendimiento del servicio y una generación de ingresos ineficientes, puede probar sus servicios en un entorno repetible y controlado sin costo alguno con CloudSim. (Greyrat, 2022)

Esta información es muy relevante para proyectos que requieren simulación en entornos de computación en la nube. Basándose en simular un proveedor de Infraestructura como Servicio (IaaS), CloudSim ofrece varias ventajas clave, como dice el artículo anterior, no requiere inversión en hardware o licencias, se pueden modelar diferentes configuraciones de proveedores de IaaS sin necesidad de costos iniciales. CloudSim permite diseñar y probar configuraciones complejas de IaaS adaptadas a necesidades específicas. Por ejemplo, modelar la distribución de recursos entre múltiples usuarios o experimentar con diferentes algoritmos de gestión de recursos para evaluar su eficiencia. Me interesa analizar el rendimiento de un proveedor de IaaS tomando en cuenta algo como las cargas de trabajo o fallas simuladas, según lo investigado, la herramienta permite registrar y analizar los resultados de las simulaciones, lo que facilita optimizar las políticas de asignación de recursos en función del rendimiento deseado.

**Entre algunas observaciones:**

* Se busca Determinar casos de uso específicos que simular, como la respuesta de la infraestructura ante picos de demanda de uso.
* Evaluar la herramienta en la parte de los recursos en CloudSim para reflejar mejor cómo operaría este proveedor de IaaS en el mundo real.
* Usar las simulaciones para comparar diferentes configuraciones y medir métricas clave como tiempo de respuesta, uso de recursos y costos operativos.
* Simular fallos en los recursos para evaluar la resiliencia de la infraestructura y ajustar tu estrategia de recuperación

**Desarrollo y Análisis de Resultados**

Para el desarrollo y análisis de los resultados de la investigación, se utilizó la versión 7.0 del CloudSim, en el que se simularon varios entornos de ejemplos de código para un proveedor de infraestructura como servicio.

Entre los ejemplos que se utilizaron, me gustaría destacar 2, que a mi parecer son los más interesantes ya que poseen estas características propias de un IaaS.

**Los Ejemplos son:**

## **Ejemplo 1:**

Se implementa una simulación con características propias de una infraestructura como servicio (IaaS), ya que modela recursos físicos (hosts, almacenamiento, ancho de banda) y su uso por máquinas virtuales (VMs) y tareas (Cloudlets).

En particular, tiene los siguientes elementos clave de un modelo IaaS:

* **Host físico:** Se definen hosts con características específicas como CPU, RAM, almacenamiento, y ancho de banda.
* **Máquinas virtuales (VMs):** Estas son instancias virtualizadas que los hosts alojan y utilizan para ejecutar tareas.
* **Cloudlets (tareas):** Representan las aplicaciones que se ejecutan en las VMs. Además, integra el concepto de red mediante la clase NetworkDatacenter, y la configuración de switches, añadiendo funcionalidad para manejar la transferencia de datos entre VMs.
* **Broker:** Actúa como intermediario entre los usuarios (simulados) y el datacenter, gestionando la asignación de tareas a las VMs.

A continuación, la respuesta de la consola al ejecutar el ejemplo:

Texto

Descripción generada automáticamente con confianza media

## **Inicialización:**

Por lo que se puede ver en este ejemplo, el CloudSim y el broker se inicializan. Se crea un datacenter (Datacenter\_0) con múltiples hosts (máquinas físicas), que son los proveedores de recursos.

En el ejemplo, el datacenter tiene dos hosts, cada uno asignado a una VM.

Se crean dos máquinas virtuales (VM #0 y VM #1) con configuraciones específicas: VM #0 utiliza un planificador de tareas en tiempo compartido (time-shared). VM #1 utiliza un planificador en espacio compartido (space-shared).

**Creación y asignación de Cloudlets:** Se crean seis tareas (Cloudlets), cada una con diferentes requisitos de procesamiento (longitud). Estas tareas se asignan a las VMs:

Los Cloudlets #0, #1, y #2 se asignan a VM #0.

Los Cloudlets #3, #4, y #5 se asignan a VM #1.

**Ejecución:** Se inician las VMs en los hosts correspondientes, y los Cloudlets son enviados a cada VM. El planificador de cada VM define cómo se ejecutan los Cloudlets:

En tiempo compartido, los recursos se dividen entre las tareas, permitiendo que varias tareas progresen simultáneamente.

En espacio compartido, cada tarea usa todos los recursos hasta que termina, ejecutándose una por una.

**Resultados:**

**Cloudlets 0-2 (en VM #0):**

Se ejecutan de forma simultánea gracias al planificador time-shared, pero su tiempo de finalización varía dependiendo de su longitud (10,000; 100,000; 1,000,000 instrucciones).

**Cloudlets 3-5 (en VM #1):**

Se ejecutan uno por uno debido al planificador space-shared. Por ejemplo: El Cloudlet #3 termina en 10.01, y el Cloudlet #4 comienza justo después.

La diferencia en los tiempos de ejecución (30, 210, 2000, etc.) refleja los diferentes tamaños y prioridades de los Cloudlets, así como las políticas de planificación de las VMs.

**Ejemplo 2:**

**Inicialización de CloudSim**: Se crea un objeto CloudSim que sirve como núcleo para manejar los eventos y la simulación.

**Creación del Datacenter:**

Un datacenter representa la infraestructura física que ofrece recursos. Aquí se crean dos hosts (servidores físicos), cada uno con:

* 8 CPUs.
* 16 GB de RAM.
* 1 millón de unidades de almacenamiento.
* También se especifican costos asociados a la utilización de los recursos del datacenter (CPU, memoria, almacenamiento, y ancho de banda).

**Creación del Broker:**

El broker actúa como intermediario entre los usuarios (que envían tareas) y el datacenter. Se encarga de asignar tareas a las máquinas virtuales (VMs) y comunicarse con el datacenter.

**Creación de Máquinas Virtuales (VMs):**

Se crean 4 VMs, cada una con:

* 2 CPUs.
* 4 GB de RAM.
* 10 GB de almacenamiento.
* Estas VMs representan los recursos virtualizados que el datacenter asigna a los usuarios.

**Creación de Tareas (Cloudlets):**

Se crean 6 tareas (cloudlets) que necesitan ser procesadas. Cada tarea tiene:

* 10,000 instrucciones.
* Un archivo de entrada y salida de 300 unidades de tamaño.
* Las tareas representan aplicaciones o cargas de trabajo que los usuarios desean ejecutar.

**Ejecución de la Simulación:**

Se inicia la simulación, permitiendo que el broker asigne las tareas (cloudlets) a las máquinas virtuales y coordine su ejecución.

**Visualización de Resultados:**

Al finalizar la simulación, se muestran los resultados en una tabla, incluyendo métricas como:

* El tiempo de inicio y finalización de las tareas.
* El estado de las tareas (éxito o falla).
* El VM en el que se ejecutó cada tarea.

Tabla

Descripción generada automáticamente

Se mostrará solo la tabla final debido a que la ejecución de este ejemplo es muy extensa, pero con la explicación previa y los resultados de la imagen queda claro el objetivo del código.

**Conclusión**

Para concluir con una parte más generalizada de los resultados tras el análisis de los ejemplos, se puede decir que, en el caso del ejemplo 1, el código simula el comportamiento de una infraestructura en la nube, donde se ejecutan cloudlets (trabajos) sobre máquinas virtuales (VMs) que están alojadas en un centro de datos.El código crea una serie de cloudlets y asigna su ejecución a las VMs de acuerdo con estos modelos de asignación.

En caso del ejemplo 2, es más desarrollado, por eso precisamente escogí estos 2 ejemplos comparándolos, en el ejemplo 1, se utilizaba un DatacenterBroker directamente en la simulación, lo que es más común y directo. En este ejemplo, se introduce un nuevo concepto: el GlobalBroker. Este actúa como un intermediario entre el sistema principal de simulación y el broker que gestiona las máquinas virtuales (VMs) y los cloudlets. La creación de este GlobalBroker se maneja como una entidad separada y sus eventos se gestionan mediante el método processEvent.

Esto marca una diferencia porque el flujo de trabajo está más descentralizado y el GlobalBroker coordina la creación de un broker normal y la asignación de recursos. Este enfoque puede ser útil en simulaciones más complejas donde se necesitan intermediarios o gestores globales.

En el código anterior, las máquinas virtuales (VMs) y los cloudlets se creaban directamente en el método main y se enviaban al broker inmediatamente después de su creación.

En este código, el proceso de creación de VMs y cloudlets se realiza mediante el GlobalBroker, que es quien organiza y delega la creación de estas entidades. El broker es creado de forma dinámica dentro del GlobalBroker, lo que da un enfoque más flexible y modular.

Este cambio introduce una abstracción adicional, lo que facilita la adaptación del sistema a situaciones más complejas donde varios brokers pueden estar involucrados o gestionados por una capa de administración global. En este otro código, el GlobalBroker gestiona eventos. El método processEvent maneja el evento de creación del broker, donde se crean las VMs y los cloudlets, se envían al broker, y se reanuda la simulación, este patrón permite una mayor flexibilidad y control sobre el ciclo de vida de la simulación, permitiendo la inyección de otros eventos en el futuro (si se desea, como la reconfiguración del sistema, la creación de más recursos, etc.).

**Estructura del Broker:**

El GlobalBroker actúa como una capa adicional sobre el DatacenterBroker, lo que permite la creación dinámica del broker y su integración con el sistema. Además, el GlobalBroker extiende SimEntity, lo que le permite interactuar con el sistema de simulación como una entidad propia, gestionando eventos específicos y la creación de brokers.

Esto da como resultado una simulación que puede ser más fácilmente escalable o modular, permitiendo la incorporación de nuevos brokers en tiempo de ejecución.

La creación de cloudlets y VMs es bastante similar en ambos códigos, pero en este ejemplo, la cantidad de VMs y cloudlets se pasa como parámetros más dinámicos a través del GlobalBroker en lugar de estar fijada de manera más estática.

**Estructura del Datacenter:**

En ambos ejemplos, se crea un Datacenter con un conjunto de Hosts y PEs (procesadores). Sin embargo, este código tiene dos centros de datos (Datacenter\_0 y Datacenter\_1), lo que agrega más flexibilidad al simular una configuración de nube distribuida o multi-datacenter.

Además, se mantiene la estructura detallada de cómo se crean los PEs (procesadores) y los Hosts, pero con la capacidad de usar múltiples datacenters, lo que permite una simulación más rica de la distribución y gestión de recursos, se especifican propiedades adicionales como costos de memoria, almacenamiento y ancho de banda en la creación del Datacenter, lo que permite a la simulación gestionar estos aspectos económicos en su flujo de trabajo.

# **Bibliografía**

Greyrat. (2022). *Que es el CloudSim.* Barcelona Geeks.

[¿Qué es Cloud Sim? – Barcelona Geeks](https://barcelonageeks.com/que-es-cloudsim/)

CloudSim plus / 7.0, tomado de GitHub, para las pruebas y análisis de distintos ejemplos

<https://github.com/cloudsimplus/cloudsimplus-examples>