

Universidad San Carlos de Guatemala

Introducción a la programación y computación 2

17 de diciembre del 2025

Auxiliar Luis Antonio

Ingeniero Marlon



## Proyecto 1

Alison Sofia De Paz Gómez

202400822

Luis Pedro Gonzales Xiloj

202406139

## Introducción

La computación en la nube ha revolucionado la manera en que las organizaciones gestionan sus recursos tecnológicos. Plataformas como Amazon y Google administran millones de recursos distribuidos globalmente, requiriendo sistemas sofisticados de orquestación y balanceo de carga. Es por eso que hay una necesidad de comprender los fundamentos de estos sistemas mediante la implementación de simuladores que repliquen su comportamiento.

CloudSync nace como una respuesta educativa a esta necesidad, proporcionando un entorno de aprendizaje donde se pueden experimentar los desafíos reales de la gestión de infraestructura cloud. El sistema simula una plataforma distribuida capaz de administrar múltiples centros de datos, cada uno con su propia capacidad de CPU, RAM y almacenamiento. Sobre estos centros se despliegan máquinas virtuales que, a su vez, pueden alojar contenedores para ejecutar aplicaciones específicas.

El sistema se fundamenta en principios de diseño orientado a objetos, a diferencia de las estructuras nativas de Python, se han implementado listas simples enlazadas, listas simplemente enlazadas con referencias a primer y último elemento, y colas de prioridad desde cero, garantizando un control total sobre el comportamiento y las operaciones de cada componente.

Esto ayuda a procesar archivos XML que contienen configuraciones iniciales y secuencias de instrucciones, permitiendo cargar estados complejos del sistema y ejecutar operaciones como creación de recursos, migraciones y procesamiento de solicitudes. Esta capacidad de procesamiento XML, combinada con la generación de reportes visuales mediante Graphviz, proporciona una visión completa del ciclo de vida de la gestión de recursos en la nube.

# MARCO TEÓRICO

## Computación en la Nube

La computación en la nube permite el acceso conveniente y bajo demanda a un conjunto compartido de recursos computacionales para que pueden ser rápidamente aprovisionados y liberados con mínimo esfuerzo.

En CloudSync, se puede crear y destruir máquinas virtuales dinámicamente, la medición precisa del uso de recursos en cada componente del sistema, y la capacidad de los usuarios de solicitar recursos mediante una cola de prioridad sin intervención manual del administrador.

Las estructuras de datos enlazadas son la columna vertebral de CloudSync. A diferencia de las estructuras basadas en arreglos, las listas enlazadas permiten un crecimiento fluido y operaciones de inserción y eliminación eficientes.

**Lista Simple Enlazada:** Cada nodo contiene un dato y una referencia al siguiente nodo. se utiliza para gestionar las máquinas virtuales dentro de cada centro de datos y los contenedores dentro de cada máquina virtual.

**Lista Simplemente Enlazada con Referencias:** Una variación que mantiene referencias tanto al primer como al último nodo, permitiendo inserciones al final en. Se usa para gestionar los centros de datos del sistema, donde las operaciones de agregación son frecuentes.

**Cola de Prioridad** es importante para ver el orden de las solicitudes. Cada solicitud tiene una prioridad, donde valores mayores indican mayor urgencia. La implementación utiliza una lista enlazada ordenada donde las inserciones mantienen el orden de prioridad.

XML es un lenguaje de marcado que define un conjunto de reglas para codificar documentos en un formato legible tanto para humanos como para máquinas. En CloudSync, XML sirve como formato de entrada y salida, permitiendo definir configuraciones complejas de manera estructurada.

Graphviz es una herramienta de código abierto para la visualización de grafos. El sistema CloudSync genera archivos en formato DOT que Graphviz procesa para crear representaciones visuales de la infraestructura cloud. Los grafos dirigidos permiten representar las relaciones jerárquicas entre centros de datos, máquinas virtuales y contenedores, facilitando la comprensión visual del estado del sistema.

**Capa de Modelos:** Contiene las clases que representan las entidades del dominio, cada clase encapsula los datos y la lógica de negocio específica de su entidad.

**Capa de Estructuras de Datos:** Implementa las estructuras enlazadas personalizadas que soportan las operaciones fundamentales del sistema.



**Capa de Controladores:** Los gestores coordinan las operaciones entre modelos y estructuras de datos, implementando la lógica de negocio compleja.

**Capa de Entrada/Salida:** Maneja el procesamiento XML y la generación de reportes.

**Capa de Presentación:** La interfaz de consola que permite la interacción del usuario con todas las funcionalidades del sistema.

**Centro Datos:** enseña un centro de datos físico o lógico con capacidad definida de CPU, RAM y almacenamiento. tiene dos tipos de valores: totales y disponibles. Ayuda a una lista de máquinas virtuales alojadas y proporciona métodos para verificar disponibilidad, consumir recursos y liberarlos cuando se eliminan máquinas virtuales.

**Máquina Virtual:** Modela una máquina virtual que consume recursos de un centro de datos y puede alojar múltiples contenedores. Los recursos de la VM se reservan del centro de datos, pero los contenedores consumen recursos de la VM, no directamente del centro. Los recursos de CPU en contenedores se expresan en porcentaje de la CPU total de la VM.

**Contenedor:** Representa una aplicación dentro de una máquina virtual. Consume un porcentaje del CPU de la VM y una cantidad específica de RAM en megabytes. Mantiene un estado que puede modificarse dinámicamente.

**Solicitud:** Encapsula una petición de cliente que requiere recursos del sistema. Contiene información sobre el tipo de operación, prioridad, recursos necesarios y cliente solicitante. Las solicitudes se procesan según su prioridad mediante una cola especializada.

**Asignación de Recursos por Disponibilidad:** Cuando se procesa una solicitud de tipo Deploy o Backup, el proyecto tiene que seleccionar el centro de datos óptimo. El algoritmo implementado calcula para cada centro la suma de recursos y selecciona aquel con mayor disponibilidad total.

**Migración de Máquinas Virtuales:** El proceso de migración requiere que se valide muchas veces:

1. Verificar que la VM origen existe
2. Verificar que el centro destino existe y es diferente del origen
3. Validar que el centro destino tiene recursos suficientes
4. Liberar recursos en el centro origen
5. Remover VM de la lista del centro origen
6. Consumir recursos en el centro destino
7. Agregar VM a la lista del centro destino
8. Actualizar la referencia de centro en la VM

**Procesamiento de Cola de Prioridad:** Las solicitudes se procesan siempre el elemento al frente de la cola. El procesamiento incluye validaciones de recursos y ejecución de la acción correspondiente según el tipo de solicitud.

## Estructuras de Datos Personalizadas

La implementación de estructuras enlazadas desde cero fue uno de los casos más difíciles y del proyecto. Se desarrollaron tres estructuras principales:

**ListaSimple:** Utiliza nodos con referencia al siguiente elemento y mantiene únicamente una referencia a la cabeza. Implementa operaciones de inserción al final, búsqueda por identificador, eliminación y listado completo. Esta estructura se emplea para gestionar las VMs dentro de centros y los contenedores dentro de VMs.

**ListaSimpleEnlazada:** Mejora de la lista simple que mantiene referencias tanto al primer como al último elemento, permitiendo inserciones al final. Se utiliza para gestionar los centros de datos, donde las operaciones de agregación son frecuentes pero las eliminaciones son raras.

**ColaPrioridad:** Implementación de cola de prioridad basada en lista enlazada ordenada. Las inserciones mantienen el orden de mayor a menor prioridad, realizando comparaciones durante la inserción para encontrar la posición correcta. El desencolar simplemente extrae el primer elemento, garantizando que siempre se procesa la solicitud más urgente.

## Gestión de Recursos

**Nivel Centro de Datos:** Cada centro mantiene contadores de recursos totales y disponibles. Cuando se crea una VM, se restan los recursos del centro. Cuando se elimina una VM, se devuelven.

**Nivel Máquina Virtual:** La VM tiene recursos asignados del centro y mantiene su propia contabilidad de recursos disponibles para contenedores. Los recursos de contenedores se expresan en diferentes unidades y deben convertirse a las unidades de la VM.

**Nivel Contenedor:** Los contenedores no mantienen contabilidad propia, simplemente registran cuántos recursos consumen.

Esta jerarquía asegura que los recursos se contabilicen correctamente en cada nivel sin duplicación ni pérdida.

## Procesamiento XML

El parser XML implementado maneja tres secciones principales del archivo de entrada:

**Sección centrosDatos:** Extrae información de ubicación y capacidad de cada centro, creándolos en el sistema mediante el gestor correspondiente.

**Sección maquinasVirtuales:** Procesa cada VM con sus recursos y centro asignado. Dentro de cada VM, procesa recursivamente sus contenedores, creando la jerarquía completa.

**Sección solicitudes:** Crea objetos Solicitud y los inserta en la cola de prioridad, respetando automáticamente el orden por urgencia.

**Sección instrucciones:** Ejecuta comandos como creación de VMs, migraciones o procesamiento de solicitudes.

El generador XML de salida recorre todas las estructuras del sistema, calculando estadísticas y formateando la información según el esquema requerido.

#### **4.4 Generación de Reportes**

Los reportes Graphviz se generan construyendo cadenas de texto en formato DOT. Cada reporte implementa una lógica específica:

**Reporte de Centros:** Muestra todos los centros como nodos con información de recursos y conecta desde un nodo título hacia cada centro.

**Reporte de VMs de un Centro:** Crea un nodo para el centro seleccionado y conecta todas sus VMs, mostrando recursos asignados y disponibles.

**Reporte de Contenedores de una VM:** Similar al anterior, pero descendiendo un nivel más en la jerarquía. Los colores de los nodos de contenedores varían según su estado.

**Reporte de Cola de Solicitudes:** Representa la cola como una secuencia lineal de nodos, donde el color indica la urgencia (rojo para prioridad alta, naranja para media, verde para baja).





## CONCLUSIONES

La implementación de CloudSync ha demostrado ser un ejercicio completo de ingeniería de software que integra múltiples conceptos fundamentales de ciencias de la computación. El proyecto ha permitido comprender profundamente el funcionamiento interno de estructuras de datos que normalmente se utilizan como cajas negras en lenguajes de alto nivel.

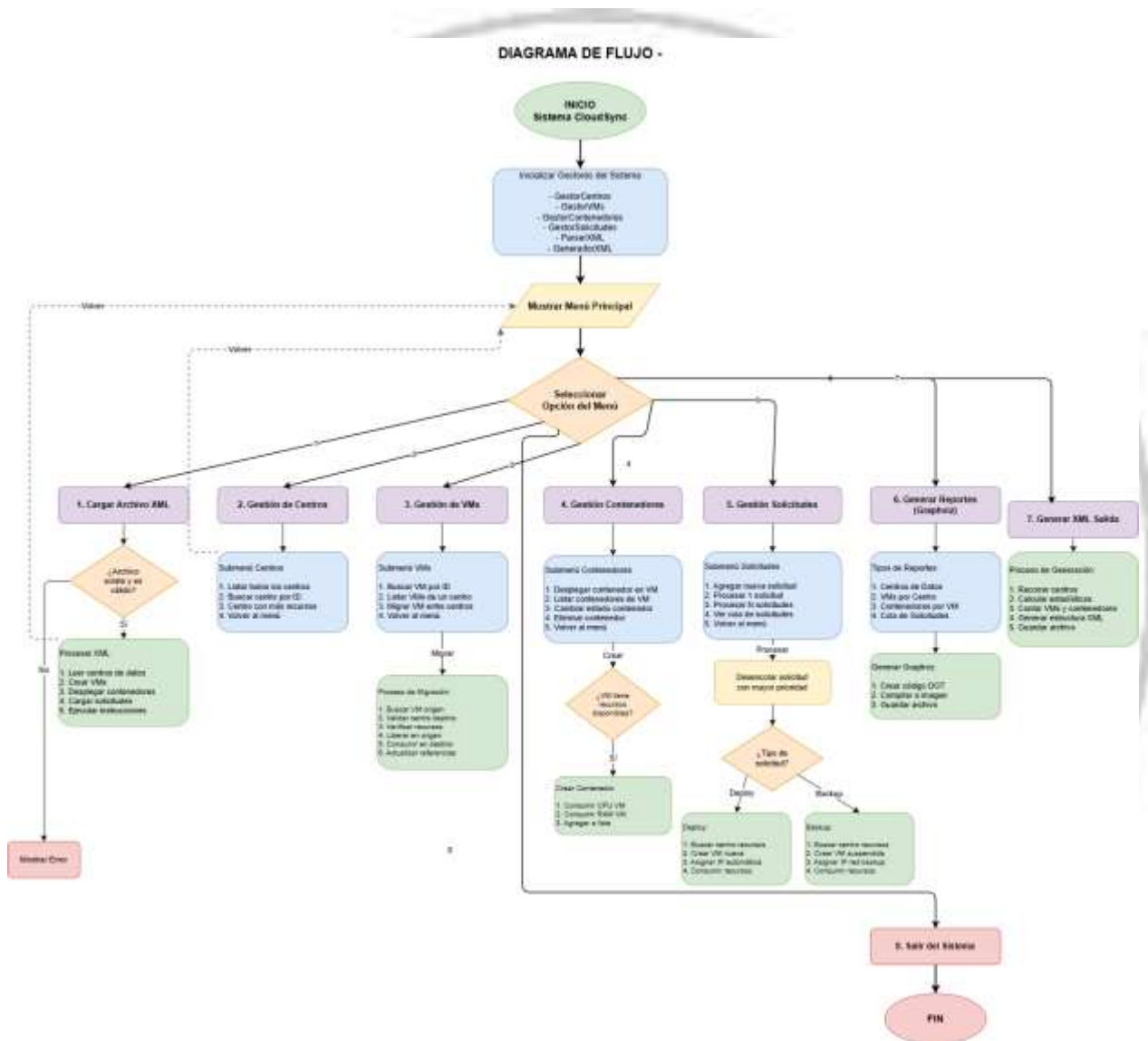
La arquitectura de capas implementada facilita el mantenimiento y extensión del sistema. La separación clara entre modelos, controladores y utilidades permite modificar componentes individuales sin afectar al resto del sistema.

El sistema de gestión de recursos implementado, aunque simplificado comparado con orquestadores reales como Kubernetes, captura la esencia del problema: asignación eficiente de recursos limitados a demandas variables, con restricciones de capacidad y priorización de solicitudes.

La experiencia de construir todas las estructuras de datos desde cero proporciona una comprensión profunda de sus costos temporales y espaciales, algo que se pierde al usar únicamente implementaciones nativas del lenguaje.

El procesamiento XML y la generación de reportes visuales añaden componentes prácticos que acercan el proyecto a sistemas reales, donde la interoperabilidad mediante formatos estándar es esencial.

# Diagrama de flujo





# Diagrama de clases

