



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



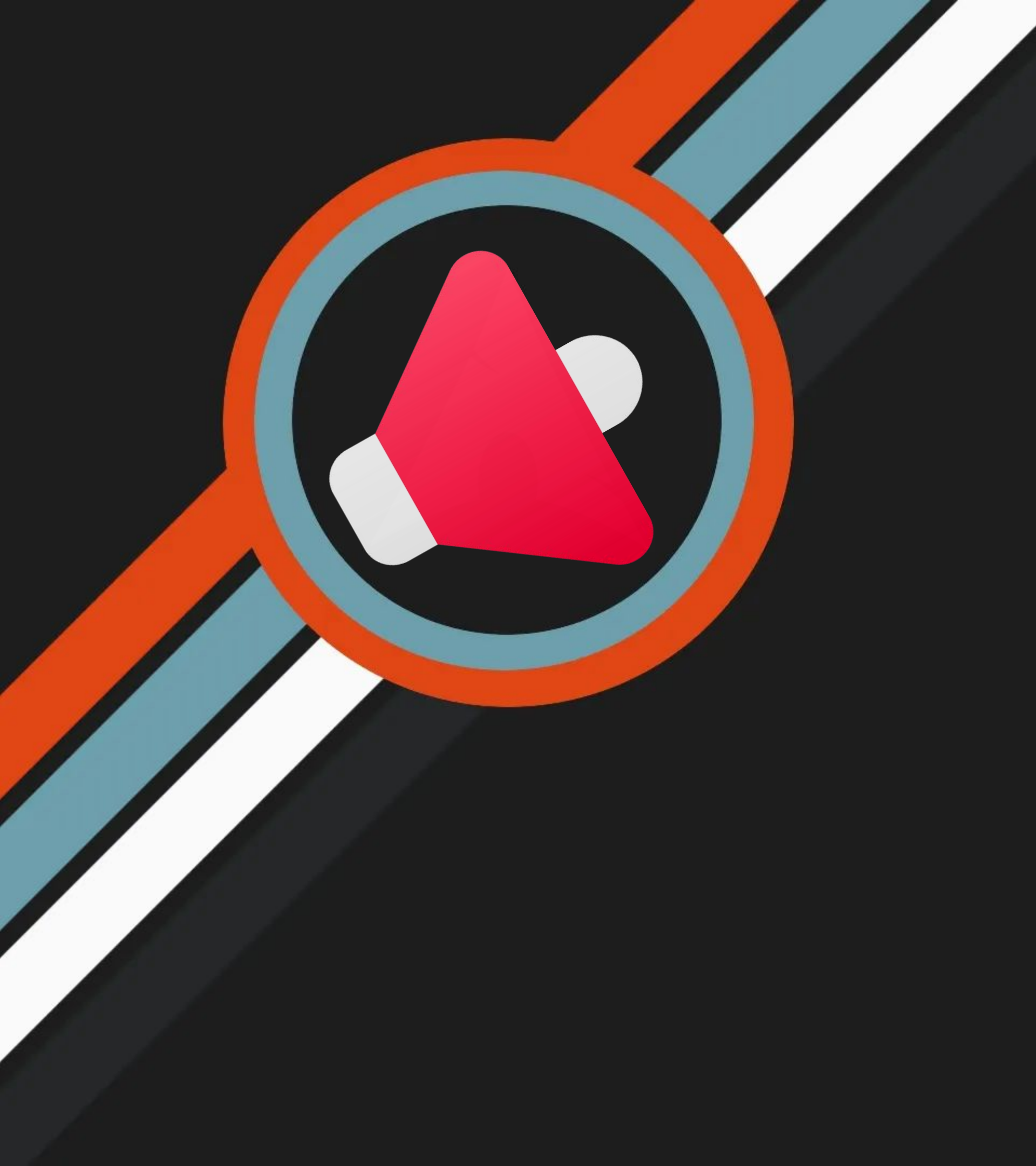
Día, Fecha:	Viernes, 24/10/2025
Hora de inicio:	17:20

Sistemas Operativos 2 [A]

Steven S. Jocol Gomez

UNIDAD 4: SISTEMAS ESPECIALIZADOS Y DISTRIBUIDOS

Escuela de Ingeniería de Ciencias Y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala



Anuncios Importantes

Entrega Proyecto
Calificacion



AGENDA



1. Sistemas Distribuidos
2. Arquitectura de sistemas distribuidos
3. Virtualización y Gestión de Recursos en Sistemas en la Nube
4. Soporte del sistema operativo para multi-tenencia y elasticidad
5. Sistemas operativos específicos de la nube (OSv, MirageOS)
6. Desafíos y soluciones de seguridad en la nube

COMPETENCIA(S) QUE DESARROLLAREMOS



- Entender el marco de referencia o estructura lógica general de un sistema operativo, que le permita la utilización, análisis y diseño de sistemas operativos.
- Evaluar sistemas operativos para soluciones específicas.



LABORATORIO SISTEMAS OPERATIVOS 2

SISTEMAS DISTRIBUIDOS



SISTEMAS DISTRIBUIDOS

Un sistema distribuido es un conjunto de programas que aprovechan recursos computacionales en múltiples nodos para alcanzar un objetivo común. En estos sistemas, también conocidos como "computación distribuida" o "bases de datos distribuidas", los nodos se comunican y sincronizan a través de una red compartida. Los nodos pueden ser dispositivos de hardware físico o procesos de software independientes, e incluso otros sistemas encapsulados. La meta principal de los sistemas distribuidos es reducir cuellos de botella y eliminar puntos únicos de falla, aumentando así la eficiencia, resiliencia y escalabilidad del sistema en su conjunto.



EJEMPLOS COMUNES

Internet

Una red global de computadoras conectadas que permite la transmisión de información a **nivel mundial**, donde cada nodo puede funcionar de manera independiente y al mismo tiempo como parte de una red colaborativa.

Servicios de mensajería

(p. ej., WhatsApp, Telegram): Las aplicaciones de mensajería dependen de sistemas distribuidos para enviar y recibir mensajes en **tiempo real** a través de redes de servidores distribuidos en distintas ubicaciones geográficas.

Aplicaciones en la nube

(p. ej., Google Drive, Dropbox): Servicios de almacenamiento y procesamiento donde los datos están distribuidos en **varios servidores** para aumentar la accesibilidad y redundancia.

Redes de distribución de contenido

(CDN) Sistemas como Akamai y Cloudflare distribuyen copias de contenido (videos, imágenes, archivos) en múltiples servidores alrededor del mundo, optimizando la disponibilidad y velocidad de acceso.

CARACTERÍSTICAS DE LOS SISTEMAS DISTRIBUIDOS

Cada una de estas propiedades es esencial para el diseño y funcionamiento de un sistema distribuido robusto

Transparencia

La capacidad de hacer que los aspectos de la distribución (ubicación, fallos, concurrencia) sean "invisibles" o no visibles para los usuarios.

01

TRANSPARENCIA DE ACCESO

Permite acceder a los recursos sin importar dónde estén ubicados.

02

TRANSPARENCIA DE UBICACIÓN

Los usuarios no necesitan saber la ubicación física de los recursos.

03

TRANSPARENCIA DE REPLICACIÓN

Múltiples copias de datos o servicios aparecen como una sola unidad para los usuarios.

04

TRANSPARENCIA DE CONCURRENCIA

Varios usuarios pueden interactuar con el sistema sin interferirse entre sí.

CARACTERÍSTICAS DE LOS SISTEMAS DISTRIBUIDOS

Cada una de estas propiedades es esencial para el diseño y funcionamiento de un sistema distribuido robusto

Escalabilidad

La capacidad del sistema para crecer y adaptarse a medida que la demanda de usuarios o recursos aumenta.

Redes sociales como Facebook y Twitter deben escalar sus sistemas para manejar miles de millones de usuarios activos y peticiones en tiempo real, sin comprometer el rendimiento

01

ESCALABILIDAD HORIZONTAL VS. VERTICAL

- **Horizontal:** Añadir más nodos al sistema (por ejemplo, agregar servidores adicionales para manejar más tráfico web).
- **Vertical:** Aumentar la capacidad de los nodos existentes (por ejemplo, añadir más memoria o procesadores a un servidor).

02

RETOS EN LA ESCALABILIDAD

- **Consistencia de datos:** Mantener la coherencia de la información entre nodos puede ser más complejo al agregar más nodos.
- **Latencia de comunicación:** A medida que el sistema crece, la distancia entre nodos también puede crecer, afectando el tiempo de respuesta.

CARACTERÍSTICAS DE LOS SISTEMAS DISTRIBUIDOS

Cada una de estas propiedades es esencial para el diseño y funcionamiento de un sistema distribuido robusto

Confiabilidad

Capacidad del sistema para seguir funcionando correctamente incluso cuando ocurren fallos en ciertos componentes.

Netflix utiliza múltiples servidores en diversas ubicaciones para asegurarse de que el contenido esté disponible para los usuarios incluso si algunos servidores fallan.

01

TOLERANCIA A FALLOS

Los sistemas distribuidos a menudo incluyen mecanismos de redundancia y recuperación que permiten continuar con las operaciones a pesar de errores o caídas de ciertos nodos.

02

REDUNDANCIA Y RECUPERACIÓN

- **Redundancia:** Duplicación de datos y servicios en varios nodos.
- **Recuperación ante fallos:** Estrategias para reanudar el servicio lo antes posible después de un fallo (p. ej., balanceo de carga, copias de seguridad automáticas).

CARACTERÍSTICAS DE LOS SISTEMAS DISTRIBUIDOS

Cada una de estas propiedades es esencial para el diseño y funcionamiento de un sistema distribuido robusto

Concurrencia

La capacidad de manejar múltiples procesos o peticiones simultáneas en el sistema, facilitando el uso compartido de recursos por varios usuarios al mismo tiempo.

En bases de datos distribuidas como las utilizadas por bancos, la concurrencia asegura que varias transacciones puedan realizarse simultáneamente sin que los datos pierdan consistencia o integridad.

01

SINCRONIZACIÓN Y CONSISTENCIA

- **Sincronización:** Controla la interacción de múltiples procesos que pueden compartir recursos o información.
- **Consistencia:** Asegura que todos los usuarios vean una misma versión de los datos en un entorno distribuido.

02

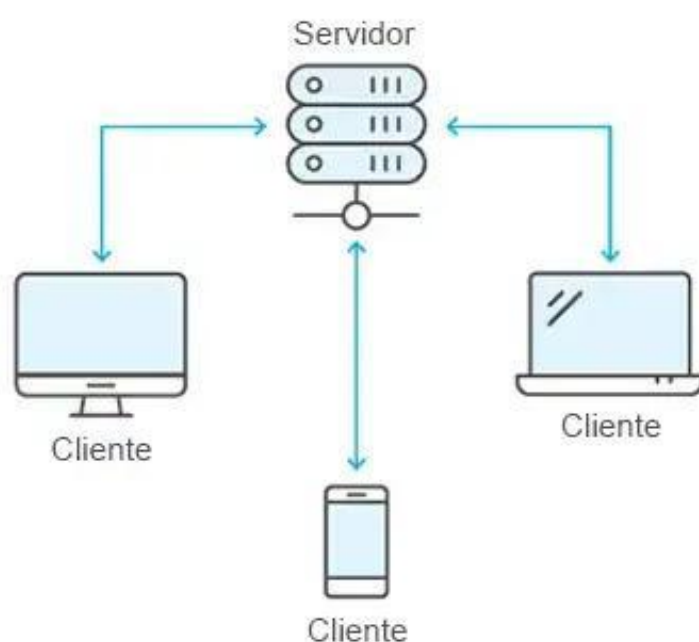
CONTROL DE CONCURRENCIA

- Evita condiciones de carrera y garantiza la integridad de los datos.
- Puede implicar mecanismos como bloqueos, semáforos, y algoritmos de consenso en sistemas distribuidos.

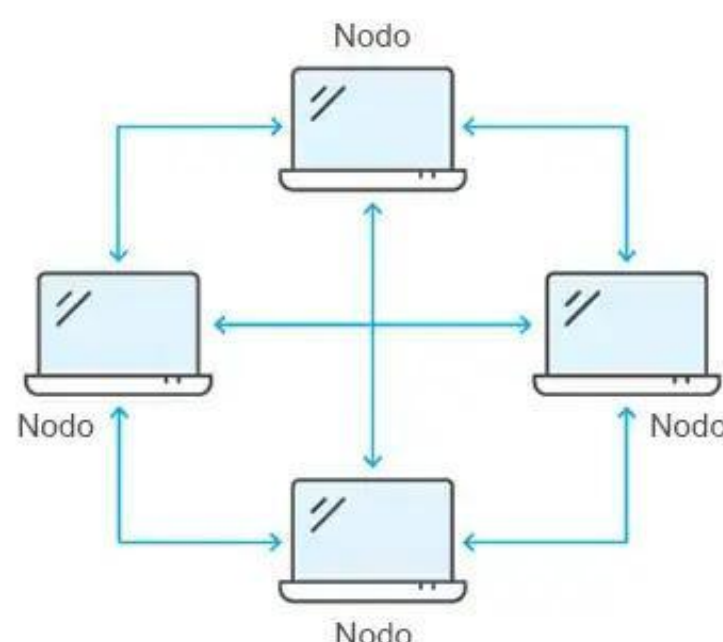
MODELOS DE ARQUITECTURA EN SISTEMAS DISTRIBUIDOS

Cliente-Servidor vs P2P

Red Cliente-Servidor



Red P2P



01

CLIENTE SERVIDOR

La red cliente-servidor está compuesta por al menos un servidor central que controla la red y una serie de dispositivos cliente que se conectan al servidor para realizar tareas específicas.

Ejemplos típicos: *servidores web y bases de datos.*

02

PEER-TO-PEER (P2P)

Descripción del modelo en el que cada nodo actúa como cliente y servidor, permitiendo que compartan recursos de manera descentralizada.

Ejemplos comunes: *redes de intercambio de archivos y blockchain.*

CONSENSO Y COORDINACIÓN EN SISTEMAS DISTRIBUIDOS

¿QUÉ ES EL CONSENSO?

El consenso en sistemas distribuidos se refiere a la capacidad de **varios nodos** (o máquinas) para llegar a un acuerdo sobre un valor o estado compartido, a pesar de fallas o discrepancias.

NECESIDAD

Es fundamental para mantener la consistencia en sistemas distribuidos, donde la comunicación **no es siempre fiable**.

PROBLEMAS DE CONSENSO

- **Elecciones en una Red:** Elegir un nodo líder en un grupo distribuido, por ejemplo, para coordinar acciones en un clúster.
- **Consistencia en Bases de Datos Distribuidas:** Asegurar que todas las réplicas de datos estén sincronizadas y representen el mismo estado, aun cuando se produzcan fallos.

IMPORTANCIA

El consenso asegura que el sistema pueda continuar operando correctamente y ofrece una única fuente de "**verdad**" en un entorno de múltiples nodos.

COORDINACIÓN EN SISTEMAS DISTRIBUIDOS

Importancia de la Coordinación y Sincronización

- Los sistemas distribuidos requieren que múltiples nodos trabajen juntos de **forma coordinada** para evitar problemas como la doble escritura o los conflictos de datos.
- La coordinación permite que los procesos y nodos realicen tareas **sincronizadamente**, gestionando el acceso a los recursos y manteniendo la consistencia de datos en tiempo real.

Coordinación en Sistemas de Archivos Distribuidos

Google File System (GFS): GFS necesita coordinación para que los nodos puedan leer y escribir archivos de manera sincronizada, evitando la corrupción de datos y manteniendo la integridad del sistema.

GFS utiliza un nodo central de control (master) que coordina qué fragmentos del archivo deben escribirse y en qué orden. Otros nodos de almacenamiento actúan como réplicas para garantizar la disponibilidad y consistencia de los datos

SISTEMAS EN LA NUBE

Los sistemas en la nube son infraestructuras distribuidas que permiten a los usuarios acceder a recursos de cómputo, almacenamiento y red de forma remota a través de internet.



- **Infraestructura distribuida:** La nube es una arquitectura en la que los recursos de cómputo y almacenamiento están repartidos a lo largo de una red de servidores ubicados en diferentes ubicaciones geográficas.
- **Acceso remoto a través de Internet:** Los usuarios pueden acceder a estos recursos desde cualquier lugar, a través de internet, lo que facilita el acceso global y la colaboración remota.
- **Modelo de entrega bajo demanda:** A diferencia de la infraestructura tradicional, los recursos se proporcionan **bajo demanda**, lo que significa que los usuarios pueden aumentar o reducir el uso de recursos según sus necesidades actuales, sin preocuparse por los detalles de hardware.

VENTAJAS DE LOS SISTEMAS EN LA NUBE

Escalabilidad

Automática: La capacidad de agregar o reducir recursos de manera automática según la carga de trabajo o el tráfico, lo que permite que aplicaciones y servicios sean escalables de manera flexible.

En eventos de alta demanda, como el Black Friday en comercio electrónico, los recursos pueden escalar automáticamente para gestionar el incremento en el tráfico.

Accesibilidad

Disponibilidad Global: Acceso a servicios y aplicaciones en la nube desde cualquier dispositivo conectado a internet, independientemente de la ubicación.

Aplicaciones como Google Drive o Dropbox permiten a los usuarios acceder a sus archivos desde cualquier lugar.

Costos Reducidos

Pago por uso: Las empresas y usuarios solo pagan por los recursos que realmente utilizan, lo que reduce la necesidad de comprar y mantener servidores físicos.

Reducción de gastos en infraestructura: Las empresas pueden evitar el alto costo inicial de infraestructura propia.

Startups tecnológicas pueden lanzar aplicaciones usando servicios en la nube, sin incurrir en gastos altos en infraestructura de TI.

TIPOS DE SERVICIOS EN LA NUBE

IaaS (Infraestructura como Servicio)

Ofrece infraestructura física (como servidores, redes, y almacenamiento) en forma virtual a través de internet.

- **Control total sobre los recursos:** Los usuarios pueden configurar, personalizar y gestionar sus propios entornos de cómputo.
- **Flexibilidad:** Adaptable para una amplia gama de aplicaciones, desde pruebas y desarrollo hasta grandes despliegues de producción.

01

AMAZON EC2 (ELASTIC COMPUTE CLOUD)

Proporciona máquinas virtuales que pueden ejecutarse con diferentes configuraciones de CPU y memoria.

02

GOOGLE COMPUTE ENGINE

Facilita el uso de máquinas virtuales en la nube y la capacidad de escalarlas según las necesidades de la aplicación

TIPOS DE SERVICIOS EN LA NUBE

PaaS (Plataforma como Servicio)

Brinda una plataforma de desarrollo y despliegue que permite a los desarrolladores construir, probar y lanzar aplicaciones sin gestionar directamente la infraestructura subyacente.

- **Productividad mejorada:** Los desarrolladores pueden centrarse en el código y la lógica de negocio en lugar de en la administración de servidores.
- **Escalabilidad y despliegue rápido:** La plataforma se ocupa de la gestión de la infraestructura, lo que permite un despliegue rápido de aplicaciones.

01

GOOGLE APP ENGINE:

Permite a los desarrolladores crear aplicaciones escalables sin preocuparse por el manejo de servidores.

02

MICROSOFT AZURE APP SERVICES

Plataforma de desarrollo para aplicaciones web, móviles y de API.

TIPOS DE SERVICIOS EN LA NUBE

SaaS (Software como Servicio)

Ofrece software y aplicaciones a través de internet. Los usuarios pueden acceder a aplicaciones a través de navegadores web, eliminando la necesidad de instalar o mantener software en dispositivos individuales.

- **Mantenimiento y actualización centralizados:** Los proveedores de SaaS gestionan las actualizaciones, la seguridad y el mantenimiento de la aplicación.
- **Facilidad de acceso:** Las aplicaciones son accesibles desde cualquier dispositivo con acceso a internet, facilitando la colaboración y el trabajo remoto.

01

GOOGLE WORKSPACE

Herramientas de productividad como Gmail, Google Docs y Google Sheets, todas accesibles en la nube.

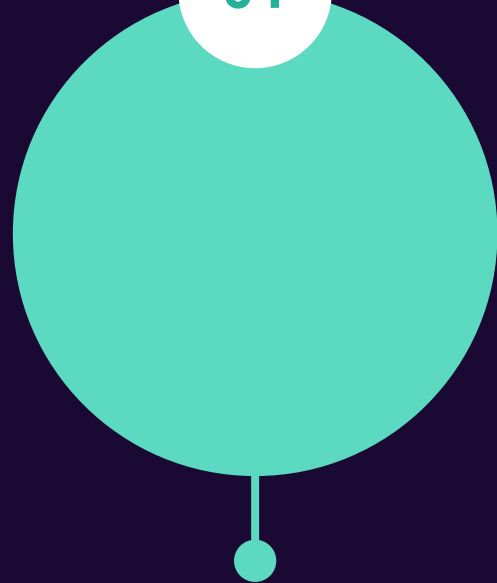
02

MICROSOFT OFFICE 365

Incluye herramientas como Word, Excel y PowerPoint en un modelo de suscripción en la nube.

MODELOS DE IMPLEMENTACIÓN DE LA NUBE

01

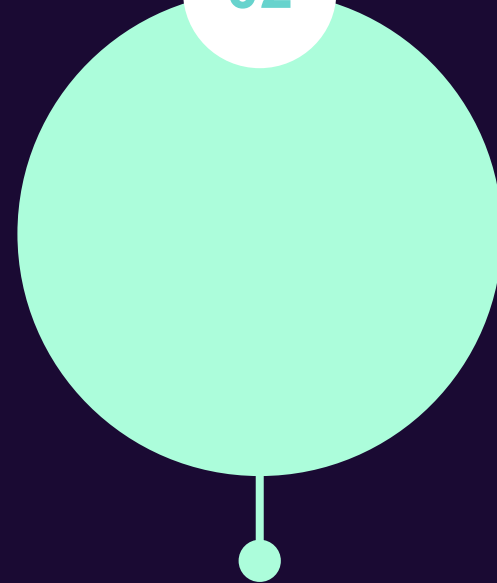


NUBE PÚBLICA

- Proporcionada por empresas externas (ej., Amazon, Google, Microsoft) y está disponible para cualquier organización o usuario.

Amazon Web Services (AWS) ofrece almacenamiento, cómputo y bases de datos en un entorno accesible para cualquier cliente.

02

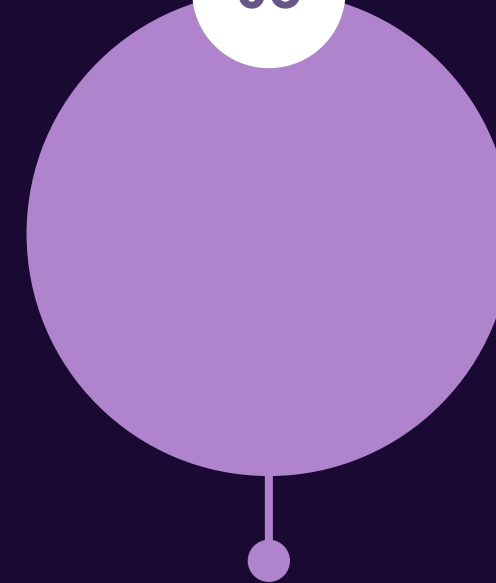


NUBE PRIVADA

- Infraestructura reservada para una única organización, que ofrece mayor control y seguridad sobre los datos y aplicaciones.

Una gran corporación financiera puede tener su propia nube privada para garantizar la seguridad de la información sensible.

03

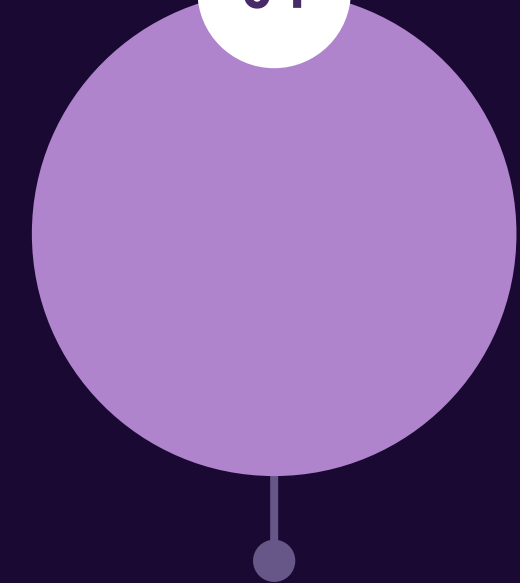


NUBE HÍBRIDA

- Combinación de nube pública y privada, permitiendo que las organizaciones compartan datos y aplicaciones entre ambos entornos.

Empresas que utilizan una nube privada para almacenar datos sensibles y una nube pública para ejecutar aplicaciones menos críticas.

04



MULTINUBE

- Uso de varios servicios de nube pública de diferentes proveedores para evitar la dependencia de un solo proveedor y aprovechar servicios específicos.

Una empresa puede usar Amazon S3 para almacenamiento y Google Cloud Platform para análisis de Big Data.

SISTEMAS OPERATIVOS ESPECÍFICOS PARA LA NUBE

OSV

OSv

Es un sistema operativo ligero y optimizado para ejecutar aplicaciones en entornos virtualizados, diseñado específicamente para **maximizar la eficiencia** en plataformas de nube.

- **Optimización para entornos virtuales:** OSv está diseñado específicamente para máquinas virtuales y contenedores, eliminando funciones innecesarias como la gestión de múltiples usuarios y servicios de red complejos.
- **Sistema de archivos en memoria:** Utiliza un sistema de archivos en memoria que minimiza las operaciones de entrada/salida (I/O), lo que reduce la latencia y mejora el rendimiento de las aplicaciones en la nube.
- **Compatibilidad con lenguajes de alto nivel:** Soporta aplicaciones en lenguajes como Java, Python y Node.js, facilitando su adopción en diversas plataformas.
- **Uso en aplicaciones:** Es ideal para aplicaciones de procesamiento intensivo en la nube, como bases de datos y servicios web de baja latencia.

OSv. 
designed for the cloud

Casos de Uso

- **Apache Cassandra en la nube:** OSv se utiliza para ejecutar aplicaciones como Apache Cassandra, donde su diseño ligero permite una mayor eficiencia en el uso de recursos.
- **Servicios de microservicios:** En arquitecturas de microservicios, OSv ofrece un entorno que reduce el consumo de recursos y el tiempo de inicio, permitiendo escalabilidad rápida en entornos de nube.

SISTEMAS OPERATIVOS ESPECÍFICOS PARA LA NUBE

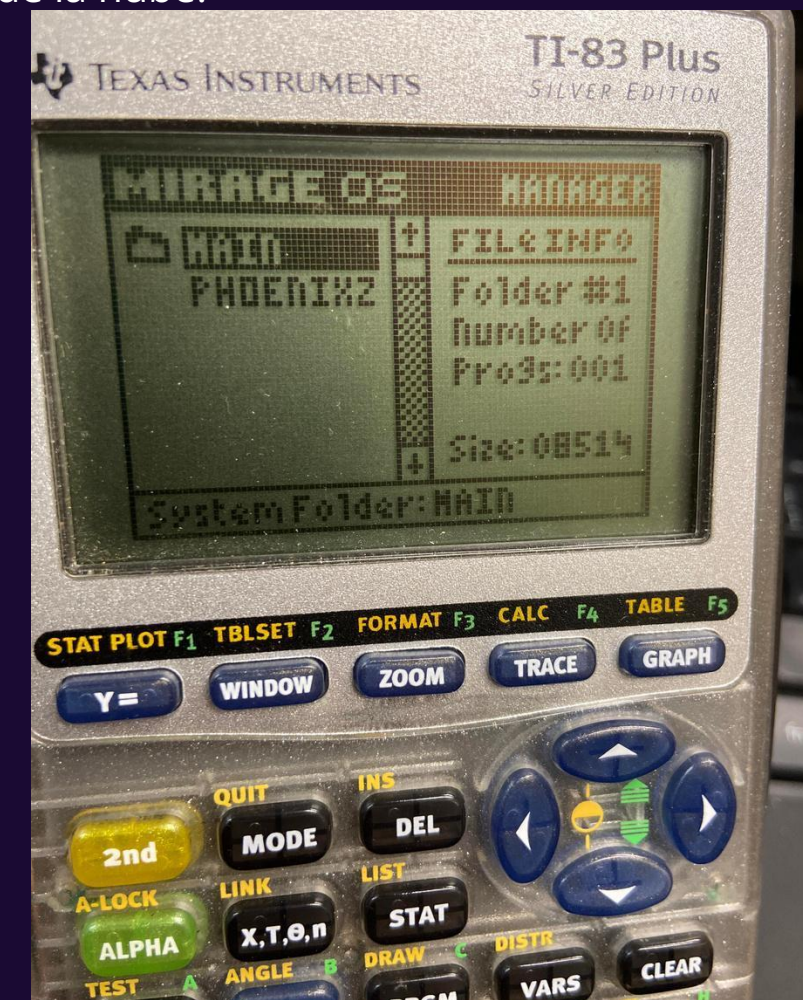
MirageOS

MirageOS es un sistema operativo basado en [unikernel](#) que permite compilar aplicaciones junto con solo los componentes esenciales del sistema operativo, eliminando así la necesidad de un sistema operativo completo.

- **Arquitectura de Unikernel:** MirageOS empaqueta la aplicación con solo los elementos esenciales para su ejecución, lo que reduce la huella de recursos y la superficie de ataque en comparación con sistemas operativos tradicionales.
- **Diseño seguro:** Su naturaleza de unikernel proporciona un alto nivel de seguridad al eliminar funcionalidades innecesarias, lo que disminuye las posibilidades de vulnerabilidades.
- **Aplicaciones de red especializadas:** Se utiliza frecuentemente en servicios de red, como firewalls, balanceadores de carga y proxies en la nube, donde se requieren bajo consumo de recursos y alta seguridad.
- **Lenguaje de programación OCaml:** MirageOS aprovecha las características de OCaml, lo que facilita la verificación de tipo estático y la seguridad de la memoria, aspectos cruciales en aplicaciones de red.

Casos de Uso

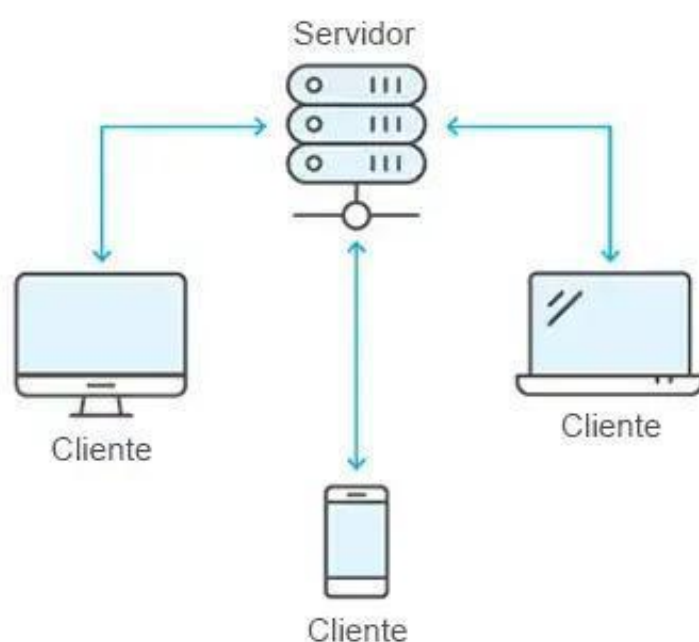
- **Firewalls y redes virtuales:** Utilizado en entornos de red segura, como firewalls y servidores DNS, que necesitan alta eficiencia y bajo consumo.
- **Aplicaciones de IoT:** Ideal para dispositivos IoT, donde los recursos son limitados y la seguridad es crítica, permitiendo que dispositivos remotos se conecten de manera segura a través de la nube.



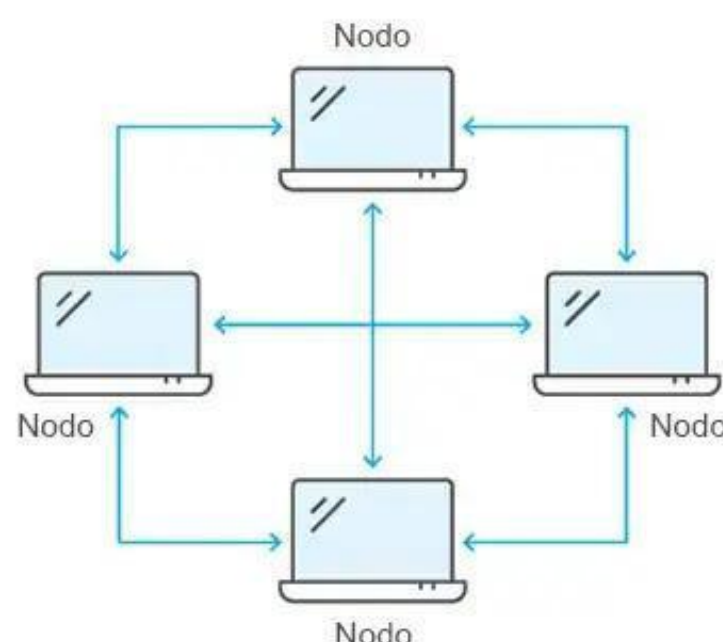
MODELOS DE ARQUITECTURA EN SISTEMAS DISTRIBUIDOS

Cliente-Servidor vs P2P

Red Cliente-Servidor



Red P2P



01

CLIENTE SERVIDOR

La red cliente-servidor está compuesta por al menos un servidor central que controla la red y una serie de dispositivos cliente que se conectan al servidor para realizar tareas específicas.

Ejemplos típicos: *servidores web y bases de datos.*

02

PEER-TO-PEER (P2P)

Descripción del modelo en el que cada nodo actúa como cliente y servidor, permitiendo que compartan recursos de manera descentralizada.

Ejemplos comunes: *redes de intercambio de archivos y blockchain.*

VENTAJAS DE LOS SISTEMAS OPERATIVOS PARA LA NUBE

EFICIENCIA

- **Uso mínimo de recursos:** Estos sistemas operativos ligeros eliminan servicios y componentes innecesarios, lo que reduce el consumo de CPU y memoria, optimizando el rendimiento en entornos virtualizados.
- **Reducción de latencia:** Están diseñados para utilizar sistemas de archivos en memoria y optimizar la entrada/salida (I/O), lo que es crucial para aplicaciones en tiempo real y de baja latencia.
- **Escalabilidad rápida:** Su estructura ligera permite que las aplicaciones escalen rápidamente en respuesta a la demanda

SEGURIDAD

- **Menor superficie de ataque:** Al reducir el sistema operativo solo a los componentes esenciales, se elimina la mayoría de las vulnerabilidades asociadas con los sistemas operativos tradicionales.
- **Control total sobre el entorno:** Los desarrolladores pueden construir aplicaciones con componentes específicos, eliminando funciones innecesarias que podrían representar riesgos de seguridad.

PORTABILIDAD

- **Facilidad de migración entre nubes:** Estos sistemas operativos son altamente portables, lo que permite que las aplicaciones se ejecuten en múltiples plataformas en la nube con mínimas modificaciones.
- **Compatibilidad con contenedores y VMs:** Su diseño ligero los hace ideales para contenedores y máquinas virtuales, facilitando el despliegue en diversos entornos de nube, ya sea pública o privada.

DESAFÍOS DE **SEGURIDAD** EN LOS SISTEMAS EN LA NUBE

Principales Amenazas



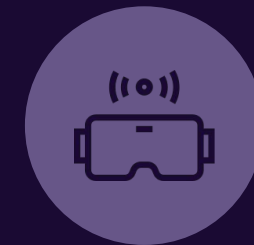
ACCESOS NO AUTORIZADOS

Los accesos no autorizados son una grave amenaza para la confidencialidad de los datos en la nube, ya que los atacantes pueden acceder a cuentas de usuario o sistemas mediante credenciales robadas o vulnerabilidades de seguridad. **Ejemplo:** **Phishing** o **ataques de ingeniería social** que buscan obtener contraseñas de usuarios.



AMENAZAS INTERNAS

Las amenazas internas surgen de empleados o administradores que pueden actuar con intenciones maliciosas o **comprometer** la seguridad por descuido o falta de capacitación. Por ejemplo, un empleado puede exponer accidentalmente datos sensibles al configurarlos incorrectamente en un servicio en la nube.



ATAQUES DE DENEGACIÓN DE SERVICIO (DDOS)

Los ataques DDoS tienen como objetivo saturar los recursos de un servicio en la nube, lo que puede resultar en la interrupción total de los servicios y afectar tanto la **disponibilidad** como la confianza del cliente. Un ejemplo es un ataque DDoS que inunda un servidor web con tráfico falso, volviendo el servicio inalcanzable para los usuarios legítimos.

¿Sabías qué?

Los desafíos de seguridad en los sistemas en la nube son diversos y complejos, abarcando amenazas externas e internas, así como la necesidad de cumplir con regulaciones de privacidad.

MECANISMOS DE SEGURIDAD

CIFRADO DE DATOS

Implementar cifrado en tránsito y en almacenamiento protege la **información sensible**, asegurando que, incluso si los datos son interceptados, no sean legibles sin la clave de cifrado.

FIREWALLS Y REDES SEGURAS

Configurar firewalls y otras medidas de seguridad de red para restringir el acceso únicamente a usuarios o sistemas autorizados. Esto proporciona una **capa adicional** de defensa contra accesos no autorizados y ataques externos.

MONITOREO Y DETECCIÓN DE AMENAZAS

Emplear inteligencia artificial y machine learning para monitorear patrones de comportamiento en tiempo real y detectar **actividades inusuales** que podrían indicar un ataque o vulnerabilidad. Facilita respuestas rápidas a incidentes de seguridad y la identificación proactiva de amenazas antes de que causen daños.



Actividad: Kahoot Sistemas en la Nube

CONCEPTOS CLAVE APRENDIDOS

- **Sistemas Distribuidos:** Son conjuntos de nodos interconectados que cooperan para un fin común, ofreciendo transparencia, escalabilidad, confiabilidad y concurrencia.
- **Sistemas en la Nube:** Proveen acceso remoto a recursos escalables bajo demanda, con modelos como IaaS, PaaS y SaaS, y despliegues públicos, privados, híbridos o multinube.
- **Sistemas Operativos para la Nube (OSv y MirageOS):** OSv y MirageOS son sistemas ligeros y optimizados para la nube, centrados en eficiencia, seguridad y despliegue rápido en entornos virtualizados.

VALOR DE LA SEMANA

- **Trabajo en equipo:** Cuando se trata de sistemas distribuidos, hablamos de que se logra más teniendo muchos equipos trabajando juntos, esto aplica para todo, no solo la programación, es importante saber adaptarse a un equipo y colaborar para lograr metas comunes, y a veces, individuales.

REFERENCIAS

What is a distributed system?

<https://www.atlassian.com/microservices/microservices-architecture/distributed-architecture>

Cloud Operative System

<https://wasabi.com/glossary/cloud-operating-system-cloud-os>

MirageOS

<https://mirage.io/>

¡GRACIAS POR
LA ATENCIÓN!

¿Dudas?