

Universidad Técnica de Ambato



Faculta de Ingeniera en Sistemas, Electrónica e Industrial



Fundamentos de las Aplicaciones Distribuidas

Albán Chávez Melanie Elizabeth

Sexto Software A

Ing. José Caiza Mg.

28 de marzo de 2025



1. Introducción

Los sistemas distribuidos han experimentado un notable desarrollo en las últimas décadas, convirtiéndose en una base esencial para el software moderno. Se componen de múltiples elementos independientes que trabajan en conjunto para formar un sistema unificado, permitiendo compartir recursos, optimizar el rendimiento y proporcionar servicios escalables y accesibles. La distribución funcional, determinada por el ámbito de aplicación, junto con el balanceo de carga, son factores clave que impulsan su adopción.

No obstante, estos sistemas también presentan retos significativos, como la concurrencia, la seguridad, la heterogeneidad y la falta de un contexto global, lo que los convierte en soluciones complejas pero imprescindibles en la actualidad. Su evolución ha dado lugar a arquitecturas avanzadas, destacándose la computación en la nube, que permite el acceso a recursos informáticos a través de internet sin necesidad de infraestructura propia. Este modelo optimiza costos, mejora la escalabilidad y brinda mayor flexibilidad, facilitando la implementación de servicios distribuidos.

Además, los sistemas distribuidos abarcan componentes esenciales como los modelos cliente-servidor, los mecanismos de comunicación (RPC, REST) y el papel de los sistemas operativos en estos entornos. También incluyen la gestión de bases de datos distribuidas, el trabajo colaborativo (GroupWare) y los modelos de integración de aplicaciones, consolidando su relevancia en el desarrollo de tecnología moderna002E

2. Marco teórico

Los sistemas distribuidos representan una evolución fundamental en la arquitectura de sistemas computacionales, donde múltiples nodos independientes colaboran para presentarse como un sistema unificado [1]. Esta arquitectura se caracteriza por su capacidad para ocultar la distribución física de recursos, permitiendo a los usuarios interactuar con el sistema como si fuera una sola entidad.

La transparencia en estos sistemas abarca varios aspectos: acceso, ubicación, migración, réplica, concurrencia, fallos y persistencia [1]. Esta característica es crucial para lograr la ilusión de un sistema único. La escalabilidad se manifiesta en la capacidad de adaptarse a incrementos en la carga de trabajo mediante la adición de nuevos nodos, manteniendo o mejorando el rendimiento [1].

Un aspecto crítico es la concurrencia, donde múltiples procesos acceden simultáneamente a recursos compartidos, requiriendo mecanismos sofisticados de sincronización para evitar condiciones de carrera y garantizar la consistencia de los datos [1]. La tolerancia a **fallos** se logra mediante redundancia y protocolos de recuperación que permiten al sistema continuar operando incluso cuando algunos componentes fallan [1].



Ejemplos paradigmáticos incluyen la propia Internet, sistemas de bases de datos distribuidas como Cassandra, plataformas blockchain y servicios de computación en la nube como AWS [1]. Estos sistemas demuestran la versatilidad y aplicabilidad de los principios de distribución en diversos contextos.

2.1 Evolución Histórica y Arquitecturas

La evolución de los sistemas distribuidos ha seguido una trayectoria marcada por avances tecnológicos y cambios en los paradigmas computacionales [1]. Desde los mainframes centralizados de los años 60-70, donde todas las operaciones se realizaban en una computadora central, se transitó hacia modelos más distribuidos.

El modelo cliente-servidor emergió como dominante en los 80-90, estableciendo una clara separación de responsabilidades entre proveedores y consumidores de servicios [1]. Este modelo permitió mayor escalabilidad y especialización de componentes. Posteriormente, arquitecturas más descentralizadas como peer-to-peer ganaron relevancia, eliminando la dependencia de servidores centrales [1].

En la era moderna, la computación en la nube ha redefinido los sistemas distribuidos, ofreciendo recursos elásticos bajo demanda [1]. Paralelamente, tecnologías como contenedores (Docker, Kubernetes) y microservicios han permitido construir sistemas más modulares y escalables [1].

2.2 Componentes Clave

Los sistemas operativos distribuidos juegan un papel fundamental al abstraer la complejidad de la distribución [1]. Gestionan recursos compartidos, planifican procesos distribuidos e implementan mecanismos de comunicación interprocesos. La memoria compartida distribuida (DSM) crea la ilusión de un espacio de direcciones unificado a través de múltiples máquinas [1].

En el modelo cliente-servidor, los servidores especializados (de archivos, bases de datos, aplicaciones) ofrecen servicios a múltiples clientes [1]. Este modelo favorece la seguridad centralizada y la administración unificada de recursos. Las bases de datos distribuidas implementan estrategias como fragmentación y replicación para optimizar el acceso a datos distribuidos geográficamente [1].

2.3 Aplicaciones y Desafíos

El Groupware representa una clase importante de aplicaciones distribuidas que facilitan la colaboración en tiempo real [1]. Estas herramientas deben resolver desafíos como la sincronización de estados y el control de concurrencia en entornos con alta latencia.



La computación en la nube ha democratizado el acceso a infraestructura distribuida, ofreciendo modelos como IaaS, PaaS y SaaS [1]. Sin embargo, plantea retos en seguridad, privacidad y dependencia de proveedores externos.

Entre los desafíos persistentes se encuentran garantizar la consistencia en entornos distribuidos (teorema CAP), manejar la heterogeneidad de componentes y mantener la seguridad en sistemas altamente interconectados [1].

3. Herramientas empleadas

- **Plataforma educativa:** Como entorno virtual de aprendizaje para la gestión de contenidos y actividades formativas.
- **Canva:** Herramienta de diseño gráfico empleada para la creación de material visual e infografías.
- **Inteligencia Artificial:** Sistemas de IA utilizados como apoyo en la investigación y generación de contenidos.
- **Recursos Bibliográficos:** Documentación especializada, libros y artículos académicos que fundamentan la investigación.

4. Desarrollo

A continuación, se presentará una infografía que resume de manera detallada toda la información sobre las aplicaciones distribuidas.



5. Conclusiones

Los sistemas distribuidos representan el paradigma fundamental de la computación moderna, permitiendo el desarrollo de aplicaciones escalables, resilientes y de alto rendimiento. Su capacidad para integrar múltiples nodos como una unidad coherente mediante transparencia, concurrencia controlada y tolerancia a fallos ha revolucionado áreas como la nube, blockchain y el trabajo colaborativo, aunque plantea desafíos técnicos en sincronización, seguridad y gestión de la consistencia distribuida.

Su arquitectura descentralizada y capacidad de aprovechar recursos geográficamente dispersos los posiciona como solución ideal para aplicaciones que demandan alta disponibilidad, escalabilidad global y procesamiento masivo de datos en tiempo real.

6. Referencias Bibliográficas

[1] P. Bazán et al., *Aplicaciones, servicios y procesos distribuidos: Una visión para la construcción de software*, Facultad de Informática, Universidad Nacional de La Plata, 2025.

Aplicaciones, Sistemas y Operaciones Distribuidas

Un sistema distribuido es un conjunto de computadoras interconectadas que trabajan juntas para ofrecer servicios y procesamiento de datos de manera coordinada.



Características

Transparencia: Oculta la distribución física a los usuarios.

Escalabilidad: Puede crecer sin afectar el rendimiento.

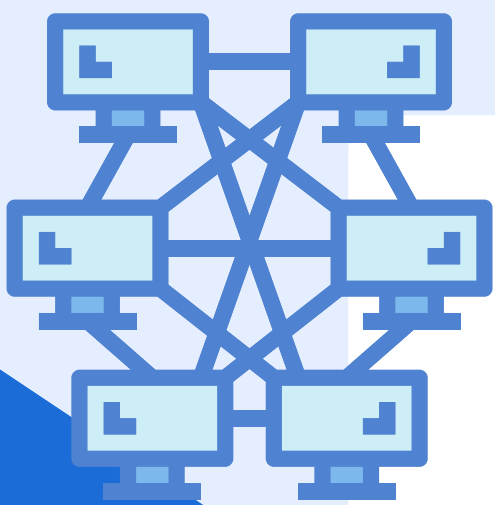
Concurrencia: Varios procesos pueden ejecutarse al mismo tiempo.

Tolerancia a fallos: Puede seguir operando a pesar de errores en algunos nodos.



Ejemplos

Internet, bases de datos distribuidas, blockchain, computación en la nube, videojuegos en línea.

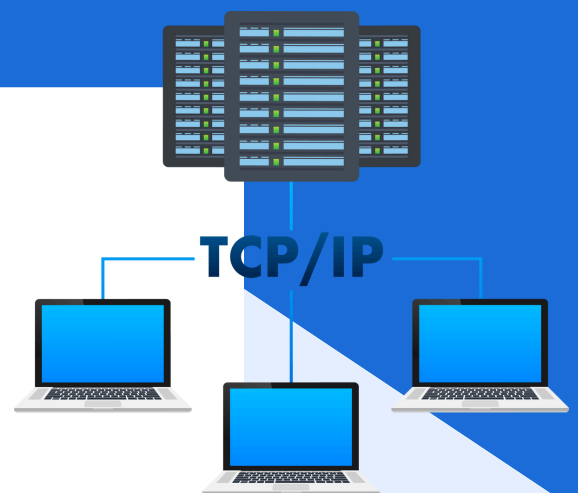


Modelos arquitectónicos

Centralizado: Un servidor principal gestiona las solicitudes.

Descentralizado: No hay un único servidor central.

Híbrido: Combinación de ambos modelos.

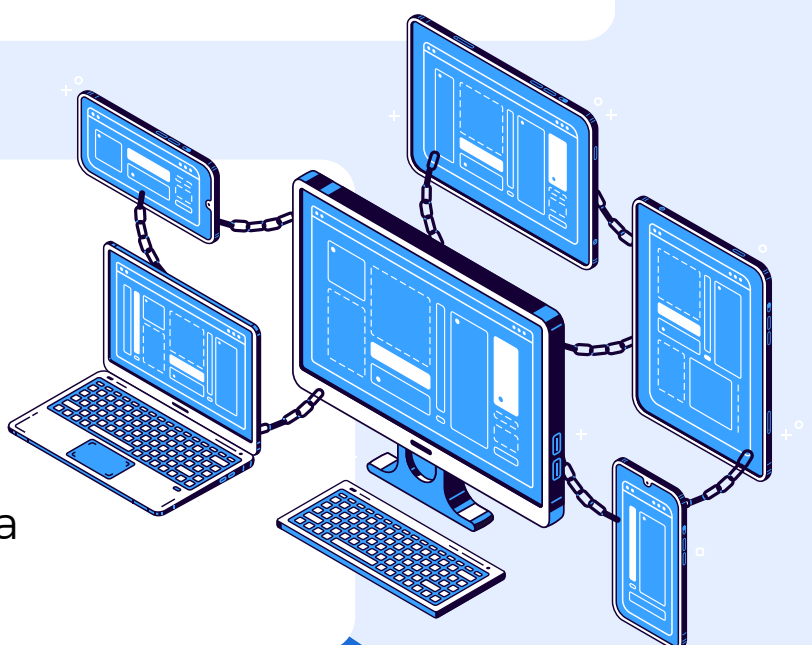


Evolución

Años 60-70: Computación centralizada (mainframes).

Años 80-90: Computación cliente-servidor y redes locales.

Años 2000+: Computación en la nube y microservicios.



Distribución y Sistemas Operativos



Relación entre sistemas operativos y sistemas distribuidos

Un sistema operativo en un entorno distribuido gestiona:

- Recursos compartidos (CPU, memoria, almacenamiento).
- Procesos concurrentes en diferentes máquinas.
- Sincronización entre procesos en red.

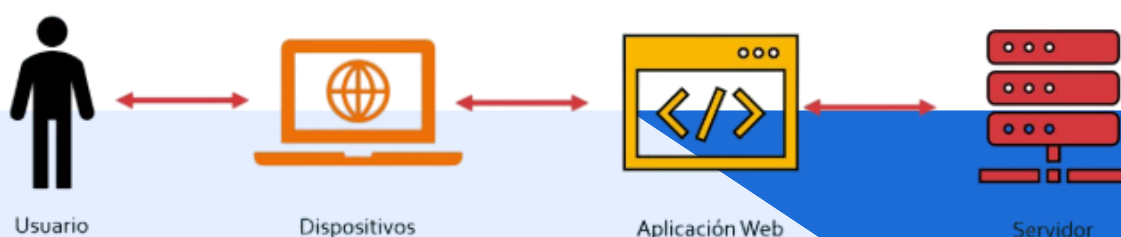
Cliente y Servidor en Sistemas Distribuidos

Interacción donde el cliente envía solicitudes de servicio y el servidor las procesa y responde, facilitando la comunicación y el intercambio de datos.

Características

Cliente: Dispositivo que solicita servicios (Ej. navegador web).

Servidor: Proveedor de servicios (Ej. servidor de correo).



Tipos de Servidores

Servidor de archivos: Almacena y distribuye archivos.

Servidor de correo: Gestiona correos electrónicos.

Servidor de aplicaciones: Hospeda software empresarial.

Servidor de bases de datos: Maneja consultas SQL.



Ventajas del modelo cliente-servidor

- Mejor distribución de carga.
- Seguridad centralizada.
- Escalabilidad en grandes infraestructuras.



Arquitecturas Distribuidas



Modelos de Distribución

- **Peer-to-Peer:** No hay servidores fijos, todos los nodos pueden actuar como clientes y servidores (Ej. Torrents).
- **Computación en malla:** Varios servidores colaboran en la ejecución de tareas (Ej. supercomputación).
- **Computación en la Nube:** Se accede a recursos de servidores remotos en internet.



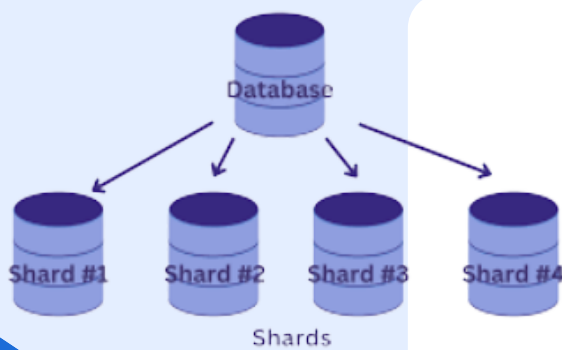
Comunicación entre nodos

- **Mensajes (RPC, REST, gRPC):** Comunicación mediante paquetes de datos.
- **Memoria compartida:** Los nodos acceden a un mismo espacio de memoria.
- **Eventos y suscripciones:** Modelo basado en notificaciones entre sistemas.



Servidores de Base de Datos y Distribución

Un sistema de bases de datos distribuido almacena información en múltiples servidores.



Modelos de distribución de datos

- **Fragmentación:** Dividir datos en diferentes ubicaciones.
- **Replicación:** Copiar datos en varios servidores.
- **Combinación:** Mezcla de fragmentación y replicación.

Ejemplos

Google Spanner, Amazon DynamoDB, Cassandra.



Amazon DynamoDB



Desafíos de bases de datos distribuidas

- Sincronización entre copias.
- Garantizar la consistencia y disponibilidad (Teorema CAP).

GroupWare

Software colaborativo que permite a múltiples usuarios trabajar en conjunto en tiempo real.

Ejemplos

- **Google Docs:** Edición de documentos en línea.
- **Slack, Microsoft Teams:** Comunicación empresarial.
- **Git:** Control de versiones para desarrollo de software.



Desafíos del software colaborativo

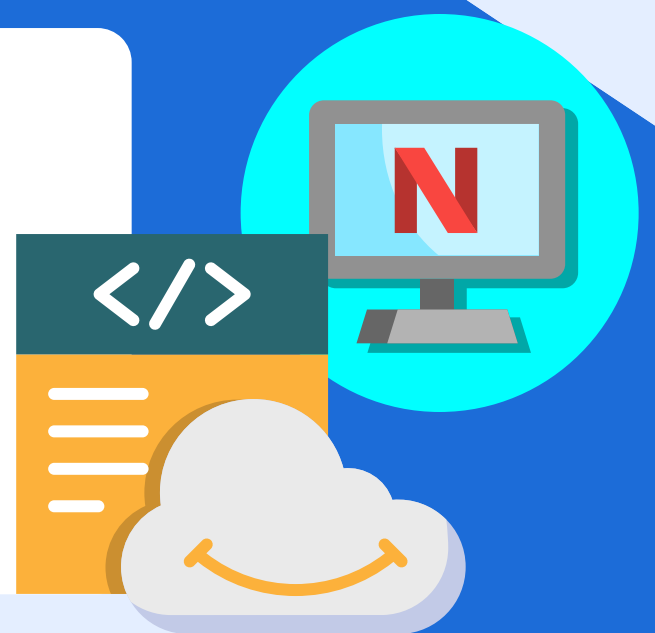
- Control de concurrencia.
- Latencia en la sincronización de cambios.

Computación en la Nube

Modelo que permite acceder a servicios informáticos a través de internet.

Modelos de servicio en la nube

- **IaaS (Infraestructura como Servicio):** Servidores virtuales (Ej. AWS EC2).
- **PaaS (Plataforma como Servicio):** Entornos de desarrollo listos (Ej. Heroku).
- **SaaS (Software como Servicio):** Aplicaciones listas para el usuario (Ej. Google Drive, Netflix).



Beneficios

- Reducción de costos.
- Escalabilidad automática.
- Acceso desde cualquier parte del mundo.

Desafíos

- Dependencia de internet.
- Riesgos de seguridad en la transferencia de datos.



Bibliografía

Bazán, P., et al. (2017). Aplicaciones, servicios y procesos distribuidos: una visión para la construcción de software. UNLP .



5. Conclusiones

Los sistemas distribuidos representan el paradigma fundamental de la computación moderna, permitiendo el desarrollo de aplicaciones escalables, resilientes y de alto rendimiento. Su capacidad para integrar múltiples nodos como una unidad coherente mediante transparencia, concurrencia controlada y tolerancia a fallos ha revolucionado áreas como la nube, blockchain y el trabajo colaborativo, aunque plantea desafíos técnicos en sincronización, seguridad y gestión de la consistencia distribuida.

Su arquitectura descentralizada y capacidad de aprovechar recursos geográficamente dispersos los posiciona como solución ideal para aplicaciones que demandan alta disponibilidad, escalabilidad global y procesamiento masivo de datos en tiempo real.

6. Referencias Bibliográficas

[1] P. Bazán et al., *Aplicaciones, servicios y procesos distribuidos: Una visión para la construcción de software*, Facultad de Informática, Universidad Nacional de La Plata, 2025.