

# Sistemas Distribuidos

## Bloque I

Introducción a los Sistemas Distribuidos:  
caracterización y arquitecturas.

Tema 1.1:

Introducción a los Sistemas Distribuidos



# Índice

---

- ¿Qué es un Sistema distribuido?
- Características
- Ventajas
- Desafíos
- Arquitecturas
- Comunicación
- Tolerancia a fallos, recuperación y seguridad
- Aplicaciones actuales

# ¿Qué es un Sistema Distribuido?

---

- **Definición básica:** Conjunto de computadoras independientes que se presentan al usuario como una única computadora coherente.
- **Objetivo principal:** Conectar recursos y usuarios de manera transparente.

# Características

---

- **Concurrencia**

- Varios procesos pueden correr simultáneamente en diferentes nodos.
- Es necesario coordinar acciones y evitar conflictos.

- **Escalabilidad**

- Capacidad para crecer y gestionar un aumento en usuarios y tareas.
- Puede ser vertical (añadir más recursos a un nodo) o horizontal (añadir más nodos).

- **Fallos independientes**

- Un fallo en un nodo no debería causar la caída del sistema completo.
- La detección y recuperación son esenciales.

- **Heterogeneidad**

- Diversidad de hardware, sistemas operativos, redes y aplicaciones.
- El middleware ayuda a unificar esta diversidad.

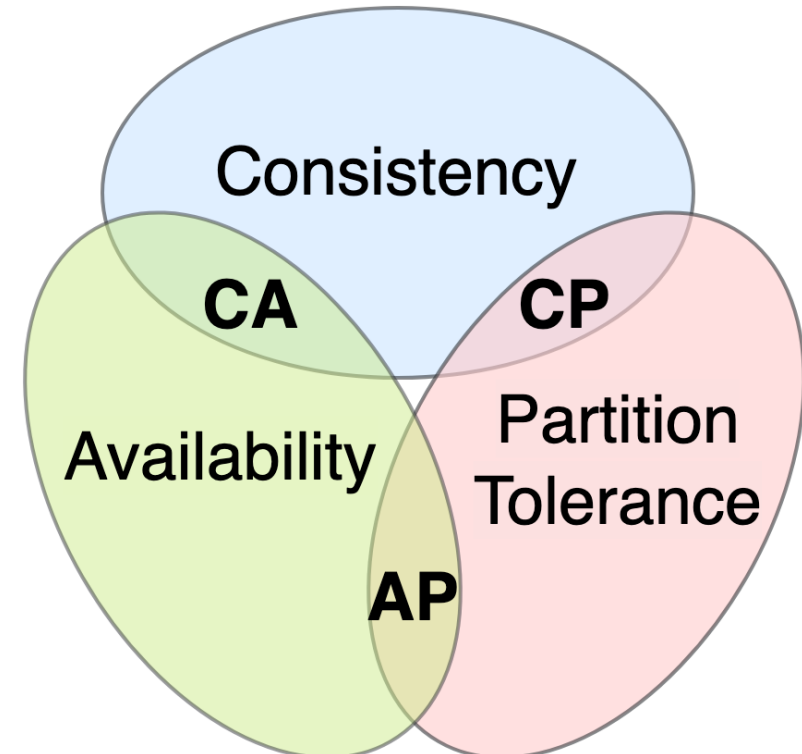
# Ventajas

---

- Compartir recursos
- Tolerancia a fallos y alta disponibilidad
- Escalabilidad
- Flexibilidad y adaptabilidad
- Reducción de costes y aumento de rendimiento

# Desafíos

- Coordinación y comunicación entre nodos
- Gestión de fallos y recuperación
- Seguridad y privacidad
- Consistencia y replicación
- Diseño y desarrollo



Fuente: [https://en.wikipedia.org/wiki/CAP\\_theorem](https://en.wikipedia.org/wiki/CAP_theorem)

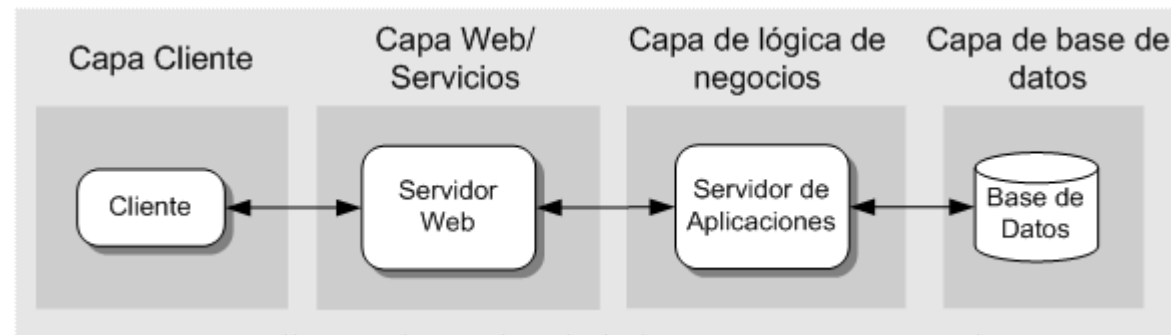
# Arquitecturas

- **Modelo Cliente-Servidor**

- Un servidor provee recursos o servicios, y los clientes consumen esos servicios.
- Variantes:
  - Grueso delgado (Dependiendo de dónde resida la lógica de la aplicación).
  - P2P (Cada nodo puede ser tanto cliente como servidor).

- **Arquitectura en Capas**

- Capas separan las responsabilidades y simplifican la gestión y el desarrollo.
- OSI Model con sus 7 capas desde física hasta aplicación.



Fuente: <https://geeks.ms/jkpelaez/2009/05/30/arquitectura-basada-en-capas/>

- **Arquitectura Basada en Objetos**

- **Objetos distribuidos:** Entidades que encapsulan datos y métodos y pueden ser invocados remotamente.
- **CORBA** (Arquitectura de Objetos de Broker Común) es un estándar que permite la comunicación entre objetos en diferentes sistemas.
- **Java RMI:** Permite a un objeto invocar métodos de otro objeto en una máquina distinta.



# Comunicación

- **Protocolos y Mensajes:**

- HTTP, FTP, SMTP son ejemplos de protocolos.
- Un protocolo define las reglas para el intercambio de mensajes.

- **Invocación Remota:**

- **RPC (Remote Procedure Call):** Permite que programas llamen a procedimientos en otra máquina.
- **RMI (Remote Method Invocation):** Similar al RPC pero orientado a objetos.

- **Middleware:** Software que proporciona servicios para ayudar a las aplicaciones distribuidas a comunicarse y gestionar datos.

- Ejemplo: **Message Queue** (cola de mensajes) para la comunicación asíncrona entre aplicaciones.

# Tolerancia a fallos, recuperación y seguridad

- **Tipos de fallos:**

- Fallo de nodo, fallo de comunicación, fallo de software, etc.

- **Técnicas de recuperación:**

- **Check-pointing:** Guardar el estado de una aplicación en un punto específico.
- **Logging:** Registrar las operaciones para poder "reproducirlas" en caso de fallo.

- **Técnicas de cifrado:**

- Simétrico (mismo cifrado y descifrado de clave) y Asimétrico (clave pública y privada).

- **Certificados:**

- Proporcionados por una Autoridad de Certificación, garantizan la identidad de un nodo.

# Aplicaciones actuales – Cloud Computing

- **Definición:** Entrega de servicios informáticos a través de Internet.
- **Tipos:** IaaS, PaaS, SaaS.
- **Proveedores líderes:** Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure.
- **Aplicaciones:** Almacenamiento en la nube, bases de datos en la nube, Inteligencia Artificial, Machine Learning.

# Aplicaciones actuales – BBDD

---

- **Definición:** Sistemas que almacenan datos en múltiples ubicaciones, pero son tratados como una única entidad.
- **Ventajas:** Tolerancia a fallos, alta disponibilidad, escalabilidad.
- **Ejemplos:** Cassandra, MongoDB, CockroachDB.
- **Aplicaciones:** Comercio electrónico, servicios financieros, plataformas sociales.

# Aplicaciones actuales – Blockchain

---

- **Definición:** Registro distribuido y cifrado de transacciones.
- **Características:** Transparente, inmutable, seguro.
- **Ejemplos:** Bitcoin, Ethereum.
- **Aplicaciones:** Monedas digitales, contratos inteligentes, registros de tierras.

# Aplicaciones actuales – Sistemas de archivos

- **Definición:** Almacenamiento de datos que permite el acceso concurrente desde múltiples nodos.
- **Ventajas:** Tolerancia a fallos, escalabilidad, eficiencia.
- **Ejemplos:** Hadoop Distributed FileSystem (HDFS), Google File System (GFS).
- **Aplicaciones:** Análisis de big data, almacenamiento de backup, streaming de contenidos.

# Aplicaciones actuales – Otras

- **IoT (Internet de las cosas):** Dispositivos conectados que recopilan y comparten datos.
- **Microservicios:** Aplicaciones descompuestas en servicios más pequeños e independientes.
- **Computación Edge:** Procesamiento de datos cerca de la fuente de datos, como dispositivos IoT.
- **Redes Sociales:** Servicios como Facebook y Twitter que operan en plataformas distribuidas globales.

# Sistemas Distribuidos

## Bloque I

Introducción a los Sistemas Distribuidos:  
caracterización y arquitecturas.

### Tema 1.1:

Introducción a los Sistemas Distribuidos



©2023 Autor Nicolás H. Rodríguez Uribe  
Algunos derechos reservados  
Este documento se distribuye bajo la licencia  
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons,  
disponible en  
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

Nicolás Rodríguez  
[nicolas.rodriguez@urjc.es](mailto:nicolas.rodriguez@urjc.es)

