M4 AE1 Mapa Visual del Teorema de CAP

Jonathan Vásquez 2025

Teorema de CAP y Tecnologías de Bases de Datos Distribuidas

1. ¿Qué es el Teorema de CAP?

El **teorema de CAP**, propuesto por Eric Brewer, afirma que en un sistema distribuido es imposible garantizar simultáneamente estas tres propiedades:

- **C: Consistencia**
- **A: Disponibilidad (Availability)**
- **P: Tolerancia a particiones (Partition Tolerance)**

Solo se pueden cumplir **dos de las tres** al mismo tiempo en un sistema distribuido real.

2. Definición de cada propiedad con ejemplos

- **Consistencia (C):**
  Todos los nodos ven los mismos datos al mismo tiempo.
  Ejemplo: Si compras un producto online, el stock se actualiza inmediatamente en todas las sucursales.

- **Disponibilidad (A):**
  El sistema siempre responde, incluso si algunos nodos están caídos.
  Ejemplo: Un buscador online que siempre entrega resultados, aunque no sean los más actualizados.

- **Tolerancia a Particiones (P):**
  El sistema sigue funcionando aunque se interrumpa la comunicación entre nodos.
  Ejemplo: Una app de mensajería donde los mensajes se almacenan localmente hasta que se recupere la conexión.

3. Ubicación de tecnologías reales en el triángulo CAP

Base de Datos	Clasificación CAP
MongoDB	CP (Consistencia + Partición)
Cassandra	AP (Disponibilidad + Partición)
PostgreSQL	CA (Consistencia + Disponibilidad, en entornos no distribuidos)
Redis	CP (usualmente, aunque depende del modo de configuración)

4. Casos de uso prioritarios para cada combinación

- **CP (Consistencia + Partición):**
  Ideal para sistemas financieros, bancarios o ecommerce donde los datos deben ser siempre coherentes.
  Ej: MongoDB, Redis.

- **AP (Disponibilidad + Partición):**
  Adecuado para redes sociales, chats, métricas o sistemas de análisis donde se toleran pequeñas inconsistencias.
  Ej: Cassandra.

- **CA (Consistencia + Disponibilidad):**
  Solo viable si no hay particiones (por ejemplo, en sistemas centralizados o de una sola máquina).
  Ej: PostgreSQL.

5. Conclusión

El Teorema de CAP es fundamental al elegir la base de datos más adecuada para una solución distribuida. No existe un sistema perfecto que cumpla con las tres propiedades a la vez, por lo que es vital entender el contexto del proyecto para priorizar la propiedad más crítica. Elegir la tecnología adecuada puede marcar la diferencia entre un sistema robusto y uno inestable.


Consistency (Consistencia)
C
Cada lectura recibe la escritura más reciente (o un error)
PostgreSQL
MongoDB
El sistema siempre responde, aunque la respuesta no sea la más reciente
Redis
Cassandra
El sistema sigue funcionando aunque haya fallas en la comunicación entre nodos.
A
P
Availability (Disponibilidad)
Partition Tolerance (Tolerancia a Particiones)