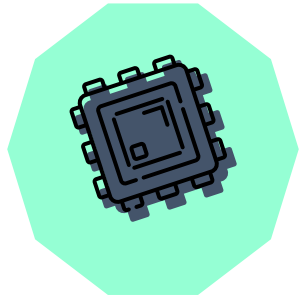


Arquitectura inmutable

Paula Suárez Prieto - UO269745
Alba Guerrero García - UO266007

Índice

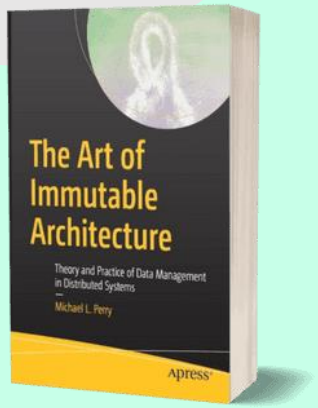
01

Introducción

Michael L Perry

Es un arquitecto de software estadaounidense reconocido por sus numerosas contribuciones entre las que destaca su tecnica de análisis y diseño basada en hechos inmutables históricos.

Ha obtenido en numerosas ocasiones el premio MVP de Microsoft



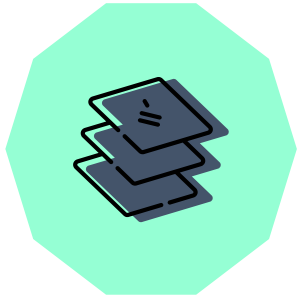
The Art of Inmutable Architecture – Michael L Perry

Aportaciones a la comunidad, desarrollando proyectos de código abierto como [Jinaga]

JINAGA

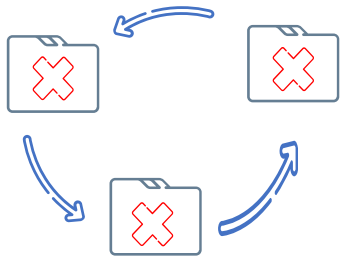
02

Arquitectura inmutable



¿Qué es la arquitectura inmutable?

La arquitectura inmutable es una restricción **autoimpuesta** sobre la modificación y eliminación de datos.



Intercambiar **información** y conocimiento sobre registros **inmutables**

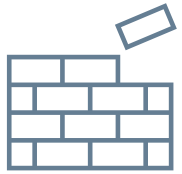
¿Entonces la arquitectura **no** puede **cambiar**?

Arquitectura inmutable vs Infraestructura inmutable

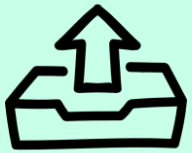
v1 Nueva v2

Infraestructura inmutable

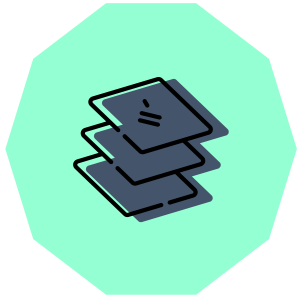
≠



Arquitectura inmutable



Patrón **Bandeja de salida**



¿Por qué arquitectura inmutable?

Auditable

Conocemos la estructura de datos del pasado

Fiable

Sabemos la información de los nodos

Consistente

Cuando se comuniquen entre sí todos están alineados

03

Aspectos a destacar

Propiedades de una arquitectura inmutable

Identidad de los registros

Comprobación de **hashes**

Independencia de ubicación

La identidad no debe estar ligada a la **ubicación**

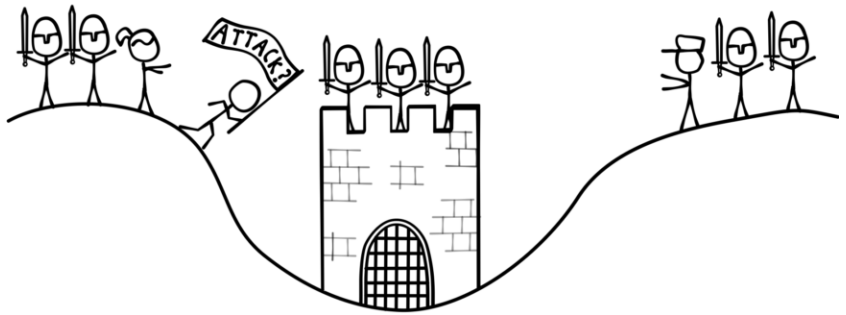
Patrón **Content Address Storage**

Hechos históricos

Sobre los que se toman **decisiones**

Problemas de los sistemas distribuidos

The Two Generals



Ejemplo

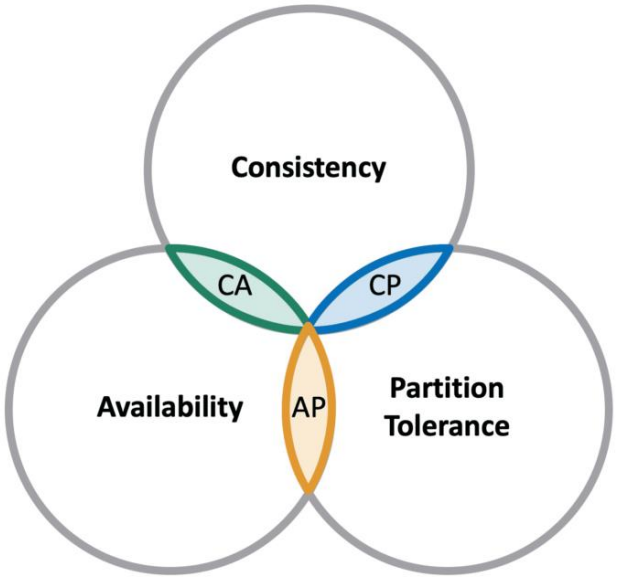
Problema técnico

↓

Problema de **negocio**

Problemas de los sistemas distribuidos

CAP Theorem

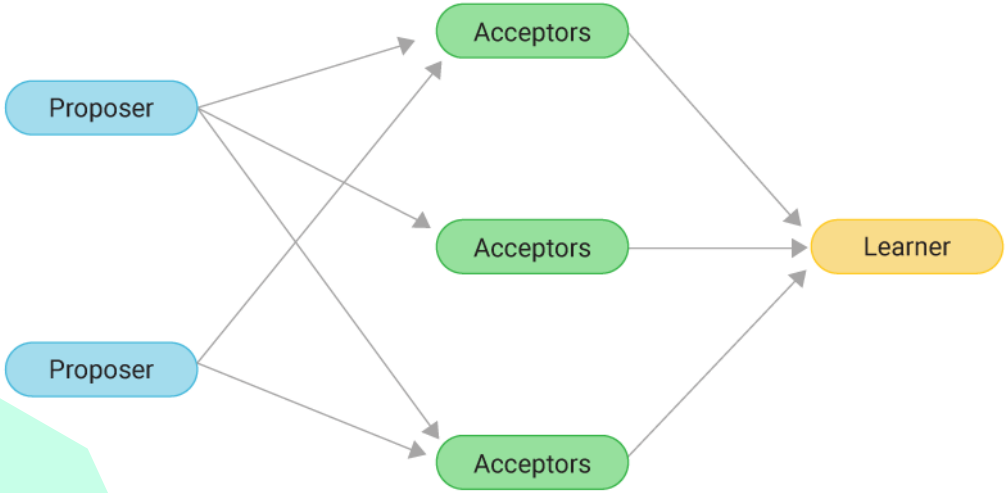


Solución

Trade-off → Fuerte consistencia eventual

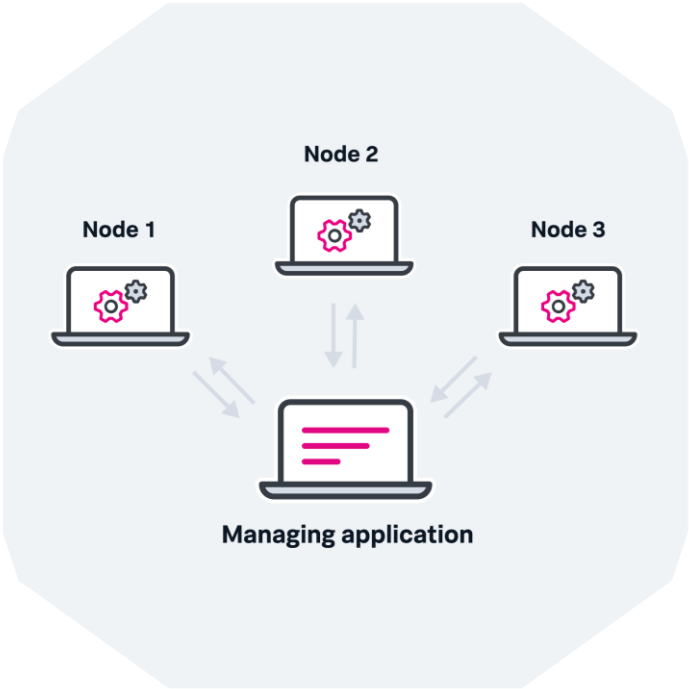
↓

Distributed Consensus Algorithms



Problemas de los sistemas distribuidos

Datos replicados



Solución

CRDT (Conflict-free Replicated Data Type)

a + b es igual a **b + a**

04

Conclusiones

Beneficios

Beneficios de una arquitectura inmutable

- **Razonar** sobre los sistemas
- **Entender** las restricciones
- Aplicaciones con **estado**

→ **Resolver problemas excepcionales**

“There are a lot of different beautiful architectures, so the **art** is really the **choice** and the **balance** and picking the correct **immutable architecture** for solving the problem we need to solve and distributed systems.”

—Michael Perry

Bibliografía

- Brown, S. A. (8 de Mayo de 2022). *The Two Generals Problem*. Recuperado el 28 de Marzo de 2023, de Hayden James: https://haydenjames.io/the-two-generals-problem/
- Chandrakant, K. (20 de Marzo de 2023). *Consensus Algorithms in Distributed Systems*. Recuperado el 28 de Marzo de 2023, de Baeldung: https://www.baeldung.com/cs/consensus-algorithms-distributed-systems
- *Episode 447: Michael Perry on Immutable Architecture*. (18 de Febrero de 2021). Recuperado el 28 de Marzo de 2023, de Software Engineering Radio: https://www.se-radio.net/2021/02/episode-447-michael-perry-on-immutable-architecture/
- *Michael Perry*. (s.f.). Recuperado el 28 de Marzo de 2023, de https://michaelperry.net/
- Ozkaya, M. (8 de Septiembre de 2021). *Outbox Pattern for Microservices Architectures*. Recuperado el 28 de Marzo de 2023, de Medium: https://medium.com/design-microservices-architecture-with-patterns/outbox-pattern-for-microservices-architectures-1b8648dfaa27
- *What is the CAP theorem?* (s.f.). Recuperado el 28 de Marzo de 2023, de IBM: https://www.ibm.com/topics/cap-theorem#:~:text=The%20CAP%20theorem%20maintains%20that,%2C%20availability%2C%20and%20partition%20tolerance

¡Gracias!

¿Alguna pregunta?