

Universidad de Monterrey



Integración de aplicaciones computacionales

Tarea 2

**Miniensayo: ¿Cloud-native o cloud-enabled? Análisis de una app real**

**Maestro:** Dr. Raúl Morales Salcedo

**Nombre:** Margarita Concepción Cuervo Citalán #581771

**Carrera:** ITC, 9no Semestre

**Día y hora del grupo de la clase:** lunes y jueves 10:00h

10 de agosto de 2025.

*Doy mi palabra de haber realizado esta actividad con integridad académica.*

El dilema entre optar por una aplicación **cloud-native** o **cloud-enabled** representa una decisión estratégica clave en proyectos de migración o desarrollo en la nube. Por ello, en este ensayo se aborda este contraste desde una definición técnica, se analiza una aplicación real y se extraen conclusiones basadas en la literatura académica actual.

Empezando por definir una aplicación **cloud-native**, nos referimos a aquellas diseñadas desde cero para la nube, utilizando microservicios, contenedores, Kubernetes, CI/CD, escalado automático y resiliencia por diseño, tal y como promueven la CNCF y estándares académicos recientes (Jain, 2024).

Por otro lado, las aplicaciones **cloud-enabled** son sistemas heredados que se trasladan a la nube con modificaciones mínimas (*lift-and-shift*), sin aprovechar plenamente las capacidades nativas del entorno, lo que limita la escalabilidad, la automatización y la eficiencia (Jain, 2024).

Un ejemplo claro del enfoque cloud-native se observa en la compañía **Ericsson**, gigante en TIC, que implementó un ecosistema denominado *Application Development Platform* (ADP) para impulsar el desarrollo de aplicaciones nativas de la nube. Entre las acciones más destacadas se encuentran:

- **Adopción de microservicios empaquetados como contenedores**, orquestados en Kubernetes con Helm (Badampudi, Usman & Chen, 2024).
- **Desarrollo de un mercado interno (ADP Marketplace)** con más de 280 microservicios (50+ con alta madurez) reutilizados por 70 equipos, fomentando eficiencia, normalización y automatización (Badampudi, Usman & Chen, 2024).
- **Implementación de “design rules” automatizadas** integradas en *pipelines* CI para garantizar consistencia en nomenclaturas, métricas y registros (Badampudi, Usman & Chen, 2024).

Este caso demuestra cómo una arquitectura cloud-native escalable y reutilizable mejora la cohesión entre equipos, reduce la duplicación de esfuerzos y refuerza la gobernanza técnica del ecosistema.

Otro caso interesante es el de la plataforma educativa **Remind**, que cuenta con millones de usuarios. Originalmente operaba con una arquitectura monolítica en PaaS, lo que correspondía a un enfoque **cloud-enabled**, ya que estaba alojada en la nube pero sin aprovechar sus capacidades nativas. Posteriormente, la compañía llevó a cabo una migración hacia un modelo **cloud-native**, adoptando microservicios desplegados en Amazon ECS. Entre los logros alcanzados tras esta transformación se destacan:

- **Reducción a la mitad de la latencia** ( $\times 2$ ) en el percentil 99, mejorando la estabilidad, la visibilidad del rendimiento y la eficiencia operativa (Ajeenkya, 2022).
- **Adopción de contenedores y balanceo automático** mediante ELB, evitando sobrecargar al equipo con la gestión compleja de clústeres (Ajeenkya, 2022).

Este caso demuestra cómo una aplicación que inicialmente era cloud-enabled puede evolucionar a cloud-native, logrando mejoras significativas en escalabilidad y eficiencia.

Asimismo, estudios como el de Balalaie et al. (2015) demuestran que la migración de un monolito hacia microservicios es un proceso complejo, pero que otorga escalabilidad, disponibilidad y flexibilidad cuando el contexto lo requiere. También, casos como el del banco danés **Danske Bank** (FX Core) evidencian que la conversión de sistemas críticos a microservicios mejora notablemente la escalabilidad real (Dragoni et al., 2017).

**En conclusión**, el enfoque cloud-native, ejemplificado por Ericsson y Remind, ofrece eficiencia operativa, calidad, escalabilidad y una gobernanza técnica sólida. Las aplicaciones cloud-enabled tienen valor como solución intermedia, pero no resultan sostenibles sin una planificación para evolucionar hacia una arquitectura nativa. En contextos donde la innovación, el crecimiento y la resiliencia son prioritarios, invertir en un diseño cloud-native es más rentable a largo plazo; en cambio, si se requiere una migración inmediata, lo cloud-enabled puede servir como un puente estratégico.

---

## Referencias bibliográficas

1. Badampudi, D., Usman, M. & Chen, X. (2024). *Large-scale reuse of microservices using CI/CD and InnerSource practices – a case study (Ericsson ADP)*, Empirical Software Engineering, Springer Nature  
Link: <https://link.springer.com/article/10.1007/s10664-024-10595-w?utm>
2. Jain, P. (2024). *A Comparative Analysis of Cloud-Native, Cloud-Enabled and Cloud-Agnostic Digital Transformation*. ResearchGate. Retrieved from: [https://www.researchgate.net/publication/381301192\\_A\\_Comparative\\_Analysis\\_of\\_Cloud-Native\\_Cloud-Enabled\\_and\\_Cloud-Agnostic\\_Digital\\_Transformation](https://www.researchgate.net/publication/381301192_A_Comparative_Analysis_of_Cloud-Native_Cloud-Enabled_and_Cloud-Agnostic_Digital_Transformation)
3. Ajeenkya, S. (2022). *Microservices: Architecture and Case Study from Various Organizations*. LinkedIn. Retrieved from: <https://www.linkedin.com/pulse/microservices-architecture-case-study-from-various-suryawanshi/>
4. Balalaie et al. (2015). *Migrating to Cloud-Native Architectures Using Microservices: An Experience Report*. arXiv/IEEE. Retrieved from: <https://arxiv.org/abs/1507.08217>
5. Dragoni et al.(2017). *Microservices: Migration of a Mission Critical System (Danske Bank FX Core)*. arXiv. Retrieved from: <https://arxiv.org/abs/1704.04173>