ARCHITECTURE MODERNIZATION

Miguel Fernández Huerta

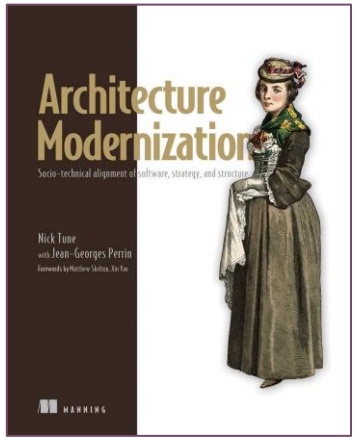
Bruno Isla Sierra

Adrián Mahía Loredo

Enlace al podcast: se-radio

ENTREVISTADOS

- **Jean-Georges Perrin.**
  - Director de innovación en ABI Data.
  - Presidente del estándar Open Data Contract.
  - Cofundador del grupo de usuarios IDA y autor de varios libros como "*Architecture Modernization*" junto a Nick.
  - IBM Champion de por vida.
  - PayPal Champion.
  - Y recientemente nombrado Data Mesh MVP.



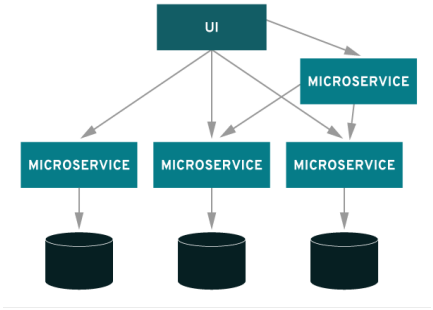
- **Nick Tune.**
  - Trabaja con líderes de producto y tecnología para:
    - Definir estrategias.
    - Modelar dominios.
    - Diseñar arquitecturas.
    - Y formar equipos de entrega continua.
  - Autor del libro "Principles and Practices of Domain-Driven Design".

PROPUESTA DE LA MODERNIZACIÓN DE ARQUITECTURA

- **Problemas:**
  - El mundo cambia, pero el software **permanece estático**.
  - El modelo de negocio de la empresa evoluciona.
  - Surgen nuevas tecnologías, patrones y arquitecturas.

¿Resultado? → El software queda obsoleto

- **Solución: la modernización.**
  - Eliminar esas desventajas de los sistemas heredados usando prácticas, métodos y reflexiones modernas.

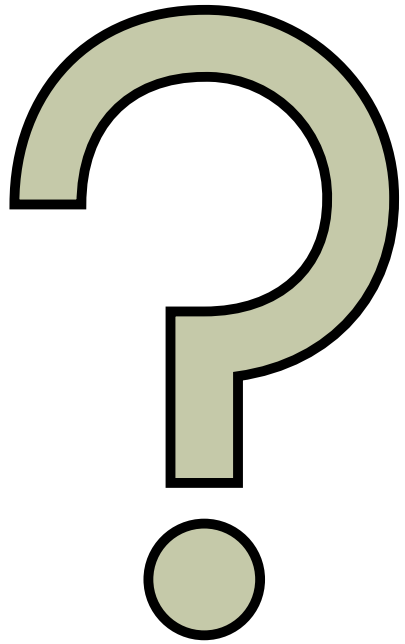


ENFOQUE DE LA MODERNIZACIÓN DE ARQUITECTURA

- Pregunta: ¿Su objetivo son **cambios a gran escala** o **mejoras incrementales**?

- **Depende del contexto**
  - Muchas empresas evitan proyectos de modernización que tarden mucho en terminar.
  - Pero si el sistema impide el crecimiento, se necesitan cambios grandes.

- **Si la deuda técnica es:**
  - **Alta** => Hay que realizar un proyecto de modernización grande (mayor esfuerzo).
  - **Baja** => El proceso de modernización se puede llevar a cabo de una forma más gradual (lo ideal).



LA MODERNIZACIÓN ES UN PROCESO CONTINUO, NO UN EVENTO ÚNICO

- Tanto si partimos teniendo una alta deuda técnica acumulada como baja…
  - La modernización **SIEMPRE** debe ser **continua** para evitar:
    - Acumulación de deuda técnica.
    - Y costos innecesarios en el futuro.

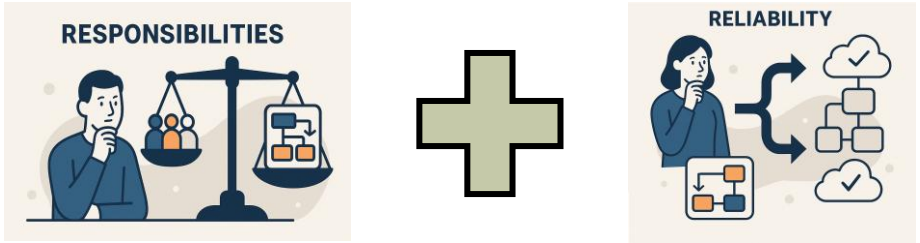
- La empresa no nos lo pone fácil…
  - Quieren que su software tenga más funcionalidades que satisfagan más necesidades de sus clientes.

Nuevas funcionalidades 

Modernizar

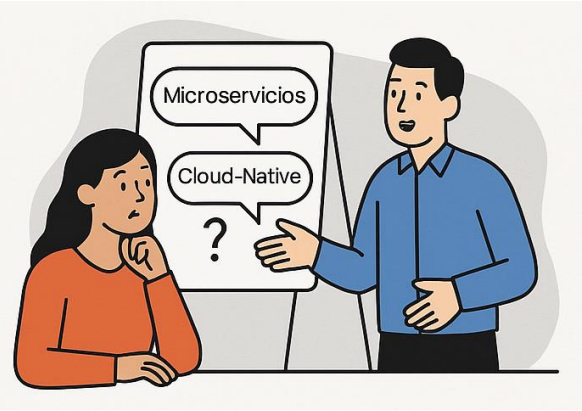
LA ARQUITECTURA DE SOFTWARE MODERNA ES SOCIAL Y TÉCNICA

- Hay que tomar decisiones a nivel de:
  - **Responsabilidades:** Decidir qué equipo se encarga de qué parte del sistema.
  - **Fiabilidad:** Decidir cómo dividir la arquitectura y los equipos para minimizar errores.



- Enfoque recomendado:
  - **Diseño orientado al dominio (DDD)**: nos ayuda a definir límites claros.
  - Identificar las **transacciones críticas**: determinar qué datos requieren actualización atómica.
  - Distribuir **responsabilidades en función de estos límites.**

LA IMPORTANCIA DEL ASPECTO SOCIAL



- El aspecto social del proyecto puede percibirse de diferente forma dependiendo de la etapa.
  - **Inicio del proyecto.**
    - La modernización comienza como una decisión estratégica.
    - El aspecto social se centra en **lograr aceptación**.
  - **Durante la implementación.**
    - La arquitectura evoluciona, y con ella, la forma en que trabajamos.
    - Aparecen **términos** como microservicios o cloud-native que suenan innovadores, pero pueden generar dudas, confusión o miedo dentro del equipo.
    - El aspecto social se centra en **facilitar adaptación**.

¿CÓMO DECIDIMOS QUÉ MODERNIZAR EN EL SISTEMA?

1° Analizar TODO el sistema.

- **Objetivo:** pensar en cómo **mejorar cada parte** del sistema para hacerlo **más eficiente**, no solo actualizarlo.

2° Identificar qué aspectos revisar.

- UX (*User Experience*)
- Modelo de dominio y datos
- Código mal diseñado

3° Pensar qué cosas mantener y qué cambiar.

- **Entender el valor del cambio**: ¿Mejorará el sistema si modernizamos esta parte?
- Determinar cuál es el **mejor retorno de la inversión (ROI)**: ¿Qué área ofrece el **mayor valor** con el **menor esfuerzo**?

CÓMO MODERNIZAR LOS SISTEMAS

- Existen muchas herramientas, pero no se pueden usar todas.
- La modernización conlleva varios pasos:
  - ✓ Definir unos objetivos de negocio
  - ✓ Comprender el sistema actual a modernizar
  - ✓ Planificar el sistema futuro
  - ✓ Diseñar la estrategia para la transición

- Es crucial **entender** el problema que se está resolviendo **antes** de elegir una técnica, además de tener claro el objetivo de a dónde quieres llegar con el proceso
  - ➢ *Congress law*, *behavioural code analysis,* ***wardley maps****...*

El proceso de modernización generalmente implica:

- **Avanzar desde el 'por qué'…** (estrategia de negocio)
- **… hasta el 'cómo'** (implementación)

Un ejemplo del mundo real ilustra cómo abordar la modernización centrándose en los objetivos de negocio, y después identificando las áreas claves para el cambio.

EJEMPLO UK GOVERNMENT

- Nick habla de primera mano de uno de los mayores problemas que tienen los proyectos grandes, y que indican una alta necesidad de modernización.
- Este era el caso del Gobierno de Inglaterra, en el que los sistemas informáticos, pese a llevar relativamente poco tiempo en funcionamiento, se regían por las normas de negocio (*business rules*).
- Estas eran muy antiguas, y siguiéndolas, el trabajo se hacía imposible.
- **Todo indicaba que era necesaria una modernización a gran escala.**

La importancia de los datos en la arquitectura

- Cualquier sistema es inútil sin datos.
- Todas las aplicaciones modernas se mueven alrededor de estos.
- **Problema:** la estructura de los datos ha cambiado mucho → la práctica de la ingeniería de datos no.
  - Esto crea equipos muy centralizados ☹
  - Puede haber problemas de organización y de arquitectura de empresa

LOS SISTEMAS DE DATOS SE HAN MODERNIZADO

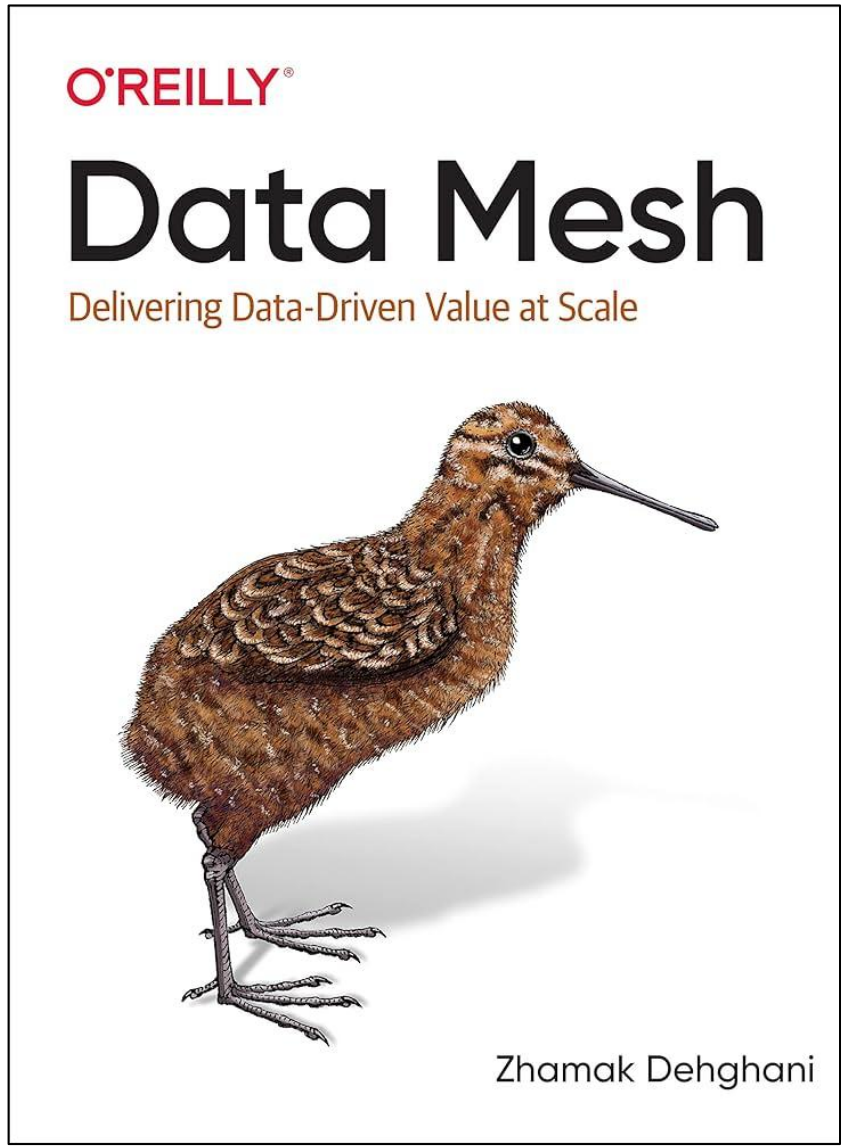
- Antiguamente, el único manejo necesario consistía en hacer *queries* simples.
- Sin embargo, en la actualidad el panorama ha cambiado:
  - Análisis de comportamiento de usuarios
  - Extraer conclusiones de los datos
  - Y mucho más que antes no se contemplaba
- **Las prácticas de ingeniería de datos no han sabido actualizarse y adaptarse** a estos nuevos requerimientos.
  - No han sabido modernizarse

MODERNIZACIÓN DE LA ESTRUCTURA ORGANIZATIVA DE LOS DATOS

- En organizaciones grandes, es normal tener un "**data team**", centrado en la infraestructura de datos, y un "**architecture or software team**", centrado en lo que consisten los sistemas y aplicaciones.
- Estos equipos trabajan de manera **separada**, y esto puede crear cuellos de botella.
- Por tanto, obstaculizan la modernización de la arquitectura general de un proyecto, ya que tienen formas diferentes de trabajar.
  - **Problemas sociales** y **tecnológicos** →

- Por otro lado, **dentro de los propios equipos de datos**, sus componentes se suelen centrar demasiado en <u>mantener sistemas antiguos en vez de modernizarse</u>
  - Acumulan **deuda técnica**

- Suelen seguir unos modelos de cascada llamados "*mini waterfall methods*".
  - Estos, en realidad, **no son escalables**.
  - Ni tampoco **ágiles**
  - **En el mundo de los datos, NO se están usando técnicas actuales de modernización**.



Una posible solución: **Data Mesh**

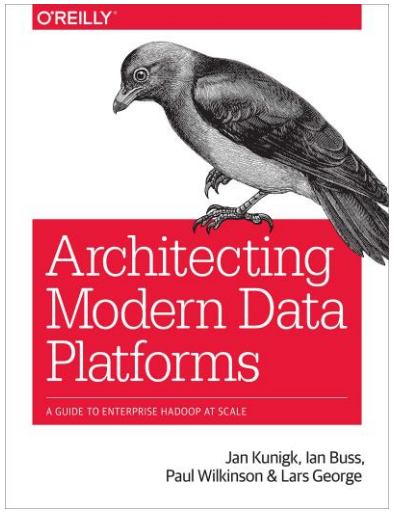
- Se basa en una **gestión descentralizada de los datos** que los organiza en torno a dominios de negocio específicos.
- En vez de tener un equipo centralizado de datos, se **delega la propiedad y responsabilidad de los datos** a los equipos de dominio que los producen y consumen.
- Logramos entonces:
  - ✓ Escalabilidad
  - ✓ Agilidad
  - ✓ Alineación entre equipos

EJEMPLO DE MODERNIZACIÓN DE DATOS: MOVERSE A LA NUBE

- Actualmente, la tendencia de las empresas es mover la gestión de sus bases de datos a la nube → *Outsourcing*
- El problema es que, a menudo, simplemente copian (‘*lift and shift*’) lo que tenían localmente a la nube sin reestructurar la arquitectura.
- Tal cosa trae consigo varios inconvenientes:
  - Si las tecnologías usadas son diferentes, tendremos esquemas no optimizados
  - Los modelos de facturación y de rendimiento cambiarán
  - Aumento de la deuda técnica

ENFOQUE CORRECTO

- En vez de copiar la estructura actual, se debe **invertir tiempo en reestructurar toda la gestión de los datos**, y por tanto, **modernizarla**.
- Esta migración es una muy buena oportunidad para ejecutar esta modernización, pudiendo aplicar conceptos modernos de ingeniería de datos que no estuvieran en la versión *legacy*.



✓ Por ejemplo, si se cambia de tecnología, de *Redshift* a *SQL Server*, se debería aprovechar para optimizar los *data flows* a la nueva.

¿CUÁNDO DEBEMOS MODERNIZAR LA ARQUITECTURA?

Según Jean-Georges…

Hay varios indicadores, pero los **tres motivos más habituales son**:

- **Se quiere ganar más dinero**
- **O bien ahorrar dinero**
- **Cumplir con normativas**: si una aplicación de fuera de la UE se quiere poner a funcionar aquí, se debe reestructurar, por ejemplo, en la gestión de datos para cumplir con el RGPD.

Estos tres motivos, según Jean-Georges, son los más habituales para modernizar arquitecturas de datos.

Sin embargo, coinciden con los motivos para comenzar una modernización a mayor escala de cualquier otra arquitectura de *software*.

¿CUÁNDO DEBEMOS MODERNIZAR LA ARQUITECTURA?

Según Nick Tune…

Costes y soporte	Crecimiento del negocio	Pérdida de oportunidades

Conclusión: para que el negocio sea más **ágil**, **barato** y **competitivo**

¿VALE SIEMPRE LA PENA MODERNIZAR?

No siempre…

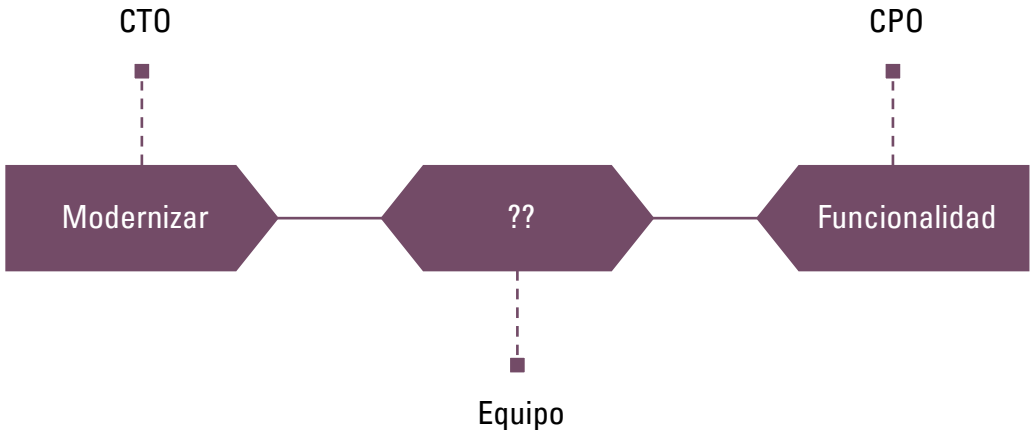
- Cuando el coste y el esfuerzo no se justifican (Nick Tune)
- Comprar en lugar de construir (Jean-George)

SISTEMAS QUE NO NECESITAN MODERNIZACIÓN

Las claves según Nick Tune

< 3% de su experiencia

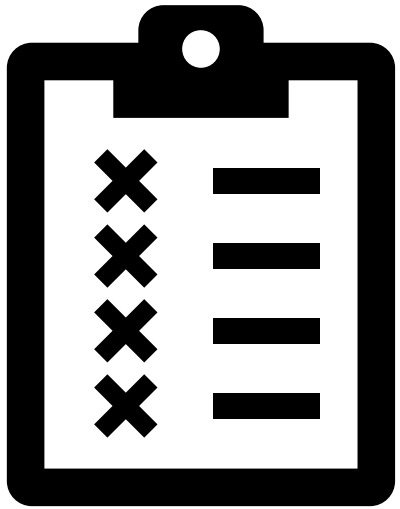
MODERNIZACIÓN VS OTRAS PRIORIDADES



FRACASO DE LOS PLANES DE MODERNIZACIÓN

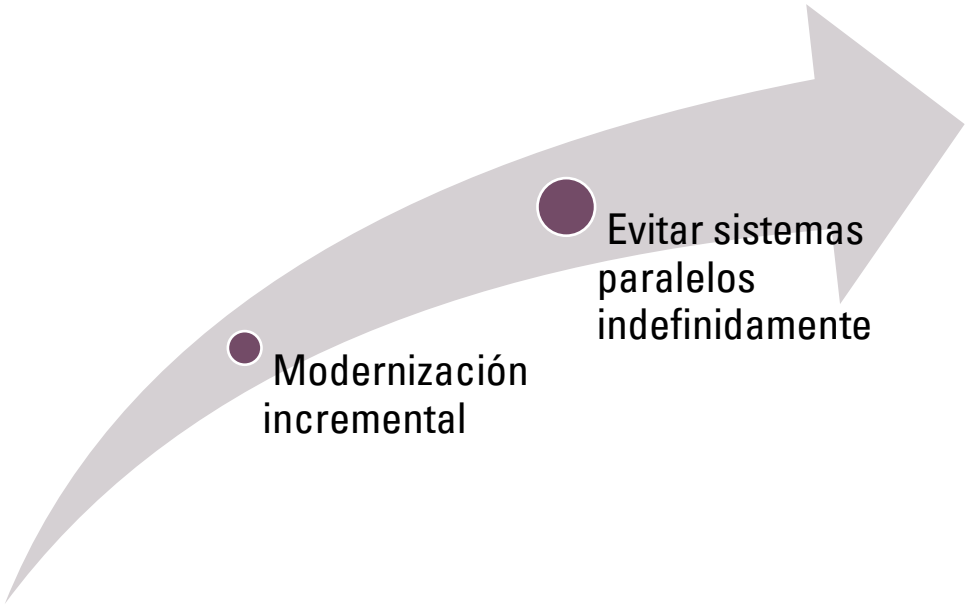
"A veces prometen que te dejarán tiempo para modernizar, pero en cuanto surge una nueva *feature* importante… la modernización se pospone indefinidamente."

- *Nick Tune*



FRACASO DE LOS PLANES DE MODERNIZACIÓN

FRACASO DE LOS PLANES DE MODERNIZACIÓN



"O lo haces, o no lo haces. Pero lo peor es quedarte a medias y crear un sistema *Frankenstein*."

- *Nick Tune*

Aumentar % de éxito según Jean-Georges

GRACIAS POR VUESTRA ATENCIÓN

¿Alguna pregunta?