

# Modernización de la Arquitectura

Software Engineering Radio Podcast #638

Claudia Nistal Martínez  
Marcelo Díez Domínguez  
Marcos Llanos Vega





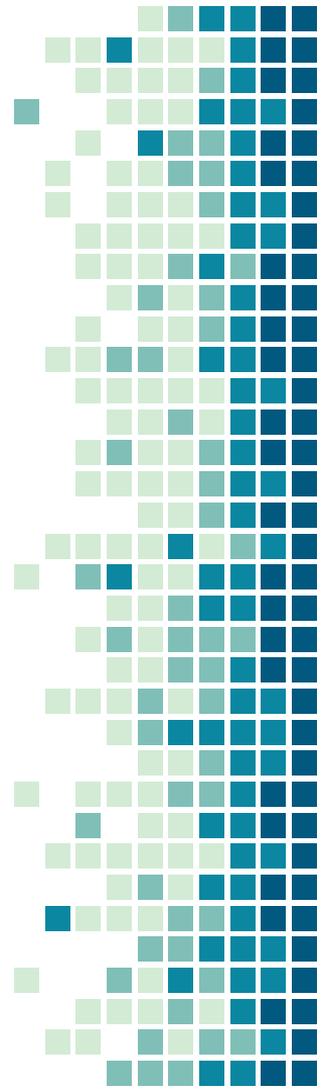
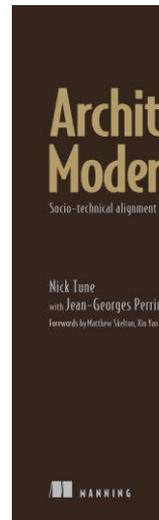
## Nick Tune

Experto en DDD



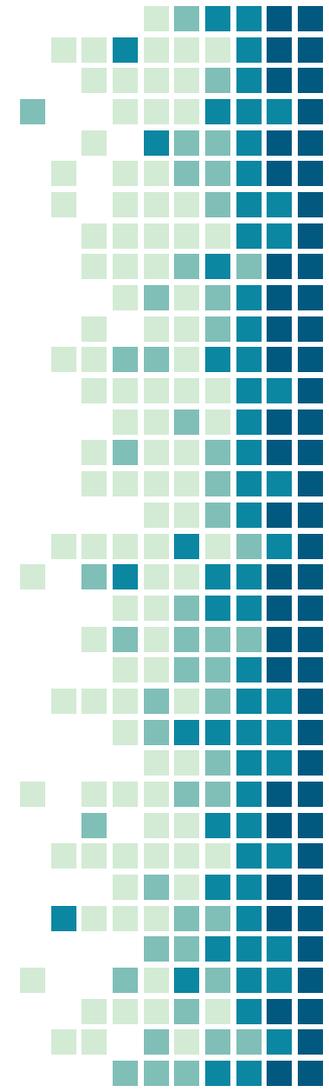
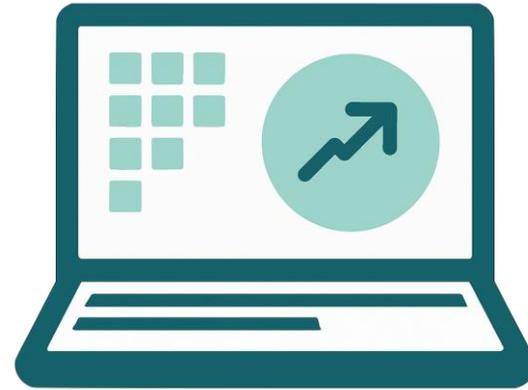
## Jean-Georges Perrin

Director de Innovación en  
ABI Data



# ¿Qué es la Modernización de la Arquitectura?

- Eliminar desventajas de sistemas antiguos
- Uso de prácticas modernas



# Modernización de la Arquitectura en una Empresa

## Gradual

Refactorización constante

Baja deuda técnica

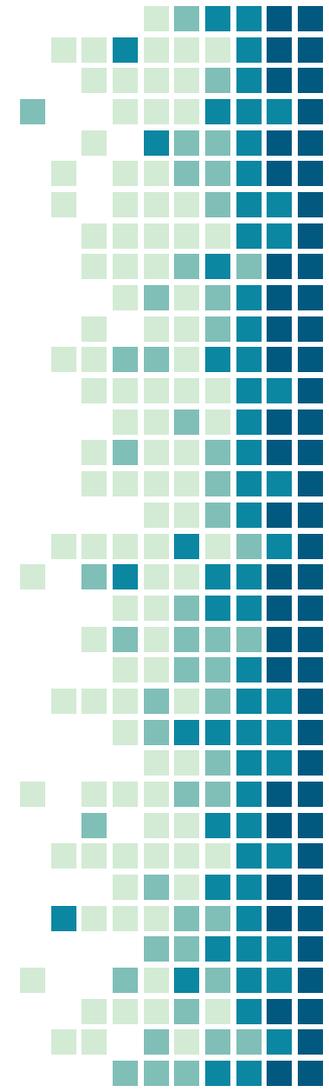
Cambios pequeños y continuos

## Masiva

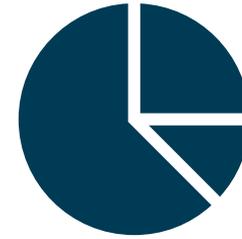
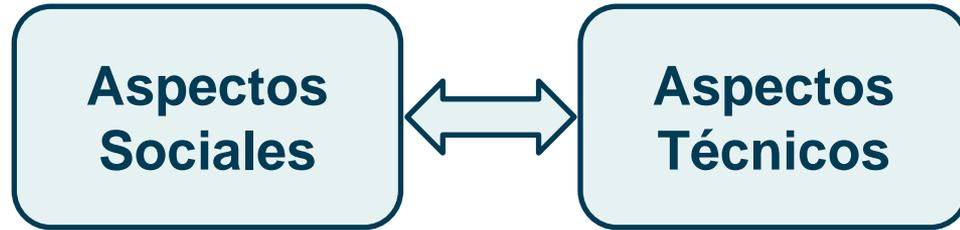
Bloqueos por legado

Parálisis para modernizar

Modernización "forzada" y costosa

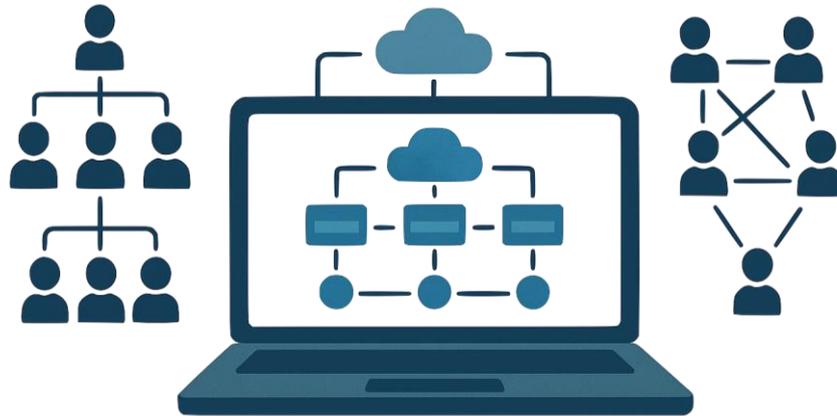


# Arquitectura del Software Sociotécnica



# Ley de Conway

*"La forma de la estructura en un equipo sigue la forma del sistema que se está construyendo"* - Melvin Conway, 1968



# Análisis del Sistema a Modernizar

- Las tecnologías
- El diseño
- Las funciones
- La estructura
- Las habilidades del equipo

Tenemos que revisar todos estos aspectos para ver cómo podemos mejorarlos.



# ¿Qué aspectos revisar?



## Interfaz de Usuario

¿Solo reescribimos el código antiguo?

¿Aprovechamos para mejorar la experiencia del usuario?



## Modelo de Dominio y de Datos

- Reconstruir con nueva tecnología **no basta** si el diseño sigue mal acoplado
- El modelo de dominio refleja los **conceptos del negocio**: si es débil, lo será también la nueva versión.
- **Invertir en un buen modelo de dominio** asegura una base sólida y mantenible.

# ¿Cuál es el potencial de Negocio?

Siempre debemos de hacernos esta pregunta:

"Si lo modernizáramos todo, ¿cuánto valor generaría y cuánto costaría?"

- Evaluar el mejor Retorno de Inversión (ROI)
  - ROI evalúa la rentabilidad de una inversión
  - Analizar cada parte del sistema por separado

$$\text{ROI} = \frac{(\text{Ganancias} - \text{Inversiones})}{\text{Inversiones}}$$


**ROI > 0**  
Se recuperó el total de la inversión y se obtuvieron plusvalías.

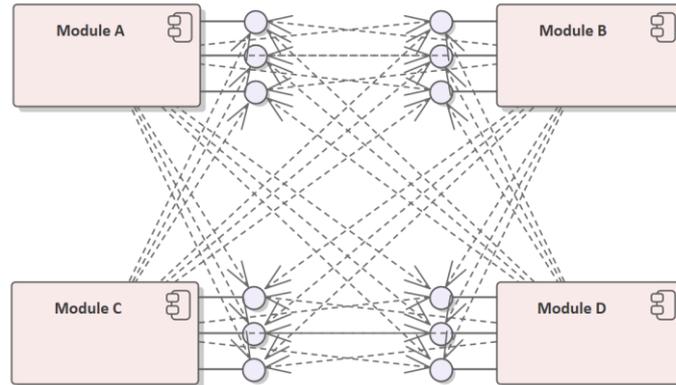
**ROI = 0**  
Se recuperó la totalidad de la inversión. No hubo ganancias ni pérdidas.

**ROI < 0**  
No se recuperó la inversión total y hubo pérdidas.

# ¿Cuál es el potencial de Negocio?

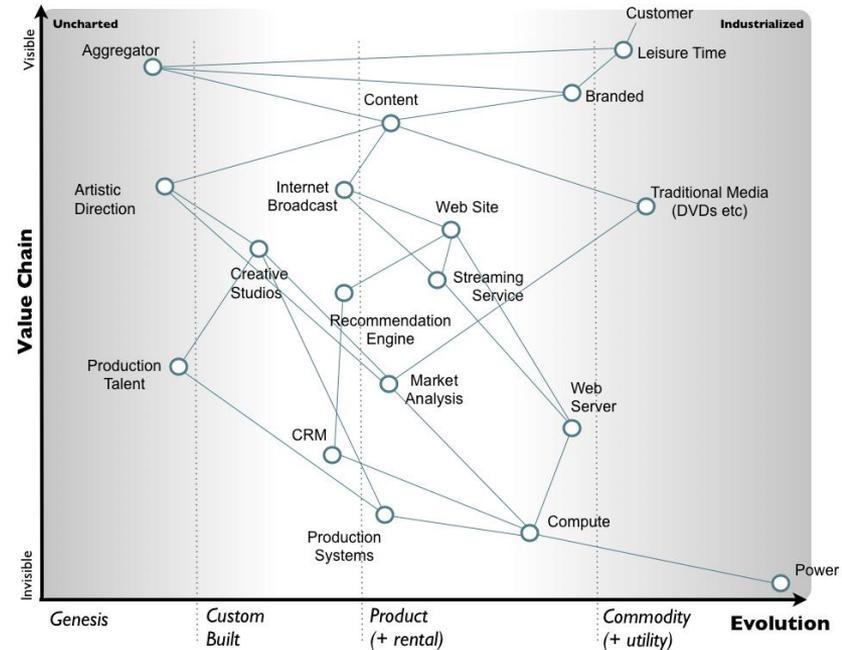
## No hay reglas fijas:

- Se puede cambiar tecnología, tecnología y diseño...
- El grado de acoplamiento del sistema determina el cambio
- Algunas veces conviene replantear el modelo completo.



# Puntos a tener en cuenta en la Modernización

- Estrategia
  - Mapeo de Wardley
- Estado actual
- Estado futuro
- Desacoplar la arquitectura
- Migrar del estado actual al futuro



# Ingeniería de Datos y Arquitectura

## Nada se puede hacer sin datos

En todos los sistemas empresariales modernos siempre hay un equipo de Ingeniería de Datos.



## Equipos de Ingeniería de Datos

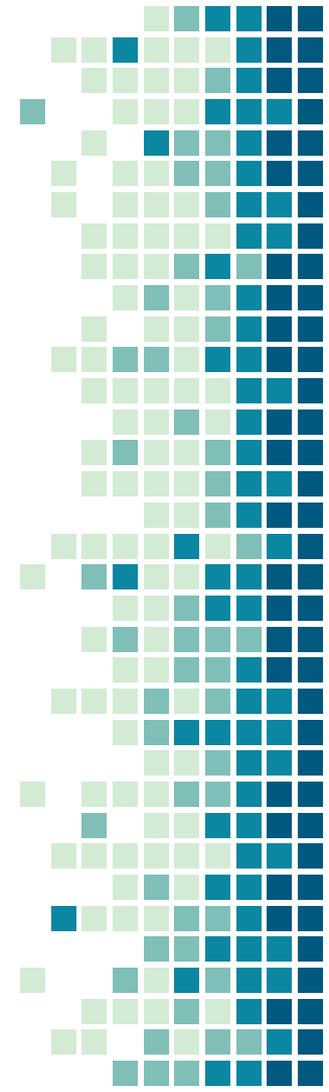
- Suelen ser equipos muy centralizados.
- No se llevan bien con la organización del software.
- No son escalables.
- No se puede modernizar una arquitectura sin modernizar también los datos.

# Los datos como motivación de Modernización

**Una empresa quiere migrar sus datos a la nube**

- Las tecnologías son diferentes
- Las expectativas, la facturación y el rendimiento son diferentes

Cuando se modernizan partes que involucran datos hay que modernizar también su estructura y configuración



# ¿Cuándo Sí debemos modernizar?

Ahorrar dinero

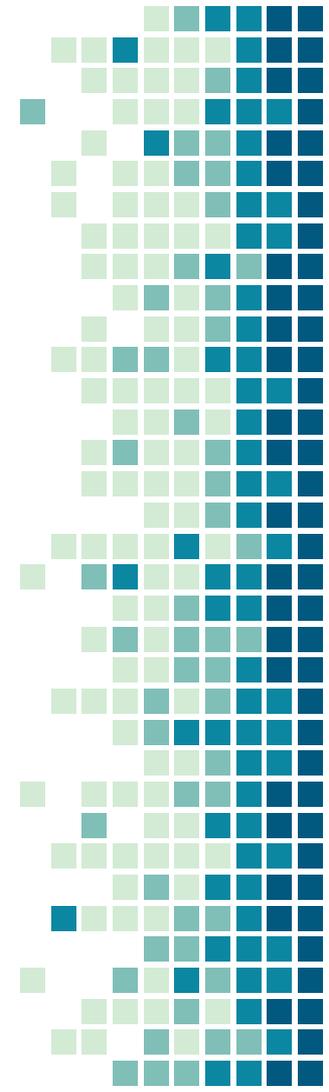
Altos costes  
diarios de soporte  
y mantenimiento

Ganar más dinero

Añadir nuevas  
funcionalidades  
requiere mucho  
tiempo y dinero

Obligación

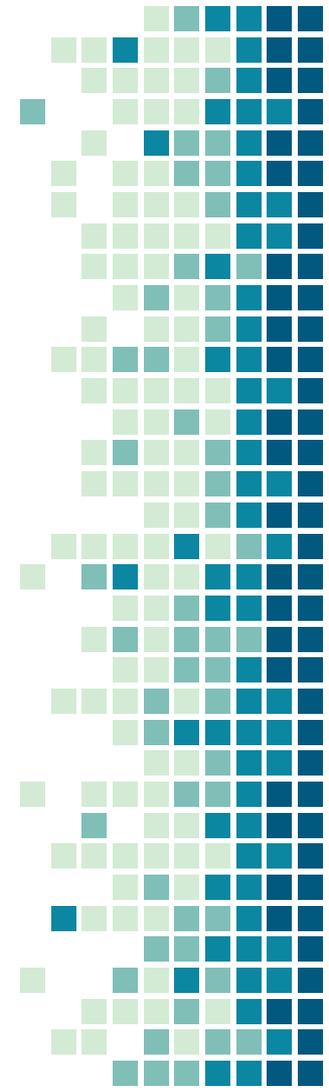
Organismos  
reguladores  
(RGPD, datos  
financieros,  
sanitarios, etc)



# ¿Cuándo NO debemos modernizar?

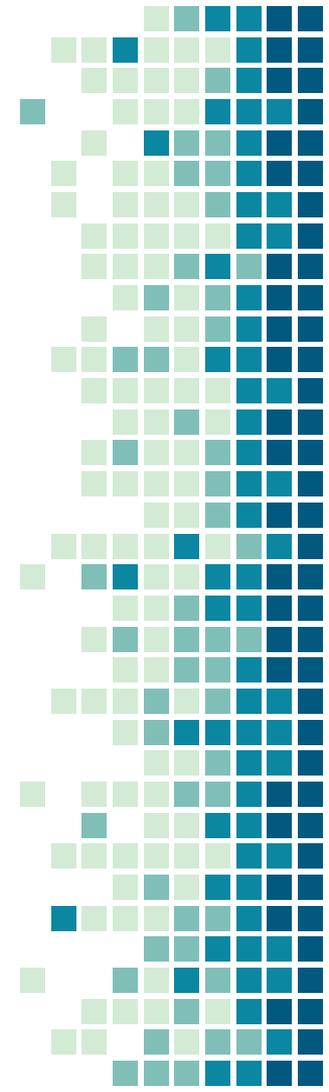
No se quiere perder tiempo ni dinero en algo que no añada nuevas funcionalidades al producto final

Es más fácil, rápido e incluso barato deshacerte de tu viejo sistema y adquirir uno nuevo



# Arquitectura evolutiva

- Idealmente permite crear sistemas que se ajustan para adaptarse a las necesidades cambiantes del negocio
- Sus sistemas se caracterizan por el mínimo acoplamiento.
- Muy pocas empresas lo aplican.
  - Y de éstas, muchas sólo lo hacen parcialmente.



# Características de estas empresas

Equipos de desarrollo  
autónomos

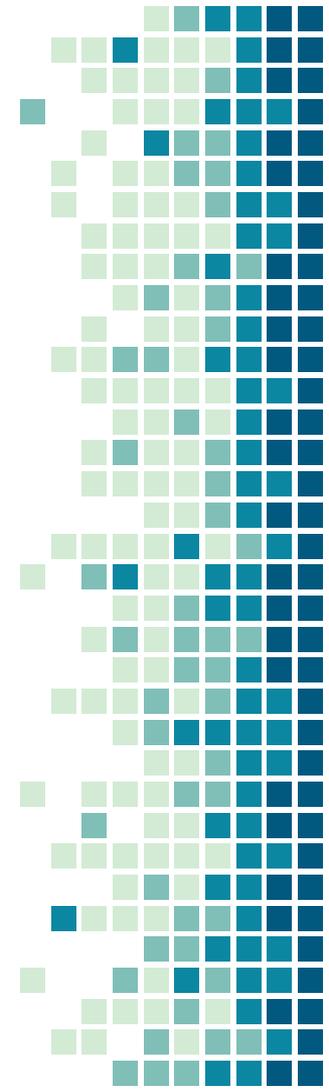
Prácticas de  
programación extrema  
con TDD y  
programación en  
pareja

Centrados en la mejora  
continua de su trabajo

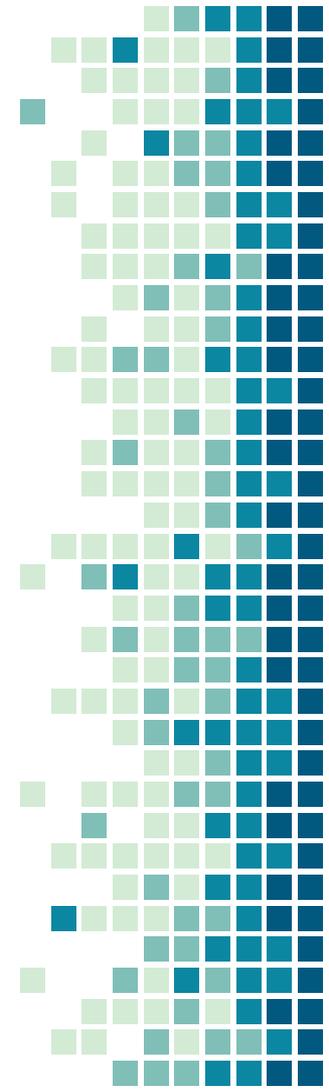
Tratando de  
refactorizar el código  
continuamente

Siempre intentan  
aprender cosas nuevas

La empresa da  
formaciones a sus  
trabajadores



- Son equipos que aplican la máxima calidad a lo que hacen. Luego no necesitan grandes trabajos de modernización.
- No se limita al equipo de desarrollo.
- El CTO tiene que entender y aplicar esa calidad.
- El CEO debe confiar en ellos.
- Los resultados acaban avalando este modelo.



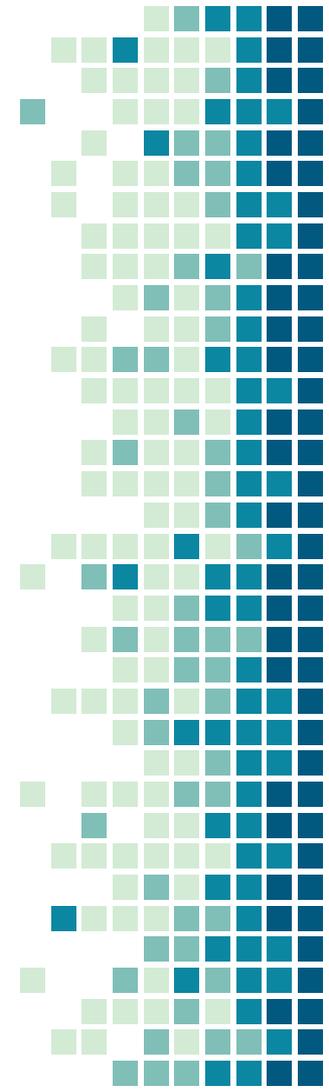
# ¿Cómo equilibrar la modernización con el resto del negocio?

*El tiempo que se dedica a la modernización no se dedica a otras funciones*



Hay que buscar el equilibrio entre modernizar y añadir nuevas funcionalidades:

- Se trata de pensar en el futuro a largo plazo y no tanto en el presente
- Toda la empresa debe ir en la misma dirección



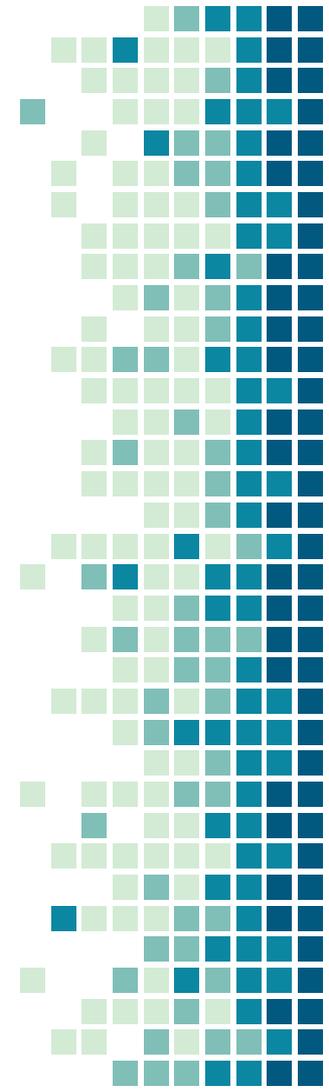
# Éxito vs fracaso de la modernización

Normalmente fracasan porque no se toman en serio:

- Se va aplazando cada vez que surge una nueva funcionalidad
- Al final nunca se termina

Sólo se terminan cuando hay una necesidad clara e inevitable:

- Obligación por ley



# Causas de éxito de la modernización

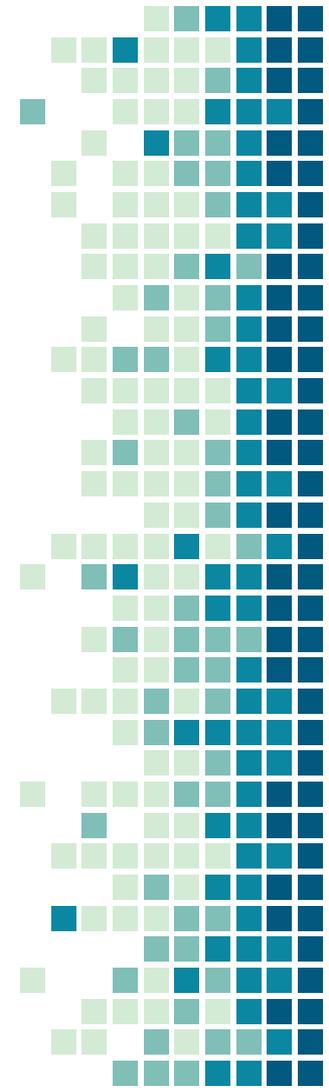
Tener una  
necesidad clara  
e inevitable  
para llevarla a  
cabo

Todas las  
partes  
involucradas la  
apoyan

Analizar con  
detenimiento  
el sistema  
original

Planificar con  
detalle todo el  
proceso de  
migración

UNA VEZ COMENZADO LO HAY QUE TERMINAR, NO SE PUEDE DEJAR A MEDIAS



# Causas de fracaso de la modernización

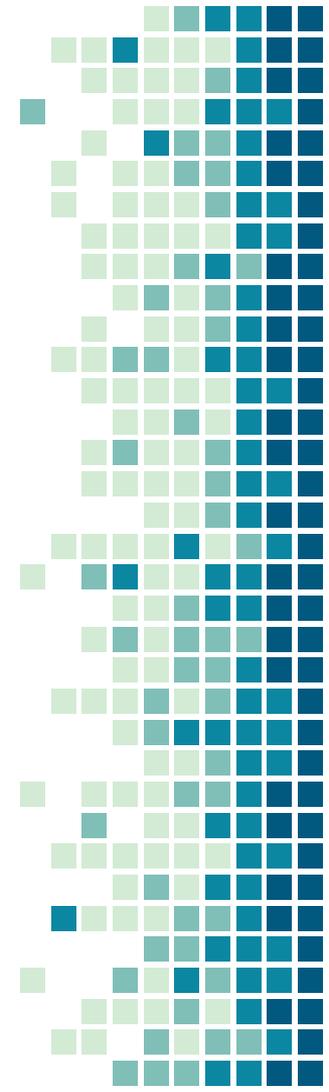
No hay una necesidad clara e inevitable, por lo que se va posponiendo

Las partes no lo apoyan o no tienen habilidades para llevarlo a cabo

No se analiza el sistema de partida, por lo que luego se encuentran problemas

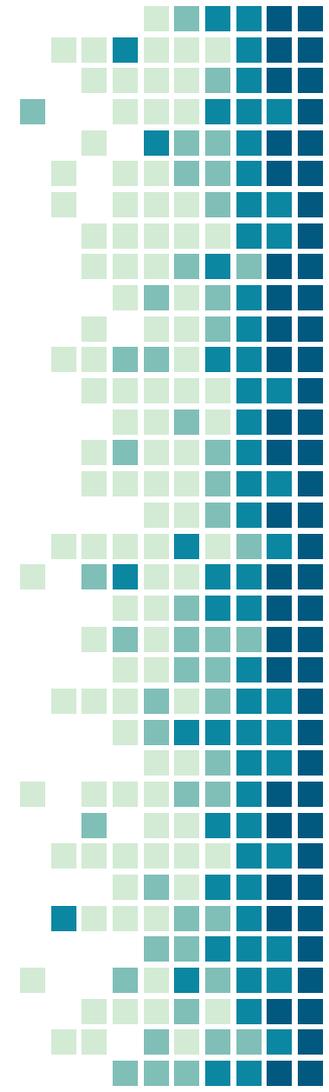
Se deja de lado para añadir nuevas funciones al sistema original

SE VA POSPONIENDO Y FINALMENTE SE DEJA A MEDIAS



# Si no lo hacemos bien:

- El proceso de modernización hay que llevarlo a cabo completo para que sea útil
- Si se deja a medias será peor incluso que no hacer nada
- Tendríamos un sistema más complejo que el inicial, al contar con una parte del viejo y otro del nuevo





# Preguntas

