

ARQUITECTURA DE SOFTWARE

¿Quién necesita de la AS?

Ing. Fernando Soriano e Ing. Zacarías Monzón



NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Un cliente le dice a su proveedor de SW:
“El sistema hace lo que esperaba para resolver mis problemas y tal como acordamos, pero desde hace un mes está lento y cada tanto debo reiniciarlo”

NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Un PM le pregunta a sus developers:

“Este proyecto es muy similar al que hicimos el año pasado para otro cliente. Recuerdan cómo resolvimos el problema de sincronización de datos?”

NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Un cliente le dice a su proveedor de SW:
“El sistema que me instalaste me permitió crecer tanto que ahora abriré sucursales en el exterior. Me sirve para usarlo allí también no?”

NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Un PO reclama al equipo:

“Tan difícil es agregar la funcionalidad de envío de mails?”

NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Un comercial dijo una vez:

“El cliente nos pidió que debemos adaptar el sistema a la nueva reglamentación fiscal. Le dije que en dos semanas lo tenía.

NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Un cliente puso como condición:

“Optamos por su propuesta, pero sepan que la prueba de integración con nuestros servicios no puede hacerse contra el productivo.”

NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Uno de QA hablando con otro:

“Che, ya concluye el proyecto y nuestro cliente nos exige testear una cobertura del 85% como mínimo. Estamos en el 50%, y cada vez tardamos más en hacerlo. Avisamos al PM?”

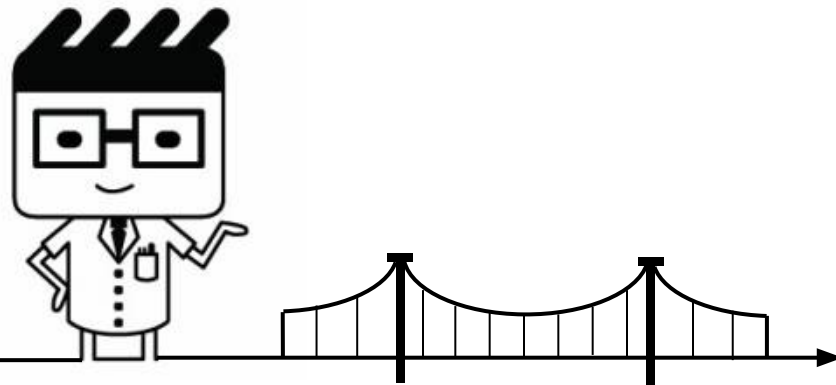
NECESITAMOS ARQUITECTURA?

Algunas situaciones...

Un developer hablando en un after:

“La verdad no sé qué pasó, cuando fuimos a integrar lo que hicieron los de backend, los servicios no eran ni parecidos a lo que nos habían dicho.”

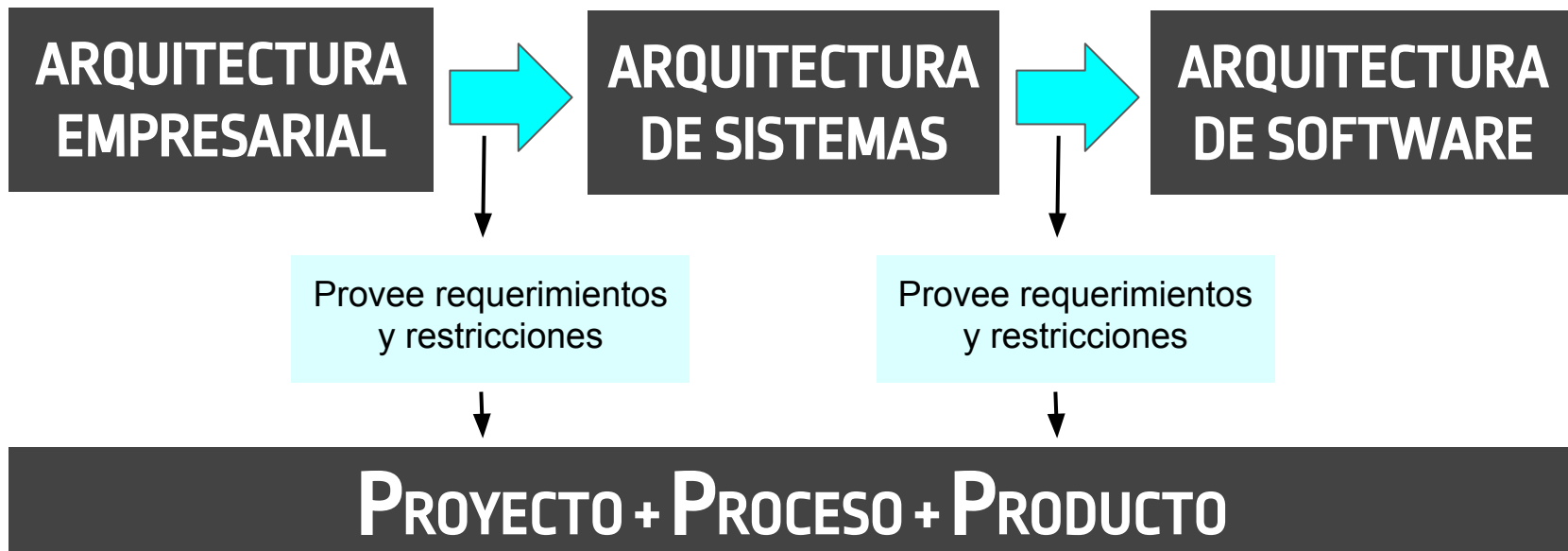
¿QUÉ ES LA ARQUITECTURA?



“Los sistemas son construidos para satisfacer las necesidades de negocio. La arquitectura es un puente entre esos objetivos de negocio y el sistema resultante.”¹

1 Len Bass, Paul Clements, Rick Kazman (2012) Traducido de “Software architecture in practice”. 2nd Edition. ISBN:0321815734 9780321815736

¿QUÉ ES LA ARQUITECTURA?



PROPUESTA: Diseñemos

APP ciudadana para transporte público



CASO: APP ciudadana para transporte público

- Es solo una app móvil para celulares?
- Quiénes la usarían?
- Cómo se sustenta todo el ciclo de vida?
- Debe soportar características de accesibilidad?
- Quién es el verdadero sponsor y que espera de esto?
- El equipo tiene experiencia en Apps móviles?
- Sólo manejará información de colectivos?
- Va a monetizar de alguna manera?
- Será multilingüe?...

CASO: APP ciudadana para transporte público

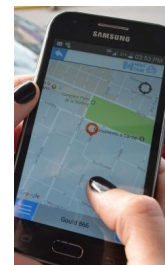
Listo! Esta es la arquitectura...



UBICACIÓN

UBICACIONES

CUÁNDO LLEGA?



Proveedor

DATOS

Servicio de
cuándo llega

CONSULTA

Usuario

Una forma de hacer arquitectura

Proponemos un poco más de trabajo

Requerimientos

Restricciones
de arquitectura

1. Identificar
atributos

Quality
Attributes

2. Diseño de la
arquitectura

SAD w/
Views

1. Identificar atributos

Brainstorming

- Junto al cliente, se generan escenarios a partir de historias cortas.
 - *“Cuando el usuario indica la línea en la que desea viajar y donde desea tomarlo el sistema debe responder en 5 segundos.”*
- Preguntamos:
 - Qué sucede si no se dispone de información del medio de transporte?
 - Qué cantidad de usuarios simultáneos imaginamos?
 - La infraestructura de comunicación nos permite cumplirlo?
 - Existen horarios para bajar el sistema y hacer mantenimiento?
 - Qué ocurre si el servicio no responde, habrá alternativo?



1. Identificar atributos

Priorización - Escenarios

Escenario	Descripción	Votos

1. Identificar atributos

Priorización - Escenarios

Escenario	Descripción	Votos
1	Cuando el usuario indica línea y parada solicitada el sistema responderá como máximo en 5 segundos.	
2	El sistema debe permitir incorporar otros medios de transporte.	
3	El sistema debe soportar desconexión de la unidad de transporte.	

1. Identificar atributos

Refinamiento de los escenarios

Escenario: Nombre del escenario, podría implementarse alguna regla de cod.	
Objetivo de negocio	Para qué, por qué , cómo justificamos este escenario.
Atributo de calidad impactado	
Estimulo	Falla, caída, respuesta incorrecta, acción del usuario.
Respuesta	Que se espera como respuesta ante este escenario.
Preguntas	Preguntas necesarias para refinar el escenario.
Inconvenientes	Riesgos o dificultad

Template basado en "Evaluating Software Architectures: Methods and Case Studies", Paul Clements

1. Identificar atributos

Ya tenemos un atributo identificado

Escenario: El sistema debe soportar desconexión de la unidad de transporte	
Objetivo de negocio	Dar respuesta independientemente de dificultades técnicas en la unidad de transporte.
Atributo de calidad impactado	Disponibilidad
Estimulo	Fallo de conexión con la unidad de transporte.
Respuesta	Horario estimado de arribo
Preguntas	Podemos utilizar tablas horarias? Existen algoritmos de predicción? Tenemos los datos para hacerlo?
Inconvenientes	Existen riesgos de error por cortes o demoras en ruta. Se debe informar al usuario de esta situación.

▶ ATRIBUTOS DE CALIDAD

→ **ADECUACIÓN
FUNCIONAL**

→ **MANTENIBILIDAD**

→ **FLEXIBILIDAD**

→ **COMPATIBILIDAD**

→ **EFICIENCIA DE
DESEMPEÑO**

▶ Comportamiento temporal

▶ Utilización de recursos

▶ Capacidad

→ **SEGURIDAD**

→ **PORTABILIDAD**

▶ Adaptabilidad

▶ Facilidad de Instalación

▶ Capacidad de ser reemplazado

→ **USABILIDAD**

1. Identificar atributos

Disponibilidad

Disponibilidad	Tiempo fuera de línea			
	Por año	Por mes	Por semana	Por día
90%	36.53 días	73.05 horas	16.80 horas	2.40 horas
99%	3.65 días	7.31 horas	1.68 horas	14.40 minutos
99.99% (4 nueves)	8.77 horas	43.83 minutos	10.08 minutos	1.44 minutos
99.99999% (7 nueves)	3.16 segundos	262.98 ms	60.48 ms	8.64 ms
99.9999999% (9 nueves)	31.56 ms	2.63 ms	604.80 μ s	86.40 μ s

Floyd Piedad, Michael Hawkins (2001). *High Availability: Design, Techniques, and Processes*. Prentice Hall. [ISBN 9780130962881](#).

1. Identificar atributos

Utility tree

Atributo de calidad	Subclasificación	Escenario	Prioridad ¹	Complejidad ²
Disponibilidad	Caídas	El sistema debe soportar desconexión del medio de transporte	A	A
Usabilidad	Contextos	El sistema debe permitir incorporar otros medios de transporte.	A	M
Performance	Tiempo de respuesta	Cuando el usuario indica línea y parada solicitada el sistema responderá como máximo en 10 segundos.	A	A

¹ Prioridad para el éxito del sistema (Alta, Media, Baja).

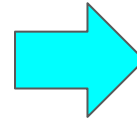
² Dificultad/Riesgo para lograrla (Alta, Media, Baja).



Resumen

1. Identificar atributos

- Brainstorming
 - Escenarios
 - Valoración
- Refinamiento de los escenarios
- Utility tree

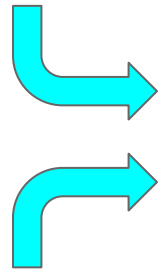


**Quality
Attributes**

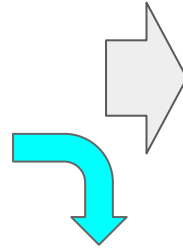
2. Diseño de la arquitectura

Proceso de diseño

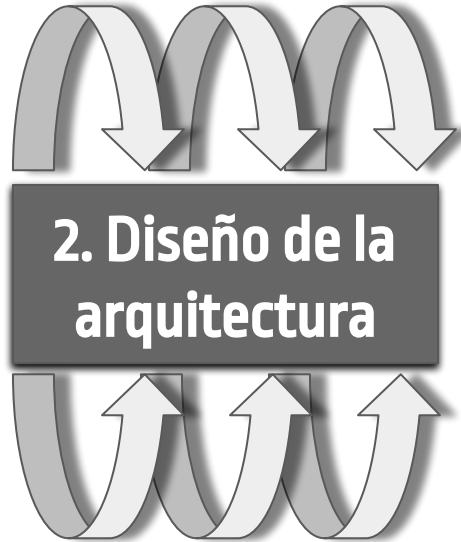
Requerimientos



1. Identificar atributos



2. Diseño de la arquitectura



Restricciones de arquitectura

Quality Attributes

SAD w/
Views

2. Diseño de la arquitectura

Cómo lo documentamos

Documentamos arquitecturas para:

- Educar
- Comunicar
- **Análisis**

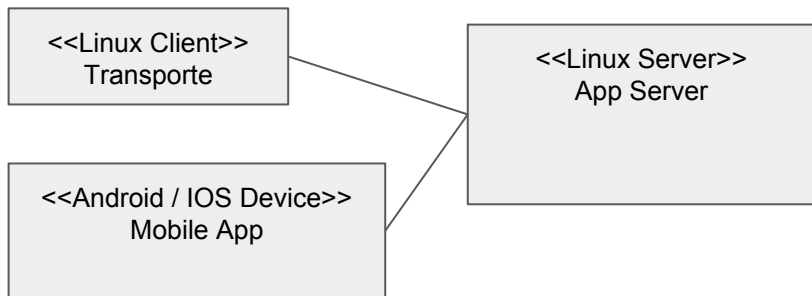
2. Diseño de la arquitectura

Cómo lo documentamos

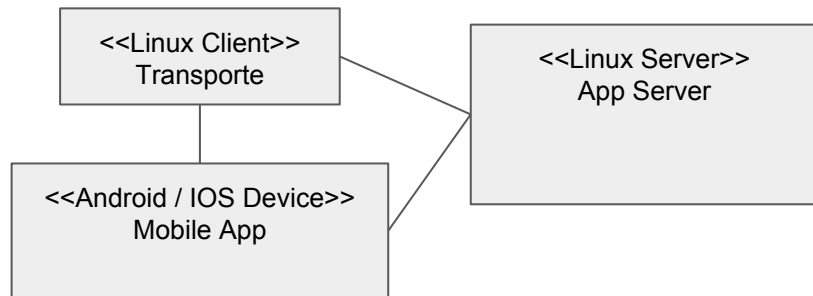
- Una propuesta puede ser utilizando vistas (“Views”).
- Las vistas son representaciones de estructuras que ayudan a entender la complejidad de los sistemas.
- Son diagramas de líneas y cajas que deben tener referencia o utilizar un estándar de notación.
- Pueden ser vistas de módulos, de componentes y conectores o de asignación.

2. Diseño de la arquitectura

Una vista de asignación (draft)



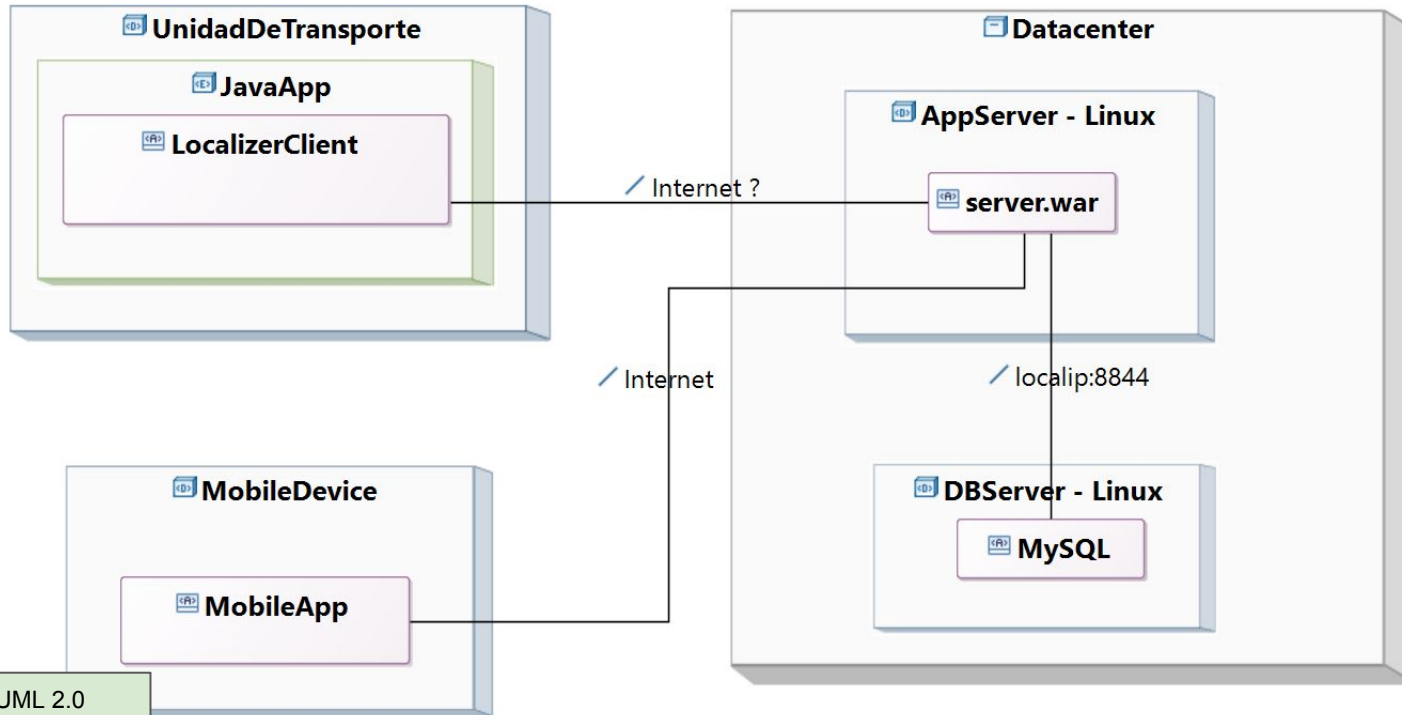
Client server / Event driven



Peer to peer & Client Server / Event Driven

2. Diseño de la arquitectura

Una vista de asignación (uml2)



Notación: UML 2.0

2. Diseño de la arquitectura

Iteramos y evaluamos

	Disponibilidad	Modificabilidad	Performance
Servidor	!	!	!
Localizador	!		
App mobile			!



2. Diseño de la arquitectura

¿Cómo evaluamos?

¿Cumple el componente **servidor** con el atributo de calidad **DISPONIBILIDAD**?.

Para eso podemos hacer un checklist !!!

ANÁLISIS Y DISEÑO DE ARQUITECTURA

Checklist: Disponibilidad

¿Cómo debe comportarse el componente?

- ¿Debe detectar el fallo? ¿Debe evitarlo? ¿Puede estar temporalmente fuera de servicio? ¿Puede operar con condiciones degradadas?

¿Qué responsabilidades tiene?

- Es responsable del reemplazo de artefactos, logue el fallo, determina si opera bajo condiciones degradadas

Modelo de datos

- Determinar qué mensajes/datos/request pueden perderse o debe asegurarse su arribo. Que puede estar temporalmente fuera de servicio.

Tiempo de asociación de componentes

- Los componentes se asocian en el arranque, pueden asociarse desasociarse en tiempo de ejecución.

ANÁLISIS Y DISEÑO DE ARQUITECTURA

Checklist: Disponibilidad del servidor

¿Cómo debe comportarse el componente?

- Si el servidor se cae debe darse igualmente una respuesta. Dentro de los 10 seg.
- La respuesta no puede depender de que el servidor tenga información brindada, ya sea push o pull del medio de transporte, si no se dispone aún así debe dar una respuesta.

¿Qué responsabilidades tiene?

- El servidor debe resolver en caso de caída en forma totalmente transparente a la app cliente.

Modelo de datos

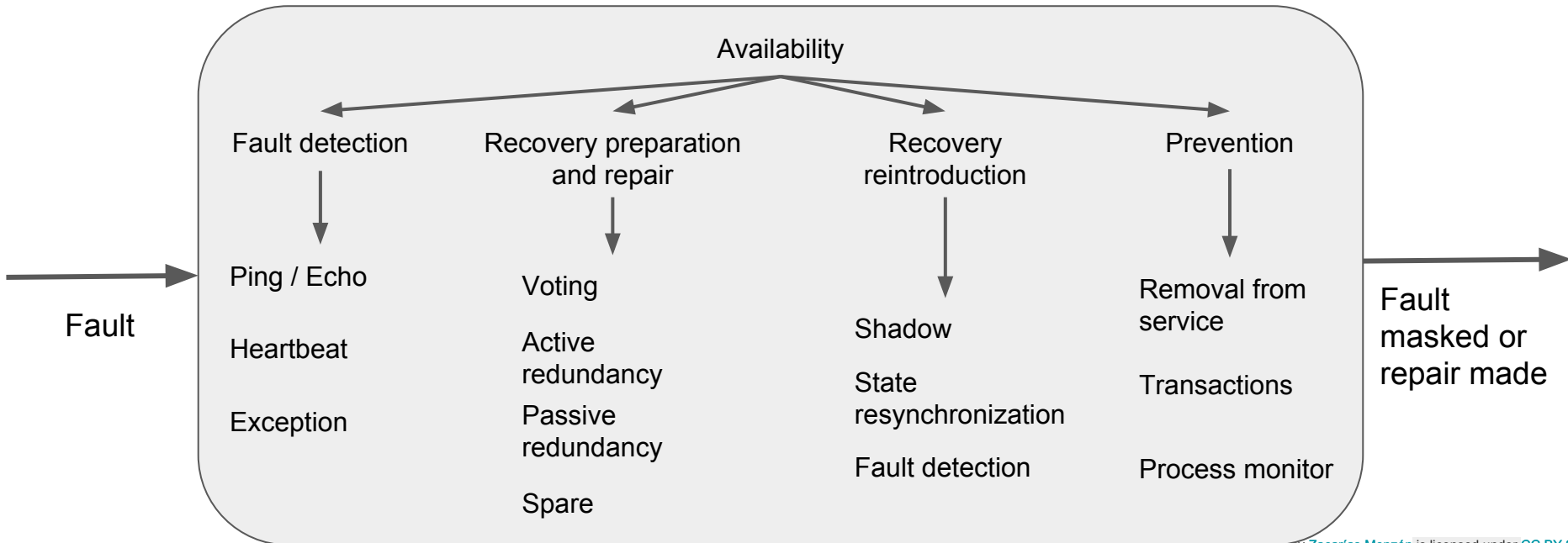
- Todo request de la app cliente debe ser respondido.
- Pueden perderse mensajes de la aplicación localizadora.

Tiempo de asociación de componentes

- Los componentes no deben requerir asociación persistente.

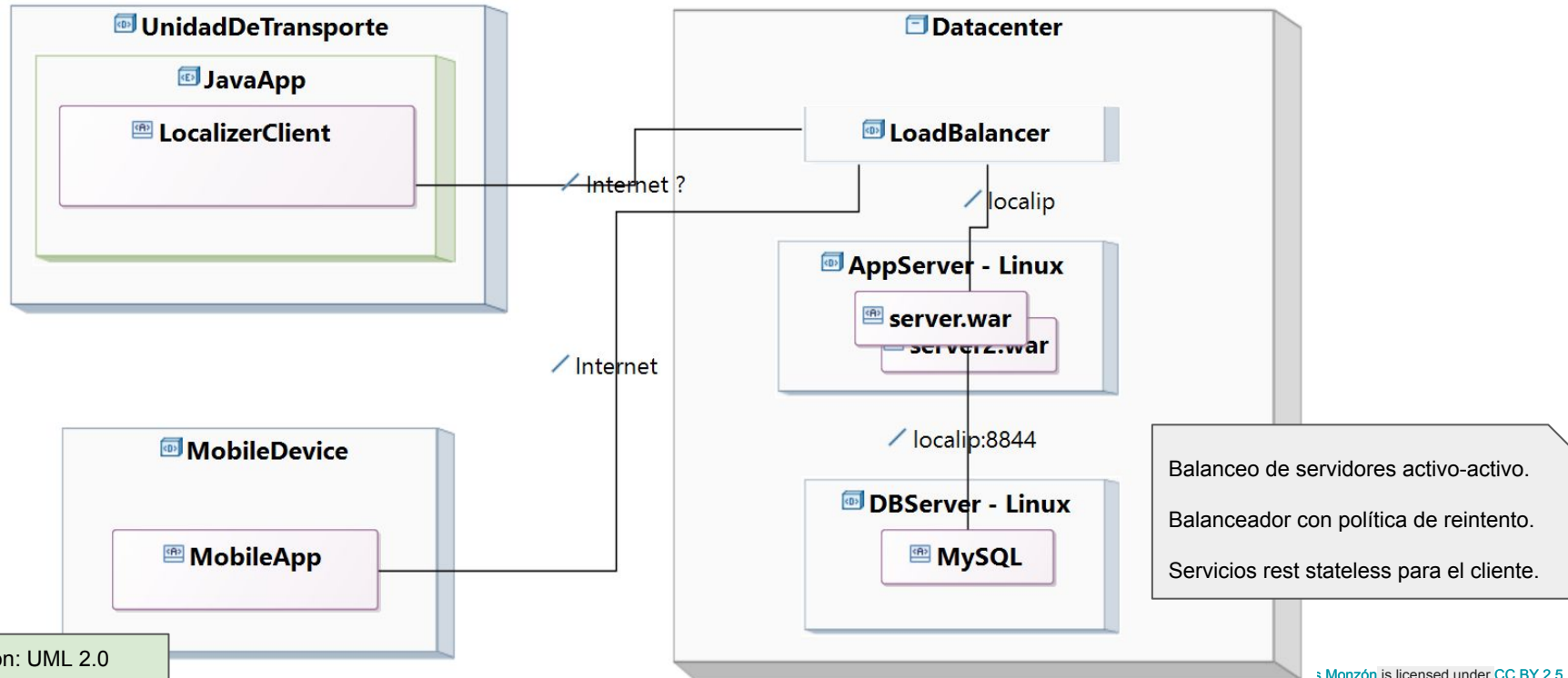
ANÁLISIS Y DISEÑO DE ARQUITECTURA

¿Cómo lo resolvemos?, tácticas.



ANÁLISIS Y DISEÑO DE ARQUITECTURA

¿Qué hacemos con el servidor?



ANÁLISIS Y DISEÑO DE ARQUITECTURA

Checklist: Modificabilidad del servidor

Asignación de responsabilidades

- ¿Qué nuevas responsabilidades incorporará el componente?
- ¿Qué responsabilidades serán impactadas por el cambio?, ¿Qué módulos?

Coordinación

- ¿Qué atributo de calidad o funcionalidad puede cambiar en tiempo de ejecución?
- ¿Qué protocolos, direccionamientos o dispositivos pueden cambiar en ejecución?

Modelo de datos

- ¿Qué entidades pueden cambiar?

Cambios de infraestructura

- ¿Cambia el comportamiento de procesos, threads, procesadores?

Gestión de recursos

- ¿Cómo impacta el agregado, eliminación o modificación de recursos?

Tiempo de instanciación

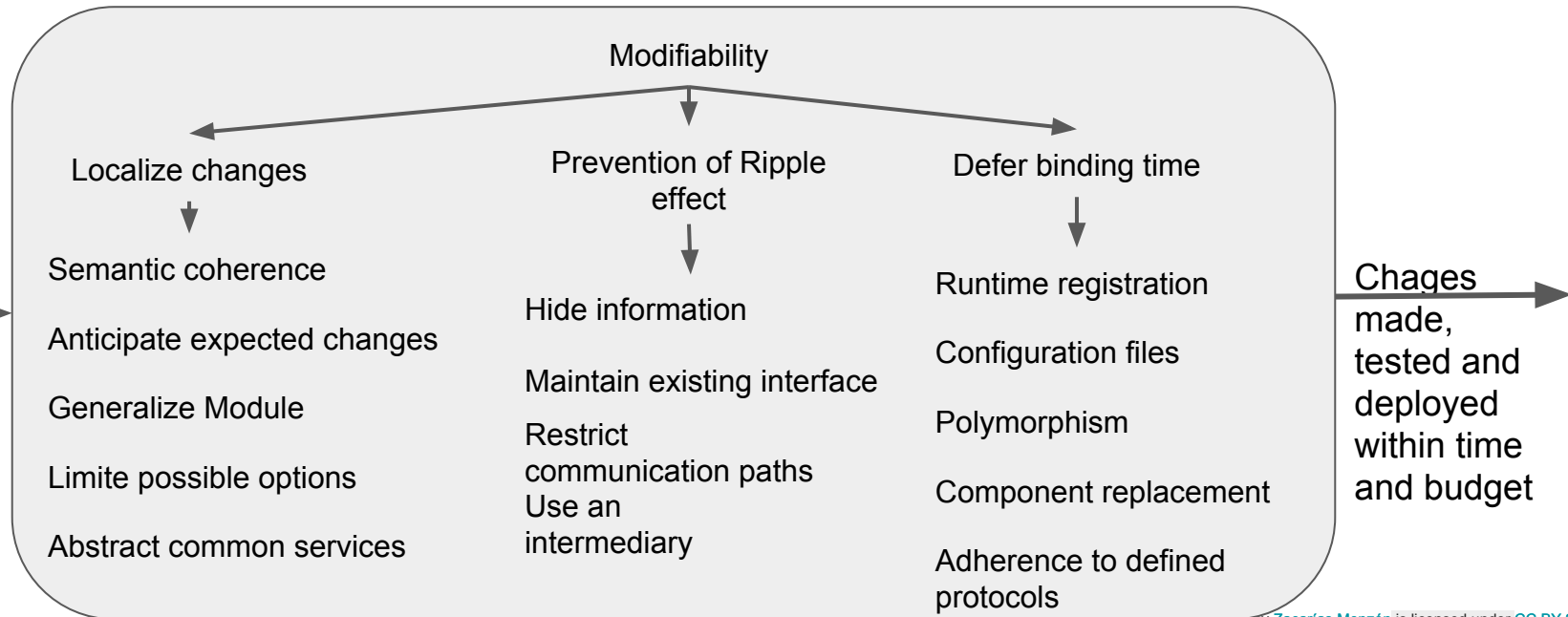
- ¿Cuál es el último momento que en el cambio puede ser hecho, determinar el costo de diferir vs el de otro mecanismo.?

Selección de tecnología

- ¿La tecnología ayuda a realizar, testear y deployar los cambios?

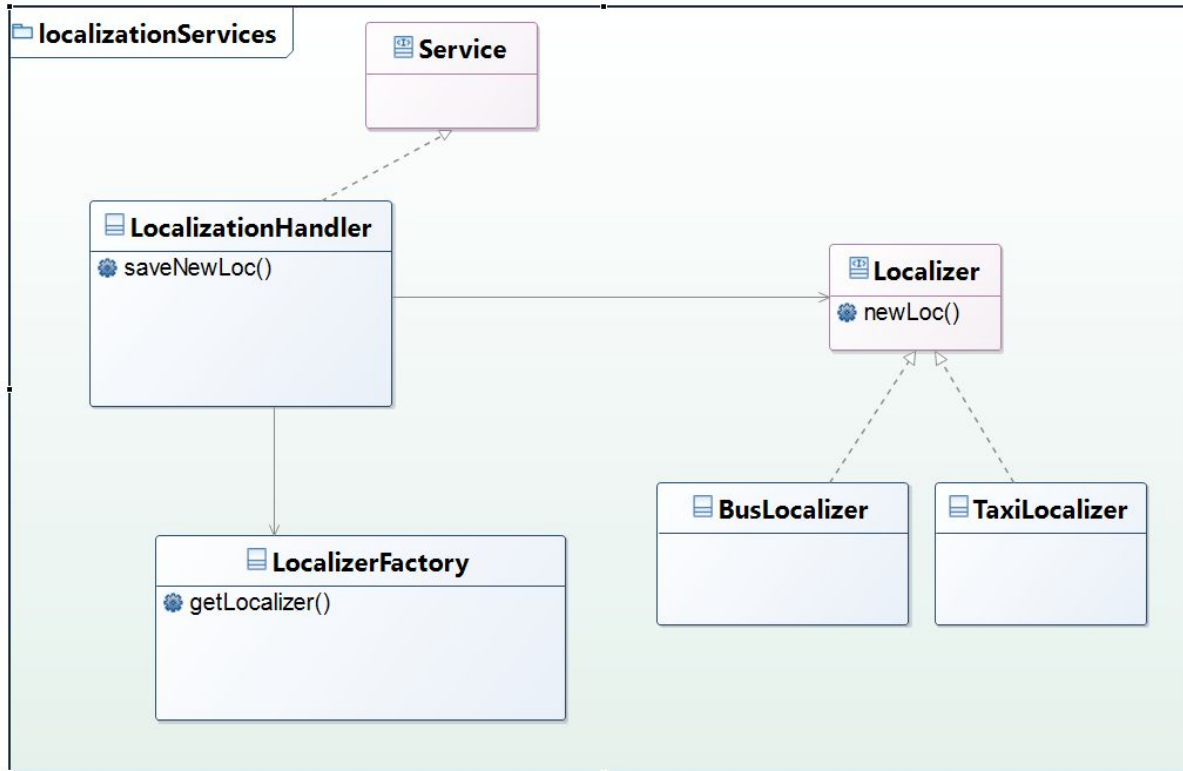
ANÁLISIS Y DISEÑO DE ARQUITECTURA

Tácticas de modificabilidad



ANÁLISIS Y DISEÑO DE ARQUITECTURA

¿Qué hacemos con el servidor?





Conclusiones

La arquitectura de software y el Proyecto

- Insumo para la planificación
 - Alcance
 - Coherencia en cada etapa (mapa de proyecto)
 - Recursos técnicos y equipo
- Costos (ciclo de vida completo)
 - Evaluación de diferentes costos
 - Mejor estimación de tiempos
 - Vida útil del producto
- Comercialización
 - Licenciamiento, Delivery, Mercado



Conclusiones

La arquitectura de software y el Proceso

- Actividades que hacen a la AS
 - Crear el negocio del sistema
 - Comprender requerimientos. En particular los QA
 - Crear o seleccionar la arquitectura
 - Documentar y comunicar la arquitectura
 - Analizar o evaluar la arquitectura
 - Asegurar la construcción del sistema sobre esa arquitectura

Conclusiones

La arquitectura de software y el Producto

- Demanda actual del SW (de nuevo QAs!)
 - Performance y disponibilidad
 - Facilidad de uso (experiencia de usuario)
 - Interoperabilidad
 - Seguridad
 - Flexibilidad y mantenibilidad
 - Escalabilidad
 - Facilidad de entrega y despliegue

“La arquitectura es un factor fundamental para lograr que el producto satisfaga los objetivos de negocio.”

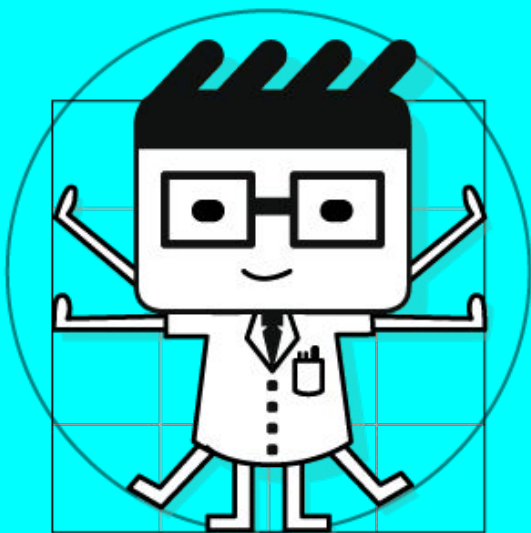


Conclusiones

¿Quién necesita de la AS?

TODOS

los que pretendemos desarrollar software que responda a las necesidades y objetivos de la organización.



ARQUITECTURA DE SOFTWARE

GRACIAS

Ing. Fernando Soriano e Ing. Zacarías Monzón

