

Fundamentos de Definición de Arquitectura de Software

Mauricio Naranjo
Chief Architect
mnaranjo@lucasian.com

Lucasian Labs Ltda.

Septiembre 27 a Octubre 01 de 2005
Bogotá, Colombia



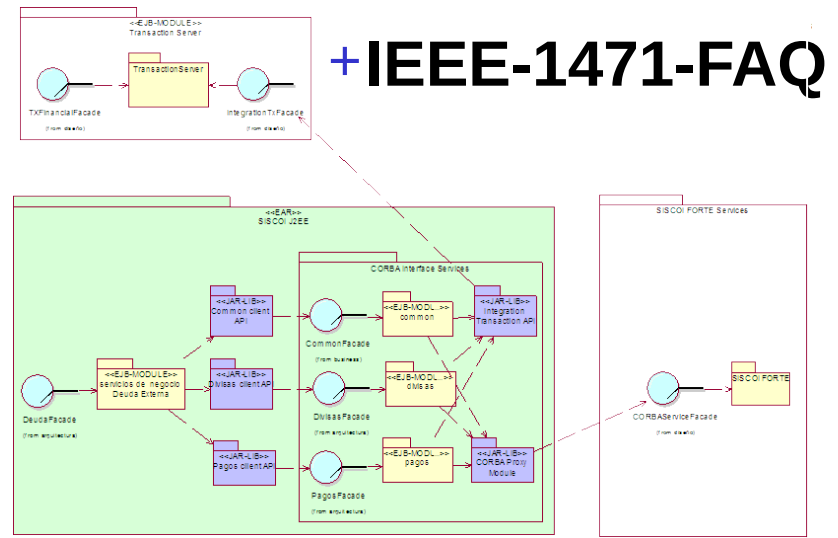
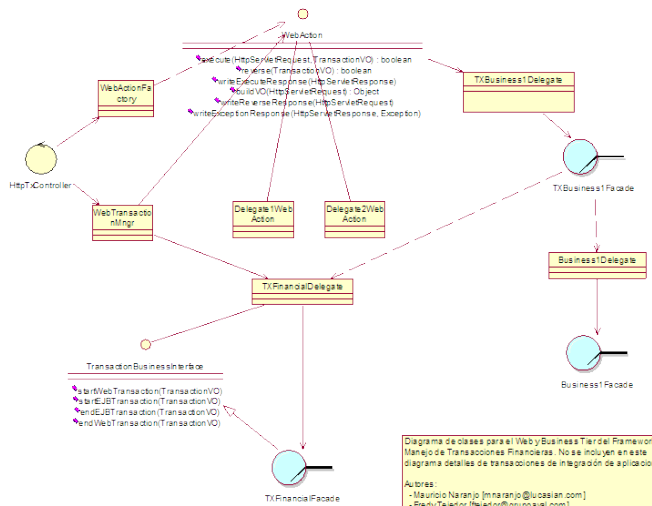
Agenda Conferencia

- + Que es Arquitectura de Software?
- + Rol y Responsabilidades del Arquitecto
- + Arquitectura Vs. Diseño
- + Estrategias de Definición de Arquitectura
- + Ejemplo de Definición de Arquitecturas
- + Calidades Sistémicas y Calidad de Servicio
- + Lecciones Aprendidas en Consultoría

Arquitectura de Software

+Que es una arquitectura?

+“No estamos seguros, pero la reconocemos cuando vemos una”



Arquitectura de Software

+ IEEE 1471

El nivel conceptual más alto de un sistema en su ambiente.

- + Arquitectura es la organización fundamental de un sistema descrita en:
 - Sus componentes.
 - Relación entre ellos y con el ambiente.
 - Principios que guían su diseño y evolución.

+ Software Architecture in Practice - Kazman

“La estructura de estructuras de un sistema, la cual abarca componentes de software, propiedades externas visibles de estos componentes y sus relaciones”.

Discusión

- + Definir la arquitectura en los proyectos actuales es crítico...

por que?



XXV XXXV XXXV XXXV XXXV XXXV XXXV XXXV XXXV XXXV

Evolución de Arquitecturas

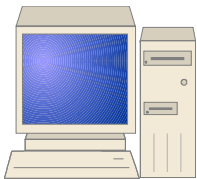
- + Dos factores primarios en la ingeniería de software que han incrementado la importancia de la arquitectura:



Evolución de Arquitecturas

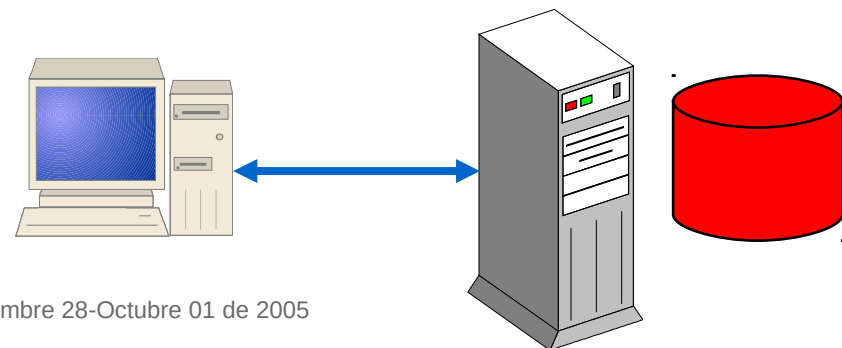
+ Aplicaciones Monolíticas

- + Interfaces gráficas de usuario (GUI).
- + Servicios de presentación, negocios y persistencia en la misma máquina.
- + No hay concurrencia de usuarios.
- + Alto acoplamiento entre tiers.



Arquitectura Cliente-Servidor

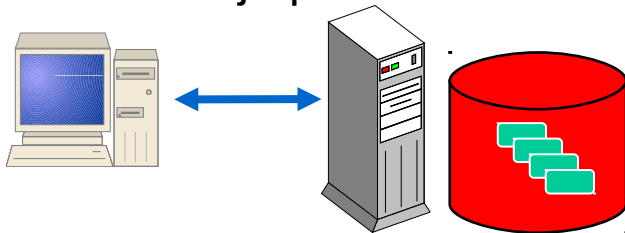
- + Clientes pesados, no estándar
- + Conexiones dedicadas a BD
- + Protocolos pesados
- + Ejecución remota de SQLs
- + Alta administración
- + Bajo rendimiento
- + Alto tráfico de red
- + Baja accesibilidad



Evolución de Arquitecturas

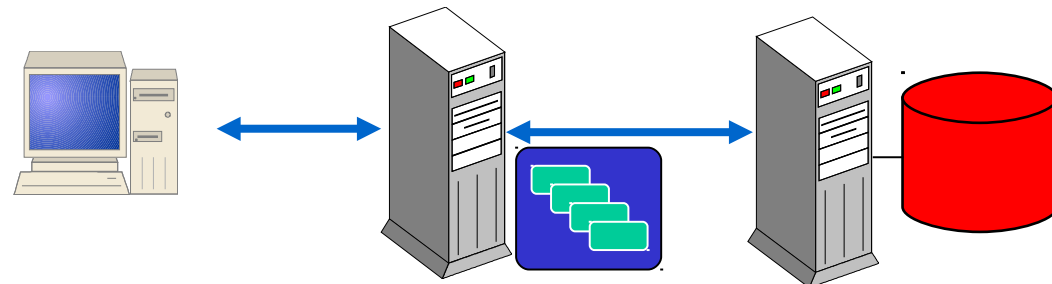
+ Arquitectura Cliente-Servidor Mejorada

- + Lógica de negocios en BD
- + Clientes pesados, no estándar.
- + Conexiones dedicadas a la BD.
- + Mejora en rendimiento
- + Alta administración
- + Baja escalabilidad
- + Baja flexibilidad
- + Baja portabilidad



Arquitectura de 3 niveles

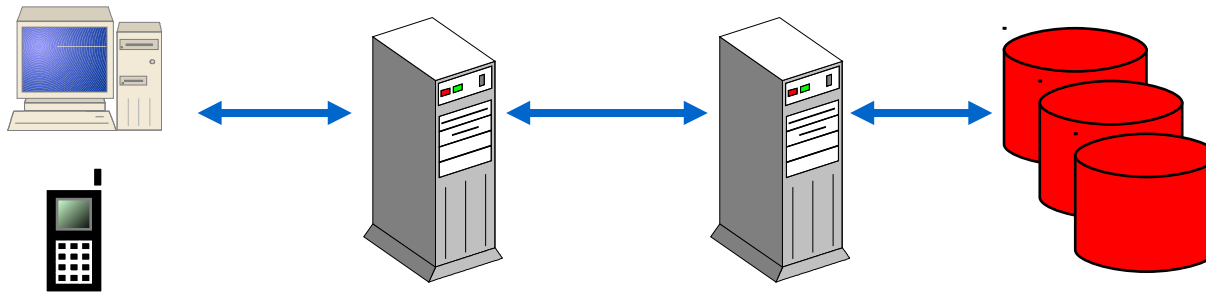
- + Reutilización de lógica de negocio para diferentes clientes o sistemas.
- + Mejora la escalabilidad.
- + Mejora la flexibilidad.
- + Independencia de la base de datos.



XXV XXXV XXXV XXXV XXXV XXXV XXXV XXXV XXXV XXXV

Evolución de Arquitecturas

+ Arquitectura de N-niveles



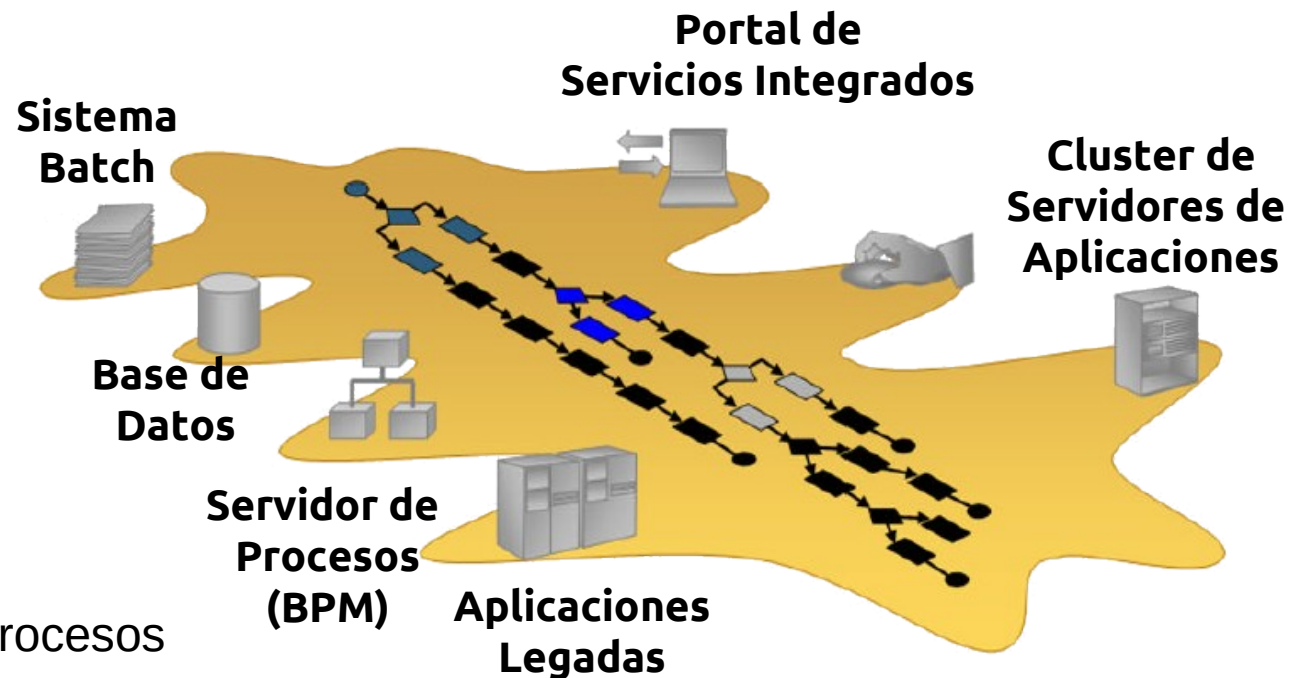
- + Bajo costo de administración de clientes.
- + Alta accesibilidad.
- + Alta flexibilidad.
- + Alta disponibilidad y tolerancia a fallos.
- + Alta escalabilidad.
- + Independencia de DB

Evolución de Arquitecturas

+ Visión de Arquitectura Orientada a Servicios (SOA)

+ Requerimientos Arquitectónicos

- + Heterogeneidad
- + Escalabilidad
- + Disponibilidad
- + Distribución
- + Manejabilidad de Procesos
- + Administración y monitoreo de procesos, servicios e infraestructura



Que es un Arquitecto de Software?

- **Rational Unified Process**

Arquitecto es un rol en un proyecto de desarrollo de software el cual es responsable de:

- Liderar el proceso de arquitectura.
- Producir los artefactos necesarios:
Documento de descripción de arquitectura
- Modelos y prototipos de arquitectura.



- **SUN SL-425:**

El arquitecto:

- Visualiza el comportamiento del sistema.
- Crea los planos del sistema.
- Define la forma en la cual los elementos del sistema trabajan en conjunto.
- Responsable de integrar los requerimientos no-funcionales (NRFs) en el sistema.



Discusión

- + Existe alguna diferencia entre arquitectura y diseño de software?

Votación:
son iguales?



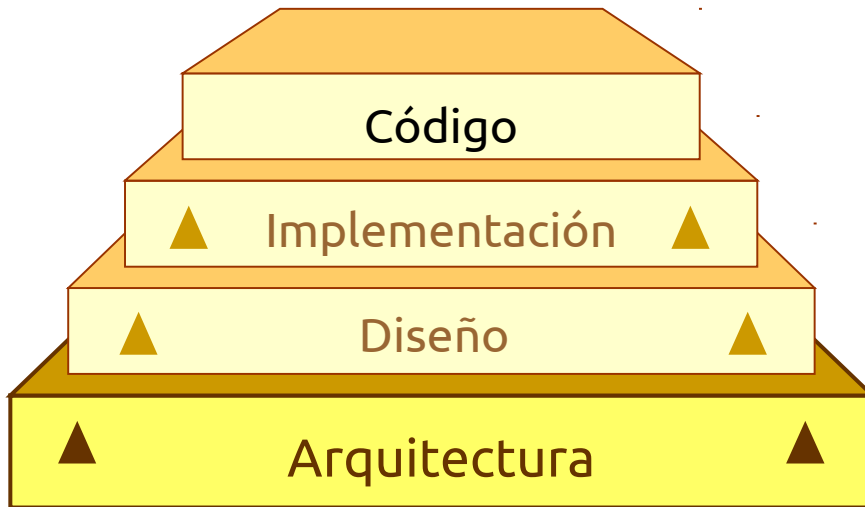
Arquitectura Vs. Diseño

+ La arquitectura y el diseño difieren en tres áreas:

	Arquitectura	Diseño
Nivel de Abstracción	Alto nivel	Bajo nivel. Enfoque específico en detalles
Entregables	Planear subsistemas, interfaces con sistemas externos, servicios horizontales, frameworks, componentes reutilizables, prototipo arquitectónico	Diseño detallado componentes. Especificaciones de codificación
Áreas de Enfoque	Selección de tecnologías, Requerimientos no funcionales (QoS), Manejo de riesgos	Requerimientos funcionales

Arquitectura Vs. Diseño

- + La arquitectura envuelve un conjunto de decisiones estratégicas de diseño, lineamientos, reglas y patrones que restringen el diseño y la implementación de un software.



Las decisiones de arquitectura causan un alto impacto en los proyectos de IT

Discusión

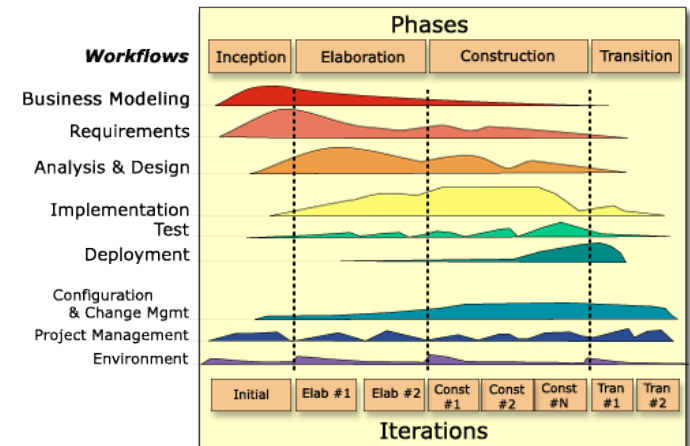
- + Cuales son los principios fundamentales en los métodos de desarrollo de software modernos?



Arquitectura y Procesos de Desarrollo

Principios Fundamentales de Procesos Modernos

- + Desarrollo iterativo e incremental.
- + **Conducido por las calidades sistémicas.**
- + **Centrado en la arquitectura.**
- + Dirigido por los casos de uso.
- + Basada en Modelos.
- + Mejores prácticas de diseño.



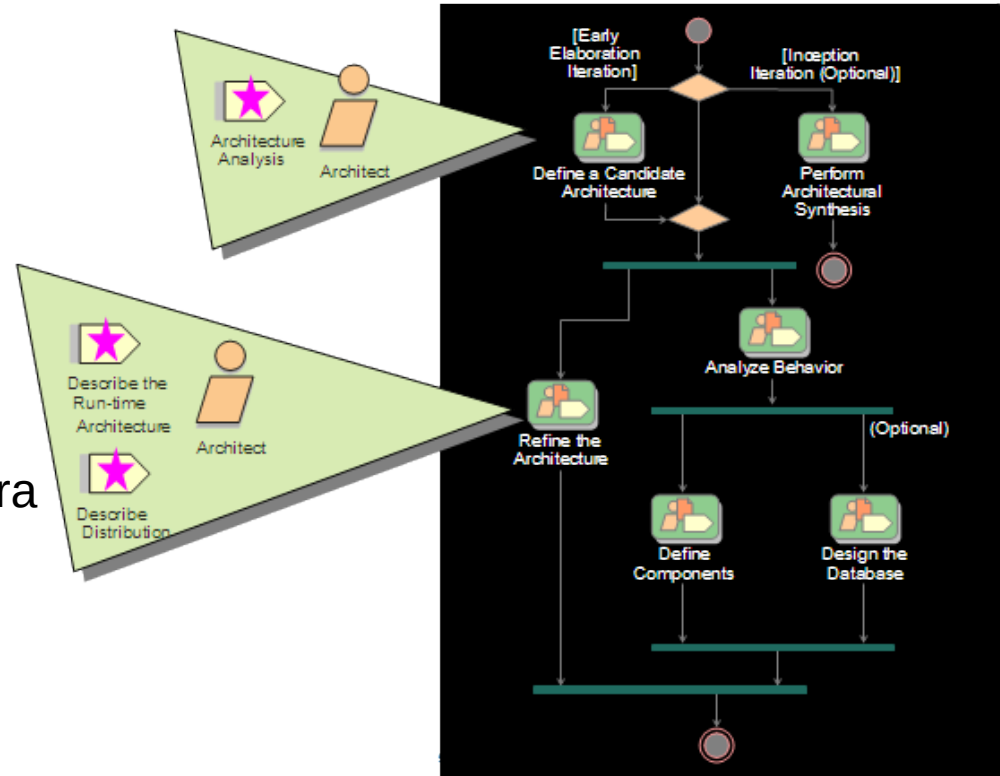
Arquitectura y Procesos de Desarrollo

+ Que es un Proceso de Arquitectura?

+ Rational Unified Process:

+ Secuencia de actividades que conllevan a la producción de artefactos arquitectónicos:

- Descripción de arquitectura
- Prototipo arquitectónico



Arquitectura y Procesos de Desarrollo

Rational Unified Process:

En el proceso de definición de arquitectura se producen:

- + Arquitectura Inicial.
- + Arquitectura de Referencia.
- + Documento de Descripción de arquitectura (SAD):
 - Subsistemas
 - Componentes
 - Arquitectura Runtime.
- + Guías para el proyecto y estándares de Diseño.

SunTone AM:

Adicionalmente se producen:

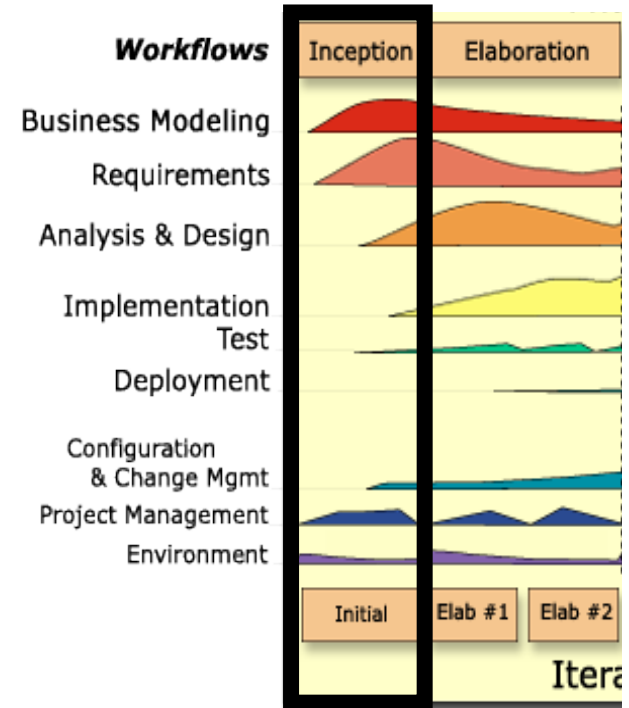
- + Matriz Tecnológica de Layers y Tiers
- + Template de Arquitectura



Definición de Arquitectura en RUP

Fase de Inicio

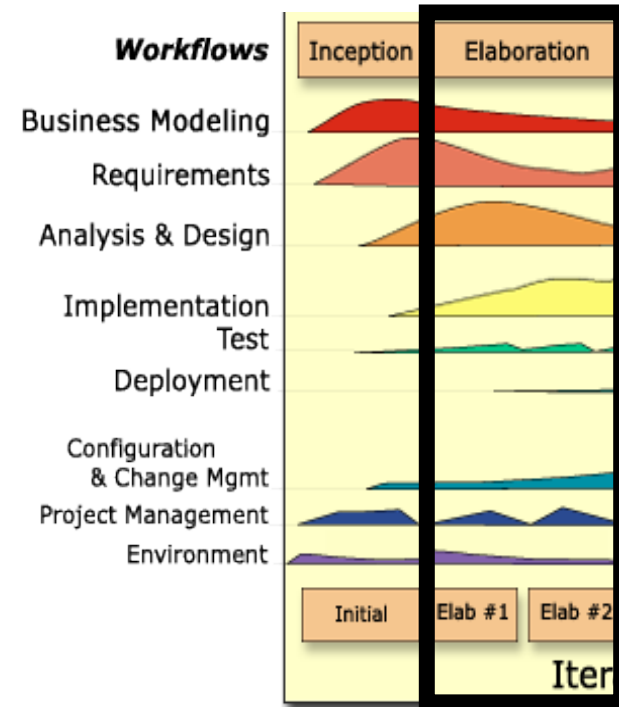
- + Con respecto a la arquitectura, en la fase de inicio de los proyectos se establece:
 - **Requerimientos no-funcionales**
 - **Lista de riesgos y restricciones**
 - **Arquitectura inicial**



Definición de Arquitectura en RUP

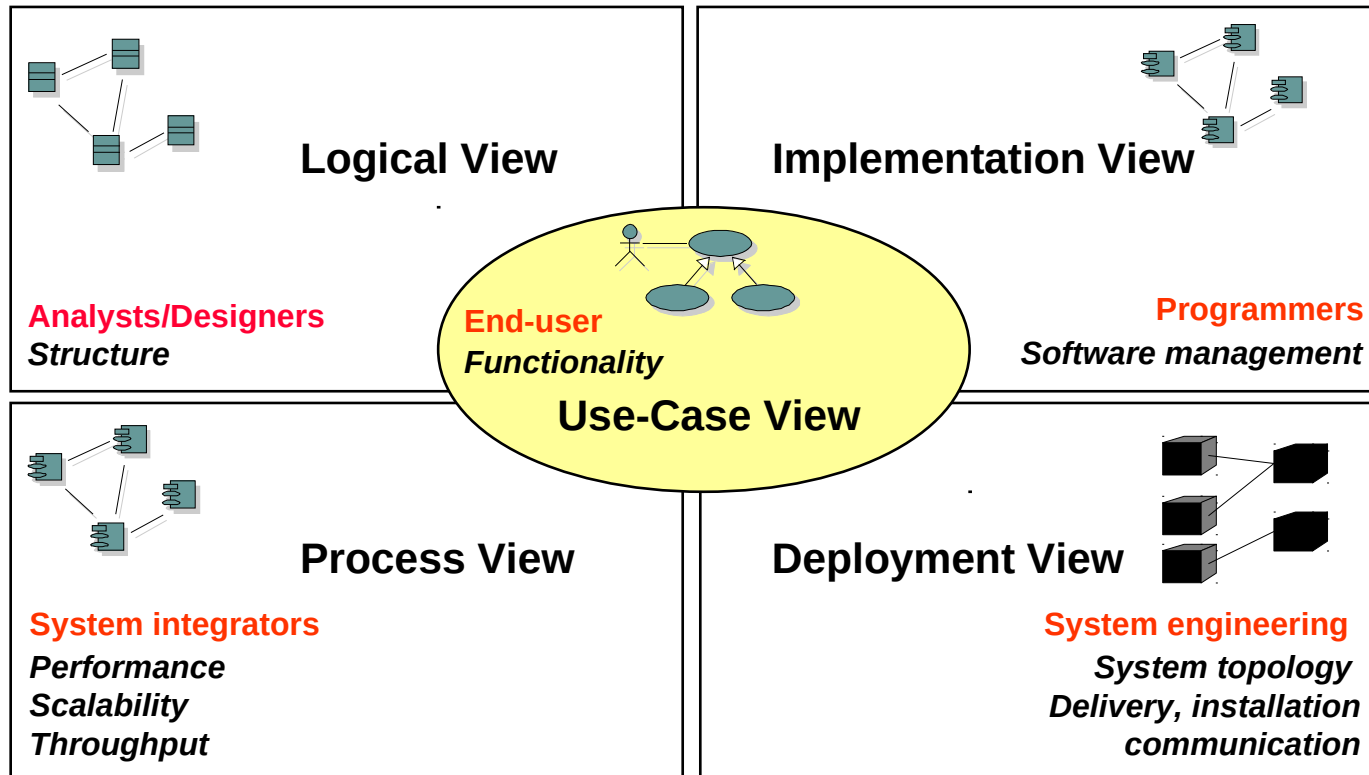
Fase de Elaboración

- + Con respecto a la arquitectura, en la fase de elaboración se establece:
 - **Arquitectura línea base.**
- + **Entregables:**
 - Documento de Definición de Arquitectura.
 - Prototipo evolutivo de arquitectura.
 - Guías y Estándares de Diseño.

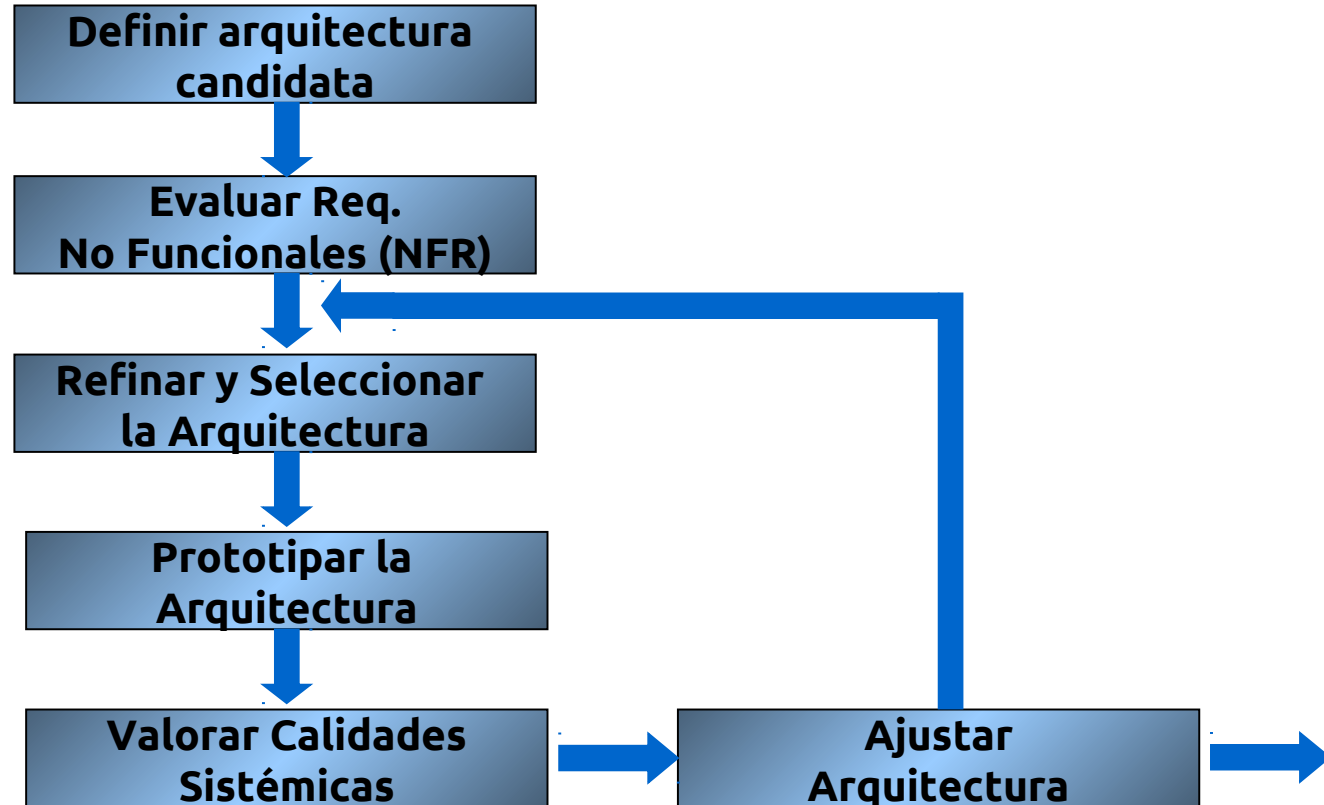


Definición de Arquitectura en RUP

- + **Modelo de Vista 4+1**
- + Framework para Descripción de Arquitectura, basado en vistas lógicas y físicas UML y una vista funcional de casos de uso.

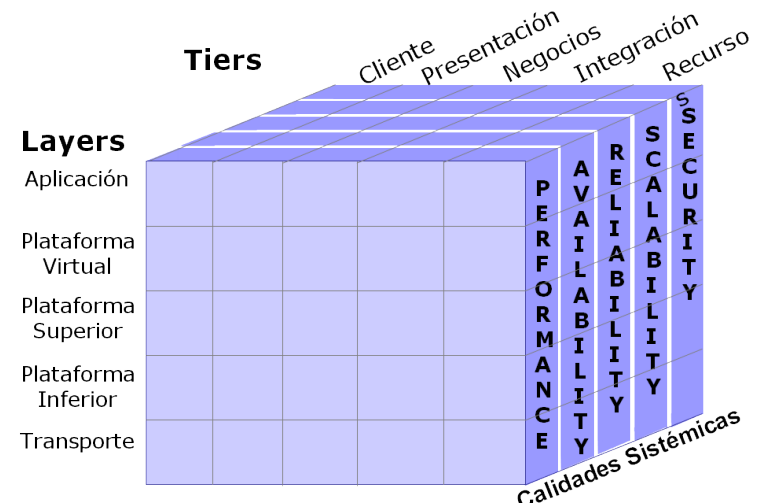


Definición de Arquitectura en RUP

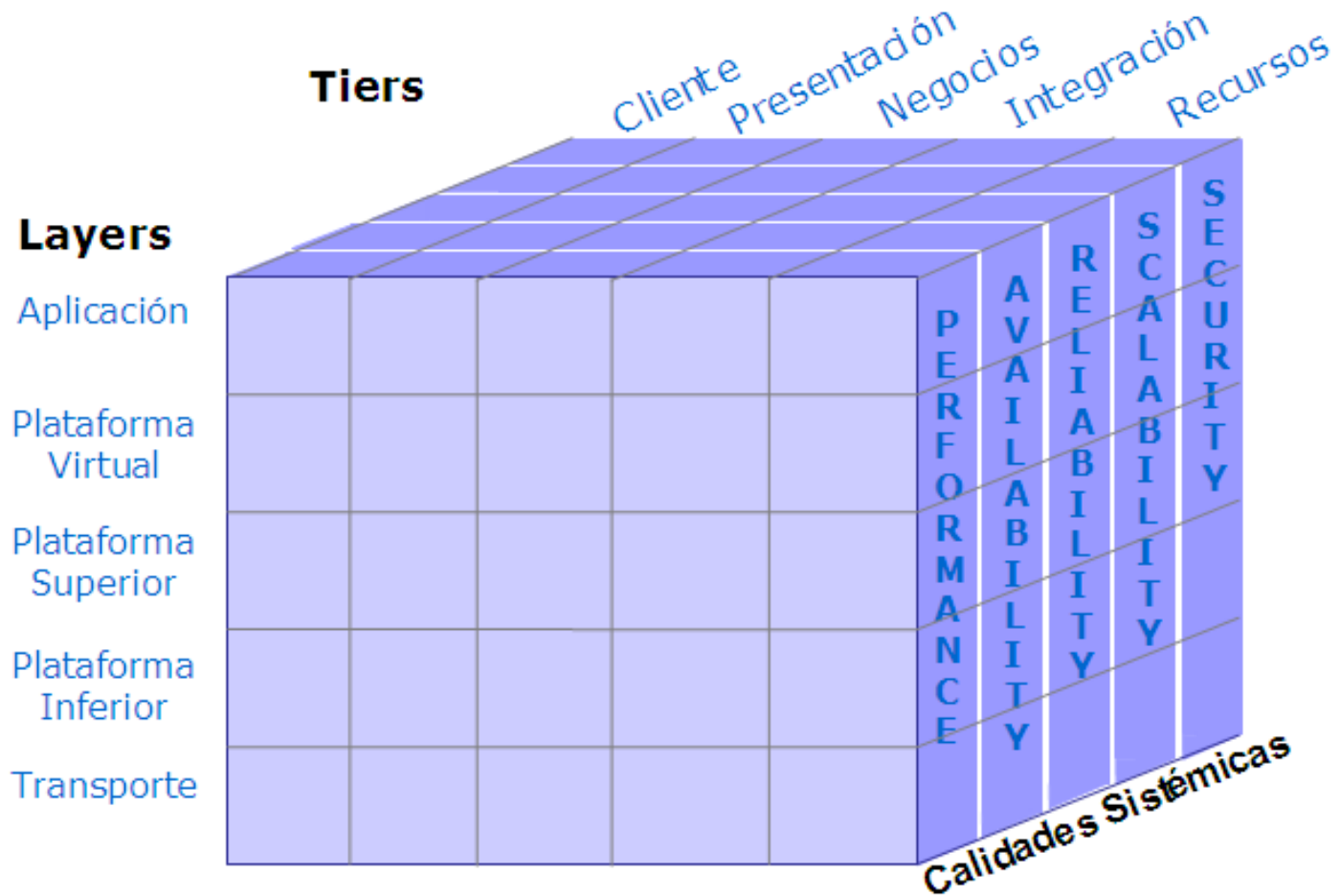


Definición de Arquitectura en SunTone AM

- + Metodología de desarrollo de software análoga al Unified Process (UP) con un fuerte énfasis en Calidad de Servicio y Patrones de diseño.
- + El cubo: framework conceptual, el cual provee una vista tridimensional:
 - Tiers lógicos
 - Layers tecnológicos
 - Calidades sistémicas



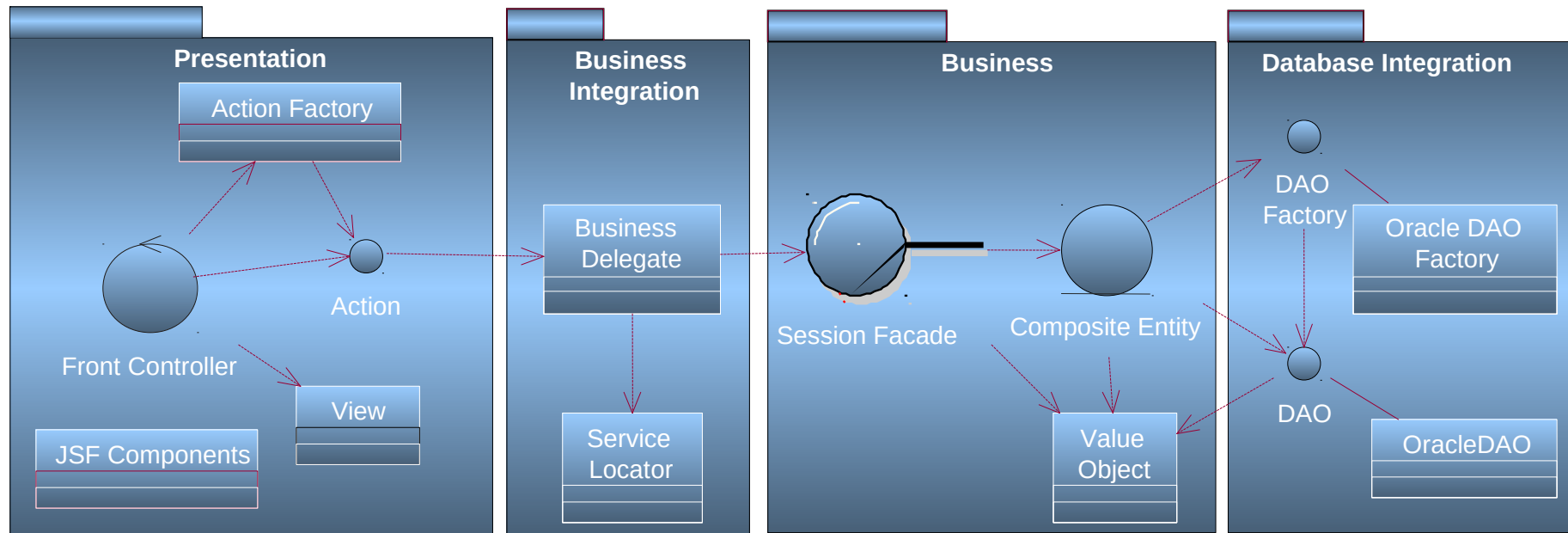
Definición de Arquitectura en SunTone AM



Definición de Arquitectura en SunTone

+ Principios Arquitectónicos

- + La arquitectura es primordialmente necesaria para crear un framework para el desarrollo basado en patrones y para la entrega de calidades sistémicas predecibles.

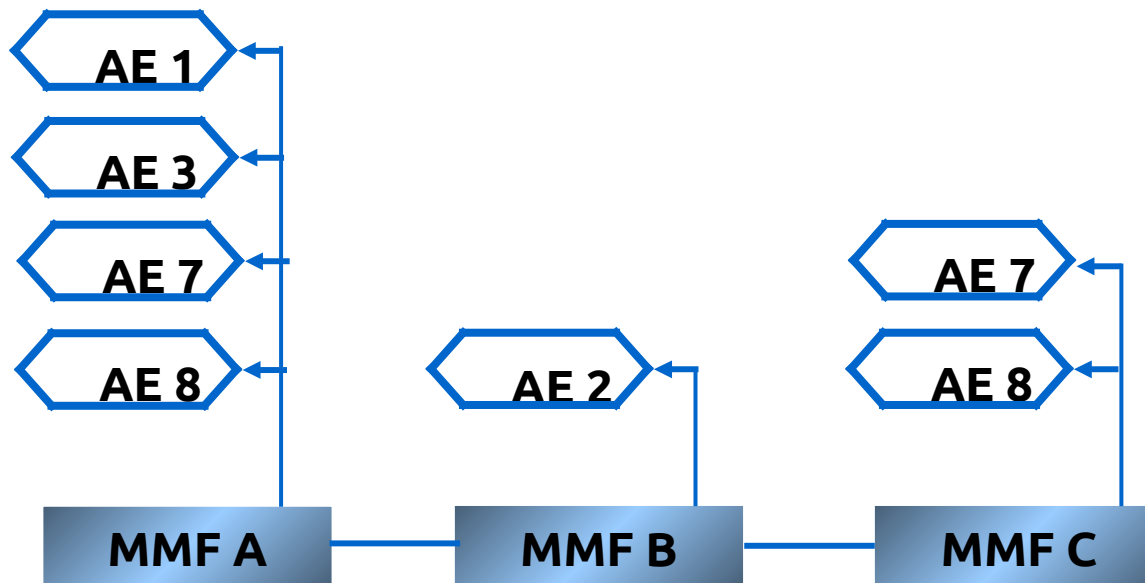


Definición de Arquitectura en IFM

- + **Principios Arquitectónicos**
- + El proceso de creación de arquitectura debe ser un proceso de creación de valor.
- + La arquitectura se descompone en elementos arquitectónicos (AEs).
- + La arquitectura se crea incrementalmente acorde a un proceso secuencial dirigido por el ROI.

Definición de Arquitectura en IFM

- + **Principios Arquitectónicos**
- + La instanciación de los elementos arquitectónicos (AEs) se realiza incrementalmente acorde a la secuencia de MMFs, determinada por el ROI.



Ejemplo de Definición de Arquitectura

- + **eBank Trusted Hosting**
- + **Workshop de Arquitectura y Diseño de Aplicaciones J2EE**
- + **Curso WS50I - Lucasian Labs Ltda.**
- + Entidad que presta el hosting de los servicios de banca personal en Internet para un grupo de bancos.

eBank
Trusted Hosting



Ejemplo de Definición de Arquitectura

+ Dado un conjunto de requerimientos primarios

Requerimientos Primarios

A continuación se presentan los requerimientos primarios definidos en el documento de visión del proyecto:

1. Los clientes deben acceder al sitio de los Bancos, identificándose con un usuario y una contraseña.
2. La autenticación la debe realizar el servidor LDAP de eBank, Sun ONE LDAP Directory 5.1.
3. Si el login es exitoso, se presenta una pantalla inicial con los datos personales de los usuarios, y con el resumen de las cuentas débito y crédito del usuario de cada uno de los bancos.
4. Los datos personales de los usuarios están en el CRM de los Bancos, los cuales están en una base de datos centralizada. Actualmente se está utilizando un CRM Inhouse (llamado Joustomer, el cual expone sus servicios de negocios como Stateless Session EJBs. Como política corporativa, el CRM es la única aplicación que contiene los datos personales de los usuarios.
5. Los clientes deben ver los detalles de las transacciones realizadas con cada una de sus cuentas, corriente y ahorros.
6. Las transacciones se deben presentar en bloques de 20 registros, sin embargo tenga en cuenta que los servicios de negocio por factores de latencia de red, envían bloques de 200 registros.
7. Los clientes pueden realizar transferencias de efectivo entre cuentas registradas de los bancos.
8. Los débitos de las transferencias se realizan obligatoriamente en línea.
9. Se debe implementar un servicio de transferencias interbancarias, a través de una nueva entidad, MSBank, la cual presta el hosting para transferencias. Las transferencias interbancarias se realizan en batch, 1 vez al día a las 11:00 PM.
10. Los clientes pueden decidir cancelar las transferencias interbancarias al final de la sesión de Internet.
11. Si los clientes salen de la interfaz Web sin confirmar las transferencias interbancarias, entonces tienen plazo para confirmar las transferencias durante el día. Si no las confirman durante el día, un proceso a las 10:00 PM elimina las transacciones interbancarias pendientes por confirmar.
12. MSBank ha expuesto un Web Service en Internet, para realizar las transferencias, el cual recibe un archivo XML con el total de las transferencias de cada banco.
13. MSBank tiene una base de datos que le permite validar las cuentas de todos los bancos que tiene suscritos, y puede rechazar algunas de las transferencias. Si hay un problema durante una transferencia, entonces se debe enviar un e-mail al cliente notificándole el evento.
14. Los clientes pueden definir alertas para las transferencias mayores a cierto monto.
15. El área de producción de los Bancos, ha determinado que se debe tener métricas de calidad de servicio en tiempo real respecto al tiempo de respuesta de los servicios de la interfaz Web.
16. Existe un usuario administrador de negocios al cual debe poder analizar las métricas de calidad de servicios, para cada una de las transacciones financieras.
17. Los Bancos han decidido que eBank va a proveer adicionalmente el hosting de los Web Site corporativos, pero no desean que se mezclen los servicios financieros en Internet con los del Web Site.
18. Los servicios financieros deben tener alta disponibilidad ante fallos del servidor de aplicaciones.
19. El departamento de mercadeo de eBank, pronostica que el 10% de los usuarios de los dos Bancos, (150.000 usuarios), utilizará la interfaz Web desde el primer mes y que la demanda en los próximos 6 meses debe aumentar el número de clientes Internet en un 200% aproximadamente.
20. eBank cobra a los bancos 75 centavos de dólar por cualquier tipo de transferencia. Los bancos cobran 1 dólar a los clientes por transferencia.


Ejemplo de Definición de Arquitectura

- + Identificación de requerimientos funcionales y de calidad de servicio (QoS)

Solución del Workshop, Fase 1: Definición de Arquitectura

A continuación se presenta una solución propuesta para la fase 1 del workshop, para cada uno de los entregables.

Systemic Quality	Requirements																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Performance				X				?							#				X	
Reliability				X				?	X											
Availability		X		X				X	X								X	X		
Throughput				X				?											X	
Manageability																X	#			
Security				X					X											
Identity	X	X																		
Authority																				
Confidentiality																				
Integrity																				
Auditability															X					
Testability																				
Serviceability																				
Planability																				
Realizability																				
Escalability																			X	


Lucasian Labs - www.lucasian.com
 The Software Architecture and Engineering Company
Workshop de Arquitectura y Diseño de Aplicaciones J2EE
 Rev. 2003-Q3

Página 8

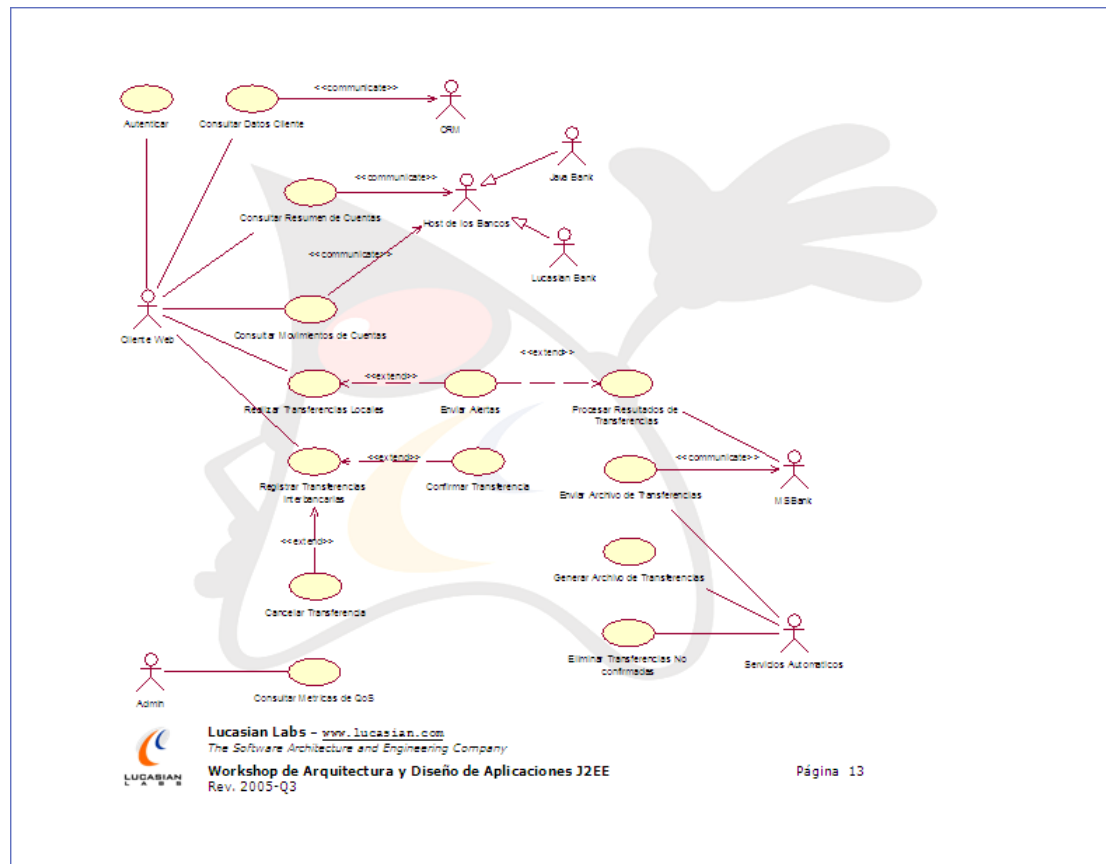
Ejemplo de Definición de Arquitectura

+ Identificación de supuestos, riesgos y restricciones

Supuestos
Los sistemas de los bancos (CRM, host de los bancos) son confiables y de alto desempeño.
La comunicación con los sistemas de los bancos y MSBank es confiable y segura.
eBank tiene un canal con suficiente ancho de banda para salir a producción.
La aplicación desarrollada va a ser para los dos bancos.
La información de MSBank está actualizada. Evitar problemas de validación de cuentas.
Se asume que existe una aplicación que analiza la información de calidad de servicio, ejemplo HP OpenView.
Se asume que los créditos de las transferencias entre cuentas de la misma entidad se realizan en línea.
Se asume que la lógica de negocios de los bancos está en servidores de aplicaciones J2EE del mismo fabricante.
Los procesos batch no tienen dependencias para inicio.
Restricciones
La autenticación se debe realizar con Sun ONE LDAP Directory.
Utilizar el CRM, JCustomer para obtener información de los clientes.
Se transfieren bloques de registros de 200, por la latencia de red.
El debito de las transferencias se realiza en línea.
Las transferencias enviadas a MSBank son en Batch, Web Services y archivos en formato XML.
La solución debe ser J2EE.
Los servicios del CRM están expuestos como Stateless Session EJBs.
Los servicios Web de los Bancos están separados de los servicios financieros.
Los procesos Batch se ejecutan a las 10:00 PM y 11:00 PM.
Riesgos Tecnológicos
Los servicios externos no estén disponibles.
La demanda aumente por encima de los estimativos.
El tráfico del sitio estático agote el ancho de banda de los servicios financieros.

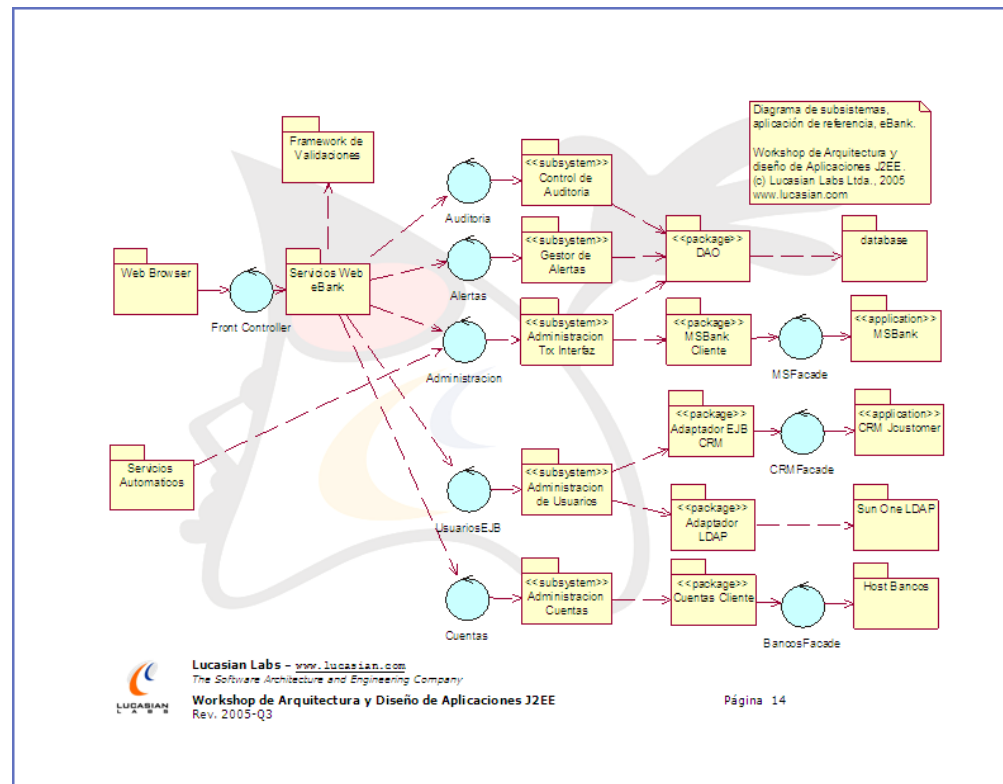
Ejemplo de Definición de Arquitectura

+ Identificación de Actores y Casos de Uso primarios



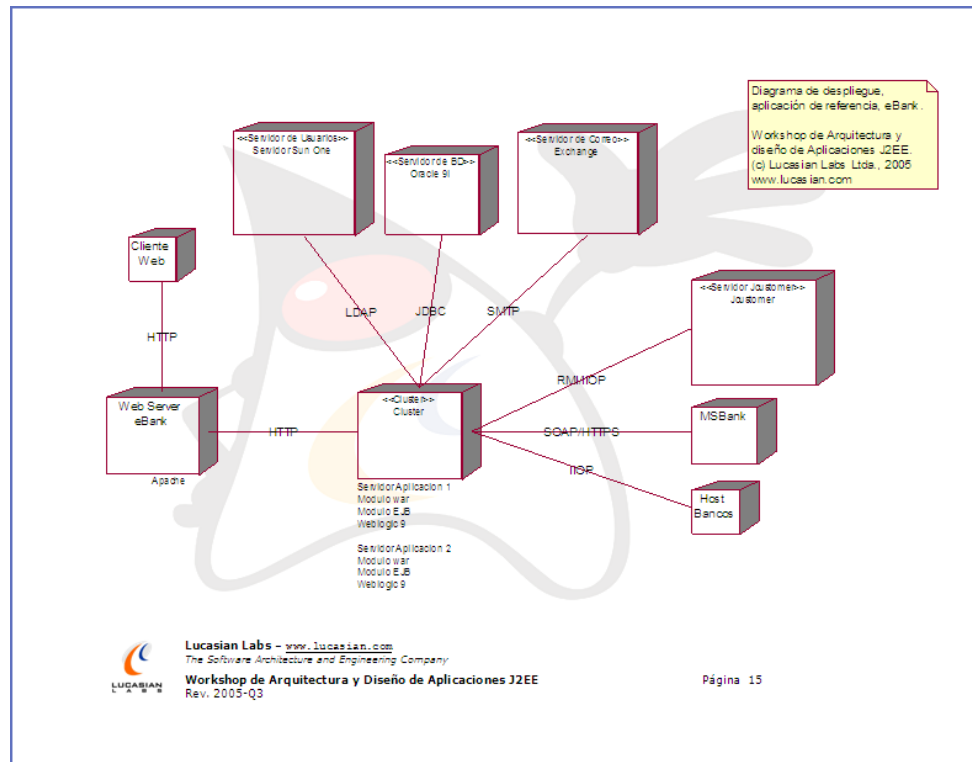
Ejemplo de Definición de Arquitectura

- + Arquitectura Lógica. Identificación de tiers lógicos, subsistemas y paquetes



Ejemplo de Definición de Arquitectura

+ Diseño de Arquitectura Runtime. Diagrama de Despliegue.



Ejemplo de Definición de Arquitectura

+ Plataforma Tecnológica. Definición de la matriz tecnológica de layers y tiers

Layers	Cliente	Presentación	Negocios	Integración	Recursos	
Aplicación	HTML, CSS, JavaScript	Páginas JSF, Java Beans, Tag Libraries,	Stateless Session Beans, Timer Bean, MDBs	Data Access Objects	Tablas y Secuencias	Web Service, Session Beans
Plataforma Virtual	HTML 4.0, Java Script 1.2	J2EE 1.4: Java Servlets 2.3 Java Naming and Directory Interface 1.2.1	J2EE 1.4: Enterprise Java Beans 2.0 JTA 1.0.1 JMS 1.x Log4J 1.2.8	JDBC 2.1	SQL 99	J2EE 1.3: Enterprise Java Beans 2.0 JTA 1.0.1 SOAP 1.x
Plataforma Superior	Internet Explorer 6 o superior	Jakarta Tomcat 5.x / Apache 1.3.x	JBoss 4.x		Oracle DB 10g	Host de los Bancos, JCustomer, SunOne LDAP Directory, MS Bank



Lucasian Labs - www.lucasian.com
The Software Architecture and Engineering Company

Workshop de Arquitectura y Diseño de Aplicaciones J2EE
Rev. 2005-Q3

Página 16

- + Los requerimientos no funcionales son fuentes comunes de riesgo...

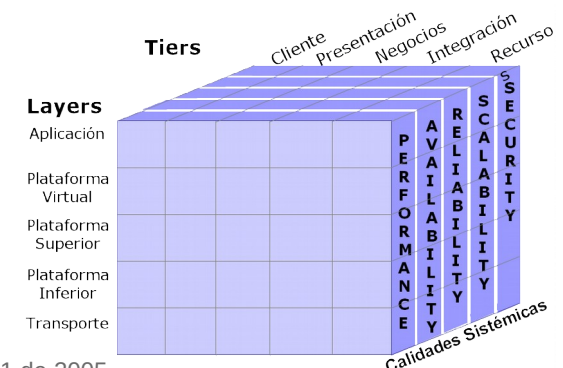
por que?





Calidades Sistémicas

- + El manejo inadecuado de los requerimientos no funcionales, es una de las fuentes más importante de riesgo en los proyectos:
 - Reglas de negocio de alta complejidad.
 - Calidades sistémicas
 - Seguridad
 - Rendimiento
 - Escalabilidad
 - Disponibilidad
 - Extensibilidad
- + La calidad de servicio (QoS = Quality Of Service) es un riesgo primario relacionado con la arquitectura.



Calidades Sistémicas - Manifiestas

- + Observables por los usuarios del sistema.
- + **Performance.** Tiempo de respuesta desde el punto de vista del usuario.
- + **Reliability.** Grado de probabilidad de realizar operaciones correctamente.
- + **Availability.** Porcentaje de tiempo que un sistema puede procesar solicitudes.



Calidades Sistémicas - Operacionales

Observables cuando el sistema está operando en producción.


- + **Throughput.** Solicitudes atendidas por unidad de tiempo.
- + **Manageability.** Cantidad inversa de esfuerzo para realizar labores administrativas.
- + **Serviceability.** Esfuerzo para actualizar el sistema para reparar errores.
- + **Security.** Prevención de uso indeseado, por abuso o uso inapropiado:
 - Identidad
 - Autoridad
 - Confidencialidad
 - Auditabilidad
 - Integridad
- + **Testability.** Esfuerzo invertido para detectar y aislar errores.



Calidades Sistémicas - Evolutivas

- + Relacionadas con el comportamiento del sistema cuando sufre algún cambio.
- + **Escalability**. La habilidad para soportar la calidad de servicio requerida conforme la carga aumenta.
- + **Flexibility**. Esfuerzo ahorrado cuando se hace un cambio de configuración.
- + **Portability**. Esfuerzo ahorrado cuando se migra a una infraestructura diferente.
- + **Reusability**. Esfuerzo ganado en la utilización de componentes existentes.
- + **Extensibility**. Esfuerzo ahorrado para adicionar nuevas funcionalidades.
- + **Maintainability**. Esfuerzo ahorrado para revisar y corregir errores.

Lecciones Aprendidas en Consultoría

- + Defina una persona o un grupo de personas experimentadas, encargadas de definir y validar arquitectura de sus proyectos.
 - + Establezca los requerimientos de calidad de servicio con los expertos del dominio y con los usuarios finales.
 - + Involucre al equipo de trabajo en el proceso de definición de arquitectura.
 - + Documente y comunique la arquitectura y lineamientos de diseño y logre aceptación. No la imponga.
 - + Sea firme con las decisiones, valore impactos e identifique riesgos.
- 
- A stylized illustration in the bottom right corner. It features a black silhouette of a person in a suit, holding a large, glowing yellow lightbulb. The lightbulb has a black base and a yellow glow with radiating lines. The person is standing on a black, angular base. The background of the illustration is a light blue square.





Lecciones Aprendidas en Consultoría

-



Fundamentos de Definición de Arquitectura de Software

Mauricio Naranjo
Chief Architect
mnaranjo@lucasian.com

Lucasian Labs Ltda.
www.lucasian.com

