



ARQUITECTURA Y DISEÑO DE SISTEMAS

ARQUITECTURA DE SOFTWARE – PARTE 2

ELSA ESTEVEZ

UNIVERSIDAD NACIONAL DEL SUR

DEPARTAMENTO DE CIENCIAS E INGENIERIA DE LA COMPUTACION



- 1 ARQUITECTURA Y LA MANTENIBILIDAD DEL SOFTWARE
- 2 RECOMENDACIONES PARA UNA BUENA ARQUITECTURA
- 3 ROL DEL ARQUITECTO
- 4 ACTIVIDADES DEL ARQUITECTO
- 5 CLASES DE ARQUITECTOS
- 6 PROCESO DE DEFINICIÓN DE ARQUITECTURA
- 7 ARQUITECTURA Y LOS CICLOS DE VIDA

OBJETIVOS



Recordando una definición...

Toda la arquitectura es diseño, pero no todo el diseño es arquitectura. La arquitectura representa las decisiones de diseño que le dan forma a un sistema, donde la **importancia** es medida por el **costo del cambio**.

Grady Booch





... hacer que todo sea fácil de cambiar hace que el sistema entero sea muy complejo...

Ralph Johnson

DESAFÍO

¿Cómo se puede acotar la complejidad incrementando la flexibilidad del sistema?





- Reconocer **dónde es necesaria la flexibilidad** para reducir el impacto y costo del cambio
- Tratar de **diferir la obligación de tomar una decisión específica** de arquitectura que nos bloquearía en una solución muy particular hasta que tengamos el conocimiento necesario que nos permita tomar una decisión lo más fundamentada posible
- **Aislar las decisiones de las cuales no estamos seguros**, asegurando que dichas decisiones iniciales sean fáciles de cambiar a medida que vayan surgiendo el conocimiento e información suficientes para asegurar una buena alternativa de solución.



Identificar las costuras en un sistema involucra identificar líneas claras de demarcación de la arquitectura. En ambos márgenes de aquellas líneas, se encuentran componentes que pueden cambiar independientemente, sin afectar a los componentes del otro lado, siempre y cuando los componentes a ambos lados cumplan con el contrato especificado por la interfaz.

Booch, UML User Guide

Modularidad combinada con **patrones de diseño** y los **principios SOLID** representan nuestra mejor esperanza para **minimizar el costo del cambio**.



- 1 ARQUITECTURA Y LA MANTENIBILIDAD DEL SOFTWARE
- 2 RECOMENDACIONES PARA UNA BUENA ARQUITECTURA
- 3 ROL DEL ARQUITECTO
- 4 ACTIVIDADES DEL ARQUITECTO
- 5 CLASES DE ARQUITECTOS
- 6 PROCESO DE DEFINICIÓN DE ARQUITECTURA
- 7 ARQUITECTURA Y LOS CICLOS DE VIDA

RECOMENDACIONES PARA EL PROCESO



- La arquitectura debería ser el producto de uno o un pequeño grupo de arquitectos
 - ✓ Integridad conceptual
 - ✓ Consistencia técnica
- El arquitecto debería basar la arquitectura en **una lista priorizada de requerimientos de atributos de calidad**
- La arquitectura debería **documentarse usando vistas**, abarcando los intereses de los stakeholders más importantes.
- La arquitectura debería ser **evaluada en su capacidad de cumplir con atributos de calidad** más importantes definidos para el sistema
- La arquitectura debiera **conducir a una implementación incremental**
 - ✓ Para evitar tener que integrar todo de una vez
 - ✓ Para descubrir problemas en forma temprana

RECOMENDACIONES PARA LA ESTRUCTURA – 1



- La arquitectura debe ofrecer **módulos bien definidos** cuyas responsabilidades funcionales respeten los principios de **ocultamiento de la información** y la **separación de intereses**
- Los atributos de calidad deberían cumplirse **usando patrones** y tácticas arquitecturales bien conocidas
- La arquitectura no debería depender de una versión particular de un producto comercial o herramienta.
 - Cambios a versiones más nuevas deberían ser directos y poco costosos.
- En un escenario de concurrencia, los módulos pueden ser instanciados múltiples veces y estar corriendo en paralelo
- Cada proceso debería ser escrito de tal forma que su asignación a un procesador específico pueda ser cambiada.

RECOMENDACIONES PARA LA ESTRUCTURA – 2



- La arquitectura debe ofrecer una pequeña cantidad de formas para que los componentes interactúen.
- La arquitectura debe contener un conjunto específico (y pequeño) de áreas de contención de recursos
 - ✓ Si la performance es crítica, el arquitecto debería producir (y forzar) límites de tiempo para las funcionalidades más representativas.
 - ✓ Si la utilización de la red es importante, el arquitecto produciría (y forzaría) para cada equipo de desarrollo un conjunto de lineamientos que resulten en un tráfico de red mínimo.



- 1 ARQUITECTURA Y LA MANTENIBILIDAD DEL SOFTWARE
- 2 RECOMENDACIONES PARA UNA BUENA ARQUITECTURA
- 3 **ROL DEL ARQUITECTO**
- 4 ACTIVIDADES DEL ARQUITECTO
- 5 CLASES DE ARQUITECTOS
- 6 PROCESO DE DEFINICIÓN DE ARQUITECTURA
- 7 ARQUITECTURA Y LOS CICLOS DE VIDA

COMO LLEGAR A SER UN ARQUITECTO DE SOFTWARE?

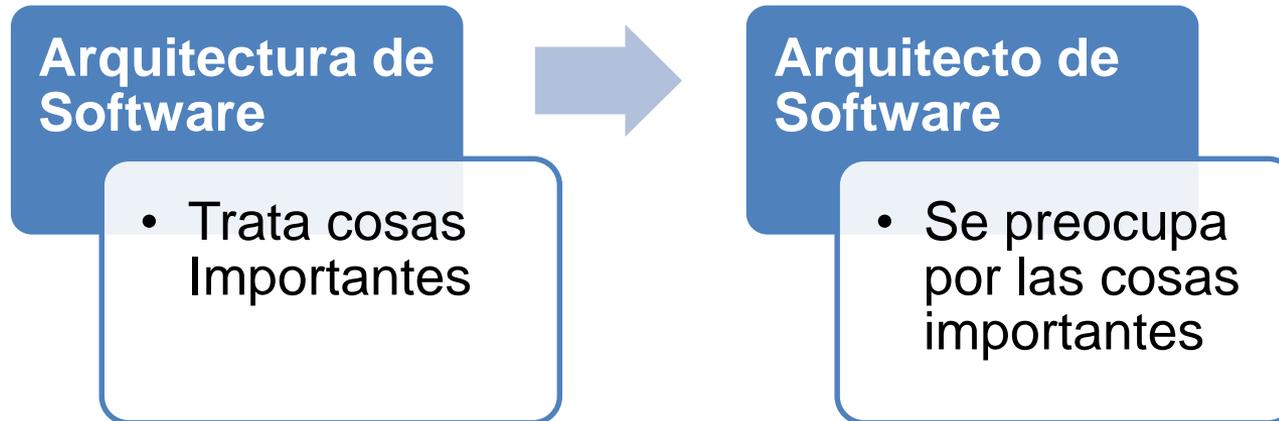


- Llegar a ser un Arquitecto de Software no es algo que ocurre simplemente de la noche a la mañana.
- es un rol, no una categoría en el trabajo.
- es un proceso evolutivo donde gradualmente se gana la experiencia y la confianza que se necesita para asumir el rol
- a diferencia del término “Desarrollador de Software” que es bastante bien entendido, “Arquitecto de Software” no lo es.

DE QUE SE OCUPA?



Una de las definiciones de Arquitectura de Software propuestas por Ralph Jonhson dice que “trata las cosas importantes”, entonces el arquitecto sería la persona que “se preocupa por las cosas importantes” (??)



DOS PRECONCEPTOS...



Los arquitectos son bastante más lentos que los desarrolladores en obtener una solución, especialmente si el problema es simple.

Los arquitectos generalmente vienen con soluciones complejas y costosas, especialmente si el problema es simple.



DOS TIPOS DE ARQUITECTOS



Martin Fowler propone dos tipos de Arquitectos:

Architectus Reloadus

Especie de arquitecto sacado de
Matrix Reloaded

Vs

Architectus Oryzus

Ejemplificado por Dave Rice,
conocido arquitecto de
ThoughtWorks



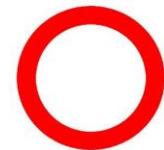
ARQUITECTO RELOADUS

- Es la persona que toma todas las decisiones importantes.
- Cree que es necesaria una única mente para asegurar la integridad conceptual del sistema.
- Cree que los miembros del equipo no tienen las habilidades suficientes para tomar dichas decisiones.
- Pregona que dichas decisiones tienen que ser tomadas de manera temprana, tal que todos en el equipo puedan seguirlas.



ARQUITECTO ORYZUS

- Está al tanto de lo que está ocurriendo en el proyecto: detectando problemas importantes y resolviéndolos antes que se transformen en problemas serios.
- Intensa colaboración en el equipo.
- Es el mentor del equipo de desarrollo.
- Es el guía de la aventura de desarrollar un software.





- 1 ARQUITECTURA Y LA MANTENIBILIDAD DEL SOFTWARE
- 2 RECOMENDACIONES PARA UNA BUENA ARQUITECTURA
- 3 ROL DEL ARQUITECTO
- 4 ACTIVIDADES DEL ARQUITECTO**
- 5 CLASES DE ARQUITECTOS
- 6 PROCESO DE DEFINICIÓN DE ARQUITECTURA
- 7 ARQUITECTURA Y LOS CICLOS DE VIDA

ACTIVIDADES DEL ARQUITECTO





- Los interesados también deben definir los requerimientos no funcionales (o cualidades del sistema, o atributos de calidad) que necesitan.
- A veces responden con frases como: “el sistema debe ser rápido”
- Para que se puedan satisfacer, los requerimiento no funcionales deben ser:
 - ✓ Específicos
 - ✓ Medibles
 - ✓ Realizables
 - ✓ Comprobables

GESTIÓN DE REQUERIMIENTOS NO FUNCIONALES – 2



EJEMPLOS

PERFORMANCE	Cantidad máxima de segundos para responder
ESCALABILIDAD	Progresión de usuarios en el tiempo, volumen y concurrencia
DISPONIBILIDAD	Tiempo que el sistema podría estar no disponible por año
SEGURIDAD	Autenticación, autorización, encriptación en la transmisión de la información, etc.
AUDITORÍA	Qué operaciones debieran quedar auditadas. Qué información se requiere mantener.
EXTENSIBILIDAD	Se prevé que haya que agregar o modificar comportamiento.
INTERNACIONALIZACIÓN	Se necesita que el software sea multi-lenguaje.



- Requerimientos + restricciones → Resolución del Problema
- Todo sistema de software tiene una arquitectura, pero no todo sistema de software tiene una arquitectura **definida**.
- La definición de la arquitectura trata de introducir **estructura, lineamientos, principios y liderazgo** a los aspectos técnicos de un proyecto de software.



- Algunos factores a tener en cuenta
 - Costo
 - Relación con proveedores
 - Interoperabilidad
 - Entrega
 - Dependencias con otras tecnologías
 - Curva de aprendizaje
 - Licenciamiento
 - Estrategia de la tecnología
 - Soporte
 - Políticas de actualización
 - Conocimiento del equipo
- La selección de tecnología se trata de manejar riesgos:
 - ✓ Reducir riesgos donde hay incertidumbre o alta complejidad
 - ✓ Introducir riesgo donde haya beneficios para obtener



- La arquitectura funciona?
 - Satisface los requerimientos no funcionales
 - Provee los fundamentos necesarios para el resto del código
 - Funciona como una plataforma para resolver los problemas de negocio subyacentes
- ¿Cómo se verifica si funciona?
 - Construir pruebas de concepto que permitan asegurar el comportamiento de algunos componentes
 - Desarrollar una batería de tests que permitan evaluar los requerimientos no funcionales durante el ciclo de vida del proyecto. De esta manera podemos tener monitoreados estos criterios durante todo el proyecto.
 - Técnicas formales (ATAM)

Como un buen chef, el arquitecto también debe probar lo que produce



El arquitecto de software es el responsable de asegurar que la arquitectura ha sido entendida por todos los interesados en el sistema de software:

- Equipo de desarrollo
- Especialistas en seguridad
- Administradores de bases de datos
- Operaciones / mantenimiento

ACTIVIDADES DEL ARQUITECTO





A menudo, la arquitectura de software es definida y luego pasada al equipo de desarrollo, considerando el desarrollo de software como si fuera una carrera de postas.

Esto es contraproducente porque la **arquitectura necesita ser cuidada y evolucionada a lo largo del proyecto**, ya sea para lograr mejoras o a medida que cambien los requerimientos.

El arquitecto de software es el responsable de conocer toda la película de la arquitectura del sistema.



El arquitecto de software es la persona indicada para tomar el liderazgo técnico del proyecto.

El arquitecto debe asegurar:

- ✓ que todo es tenido en cuenta
- ✓ que el equipo está siendo dirigido en la dirección correcta, de manera continua.



El arquitecto de software debe:

- proveer asistencia a los desarrolladores para resolver problemas particulares o sortear algunos impedimentos que los bloquean,
- aportar su experiencia
- promover que se comparta el conocimiento entre los miembros del equipo



El arquitecto de software debe:

- definir estándares de codificación
- delinear principios de diseño a cumplir
- utilizar herramientas de análisis de código junto con integración continua para evaluar el cumplimiento de los mismos
- testeo unitario automático
- herramientas de análisis de cobertura de tests



Muchos arquitectos son programadores experimentados, así que tiene sentido mantener esas habilidades al día.

El arquitecto podría experimentar el mismo sufrimiento que el resto de los desarrolladores, lo que lo ayudaría a entender cómo la arquitectura es vista desde la perspectiva de los desarrolladores.

Un arquitecto que codifica es un arquitecto más efectivo y más feliz



- 1 ARQUITECTURA Y LA MANTENIBILIDAD DEL SOFTWARE
- 2 RECOMENDACIONES PARA UNA BUENA ARQUITECTURA
- 3 ROL DEL ARQUITECTO
- 4 ACTIVIDADES DEL ARQUITECTO
- 5 CLASES DE ARQUITECTOS**
- 6 PROCESO DE DEFINICIÓN DE ARQUITECTURA
- 7 ARQUITECTURA Y LOS CICLOS DE VIDA



ARQUITECTO DE PRODUCTO

- es el responsable de la entrega de una o más versiones (releases) de un producto de software a clientes externos
- supervisa la integridad técnica del producto
- debe identificar los stakeholders usuarios antes de la primera versión

ARQUITECTO DE DOMINIO

- se enfoca en un dominio específico, tales como “arquitecto de negocio”, “arquitecto de datos”, “arquitecto de red”, etc.
- son muy valiosos para sistemas grandes, complejos e innovadores ya que aportan su experiencia específica al grupo de arquitectos.



ARQUITECTO DE SOLUCIÓN

- es quien tiene una visión amplia y de alto nivel de la solución global

ARQUITECTO EMPRESARIAL

- tiene responsabilidad de la arquitectura global de los sistemas de la empresa; es decir, de la interconexión de los distintos sistemas utilizados en la empresa



- 1 ARQUITECTURA Y LA MANTENIBILIDAD DEL SOFTWARE
- 2 RECOMENDACIONES PARA UNA BUENA ARQUITECTURA
- 3 ROL DEL ARQUITECTO
- 4 ACTIVIDADES DEL ARQUITECTO
- 5 CLASES DE ARQUITECTOS
- 6 PROCESO DE DEFINICIÓN DE ARQUITECTURA**
- 7 ARQUITECTURA Y LOS CICLOS DE VIDA



- Debe estar conducida por los **intereses de los stakeholders**.
- Debe alentar la **comunicación efectiva** de las decisiones, principios y soluciones arquitectónicas a los stakeholders.
- Debe **asegurar la adherencia** a las decisiones y principios de arquitectura durante todo el ciclo de vida.
- Debe ser **estructurada**. Constar de una serie de pasos o tareas, con una clara definición de los objetivos, entradas y salidas de cada paso.
- Debe ser **pragmática**, considerando cuestiones del mundo real – tiempo, dinero, conocimientos técnicos, consideraciones políticas, etc.
- Debe ser **flexible**, tal que pueda ser ajustada en determinadas circunstancias.
- Debe ser **neutral de la tecnología**.
- Debe **integrarse** en el ciclo de vida de desarrollo de software elegido.
- Debe **alinearse** con buenas **prácticas de la ingeniería de software y estándares de gestión de calidad**



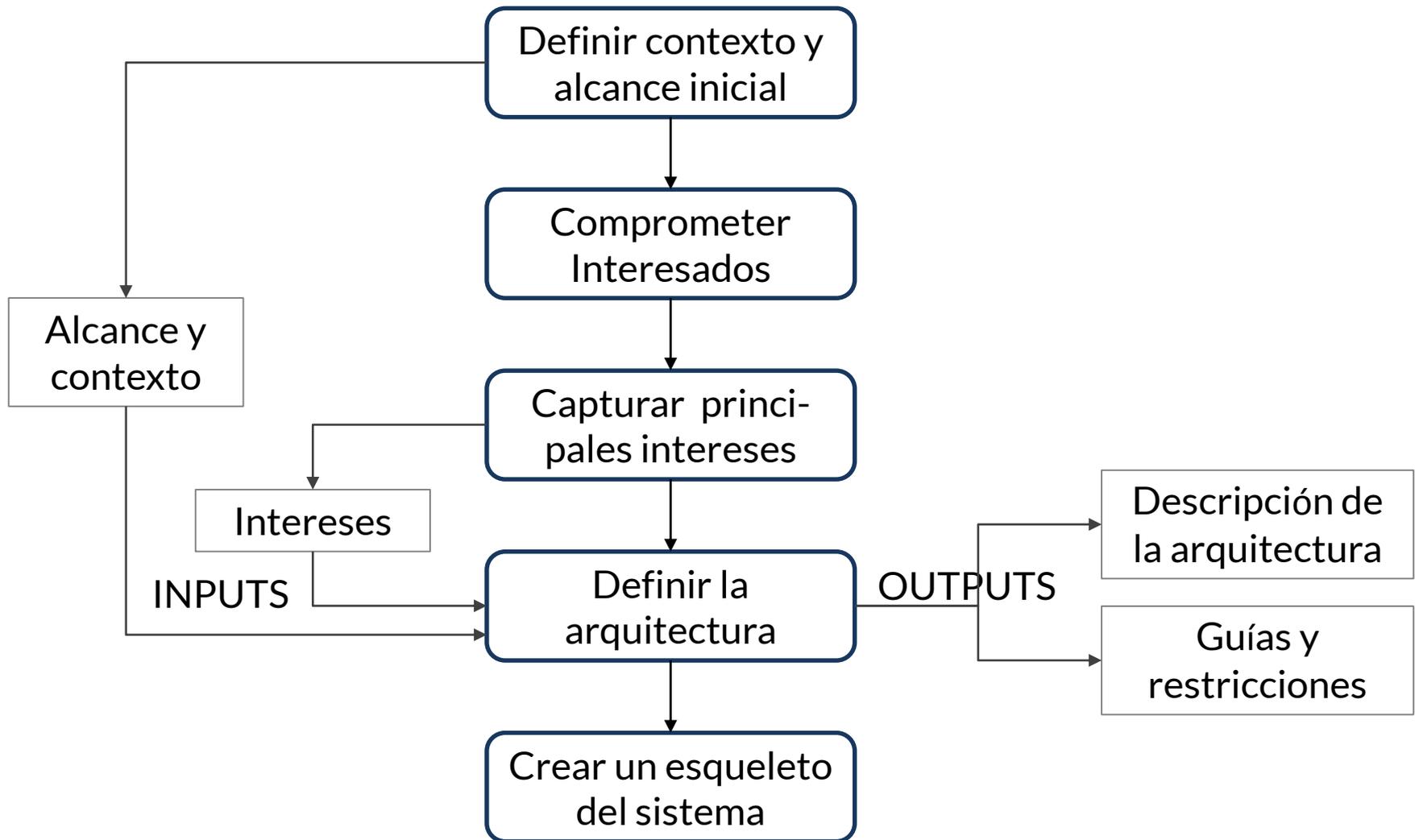
DIRECTAS

- Arquitectura del sistema

INDIRECTAS

- Clarificación de requerimientos y restricciones
- Manejo de las expectativas de los interesados
- Identificación y evaluación de opciones
- Descripción de los criterios de la aceptación para la arquitectura
- Creación de lineamientos de diseño

ACTIVIDADES DE SOPORTE



DESCRIPCIÓN DE ACTIVIDADES – 1



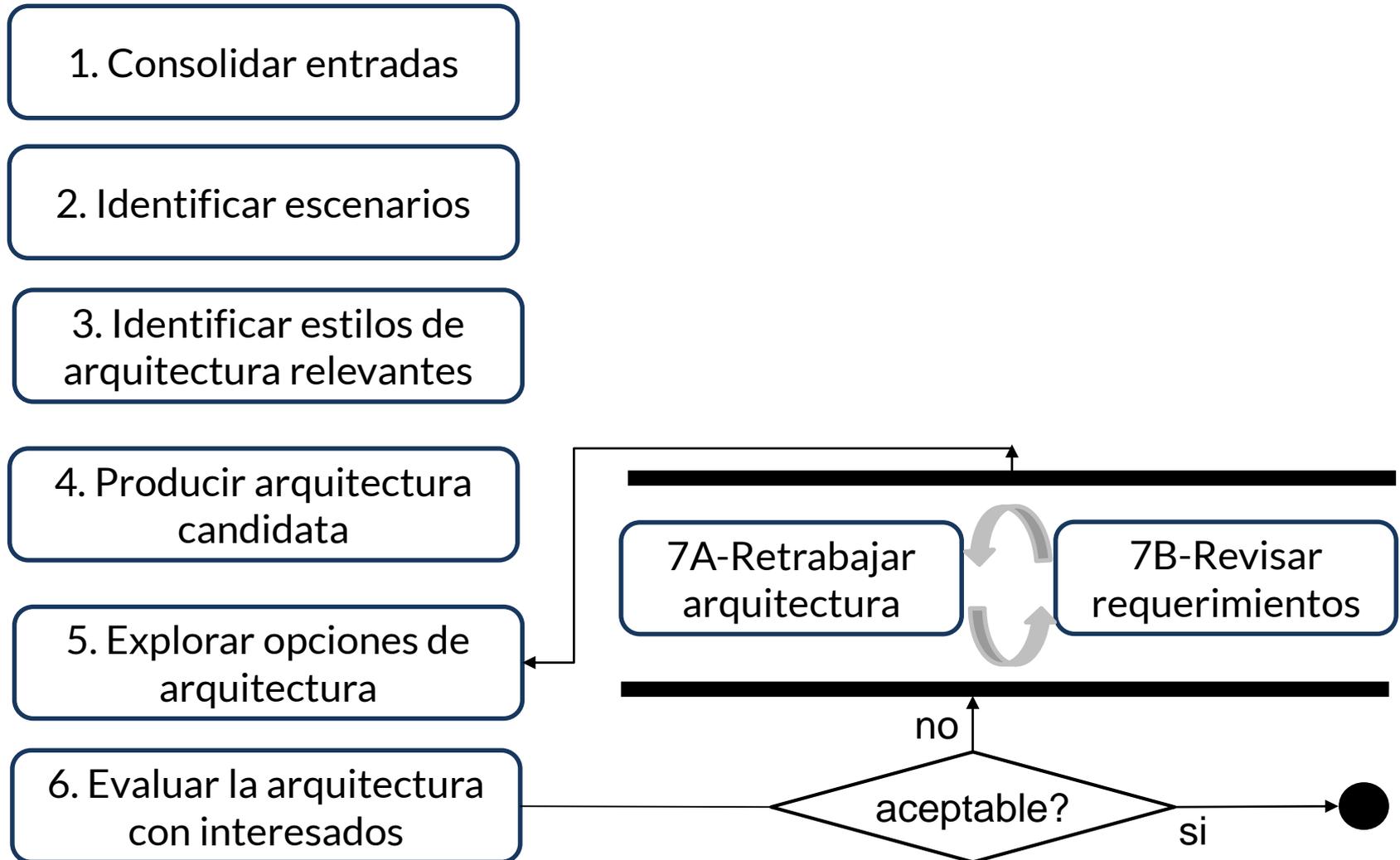
	Entradas	Salidas
Definir el alcance y contexto inicial		
Definir los límites y responsabilidades del sistema, junto con su contexto operacional y organizacional	Visión del cliente. Estrategia organizacional. Arquitectura empresarial.	Metas iniciales del sistema. Alcance de sus responsabilidades. Definición del contexto inicial.
Comprometer a los stakeholders		
Identificar los stakeholders importantes del sistema y establecer una relación de trabajo con ellos	Alcance y contexto. Estructura de la organización	Definición de grupos de stakeholders. Gente comprometida a representar al grupo
Capturar principales intereses		
Entender los intereses de cada grupo de stakeholders y las prioridades que le asignan.	Lista de stakeholders. Alcance y contexto.	Definición inicial del conjunto de intereses priorizados para cada grupo de stakeholders.

DESCRIPCIÓN DE ACTIVIDADES – 2



	Entradas	Salidas
Definir la arquitectura		
Crear la definición de arquitectura para el sistema	Lista de stakeholders. Alcance y contexto.	Definición de la arquitectura. Lineamientos y restricciones
Crear un esqueleto del sistema (opcional)		
Crear una implementación funcionando de la arquitectura que puede evolucionar a la versión final durante la fase de construcción	Definición de arquitectura. Lineamientos y restricciones	El versión limitada del sistema funcionando, ilustrando cómo el sistema resuelve algún aspecto específico.

ACTIVIDADES DE DEFINICIÓN DE ARQUITECTURA



ACTIVIDAD 1



Paso 1	Consolidar las entradas
Objetivos	Entender, validar y refinar las entradas iniciales
Entradas	Entradas al proceso crudas: alcance y contexto, intereses de los stakeholders
Salidas	Entradas consolidadas, con las principales inconsistencias resueltas, preguntas abiertas respondidas y (en un mínimo) áreas que requieren más investigación identificadas.
Actividades	Resolver las inconsistencias detectadas en las entradas crudas Responder preguntas abiertas Profundizar cuando sea necesario para producir una línea base sólida.

ACTIVIDAD 2



Paso 2	Identificar escenarios
Objetivos	Identificar los escenarios que ilustran los requerimientos más importantes del sistema.
Entradas	Entradas consolidadas
Salidas	Escenarios arquitectónicos
Actividades	Producir un conjunto de escenarios que caractericen los más importantes atributos de calidad y puedan ser usados para evaluar cuán bien la arquitectura satisface los requerimientos funcionales y de calidad.

ACTIVIDAD 3



Paso 3	Identificar los estilos/patrones arquitectónicos relevantes
Objetivos	Identificar uno o mas estilos/patrones arquitectónicos que podrían ser usados como base para la organización general del sistema.
Entradas	Entradas consolidadas. Escenarios arquitectónicos.
Salidas	Estilos arquitectónicos a considerar como la base de las estructuras arquitectónicas del sistema.
Actividades	Revisar catálogos de estilos/patrones existentes, considerando aquellos que nos han funcionado bien en el pasado. Identificar estilos/patrones que sean relevantes para la arquitectura.

ACTIVIDAD 4



Paso 4	Producir una arquitectura candidata
Objetivos	Crear una primera aproximación de la arquitectura que refleje los intereses arquitectónicos primarios y que pueda actuar como la base para posteriores evaluaciones y refinamientos.
Entradas	Entradas consolidadas. Estilos arquitectónicos relevantes.
Salidas	Vistas borrador de la arquitectura.
Actividades	Producir un conjunto inicial de vistas de la arquitectura para definir las ideas arquitectónicas iniciales, usando los lineamientos definidos en los estilos/patrones seleccionados.

ACTIVIDAD 5



Paso 5	Explorar las opciones arquitecturales
Objetivos	Explorar las distintas posibilidades arquitectónicas para el sistema y tomar las decisiones claves para elegir entre ellas
Entradas	Entradas consolidadas. Vista borrador de la arquitectura. Escenarios arquitectónicos.
Salidas	Vistas arquitecturales más detalladas y/o apropiadas para algunas partes de la solución
Actividades	Donde exista más de una posible solución, evaluar las fortalezas y debilidades de cada una y elegir la más apropiada.



Paso 5	Evaluar la arquitectura con los stakeholders
Objetivos	Realizar una evaluación de la arquitectura con los stakeholders clave, capturando los problemas y deficiencias y obteniendo la aceptación de la arquitectura por parte los stakeholders
Entradas	Entradas consolidadas. Vista borrador de la arquitectura
Salidas	Comentarios de revisión de la arquitectura
Actividades	Evaluar la arquitectura con un conjunto representativo de la colección de stakeholders. Capturar y acordar cualquier mejora o comentario sobre los modelos.

ACTIVIDAD 7A



Paso 7A	Retrabajar la arquitectura
Objetivos	Direccionar cualquier problema que haya surgido durante la tarea de evaluación
Entradas	Vistas arquitectónicas. Comentarios de revisión. Estilos/patronos arquitectónicos relevantes.
Salidas	Vistas arquitecturales modificadas. Áreas para futura investigación.
Actividades	Tomar los resultados de la evaluación arquitectónica y direccionarlos para producir una arquitectura que cumple mejor sus objetivos.



Paso 7A	Revisar requerimientos
Objetivos	Considerar cualquier cambio a los requerimientos originales del sistema que haya saltado a la luz durante la evaluación de la arquitectura
Entradas	Vistas arquitectónicas. Comentarios de revisión.
Salidas	Requerimientos revisados (si hubiera)
Actividades	El trabajo realizado hasta el momento puede revelar deficiencias o inconsistencias en los requerimientos, o requerimientos que no son factibles o son muy costosos de implementar. En este caso, puede ser necesario revisar estos requerimientos con los stakeholders y obtener su acuerdo a las revisiones necesarias.

CRITERIO DE FINALIZACIÓN - ALTERNATIVAS



En un mundo ideal, se continuará con la definición de la arquitectura hasta que sea perfecta y totalmente documentada.

En la práctica, esto no es posible.

Que no haya comentarios significativos en las actividades de evaluación de la arquitectura, y que no se requieran cambios significativos a la arquitectura

No es muy frecuente que se pueda lograr ese acuerdo



Finalizar la definición de la arquitectura cuando:

- la mayoría de los intereses de los stakeholders más importantes estén resueltos;
- el arquitecto (o grupo de) se sienta seguro que el proyecto puede continuar con un nivel de riesgo aceptable

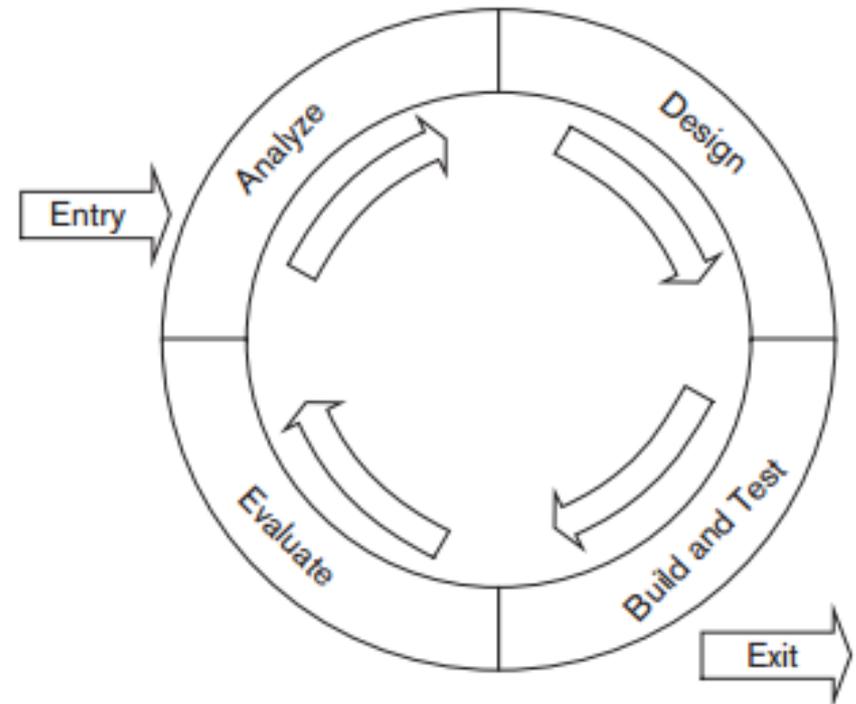
Producir una descripción arquitectónica que sea lo suficientemente buena para satisfacer las necesidades de los usuarios, en lugar de luchar por una versión perfecta que tomará muchos más recursos sin proveer ningún beneficio real a los stakeholders.



- 1 ARQUITECTURA Y LA MANTENIBILIDAD DEL SOFTWARE
- 2 RECOMENDACIONES PARA UNA BUENA ARQUITECTURA
- 3 ROL DEL ARQUITECTO
- 4 ACTIVIDADES DEL ARQUITECTO
- 5 CLASES DE ARQUITECTOS
- 6 PROCESO DE DEFINICIÓN DE ARQUITECTURA
- 7 ARQUITECTURA Y LOS CICLOS DE VIDA**



- La definición de arquitectura forma parte de la fase de Análisis
- El Proceso de Definición de Arquitectura es iterativo de por sí, encajando perfectamente con esta aproximación.





MANIFIESTO AGIL

Individuos e interacciones

Procesos y herramientas

Software funcionando

Documentación

Colaboración con el cliente

Negociación de contratos

Responder al cambio

Seguir un plan

sobre

- ¿Las técnicas y métodos de arquitectura se adaptan a un contexto ágil?
Por supuesto!
- La principal tensión entre los métodos de arquitectura y un ciclo de vida ágil se podría ver en la documentación y la evaluación.



- YAGNI: You ain't gonna need it (No lo vas a necesitar)
 - ✓ Documentar solamente lo que se necesita, no desperdiciar tiempo tratando de anticipar todas las posibles necesidades.
 - ✓ Toda la documentación debe tener un uso y una audiencia en mente.
 - ✓ Escribir para el lector.



- Documentando con Vistas y otros aspectos
 - ✓ Usa vistas arquitectónicas como la “unidad” de documentación
 - ✓ Prescribe producir una vista sí y sólo sí direcciona intereses sustanciales de una comunidad de stakeholders
 - ✓ Como la documentación no es una actividad que detiene todo lo demás hasta que esté completa, el método de selección de vistas sugiere producir la documentación de manera priorizada para satisfacer las necesidades de los stakeholders que la necesitan ahora.



- ¿Qué documentamos?
 - ✓ Lo que necesitamos enseñar a las personas nuevas en el proyecto
 - ✓ Lo que conlleva riesgos significativos si no se maneja apropiadamente
 - ✓ Lo que necesitan los lectores para hacer su trabajo



- Se desprende del Método de Análisis de Tradeoffs de Arquitectura (ATAM)
- No se trata de analizar todo, ni mucho de la arquitectura.
- El foco está determinado por un conjunto de escenarios de atributos de calidad que representen los intereses más importantes de los stakeholders.
- “Más importante” es determinado por la cantidad de valor que el escenario aporta a los stakeholders, o la cantidad de riesgo presente en el escenario.
- Una vez que estos escenarios son elicitados, validados y priorizados, nos darán una agenda de evaluación basada en:
 - ✓ Lo que es más importante para el éxito del sistema
 - ✓ Lo que representa el mayor riesgo para el éxito del sistema



- La arquitectura tiene como objetivo modularizar al sistema y disminuir a futuro los costos de los cambios.
- Existen recomendaciones a tener en cuenta para el proceso de desarrollo de la arquitectura, por ejemplo:
 - ✓ Que se desarrolle por un grupo pequeño de arquitectos para asegurar integridad conceptual y consistencia técnica
 - ✓ Se debe basar en una lista priorizada de atributos de calidad
 - ✓ Se debe documentar utilizando distintas vistas...
- El rol del arquitecto es muy importante – liderar, facilitar la colaboración, ...



- El arquitecto debe desarrollar distintas actividades:
 - ✓ Definición de la arquitectura – gestión requerimientos, definición, selección tecnologías, evaluación, colaboración
 - ✓ Entrega de la arquitectura – propietario; liderazgo; coaching y mentoring; aseguramiento de calidad; diseño, desarrollo y testing
- Distintas clases de arquitectos – de producto, de dominio, de solución, empresarial



- Proceso de desarrollo de la arquitectura
 - ✓ Actividades de soporte:
 - Definir contexto y alcance inicial
 - Comprometer interesados
 - Capturar principales intereses
 - Definir la arquitectura
 - Crear un esqueleto de sistema
 - ✓ Actividades de definición de la arquitectura
 - Considerar entradas
 - Identificar escenarios
 - Identificar estilos relevantes
 - Producir arquitectura candidata
 - Explorar opciones
 - Evaluar la arquitectura
 - Retrabajar la arquitectura
 - Revisar requerimientos

- Ubicación de la definición de la arquitectura en diferentes ciclos de vida

Elsa Estevez
ece@cs.uns.edu.ar