



Universidad Veracruzana

Facultad de Estadística e Informática

Maestría en Ingeniería de Software

Perfil UML para la documentación de arquitecturas orientadas a aspectos para líneas de productos de software

M.I.S. Jorge Octavio Ocharán Hernández

Xalapa, Ver. miércoles 13 de octubre de 2010

Contenido

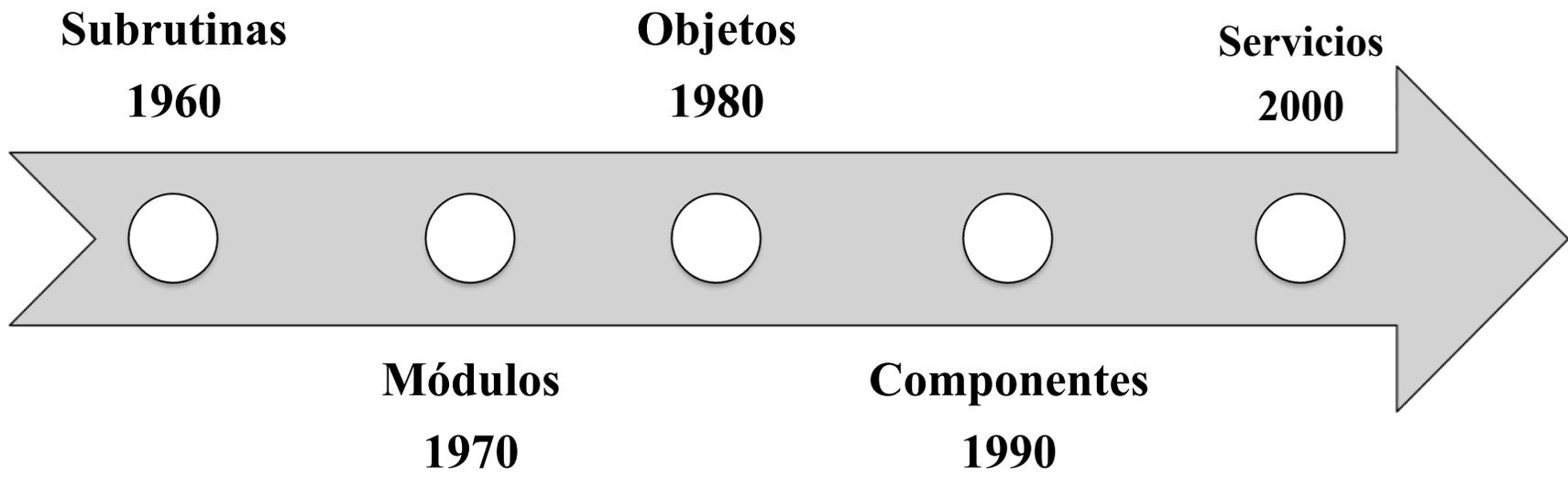
1. Motivación
2. Definición del Problema
3. Aproximación al Problema
4. Resultados
5. Caso de Estudio
6. Conclusiones
7. Trabajo Futuro

MOTIVACIÓN

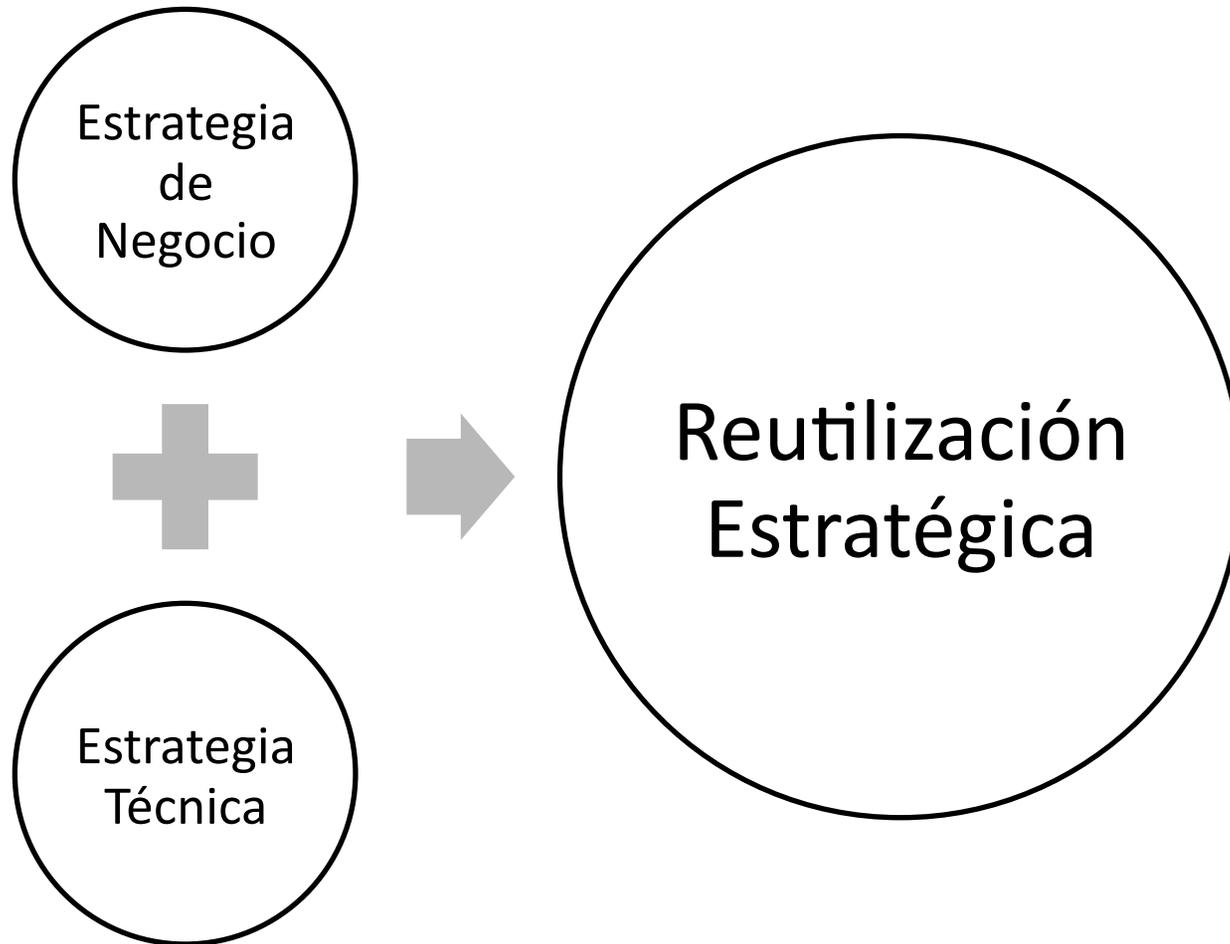
Pocos sistemas son únicos

- Se desarrollan familias de productos similares diferenciados por algunas características.
- Se busca la reutilización de artefactos.
- Las estrategias tradicionales de reutilización de artefactos han tenido **poco beneficio económico.**

Reutilización de artefactos

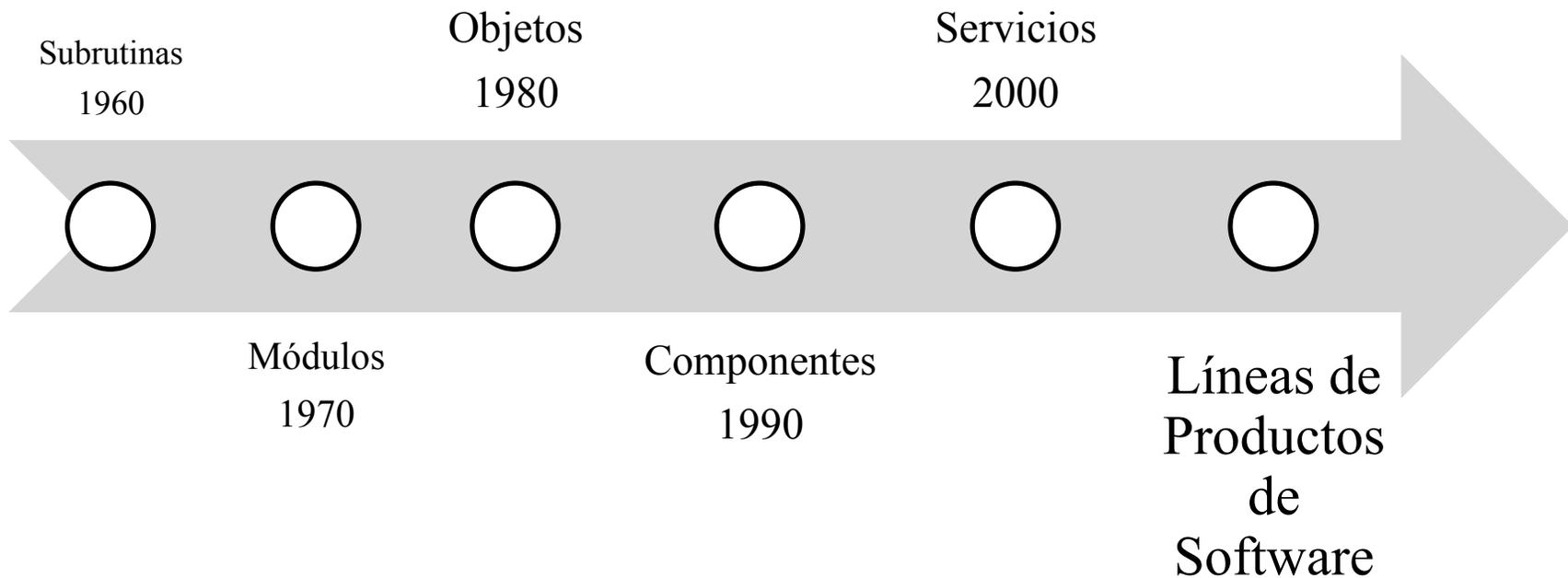


Reutilización de artefactos (cont.)



Reutilización de artefactos (cont.)

De Ad hoc a Sistemática



Líneas de Productos de Software

“Conjunto de software intensivo que comparte un conjunto administrado de características que satisfacen las necesidades específicas de un segmento de mercado o misión y que son desarrolladas a partir de un conjunto de activos esenciales de una forma preestablecida.”

Software Engineering Institute. A Framework for Software Product Line Practice.

Líneas de Productos de Software (cont.)

Las LPS amortizan la inversión de los siguientes artefactos:

- Especificación de Requerimientos
- Modelo de dominio
- Arquitectura de Software
- Documentación
- Plan de pruebas, casos de prueba, datos de prueba

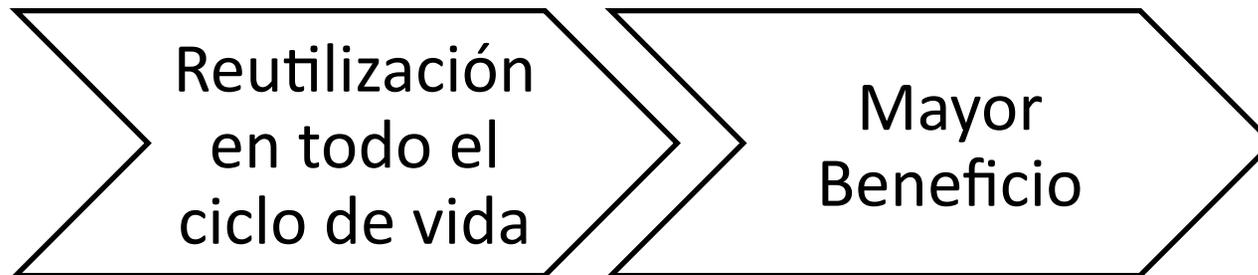
Líneas de Productos de Software (cont.)

Las LPS amortizan la inversión de los siguientes artefactos:

- Personas: su conocimiento y habilidades
- Procesos, métodos y herramientas
- Presupuestos, calendarios y planes de trabajo
- Componentes y servicios

Líneas de Productos de Software

Líneas de productos = Reutilización estratégica



Beneficios organizacionales

- Ganancias en productividad
- Disminución del “*time to market*”
- Mayor calidad de los productos
- Reducción de riesgos
- Incremento en la satisfacción del cliente

Beneficios organizacionales (cont.)

- Uso más eficiente de los recursos humanos
- Personalización en masa
- Mantener presencia en el mercado
- Disminución de costos
- Presencia en nuevos mercados

Clements, P. & Northrop, L. *Software Product Lines: Practices and Patterns*.
Boston, MA: Addison-Wesley, 2002.

Importancia de la arquitectura

Decisiones de
diseño tempranas

- Difíciles de cambiar
- Crítico que sean correctas
- Vehículo de comunicación

Primer artefacto del
diseño

- Desempeño
- Facilidad de modificación
- Fiabilidad
- Seguridad

Clave para la
reutilización
sistemática

- Abstracción transferible y reutilizable

La Arquitectura de una LPS

- Se espera que persista en la LPS
- Cambiará poco y lentamente
- Define el conjunto de componentes de los activos esenciales y sus artefactos de soporte (documentación y pruebas).

Limitaciones de los componentes

- Divide y vencerás – Separación de intereses
- Idealmente los intereses deberían estar localizados en los componentes.
- Hay intereses que se mezclan en diferentes componentes (aspectos).

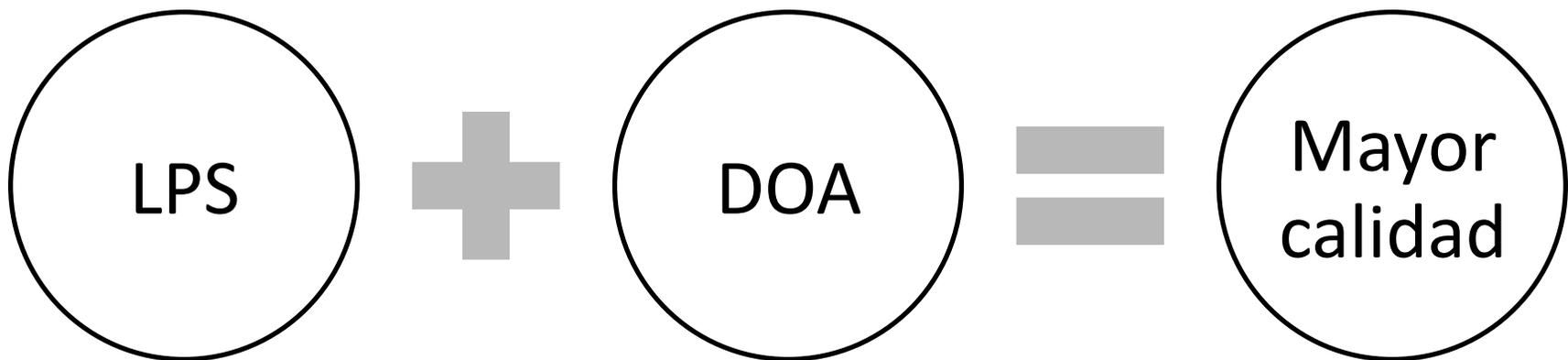
Limitaciones de los componentes (cont.)

- *Tangling*. La realización de un componente satisface varios intereses.
- *Scattering*. La realización de un interés en particular se encuentra dispersa en diferentes componentes.

Desarrollo Orientado a Aspectos

- Un nuevo tipo de modularización
- Mejor separación de intereses mezclados
- Se pueden identificar aspectos desde las primeras etapas

Inclusión del DOA a las LPS



Beneficios esperados

- Mejor separación de intereses
- Mayor facilidad de evolución
- Mayor reutilización
- Mejor comprensión

DEFINICIÓN DEL PROBLEMA

Definición del Problema

- No existe un consenso acerca de la manera de documentar una ALPS ni una AOA. Como consecuencia:

NO existe una forma de describir una AOA de una LPS.

Objetivos

- Desarrollar una estrategia para la documentación de AOALPS, que facilite su:
 - Creación
 - Evolución
 - Derivación

APROXIMACIÓN

Aproximación

- Se estudió las estrategias de documentación para:
 - Arquitecturas de software
 - Arquitecturas de LPS
 - Arquitecturas orientadas a aspectos

Aproximación (cont.)

- Se busco amalgamar las mejores prácticas, a juicio del autor, de las diferentes áreas en una propuesta coherente.

Requisitos

R1. Utilizar las estrategias de documentación de arquitecturas, de documentación de arquitecturas orientadas a aspectos y de documentación de LPS existentes y de mayor aceptación.

R2. El lenguaje de modelado para la documentación debe ser de propósito general y basarse en UML 2.0.

Requisitos (cont.)

- **R3.** Debe ser completa, es decir, debe soportar todas las abstracciones de cada concepto de DSOA y representar la variabilidad de la LPS.
- **R4.** Debe promover la simplicidad y evitar las extensiones innecesarias.

PROPUESTA

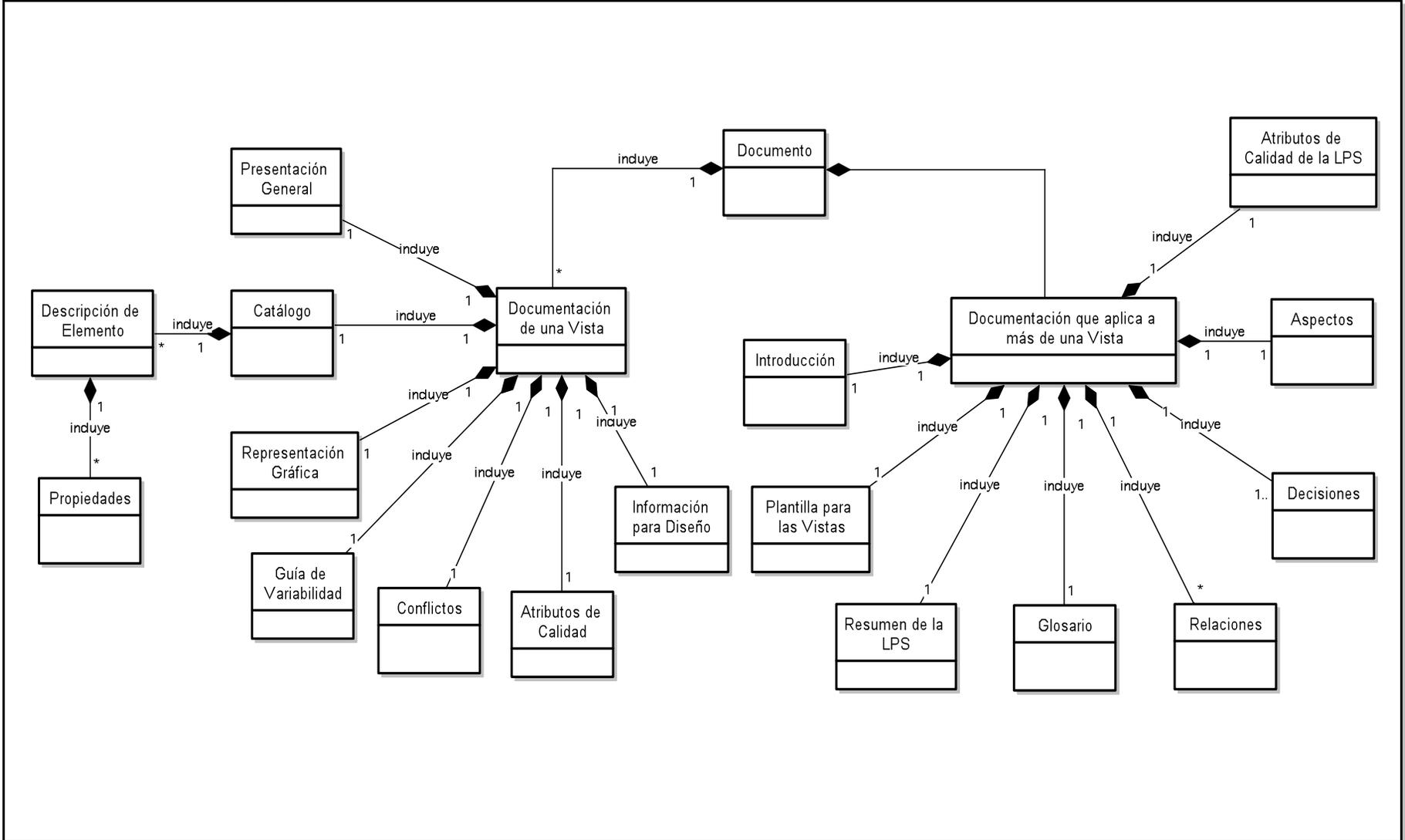
Mejores prácticas

- “*Documenting Software Architectures: Views and beyond*” (Clements *et al.*)
- Modelo Ortogonal de Variabilidad (Pohl *et al.*)
- “*Approach for Modeling Aspects in Architectural Views*”. (Tekinerdogan *et al.*)

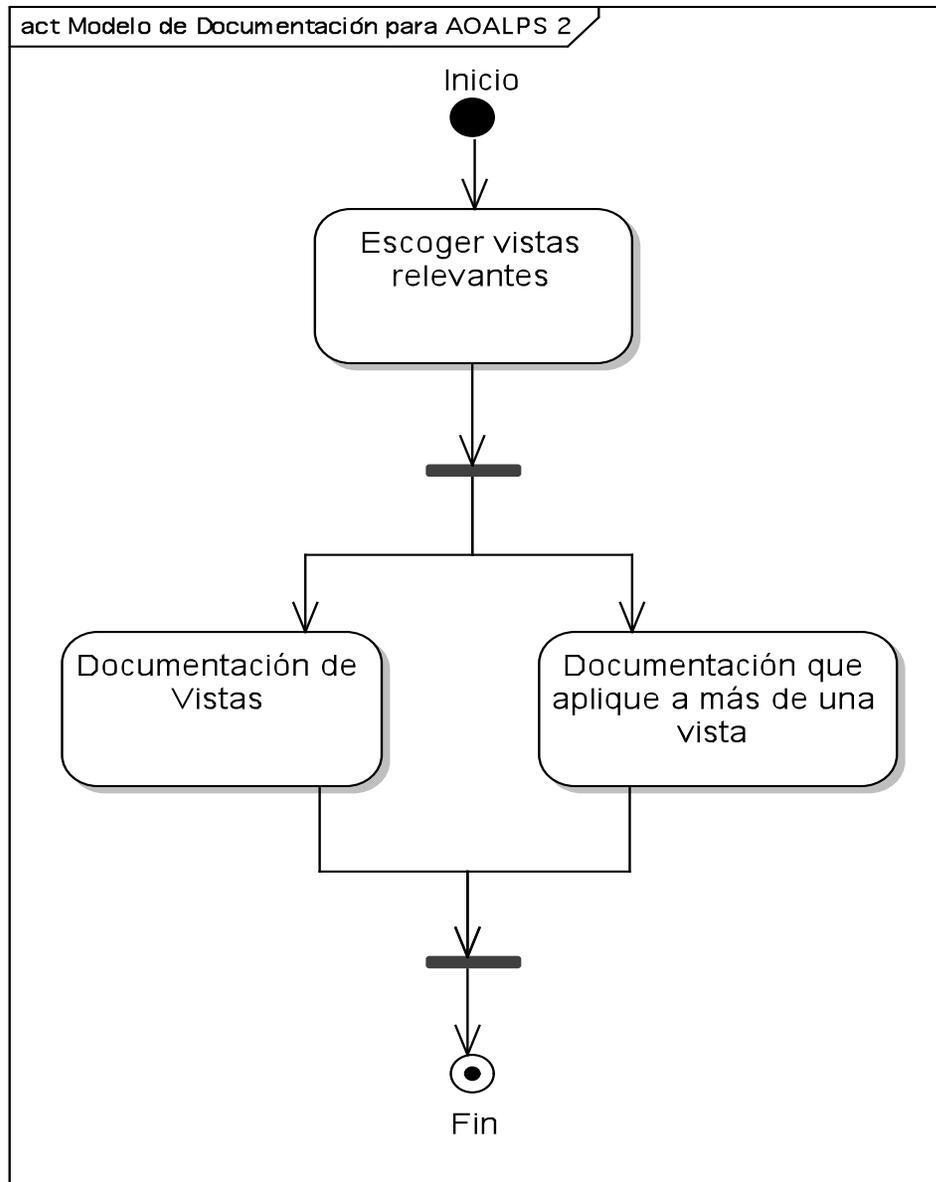
Paquete de documentación

class Domain Objects

Paquete de Documentación de AOALPS



Actividades



Escoger vistas relevantes

1. Producir una lista de vistas candidatas
2. Combinar vistas
3. Priorizar

Plantilla para Documentación de Vistas

Nombre de la vista

Sección 1. Presentación General

Sección 1.1 Antecedentes

Sección 1.2 Presentación

Sección 1.3 Estilo arquitectónico

Sección 2. Catálogo de Elementos

Sección 2.1 Elementos y sus propiedades

Sección 2.2 Relaciones y sus propiedades

Sección 2.3 Interfaces de los elementos

Sección 2.4 Comportamiento de los elementos

Sección 3. Representación Gráfica

Plantilla (cont.)

Sección 4. Guía de Variabilidad

Sección 5. Conflictos

Sección 6. Atributos de calidad

Sección 6.1 Atributos mostrados y su descripción

Sección 6.2 Importancia de los atributos mostrados

Sección 7. Información para diseño

Plantilla para Documentación que aplica a más de una vista

Sección 1. Introducción

Sección 1.1 Propósito

Sección 1.2 A quién va dirigido

Sección 1.3 Información documentada

Sección 1.4 Vistas y sus interesados

Sección 1.5 Escenarios

Sección 1.6 Organización del resto de la documentación

Sección 1.7 Versión de la documentación

Sección 2. Plantilla para las vistas

Plantilla (cont.)

Sección 3. La línea de productos de software

Sección 3.1 Función de la línea de productos de software

Sección 3.2 Usuarios de la LPS

Sección 3.3 Restricciones

Sección 4. Relaciones entre vistas

Sección 5. Decisiones que trastocan varias vistas

Sección 6. Aspectos

Sección 7. Atributos de calidad

Sección 8. Glosario

Representación gráfica

- No se cuenta con ella.
- Algunos problemas con UML.

UML

- Lenguaje gráfico para especificar, documentar y construir los artefactos que modelan el sistema.
- Es de propósito general.
- Se puede extender por medio de perfiles.

Perfiles

- Disponer una terminología y vocabulario propio de un dominio de aplicación.
- Definir una sintaxis para construcciones que no cuentan con una notación propia.
- Definir una nueva notación para símbolos ya existentes, más acorde con el dominio de la aplicación objetivo.

Perfiles

- Añadir cierta semántica que no aparece determinada de forma precisa en el metamodelo.
- Añadir restricciones a la existentes en el metamodelo, restringiendo su forma de utilización.
- Añadir información que puede ser útil a la hora de transformar el modelo a otros modelos, o a código *

Representación gráfica

- Basado en el metamodelo de UML 2.0
- El perfil incluye:
 - Elementos arquitectónicos
 - Aspectos arquitectónicos
 - Elementos arquitectónicos de una LPS
- Se soportan los estilos arquitectónicos más comunes

Vistas de módulos

- ArchitecturalModule
- ArchitecturalLayer
- ArchitecturalVariant
- ArchAspectualModule
- ArchitecturalRelation
 - uses
 - generalizes
 - decompose in
 - allowed to use

Vistas de C&C

- ArchitecturalComponent
- ComplexArchitecturalComponent
- ArchitecturalComponentVariant
- ArchAspectualComponent
- ArchAspectualComponentVariant
- ArchitecturalConnector
- ArchitecturalConnectorVariant

Algunos estilos arquitectónicos

- Filter
- SharedDataRepository
- Client
- Server
- Peer
- ConcurrentUnit
- Pipe
- DataReading
- DataWriting
- Invocation
- Communication

CASO DE ESTUDIO

Áncora-Soft

- LPS que da soporte a la metodología de análisis de requerimientos Áncora (*Sumano López 2006*).
- Productos enfocados a estudiantes, docentes y desarrolladores en general

Áncora-Soft (cont.)

- Arquitectura de la LPS desarrollada con AOPLA (*Cortés Verdín 2009*).
- Se genera una arquitectura de referencia.
- No se considera información para el diseño.

Vistas a documentar

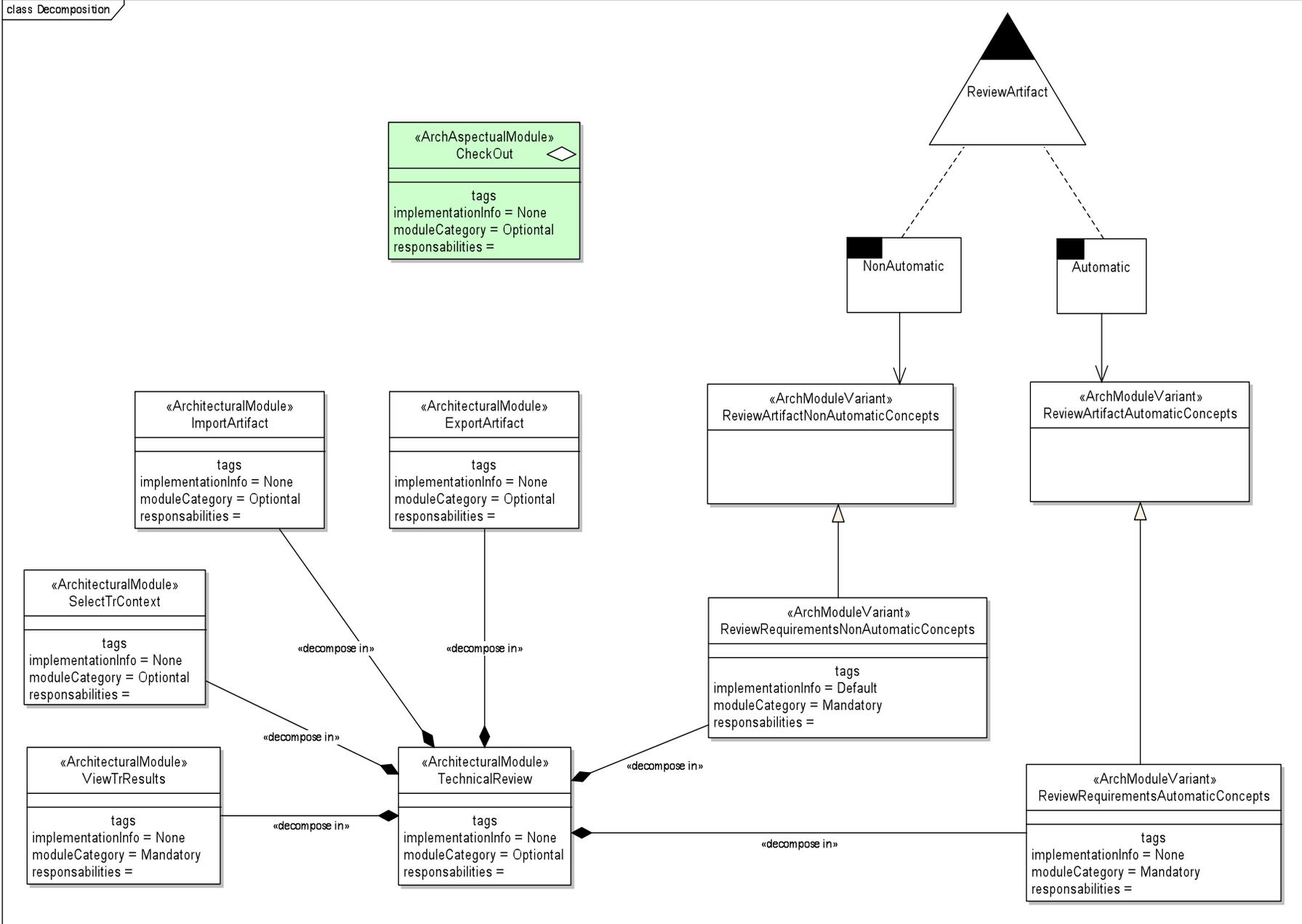
Interesados	Vista Módulos				Vista C&C	Vista Asignación	
	Descomposición	Usos	Generalización	Capas	Componentes y Conectores	Despliegue	Implementación
Administración	G	G	N	G	G	D	N
Desarrolladores	D	D	G	D	D	G	D
Probadores e integradores	N	D	N	D	D	G	N
Arquitecto de software	D	D	D	D	D	D	D
Mercadotecnia	G	N	N	N	N	G	N
Usuarios Finales	N	N	N	N	N	G	N

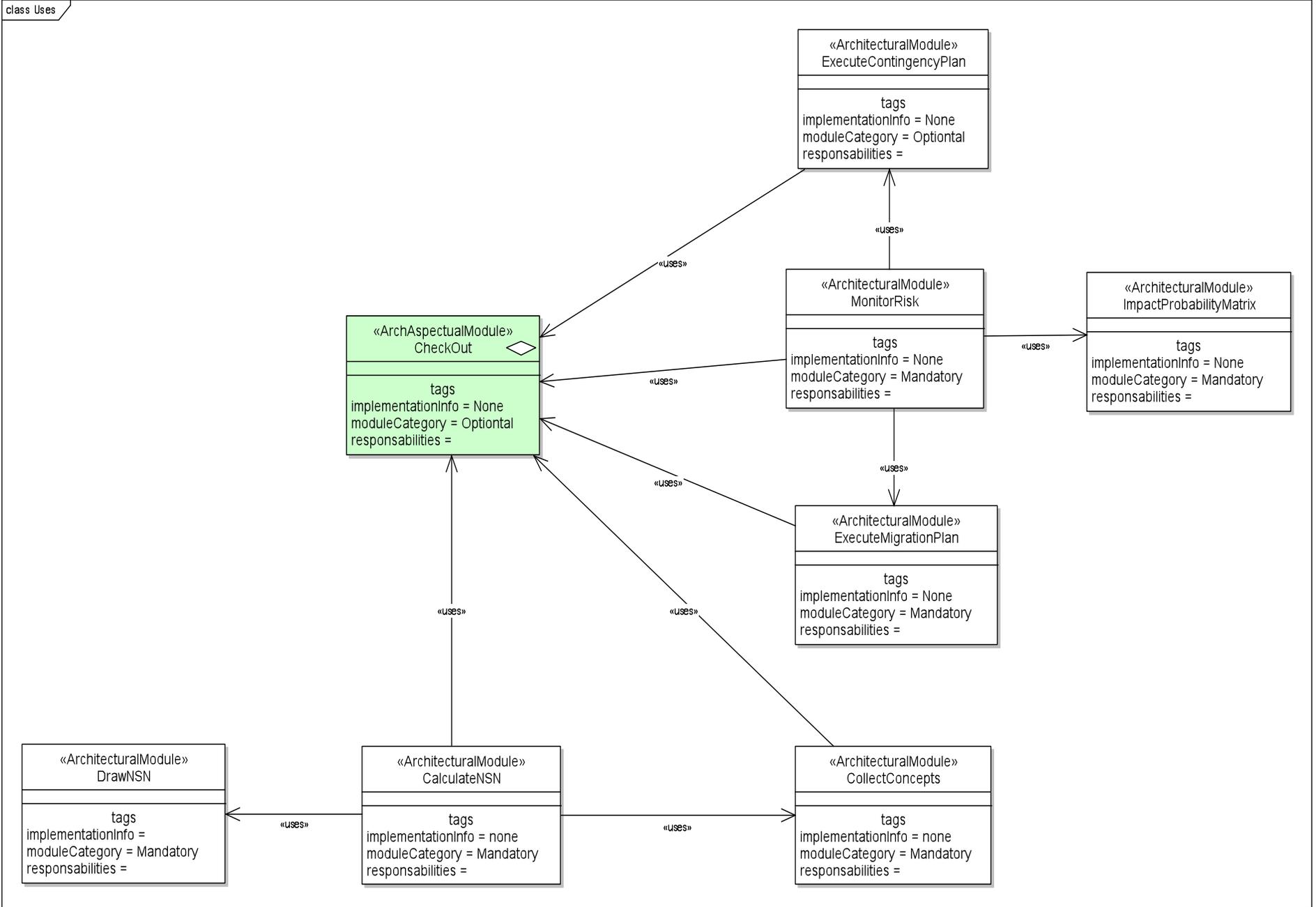
Clave: N – Ninguna información, G – Información General y D – Información detallada

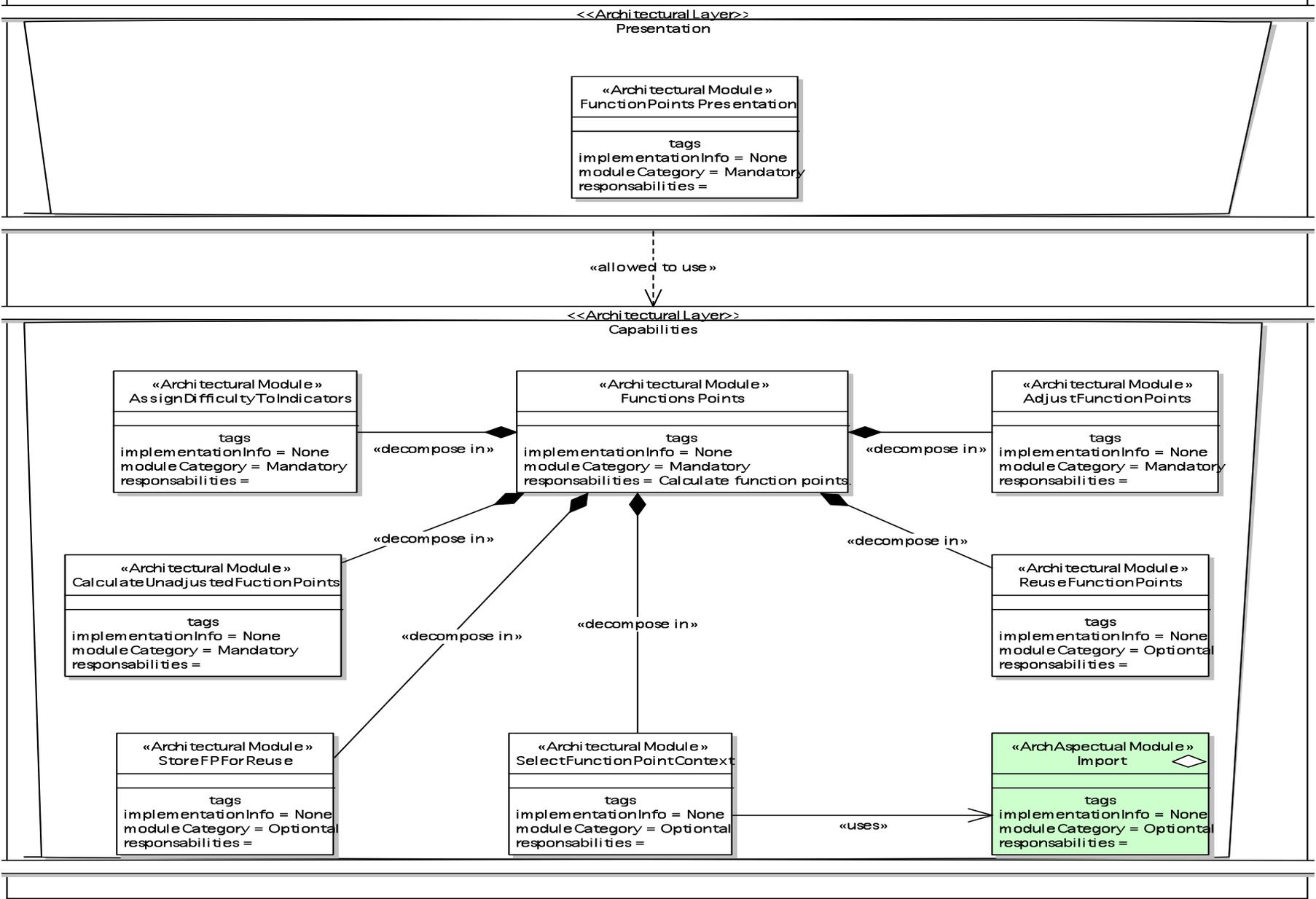
Vistas a documentar

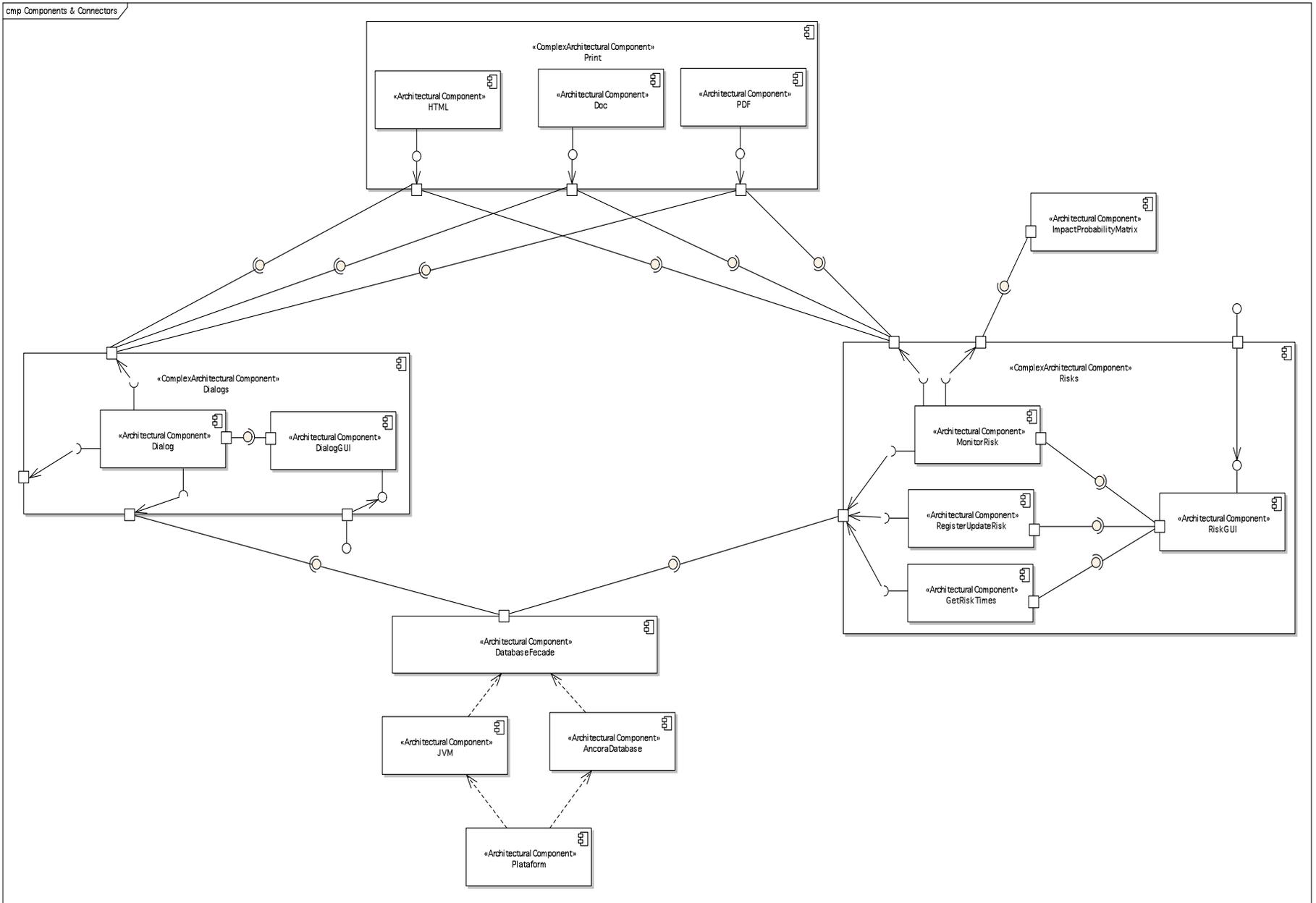
Interesados	Vista Módulos				Vista C&C	Vista Asignación	
	Descomposición	Usos	Generalización	Capas	Componentes y Conectores	Despliegue	Implementación
Administración	G	G	N	G	G	D	N
Desarrolladores	D	D	G	D	D	G	D
Probadores e integradores	N	D	N	D	D	G	N
Arquitecto de software	D	D	D	D	D	D	D
Mercadotecnia	G	N	N	N	N	G	N
Usuarios Finales	N	N	N	N	N	G	N

Clave: N – Ninguna información, G – Información General y D – Información detallada









Lecciones aprendidas

- En el caso de la vista de Descomposición se muestran más claramente los puntos de variación y las variantes por lo que debería ser uno de los artefactos principales para la derivación de los productos de la LPS.

Lecciones aprendidas (cont.)

- Los módulos arquitectónicos aspectuales se muestran totalmente identificados así como sus relaciones. Esto resulta de gran utilidad al momento de realizar una evaluación de una arquitectura orientada a aspectos.

Lecciones aprendidas (cont.)

- En la vista de Usos se muestra claramente que módulos arquitectónicos interactúan con los módulos arquitectónicos aspectuales. Este podría ser un indicador de que porcentaje del código de la aplicación final estaría entrelazado o disperso en caso de no utilizar la DOA.

Lecciones aprendidas (cont.)

- Aunque no es parte de un estándar, la utilización de colores en los diferentes diagramas para distinguir diferentes elementos hace a los diagramas mucho más legibles y se pueden ubicar los elementos más fácilmente.

Lecciones aprendidas (cont.)

- La manera en que se documenten los atributos de calidad tendrá una gran variación ya que se espera que la metodología con la que se diseñe la AOALPS contenga un modelo de calidad.

CONCLUSIONES

Conclusiones

- La documentación de la AOALPS es vital para el éxito de la LPS.
- Se identificaron los requisitos que debe cumplir una Documentación de AOALPS.
- Se propusieron dos artefactos:
 - Una estrategia para la documentación
 - Un perfil UML

Ventajas

- Simple: Fácil de comprender y realizar.
Facilita la actualización
- Curva de aprendizaje poco pronunciada.
- Conformidad con el estándar IEEE 1471

Desventajas

- Se depende de un modelo de calidad externo para documentar las decisiones arquitectónicas
- Se necesita que la propuesta sea utilizada en más proyectos

TRABAJO FUTURO

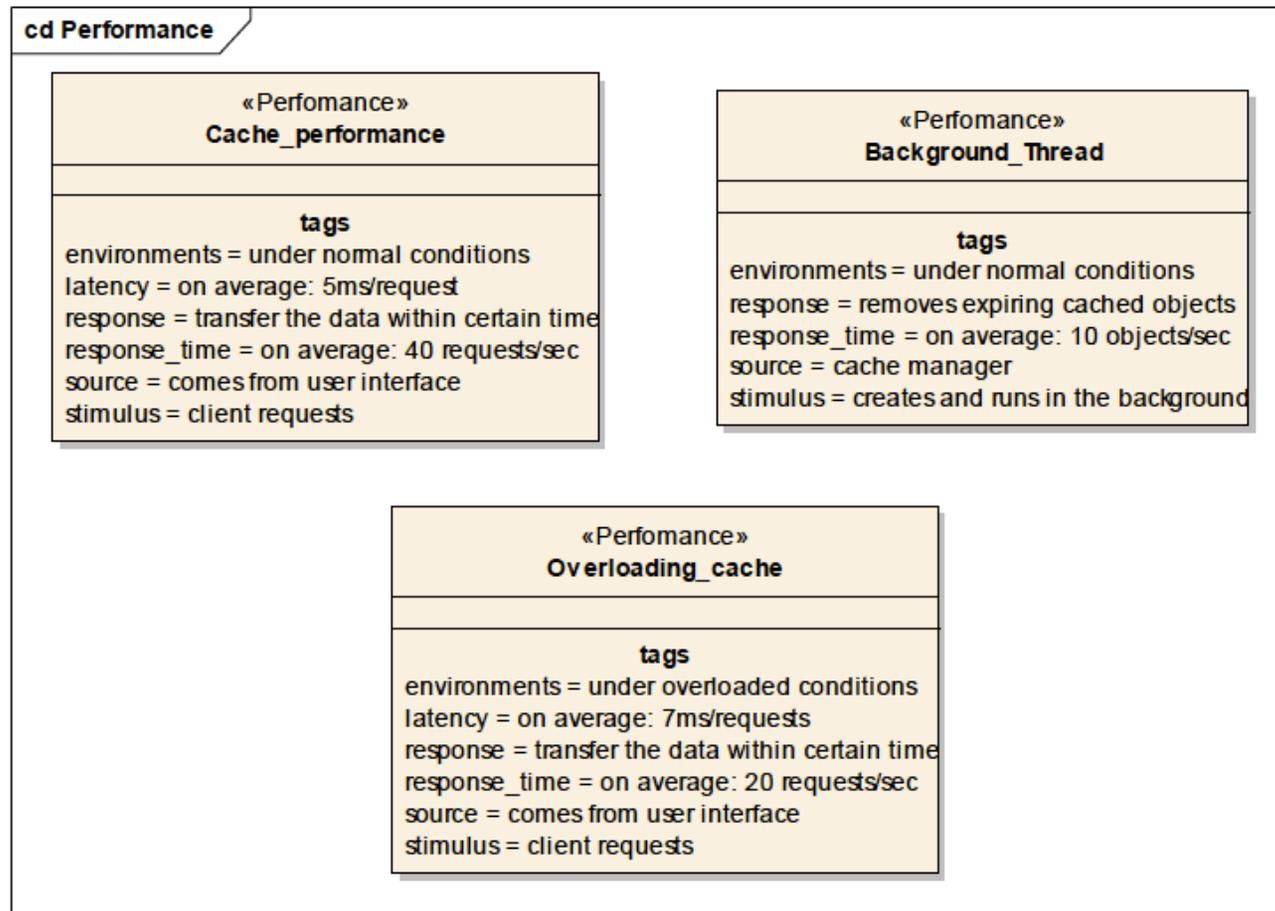
Trabajo futuro

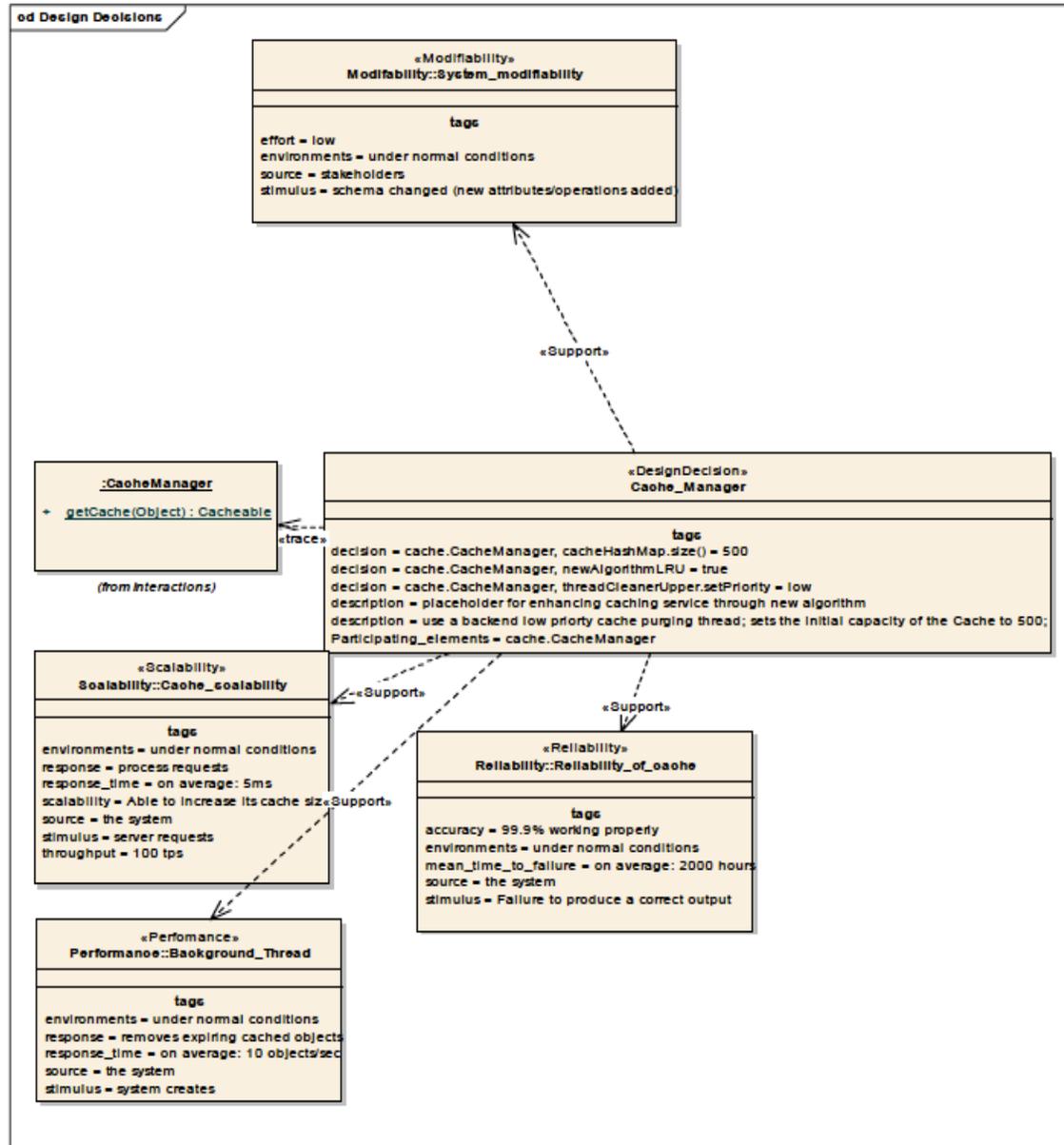
- Exploración del tema desde un punto de vista formal – validación y verificación rigurosa.
- Inclusión de vistas diferentes a las propuestas debido a condiciones especiales.
- Una mejor representación gráfica de los elementos arquitectónicos.

Trabajo futuro

- Comprobar con más casos de prueba y trabajo práctico.
- Probar en que medida la presente propuesta facilita la evaluación de una AOALPS.
- Explorar la trazabilidad de la AOALPS a largo de todas las etapas de desarrollo.
- Documentación gráfica de decisiones de diseño y requisitos no funcionales. (Zhu y Gorton)

Decisiones de diseño y RNF





SECCIÓN DE PREGUNTAS Y RESPUESTAS

Gracias por su atención.

jocharan@uv.mx