

## COMPARACIÓN DE MÉTODOS PARA LA ARQUITECTURA DEL SOFTWARE: UN MARCO DE REFERENCIA PARA UN MÉTODO ARQUITECTÓNICO UNIFICADO

FRANCISCA LOSAVIO<sup>1</sup>, CHRISTIAN GUILLÉN-DRIJA<sup>2</sup>

<sup>1</sup>Universidad Central de Venezuela. Facultad de Ciencias. Escuela de Computación. Centro ISYS.

Apdo. 47567. Los Chaguaramos. 1041-A. Caracas. e-mail: flosav@cantv.net

<sup>2</sup>Universidad Pedagógica Experimental Libertador, Instituto Pedagógico de Miranda “J.M. Siso Martínez”,

La Urbina. Estado Miranda. Venezuela. e-mail: cguillen@ipmjmsm.upel.edu.ve

Recibido: abril de 2009

Recibido en forma final revisado: diciembre de 2009

### RESUMEN

Debido a la relevancia que ha adquirido la visión arquitectónica del software en el proceso de desarrollo, se han propuesto diversos métodos, tanto de diseño como de evaluación arquitectónica. Cada uno de ellos se fundamenta en conceptos que pueden ser equivalentes, complementarios o alternativos. Un estudio comparativo de tales métodos favorece la identificación de los procedimientos y actividades que mejor respondan al complejo proceso de generar una arquitectura en función de un conjunto de requisitos iniciales. Este trabajo presenta un marco de comparación que luego es aplicado tanto a métodos de diseño arquitectónico como a métodos de evaluación arquitectónica, identificándose un conjunto de características que consideramos deseables en un método de diseño arquitectónico. Con base en tales características, presentamos una primera versión de un método unificado que contempla el proceso completo de diseño arquitectónico.

*Palabras clave:* Arquitectura de software, Calidad de software, Métodos de diseño arquitectónico, Métodos de evaluación arquitectónica, Características de calidad.

### COMPARISON OF SOFTWARE ARCHITECTURE METHODS: A FRAMEWORK FOR A UNIFIED ARCHITECTURE METHOD

#### ABSTRACT

Due to the growing interest in the current development of the architectural vision of software, a great number of architectural design and evaluation methods have been proposed. They are generally based on equivalent, complementary or alternative concepts. A comparative study of such methods allows us to determine procedures and activities that satisfy the process of generating architecture in terms of the initial set of requirements. In this paper, we present a comparative framework which is applied to architectural design methods as well as architectural evaluation methods. We obtained a set of desirable characteristics in an architectural design method. Based on those characteristics, a first draft of a framework for a unified method that contemplates the whole design process is presented.

*Keywords:* Software architecture, Software quality, Architectural design methods, Architecture evaluation methods, Quality characteristics.

#### INTRODUCCIÓN

Los métodos de diseño arquitectónico y los métodos de evaluación arquitectónica consideran la concepción de un sistema de software a un alto nivel de abstracción, con base en una visión arquitectónica. Algunos de estos métodos tienen como objetivo la generación de una arquitectura que responda a un conjunto de requisitos iniciales. Otros están dirigidos a evaluar configuraciones arquitectónicas ya existentes para escoger la mejor de acuerdo a un conjunto de requisitos. Todos se ubican en las fases tempranas del

proceso de diseño de software y en la actualidad son considerados parte esencial del mismo. En el transcurso de este trabajo nos referiremos a ambas categorías como *métodos arquitectónicos*.

El objetivo de este trabajo ha sido el de comparar un conjunto de métodos arquitectónicos para identificar semejanzas y diferencias, así como explorar las heurísticas subyacentes en cada propuesta. A partir de este estudio comparativo se obtuvo un conjunto de características que proponemos como deseables en un método de diseño arquitectónico

basado en características de calidad. Más que realizar una evaluación de los métodos involucrados, lo que implica un proceso de valoración en función de un conjunto de criterios previamente definido, nuestra intención es realizar una comparación y clasificación de las propuestas existentes, que nos permita identificar las características más notorias en cada caso. Para la selección de los criterios de comparación, nos basamos en el estudio de varias propuestas realizadas en este sentido. Aunado a lo anterior, complementamos el cuerpo de criterios de comparación agregando otros que consideramos importantes en función de las problemáticas antes mencionadas. Finalmente, con base en el conjunto de características consideradas como deseables, se propone una primera versión de un marco conceptual, de referencia o framework que contemple un método de diseño arquitectónico, constituyéndose éste en una estructura unificadora que especifica los diferentes elementos del proceso completo de diseño arquitectónico.

Este artículo contiene, además de la introducción y las conclusiones, 4 secciones más: la sección 2, describe brevemente algunos trabajos en los que se comparan métodos centrados en la arquitectura de software; la sección 3, presenta un marco de comparación; la sección 4, el conjunto de características que hemos considerado como deseables en un método de diseño arquitectónico; y por último, la sección 5, propone la primera versión de un marco conceptual para un método de diseño arquitectónico unificado.

## ANTECEDENTES

Varios son los esfuerzos realizados con el fin de comparar métodos, tanto de diseño como de evaluación arquitec-

tónica. Por una parte, se pueden nombrar los trabajos de: Obbink *et al.* (2007); Tekinerdogan & Mehmet (2000), Losavio *et al.* (2001), los cuales comparan métodos de diseño arquitectónico. Por otra parte, podemos mencionar los trabajos de: Hammer *et al.* (2002); Kazman & Nord (2003); Dobrica & Niemelä (2002); Babar *et al.* (2004); Babar & Gorton (2004); y Grimán *et al.* (2006), en los que se comparan los métodos de evaluación arquitectónica. Por último, Thiel (2005) ejecuta un ejercicio de comparación tanto de métodos de diseño como de métodos de evaluación.

En cuanto a la comparación de métodos de diseño arquitectónico, Losavio *et al.* (2001) aplican DESMET (Kitchenham, 1996), para determinar la técnica de evaluación más idónea aplicable a un conjunto de métodos de diseño. Al usar entonces el análisis a través del filtrado de características identifican los aspectos generales que deberían estar contenidos en un método de diseño arquitectónico basado en atributos de calidad. Como resultado, obtienen los criterios contenidos en la tabla 1. Obbink *et al.* (2007), proponen la necesidad de identificar las similitudes entre los métodos de diseño con el objetivo de obtener un modelo general. Los investigadores llegan a la conclusión de que un método de diseño debe contemplar actividades de análisis de requisitos y de evaluación. Tales actividades son entonces utilizadas como criterios para comparar varios métodos. Por su parte, Tekinerdogan & Mehmet (2000) proponen una clasificación de los distintos enfoques de diseño arquitectónico según las fuentes utilizadas para la identificación de las abstracciones arquitectónicas claves. En dicha clasificación se identifican tres enfoques fundamentales: los centrados en artefactos, los centrados en casos de uso y los centrados en el dominio. Con respecto a los métodos de evaluación ar-

**Tabla 1.** Criterios de comparación según Losavio *et al.* (2001).

<b>Criterio</b>	<b>Descripción</b>
Identificación de un enfoque arquitectónico amplio (estilos).	En otras palabras debe basarse en la utilización de estilos arquitectónicos para la generación de una arquitectura candidata inicial. Estos son los determinantes principales de los atributos de calidad arquitectónicos, por lo que deben ser identificados en las fases iniciales del proceso de desarrollo.
Especificación de la arquitectura.	Lo que implica que la arquitectura debe ser expresada en términos de una notación precisa de acuerdo a diferentes vistas arquitectónicas.
Levantamiento de los requisitos de calidad para el establecimiento de los objetivos de calidad del sistema.	El método debe ofrecer una técnica para capturar los requisitos no funcionales, los cuales se constituyen en objetivos de calidad a alcanzar por la arquitectura.
Evaluación de la arquitectura con respecto a los objetivos de calidad establecidos.	El método debe incluir actividades dirigidas a predecir los atributos de calidad con base en la arquitectura. Para poder ejecutar una evaluación arquitectónica, es necesario contar tanto con un conjunto de declaraciones de los requisitos de calidad y con una especificación de la arquitectura en la que se articulen claramente las decisiones de diseño en términos de eventos que afectan los objetivos de calidad establecidos (estímulos) y la reacción de la arquitectura frente a tales eventos (respuestas).
Diseño de un modelo de calidad.	El método debe contener actividades dirigidas a la estructuración de un cuerpo de requisitos de calidad, caracterizando cada atributo de calidad.

quitectónica, Dobrica & Niemelä (2002), proponen un conjunto de criterios para su comparación y caracterización, pero sin ofrecer una definición de los mismos. Sin embargo,

asocian interrogantes a cada uno de los criterios que conforman el marco de comparación, por lo que se puede al menos inferir el significado de cada uno de estos (tabla 2).

**Tabla 2.** Elementos del marco de comparación de Dobrica & Niemelä (2002)

<b>Elemento del marco de comparación</b>	<b>Interrogante</b>	<b>Descripción</b>
Objetivo específico del método	¿Cuál es el objetivo específico del método?	Todos los métodos de evaluación tienen un objetivo general en común, sin embargo pueden diferir en cuanto a los objetivos específicos que persiguen.
Técnicas de evaluación	¿Cuáles son las técnicas de evaluación incluidas en el método?	Según los autores, existen dos grupos de técnicas de evaluación: técnicas de interrogación y técnicas de medición. Las técnicas de interrogación se caracterizan por generar preguntas cualitativas que están dirigidas a un arquitecto. Las técnicas de medición: sugieren la ejecución de mediciones cuantitativas, y suelen aplicarse para obtener información de atributos de calidad específicos, por lo tanto no son ampliamente utilizadas como las técnicas de interrogación.
Atributos de calidad	¿Cuáles son los atributos de calidad considerados?	Dos tendencias son identificadas: los métodos que consideran sólo un atributo de calidad y los que consideran varios atributos.
Descripción de la arquitectura de software	¿Cuáles son las vistas en las que se enfoca el método?	Las vistas son los artefactos sobre los cuales se realizará la evaluación. Existen varios tipos de vista, que muestran distintas facetas de la arquitectura.
Stakeholders involucrados	¿Cuáles son los stakeholder que participan en el diseño del sistema?	Los stakeholders son los distintos interesados en el sistema.
Actividades del método	¿En qué orden y de qué forma se usan las técnicas de evaluación en función de los objetivos específicos del método? ¿Cuál es el resultado de cada una de ellas?	Implica considerar la cantidad de actividades, entradas y salidas de cada una de ellas y sus implicaciones en el resultado final del método.
Reutilización de una base de conocimiento	¿Se considera la reutilización de una base de conocimiento?	La base de conocimiento continuamente debe irse construyendo con los resultados obtenidos de las distintas evaluaciones realizadas. La base de conocimiento permite disminuir los esfuerzos de evaluación, optimizando la aplicación de las actividades.
Validación del método	¿El método ha sido validado a través de su aplicación en casos reales en la industria?	Este criterio permite determinar la madurez del método.

Babar *et al.* (2004), reconocen que el trabajo de Dobrica & Niemelä (2002) es el primer esfuerzo estructurado para proponer una taxonomía en esta línea de investigación, sin embargo, señalan que estos no explican claramente los componentes de su marco de comparación, ni justifican explícitamente las razones por las cuales los incluyen, infiriendo que los mismos pueden ser considerados como características deseables en un método. Posteriormente Babar & Gorton (2004), reorganizan los criterios dentro de cuatro categorías: contexto, participantes o stakeholders, contenidos y confiabilidad. En la tabla 3, se muestran las dos versiones del marco de comparación. Se puede observar que las dos versiones son muy similares, excepto que en la segunda se omite el criterio repositorio de experiencia y se agregan los criterios: entradas y salidas, dominio de aplicación y beneficios. Hammer *et al.* (2002) proponen un

marco de comparación, pero no presentan argumentos que sustenten la inclusión de los criterios que lo componen, ni tampoco las definiciones de los mismos. No obstante, se puede inferir el significado de cada uno de estos criterios (tabla 4). Grimán *et al.* (2006) aplican un análisis de características a tres métodos de evaluación arquitectónica a través de un caso de estudio, obteniendo como resultado un conjunto de 49 métricas agrupadas en características y sub-características (tabla 5).

Por último, Thiel (2005) realiza una comparación tanto de métodos de evaluación como de métodos de diseño (tabla 6). En dicha comparación se evidencia la existencia de criterios que son aplicados por igual, tanto a los métodos de evaluación como a los métodos de diseño.

**Tabla 3.** Comparación de los criterios de Babar *et al.* (2004) y Babar & Gorton (2004).

Babar <i>et al.</i> (2004)	Babar & Gordon (2004)	
Componentes	Componentes	Elementos
Definición de arquitectura de software	Contexto	Definición de arquitectura de software
Objetivos del método		Objetivos del método
Número de atributos de calidad		Atributos de calidad
Estadio del proyecto en el cual es aplicable		Estadio del proyecto en el cual es aplicable
		Entradas y salidas
		Dominio de aplicación
	Stakeholders	Beneficios
Stakeholders involucrados		Stakeholders involucrados
Soporte al proceso		Soporte al proceso
Soporte a aspectos no técnicos		Aspectos socio-técnicos
Recursos requeridos		Recursos requeridos
Actividades del método	Contenidos	Actividades del método
Descripción arquitectónica		Descripción arquitectónica
Enfoques de evaluación		Enfoques de evaluación
Herramientas de soporte		Herramientas de soporte
Estado de madurez	Confiabilidad	Estado de madurez
Validación del método		Validación del método
Repositorio de experiencia		

**Tabla 4.** Descripción inferida por los autores de los criterios utilizados por Hammer *et al.* (2002).

<b>Criterios</b>	<b>Descripción</b>
Contexto	Expresa el ámbito del método y su ubicación en el proceso de desarrollo.
Propósito	En este caso, el propósito incluye a los objetivos específicos del método y los atributos de calidad analizados.
Factores claves en la ejecución del proceso	Se refiere a los factores que condicionan el buen desarrollo o aplicación del método.
Prerrequisitos y entradas	Los prerrequisitos son las condiciones previas que deben ser cubiertas para el desarrollo del método; mientras que las entradas vienen dadas por los artefactos que contienen la información inicial necesaria para la evaluación.
Pasos	Constituyan las actividades del método.
Roles	Se refiere en este caso a los stakeholders y no a los papeles que pueden desempeñarse durante la ejecución del método.
Estimación del esfuerzo	Se refiere a la existencia de actividades y herramientas que específicamente se centren en la estimación del esfuerzo realizado por el equipo de evaluación durante la aplicación del método.
Herramientas de soporte	Se refiere a herramientas informáticas que apoyen la generación de artefactos que reflejen los resultados de la evaluación.
Alternativas	Indica los métodos que pueden ser utilizados en su lugar.
Resultados	Incluye el conjunto de artefactos finales que son incluidos en el reporte final de la presentación de resultados.
Fortalezas	Indican de alguna manera los beneficios obtenidos por la organización y por equipo de evaluación, derivadas de la aplicación del método.

**Tabla 5.** Características y sub-características propuestas por Grimán *et al.* (2006).

<b>Característica</b>	<b>Definición de sub-características y atributos</b>
General	Estructura: conformación del método (pasos).
	Enfoque: que puede ser inductivo o deductivo.
	Equipo evaluador: grupo de especialistas responsables de la aplicación del Método.
	Documentación: todos los entregables obtenidos durante la aplicación del método y que documentan las decisiones o conclusiones obtenidas.
	Herramientas: destinadas a favorecer el aspecto objetivo del método.
	Costos: en cuanto a recursos, tiempo y tecnología necesaria.
Específica	Arquitectura del software: como producto final del proceso de diseño arquitectónico.
	Calidad: se refiere al nivel de logro de requisitos no funcionales.
	Vistas de calidad: que pueden ser internas o externas según ISO 9126.
	Requisitos de calidad: dados por las necesidades que el usuario quiere satisfacer a través de los servicios que usa.
	Medidas de calidad: referidas a las métricas de calidad que permiten cuantificar la satisfacción de las características de calidad.

**Tabla 6.** Criterios propuestos por Thiel (2005).

	Criterios	Descripción
Métodos de Evaluación	Ámbito del método	Se refiere al número de atributos considerados.
	Objetivo del método	Generalmente viene dado por la identificación de fortalezas y debilidades.
	Atributos de calidad	Atributos que en particular son considerados en el método para la determinación de la idoneidad del diseño.
	Extensión de la evaluación	Indica si el método está orientado al análisis de una arquitectura, o si por el contrario soporta la comparación de varias soluciones candidatas.
	Técnicas de evaluación	Indica el tipo de técnica que se utiliza, cuestionarios, escenarios, etc.
Aplicables en ambos casos	Dominio de aplicación	Indica si se aplica a un dominio específico y si es de aplicación general.
	Validación del método	Indica la madurez del método, lo cual viene reflejado en la cantidad de casos de estudio.
Métodos de Diseño	Enfoque de diseño	Indica si el diseño está dirigido al logro de funcionalidades o al logro de atributos de calidad.
	Objetivos del método	Indica el entregable principal que se obtiene como resultado de la aplicación del método.
	Requerimientos funcionales y de calidad	Hace referencia al tipo de requerimientos que se analiza para la adopción de soluciones arquitectónicas.
	Vistas arquitectónicas consideradas	Una vista arquitectónica es introducida como una abstracción de la arquitectura desde una perspectiva en particular.
	Conceptos que soportan el diseño	Tales como casos de uso, escenarios, prototipos, patrones de diseño, heurísticas de diseño, etc. Se refiere a las técnicas de análisis

## MARCO DE COMPARACIÓN

En este trabajo se han seleccionado los siguientes métodos: SAAM (Scenario-Based Analysis of Software Architecture) (Bass *et al.* 2003); ALPSM (Architecture Level Prediction of Software Maintenance) (Bengtsson & Bosch, 1999); AQA (Architecture Quality Assessment) (Hilliard II *et al.* 1997); SAE (Software Architecture Evaluation) (AT&T, 1993); FAAM (Family-Architecture Analysis Method) (Dolan, 2001); ASAAM (Scenario-Based Analysis of Software Architecture) (Tekinerdogan, 2003); ALMA (Architecture Level Modifiability Analysis) (Bengtsson *et al.* 2004); QAW (Quality Attribute Workshops) (Barbacci *et al.* 2003); ATAM (Architecture Tradeoff Analysis Method) (Clements *et al.* 2002); PASA (Performance Assessment of Software Architectures) (Connie & Williams, 2002); ARID (Active Reviews for Intermediate Designs) (Clements, 2000); CBSP (Grünbacher *et al.* 2003); PRESKRIPTOR (Brandozzi & Perry, 2002); VAP (Visual Architecture Process) (Bredemeyer & Malan, 2005); CBAM-WIN WIN (Cost Benefit Analysis Method, combinado con el método de negociación de requisitos WIN WIN) (In *et al.* 2001); ABD (Architecture Based Design) (Bass *et al.* 2000); SACAM (Software Architecture Comparison Analysis Method) (Bachmann

*et al.* 2003); SARM (Software Architecture Reengineering Method) (Bengtsson & Bosch, 1998); Bosch (2000); Proteus (Chung *et al.* 2002); Lamsweerde (2003); MECABIC (Método de Evaluación de Arquitecturas de Software Basadas en Componentes) (González *et al.* 2005); Losavio-Chirinos-Lévy-Ramdane Cherif (2003); CBAM (Cost Benefit Analysis Method) (Asundi & Kazman, 2001); QUADRAD (Quality-Driven Architecture Development) (Thiel, 2005); ADD (Attribute-Driven Design) (Bass *et al.* 2003); ASAA (Applied Software Architecture Approach) (Hofmeister *et al.* 2000).

Una revisión de los distintos enfoques mencionados en la sección anterior ha permitido proponer un marco de comparación propio, el cual se presenta a continuación junto con los resultados obtenidos al ser aplicado al conjunto de métodos seleccionados. Esta propuesta intenta comparar los métodos centrados en la visión arquitectónica del software desde dos perspectivas: la primera toma en cuenta las disciplinas propias del proceso de diseño arquitectónico; y la segunda considera un conjunto de elementos comunes al proceso general del desarrollo del software. En la figura 1 se muestra un esquema de los criterios de comparación adoptados.

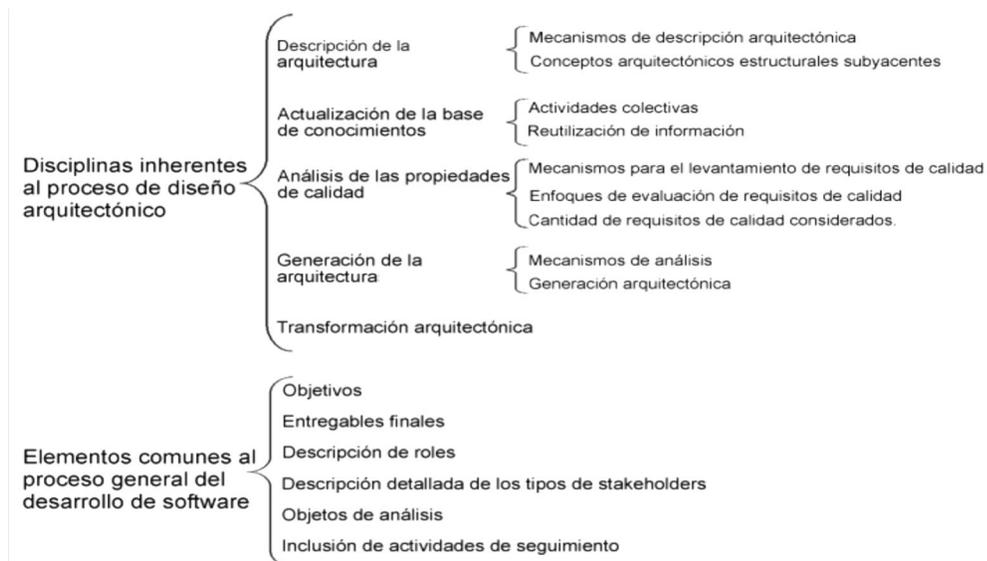


Figura 1. Marco de comparación de métodos arquitectónicos.

### Disciplinas inherentes al proceso de diseño arquitectónico

A continuación se presentan cinco disciplinas o conjunto de actividades básicas consideradas como esenciales y que deben estar presentes en cualquier proceso de diseño arquitectónico. Estas son:

**a. Descripción de la arquitectura:** se refiere a las formas a través de las cuales el arquitecto de software expresa distintos aspectos de la arquitectura tales como la estructura, relaciones y comportamientos de cada uno de los componentes involucrados. Los criterios que permitieron comparar a los distintos métodos estudiados desde la perspectiva de esta disciplina fueron: mecanismos de descripción arquitectónica y conceptos arquitectónicos subyacentes.

**b. Actualización de la base de conocimientos:** esta disciplina se refiere a la manera a través de la cual se registra la información obtenida como consecuencia de la aplicación del método a nuevos proyectos. Tal registro permite la reutilización de información de experiencias previas, lo que a su vez permite realizar ajustes en el desarrollo de las actividades y técnicas que conforman el método.

**c. Análisis de las propiedades de calidad:** donde los aspectos relevantes a considerar son los mecanismos propuestos por cada método para el levantamiento y para la evaluación de las propiedades no funcionales. Otro aspecto que es importante considerar es si el método está dirigido a tratar sólo una propiedad de calidad o si por el contrario, analiza múltiples propiedades al unísono.

**d. Generación de la arquitectura:** se refiere a la identificación de los mecanismos a través de los cuales, la información capturada es analizada en virtud de los objetivos del método. Para la comparación de los métodos seleccionados, desde la perspectiva de esta disciplina, se tomaron en cuenta los siguientes aspectos: mecanismos de análisis y mecanismos de generación arquitectónica.

**e. Transformación arquitectónica:** disciplina centrada en la transformación de una arquitectura con el objetivo de responder a requisitos de calidad.

## DESCRIPCIÓN DE LA ARQUITECTURA

### Mecanismos de descripción arquitectónica

Un método de diseño arquitectónico se caracteriza por mantener un nivel de abstracción acorde con la visión arquitectónica del software, en la que se obvian detalles propios de las fases en donde se ejecuta un diseño detallado del sistema. Contar con una notación con la semántica necesaria para expresar los elementos arquitectónicamente significativos es vital para el mantenimiento del nivel de abstracción adecuado al momento de identificar las soluciones arquitectónicas. Usualmente, una notación responde a formas de expresar una arquitectura conocida como vistas arquitectónicas. Una revisión de este criterio en el conjunto de métodos seleccionados arroja como resultado la identificación de cuatro grupos fundamentales (tabla 7). Un primer grupo incluye a los métodos que exigen la descripción de las arquitecturas por medio de vistas. Entendiéndose por vista a un modelo que muestra determinados aspectos del sistema

**Tabla 7.** Notación Arquitectónica.

Descripción de arquitecturas por medio de vistas	Estándar UML	Notación propia	Sin indicaciones sobre exigencias de notación	Otros
ALMA, ATAM, VAP, ABD, ADD, ASAA, SACAM, ALMA.	Losavio-Levy-Ramdane Cherif, QUADRAD, ASAAM	PRESKRIPTOR, CBSP, NFR	ALPSM, AQA, SAE, QAW, SARM, Bosch, Proteus, Lamsweerde, MECABIC, CBAM, CBAM-WIN WIN	SAAM, PASA, ARID

(Krutchen, 1995). En un segundo grupo, podemos incluir a aquellos que exigen o al menos reconocen la necesidad de utilizar el estándar UML como notación. Un tercer grupo incluye a los métodos que proponen una notación propia y un cuarto grupo está constituido por los métodos que no ofrecen indicaciones con respecto a formas de expresar la arquitectura. Es recomendable que un método de diseño arquitectónico cuente con **mecanismos de descripción arquitectónica**. Tales mecanismos deberían fundamentarse en la medida de lo posible en estándares. El uso de una notación particular podría implicar una curva de aprendizaje más pronunciada influyendo en la facilidad de uso del método. Una alternativa podría ser el uso de UML2.0, pues considera algunos conceptos reconocidos como importantes dentro de la comunidad de arquitectos.

### Conceptos arquitectónicos estructurales subyacentes

Los conceptos arquitectónicos estructurales comúnmente aceptados son: componentes, conectores, estilos arquitectónicos, patrones, puertos, interfaces y vistas. Estos conceptos son utilizados para ocultar detalles de implementación y de diseño detallado. En la tabla 8 se pueden observar los conceptos asumidos por cada método estudiado.

### ACTUALIZACIÓN DE LA BASE DE CONOCIMIENTOS

#### Actividades colectivas

La mayoría de los métodos estudiados evidencia un alto grado de dependencia con respecto a la opinión de los expertos. Cada uno de estos, posee un área de experticia que puede ayudar a identificar las soluciones más idóneas, por lo que su participación durante el proceso de diseño arquitectónico debe realizarse de forma tal que las distintas opiniones y puntos de vista puedan ser contrastados. Como resultado de la comparación se identifican cinco grupos: el primero incluye a los métodos en los que todas las actividades son explícitamente grupales. En el segundo, podemos ubicar a aquellos métodos en los que la mayoría de las actividades son realizadas de manera grupal. En un tercer

grupo se incluyen a los métodos en los que sólo algunas actividades son presentadas como grupales. En un cuarto grupo, se ubican a aquellos métodos en los que todas las actividades pueden ser realizadas de manera individual o grupal. Finalmente, un quinto grupo incluye a los métodos en los que no fue posible determinar cuáles actividades eran grupales o individuales (tabla 9).

#### Reutilización de información

La reutilización de información se apoya en actividades y artefactos que registren la experticia adquirida por la organización durante la aplicación de un método a distintos casos o proyectos. Lo anterior adquiere gran importancia cuando la ejecución del método se realiza en el contexto de un dominio de aplicación específico o en el caso de empresas dedicadas a la generación de familias de productos. El registro del conocimiento adquirido, debería ejecutarse paralelamente a la aplicación del método.

En relación a este aspecto, únicamente los métodos FAAM, ATAM, ABD y Bosch incluyen actividades dirigidas en tal dirección. La reutilización debería ser en dos sentidos: en relación al desarrollo del método en sí; y en relación al conocimiento adquirido sobre los mecanismos y soluciones arquitectónicas probadas por la organización.

#### ANÁLISIS DE LAS PROPIEDADES DE CALIDAD

##### Mecanismos para el levantamiento de requisitos de calidad

Se refiere a la heurística para la identificación y especificación de las propiedades de calidad asociadas a los requisitos funcionales y no funcionales. Estos requisitos deben haber sido levantados y especificados, por ejemplo, en un documento SRS (Software Requirements Specification), utilizado comúnmente, por lo tanto esta disciplina puede incluirse con facilidad en la ingeniería de requisitos. Los requisitos de calidad deben ser analizados para determinar la idoneidad de una arquitectura con respecto a estos o para decidir entre varias alternativas arquitectónicas. En el ám-

**Tabla 8.** Conceptos arquitectónicos utilizados en los métodos estudiados.

	Conceptos Arquitectónicos									Otros Conceptos							
	Componentes	Conectores	Puertos	Interfaces	Vistas	Estilos arquitectónicos	Patrones Arquitectónicos	Propiedades	Estrategias Arquitectónicas	Patrones de Diseño	Unidades de Software	Conexiones	Restricciones	Módulos	Relaciones	Topologías	N/A
SAAM	•																
ATAM	•																
AQA	•				•						•	•					
FAAM	•						•								•		
ALMA	•																
SAE																	•
ALPSM	•																
ASAAM	•																
SACAM	•	•															
PASA							•			•							
CBAM-WIN WIN								•									
CBAM								•									
QAW																	•
CBSP	•	•														•	
ARID																	•
PRESKRIPTOR	•	•					•										
MECABIC	•		•		•												
ASAA	•	•		•													
Losavio	•	•					•	•				•					
VAP	•			•							•						
Proteus						•	•										
Bosch	•	•				•	•		•								
Lamsweerde	•	•				•	•										
QUADRAD	•	•			•		•										
ADD				•									•				
ABD	•																
SARM	•																

**Tabla 9.** Cantidad de Actividades colectivas.

Todas las actividades son grupales	SAE, QAW, PASA, PRESKRIPTOR, CBAM-WIN WIN, MECABIC, QUADRAD
Mayoría de las actividades son grupales	ATAM, VAP, CBAM
Algunas actividades son grupales	ALMA, ARID, CBSP, Losavio-Levy-Ramdane Cherif, SAAM
Todas las actividades pueden ser realizadas en grupo o individualmente	FAAM
Sin determinar	ALPSM, AQA, ASAAM, ABD, SACAM, BOSCH, Proteus, Lamsweerde, ADD, SARM, ASAA

bito del diseño arquitectónico, tales requisitos son el punto de partida para la generación de una arquitectura inicial o se constituyen en directrices para una transformación arquitectónica.

La comparación de los métodos seleccionados muestra que, en la mayoría, el mecanismo de levantamiento de requisitos más utilizado es la generación de escenarios, seguido por el análisis de casos de uso y la construcción de modelos de calidad que pueden estar representados por modelos de objetivos o por un árbol de utilidad. En la tabla 10 se puede observar que algunos de los métodos estudiados combinan los mecanismos mencionados.

### Enfoques de evaluación de requisitos de calidad

En general los enfoques de evaluación de atributos de calidad que utilizan los distintos métodos pueden ser: análisis de escenarios, simulación, modelos matemáticos y razonamiento basado en la experiencia (Bosch, 2000). En la evaluación basada en el análisis de escenarios, un conjunto de estos es desarrollado para expresar de forma más concreta los requisitos de calidad. La simulación se puede realizar a través de la utilización de un ADL que esté soportado

por herramientas capaces de generar modelos de simulación. Por otra parte, los modelos matemáticos pueden ser una alternativa a la simulación; sin embargo ambos pueden coexistir. El cuarto enfoque tiene un componente subjetivo alto, lo cual no debería conducir a menospreciarlo, puesto que la mayoría de los arquitectos experimentados han adquirido una capacidad para intuir aspectos potencialmente problemáticos en un diseño. El problema realmente no está en la subjetividad de cada especialista sino más bien en que tales opiniones sean argumentadas de manera objetiva para que puedan ser valoradas por otros especialistas. Técnicas de votación y tormenta de ideas permiten que la pertinencia de las distintas argumentaciones pueda ser contrastada para lograr un cuerpo de decisiones sólidamente sustentadas. La tabla 11 resume los enfoques identificados en los métodos estudiados.

### Cantidad de requisitos de calidad considerados

Un método puede considerar la evaluación de un único atributo de calidad, o varios atributos al mismo tiempo. Una revisión de varios atributos de calidad podría implicar una mayor complejidad en las actividades de análisis, demandando una mayor participación de distintos especialistas y

**Tabla 10.** Mecanismos para el levantamiento de requisitos de calidad.

Generación de escenarios	Análisis de casos de uso	Árbol de utilidad	Modelo de objetivos	Levantamiento previo de requisitos	Otros
SAAM, ALPSM ASAAM QAW, ABD, SACAM QUADRAD	MECABIC		PRESKRIPTOR Lamsweerde Proteus ALMA	CBSP, SARM, Bosch, ARID, CBAM y ASAA	VAP, CBAM- WIN WIN, SAE, AQA
PASA, ADD					
	Losavio-Chirinos- Levy-Ramdane Cherif (combinado con ISO/IEC 9126-1)				
	FAAM (casos de cambio)				

**Tabla 11.** Enfoques de evaluación de requisitos de calidad.

Enfoque	Métodos
Análisis de escenarios	ATAM, ALPSM, MECABIC, CBAM, QUADRAD, FAAM
Simulación	Bosch
Modelos matemáticos	Bosch
Razonamiento basado en la experiencia	ALMA, AQA, SAE, Losavio-Levy-Ramdane Cherif
Sin especificar	SARM
Relaciones AND/OR (tradeoff)	Proteus, Lamsweerde
Análisis de factores	ASAA

la necesaria negociación entre estos, así como la documentación de los efectos que unos atributos tengan sobre otros.

A este respecto, se identifica un primer grupo integrado por métodos que contemplan distintos atributos de calidad. En dicho grupo tenemos a los siguientes métodos: QAW, ATAM, CBSP, PROCESO PRESKRIPTOR, VAP, CBAM-WIN WIN, ABD, SACAM, SARM, Bosch, MECABIC, Losavio-Chirinos-Levy-Ramdane Cherif, CBAM, QUADRAD, Lamsweerde y ADD. En un segundo grupo tenemos a aquellos métodos que consideran a uno o a unos pocos atributos de calidad. Este grupo está integrado por los siguientes métodos: SAAM, ALPSM, AQA, SAE, FAAM, ASAAM, ALMA, PASA y Proteus. ASAA se refiere a “factores” que influyen en la arquitectura. En el caso de ARID, por su propia naturaleza, no se hace referencia a los atributos de calidad, puesto que su objetivo es evaluar diseños detallados de unidades de software coherentes tales como módulos o componentes, lo que implica un nivel de abstracción muy bajo con respecto a los otros métodos estudiados.

## GENERACIÓN DE LA ARQUITECTURA

### Mecanismos de análisis

La tabla 12 muestra un resumen de los mecanismos de análisis a través de los cuales se busca encontrar posibles soluciones. En este trabajo se han identificado los siguientes: análisis de escenarios, análisis de vistas arquitectónicas, análisis de casos de uso, generación de especificaciones, listas de chequeo o cuestionarios, análisis de efectos colaterales (tradeoff), estimaciones cuantitativas, entre otros.

**Tabla 12.** Mecanismos de análisis.

QUADRAD, ALMA, SAAM, ALPSM, ASAAM, ATAM, QAW, ARID, SARM	Análisis de escenarios
ADD, CBSP	Análisis de escenarios Análisis de casos de uso
FAAM, VAP	Análisis de vistas arquitectónicas
AQA, SAE, MECABIC	Cuestionarios
PRESKRIPTOR, Losavio-Chirinos-Levy Ramdane Cherif	Generación de especificaciones
Proteus, Lamsweerde	Análisis de efectos colaterales (tradeoff)
PASA, Bosch	Identificación de estilos y patrones arquitectónicos
CBAM, CBAM-WIN WIN	Estimaciones cuantitativas

## Generación arquitectónica

Los mecanismos de generación arquitectónica son los que posibilitan la estructuración de una arquitectura base. Entre los métodos de diseño arquitectónico, se identifican aquellos que proponen mecanismos que intentan dar un carácter más formal a los modelos generados y que además utilizan una heurística basada en la derivación progresiva de los componentes, topología y demás propiedades de una arquitectura. De los métodos estudiados, CBSP, PRESKRIPTOR y el método propuesto por Lamsweerde, muestran esta tendencia. En otros métodos no se indican claramente los mecanismos que permitan la transición progresiva y sistemática desde un conjunto de requisitos hasta una primera propuesta arquitectónica.

## TRANSFORMACIÓN ARQUITECTÓNICA

Usualmente, la evaluación de una arquitectura resulta en la identificación de fortalezas y debilidades en esta. Si existen aspectos riesgosos, estos deben ser resueltos a través de un proceso de transformación arquitectónica que consiste en un rediseño de la misma. Tal transformación se lleva a cabo a través de la aplicación de mecanismos o patrones arquitectónicos, convirtiendo requisitos en funcionalidades o distribuyendo responsabilidades entre los componentes de la arquitectura (Bosch, 2000).

Como es de esperarse, ninguno de los métodos de evaluación contempla algún tipo de transformación arquitectónica. En cuanto a los métodos de diseño, SARM, hace referencia explícitamente a la transformación arquitectónica, proponiendo cinco categorías de transformaciones: estilos arquitectónicos, patrones arquitectónicos, patrones de diseño conversión de requisitos en funcionalidades y distribución de requisitos. Igualmente, en Bosch se contemplan como mecanismos de transformación a los estilos arquitectónicos, patrones arquitectónicos y patrones de diseño. En el método de Losavio-Chirinos-Levy-Ramdane Cherif únicamente se hace referencia a los patrones arquitectónicos; así como en CBAM se hace referencia a estrategias arquitectónicas. En QUADRAD se producen transformaciones por medio de la aplicación de decisiones arquitectónicas. En ADD, el sistema se descompone recursivamente en módulos.

## ELEMENTOS COMUNES AL PROCESO GENERAL DEL DESARROLLO DEL SOFTWARE

A continuación se presentan algunos elementos que en general están presentes en cualquier proceso de desarrollo de software.

### Objetivos

Muchos métodos comparten un objetivo general, como

puede ser identificar requisitos, evaluar o diseñar una arquitectura. Sin embargo, un conjunto de métodos de evaluación o de diseño se diferencia entre sí por los objetivos específicos que persigue (tabla 13).

### Entregables finales

Una comparación de los métodos estudiados, evidencia que el entregable final que más se genera es precisamente la arquitectura (en 14 de los 27 métodos estudiados), seguida de los enfoques arquitectónicos (10/27). Junto a estos se pueden nombrar los siguientes: lista de riesgos (7/27), puntos sensibles (7/27), lista de fortalezas (6/27), requisitos de calidad (5/27) y lista de restricciones (4/27).

### Descripción de roles en la aplicación del método

La descripción de los roles que intervienen en la aplicación de un método son importantes, pues ayuda a la distribución de responsabilidades propias del método; además, favorece la aplicabilidad del mismo en situaciones reales. En lo referente a este aspecto, encontramos que la gran mayoría

de los métodos estudiados no especifica detalladamente los roles de los participantes. La tabla 14 muestra un resumen de los resultados obtenidos.

### Descripción detallada de los tipos de stakeholders

Es importante disponer de los perfiles de los distintos interesados (stakeholders) de los cuales se necesita su participación en el método. Cada especialista posee una experticia que puede nutrir determinados aspectos de la arquitectura.

En relación a este aspecto, los siguientes métodos no describen explícitamente los tipos de stakeholder participantes: ASAAM, ALMA, PASA, CBSP, SACAM, SARM, BOSH, Proteus, Lamsweerde, Losavio-Chirinos-Levy-Ramdane Cherif, ADD y ASAA. En este caso, podría inferirse que el tipo stakeholder que participa es el arquitecto de software. Por otra parte, APSM y CBAM utilizan el término stakeholders en general, dejando a criterio del equipo los tipos que se consideren necesarios según las características del sistema a desarrollar. Los métodos PRESKRIPTOR, VAP y CBAM-WIN WIN, únicamente hacen referencia a los ar-

**Tabla 13.** Objetivos.

Objetivos Generales	Objetivos Específicos	Métodos
Levantamiento de requisitos	Identificación de requisitos	QAW
	Identificación de requisitos y riesgos	CBAM-WIN WIN, CBAM, ASAA
Diseño de arquitectura	Derivación arquitectónica	CBSP, PRESKRIPTOR, Lamsweerde y Proteus
	Identificación de arquitectura inicial	VAP, ABD, Bosch, QUADRAD, ADD, Losavio-Levy-Ramdane Cherif
	Transformación arquitectónica	SARM
Evaluación de la arquitectura	Evaluación de un atributo	SAAM, ALMA, ALPSM, PASA
	Evaluación de varios atributos	ATAM, ASAAM, AQA, SACAM, MECABIC
	Evaluación detallada	ARID
	Evaluación de dominio específico	SAE, FAAM

**Tabla 14.** Descripción de roles.

No especifican roles	ALPSM, AQA, ASAAM, ALMA, PASA, PRESKRIPTOR, VAP, CBAM-WIN WIN, ABD, SACAM, SARM, Proteus, Lamsweerde, MECABIC, Losavio-Chirinos-Levy-Ramdane Cherif, CBAM, QUADRAD, ADD y ASAA		
Especifican un rol	SAAM	Rol: revisor o crítico	Cada especialista es un revisor durante la aplicación del método
	SAE	Rol: evaluador	Más bien se habla de equipo evaluador
	FAAM	Rol: facilitador	Puede ser asumido, según el paso que se esté desarrollando, por los stakeholders o el arquitecto
	CBSP	Rol: arquitecto	
	BOSCH		
Especifican dos a más roles	QAW	Roles: guía, secretario	

arquitectos de software, mientras que ABD, sólo hace mención a los diseñadores. Los métodos AQA y MECABIC, mencionan a los evaluadores de software y a los arquitectos de software, mientras que ARID menciona a los ingenieros de software y a los arquitectos. Los métodos SAAM, SAE, FAAM, QUADRAD nombran varios stakeholders. Específicamente, SAAM demanda la participación del usuario final, cliente, especialista en mercadeo, administrador de sistema, mantenedor, desarrollador y arquitecto. SAE, menciona a los desarrolladores, arquitectos y administradores de proyecto. FAAM presenta una descripción bien detallada de los stakeholders, colocando especial énfasis en el redimensionamiento de las tareas de cada uno en el contexto de la evaluación de arquitecturas de la familia de sistemas de información. QUADRAD hace referencia a los siguientes stakeholders: arquitecto, ingeniero de requisitos, ejecutivo de negocios, equipo de evaluadores, y de manera general a los interesados en el sistema. QAW nombra algunos stakeholders aunque no se ofrece una descripción detallada de las responsabilidades de estos. Finalmente, un caso especial lo constituye ATAM, pues proporciona la descripción más completa de los distintos tipos de stakeholders que pueden participar en el método.

### **Identificación de objetos de análisis**

Constituyen las entidades sobre las que se centran las actividades de análisis. Estos objetos están determinados por los objetivos del método y por la infraestructura conceptual sobre la cual este se fundamenta.

En el caso de los métodos de evaluación, los objetos de análisis son en general la arquitectura y su respuesta frente a un grupo de requisitos de calidad. Lo anterior es complementado en muchos casos con el análisis de las relaciones entre los atributos de calidad y mecanismos arquitectónicos. En el caso de los métodos de diseño, los objetos de análisis vienen dados por un conjunto de requisitos (funcionales y no funcionales). Junto al análisis de requisitos de calidad, las relaciones entre estos, es también objeto de análisis en muchos casos.

### **Inclusión de actividades para el seguimiento (trazabilidad)**

Parece conveniente otorgar un carácter más holístico al diseño arquitectónico dentro del proceso de desarrollo. Aunque éste se centra en el ámbito de la solución del problema, debería trascenderlo para extenderse hacia la ingeniería de requisitos y al unísono hacia las fases posteriores de diseño detallado. Los métodos de diseño arquitectónico deberían contemplar actividades que permitan al arquitecto realizar el seguimiento de las decisiones tomadas durante esta fase

en las subsecuentes actividades propias del diseño detallado del sistema. En este sentido, únicamente AQA, SAE y VAP hacen alguna propuesta. En el primer caso, se hace referencia a la posibilidad de que los arquitectos utilicen los entregables finales para realizar las modificaciones necesarias que disminuyan los riesgos identificados, sin que esto sea obligatorio. Por otra parte, en SAE se genera un reporte final que resume las fortalezas y debilidades de la arquitectura. De igual forma, VAP establece actividades de seguimiento en la fase de despliegue arquitectónico.

### **CARACTERÍSTICAS DESEABLES EN UN MÉTODO DE DISEÑO ARQUITECTÓNICO**

Del marco de comparación propuesto se ha inferido un conjunto de características deseables en un método de diseño arquitectónico, a saber:

1. Poseer un modelo conceptual de la arquitectura: el cuerpo de conceptos sobre los que se fundamenta el método debe contemplar todas las definiciones que son universalmente aceptadas en el área de la arquitectura de software, preferiblemente estándares.
2. Tratar explícitamente los requisitos no funcionales: de estos depende fundamentalmente el diseño arquitectónico final del sistema.
3. Proporcionar mecanismos de derivación arquitectónica: que permitan al arquitecto de software convertir un conjunto de requisitos en elementos de valor arquitectónico.
4. Aplicar mecanismos de descripción arquitectónica con el nivel de abstracción adecuado.
5. Tener como objetivo el logro de cualquier combinación de características de calidad en general.
6. Proporcionar actividades de negociación entre los especialistas involucrados (Stakeholders).
7. Proporcionar actividades de transformación arquitectónica para satisfacer el modelo de calidad del sistema.
8. Estar apoyado por herramientas de software que auxilien al arquitecto en la recolección de datos.
9. Estar validado: como resultado de la aplicación del método a varios casos.
10. Ser fácil de usar: característica que es favorecida si el método cuenta con un marco conceptual sencillo.

## UNA PROPUESTA DE PROCESO DE DISEÑO ARQUITECTÓNICO

Como resultado de la comparación de los distintos métodos arquitectónicos, se propone una primera versión de un marco conceptual para un método de diseño arquitectónico. La figura 2 muestra los artefactos o productos que se obtienen en cada etapa del marco conceptual propuesto para este proceso. A continuación se describen las etapas del mismo.

**Etapas de preparación:** en la que se organizan los requisitos. Se supone la existencia previa de un conjunto de requisitos. Esta etapa se divide en dos actividades:

a. Generación del Modelo de Casos de uso: compuesto por el diagrama de casos de uso y una especificación detallada de los mismos. A través de las relaciones *extend*, *incluye* y *use*, se originan casos de uso específicamente arquitectónicamente significativos.

b. Generación del Modelo de Calidad: cuyo objetivo es capturar los requisitos no funcionales del sistema. Para esto, se puede partir de los atributos que fueron identificados y asignados a los casos de uso durante su especificación. El artefacto final debe ser un *árbol de calidad* (Kazman *et al.* 1998). Este se refina hasta que en las hojas se encuentren escenarios en los que, según el criterio de los especialistas; se pongan de manifiesto todos los atributos identificados. Posteriormente, se clasifican los escenarios ubicados en las hojas según la taxonomía de artefactos CBSP propuesta por Grünbacher *et al.* (2003). Esta taxonomía está compuesta por las siguientes categorías: artefactos que involucran componentes (C); artefactos relacionados con propiedades

de componentes (CP); artefactos que involucran a un conector (B); artefactos que involucran a una amplia sección de componentes del sistema (S); artefactos relacionados con propiedades de un conector (BP) y artefactos relacionados con propiedades del sistema (SP).

**Etapas de derivación arquitectónica:** en esta etapa se genera una arquitectura base sobre la cual se pueden realizar transformaciones, para así lograr que ésta responda a los requisitos. Se compone de dos actividades:

a. Generación del modelo de componentes: partiendo del modelo de casos de uso y del modelo de calidad. En primer lugar, se toman los casos de uso ubicados como hojas del diagrama de casos de uso y se les asigna un componente, expresado en UML. Por otra parte, se hace algo similar a partir del modelo de calidad expresado en el árbol de calidad; a cada escenario hoja, clasificado como un artefacto C, se le asigna un componente, mientras que a los que se les ha clasificado como artefactos CP, se les intenta ubicar como propiedades de los componentes.

b. Generación del modelo CC (modelo de componentes-conectores): lo que se logra asignando relaciones entre los componentes generados en la actividad anterior. Tales relaciones pueden derivarse en conectores o en relaciones de composición. Por otra parte, a los escenarios que en el árbol de calidad fueron clasificados como artefactos B, se les asignan conectores que a su vez son asignados a los componentes existentes. A los escenarios que se identificaron como artefactos BP, se les intenta ubicar como propiedades de los conectores.

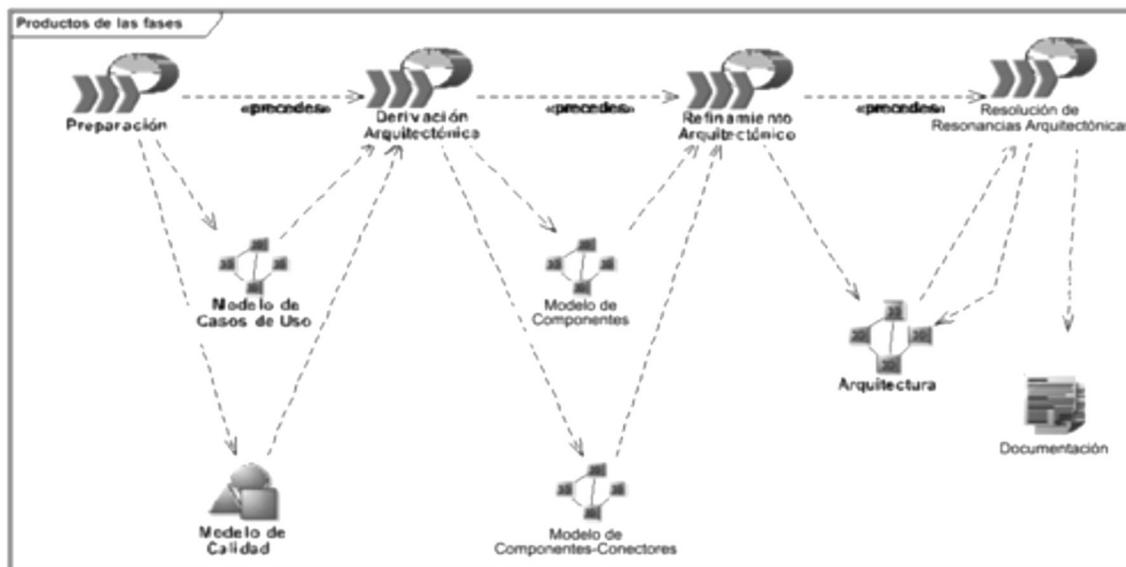


Figura 2. Productos de las fases del método propuesto. Descripción realizada utilizando SPEM (Software Process Engineering Metamodel Specification) (OMG, 2005b).

### **Etapas de refinamiento arquitectónico (agrupa las siguientes actividades):**

- a. Identificación de subsistemas, estilos y patrones: lo que se logra a partir de los escenarios identificados como artefactos S o SP. Igualmente se determina la estructura interna de tales subsistemas a partir de los componentes identificados en el modelo CC.
- b. Identificar los estilos y/o patrones arquitectónicos que respondan a los artefactos SP.
- c. Aplicación de estilos y patrones arquitectónicos: una vez identificados se aplican al modelo CC, lo que transformará la arquitectura para conseguir responder a los requisitos de calidad. El resultado es la arquitectura base.

### **Etapas de resolución de resonancias arquitectónicas (compuesta por las siguientes actividades):**

- a. Evaluación de la arquitectura: se valida la arquitectura según ATAM (Kazman *et al.* 1998). Como resultado, se generan los siguientes entregables: un conjunto de riesgos; un conjunto de fortalezas; un conjunto de aspectos sensibles y efectos colaterales; una lista de enfoques y mecanismos arquitectónicos.
- b. Transformación de arquitectónica: Si el conjunto de riesgos es notable, entonces se deben aplicar los enfoques y mecanismos arquitectónicos que permitan solucionarlos, lo que conduciría a una transformación de la arquitectura obtenida.

### **CONCLUSIONES**

La comparación realizada permitió identificar un cuerpo de características deseables en un método de diseño arquitectónico. A partir de éstas se ha propuesto un método de diseño arquitectónico que integra distintas propuestas que se han realizado alrededor del diseño arquitectónico colocando énfasis en la coherencia entre los distintos entregables generados en cada etapa, el tratamiento de los atributos de calidad, y la apropiada selección de mecanismos de descripción que aseguren la correcta comunicación entre los distintos especialistas.

Por otra parte no se ha realizado una evaluación de métodos, sin embargo, el conjunto de características deseables puede ser aplicado en este ámbito. En futuros trabajos se refinarán tales características para obtener métricas de permitan su aplicación en este sentido. Actualmente, el método está siendo aplicado a un caso de estudio en el área de los Sistemas Tutoriales Inteligentes y también al diseño arqui-

tectónico de aplicaciones basadas en servicios web.

### **AGRADECIMIENTO**

Los autores agradecen altamente a los árbitros por sus observaciones y aportes constructivos y al Consejo de desarrollo científico y Humanístico (CDCH) de la Universidad Central de Venezuela por haber financiado parcialmente esta investigación.

### **REFERENCIAS**

- ASUNDI, J. & KAZMAN, R. (2001). A Foundation for the Economic Analysis of Software Architectures. Third International Workshop on Economics-Driven Software Engineering Research, s/n.
- AT&T (1993). Best Current Practices: Software Architecture Validation, s/n.
- BABAR, M. & GORTON, I. (2004). Comparison of scenario-based software architecture evaluation methods. National ICT Australia Ltd. and University of New South Wales, Australia, s/n.
- BABAR, M., JEFFERY, R., ZHU, L. (2004). A Framework for Classifying and Comparing Software Architecture Evaluation Methods. National ICT Australia Ltd. And University of New South Wales, Australia, s/n.
- BACHMANN, F., STOERMER, C., VERHOEF, C. (2003). SACAM: The Software Architecture Comparison Analysis Method. CMU/SEI-2003-TR-006. Carnegie Mellon Software Engineering Institute. Pittsburg, s/n.
- BARBACCI, M., ELLISON, R., LATTANZE, A., STAFFORD, J., WEINSTOCK, C., WOOD, W. (2003). Quality Attribute Workshops (QAWs). CMU/SEI-2003-TR-016. Carnegie Mellon Software Engineering Institute. Pittsburg, s/n.
- BASS, L., CHASTEK, G., DONOHOE, P., PERUZZI, F. (2000). The Architecture Based Design Method. CMU/SEI-2000-TR-001. Carnegie Mellon Software Engineering Institute. Pittsburg, s/n.
- BASS, L., CLEMENTS, P., KAZMAN, R. (2003). Software Architecture in Practice. Massachusetts: Addison-Wesley. Segunda edición, p. 560.
- BENGTSSON, P. & BOSCH, J. (1998). Scenario-based Software Architecture Reengineering. Proceedings of the 5th International Conference on Software Reuse, IEEE, Victoria, Canada, pp. 308-317.

- BENGTSSON, P. & BOSCH, J. (1999). Architecture-Level Prediction of software Maintenance. Proceedings of 3rd EuroMicro Conference on Maintenance and Reengineering, Los Alamitos, CA: IEEE Cs Press, pp. 139-147.
- BENGTSSON, P., BOSCH, J., LASSING, N., VAN VLIET, H. (2004). Architecture-level modifiability analysis (ALMA). The journal of systems and software, (69): 129-147.
- BOSCH, J. (2000). Design & Use of Software Architecture. Addison-Wesley, p. 354.
- BRANDOZZI, M. & PERRY, D. (2002). Architectural Prescriptions for Dependable Systems. ICSE WADS 2.
- BREDEMEYER, D. & MALAN, R. (2005). The Visual Architecting Process. Bredemeyer Consulting. s/n.
- CHUNG, L., COOPER, K., YI, A. (2002). Developing Adaptable Software Architectures Using Design Patterns: a NFR Approach. Department of Computer Science, University of Texas at Dallas.
- CLEMENTS, P. (2000). Active Reviews for Intermediate Designs. CMU/SEI-2000-TN-009. Carnegie Mellon Software Engineering Institute. Pittsburg, s/n.
- CLEMENTS, P., KAZMAN, R., KLEIN, M. (2002). Evaluating Software Architecture: Methods and Case Studies. Boston: Addison-Wesley, p. 447.
- CONNIE, S. & WILLIAMS, L. (2002). PASASM: A Method for the Performance Assessment of Software Architectures. Software Engineering Research and Performance Engineering Services, s/n.
- DOBRICA, L. & NIEMELÄ, E. (2002). A Survey on Software Architecture Analysis Methods. IEEE transactions on Software Engineering; (28)7.
- DOLAN, T. (2001). Architecture Assessment of Information-System Families. Tesis de Doctorado. Technische Universiteit Eindhoven, Holanda.
- GONZÁLEZ, A., GRIMÁN, A., MENDOZA, L., MIJARES, M., PÉREZ, M. (2005). Método de Evaluación de Arquitecturas de Software Basadas en Componentes (MECABIC). Universidad Simón Bolívar, s/n.
- GRIMÁN, A., PÉREZ, M., MENDOZA, L., LOSAVIO, F. (2006). Feature Analysis for Architectural Evaluation Methods. Journal of Systems & Software, Elsevier Science (North Holland) (79)6: pp. 871-888.
- GRUNBACHER, P., EGYED, A., MEDVIDOVIC, N. (2003). Reconciling Software Requirements and Architecture: The CBSP Approach. 2nd International Workshop on Traceability In Emerging Forms of Software Engineering (TEFSE) with ASE 2003, Montreal, Canada, s/n.
- HAMMER, D., IONITA, M., OBBINK, H. (2002). Scenario-Based Software Architecture Evaluation Methods: An Overview, s/n.
- HILLIARD, I., KURLAND, R., LITVINTCHOUK, S. (1997). MITRE's Architecture Quality Assessment. Software Engineering & Economics Conference. The MITRE Corporation, s/n.
- HOFMEISTER, C., NORD, R., SONI, D. (2000). Applied Software Architecture. Addison-Wesley, p. 387.
- KAZMAN, H.R., & OLSON, D. (2001). From Requirements Negotiation to Software Architectural Decisions, s/n.
- ISO/IEC 9126-1. (2001) Quality characteristics and guidelines for their use. International Organisation for Standardisation / International Electrotechnical Commission, s/n.
- KAZMAN, R. & NORD, R. (2003). A Life-Cycle View of Architecture Analysis and Design Methods. CMU/SEI-2003-TN-026, s/n.
- KAZMAN, R., KLEIN, M., BARBACCI, T., LONGSTAFF, H., LIPSON, H., CARRIERE, J. (1998). The Architecture Tradeoff Analysis Method. IEEE, ICECCS, s/n.
- KITCHENHAM, B. (1996). DESMET: A Method for Evaluating Software Engineering Methods and Tools. Department of Computer Science. University of Keele. U.K., s/n.
- KRUTCHEN, P. (1995). Architectural Blueprints—The “4+1” View Model of Software Architecture. Rational Software Corp. IEEE Software 12 (6):p. 42-50.
- LAMSWEERDE, A. (2003). From System Goals to Software Architecture. Formal Methods for Software Architectures. Springer-Verlag, s/n.
- LOSAVIO, F., CHIRINOS, L., LÉVY, N., RAMDANE-CHERIF, A. (2003). Quality Characteristics for Software Architecture. Journal of Object Technology (JOT). Vol 2, No. 2, pp.133-150, [http://www.jot.fm/issues/issue\\_2003\\_03/article2](http://www.jot.fm/issues/issue_2003_03/article2). s/n.

- LOSAVIO, F., CHIRINOS, L., PÉREZ, M. (2001). Feature Analysis for quality-based architectural design methods. XI Encuentro Chileno de Computación, Jornadas Chilenas de Computación 2001 (SCCC), Punta Arenas (Magallanes), Chile, 5-10 Noviembre, proceedings CD-rom, [www.umag.cl/ec](http://www.umag.cl/ec). s/n.
- ORBINK, H., NORD, R., KRUCHTEN, P., HOFMEISTER, C., AMERICA, P., RAN, A. (2007). A general model of software architecture design derived from five industrial approaches. *The Journal of Systems and Software* (80):106-126.
- OMG (Object Management Group) (2005a). Unified Modeling Language: Superstructure. version 2.0. formal/05-07-04, s/n.
- OMG (Object Management Group) (2005b). Software Process Engineering Metamodel Specification. Version 1.1. formal/05-01-06, s/n.
- TEKINERDOGAN, B. & MEHMET, A. (2000). Classifying and Evaluating Architecture Design Methods. TRESE Group, Department of Computer Science, University of Twente, s/n.
- TEKINERDOGAN, B. (2003). ASAAM: Aspectual Software Architecture Analysis Method. Turkey: Department of Computer Engineering, Bilkent University, s/n.
- THIEL, S. (2005). Framework to Improve the Architecture Quality of Software-Intensive Systems. Tesis de Doctorado. Universität Duisburg-Essen, Alemania, s/n.