

## INTRODUCCIÓN A LA ARQUITECTURA DE SOFTWARE

Extracto de un documento escrito por:

Carlos Billy Reynoso - UNIVERSIDAD DE BUENOS AIRES

### 1. DEFINICIONES.

No es novedad que ninguna definición de la AS es respaldada unánimemente por la totalidad de los arquitectos. El número de definiciones circulantes alcanza un orden de tres dígitos, amenazando llegar a cuatro. De hecho, existen grandes compilaciones de definiciones alternativas o contrapuestas, como la colección que se encuentra en el SEI (<http://www.sei.cmu.edu/architecture/definitions.html>), a la que cada quien puede agregar la suya. En general, las definiciones entremezclan despreocupadamente:

- (1) el trabajo dinámico de estipulación de la arquitectura dentro del proceso de ingeniería o el diseño (su lugar en el ciclo de vida),
- (2) la configuración o topología estática de sistemas de software contemplada desde un elevado nivel de abstracción y
- (3) la caracterización de la disciplina que se ocupa de uno de esos dos asuntos, o de ambos.

Una definición reconocida es la de Clements: La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

En una definición semejante, hay que aclararlo, la idea de “componente” no es la de la correspondiente tecnología de desarrollo (COM, CORBA Component Model, EJB), sino la de elemento propio de un estilo. Un componente es una cosa, una entidad, a la que los arquitectos prefieren llamar “componente” antes que “objeto”, por razones que se verán en otros documentos de esta serie pero que ya es obvio imaginar cuáles han de ser.

A despecho de la abundancia de definiciones del campo de la AS, existe en general acuerdo de que ella se refiere a la estructura a grandes rasgos del sistema, estructura consistente en componentes y relaciones entre ellos. Estas cuestiones estructurales se vinculan con el diseño, pues la AS es después de todo una forma de diseño de software que se manifiesta tempranamente en el proceso de creación de un sistema; pero este diseño ocurre a un nivel más abstracto que el de los algoritmos y las estructuras de datos. En el que muchos consideran un ensayo seminal de la disciplina, Mary Shaw y David Garlan sugieren que dichas cuestiones estructurales incluyen organización a grandes rasgos y estructura global de control; protocolos para la comunicación, la sincronización y el acceso a datos; la asignación de funcionalidad a elementos del diseño; la distribución física; la composición de los elementos de diseño; escalabilidad y rendimiento; y selección entre alternativas de diseño.

En una definición tal vez demasiado amplia, David Garlan establece que la AS constituye un puente entre el requerimiento y el código, ocupando el lugar que en los gráficos antiguos se reservaba para el diseño. La definición “oficial” de AS se ha acordado que sea la que brinda el documento de IEEE Std 1471-2000, adoptada también por Microsoft, que reza así:

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Aunque las literaturas de ambos campos rara vez convergen, entendemos que es productivo contrastar esa definición con la de ingeniería de software, conforme al estándar IEEE 610.12.1990:

La aplicación de una estrategia sistemática, disciplinada y cuantificable al desarrollo, aplicación y mantenimiento del software; esto es, la aplicación de la ingeniería al software.

Obsérvese entonces que la noción clave de la arquitectura es la organización (un concepto cualitativo o estructural), mientras que la ingeniería tiene fundamentalmente que ver con una sistematicidad susceptible de cuantificarse.

Ante el número y variedad de definiciones existentes de AS, Mary Shaw y David Garlan proporcionaron una sistematización iluminadora, explicando las diferencias entre definiciones en función de distintas clases de modelos. Destilando las definiciones y los puntos de vista implícitos o explícitos, los autores clasifican los modelos de esta forma:

**1) Modelos estructurales:** Sostienen que la AS está compuesta por componentes, conexiones entre ellos y (usualmente) otros aspectos tales como configuración, estilo, restricciones, semántica, análisis, propiedades, racionalizaciones, requerimientos, necesidades de los participantes. El trabajo en esta área está caracterizada por el desarrollo de lenguajes de descripción arquitectónica (ADLs).

**2) Modelos de framework:** Son similares a la vista estructural, pero su énfasis primario radica en la (usualmente una sola) estructura coherente del sistema completo, en vez de concentrarse en su composición. Los modelos de framework a menudo se refieren a dominios o clases de problemas específicos. El trabajo que ejemplifica esta variante incluye arquitecturas de software específicas de dominios, como CORBA, o modelos basados en CORBA, o repositorios de componentes específicos, como PRISM.

**3) Modelos dinámicos:** Enfatizan la cualidad conductual de los sistemas. “Dinámico” puede referirse a los cambios en la configuración del sistema, o a la dinámica involucrada en el progreso de la computación, tales como valores cambiantes de datos.

**4) Modelos de proceso:** Se concentran en la construcción de la arquitectura, y en los pasos o procesos involucrados en esa construcción. En esta perspectiva, la arquitectura es el resultado de seguir un argumento (script) de proceso. Esta vista se ejemplifica con el actual trabajo sobre programación de procesos para derivar arquitecturas.

**5) Modelos funcionales:** Una minoría considera la arquitectura como un conjunto de componentes funcionales, organizados en capas que proporcionan servicios hacia arriba. Es tal vez útil pensar en esta visión como un framework particular.

Ninguna de estas vistas excluye a las otras, ni representa un conflicto fundamental sobre lo que es o debe ser la AS. Por el contrario, representan un espectro en la comunidad de investigación sobre distintos énfasis que pueden aplicarse a la arquitectura: sobre sus partes constituyentes, su totalidad, la forma en que se comporta una vez construida, o el proceso de su construcción. Tomadas en su conjunto, destacan más bien un consenso.

Independientemente de las discrepancias entre las diversas definiciones, es común entre todos los autores el concepto de la arquitectura como un punto de vista que concierne a un alto nivel de abstracción. Revisamos las diversas definiciones del concepto de abstracción en un apartado específico más adelante en este estudio.

Es casi seguro que la percepción de la AS que prevalece entre quienes no han tenido contacto con ella, así como los estereotipos dominantes sobre su naturaleza, o los nombres que se escogerían como sus personalidades más importantes, difieren sustancialmente de lo que es el caso en el interior de la especialidad. Este sería acaso un ejercicio digno de llevarse a cabo alguna vez.

## 2. CAMPOS DE LA ARQUITECTURA DE SOFTWARE.

La AS es hoy en día un conjunto inmenso y heterogéneo de áreas de investigación teórica y de formulación práctica, por lo que conviene aunque más no sea enumerar algunos de sus campos y sus focos. Una semblanza semejante (de la que aquí se proporciona sólo un rudimento) dudosamente debería ser estática.

La AS comenzó siendo una abstracción descriptiva puntual que en los primeros años no investigó de manera sistemática ni las relaciones que la vinculaban con los requerimientos previos, ni los pasos metodológicos a dar luego para comenzar a componer el diseño. Pero esa sincronidad estructuralista no pudo sostenerse. Por el contrario, daría la impresión que, a medida que pasa el tiempo, la AS tiende a redefinir todos y cada uno de los aspectos de la disciplina madre, la ingeniería de software, sólo que a un mayor nivel de abstracción y agregando una nueva dimensión reflexiva en lo que concierne a la fundamentación formal del proceso.

Hay unas pocas caracterizaciones (y mucha actividad de copiado y pegado) en torno de las áreas que componen el territorio. David Garlan y Dewayne Perry, en su introducción al volumen de abril de 1995 de IEEE Transactions on Software Engineering dedicado a la AS, en el cual se delinean las áreas de investigación más promisorias, enumeran las siguientes:

- Lenguajes de descripción de arquitecturas
- Fundamentos formales de la AS (bases matemáticas, caracterizaciones formales de propiedades extra-funcionales tales como mantenibilidad, teorías de la interconexión, etcétera).
- Técnicas de análisis arquitectónicas
- Métodos de desarrollo basados en arquitectura
- Recuperación y reutilización de arquitectura
- Codificación y guía arquitectónica
- Herramientas y ambientes de diseño arquitectónico
- Estudios de casos

Fundamental en la concepción de Clements y Northrop es el criterio de reusabilidad como uno de los aspectos que más hacen por justificar la disciplina misma. Según estos autores, el estudio actual de la AS puede ser visto como un esfuerzo a posteriori para proporcionar un almacén estructurado de este tipo de información reutilizable de diseño de alto nivel propio de una familia. De tal manera, las decisiones de alto nivel inherentes a cada miembro de una familia de programas no necesitan ser re-inventadas, re-validadas y re-descriptas. Un razonamiento arquitectónico es además un argumento sobre las cuestiones estructurales de un sistema. Se diría también que el concepto de estilo es la encarnación principal del principio de reusabilidad en el plano arquitectónico.

Paul Clements define cinco temas fundamentales en torno de los cuales se agrupa la disciplina:

- **Diseño o selección de la arquitectura:** Cómo crear o seleccionar una arquitectura en base de requerimientos funcionales, de rendimiento o de calidad.
- **Representación de la arquitectura:** Cómo comunicar una arquitectura. Este problema se ha manifestado como el problema de la representación de arquitecturas utilizando recursos lingüísticos, pero el problema también incluye la selección del conjunto de información a ser comunicada.
- **Evaluación y análisis de la arquitectura:** Cómo analizar una arquitectura para predecir cualidades del sistema en que se manifiesta. Un problema semejante es cómo comparar y escoger entre diversas arquitecturas en competencia.
- **Desarrollo y evolución basados en arquitectura:** Cómo construir y mantener un sistema dada una representación de la cual se cree que es la arquitectura que resolverá el problema correspondiente.
- **Recuperación de la arquitectura:** Cómo hacer que un sistema legacy evolucione cuando los cambios afectan su estructura; para los sistemas de los que se carezca de documentación confiable, esto involucra primero una "arqueología arquitectónica" que extraiga su arquitectura.

Mary Shaw considera que en el 2001 los campos más promisorios de la AS siguen teniendo que ver con el tratamiento sistemático de los estilos, el desarrollo de lenguajes de descripción arquitectónica, la formulación de metodologías y (ahora) el trabajo con patrones de diseño. Se requieren todavía modelos precisos que permitan razonar sobre las propiedades de una arquitectura y verificar su consistencia y completitud, así como la automatización del proceso de análisis, diseño y síntesis. Sugiere que debe aprenderse una lección a partir de la experiencia de la ingeniería de software, la cual no obstante haberse desenvuelto durante treinta años no ha logrado plasmar un conjunto de paradigmas de investigación comparable al de otras áreas de las ciencias de la computación. Estima que la AS se encuentra ya en su fase de desarrollo y extensión, pero que tanto las ideas como las herramientas distan de estar maduras.

Un campo que no figura en estas listas pero sobre el cual se está trabajando intensamente es en el de la coordinación de los ADLs que sobrevivan con UML 2.0 por un lado y con XML por el otro. Ningún lenguaje de descripción arquitectónica en el futuro próximo (excepto los que tengan un nicho técnico muy particular) será viable si no satisface esos dos requisitos.

Los ejercicios que pueden hacerse para precisar los campos de la AS son incontables. Ahora que la AS se ha abismado en el desarrollo de metodologías, hace falta, por ejemplo, establecer con más claridad en qué difieren sus elaboraciones en torno del diseño, del análisis de requerimientos o de justificación económica de las llevadas a cabo por la ingeniería de software. Asimismo, se está esperando todavía una lista sistemática y exhaustiva que describa los dominios de incumbencia de la disciplina, así como un examen del riesgo de duplicación de esfuerzos entre campos disciplinarios mal comunicados, una situación que a primera vista parecería contradictoria con el principio de reusabilidad.

### 3. MODALIDADES Y TENDENCIAS.

En la década de 1990 se establece definitivamente la AS como un dominio todavía hoy separado de manera confusa de ese marco global que es la ingeniería y de esa práctica puntual que es el diseño. Aunque no hay un discurso explícito y autoconsciente sobre escuelas de AS, ni se ha publicado un estudio reconocido y sistemático que analice las particularidades de cada una, en la actualidad se pueden distinguir a grandes rasgos unas seis corrientes. Algunas distinciones están implícitas por ejemplo en, pero la bibliografía sobre corrientes y alternativas prácticamente no existe y la que sigue habrá de ser por un tiempo una de las pocas propuestas contemporáneas sobre el particular.

Ahora bien, articular una taxonomía de estrategias no admite una solución simple y determinista. En distintos momentos de su trayectoria, algunos practicantes de la AS se mueven ocasionalmente de una táctica a otra, o evolucionan de un punto de vista más genérico a otro más particular, o realizan diferentes trabajos operando en marcos distintos.

Además, con la excepción del “gran debate metodológico” entre métodos pesados y ligeros, las discusiones entre las distintas posturas rara vez se han manifestado como choques frontales entre ideologías irreconciliables, por lo que a menudo hay que leer entre líneas para darse cuenta que una afirmación cualquiera es una crítica a otra manera de ver las cosas. Fuera de la metodología, el único factor reconocible de discordia ha sido, hasta la fecha, la preminencia de UML y el diseño orientado a objetos. Todo lo demás parece ser más o menos negociable.

La división preliminar de escuelas de AS que se propone es la siguiente:

**1) Arquitectura como etapa de ingeniería y diseño orientada a objetos.** Es el modelo de James Rumbaugh, Ivar Jacobson, Grady Booch, Craig Larman y otros, ligado estrechamente al mundo de UML y Rational. No cabe duda que se trata de una corriente específica: Rumbaugh, Jacobson y Booch han sido llamados “Los Tres Amigos”; de lo que sí puede dudarse es que se trate de una postura arquitectónica. En esta postura, la arquitectura se restringe a las fases iniciales y preliminares del proceso y concierne a los niveles más elevados de abstracción, pero no está sistemáticamente ligada al requerimiento que viene antes o a la composición del diseño que viene después. Lo que sigue al momento arquitectónico se obvia, y a cualquier configuración, topología o morfología de las piezas del sistema se la llama arquitectura.

En esta escuela, si bien se reconoce el valor primordial de la abstracción (nadie después de Dijkstra osaría oponerse a ello) y del ocultamiento de información promovido por Parnas, estos conceptos tienen que ver más con el encapsulamiento en clases y objetos que con la visión de conjunto arquitectónica. Para este movimiento, la arquitectura se confunde también con el modelado y el diseño, los cuales constituyen los conceptos dominantes. En esta corriente se manifiesta predilección por un modelado denso y una profusión de diagramas, tendiente al modelo metodológico CMM o a UPM; no hay, empero, una prescripción formal. Importa más la abundancia y el detalle de diagramas y técnicas disponibles que la simplicidad de la visión de conjunto.

Cuando aquí se habla de estilos, se los confunde con patrones arquitectónicos o de diseño. Jamás se hace referencia a los lenguajes de descripción arquitectónica, que representan uno de los recursos valiosos reconocidos de la AS; sucede como si la disponibilidad de un lenguaje unificado de modelado los tornara superfluos.

La definición de arquitectura que se promueve en esta corriente tiene que ver con aspectos formales a la hora del desarrollo; esta arquitectura es isomorfa a la estructura de las piezas de código. Una definición típica y demostrativa sería la de Grady Booch; para él, la AS es “la estructura lógica y física de un sistema, forjada por todas las decisiones estratégicas y tácticas que se aplican durante el desarrollo”. Otras definiciones revelan que la AS, en esta perspectiva, concierne a decisiones sobre organización, selección de elementos estructurales, comportamiento, composición y estilo arquitectónico susceptibles de ser descriptas a través de las cinco vistas clásicas del modelo 4+1 de Kruchten.

**2) Arquitectura estructural, basada en un modelo estático de estilos, ADLs y vistas.** Constituye la corriente fundacional y clásica de la disciplina. Los representantes de esta corriente son todos académicos, mayormente de la Universidad Carnegie Mellon en Pittsburgh: Mary Shaw, Paul Clements, David Garlan, Robert Allen, Gregory Abowd, John Ockerbloom.

Se trata también de la visión de la AS dominante en la academia, y aunque es la que ha hecho el esfuerzo más importante por el reconocimiento de la AS como disciplina, sus categorías y herramientas son todavía mal conocidas en la práctica industrial. En principio se pueden reconocer tres modalidades en cuanto a la formalización; los más informales utilizan descripciones verbales o diagramas de cajas, los de talante intermedio se sirven de ADLs y los más exigentes usan lenguajes formales de especificación como CHAM y Z.

En toda la corriente, el diseño arquitectónico no sólo es el de más alto nivel de abstracción, sino que además no tiene por qué coincidir con la configuración explícita de las aplicaciones; rara vez se encontrarán referencias a lenguajes de programación o piezas de código, y en general nadie habla de clases o de objetos. Mientras algunos participantes excluyen el modelo de datos de las incumbencias de la AS (Shaw, Garlan, etc), otros insisten en su relevancia (Medvidovic, Taylor). Todo estructuralismo es estático; hasta fines del siglo XX, no está muy claro el tema de la posición del modelado arquitectónico en el ciclo de vida.

**3) Estructuralismo arquitectónico radical.** Se trata de un desprendimiento de la corriente anterior, mayoritariamente europeo, que asume una actitud más confrontativa con el mundo UML. En el seno de este movimiento hay al menos dos tendencias, una que excluye de plano la relevancia del modelado orientado a objetos para la AS y otra que procura definir nuevos metamodelos y estereotipos de UML como correctivos de la situación.

**4) Arquitectura basada en patrones.** Si bien reconoce la importancia de un modelo emanado históricamente del diseño orientado a objetos, esta corriente surgida hacia 1996 no se encuentra tan rígidamente vinculada a UML en el modelado, ni a CMM en la metodología. El texto sobre patrones que esta variante reconoce como referencia es la serie POSA de Buschmann y secundariamente el texto de la Banda de los Cuatro.

La diferencia entre ambos textos sagrados de la comunidad de patrones no es menor; en el primero, la expresión "Software Architecture" figura en el mismo título; el segundo se llama Design Patterns: Elements of reusable Object-Oriented software y su tratamiento de la arquitectura es mínimo. En esta manifestación de la AS prevalece cierta tolerancia hacia modelos de proceso tácticos, no tan macroscópicos, y eventualmente se expresa cierta simpatía por las ideas de Martin Fowler y las premisas de la programación extrema. El diseño consiste en identificar y articular patrones preexistentes, que se definen en forma parecida a los estilos de arquitectura.

**5) Arquitectura procesual.** Desde comienzos del siglo XXI, con centro en el SEI y con participación de algunos (no todos) los arquitectos de Carnegie Mellon de la primera generación y muchos nombres nuevos de la segunda: Rick Kazman, Len Bass, Paul Clements, Felix Bachmann, Fabio Peruzzi, Jeromy Carrière, Mario Barbacci, Charles Weinstock. Intenta establecer modelos de ciclo de vida y técnicas de elicitación de requerimientos, brainstorming, diseño, análisis, selección de alternativas, validación, comparación, estimación de calidad y justificación económica específicas para la arquitectura de software.

**6) Arquitectura basada en escenarios.** Es la corriente más nueva. Se trata de un movimiento predominantemente europeo, con centro en Holanda. Recupera el nexo de la AS con los requerimientos y la funcionalidad del sistema, ocasionalmente borroso en la arquitectura estructural clásica.

Los teóricos y practicantes de esta modalidad de arquitectura se inscriben dentro del canon delineado por la arquitectura procesual, respecto de la cual el movimiento constituye una especialización. En esta corriente suele utilizarse diagramas de casos de uso UML como herramienta informal u ocasional, dado que los casos de uso son uno de los escenarios posibles. Los casos de uso no están orientados a objeto.

En todos los simposios han habido intentos de fundación de otras variedades de AS, tales como una arquitectura adaptativa inspirada en ideas de la programación genética o en teorías de la complejidad, la auto-organización, el caos y los fractales, una arquitectura centrada en la acción que recurre a la inteligencia artificial heideggeriana o al postmodernismo, y una arquitectura epistemológicamente reflexiva que tiene a Popper o a Kuhn entre sus referentes; consideramos preferible omitirlas, porque por el momento ni su masa crítica es notoria ni su mensaje parece sustancial. Pero hay al menos un movimiento cismático digno de tomarse más en serio; a esta altura de los acontecimientos es muy posible que pueda hablarse también de una anti-arquitectura, que en nombre de los métodos heterodoxos se opone tanto al modelado orientado a objetos y los métodos sobre documentados impulsados por consultoras corporativas como a la abstracción arquitectónica que viene de la academia.

El Manifiesto por la Programación Ágil, en efecto, valoriza:

- Los individuos y las interacciones por encima de los procesos y las herramientas
- El software que funciona por encima de la documentación exhaustiva
- La colaboración con el cliente por encima de la negociación contractual
- La respuesta al cambio por encima del seguimiento de un plan

Habrà oportunidad de desarrollar el tratamiento de estas metodologías en otros documentos de la serie.

Hay muchas formas, por cierto, de organizar el panorama de las tendencias y las escuelas. Discernir la naturaleza de cada una puede tener efectos prácticos a la hora de comprender antagonismos, concordancias, sesgos, disyuntivas, incompatibilidades, fuentes de recursos. Nadie ha analizado mejor los límites de una herramienta como UML o de los métodos ultra-formales, por ejemplo, que los que se oponen a su uso por razones teóricas precisas, por más que éstas sean interesadas. Un buen ejercicio sería aplicar vistas y perspectivas diferentes a las que han regido esta clasificación informal, y proponer en función de otros criterios o de métodos más precisos de composición y referencias cruzadas, taxonomías alternativas de las líneas de pensamiento vigentes en la AS en la actualidad.

#### **4. DIFERENCIAS ENTRE ARQUITECTURA Y DISEÑO.**

Una vez que se reconoce la diferencia, que nunca debió ser menos que obvia, entre diseño e implementación, o entre vistas conceptuales y vistas tecnológicas ¿Es la AS solamente otra palabra para designar el diseño? Como suele suceder, no hay una sola respuesta, y las que hay no son taxativas. La comunidad de AS, en particular la de extracción académica, sostiene que ésta difiere sustancialmente del mero diseño. Pero Taylor y Medvidovic, por ejemplo, señalan que la literatura actual mantiene en un estado ambiguo la relación entre ambos campos, albergando diferentes interpretaciones y posturas:

- (1) Una postura afirma que arquitectura y diseño son lo mismo.
- (2) Otra, en cambio, alega que la arquitectura se encuentra en un nivel de abstracción por encima del diseño, o es simplemente otro paso (un artefacto) en el proceso de desarrollo de software.
- (3) Una tercera establece que la arquitectura es algo nuevo y en alguna medida diferente del diseño (pero de qué manera y en qué medida se dejan sin especificar).

Taylor y Medvidovic estiman que la segunda interpretación es la que se encuentra más cerca de la verdad. En alguna medida, la arquitectura y el diseño sirven al mismo propósito. Sin embargo, el foco de la AS en la estructura del sistema y en las interconexiones la distingue del diseño de software tradicional, tales como el diseño orientado a objetos, que se concentra más en el modelado de abstracciones de más bajo nivel, tales como algoritmos y tipos de datos. A medida que la arquitectura de alto nivel se refina, sus conectores pueden perder prominencia, distribuyéndose a través de los elementos arquitectónicos de más bajo nivel, resultando en la transformación de la arquitectura en diseño.

En su reciente libro sobre el arte de la AS, Stephen Albin se pregunta en qué difiere ella de las metodologías de diseño bien conocidas como la orientación a objetos. La AS, se contesta, es una metáfora relativamente nueva en materia de diseño de software y en realidad abarca también las metodologías de diseño, así como metodologías de análisis. El arquitecto de software contemporáneo, escribe Albin, ejecuta una combinación de roles como los de analista de sistemas, diseñador de sistemas e ingeniero de software.

Pero la arquitectura es más que una recolocación de funciones. Esas funciones pueden seguir siendo ejecutadas por otros, pero ahora caen comúnmente bajo la orquestación del chief architect. El concepto de arquitectura intenta subsumir las actividades de análisis y diseño en un framework de diseño más amplio y más coherente. Las organizaciones se están dando cuenta que el alto costo del desarrollo de software requiere ser sometido a algún control y que muchas de las ventajas prometidas por las metodologías aún no se han materializado. Pero la arquitectura es algo más integrado que la suma del análisis por un lado y el diseño por el otro. La integración de metodologías y modelos, concluye Albin, es lo que distingue la AS de la simple yuxtaposición de técnicas de análisis y de diseño.

Para Shaw y Garlan la AS es el primer paso en la producción de un diseño de software, en una secuencia que distingue tres pasos:

- (1) **Arquitectura.** Asocia las capacidades del sistema especificadas en el requerimiento con los componentes del sistema que habrán de implementarla. La descripción arquitectónica incluye componentes y conectores (en términos de estilos) y la definición de operadores que crean sistemas a partir de subsistemas o, en otros términos, componen estilos complejos a partir de estilos simples.
- (2) **Diseño del código.** Comprende algoritmos y estructuras de datos; los componentes son aquí primitivas del lenguaje de programación, tales como números, caracteres, punteros e hilos de control. También hay operadores primitivos.
- (3) **Diseño ejecutable.** Remite al diseño de código a un nivel de detalle todavía más bajo y trata cuestiones tales como la asignación de memoria, los formatos de datos, etcétera.

En opinión de Clements el diseño basado en arquitectura representa un paradigma de desarrollo que difiere de maneras fundamentales de las alternativas conocidas actualmente. En muchos sentidos, es diferente del diseño orientado a objetos (OOD) en la misma medida en que éste difería de sus predecesores. La AS deberá nutrir una comunidad de practicantes estableciendo una cultura en la que las ideas arquitectónicas puedan florecer. Una cuestión técnica que deberá abordarse es la creación de una articulación precisa de un paradigma de diseño basado en arquitectura (o posiblemente más de uno) y las cuestiones de proceso asociadas.

En una presentación de 1997, Dewayne Perry, uno de los fundadores de la disciplina, bosquejó la diferencia entre arquitectura y diseño. La arquitectura, una vez más (todo el mundo insiste en ello) concierne a un nivel de abstracción más elevado; se ocupa de componentes y no de procedimientos; de las interacciones entre esos componentes y no de las interfaces; de las restricciones a ejercer sobre los componentes y las interacciones y no de los algoritmos, los procedimientos y los tipos. En cuanto a la composición, la de la arquitectura es de grano grueso, la del diseño es de fina composición procedural; las interacciones entre componentes en arquitectura tienen que ver con un protocolo de alto nivel (en el sentido no técnico de la palabra), mientras que las del diseño conciernen a interacciones de tipo procedural (rpc, mensajes, llamadas a rutinas).

En los primeros años del nuevo siglo, la AS precisó la naturaleza del proceso de diseño como metodología en diversos modelos de diseño basados en arquitectura o ABD. Esta metodología considera que el diseño arquitectónico es el de más elevado nivel de abstracción, pero debe hacer frente al hecho de un requerimiento todavía difuso y al hecho de que, en ese plano, las decisiones que se tomen serán las más críticas y las más difíciles de modificar. Fundándose en el concepto de arquitectura conceptual de Hofmeister, Nord y Soni y en un modelos de vistas similar al 4+1 o a las vistas del modelo arquitectónico de Microsoft, el ABD describe el sistema en función de los principales elementos y las relaciones entre ellos.

El proceso se basa en tres fundamentos:

- (1) la descomposición de la función (usando técnicas bien establecidas de acoplamiento y cohesión),
- (2) la realización de los requerimientos de calidad y negocios a través de los estilos arquitectónicos, y
- (3) las plantillas de software, un concepto nuevo que incluye patrones que describen la forma en que todos los elementos de un tipo interactúan con los servicios compartidos y la infraestructura.

El modelo de diseño específico de ABD se describe en los documentos correspondientes a metodologías arquitectónicas.

Seguramente el lector encontrará mucho que agregar al planteamiento de las similitudes y diferencias entre arquitectura y diseño, más allá del hecho obvio de que el diseño arquitectónico se distingue del diseño del código, que viene determinado desde mucho antes y que es mucho más crítico para el éxito o el fracaso de una solución.

----- **FIN DEL DOCUMENTO**