

# Documentando la arquitectura de software

## Principios básicos

*por Omar Gómez*

En la actualidad, uno de los temas candentes que se habla dentro de la comunidad de desarrollo de software es el referente a las arquitecturas. La literatura nos dice que una arquitectura de software describe cómo un sistema es descompuesto en componentes, cómo éstos son interconectados, y la manera en que éstos se comunican e interactúan entre sí. Tras la anterior definición me surgieron un par de preguntas, ¿En qué grado cumplimos con ésta definición durante el rol que desempeñamos como arquitecto o diseñador?, O mejor dicho ¿Qué tan bien documentamos una arquitectura de software?.

Existen varias alternativas para documentar una arquitectura de software. La finalidad de este artículo es contar con un punto de referencia sobre ésta práctica. En el presente artículo se aborda dos de los enfoques más relevantes que han sido usados para documentar la arquitectura, a la vez se describe la tendencia actual con respecto a esta práctica, por último se describe una serie de consideraciones que se deben tener en cuenta al momento de documentar.

Comúnmente una arquitectura de software se documenta a través de un conjunto de vistas, en donde cada vista representa un aspecto o comportamiento particular del sistema. Dos de los artículos de mayor relevancia que abordan el tema del uso de vistas son el conocido, “Modelo de 4+1 vistas de la arquitectura de software” de Philippe B. Kruchten [1], y el de Robert L. Nord y compañía [2] titulado, “La arquitectura de software en aplicaciones industriales”. Ambos artículos fueron publicados en el año de 1995; El primer artículo es el más conocido, quizás esto se deba a que la propuesta de Kruchten es parte fundamental de la metodología del Proceso Unificado, que en la actualidad es una de las metodologías que goza de cierto grado de popularidad. A continuación se describe brevemente los trabajos de ambos autores.

Kruchten propone el uso de cinco vistas:

**Vista lógica.** Apoya principalmente los requisitos funcionales, lo que el sistema debe brindar en términos de servicios a sus usuarios. El sistema se descompone en una serie de abstracciones primarias, tomadas principalmente del dominio del problema en la forma de objetos o clases de objetos. Aquí se aplican los principios de abstracción, encapsulación y herencia. Esta descomposición no sólo se hace para potenciar el análisis funcional, sino también sirve para identificar mecanismos y elementos de diseño comunes a diversas partes del sistema.

**Vista de procesos.** Se tratan los aspectos de concurrencia y distribución, integridad del sistema, y tolerancia a fallos. También en ésta, se especifica en cuál hilo de control se ejecuta efectivamente una operación de una clase identificada en la vista lógica. Esta vista puede ser descrita como un conjunto de redes lógicas de procesos que son ejecutados de forma independiente, y distribuidos a lo largo de varios recursos de hardware conectados mediante un bus o a una red de datos.

**Vista de desarrollo.** Se centra en la organización real de los módulos de software en el ambiente de desarrollo. El software se empaqueta en partes pequeñas que pueden ser bibliotecas o subsistemas que son desarrollados por uno o un grupo de desarrolladores. Los subsistemas se organizan en una jerarquía de capas, cada una brinda una interfaz estrecha y bien definida hacia las capas superiores.

**Vista física.** Se toma en cuenta los requisitos no funcionales del sistema tales como, disponibilidad, confiabilidad, desempeño entre otras más. El sistema se ejecuta sobre varios nodos de procesamiento (hardware). Estos nodos son relacionados con los elementos identificados de las vistas anteriores. En esta vista se especifican varias configuraciones físicas. Por ejemplo, una para el desarrollo y las pruebas, o para el despliegue del sistema en plataformas distintas.

Kruchten define una última vista, en la que propone el uso de un pequeño subconjunto de escenarios que son instancias de casos de uso. La función de los escenarios es relacionar las cuatro vistas entre sí, de esta forma se cuenta con una perspectiva general del sistema, que ayuda a descubrir nuevos elementos o validar la arquitectura.

Por su parte Nord y compañía realizaron un estudio para conocer en una arquitectura, las estructuras que son de mayor importancia y el uso de éstas. El estudio se efectuó sobre varios sistemas de software de ámbito industrial tales como, sistemas de procesamiento de señales e imágenes, sistemas operativos en tiempo real, sistemas de comunicaciones, y sistemas de control e instrumentación. Tras el estudio realizado, Nord y colegas propusieron cuatro categorías o “vistas” para agrupar las estructuras principales de una arquitectura. Éstas son, vista conceptual, vista de interconexión de módulos, vista de ejecución, y vista de código. Dentro de cada una, se describe las estructuras principales del sistema desde una perspectiva en particular. A continuación se explica brevemente cada una de éstas.

**Vista conceptual.** Se describe el sistema en términos de sus elementos principales de diseño y las relaciones entre éstos, dentro de un dominio determinado. Esta vista es independiente de las decisiones de implementación y enfatiza en los protocolos de interacción entre los elementos de diseño.

**Vista de módulos.** Se captura la descomposición funcional y las capas del sistema. El sistema es descompuesto lógicamente en subsistemas, módulos, y unidades abstractas. Cada capa representa las distintas interfaces de comunicación permitidas entre los módulos.

**Vista de ejecución.** Se describe la estructura dinámica del sistema en términos de sus elementos en tiempo de ejecución. Por ejemplo, se modela las tareas operativas del sistema, procesos, mecanismos de comunicación y asignación de recursos. Algunos de los aspectos que se consideran en esta vista son, el desempeño y el entorno de ejecución.

**Vista de código.** Se organiza el código fuente en directorios, archivos y bibliotecas. Algunos de los aspectos que se incluyen son, los lenguajes de programación a utilizar, herramientas de desarrollo, la administración de la configuración y, la estructura y organización del proyecto.

Es común que en la actualidad se utilice alguno de los dos enfoques antes descritos. Sin embargo, la forma de documentar una arquitectura ha evolucionado significativamente. Hoy en día la tendencia sobre esta práctica se centra en dos aspectos principales:

- Los arquitectos deben documentar las vistas que sean de mayor utilidad y no ajustarse a un número fijo de vistas, como es el caso en las propuestas de Kruchten y Nord.
- Documentar la arquitectura tomando en cuenta los intereses y necesidades de las personas involucradas en el proyecto, estos intereses se traducen como las cualidades que el sistema resultante debe poseer.

Esta nueva tendencia está respaldada por dos grandes institutos, uno de ellos es el Instituto de Ingeniería del Software (SEI) con su propuesta, “Vistas y más allá de éstas, enfoque para la documentación de arquitecturas de software” [3] y el otro es el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) con el estándar, “IEEE 1471-2000, Prácticas recomendadas para la descripción arquitectónica de sistemas de software de gran demanda”, [4] elaborado por el comité de estándares del IEEE Software. A continuación se describe los dos enfoques.

El SEI, en su propuesta define tres categorías denominadas “tipos de vista” en las que prácticamente cualquier vista, dependiendo del tipo de información que contenga puede pertenecer a una de estas categorías. Los tipos de vista pueden ser:

- **Vista de módulo.** Describe cómo el sistema es estructurado en un conjunto de unidades de código.
- **Vista de conectores y componentes.** Describe cómo el sistema es estructurado en un conjunto de elementos que están en tiempo de ejecución así como la interacción entre éstos.
- **Vista de asignación.** Se describe la relación entre las unidades de software y los elementos del entorno tales como, hardware, sistemas de archivos o la organización de los equipos de desarrollo de software.

Es importante señalar que cada tipo de vista viene acompañado de un conjunto predefinido de estilos, de esta forma los arquitectos pueden hacer uso de éstos para documentar las vistas. De acuerdo a Shaw y Garlan [5], un estilo arquitectónico es una descripción de los elementos, conectores, topología, y un conjunto de restricciones sobre la interacción de los elementos. El uso de estilos beneficia el cumplimiento de las cualidades no funcionales de un sistema; en otras palabras, promueve la satisfacción de los intereses definidos por parte del personal involucrado en el proyecto.

El SEI recomienda se cuente con una guía de estilos que contenga entre otros aspectos, la descripción relevante del estilo, elementos, relaciones, propiedades, situaciones en las que no es recomendable aplicarlo, circunstancias en las que se recomienda usar el estilo, y posibles enfoques analíticos que el arquitecto puede utilizar. Para la elaboración de la guía de estilos se puede tomar como referencia el informe técnico realizado por Mark Klein y Rick Kazman [6] titulado, “Estilos Arquitectónicos Basados en Atributos”. En éste, los autores proponen un marco de trabajo para llevar a cabo un razonamiento cualitativo o cuantitativo de los atributos de calidad presentes en un estilo arquitectónico. En la parte final del informe, los autores presentan una serie de ejemplos que describen el uso del marco de trabajo con los atributos de calidad: desempeño, facilidad de modificación y disponibilidad. En caso de no encontrar en la literatura algún estilo que satisfaga las necesidades del arquitecto, es totalmente válido elaborar estilos propios y agregarlos a la guía.

La documentación de las vistas se realiza a través de lo que se denomina “paquetes de vista”. Una vista puede llegar a contener gran cantidad de elementos, ocasionando en el personal involucrado dificultad en su interpretación. La solución a esto es utilizar paquetes de vista, los cuales contienen un número reducido de elementos, logrando así una mejor comprensión ya que solo se muestra un fragmento particular del sistema. De esta manera, una vista se descompone en uno o más paquetes de vista.

Para seleccionar las vistas a documentar, se sigue un procedimiento basado con respecto a las estructuras que se encuentran presentes de manera inherente en el sistema a construir, y en los intereses primarios del personal involucrado. El procedimiento consta de los siguientes pasos:

**1) Elaborar una lista de vistas candidatas.** En este paso se elabora una tabla con la siguiente información, en las columnas se enumera el conjunto de posibles vistas a documentar, mientras que en las filas se enumera el personal involucrado. Posteriormente en cada una de las celdas se especifica el grado de información que requiere cada una de las personas involucradas en el proyecto. Los valores posibles para las celdas pueden ser: requerido a detalle, de manera general, o ninguno. Este paso concluye una vez que se han seleccionado las vistas de mayor interés por parte de las personas involucradas.

**2) Combinar las vistas.** Puede que las vistas elegidas en el paso anterior sean imprácticas de documentar debido al número de vistas seleccionadas, en este paso se reduce la lista de vistas de una manera que pueda ser manejable por el arquitecto. La reducción se lleva a cabo combinando varias vistas, de este modo una vista combinada muestra la información nativa de dos o más vistas separadas.

**3) Priorizar las vistas.** En este paso, el arquitecto debe tener el conjunto mínimo de vistas que satisfacen los intereses del personal involucrado. Después, en conjunto con el administrador del proyecto se procede a priorizar cada una de las vistas resultantes.

Una vez que las vistas se han seleccionado y priorizado, se inicia la documentación de éstas. El SEI cuenta con una plantilla [7] que se puede utilizar de referencia para este propósito. De acuerdo al SEI, la documentación de una arquitectura debe contener los siguientes apartados:

- **Presentación primaria.** Muestra los elementos y sus relaciones entre sí, usualmente se representa de manera gráfica.
- **Catálogo de elementos.** Contiene los detalles de éstos, sus propiedades e interfaces.
- **Diagrama de contexto.** Muestra la relación entre el sistema o porción de éste y su entorno.
- **Guía de variabilidad.** Muestra los posibles puntos de variación en caso de que las vistas sean modificadas.
- **Antecedentes de la arquitectura.** Explica la justificación de la arquitectura así como los supuestos, y los resultados de los análisis realizados.
- **Otra información.** En esta sección se incluyen prácticas y políticas de la organización.
- **Paquetes de vista relacionados.** Básicamente, en esta sección se definen las relaciones entre los distintos paquetes de vista.

Para concluir con la propuesta del SEI, en la figura 1 se muestra los principales conceptos de este enfoque.

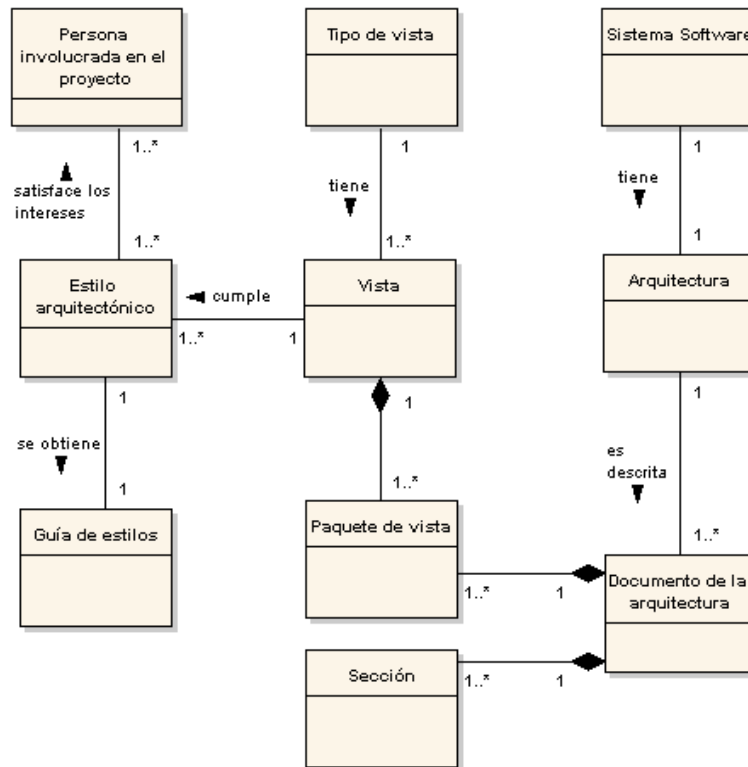


Figura 1. Relación de conceptos de la propuesta “vistas y más allá de éstas” del SEI.

Por su parte, el IEEE Software propone el estándar IEEE 1471 que define un conjunto de recomendaciones centradas básicamente en dos ideas, un marco conceptual para describir arquitecturas, y un conjunto de prácticas a seguir. Si algún arquitecto le interesa que la documentación de sus arquitecturas cumpla con el estándar IEEE 1471, éste debe seguir las prácticas que el estándar recomienda.

La descripción de la arquitectura se organiza en un conjunto de vistas. Cada vista modela una parte del sistema y satisface uno o más intereses de las personas involucradas. Los distintos intereses se deben considerar durante la construcción del sistema. Si alguno de éstos no es considerado en al menos una de las vistas se dice que la descripción de la arquitectura está incompleta.

Cada vista se documenta de acuerdo a un punto de vista determinado. En éste, se definen las notaciones, técnicas y reglas para construir e interpretar un vista. A su vez, éste determina cómo el contenido de una vista satisface uno o más intereses de las personas involucradas. Un punto de vista es un artefacto que puede ser reutilizado. En este marco conceptual, una vista es la instancia de un punto de vista dado. Un conjunto (biblioteca) de puntos de vista es análogo a la guía de estilos que propone el SEI. En la figura 2 se muestra un extracto del marco conceptual del estándar IEEE 1471.

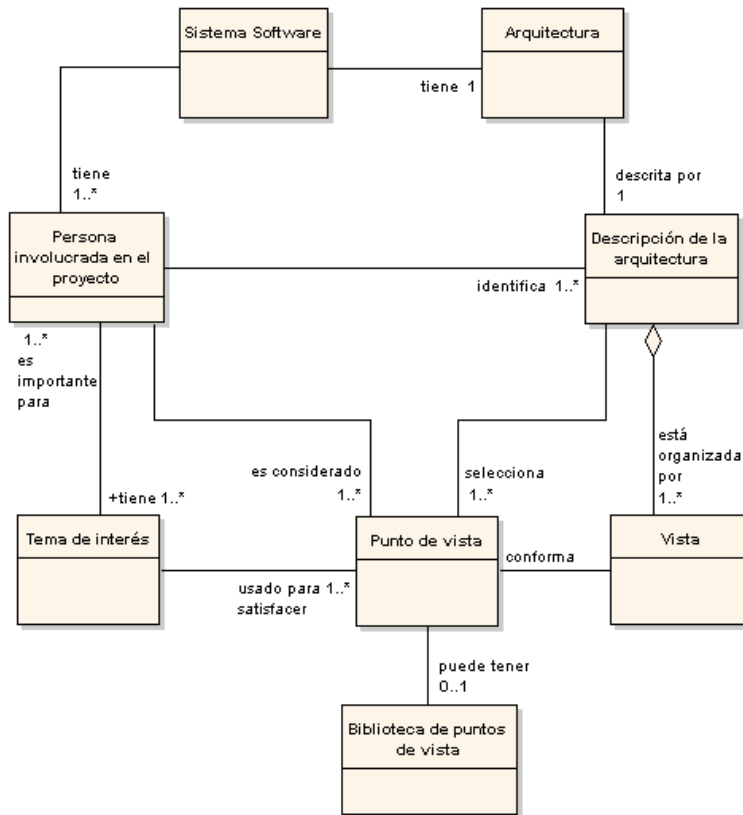


Figura 2. Extracto del marco conceptual del estándar IEEE 1471.

Si se desea que la documentación de la arquitectura cumpla con el estándar, se deben seguir las siguientes prácticas:

**Identificación e información general.** En ésta práctica se lleva lo relacionado al control de versiones del documento, fecha, histórico de revisiones, estado, contexto del sistema, declaración del alcance, entre otros más.

**Identificación del personal involucrado y sus intereses.** Se describen los diversos tipos de personas involucradas en el proyecto. Tales como, administradores, diseñadores, desarrolladores, usuarios, patrocinadores. A su vez, se incluye la diversidad de intereses que la arquitectura debe satisfacer.

**Puntos de vista.** Cada de uno de éstos debe contener un nombre, personal involucrado, intereses que satisface, lenguaje o técnicas de modelado utilizados durante la construcción de la vista, algún método analítico para analizar de manera cualitativa o cuantitativa los atributos de calidad que satisface el punto de vista, y la justificación de éste.

**Vistas.** Cada vista debe tener un identificador, una breve introducción y la representación del sistema con respecto a un punto de vista en particular.

**Consistencia entre vistas.** En está se registran las inconsistencias entre las vistas, así como algún tipo de procedimiento que indique la consistencia entre éstas.

**Justificación.** Se debe incluir la justificación del tipo de arquitectura seleccionada, y de los puntos de vista utilizados.

Actualmente, el IEEE e ISO están revisando en conjunto este estándar con el objetivo de tener una versión actualizada en el siguiente año para someterla a votación [8].

Estos dos últimos enfoques tienen varias similitudes entre sí. Por ejemplo, ambos se centran en las personas involucradas y en sus diversos intereses, ninguno de estos prescribe un número fijo de vistas, los dos enfoques hacen uso de la reutilización de artefactos; uno a través de la guía de estilos y el otro por medio de una biblioteca de puntos de vista, en ambos la arquitectura debe estar justificada y los dos enfoques recomiendan el uso de análisis cualitativos o cuantitativos.

Una de las diferencias entre los dos enfoques es la siguiente. En el enfoque del SEI primero se selecciona un conjunto posible de vistas con respecto a las estructuras que están presentes en el sistema a construir, después se validan las vistas de acuerdo a un consenso de intereses por parte del personal involucrado, posteriormente se procede a documentar las vistas. Por otra parte, el IEEE primero determina los intereses del personal involucrado, después en base a estos intereses se selecciona un conjunto de puntos de vista que los satisfaga, por último se documenta cada una de las vistas basándose en los puntos de vista seleccionados.

**Conclusiones.** Los enfoques presentados en este artículo pueden servir como punto de referencia para los arquitectos interesados en mejorar la documentación de sus arquitecturas. Uno de los principales beneficios de llevar a cabo ésta práctica es el poder efectuar evaluaciones sobre la arquitectura documentada [9], con el fin de determinar si se están cumpliendo o no los intereses del personal involucrado. Para finalizar, algunos de los puntos que se deben considerar al momento de documentar un arquitectura son:

- Documentar la arquitectura tomando en cuenta las necesidades e intereses de cada persona que forma parte del proyecto.
- Acompañar la documentación de las vistas con un modelo analítico que ayude a predecir el comportamiento de los atributos de calidad.
- UML no es el único lenguaje para documentar la arquitectura. Existen diferentes notaciones y lenguajes para este propósito. Por ejemplo, existen diversos lenguajes para descripción de arquitecturas (ADLs) que describen aspectos particulares de ésta.
- Mantener una relación consistente entre las vistas.
- Elaborar plantillas de estilos para promover la reutilización de artefactos dentro de la organización.
- Mantener actualizada la matriz de trazabilidad entre los requisitos y los elementos de la arquitectura.
- Tener bajo una línea base el documento de la arquitectura.

## Referencias

- [1] Philippe Kruchten. *"The 4+1 View Model of Architecture"*. IEEE Software., Los Alamitos, CA, USA, Volume 12, Number 6, Pages 42-50, IEEE Computer Society Press, 1995.
- [2] Dilip Soni; Robert L. Nord & Christine Hofmeister. *"Software architecture in industrial applications"*. ICSE '95: Proceedings of the 17th international conference on Software Engineering, New York, NY, USA, Pages 196-207, ACM, 1995.
- [3] Paul Clements; Felix Bachmann; Len Bass; David Garlan; James Ivers; Reed Little; Robert Nord & Judith Stafford. *"Documenting Software Architectures: Views and Beyond"*., Addison Wesley Professional, 2002.
- [4] IEEE Architecture Working Group. *"IEEE Standard 1471-2000, Recommended practice for Architectural Description of Software-Intensive Systems"*., Pages 1-23, IEEE, 2000.
- [5] Mary Shaw & David Garlan. *"Software Architecture: Perspectives on an Emerging Discipline"*., April, Prentice Hall, 1996.
- [6] Mark Klein; Rick Kazman & Robert Nord. *"Attribute-Based Architectural Styles"*., Software Engineering Institute, Technical Report CMU/SEI-99-TR-22, Carnegie Mellon University, 1999
- [7] Software Engineering Institute, SEI. *"Views and Beyond Architecture Documentation Template"*., <[http://www.sei.cmu.edu/architecture/arch\\_doc.html](http://www.sei.cmu.edu/architecture/arch_doc.html)>, Template 05, February 2006.
- [8] Rich Hilliard. *"ISO/IEC 42010/IEEE 1471: Architectural Description"*., <[http://www.sei.cmu.edu/architecture/IEEE\\_1471.html](http://www.sei.cmu.edu/architecture/IEEE_1471.html)>
- [9] Omar Gómez. *"Evaluando la Arquitectura de Software, métodos de evaluación"*., Revista Software Gurú, Año 03 No. 02., 2007.