**¿Qué es la arquitectura de software?**

Antes de elaborar sobre el tema, es conveniente definir el concepto ya que hoy en día el término de arquitectura se usa para referirse a varios aspectos relacionados con las TI. De acuerdo al Software Engineering Institute (SEI), la Arquitectura de Software se refiere a “las estructuras de un sistema, compuestas
de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.”[1]

El término “elementos” dentro de la definición del SEI es vago a propósito, pues puede referirse a distintas entidades relacionadas con el sistema. Los elementos pueden ser entidades que existen en tiempo de ejecución (objetos, hilos), entidades lógicas que existen en tiempo de desarrollo (clases, componentes) y entidades físicas (nodos, directorios). Por otro lado, las relaciones entre elementos dependen de propiedades visibles (o públicas) de los elementos, quedando ocultos los detalles de implementación. Finalmente, cada conjunto de elementos relacionados de un tipo particular corresponde a una estructura distinta, de ahí que la arquitectura esta compuesta por distintas estructuras.

**¿Por qué es importante la arquitectura de software?**

La arquitectura de software es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema. Ejemplos de atributos de calidad son el desempeño, que tiene que ver con el tiempo de respuesta del sistema a las peticiones que se le hacen, la usabilidad, que tiene que ver con qué tan sencillo les resulta a los usuarios realizar operaciones con el sistema, o bien la modificabilidad, que tiene que ver con qué tan simple resulta introducir cambios en el sistema. Los atributos de calidad son parte de los requerimientos (no funcionales) del sistema y son características que deben expresarse de forma cuantitativa. No tiene sentido, por ejemplo, decir que el sistema debe devolver una petición “de manera rápida”, o presentar una página “ligera”, ya que no es posible evaluar objetivamente si el sistema cubre o no esos requerimientos.

La manera en que se estructura un sistema permitirá o impedirá que se satisfagan los atributos de calidad. Por ejemplo, un sistema estructurado de tal manera que una petición deba transitar por muchos componentes antes de que se devuelva una respuesta podría tener un desempeño pobre. Por otro lado, un sistema estructurado de tal manera que los componentes estén altamente acoplados entre ellos limitará severamente la modificabilidad. Curiosamente, la estructuración tiene un impacto mucho menor respecto a los requerimientos funcionales del sistema. Por ejemplo, un sistema difícil de modificar puede satisfacer plenamente los requerimientos funcionales que se le imponen.

Además de los atributos de calidad, la arquitectura de software juega un papel fundamental para guiar el desarrollo. Una de las múltiples estructuras que la componen se enfoca en partir el sistema en componentes que serán desarrollados por individuos o grupos de individuos. La identificación de esta estructura de asignación de trabajo es esencial para apoyar las tareas de planeación del proyecto.

Finalmente, los diseños arquitectónicos que se crean en una organización pueden ser reutilizados para crear sistemas distintos. Esto permite reducir costos y aumentar la calidad, sobre todo si dichos diseños han resultado previamente en sistemas exitosos.

**El ciclo de desarrollo de la arquitectura**

Dentro de un proyecto de desarrollo, e independientemente de la metodología que se utilice, se puede hablar de “desarrollo de la arquitectura de software”. Este desarrollo, que precede a la construcción del sistema, esta dividido en las siguientes etapas: requerimientos, diseño, documentación y evaluación. Cabe señalar que las actividades relacionadas con el desarrollo de la arquitectura de software generalmente forman parte de las actividades definidas dentro de las metodologías de desarrollo.

A continuación se describen dichas etapas.

**• Requerimientos**. La etapa de requerimientos se enfoca en la captura, documentación y priorización de requerimientos que influencian la arquitectura. Como se mencionó anteriormente, los atributos de calidad juegan un papel preponderante dentro de estos requerimientos, así que esta etapa hace énfasis en ellos. Otros requerimientos, sin embargo, son también relevantes para la arquitectura, estos son los requerimientos funcionales primarios y las restricciones.

• **Diseño**. La etapa de diseño es la etapa central en relación con la arquitectura y probablemente la más compleja. Durante esta etapa se definen las estructuras que componen la arquitectura. La creación de estas estructuras se hace en base a patrones de diseño, tácticas de diseño y elecciones tecnológicas. El diseño que se realiza debe buscar ante todo satisfacer los requerimientos que influencian a la arquitectura, y no simplemente incorporar diversas tecnologías por que están “de moda”.

• **Documentación.** Una vez creado el diseño de la arquitectura, es necesario poder comunicarlo a otros involucrados dentro del desarrollo. La comunicación exitosa del diseño muchas veces depende de que dicho diseño sea documentado de forma apropiada. La documentación de una arquitectura involucra la representación de varias de sus estructuras que son representadas a través de distintas vistas. Una vista generalmente contiene un diagrama, además de información adicional, que apoya en la comprensión de dicho diagrama.

• **Evaluación.** Dado que la arquitectura de software juega un papel crítico en el desarrollo, es conveniente evaluar el diseño una vez que este ha sido documentado con el fin de identificar posibles problemas y riesgos. La ventaja de evaluar el diseño es que es una actividad que se puede realizar de manera temprana (aún antes de codificar), y que el costo de corrección de los defectos identificados a través de la evaluación es mucho menor al costo que tendría el corregir estos defectos una vez que el sistema ha sido construido.

**El rol de arquitecto**
Las actividades descritas anteriormente requieren de habilidades particulares que son la responsabilidad del arquitecto de software. El arquitecto es un líder técnico que debe conocer los principios relacionados con la arquitectura de software, tener un amplio conocimiento respecto a la tecnología, y tener excelentes habilidades de comunicación escrita y oral.

Desafortunadamente, en la actualidad pocos arquitectos de software que laboran en la industria han recibido una formación teórica respecto al tema. Esto se debe a que no es sino hasta épocas recientes que se han establecido de manera más formal los conceptos relacionados con la arquitectura de software, y que actualmente pocas instituciones ofrecen cursos enfocados en el tema. El desconocimiento de los principios relativos a la arquitectura de software frecuentemente impacta de manera negativa a los proyectos de desarrollo.

**Apenas empezamos**
A lo largo de las distintas entregas de esta columna que inicia se buscará dar una panorámica del tema de arquitectura de software y se discutirá de manera más detallada aspectos como:

•Requerimientos que influyen en la arquitectura,
•El diseño, documentación y evaluación de la arquitectura,
•Los retos relacionados con la introducción de prácticas de arquitectura de software en un contexto organizacional,
•La arquitectura de software dentro de las metodologías de desarrollo,
•El perfil del arquitecto de software.