

# LA ARQUITECTURA DE SOFTWARE

Desde su primera mención en un informe técnico en la década de 1970 titulado *Software Engineering Techniques*, diversos autores se propusieron definir el término de arquitectura de software. Por ese motivo, en vez de crear nuestra propia definición del término, haremos uso de las cuatro definiciones existentes a fin de resaltar sus diferentes características. Las tres primeras que usaremos son las definiciones del término. Estas fueron formuladas por autores que destacan en el área desde su introducción y son usadas actualmente por la gran mayoría de los profesores, alumnos y practicantes del área. Por otro lado, también mostraremos la definición de arquitectura de software. Esta es parte del estándar ISO/IEEE 1471-2000 y su creación fue motivada justamente para hacer que estudiantes, profesores y practicantes de arquitectura de software estén de acuerdo sobre el término.

## LA DEFINICIÓN DE ARQUITECTURA DE SOFTWARE DE PERRY Y WOLF

Perry y Wolf introdujeron su definición de arquitectura de software en su artículo seminal *Foundations and the Study of Software Architecture*. La definición que ellos proponen consiste en:

Arquitectura = {elementos, organización, decisiones}

De acuerdo con esa definición, la arquitectura de software es un conjunto de elementos arquitecturales que poseen alguna organización. Los elementos y su organización son definidos por decisiones tomadas para satisfacer objetivos y restricciones. Destacan tres tipos de elementos arquitecturales:

- Elementos que usan o transforman información;
- Elementos que contienen la información para ser usada y transformada;
- Elementos que conectan elementos de cualquier tipo entre sí.

Después la organización dicta las relaciones entre los elementos arquitecturales. Esas relaciones poseen propiedades y restringen como los elementos deben interactuar de forma que satisfaga los objetivos del sistema. Adicionalmente, esas relaciones deben ser ponderadas de modo que puedan indicar su importancia en el proceso de selección de alternativas.

**EJEMPLO:** Un elemento de datos muy presente en el software del Videoclub y en sistemas de información en general es la base de datos. Esta es la responsable de guardar y recuperar los datos en el sistema.

En el Videoclub, inicialmente, están presentes tres tipos de datos:

1. **Información textual:** informaciones de registros de alta de los usuarios e informaciones textuales sobre las películas;
2. **Imágenes:** imágenes que componen la identidad visual del sistema, foto del usuario presente en su perfil e imágenes de publicidad de las películas;
3. **Vídeos:** películas completas, trailers y documentación de “making off” disponibles para streaming.

Por eso, consideramos un elemento de datos para cada tipo. Así, tenemos la base de datos responsable de las informaciones textuales, la base de datos responsable de las imágenes y la base de datos responsable de los vídeos. Esa separación de responsabilidades permite que la implementación de cada tipo de datos disponga de servicios diferenciados o que obtenga algún provecho de la naturaleza de sus datos para atender a algún atributo de calidad (rendimiento, escalabilidad, etc.). De esa manera, el elemento responsable del texto puede ser optimizado para la búsqueda por palabras-clave, mientras el responsable de los vídeos puede ser optimizado para recuperar grandes flujos de datos por cada respuesta. Por otro lado, también tiene sentido dividir lógicamente los elementos de datos en: elementos de datos de usuarios y elementos de datos de películas. Note que esa división es transversal a la división en elementos de texto, imágenes y vídeos y, por lo tanto, el

elemento de datos de usuarios puede estar compuesto por un elemento de datos textuales y otro elemento de datos de imágenes, de la misma manera que el elemento de datos de películas puede contener el elemento de datos textuales, de imágenes y de vídeos.

Como ejemplo de elemento de procesamiento, citamos la lógica de negocio del Videoclub. Este contiene las reglas de negocio que componen el Videoclub. Note que todavía podemos dividir ese elemento de procesamiento en elementos más especializados: el elemento de procesamiento responsable de crear, editar, recuperar y eliminar usuarios, el responsable de crear, editar, recuperar y eliminar informaciones de películas, el responsable del alquiler de películas y el responsable de controlar la sesión de streaming, entre otros. Esa división, así como la división de los elementos de datos, puede ser realizada en favor de la atención a los atributos de calidad.

Sin embargo, un elemento no es capaz de crear, editar, recuperar o eliminar usuarios sin comunicarse con los datos de los usuarios. De la misma manera, el elemento responsable de manipular las informaciones de las películas debe comunicarse con los elementos que guardan los datos de las películas.

También, para controlar la sesión de streaming, el responsable debe obtener la película del elemento de datos que contiene las películas completas. Esa comunicación es realizada por los diversos elementos de conexión del Videoclub. Entre ellos el driver JDBC Java Database Connectivity, que permite la

comunicación con la base de datos responsable de los usuarios; con el protocolo FTP, para la transferencia de vídeos; con el protocolo HTTP, para transferencias a partir de la base de imágenes; o el REST Representational State Transfer, que es una especialización del HTTP y es usado para la comunicación entre elementos de procesamiento.

## ARQUITECTURA DE SOFTWARE POR GARLAN Y SHAW

Además de tener una visión más concreta sobre la arquitectura que Perry y Wolf, Garlan y Shaw son más explícitos cuando mencionan el propósito de aplicar conocimientos de arquitectura en un sistema de software. Para ellos, la arquitectura de software se hace necesaria cuando el tamaño y la complejidad de los sistemas de software crecen. Así, el problema de construir sistemas va más allá de la elección de los algoritmos y de las estructuras de datos correctos. Ese problema envolverá también las decisiones sobre las estructuras que formarán el sistema, la estructura global de control que será usada, los protocolos de comunicación, la sincronización y el acceso a datos, la atribución de la funcionalidad a los elementos del sistema y la distribución física de los elementos del sistema. Además de eso, el problema envolverá las decisiones que impactarán en el comportamiento del sistema en términos de escala y rendimiento, entre otros atributos de calidad.

La visión sobre arquitectura de software de Garlan y Shaw contiene tres aspectos. El primero que estos citan es que sean explícitos cuando debemos aplicar conocimientos de arquitectura de software, es decir, cuando trabajamos con grandes sistemas. El segundo aspecto que citan es ser claros en

la separación entre las tareas de diseño detallado y diseño arquitectural; el primero se preocupa de los algoritmos y de las estructuras de datos, mientras que el segundo se preocupa de los elementos y de la organización del sistema como un todo, estando relacionado con la estructura del sistema, el control, la comunicación o la implantación. Y el tercero que citan es que el proceso de diseño de la arquitectura necesita preocuparse de los atributos de calidad del sistema, en alcanzar la escalabilidad o en el rendimiento, por ejemplo.

**EJEMPLO:** En la arquitectura de un sistema operativo, para alcanzar atributos de rendimiento y de portabilidad, deberá preocuparse de los diversos aspectos que compondrán el sistema. Está claro que algunos algoritmos también serán responsables del rendimiento del Sistema Operativo en cuestión, como el responsable de la ordenación por prioridad de los procesos en ejecución o del alojamiento en memoria de un nuevo proceso; pero la organización del sistema en capas de abstracción (abstracción de hardware, sistema de archivos y drivers, gestión de procesos, API del sistema, bibliotecas y aplicaciones), la comunicación entre estas capas (una capa sólo puede comunicarse con la capa siguiente o aplicaciones y las bibliotecas sólo pueden comunicarse con la API del sistema, etc.) y la sincronización (una aplicación que sugiere archivar los datos pero será el sistema de archivos quién decidirá cuando sucederá esa acción) también impactan en su rendimiento.

Note que esa organización también impacta en la portabilidad: cuanto menos acoplado estén el resto de las capas fuera de la capa de abstracción de hardware, más fácil serán de realizar los

cambios necesarios para que el sistema operativo esté disponible para una nueva plataforma de hardware, teniendo sólo que volver a implementar esa capa.

## ARQUITECTURA DE SOFTWARE POR BASS ET AL

Como veremos a continuación, la definición de Bass et al es bastante similar a la que podemos ver en el estándar ISO/IEEE 1471-2000. Sin embargo, veremos que propiedades de los elementos arquitecturales deben ser consideradas:

*La arquitectura de un programa o de sistemas computacionales es la estructura o estructuras del sistema, la cual está compuesta de elementos de software, de propiedades externamente visibles de esos elementos y de las relaciones entre ellas.*

Como ya fue observado por Gorton, esa definición es explícita en cuanto al papel de la abstracción en la arquitectura (cuando habla de propiedades externamente visibles) y también en cuanto al papel de las múltiples visiones arquitecturales (estructuras del sistema). Debemos también mencionar el uso del término “elementos de software” como las piezas fundamentales de la arquitectura. En la edición anterior de esa definición, sus autores usaban “componentes de software” en vez de “elementos de software”.

## *LA ARQUITECTURA DE SOFTWARE POR EL ESTÁNDAR ISO/IEEE 1471-2000*

El propósito de la creación del estándar ISO/IEEE 1471-2000 fue la de ayudar en el consenso entre autores, estudiantes y profesionales sobre lo que es y para que sirve la arquitectura de software. Por ello, ese estándar no sólo define la arquitectura del software, sino que también introduce un esquema conceptual para la descripción arquitectural. Su definición de arquitectura de software, la cual nosotros adoptaremos a lo largo del libro, es la siguiente:

*La arquitectura es la organización fundamental de un sistema incorporada en sus componentes, sus relaciones con el entorno y los principios que conducen su diseño y evolución.*

Podemos darnos cuenta de que la definición de arriba es consistente con las anteriores ya que menciona que la arquitectura también comprende la estructura (o elementos o componentes), relaciones y decisiones (o principios). Sin embargo, esta además añade una preocupación por la arquitectura: conducir la evolución del software.

La evolución del software es el fenómeno de cambio/modificación que sucede en el software a lo largo de los años y de las múltiples versiones, desde su inicio hasta el completo abandono del sistema. Ese cambio no está sólo relacionado con la adición y eliminación de funcionalidades, sino que también está relacionada con el mantenimiento del

código a lo largo del ciclo de vida del software. Ese mantenimiento puede tanto mejorar como deteriorar los atributos externos de calidad del software, los cuales son percibidos por los usuarios (Ej., rendimiento, tolerancia a fallos, disponibilidad), en cuanto a los atributos internos de calidad del software, los cuales son percibidos por los participantes en el desarrollo (Ej., testeo, legibilidad, reusabilidad, etc....).

Uno de los principales objetivos en una arquitectura es la de alcanzar la calidad deseada por los interesados en el sistema, con ello se prueba la relevancia de la arquitectura en la conducción de la evolución del software, con la arquitectura de un proyecto consolidado, se podrán tomar decisiones que contribuirán para la preservación de la calidad del sistema durante su ciclo de vida.

Antes de entrar en detalles sobre los diversos aspectos de la arquitectura de software, debemos entrar en consenso sobre el término “componente de software”. En Ingeniería de Software, “componentes” tiene varios significados divergentes. Un significado, de acuerdo con el Standard Computer Dictionary, es que un componente es una de las partes que componen el sistema. De esa manera, “componente” puede ser sustituido por “módulo”, “unidad” o mismo “elemento” de software. Ese es el significado de “componente” usado en el estándar ISO/IEEE 1471-2000