

Capítulo 3: Atributos de calidad del software

Gran parte de la vida de un arquitecto es gastada en diseñar sistemas de software que cumplan con un conjunto de requerimientos de atributos de calidad.

Los requerimientos de atributos de calidad son parte de los requerimientos no funcionales de una aplicación, estos capturan muchas facetas de cómo los requerimientos funcionales de una aplicación son alcanzados.

Los requerimientos de atributos de calidad deben ser específicos en cuanto a como una aplicación debería alcanzar una necesidad dada.

Atributos de calidad más relevantes para aplicaciones de TI:

- Performance
- Escalabilidad
- Modificabilidad
- Seguridad
- Availability (Disponibilidad)
- Integración
- Otros:
 - o Portabilidad
 - o Testability (pruebas)
 - o Supportability (soporte)

Performance

Es una de las cualidades de una aplicación que puede, a menudo, ser fácilmente cuantificada y validada.

Un requerimiento de calidad de performance define una métrica que la cantidad de trabajo en una aplicación debe realizarse en un momento dado, y/o en un plazo, para su operación correcta.

El performance usualmente se manifiesta en las siguientes medidas:

- Throughput (rendimiento de proceso)
- Tiempo de respuesta
- Deadlines (Plazos)

Throughput (rendimiento de proceso)

Es la medida de la cantidad de trabajo que una aplicación debe realizar en una unidad de tiempo. El trabajo, típicamente, es medido en transacciones por segundo (tps), o mensajes procesados por segundo (mps). Tipos de rendimiento de proceso:

- Average throughput (rendimiento de procesamiento medio)
- Peak throughput (rendimiento de procesamiento máximo)

Response Time (tiempo de respuesta)

Es la medida de latencia que una aplicación exhibe al procesar transacciones de negocios. El tiempo de respuesta es casi siempre asociado con el tiempo que una aplicación tarda en responder a una entrada.

Plazos (deadlines)

Cualquier aplicación que tiene un tiempo de ventana limitado para completarse tendrá un requerimiento de funcionamiento de plazo (performance deadline requirement).

Escalabilidad

“Que tan bien trabajará una solución a un problema, cuando el tamaño del problema crece”

La escalabilidad se refiere a cómo un diseño puede hacer frente a algún aspecto de los requerimientos de una aplicación que están creciendo en tamaño.

¿Qué se espera que crezca?

- Request load (petición de carga)
- Simultaneous connections (conexiones simultaneas)
- Data size (tamaño de los datos)

Request load (Petición de carga)

En un mundo perfecto y sin capacidad adicional de hardware, conforme la carga se incrementa, el rendimiento de proceso de una aplicación debería mantenerse constante, y el tiempo de respuesta por petición debe incrementarse solo de forma lineal. Una **solución escalable** permitirá, entonces, que se despliegue capacidad de procesamiento adicional para incrementar el rendimiento de proceso y decrementar el tiempo de respuesta.

- Scale-up: La aplicación se replica en diferentes maquinas. Funciona si una aplicación es multi-hilo o se pueden ejecutar en ella, instancias de procesos multi-hilo en la misma máquina.
- Scale-out: Una instancia de la aplicación es ejecutada en una maquina-multiprocesamiento. Funciona si hay poco o nada de trabajo requerido al manejar la distribución de peticiones entre las múltiples maquinas.

La escalabilidad debe ser alcanzada sin hacer modificaciones a la arquitectura subyacente. En la realidad, mientras la carga se incrementa, las aplicaciones son expuestas a decrementos en el rendimiento de procesos y por esto mismo, crecimientos en los tiempos de respuesta.

Tamaño de los datos

El comportamiento de una aplicación al aumentar el procesamiento de datos. Una solución para esto puede ser diseñar buscar y recuperar datos de un repositorio de un tamaño específico.

Modificabilidad

Considerar cambios probables a la aplicación es una buena práctica durante la formulación de la arquitectura.

La modificabilidad está medida en que tan fácil puede ser cambiar una aplicación para abastecerla de requerimientos funcional o no funcional. Predecir modificabilidad, requiere una estimación del esfuerzo y/o costo para hacer un cambio.

Las medidas de modificabilidad son únicamente relevantes dentro del contexto de una solución arquitectónica dada. Esta solución debe ser expresada, al menos estructuralmente, como una colección de componentes, relaciones entre ellos y una descripción de cómo los componentes interactúan con el ambiente.

Seguridad

En el nivel de arquitectura, los requerimientos mas comunes son:

- Autenticación
- Autorización
- Encriptación
- Integridad
- No repudiación

Disponibilidad

Esta relacionada con la confiabilidad. Si una aplicación no esta disponible para su uso cuando se necesita, entonces no cumple con los requerimientos funcionales de la misma.

La replicación de componentes es una estrategia usada y probada para alta disponibilidad.

La recuperación esta relacionada de cerca con la disponibilidad. Una aplicación es recuperable si tiene la capacidad de reestablecer los niveles de funcionamiento requerido y recuperar los datos afectados en una falla de aplicación o sistema.

Integración

La integración se refiere a la facilidad con la que una aplicación puede fácilmente ser incorporada en un contexto más amplio de la aplicación.

La estrategia más amplia para proveer integración es a través de integración de los datos o proveyendo una API.

La integración envuelve almacenamiento de los datos que una aplicación manipula de manera que otras aplicaciones pueden acceder.

Con integración de los datos, las maneras en que los datos son usados por otras aplicaciones se salen bastante del control del dueño original de los datos. Esto puede ser causado porque la integridad de los datos y las reglas del negocio impuestas por la aplicación son evitadas. La alternativa es alcanzar interoperabilidad mediante un API.

Otros atributos de calidad:

- Portabilidad: una aplicación puede fácilmente ser ejecutada en diferentes plataformas de software/hardware.
- Testability: Qué tan fácil o difícil es probar una aplicación. Mientras más complejo sea un diseño, más difícil es probarla a fondo.
- Supportability: Es la medida de qué tan fácil puede una aplicación ser soportada luego de que ha sido implantada.