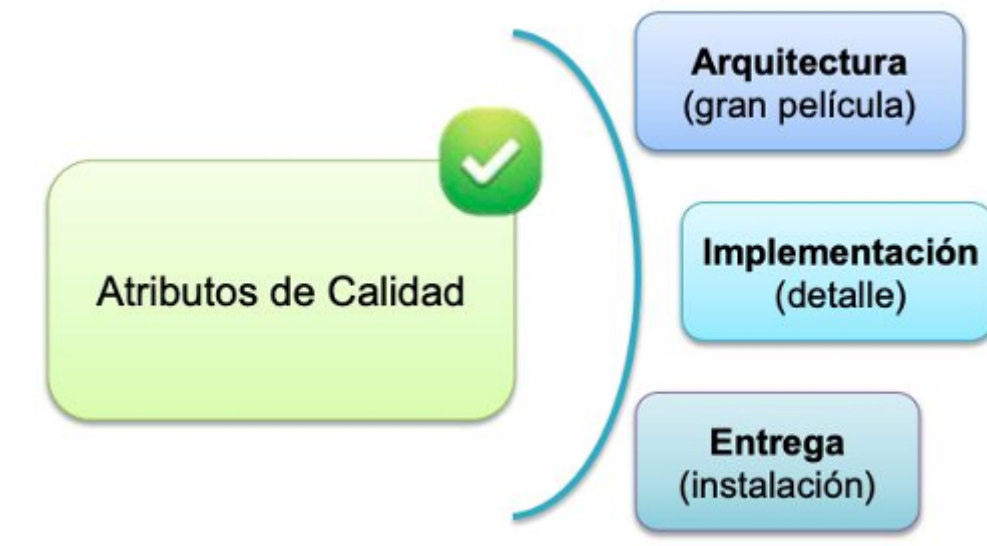


ATRIBUTOS DE CALIDAD

Un atributo de calidad es una medida o propiedad testable de un sistema que es usada para indicar cómo es el sistema satisface las necesidades de sus stakeholders.

Los atributos de calidad deben ser considerados a lo largo de todo el ciclo de vida del software para tener éxito. No dependen solamente de la etapa de diseño o de la implementación/entrega.



Los atributos pueden entrar en conflictos unos con otros.

- Seguridad vs Confiabilidad
- Portabilidad vs Performance
- Seguridad vs Usabilidad
- Performance vs Modificabilidad

Hay que consensuar una priorización de los atributos de calidad con los intereses involucrados para lograr diseñar un sistema lo suficientemente bueno para todos los interesados.

Disponibilidad - Availability

Es la capacidad del sistema para estar en funcionamiento y ser accesible cuando se necesita. Se mide generalmente en términos de porcentaje, que representa la proporción de tiempo en que el sistema está operativo en relación al tiempo total.

Performance

Corresponde a los tiempos de respuesta de la aplicación en relación a las funcionalidades o actividades soportadas por la misma. Hay dos formas de medir el rendimiento de una aplicación:

- Latencia: tiempo dedicado a responder un evento.
- Capacidad: el número de eventos que pueden ocurrir en un tiempo determinado.

Interoperabilidad

Mide la capacidad de intercambio de información de la aplicación con otros sistemas o con el entorno donde opera. Una aplicación bien diseñada facilita la integración con otros sistemas.

Usabilidad

Se ve a través de:

- Comprensibilidad: facilidad del usuario de comprender el sistema.
- Fácil uso.
- Intuitiva.
- Comodidad.

Seguridad

Permite medir la vulnerabilidad de las aplicaciones a ataques accidentales o maliciosos, la posibilidad de defensa del sistema ante pérdidas o robo de información.

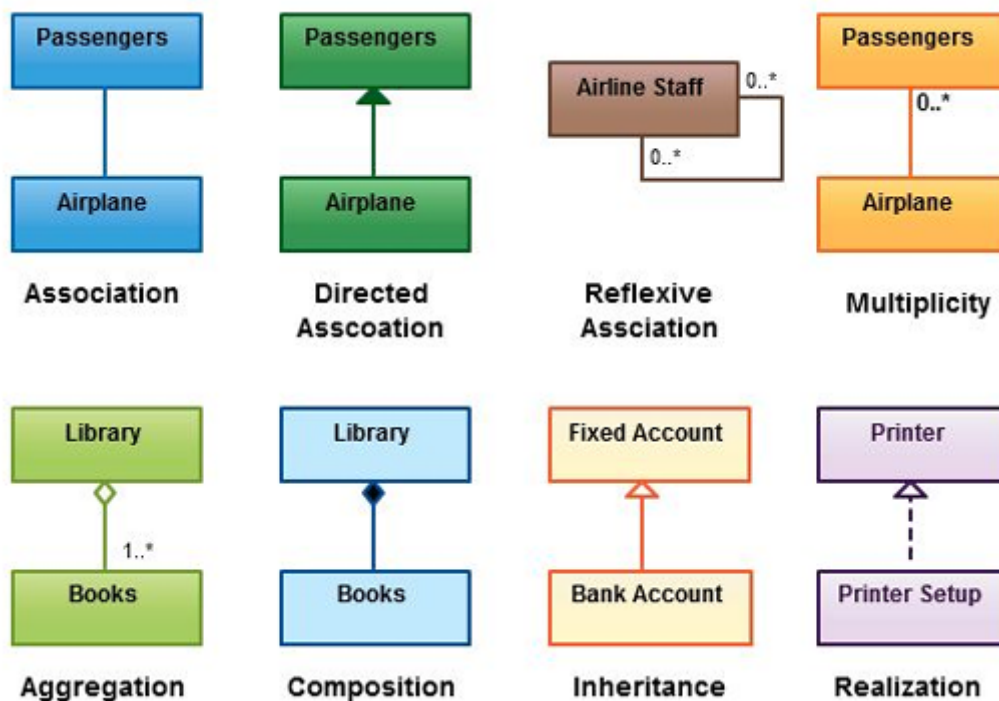
- Autenticación.
- Autorización.
- Encriptación de datos.

Escalabilidad

INTRODUCCIÓN A DIAGRAMAS UML

El lenguaje unificado de modelado o UML es una forma de representar visualmente la arquitectura y diseño de sistemas de software complejos. Facilita la comprensión de ideas y sistemas complejos mediante la visualización.

Relaciones



Asociación

La clase "A" tiene a la clase "B" como parámetro.

Inheritance/Generalización

La clase "B" hereda las características de la clase "A".

Realización

La clase "A" implementa una interfaz o clase abstracta.

Dependencia

Asociación más débil.

Agregación

Son independientes, pero conviven juntos, ambos interaccionan y se envían mensajes.

Composición

Un objeto compone a otro y tienen el mismo ciclo de vida.