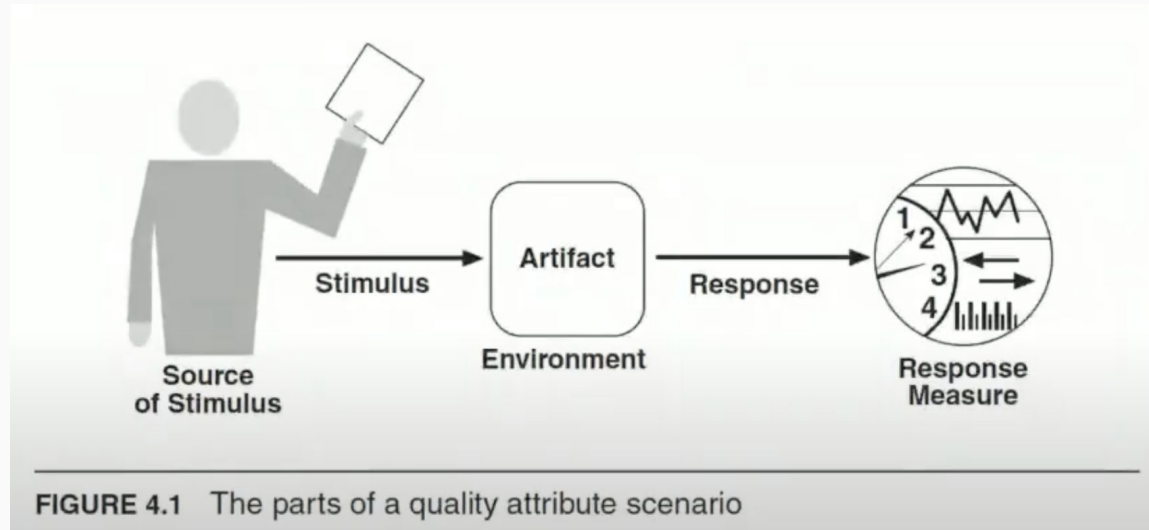


# Atributos de Calidad

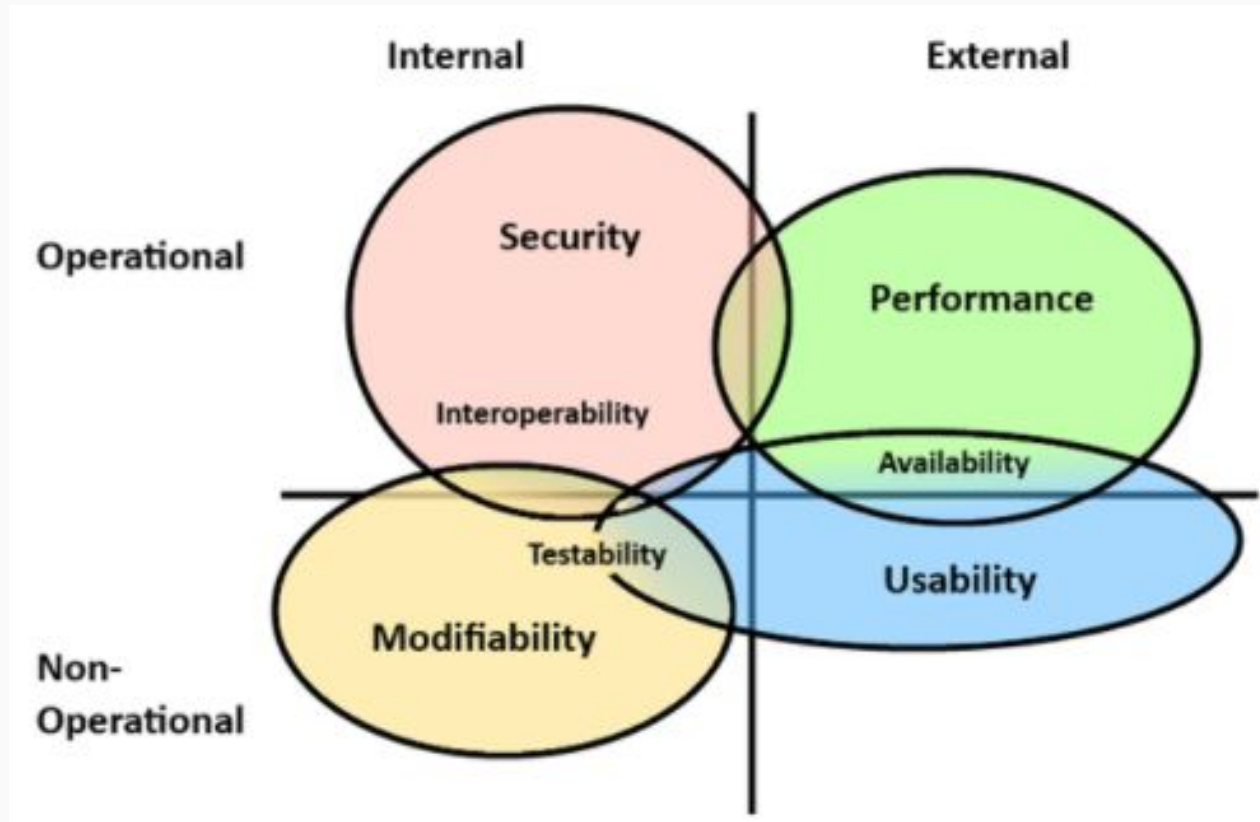
FIUBA

# Definición

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



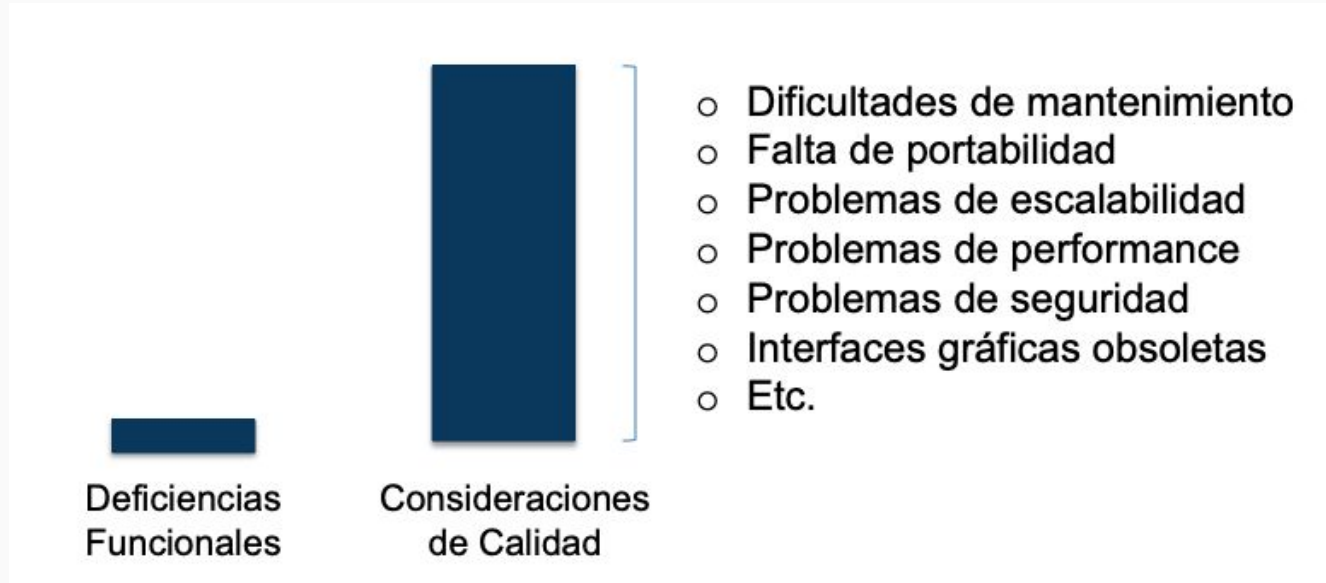
# Clasificación



Otra clasificación puede ser los atributos de calidad definidos en la norma ISO/IEC FCD 25010:

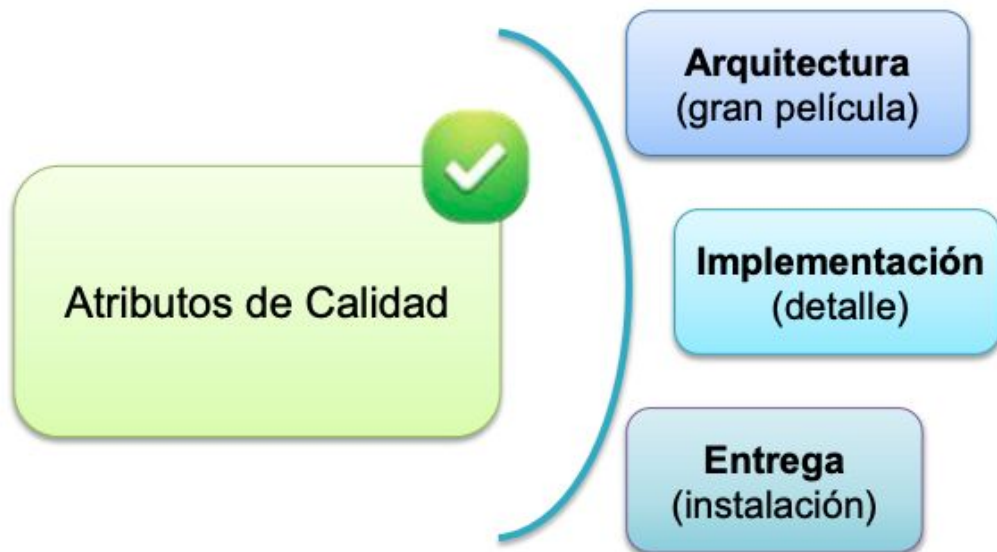


# Por que se rediseñan los sistemas?



Las consideraciones de calidad se desprenden de las necesidades del negocio y deben jugar un rol fundamental durante el ciclo de vida del desarrollo de software. No todo es funcionalidad en un sistema.

- 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 (y/o arquitectura)
  - ✓ No dependen solamente de la implementación o entrega



# Atributos de Calidad

Disponibilidad

Reusabilidad

Performance

Robustez

Flexibilidad

Testeabilidad

Interoperabilidad

Usabilidad

Mantenibilidad

Integridad

Portabilidad

Confiabilidad

Etc

Los atributos de calidad pueden entrar en conflicto unos con otros.

- Seguridad vs confiabilidad
- Portabilidad vs performance
- Seguridad vs usabilidad
- Performance vs modificabilidad



El objetivo es evaluar cuantitativa y/o cualitativamente múltiples atributos de calidad.

- Consensuar una priorización de los atributos de calidad con los interesados involucrados
- Diseñar un sistema lo suficientemente bueno para todos los interesados



- ¿Qué haría fallar al sistema?
- ¿Qué tan probable es que ocurra?
- ¿Cuánto tiempo lleva repararlo?

La disponibilidad es un importante atributo de calidad de software que se refiere a la capacidad del sistema para estar en funcionamiento y accesible cuando se le necesita. En otras palabras, se trata de la capacidad de un sistema o aplicación de estar disponible y operativo para los usuarios durante un período de tiempo especificado, generalmente medido en términos de horas, días o años. La disponibilidad es fundamental en una amplia variedad de sistemas de software, desde aplicaciones web y móviles hasta sistemas críticos como la infraestructura de servicios públicos y la atención médica. A continuación, se describen los aspectos clave de la disponibilidad:

Medición de la Disponibilidad: La disponibilidad se mide generalmente en términos de un porcentaje, que representa la proporción de tiempo en el que el sistema está operativo en relación con el tiempo total. Por ejemplo, si un sistema está en funcionamiento durante 364 días al año, su disponibilidad sería del 99.73% (esto se calcula como  $(364 / 365) * 100\%$ ).

Factores Clave: La disponibilidad del software se ve influenciada por varios factores, que incluyen:

- Diseño Robusto:
- Gestión de Recursos
- Mantenimiento y Actualizaciones
- Supervisión y Detección de Fallos
- Recuperación ante Desastres:

## Importancia de la Disponibilidad:

- Negocios
- Usuarios Finales
- Seguridad

## Ejemplos de Estrategias para Mejorar la Disponibilidad:

- Balanceo de carga
- Replicación de Datos
- Respaldo y Recuperación
- Escalabilidad

La disponibilidad es esencial para garantizar que los sistemas de software cumplan con las expectativas de los usuarios y las necesidades del negocio. Por lo tanto, los ingenieros de software deben diseñar y desarrollar aplicaciones teniendo en cuenta la disponibilidad desde el principio, implementando estrategias para mitigar fallos y garantizar la continuidad del servicio.

# Performance

Corresponde a los tiempos de respuesta de la aplicación en relación a las funcionalidades o actividades soportadas por la misma.

Se consideran dos formas principales para medir el rendimiento de una aplicación:

Latencia: Tiempo dedicado a responder a un evento.

Capacidad: El número de eventos que pueden ocurrir en un tiempo determinado.

# Interoperabilidad

Interoperabilidad es un atributo que 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.

Para mejorar la interoperabilidad, es conveniente utilizar interfaces externas bien diseñadas, normas de intercambio y estándares, entre otras.

# Usabilidad

La usabilidad se puede ver a través de:

- **Comprensibilidad:** este atributo refleja que tan fácil es para el usuario comprender el sistema, que conocimientos previos requiere el usuario para poder trabajar con el software
- **Fácil uso / Eficiente:** Que permita realizar las operación de manera rápida y efectiva
- **Fácil de Recordar / Intuitiva / Estandar**
- **Atractividad / Agradable / Cómoda**

Es el atributo que permite medir la vulnerabilidad de las aplicaciones a ataques accidentales o maliciosos, versus la posibilidad de defensa del sistema ante pérdidas o robo de información estratégica y valiosa para la organización.

Estos ataques pueden ser tanto externos como internos.

Autenticación, autorización, encriptación de datos, auditoría, entre otros.

- Capacidad de detección de ataques de denegación de servicio (DDoS), y respuesta ante éstos.
- Restricciones de acceso de usuarios de acuerdo a las políticas de autenticación y autorización.
- Prevención de la inyección de consultas SQL
- Encriptación de claves, contenidos y datos empresariales.
- Conexión segura

# Escalabilidad

Es la capacidad de manejar la carga de trabajo de la aplicación sin afectar el rendimiento de la misma, es la posibilidad de crecimiento sin perjudicar su funcionamiento operativo.

Como mejorar la escalabilidad:

- Vertical: Para crecer, se agregan más recursos físicos a la infraestructura que soporta al aplicativo, tales como memoria, almacenamiento en disco, procesador o capacidad de cómputo, ancho de banda, entre otras, para un aplicativo.
- Horizontal: Se incrementa el número de computadores para dividir la carga de trabajo de la aplicación.



Entre los indicadores claves para medir la escalabilidad se encuentran:

- Si el sistema permite el escalamiento vertical o distribución de la carga de trabajo en distintas computadoras.
- El tiempo necesario para aumentar el escalamiento.
- Las limitaciones de escalamiento en la infraestructura operativa: número de servidores máximo, memoria, discos, o capacidades de la red.
- Posibilidades de escalamiento: incremento en el número de transacciones o carga de trabajo.

Software Architecture in Practice (3rd Edition) by Len Bass (Author), Paul Clements (Author), Rick Kazman (Author)