

# REQUERIMIENTOS NO FUNCIONALES

por:

Jaider Fabian Molina Velez

Luis Fernando Rios Zapata

Professor

Robinson Coronado

Arquitectura de Software - 20201

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE SISTEMAS

2020

# Requerimientos No Funcionales

## Introducción

Cuando se inicia un nuevo proyecto para el desarrollo de una aplicación, la ingeniería de requerimientos es vital para identificar y clasificar todas las necesidades del negocio, de los clientes (internos y externos) así como restricciones o normas técnicas, jurídicas y de comportamiento. Esas restricciones o normas se llaman Requerimientos No Funcionales y algunas de ellas se pueden resolver con patrones de diseño y/o frameworks dentro de una arquitectura de software, a continuación veremos la definición de algunas y el ejemplo práctico de la Auditabilidad.

## Objetivo General

Identificar los RNF más usuales y corroborar que patrones de diseño o frameworks lo pueden resolver.

## Requerimientos No Funcionales

Los requisitos no funcionales son aquellos aspectos del sistema de TI que, si bien no afectan directamente la funcionalidad comercial de la aplicación, tienen un impacto profundo en la eficiencia y efectividad de los sistemas comerciales para los usuarios finales, así como las personas responsables de respaldar el programa

La definición de estos requisitos es un factor esencial en el desarrollo de una solución total para el cliente que cumpla los objetivos comerciales. Los requisitos no funcionales (NFR) se utilizan principalmente para impulsar los aspectos operativos de la arquitectura; en otras palabras, para abordar las principales áreas operativas y técnicas del sistema para garantizar la solidez y solidez de la aplicación.

### Durabilidad

La durabilidad es la capacidad de un producto físico para permanecer funcional, sin requerir un mantenimiento o reparación excesivos , cuando se enfrenta a los desafíos del funcionamiento normal durante su vida útil de diseño . Existen varias medidas de durabilidad en uso, incluidos años de vida, horas de uso y número de ciclos operativos.

La durabilidad del producto se basa en una buena capacidad de reparación y regeneración junto con el mantenimiento . Todo producto duradero debe poder adaptarse a los desarrollos técnicos, tecnológicos y de diseño. Esto debe ir acompañado de la voluntad por parte de los consumidores de renunciar a tener la versión "más reciente" de un producto.

### Resiliencia red

la resiliencia es la capacidad de proporcionar y mantener un nivel de servicio aceptable frente a fallas y desafíos para el funcionamiento normal. Las amenazas y desafíos para los servicios pueden variar desde una simple mala configuración por desastres naturales a gran escala hasta ataques dirigidos. Como tal, la resiliencia de la red toca una amplia gama de temas. Para aumentar la resiliencia de una determinada red de comunicaciones, es necesario

identificar los probables desafíos y riesgos y definir las métricas de resiliencia adecuadas para proteger el servicio.

La importancia de la resiliencia de la red aumenta continuamente, ya que las redes de comunicación se están convirtiendo en un componente fundamental en el funcionamiento de las infraestructuras críticas. En consecuencia, los esfuerzos recientes se centran en interpretar y mejorar la resiliencia de la red y la informática con aplicaciones a infraestructuras críticas. Como ejemplo, se puede considerar como un objetivo de resiliencia el suministro de servicios a través de la red, en lugar de los servicios de la propia red. Esto puede requerir una respuesta coordinada tanto de la red como de los servicios que se ejecutan en la parte superior de la red.

## Legibilidad

La legibilidad se refiere a la facilidad con la que un lector humano puede comprender el propósito, el flujo de control y el funcionamiento del código fuente. Afecta los aspectos de calidad, incluida la portabilidad, la usabilidad y, lo más importante, la capacidad de mantenimiento.

La legibilidad es importante porque los programadores pasan la mayor parte de su tiempo leyendo, tratando de comprender y modificar el código fuente existente, en lugar de escribir código fuente nuevo. El código ilegible a menudo conduce a errores, ineficiencias y código duplicado.

Seguir un estilo de programación coherente a menudo ayuda a mejorar la legibilidad. Sin embargo, la legibilidad es más que solo estilo de programación. Muchos factores, que tienen poco o nada que ver con la capacidad de la computadora para compilar y ejecutar el código de manera eficiente, contribuyen a la legibilidad. Algunos de estos factores incluyen:

- Diferentes estilos de sangría (espacios en blanco)
- Comentarios
- Descomposición
- Convenciones de nomenclatura para objetos (como variables, clases, procedimientos, etc.)

## Extensibilidad

La extensibilidad es un principio de diseño de sistemas e ingeniería de software en el que la implementación tiene en cuenta el crecimiento futuro. El término extensibilidad también

puede verse como una medida sistémica de la capacidad de extender un sistema y el nivel de esfuerzo requerido para implementar la extensión. Las extensiones pueden ser mediante la adición de nueva funcionalidad o mediante la modificación de la funcionalidad existente. El tema central es proporcionar cambios, generalmente mejoras, mientras se minimiza el impacto en las funciones existentes del sistema.

Un sistema extensible es aquel cuya estructura interna y flujo de datos se ven mínimamente o no afectados por una funcionalidad nueva o modificada, por ejemplo, volver a compilar o cambiar el código fuente original puede ser innecesario al cambiar el comportamiento de un sistema, ya sea por el creador o por otros programadores. Debido a que los sistemas de software son de larga duración y se modificarán para incorporar nuevas características y funcionalidades adicionales que demanden los usuarios, la extensibilidad permite a los desarrolladores expandir o agregar capacidades del software y facilita la reutilización sistemática. Algunos de sus enfoques incluyen facilidades para permitir que se inserten las rutinas del programa de los usuarios y la capacidad de definir nuevos tipos de datos, así como para definir nuevas etiquetas de marcado de formato.

La extensibilidad es la capacidad de un sistema para adaptarse a cambios futuros a través de una arquitectura, diseño o implementación flexibles.

Las aplicaciones extensibles tienen una resistencia excelente, lo que evita los costosos procesos de adquirir grandes aplicaciones inflexibles y volver a organizarlas debido a cambios en las necesidades comerciales. La extensibilidad permite a las organizaciones aprovechar las oportunidades y responder a los riesgos y, si bien existe una diferencia significativa, la extensibilidad a menudo se mezcla con la modificabilidad. La modificabilidad significa que es posible cambiar el software, mientras que la extensibilidad significa que el cambio se ha planificado y no tendrá ningún esfuerzo. En ocasiones, la adaptabilidad se aprovecha erróneamente de la extensibilidad. Sin embargo, la adaptabilidad se ocupa de cómo se gestionan y gobiernan las interacciones del usuario con el sistema.

La extensibilidad permite que un sistema, personas, tecnología, información y procesos trabajen juntos para lograr los siguientes atributos:

Manejar nuevos tipos de información

Administrar entidades comerciales nuevas o modificadas

Consumir o proporcionar nuevos feeds

Atributo NFR Valor objetivo Opciones de extensión La separación lógica de la aplicación en diferentes niveles (cliente, presentación, lógica de negocios, integración y nivel EIS) permite que un sistema sea flexible y fácil de mantener Aprovechando los patrones de diseño en toda la arquitectura de la aplicación Orientación del objeto como encapsulación, herencia, acoplamiento bajo y una alta cohesión se aprovechan en el diseño de la aplicación Está respaldado por la documentación adecuada de la aplicación Independencia de la interfaz de la implementación.

## Recuperación

En el caso de una calamidad natural, por ejemplo, una inundación o un huracán, toda la instalación donde está alojada la aplicación puede volverse inoperable o inaccesible. Las aplicaciones críticas para el negocio deben tener una estrategia para recuperarse de tales desastres dentro de un período de tiempo razonable. La solución que implementa varios procesos debe integrarse con el plan de recuperación de desastres de la empresa existente. Los procesos deben analizarse para comprender la criticidad de cada proceso para el negocio, el impacto de la pérdida para el negocio en caso de no disponibilidad del proceso. Sobre la base de este análisis, se deben desarrollar procedimientos de desastre apropiados y se deben delinear planes. Como parte de la recuperación ante desastres, las copias de seguridad electrónicas de los datos y los procedimientos deben mantenerse en el lugar de recuperación y ser recuperables dentro de los plazos adecuados para la restauración del funcionamiento del sistema. En el caso de alta criticidad, se debe implementar la replicación en tiempo real en un sitio espejo.

Los atributos son:

- **Proceso de recuperación:** Objetivos de tiempo de recuperación (RTO) y objetivos de punto de recuperación (RPO)
- **Restaurar tiempo:** Tiempo requerido para intercambiar al respaldo si falla el primario.
- **RPO/tiempo de respaldo:** Tiempo que toma realizar un respaldo de la información.
- **Backup frecuencias:** Frecuencia de respaldo de los datos transaccionales, configuración de datos y código.

## Portabilidad

La portabilidad es la facilidad con la que un sistema de software se puede transferir desde su entorno actual de hardware o software a otro entorno. Los requisitos de portabilidad abordan la preocupación del usuario sobre la facilidad con que se transporta el sistema. Al obtener los requisitos de portabilidad, tenga en cuenta los aspectos de la portabilidad con respecto a los datos, el programa, el usuario final y la documentación del desarrollador.

EJEMPLO: [Independencia del dispositivo] Los estudiantes verán videos de lecciones de cursos a pedido desde múltiples sistemas operativos, incluidos Microsoft Windows, macOS y Android.

## Auditabilidad

La auditabilidad es el grado en que se pueden rastrear las transacciones, desde el originador hasta el aprobador y la disposición final, a través de un sistema por un auditor. Parte de la auditabilidad proviene de la documentación del sistema y parte proviene de la visibilidad de todas las modificaciones relacionadas con la integridad del sistema y de los registros de datos. Los registros deben permitirle al auditor determinar quién hizo qué y cuándo. Los auditores pueden ser internos o externos a la organización. Una vez que haya identificado quiénes serán los auditores de este sistema, trabaje con ellos para asegurarse de que satisfaga sus necesidades. Naturalmente, querrán validar y documentar que lo hace.

Por ejemplo, en un sistema de vinculación de clientes con crédito se necesita saber quien y cuando se registran los clientes, inicialmente en un estado “en estudio” y también se necesita saber quien y cuando los “aprueba”. El framework de java Spring permite implementar módulos para hacer auditoria a estas transacciones a través de ciertas clase de JPA Entity, entre ellas: `AuditingEntityListener`, `data.annotation`, `AuditorAware` y `@EnableJpaAuditing`.

En el repositorio

<https://github.com/lfernandorios/arquisoft-20201/tree/develop/sources/nfr-auditability> hay una implementación de estas librerías que se pueden extender a nuevas Entities y registrar la auditoria en cada de las operaciones CUD.