

# Ejemplos de requerimientos no funcionales de producto

## Eficiencia

- El sistema debe ser capaz de procesar N transacciones por segundo. Esto se medirá por medio de la herramienta **SoapUI aplicada al Software Testing de servicios web**.
- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

## Seguridad lógica y de datos

- Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
- El nuevo sistema debe desarrollarse aplicando **patrones y recomendaciones de programación que incrementen la seguridad de datos**.
- Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema.
- Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA.
- Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad.

## Seguridad industrial

- El sistema no continuará operando si la temperatura externa es menor a 4 grados Celsius.
- El sistema no continuará operando en caso de fuego. (Ej. Un ascensor).

## Usabilidad

- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
- La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.
- El sistema debe contar con manuales de usuario estructurados adecuadamente.
- El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
- El sistema debe contar con un módulo de ayuda en línea.
- La aplicación web debe poseer un **diseño “Responsive”** a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
- El sistema debe poseer interfaces gráficas bien formadas.

## Dependibilidad

- El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
- El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
- La tasa de tiempos de falla del sistema no podrá ser mayor al 0,5% del tiempo de operación total.
- El promedio de duración de fallas no podrá ser mayor a 15 minutos.
- La probabilidad de falla del Sistema no podrá ser mayor a 0,05.

## Otros ejemplos de requerimientos de producto

- El sistema será desarrollado para las plataformas PC y Macintosh.
- La aplicación debe ser compatible con todas las versiones de Windows, desde Windows 95.
- La aplicación deberá consumir menos de 500 Mb de memoria RAM.
- La aplicación no podrá ocupar más de 2 GB de espacio en disco.
- La nueva aplicación debe manejar fuentes del alfabeto en Inglés, Idiomas latinos (Español, Frances, Portugués, Italiano), Árabe y Chino.
- La interfaz de usuario será implementada para navegadores web únicamente con HTML5 y JavaScript.

## Ejemplos de requerimientos no funcionales organizacionales

- El procedimiento de desarrollo de software a usar debe estar definido explícitamente (en manuales de procedimientos) y debe cumplir con los estándares ISO 9000.
- La metodología de desarrollo de software será **Behaviour Driven Development (BDD) apoyada en Cucumber**.
- El sistema debe ser desarrollado utilizando las herramientas CASE XYZ.
- El proceso de desarrollo se gestionará por medio de una determinada **herramienta web para gestionar el proceso de desarrollo de software**.
- Debe especificarse un plan de recuperación ante desastres para el sistema a ser desarrollado.
- Cada dos semanas deberán producirse reportes gerenciales en los cuales se muestre el esfuerzo invertido en cada uno de los componentes del nuevo sistema.
- Las pruebas de software se gestionaran con una **herramienta de gestión de software testing**.
- Las pruebas de software se ejecutarán utilizando **Selenium y Ruby** como herramienta y lenguaje Scripting para automatización de software testing.

## Ejemplos de requerimientos no funcionales externos

- Sistemas de datos médicos: El nuevo sistema y sus procedimientos de mantenimiento de datos deben cumplir con las leyes y reglamentos de protección de datos médicos.
- El nuevo sistema se acogerá a las reglas de las licencias generales públicas (GNU), es decir será gratuito, código abierto en el que cualquiera podrá cambiar el software, sin patentes y sin garantías.
- Las páginas web a ser desarrolladas deben cumplir con la ley de tratamiento en condiciones de igualdad para personas con discapacidad.
- El sistema no revelará a sus operadores otros datos personales de los clientes distintos a nombres y números de referencia.

## Requerimientos no funcionales y requerimientos funcionales

Los requerimientos no funcionales suelen expresarse de una manera general y sin hacer referencia a algún módulo, transacción o característica del sistema, pues sino pasarían a ser **requerimientos funcionales**.

Por ejemplo:

*El sistema debe asegurar que los datos estén protegidos del acceso no autorizado*

Es recomendable acompañar las definiciones de requerimientos no funcionales con criterios de aceptación que puedan medirse.

Los requerimientos no funcionales pueden a su vez derivar en requerimientos funcionales, tomando como ejemplo el anterior:

*El sistema incluirá un procedimiento de autorización de usuarios, en el cual los usuarios deben identificarse usando un nombre de usuario y contraseña. Sólo los usuarios autorizados de esta forma podrán acceder a los datos del sistema.*

Escrito de esta forma, el requerimiento pasa a ser funcional.

Sin embargo, no todos los requerimientos no funcionales pueden derivarse en requerimientos funcionales, por lo cual es muy importante definir los criterios de aceptación.